ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Η ΜΕΘΟΔΟΣ LASSO ΣΤΗ ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ ΚΑΙ ΓΕΝΙΚΕΥΣΕΙΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΓΓΕΛΑΚΟΠΟΥΛΟΣ ΧΑΡΑΛΑΜΠΟΣ

ΕΠΙΒΛΕΠΩΝ:
ΔΗΜΗΤΡΙΟΣ ΦΟΥΣΚΑΚΗΣ
Αν. Καθηγητής Ε.Μ.Π

ΑΘΗΝΑ, Σεπτέμβριος 2020

Στους Γονείς μου

Euqaristðec

Θα ήθελα να ευχαριστήσω τον Επιβλέποντα Καθηγητή της Διπλωματικής μου εργασίας, κύριο Δημήτρη Φουσκάκη, για όλη τη βοήθεια και την καθοδήγησή του κατά τη διάρκεια συγγραφής της. Επίσης τον ευχαριστώ ιδιαίτερα, διότι μέσα από τα μαθήματά του με βοήθησε να ξαναβρώ το πάθος και την αγάπη μου για τα Μαθηματικά. Επιπλέον θα ήθελα να ευχαριστήσω την οικογένειά μου και κυρίως τους γονείς μου Παναγιώτη και Βασιλική, οι οποίοι με στηρίζουν σε όλες μου τις επιλογές.

Perlhyh

Η παρούσα διπλωματική εργασία αφορά τη μέθοδο Lasso στη γραμμική παλινδρόμηση καθώς και ορισμένες γενικεύσεις αυτής. Στα προβλήματα που θα μελετήσουμε, θεωρούμε την εξάρτηση μιας μεταβλητής απόκρισης από κάποιες άλλες επεξηγηματικές μεταβλητές. Το γενικό πλαίσιο αφορά ένα δείγμα που αποτελείται από N παρατηρήσεις και p παράγοντες. Με βάση το δείγμα αυτό και με χρήση ορισμένων μεθόδων, θέλουμε να κατασκευάσουμε ένα γραμμικό μοντέλο που θα περιγράφει τη σχέση μεταξύ της μεταβλητής απόκρισης και των επεξηγηματικών μεταβλητών. Το κριτήριο για την επιλογή του «βέλτιστου» μοντέλου συχνά είναι υποκειμενικό. Σίγουρα όμως μας ενδιαφέρει το μοντέλο μας να έχει όσο το δυνατόν υψηλότερη ακρίβεια στις προβλέψεις για την εξαρτημένη μεταβλητή, κυρίως για δεδομένα που δεν ανήκουν στο σύνολο που διαθέτουμε (π.χ. μελλοντικά δεδομένα). Επίσης, είναι προτιμότερο το μοντέλο που θα επιλέξουμε να είναι όσο το δυνατόν πιο φειδωλό, δηλαδή να περιέχει μόνο τις μεταβλητές εκείνες που συνεισφέρουν πραγματικά στην ερμηνεία της μεταβλητής που μελετάμε. Επομένως, πιο απλά μοντέλα μας βοηθούν στην καλύτερη ερμηνεία και σε μείωση του κόστους και του χρόνου, εφόσον δε χρειάζεται να κάνουμε περιττές μετρήσεις για επιπλέον παράγοντες που δε συνεισφέρουν στο μοντέλο. Ειδικότερα σε προβλήματα όπου ο αριθμός των παραγόντων είναι αρκετά μεγαλύτερος από το πλήθος των παρατηρήσεων που διαθέτουμε (large p, small N problems), θεωρούμε ότι σημαντικό ρόλο θα παίζουν μόνο ορισμένοι παράγοντες. Επομένως είναι επιτακτική η ανάγκη για τη χρήση μεθόδων που παράγουν αραιά (sparse) αλλά ταυτόχρονα και ακριβή μοντέλα. Οι μέθοδοι συρρίκνωσης χρησιμοποιούνται ευρέως σε τέτοιου είδους προβλήματα και η μέθοδος Lasso είναι μία από τις κυριότερες.
Στο πρώτο Κεφάλαιο της εργασίας αυτής θα αναλύσουμε τη μέθοδο Lasso για τα γραμμικά μοντέλα και θα δούμε πως αυτή μπορεί να χρησιμοποιηθεί ως μια εναλλακτική προσέγγιση της μεθόδου ελαχίστων τετραγώνων, στο πρόβλημα προσαρμογής ενός γραμμικού μοντέλου. Επίσης, από τη σύγκριση με τη μέθοδο Ridge, θα δούμε γιατί η μέθοδος Lasso έχει την ιδιότητα να παράγει αραιά μοντέλα. Στο Κεφάλαιο 2 παρουσιάζουμε κάποιες γενικεύσεις και επεκτάσεις της μεθόδου Lasso, όπως είναι οι Elastic Net, Group Lasso κ.α. Αυτές οι μέθοδοι βελτιώνουν τη μέθοδο Lasso, σε περιπτώσεις όπου υπάρχει υψηλή συσχέτιση μεταξύ των επεξηγηματικών μεταβλητών (φαινόμενο πολυσυγγραμμικότητας) ή όταν αυτές μπορούν με κάποιο τρόπο να δομηθούν σε ομάδες (π.χ κατηγορικές μεταβλητές). Στο Κεφάλαιο 3 κάνουμε μια εισαγωγή στη στατιστική συμπερασματολογία για τις εκτιμήτριες που προκύπτουν με βάση τις μεθόδους συρρίκνωσης που έχουμε αναπτύξει. Θεωρούμε την Μπεϋζιανή προσέγγιση των μεθόδων Lasso και Ridge. Επίσης, εξετάζουμε πως με τη χρήση μεθόδων επαναδειγματοληψίας, όπως η Bootstrap, μπορούμε να εξάγουμε συμπεράσματα για τις εκτιμήτριες του μοντέλου μας. Στο τέταρτο και τελευταίο Κεφάλαιο εφαρμόζουμε ορισμένες από τις τεχνικές συρρίκνωσης, πάνω σε ένα πραγματικό σύνολο δεδομένων. Κατασκευάζουμε ένα γραμμικό μοντέλο με σκοπό την πρόβλεψη του αριθμού θανάτων που οφείλονται στη νόσο του καρκίνου σε διάφορες κομητείες των Η.Π.Α. Επίσης, αξιολογούμε το μοντέλο μας χρησιμοποιώντας τεχνικές όπως Cross Validation και τέλος παρουσιάζουμε κάποια συμπεράσματα. Μεγάλο μέρος της εργασίας περιλαμβάνει εφαρμογές, όπως προσομοιώσεις μαζί με αντίστοιχα διαγράμματα

και σχήματα. Για όλες τις εφαρμογές έγινε χρήση του στατιστικού πακέτου R και όλοι οι κώδικες και τα διαγράμματα βρίσκονται στις αντίστοιχες ενότητες.

Summary

The present thesis deals with the Lasso method in linear regression, as well as some of its generalizations. In the problems that we study, we consider the dependence of a response variable on some other explanatory variables. The general setup refers to a sample consisting of N observations and p factors. Based on this sample and the use of certain methods, we want to construct a linear model that describes the relationship between the response variable and the explanatory variables. The criterion for choosing the \optimal" model is often subjective. But we certainly want our model to have the best possible accuracy in predicting the dependent variable, especially for data that does not belong to the set that we have (e.g. future data). Also, it is preferable for the model we choose to be as sparse as possible, that is to contain only those variables that really contribute to the interpretation of the variable we are studying. Therefore, simpler models help us better interpret and reduce costs and time, as long as we do not need to make unnecessary measurements for additional factors that do not contribute to the model. Particularly, in problems where the number of factors is much larger than the number of observations we have (large p, small N problems), we believe that only certain factors will play an important role. Therefore, the need for the use of methods that produce sparse and also accurate models, is imperative. Shrinkage methods are widely used in such problems and the Lasso is one of the main ones.

In the first Chapter of this thesis, we will analyze the Lasso method for linear models and we will see how it can be used as an alternative approach to the least squares method, in the problem of fitting a linear model. Also, from the comparison with Ridge regression we will see why Lasso has the property of producing sparse models. In Chapter 2 we present some generalizations and extensions of the Lasso, such as Elastic Net, Group Lasso etc. These methods improve the Lasso in cases where there is a high correlation between the explanatory variables (multicollinearity issues) or they can somehow be structured into groups (e.g. categorical variables). In Chapter 3 we make an introduction to statistical inference for the estimators that result on the shrinkage methods we have developed. We consider the Bayesian approach to Lasso and Ridge. We also look at how we can draw conclusions about our model's estimates by using re-sampling methods, such as Bootstrap. In the fourth and last Chapter we apply some of the shrinkage techniques, to a real data set. We are building a linear model to predict the number of deaths due to cancer in various U.S counties. We also evaluate our model using techniques such as Cross Validation and finally present some conclusions. Much of this thesis includes simulations along with corresponding diagrams. For all the applications we used the R statistical software and all the codes and diagrams are in the respective sections.

Perieqìmena

Kefˆlaio 1

H mèjodoc Lasso sta grammikˆ montèla

1.1 Εισαγωγή

Θεωρούμε το γενικό γραμμικό μοντέλο με μεταβλητή απόκρισης $\mathbf{y}$ 2 R^N και p επεξηγηματικές μεταβλητές $X_1, \ldots, X_p$. Η σχέση που περιγράφει τη γραμμική εξάρτηση μεταξύ της μεταβλητής απόκρισης και των επεξηγηματικών μεταβλητών είναι η εξής:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1.1}$$

όπου $\mathbf{X}$ 2 $\mathsf{R}^{N \ (p+1)}$ ο πίνακας σχεδιασμού, $\boldsymbol{\beta}$ 2 R^{p+1} το διάνυσμα των συντελεστών και $\boldsymbol{\varepsilon}$ 2 R^N το σφάλμα στο μοντέλο, με μέση τιμή $\mathsf{E}(\boldsymbol{\varepsilon}) = \mathbf{0}$ και πίνακα διασποράς $\mathbf{V}(\boldsymbol{\varepsilon}) = \sigma^2 \boldsymbol{I}_N$, με $\boldsymbol{I}_N$ τον μοναδιαίο πίνακα. Επίσης, υποθέτουμε ότι υπάρχει ανεξαρτησία των σφαλμάτων και ότι ακολουθούν κανονική κατανομή $N(0, \sigma^2)$. Έστω ότι διαθέτουμε ένα τυχαίο δείγμα N παρατηρήσεων $f(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)g$, με $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, N$ είναι η i-οστή γραμμή του πίνακα $\mathbf{X}$. Θα χρησιμοποιήσουμε τις παρατηρήσεις αυτές προκειμένου να προσαρμόσουμε το μοντέλο. Σκοπός μας δηλαδή είναι να εκτιμήσουμε τις παραμέτρους $\beta_0, \beta_1, \ldots, \beta_p$ και εν συνεχεία να κάνουμε προβλέψεις για τη μεταβλητή απόκρισης $\mathbf{y}$. Η πιο συνηθισμένη μέθοδος για την προσαρμογή ενός γραμμικού μοντέλου, είναι η μέθοδος ελαχίστων τετραγώνων. Μέσω αυτής ελαχιστοποιούμε την παράσταση:

$$S(\boldsymbol{\beta}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i \quad \beta_0 \quad \sum_{j=1}^{p} \beta_j x_{ij} \right)^2, \tag{1.1.2}$$

δηλαδή το άθροισμα των τετραγώνων των υπολοίπων. Ο παράγοντας $1/2N$ χρησιμοποιείται ώστε να γίνουν οι σχετικές απλοποιήσεις στις πράξεις για τον υπολογισμό της εκτιμήτριας. Με άλλα λόγια ψάχνουμε τους συντελεστές εκείνους για τους οποίους οι προβλεπόμενες τιμές για την μεταβλητή απόκρισης είναι όσο το δυνατόν πιο κοντά στις τιμές που έχουμε παρατηρήσει γι΄ αυτή. Παραγωγίζοντας την παράσταση (1.1.2) ως προς $\boldsymbol{\beta}$ και θέτοντας την παράγωγο ίση με το 0, καταλήγουμε στην εκτιμήτρια ελαχίστων τετραγώνων, η οποία δίνεται από τον τύπο:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X}) \ {}^1\mathbf{X}^T\mathbf{y}. \tag{1.1.3}$$

Για να υπάρχει η εκτιμήτρια αυτή θα πρέπει ο πίνακας $\mathbf{X}^T\mathbf{X}$ να έχει μη μηδενική ορίζουσα και άρα να είναι αντιστρέψιμος. Η μέθοδος ελαχίστων τετραγώνων αν και απλή στην εφαρμογή της,

έχει ορισμένα μειονεκτήματα-ελλείψεις. Στην περίπτωση όπου ο αριθμός των παραγόντων p είναι μεγαλύτερος του πλήθους των παρατηρήσεων N που διαθέτουμε, τότε η εκτιμήτρια που προκύπτει από τη μέθοδο ελαχίστων τετραγώνων δεν είναι μοναδική, εφόσον το $N \times p$ σύστημα για τον υπολογισμό της είναι αόριστο. Επίσης, θα θέλαμε το τελικό μας μοντέλο να περιέχει μόνο τις μεταβλητές εκείνες που σχετίζονται άμεσα με την εξαρτημένη μεταβλητή. Με τη μέθοδο ελαχίστων τετραγώνων συχνά καταλήγουμε σε ένα μοντέλο που περιέχει μεταβλητές οι οποίες δε συνεισφέρουν στην επεξήγηση της μεταβλητής απόκρισης. Αντιθέτως οδηγούν σε πολυπλοκότητα με αποτέλεσμα να δυσχεραίνουν την ερμηνεία του μοντέλου μας. Ένας δεύτερος λόγος έχει να κάνει με την ακρίβεια και την προβλεπτική ικανότητα του μοντέλου μας. Μας ενδιαφέρει να έχουμε ακρίβεια στις προβλέψεις μας, όσον αφορά μελλοντικά δεδομένα ή παρατηρήσεις που δεν περιέχονται στο αρχικά διαθέσιμο σύνολο. Δεδομένου ότι το γραμμικό μοντέλο περιγράφει κατάλληλα τη σχέση μεταξύ εξαρτημένης μεταβλητής και επεξηγηματικών μεταβλητών, οι εκτιμήτριες ελαχίστων τετραγώνων θα είναι αμερόληπτες (unbiased), αλλά γενικά θα έχουν μεγάλη διασπορά και άρα μεγάλα τυπικά σφάλματα. Κατά συνέπεια το μοντέλο που προκύπτει σε αυτήν την περίπτωση δεν είναι κατάλληλο, με την έννοια ότι το σφάλμα των προβλέψεών μας για την εξαρτημένη μεταβλητή θα είναι μεγάλο, εφόσον οι εκτιμήτριές μας δεν είναι αξιόπιστες. Θα πρέπει λοιπόν με κάποιο τρόπο να εξισορροπήσουμε τη σχέση μεταξύ μεροληψίας και διασποράς των εκτιμητριών, το λεγόμενο *Bias-Variance trade o* . Για τους λόγους λοιπόν που αναφέραμε, είναι πολλές φορές ορθότερο να χρησιμοποιήσουμε κάποια εναλλακτική μέθοδο για την προσαρμογή ενός γραμμικού μοντέλου παλινδρόμησης, από αυτή της μεθόδου των ελαχίστων τετραγώνων.

1.2 Η εκτιμήτρια Lasso

Η μέθοδος Lasso[1] (Tibshirani, 1996) αποτελεί μια εναλλακτική μέθοδο από αυτή των ελαχίστων τετραγώνων, για την προσαρμογή ενός γραμμικού μοντέλου. Η χρήση της συνίσταται κυρίως σε περιπτώσεις όπου ο αριθμός των επεξηγηματικών μεταβλητών είναι μεγαλύτερος του αριθμού των διαθέσιμων παρατηρήσεων ($p > N$). Το σημαντικότερο πλεονέκτημα της μεθόδου αυτής, είναι η ικανότητά της να παράγει αραιά μοντέλα, δηλαδή μοντέλα τα οποία περιέχουν μόνο ένα υποσύνολο παραγόντων από τους αρχικά διαθέσιμους. Αυτό οδηγεί σε ευκολότερη ερμηνεία και μείωση της πολυπλοκότητας στο τελικό μοντέλο, αφού τελικά καταλήγουμε μόνο με τις μεταβλητές εκείνες που παίζουν σημαντικό ρόλο στην επεξήγηση της μεταβλητής απόκρισης. Η εκτιμήτρια Lasso δίνει λύση στο εξής πρόβλημα ελαχιστοποίησης:

$$\min_{2\mathrm{R}^p} \left\{ \frac{1}{2N} \sum_{i=1}^{N} \left(y_i \quad \beta_0 \quad \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \right\}, \qquad \mu\varepsilon \quad \sum_{j=1}^{p} j\beta_j j \quad s, \tag{1.2.1}$$

όπου το s 0, είναι ένα άνω φράγμα για το άθροισμα των απολύτων τιμών των συντελεστών του μοντέλου. Η παράσταση που ελαχιστοποιεί η εκτιμήτρια Lasso είναι ίδια με αυτή της μεθόδου των ελαχίστων τετραγώνων, αλλά τώρα θέτουμε και έναν επιπλέον περιορισμό στους συντελεστές. Έτσι λοιπόν, όταν το s είναι αρκετά μεγάλο δεν περιορίζουμε τους συντελεστές, καταλήγοντας έτσι στην εκτιμήτρια ελαχίστων τετραγώνων. Αντιθέτως όταν το s μειώνεται, τότε περιορίζουμε σε μεγαλύτερο βαθμό τους συντελεστές, με αποτέλεσμα αρκετοί από αυτούς να τείνουν προς το 0 και κάποιοι να είναι ακριβώς μηδενικοί. Η παράμετρος s συνήθως επιλέγεται μέσω μιας διαδικασίας που ονομάζεται *Cross Validation* (Hastie et al, 2001). Περιορίζοντας τους συντελεστές των

[1] proèrqetai apì ta arqikˆ Least Absolute Selection and Shrinkage Operator.

ανεξάρτητων μεταβλητών μπορούμε να πετύχουμε περισσότερη ακρίβεια προβλέψεων στο μοντέλο μας. Όπως αναφέραμε, οι εκτιμήτρες ελαχίστων τετραγώνων είναι αμερόληπτες αλλά έχουν μεγάλη διασπορά. Θέτοντας περιορισμό στις παραμέτρους εισάγουμε ένα μικρό ποσό μεροληψίας αλλά επιτυγχάνουμε σημαντική μείωση στη διασπορά τους. Έτσι το μέσο τετραγωνικό σφάλμα των εκτιμητριών, που αποτελείται από το άθροισμα της μεροληψίας στο τετράγωνο και της διασποράς,

$$MSE = bias^2 + Variance$$

συνήθως μειώνεται, με αποτέλεσμα να οδηγούμαστε σε καλύτερες προβλέψεις για τη μεταβλητή απόκρισης.
Είναι σημαντικό προτού προχωρήσουμε στον υπολογισμό της εκτιμήτριας να τυποποιήσουμε τις τιμές των ανεξάρτητων μεταβλητών, ώστε να έχουν μέση τιμή 0 και διασπορά ίση με τη μονάδα, δηλαδή

$$x_{ij} \quad \frac{x_{ij} \quad x_j}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} \quad x_j)^2}}, \quad x_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij},$$

για κάθε $j = 1, \ldots, p$ και $i = 1, \ldots, N$. Έτσι οι εκτιμήτριες Lasso που θα προκύψουν θα είναι ανεξάρτητες από τις μονάδες μέτρησης κάθε επεξηγηματικής μεταβλητής. Για λόγους απλότητας μπορούμε επίσης να κεντράρουμε και τη μεταβλητή απόκρισης y, ώστε $y = 0$. Έτσι μπορούμε να παραλείψουμε τη σταθερά β_0 από την ελαχιστοποίηση της παράστασης (1.2.1), διότι προκύπτει ότι

$$\hat{\beta_0} = y \quad \sum_{j=1}^{p} \hat{\beta_j} x_j \tag{1.2.2}$$

με y και x_j, $j = 1, \ldots, p$, οι μέσες τιμές υπολογισμένες στις αρχικές παρατηρήσεις (μη τυποποιημένες). Μια ισοδύναμη μορφή του προβλήματος (1.2.1) προκύπτει μέσω της λεγόμενης Λαγκρανζιανής δυϊκότητας (Lagrange duality). Άρα παραλείποντας τη σταθερά β_0, το πρόβλημα ελαχιστοποίησης μπορεί να γραφεί ισοδύναμα:

$$\min_{2R^p}\left\{\frac{1}{2N}\sum_{i=1}^{N}\left(y_i \quad \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\}, \quad \text{με } \lambda \quad 0, \tag{1.2.3}$$

όπου λ η παράμετρος συντονισμού (tuning parameter) ή παράμετρος ποινής (penalty parameter). Όπως και με την παράμετρο s, η παράμετρος λ καθορίζει το μέγεθος της ποινής που επιβάλλουμε στους συντελεστές. Όσο η τιμή του λ πλησιάζει το μηδέν, τόσο λιγότερο ποινικοποιούμε τους συντελεστές του μοντέλου, με αποτέλεσμα η εκτιμήτρια που προκύπτει να πλησιάζει την εκτιμήτρια ελαχίστων τετραγώνων. Αντιθέτως, όσο η τιμή του λ αυξάνει, τόσο περισσότεροι συντελεστές εκτιμούνται ως μηδενικοί και έτσι καταλήγουμε σε ένα αραιό μοντέλο. Παρόμοια συμπεράσματα ισχύουν και με τη χρήση του παράγοντα συρρίκνωσης (shrinkage factor), ο οποίος ισούται με την τιμή

$$\text{sf} = \frac{\|\beta\|_1}{\max\|\beta\|_1} \; 2 \; [0, 1].$$

Καθώς η l_1 νόρμα των συντελεστών πλησιάζει τη μέγιστη τιμή της, ισοδύναμα ο παράγοντας συρρίκνωσης τείνει προς τη μονάδα, τόσο η εκτιμήτρια Lasso τείνει προς την εκτιμήτρια ελαχίστων τετραγώνων. Αντιθέτως, όσο ο παράγοντας συρρίκνωσης πλησιάζει το 0, τόσο περιορίζεται η l_1 νόρμα των συντελεστών, άρα και περισσότεροι συντελεστές εκτιμούνται ως μηδενικοί.

Το πρόβλημα ελαχιστοποίησης μπορεί να γραφεί εναλλακτικά ως εξής:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\}, \qquad \lambda \geq 0, \tag{1.2.4}$$

αφού $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$. Η ισοδυναμία μέσω της Λαγκρανζιανής δυϊκότητας προκύπτει από το γεγονός ότι η συνάρτηση $f(\beta) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$ είναι κυρτή και το σύνολο $\{\beta \in \mathbb{R}^p : \|\beta\|_1 \leq s\}$ είναι επίσης κυρτό.

Απόδειξη. Έστω $\beta_1, \beta_2 \in \mathbb{R}^p$ και $t \in [0,1]$. Τότε

$$\begin{aligned} f(t\beta_1 + (1-t)\beta_2) &= \frac{1}{2N}\|\mathbf{y} - \mathbf{X}[t\beta_1 + (1-t)\beta_2]\|_2^2 \\ &= \frac{1}{2N}\|t(\mathbf{y} - \mathbf{X}\beta_1) + (1-t)(\mathbf{y} - \mathbf{X}\beta_2)\|_2^2 \\ &\leq \frac{1}{2N}\left(t\|\mathbf{y} - \mathbf{X}\beta_1\|_2 + (1-t)\|\mathbf{y} - \mathbf{X}\beta_2\|_2\right)^2 \\ &\leq \frac{1}{2N}\left(t\|\mathbf{y} - \mathbf{X}\beta_1\|_2^2 + (1-t)\|\mathbf{y} - \mathbf{X}\beta_2\|_2^2\right) \\ &= t f(\beta_1) + (1-t) f(\beta_2), \end{aligned}$$

όπου έγινε χρήση της τριγωνικής ανισότητας καθώς και ότι η συνάρτηση $g(x) = x^2, x \in \mathbb{R}$ είναι κυρτή. □

Εύκολα προκύπτει ότι και το σύνολο $\{\beta \in \mathbb{R}^p : \|\beta\|_1 \leq s\}$ είναι επίσης κυρτό. Προκειμένου να υπολογίσουμε τώρα την εκτιμήτρια Lasso είναι χρήσιμο να εισάγουμε τις έννοιες του υποδιαφορικού (subdifferential) και της υποκλίσης (subgradient) για μια κυρτή συνάρτηση.

Ορισμός 1. Έστω $f : \mathbb{R}^p \to \mathbb{R}$ μια κυρτή συνάρτηση, όχι απαραίτητα διαφορίσιμη σε όλο το $\mathbb{R}^p$. Ορίζουμε ως υποδιαφορικό (*subdifferential*) της f στο $\mathbf{x}$, το σύνολο:

$$\partial f(\mathbf{x}) = \{\mathbf{z} \in \mathbb{R}^p : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{z}, \mathbf{y} - \mathbf{x} \rangle \text{ για κάθε } \mathbf{y} \in \mathbb{R}^p\}.$$

Ένα διάνυσμα $\mathbf{z}$ το οποίο ανήκει στο υποδιαφορικό της f καλείται υποκλίση (*subgradient*) της f στο $\mathbf{x}$.

Αν μια συνάρτηση f είναι κυρτή και διαφορίσιμη σε ένα σημείο $\mathbf{x} \in \mathbb{R}^p$, τότε το υποδιαφορικό της είναι το μονοσύνολο:

$$\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\},$$

όπου $\nabla f(\mathbf{x})$ η κλίση της f στο $\mathbf{x}$. Τέλος ισχύει το παρακάτω Λήμμα.

Λήμμα 1. Gia kˆje kurt sunˆrthsh $f : \mathbb{R}^p \to \mathbb{R}$ isqÔei:

$$\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^p} f(\mathbf{x}) \iff \mathbf{0} \in \partial f(\mathbf{x}^*). \tag{1.2.5}$$

Απόδειξη. Και οι δύο συνθήκες είναι ισοδύναμες με

$$f(\mathbf{y}) \geq f(\mathbf{x}^*) + \langle \mathbf{0}, \mathbf{y} - \mathbf{x}^* \rangle \text{ για κάθε } \mathbf{y} \in \mathbb{R}^p.$$

□

Γυρνώντας τώρα στο πρόβλημα (1.2.4) μια ικανή και αναγκαία συνθήκη για μία λύση $\boldsymbol{\beta}$, προκύπτει από το Λήμμα 1 ως εξής:

$$\mathbf{0} \in \partial S(\boldsymbol{\beta}) \iff \mathbf{0} \in \partial(f(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1) = \nabla f(\boldsymbol{\beta}) + \lambda\partial\|\boldsymbol{\beta}\|_1 \tag{1.2.6}$$

με

$$S(\boldsymbol{\beta}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{και} \quad f(\boldsymbol{\beta}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

Όμως $f(\boldsymbol{\beta}) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2$ και για κάθε $j = 1, \ldots, p$ θα είναι

$$(\nabla f(\boldsymbol{\beta}))_j = \frac{\partial f(\boldsymbol{\beta})}{\partial \beta_j} = -\frac{1}{N}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})x_{ij} = -\frac{1}{N}\langle \mathbf{y} - \mathbf{X}\boldsymbol{\beta}, \mathbf{x}_j\rangle = -\frac{1}{N}\langle \mathbf{x}_j, \mathbf{y} - \mathbf{X}\boldsymbol{\beta}\rangle. \tag{1.2.7}$$

Επομένως προκύπτει ένα σύστημα p εξισώσεων

$$-\frac{1}{N}\langle \mathbf{x}_j, \mathbf{y} - \mathbf{X}\boldsymbol{\beta}\rangle + \lambda s_j = 0, \tag{1.2.8}$$

όπου s_j η j-συνιστώσα του διανύσματος υποκλίσης $\mathbf{s}$ το οποίο ανήκει στο υποδιαφορικό $\partial\|\boldsymbol{\beta}\|_1$ της νόρμας $\|\boldsymbol{\beta}\|_1$. Το σύστημα αυτό αποτελεί τις συνθήκες **Karush–Kuhn–Tucker (KKT)** για το πρόβλημα (1.2.4). Στη συνέχεια παρουσιάζουμε τη διαδικασία υπολογισμού της εκτιμήτριας Lasso.

1.3 Υπολογισμός της εκτιμήτριας Lasso

Προχωράμε τώρα στην επίλυση του προβλήματος ελαχιστοποίησης με τη μέθοδο Lasso, αναλύοντας πρώτα την απλή περίπτωση όπου στο μοντέλο μας υπάρχει μία μόνο επεξηγηματική μεταβλητή. Στη συνέχεια παρουσιάζουμε τη γενική περίπτωση πολλών επεξηγηματικών μεταβλητών.

1.3.1 Μια επεξηγηματική μεταβλητή

Αρχικά θεωρούμε το πρόβλημα στο οποίο η μεταβλητή απόκρισης εξαρτάται μόνο από έναν παράγοντα. Δοθέντος τυχαίου δείγματος $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ και θεωρώντας ότι οι παρατηρήσεις μας είναι κεντραρισμένες ώστε:

$$\bar{y} = 0, \ \bar{x} = 0 \quad \text{και} \quad \frac{1}{N}\sum_{i=1}^{N} x_i^2 = 1,$$

η συνάρτηση που ελαχιστοποιούμε γράφεται:

$$\min_{\beta\in\mathbb{R}}\left\{\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta x_i)^2 + \lambda|\beta|\right\}, \quad \lambda \geq 0. \tag{1.3.1}$$

Θεωρούμε την πρώτη παράγωγο της συνάρτησης ελαχιστοποίησης $S(\beta)$ ως προς β και θέτοντας την ίση με το μηδέν έχουμε:

1. Αν $\beta > 0$ τότε:

$$\frac{\partial S(\beta)}{\partial \beta} = 0 \ , \ \frac{1}{N}\sum_{i=1}^{N}(y_i \quad \beta x_i)(\ \ x_i) + \lambda = 0 \ ) \ \hat{\beta} = \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle \quad \lambda.$$

2. Αν $\beta < 0$ τότε :

$$\frac{\partial S(\beta)}{\partial \beta} = 0 \ , \ \frac{1}{N}\sum_{i=1}^{N}(y_i \quad \beta x_i)(\ \ x_i) \quad \lambda = 0 \ ) \ \hat{\beta} = \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle + \lambda,$$

3. Αν $\beta = 0$, τότε $\hat{\beta} = 0$,

όπου $\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{i=1}^{N} x_i y_i$ το εσωτερικό γινόμενο των διανυσμάτων $\mathbf{x}, \mathbf{y}$. Συνοπτικά λοιπόν θα έχουμε

$$\hat{\beta} = \begin{cases} \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle \quad \lambda, & \text{αν } \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle > \lambda \\ \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle + \lambda, & \text{αν } \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle < \quad \lambda \\ 0, & \text{αν } \frac{1}{N}|\langle \mathbf{x}, \mathbf{y}\rangle| \quad \lambda \end{cases}$$

δηλαδή

$$\hat{\beta} = S_\lambda\left(\frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle\right), \tag{1.3.2}$$

με $S_\lambda(z) = \text{sign}(z)(|z| \quad \lambda)_+$[2] ο τελεστής Soft-thresholding (Donoho and Johnstone, 1994). Ένας άλλος τρόπος προκύπτει με τη χρήση υποκλίσεων. Για να είναι το σημείο $\beta \in \mathbb{R}$ σημείο ελαχίστου της συνάρτησης

$$S(\beta) = \frac{1}{2N}\sum_{i=1}^{N}(y_i \quad \beta x_i)^2 + \lambda|\beta| = f(\beta) + \lambda|\beta|,$$

αρκεί

$$0 \in \partial S(\beta) = \partial\{f(\beta) + \lambda|\beta|\} = \{f(\beta) + \lambda\partial|\beta|,$$

άρα από τις συνθήκες KKT έχουμε

$$\frac{1}{N}\sum_{i=1}^{N}(y_i \quad \beta x_i)x_i + \lambda s = 0 \ ) \ \hat{\beta} = \frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle \quad \lambda s$$

όπου s η υποκλίση της $|\beta|$ με $s = \text{sign}(\beta)$ αν $\beta \neq 0$ και $s \in [\ 1, 1]$ αν $\beta = 0$. Άρα και πάλι με χρήση του τελεστή soft-thresholding η εκτιμήτρια Lasso, όταν έχουμε έναν παράγοντα θα γράφεται ως $\hat{\beta} = S_\lambda\left(\frac{1}{N}\langle \mathbf{x}, \mathbf{y}\rangle\right)$.

[2] $z_+ = \max\{z, 0\}, z \in \mathbb{R}$, είναι το θετικό μέρος του αριθμού z.

1.3.2 Πολλές επεξηγηματικές μεταβλητές

Θεωρούμε τώρα τη γενική περίπτωση όπου η μεταβλητή απόκρισης εξαρτάται από p επεξηγηματικές μεταβλητές, τις $X_1, \ldots, X_p$. Η εκτιμήτρια Lasso είναι λύση του προβλήματος:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{1.3.3}$$

με $\lambda \geq 0$, η παράμετρος συντονισμού. Η συνάρτηση που πρέπει να ελαχιστοποιήσουμε είναι κυρτή αλλά μη διαφορίσιμη στα σημεία όπου $\beta_j = 0$. Για την επίλυση τέτοιου είδους προβλημάτων ελαχιστοποίησης, έχουν προταθεί διάφοροι αλγόριθμοι. Στη συνέχεια του Κεφαλαίου παρουσιάζουμε έναν πολύ σημαντικό από αυτούς, τον αλγόριθμο Coordinate Descent.
Η συνάρτηση του προβλήματος (1.3.3) μπορεί να γραφεί ως:

$$S(\boldsymbol{\beta}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \sum_{k \neq j} \beta_k x_{ik} - \beta_j x_{ij}\right)^2 + \lambda \sum_{k \neq j} |\beta_k| + \lambda |\beta_j|. \tag{1.3.4}$$

Ορίζουμε το μερικό υπόλοιπο $r_i^{(j)} = y_i - \sum_{k \neq j} \hat{\beta}_k x_{ik}$, ως προς το δείκτη j. Θεωρώντας την πρώτη παράγωγο της $S(\boldsymbol{\beta})$ ως προς το συντελεστή β_j και κρατώντας σταθερούς τους συντελεστές $\hat{\beta}_k$, για $k \neq j$ έχουμε:

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \beta_j} = 0 \ , \ \frac{1}{N} \sum_{i=1}^{N} \left(r_i^{(j)} - \beta_j x_{ij}\right)(-x_{ij}) + \lambda s_j = 0$$

$$) \ \hat{\beta}_j = \frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle - \lambda s_j, \quad \text{για} \quad j = 1, \ldots, p,$$

με s_j η υποκλίση (subgradient) της $|\beta_j|$,

$$(s_j = \text{sign}(\beta_j), \quad \text{αν} \quad \beta_j \neq 0 \quad \text{και} \quad s_j \in [-1, 1], \quad \text{αν} \quad \beta_j = 0).$$

Οπότε η συνιστώσα j του διανύσματος $\hat{\boldsymbol{\beta}}$ της εκτιμήτριας θα είναι:

$$\hat{\beta}_j = \begin{cases} \frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle - \lambda, & \text{αν } \frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle > \lambda \\ \frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle + \lambda, & \text{αν } \frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle < -\lambda \\ 0, & \text{αν } \frac{1}{N} \left| \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle \right| \leq \lambda \end{cases}$$

δηλαδή

$$\hat{\beta}_j = S_\lambda \left(\frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle \right). \tag{1.3.5}$$

Στην ειδική περίπτωση όπου οι στήλες του πίνακα $\mathbf{X}$ είναι ορθογώνιες, δηλαδή $\langle \mathbf{x}_k, \mathbf{x}_j \rangle = 0$, για $k \neq j$, θα είναι:

$$\hat{\beta}_j = S_\lambda \left(\frac{1}{N} \langle \mathbf{r}^{(j)}, \mathbf{x}_j \rangle \right) = S_\lambda \left(\frac{1}{N} \langle \mathbf{y} - \sum_{k \neq j} \hat{\beta}_k \mathbf{x}_k, \mathbf{x}_j \rangle \right)$$
$$= S_\lambda \left(\frac{1}{N} \langle \mathbf{y}, \mathbf{x}_j \rangle - \frac{1}{N} \sum_{k \neq j} \hat{\beta}_k \langle \mathbf{x}_k, \mathbf{x}_j \rangle \right) = S_\lambda \left(\frac{1}{N} \langle \mathbf{y}, \mathbf{x}_j \rangle \right),$$

όπου $\frac{1}{N}\langle \mathbf{y}, \mathbf{x}_j \rangle$ είναι η εκτιμήτρια ελαχίστων τετραγώνων, αφού για τη μέθοδο ελαχίστων τετραγώνων θα έχουμε:

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \beta_j} = 0 \;\;,\;\; \frac{1}{N}\sum_{i=1}^{N}\Big(y_i \quad \sum_{k \neq j}\beta_k x_{ik} \quad \beta_j x_{ij}\Big)\big(\quad x_{ij}\big) = 0 \;\;\big)$$

$$\sum_{i=1}^{N} y_i x_{ij} = \sum_{i=1}^{N}\Big(\sum_{k \neq j}\beta_k x_{ik} x_{ij} + \beta_j x_{ij}^2\Big) = \sum_{k \neq j}\beta_k \underbrace{\sum_{i=1}^{N} x_{ik} x_{ij}}_{0} + \beta_j \sum_{i=1}^{N} x_{ij}^2 \;\;\big)$$

$$\langle \mathbf{y}, \mathbf{x}_j \rangle = \beta_j N \;\;\big)\;\; \hat{\beta}_j = \frac{1}{N}\langle \mathbf{y}, \mathbf{x}_j \rangle \text{ για κάθε } j = 1, \ldots, p,$$

αφού $\sum_{i=1}^{N} x_{ik} x_{ij} = \langle \mathbf{x}_k, \mathbf{x}_j \rangle = 0$.

Από τη σχέση (1.3.5) παρατηρούμε ότι κάθε συνιστώσα $\hat{\beta}_j$ εξαρτάται από τις υπόλοιπες $\hat{\beta}_k$ για $k \neq j$, μέσω του υπολοίπου $\mathbf{r}^{(j)}$. Άρα χρειαζόμαστε μια επαναληπτική διαδικασία η οποία να ελαχιστοποιεί με κυκλικό τρόπο τη συνάρτηση $S(\boldsymbol{\beta})$ ως προς κάθε συνιστώσα της, κρατώντας τις υπόλοιπες σταθερές και ανανεώνοντας κάθε μία από αυτές σύμφωνα με τη σχέση (1.3.5). Στη συνέχεια παρουσιάζουμε τη διαδικασία αυτή.

1.4 Coordinate Descent

Ο αλγόριθμος Coordinate Descent (Friedman et al, 2007) είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την ελαχιστοποίηση συγκεκριμένου τύπου συναρτήσεων. Στη γενική περίπτωση ελαχιστοποιεί συναρτήσεις με την εξής μορφή:

$$S(\boldsymbol{\beta}) = f(\boldsymbol{\beta}) + \sum_{j=1}^{p} h_j(\beta_j) \tag{1.4.1}$$

όπου $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$ ένα διάνυσμα παραμέτρων ως προς το οποίο ελαχιστοποιούμε τη συνάρτηση $S(\boldsymbol{\beta})$. Η συνάρτηση $f : \mathbb{R}^p \to \mathbb{R}$ είναι διαφορίσιμη και κυρτή, ενώ οι συναρτήσεις $h_j : \mathbb{R} \to \mathbb{R}$ είναι κυρτές (όχι απαραίτητα διαφορίσιμες). Σύμφωνα με τον Tseng (2001) ο αλγόριθμος Coordinate Descent συγκλίνει στο ολικό ελάχιστο μιας κυρτής συνάρτησης $S(\boldsymbol{\beta})$ η οποία έχει τη μορφή (1.4.1). Αυτό οφείλεται στο γεγονός ότι η συνάρτηση $\sum_{j=1}^{p} h_j(\beta_j)$ είναι διαχωρίσιμη (separable). Η παράσταση που ελαχιστοποιεί η μέθοδος Lasso:

$$S(\boldsymbol{\beta}) = \frac{1}{2N}\sum_{i=1}^{N}\Big(y_i \quad \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \quad \lambda \quad 0$$

ανήκει σε αυτού του είδους τις συναρτήσεις γιατί η

$$f(\boldsymbol{\beta}) = f(\beta_1, \ldots, \beta_p) = \frac{1}{2N}\sum_{i=1}^{N}\Big(y_i \quad \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2$$

είναι παραγωγίσιμη και κυρτή στον $\mathbb{R}^p$ και οι συναρτήσεις $h_j(\beta_j) = \lambda|\beta_j|$ είναι κυρτές στο $\mathbb{R}$ (αλλά μη διαφορίσιμες στα σημεία όπου $\beta_j = 0$, για κάθε $j = 1, \ldots, p$) και το άθροισμα τους είναι διαχωρίσιμο. Παρακάτω δίνουμε μια περιγραφή του αλγορίθμου για την περίπτωση της μεθόδου Lasso.

Αλγόριθμος Coordinate Descent

Ξεκινάμε από μια αρχική αυθαίρετη τιμή, έστω $\boldsymbol{\beta}^{(0)} = \left(\beta_1^{(0)}, \beta_2^{(0)}, \ldots, \beta_p^{(0)}\right)$. Στο πρώτο βήμα κρατάμε σταθερά τα $\left(\beta_2^{(0)}, \beta_3^{(0)}, \ldots, \beta_p^{(0)}\right)$ και ελαχιστοποιούμε την $S\left(\beta_1, \beta_2^{(0)}, \beta_3^{(0)}, \ldots, \beta_p^{(0)}\right)$ ως προς το β_1. Εν συνεχεία κρατάμε σταθερά τα $\left(\beta_1^{(1)}, \beta_3^{(0)}, \ldots, \beta_p^{(0)}\right)$ με $\beta_1^{(1)}$ η τιμή που προέκυψε από την πρώτη ελαχιστοποίηση, και τώρα ελαχιστοποιούμε την $S\left(\beta_1^{(1)}, \beta_2, \beta_3^{(0)}, \ldots, \beta_p^{(0)}\right)$ ως προς β_2. Συνεχίζουμε μέχρι να καταλήξουμε στην πρώτη εκτιμήτρια $\boldsymbol{\beta}^{(1)} = \left(\beta_1^{(1)}, \ldots, \beta_p^{(1)}\right)$. Επαναλαμβάνουμε τη διαδικασία μέχρι ο αλγόριθμος να συγκλίνει σε κάποια τιμή έστω $\hat{\boldsymbol{\beta}}$ η οποία θα είναι και η εκτιμήτρια Lasso. Ο αλγόριθμος συγκλίνει λόγω της κυρτότητας της συνάρτησης $S(\boldsymbol{\beta})$. Αφού είναι κυρτή, αν βρεθεί ένα τοπικό της ελάχιστο θα είναι και ολικό ελάχιστο. Σχηματικά ο αλγόριθμος Coordinate Descent έχει την εξής μορφή:

Coordinate Descent

Αρχικοποίηση: $\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)} \in \mathbb{R}^p$

Επανάληψη: $k = 1, 2, \ldots$ (μέχρι σύγκλιση)

$$\begin{aligned}
\beta_1^{(k)} &= \underset{\beta_1}{\operatorname{argmin}}\, S\left(\beta_1, \beta_2^{(k-1)}, \ldots, \beta_p^{(k-1)}\right) \\
\beta_2^{(k)} &= \underset{\beta_2}{\operatorname{argmin}}\, S\left(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \ldots, \beta_p^{(k-1)}\right) \\
&\vdots \\
\beta_p^{(k)} &= \underset{\beta_p}{\operatorname{argmin}}\, S\left(\beta_1^{(k)}, \ldots, \beta_{p-1}^{(k)}, \beta_p\right)
\end{aligned}$$

Έξοδος: $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$

Επομένως ελαχιστοποιούμε επαναληπτικά την παράσταση $S(\boldsymbol{\beta})$ λαμβάνοντας κάθε φορά μια εκτίμηση $\beta_j^{(k)}$ και σε κάθε επόμενο βήμα (μέχρι και την εύρεση του $\beta_p^{(k)}$) χρησιμοποιούμε αυτή τη νέα τιμή. Τελικά ο αλγόριθμος θα συγκλίνει στην εκτιμήτρια Lasso $\hat{\boldsymbol{\beta}}$. Επομένως για την j συνιστώσα της εκτιμήτριας Lasso θα έχουμε:

$$\hat{\beta}_j = S_\lambda\left(\frac{1}{N}\langle \mathbf{r}^{(j)}, \mathbf{x}_j\rangle\right) = S_\lambda\left(\frac{1}{N}\langle \mathbf{y} - \sum_{k \neq j} \hat{\beta}_k \mathbf{x}_k, \mathbf{x}_j\rangle\right) = S_\lambda\left(\frac{1}{N}\langle \mathbf{r} + \hat{\beta}_j \mathbf{x}_j, \mathbf{x}_j\rangle\right)$$

$$= S_\lambda\left(\hat{\beta}_j + \frac{1}{N}\langle \mathbf{r}, \mathbf{x}_j\rangle\right), \quad \mathbf{r} = \mathbf{y} - \sum_{j=1}^{p} \hat{\beta}_j \mathbf{x}_j,$$

άρα σε κάθε βήμα k η νέα τιμή της j συνιστώσας θα είναι

$$\beta_j^{(k)} \quad S_\lambda \left(\beta_j^{(k-1)} + \frac{1}{N} \langle \mathbf{r}, \mathbf{x}_j \rangle \right), \tag{1.4.2}$$

όπου για το υπόλοιπο $\mathbf{r}$ χρησιμοποιούμε τις τιμές που έχουν οι εκτιμήτριες σε εκείνο το βήμα. Συχνά θα θέλαμε να βρούμε την εκτιμήτρια Lasso για διάφορες τιμές της παραμέτρου συντονισμού λ. Θέλουμε δηλαδή τις λύσεις $\hat{\boldsymbol{\beta}}_\lambda$, για λ 2 , όπου = $\{\lambda_1, \ldots, \lambda_T\}$ ένα σύνολο τιμών της παραμέτρου ποινής, για το οποίο ενδιαφερόμαστε να υπολογίσουμε την εκτιμήτρια. Συνήθως οι τιμές του Λ είναι σε φθίνουσα σειρά ($\lambda_1 >$ $> \lambda_T$) όπου μπορούμε να χρησιμοποιήσουμε ως αρχική τιμή την ελάχιστη τιμή του λ για την οποία η εκτιμήτρια Lasso έχει όλες τις συνιστώσες της $\hat{\beta}_j$ ίσες με το μηδέν. Αυτή η τιμή είναι η $\lambda = \max_j \frac{1}{N} |\langle \mathbf{y}, \mathbf{x}_j \rangle|$, αφού για κάθε β_j έχουμε:

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \beta_j} = 0 \quad) \quad \frac{1}{N} \sum_{i=1}^{N} \Big(y_i \quad \sum_{j=1}^{p} \beta_j x_{ij}\Big) x_{ij} + \lambda s_j = 0$$

$$^{\beta_j=0} \; \lambda = \frac{1}{s_j} \frac{\langle \mathbf{y}, \mathbf{x}_j \rangle}{N} \; ^{\lambda})^{0} \; \lambda = \frac{1}{|s_j|} \frac{|\langle \mathbf{y}, \mathbf{x}_j \rangle|}{N}$$

$$^{|s_j|\ 1} \; \frac{1}{N} |\langle \mathbf{y}, \mathbf{x}_j \rangle|, \text{ για κάθε } j = 1, \ldots, p,$$

Επομένως $\lambda = \max_j \frac{1}{N} |\langle \mathbf{y}, \mathbf{x}_j \rangle|$ και μπορούμε να ξεκινήσουμε από αυτή την τιμή για τον υπολογισμό των εκτιμητριών, αφού για κάθε τιμή μεγαλύτερη από αυτή γνωρίζουμε ότι η μέθοδος Lasso θα παράγει μηδενικές λύσεις. Ξεκινάμε λοιπόν να υπολογίζουμε το $\hat{\boldsymbol{\beta}}_{\lambda_1}$, χρησιμοποιώντας για αρχική τιμή την $\boldsymbol{\beta}^{(0)} = \mathbf{0}$. Ύστερα υπολογίζουμε το $\hat{\boldsymbol{\beta}}_{\lambda_2}$ χρησιμοποιώντας ως αρχική τιμή $\boldsymbol{\beta}^{(0)} = \hat{\boldsymbol{\beta}}_{\lambda_1}$, κ.ο.κ. Έτσι κάθε φορά που εφαρμόζουμε τον αλγόριθμο ξεκινάμε από την λύση που προέκυψε από την προηγούμενη εκτέλεσή του, η οποία ονομάζεται \warm-start". Καταλήγουμε λοιπόν σε T εκτιμήτριες τις $\hat{\boldsymbol{\beta}}_{\lambda_1}, \ldots, \hat{\boldsymbol{\beta}}_{\lambda_T}$, οι οποίες λέμε ότι αποτελούν ένα μονοπάτι λύσεων. Για το λόγο αυτό η διαδικασία ονομάζεται Pathwise Coordinate descent. Το πακέτο glmnet στην R χρησιμοποιεί αυτόν τον αλγόριθμο.

