Εργασία στην Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση

Ευάγγελος Τσόγκας
Μεταπτυχιακός Φοιτητής
Email: evaggelostsogkas@mail.ntua.gr
Αριθμός Μητρώου: 03400120

ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση
Εθνικό Μετσόβιο Πολυτεχνείο
Ιούνιος 2021

Άσκηση 1

Μας δίνεται η σ.π.π. της τυποποιημένης κανονικής κατανομής, $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, καθώς και το ολοκλήρωμα

$$J = \int_{-\infty}^{+\infty} (x+a)^2 \phi(x)\, dx = 1 + a^2$$

(α) Θέλουμε να εκτιμήσουμε το J με χρήση Monte Carlo ολοκλήρωσης, προσομοιώνοντας τιμές από την τυποποιημένη κανονική κατανομή. Το ολοκλήρωμα μπορεί να γραφεί ως

$$J = \int_{-\infty}^{+\infty} h(x)\phi(x)\, dx$$

Δεδομένου ότι η $\phi(x)$ είναι η σ.π.π. της κατανομής από την οποία θέλουμε να προσομοιώσουμε τιμές, ο εκτιμητής του J με Monte Carlo ολοκλήρωση είναι ο

$$\hat{J}_N = \frac{1}{N} \sum_{i=1}^{N} h(x_i)$$

Στον πίνακα 1 φαίνονται οι εκτιμήσεις $\hat{J}_N$ και οι πραγματικές τιμές του ολοκληρώματος J για $a = 0, 1, 2, 3, 4$ και αριθμό προσομοιωμένων τιμών $N = 100, 1000$. Παρατηρούμε πως οι εκτιμήσεις Monte Carlo είναι αρκετά κοντά στις πραγματικές τιμές του ολοκληρώματος J για κάθε τιμή a, N.

a	$\hat{J}_{100}$	$\hat{J}_{1000}$	J
0	1.179	1.077	1
1	1.981	1.951	2
2	5.357	5.061	5
3	10.601	10.213	10
4	17.582	16.626	17

Πίνακας 1: Αποτελέσματα εκτίμησης J με χρήση Monte Carlo ολοκλήρωσης.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή των αποτελεσμάτων του πίνακα 1:

```
for(N in c(100, 1000)) {
  for(a in 0:4) {

    # simulate N values from standard normal distribution
    x <- rnorm(N, 0, 1)

    # Monte Carlo estimator of J
    J_est <- 1/length(x) * sum((x+a)**2)

```

```
    # actual value of J
    J <- 1 + a**2

    cat("\n\nN =", N, "a =", a)
    cat("\nEstimated J =", J_est)
    cat("\nTrue J =", J)
  }
}
```

(β) Στη συνέχεια, αποδεικνύουμε πως ο εκτιμητής του ερωτήματος (α) είναι αμερόληπτος και βρίσκουμε τη θεωρητική του τυπική απόκλιση. Για να είναι αμερόληπτος ο εκτιμητής θα πρέπει να έχει μηδενική μεροληψία, δηλαδή:

$$bias(\hat{J}_N) = E[\hat{J}_N] - J = 0 \Leftrightarrow E[\hat{J}_N] = J$$

Πράγματι αυτό αποδεικνύεται ως εξής:

$$E[\hat{J}_N] = E[\frac{1}{N}\sum_{i=1}^{N} h(x_i)] = \frac{1}{N}\sum_{i=1}^{N} E[h(x_i)] = E[h(x)] = \int_{-\infty}^{+\infty} h(x)\phi(x)\,dx = J$$

Προκειμένου να βρούμε την τυπική απόκλιση του εκτιμητή, αρχικά υπολογίζουμε τη διασπορά:

$$\begin{aligned}
Var(\hat{J}_N) &= E[(\hat{J}_N - E[\hat{J}_N])^2] = E[\hat{J}_N^2] - E[\hat{J}_N]^2 \\
&= E[(\frac{1}{N}\sum_{i=1}^{N} h(x_i))^2] - E[\frac{1}{N}\sum_{i=1}^{N} h(x_i)]^2 \\
&= \frac{1}{N^2}(\sum_{i=1}^{N} E[(x_i + a)^4] - \sum_{i=1}^{N} E[(x_i + a)^2]^2) \\
&= \frac{1}{N^2}(\sum_{i=1}^{N} (E[x_i^4] + 4aE[x_i^3] + 6a^2E[x_i^2] + 4a^3E[x_i] + a^4) \\
&- \sum_{i=1}^{N} (E[x_i^2] + 2aE[x_i] + a^2)^2) \\
&= \frac{1}{N^2}(N(3 + 6a^2 + a^4) - N(1 + 2a^2 + a^4)) \\
&= \frac{4a^2 + 2}{N}
\end{aligned}$$

Επομένως, η τυπική απόκλιση είναι η

$$\sigma(\hat{J}_N) = \sqrt{Var(\hat{J}_N)} = \sqrt{\frac{4a^2 + 2}{N}}$$

Σημειώνεται ότι στους παραπάνω υπολογισμούς χρησιμοποιήσαμε το γεγονός πως για την τυποποιημένη κανονική κατανομή ισχύει ότι $E[X] = 0$, $E[X^2] = 1$, $E[X^3] = 0$ και $E[X^4] = 3$.