1.5 Μοναδικότητα της εκτιμήτριας Lasso

Ένα εύλογο ερώτημα που προκύπτει είναι αν η εκτιμήτρια Lasso για μια συγκεκριμένη τιμή της παραμέτρου ποινής λ είναι μοναδική και αν όχι υπό ποιες συνθήκες θα μπορούσαμε να πούμε ότι υπάρχει μοναδική λύση. Στη συνέχεια δίνουμε την απάντηση σε αυτό το ερώτημα.

Πρόταση 1. Έστω $\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2$ δύο εκτιμήτριες Lasso οι οποίες αντιστοιχούν στην ίδια τιμή λ της παραμέτρου ποινής. Τότε για τις προσαρμοσμένες τιμές $\hat{f}_1 = \mathbf{X}\hat{\boldsymbol{\beta}}_1$ και $\hat{f}_2 = \mathbf{X}\hat{\boldsymbol{\beta}}_2$ θα ισχύει ότι $\hat{f}_1 = \hat{f}_2$.

Apìdeixh. Θέτουμε $\hat{\boldsymbol{\beta}} = \frac{\hat{\boldsymbol{\beta}}_1 + \hat{\boldsymbol{\beta}}_2}{2}$. Υποθέτουμε ότι

$$\mathbf{X}\hat{\boldsymbol{\beta}}_1 \neq \mathbf{X}\hat{\boldsymbol{\beta}}_2 \text{ άρα } \mathbf{y} \quad \mathbf{X}\hat{\boldsymbol{\beta}}_1 \neq \mathbf{y} \quad \mathbf{X}\hat{\boldsymbol{\beta}}_2.$$

Τότε θα είναι:

$$\begin{aligned} S(\hat{\boldsymbol{\beta}}) &= \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2 + \lambda\|\hat{\boldsymbol{\beta}}\|_1 \\ &= \frac{1}{2N}\left\|\frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_1) + \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_2)\right\|_2^2 + \frac{\lambda}{2}\|\hat{\boldsymbol{\beta}}_1 + \hat{\boldsymbol{\beta}}_2\|_1 \\ &\quad \frac{1}{2N}\left(\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_1\|_2 + \frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_2\|_2\right)^2 + \frac{\lambda}{2}(\|\hat{\boldsymbol{\beta}}_1\|_1 + \|\hat{\boldsymbol{\beta}}_2\|_1) \\ &< \frac{1}{2N}\left(\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_1\|_2^2 + \frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_2\|_2^2\right) + \frac{\lambda}{2}(\|\hat{\boldsymbol{\beta}}_1\|_1 + \|\hat{\boldsymbol{\beta}}_2\|_1) \\ &= \frac{1}{2}S(\hat{\boldsymbol{\beta}}_1) + \frac{1}{2}S(\hat{\boldsymbol{\beta}}_2). \end{aligned}$$

Στην πρώτη και δεύτερη ανίσωση χρησιμοποιήσαμε αντίστοιχα την τριγωνική ανισότητα και ότι η συνάρτηση $g(x) = x^2, x \in \mathbb{R}$ είναι αυστηρά κυρτή. Καταλήγουμε λοιπόν σε άτοπο, διότι οι $\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2$ θα ελαχιστοποιούν την $S(\boldsymbol{\beta})$ ως λύσεις, άρα

$$\frac{1}{2}S(\hat{\boldsymbol{\beta}}_1) + \frac{1}{2}S(\hat{\boldsymbol{\beta}}_2) \not< S(\hat{\boldsymbol{\beta}}).$$

Συνεπώς η προσαρμοσμένη τιμή τους θα είναι ίδια: $\mathbf{X}\hat{\boldsymbol{\beta}}_1 = \mathbf{X}\hat{\boldsymbol{\beta}}_2$. □

Έστω τώρα ένα $\lambda > 0$ για το οποίο οι $\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2$ ελαχιστοποιούν τη συνάρτηση $S(\boldsymbol{\beta})$. Τότε θα υπάρχουν υποκλίσεις $\hat{z}_1, \hat{z}_2 \in \partial\|\hat{\boldsymbol{\beta}}\|_1$ αντίστοιχα ώστε

$$\begin{aligned} -\nabla f(\hat{\boldsymbol{\beta}}_1) + \lambda\hat{z}_1 = \mathbf{0} \;\Rightarrow\; \frac{\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_1)}{N} + \lambda\hat{z}_1 = \mathbf{0} \\ -\nabla f(\hat{\boldsymbol{\beta}}_2) + \lambda\hat{z}_2 = \mathbf{0} \;\Rightarrow\; \frac{\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_2)}{N} + \lambda\hat{z}_2 = \mathbf{0} \end{aligned}$$

όπου $f(\boldsymbol{\beta}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ και αφού $\mathbf{X}\hat{\boldsymbol{\beta}}_1 = \mathbf{X}\hat{\boldsymbol{\beta}}_2$ θα είναι και $\hat{z}_1 = \hat{z}_2$.
Έστω $\hat{\boldsymbol{\beta}}$ μια εκτιμήτρια που αντιστοιχεί στην παράμετρο λ. Για $\hat{z} \in \partial\|\hat{\boldsymbol{\beta}}\|_1$ ορίζουμε το σύνολο

$$J = \{j : |\hat{z}_j| = 1\}$$

οπότε αν $\beta_j \neq 0$ τότε αφού $\hat{z}_j = \text{sign}(\beta_j) \Rightarrow |\hat{z}_j| = 1$ θα ισχύει $j \in J$. Διαφορετικά αν $\beta_j = 0$ τότε $j \in J^c = \{j : \hat{z}_j \in [-1, 1]\}$. Επομένως για κάθε $j \in J^c$ θα είναι $\hat{\beta}_j = 0$ και γράφουμε $\hat{\boldsymbol{\beta}}_{J^c} = \mathbf{0}$. Εφόσον τώρα ισχύει

$$\frac{\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{N} + \lambda\hat{z} = \mathbf{0} \;\Rightarrow\; \hat{z} = \frac{\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{\lambda N}$$

επομένως θα έχουμε:

$$\lambda N\hat{z}_j = (\mathbf{X}^\mathbf{T}\mathbf{y})_j - (\mathbf{X}^\mathbf{T}\mathbf{X}\hat{\boldsymbol{\beta}})_j = \mathbf{x}_j^T\mathbf{y} - \mathbf{x}_j^T\mathbf{X}\boldsymbol{\beta},$$

αφού

$$\mathbf{X^T y} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{y} \\ \vdots \\ \mathbf{x}_p^T\mathbf{y} \end{bmatrix}_{p\times 1} \quad \text{και} \quad \mathbf{X^T X}\hat{\boldsymbol{\beta}} = \begin{bmatrix} \sum_{k=1}^p \mathbf{x}_1^T\mathbf{x}_k\hat{\beta}_k \\ \vdots \\ \sum_{k=1}^p \mathbf{x}_p^T\mathbf{x}_k\hat{\beta}_k \end{bmatrix}_{p\times 1}$$

άρα $(\mathbf{X^T y})_j = \mathbf{x}_j^T\mathbf{y}$ και $(\mathbf{X^T X}\hat{\boldsymbol{\beta}})_j = \sum_{k=1}^p \mathbf{x}_j^T\mathbf{x}_k\hat{\beta}_k = \mathbf{x}_j^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{x}_j^T\mathbf{X}_J\hat{\boldsymbol{\beta}}_J$.[3]
Συνεπώς για $j \in J$ θα έχουμε:

$$\lambda N\hat{z}_j = \mathbf{x}_j^T\mathbf{y} \quad \mathbf{x}_j^T\mathbf{X}_J\hat{\boldsymbol{\beta}}_J,$$

και σε όλο το σύνολο J γράφουμε:

$$\lambda N\hat{z}_J = \mathbf{X}_j^T\mathbf{y} \quad \mathbf{X}_J^T\mathbf{X}_J\hat{\boldsymbol{\beta}}_J.$$

Αν ο πίνακας $\mathbf{X}_J^T\mathbf{X}_J$ είναι αντιστρέψιμος τότε η εκτιμήτρια στο J θα είναι:

$$\hat{\boldsymbol{\beta}}_J = (\mathbf{X}_J^T\mathbf{X}_J)^{-1}(\mathbf{X}_j^T\mathbf{y} \quad \lambda N\hat{z}_J). \tag{1.5.1}$$

Τέλος θεωρούμε δύο λύσεις $\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2$ (για το ίδιο λ). Τότε αυτές θα είναι ίσες στο σύνολο J^c, δηλαδή και οι δύο ίσες με το $\mathbf{0}$. Αν ο πίνακας $\mathbf{X}_J^T\mathbf{X}_J$ είναι αντιστρέψιμος τότε αυτές θα είναι ίσες με:

$$(\hat{\boldsymbol{\beta}}_1)_J = (\hat{\boldsymbol{\beta}}_2)_J = (\mathbf{X}_J^T\mathbf{X}_J)^{-1}(\mathbf{X}_j^T\mathbf{y} \quad \lambda N\hat{z}_J)$$

και στο σύνολο J. Συμπεραίνουμε λοιπόν ότι η εκτιμήτρια με τη μέθοδο Lasso είναι μοναδική αν ο πίνακας $\mathbf{X}_J^T\mathbf{X}_J$ είναι αντιστρέψιμος.

1.6 Cross Validation

Όπως έχουμε ήδη αναφέρει η επιλογή της παραμέτρου ποινής λ είναι πολύ σημαντική για το μοντέλο μας. Μια αρκετά μικρή τιμή της δε θέτει μεγάλο περιορισμό στους συντελεστές του μοντέλου και έτσι η εκτιμήτρια που θα προκύψει με τη μέθοδο Lasso θα είναι κοντά στην εκτιμήτρια ελαχίστων τετραγώνων. Αντιθέτως, όσο η τιμή της παραμέτρου ποινής αυξάνει, τόσο περισσότερο περιορίζουμε τους συντελεστές των επεξηγηματικών μεταβλητών, με αποτέλεσμα αρκετοί από αυτούς να μηδενίζονται. Έτσι όμως υπάρχει περίπτωση να παραλείψουμε και ορισμένους σημαντικούς παράγοντες από το μοντέλο μας. Ποια είναι λοιπόν η βέλτιστη επιλογή της παραμέτρου λ, ώστε να έχουμε καλή προσαρμογή στα δεδομένα μας και ταυτόχρονα να μη χάνουμε αρκετή πληροφορία έχοντας καταλήξει σε ένα πολύ αραιό μοντέλο; Η απάντηση εξαρτάται από το κριτήριο που θα επιλέξουμε. Συνήθως αυτό που μας ενδιαφέρει είναι το μοντέλο μας να έχει υψηλή προβλεπτική ικανότητα για παρατηρήσεις που δεν ανήκουν στο αρχικό σύνολο δεδομένων (unseen data). Στη πράξη όμως τις περισσότερες φορές δε διαθέτουμε τέτοιες παρατηρήσεις, ώστε να εξετάσουμε την προβλεπτική ακρίβεια του μοντέλου μας. Η μέθοδος Cross Validation είναι μια τεχνική που μπορούμε να χρησιμοποιήσουμε ώστε να δώσουμε λύση σε τέτοιου είδους προβλήματα.

[3] $\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}_J\hat{\boldsymbol{\beta}}_J$ διότι $\hat{\boldsymbol{\beta}}_{J^c} = \mathbf{0}$

1.6.1 K-fold Cross Validation

Θεωρούμε και πάλι ότι διαθέτουμε ενα σύνολο από N ανεξάρτητες παρατηρήσεις $f(\mathbf{x}_i, y_i)g_{i=1}^N$. Η διαδικασία που εφαρμόζεται έχει ως εξής: Χωρίζουμε τα δεδομένα σε K μη αλληλοεπικαλυπτόμενες ομάδες-φακέλους (folds) ίδιου αν είναι δυνατόν μεγέθους. Κρατάμε τον πρώτο φάκελο εκτός και προσαρμόζουμε το μοντέλο χρησιμοποιώντας τις παρατηρήσεις των υπόλοιπων K 1 φακέλων. Έτσι έχουμε δύο σετ δεδομένων. Το σύνολο που περιέχει τις παρατηρήσεις με τη βοήθεια των οποίων προσαρμόζουμε το μοντέλο, το οποίο ονομάζεται σύνολο προσαρμογής (training set), και το σύνολο που θα χρησιμοποιηθεί για να ελέγξουμε τις εκτιμήσεις μας, το οποίο ονομάζεται σύνολο ελέγχου (test set). Υπολογίζουμε το μέσο τετραγωνικό σφάλμα

$$\mathsf{MSE}_1 = \frac{1}{k}\sum_{i2K_1}(y_i \quad \hat{y}_i)^2, \tag{1.6.1}$$

όπου k το μέγεθος του πρώτου φακέλου K_1 και $\hat{y}_i$ οι τιμές της εκτίμησης της εξαρτημένης μεταβλητής υπολογισμένες στα δεδομένα της πρώτης ομάδας (test set). Στη συνέχεια αφήνουμε το δεύτερο φάκελο εκτός και προσαρμόζουμε το μοντέλο με βάση τους φακέλους $1, 3, \ldots . K$. Βρίσκουμε το MSE_2. Επαναλαμβάνουμε τη διαδικασία αφήνοντας κάθε φορά έναν φάκελο εκτός, έως ότου και οι K ομάδες παρατηρήσεων γίνουν από μία φορά test set. Έτσι έχουμε στη διάθεσή μας K το πλήθος μέσα τετραγωνικά σφάλματα. Παίρνοντας τη μέση τους τιμή έχουμε το Cross Validation Error που αποτελεί μια εκτίμηση για το μέσο τετραγωνικό σφάλμα των μελλοντικών παρατηρήσεων (test MSE). Δηλαδή:

$$\mathsf{CV}_{(K)} = \frac{1}{K}\sum_{i=1}^{K} MSE_i \tag{1.6.2}$$

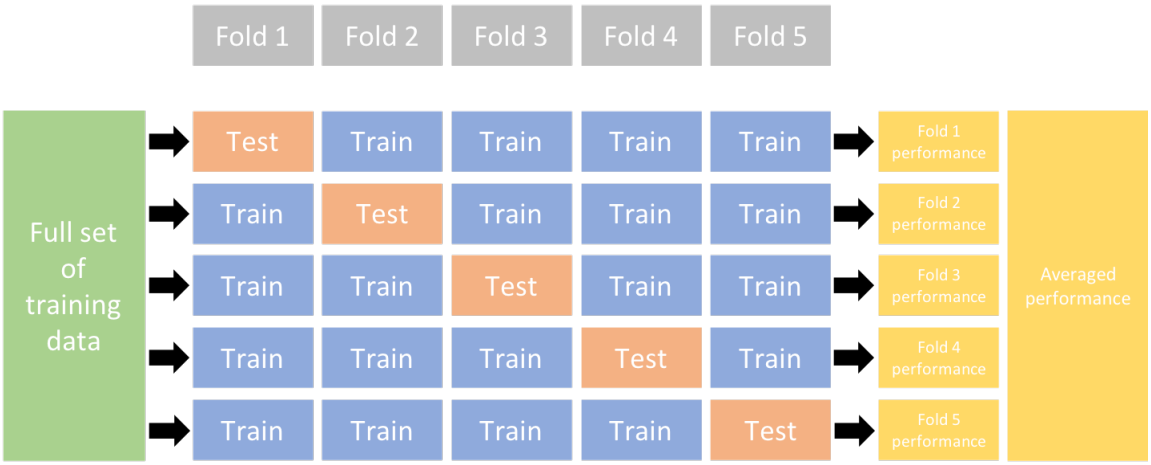
Συνηθισμένες επιλογές για τον αριθμό των φακέλων είναι $K = 5$ ή $K = 10$. Η περίπτωση $K = N$ ονομάζεται *Leave One Out Cross Validation*, αφού κάθε φορά αφήνουμε μία παρατήρηση εκτός και προσαρμόζουμε το μοντέλο στις υπόλοιπες N 1 παρατηρήσεις. Μια σχηματική απεικόνιση για την περίπτωση $K = 5$ φαίνεται στο Σχήμα 1.1. Για να επιλέξουμε τώρα την παράμετρο ποινής λ στη μέθοδο Lasso, αρχικά χρησιμοποιούμε ένα πλέγμα τιμών (grid) = $f\lambda_1, \ldots, \lambda_T g$ και για κάθε μία τιμή στο προσαρμόζουμε το μοντέλο. Εκτελούμε τη μέθοδο Cross Validation όπως περιγράψαμε και για κάθε μία τιμή της παραμέτρου ποινής καταλήγουμε σε ένα διαφορετικό σφάλμα. Συνήθως επιλέγουμε την τιμή εκείνη για την οποία το σφάλμα αυτό είναι το ελάχιστο.

1.7 Παλινδρόμηση κορυφογραμμής

Μια παρόμοια μέθοδος με αυτήν της μεθόδου Lasso είναι η παλινδρόμηση κορυφογραμμής ή Ridge regression (Hoerl and Kennard, 1970). Η εκτιμήτρια σε αυτή τη περίπτωση ελαχιστοποιεί τη συνάρτηση:

$$S(\boldsymbol{\beta}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i \quad \beta_0 \quad \sum_{j=1}^{p}\beta_j x_{ij}\right)^2, \qquad \text{με} \quad \sum_{j=1}^{p}\beta_j^2 \quad s. \tag{1.7.1}$$

Ο όρος ποινής μοιάζει αρκετά με αυτόν της μεθόδου Lasso, αλλά εδώ περιορίζουμε το άθροισμα των τετραγώνων των συντελεστών του μοντέλου. Μια ισοδύναμη μορφή της (1.7.1) λόγω της



Σχήμα 1.1: Cross Validation για K = 5 folds. Κάθε φορά αφήνουμε ένα φάκελο εκτός (test set) και προσαρμόζουμε το μοντέλο χρησιμοποιώντας τους υπόλοιπους φακέλους (training set). Τέλος εκτιμούμε την προβλεπτική ικανότητα τους μοντέλου μας μέσω του σφάλματος Cross Validation. (Phg : *Hands-On Machine Learning with R, Boehmke-Greenwell).*

Λαγκρανζιανής δυϊκότητας είναι η:

$$S(\boldsymbol{\beta}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i \quad \beta_0 \quad \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{1.7.2}$$

με λ 0 η παράμετρος συντονισμού. Και εδώ μπορούμε να κάνουμε τυποποίηση στις τιμές των ανεξάρτητων μεταβλητών και να κεντράρουμε την μεταβλητή απόκρισης, οπότε και παραλείπεται η σταθερά β_0 από την ελαχιστοποίηση του προβλήματος Ridge. Γράφοντας το πρόβλημα με τη μορφή πινάκων, η εκτιμήτρια με τη μέθοδο Ridge προκύπτει ελαχιστοποιώντας την:

$$S(\boldsymbol{\beta}) = \frac{1}{2N}k\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta}k_2^2 + \lambda k\boldsymbol{\beta}k_2^2, \tag{1.7.3}$$

οπότε θεωρώντας την πρώτη παράγωγο ως προς $\boldsymbol{\beta}$ κι θέτοντας την ίση με το μηδέν έχουμε:

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{0} \;) \quad \frac{1}{N}(\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta})^T\mathbf{X} + 2\lambda\boldsymbol{\beta}^T = \mathbf{0}$$

και παίρνοντας αντίστροφο και στα δύο μέλη και με χρήση της ιδιότητας $(\mathbf{AB})^{\mathbf{T}} = \mathbf{B}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}$ των πινάκων, έχουμε:

$$\frac{1}{N}\mathbf{X}^T(\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0},$$

άρα η εκτιμήτρια με τη μέθοδο Ridge θα είναι:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + 2\lambda N\mathbf{I}_p) \;^1\mathbf{X}^{\mathbf{T}}\mathbf{y}. \tag{1.7.4}$$

Ένας άλλος τρόπος προκύπτει παρατηρώντας ότι η συνάρτηση ελαχιστοποίησης μπορεί να γραφεί ως εξής:

$$S(\boldsymbol{\beta}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i \quad \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \frac{1}{2N}\sum_{j=1}^{p}(0 \quad p\overline{2\lambda N}\beta_j)^2.$$

Έτσι είναι σα να θεωρούμε ότι έχουμε p επιπλέον παρατηρήσεις όπου η μεταβλητή απόκρισης είναι μηδενική και ο πίνακας σχεδιασμού είναι ο $\sqrt{2\lambda N}\mathbf{I}_p$. Οπότε τα συνολικά δεδομένα μπορούν να γραφούν ως:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(N+p)\times 1} \quad \text{και} \quad \mathbf{X} = \begin{bmatrix} \mathbf{X} \\ \sqrt{2\lambda N}\mathbf{I}_p \end{bmatrix}_{(N+p)\times p}.$$

και μπορούμε να δούμε το πρόβλημα σα να ελαχιστοποιούμε την $S(\boldsymbol{\beta})$ με τη μέθοδο ελαχίστων τετραγώνων. Οπότε η εκτιμήτρια Ridge θα είναι:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{y} = \left[\begin{pmatrix}\mathbf{X}^T & \sqrt{2\lambda N}\mathbf{I}_p\end{pmatrix}\begin{pmatrix}\mathbf{X} \\ \sqrt{2\lambda N}\mathbf{I}_p\end{pmatrix}\right]^{-1}\begin{pmatrix}\mathbf{X}^T & \sqrt{2\lambda N}\mathbf{I}_p\end{pmatrix}\begin{bmatrix}\mathbf{y} \\ \mathbf{0}_p\end{bmatrix}$$

$$= (\mathbf{X}^\mathbf{T}\mathbf{X} + 2\lambda N\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y}$$

και έτσι καταλήξαμε στον ίδιο τύπο. Η διαδικασία αυτή ονομάζεται προσαύξηση των δεδομένων. Η αναμενόμενη τιμή της εκτιμήτριας θα είναι:

$$\mathrm{E}[\hat{\boldsymbol{\beta}}] = (\mathbf{X}^\mathbf{T}\mathbf{X} + 2\lambda N\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \neq \boldsymbol{\beta} \text{ για } \lambda > 0$$

και άρα η εκτιμήτρια με τη μέθοδο Ridge είναι μεροληπτική. Η παλινδρόμηση Ridge είναι μια τεχνική που μπορεί να εφαρμοστεί σε περιπτώσεις όπου παρατηρείται το φαινόμενο της πολυσυγγραμμικότητας (multicollinearity). Με τον όρο πολυσυγγραμμικότητα εννοούμε ότι υπάρχει υψηλή συσχέτιση μεταξύ δύο ή περισσότερων επεξηγηματικών μεταβλητών. Αυτό μπορεί να οδηγήσει σε ορισμένα προβλήματα όσον αφορά τη σταθερότητα των συντελεστών των ανεξάρτητων μεταβλητών. Για παράδειγμα, έστω ότι μια μεταβλητή X_j εξαρτάται γραμμικά με ορισμένες από τις υπόλοιπες συμμεταβλητές $X_k, k \neq j$, δηλαδή ισχύει $X_j = a + \sum_{k\neq j}\beta_k X_k$. Τότε το πλήθος των γραμμικά ανεξάρτητων στηλών του $\mathbf{X}^\mathbf{T}\mathbf{X}$ θα είναι μικρότερο του p (συνολικό πλήθος παραγόντων), άρα θα ισχύει $\mathrm{rank}(\mathbf{X}^\mathbf{T}\mathbf{X}) < p \Rightarrow \det(\mathbf{X}^\mathbf{T}\mathbf{X}) = 0$ και έτσι ο πίνακας δεν αντιστρέφεται. Επομένως η εκτιμήτρια ελαχίστων τετραγώνων δεν υπάρχει. Στην πράξη σπάνια παρατηρείται τέλεια πολυσυγγραμμικότητα μεταξύ των παραγόντων. Παρ΄ όλα αυτά, πολύ συχνά θα υπάρχει συσχέτιση μεταξύ κάποιων επεξηγηματικών μεταβλητών, με αποτέλεσμα να οδηγούμαστε σε εκτιμήτριες με μεγάλα τυπικά σφάλματα. Κατά συνέπεια, το μοντέλο που προκύπτει, δεν μπορεί να δώσει αξιόπιστες προβλέψεις. Με τη μέθοδο Ridge η εκτιμήτρια υπάρχει πάντα, ακόμη και αν ο πίνακας $\mathbf{X}^\mathbf{T}\mathbf{X}$ είναι μη αντιστρέψιμος. Θέτοντας όπως είδαμε ένα φράγμα στους συντελεστές του μοντέλου, μπορεί η μεροληψία της εκτιμήτριας να αυξάνεται, αλλά περιμένουμε γενικά σημαντική μείωση στη διασπορά της. Έτσι το μοντέλο με την εκτιμήτρια Ridge συνήθως οδηγεί σε μικρότερο μέσο τετραγωνικό σφάλμα, άρα και υψηλότερη ικανότητα πρόβλεψης από το μοντέλο με την εκτιμήτρια ελαχίστων τετραγώνων. Η μέθοδος Ridge προηγείται χρονικά της μεθόδου Lasso. Και οι δύο μέθοδοι έχουν ως επί το πλείστον εφαρμογή σε περιπτώσεις όπου ο αριθμός των παραγόντων p είναι ίδιος ή αρκετά μεγαλύτερος του αριθμού N των παρατηρήσεων ($p \geq N$).
Εντούτοις, η σημαντική διαφοροποίηση ανάμεσα στις δύο μεθόδους, έγκειται στο γεγονός ότι η μέθοδος Lasso για κατάλληλα επιλεγμένη τιμή του λ παράγει ένα αραιό μοντέλο, με την έννοια ότι μηδενίζει τους συντελεστές των μη σημαντικών μεταβλητών κρατώντας έτσι μόνο ένα υποσύνολο

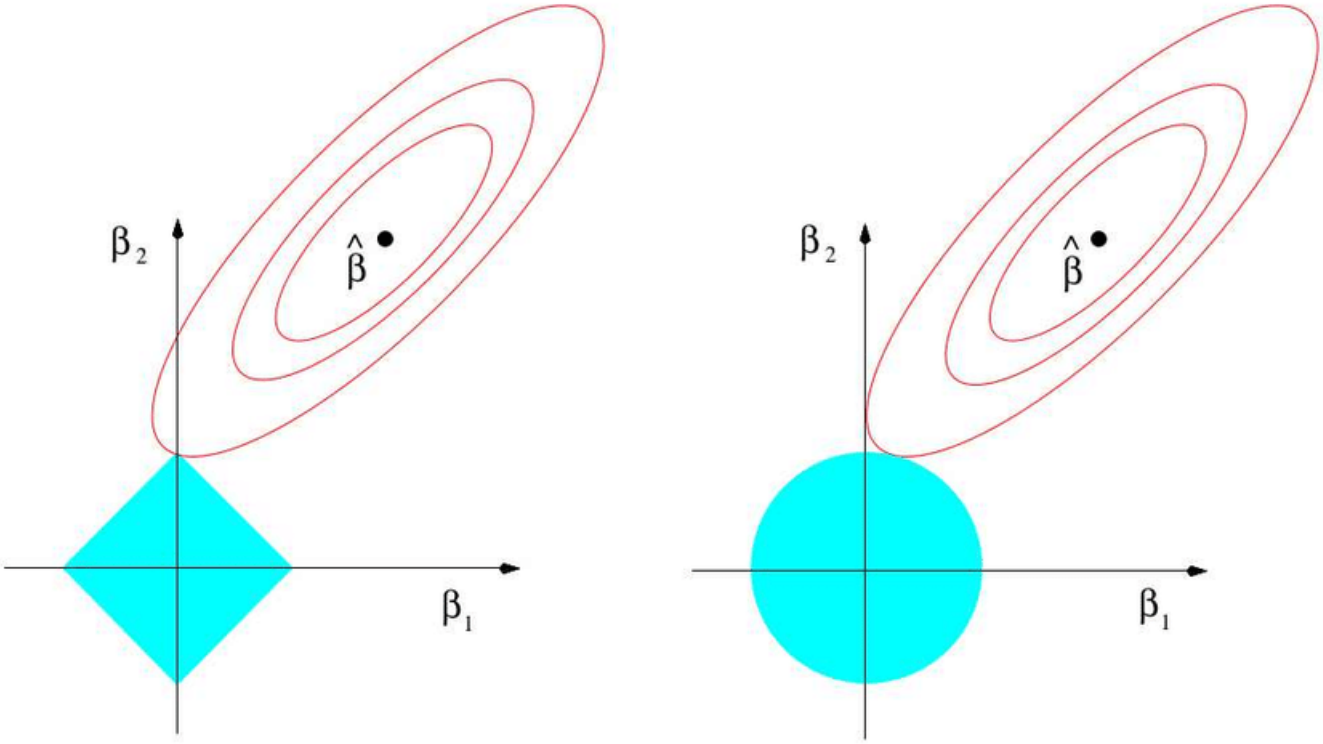
αυτών στο τελικό μοντέλο. Αντιθέτως, με τη μέθοδο Ridge αν και η ποινή της περιορίζει σημαντικά τους συντελεστές (για κατάλληλα μεγάλο λ οι παράμετροι τείνουν στο 0), τελικά δε θέτει κανέναν ίσο με 0 με αποτέλεσμα όλες οι μεταβλητές να περιλαμβάνονται στο μοντέλο. Ο λόγος για τον οποίο η μέθοδος Lasso έχει αυτήν τη σημαντική ιδιότητα, γίνεται αντιληπτός μέσω της γεωμετρίας του προβλήματος. Θεωρούμε την περίπτωση όπου έχουμε δύο συμμεταβλητές X_1, X_2 με συντελεστές β_1, β_2 αντίστοιχα. Τότε ο περιορισμός στη μέθοδο Ridge γράφεται $\beta_1^2 + \beta_2^2 \quad s$, όπου στο επίπεδο παριστάνει τον κυκλικό δίσκο με κέντρο το $O(0,0)$ και ακτίνα $\sqrt{s}$. Στη μέθοδο Lasso το σύνολο περιορισμού είναι το $|\beta_1| + |\beta_2| \quad s$ και παριστάνει την περιοχή που περικλείεται από το ρόμβο κέντρου 0 (όπως φαίνεται στο Σχήμα 1.2). Και στις δύο περιπτώσεις ελαχιστοποιούμε τη συνάρτηση

$$S(\beta_1, \beta_2) = \frac{1}{2N} \sum_{i=1}^{N} (y_i \quad \beta_1 x_{i1} \quad \beta_2 x_{i2})^2$$

που παριστάνεται με ελλείψεις που έχουν κέντρο την εκτιμήτρια ελαχίστων τετραγώνων, έστω $\hat{\boldsymbol{\beta}}$. Οι λύσεις και στις δύο μεθόδους είναι τα σημεία εκείνα του επιπέδου, όπου η έλλειψη τέμνει το αντίστοιχο σύνολο περιορισμού για πρώτη φορά. Στη περίπτωση Lasso, όταν η έλλειψη τέμνει το ρόμβο σε μια από τις δύο γωνίες τότε ο άλλος συντελεστής μηδενίζεται. Αντιθέτως ο κυκλικός δίσκος δεν έχει γωνίες, άρα τα σημεία τομής του με την έλλειψη θα έχουν κάποια συντεταγμένη πολύ κοντά στο 0, αλλά όχι ακριβώς ίση με το 0. Στη γενική περίπτωση όπου έχουμε p παράγοντες το σύνολο περιορισμού της Lasso παρουσιάζει πολλές γωνίες και κατά συνέπεια υπάρχει μεγάλη πιθανότητα αρκετοί συντελεστές να μηδενίζονται. Από το Σχήμα 1.2 μπορεί να γίνει επίσης αντιληπτό ότι όσο μειώνουμε τη παράμετρο συντονισμού λ (αντίστοιχα αυξάνουμε το φράγμα s, άρα τα σύνολα περιορισμού μεγαλώνουν), οι συντελεστές δεν περιορίζονται αρκετά και έτσι τείνουν προς την εκτιμήτρια ελαχίστων τετραγώνων.

1.8 Συμπέρασμα

Η μέθοδος Lasso μπορεί να χρησιμοποιηθεί ως μια εναλλακτική διαδικασία από αυτή της μεθόδου των ελαχίστων τετραγώνων, με σκοπό την προσαρμογή ενός μοντέλου γραμμικής παλινδρόμησης. Μέσω αυτής ελαχιστοποιούμε το άθροισμα των τετραγώνων των υπολοίπων μαζί με μια l_1-ποινή ή περιορισμό. Είδαμε ότι η επιλογή της παραμέτρου ποινής μπορεί να γίνει με τη διαδικασία του Cross Validation. Έτσι, από ένα πλήθος αρχικών τιμών λ επιλέγουμε την τιμή εκείνη που ελαχιστοποιεί το σφάλμα cross validation, δηλαδή μια εκτίμηση του μέσου τετραγωνικού σφάλματος για μελλοντικές παρατηρήσεις. Με τη βοήθεια του αλγορίθμου Coordinate descent μπορεί κάποιος με επαναληπτικές μεθόδους να καταλήξει στην εκτιμήτρια Lasso. Ο αλγόριθμος αυτός συγκλίνει στην εκτιμήτρια Lasso γιατί η συνάρτηση που ελαχιστοποιούμε είναι κυρτή και ο l_1-περιορισμός αποτελεί επίσης ένα κυρτό σύνολο. Επιπλέον από τη σύγκριση που έγινε με την παλινδρόμηση κορυφογραμμής, συμπεραίνουμε ότι λόγω της γεωμετρίας του l_1-περιορισμού, είναι πιθανότερο πολλές από τις εκτιμήσεις των συντελεστών να μηδενίζονται και τελικά λίγες συμμεταβλητές (αυτές με τους μη μηδενικούς συντελεστές) να βρίσκονται στο τελικό μοντέλο. Η μέθοδος είναι ιδιαίτερα χρήσιμη σε προβλήματα όπου το πλήθος των παραμέτρων p σε ένα μοντέλο είναι αρκετά μεγαλύτερο από το πλήθος N των παρατηρήσεων που διαθέτουμε. Θα θέλαμε λοιπόν στο τελικό μας μοντέλο να περιλαμβάνονται μόνο οι μεταβλητές εκείνες, οι οποίες ερμηνεύουν και σχετίζονται ουσιαστικά με τη μεταβλητή απόκρισης. Με τη μέθοδο Lasso πετυχαίνουμε τελικά να έχουμε ένα φειδωλό τελικό μοντέλο, το οποίο αρκετά συχνά θα περιέχει τις μεταβλητές που συνεισφέρουν ουσιαστικά στην ερμηνεία της μεταβλητής που μελετάμε. Επίσης πολλές φορές το μοντέλο αυτό θα έχει καλύτερη



Σχήμα 1.2: (Αριστερά) Ο ρόμβος είναι το σύνολο περιορισμού της Lasso. (Δεξιά) Ο κυκλικός δίσκος είναι το σύνολο περιορισμού της Ridge. Το σημείο $\hat{\beta}$ που βρίσκεται στο κέντρο των ελλείψεων είναι η εκτιμήτρια ελαχίστων τετραγώνων. Τα σημεία τομής του περιγράμματος της έλλειψης με το ρόμβο και τον κυκλικό δίσκο, αποτελούν τις εκτιμήτριες Lasso και Ridge αντίστοιχα. (Phg : *The Elements of Statistical Learning,* sel. 71).

προβλεπτική ικανότητα σε σχέση με το μοντέλο που προκύπτει από τη μέθοδο ελαχίστων τετραγώνων. Στη συνέχεια της εργασίας, θα αναλύσουμε ορισμένες γενικεύσεις και επεκτάσεις της μεθόδου Lasso.

Kef^laio 2

GenikeÔseic thc mejìdou Lasso

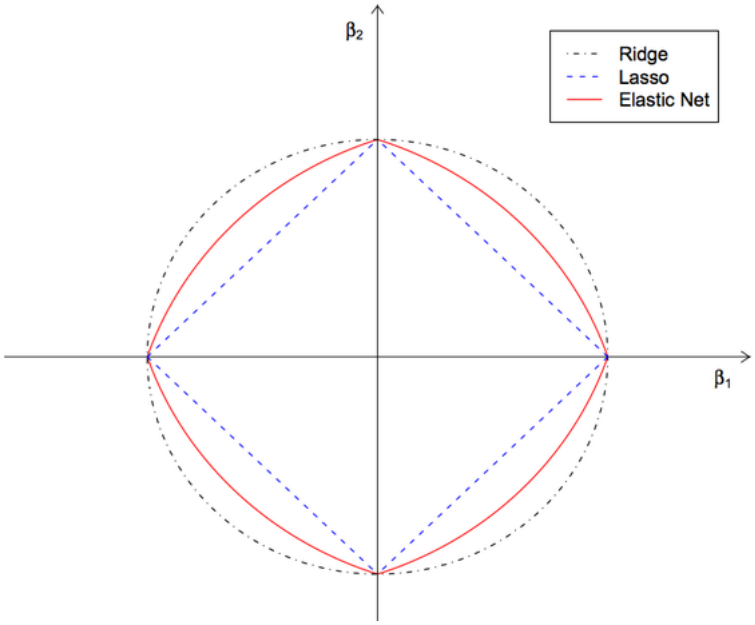
2.1 Εισαγωγή

Στο προηγούμενο Κεφάλαιο έγινε μια εισαγωγή στη μέθοδο Lasso και είδαμε πως αυτή μπορεί να χρησιμοποιηθεί σε προβλήματα παλινδρόμησης, για την εκτίμηση των συντελεστών των ανεξάρτητων μεταβλητών και την επιλογή των σημαντικών παραγόντων σε ένα γραμμικό μοντέλο. Στο Κεφάλαιο αυτό παρουσιάζουμε ορισμένες γενικεύσεις και επεκτάσεις της μεθόδου, που χρησιμοποιούνται συχνά, προκειμένου να βελτιώσουν ορισμένα μειονεκτήματά της και να επεκταθούν και σε άλλες περιπτώσεις προβλημάτων, στα οποία η μέθοδος Lasso ενδέχεται να αποτύχει. Για παράδειγμα, σε προβλήματα όπου υπάρχει υψηλή συσχέτιση μεταξύ των ανεξάρτητων μεταβλητών, είναι προτιμότερη η χρήση της μεθόδου Elastic Net, η οποία συνδυάζει τις ποινές των μεθόδων Lasso και Ridge, επιλέγοντας ή όχι τις συσχετισμένες μεταβλητές μαζί (grouping e ect). Σε άλλα προβλήματα ενδέχεται οι παράγοντες να είναι δομημένοι σε ομάδες. Ομάδες μπορούν να προκύψουν με τη χρήση κατηγορικών μεταβλητών, οι οποίες εισάγονται στο μοντέλο μέσω κάποιων ψευδομεταβλητών (dummy variables). Η μέθοδος Group Lasso συχνά χρησιμοποιείται για την επιλογή ή όχι, ολόκληρων γκρουπ ανεξάρτητων μεταβλητών στο τελικό μοντέλο. Επίσης σε προβλήματα όπου οι ανεξάρτητες μεταβλητές μπορούν να διαταχθούν με κάποιο τρόπο (π.χ. ως προς το χρόνο) τότε μπορεί να χρησιμοποιηθεί και η μέθοδος Fused Lasso. Όπως είδαμε, η μέθοδος Lasso μπορεί να χρησιμοποιηθεί ταυτόχρονα και για εκτίμηση των συντελεστών αλλά και για την επιλογή των μεταβλητών σε ένα μοντέλο. Επομένως, προκειμένου να επιλέξει ένα μοντέλο το οποίο θα είναι φειδωλό αλλά ταυτόχρονα θα έχει και υψηλή προβλεπτική ικανότητα, μπορεί τελικά να συμπεριλάβει επιπλέον παράγοντες (που αποτελούν «θόρυβο») στο τελικό μοντέλο. Η μέθοδος Adaptive Lasso και τα κριτήρια που περιέχουν μη κυρτούς όρους ποινής, χρησιμοποιούνται για να διορθώσουν αυτή την αδυναμία της μεθόδου και συνήθως παράγουν ακόμη πιο αραιά και ακριβή μοντέλα. Στη συνέχεια του Κεφαλαίου θα αναλύσουμε περαιτέρω όλες τις παραπάνω επεκτάσεις της μεθόδου Lasso.

2.2 Elastic Net

Η μέθοδος Elastic Net (Zou and Hastie, 2005) εφαρμόζεται κυρίως σε προβλήματα όπου υπάρχει υψηλή συσχέτιση μεταξύ των ανεξάρτητων μεταβλητών σε ένα γραμμικό μοντέλο. Από εμπειρικές μελέτες προκύπτει ότι όταν έχουμε δεδομένα στα οποία οι παράγοντες έχουν υψηλή συσχέτιση, τότε η μέθοδος Ridge έχει καλύτερα αποτελέσματα από τη μέθοδο Lasso, όσον αφορά την προβλεπτική ικανότητα του μοντέλου. Επίσης σε ομάδες υψηλά συσχετισμένων μεταβλητών η μέθοδος

Lasso επιλέγει συνήθως μία από αυτές τις μεταβλητές χωρίς να εξετάζει ποια από αυτές είναι κατάλληλη. Επιπλέον, στις περιπτώσεις όπου ο αριθμός των παραγόντων p είναι μεγαλύτερος του αριθμού των παρατηρήσεων N, η μέθοδος Lasso επιλέγει το πολύ N το πλήθος επεξηγηματικές μεταβλητές. Αυτό αποτελεί έναν περιορισμό διότι στην πραγματικότητα μπορεί να χρειάζονται παραπάνω από N επεξηγηματικές μεταβλητές για την ερμηνεία της μεταβλητής απόκρισης. Η μέθοδος Elastic Net συνδυάζει τις μεθόδους Lasso και Ridge μέσω μιας κοινής ποινής. Χρησιμοποιώντας αυτή τη νέα ποινή καταλήγουμε τελικά σε ένα αραιό μοντέλο έχοντας επιλέξει τους σημαντικούς παράγοντες που σχετίζονται με την εξαρτημένη μεταβλητή. Αυτό έχει να κάνει με τον l_1 περιορισμό της μεθόδου Lasso. Επιπλέον πετυχαίνουμε μείωση των συντελεστών των μη σημαντικών μεταβλητών και έτσι βελτιώνεται η προβλεπτική ακρίβεια του μοντέλου. Αυτό οφείλεται στον l_2-περιορισμό της μεθόδου Ridge. Στο Σχήμα 2.1 βλέπουμε τα σύνολα περιορισμού που εφαρμόζονται σε κάθε μια από τις τρεις μεθόδους, όταν έχουμε δύο παράγοντες. Το σύνολο περιορισμού της μεθόδου Elastic Net διαθέτει τα γεωμετρικά χαρακτηριστικά και των δύο συνόλων Lasso, Ridge. Η μέθοδος Elastic Net δίνει λύση στο εξής πρόβλημα ελαχιστοποίησης:



Σχήμα 2.1: Τα σύνολα περιορισμού των μεθόδων Ridge-Lasso-Elastic Net στην περίπτωση των δύο παραγόντων. Ο περιορισμός Elastic Net (κόκκινο) συνδυάζει τα χαρακτηριστικά των συνόλων περιορισμού Ridge και Lasso.

$$\min_{(\beta_0,\ )2\mathrm{R}\times\mathrm{R}^p}\left\{\frac{1}{2}\sum_{i=1}^{N}\left(y_i \quad \beta_0 \quad \mathbf{x}_i^T\boldsymbol{\beta}\right)^2 + \lambda\left[\frac{1}{2}(1 \quad \alpha)k\boldsymbol{\beta}k_2^2 + \alpha k\boldsymbol{\beta}k_1\right]\right\}. \tag{2.2.1}$$

Η παράμετρος λ 0 είναι η παράμετρος συντονισμού και η νέα παράμετρος α 2 [0, 1] ουσιαστικά συνδυάζει τις ποινές των μεθόδων Lasso και Ridge. Για α = 1 το πρόβλημα ελαχιστοποίησης ανάγεται στο απλό πρόβλημα Lasso, ενώ για α = 0 καταλήγουμε στη μέθοδο Ridge. Η παράμετρος α συνήθως επιλέγεται με υποκειμενικά κριτήρια. Αν το α είναι κοντά στη μονάδα τότε η μέθοδος λειτουργεί περισσότερο σαν τη μέθοδο Lasso, ενώ αν το α είναι κοντά στο 0 τότε η μέθοδος λειτουργεί σαν τη Ridge. Η παράσταση που ελαχιστοποιεί το πρόβλημα (2.2.1) είναι κυρτή (ως

άθροισμα κυρτών συναρτήσεων) και η ποινή είναι διαχωρίσιμη ($h_j(\beta_j) = \lambda f \frac{1}{2}(1 \quad \alpha)\beta_j^2 + \alpha j \beta_j j g$, κυρτή αλλά μη διαφορίσιμη στο 0 για κάθε $j = 1, \ldots, p$), επομένως μπορεί να εφαρμοστεί ο αλγόριθμος Coordinate Descent για την εύρεση των συντελεστών των παραγόντων του μοντέλου. Μπορούμε να κεντράρουμε τις ανεξάρτητες μεταβλητές ώστε να έχουν μέση τιμή 0. Τότε προκύπτει ότι για τη σταθερά β_0 θα είναι $\hat{\beta}_0 = y$. Μένει λοιπόν να υπολογίσουμε και τις υπόλοιπες εκτιμήτριες $\hat{\beta}_j$ για $j = 1, \ldots, p$. Ορίζουμε το μερικό υπόλοιπο $r_{ij} = y_i \quad \hat{\beta}_0 \quad \sum_{k \neq j} \hat{\beta}_k x_{ik}$ ως προς τη μεταβλητή j για κάθε $j = 1, \ldots, p$. Θεωρούμε τη συνάρτηση ελαχιστοποίησης του (2.2.1), έστω $S(\boldsymbol{\beta})$ και παραγωγίζοντάς την ως προς β_j για κάθε $j = 1, \ldots, p$ (πρόκειται για το σύστημα p εξισώσεων KKT) και κρατώντας σταθερούς τους συντελεστές $\hat{\beta}_k$, για $k \neq j$ έχουμε:

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \beta_j} = 0 \;) \; \sum_{i=1}^{N}(r_{ij} \quad \beta_j x_{ij})(\quad x_{ij}) + \lambda(1 \quad \alpha)\beta_j + \lambda\alpha s_j = 0.$$

Άρα $\hat{\beta}_j = \frac{\sum_{i=1}^N r_{ij}x_{ij} \quad \lambda\alpha s_j}{\sum_{i=1}^N x_{ij}^2 + \lambda(1 \quad \alpha)}$, όπου s_j είναι η υποκλίση της $j\beta_j j$ ($s_j = \text{sign}(\beta_j)$, αν $\beta_j \neq 0$ και $s_j \in [\quad 1, 1]$, αν $\beta_j = 0$). Οπότε συνοπτικά θα έχουμε:

$$\hat{\beta}_j = \begin{cases} \frac{\sum_{i=1}^N r_{ij}x_{ij} \quad \lambda\alpha}{\sum_{i=1}^N x_{ij}^2 + \lambda(1 \quad \alpha)}, & \text{αν } \sum_{i=1}^N r_{ij}x_{ij} > \lambda\alpha \\ \frac{\sum_{i=1}^N r_{ij}x_{ij} + \lambda\alpha}{\sum_{i=1}^N x_{ij}^2 + \lambda(1 \quad \alpha)}, & \text{αν } \sum_{i=1}^N r_{ij}x_{ij} < \quad \lambda\alpha \\ 0, & \text{αν } \left|\sum_{i=1}^N r_{ij}x_{ij}\right| \quad \lambda\alpha \end{cases}$$

και με τη χρήση του τελεστή soft-thresholding:

$$\hat{\beta}_j = \frac{S_{\lambda\alpha}(\sum_{i=1}^N r_{ij}x_{ij})}{\sum_{i=1}^N x_{ij}^2 + \lambda(1 \quad \alpha)}. \tag{2.2.2}$$

2.2.1 Εφαρμογή

Προκειμένου να παρουσιάσουμε τα χαρακτηριστικά της μεθόδου Elastic net, θα προσομοιώσουμε τιμές από το γραμμικό μοντέλο

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Θα χρησιμοποιήσουμε δύο παραδείγματα. Στο πρώτο παράδειγμα θεωρούμε ότι το δείγμα μας αποτελείται από $N = 100$ παρατηρήσεις και από $p = 10$ παράγοντες. Επίσης για κάθε ανεξάρτητη μεταβλητή θεωρούμε ότι $X_j \quad N(0, 1)$ και ότι $\boldsymbol{\Sigma}_{i,j} = 0.9^{ji \quad jj}$, για κάθε $i, j = 1, \ldots, 10$, όπου $\boldsymbol{\Sigma}$ ο πίνακας διασποράς-συνδιασποράς. Για τα σφάλματα θεωρούμε ότι $\varepsilon_i \quad N(0, 1), i = 1, \ldots, 100$. Ο σταθερός όρος στο μοντέλο μας είναι $\beta_0 = 2$ και το διάνυσμα των παραμέτρων:

$$\boldsymbol{\beta} = (3, 0, 0, \quad 1, 0, 1.5, \quad 0.5, 0, 2, 0).$$

Για 11 διαφορετικές τιμές της παραμέτρου α στο [0, 1], προσαρμόζουμε το μοντέλο μας με βάση τις παρατηρήσεις που ανήκουν στο σύνολο train (80%) και χρησιμοποιούμε το σύνολο test (20%) για να εξετάσουμε την προβλεπτική ακρίβεια του κάθε μοντέλου. Τη διαδικασία αυτή την επαναλαμβάνουμε 100 φορές. Με χρήση του παρακάτω κώδικα αρχικά εγκαθιστούμε και φορτώνουμε τα πακέτα mvtnorm (για την προσομοίωση τιμών από την πολυμεταβλητή κανονική κατανομή) και glmnet (για την προσαρμογή του μοντέλου με τη μέθοδο Elastic Net). Για να είναι τα αποτελέσματά μας αναπαραγωγίσιμα θέτουμε ένα random.seed. Ύστερα δίνουμε τιμές για τον αριθμό των παρατηρήσεων N, τον αριθμό των παραγόντων p και το διάνυσμα των παραμέτρων beta. Εν συνεχεία κατασκευάζουμε το διάνυσμα alpha που θα περιέχει τιμές για την παράμετρο α. Αρχικοποιούμε τους πίνακες el_net_MAE και covariates. Ο πρώτος πίνακας θα περιέχει το μέσο απόλυτο σφάλμα (Mean Absolute Error) για κάθε ένα από τα 100 δείγματα που θα προσομοιώσουμε και για κάθε μία τιμή της παραμέτρου α. Όμοια ο πίνακας covariates θα περιέχει το πλήθος των ανεξάρτητων μεταβλητών που θα περιέχονται σε κάθε μοντέλο ανάλογα με την τιμή της παραμέτρου α. Επιπλέον χωρίζουμε τα δεδομένα μας σε train και test set με χρήση της εντολής sample. Επίσης κατασκευάζουμε τον πίνακα διασποράς-συνδιασποράς $\mathbf{\Sigma}$, όπως περιγράψαμε νωρίτερα. Εν συνεχεία μέσα στο βρόχο for και με χρήση της εντολής rmvnorm προσομοιώνουμε τιμές για τις επεξηγηματικές μεταβλητές μας. Ύστερα κατασκευάζουμε τιμές για την μεταβλητή απόκρισης y, με βάση το γραμμικό μας μοντέλο. Μέσα σε ένα νέο βρόχο for, για τις 11 τιμές της παραμέτρου α, προσαρμόζουμε το μοντέλο με βάση τις παρατηρήσεις που ανήκουν στο σύνολο train και εκτελούμε 10-fold Cross Validation για τον υπολογισμό της παραμέτρου λ που ελαχιστοποιεί το σφάλμα cross-validation. Αυτό επιτυγχάνεται με χρήση της εντολής cv.glmnet. Στη συνέχεια υπολογίζουμε το διάνυσμα των συντελεστών (χωρίς τη σταθερά) για την τιμή της παραμέτρου ποινής λ_{min}. Στο διάνυσμα el_net.pred αποθηκεύουμε τις προβλέψεις για τη μεταβλητή απόκρισης που είναι υπολογισμένες για τις παρατηρήσεις που ανήκουν στο σύνολο test (όρισμα newx=x[test,]). Τέλος, υπολογίζουμε το μέσο απόλυτο σφάλμα καθώς και το πλήθος των ανεξάρτητων μεταβλητών που περιέχονται στο εκάστοτε μοντέλο.

```
> install.packages("mvtnorm")
> install.packages("glmnet")
> library(mvtnorm)
> library(glmnet)

> #----------------
> #1st simulation
> set.seed(1)
> N=100
> p=10
> beta=c(3,0,0,-1,0,1.5,-0.5,0,2,0)
> alpha=seq(0,1,by=0.1)
> el_net_MAE=matrix(0,100,11)
> covariates=matrix(0,100,11)
> train=sample(1:N,0.8*N)
> test=-train
> sigma=matrix(0,p,p)
> for(i in 1:p){
    for(j in 1:p){
      sigma[i,j]=0.9^(abs(i-j))
```

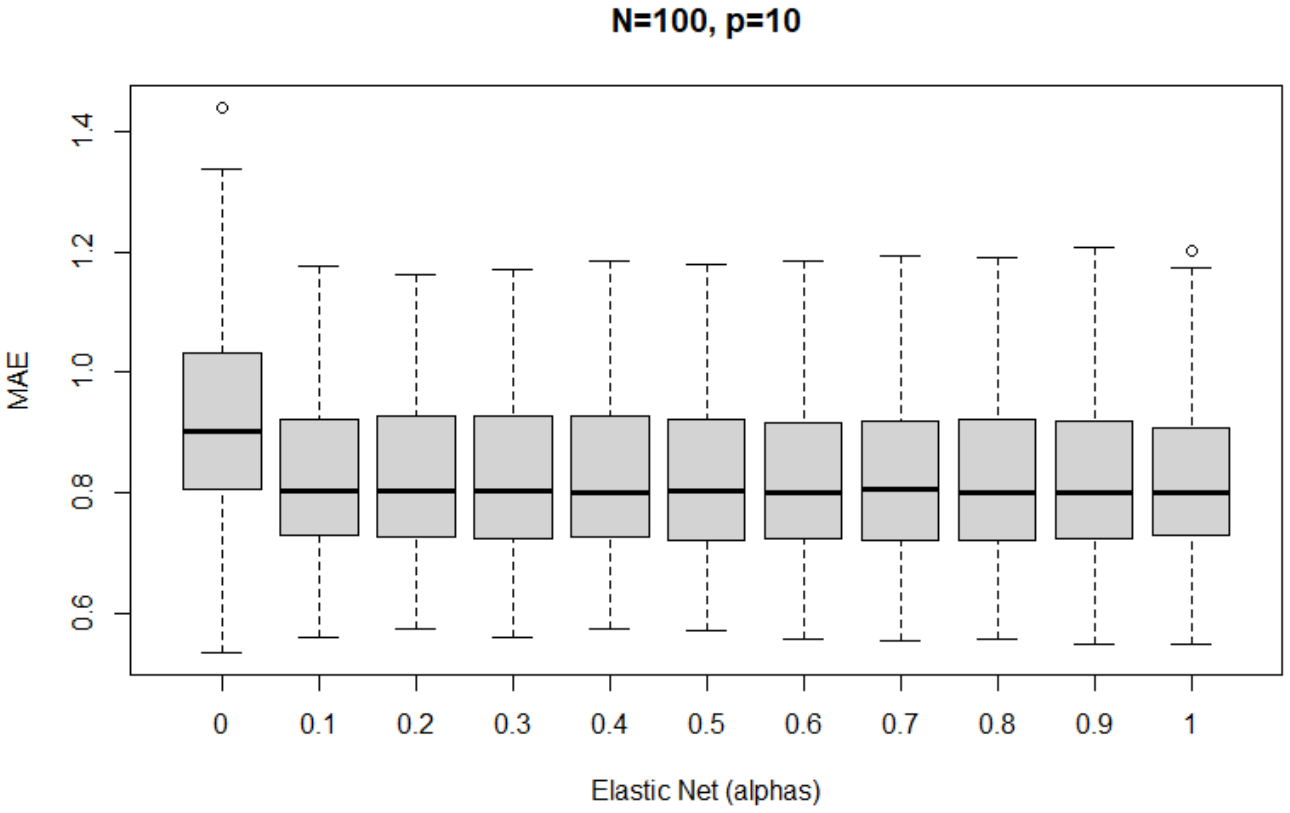
```
    }
  }
> for(k in 1:100){
   x=rmvnorm(N,mean=rep(0,p),sigma = sigma)
   y=2+x%*%beta+rnorm(N)

   #for alphas in [0,1] fit elastic net
   for(i in 1:11){
    el_net=cv.glmnet(x[train,],y[train],alpha=alpha[i])
    el_net.coef=coef(el_net,s="lambda.min")[-1]
    el_net.pred=predict(el_net,s=el_net$lambda.min,newx=x[test,])
    el_net_MAE[k,i]=mean(abs(y[test]-el_net.pred))
    covariates[k,i]=length(which(sapply(el_net.coef,
                                function(x) x!=0)))
   }
  }
```

Εφόσον έχουμε υπολογίσει τους πίνακες el_net_MAE και covariates, θα κατασκευάσουμε τα θηκοδιαγράμματα (boxplots) για το μέσο απόλυτο σφάλμα, για κάθε μία τιμή της παραμέτρου α που χρησιμοποιήσαμε. Με βάση τον παρακάτω κώδικα κατασκευάζουμε το Διάγραμμα 2.2.

```
> boxplot(el_net_MAE,xaxt="n")
> title(main="N=100, p=10",x="Elastic Net (alphas)",y="MAE")
> axis(side=1,at=c(1:11),labels=seq(0,1,by=0.1))

> apply(el_net_MAE,2,median)
 [1] 0.9012343 0.8032757 0.8036093 0.8023998 0.7996146 0.8020319
 [7] 0.7992082 0.8050156 0.8004113 0.8011426 0.8009798

> apply(covariates,2,median)
 [1] 10 10  9  9  8  8  8  8  7  7  8
```

Σε αυτό το Διάγραμμα παρατηρούμε ότι η προσαρμογή του μοντέλου με χρήση της μεθόδου Ridge (α = 0), δεν δίνει τόσο καλές προβλέψεις για τη μεταβλητή απόκρισης, αφού η τιμή της διαμέσου για το σφάλμα είναι σαφώς υψηλότερη από τις υπόλοιπες διαμέσους των σφαλμάτων. Για τη μέθοδο Lasso (α = 1) βλέπουμε ότι η διάμεσος του σφάλματος είναι μικρότερη από τη μέθοδο Ridge. Όμως αν συγκρίνουμε τις διαμέσους των σφαλμάτων και για τις υπόλοιπες τιμές του α, παρατηρούμε ότι η ελάχιστη τιμή 0.7992082 αντιστοιχεί στη μέθοδο Elastic Net με α = 0.6. Επίσης από τα παραπάνω αποτελέσματα βλέπουμε ότι η διάμεσος του πλήθους των επεξηγηματικών μεταβλητών που περιέχονται στο μοντέλο για τη μέθοδο Ridge είναι 10 και για τη μέθοδο Lasso είναι 8. Αυτό είναι αναμενόμενο αφού η μέθοδος Ridge δεν κάνει επιλογή μεταβλητών αρά αναμένουμε και οι 10 παράγοντες να συμπεριλαμβάνονται στο τελικό μοντέλο. Επίσης, αφού η μέθοδος Lasso χρησιμοποιείται και για επιλογή μεταβλητών περιμένουμε να επιλέγει τελικά λιγότερες από τις αρχικά διαθέσιμες επεξηγηματικές μεταβλητές. Και η μέθοδος Elastic Net για α = 0.6, δίνει επίσης διάμεσο ίση με 8 για το πλήθος των ανεξάρτητων μεταβλητών που συμπεριλαμβάνονται στο μοντέλο.

Στο δεύτερο παράδειγμα εξετάζουμε την περίπτωση $p > N$. Προσομοιώνουμε 100 δείγματα μεγέθους N = 20 παρατηρήσεων και p = 30 ανεξάρτητων μεταβλητών. Θέτουμε και πάλι τις ίδιες



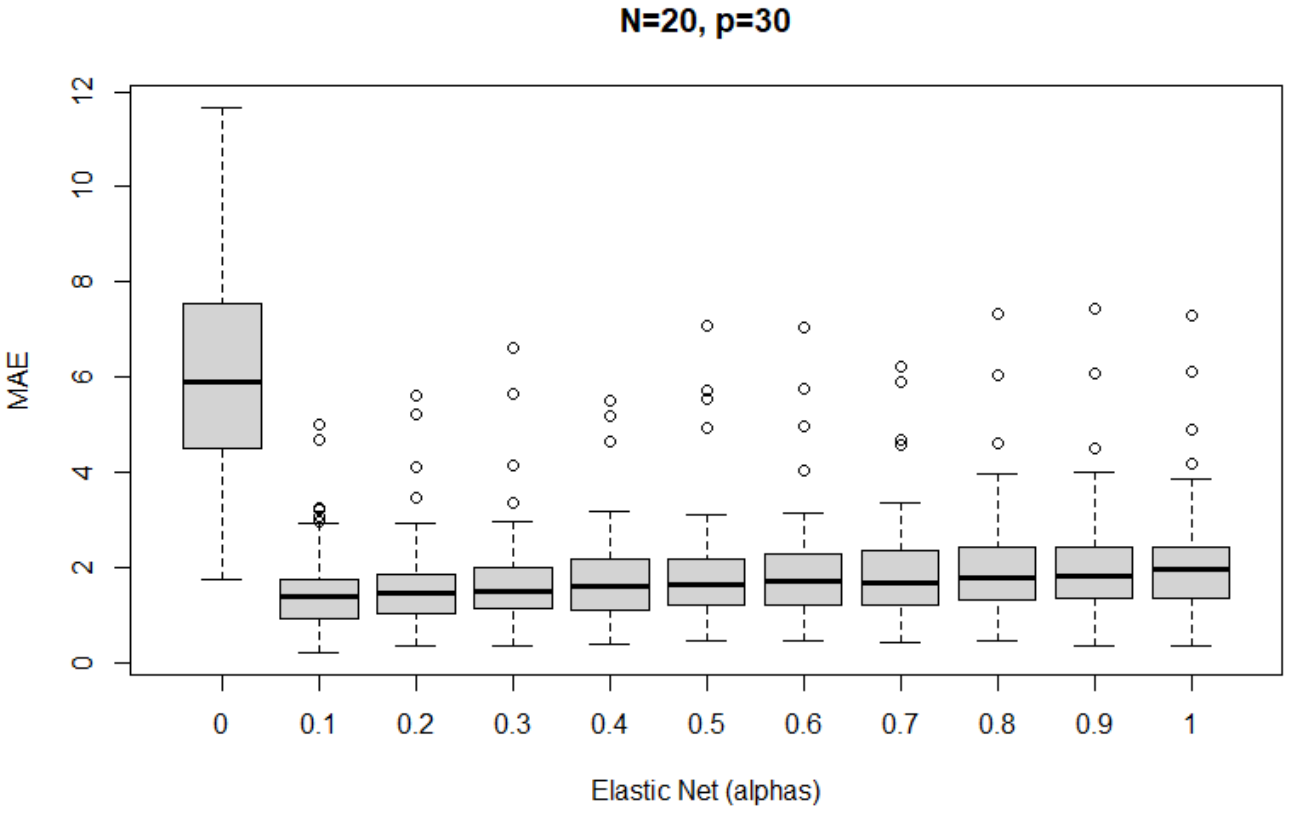
Διάγραμμα 2.2: Θηκοδιαγράμματα του μέσου απολύτου σφάλματος για 11 τιμές της παραμέτρου α της μεθόδου Elastic Net. (Περίπτωση $N = 100, p = 10$)

συνθήκες με πριν, δηλαδή $X_j \quad N(0,1)$ για $j = 1,\ldots,p$. Ο πίνακας διασποράς-συνδιασποράς είναι $\Sigma_{i,j} = 0.9^{|i \quad j|}$, για $i,j = 1,\ldots,p$ και για τα σφάλματα ισχύει $\varepsilon_i \quad N(0,1), i = 1,\ldots,N$. Επίσης θεωρούμε ότι ο σταθερός όρος στο μοντέλο μας είναι $\beta_0 = 1$ και το διάνυσμα των παραμέτρων:

$$\boldsymbol{\beta} = (\underbrace{1.5,\ldots,1.5}_{10},\underbrace{0,\ldots,0}_{10},\underbrace{0.5,\ldots,\quad 0.5}_{10}).$$

Χρησιμοποιώντας τις ίδιες εντολές με το προηγούμενο παράδειγμα (αλλάζοντας φυσικά όπου χρειάζεται τις αντίστοιχες τιμές π.χ. για τα N, p, beta), προσαρμόζουμε το μοντέλο μας για τις 11 διαφορετικές τιμές της παραμέτρου α *2* [0, 1] και υπολογίζουμε τους πίνακες el_net_MAE και covariates. Στη συνέχεια κατασκευάζουμε το Διάγραμμα 2.3. Στο Διάγραμμα αυτό παρουσιάζονται τα θηκοδιαγράμματα για το μέσο απόλυτο σφάλμα, για κάθε μία τιμή της παραμέτρου α που χρησιμοποιήσαμε. Παρατηρούμε ότι και σε αυτήν την περίπτωση, το μοντέλο που προκύπτει με χρήση της μεθόδου Ridge δεν έχει καλή προβλεπτική ικανότητα σε σχέση με τα υπόλοιπα μοντέλα. Η ελάχιστη διάμεσος για το σφάλμα, προκύπτει με χρήση της μεθόδου Elastic Net για $\alpha = 0.1$, ενώ η διάμεσος του σφάλματος για τη μέθοδο Lasso είναι η δεύτερη μεγαλύτερη μετά τη μέθοδο Ridge. Οι διάμεσοι των σφαλμάτων καθώς και του πλήθους των επεξηγηματικών μεταβλητών που περιέχονται στο μοντέλο για κάθε τιμή του α φαίνονται παρακάτω.

```
> apply(el_net_MAE,2,median)
```



Διάγραμμα 2.3: Θηκοδιαγράμματα του μέσου απολύτου σφάλματος για 11 τιμές της παραμέτρου α της μεθόδου Elastic Net. (Περίπτωση $N = 20, p = 30$)

```
 [1] 5.893449 1.248830 1.407449 1.479308 1.550380 1.581548
 [7] 1.704253 1.750621 1.771033 1.736906 1.845559
> apply(covariates,2,median)
 [1] 30 21 19 18 17 16 15 14 14 13 12
```

Από τα δύο προσομοιωμένα μας παραδείγματα συμπεραίνουμε ότι η μέθοδος Elastic Net για κατάλληλη τιμή της παραμέτρου α μπορεί να ξεπεράσει τις μεθόδους Ridge και Lasso, όσον αφορά την προβλεπτική ακρίβεια του μοντέλου μας. Επίσης είδαμε ότι μπορεί να παράξει και αραιά μοντέλα, επιλέγοντας ένα υποσύνολο από τις αρχικά διαθέσιμες επεξηγηματικές μεταβλητές.

2.3 Group Lasso

Σε αρκετά προβλήματα παλινδρόμησης ενδέχεται οι ανεξάρτητες μεταβλητές να είναι δομημένες σε ομάδες. Χαρακτηριστικό παράδειγμα είναι η δομή ομάδων που προκύπτει όταν η μεταβλητή απόκρισης δεν εξαρτάται μόνο από συνεχείς μεταβλητές, αλλά εξαρτάται και από κατηγορικές μεταβλητές. Κάθε κατηγορική μεταβλητή έχει ορισμένες κατηγορίες-επίπεδα. Για παράδειγμα αν μια μεταβλητή εκφράζει το επίπεδο ετησίου εισοδήματος των υπαλλήλων μιας εταιρείας, τότε αυτή μπορεί να αναλυθεί μέσω των κατηγοριών: χαμηλό, μεσαίο και υψηλό, αντίστοιχα επίπεδο εισοδήματος για κάθε υπάλληλο. Στη γενική περίπτωση όπου διαθέτουμε πολλές κατηγορικές μεταβλητές, μας ενδιαφέρει

να δούμε ποιές από αυτές τελικά επηρεάζουν ή όχι τη μεταβλητή απόκρισης. Αρχικά για να τις συμπεριλάβουμε στο γραμμικό μας μοντέλο θα πρέπει να τις κωδικοποιήσουμε χρησιμοποιώντας τις λεγόμενες εικονικές μεταβλητές (ή ψευδομεταβλητές). Στο παράδειγμά μας, αν Z είναι το επίπεδο εισοδήματος ενός υπαλλήλου τότε μπορούμε να χρησιμοποιήσουμε το διάνυσμα $Z = (Z_1, Z_2, Z_3)$ των ψευδομεταβλητών Z_1, Z_2, Z_3, όπου κάθε μία εκφράζει ένα από τα επίπεδα εισοδήματος: χαμηλό, μεσαίο και υψηλό αντίστοιχα. Δηλαδή χρησιμοποιώντας δείκτριες συναρτήσεις για κάθε επίπεδο θα έχουμε[1]:

$$Z_1 = \begin{cases} 1, & \text{χαμηλό} \\ 0, & \text{διαφορετικά} \end{cases} \quad Z_2 = \begin{cases} 1, & \text{μεσαίο} \\ 0, & \text{διαφορετικά} \end{cases} \quad Z_3 = \begin{cases} 1, & \text{υψηλό} \\ 0, & \text{διαφορετικά} \end{cases} .$$

Ας θεωρήσουμε την απλή περίπτωση όπου η μεταβλητή απόκρισης, έστω Y εξαρτάται μόνο από αυτή την κατηγορική μεταβλητή και άλλη μία συνεχή μεταβλητή, έστω W. Τότε το γραμμικό μοντέλο μπορεί να γραφεί ως εξής:

$$E[Y|W, Z] = \beta W + Z\boldsymbol{\theta}, \tag{2.3.1}$$

όπου $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^T$ το διάνυσμα των συντελεστών για τις εικονικές μεταβλητές. Τότε είναι λογικό να θεωρήσουμε ότι αν η κατηγορική μεταβλητή Z δε συνεισφέρει στην επεξήγηση της Y, το διάνυσμα $\boldsymbol{\theta}$ θα είναι το μηδενικό. Αντιθέτως αν αυτή η μεταβλητή εξηγεί τη μεταβλητή απόκρισης, τότε περιμένουμε το διάνυσμα $\boldsymbol{\theta}$ να έχει όλες τις συνιστώσες του μη μηδενικές. Στη γενική περίπτωση όπου διαθέτουμε J κατηγορικές μεταβλητές, θέλουμε να δούμε ποιές από αυτές είναι σημαντικές για την ερμηνεία της εξαρτημένης μεταβλητής. Οπότε είναι λογικό είτε να συμπεριλάβουμε ολόκληρα γκρουπ (εκτιμώντας τους συντελεστές σε κάθε γκρουπ ως μη μηδενικούς) είτε να μην χρησιμοποιήσουμε ορισμένες ομάδες στο τελικό μοντέλο (εκτιμώντας τους συντελεστές τους ως μηδενικούς). Η μέθοδος Group Lasso (Yuan and Lin, 2006) χρησιμοποιείται για τέτοιου είδους προβλήματα, δηλαδή προβλήματα στα οποία οι ανεξάρτητες μεταβλητές μπορούν με κάποιο τρόπο να ομαδοποιηθούν. Στη συνέχεια δίνουμε τον ορισμό του κριτηρίου που χρησιμοποιεί αυτή η μέθοδος και προχωράμε στον υπολογισμό της εκτιμήτριας.

2.3.1 Υπολογισμός της εκτιμήτριας Group Lasso

Ας θεωρήσουμε ότι οι ανεξάρτητες μεταβλητές είναι δομημένες σε J ομάδες. Τότε η μέθοδος Group Lasso επιλύει το εξής πρόβλημα ελαχιστοποίησης:

$$\min_{(\theta_1,\ldots,\theta_J)} \left\{ \frac{1}{2} \|\mathbf{y} - \sum_{j=1}^{J} \mathbf{Z}_j \boldsymbol{\theta}_j\|_2^2 + \lambda \sum_{j=1}^{J} \sqrt{p_j} \|\boldsymbol{\theta}_j\|_2 \right\} \tag{2.3.2}$$

όπου $\mathbf{y} \in \mathbb{R}^N$ το διάνυσμα των τιμών της εξαρτημένης μεταβλητής, $\mathbf{Z}_j \in \mathbb{R}^{N \times p_j}$ ο πίνακας που περιέχει τις ανεξάρτητες μεταβλητές που ανήκουν στην ομάδα j, $\boldsymbol{\theta}_j \in \mathbb{R}^{p_j}$ το διάνυσμα των συντελεστών που αντιστοιχούν στην ομάδα j και p_j είναι το πλήθος στοιχείων της ομάδας j, για κάθε $j = 1, \ldots, J$. Πάλι μπορούμε να κεντράρουμε τις τιμές των ανεξάρτητων μεταβλητών και της

[1] Sto sugkekrimèno par^deigma, gia k^je epÛpedo thc kathgorik c mac metablht c qrhsimopoioÔme kai apì mða eikonik metablht . 'Etsi sto montèlo mac de qrei^zetai na sumperil^boume to stajerì ìro. Enallaktik^ ja mporoÔsame na qrhsimopoi soume dÔo mìno yeudometablhtèc, p.q gia tic kathgorÛec qamhlì kai mesaÛo eisìdhma. Se aut thn perÛptwsh ja eÛqame diaforetik ermhneÛa gia touc suntelestèc tou montèlou, all^ kai ta dÔo montèla ja tan isodÔnama.

μεταβλητής απόκρισης, οπότε η σταθερά παραλείπεται από το πρόβλημα ελαχιστοποίησης. Η μέθοδος λειτουργεί σαν την απλή μέθοδο Lasso αλλά τώρα σε επίπεδο ομάδων. Για κατάλληλες τιμές της παραμέτρου λ υπάρχει περίπτωση ένα ολόκληρο γκρουπ παραγόντων να μη συμπεριλαμβάνεται στο τελικό μοντέλο. Ο συντελεστής $\sqrt{p_j}$ στην ποινή της μεθόδου είναι χρήσιμος στην περίπτωση όπου οι ομάδες δεν έχουν το ίδιο πλήθος παραγόντων. Έτσι χρησιμοποιώντας αυτόν το συντελεστή ουσιαστικά κάθε ομάδα συντελεστών ποινικοποιείται ανάλογα με το μέγεθός της. Αν κάθε γκρουπ αποτελείται μόνο από έναν παράγοντα, δηλαδή όταν $p_j = 1$, για κάθε $j = 1, \ldots, J$, τότε $\|\boldsymbol{\theta}_j\|_2 = |\theta_j|$ και η μέθοδος Group Lasso ισοδυναμεί με την απλή μέθοδο Lasso. Για την ελαχιστοποίηση του (2.3.2) θεωρούμε για κάθε $j = 1, \ldots, J$ τις εξισώσεις (zero-subgradient equations):

$$\mathbf{Z}_j^T(\mathbf{y} - \sum_{j=1}^{J} \mathbf{Z}_j\boldsymbol{\theta}_j) + \lambda\mathbf{s}_j = \mathbf{0} \tag{2.3.3}$$

όπου το $\mathbf{s}_j \in \mathbb{R}^{p_j}$ είναι το διάνυσμα υποκλίσης που ανήκει στο υποδιαφορικό του $\sqrt{p_j}\|\boldsymbol{\theta}_j\|_2$. Αν $\boldsymbol{\theta}_j \neq \mathbf{0}$ τότε $\mathbf{s}_j = \sqrt{p_j}\frac{\boldsymbol{\theta}_j}{\|\boldsymbol{\theta}_j\|_2}$, ενώ για $\boldsymbol{\theta}_j = \mathbf{0}$ αφού το $\mathbf{s}_j$ είναι η υποκλίση της $\sqrt{p_j}\|\boldsymbol{\theta}_j\|_2$ στο $\mathbf{0}$, θα έχουμε (από τον ορισμό της υποκλίσης) $\sqrt{p_j}\|\boldsymbol{\theta}_j\|_2 \geq \mathbf{s}_j^T\boldsymbol{\theta}_j$ για κάθε $\boldsymbol{\theta}_j \in \mathbb{R}^{p_j}$, οπότε θέτοντας $\boldsymbol{\theta}_j = \mathbf{s}_j$ προκύπτει $\|\mathbf{s}_j\|_2 \leq \sqrt{p_j}$. Θεωρώντας τώρα το μερικό υπόλοιπο

$$\mathbf{r}_j = \mathbf{y} - \sum_{k \neq j} \mathbf{Z}_k\hat{\boldsymbol{\theta}}_k, \quad j = 1, \ldots, J$$

ως προς την ομάδα j (κρατάμε σταθερά τα διανύσματα $\hat{\boldsymbol{\theta}}_k$, για $k \neq j$), τότε θα έχουμε:

$$\mathbf{Z}_j^T(\mathbf{r}_j - \mathbf{Z}_j\boldsymbol{\theta}_j) + \lambda\mathbf{s}_j = \mathbf{0}.$$

Από τις συνθήκες που ισχύουν για την υποκλίση $\mathbf{s}_j$ θα είναι

$$\hat{\boldsymbol{\theta}}_j = \mathbf{0} \ , \ \|\mathbf{s}_j\|_2 < \sqrt{p_j} \ \Rightarrow \ \|\mathbf{Z}_j^T\mathbf{r}_j\|_2 < \lambda\sqrt{p_j}, \tag{2.3.4}$$

διαφορετικά θα είναι $\mathbf{s}_j = \sqrt{p_j}\frac{\boldsymbol{\theta}_j}{\|\boldsymbol{\theta}_j\|_2}$ οπότε:

$$\left(\mathbf{Z}_j^T\mathbf{Z}_j + \frac{\lambda\sqrt{p_j}}{\|\boldsymbol{\theta}_j\|_2}\mathbf{I}_{p_j}\right)\boldsymbol{\theta}_j = \mathbf{Z}_j^T\mathbf{r}_j$$

και αν ο πίνακας είναι αντιστρέψιμος τελικά θα έχουμε:

$$\hat{\boldsymbol{\theta}}_j = \left(\mathbf{Z}_j^T\mathbf{Z}_j + \frac{\lambda\sqrt{p_j}}{\|\hat{\boldsymbol{\theta}}_j\|_2}\mathbf{I}_{p_j}\right)^{-1}\mathbf{Z}_j^T\mathbf{r}_j. \tag{2.3.5}$$

Όμως παρατηρούμε ότι η εκτιμήτρια $\hat{\boldsymbol{\theta}}_j$ εξαρτάται από την ποσότητα $\|\hat{\boldsymbol{\theta}}_j\|_2$. Δεν υπάρχει κλειστή μορφή της λύσης εκτός αν ο πίνακας $\mathbf{Z}_j$ είναι ορθοκανονικός, δηλαδή ισχύει $\mathbf{Z}_j^T\mathbf{Z}_j = \mathbf{I}_{p_j}$ τότε η εκτιμήτρια παίρνει την εξής μορφή:

$$\hat{\boldsymbol{\theta}}_j = \left(\mathbf{I}_{p_j} + \frac{\lambda\sqrt{p_j}}{\|\hat{\boldsymbol{\theta}}_j\|_2}\mathbf{I}_{p_j}\right)^{-1}\mathbf{Z}_j^T\mathbf{r}_j = \left(1 + \frac{\lambda\sqrt{p_j}}{\|\hat{\boldsymbol{\theta}}_j\|_2}\right)^{-1}\mathbf{Z}_j^T\mathbf{r}_j$$

$$= \frac{\|\hat{\boldsymbol{\theta}}_j\|_2}{\lambda\sqrt{p_j} + \|\hat{\boldsymbol{\theta}}_j\|_2}\mathbf{Z}_j^T\mathbf{r}_j$$

και παίρνοντας νόρμα και στα δύο μέλη:

$$\|\hat{\boldsymbol{\theta}}_j\|_2 = \frac{\|\hat{\boldsymbol{\theta}}_j\|_2}{\lambda\sqrt{p_j} + \|\hat{\boldsymbol{\theta}}_j\|_2}\|\mathbf{Z}_j^T\mathbf{r}_j\|_2 \;\Big)\; \|\hat{\boldsymbol{\theta}}_j\|_2 = \|\mathbf{Z}_j^T\mathbf{r}_j\|_2 \quad \lambda\sqrt{p_j}$$

οπότε τελικά αντικαθιστώντας παίρνουμε:

$$\hat{\boldsymbol{\theta}}_j = \left(1 \quad \frac{\lambda\sqrt{p_j}}{\|\mathbf{Z}_j^T\mathbf{r}_j\|_2}\right)\mathbf{Z}_j^T\mathbf{r}_j. \tag{2.3.6}$$

Συνδυάζοντας την τελευταία σχέση μαζί με την σχέση (2.3.4) και χρησιμοποιώντας τη συνάρτηση θετικό μέρος ($z_+ = \max\{z, 0\}$) μπορούμε τελικά να γράψουμε:

$$\hat{\boldsymbol{\theta}}_j = \left(1 \quad \frac{\lambda\sqrt{p_j}}{\|\mathbf{Z}_j^T\mathbf{r}_j\|_2}\right)_+ \mathbf{Z}_j^T\mathbf{r}_j. \tag{2.3.7}$$

Η λύση στο πρόβλημα (2.3.2) δηλαδή η εύρεση όλων των εκτιμητριών $\hat{\boldsymbol{\theta}}_j$ επιτυγχάνεται εφαρμόζοντας επαναληπτικά τη σχέση (2.3.7) για κάθε $j = 1, \ldots, J$. Επομένως σε κάθε επανάληψη τα block-διανύσματα $\{\hat{\boldsymbol{\theta}}_k, k \neq j\}$ παραμένουν σταθερά και ελαχιστοποιούμε ως προς $\boldsymbol{\theta}_j$. Η συνάρτηση ελαχιστοποίησης του προβλήματος (2.3.2) είναι κυρτή (ως άθροισμα κυρτών συναρτήσεων) και ο όρος ποινής είναι διαχωρίσιμος, επομένως η επαναληπτική διαδικασία θα συγκλίνει στη λύση του προβλήματος. Ο αλγόριθμος που εφαρμόζεται ονομάζεται Block Coordinate Descent. Σχηματικά λοιπόν θα είναι:

Block Coordinate Descent

Αρχικοποίηση: $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$ με $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\theta}_1^{(0)}, \ldots, \boldsymbol{\theta}_J^{(0)})$ αυθαίρετο.

Επανάληψη: (μέχρι σύγκλιση)

Για $j = 1, \ldots, J$:

- $\mathbf{r}_j = \mathbf{y} \quad \sum_{k \neq j} \mathbf{Z}_k\hat{\boldsymbol{\theta}}_k$
- Αν $\|\mathbf{Z}_j^T\mathbf{r}_j\|_2 \quad \lambda\sqrt{p_j}$ τότε $\hat{\boldsymbol{\theta}}_j = \mathbf{0}$
- Αλλιώς (αν $\mathbf{Z}_j^T\mathbf{Z}_j = \mathbf{I}_{p_j}$) τότε:

$$\hat{\boldsymbol{\theta}}_j = \left(1 \quad \frac{\lambda\sqrt{p_j}}{\|\mathbf{Z}_j^T\mathbf{r}_j\|_2}\right)\mathbf{Z}_j^T\mathbf{r}_j$$

Έξοδος: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J)$

Ξεκινάμε από την αρχικοποίηση του διανύσματος $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)} = (\boldsymbol{\theta}_1^{(0)}, \ldots, \boldsymbol{\theta}_J^{(0)})$ και στην πρώτη επανάληψη υπολογίζουμε το υπόλοιπο $\mathbf{r}_1 = \mathbf{y} \quad \sum_{k=1} \mathbf{Z}_k \boldsymbol{\theta}_k^{(0)}$. Αν ισχύει $k\mathbf{Z}_1^T \mathbf{r}_1 k_2 \quad \lambda \sqrt{p_1}$ τότε $\hat{\boldsymbol{\theta}}_1^{(1)} = \mathbf{0}$, αλλιώς αν $\mathbf{Z}_1^T \mathbf{Z}_1 = \mathbf{I}_{p_1}$ τότε θέτουμε $\hat{\boldsymbol{\theta}}_1^{(1)} = \left(1 \quad \frac{\lambda \sqrt{p_1}}{k\mathbf{Z}_1^T \mathbf{r}_1 k_2}\right) \mathbf{Z}_1^T \mathbf{r}_1$. Στη συνέχεια υπολογίζουμε το διάνυσμα $\hat{\boldsymbol{\theta}}_2^{(1)}$ κρατώντας σταθερά τα διανύσματα $(\hat{\boldsymbol{\theta}}_1^{(1)}, \boldsymbol{\theta}_2^{(0)}, \ldots, \boldsymbol{\theta}_J^{(0)})$ (πλέον χρησιμοποιούμε τη νέα τιμή $\hat{\boldsymbol{\theta}}_1^{(1)}$ που υπολογίσαμε στο πρώτο βήμα). Συνεχίζουμε μέχρι να υπολογίσουμε όλα τα διανύσματα οπότε καταλήγουμε σε μια πρώτη εκτιμήτρια, έστω $\boldsymbol{\theta}^{(1)} = (\hat{\boldsymbol{\theta}}_1^{(1)}, \ldots, \hat{\boldsymbol{\theta}}_J^{(1)})$. Επαναλαμβάνουμε τη διαδικασία μέχρι ο αλγόριθμος να συγκλίνει σε κάποια εκτιμήτρια, έστω $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J)$, η οποία θα είναι και η εκτιμήτρια Group Lasso. Το πακέτο gglasso στην R μπορεί να χρησιμοποιηθεί για την προσαρμογή του μοντέλου με τη μέθοδο Group Lasso.

2.3.2 Sparse Group Lasso

Η μέθοδος Group Lasso είδαμε ότι επιλέγει ολόκληρες ομάδες-γκρουπ ανεξάρτητων μεταβλητών προκειμένου να τις συμπεριλάβει ή όχι στο τελικό μοντέλο. Όταν μια ομάδα μεταβλητών συμπεριλαμβάνεται στο μοντέλο, τότε όλοι οι συντελεστές των παραγόντων που ανήκουν σε αυτή την ομάδα θα είναι μη μηδενικοί. Αυτό οφείλεται στο γεγονός ότι ποινικοποιούμε την l_2-νόρμα των συντελεστών σε κάθε γκρουπ. Ωστόσο, σε ορισμένες περιπτώσεις θα θέλαμε να υπάρχει αραιότητα και μέσα σε κάθε γκρουπ που τελικά επιλέγεται για την ερμηνεία της μεταβλητής απόκρισης. Η μέθοδος Sparse Group Lasso (Friedman et al, 2010) είναι σχεδιασμένη για να παράγει αραιές λύσεις και μέσα σε κάθε γκρουπ, συνδυάζοντας τους περιορισμούς l_1 και l_2 σε επίπεδο ομάδας. Η μέθοδος αυτή επιλύει το εξής πρόβλημα ελαχιστοποίησης:

$$\min_{f \ _j 2 \mathbb{R}^{p_j} g_{j=1}^J} \left\{ \frac{1}{2} \left\| \mathbf{y} \quad \sum_{j=1}^{J} \mathbf{Z}_j \boldsymbol{\theta}_j \right\|_2^2 + \lambda \sum_{j=1}^{J} [(1 \quad \alpha) k\boldsymbol{\theta}_j k_2 + \alpha k\boldsymbol{\theta}_j k_1] \right\} \tag{2.3.8}$$

όπου α 2 [0, 1] και $\lambda \quad 0$ η παράμετρος ποινής (για απλότητα στους συμβολισμούς μπορούμε να παραλείψουμε τους συντελεστές $\sqrt{p_j}$). Η ποινή εδώ μοιάζει με την ποινή που εφαρμόζεται στη μέθοδο Elastic Net, εδώ όμως ποινικοποιούμε ολόκληρες ομάδες συντελεστών. Όταν η παράμετρος α είναι 0 τότε η συνάρτηση ελαχιστοποίησης είναι ίδια με αυτή της μεθόδου Group Lasso, ένω για $\alpha = 1$ η μέθοδος ισοδυναμεί με την απλή μέθοδο Lasso. Η παράσταση ελαχιστοποίησης της (2.3.8) είναι κυρτή (ως άθροισμα κυρτών συναρτήσεων) και ο όρος ποινής είναι διαχωρίσιμος, οπότε για τον υπολογισμό της εκτιμήτριας μπορεί να εφαρμοστεί ο αλγόριθμος Block Coordinate Descent.