(γ) Έπειτα, επαναλαμβάνουμε τα ερωτήματα (α) και (β), αλλά αυτή τη φορά εφαρμόζοντας δειγματοληψία σπουδαιότητας, μια τεχνική μείωσης διασποράς η ιδέα της οποίας είναι να προσομοιώσουμε τιμές από μια δοκιμαστική κατανομή με σ.π.π. $g(x)$ αντί της $\phi(x)$. Αν επιλεχθεί μια "καλή" $g(x)$ τότε η διασπορά μπορεί να ελαττωθεί, αλλά σε αντίθετη περίπτωση υπάρχει κίνδυνος να αυξηθεί. Στην περίπτωσή μας χρησιμοποιούμε τη συνάρτηση $g(x) = \phi(x - a)$, δηλαδή τη σ.π.π. της κανονικής κατανομής με μέση τιμή a και τυπική απόκλιση 1. Έστω, λοιπόν, η συνάρτηση

$$\begin{aligned} y(x) = \frac{h(x)\phi(x)}{g(x)} &= \frac{(x+a)^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}}{\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2}}} \\ &= (x+a)^2 e^{\frac{1}{2}(a^2 - 2ax)}, \end{aligned}$$

τότε το ολοκλήρωμα J παίρνει τη μορφή

$$J = \int_{-\infty}^{+\infty} y(x) g(x)\, dx = E[y(x)] = 1 + a^2$$

και ο εκτιμητής του J με Monte Carlo ολοκλήρωση είναι ο

$$\hat{J}_N = \frac{1}{N} \sum_{i=1}^{N} y(x_i) = \frac{1}{N} \sum_{i=1}^{N} (x_i + a)^2 e^{\frac{1}{2}(a^2 - 2ax_i)}$$

Στον πίνακα 2 φαίνονται οι εκτιμήσεις $\hat{J}_N$ και οι πραγματικές τιμές του ολοκληρώματος J για $a = 0, 1, 2, 3, 4$ και αριθμό προσομοιωμένων τιμών $N = 100, 1000$. Παρατηρούμε πως οι εκτιμήσεις Monte Carlo για μεγάλο a δεν είναι καλές, καθώς όσο μεγαλώνει το a τόσο περισσότερο διαφέρει η $g(x)$ από τη $\phi(x)$ και η διασπορά αυξάνεται.

a	$\hat{J}_{100}$	$\hat{J}_{1000}$	J
0	0.865	0.990	1
1	2.057	1.917	2
2	4.177	4.779	5
3	9.703	6.603	10
4	3.942	7.248	17

Πίνακας 2: Αποτελέσματα εκτίμησης J με χρήση Monte Carlo ολοκλήρωσης και δειγματοληψίας σπουδαιότητας.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή των αποτελεσμάτων του πίνακα 2:

```
for(N in c(100, 1000)) {
  for(a in 0:4) {

    # simulate N values from normal distribution with mean = a and
    sd=1
    x <- rnorm(N, a, 1)

    # Monte Carlo estimator of J
    J_est <- 1/length(x) * sum((x+a)**2 * exp(1/2*(a**2 - 2*a*x)))

    # actual value of J
    J <- 1 + a**2

    cat("\n\nN =", N, "a =", a)
    cat("\nEstimated J =", J_est)
    cat("\nTrue J =", J)
  }
}
```

Ομοίως με το ερώτημα (β) αποδεικνύεται πως ο παραπάνω εκτιμητής είναι αμερόληπτος ως εξής:

$$E[\hat{J}_N] = E[\frac{1}{N}\sum_{i=1}^{N} y(x_i)] = \frac{1}{N}\sum_{i=1}^{N} E[y(x_i)] = E[y(x)] = J$$

Για να βρούμε τη θεωρητική τυπική απόκλιση του εκτιμητή υπολογίζουμε αρχικά τη διασπορά του:

$$\begin{aligned} Var(\hat{J}_N) &= E[(\hat{J}_N - E[\hat{J}_N])^2] = E[\hat{J}_N^2] - E[\hat{J}_N]^2 \\ &= \frac{1}{N}(\int_{-\infty}^{+\infty} y^2(x)g(x)\,dx - (1+a^2)^2) \\ &= \frac{1}{N}(\int_{-\infty}^{+\infty} (x+a)^4 e^{a^2-2ax}\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-a)^2}{2}}\,dx - (1+a^2)^2) \\ &= \frac{1}{N}(3e^{a^2} - a^4 - 2a^2 - 1) \end{aligned}$$

Επομένως, η τυπική απόκλιση είναι η

$$\sigma(\hat{J}_N) = \sqrt{Var(\hat{J}_N)} = \sqrt{\frac{3e^{a^2} - a^4 - 2a^2 - 1}{N}}$$

Έχουμε λοιπόν τα θεωρητικά τυπικά σφάλματα $se(\hat{J}_N(\alpha)) = \sqrt{\frac{4a^2+2}{N}}$ και $se(\hat{J}_N(\gamma)) = \sqrt{\frac{3e^{a^2}-a^4-2a^2-1}{N}}$ των εκτιμητών των ερωτημάτων (α) και (γ) αντίστοιχα. Στον πίνακα 3 φαίνονται οι τιμές των παραπάνω τυπικών σφαλμάτων για $N = 1000$ και $\alpha = 0, 1, 2, 3, 4$. Όπως περιμέναμε, φαίνεται πως το τυπικό σφάλμα του εκτιμητή $\hat{J}_N(\gamma)$ αυξάνεται εκθετικά, καθώς αυξάνεται η τιμή του α.

a	$se(\hat{J}_{1000}(\alpha))$	$se(\hat{J}_{1000}(\gamma))$
0	0.045	0.045
1	0.077	0.064
2	0.134	0.373
3	0.195	4.920
4	0.257	163.273

Πίνακας 3: Σύγκριση των θεωρητικών τυπικών σφαλμάτων των εκτιμητών των ερωτημάτων (α) και (γ) αντίστοιχα.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή των αποτελεσμάτων του πίνακα 3:

```
N <- 1000
for(a in 0:4) {
  # estimate theoretical standard error of J (a)
  se_a <- sqrt((4*a**2 + 2) / N)

  # estimate theoretical standard error of J (c)
  se_c <- sqrt((3*exp(a**2) - a**4 - 2*a**2 - 1) / N)

  cat("\n\na =", a)
  cat("\nTheoretical se (a) =", se_a)
  cat("\nTheoretical se (c) =", se_c)
}
```

(δ) Τέλος, χρησιμοποιώντας την τεχνική δειγματοληψίας με επανάθεση, Bootstrap, εκτιμούμε το τυπικό σφάλμα του εκτιμητή του ερωτήματος (α) για $N = 1000$ προσομοιωμένες τιμές και $\alpha = 4$. Το τυπικό σφάλμα με την τεχνική Bootstrap υπολογίζεται ως:

$$se_{boot}(\hat{J}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{J}_b - \bar{J})^2},$$

όπου $\bar{J} = \frac{1}{B}\sum_{b=1}^{B}\hat{J}_b$ είναι η μέση τιμή των B εκτιμητών Bootstrap. Επιλέγουμε να δημιουργήσουμε $B = 1000$ Bootstrap δείγματα, αφού εμπειρικά είναι αρκετά για τη λήψη ικανοποιητικών αποτελεσμάτων. Η τιμή του θεωρητικού τυπικού σφάλματος του εκτιμητή είναι η

$$se(\hat{J}) = 0.257$$

και η εκτίμησή του με την τεχνική Bootstrap είναι η

$$se_{boot}(\hat{J}) = 0.252$$

Παρατηρούμε πως η εκτίμηση είναι πολύ καλή. Παρακάτω φαίνεται ο σχετικός κώδικας ο οποίος περιλαμβάνει συνάρτηση για την εκτίμηση του τυπικού σφάλματος με Bootstrap.

```r
# calculates the standard error of J Monte Carlo estimator using
   the bootstrap method
bootstrap_se <- function(x, B, a) {

  # initialize J estimates
  J_est <- rep(NA, B)
  # perform bootstrap sampling
  for (b in 1:B) {
    # sample with replacement
    bootstrap <- sample(x, replace = TRUE)
    # estimate J from the bootstrap sample
    J_est[b] <- (1/length(bootstrap) * sum((bootstrap+a)**2))
  }

  # calculate standard error
  se <- sqrt(1/(B-1) * sum((J_est - mean(J_est))**2))
  return (se)
}


N <- 1000
a <- 4
B <- 1000 # number of bootstrap samples
# simulate N values from the standard normal distribution
x <- rnorm(N, 0, 1)

# estimated standard error using the bootstrap method
se_est <- bootstrap_se(x, B, a)

# estimate theoretical standard error
se_th <- sqrt((4*a**2 + 2) / N)

cat("Bootstrap se =", se_est)
cat("Theoretical se =", se_th)
```

Άσκηση 2

Θέλουμε να προσομοιώσουμε 1000 τιμές από την σ.π.π.

$$f(x) = \frac{1}{e^3 - 1} e^x, x \in [0, 3]$$

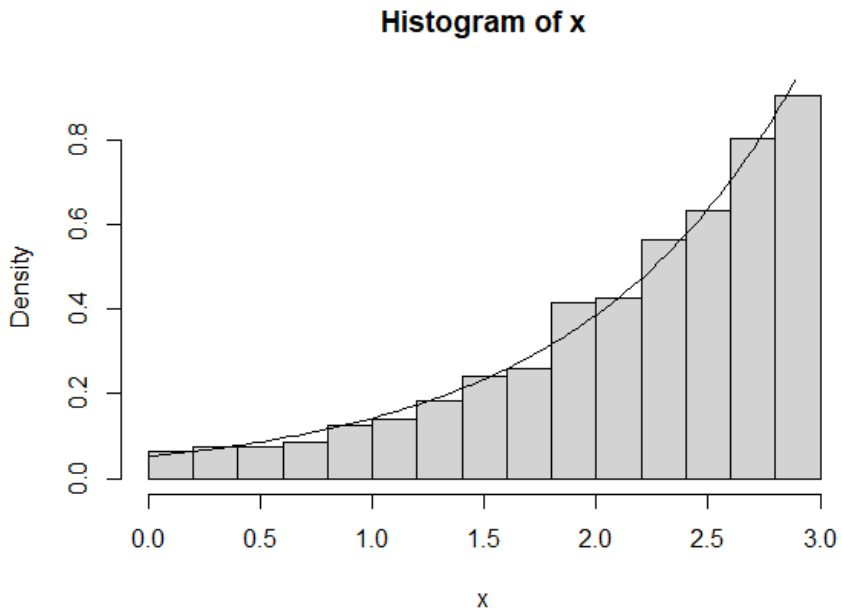
(α) Αρχικά, θα χρησιμοποιήσουμε τη μέθοδο αντιστροφής η οποία βασίζεται στην παρατήρηση ότι η τυχαία μεταβλητή $x = F^{-1}(u)$, όπου $u \sim U(0, 1)$, ακολουθεί την επιθυμητή κατανομή F. Από τον ορισμό βρίσκουμε τη συνάρτηση κατανομής $F(x)$ ως:

$$F(x) = \int_{-\infty}^{x} f(t)\,dt = \int_{0}^{x} \frac{e^t}{e^3-1}\,dt = \frac{1}{e^3-1}\int_{0}^{x} e^t\,dt = \frac{e^x-1}{e^3-1}$$

Άρα, με τη μέθοδο αντιστροφής:

$$x = F^{-1}(u) \Leftrightarrow u = \frac{e^x-1}{e^3-1} \Leftrightarrow e^x = (e^3-1)u+1 \Leftrightarrow x = ln((e^3-1)u+1)$$

Προσομοιώνουμε, λοιπόν, τιμές $u \sim U(0,1)$ και βρίσκουμε το ιστόγραμμα από τις παραπάνω τιμές x. Στο σχήμα 1 φαίνεται πως το ιστόγραμμα των x έχει προσεγγίσει σε ικανοποιητικό βαθμό το διάγραμμα της $f(x)$.