2.4 Fused Lasso

Η μέθοδος Fused Lasso (Tibshirani et al, 2005) χρησιμοποιείται σε περιπτώσεις όπου οι ανεξάρτητες μεταβλητές μπορούν να διαταχθούν με κάποιο φυσικό τρόπο (π.χ. ως προς το χρόνο). Αποτελεί μια γενίκευση της μεθόδου Lasso διότι εκτός από την l_1-νόρμα των συντελεστών του μοντέλου, ποινικοποιεί το άθροισμα των απολύτων τιμών των διαδοχικών διαφορών τους. Η μέθοδος είναι ιδιαίτερα χρήσιμη όταν ο αριθμός των παραγόντων p είναι αρκετά μεγαλύτερος του αριθμού των παρατηρήσεων N. Το κριτήριο ελαχιστοποίησης της μεθόδου Fused Lasso είναι το εξής:

$$\min_{2\mathbb{R}^p} \sum_{i=1}^{N} (y_i \quad \sum_{j=1}^{p} \beta_j x_{ij})^2, \quad \text{με} \sum_{j=1}^{p} j\beta_j j \quad s_1 \text{ και} \sum_{j=2}^{p} j\beta_j \quad \beta_{j \ 1} j \quad s_2$$

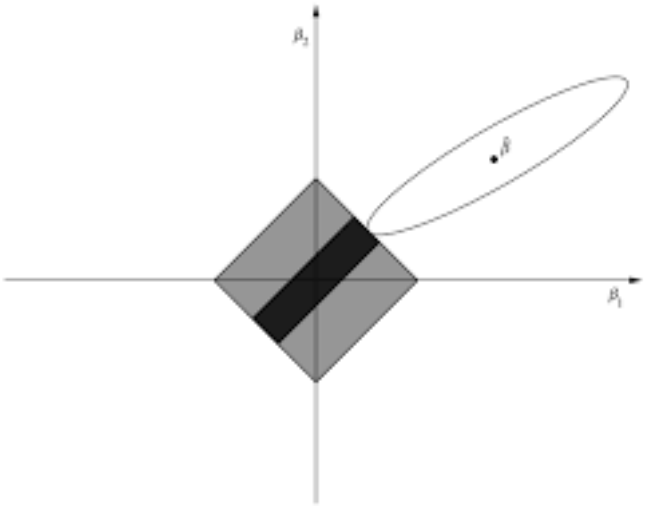
όπου έχουμε υποθέσει ότι οι τιμές των ανεξάρτητων μεταβλητών είναι τυποποιημένες και η εξαρτημένη μεταβλητή έχει μέση τιμή 0. Έτσι μπορούμε να παραλείψουμε τη σταθερά από το πρόβλημα ελαχιστοποίησης. Για κατάλληλες τιμές των s_1, s_2 0 ο πρώτος περιορισμός $\sum_{j=1}^{p} j\beta_j j$ s_1 ενισχύει την αραιότητα μέσα στο μοντέλο, ώστε να επιλεγεί τελικά ένα υποσύνολο από τους αρχικούς παράγοντες και ο δεύτερος περιορισμός $\sum_{j=2}^{p} j\beta_j$ β_j $_1 j$ s_2 περιορίζει τις διαφορές των γειτονικών συντελεστών με αποτέλεσμα κάποιοι από αυτούς να είναι ίδιοι. Επομένως το μονοπάτι των συντελεστών που προκύπτει από αυτή τη μέθοδο είναι ανά τμήματα σταθερό. Τα σύνολα περιορισμού για την απλή περίπτωση $p = 2$ παραγόντων φαίνονται στο Σχήμα 2.4. Το πρόβλημα μπορεί να γραφεί ισοδύναμα σε Λαγκρανζιανή μορφή ως εξής:

$$\min_{2R^p} \left\{ \sum_{i=1}^{N} \left(y_i \quad \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda_1 \sum_{j=1}^{p} j\beta_j j + \lambda_2 \sum_{j=2}^{p} j\beta_j \quad \beta_j \ _1 j \right\}, \tag{2.4.1}$$

όπου λ_1, λ_2 είναι οι παράμετροι ποινής. Στην ειδική περίπτωση όπου ο πίνακας σχεδιασμού είναι ο ταυτοτικός $\mathbf{X} = \mathbf{I}_N$, το πρόβλημα ελαχιστοποίησης γράφεται ως εξής:

$$\min_{2R^p} \left\{ \sum_{i=1}^{N} (y_i \quad \beta_i)^2 + \lambda_1 \sum_{i=1}^{N} j\beta_i j + \lambda_2 \sum_{i=2}^{N} j\beta_i \quad \beta_i \ _1 j \right\}, \tag{2.4.2}$$

και η λύση του ονομάζεται *Fused Lasso Signal Approximator (FLSA)*.



Σχήμα 2.4: Τα σύνολα περιορισμού στη μέθοδο Fused Lasso για την απλή περίπτωση $p = 2$ παραγόντων. Με γκρι χρώμα παριστάνεται το σύνολο $j\beta_1 j + j\beta_2 j$ s_1 και με μαύρο το σύνολο $j\beta_2 \quad \beta_1 j$ s_2. Το σημείο τομής τους με την έλλειψη του RSS αποτελεί την εκτιμήτρια Fused Lasso. (Phg : *Sparsity and smoothness via the fused lasso, Tibshirani et al, 2005*)

2.4.1 Προσαρμογή με τη μέθοδο Fused Lasso

Το πρόβλημα (2.4.2) αποτελεί ένα κυρτό πρόβλημα ελαχιστοποίησης άρα η λύση του είναι καλά ορισμένη. Συνήθως σε τέτοιου είδους προβλήματα χρησιμοποιούμε τον αλγόριθμο Coordinate Descent για την εύρεση των συντελεστών του μοντέλου. Όμως στη συγκεκριμένη περίπτωση ο

όρος ποινής είναι μη διαχωρίσιμος αφού η συνάρτηση $h(\boldsymbol{\beta}) = \sum |\beta_i - \beta_{i-1}|$ δε μπορεί να γραφεί στη μορφή $\sum h_j(\beta_j)$. Κατά συνέπεια ο αλγόριθμος Coordinate Descent δεν εγγυάται σύγκλιση. Για το λόγο αυτό έχουν δοθεί διαφορετικές προσεγγίσεις για την εκτίμηση των συντελεστών του μοντέλου με τη μέθοδο Fused Lasso. Αρχικά δίνουμε το παρακάτω Λήμμα, το οποίο αφορά τη δομή της εκτιμήτριας για το πρόβλημα (2.4.2).

Λήμμα 2 (Friedman et al, 2007). *Έστω $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ μια εκτίμήτρια που ελαχιστοποιεί το κριτήριο* (2.4.2). *Τότε για κάθε $\lambda_1' > \lambda_1$ ισχύει:*

$$\hat{\boldsymbol{\beta}}(\lambda_1', \lambda_2) = S_{\lambda_1' - \lambda_1}(\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)).$$

Αν θέσουμε $\lambda_1 = 0$ τότε προκύπτει:

$$\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2) = S_{\lambda_1}(\hat{\boldsymbol{\beta}}(0, \lambda_2)) \tag{2.4.3}$$

(όπου στη θέση του λ_1' θέσαμε λ_1 για απλότητα), επομένως αν υπολογίσουμε την εκτιμήτρια $\hat{\boldsymbol{\beta}}(0, \lambda_2)$ για την περίπτωση $\lambda_1 = 0$, τότε κάθε άλλη λύση $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ για $\lambda_1 > 0$ προκύπτει με εφαρμογή του τελεστή soft thresholding. Επομένως αρκεί να επιλύσουμε το εξής πρόβλημα ελαχιστοποίησης (και πάλι για απλότητα συμβολισμών θέτουμε λ αντί για λ_2):

$$\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{N} (y_i - \beta_i)^2 + \lambda \sum_{i=2}^{N} |\beta_i - \beta_{i-1}| \right\}. \tag{2.4.4}$$

Στη συνέχεια δίνουμε μια προσέγγιση στην επίλυση αυτού του προβλήματος με τη χρήση του δυναμικού προγραμματισμού.

2.4.2 Δυναμικός προγραμματισμός για τη μέθοδο Fused Lasso

Ο δυναμικός προγραμματισμός (dynamic programming) είναι μια τεχνική για την επίλυση υπολογιστικά δύσκολων προβλημάτων, τα οποία μπορούν να διασπαστούν σε υποπροβλήματα που είναι ευκολότερα στην επίλυση. Χρησιμοποιείται κυρίως σε προβλήματα βελτιστοποίησης (optimization). Στη περίπτωση της μεθόδου Fused Lasso σύμφωνα με τον Johnson (2013) για τον υπολογισμό της εκτιμήτριας στο πρόβλημα (2.4.4) η συνάρτηση ελαχιστοποίησης μπορεί να γράφει ως εξής:

$$f(\boldsymbol{\beta}) = (y_1 - \beta_1)^2 + \lambda|\beta_2 - \beta_1| + \left\{ \sum_{i=2}^{N} (y_i - \beta_i)^2 + \lambda \sum_{i=3}^{N} |\beta_i - \beta_{i-1}| \right\}. \tag{2.4.5}$$

Έτσι απλοποιούμε το αρχικό πρόβλημα σε επιμέρους υποπροβλήματα. Το πρώτο από αυτά είναι η ελαχιστοποίηση της συνάρτησης

$$g(\beta_1, \beta_2) = (y_1 - \beta_1)^2 + \lambda|\beta_2 - \beta_1| \tag{2.4.6}$$

η οποία εξαρτάται μόνο από τις δύο πρώτες παραμέτρους. Ελαχιστοποιώντας την ως προς το συντελεστή β_1, λαμβάνουμε την εκτίμηση $\hat{\beta}_1(\beta_2)$ για την πρώτη παράμετρο β_1, η οποία είναι συνάρτηση της παραμέτρου β_2. Στη συνέχεια θα θεωρήσουμε αυτή τη νέα τιμή για να ελαχιστοποιήσουμε τη νέα συνάρτηση $f_2 : \mathbb{R}^{N-1} \to \mathbb{R}$

$$f_2(\beta_2, \ldots, \beta_N) = f(\hat{\beta}_1(\beta_2), \ldots, \beta_N) \tag{2.4.7}$$

η οποία μπορεί να γραφεί ως:

$$f_2(\beta_2,\dots,\beta_N) = h(\beta_2,\beta_3) + \left\{ \sum_{i=3}^{N}(y_i \quad \beta_i)^2 + \lambda \sum_{i=4}^{N} j\beta_i \quad \beta_{i\ 1} j \right\}$$

με $h(\beta_2,\beta_3) = g(\hat{\beta}_1(\beta_2)) + (y_2 \quad \beta_2)^2 + \lambda j\beta_3 \quad \beta_2 j$. Ελαχιστοποιώντας τώρα την $h(\beta_2,\beta_3)$ ως προς β_2 λαμβάνουμε την εκτίμηση $\hat{\beta}_2 = \hat{\beta}_2(\beta_3)$. Συνεχίζουμε τη διαδικασία μέχρι και τον υπολογισμό της $\hat{\beta}_N$ και στη συνέχεια γυρνώντας προς τα πίσω και αντικαθιστώντας, λαμβάνουμε τις τελικές εκτιμήτριες του μοντέλου $\hat{\beta}_{N\ 1} = \hat{\beta}_{N\ 1}(\hat{\beta}_N),\dots,\hat{\beta}_1 = \hat{\beta}_1(\hat{\beta}_2)$.

2.5 Adaptive Lasso

Πολλές φορές η μέθοδος Lasso μπορεί να οδηγήσει σε εκτιμήτριες οι οποίες έχουν μεγάλη μεροληψία. Αυτό συμβαίνει διότι οι συντελεστές των ανεξάρτητων μεταβλητών δέχονται την ίδια ποινή ανεξαρτήτως του μεγέθους τους. Για παράδειγμα όταν ένας μη μηδενικός συντελεστής είναι μεγάλος, τότε μπορεί να δέχεται μεγάλη ποινή με αποτέλεσμα να συρρικνωθεί αρκετά και επομένως η εκτίμησή του να έχει μεγάλη μεροληψία. Επιπλέον τις περισσότερες φορές επιλέγουμε την παράμετρο ποινής (π.χ με Cross Validation), προκειμένου το μοντέλο που θα προκύψει να είναι το βέλτιστο ως προς την προβλεπτική ικανότητα. Όμως, αυτό γενικά έρχεται σε αντίθεση με την ιδιότητα της μεθόδου να επιλέγει το υποσύνολο των μεταβλητών που σχετίζονται με τη μεταβλητή απόκρισης (variable selection). Δηλαδή υπάρχει περίπτωση επιλέγοντας το λ εκείνο που ελαχιστοποιεί το σφάλμα cross-validation, να καταλήξουμε σε ένα μοντέλο που περιέχει πάρα πολλές μεταβλητές. Βέβαια το μοντέλο αυτό είναι πολύ πιθανό να περιέχει όλες τις μεταβλητές που σχετίζονται με τη μεταβλητή απόκρισης (Bühlmann, van de Geer, 2011). Έτσι, συχνά χρειάζεται να αυξήσουμε το μέγεθος της παραμέτρου λ, προκειμένου να περιοριστούν περαιτέρω κάποιοι συντελεστές και κατά συνέπεια να οδηγηθούμε σε ένα πιο αραιό μοντέλο. Για τους λόγους αυτούς, προτάθηκε η μέθοδος Adaptive Lasso από τον Zou (2006). Η μέθοδος Adaptive Lasso επιλύει το εξής πρόβλημα ελαχιστοποίησης:

$$\min_{\beta\in\mathbb{R}^p}\left\{ \frac{1}{2} k\mathbf{y} \quad \mathbf{X}\beta k_2^2 + \lambda \sum_{j=1}^{p} w_j j\beta_j j \right\} \tag{2.5.1}$$

με $w_j = \frac{1}{j\hat{\beta}_{\text{init},j} j}$, όπου $\hat{\beta}_{\text{init}}$ είναι μια αρχική εκτιμήτρια για το διάνυσμα των συντελεστών. Το παραπάνω πρόβλημα αποτελεί μια παραλλαγή του προβλήματος ελαχιστοποίησης της μεθόδου Lasso και για δοθέντα w_j είναι κυρτό. Το μέγεθος της ποινής που επιβάλλεται σε κάθε συντελεστή β_j εξαρτάται από τις ποσότητες-βάρη w_j. Η ποινή που επιβάλλεται σε κάθε συντελεστή είναι αντιστρόφως ανάλογη του μεγέθους του. Έτσι όταν η αρχική εκτίμηση $\hat{\beta}_{\text{init},j}$ είναι μεγάλη τότε ο συντελεστής β_j ποινικοποιείται σε μικρότερο βαθμό (άρα έχουμε διόρθωση μεροληψίας) και αντίστοιχα όταν η αρχική εκτίμηση είναι μικρή, τότε ο συντελεστής δέχεται μεγαλύτερη ποινή με αποτέλεσμα να γίνει μηδενικός (άρα καταλήγουμε σε ένα ακόμα πιο αραιό μοντέλο). Προφανώς αν $\hat{\beta}_{\text{init},j} = 0$ τότε $\hat{\beta}_j = 0$. Όταν το πλήθος των παρατηρήσεων είναι μεγαλύτερο του αριθμού των παραγόντων τότε ως αρχική εκτίμηση μπορεί να χρησιμοποιηθεί η εκτιμήτρια ελαχίστων τετραγώνων. Στην περίπτωση όπου $p \quad N$, οι Bühlmann,van de Geer, (2011) προτείνουν ως αρχική εκτιμήτρια την εκτιμήτρια Lasso, υπολογισμένη για την τιμή της παραμέτρου ποινής που ελαχιστοποιεί το

σφάλμα Cross Validation, έστω λ_{min}. Μπορούμε λοιπόν να πούμε ότι η μέθοδος Adaptive Lasso είναι μια διαδικασία που αποτελείται από δύο στάδια. Στο πρώτο στάδιο υπολογίζουμε την αρχική εκτιμήτρια μέσω της μεθόδου Lasso ως $\hat{\boldsymbol{\beta}}_{\text{init}} = \hat{\boldsymbol{\beta}}(\lambda_{\text{min}})$. Στο δεύτερο στάδιο επαναλαμβάνουμε το Cross Validation προκειμένου να υπολογίσουμε την παράμετρο λ της σχέσης (2.5.1) και ύστερα να βρούμε τη νέα εκτιμήτρια Adaptive Lasso. Αυτό οδηγεί σε μικρότερο υπολογιστικό κόστος, διότι αντί να υπολογίσουμε εξαρχής τη λύση του (2.5.1) (ουσιαστικά έχουμε δύο παραμέτρους συρρίκνωσης που πρέπει να υπολογιστούν ταυτόχρονα μέσω Cross Validation), λύνουμε πρώτα δύο πιο απλά προβλήματα ελαχιστοποίησης, ένα για κάθε στάδιο. Στη συνέχεια παρουσιάζουμε ένα τρόπο για τον υπολογισμό της εκτιμήτριας με τη μέθοδο Adaptive Lasso.

2.5.1 Υπολογισμός της εκτιμήτριας Adaptive Lasso

Στη γενική περίπτωση για τον υπολογισμό της εκτιμήτριας Adaptive Lasso μπορούμε να χρησιμοποιήσουμε την αναπαραμέτρηση

$$\mathbf{x}_j = \left|\hat{\beta}_{\text{init},j}\right|\mathbf{x}_j \quad \text{και} \quad \beta_j = \frac{\beta_j}{\left|\hat{\beta}_{\text{init},j}\right|} \tag{2.5.2}$$

για τα j με $\hat{\beta}_{\text{init},j} \neq 0$. Τότε το πρόβλημα (2.5.1) μπορεί να γραφεί ως:

$$\min_{\beta \in \mathbb{R}^p}\left\{\frac{1}{2}\|\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\right\} \tag{2.5.3}$$

εφόσον έχουμε παραλείψει τις μεταβλητές X_j για τις οποίες $\hat{\beta}_{\text{init},j} = 0$. Επομένως αρκεί να λύσουμε αυτό το πρόβλημα Lasso και να υπολογίσουμε την εκτιμήτρια, έστω $\hat{\tilde{\boldsymbol{\beta}}}$. Τότε για την εκτιμήτρια Adaptive Lasso θα είναι:

$$\hat{\beta}_j = \hat{\tilde{\beta}}_j\left|\hat{\beta}_{\text{init,j}}\right|, \quad \text{αν } \hat{\beta}_{\text{init,j}} \neq 0 \tag{2.5.4}$$

και $\hat{\beta}_j = 0$ αν $\hat{\beta}_{\text{init,j}} = 0$. Επομένως μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο Coordinate Descent για να επιλύσουμε το πρόβλημα (2.5.3) και στη συνέχεια με αντικατάσταση από τον παραπάνω τύπο να βρούμε την εκτιμήτρια Adaptive Lasso.

2.5.2 Εφαρμογή

Ως εφαρμογή θα συγκρίνουμε τις μεθόδους Lasso και Adaptive Lasso. Για το σκοπό αυτό, προσομοιώνουμε 50 δείγματα μεγέθους $N = 70$ παρατηρήσεων και $p = 1000$ παραγόντων. Για τις ανεξάρτητες μεταβλητές θεωρούμε ότι $X_j \sim N(0, 1)$ για $j = 1, \ldots, 1000$ και για τη μεταβλητή απόκρισης ισχύει

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

όπου για τα σφάλματα έχουμε $\varepsilon_i \sim N(0, 1), i = 1, \ldots, 70$ και για το διάνυσμα των παραμέτρων:

$$\boldsymbol{\beta} = (1, \quad 2, 1.5, \underbrace{0, \ldots, 0}_{997}).$$

Ο σταθερός όρος στο μοντέλο μας θεωρείται ίσος με το μηδέν. Με χρήση του παρακάτω κώδικα, αρχικά θέτουμε τις τιμές για τις παραμέτρους $N, p, \boldsymbol{\beta}$ και αρχικοποιούμε τους πίνακες coef_lasso

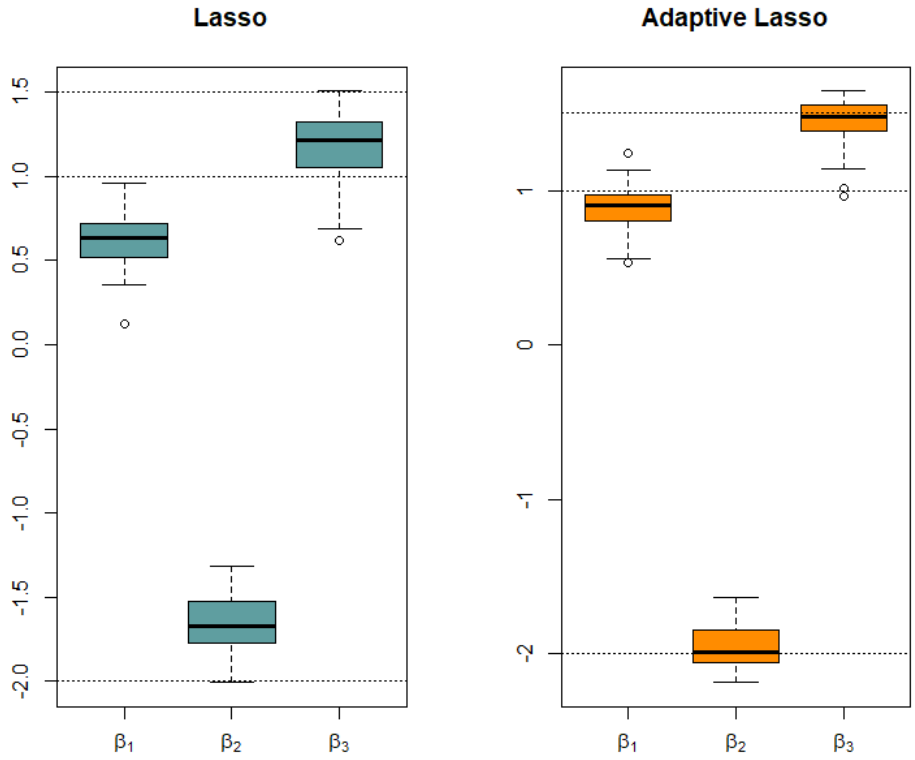
και coef_ad_lasso, στους οποίους θα εκχωρήσουμε τις εκτιμήσεις των συντελεστών για τις μεταβλητές X_1, X_2, X_3, για κάθε μία από τις δύο μεθόδους στα 50 δείγματα που θα προσομοιώσουμε. Επίσης αρχικοποιούμε και τα διανύσματα lasso_vars και ad_lasso_vars που θα περιέχουν το πλήθος των μεταβλητών που θα συμπεριλαμβάνονται στο μοντέλο, ύστερα από προσαρμογή του και με τις δύο μεθόδους Lasso και Adaptive Lasso, για κάθε ένα από τα 50 δείγματα. Στη συνέχεια, μέσα στο βρόχο for προσομοιώνουμε τιμές για τις επεξηγηματικές μεταβλητές και υπολογίζουμε τη μεταβλητή απόκρισης, με βάση τις σχέσεις που περιγράψαμε. Με την εντολή cv.glmnet εκτελούμε 10-fold CV και εν συνεχεία υπολογίζουμε τους συντελεστές (χωρίς τη σταθερά) της μεθόδου Lasso, για την παράμετρο λ που ελαχιστοποιεί το σφάλμα cross-validation. Επίσης βρίσκουμε ποιοί συντελεστές είναι μη μηδενικοί με τη βοήθεια της εντολής sapply και ύστερα παίρνουμε το πλήθος τους. Για τη μέθοδο Adaptive Lasso θα χρησιμοποιήσουμε ως αρχική εκτιμήτρια την εκτιμήτρια Lasso. Το όρισμα penalty.factor στην εντολή cv.glmnet χρησιμοποιείται για να δώσουμε διαφορετική ποινή σε κάθε έναν συντελεστή. Και για τη μέθοδο Adaptive Lasso υπολογίζουμε τους συντελεστές για τους τρεις πρώτους παράγοντες, καθώς και το πλήθος των ανεξάρτητων μεταβλητών που περιέχονται τελικά στο μοντέλο.

```
> library(glmnet)
> set.seed(1)
> N=70
> p=1000
> beta=c(1,-2,1.5,rep(0,997))
> coef_lasso=matrix(0,50,3)
> coef_ad_lasso=matrix(0,50,3)
> lasso_vars=rep(0,50)
> ad_lasso_vars=rep(0,50)
> for (k in 1:50){
   x=matrix(rnorm(N*p),N,p)
   y=x%*%beta+rnorm(N)

   lasso=cv.glmnet(x,y)
   lasso.coef=coef(lasso,s="lambda.min")[-1]
   coef_lasso[k,1:3]=lasso.coef[1:3]
   lasso_vars[k]=length(which(sapply(lasso.coef,
                        function(x) x!=0)))

   ad_lasso=cv.glmnet(x,y,penalty.factor=1/(abs(lasso.coef)))
   ad_lasso.coef=coef(ad_lasso,s="lambda.min")[-1]
   coef_ad_lasso[k,1:3]=ad_lasso.coef[1:3]
   ad_lasso_vars[k]=length(which(sapply(ad_lasso.coef,
                           function(x) x!=0)))

 }
```

Στη συνέχεια εμφανίζουμε τις διαμέσους του πλήθους των επεξηγηματικών μεταβλητών για κάθε μία μέθοδο. Επίσης κατασκευάζουμε τα θηκοδιαγράμματα των συντελεστών των τριών πρώτων παραγόντων (βλ. Διάγραμμα 2.5).

```
> c(median(lasso_vars),median(ad_lasso_vars))
```

```
[1] 23  5
> par(mfrow=c(1,2))
> expr=c(expression(beta[1]),expression(beta[2]),
         expression(beta[3]))
> boxplot(coef_lasso[,1:3],xaxt="n",col="cadetblue")
> title(main = "Lasso")
> axis(side = 1,at=1:3,labels=expr)
> abline(h=c(1,-2,1.5),lty="dotted")
> boxplot(coef_ad_lasso[,1:3],xaxt="n",col="darkorange")
> title(main="Adaptive Lasso")
> axis(side = 1,at=1:3,labels=expr)
> abline(h=c(1,-2,1.5),lty="dotted")
```



Διάγραμμα 2.5: Θηκοδιαγράμματα των συντελεστών των μεταβλητών X_1, X_2, X_3, που έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης. Η προσαρμογή έγινε με χρήση των μεθόδων Lasso και Adaptive Lasso.

Στα 50 δείγματα που προσομοιώσαμε, η διάμεσος του πλήθους των επεξηγηματικών μεταβλητών που περιέχονται στο μοντέλο ύστερα από προσαρμογή με τη μέθοδο Lasso, είναι 23. Άρα στο συγκεκριμένο παράδειγμα προκύπτει ότι με τη μέθοδο Lasso, συμπεριλαμβάνονται επιπλέον μεταβλητές από αυτές που παίζουν ουσιαστικό ρόλο στην ερμηνεία της μεταβλητής απόκρισης. Αντιθέτως, με τη μέθοδο Adaptive Lasso, παρατηρούμε ότι η διάμεσος του πλήθους των επεξηγηματικών μεταβλητών είναι μόλις 5, ένω ουσιαστική επίδραση έχουν μόνο 3 από τις 1000

επεξηγηματικές μεταβλητές. Άρα με τη μέθοδο Adaptive Lasso καταλήγουμε σαφώς σε ένα πιο αραιό μοντέλο. Επίσης οι εκτιμήτριες των συντελεστών για τις μεταβλητές X_1, X_2, X_3 που προκύπτουν από τη μέθοδο Adaptive Lasso είναι γενικά πιο κοντά στις πραγματικές τους τιμές, από εκείνες που προκύπτουν από τη μέθοδο Lasso. Αυτό είναι φανερό από το Διάγραμμα 2.5, αφού για παράδειγμα η διάμεσος για το συντελεστή β_2 είναι σχεδόν ίση με την πραγματική του τιμή -2 (δεξιά), σε σχέση με την αντίστοιχη διάμεσο του β_2 (αριστερά). Οι οριζόντιες (dotted) γραμμές αντιστοιχούν στις πραγματικές τιμές των συντελεστών των τριών πρώτων παραγόντων. Τα αποτελέσματα αυτά συνοψίζονται και στον Πίνακα 2.1.

Μέθοδος	Διάμεσος αριθμού επιλεγμένων μεταβλητών	Διάμεσος τιμών των συντελεστών		
		β_1	β_2	β_3
Lasso	23	0.633	-1.672	1.212
Adaptive Lasso	5	0.908	-1.992	1.474
(Πραγματικό μοντέλο)	3	1	-2	1.5

Πίνακας 2.1: Προσαρμογή του μοντέλου με χρήση των μεθόδων Lasso και Adaptive Lasso. Η μεσαία στήλη περιλαμβάνει τη διάμεσο του πλήθους των ανεξάρτητων μεταβλητών που περιέχονται στο μοντέλο, ενώ η δεξιά στήλη περιλαμβάνει τη διάμεσο των συντελεστών $\beta_1, \beta_2, \beta_3$, καθώς και τις πραγματικές τους τιμές.

2.6 Μη κυρτές ποινές

Είδαμε ότι με τη μέθοδο Lasso επιλέγουμε τελικά ένα υποσύνολο από τις αρχικές μας διαθέσιμες επεξηγηματικές μεταβλητές ώστε να συμπεριλάβουμε στο τελικό μοντέλο. Επειδή όμως η μέθοδος χρησιμοποιείται ταυτόχρονα και για την εκτίμηση των συντελεστών του μοντέλου αλλά και για την επιλογή των σημαντικών μεταβλητών, ενδέχεται οι συντελεστές ορισμένων μεταβλητών να μη μειωθούν αρκετά. Κατά συνέπεια όταν ο αριθμός των παραγόντων είναι μεγάλος και μόνο λίγες μεταβλητές έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης, η προσαρμογή με τη μέθοδο Lasso μπορεί να οδηγήσει σε μοντέλα τα οποία περιέχουν επιπλέον περιττές μεταβλητές. Προκειμένου λοιπόν να καταλήξουμε σε πιο αραιά μοντέλα, έχουν μελετηθεί και κριτήρια που περιέχουν μη κυρτούς όρους ποινής. Μια πρώτη εναλλακτική προσέγγιση είναι να χρησιμοποιήσουμε τις l_q-νόρμες, για $0 \quad q < 1$, αφού για $q \in [0,1)$ το σύνολο περιορισμού είναι μη κυρτό. Η περίπτωση l_0 για $q = 0$ ($\|\beta\|_0 = \sum_{j=1}^{p} I(\beta_j \neq 0)$), με I να είναι η δείκτρια συνάρτηση, αντιστοιχεί στην επιλογή του βέλτιστου υποσυνόλου ανεξάρτητων μεταβλητών (best-subset selection). Η νόρμα αυτή μετράει το πλήθος των μη μηδενικών συντελεστών. Με τη μέθοδο αυτή εξετάζουμε τα μοντέλα που περιέχουν όλους τους πιθανούς συνδυασμούς παραγόντων (συνολικά 2^p μοντέλα) και επιλέγουμε τους σημαντικότερους από αυτούς, με βάση ορισμένα στατιστικά κριτήρια (π.χ BIC, C_p, R^2_{adj}). Όταν όμως ο αριθμός των παραγόντων p είναι μεγάλος (π.χ $p > 40$) το πρόβλημα υπολογισμού όλων των πιθανών συνδυασμών είναι υπολογιστικά δύσκολο. Έτσι έχουν αναπτυχθεί και άλλες εναλλακτικές μη κυρτές ποινές. Για αυτού του είδους τις ποινές το πρόβλημα ελαχιστοποίησης είναι το εξής:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} P_{\lambda,\gamma}(\beta_j) \right\}, \tag{2.6.1}$$

όπου ο όρος $P_{\lambda,\gamma}(\beta_j)$ ορίζει μια οικογένεια ποινών οι οποίες είναι κοίλες στο $|\beta_j|$, γ είναι μια παράμετρος που ελέγχει το βαθμό στον οποίο η ποινή είναι κοίλη και λ η παράμετρος ποινής. Στην

περίπτωση όπου έχουμε έναν παράγοντα μπορούμε να ελαχιστοποιήσουμε τη συνάρτηση:

$$f(\beta) = \frac{1}{2}(\beta \quad \tilde{\beta}) + \lambda P_{\lambda,\gamma}(\beta), \tag{2.6.2}$$

όπου ως $\tilde{\beta}$ μπορούμε να θεωρήσουμε την εκτιμήτρια ελαχίστων τετραγώνων. Για διαφορετικές ποινές ορίζεται και ο αντίστοιχος γενικευμένος τελεστής Soft-thresholding με

$$S_\gamma(\tilde{\beta}, \lambda) = \underset{\beta 2 \mathsf{R}}{\operatorname{argmin}} f(\beta) \tag{2.6.3}$$

Ορισμένα παραδείγματα (Mazumder et al, 2010) για τη συνάρτηση $P_{\lambda,\gamma}(\beta)$ είναι τα εξής:

1. Η ποινή l_γ με $\lambda P_{\lambda,\gamma}(\beta) = \lambda |\beta|^\gamma$ για γ 2 [0, 1] (Friedman, 2008).

2. Η λογαριθμική ποινή (log-penalty) με

$$\lambda P_{\lambda,\gamma}(\beta) = \frac{\lambda}{\log(\gamma + 1)} \log(\gamma |\beta| + 1), \gamma > 0$$

   (για γ ! 0_+ έχουμε τον l_1-περιορισμό (Lasso) ενώ για γ ! + 1 έχουμε τον l_0-περιορισμό (Best-subset selection)).

3. Η ποινή SCAD (Fan and Li, 2001) που ορίζεται ως:

   $\frac{d}{d\beta} P_{\lambda,\gamma}(\beta) = \mathrm{I}\{\beta \quad \lambda\} + \frac{(\gamma\lambda \quad \beta)_+}{(\gamma \quad 1)\lambda} \mathrm{I}\{\beta > \lambda\}$ για $\beta > 0, \gamma > 2$

   $P_{\lambda,\gamma}(\beta) = P_{\lambda,\gamma}(\quad \beta)$

   $P_{\lambda,\gamma}(0) = 0$

4. Η ποινή MC+ (Zhang, 2010) που ορίζεται ως:

$$\begin{aligned} \lambda P_{\lambda,\gamma}(\beta) &= \lambda \int_0^{|\beta|} \left(1 \quad \frac{x}{\lambda\gamma}\right)_+ dx \\ &= \lambda \left(|\beta| \quad \frac{\beta^2}{2\lambda\gamma}\right) \mathrm{I}\{|\beta| < \lambda\gamma\} + \frac{\lambda^2\gamma}{2} \mathrm{I}\{|\beta| \quad \lambda\gamma\} \end{aligned}$$

   (για γ ! + 1 έχουμε τον l_1-περιορισμό ενώ για γ ! 1_+ έχουμε τον l_0-περιορισμό).

Γενικά οι μη κυρτοί περιορισμοί ποινικοποιούν σε μεγαλύτερο βαθμό τους συντελεστές των ανεξάρτητων μεταβλητών, απ΄ ότι ο l_1-περιορισμός της μεθόδου Lasso. Όμως οδηγούν σε μη κυρτά προβλήματα ελαχιστοποίησης τα οποία είναι σαφώς δυσκολότερα στην επίλυσή τους. Μια προσέγγιση για την ελαχιστοποίηση του προβλήματος (2.6.1) δίνεται από τους Mazumder, Friedman, Hastie (2010) οι οποίοι προτείνουν τον αλγόριθμο SparseNet. Ο αλγόριθμος χρησιμοποιείται στο πακέτο sparsenet της R.

2.7 Συμπέρασμα

Στο Κεφάλαιο αυτό παρουσιάσαμε κάποιες γενικεύσεις της μεθόδου Lasso και είδαμε πως αυτές μπορούν να χρησιμοποιηθούν για να διορθώσουν ορισμένες ελλείψεις της. Από τα προσομοιωμένα μας παραδείγματα, με τη χρήση της μεθόδου Elastic Net είδαμε πως αυτή η μέθοδος μπορεί να ξεπεράσει τη μέθοδο Lasso σε περιπτώσεις όπου υπάρχει υψηλή συσχέτιση μεταξύ των επεξηγηματικών μεταβλητών. Για κατάλληλη τιμή της παραμέτρου α καταλήξαμε σε μοντέλα τα οποία ήταν και φειδωλά αλλά είχαν και υψηλότερη προβλεπτική ακρίβεια σε σχέση με τα μοντέλα που καταλήξαμε με τη μέθοδο Lasso. Επιπλέον με τη μέθοδο Group Lasso είδαμε πως μπορούμε να κάνουμε επιλογή ολόκληρων ομάδων ανεξάρτητων μεταβλητών στο τελικό μας μοντέλο. Όταν μια ομάδα συμπεριλαμβάνεται στο μοντέλο, τότε όλοι οι συντελεστές των μεταβλητών που ανήκουν σε αυτήν, είναι μη μηδενικοί. Πολλές φορές όμως μπορεί να μη χρειάζονται όλες αυτές τις μεταβλητές, οπότε εφαρμόζουμε τη μέθοδο Sparse Group Lasso για να πετύχουμε αραιότητα και μέσα σε κάθε γκρουπ. Επίσης, παρουσιάσαμε τη μέθοδο Fused Lasso, η οποία μπορεί να χρησιμοποιηθεί σε προβλήματα στα οποία οι ανεξάρτητες μεταβλητές μπορούν να διαταχθούν π.χ ως προς το χρόνο. Τέλος, με τη μέθοδο Adaptive Lasso και με τη χρήση μη κυρτών όρων ποινής είδαμε πως μπορούμε να καταλήξουμε σε μοντέλα τα οποία είναι πιο αραιά και με καλύτερη προβλεπτική ικανότητα, διορθώνοντας έτσι την αδυναμία της μεθόδου Lasso σε περιπτώσεις όπου επιλέγει επιπλέον περιττές μεταβλητές στο μοντέλο. Στο επόμενο Κεφάλαιο θα αναλύσουμε διάφορες τεχνικές προκειμένου να εξάγουμε συμπεράσματα για τους εκτιμητές που προκύπτουν από την προσαρμογή ενός γραμμικού μοντέλου, με τη χρήση ορισμένων μεθόδων συρρίκνωσης.

Kefˆlaio 3

Statistik Sumperasmatologa

3.1 Εισαγωγή

Στα προηγούμενα Κεφάλαια είδαμε πως με τη χρήση ορισμένων τεχνικών συρρίκνωσης, μπορούμε να προσαρμόσουμε ένα γραμμικό μοντέλο. Μέσω αυτών των μεθόδων εκτιμούμε τους συντελεστές των επεξηγηματικών μεταβλητών και έτσι καταλήγουμε σε ένα τελικό μοντέλο, το οποίο μπορούμε να χρησιμοποιήσουμε για προβλέψεις, όσον αφορά τη μεταβλητή απόκρισης. Για να είναι οι προβλέψεις μας όσο το δυνατόν καλύτερες, θα πρέπει και οι εκτιμήσεις μας για τους συντελεστές να είναι αξιόπιστες, δηλαδή οι συντελεστές να έχουν μικρά τυπικά σφάλματα. Όμως, μέσω των μεθόδων συρρίκνωσης, οι εκτιμήτριες που προκύπτουν δεν έχουν κάποιο τύπο που να δίνει το τυπικό τους σφάλμα, ώστε να εξετάσουμε κατά πόσο είναι αξιόπιστες και άρα να στηριχθούμε σε αυτές για μελλοντικές προβλέψεις. Στο Κεφάλαιο αυτό, θα αναλύσουμε διάφορες μεθόδους επαναδειγματοληψίας από το αρχικό μας δείγμα, τις οποίες μπορούμε να χρησιμοποιήσουμε προκειμένου να κάνουμε συμπερασματολογία για τους εκτιμητές του μοντέλου. Επίσης θα εφαρμόσουμε τις μεθόδους αυτές πάνω σε ένα προσομοιωμένο παράδειγμα με χρήση της R.

3.2 Μπεϋζιανή προσέγγιση των μεθόδων Lasso και Ridge

Θεωρούμε το πολλαπλό γραμμικό μοντέλο

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3.2.1}$$

όπου τα σφάλματα είναι ανεξάρτητα και ακολουθούν την κανονική κατανομή με μέση τιμή 0 και διασπορά σ^2. Σύμφωνα με την Μπεϋζιανή προσέγγιση, οι παράμετροι του μοντέλου θεωρούνται τυχαίες ποσότητες, οι οποίες έχουν μια εκ των προτέρων (prior) κατανομή. Οι Park και Casella (2008), θεωρούν ως μια κατάλληλη εκ των προτέρων κατανομή για τις παραμέτρους, την κατανομή Laplace (ή double exponential). Πρόκειται για μία συνεχή κατανομή η οποία χαρακτηρίζεται από δύο παραμέτρους (μ, b), όπου μ είναι η παράμετρος θέσης και b η παράμετρος κλίμακας. Αν υποθέσουμε ότι διαθέτουμε ένα δείγμα με N παρατηρήσεις και p επεξηγηματικές μεταβλητές, τότε σύμφωνα με τη σχέση (3.2.1), για τη μεταβλητή απόκρισης θα ισχύει $y_i | \mathbf{x}_i, \boldsymbol{\beta}, \sigma^2 \quad N(\mathbf{x}_i^T\boldsymbol{\beta}, \sigma^2)$ για κάθε $i = 1, \ldots, N$. Επίσης για κάθε συντελεστή υποθέτουμε a-priori ότι $\beta_j \quad Laplace(0, \frac{\sigma}{\lambda})$, με $j = 1, \ldots, p$, και $\lambda > 0$. Επομένως οι αντίστοιχες συναρτήσεις πυκνότητας πιθανότητας θα

είναι:

$$f(y_i|\boldsymbol{\beta},\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{\frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2}{2\sigma^2}\right\} \tag{3.2.2}$$

$$p(\beta_j) = \frac{\lambda}{2\sigma}e^{-\frac{\lambda}{\sigma}|\beta_j|}. \tag{3.2.3}$$

Άρα η συνάρτηση πιθανοφάνειας του δείγματος δοθέντων των παραμέτρων θα είναι:

$$\begin{aligned} L(\boldsymbol{\beta};\mathbf{y},\sigma^2) = f(\mathbf{y}|\boldsymbol{\beta},\sigma^2) &= \prod_{i=1}^{N} f(y_i|\boldsymbol{\beta},\sigma^2) = \prod_{i=1}^{N}\frac{1}{\sigma\sqrt{2\pi}}\exp\left\{\frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2}{2\sigma^2}\right\} \\ &= \frac{1}{(\sigma\sqrt{2\pi})^N}\exp\left\{\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2\right\} \\ &= \frac{1}{(\sigma\sqrt{2\pi})^N}\exp\left\{\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\right\} \end{aligned} \tag{3.2.4}$$

(το γινόμενο προκύπτει λόγω ανεξαρτησίας των y_i). Επίσης η από κοινού σ.π.π των συντελεστών θα γράφεται (πάλι λόγω ανεξαρτησίας) ως:

$$\begin{aligned} p(\boldsymbol{\beta}) = \prod_{j=1}^{p} p(\beta_j) &= \prod_{j=1}^{p}\frac{\lambda}{2\sigma}e^{-\frac{\lambda}{\sigma}|\beta_j|} \\ &= \left(\frac{\lambda}{2\sigma}\right)^p \exp\left\{-\frac{\lambda}{\sigma}\sum_{j=1}^{p}|\beta_j|\right\} \\ &= \left(\frac{\lambda}{2\sigma}\right)^p \exp\left\{-\frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right\}. \end{aligned} \tag{3.2.5}$$

Σύμφωνα με το θεώρημα Bayes θα είναι:

$$p(\boldsymbol{\beta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{\int f(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})d\boldsymbol{\beta}}, \tag{3.2.6}$$

όπου $p(\boldsymbol{\beta}|\mathbf{y})$ είναι η εκ των υστέρων (posterior) κατανομή των παραμέτρων, δοθέντων των παρατηρήσεων $\mathbf{y}$. Το ολοκλήρωμα στον παρονομαστή παίζει το ρόλο της σταθεράς κανονικοποίησης για την $p(\boldsymbol{\beta}|\mathbf{y})$ και αφού ολοκληρώνουμε ως προς $\boldsymbol{\beta}$ δεν θα εξαρτάται από τις παραμέτρους. Επομένως η σχέση (3.2.6) μπορεί να γραφεί και ως:

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}), \tag{3.2.7}$$

όπου το σύμβολο $\propto$ δηλώνει αναλογία. Ύστερα από σχετικές απλοποιήσεις και συγχωνεύσεις σταθερών, θα έχουμε:

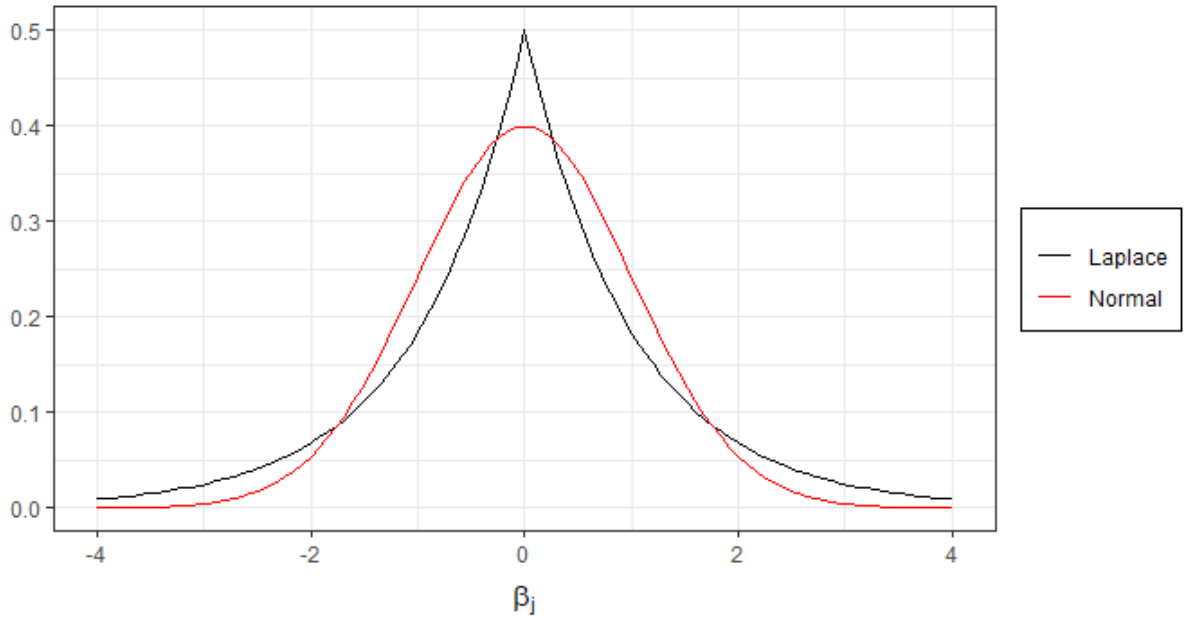
$$p(\boldsymbol{\beta}|\mathbf{y}) = c\exp\left\{\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 - \frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1\right\}.$$

Παίρνοντας τώρα λογάριθμο και στα δύο μέλη προκύπτει τελικά :

$$\log p(\boldsymbol{\beta}|\mathbf{y}) = \log c \quad \frac{1}{\sigma^2}\left(\frac{1}{2}\|\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sigma\lambda\|\boldsymbol{\beta}\|_1\right). \tag{3.2.8}$$

Συμπεραίνουμε λοιπόν ότι η τιμή που μεγιστοποιεί την συνάρτηση $p(\boldsymbol{\beta}|\mathbf{y})$, ισοδύναμα το λογάριθμο $\log p(\boldsymbol{\beta}|\mathbf{y})$, συμπίπτει με την εκτιμήτρια Lasso με παράμετρο ποινής $\sigma\lambda$, αφού θα ελαχιστοποιεί τη σχέση $\frac{1}{2}\|\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sigma\lambda\|\boldsymbol{\beta}\|_1$. Αν τώρα υποθέσουμε ότι οι συντελεστές β_j ακολουθούν a-priori την κανονική κατανομή με μέση τιμή 0 και διασπορά που εξαρτάται από τις παραμέτρους σ, λ, δηλαδή $\beta_j \quad N(0, \frac{\sigma^2}{\lambda})$, τότε με παρόμοιο τρόπο προκύπτει ότι η τιμή που μεγιστοποιεί την εκ των υστέρων σ.π.π. των παραμέτρων είναι η εκτιμήτρια Ridge, διότι θα ελαχιστοποιεί την παράσταση $\|\mathbf{y} \quad \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$. Στο Διάγραμμα 3.1 φαίνονται οι γραφικές παραστάσεις των εκ των προτέρων κατανομών, για τις δύο περιπτώσεις Lasso και Ridge. Παρατηρούμε ότι η κατανομή Laplace παρουσιάζει μία αιχμηρή κορυφή στο 0. Επομένως είναι λογικό να θεωρήσουμε αυτή την κατανομή (ως prior) για τους συντελεστές του μοντέλου, εφόσον πιστεύουμε εξαρχής ότι λίγες μεταβλητές θα παίζουν σημαντικό ρόλο και άρα αρκετοί από τους συντελεστές τους θα είναι μηδενικοί. Επίσης είναι λογικό η διασπορά να εξαρτάται και από την παράμετρο λ, αφού περιμένουμε ότι όσο αυξάνουμε την παράμετρο λ, τόσο περιορίζουμε τους συντελεστές, με αποτέλεσμα οι τιμές τους να είναι πιο κοντά στο 0 και επομένως να έχουν μικρότερη διασπορά. Παρόμοια αποτελέσματα ισχύουν και για την μέθοδο Ridge, μόνο που τώρα παρατηρούμε ότι η εκ των προτέρων κανονική κατανομή των συντελεστών είναι πιο επίπεδη γύρω από το 0. Συνεπώς αναμένουμε οι περισσότεροι από τους συντελεστές να είναι κατανεμημένοι γύρω από το 0 (αλλά όχι ακριβώς μηδενικοί).