Σχήμα 1: Ιστόγραμμα προσομοιωμένων τιμών με τη μέθοδο αντιστροφής, συνοδευόμενο από το διάγραμμα της $f(x)$.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή του σχήματος 1:

```
# simulate 1000 values from the uniform distribution
u <- runif(1000, 0, 1)

# inverse transform sampling
x <- log((exp(3)-1)*u + 1)

# plot histogram
hist(x, probability = TRUE)

# plot pdf
f <- function(x) {(exp(x))/(exp(3)-1)}
curve(f, add=TRUE)
```

(β) Στη συνέχεια, θα χρησιμοποιήσουμε τη μέθοδο απόρριψης για την προσομοίωση. Η μέθοδος απόρριψης είναι χρήσιμη για να προσομοιώσουμε τιμές από κατανομές των οποίων η F^{-1} δεν μπορεί να γραφεί σε κλειστή μορφή, οπότε δεν θα μπορούσαμε να χρησιμοποιήσουμε τη μέθοδο της αντιστροφής. Υποθέτουμε, λοιπόν, ότι είναι πιο εύκολο να παράγουμε τιμές από μια σ.π.π. $g(x)$ αντί της $f(x)$, για την οποία μπορούμε να βρούμε κάποιο $M > 0$ τέτοιο ώστε:

$$\frac{f(x)}{g(x)} \leq M$$

Παράγουμε, λοιπόν, $Y \sim g(y)$ και $u \sim U(0,1)$ μέχρις ότου $u \leq \frac{f(y)}{Mg(y)}$, όπου θέτουμε $X = y$ και σταματάμε. Κατά αυτόν τον τρόπο δεχόμαστε μόνο τα σημεία που πέφτουν κάτω από την $f(y)$. Επιλέγουμε ως $g(y)$ την σ.π.π. της $U(0,3)$, στο ίδιο πεδίο ορισμού με την $f(x)$ ώστε να την μιμείται όσο το δυνατόν καλύτερα. Άρα,

$$g(y) = \frac{1}{3}$$

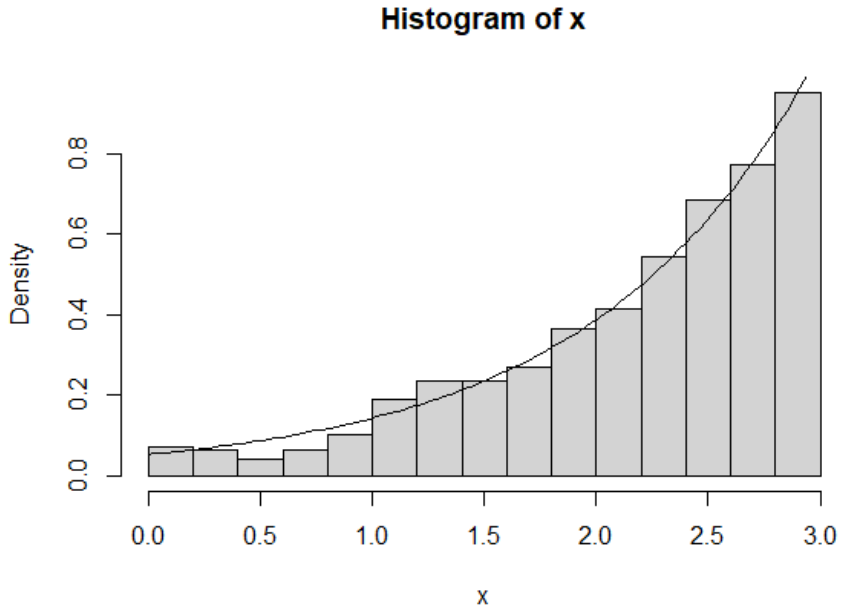
Προκειμένου το M να είναι το μικρότερο δυνατό, ώστε να μιμείται καλύτερα η $g(y)$ το σχήμα της $f(y)$, επιλέγουμε το M ως εξής:

$$f(y) \leq Mg(y) \Rightarrow M \geq \frac{f(y)}{g(y)} \Rightarrow M = max_y \frac{f(y)}{g(y)} = max_y \frac{3e^y}{e^3 - 1}$$

Η συνάρτηση που θέλουμε να μεγιστοποιήσουμε είναι γνησίως αύξουσα, οπότε εμφανίζει μέγιστη τιμή για $y = 3$ στο διάστημα $[0, 3]$. Επομένως,

$$M = \frac{3e^3}{e^3 - 1} = 3.16$$

Στο σχήμα 2 φαίνεται πως το ιστόγραμμα των τιμών x που προσομοιώσαμε με την παραπάνω μεθοδολογία έχει προσεγγίσει σε ικανοποιητικό βαθμό το διάγραμμα της $f(x)$.



Σχήμα 2: Ιστόγραμμα προσομοιωμένων τιμών με τη μέθοδο απόρριψης, συνοδευόμενο από το διάγραμμα της $f(x)$.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή του σχήματος 2:

```
N <- 1000
x <- rep(NA, N)  # final simulated values

# rejection sampling
for(i in 1:N) {
  while(TRUE) {
    # y ~ U(0, 3)
    y <- runif(1, 0, 3)

    # u ~ U(0, 1)
    u <- runif(1, 0, 1)

    # accept value
    if (u <= 3*exp(y) / (3.16 * (exp(3)-1))) {
      x[i] <- y
      break
    }
  }
}

# plot histogram
hist(x, probability = TRUE)

# plot PDF
f <- function(x) {(exp(x))/(exp(3)-1)}
curve(f, add=TRUE)
```

(γ) Έπειτα, προσομοιώνοντας 100 τιμές από την $f(x)$ μέσω της μεθόδους αντιστροφής όπως και στο ερώτημα (α), θέλουμε να εκτιμήσουμε την $f(x)$ από αυτά τα δεδομένα χρησιμοποιώντας τον Epanechnikov πυρήνα. Η ιδέα είναι ότι σε κάθε παρατήρηση τοποθετούμε έναν πυρήνα με κέντρο την παρατήρηση αυτή και εκτιμούμε την $f(x)$ αθροίζοντας τα βάρη που προκύπτουν από κάθε πυρήνα. Ορίζουμε, λοιπόν, την εκτιμήτρια:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

όπου $K(x)$ είναι ο πυρήνας και h είναι μια παράμετρος που ονομάζεται πλάτος ή παράθυρο και επηρεάζει σε μεγάλο βαθμό την ποιότητα της εκτίμησης. Τίθεται λοιπόν, το ζήτημα της εύρεσης του βέλτιστου πλάτους h. Για το σκοπό αυτό θα μεγιστοποιήσουμε την cross-validated πιθανοφάνεια. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε leave-one-out cross-validation υπολογίζοντας την πιθανοφάνεια από τη σχέση:

$$L(h, i) = \prod_{i=1}^{n} \hat{f}_{h,-i}(x)$$

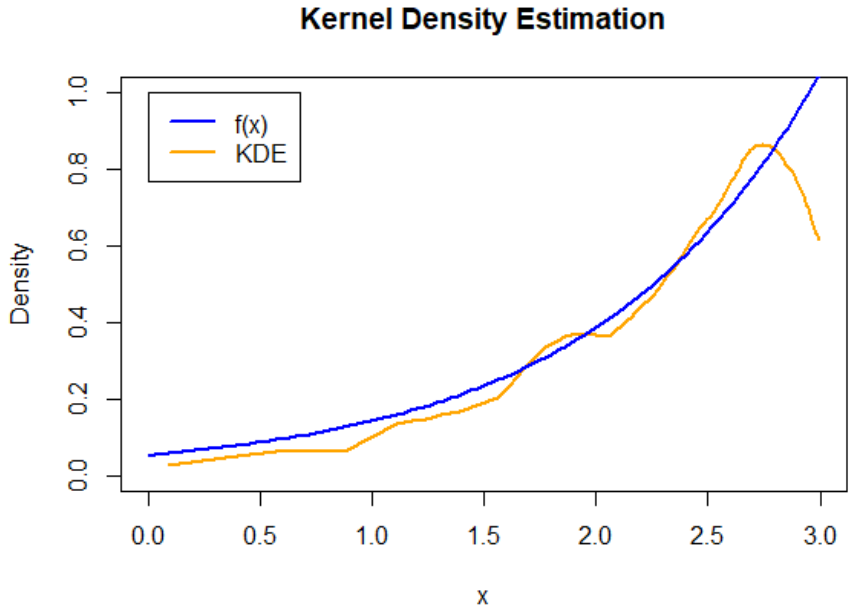
όπου $\hat{f}_{h,-i}(x)$ είναι η εκτιμήτρια πυκνότητας στην παρατήρηση που αφήσαμε εκτός. Ισοδύναμα μπορούμε να μεγιστοποιήσουμε τη λογαριθμική πιθανοφάνεια:

$$\begin{aligned} L(h) &= \sum_{i=1}^{n} log\left(\frac{1}{(n-1)h} \sum_{j=1, j\neq i}^{n} K\left(\frac{x_j - x_i}{h}\right)\right) \\ &= \frac{1}{n} \sum_{i=1}^{n} log \sum_{j=1, j\neq i}^{n} K\left(\frac{x_j - x_i}{h}\right) - log((n-1)h) \end{aligned}$$

όπου ο Epanechnikov πυρήνας έχει τη μορφή:

$$K(x) = \frac{3}{4}(1 - x^2), \text{ για } |x| \leq 1 \text{ και } 0 \text{ αλλού.}$$

Στο σχήμα 3 φαίνεται η εκτίμηση της $f(x)$ (KDE) χρησιμοποιώντας πλάτος $h = 0.35$, όπως το βρήκαμε μεγιστοποιώντας την cross-validated πιθανοφάνεια και δοκιμάζοντας τιμές στο διάστημα [0.01, 0.5] με βήμα 0.01. Παρατηρούμε πως η εκτίμηση είναι αρκετά καλή, εκτός ίσως για τιμές κοντά στο δεξί άκρο του πεδίου ορισμού, αφού εκεί η $\hat{f}(x)$ αρχίζει και πέφτει προς το 0.



Σχήμα 3: Εκτίμηση της $f(x)$ χρησιμοποιώντας Epanechnikov πυρήνα και πλάτος $h = 0.35$.