Τις περισσότερες φορές ο ακριβής υπολογισμός της εκ των υστέρων σ.π.π. των συντελεστών



Διάγραμμα 3.1: Οι εκ των προτέρων κατανομές των συντελεστών για τις περιπτώσεις Lasso και Ridge. Η κατανομή Laplace έχει αιχμηρή κορυφή στο 0, άρα αναμένουμε αρκετοί συντελεστές να είναι ακριβώς μηδενικοί (περίπτωση Lasso). Η κανονική κατανομή είναι επίπεδη στο 0, άρα περιμένουμε αρκετοί συντελεστές να είναι κατανεμημένοι γύρω από το 0 (περίπτωση Ridge).

δοθέντων των παρατηρήσεων είναι αρκετά δύσκολος. Αυτό οφείλεται κυρίως στη μορφή του ολοκληρώματος της σχέσης (3.2.6), το οποίο συνήθως αποτελεί ένα σύνθετο ολοκλήρωμα σε πολλές

διαστάσεις. Έτσι χρησιμοποιούνται τεχνικές Markov Chain Monte Carlo (MCMC) (Andrieu et al, 2003) προκειμένου να γίνει προσομοίωση τιμών από την εκ των υστέρων σ.π.π των παραμέτρων. Άρα χωρίς να ξέρουμε τον ακριβή τύπο αυτής της κατανομής μπορούμε να πάρουμε αρκετές πληροφορίες για αυτήν μέσω των προσομοιωμένων τιμών της. Έτσι μπορούμε να εκτιμήσουμε τη μέση τιμή, τη διασπορά, ακόμα και το γράφημα της σ.π.π με τη μέθοδο των πυρήνων (Kernel density estimation). Οι εκτιμητές αυτοί ονομάζονται Monte Carlo εκτιμητές και αποδεικνύεται ότι για μεγάλο πλήθος δείγματος, συγκλίνουν στην ποσότητα που θέλουμε να εκτιμήσουμε (εφόσον η δειγματοληψία έχει γίνει με τυχαίο τρόπο). Συνεπώς, με αυτό τον τρόπο μπορούμε να κάνουμε συμπερασματολογία και για τις εκτιμήτριες Lasso και Ridge.

3.3 Bootstrap

Έστω τώρα ότι έχουμε προσαρμόσει το μοντέλο μας χρησιμοποιώντας τη μέθοδο Lasso, όπως περιγράψαμε στο πρώτο Κεφάλαιο. Δηλαδή ακολουθούμε την εξής διαδικασία: Αρχικά θεωρούμε ένα πλέγμα (grid) τιμών για την παράμετρο ποινής λ, έστω $= f\lambda_m g_{m=1}^{M}$ (σε φθίνουσα διάταξη). Για κάθε μία τιμή λ_m 2 προσαρμόζουμε το μοντέλο χρησιμοποιώντας όλες τις παρατηρήσεις που διαθέτουμε. Στη συνέχεια χρησιμοποιούμε Cross Validation (CV) για να βρούμε το λ εκείνο που ελαχιστοποιεί το σφάλμα cross-validation. Πιο συγκεκριμένα μπορούμε να χωρίσουμε τα δεδομένα μας με τυχαίο τρόπο, π.χ. σε k = 10 μη αλληλοεπικαλυπτόμενες ομάδες, ίδιου αν είναι δυνατόν μεγέθους. Αφήνοντας κάθε φορά μία ομάδα παρατηρήσεων εκτός (test set) προσαρμόζουμε το μοντέλο χρησιμοποιώντας τα υπόλοιπα 9 γκρουπ (training set) για όλα τα λ 2 και υπολογίζουμε τα αντίστοιχα μέσα τετραγωνικά σφάλματα πρόβλεψης. Για κάθε λ παίρνουμε τον αντίστοιχο μέσο όρο των σφαλμάτων και έτσι καταλήγουμε σε M διαφορετικά CV σφάλματα (όσα δηλαδή και το πλήθος του συνόλου). Από αυτά βρίσκουμε το μικρότερο και κρατάμε την τιμή της παραμέτρου ποινής, έστω λ_{min} που αντιστοιχεί σε αυτό, το ελάχιστο δηλαδή σφάλμα πρόβλεψης. Τέλος υπολογίζουμε την εκτιμήτρια Lasso, δηλαδή επιστρέφουμε το διάνυσμα των συντελεστών από την αρχική προσαρμογή του μοντέλου, με χρήση της παραμέτρου ποινής λ_{min}. Αφού έχουμε υπολογίσει την εκτιμήτρια αυτή, θέλουμε με κάποιο τρόπο να κάνουμε συμπερασματολογία γι΄ αυτή. Δηλαδή μας ενδιαφέρει να εξετάσουμε πόσο καλή είναι η εκτίμησή μας (π.χ τι τυπικό σφάλμα έχει, από ποια κατανομή προέρχεται κ.τ.λ.) Όμως η εκτιμήτρια Lasso, όπως έχουμε δει, αποτελεί μια σύνθετη εκτιμήτρια και δεν υπάρχει ακριβής τύπος που να δίνει, για παράδειγμα, το τυπικό της σφάλμα. Η μέθοδος Bootstrap (Efron, 1979) χρησιμοποιείται κυρίως για προβλήματα όπου θέλουμε να εκτιμήσουμε τις στατιστικές ιδιότητες τέτοιων πολύπλοκων εκτιμητών. Έστω ότι έχουμε ένα σετ δεδομένων με N παρατηρήσεις της μορφής $(Y, X_1, \ldots, X_p)$, δηλαδή διαθέτουμε το εξής δείγμα $(\mathbf{z}_1, \ldots, \mathbf{z}_N)$, όπου:

$$\mathbf{z}_1 = (y_1, x_{11}, \ldots, x_{1p})$$

$$\mathbf{z}_2 = (y_2, x_{21}, \ldots, x_{2p})$$

$$\vdots$$

$$\mathbf{z}_N = (y_N, x_{N1}, \ldots, x_{Np}).$$

Με το μη παραμετρικό Bootstrap εκτιμούμε τη συνάρτηση κατανομής F των $(\mathbf{z}_1, \ldots, \mathbf{z}_N)$, δηλαδή την από κοινού σ.κ των $(Y, X_1, \ldots, X_p)$ μέσω της εμπειρικής συνάρτησης κατανομής $\hat{F}_N$[1]. Αυτό ισοδυναμεί με το να κάνουμε δειγματοληψία με epanˆjesh από το αρχικό δείγμα, αφού η εμπειρική κατανομή δίνει την ίδια πιθανότητα επιλογής $1/N$ σε κάθε παρατήρηση. Έτσι παίρνουμε τα λεγόμενα Bootstrap δείγματα, ίδιου μεγέθους με το αρχικό δείγμα. Κάθε τέτοιο δείγμα μπορούμε να το συμβολίσουμε με $(\mathbf{z}_1, \ldots, \mathbf{z}_N)$ και είναι δυνατόν να περιέχει κάποιες παρατηρήσεις παραπάνω από μία φορά και κάποιες άλλες μπορεί και καθόλου. Έστω λοιπόν ότι σε κάθε τέτοιο δείγμα υπολογίζουμε, με τον τρόπο που περιγράψαμε στην αρχή της Ενότητας, την εκτιμήτρια Lasso. Μπορούμε να επαναλάβουμε B φορές (για B αρκούντως μεγάλο) την δειγματοληψία και σε κάθε νέο δείγμα να υπολογίζουμε τις εκτιμήτριες Lasso. Με αυτό τον τρόπο καταλήγουμε στις εκτιμήτριες $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_B$, όπου κάθε τέτοιο διάνυσμα συντελεστών θα γράφεται:

$$\hat{\boldsymbol{\beta}}_1 = (\hat{\beta}_{11}, \hat{\beta}_{12}, \ldots, \hat{\beta}_{1p})$$

$$\hat{\boldsymbol{\beta}}_2 = (\hat{\beta}_{21}, \hat{\beta}_{22}, \ldots, \hat{\beta}_{2p})$$

$$\vdots$$

$$\hat{\boldsymbol{\beta}}_B = (\hat{\beta}_{B1}, \hat{\beta}_{B2}, \ldots, \hat{\beta}_{Bp}).$$

Τώρα μπορούμε εύκολα να υπολογίσουμε το τυπικό σφάλμα για κάθε μία συνιστώσα, έστω π.χ. για τη $\hat{\beta}_j$, το οποίο θα δίνεται από τον τύπο:

$$\text{se}(\hat{\beta}_j) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\hat{\beta}_{ij} - \beta_{j(\cdot)})^2}, \tag{3.3.1}$$

όπου η ποσότητα

$$\beta_{j(\cdot)} = \frac{1}{B}\sum_{i=1}^{B}\hat{\beta}_{ij}$$

συμβολίζει τη μέση τιμή όλων αυτών των B συντελεστών για τη j συνιστώσα. Ένας άλλος τρόπος για να εκτιμήσουμε τα τυπικά σφάλματα των συντελεστών είναι να παράξουμε Bootstrap δείγματα από τα υπόλοιπα. Αρχικά προσαρμόζουμε το μοντέλο μας στα δεδομένα $(Y, X_1, \ldots, X_p)$ και υπολογίζουμε τις εκτιμήτριες $\hat{\beta}_0, \hat{\boldsymbol{\beta}}$ καθώς και τα υπόλοιπα $e_i = y_i - \hat{y}_i$, για κάθε $i = 1, \ldots, N$. Στη συνέχεια θα κάνουμε δειγματοληψία με επανάθεση από το αρχικό δείγμα των υπολοίπων, οπότε παίρνουμε ένα Bootstrap δείγμα από τα υπόλοιπα, έστω $(e_1, \ldots, e_N)$. Ύστερα κατασκευάζουμε τιμές για τη μεταβλητή απόκρισης με βάση τη σχέση $y_i = \hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}} + e_i$, όπου κρατάμε σταθερό τον πίνακα σχεδιασμού $\mathbf{X}$. Στο νέο τώρα σύνολο δεδομένων $(Y, X_1, \ldots, X_p)$ υπολογίζουμε την εκτιμήτρια Lasso, έστω $\hat{\boldsymbol{\beta}}_1$. Επαναλαμβάνουμε την παραπάνω διαδικασία έστω B φορές, οπότε καταλήγουμε και πάλι σε B το πλήθος εκτιμήτριες Lasso $(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_B)$.

Το τυπικό σφάλμα (3.3.1) αποτελεί μια εκτίμηση για το τυπικό σφάλμα της αρχικής εκτιμήτριας. Μέσω αυτού έχουμε τη δυνατότητα για παράδειγμα, να υπολογίσουμε διαστήματα εμπιστοσύνης, να κάνουμε ελέγχους υποθέσεων και γενικότερα έχοντας στη διάθεση μας αρκετές τέτοιες εκτιμήτριες μπορούμε να προσεγγίσουμε και την κατανομή τους με τη μέθοδο των πυρήνων. Το μη

[1] Gia èna tuqaîo deîgma $x_1, \ldots, x_N$, h empeirik sunˆrthsh katanom c se èna shmeîo x dînetai apì ton tôpo $\hat{F}_N(x) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{x_i \leq x\} = \frac{(\text{pl joc twn } x_i \leq x)}{N}$.

παραμετρικό Bootstrap είναι μία γενική μέθοδος που εφαρμόζεται σε τέτοιου είδους προβλήματα και δεν χρειάζεται να κάνουμε καμία υπόθεση για την κατανομή που ακολουθεί το δείγμα. Παρ΄ όλα αυτά, σε περιπτώσεις όπου γνωρίζουμε εξαρχής από ποιά κατανομή προέρχονται τα δεδομένα ή η υπόθεση για μια συγκεκριμένη κατανομή αυτών είναι λογική, τότε είναι προτιμότερο να χρησιμοποιήσουμε το παραμετρικό Bootstrap. Στην περίπτωση της γραμμικής παλινδρόμησης μπορούμε να υποθέσουμε ότι τα σφάλματα θα ακολουθούν την κανονική κατανομή, έστω $N(0, \sigma^2)$. Επομένως μπορούμε να προσαρμόσουμε το μοντέλο μας με τη μέθοδο ελαχίστων τετραγώνων ή με τη μέθοδο Lasso και να πάρουμε αρχικές εκτιμήσεις για τις παραμέτρους. Στη συνέχεια με σταθερό τον πίνακα σχεδιασμού $\mathbf{X}$, προσομοιώνουμε N τιμές (όσες το αρχικό δείγμα) για την εξαρτημένη μεταβλητή μέσω της σχέσης:

$$y_i \quad N(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}, \hat{\sigma}^2) \tag{3.3.2}$$

όπου $\hat{\boldsymbol{\beta}}$ η εκτιμήτρια που υπολογίσαμε αρχικά και $\hat{\sigma}^2$ είναι η δειγματική διασπορά των υπολοίπων. Επαναλαμβάνουμε την παραπάνω διαδικασία, έστω B φορές, οπότε έχουμε στη διάθεσή μας B Bootstrap δείγματα μεγέθους N. Σε κάθε ένα από αυτά υπολογίζουμε την εκτιμήτρια Lasso, οπότε καταλήγουμε σε B διαφορετικές εκτιμήτριες, άρα είμαστε σε θέση να κάνουμε συμπερασματολογία. Το παραμετρικό Bootstrap βασίζεται σε ισχυρότερες υποθέσεις όσον αφορά την κατανομή του δείγματος, αλλά δίνει πιο ακριβείς προβλέψεις όταν το αρχικό μοντέλο που υποθέτουμε είναι σωστό.

3.4 Προσομοίωση

Προκειμένου να εξετάσουμε τις μεθόδους επαναδειγματοληψίας που αναφέραμε, θα προσομοιώσουμε τιμές από ένα γραμμικό μοντέλο. Για το πλήθος των παρατηρήσεων του δείγματος υποθέτουμε ότι $N = 30$ και για το πλήθος των επεξηγηματικών μεταβλητών $p = 50$. Για την κατανομή των ανεξάρτητων μεταβλητών υποθέτουμε ότι $\mathbf{X} \quad N_{50}(\mathbf{0}, \boldsymbol{\Sigma})$, όπου ο πίνακας διασποράς-συνδιασποράς είναι $\boldsymbol{\Sigma}_{i,j} = 0.5^{|i \ j|}$ για $i, j = 1, \ldots, 50$. Δηλαδή για κάθε ανεξάρτητη μεταβλητή θα είναι $X_j \quad N(0, 1)$ και επιπλέον $\mathrm{Cov}(X_i, X_j) = 0.5^{|i \ j|}$. Η μεταβλητή απόκρισης εξαρτάται μόνο από τις μεταβλητές X_1, X_6, X_{15} και θέτουμε το διάνυσμα των συντελεστών τους να είναι $(\beta_1, \beta_6, \beta_{15}) = (4.2, \ 3.7, 2.3)$. Οι συντελεστές των υπόλοιπων μεταβλητών ισούνται με το 0. Επιπλέον η σταθερά για το μοντέλο μας θα είναι $\beta_0 = 3$. Για τα σφάλματα υποθέτουμε ότι $\varepsilon_i \quad N(0, 1)$ για κάθε $i = 1, \ldots, 30$. Το μοντέλο μας θα γράφεται στην παρακάτω μορφή:

$$\mathbf{y} = \beta_0 \mathbf{1}_N + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{3.4.1}$$

Με τον παρακάτω κώδικα προσομοιώνουμε τιμές με βάση τη σχέση αυτή, χρησιμοποιώντας το πακέτο mvtnorm στην R. Αρχικά θέτουμε ένα random seed ώστε τα αποτελέσματά μας να είναι αναπαραγωγίσιμα.

```
> set.seed(1)
> library(mvtnorm)
> n=30
> p=50
> Sigma=matrix(0,p,p)
> for(i in 1:p){
   for(j in 1:p){
     Sigma[i,j]=0.5^(abs(i-j))
   }
```

```
  }
> x=rmvnorm(n,rep(0,p),Sigma)
> y=3+4.2*x[,1]-3.7*x[,6]+2.3*x[,15]+rnorm(n)
> df=data.frame(x,y)
```

Εν συνεχεία θα προσαρμόσουμε το μοντέλο μας με τη μέθοδο Lasso και θα εκτιμήσουμε τους συντελεστές των επεξηγηματικών μεταβλητών καθώς και τη σταθερά. Για τα παρακάτω χρησιμοποιούμε το πακέτο glmnet.

```
> #lasso estimate
> library(glmnet)
> x=model.matrix(df$y .,data=df)[,-1]
> lasso=glmnet(x,y)
> lasso.cv=cv.glmnet(x,y) #10fold-CV
> minlam=lasso.cv$lambda.min
> lambda.1se=lasso.cv$lambda.1se
> print(c(minlam,lambda.1se))
[1] 0.03704074 0.37912342
```

Με την εντολή model.matrix() κατασκευάζουμε αρχικά τον πίνακα σχεδιασμού x αφαιρώντας την πρώτη στήλη με τις μονάδες που αντιστοιχεί στο σταθερό όρο. Στη συνέχεια με την εντολή glmnet(), στην οποία θέτουμε σαν ορίσματα τον πίνακα x και το διάνυσμα της επεξηγηματικής μεταβλητής y, προσαρμόζουμε το μοντέλο με τη μέθοδο Lasso. Με την εντολή αυτή μπορούμε επίσης να προσαρμόσουμε το μοντέλο και με άλλες τεχνικές, όπως Ridge και Elastic Net, δίνοντας ανάλογη τιμή στο όρισμα alpha (π.χ για alpha = 1 (default) προσαρμόζουμε το μοντέλο με Lasso, ενώ για alpha = 0 με Ridge). Η εντολή glmnet() επιλέγει αυτόματα ένα πλέγμα τιμών για την παράμετρο λ (μπορούμε να το ρυθμίσουμε με το όρισμα lambda) και επίσης (by default) κάνει τυποποίηση των μεταβλητών (όρισμα standardize με τιμές TRUE (default) ή FALSE). Με τη χρήση της εντολής cv.glmnet() εκτελούμε 10 fold Cross Validation (μπορούμε να ρυθμίσουμε την επιλογή αυτή από το όρισμα nfolds) και εκχωρούμε το αποτέλεσμα στο αντικείμενο lasso.cv. Εντοπίζουμε τις τιμές της παραμέτρου ποινής $\lambda_{\text{min}}, \lambda_{\text{1se}}$ και τις εμφανίζουμε με την εντολή print(). Η πρώτη τιμή 0.037 (minlam) αντιστοιχεί στο ελάχιστο σφάλμα Cross Validation. Η τιμή 0.379 αντιστοιχεί στην τιμή lambda.1se και είναι η μεγαλύτερη τιμή του λ για την οποία το αντίστοιχο σφάλμα της είναι εντός ενός τυπικού σφάλματος του ελάχιστου. Γενικά η τιμή αυτή είναι μεγαλύτερη από την τιμή lambda.min επομένως οδηγεί σε πιο φειδωλά μοντέλα. Παρακάτω με την εντολή coef() υπολογίζουμε τους συντελεστές του μοντέλου. Ως ορίσματα θέτουμε το αντικείμενο lasso και με την τιμή s καθορίζουμε την παράμετρο ποινής για την οποία θέλουμε να υπολογίσουμε τους συντελεστές. Οι συντελεστές επιστρέφονται για τις αρχικές (μη τυποποιημένες) μεταβλητές. Εδώ χρησιμοποιούμε την τιμή (minlam) που ελαχιστοποιεί το σφάλμα cross-validation.

```
> #coefficients of model with min CV error
> lasso.coef=coef(lasso,s=minlam)
> print(lasso.coef)
51 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  3.563117190
X1           4.068519168
X2           .
X3           .
```

```
X4             .
X5          -0.194167138
X6          -3.567950434
X7             .
X8             .
X9          -0.002303783
X10            .
X11            0.128651037
X12            .
X13            .
X14            0.217729787
X15            2.045189711
X16            .
X17            0.036930312
X18            .
X19            0.121079099
X20         -0.370481607
X21            0.027856021
X22            .
X23            .
X24         -0.324551797
X25            .
X26         -0.013609738
X27            .
X28         -0.273523665
X29            .
X30            .
X31            .
X32            .
X33            .
X34            0.382531232
X35            .
X36            .
X37         -0.237102471
X38         -0.139343561
X39         -0.289135052
X40            0.171835676
X41            .
X42         -0.248582848
X43            0.027688932
X44            .
X45            .
X46            .
X47            .
X48            0.196844673
X49            0.038304751
```

```
X50             .
```

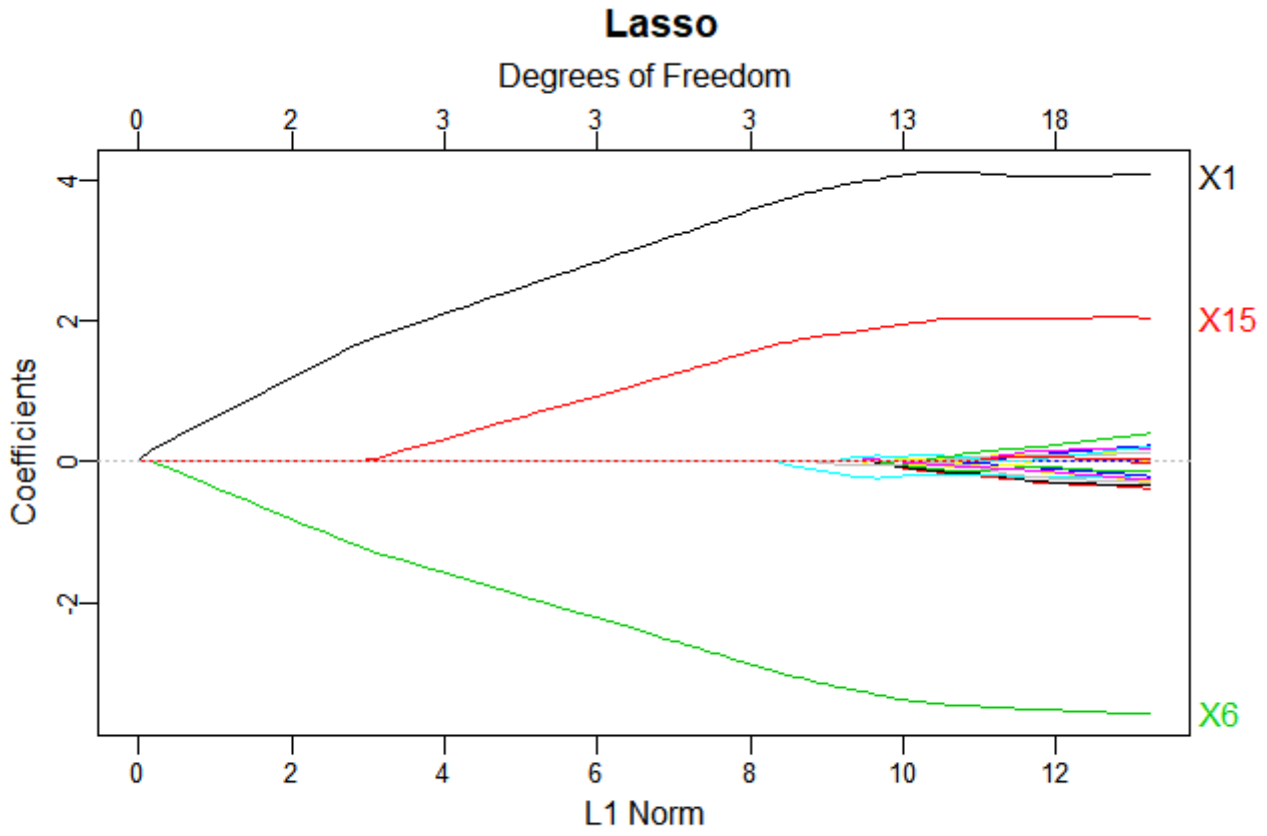
Από τα παραπάνω αποτελέσματα παρατηρούμε ότι οι συντελεστές που προκύπτουν με τη μέθοδο Lasso είναι σχετικά κοντά στις πραγματικές τιμές του μοντέλου. Για παράδειγμα ο συντελεστής της μεταβλητής X_6 εκτιμάται από την τιμή -3.57 (με στρογγυλοποίηση) και η πραγματική τιμή είναι -3.7. Αρκετοί συντελεστές εκτιμούνται από την τιμή 0 (όπου τελεία ο αντίστοιχος συντελεστής ισούται με 0), πράγμα που σημαίνει ότι η μέθοδος Lasso παράγει ένα αραιό μοντέλο. Όμως βλέπουμε ότι πολλοί συντελεστές είναι διάφοροι του μηδενός (αλλά κοντά στο μηδέν), ενώ στο πραγματικό μοντέλο οι αντίστοιχες επεξηγηματικές μεταβλητές δεν έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης. Με τις παρακάτω εντολές παίρνουμε το Διάγραμμα 3.2.

```
> install.packages("plotmo")
> library(plotmo)
> plot_glmnet(lasso,xvar="norm",label=3,col=1:50)
> axis(side=3,labels=FALSE)
> title(main="Lasso",line=3)
```

Στο Διάγραμμα αυτό παρουσιάζεται το μονοπάτι για τους συντελεστές του μοντέλου σε σχέση με τη νόρμα l_1 (coe cient path). Όπως είναι λογικό, για μικρές τιμές της νόρμας (αντίστοιχα μεγάλες τιμές της παραμέτρου λ) παρατηρούμε ότι όλοι οι συντελεστές είναι μηδενικοί εκτός από αυτούς που αντιστοιχούν στις μεταβλητές X_1, X_6, X_{15}. Καθώς αυξάνεται η τιμή της l_1-νόρμας (ισοδύναμα μειώνεται η τιμή της παραμέτρου συντονισμού) βλέπουμε ότι όλο και περισσότεροι συντελεστές είναι μη μηδενικοί και άρα περισσότερες επεξηγηματικές μεταβλητές συμμετέχουν στο μοντέλο (οι βαθμοί ελευθερίας στον πάνω άξονα αντιστοιχούν στο πλήθος των ανεξάρτητων μεταβλητών που περιέχονται στο μοντέλο). Επιπλέον με τις παρακάτω εντολές παίρνουμε την καμπύλη που φαίνεται στο Διάγραμμα 3.3.

```
> min(lasso.cv$cvm)
[1] 1.836076
> (lasso.cv$cvm)[which(lasso.cv$lambda==lambda.1se)]
[1] 2.382199
> plot(lasso.cv)
> title(main="Cross Validation Curve",line=3)
```

Στο διάνυσμα cvm του αντικειμένου lasso.cv περιέχονται τα σφάλματα Cross Validation για κάθε τιμή του λ, που έχει χρησιμοποιηθεί για την προσαρμογή του μοντέλου. Με την εντολή which(lasso.cv$lambda==lambda.1se) εντοπίζουμε τη θέση στην οποία η παράμετρος ποινής ισούται με την τιμή lambda.1se, και στη συνέχεια παίρνουμε το αντίστοιχο σφάλμα. Οι τιμές για το ελάχιστο σφάλμα CV και για το σφάλμα που αντιστοιχεί στην παράμετρο lambda.1se είναι 1.836 και 2.382 αντίστοιχα. Στο Διάγραμμα 3.3 φαίνεται η καμπύλη του σφάλματος Cross Validation σε σχέση με το λογάριθμο της παραμέτρου ποινής λ, για την περίπτωση του 10-fold CV. Η αριστερή κάθετη (dotted) γραμμή αντιστοιχεί στο λ που οδηγεί στο ελάχιστο σφάλμα, ενώ η δεξιά κάθετη (dotted) γραμμή αντιστοιχεί στο λ «ενός τυπικού σφάλματος». Επίσης από τον πάνω άξονα βλέπουμε ότι το μοντέλο που προκύπτει με το ελάχιστο σφάλμα θα περιέχει 23 ανεξάρτητες μεταβλητές (και τη σταθερά), ενώ το μοντέλο που προσαρμόζεται χρησιμοποιώντας την τιμή του lambda.1se περιέχει μόλις 4 παράγοντες και το σταθερό όρο.

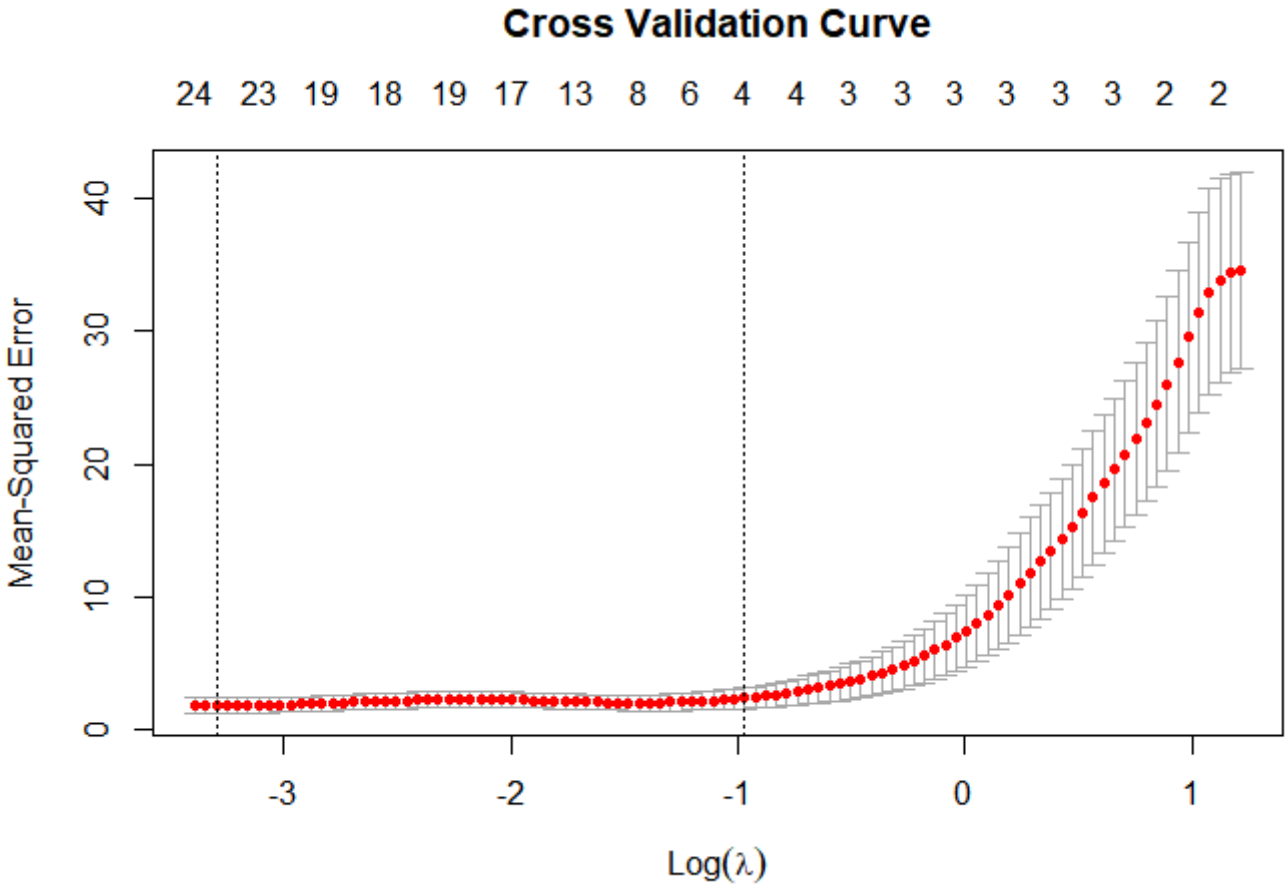


Διάγραμμα 3.2: Το μονοπάτι των συντελεστών σε σχέση με την l_1-νόρμα, για το προσομοιωμένο μοντέλο με χρήση της μεθόδου Lasso. Οι μεταβλητές X_1, X_6, X_{15} έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης.

3.4.1 Μη παραμετρικό Bootstrap

Μέχρι στιγμής έχουμε υπολογίσει την εκτιμήτρια Lasso για το προσομοιωμένο μας παράδειγμα, χρησιμοποιώντας την παράμετρο ποινής λ που ελαχιστοποιεί το σφάλμα Cross Validation. Στη συνέχεια θα εφαρμόσουμε το μη παραμετρικό Bootstrap για να βγάλουμε ορισμένα συμπεράσματα για την εκτίμησή μας. Θα χρησιμοποιήσουμε το πακέτο boot και συγκεκριμένα τη συνάρτηση boot() που ανήκει σε αυτό. Αρχικά κατασκευάζουμε μια συνάρτηση που υπολογίζει το στατιστικό ενδιαφέροντος. Στη συγκεκριμένη περίπτωση μας ενδιαφέρουν οι συντελεστές του μοντέλου. Στη συνέχεια καλούμε τη συνάρτηση boot(), μέσω της οποίας κάνουμε δειγματοληψία με επανάθεση από το αρχικό δείγμα και σε κάθε επανάληψη Bootstrap υπολογίζουμε τους συντελεστές με τη μέθοδο Lasso.

```
> #-----------------------------------
> #-----------------------------------
> #NON-PARAMETRIC BOOTSTRAP
> install.packages("boot")
> library(boot)
> non_coef_stat=function(data,index){
```



Διάγραμμα 3.3: Η καμπύλη του σφάλματος Cross Validation στην περίπτωση 10-fold CV για το προσομοιωμένο μας παράδειγμα. Η αριστερή κάθετη γραμμή αντιστοιχεί στη τιμή της παραμέτρου λ που οδηγεί στο ελάχιστο CV error ενώ η δεξιά αντιστοιχεί στο λ που προκύπτει από τον κανόνα του ενός τυπικού σφάλματος.

```
  x=model.matrix(data$y .,data)[,-1]
  y=data$y
  lasso.fit=cv.glmnet(x[index,],y[index])
  minlam=lasso.fit$lambda.min
  return(as.numeric(coef(lasso.fit,s=minlam)))
}
```

Με τον παραπάνω κώδικα αρχικά κατασκευάζουμε τη συνάρτηση με όνομα non_coef_stat, η οποία δέχεται ως ορίσματα ένα πλαίσιο δεδομένων data καθώς και έναν δείκτη index που θα χρησιμοποιηθεί για τη δειγματοληψία με επανάθεση. Κατασκευάζουμε τον πίνακα σχεδιασμού **x** και τη μεταβλητή y (που είναι η στήλη y του data). Στη συνέχεια προσαρμόζουμε το μοντέλο χρησιμοποιώντας συγκεκριμένες γραμμές του πίνακα x και τα αντίστοιχα στοιχεία της στήλης y που ορίζονται μέσω του index. Εκτελούμε 10-fold CV και βρίσκουμε το λ (minlam) που αντιστοιχεί στο ελάχιστο σφάλμα. Τέλος, η συνάρτηση επιστρέφει τους συντελεστές του μοντέλου, τους οποίους έχουμε μετατρέψει σε αριθμητικό διάνυσμα με την εντολή as.numeric(). Για παράδειγμα με την παρακάτω εντολή εμφανίζονται η σταθερά και οι 5 πρώτοι συντελεστές του μοντέλου.

```
> head(non_coef_stat(df,1:nrow(df)))
[1]  3.563  4.068 0.000  0.000  0.000 -0.194
```

Στη συνέχεια καλούμε τη συνάρτηση boot() η οποία δέχεται ως ορίσματα το πλαίσιο δεδομένων df, τη συνάρτηση non_coef_stat που υλοποιήσαμε παραπάνω καθώς και τον αριθμό R των επαναλήψεων που θέλουμε να εκτελεστούν. Εδώ παράγουμε 1000 νέα Bootstrap δείγματα με επανάθεση από το αρχικό μας δείγμα.

```
> #1000 replications
> non_boot.obj=boot(df,non_coef_stat,R=1000)
> print(non_boot.obj)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = df, statistic = non_coef_stat, R = 1000)


Bootstrap Statistics :
         original         bias    std. error
t1*   3.563117190 -0.2610693242  0.32308010
t2*   4.068519168 -0.4082874408  0.64986938
t3*   0.000000000  0.0872673730  0.25512868
t4*   0.000000000  0.0252348102  0.10712545
t5*   0.000000000  0.0027487828  0.10943031
t6*  -0.194167138  0.1115066458  0.15688136
t7*  -3.567950434  0.4070171060  0.37698337
t8*   0.000000000 -0.0476848230  0.14835436
t9*   0.000000000 -0.0042113434  0.09542799
t10* -0.002303783 -0.0850300883  0.26933599
t11*  0.000000000 -0.0040123168  0.05386172
t12*  0.128651037  0.0046533783  0.21867437
t13*  0.000000000  0.0794963933  0.16494744
t14*  0.000000000 -0.0126325870  0.08771906
t15*  0.217729787 -0.1641074493  0.12320570
t16*  2.045189711 -0.4157704451  0.47349968
```

Παραπάνω βλέπουμε τα αποτελέσματα που προκύπτουν από το μη παραμετρικό Bootstrap (εδώ παρουσιάζουμε μόνο τα στατιστικά για τη σταθερά t1* και τους 15 πρώτους συντελεστές t2*,...,t16* ενώ στην πραγματικότητα εμφανίζονται τα στατιστικά για όλους του συντελεστές t1*,...,t51*). Η πρώτη στήλη (original) παρουσιάζει τις αρχικές μας εκτιμήσεις που προέκυψαν από τη μέθοδο Lasso. Η δεύτερη στήλη (bias) περιλαμβάνει τις εκτιμήσεις της μεροληψίας για κάθε έναν συντελεστή, ενώ η τρίτη στήλη περιλαμβάνει τις εκτιμήσεις των τυπικών σφαλμάτων για κάθε συντελεστή. Για παράδειγμα ο συντελεστής $\hat{\beta}_1$ αρχικά εκτιμάται από την τιμή 4.068 και η μεροληψία του από την τιμή 0.408. Το τυπικό του σφάλμα εκτιμάται από την τιμή 0.649. Οι εκτιμητές που έχουν προκύψει σε κάθε μία από τις 1000 επαναλήψεις βρίσκονται στον πίνακα t της λίστας non_boot.obj. Οπότε εκχωρούμε στον πίνακα non_boot_data όλους τους bootstrap συ-

ντελεστές εκτός της πρώτης στήλης στην οποία βρίσκεται ο σταθερός όρος (αφαιρούμε την πρώτη στήλη ώστε στη συνέχεια να κάνουμε γραφήματα μόνο για τους συντελεστές των ανεξάρτητων μεταβλητών).

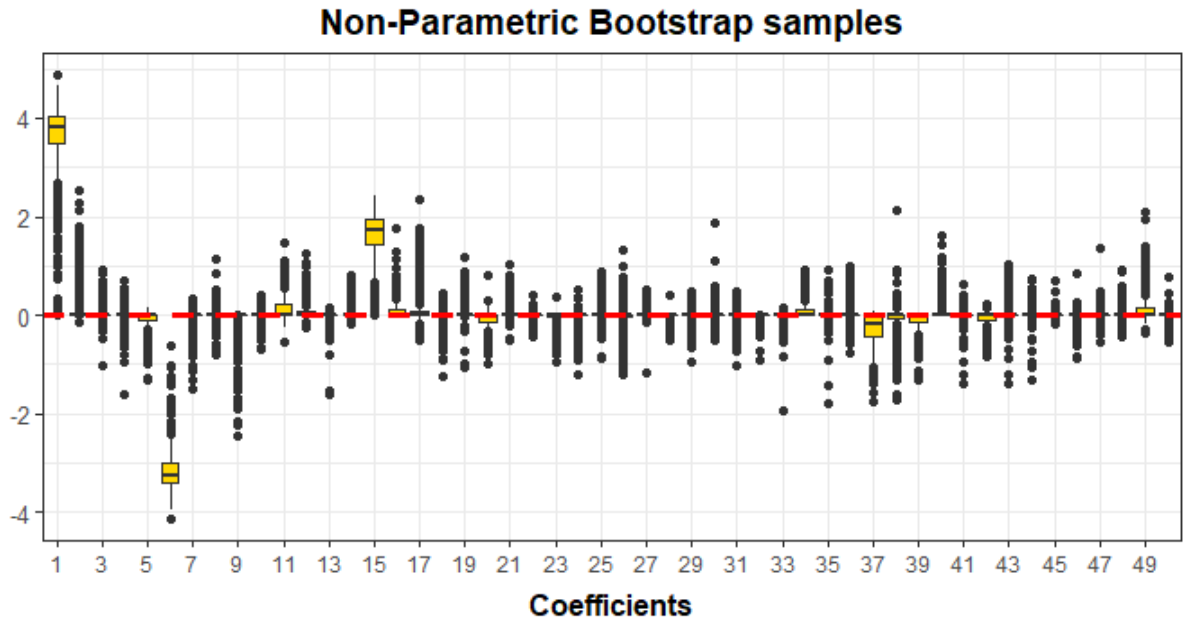
```
> #bootstrap coefficients without intercept
> non_boot_data=(non_boot.obj$t)[,-1]
```

Στη συνέχεια χρησιμοποιούμε την εντολή melt() (από το πακέτο reshape2) ώστε όλες οι στήλες του πίνακα των συντελεστών να στοιβαχτούν σε μία στήλη με όνομα Var2 και οι αντίστοιχες τιμές σε μία νέα στήλη με όνομα value. Ο μετασχηματισμός αυτός θα χρησιμοποιηθεί για την κατασκευή του θηκοδιαγράμματος (boxplot) με χρήση του πακέτου ggplot2. Παρακάτω δίνονται οι σχετικές εντολές.

```
> install.packages("reshape2")
> install.packages("ggplot2")
> library(reshape2)
> library(ggplot2)
> non_melt=melt(non_boot_data)

> #boxplot
> box=ggplot(non_melt,aes(x=factor(Var2),y=value))+
      geom_boxplot(fill="gold")
  box=box+scale_x_discrete(name="Coefficients",
                           breaks=seq(1,50,by=2))+
   scale_y_continuous(name="",breaks=seq(-4,4.5,by=2),
     limits=c(min(non_melt$value),max(non_melt$value)))+
   geom_hline(yintercept = 0,color="red",
              size=1.2,linetype="dashed")+
   theme_bw()+ggtitle("Non-Parametric Bootstrap samples")+
   theme(plot.title=element_text(hjust=0.5,face="bold",size=14),
         axis.title.x=element_text(face ="bold",size=12),
         axis.text.y=element_text(size = 10))
> box
```

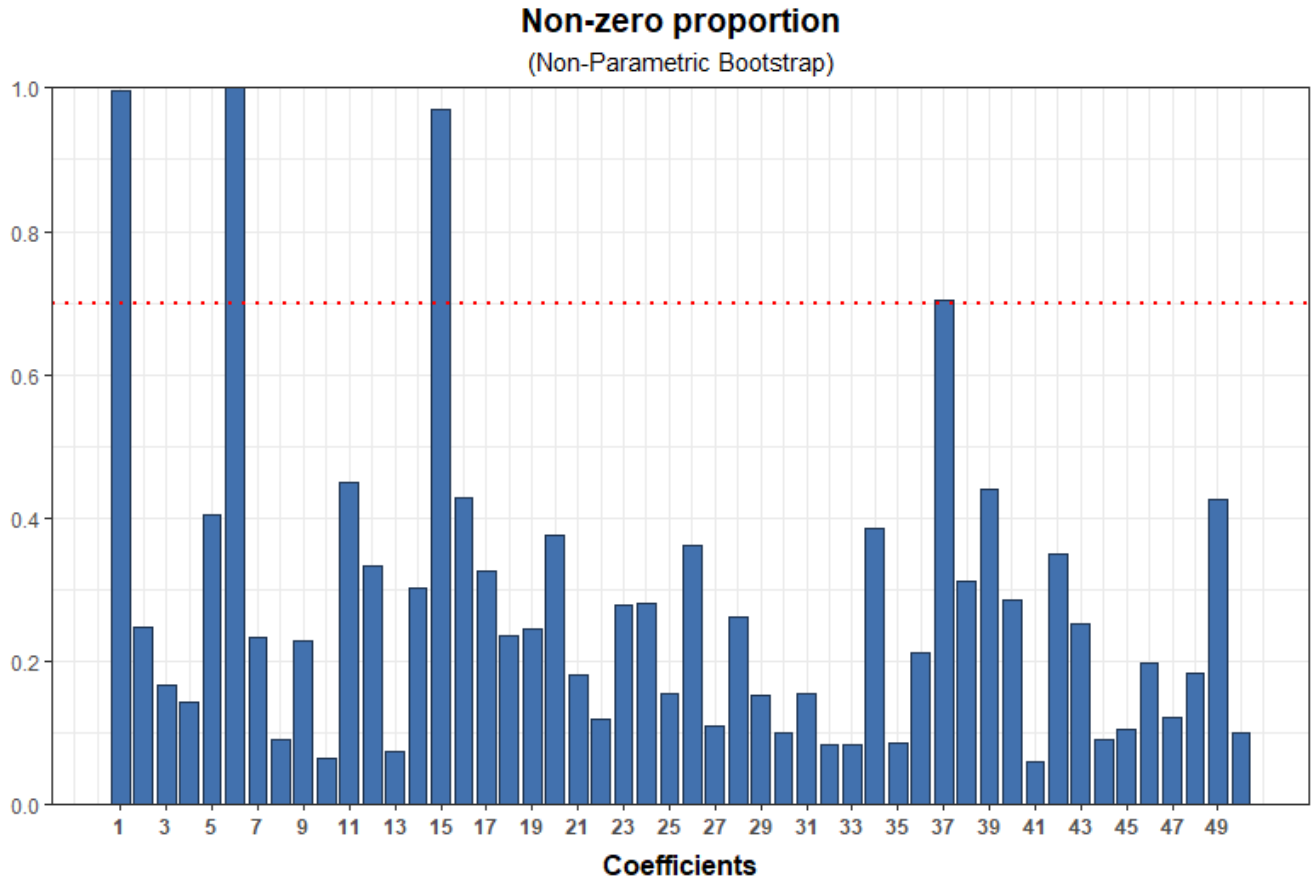
Στο Διάγραμμα 3.4 παρουσιάζονται τα θηκοδιαγράμματα των συντελεστών που έχουν προκύψει από το μη παραμετρικό Bootstrap. Παρατηρούμε ότι οι διάμεσοι των συντελεστών $\hat{\beta}_1, \hat{\beta}_6, \hat{\beta}_{15}$ βρίσκονται σχετικά κοντά στις πραγματικές τους τιμές. Όμως βλέπουμε ότι και οι περισσότεροι από τους συντελεστές των υπόλοιπων μεταβλητών που δεν έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης, εκτιμούνται κάποιες φορές ως μη μηδενικοί στις 1000 επαναλήψεις που εκτελέσαμε. Μπορούμε λοιπόν να κατασκευάσουμε ραβδογράμματα προκειμένου να δούμε πόσες φορές ο κάθε συντελεστής είναι μη μηδενικός στις 1000 επαναλήψεις. Με τις παρακάτω εντολές κατασκευάζουμε το Διάγραμμα 3.5. Στο διάνυσμα zero_proportion εκχωρούμε σε κάθε θέση το ποσοστό για κάθε συντελεστή. Ουσιαστικά μετράμε πόσες φορές ο κάθε συντελεστής είναι μη μηδενικός στα Bootstrap δείγματα και έπειτα διαιρούμε με το συνολικό πλήθος επαναλήψεων (1000). Στο Διάγραμμα 3.5 φαίνονται τα ραβδογράμματα (barplot) για τους 50 συντελεστές. Το ύψος κάθε ράβδου ισούται με την μη μηδενική αναλογία κάθε συντελεστή. Η κόκκινη οριζόντια (dotted) γραμμή βρίσκεται στην τιμή 0.7.

```
#count proportion each coefficient is NOT zero
```



Διάγραμμα 3.4: Θηκοδιαγράμματα των συντελεστών που έχουν προκύψει ύστερα από 1000 επαναλήψεις με εφαρμογή του μη παραμετρικού Bootstrap.

```
> zero_proportion=rep(0,50)
> for(i in 1:50){
   c=which(non_boot_data[,i]!=0)
   zero_proportion[i]=length(c)/1000
  }

> bar=ggplot(data.frame(zero_proportion),
     aes(x=seq_along(zero_proportion),y=zero_proportion))+
     geom_bar(stat="identity",fill="#4271AE",col="#1F3552",
              width=0.8)
> bar=bar+scale_x_continuous(name="Coefficients",
  labels=paste(seq(1,50,by=2)),breaks=seq(1,50,by=2),
  limits=c(0.5,50.5))+scale_y_continuous(name="",expand=c(0,0),
  limits=c(0,1),breaks = seq(0,1,0.2))+
  ggtitle("Non-zero proportion",
          subtitle ="(Non-Parametric Bootstrap)")+
  geom_hline(yintercept = .7,color="red",
             linetype="dotted",size=1)+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5,
        face="bold",size=14),
        plot.subtitle = element_text(hjust = 0.5),
        axis.text.x = element_text(face="bold"),
        axis.title.x = element_text(size = 12,face="bold",
        vjust=-1.1))
> bar
```



Διάγραμμα 3.5: Ραβδογράμματα μη μηδενικής αναλογίας των συντελεστών, που έχουν προκύψει ύστερα από 1000 επαναλήψεις με εφαρμογή του μη παραμετρικού Bootstrap. Η κόκκινη οριζόντια γραμμή αντιστοιχεί σε αναλογία 0.7.