Παρακάτω φαίνεται ο σχετικός κώδικας ο οποίος περιλαμβάνει συναρτήσεις για τον υπολογισμό του Epanechnikov πυρήνα, την εύρεση βέλτιστης τιμής h και την εκτίμηση της $f(x)$ για τα δεδομένα που προσομοιώσαμε μέσω της μεθόδου αντιστροφής. Σημειώνεται πως οι συναρτήσεις για την εκτίμηση της $f(x)$ και του βέλτιστου πλάτους δέχονται ως όρισμα μια συνάρτηση υπολογισμού πυρήνα, επομένως είναι δυνατό να χρησιμοποιηθεί και πυρήνας εκτός του Epanechnikov με εύκολο τρόπο, εφόσον προγραμματιστεί.

```
# calculates the value of the Epanechnikov kernel
epanechnikov <- function(x) {
  result <- 0
  if (abs(x) <= 1) {
    result <- 3/4*(1-x**2)
  }
  return (result)
}


# finds optimal h maximizing cross-validated likelihood
find_hopt <- function(x, h_values, kernel) {
  n <- length(x)
  ML <- -Inf
  hopt <- h_values[1]
  for (h in h_values) {
    # calculate cross validated likelihood for h
    L <- 0
    for (i in 1:n) {
      # calculate kernel values without xi
      k <- sapply((x[-i]-x[i])/h, kernel)
```

```
      L <- L + log(sum(k))
    }
    L <- 1/n*L - log((n-1)*h)

    # check if likelihood is greater
    if (L > ML) {
      ML <- L
      hopt <- h
    }
  }
  return (hopt)
}


# KDE given a sample x
KDE <- function(x, h, kernel) {
  n <- length(x)
  f_est <- rep(NA, n)
  for (i in 1:n) {
    k <- sapply((x-x[i])/h, kernel)
    f_est[i] <- 1/(n*h)*sum(k)
  }
  return (f_est)
}


# f(x)
f <- function(x) {
  return (exp(x)/(exp(3)-1))
}


# inverse transform sampling
u <- runif(100, 0, 1)
x <- log((exp(3)-1)*u + 1)

# find optimal h
hopt <- find_hopt(x, seq(0.01, 0.5, 0.01), epanechnikov)
hopt

# plot estimation of f(x) along with f(x)
plot(sort(x), KDE(sort(x), hopt, epanechnikov), type="l", col="
    orange", lwd=2, main="Kernel Density Estimation", xlab="x", ylab
    ="Density", xlim=c(0, 3), ylim=c(0, 1))
curve(f, add=TRUE, col="blue", lwd=2)
legend(0,1,c("f(x)", "KDE"), lwd=c(2,2), col=c("blue", "orange"))
```

(δ) Τέλος, προσομοιώνοντας 10 τιμές από την $f(x)$ μέσω της μεθόδου αντιστροφής, θέλουμε να εφαρμόσουμε έναν Bootstrap έλεγχο υπόθεσης της μηδενικής υπόθεσης $\mu = 2$ έναντι της εναλλακτικής $\mu \neq 2$ σε επίπεδο σημαντικότητας 5%, όπου το μ δηλώνει την (υποθετικά) άγνωστη μέση τιμή της κατανομής $f(x)$. Διαλέγουμε, λοιπόν, την ελεγχοσυνάρτηση $T = |\bar{x} - 2|$ η οποία θα είναι 0 αν ισχύει η μηδενική υπόθεση H_0. Η H_0 όμως ισχύει για $\mu = 2$, ενώ η μέση τιμή του δείγματός μας είναι $\bar{x} = 2.2$. Αφαιρούμε, λοιπόν, από κάθε παρατήρηση την τιμή $T = 0.2$ και προσομοιώνουμε 1000 δείγματα bootstrap από την εμπειρική κατανομή μετακινημένη κατά T, ώστε να ικανοποιείται η H_0. Για κάθε δείγμα bootstrap υπολογίζουμε την ελεγχοσυνάρτηση $\hat{T}$. Τελικά, η $\hat{p} - value$ υπολογίζεται ως:

$$\hat{p} - value = \frac{m+1}{B+1}$$

όπου m είναι ο αριθμός των δειγμάτων bootstrap με $\hat{T} > T$. Αν $\hat{p} - value < 0.05$ η μηδενική υπόθεση απορρίπτεται, αλλιώς γίνεται δεκτή. Στην περίπτωσή μας

$$\hat{p} - value = 0.11$$

οπότε δεχόμαστε τη μηδενική υπόθεση. Επίσης, βρίσκουμε ένα 95% διάστημα εμπιστοσύνης για τη μέση τιμή, βασισμένο σε ποσοστιαία σημεία. Υπολογίζουμε τις μέσες τιμές χιλίων bootstrap δειγμάτων από το αρχικό δείγμα (όχι το μετακινημένο), τις βάζουμε σε αύξουσα σειρά και βρίσκουμε τα $\frac{\alpha}{2}$, $1 - \frac{\alpha}{2}$ ποσοστιαία σημεία των τιμών αυτών για $\alpha = 0.05$. Παρατηρούμε ότι το διάστημα που βρίσκουμε κατά αυτόν τον τρόπο είναι το

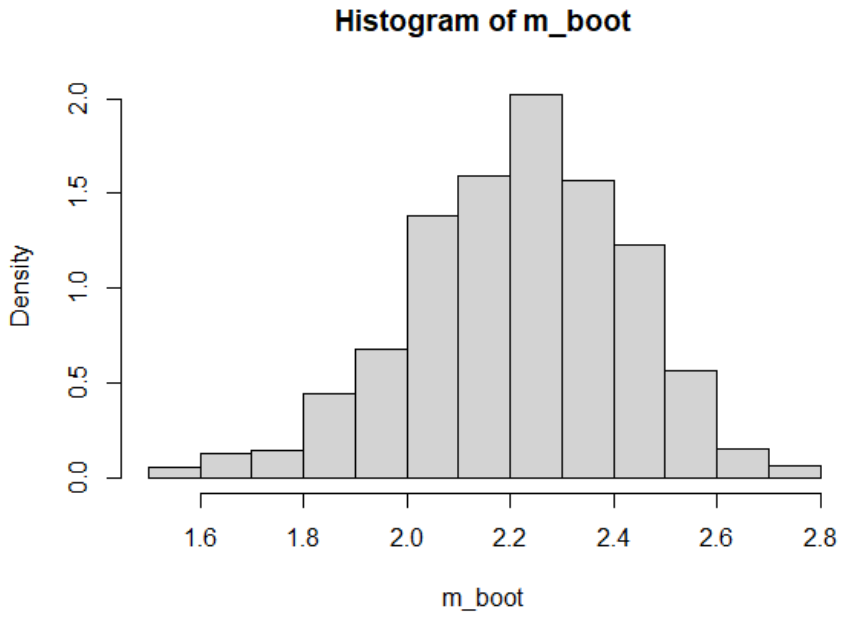
$$(1.76, 2.58)$$

και η τιμή $\mu = 2$ βρίσκεται μέσα σε αυτό, κάτι που συνάδει με το γεγονός ότι δεχτήκαμε προηγουμένως την H_0. Αυτή η μέθοδος όμως είναι ακριβής μόνο αν η κατανομή των μέσων τιμών είναι συμμετρική. Από το ιστόγραμμα του σχήματος 4 μπορούμε να θεωρήσουμε πως αυτό ισχύει και να δεχτούμε το αποτέλεσμα ως ακριβές.

Επιπλέον, βρίσκουμε και την πραγματική μέση τιμή της $f(x)$:

$$E[f(x)] = \int_0^3 x f(x)\, dx = \frac{1 + 2e^3}{e^3 - 1} = 2.1572$$

Παρατηρούμε πως η πραγματική μέση τιμή είναι πολύ κοντά στη μέση τιμή της μηδενικής υπόθεσης $\mu = 2$, επομένως, σωστά τη δεχτήκαμε τόσο με τον έλεγχο υπόθεσης όσο και με το διάστημα εμπιστοσύνης.



Σχήμα 4: Ιστόγραμμα μέσων τιμών των δειγμάτων bootstrap.

Παρακάτω φαίνεται ο σχετικός κώδικας:

```
# the test function T
test_function <- function(x) {
  return (abs(mean(x) - m_h0))
}

# returns the estimator values of every bootstrap sample
bootstrap <- function(x, B, estimator) {
  thetas <- rep(NA, B)
  for (b in 1:B) {
    bootstrap <- sample(x, replace = TRUE)
    thetas[b] <- estimator(bootstrap)
  }
  return (thetas)
}


# inverse transform sampling
n <- 10
u <- runif(10, 0, 1)
x <- log((exp(3)-1)*u + 1)

m_h0 <- 2 # null hypothesis
m <- mean(x)  # sample mean
t <- test_function(x) # test function T for the sample
cat("mean =", m)
cat("T =", t)

# center sample according to h0
if (m < m_h0) {
```

```
  x_new <- x + t
} else {
  x_new <- x - t
}

# bootstrap estimates of T
B <- 1000
t_boot <- bootstrap(x_new, B, test_function)

# calculate p-value
pvalue <- (sum(t_boot[t_boot > t])+1) / (B+1)
cat("p-value =", pvalue)

# find 95% confidence interval
a <- 0.05
m_boot <- bootstrap(x, B, mean)
m_boot <- sort(m_boot)
quantile1 <- m_boot[as.integer(round(a/2*B))]
quantile2 <- m_boot[as.integer(round((1-a/2)*B))]
cat("CI: (", quantile1, ', ', quantile2, ")", sep='')

# histogram of means
hist(m_boot, prob=T)
```

Άσκηση 3

(α) Μια στατιστική συνάρτηση $T(X_i^n)$ είναι επαρκής ως προς μια άγνωστη παράμετρο όταν κανένα άλλο στατιστικό που υπολογίζεται από το ίδιο δείγμα δεν προσφέρει επιπλέον πληροφορία για την τιμή της παραμέτρου αυτής. Έστω $X_1, ..., X_n$ ένα τυχαίο δείγμα μεγέθους n από την κατανομή $\Gamma(\alpha, \beta)$ με σ.π.π. $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$, όπου α και β είναι άγνωστες παράμετροι. Τότε, η

$$T(X_i^n) = \left(\prod_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i \right)$$

είναι η δυσδιάστατη επαρκής στατιστική συνάρτηση για (α, β). Έστω τώρα πως θέλουμε να μεγιστοποιήσουμε τη λογαριθμική πιθανοφάνεια ως προς α και β χρησιμοποιώντας τον αλγόριθμο Newton-Raphson. Για να χρησιμοποιήσουμε τη μέθοδο της μέγιστης πιθανοφάνειας καθορίζουμε πρώτα την από κοινού συνάρτηση πυκνότητας για όλες τις παρατηρήσεις:

$$\begin{aligned} f(x_1, ..., x_n | \alpha, \beta) &= \prod_{i=1}^{n} f(x_i | \alpha, \beta) = \left(\frac{\beta^\alpha}{\Gamma(\alpha)} \right)^n \left(\prod_{i=1}^{n} x_i \right)^{\alpha-1} e^{-\beta \sum_{i=1}^{n} x_i} \\ &= L(\alpha, \beta; x_1, ..., x_n) \end{aligned}$$

Επομένως, η λογαριθμική πιθανοφάνεια είναι η

$$l(\alpha,\beta) = log(L(\alpha,\beta;x_1,...,x_n)) = n\alpha log\beta - nlog\Gamma(\alpha) + (\alpha-1)\sum_{i=1}^{n} logx_i - \beta\sum_{i=1}^{n} x_i$$

Αφού μεγιστοποιούμε ως προς α, β υπολογίζουμε τις αντίστοιχες μερικές παραγώγους και τις θέτουμε ίσες με το μηδέν:

$$\frac{\partial l(\alpha,\beta)}{\partial\beta} = \frac{n\alpha}{\beta} - \sum_{i=1}^{n} x_i = 0 \Rightarrow \hat{\beta} = \frac{\alpha}{\bar{x}}$$