Όπως είναι αναμενόμενο οι συντελεστές των παραγόντων X_1, X_6, X_{15} είναι σχεδόν όλες τις φορές μη μηδενικοί. Επίσης βλέπουμε ότι η εκτίμηση του συντελεστή της μεταβλητής X_{37} είναι αρκετές φορές μη μηδενική (περίπου 700 φορές στις 1000). Το ποσοστό των υπόλοιπων συντελεστών είναι περίπου από το 46% και κάτω, πράγμα που σημαίνει ότι τις περισσότερες φορές εκτιμούνται ως μηδενικοί. Άρα συμπεραίνουμε ότι χρησιμοποιώντας το μη παραμετρικό Bootstrap οι τιμές στις 1000 επαναλήψεις για τους συντελεστές $\hat{\beta}_1, \hat{\beta}_6, \hat{\beta}_{15}$ είναι ναι μεν συγκεντρωμένες γύρω από τις πραγματικές τους τιμές, αλλά βλέπουμε ότι κάποιες φορές συμμετέχουν και επιπλέον μεταβλητές στο μοντέλο μας, ενώ θα έπρεπε οι συντελεστές τους να είναι μηδενικοί.

3.4.2 Παραμετρικό Bootstrap

Εφόσον λοιπόν γνωρίζουμε την κατανομή από την οποία προέρχεται η μεταβλητή απόκρισης (κανονική κατανομή), είναι προτιμότερο να προσομοιώσουμε τιμές για τα Bootstrap δείγματά μας με βάση αυτή την κατανομή. Θα εφαρμόσουμε δηλαδή το παραμετρικό Bootstrap για να κάνουμε συμπερασματολογία για τις εκτιμήτριες του μοντέλου μας. Για να εφαρμόσουμε τώρα το παραμετρικό Bootstrap, θα χρησιμοποιήσουμε την αρχική εκτιμήτρια Lasso, έστω $\hat{\boldsymbol{\beta}}$, καθώς και την τυπική απόκλιση $\hat{\theta}$ των σφαλμάτων $e_i = y_i - \hat{y}_i$, $i = 1, \ldots, 30$. Με σταθερό τον πίνακα σχεδιασμού $\mathbf{X}$ προσομοιώνουμε τιμές για την εξαρτημένη μεταβλητή σύμφωνα με τη σχέση $y_i \sim N(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}, \hat{\theta}^2)$ για

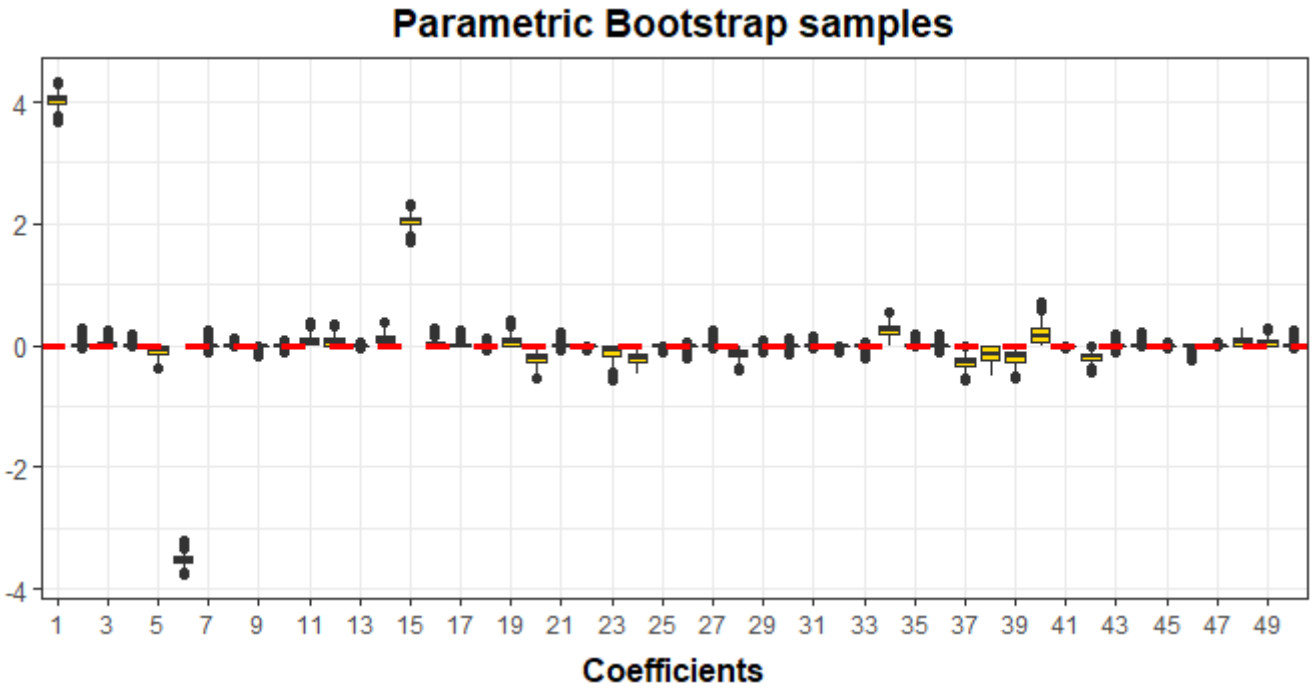
κάθε i = 1, . . . , 30. Στο νέο μας τώρα δείγμα (Y , $X_1, \ldots, X_{50}$) θα υπολογίσουμε την εκτιμήτρια Lasso για την παράμετρο ποινής που ελαχιστοποιεί το σφάλμα Cross Validation. Επαναλαμβάνοντας τη διαδικασία B φορές καταλήγουμε τελικά σε B νέες εκτιμήτριες που έχουν προκύψει σύμφωνα με το παραμετρικό Bootstrap. Με τον παρακάτω κώδικα αρχικά υπολογίζουμε τις τιμές $\hat{y}_i = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}$ και τις αποθηκεύουμε στη μεταβλητή y_hat (στο αντικείμενο lasso.coef βρίσκονται οι συντελεστές που έχουν προκύψει από την αρχική προσαρμογή του μοντέλου με τη μέθοδο Lasso). Ύστερα υπολογίζουμε την τυπική απόκλιση $\hat{\sigma}$ (sigma_hat) των σφαλμάτων. Στη συνέχεια με τον βρόχο for θα δημιουργήσουμε τον πίνακα μεγέθους 1000 × (p + 1) με όνομα par_beta που θα περιέχει τις εκτιμήτριες Lasso που προέκυψαν ύστερα από 1000 επαναλήψεις. Οπότε προσομοιώνουμε N = 30 τιμές για τη μεταβλητή απόκρισης με χρήση της εντολής rnorm() και τις αποθηκεύουμε στο διάνυσμα y.boot. Προσαρμόζουμε το μοντέλο και εκτελούμε 10-fold CV με την εντολή cv.glmnet() και τέλος αποθηκεύουμε σε κάθε γραμμή του πίνακα τους συντελεστές του μοντέλου για την τιμή της παραμέτρου λ που αντιστοιχεί στο ελάχιστο σφάλμα (με την εντολή as.numeric() μετατρέπουμε το διάνυσμα σε αριθμητικό).

```
> #-----------------------------------
> #-----------------------------------
> #PARAMETRIC BOOTSTRAP
> y_hat=lasso.coef[1]+x%*%lasso.coef[-1]
> sigma_hat=sd(y-y_hat)
> par_beta=matrix(0,1000,p+1)
> for(i in 1:1000){
   y.boot=rnorm(n,mean=y_hat,sd=sigma_hat)
   lasso.fit=cv.glmnet(x,y.boot)
   minlam=lasso.fit$lambda.min
   par_beta[i,]=as.numeric(coef(lasso.fit,s=minlam))
  }
```

Με τις παρακάτω εντολές κατασκευάζουμε το Διάγραμμα 3.6. Στο Διάγραμμα αυτό φαίνονται τα θηκοδιαγράμματα των συντελεστών ύστερα από την εφαρμογή του παραμετρικού Bootstrap. Παρατηρούμε ότι οι τιμές των συντελεστών είναι περισσότερο συγκεντρωμένες γύρω από τις αντίστοιχες πραγματικές τους τιμές, απ΄ ότι στο μη παραμετρικό Bootstrap. Επίσης είναι φανερό ότι όλοι οι συντελεστές των μεταβλητών του μοντέλου (εκτός των X_1, X_6, X_{15} που έχουν ουσιαστική επίδραση στη μεταβλητή απόκρισης) βρίσκονται τις περισσότερες φορές πολύ κοντά στο 0. Το αποτέλεσμα αυτό είναι αναμενόμενο, αφού τώρα προσομοιώνουμε τιμές από την πραγματική κατανομή που ακολουθούν τα δεδομένα και όχι από την εμπειρική κατανομή που αποτελεί μια πιο γενική εκτίμηση της πραγματικής κατανομής. Βέβαια εδώ εξαρτώμαστε από την αρχική μας εκτιμήτρια Lasso. Όσο πιο κοντά στην πραγματική τιμή βρίσκεται, αναμένουμε οι συντελεστές που θα προκύψουν με το παραμετρικό Bootstrap να είναι ακόμα πιο συγκεντρωμένοι γύρω από τις πραγματικές τους τιμές.

```
> par_boot_data=par_beta[,-1] #without b0 for plots
> par_melt=melt(par_boot_data)

> box2=ggplot(par_melt,aes(x=factor(Var2),y=value))+
              geom_boxplot(fill="gold")
> box2=box2+scale_x_discrete(name="Coefficients",
       breaks=seq(1,50,by=2))+scale_y_continuous(name="",
```

```
      breaks=seq(-4,4.5,by=2),
      limits = c(min(par_melt$value),
      max(par_melt$value)))+
  geom_hline(yintercept = 0,color="red",size=1.2,
             linetype="dashed")+theme_bw()+
  ggtitle("Parametric Bootstrap samples")+
  theme(plot.title = element_text(hjust=0.5,
        face="bold",size = 14),
        axis.title.x = element_text(face="bold",size=12,
        vjust=-1.1),
        axis.text.y =element_text(size=10))
> box2
```



Διάγραμμα 3.6: Θηκοδιαγράμματα των συντελεστών που έχουν προκύψει ύστερα από 1000 επαναλήψεις με εφαρμογή του παραμετρικού Bootstrap. Οι τιμές των συντελεστών είναι περισσότερο συγκεντρωμένες γύρω από τις πραγματικές τους τιμές σε σχέση με το μη παραμετρικό Bootstrap.

3.4.3 Bootstrap από τα υπόλοιπα

Ένας άλλος τρόπος για να κάνουμε συμπερασματολογία για τις εκτιμήτριες Lasso είναι να χρησιμοποιήσουμε τη μέθοδο Bootstrap από τα υπόλοιπα. Πιο συγκεκριμένα, υπολογίζουμε αρχικά τα υπόλοιπα $e_i = y_i \quad \hat{y}_i, i = 1, \ldots, 30$ και τα αποθηκεύουμε στη μεταβλητή res. Στη συνέχεια αρχικοποιούμε τον πίνακα coef_res, όπου σε κάθε του γραμμή θα εκχωρήσουμε τις εκτιμήτριες Lasso για κάθε μία από τις 1000 Bootstrap επαναλήψεις. Μέσα στο βρόχο for και με χρήση της εντολής sample κάνουμε δειγματοληψία με επανάθεση από το αρχικό δείγμα των υπολοίπων και στη συνέχεια εκχωρούμε στη μεταβλητή y_boot_res τιμές για τη μεταβλητή απόκρισης με βάση τη

σχέση $y_i = \hat{y}_i + e_i, i = 1, \ldots, 30$. Με την εντολή cv.glmnet και ορίσματα τον πίνακα σχεδιασμού x και τις Bootstrap τιμές της μεταβλητής απόκρισης, προσαρμόζουμε το μοντέλο με τη μέθοδο Lasso και εκτελούμε 10-fold CV (ο πίνακας σχεδιασμού $\mathbf{X}$ παραμένει σταθερός). Τέλος, εκχωρούμε σε κάθε γραμμή του πίνακα coef_res το αριθμητικό διάνυσμα των συντελεστών για την τιμή της παραμέτρου ποινής που ελαχιστοποιεί το σφάλμα Cross Validation.

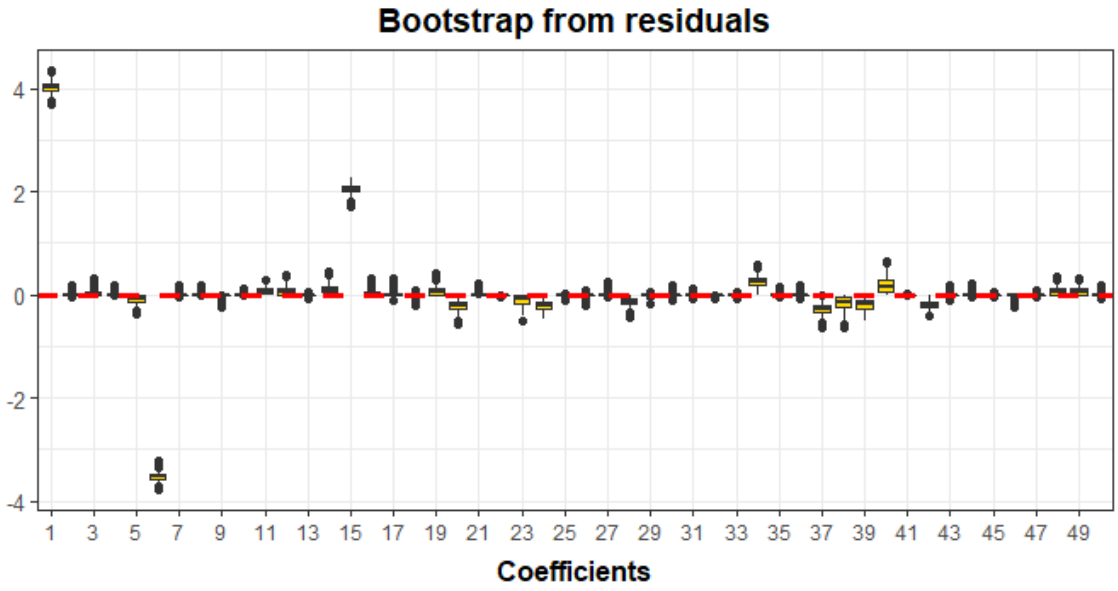
```
> #-----------------
> #BOOTSTRAP from residuals
> res=y-y_hat
> coef_res=matrix(0,1000,p+1)
> for(i in 1:1000){
    boot_res=sample(res,n,replace = TRUE)
    y_boot_res=y_hat+boot_res
    lasso.boot=cv.glmnet(x,y_boot_res)
    coef_res[i,]=as.numeric(coef(lasso.boot,s="lambda.min"))
  }
```

Με τις παρακάτω εντολές κατασκευάζουμε το Διάγραμμα 3.7. Στο Διάγραμμα αυτό παρουσιάζονται τα θηκοδιαγράμματα των συντελεστών του μοντέλου ύστερα από εφαρμογή της μεθόδου Bootstrap από τα υπόλοιπα. Παρατηρούμε ότι και σε αυτή την περίπτωση οι συντελεστές των μεταβλητών X_1, X_6, X_{15} είναι περισσότερο συγκεντρωμένοι στις πραγματικές τους τιμές και ότι οι υπόλοιποι συντελεστές είναι κοντά στο 0, όπως και στο παραμετρικό Bootsrap. Η εφαρμογή της μεθόδου Bootstrap από τα υπόλοιπα είναι στατιστικά ορθότερη από το μη παραμετρικό Bootstrap, διότι ο πίνακας σχεδιασμού $\mathbf{X}$ είναι γνωστός από πριν και παραμένει σταθερός στις προσομοιώσεις που κάνουμε.

```
> melt_res=melt(coef_res[,-1])
> box3=ggplot(melt_res,aes(x=factor(Var2),y=value))+
       geom_boxplot(fill="gold")
> box3=box3+scale_x_discrete(name="Coefficients",
       breaks=seq(1,50,by=2))+ylab("")+
  geom_hline(yintercept = 0,color="red",size =1.2,
             linetype="dashed")+
  theme_bw()+
  ggtitle("Bootstrap from residuals")+
  theme(plot.title=element_text(hjust=0.5,face="bold",size=14),
       axis.title.x=element_text(face="bold",size=12,vjust=-1.1),
       axis.text.y=element_text(size=10))
> box3
```

3.4.4 Συμπερασματολογία με χρήση 100 δειγμάτων

Προκειμένου να εξάγουμε συμπεράσματα για τις εκτιμήτριες του γραμμικού μοντέλου μας, μπορούμε να χρησιμοποιήσουμε και τις μεθόδους Ridge, Elastic Net και Adaptive Lasso για να το προσαρμόσουμε. Αρχικά θα προσομοιώσουμε 100 δείγματα μεγέθους $N = 30$ και $p = 50$ όπως ακριβώς κάναμε και με τη μέθοδο Lasso. Για κάθε μία μέθοδο θα προσαρμόσουμε το μοντέλο και θα πάρουμε τις αντίστοιχες εκτιμήτριες. Χρησιμοποιώντας τον παρακάτω κώδικα αρχικοποιούμε τους (100×51) πίνακες lasso_matrix, ridge_matrix, el_net_matrix και ad_lasso_matrix, ώστε τελικά



Διάγραμμα 3.7: Θηκοδιαγράμματα των συντελεστών που έχουν προκύψει ύστερα από 1000 επαναλήψεις με εφαρμογή της μεθόδου Bootstrap από τα υπόλοιπα.

σε κάθε γραμμή τους να περιέχουν τις αντίστοιχες εκτιμήτριες για κάθε ένα από τα 100 δείγματα. Με χρήση του βρόχου for προσομοιώνουμε κάθε φορά ένα τα 100 δείγματα κατασκευάζοντας τον πίνακα που περιέχει τις ανεξάρτητες μεταβλητές x (με χρήση της εντολής rmvnorm) καθώς και την μεταβλητή απόκρισης y. Ύστερα με την εντολή cv.glmnet με ορίσματα τα x,y καθώς και την παράμετρο alpha εκτελούμε 10-fold CV. Για παράδειγμα, για τη μέθοδο Ridge θα θέσουμε alpha = 0 ενώ για τη μέθοδο Elastic Net χρησιμοποιούμε την παράμετρο alpha = 0.5. Για την μέθοδο Adaptive Lasso πρώτα θα υπολογίσουμε τους συντελεστές που προκύπτουν από τη μέθοδο Lasso και θα τους αποθηκεύσουμε στο αριθμητικό διάνυσμα με όνομα lasso.coef (χωρίς τη σταθερά). Υπενθυμίζουμε ότι η ποινή για τη μέθοδο Adaptive Lasso είναι $\lambda \sum_{j=1}^{p} w_j \beta_j$ με $w_j = 1/j\hat{\beta}_{\text{init},j}j$, οπότε είναι σα να προσαρμόζουμε το μοντέλο με τη μέθοδο Lasso, αλλά με αρχικά βάρη w_j για κάθε συντελεστή. Αυτό μπορούμε να το ρυθμίσουμε χρησιμοποιώντας το όρισμα penalty.factor στην εντολή cv.glmnet. Τέλος σε κάθε γραμμή των πινάκων που έχουμε κατασκευάσει αποθηκεύουμε τους αντίστοιχους συντελεστές. Οι συντελεστές αυτοί είναι υπολογισμένοι για την τιμή της παραμέτρου λ (lambda.min) που ελαχιστοποιεί το σφάλμα Cross Validation.

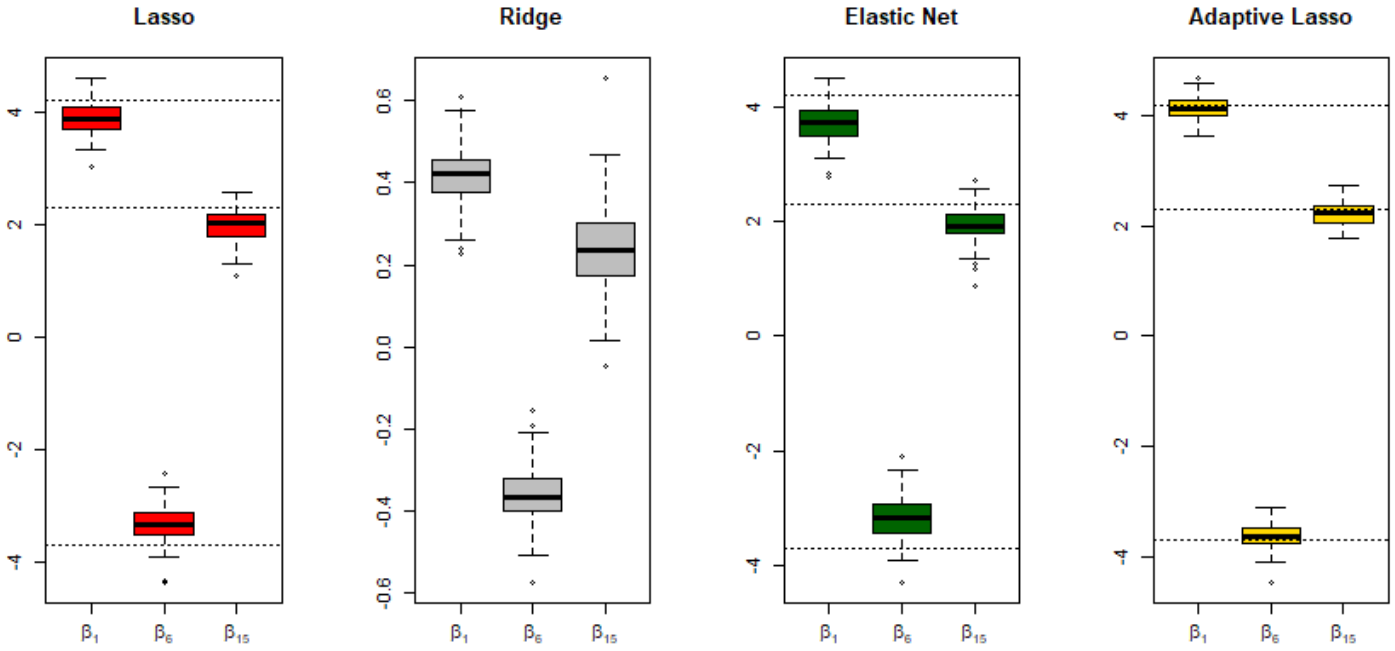
```
> set.seed(2)
> lasso_matrix=matrix(0,100,51)
> ridge_matrix=matrix(0,100,51)
> el_net_matrix=matrix(0,100,51)
> ad_lasso_matrix=matrix(0,100,51)
> for(i in 1:100){
   x=rmvnorm(30,mean=rep(0,50),Sigma)
   y=3+4.2*x[,1]-3.7*x[,6]+2.3*x[,15]+rnorm(30)

   lasso=cv.glmnet(x,y)
   ridge=cv.glmnet(x,y,alpha=0)
   el_net=cv.glmnet(x,y,alpha=0.5)
```

```
   lasso.coef=(as.numeric(coef(lasso,s="lambda.min")))[-1]
   ad_lasso=cv.glmnet(x,y,penalty.factor=1/abs(lasso.coef))

   lasso_matrix[i,]=as.numeric(coef(lasso,s="lambda.min"))
   ridge_matrix[i,]=as.numeric(coef(ridge,s="lambda.min"))
   el_net_matrix[i,]=as.numeric(coef(el_net,s="lambda.min"))
   ad_lasso_matrix[i,]=as.numeric(coef(ad_lasso,s="lambda.min"))
 }
```

Στη συνέχεια με τις παρακάτω εντολές κατασκευάζουμε το Διάγραμμα 3.8.

```
> par(mfrow=c(1,4))
> expr=c(expression(beta[1]),expression(beta[6]),
         expression(beta[15]))
> boxplot(lasso_matrix[,2],lasso_matrix[,7],
          lasso_matrix[,16],main="Lasso",col="red")
> axis(side=1,at=c(1,2,3),labels=expr)
> abline(h=c(4.2,-3.7,2.3),lty="dotted")
> boxplot(ridge_matrix[,2],ridge_matrix[,7],
          ridge_matrix[,16],main="Ridge",col="gray")
> axis(side=1,at=c(1,2,3),labels=expr)
> abline(h=c(4.2,-3.7,2.3),lty="dotted")
> boxplot(el_net_matrix[,2],el_net_matrix[,7],
          el_net_matrix[,16],main="Elastic Net",
          col="darkgreen")
> axis(side=1,at=c(1,2,3),labels= expr)
  abline(h=c(4.2,-3.7,2.3),lty="dotted")
> boxplot(ad_lasso_matrix[,2],ad_lasso_matrix[,7],
          ad_lasso_matrix[,16],main="Adaptive Lasso",
          col="gold")
> axis(side=1,at=c(1,2,3),labels=expr)
> abline(h=c(4.2,-3.7,2.3),lty="dotted")
```

Στο Διάγραμμα αυτό παρουσιάζονται τα θηκοδιαγράμματα των συντελεστών των σημαντικών μεταβλητών του μοντέλου (X_1, X_6, X_{15}) για τα 100 δείγματα που προσομοιώσαμε. Για κάθε μία μέθοδο, οι οριζόντιες (dotted) γραμμές αντιστοιχούν στις πραγματικές τιμές των συντελεστών, δηλαδή στις τιμές β_1 = 4.2, β_6 = 3.7 και β_{15} = 2.3. Παρατηρούμε ότι οι μέθοδοι Lasso και Elastic Net έχουν περίπου όμοια αποτελέσματα και οι διάμεσοι των συντελεστών δε διαφοροποιούνται σε μεγάλο βαθμό από τις πραγματικές τιμές. Η προσαρμογή του μοντέλου με τη μέθοδο Adaptive Lasso φαίνεται να είναι η βέλτιστη, μιας και οι τιμές των συντελεστών είναι αρκετά συγκεντρωμένες γύρω από τις πραγματικές τους τιμές (ενδεικτικά αναφέρουμε ότι η διάμεσος των συντελεστών της μεταβλητής X_1 είναι 4.14, δηλαδή πολύ κοντά στη τιμή 4.2). Τέλος, βλέπουμε ότι η μέθοδος Ridge δίνει αποτελέσματα μακριά από τις πραγματικές τιμές άρα για το προσομοιωμένο μας παράδειγμα δεν είναι μια καλή μέθοδος προσαρμογής. Τα αποτελέσματα αυτά για τη μέθοδο Ridge είναι αναμενόμενα αφού αν προσαρμόσουμε το μοντέλο μας με αυτή τη μέθοδο τότε και οι 50 συντελεστές θα εκτιμούνται ως μη μηδενικοί, ενώ στην πραγματικότητα μόνο 3 από αυτούς είναι διάφοροι του μηδενός. Οι μέθοδοι Lasso και Elastic Net δίνουν σαφώς καλύτερες εκτιμήσεις εφόσον μπορούν να κάνουν ταυτόχρονα και επιλογή μεταβλητών αλλά και εκτίμηση των παραμέτρων του μοντέλου.



Διάγραμμα 3.8: Θηκοδιαγράμματα των συντελεστών των σημαντικών μεταβλητών για το προσομοιωμένο μας παράδειγμα, με χρήση 100 δειγμάτων. Η μέθοδος Adaptive Lasso δίνει εκτιμήτριες πολύ κοντά στις πραγματικές τιμές, ενώ η μέθοδος Ridge δίνει εκτιμήτριες μακριά από τις πραγματικές τιμές. Οι οριζόντιες (dotted) γραμμές αντιστοιχούν στους πραγματικούς συντελεστές των μεταβλητών X_1, X_6, X_{15}.

Τέλος, η μέθοδος Adaptive Lasso παράγει σαφώς ένα πιο αραιό μοντέλο το οποίο όπως βλέπουμε είναι και το πιο ακριβές με την έννοια ότι οι συντελεστές που προκύπτουν είναι πολύ κοντά στις πραγματικές τιμές τους.

3.5 Συμπέρασμα

Στο Κεφάλαιο αυτό είδαμε πως με ορισμένες μεθόδους επαναδειγματοληψίας, μπορούμε να εξάγουμε συμπεράσματα όσον αφορά τους εκτιμητές που προκύπτουν από διάφορες τεχνικές συρρίκνωσης. Αρχικά, παρουσιάσαμε τη Μπεϋζιανή προσέγγιση των μεθόδων Lasso και Ridge, σύμφωνα με την οποία οι παράμετροι του μοντέλου μπορούν να θεωρηθούν τυχαίες ποσότητες οι οποίες έχουν μια εκ των προτέρων κατανομή. Με τεχνικές MCMC μπορούμε να κάνουμε δειγματοληψία από την εκ των υστέρων κατανομή των παραμέτρων και να βγάλουμε συμπεράσματα για αυτές. Επίσης, από το προσομοιωμένο μας παράδειγμα και με χρήση του μη παραμετρικού Bootstrap, είδαμε τον τρόπο με τον οποίο μπορούμε να κάνουμε δειγματοληψία με επανάθεση από το αρχικό μας δείγμα. Υπολογίζοντας σε κάθε ένα από τα Bootstrap δείγματά μας την εκτιμήτρια Lasso, καταφέραμε να εκτιμήσουμε το τυπικό της σφάλμα και να δούμε που κυμαίνονται οι εκτιμήτριες των συντελεστών μας μέσω του αντίστοιχου θηκοδιαγράμματος. Με το παραμετρικό Bootstrap προσομοιώσαμε τιμές από την κανονική κατανομή για τη μεταβλητή απόκρισης και ύστερα επαναλάβαμε τη διαδικασία 1000 φορές με σκοπό να κάνουμε συμπερασματολογία για τις εκτιμήτριες των συντελεστών

του μοντέλου. Καταλήξαμε λοιπόν σε πιο ορθά αποτελέσματα από το μη παραμετρικό Bootstrap αφού προσομοιώσαμε τιμές με βάση το μοντέλο που γέννησε τα δεδομένα. Επίσης σε ορθότερα αποτελέσματα καταλήξαμε και με εφαρμογή της μεθόδου Bootstrap από τα υπόλοιπα. Τέλος, συγκρίναμε τις μεθόδους Lasso, Ridge, Elastic Net και Adaptive Lasso με βάση 100 προσομοιωμένα δείγματα και είδαμε ότι η μέθοδος Adaptive Lasso δίνει καλύτερα αποτελέσματα για το μοντέλο μας, με την έννοια ότι οι εκτιμήτριες που προκύπτουν είναι πιο κοντά στις πραγματικές τιμές που χρησιμοποιήσαμε.

Kefˆlaio 4

Efarmog se pragmatikˆ dedomèna

4.1 Εισαγωγή

Στο Κεφάλαιο αυτό θα εφαρμόσουμε ορισμένες από τις τεχνικές συρρίκνωσης που αναλύσαμε στα προηγούμενα Κεφάλαια, χρησιμοποιώντας ένα πραγματικό σύνολο δεδομένων. Τα δεδομένα που θα αναλύσουμε προέρχονται από την ιστοσελίδα data.world (link: https://data.world/nrippner/ols-regression-challenge) και περιλαμβάνουν συγκεντρωτικά στοιχεία από τις κομητείες της Αμερικής. Σκοπός μας είναι να κατασκευάσουμε ένα μοντέλο για την πρόβλεψη του ποσοστού θνησιμότητας που οφείλεται στη νόσο του καρκίνου, στις κομητείες των Η.Π.Α. Τα δεδομένα αποτελούνται από $N = 3047$ παρατηρήσεις και $p = 34$ επεξηγηματικές μεταβλητές οι οποίες είναι οι εξής:

TARGET_deathRate: Εξαρτημένη μεταβλητή. Μέσος κατά κεφαλήν (ανά 100,000 άτομα) αριθμός θανάτων εξαιτίας της νόσου του καρκίνου (α)

avgAnnCount: Μέσος αριθμός αναφερόμενων περιστατικών καρκίνου που διαγιγνώσκονται ετησίως (α)

avgDeathsPerYear: Μέσος αριθμός αναφερόμενων θανάτων λόγω του καρκίνου (α)

incidenceRate: Μέσος κατά κεφαλήν (ανά 100,000) αριθμός διαγνώσεων καρκίνου (α)

medianIncome: Διάμεσος του εισοδήματος για κάθε κομητεία (β)

popEst2015: Πληθυσμός κάθε κομητείας (β)

povertyPercent: Ποσοστό πληθυσμού στη φτώχεια (β)

studyPerCap: Κατά κεφαλήν αριθμός κλινικών δοκιμών για τον καρκίνο για κάθε κομητεία (α)

binnedInc: Κατά κεφαλήν εισόδημα (σε διάστημα) (β)

MedianAge: Διάμεσος ηλικίας των κατοίκων της κομητείας (β)

MedianAgeMale: Διάμεσος ηλικίας των ανδρών της κομητείας (β)

MedianAgeFemale: Διάμεσος ηλικίας των γυναικών της κομητείας (β)

Geography: Κομητεία (β)

AvgHouseholdSize: Μέσο μέγεθος νοικοκυριού κάθε κομητείας (β)

PercentMarried: Ποσοστό κατοίκων της κομητείας που είναι παντρεμένοι (β)

PctNoHS18_24: Ποσοστό κατοίκων της κομητείας ηλικίας 18-24 ετών που δεν έχουν τελειώσει το λύκειο (β)

PctHS18_24: Ποσοστό κατοίκων της κομητείας ηλικίας 18-24 ετών με υψηλότερη βαθμίδα εκπαίδευσης το λύκειο (β)

PctSomeCol18_24: Ποσοστό κατοίκων της κομητείας ηλικίας 18-24 ετών με υψηλότερη βαθμίδα εκπαίδευσης: πτυχίο κολλεγίου (β)

PctBachDeg18_24: Ποσοστό κατοίκων της κομητείας ηλικίας 18-24 ετών με υψηλότερη βαθμίδα εκπαίδευσης: πτυχίο Bachelor (β)

PctHS25_Over: Ποσοστό κατοίκων της κομητείας ηλικίας 25 ετών και άνω με υψηλότερη βαθμίδα εκπαίδευσης το λύκειο (β)

PctBachDeg25_Over: Ποσοστό κατοίκων της κομητείας ηλικίας 25 ετών και άνω με υψηλότερη βαθμίδα εκπαίδευσης: πτυχίο Bachelor (β)

PctEmployed16_Over: Ποσοστό κατοίκων της κομητείας ηλικίας 16 ετών και άνω που εργάζονται (β)

PctUnemployed16_Over: Ποσοστό κατοίκων της κομητείας ηλικίας 16 ετών και άνω που δεν εργάζονται (β)

PctPrivateCoverage: Ποσοστό κατοίκων της κομητείας με ιδιωτική κάλυψη υγείας (β)

PctPrivateCoverageAlone: Ποσοστό κατοίκων της κομητείας με ιδιωτική κάλυψη υγείας (χωρίς δημόσια βοήθεια) (β)

PctEmpPrivCoverage: Ποσοστό κατοίκων της κομητείας με ιδιωτική κάλυψη υγείας που παρέχεται από τους εργοδότες στους υπαλλήλους (β)

PctPublicCoverage: Ποσοστό κατοίκων της κομητείας στους οποίους η κυβέρνηση παρέχει κάλυψη υγείας (β)

PctPubliceCoverageAlone: Ποσοστό κατοίκων της κομητείας για τους οποίους παρέχεται κάλυψη υγείας από την κυβέρνηση αλλά συνεισφέρουν και οι ίδιοι (β)

PctWhite: Ποσοστό κατοίκων της κομητείας που αναγνωρίζονται ως Λευκοί (β)

PctBlack: Ποσοστό κατοίκων της κομητείας που αναγνωρίζονται ως Αφροαμερικανοί (β)

PctAsian: Ποσοστό κατοίκων της κομητείας που αναγνωρίζονται ως Ασιάτες (β)

PctOtherRace: Ποσοστό κατοίκων της κομητείας που δεν αναγνωρίζονται ως Λευκοί, Αφροαμερικανοί ή Ασιάτες (β)

PctMarriedHouseholds: Ποσοστό παντρεμένων νοικοκυριών (β)

BirthRate: Αριθμός γεννήσεων σε σχέση με τον αριθμό των γυναικών της κομητείας (β)

(α) Έτη: 2010-2016, (β) Εκτιμήσεις απογραφής για το έτος 2013.
Στη συνέχεια παρουσιάζουμε τη διαδικασία κατασκευής του μοντέλου μας, κάνοντας πρώτα μία διερευνητική ανάλυση των δεδομένων μας.

4.2 Επεξεργασία δεδομένων-Υπολογισμός ελλιπών τιμών

Αρχικά φορτώνουμε ορισμένες βιβλιοθήκες της R (εφόσον τις έχουμε εγκαταστήσει), τις οποίες θα χρησιμοποιήσουμε για την ανάλυση των δεδομένων μας. Στη συνέχεια φορτώνουμε τα δεδομένα μας με την εντολή read.csv() και τα αποθηκεύουμε στο πλαίσιο δεδομένων με όνομα data και διαστάσεων 3047 γραμμών και 34 στηλών. Με την εντολή View(data) μπορούμε να δούμε τα δεδομένα αυτά σε ένα νέο παράθυρο.

```
> library(ggplot2)
> library(corrplot)
> library(gridExtra)
> library(rms)
> library(dplyr)
> library(tidyr)
> library(glmnet)
> library(reshape2)

> data=read.csv("C:/Users/User/Documents/Data/cancer_reg.csv",
                header = TRUE,stringsAsFactors = FALSE)
> dim(data)
[1] 3047   34
> View(data)
```

Παρατηρούμε ότι σε ορισμένες στήλες υπάρχουν ελλιπείς τιμές (missing values) με τον συμβολισμό NA. Με την παρακάτω εντολή υπολογίζουμε το πλήθος αυτών των τιμών σε κάθε στήλη και τις διατάσσουμε σε φθίνουσα σειρά (εδώ εμφανίζουμε μόνο τα ονόματα και το πλήθος των ελλιπών τιμών για τρεις μεταβλητές, μιας και οι υπόλοιπες δεν έχουν ελλιπείς τιμές).

```
> sort(colSums(is.na(data)),decreasing = TRUE)
  PctSomeCol18_24   PctPrivateCoverageAlone   PctEmployed16_Over
             2285                       609                  152
```

Η στήλη με όνομα PctSomeCol18_24 που εκφράζει το ποσοστό κατοίκων της κομητείας ηλικίας 18-24 που έχουν αποφοιτήσει από κάποιο κολλέγιο, έχει 2285 NAs, δηλαδή περίπου 75% των παρατηρήσεων. Επομένως χρειάζεται να αφαιρέσουμε τη στήλη αυτή από τα δεδομένα μας. Χρησιμοποιούμε την εντολή select από το πακέτο dplyr. Η στήλη PctPrivateCoverageAlone που εκφράζει το ποσοστό των κατοίκων της κομητείας με ιδιωτική κάλυψη υγείας (χωρίς δημόσια βοήθεια), έχει 609 ελλιπείς τιμές. Θα μπορούσαμε να αφαιρέσουμε τις γραμμές του data στις οποίες η μεταβλητή αυτή περιέχει NA. Εναλλακτικά όμως εκτιμούμε αυτές τις τιμές από τη διάμεσο των

αντίστοιχων ποσοστών σε κάθε πολιτεία της Αμερικής. Για παράδειγμα το πρώτο στοιχείο της μεταβλητής PctPrivateCoverageAlone είναι NA (με αντίστοιχη πολιτεία την Washington). Θα πάρουμε τη διάμεσο των ποσοστών στις θέσεις όπου η πολιτεία είναι η Washington και έτσι εκτιμούμε την τιμή NA. Αρχικά χρειάζεται να χωρίσουμε τη στήλη Geography σε δύο νέες στήλες με ονόματα County/City και State. Αυτό επιτυγχάνεται με χρήση της εντολής separate. Ύστερα με το βρόχο for εκτιμούμε με τη διαδικασία που περιγράψαμε τις ελλιπείς τιμές για την μεταβλητή PctPrivateCoverageAlone. Με όμοιο τρόπο επιλέγουμε να εκτιμήσουμε και τις τιμές NA για την μεταβλητή PctEmployed16_Over (ποσοστό κατοίκων της κομητείας ηλικίας 16 και άνω που εργάζονται). Ύστερα από τις εκτιμήσεις αυτές βλέπουμε ότι δεν υπάρχουν ελλιπείς τιμές στα δεδομένα μας.

```
> data=data %>% select(-PctSomeCol18_24)
> data=data %>% separate(Geography,into=c("County/City","State"),
                         sep=", ")
> for(i in 1:3047){
    if(is.na(data$PctPrivateCoverageAlone[i])){
      data$PctPrivateCoverageAlone[i]=median(
      data$PctPrivateCoverageAlone[data$State==data$State[i]],
      na.rm = TRUE)
    }
  }
> for(i in 1:3047){
    if(is.na(data$PctEmployed16_Over[i])){
      data$PctEmployed16_Over[i]=median(
      data$PctEmployed16_Over[data$State==data$State[i]],
      na.rm = TRUE)
    }
  }
> sum(is.na(data)) #check no NA's now
[1] 0
```

Επιπλέον παρατηρούμε ότι στα δεδομένα μας υπάρχουν και ορισμένες λανθασμένες-μη λογικές τιμές όσον αφορά την μεταβλητή MedianAge. Δεν είναι λογικό π.χ. η διάμεσος της ηλικίας των κατοίκων μιας κομητείας να είναι άνω των 80. Επομένως θα χρησιμοποιήσουμε την εντολή Iter για να αφαιρέσουμε τις γραμμές του data στις οποίες η διάμεσος της ηλικίας είναι π.χ. άνω των 80 (μη λογική τιμή). Επίσης επιλέγουμε να αφαιρέσουμε από τα δεδομένα μας τις κατηγορικές μεταβλητές County/City και binnedInc. Η μεταβλητή County/City θα περιέχει πάρα πολλά επίπεδα (ένα για κάθε μία κομητεία) και ουσιαστικά για να αποφύγουμε τη χρήση τόσου μεγάλου πλήθους επιπέδων στην ανάλυσή μας, θα κάνουμε χρήση της μεταβλητής State (πολιτεία) που αποτελεί μια σύμπτυξη των κομητειών. Τέλος αφαιρούμε τη μεταβλητή binnedInc εφόσον έχουμε στη διάθεσή μας και ακριβείς τιμές της διαμέσου του εισοδήματος (medianIncome) για κάθε κομητεία. Καταλήγουμε σε ένα πλαίσιο δεδομένων με 3017 γραμμές και 32 στήλες.

```
> summary(data$MedianAge)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.30   37.70   41.00   45.27   44.00  624.00
> data=data %>% filter(!MedianAge>80)
> data=data %>% select(-c("County/City",binnedInc))
```

```
> dim(data)
  [1] 3017   32
```

4.3 Απεικόνιση δεδομένων

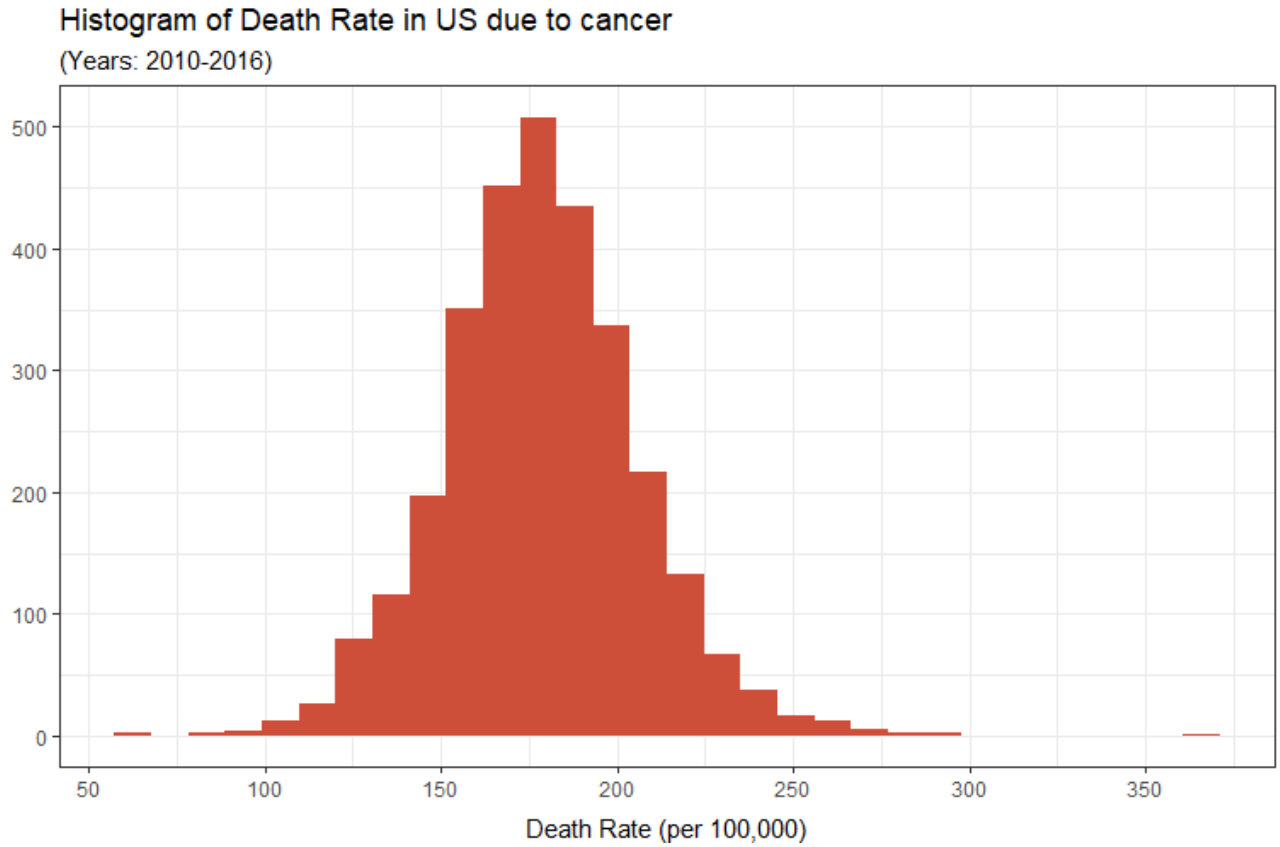
Προχωρούμε τώρα στην υλοποίηση ορισμένων διαγραμμάτων που θα μας βοηθήσουν να κατανοήσουμε καλύτερα τη δομή των δεδομένων μας, προτού προχωρήσουμε στην μοντελοποίηση. Με τις παρακάτω εντολές και αφού ορίσουμε το επιθυμητό φόντο για τα διαγράμματά μας, κατασκευάζουμε το ιστόγραμμα της εξαρτημένης μεταβλητής TARGET_deathRate (βλ. Διάγραμμα 4.1). Παρατηρούμε ότι κατά μέσο όρο για τα έτη 2010-2016, ο αριθμός (ανά 100,000 άτομα) των θανάτων που οφείλονται στη νόσο του καρκίνου συγκεντρώνεται περισσότερο στο διάστημα από 160 έως 195. Η διάμεσος και η μέση τιμή του αριθμού των θανάτων είναι 178.1 και 178.6 αντίστοιχα.

```
> theme_set(theme_bw())
> summary(data$TARGET_deathRate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   59.7   161.3   178.1   178.6   195.2   362.8
> ggplot(data,aes(x=TARGET_deathRate))+
    geom_histogram(fill="tomato3")+
    scale_x_continuous(breaks=seq(0,400,by=50))+
    labs(title = "Histogram of Death Rate in US due to cancer",
    subtitle="(Years: 2010-2016)",
    x="Death Rate (per 100,000)",y="")+
    theme(axis.title.x = element_text(vjust = -1.3))
```

Προκειμένου να εξετάσουμε τυχόν συσχετίσεις μεταξύ των δεδομένων μας, θα υπολογίσουμε τον πίνακα συσχετίσεων (correlation matrix)[1] και στη συνέχεια θα κατασκευάσουμε το διάγραμμα συσχέτισης. Με αυτό το διάγραμμα εξετάζουμε μόνο πιθανή γραμμική συσχέτιση μεταξύ των μεταβλητών μας. Μπορεί για παράδειγμα δύο μεταβλητές να συνδέονται μη γραμμικά μεταξύ τους και ο συντελεστής συσχέτισης τους να είναι κοντά στο 0. Με τον παρακάτω κώδικα υπολογίζουμε αρχικά ποιές μεταβλητές από τα δεδομένα μας έχουν αριθμητικές τιμές και τις αποθηκεύουμε στη μεταβλητή num_Var. Ύστερα υπολογίζουμε τον πίνακα συσχετίσεων correlation. Η διαγώνιος του πίνακα αυτού θα περιέχει μονάδες, αφού κάθε μια μεταβλητή είναι τέλεια γραμμικά συσχετισμένη με τον εαυτό της. Για να βρούμε τυχόν συσχετίσεις με τη μεταβλητή απόκρισης πηγαίνουμε στη στήλη 3 του πίνακα συσχέτισης και με την εντολή sapply υπολογίζουμε τις απόλυτες τιμές των συντελεστών συσχέτισης με τη μεταβλητή απόκρισης. Στη συνέχεια αποθηκεύουμε στη μεταβλητή corHigh τα ονόματα των μεταβλητών των οποίων η απόλυτη τιμή του συντελεστή συσχέτισης με τη μεταβλητή TARGET_deathRate είναι άνω του 0.4. Τέλος με χρήση της εντολής corrplot.mixed (πακέτο corrplot) κατασκευάζουμε το διάγραμμα συσχέτισης για αυτές τις μεταβλητές (Διάγραμμα 4.2).

```
> num_Var=which(sapply(data,is.numeric))
> correlation=cor(data[,num_Var])
> cor_abs=sapply(correlation[,3],function(x) abs(x))
```

[1] Ta stoiqeła tou pínaka autoÔ eínai oi deigmatikoí suntelestèc susqètishc $r_{xy} = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\sum_{i=1}^{n}(x_i - x)^2 \sum_{i=1}^{n}(y_i - y)^2}$ kai kumaínontai sto diˆsthma [-1,1]. Gia timèc kontˆ sto 1 -1 upˆrqei megˆlh grammik exˆrthsh metaxÔ twn x, y en¸ gia timèc kontˆ sto 0 den upˆrqei isqur grammik exˆrthsh.

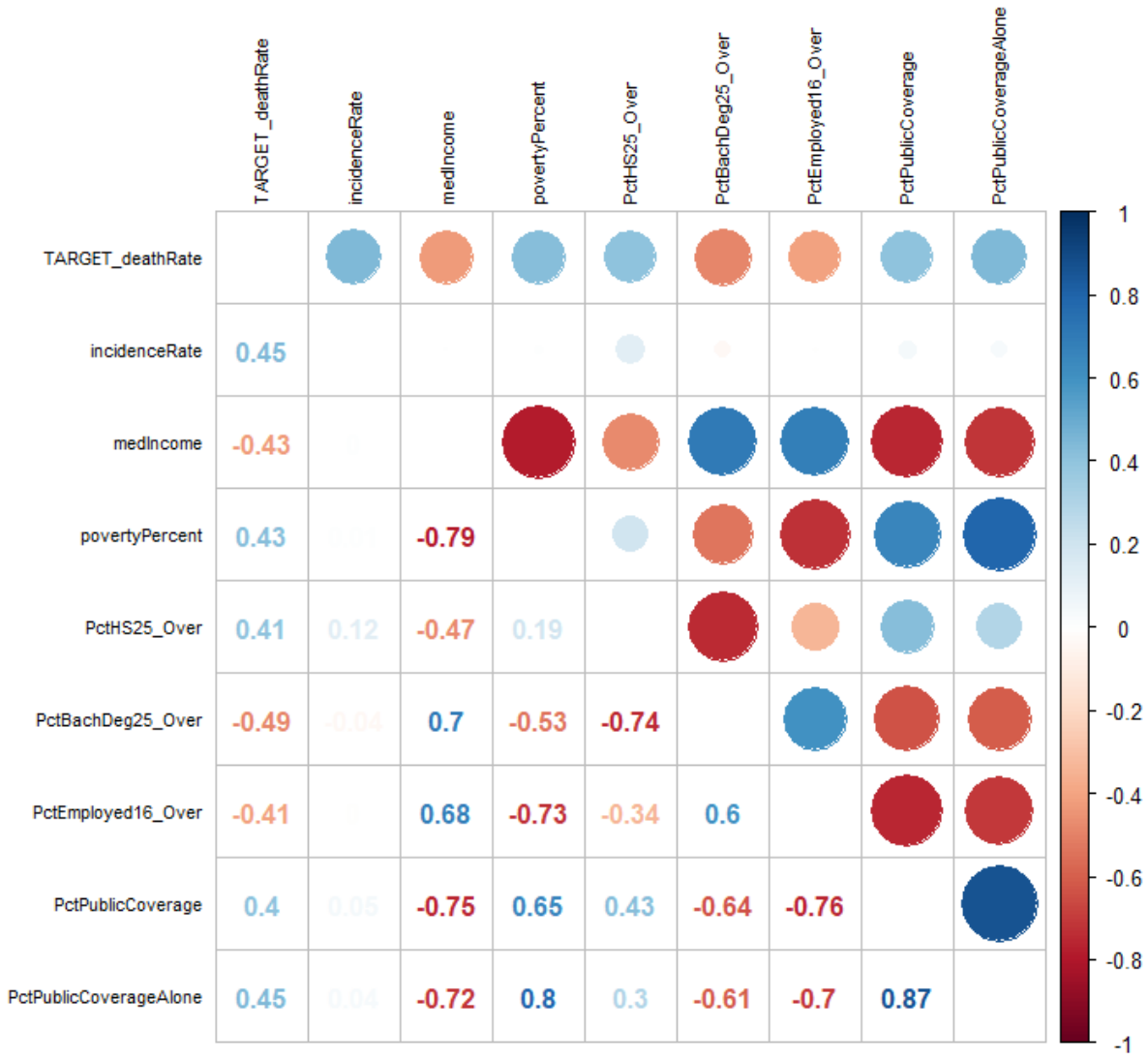


Διάγραμμα 4.1: Ιστόγραμμα των τιμών της εξαρτημένης μεταβλητής TARGET_deathRate, που εκφράζει το μέσο κατά κεφαλήν (ανά 100,000 άτομα) αριθμό θανάτων, εξαιτίας της νόσου του καρκίνου για τα έτη 2010-2016, στις κομητείες της Αμερικής.

```
> corHigh=names(which(sapply(cor_abs, function(x) x>0.4)))
> corrplot.mixed(correlation[corHigh,corHigh],tl.pos ="lt",
                 tl.col="black",tl.cex=0.7)
```

Από το Διάγραμμα 4.2 παρατηρούμε ότι την υψηλότερη γραμμική συσχέτιση με την εξαρτημένη μεταβλητή την έχει η μεταβλητή PctBachDeg25_Over με τιμή -0.49. Το Διάγραμμα διασποράς 4.3 παρουσιάζει αυτήν την γραμμική εξάρτηση. Αν και ο συντελεστής συσχέτισης δεν είναι γενικά πολύ υψηλός, παρατηρούμε ότι αύξηση του ποσοστού των κατοίκων κομητείας ηλικίας 25 ετών και άνω με πτυχίο Bachelor, οδηγεί σε σχετική μείωση του αριθμού των θανάτων (ανά 100,000) που οφείλονται στο καρκίνο.

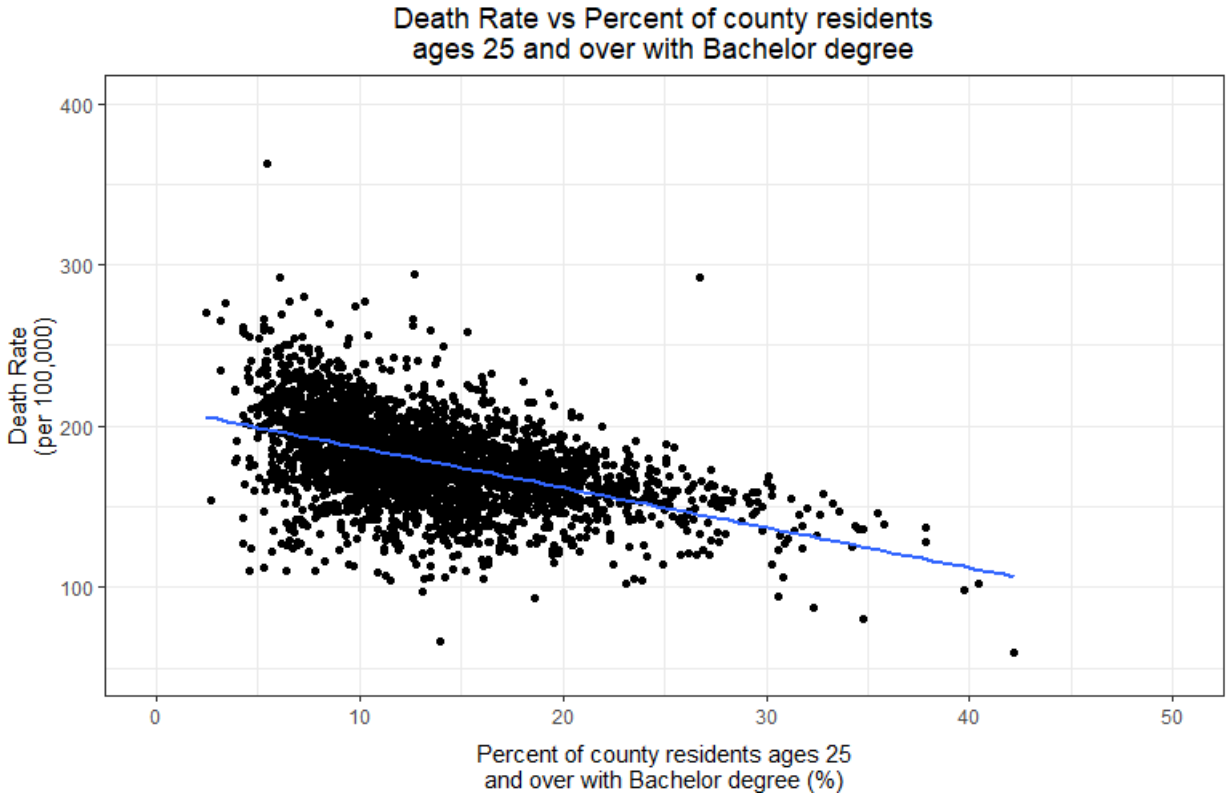
```
> ggplot(data,aes(x=PctBachDeg25_Over,y=TARGET_deathRate))+
   geom_point()+geom_smooth(method = "lm",se=F)+
   labs(title="Death Rate vs Percent of county residents\nages 25
        and over with Bachelor degree",
   x="Percent of county residents ages 25\nand over with Bachelor
      degree (%)",
   y="Death Rate\n(per 100,000)")+
   xlim(0,50)+ylim(50,400)+
   theme(plot.title=element_text(hjust=0.5),
```



Διάγραμμα 4.2: Διάγραμμα συσχετίσεων για τις μεταβλητές των οποίων η απόλυτη τιμή του συντελεστή συσχέτισης με τη μεταβλητή απόκρισης είναι άνω του 0.4.

```
        axis.title.x=element_text(vjust=-1.3))
```

Επίσης από το Διάγραμμα 4.2 παρατηρούμε ότι υπάρχουν και ορισμένες υψηλές συσχετίσεις μεταξύ ορισμένων ανεξάρτητων μεταβλητών. Για παράδειγμα ο συντελεστής συσχέτισης μεταξύ των μεταβλητών povertyPercent (ποσοστό πληθυσμού στη φτώχεια) και medIncome (διάμεσος του εισοδήματος των κατοίκων) είναι -0.79, πράγμα που σημαίνει ότι οι δύο μεταβλητές σχετίζονται σε μεγάλο βαθμό. Το ίδιο ισχύει και για τις μεταβλητές PctPublicCoverageAlone και PctPublicCoverage (συντελεστής συσχέτισης 0.87). Τέτοιες τιμές υποδηλώνουν πρόβλημα πολυσυγγραμμικότητας το οποίο θα πρέπει να ελέγξουμε, προτού προχωρήσουμε στη μοντελοποίηση. Στη συνέχεια κατασκευάζουμε το Διάγραμμα 4.4. Στο Διάγραμμα αυτό παρουσιάζεται ο μέσος αριθμός των θανάτων που οφείλονται στη νόσο του καρκίνου, ανάλογα με τις πολιτείες της Αμερικής για τα έτη 2010-2016. Δηλαδή για κάθε μία πολιτεία των Η.Π.Α κατασκευάζουμε ράβδους με

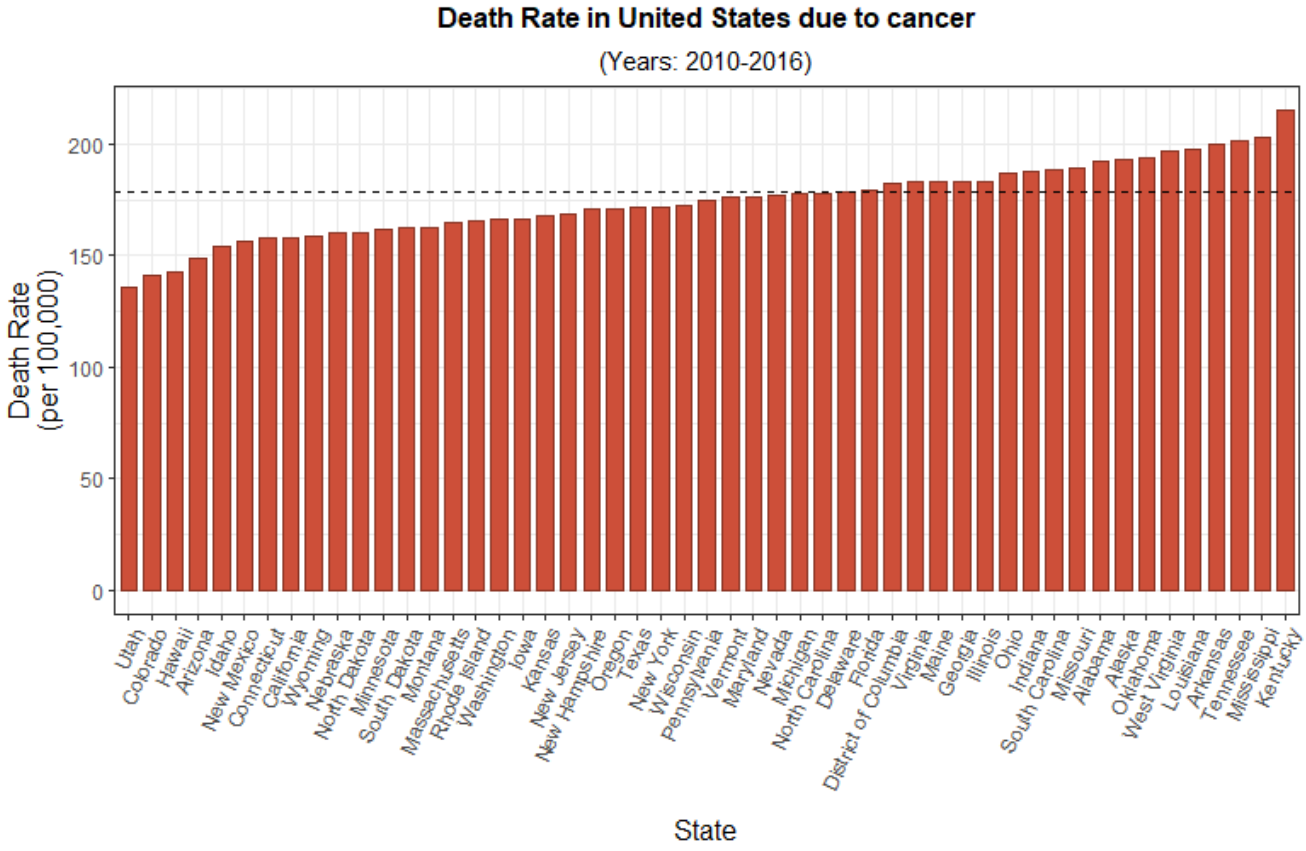


Διάγραμμα 4.3: Διάγραμμα διασποράς μεταξύ του ποσοστού θνησιμότητας (ανά 100,000 άτομα) και του ποσοστού των κατοίκων κομητείας ηλικίας 25 ετών και άνω οι οποίοι είναι κάτοχοι πτυχίου Bachelor.

ύψος ίσο με την μέση τιμή της μεταβλητής TARGET_deathRate σε κάθε πολιτεία (State). Χρησιμοποιώντας το όρισμα reorder διατάσσουμε τις ράβδους σε αύξουσα σειρά. Η οριζόντια (dotted) γραμμή αντιστοιχεί στη συνολική μέση τιμή της μεταβλητής απόκρισης που είναι ίση με 178.6.

```
> ggplot(data,aes(x=reorder(State,TARGET_deathRate,FUN=mean),
         y=TARGET_deathRate))+
  geom_bar(stat="summary",fill="tomato3",col="tomato4",
           width=0.7)+
  labs(title="Death Rate in United States due to cancer",
  subtitle="(Years: 2010-2016)",x="State",
  y="Death Rate\n(per 100,000)")+
  theme(plot.title=element_text(hjust=0.5,face="bold",
        size=12,vjust=2),
        plot.subtitle=element_text(hjust=0.5),
        axis.text.x=element_text(angle=65,hjust=1),
        axis.title.x=element_text(size=12,vjust=-1),
        axis.title.y=element_text(size=12))+
  geom_hline(yintercept=178.6,lty="dashed",cex=0.7)
```

Από το Διάγραμμα 4.4 είναι φανερό ότι υπάρχει μια διαφοροποίηση του αριθμού των θανάτων από πολιτεία σε πολιτεία. Για παράδειγμα οι πολιτείες Utah, Colorado και Hawaii παρουσιάζουν κατά μέσο όρο τον μικρότερο αριθμό θανάτων, ενώ οι πολιτείες Tennessee, Mississippi και Kentucky



Διάγραμμα 4.4: Ραβδογράμματα του μέσου αριθμού των θανάτων (ανά 100,000 άτομα) οι οποίοι οφείλονται στη νόσο του καρκίνου, στις πολιτείες των Η.Π.Α (έτη 2010-2016). Η οριζόντια (dotted) γραμμή αντιστοιχεί στη συνολική μέση τιμή (178.6) του αριθμού των θανάτων που οφείλονται στη νόσο του καρκίνου.

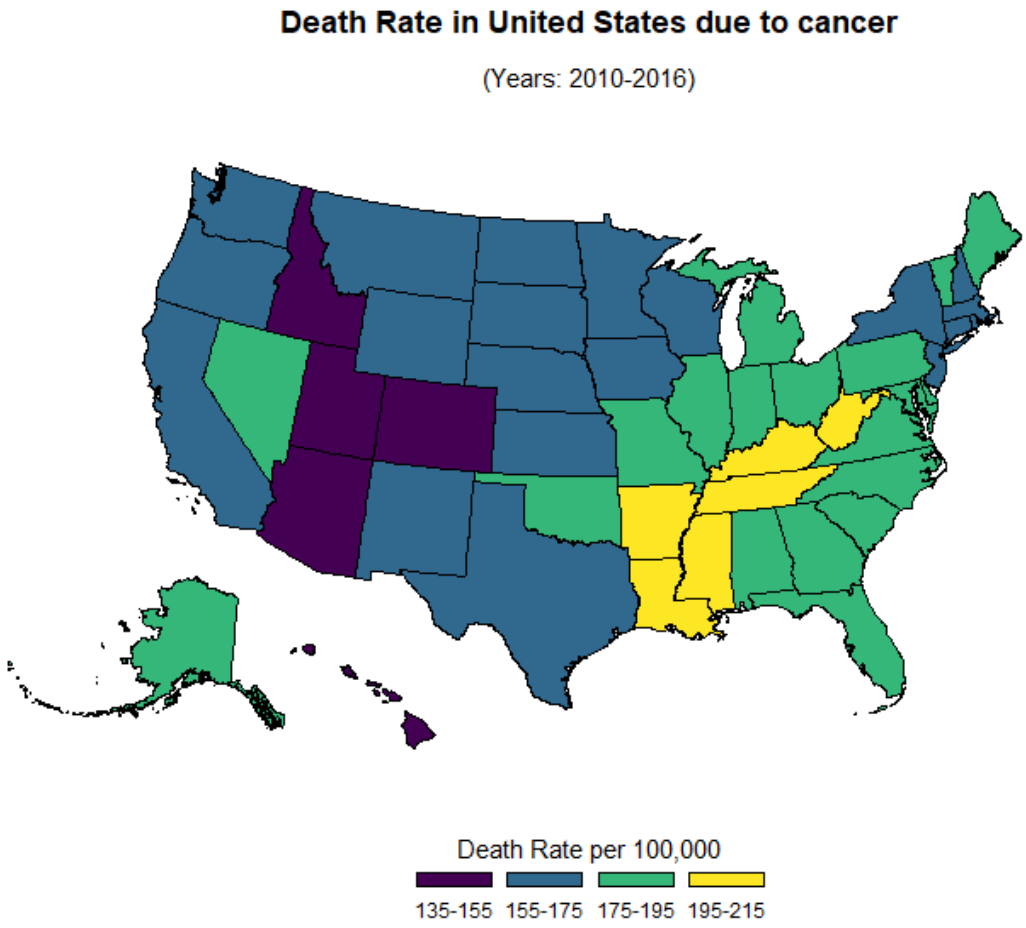
παρουσιάζουν κατά μέσο όρο το μέγιστο αριθμό θανάτων για τα έτη 2010-2016. Συνεπώς φαίνεται να υπάρχει εξάρτηση μεταξύ της μεταβλητής απόκρισης και του παράγοντα State. Επίσης με τον παρακάτω κώδικα θα κατασκευάσουμε τον χάρτη που φαίνεται στο Διάγραμμα 4.5. Αρχικά φορτώνουμε το πακέτο urbnmapr και το πλαίσιο δεδομένων states που ανήκει σε αυτό. Στο σύνολο states περιέχονται μεταβλητές όπως το γεωγραφικό μήκος και πλάτος (long, lat), το όνομα κάθε πολιτείας (state_name) και άλλες μεταβλητές που θα χρησιμοποιήσουμε για την κατασκευή του χάρτη. Συνολικά έχουμε 51 πολιτείες. Στη μεταβλητή state_name εκχωρούμε τα ονόματα όλων των πολιτειών σε αλφαβητική σειρά. Στο διάνυσμα death_rate εκχωρούμε τη μέση τιμή της εξαρτημένης μεταβλητής TARGET_deathRate για κάθε μία πολιτεία. Στη συνέχεια κατασκευάζουμε το πλαίσιο δεδομένων map_df και με την εντολή mutate προσθέτουμε μία επιπλέον στήλη με όνομα death_rate_bin η οποία θα περιέχει το διάστημα στο οποίο ανήκει η μέση τιμή που υπολογίσαμε προηγουμένως. Για παράδειγμα η πρώτη γραμμή του map_df θα περιέχει τη μέση τιμή του αριθμού των θανάτων (ανά 100,000) για την πολιτεία Alabama με τιμή 192.72, καθώς και το αντίστοιχο διάστημα 175-195. Τα διαστήματα κατασκευάζονται με την εντολή cut μέσω του ορίσματος break. Επιπλέον με την εντολή left_join ενώνουμε τα πλαίσια δεδομένων map_df και states με βάση τη μεταβλητή state_name. Το νέο πλαίσιο δεδομένων spatial_data χρησιμοποιείται για την κατασκευή του χάρτη.

```
> #map
```

```
> library(urbnmapr)
> data(states)
> state_name=sort(unique(data$State))
> death_rate=rep(0,51)
> for(i in 1:51){
   death_rate[i]=mean(data$TARGET_deathRate[data$State==
                                          state_name[i]])
> }
> map_df=data.frame(death_rate,state_name)
> map_df=map_df %>% mutate(death_rate_bin=cut(map_df$death_rate,
           breaks=c(135,155,175,195,215),
           labels=c("135-155","155-175","175-195","195-215")))

> map_df[1,]
  death_rate state_name death_rate_bin
1   192.7286    Alabama        175-195

> spatial_data=left_join(map_df,states,by="state_name")
> ggplot(spatial_data,aes(x=long,y=lat,group=group,
                          fill=death_rate_bin))+
  geom_polygon(color="black")+
  coord_map(projection="albers",lat0=39,lat0=45)+theme_void()+
  labs(fill="Death Rate")+
  ggtitle("Death Rate in United States due to cancer",
          subtitle="(Years: 2010-2016)")+
  theme(plot.title=element_text(hjust=0.5,vjust =5,face="bold"),
        plot.subtitle=element_text(hjust=0.5,vjust=3.5),
        legend.position="bottom")+
  scale_fill_viridis_d(name="Death Rate per 100,000",
           guide=guide_legend(direction="horizontal",
                 title.position="top",
                 title.hjust=0.5,
                 label.hjust=0.5,
                 label.position="bottom",
                 keyheight=0.5))
```

Παρατηρούμε ότι για τις πολιτείες με χρώμα μωβ ο μέσος αριθμός των θανάτων που οφείλονται στη νόσο του καρκίνου (ανά 100,000) άτομα κυμαίνεται στο διάστημα από 135 έως 155 θανάτους. Οι πολιτείες αυτές είναι οι Utah, Colorado, Hawaii, Arizona και Idaho, οι οποίες παρουσιάζουν κατά μέσο όρο το μικρότερο ποσοστό θνησιμότητας για τα έτη 2010-2016. Το υψηλότερο ποσοστό θνησιμότητας παρουσιάζεται σε πολιτείες όπως Kentucky, Mississippi, Tennessee κ.ά. (κίτρινο χρώμα). Το ποσοστό αυτό κυμαίνεται από 195 έως 215 θανάτους ανά 100,000 άτομα.

Στη συνέχεια θα εξετάσουμε τη σχέση μεταξύ του ποσοστού φτώχειας, του εισοδήματος και του αριθμού των θανάτων. Με τον παρακάτω κώδικα κατασκευάζουμε αρχικά ένα νέο πλαίσιο δεδομένων με όνομα new_data. Με χρήση της εντολής mutate δημιουργούμε δύο νέες στήλες με ονόματα binnedPov και inc_thous, όπου η πρώτη θα αποτελεί μια ομαδοποίηση του ποσοστού φτώχειας και η δεύτερη θα περιέχει το εισόδημα σε χιλιάδες δολλάρια. Πάλι με την εντολή mutate



Διάγραμμα 4.5: Στο χάρτη απεικονίζεται ο μέσος αριθμός των θανάτων (ανά 100,000 άτομα) που οφείλονται στη νόσο του καρκίνου, σε κάθε πολιτεία της Αμερικής για τα έτη 2010-2016.

δημιουργούμε τη νέα στήλη με όνομα binnedInc που αποτελεί μια ομαδοποίηση του εισοδήματος. Τα επιθυμητά διαστήματα κατασκευάζονται με χρήση της εντολής cut. Χρησιμοποιούμε την εντολή grid.arrange για να τοποθετήσουμε μαζί τα δύο διαγράμματα που προκύπτουν.

```
> new_data=data %>% mutate(binnedPov=cut(data$povertyPercent,
          breaks=c(0,10,20,30,40,50),right=FALSE),
          inc_thous=data$medIncome/1000)
> new_data=new_data %>% mutate(binnedInc=cut(new_data$inc_thous,
           breaks=c(20,40,60,80,100,130),
           dig.lab=10,right=FALSE))

> g1=ggplot(new_data,aes(x=binnedPov,y=TARGET_deathRate,
           fill=binnedPov))+
  geom_bar(stat="summary",width=0.5,show.legend=FALSE)+
```

```
  labs(title ="Death Rate vs Poverty in US",
       x="Percent of populace in poverty (%)",
       y="Death Rate\n(per 100,000)")+
  theme(axis.title.x=element_text(vjust=-1))+
  scale_fill_brewer(palette="Reds")

> g2=ggplot(new_data,aes(x=binnedInc,y=TARGET_deathRate,
            fill=binnedInc))+
  geom_bar(stat="summary",width=0.5,show.legend=FALSE)+
  labs(title="Death Rate vs Income in US",
       x="Median Income (in thousands $)",y="")+
  theme(axis.title.x=element_text(vjust=-1))+
  scale_fill_brewer(palette="Greens")

> grid.arrange(g1,g2,ncol=2)
```

Στο Διάγραμμα 4.6 παρατηρούμε (αριστερά) ότι όσο το ποσοστό του πληθυσμού που βρίσκεται στη φτώχεια αυξάνει, τόσο κατά μέσο όρο αυξάνει και ο αριθμός των θανάτων που οφείλονται στη νόσο του καρκίνου στις κομητείες των Η.Π.Α. Αντίστοιχα, στο δεξί Διάγραμμα παρατηρούμε ότι όσο η διάμεσος του εισοδήματος αυξάνει, τόσο ο αριθμός των θανάτων φθίνει κατά μέσο όρο. Το αποτέλεσμα αυτό φαίνεται λογικό, διότι αναμένουμε γενικά σε πληθυσμούς όπου υπάρχει οικονομική ευημερία, να υπάρχει ανάπτυξη και σε τομείς όπως η υγεία και η περίθαλψη.

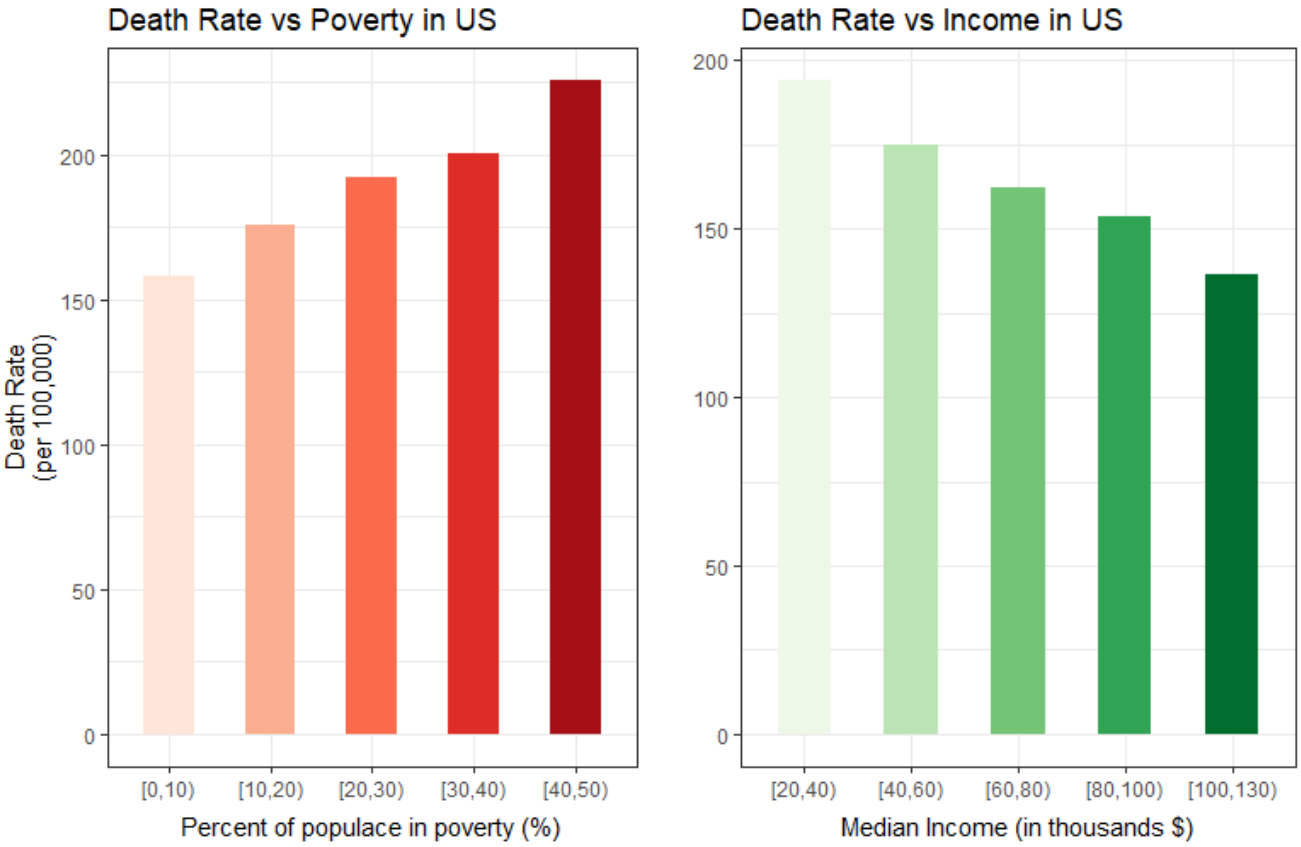
4.3.1 Πολυσυγγραμμικότητα

Όπως είδαμε προηγουμένως, ορισμένες από τις επεξηγηματικές μεταβλητές έχουν υψηλό βαθμό συσχέτισης μεταξύ τους. Προκειμένου λοιπόν να ελέγξουμε αν υπάρχει πολυσυγγραμμικότητα στα δεδομένα μας, θα χρησιμοποιήσουμε το διαγνωστικό ελέγχου VIF (Variance In ation Factor). Για κάθε μία επεξηγηματική μεταβλητή στο μοντέλο μας, η ποσότητα αυτή υπολογίζεται ως εξής:

$$\mathsf{VIF}_j = \frac{1}{1 \quad R_j^2},$$

όπου R_j^2 είναι ο συντελεστής προσδιορισμού που προκύπτει αν προσαρμόσουμε το μοντέλο με μεταβλητή απόκρισης την X_j και επεξηγηματικές μεταβλητές τις υπόλοιπες $X_k, k \neq j$. Συνήθως όταν έχουμε $\mathsf{VIF}_j > 10$, $R_j^2 > 0.90$ τότε υπάρχει πιθανό πρόβλημα πολυσυγγραμμικότητας. Στην R για τον υπολογισμό του, θα χρησιμοποιήσουμε την εντολή vif από το πακέτο rms. Με τον παρακάτω κώδικα αρχικά προσαρμόζουμε το μοντέλο με μεταβλητή απόκρισης την TARGET_deathRate με χρήση της εντολής lm. Ύστερα υπολογίζουμε τα vif (με στρογγυλοποίηση στο πρώτο δεκαδικό) και τα ταξινομούμε σε φθίνουσα σειρά (εδώ παρουσιάζουμε μόνο τις τιμές των VIF που είναι μεγαλύτερες από 10 και άρα υποδεικνύουν πρόβλημα πολυσυγγραμμικότητας).

```
> #multicollinearity
> mfull=lm(TARGET_deathRate .,data)
> VIF=round(vif(mfull),1)
> sort(VIF,decreasing = TRUE)
    MedianAge              MedianAgeMale       MedianAgeFemale
    267.3                  90.6                69.4
```



Διάγραμμα 4.6: (Αριστερά) Ο αριθμός των θανάτων (ανά 100,000) που οφείλονται στη νόσο του καρκίνου σε σχέση με το ποσοστό του πληθυσμού που βρίσκεται στη φτώχεια για τις κομητείες των Η.Π.Α. (Δεξιά) Ο αριθμός των θανάτων σε σχέση με τη διάμεσο του εισοδήματος.

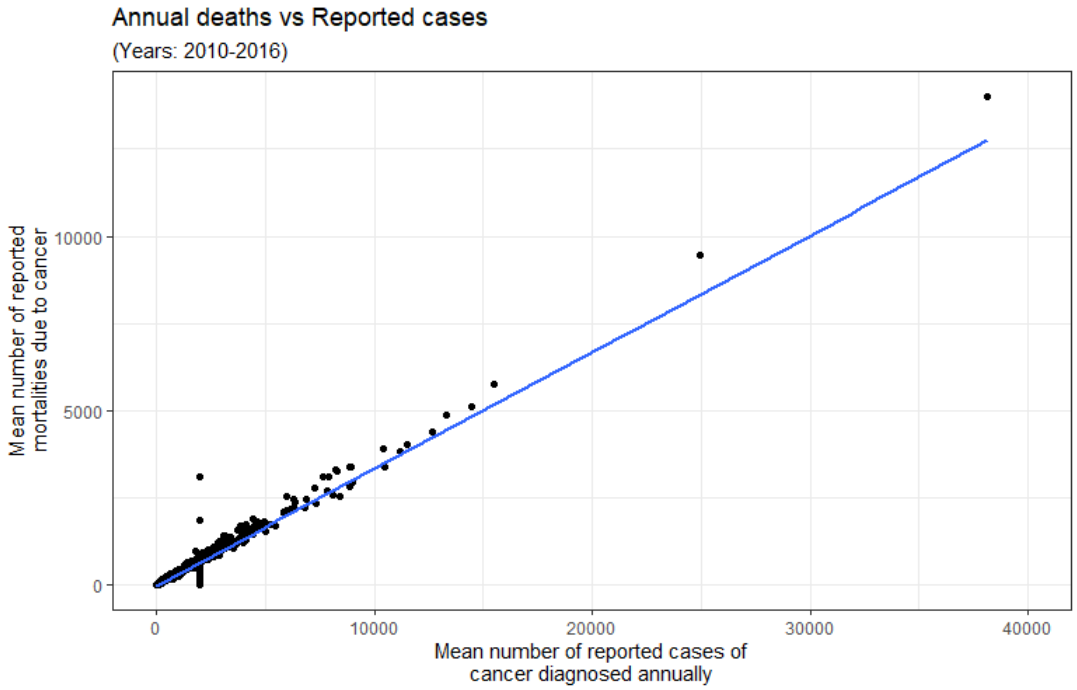
```
    avgDeathsPerYear          avgAnnCount          PctPublicCoverage
    61.1                      46.7                 34.0
    popEst2015      PctPublicCoverageAlone         PctPrivateCoverage
    30.2            29.9                           25.9
    PercentMarried            PctWhite             povertyPercent
    13.1                      10.9                 10.2
```

Από τα παραπάνω αποτελέσματα είναι ξεκάθαρο ότι υπάρχει πρόβλημα πολυσυγγραμμικότητας στα δεδομένα μας. Η μεταβλητή MedianAge (διάμεσος ηλικίας) έχει την υψηλότερη τιμή VIF 267.3, κάτι που είναι αναμενόμενο, αφού η μεταβλητή αυτή σχετίζεται άμεσα με τις μεταβλητές MedianAgeMale (διάμεσος ηλικίας αντρών) και MedianAgeFemale (διάμεσος ηλικίας γυναικών). Επίσης είναι λογικό να υπάρχει συσχέτιση μεταξύ των μεταβλητών avgDeathsPerYear (μέσος αριθμός αναφερόμενων θανάτων λόγω του καρκίνου) και avgAnnCount (μέσος αριθμός αναφερόμενων περιστατικών καρκίνου που διαγιγνώσκονται ετησίως). Όμοια συμπεράσματα προκύπτουν και για τις υπόλοιπες μεταβλητές με υψηλά VIF. Με τον παρακάτω κώδικα κατασκευάζουμε τα διαγράμματα που υποδεικνύουν τις συσχετίσεις μεταξύ ορισμένων επεξηγηματικών μεταβλητών.

```
> ggplot(data,aes(x=avgAnnCount,y=avgDeathsPerYear))+
  geom_point()+
  geom_smooth(method="lm",se=F)+xlim(0,40000)+
```

```
  labs(title="Annual deaths vs Reported cases",
  subtitle="(Years: 2010-2016)",
  x="Mean number of reported cases of\ncancer diagnosed annually",
  y="Mean number of reported\nmortalities due to cancer")
```

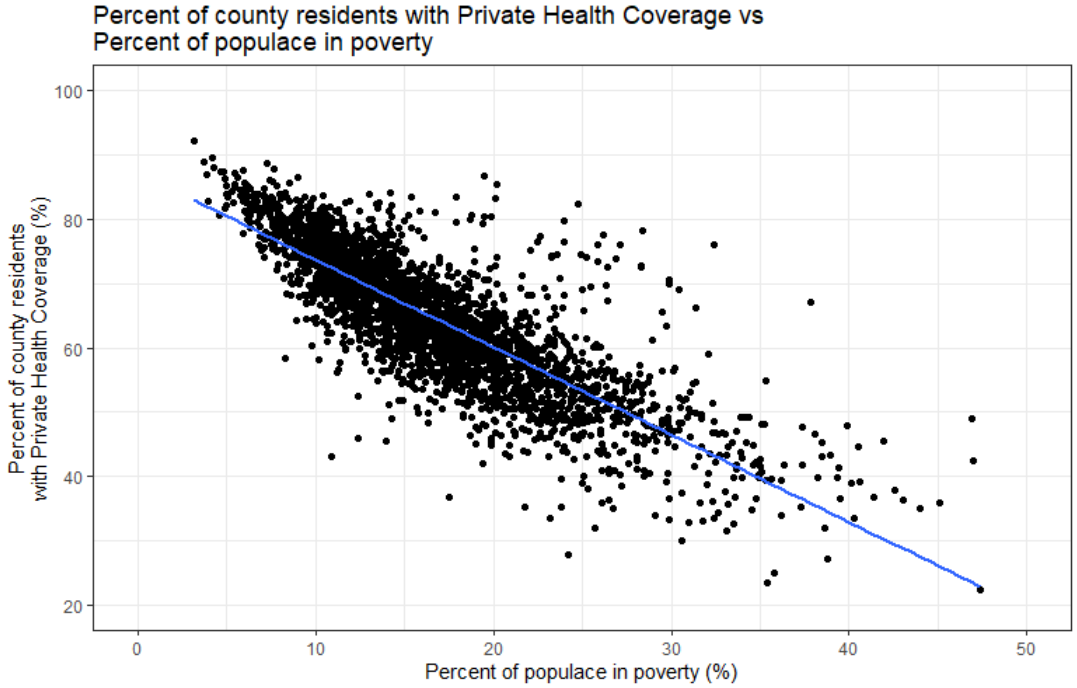
Στο Διάγραμμα 4.7 φαίνεται η εξάρτηση μεταξύ των μεταβλητών avgDeathsPerYear και avgAnnCount. Είναι λογικό να περιμένουμε αύξηση στο μέσο αριθμό των αναφερόμενων θανάτων που οφείλονται στη νόσο του καρκίνου, όταν παρατηρείται αύξηση στο μέσο αριθμό των αναφερόμενων περιστατικών καρκίνου ετησίως. Στο Διάγραμμα 4.8 παρουσιάζεται η σχέση μεταξύ του ποσοστού



Διάγραμμα 4.7: Η σχέση μεταξύ του μέσου αριθμού αναφερόμενων θανάτων που οφείλονται στη νόσο του καρκίνου και του μέσου αριθμού αναφερόμενων περιστατικών καρκίνου ετησίως, για τις κομητείες των Η.Π.Α (έτη: 2010-2016).

του πληθυσμού που βρίσκεται στη φτώχεια και του ποσοστού των κατοίκων με ιδιωτική κάλυψη υγείας στις κομητείες της Αμερικής για το έτος 2013. Σε κομητείες όπου το ποσοστό των κατοίκων που βρίσκονται στη φτώχεια είναι αυξημένο, παρατηρείται μείωση του ποσοστού των κατοίκων με ιδιωτική κάλυψη υγείας. Το αποτέλεσμα αυτό είναι λογικό αφού όταν υπάρχει υψηλό ποσοστό φτώχειας αναμένουμε οι κάτοικοι να μην έχουν την οικονομική δυνατότητα για αγορά ιδιωτικής ασφάλειας υγείας.

```
> ggplot(data,aes(x=povertyPercent,y=PctPrivateCoverage))+
  geom_point()+geom_smooth(method="lm",se=F)+
  xlim(0,50)+ylim(20,100)+
  labs(title="Percent of county residents
       with Private Health Coverage vs\nPercent of populace
       in poverty", x="Percent of populace in poverty (%)",
       y="Percent of county residents\nwith Private Health
          Coverage (%)")
```



Διάγραμμα 4.8: Η σχέση μεταξύ του ποσοστού των κατοίκων που έχουν ιδιωτική κάλυψη υγείας στις κομητείες της Αμερικής και του ποσοστού του πληθυσμού που βρίσκεται στη φτώχεια για το έτος 2013.

Τέλος, επιπλέον υψηλές συσχετίσεις στα δεδομένα μας υπάρχουν και μεταξύ επεξηγηματικών μεταβλητών όπως PctMarriedHouseholds (ποσοστό παντρεμένων νοικοκυριών) και PercentMarried (ποσοστό κατοίκων κομητείας που είναι παντρεμένοι) (θετική συσχέτιση), PctPublicCoverageAlone (ποσοστό κατοίκων κομητείας στους οποίους παρέχεται κάλυψη υγείας από την κυβέρνηση αλλά συνεισφέρουν και οι ίδιοι) και PctPrivateCoverage (ποσοστό κατοίκων κομητείας με ιδιωτική κάλυψη υγείας) (αρνητική συσχέτιση), PctBlack (ποσοστό κατοίκων της κομητείας που αναγνωρίζονται ως Αφροαμερικάνοι) και PctWhite (ποσοστό κατοίκων της κομητείας που αναγνωρίζονται ως «Λευκοί») (αρνητική συσχέτιση). Τις συσχετίσεις αυτές μπορούμε να τις δούμε στον πίνακα correlation που έχουμε κατασκευάσει.
Μια συνηθισμένη τεχνική για να αντιμετωπίσουμε το πρόβλημα της πολυσυγγραμμικότητας προτού κατασκευάσουμε το μοντέλο μας, είναι να αφαιρέσουμε εκείνες τις επεξηγηματικές μεταβλητές που παρουσιάζουν υψηλές τιμές των VIF και άρα εξηγούνται σε μεγάλο βαθμό από τις υπόλοιπες. Εναλλακτικά, θα χρησιμοποιήσουμε ορισμένες από τις τεχνικές συρρίκνωσης που έχουμε αναλύσει σε προηγούμενα κεφάλαια, ώστε να κατασκευάσουμε ένα μοντέλο που θα προβλέπει το μέσο αριθμό θανάτων (ανά 100,000) που οφείλονται στη νόσο του καρκίνου, με βάση ορισμένες από τις επεξηγηματικές μεταβλητές που διαθέτουμε. Ευελπιστούμε το τελικό μας μοντέλο να είναι όσο το δυνατόν φειδωλό και να έχει υψηλή προβλεπτική ακρίβεια.

4.4 Μοντελοποίηση

Όπως αναφέραμε, για την προσαρμογή του μοντέλου θα χρησιμοποιήσουμε ορισμένες τεχνικές συρρίκνωσης που έχουμε ήδη αναπτύξει σε προηγούμενα κεφάλαια. Αρχικά χρησιμοποιούμε τον

παρακάτω κώδικα για να κατασκευάσουμε τον πίνακα σχεδιασμού x αφαιρώντας την πρώτη στήλη που περιέχει μονάδες. Με την εντολή model.matrix μετατρέπουμε αυτόματα κάθε κατηγορική μεταβλητή σε ψευδομεταβλητές (dummy variables). Στα δεδομένα μας η μόνη κατηγορική μεταβλητή είναι η μεταβλητή State (51 κατηγορίες) οπότε χρησιμοποιούνται 50 εικονικές μεταβλητές για την κωδικοποίησή της. Επομένως ο πίνακας σχεδιασμού x θα έχει 3017 γραμμές και 80 στήλες.

```
> x=model.matrix(TARGET_deathRate .,data)[,-1]
> y=data$TARGET_deathRate
> dim(x)
[1] 3017   80
```

Στη συνέχεια θα χωρίσουμε τα δεδομένα μας σε train και test set. Θα χρησιμοποιήσουμε το 70% των παρατηρήσεών μας για να προσαρμόσουμε το μοντέλο και το υπόλοιπο 30% για να εξετάσουμε την προβλεπτική του ικανότητα και ακρίβεια.

```
> set.seed(100)
> train=sample(1:nrow(data),0.7*nrow(data))
> test=-train
```

Αρχικά προσαρμόζουμε το μοντέλο μας με τη μέθοδο ελαχίστων τετραγώνων για τα δεδομένα που ανήκουν στο σύνολο train και με την εντολή predict υπολογίζουμε τις εκτιμήσεις για την μεταβλητή απόκρισης με βάση το σύνολο test. Στη μεταβλητή MSE_linear αποθηκεύουμε το μέσο τετραγωνικό σφάλμα για τα δεδομένα που ανήκουν στο σύνολο test.

```
> #OLS
> linear.fit=lm(TARGET_deathRate .,data,subset=train)
> linear.pred=predict(linear.fit,newdata=data[test,])
> MSE_linear=mean((y[test]-linear.pred)^2)
```

Θα χρησιμοποιήσουμε τώρα την τεχνική συρρίκνωσης Ridge προκειμένου να προσαρμόσουμε το μοντέλο μας. Αρχικά χρησιμοποιούμε την εντολή glmnet με όρισμα alpha=0 και προσαρμόζουμε το μοντέλο για τα δεδομένα που ανήκουν στο σύνολο train. Ύστερα με την εντολή cv.glmnet εκτελούμε 10-fold Cross Validation με σκοπό να υπολογίσουμε την τιμή της παραμέτρου ποινής λ, που ελαχιστοποιεί το σφάλμα cross-validation (λ_{min}) καθώς επίσης και την τιμή της παραμέτρου που αντιστοιχεί στο σφάλμα 1_{se} ($\lambda_{1\text{se}}$). Οι τιμές που προκύπτουν είναι 1.37 και 12.82 αντίστοιχα. Επίσης με την εντολή sapply βρίσκουμε τους μη μηδενικούς συντελεστές (όρισμα function(x) x!=0) που προκύπτουν από τη μέθοδο Ridge (εκτός του σταθερού όρου) και ύστερα υπολογίζουμε το πλήθος τους. Όπως ήταν αναμενόμενο και για τις δύο τιμές της παραμέτρου ποινής λ το τελικό μοντέλο περιλαμβάνει και τις 80 επεξηγηματικές μεταβλητές. Αυτό συμβαίνει επειδή η μέθοδος Ridge δεν κάνει επιλογή μεταβλητών, παρ' όλα αυτά συρρικνώνει τους συντελεστές των μη σημαντικών μεταβλητών ώστε να επιτευχθεί μείωση της διασποράς των συντελεστών. Με την εντολή predict και με όρισμα newx=x[test,] υπολογίζουμε τις προβλέψεις για τη μεταβλητή απόκρισης χρησιμοποιώντας το σύνολο test, και για τις δύο τιμές $\lambda_{\text{min}}, \lambda_{1\text{se}}$ της παραμέτρου ποινής. Τέλος, αποθηκεύουμε τα μέσα τετραγωνικά σφάλματα για το test set στις μεταβλητές MSE_ridge.min και MSE_ridge.1se.

```
> #Ridge regression
> ridge.fit=glmnet(x[train,],y[train],alpha=0)
> ridge.cv=cv.glmnet(x[train,],y[train],alpha=0)

> lam_ridge.min=ridge.cv$lambda.min
```

```
> lam_ridge.1se=ridge.cv$lambda.1se
> print(c(lam_ridge.min,lam_ridge.1se))
[1]  1.374483 12.818476
> length(which(sapply(coef(ridge.cv,s=lam_ridge.min)[-1],
         function(x) x!=0)))
[1] 80
> length(which(sapply(coef(ridge.cv,s=lam_ridge.1se)[-1],
         function(x) x!=0)))
[1] 80

> ridge.pred.min=predict(ridge.fit,s=lam_ridge.min,newx=x[test,])
> ridge.pred.1se=predict(ridge.fit,s=lam_ridge.1se,newx=x[test,])
> MSE_ridge.min=mean((y[test]-ridge.pred.min)^2)
> MSE_ridge.1se=mean((y[test]-ridge.pred.1se)^2)
```

Στη συνέχεια θα εφαρμόσουμε τη μέθοδο Lasso, χρησιμοποιώντας πάλι τις ίδιες εντολές με πριν, (για τις εντολές glmnet και cv.glmnet το όρισμα alpha=1 χρησιμοποιείται by default, οπότε το παραλείπουμε). Για την τιμή της παραμέτρου ποινής λ_{min} = 0.04 το τελικό μοντέλο περιέχει 76 μεταβλητές, ενώ για την τιμή λ_{1se} = 0.77 περιέχει 41 μεταβλητές. Βλέπουμε λοιπόν ότι με τη μέθοδο Lasso καταλήγουμε σε ένα πιο φειδωλό μοντέλο και για τις δύο τιμές της παραμέτρου λ. Στη συνέχεια υπολογίζουμε πάλι τα αντίστοιχα μέσα τετραγωνικά σφάλματα για το σύνολο test και τα αποθηκεύουμε στις μεταβλητές MSE_lasso.min και MSE_lasso.1se.

```
> #Lasso regression
> lasso.fit=glmnet(x[train,],y[train])
> lasso.cv=cv.glmnet(x[train,],y[train])

> lam_lasso.min=lasso.cv$lambda.min
> lam_lasso.1se=lasso.cv$lambda.1se
> print(c(lam_lasso.min,lam_lasso.1se))
[1] 0.04296244 0.76844743
> length(which(sapply(coef(lasso.cv,s=lam_lasso.min)[-1],
         function(x) x!=0)))
[1] 76
> length(which(sapply(coef(lasso.cv,s=lam_lasso.1se)[-1],
         function(x) x!=0)))
[1] 41

> lasso.pred.min=predict(lasso.fit,s=lam_lasso.min,newx=x[test,])
> lasso.pred.1se=predict(lasso.fit,s=lam_lasso.1se,newx=x[test,])
> MSE_lasso.min=mean((y[test]-lasso.pred.min)^2)
> MSE_lasso.1se=mean((y[test]-lasso.pred.1se)^2)
```

Η μέθοδος Elastic net συνδυάζει τις ποινές των μεθόδων Ridge και Lasso μέσω της παραμέτρου α *2* [0, 1]. Για την προσαρμογή του μοντέλου μας επιλέγουμε την τιμή α = 0.5. Χρησιμοποιώντας τον παρακάτω κώδικα καταλήγουμε πάλι σε μοντέλα που περιέχουν λιγότερες επεξηγηματικές μεταβλητές από τις αρχικά διαθέσιμες. Για την τιμή λ_{min} = 0.11 το μοντέλο μας περιέχει 73 μεταβλητές ενώ για την τιμή λ_{1se} = 1.40 περιέχει 44 μεταβλητές.

```
> #Elastic Net
> el_net.fit=glmnet(x[train,],y[train],alpha=0.5)
> el_net.cv=cv.glmnet(x[train,],y[train],alpha=0.5)

> lam_el_net.min=el_net.cv$lambda.min
> lam_el_net.1se=el_net.cv$lambda.1se
> print(c(lam_el_net.min,lam_el_net.1se))
[1] 0.1135876 1.4003614
> length(which(sapply(coef(el_net.cv,s=lam_el_net.min)[-1],
         function(x) x!=0)))
[1] 73
> length(which(sapply(coef(el_net.cv,s=lam_el_net.1se)[-1],
         function(x) x!=0)))
[1] 44

> el_net.pred.min=predict(el_net.fit,s=lam_el_net.min,
                          newx=x[test,])
> el_net.pred.1se=predict(lasso.fit,s=lam_el_net.1se,
                          newx=x[test,])
> MSE_el_net.min=mean((y[test]-el_net.pred.min)^2)
> MSE_el_net.1se=mean((y[test]-el_net.pred.1se)^2)
```

Τέλος, θα εφαρμόσουμε τη μέθοδο Adaptive Lasso για την προσαρμογή του μοντέλου μας. Ως αρχική εκτίμηση χρησιμοποιούμε τους συντελεστές που προκύπτουν από τη μέθοδο Lasso. Οπότε για την τιμή λ_{min} που ελαχιστοποιεί το σφάλμα cross-validation, υπολογίζουμε το αριθμητικό διάνυσμα των συντελεστών που προκύπτουν από τη μέθοδο Lasso και το αποθηκεύουμε στη μεταβλητή lasso.coef. Εφόσον δεν επιβάλουμε ποινή στο σταθερό όρο, τον αφήνουμε εκτός από το διάνυσμα των συντελεστών. Με το όρισμα penalty.factor στις εντολές glmnet και cv.glmnet, ποινικοποιούμε κάθε συντελεστή ξεχωριστά δίνοντας του βάρος $w_j = 1/|\hat{\beta}_{Lasso,j}|, j = 1, \ldots, p$. Εάν κάποιος συντελεστής προκύψει 0 από την μέθοδο Lasso τότε θα είναι μηδενικός και από την προσαρμογή του μοντέλου με τη μέθοδο Adaptive Lasso. Από τα παρακάτω αποτελέσματα παρατηρούμε ότι για την τιμή της παραμέτρου ποινής $\lambda_{\text{min}} = 38.76$ το τελικό μας μοντέλο περιέχει 65 μεταβλητές, ενώ για την τιμή $\lambda_{\text{1se}} = 1929.31$ περιέχει 44 μεταβλητές.

```
> #Adaptive Lasso
> lasso.coef=as.numeric(coef(lasso.cv,s=lam_lasso.min))[-1]
> ad_lasso.fit=glmnet(x[train,],y[train],
                      penalty.factor=1/abs(lasso.coef))
> ad_lasso.cv=cv.glmnet(x[train,],y[train],
                        penalty.factor=1/abs(lasso.coef))

> lam_ad_lasso.min=ad_lasso.cv$lambda.min
> lam_ad_lasso.1se=ad_lasso.cv$lambda.1se
> print(c(lam_ad_lasso.min,lam_ad_lasso.1se))
[1]   38.76429 1929.30796
> length(which(sapply(coef(ad_lasso.cv,s=lam_ad_lasso.min)[-1],
         function(x) x!=0)))
[1] 65
```

```
> length(which(sapply(coef(ad_lasso.cv,s=lam_ad_lasso.1se)[-1],
         function(x) x!=0)))
[1] 44

> ad_lasso.pred.min=predict(ad_lasso.fit,s=lam_ad_lasso.min,
                            newx=x[test,])
> ad_lasso.pred.1se=predict(ad_lasso.fit,s=lam_ad_lasso.1se,
                            newx=x[test,])
> MSE_ad_lasso.min=mean((y[test]-ad_lasso.pred.min)^2)
> MSE_ad_lasso.1se=mean((y[test]-ad_lasso.pred.1se)^2)
```

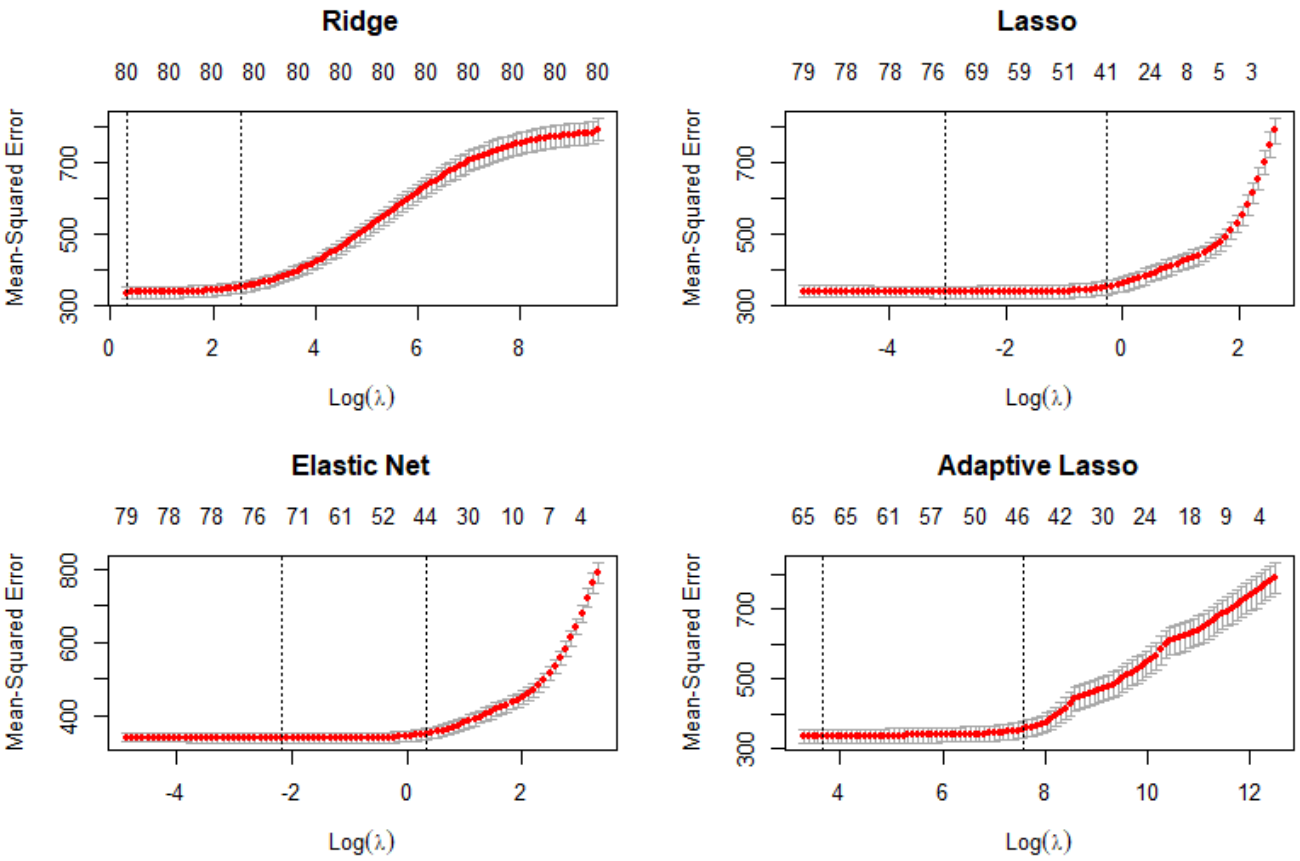
Με τον παρακάτω κώδικα κατασκευάζουμε το Διάγραμμα 4.9. Στο Διάγραμμα αυτό παρουσιάζονται οι καμπύλες του σφάλματος Cross Validation σε σχέση με το λογάριθμο της παραμέτρου ποινής λ, για κάθε μία από τις μεθόδους που χρησιμοποιήσαμε. Σε κάθε Διάγραμμα, στον άνω άξονα φαίνονται οι μεταβλητές που τελικά επιλέγονται στο μοντέλο για κάθε τιμή της παραμέτρου ποινής. Βλέπουμε ότι καθώς αυξάνεται η τιμή του λογαρίθμου της παραμέτρου ποινής λ (ισοδύναμα αυξάνεται η παράμετρος ποινής λ), όλο και λιγότερες επεξηγηματικές μεταβλητές περιέχονται στο μοντέλο μας (εκτός από τη μέθοδο Ridge).

```
> par(mfrow=c(2,2))
> plot(ridge.cv)
> title(main="Ridge",line =3)
> plot(lasso.cv)
> title(main="Lasso",line = 3)
> plot(el_net.cv)
> title(main="Elastic Net",line=3)
> plot(ad_lasso.cv)
> title(main="Adaptive Lasso",line=3)
```

4.5 Αξιολόγηση μοντέλου

Μέχρι στιγμής έχουμε προσαρμόσει το μοντέλο μας με 5 διαφορετικές τεχνικές. Αρχικά χρησιμοποιήσαμε τη μέθοδο ελαχίστων τετραγώνων και στη συνέχεια εφαρμόσαμε τις μεθόδους Ridge, Lasso, Elastic Net και Adaptive Lasso. Για κάθε μία από αυτές καταλήγουμε σε ένα διαφορετικό μοντέλο, το οποίο θα περιέχει ορισμένες από τις αρχικές διαθέσιμες επεξηγηματικές μεταβλητές ή και όλες. Επίσης κάθε ένα από τα μοντέλα μας θα έχει και ένα διαφορετικό μέσο τετραγωνικό σφάλμα πρόβλεψης για τις παρατηρήσεις που ανήκουν στο σύνολο test. Σκοπός μας είναι να επιλέξουμε τελικά το μοντέλο εκείνο, που δίνει το ελάχιστο μέσο τετραγωνικό σφάλμα για το test set, καθώς επίσης και το μοντέλο που είναι όσο το δυνατόν πιο φειδωλό. Από τη μέθοδο ελαχίστων τετραγώνων το σφάλμα που προκύπτει είναι 353.98 και όλες οι επεξηγηματικές μεταβλητές περιέχονται στο μοντέλο.

```
> MSE_linear
[1] 353.9769
> length(which(sapply(coef(linear.fit)[-1], function(x) x!=0)))
[1] 80
```



Διάγραμμα 4.9: Οι καμπύλες του σφάλματος Cross Validation για κάθε μία από τις τεχνικές συρρίκνωσης που χρησιμοποιήσαμε.

Στον Πίνακα 4.1 παρουσιάζουμε για κάθε μία μέθοδο προσαρμογής τα αντίστοιχα μέσα τετραγωνικά σφάλματα για το test set (στήλη Test MSE), καθώς και τον αριθμό των μεταβλητών που περιέχονται στο τελικό μοντέλο (στήλη Παράγοντες), για τις δύο τιμές της παραμέτρου ποινής λ_{min} και λ_{1se}. Παρατηρούμε ότι το ελάχιστο σφάλμα αντιστοιχεί στη μέθοδο Ridge με τιμή 346.11, αλλά όλες οι επεξηγηματικές μεταβλητές περιέχονται στο μοντέλο. Η μέθοδος Lasso δίνει σφάλμα 348.06 (για λ_{min}) με 76 μεταβλητές, δηλαδή μόλις 4 λιγότερες από το συνολικό αριθμό όλων των διαθέσιμων μεταβλητών. Η μέθοδος Elastic Net (για α = 0.5) δίνει μικρότερο σφάλμα από τη μέθοδο Lasso, με τιμή 347.30, καθώς επίσης και ένα μοντέλο με 73 επεξηγηματικές μεταβλητές. Τέλος, το σφάλμα που προκύπτει από τη μέθοδο Adaptive Lasso είναι 346.45 (για λ_{1se}) και περιέχει μόλις 44 επεξηγηματικές μεταβλητές. Άρα με βάση το διαχωρισμό που κάναμε στα δεδομένα μας σε train και test set, είναι λογικό να επιλέξουμε το μοντέλο που προκύπτει με χρήση της μεθόδου Adaptive Lasso για την τιμή λ_{1se}, εφόσον δίνει το δεύτερο πιο μικρό σφάλμα μετά τη μέθοδο Ridge και επίσης περιέχει μόνο 44 μεταβλητές, δηλαδή σχεδόν τις μισές απ΄ όσες είχαμε αρχικά στη διάθεσή μας. Παρ΄ όλα αυτά, θα πρέπει να εξετάσουμε αν το αποτέλεσμα αυτό για το test MSE, προέκυψε «τυχαία», με την έννοια ότι αν χωρίζαμε τα δεδομένα μας με άλλον τρόπο, τότε μπορεί να καταλήγαμε σε μία αρκετά διαφορετική τιμή για το σφάλμα κάθε μεθόδου. Προκειμένου να ελέγξουμε ποιο μοντέλο τελικά δίνει κατά μέσο όρο το ελάχιστο σφάλμα, θα επαναλάβουμε τη διαδικασία προσαρμογής 100 φορές, ώστε κάθε φορά να επιλέγουμε διαφορετικά στοιχεία για το train set και διαφορετικά στοιχεία για το test set. Η αναλογία train (70%) και

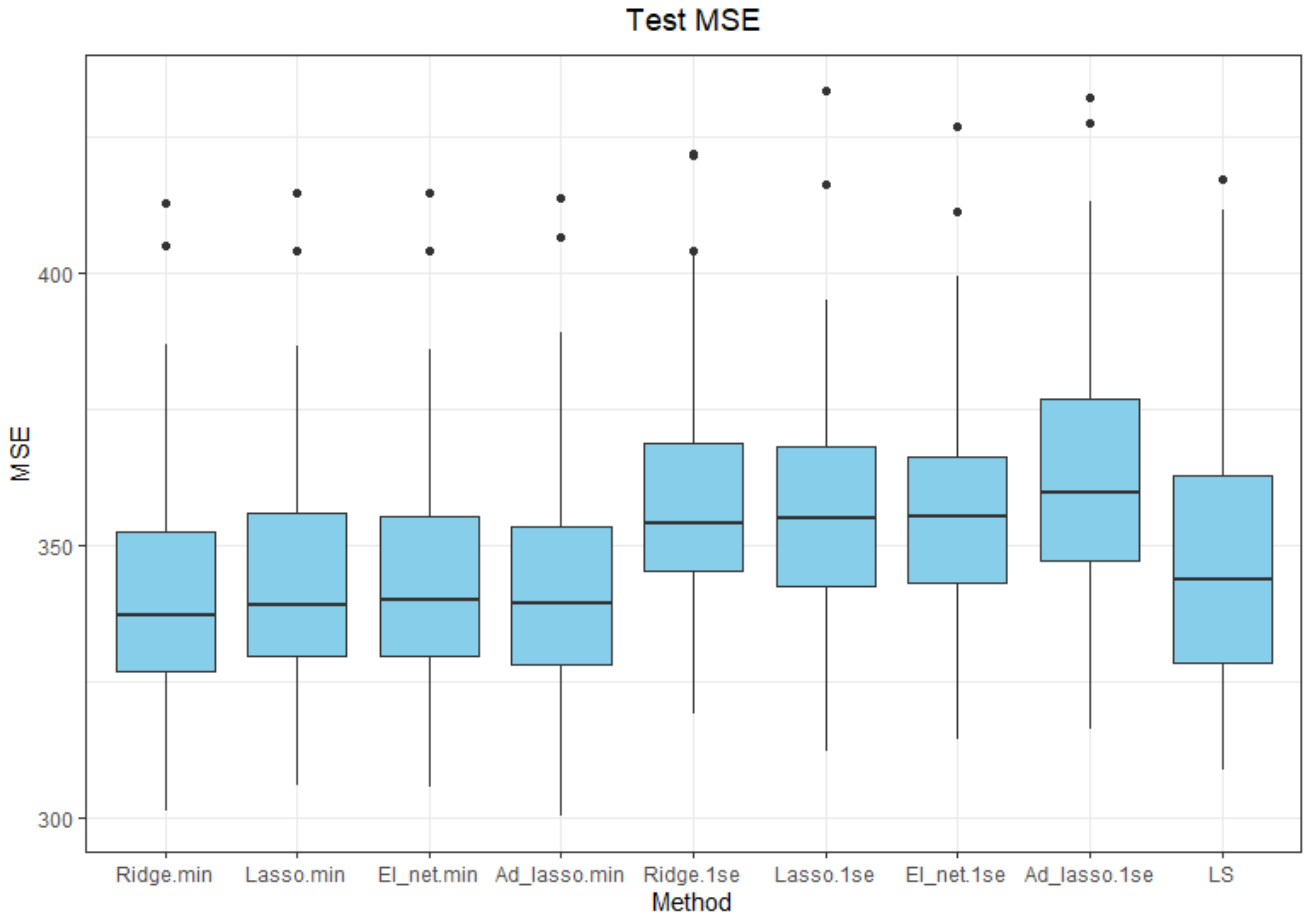
Μέθοδος	**Test MSE**		**Παράγοντες**	
LS	353.98		80	
	(λ_{min})	(λ_{1se})	(λ_{min})	(λ_{1se})
Ridge	346.11	348.66	80	80
Lasso	348.06	355.76	76	41
Elastic Net	347.30	374.88	73	44
Adaptive Lasso	348.64	346.45	65	44

Πίνακας 4.1: Μέσα τετραγωνικά σφάλματα και πλήθος επεξηγηματικών μεταβλητών που περιέχονται στο τελικό μοντέλο για κάθε μία από τις μεθόδους προσαρμογής.

test (30%) παραμένει ίδια σε κάθε επανάληψη. Οπότε, με τον παρακάτω κώδικα αρχικά κατασκευάζουμε τον πίνακα test_MSE όπου κάθε του στήλη θα περιέχει τα μέσα τετραγωνικά σφάλματα για κάθε μέθοδο προσαρμογής. Για τις πρώτες 4 στήλες προσαρμόζουμε το μοντέλο χρησιμοποιώντας ως παράμετρο ποινής τη λ_{min} και παίρνουμε τα αντίστοιχα σφάλματα, ενώ για τις υπόλοιπες 4 στήλες χρησιμοποιούμε την παράμετρο λ_{1se}. Στη τελευταία στήλη θα εκχωρήσουμε τα σφάλματα που προκύπτουν ύστερα από προσαρμογή του μοντέλου με τη μέθοδο ελαχίστων τετραγώνων. Εν συνεχεία μέσα στο βρόχο for χωρίζουμε τα δεδομένα μας σε train και test set. Με την εντολή cv.glmnet εκτελούμε 10-fold cross-validation για κάθε μία μέθοδο συρρίκνωσης. Εκχωρούμε στο διάνυσμα method τα ονόματα των μεθόδων επί δύο φορές και με την εντολή get μετατρέπουμε κάθε στοιχείο του διανύσματος method σε αντικείμενο, ώστε να μπορεί να χρησιμοποιηθεί ως όρισμα στην εντολή predict. Τέλος, κατασκευάζουμε το Διάγραμμα 4.10.

```
> test_MSE=matrix(0,100,9)
> colnames(test_MSE)=c("Ridge.min","Lasso.min","El_net.min",
        "Ad_lasso.min","Ridge.1se","Lasso.1se","El_net.1se",
        "Ad_lasso.1se","LS")
> for(i in 1:100){
    train=sample(1:nrow(data),0.7*nrow(data))
    test=-train

    Ridge=cv.glmnet(x[train,],y[train],alpha=0)
    Lasso=cv.glmnet(x[train,],y[train])
    Lasso.coef=as.numeric(coef(Lasso,s=Lasso$lambda.min))[-1]
    El_net=cv.glmnet(x[train,],y[train],alpha=0.5)
    Ad_lasso=cv.glmnet(x[train,],y[train],
                       penalty.factor=1/abs(Lasso.coef))
    Least_squares=lm(y .,data.frame(x,y),subset=train)

    method=rep(c("Ridge","Lasso","El_net","Ad_lasso"),2)
    for(j in 1:8){
      mod=get(method[j])
      if(j<=4){
```

```
        test_MSE[i,j]=mean((y[test]-predict(mod,s=mod$lambda.min,
                            newx=x[test,]))^2)
      }else{
        test_MSE[i,j]=mean((y[test]-predict(mod,s=mod$lambda.1se,
                            newx=x[test,]))^2)
      }
    }
    ls_pred=predict(Least_squares,newdata=data.frame(x,y)[test,])
    test_MSE[i,9]=mean((y[test]-ls_pred)^2)

  }

> melt=melt(test_MSE)
> ggplot(melt,aes(x=factor(Var2),y=value))+
         geom_boxplot(fill="skyblue")+
         labs(title="Test MSE",x="Method",y="MSE")+
         theme(plot.title=element_text(hjust=0.5,vjust=2))
```



Διάγραμμα 4.10: Θηκοδιαγράμματα του test MSE για κάθε μία μέθοδο προσαρμογής που χρησιμοποιήσαμε στις 100 επαναλήψεις. Τα 4 πρώτα αντιστοιχούν στην παράμετρο λ_{min}, ενώ τα υπόλοιπα 4 στην παράμετρο λ_{1se}. Το τελευταίο θηκοδιάγραμμα (LS) αντιστοιχεί στη M.E.T.

Στο Διάγραμμα 4.10 παρουσιάζονται τα θηκοδιαγράμματα των test MSE για κάθε μία μέθοδο προσαρμογής που εφαρμόσαμε. Παρατηρούμε ότι η διάμεσος του σφάλματος που αντιστοιχεί στη μέθοδο Ridge για την παράμετρο ποινής λ_{min} είναι η ελάχιστη και ακολουθεί η μέθοδος Lasso. Παρακάτω υπολογίζουμε και τις αντίστοιχες διαμέσους του test MSE.

```
>  apply(test_MSE,2,median)
   Ridge.min    Lasso.min   El_net.min  Ad_lasso.min
   337.3602     339.1238    339.9865    339.5263

   Ridge.1se    Lasso.1se   El_net.1se  Ad_lasso.1se
   354.2301     354.9596    355.3374    359.6803

   LS
   343.7130
```

4.6 Συμπέρασμα

Βασιζόμενοι στα παραπάνω αποτελέσματα για το test MSE, καλούμαστε να επιλέξουμε το μοντέλο που θα χρησιμοποιήσουμε για την πρόβλεψη της μεταβλητής απόκρισης. Είδαμε, πως με τη μέθοδο Ridge καταλήγουμε σε ένα μοντέλο που έχει την καλύτερη προβλεπτική ικανότητα, δηλαδή παρουσιάζει το ελάχιστο σφάλμα. Παρ΄ όλα αυτά και από τις υπόλοιπες μεθόδους Lasso, Elastic Net και Adaptive Lasso, συμπεραίνουμε ότι τα σφάλματα δε διαφέρουν σε μεγάλο βαθμό μεταξύ τους. Επομένως, μπορούμε τελικά να επιλέξουμε ένα μοντέλο που θα περιέχει λιγότερες επεξηγηματικές μεταβλητές από τις αρχικά διαθέσιμες. Σύμφωνα με τον Πίνακα 4.1 θα επιλέξουμε το μοντέλο που προκύπτει από τη μέθοδο Elastic Net για α = 0.5, που για το συγκεκριμένο διαχωρισμό που κάναμε για το train και test set δίνει σφάλμα (για λ_{min}) με τιμή 347.30 και περιέχει 73 από τις 80 επεξηγηματικές μεταβλητές. Προσαρμόζουμε λοιπόν το μοντέλο μας με τη μέθοδο Elastic Net για τα πλήρη δεδομένα και ύστερα υπολογίζουμε τους συντελεστές για την τιμή λ_{min} που προέκυψε από cross-validation. Οι συντελεστές που προκύπτουν φαίνονται παρακάτω.

```
> #final model
> model=glmnet(x,y,alpha=0.5)
> final_coef=coef(model,s=lam_el_net.min)
> final_coef
81 x 1 sparse Matrix of class "dgCMatrix"
                                      1
(Intercept)               128.880068832
avgAnnCount                -0.002669650
avgDeathsPerYear            0.008878547
incidenceRate               0.182225941
medIncome                   0.000024900
popEst2015                 -0.000003593
povertyPercent              0.171936194
studyPerCap                 0.000050508
MedianAge                   .
MedianAgeMale              -0.245042257
MedianAgeFemale            -0.121390415
```

```
StateAlaska                 15.414881108
StateArizona               -18.473333385
StateArkansas               12.340207559
StateCalifornia            -11.878450968
StateColorado              -15.344682770
StateConnecticut           -12.739102369
StateDelaware               -4.406512020
StateDistrict of Columbia    .
StateFlorida                 .
StateGeorgia                -4.297805774
StateHawaii                -28.660774702
StateIdaho                 -15.598034059
StateIllinois               -0.841441955
StateIndiana                 7.525376778
StateIowa                   -7.884022648
StateKansas                  3.004601406
StateKentucky               12.094907180
StateLouisiana               1.028932809
StateMaine                   1.550560874
StateMaryland                2.926552001
StateMassachusetts          -6.008037784
StateMichigan               -0.074968251
StateMinnesota              -4.884842513
StateMississippi             6.293188035
StateMissouri                9.393076683
StateMontana                -9.403753852
StateNebraska               -4.058342670
StateNevada                  .
StateNew Hampshire          -0.472139216
StateNew Jersey             -2.985479230
StateNew Mexico            -12.506300666
StateNew York              -12.058099321
StateNorth Carolina         -4.889985325
StateNorth Dakota           -0.936816712
StateOhio                    4.157496929
StateOklahoma               13.550981098
StateOregon                 -6.385843604
StatePennsylvania           -7.964073428
StateRhode Island           -1.263886426
StateSouth Carolina          1.153859615
StateSouth Dakota           -4.816262255
StateTennessee               8.024950744
StateTexas                   2.729734844
StateUtah                  -19.702355570
StateVermont                 .
StateVirginia                7.921512630
```

```
StateWashington              -5.789496048
StateWest Virginia            3.593684292
StateWisconsin               -2.021228598
StateWyoming                 -1.412277982
AvgHouseholdSize              0.708752856
PercentMarried                0.541869001
PctNoHS18_24                 -0.010551158
PctHS18_24                    0.179489074
PctBachDeg18_24              -0.187553584
PctHS25_Over                  0.224756118
PctBachDeg25_Over            -0.943719828
PctEmployed16_Over           -0.228115402
PctUnemployed16_Over          0.445010828
PctPrivateCoverage           -0.063860145
PctPrivateCoverageAlone       .
PctEmpPrivCoverage            0.123820897
PctPublicCoverage             .
PctPublicCoverageAlone        0.488270287
PctWhite                     -0.089553003
PctBlack                     -0.059857940
PctAsian                      0.123970527
PctOtherRace                 -0.563611504
PctMarriedHouseholds         -0.781927847
BirthRate                    -0.597731887
```

Προκειμένου λοιπόν να προβλέψουμε το μέσο αριθμό θανάτων (ανά 100,000 άτομα) που οφείλονται στη νόσο του καρκίνου σε διάφορες κομητείες της Αμερικής, χρησιμοποιούμε μεταβλητές όπως ο μέσος αριθμός διαγνώσεων καρκίνου (ανά 100,000) (incidenceRate), το ποσοστό φτώχειας (povertyPercent), η διάμεσος της ηλικίας των αντρών της κομητείας (MedianAgeMale) και διάφορες εικονικές μεταβλητές που σχετίζονται με ορισμένες πολιτείες της Αμερικής (State). Επίσης στην επεξήγηση του μέσου αριθμού θανάτων που οφείλονται στη νόσο του καρκίνου, συνεισφέρουν και μεταβλητές όπως το μέσο μέγεθος των νοικοκυριών σε κάθε κομητεία (AvgHouseholdSize), το ποσοστό των κατοίκων της κομητείας που είναι παντρεμένοι (PercentMarried), και μεταβλητές που σχετίζονται με τη μόρφωση και την ασφάλεια υγείας των κατοίκων της κομητείας, όπως για παράδειγμα το ποσοστό των κατοίκων της κομητείας ηλικίας 25 ετών και άνω με πτυχίο Bachelor (PctBachDeg25_Over) και το ποσοστό των κατοίκων της κομητείας στους οποίους η κυβέρνηση παρέχει κάλυψη υγείας αλλά συνεισφέρουν και οι ίδιοι (PctPublicCoverageAlone). Τέλος, με βάση το μοντέλο στο οποίο έχουμε καταλήξει, συνεισφορά έχουν και μεταβλητές όπως π.χ το ποσοστό των κατοίκων της κομητείας που αναγνωρίζονται ως λευκοί (PctWhite), το ποσοστό των παντρεμένων νοικοκυριών (PctMarriedHouseholds), ο αριθμός γεννήσεων σε σχέση με τον αριθμό των γυναικών κάθε κομητείας (BirthRate) κ.ά. Η συνεισφορά κάθε μεταβλητής εξαρτάται από το συντελεστή της. Ο σταθερός όρος για το μοντέλο μας εκτιμάται από την τιμή 128.88. Επίσης επτά από τις αρχικές διαθέσιμες μεταβλητές δεν συμμετέχουν τελικά στο μοντέλο μας, εφόσον οι συντελεστές τους εκτιμώνται ως μηδενικοί. Για παράδειγμα οι μεταβλητές MedianAge (διάμεσος ηλικίας των κατοίκων της κομητείας) και PctPublicCoverage (ποσοστό κατοίκων της κομητείας με δημόσια κάλυψη υγείας) δε συνεισφέρουν στην πρόβλεψη του αριθμού θανάτων που οφείλονται στη νόσο του καρκίνου. Με τον παρακάτω κώδικα αρχικά εκχωρούμε στη μεταβλητή names

τα ονόματα των μεταβλητών που συμμετέχουν στο τελικό μας μοντέλο και στη συνέχεια υπολογίζουμε το διαγνωστικό ελέγχου VIF για κάθε μεταβλητή και τα ταξινομούμε σε φθίνουσα σειρά (εδώ παρουσιάζουμε μόνο τις τιμές των VIF που είναι μεγαλύτερες από 10 και άρα υποδεικνύουν πρόβλημα πολυσυγγραμμικότητας).

```
> #check collinearity
> names=final_coef@Dimnames[[1]][final_coef@i+1][-1]
> m_shrinked=lm(y .,data=data.frame(x[,names],y))
> newVIF=round(vif(m_shrinked),1)
> sort(newVIF,decreasing = TRUE)
  avgDeathsPerYear      avgAnnCount        popEst2015
  54.4                  40.1               29.5

  PctPrivateCoverage    PercentMarried     PctPublicCoverageAlone
  20.5                  12.1               11.0

  PctWhite              MedianAgeFemale
  10.8                  10.6
```

Από τα παραπάνω αποτελέσματα είναι φανερό ότι οι τιμές των VIF έχουν ελαττωθεί αρκετά (σε σχέση με τις αντίστοιχες τιμές, όταν είχαμε χρησιμοποιήσει όλες τις διαθέσιμες επεξηγηματικές μεταβλητές) αλλά υπάρχουν ακόμα παράγοντες των οποίων οι τιμές του VIF είναι μεγαλύτερες του 10. Παρατηρούμε λοιπόν ότι με βάση το μοντέλο στο οποίο καταλήξαμε, λύνεται και σε ένα μικρό βαθμό το πρόβλημα της πολυσυγγραμμικότητας, εφόσον ορισμένες από τις επεξηγηματικές μεταβλητές με υψηλά VIF τελικά δε συμμετέχουν στο μοντέλο μας (π.χ Median Age, PctPublicCoverage).

Βιβλιογραφία

(Α) Διεθνής

Andrieu, C., de Freitas, N., Doucet, A. and Jordan, M. I. (2003).*An introduction to MCMC for machine learning.* Machine learning 50.1-2: 5-43.

Boehmke, B. and Greenwell, B. M. (2019). *Hands-on machine learning with R*. CRC Press.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap.* CRC press.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736.*

Friedman, J., Hastie, T., Höfling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization. *The annals of applied statistics*, 1(2), 302-332.

Giraud, C. (2014). *Introduction to high-dimensional statistics* (Vol. 138). CRC Press.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations. CRC press.*

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.

Mazumder, R., Friedman, J. H. and Hastie, T. (2011). Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association,* 106(495), 1125-1138.

Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association,* 103(482), 681-686.

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics,* 22(2), 231-245.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*: Series B (Methodological), 58(1), 267-288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* 67(1), 91-108.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondi erentiable minimization. *Journal of optimization theory and applications*, 109(3), 475-494.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* 68(1), 49-67.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association,* 101(476), 1418-1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology),* 67(2), 301-320.

(Β) Ελληνική

Οικονόμου, Π. και Καρώνη, Χ. (2010). Statistikˆ Montèla Palindrìmhshc. Εκδόσεις Συμεών Αθήνα.

Φουσκάκης, Δ. (2013). Anˆlush Dedomènwn me qr sh thc *R*. Εκδόσεις ΤΣΟΤΡΑΣ