Για την παράμετρο β βρήκαμε λύση σε κλειστή μορφή, επομένως, η λογαριθμική πιθανοφάνεια αντικαθιστώντας την εκτιμήτρια $\hat{\beta}$ γίνεται:

$$\begin{aligned} l(\alpha,\hat{\beta}) &= n\alpha log\frac{a}{\bar{x}} - nlog\Gamma(\alpha) + (\alpha-1)\sum_{i=1}^{n} logx_i - \frac{a}{\bar{x}}\sum_{i=1}^{n} x_i \\ &= n\alpha log\alpha - n\alpha log\bar{x} - nlog\Gamma(\alpha) + (\alpha-1)\sum_{i=1}^{n} logx_i - an \end{aligned}$$

Άρα, η μερική παράγωγος ως προς α είναι η:

$$\begin{aligned} \frac{\partial l(\alpha,\hat{\beta})}{\partial\alpha} &= -nlog\bar{x} + nlog\alpha + n - n(log\Gamma(\alpha))' + \sum_{i=1}^{n} logx_i - n \\ &= -nlog\bar{x} + nlog\alpha - n\psi_0(\alpha) + \sum_{i=1}^{n} logx_i \end{aligned}$$

Παρατηρούμε ότι δεν μπορούμε να βρούμε κλειστή λύση για την παράμετρο α. Θα χρησιμοποιήσουμε, λοιπόν, τον αλγόριθμο Newton-Raphson με τη γενική αναδρομική σχέση:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Άρα, για τη μεγιστοποίηση της λογαριθμικής πιθανοφάνειας ως προς α έχουμε ότι:

$$\alpha_{n+1} = a_n - \frac{\sum_{i=1}^{n} logx_i - nlog\bar{x} + nlog\alpha_n - n\psi_0(\alpha_n)}{\frac{n}{\alpha_n} - n\psi_1(\alpha_n)}$$

και επειδή η παράμετρος β εξαρτάται από την α για τη μεγιστοποίηση με Newton-Raphson:

$$\beta_n = \frac{\alpha_n}{\bar{x}}$$

Σημειώνεται ότι στις παραπάνω σχέσεις:

$$\psi_0(\alpha) = \frac{d}{d\alpha} log\Gamma(\alpha) \text{ είναι η δίγαμμα συνάρτηση και}$$

$$\psi_1(\alpha) = \frac{d^2}{d\alpha^2} log\Gamma(\alpha) \text{ η τρίγαμμα συνάρτηση.}$$

Άσκηση 4

Θεωρούμε το πρόβλημα επιλογής επεξηγηματικών μεταβλητών στην πολλαπλή γραμμική παλινδρόμηση με $n = 50$ παρατηρήσεις και $p = 15$ επεξηγηματικές μεταβλητές. Για τις 10 πρώτες μεταβλητές θέλουμε να προσομοιώσουμε από την πολυδιάστατη κανονική κατανομή με μέση τιμή 0 και πίνακα συνδιακύμανσης τον ταυτοτικό. Αυτό σημαίνει πως οι μεταβλητές αυτές είναι στατιστικά ανεξάρτητες και επομένως μπορούμε να προσομοιώσουμε τιμές για κάθε μία ξεχωριστά από την τυποποιημένη κανονική κατανομή $N(0, 1)$. Για τις υπόλοιπες επεξηγηματικές μεταβλητές προσομοιώνουμε τιμές με βάση τη σχέση:

$X_{ij} \sim N(0.2X_{i1} + 0.4X_{i2} + 0.6X_{i3} + 0.8X_{i4} + 1.1X_{i5}, 1)$, $j = 11, ..., 15$ και $i = 1, ..., 50$.

Για τη μεταβλητή απόκρισης προσομοιώνουμε τιμές με βάση τη σχέση:

$$Y_i \sim N(4 + 2X_{i1} - X_{i5} + 2.5X_{i7} + 1.5X_{i11} + 0.5X_{i13}, 1.5^2),\ i = 1, ..., 50.$$

Το πλήρες πολλαπλό γραμμικό μοντέλο έχει τη μορφή:

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_{15} X_{15} + \epsilon,\ \epsilon \sim N(0, \sigma^2).$$

Παρακάτω φαίνεται ο κώδικας για την προσομοίωση των τιμών κάθε μεταβλητής, τις οποίες αποθηκεύουμε σε ένα data table:

```
1 library(data.table)
2 library(glmnet)
3
4 n=50
5 vnames <- c('X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', '
     X10', 'X11', 'X12', 'X13', 'X14', 'X15')
6
```

```r
# simulate values for the variables X1-X10
data <- data.table(NULL)
for (v in vnames[1:10]) {
  data[, v] <- rnorm(n)
}

# simulate values for the variables X11-X15
for (v in vnames[11:15]) {
  for (i in 1:n) {
    mean <- 0.2*data[[i, 'X1']] + 0.4*data[[i, 'X2']] + 0.6*data[[i, 'X3']] + 0.8*data[[i, 'X4']] + 1.1*data[[i, 'X5']]
    data[i, v] <- rnorm(1, mean, 1)
  }
}

# simulate values for the response variable Y
for (i in 1:n) {
  mean <- 4 + 2*data[[i, 'X1']] - data[[i, 'X5']] + 2.5*data[[i, 'X7']] + 1.5*data[[i, 'X11']] + 0.5*data[[i, 'X13']]
  data[i, 'Y'] <- rnorm(1, mean, 1.5)
}
```

(α) Στη συνέχεια, εξερευνούμε τον χώρο όλων των πιθανών μοντέλων, δηλαδή $2^{15} = 32768$ μοντέλα, επιλέγοντας το καλύτερο ως αυτό με την μικρότερη τιμή του κριτηρίου BIC. Παρακάτω φαίνεται ο κώδικας με τη συνάρτηση που εκτελεί αυτή τη διαδικασία:

```r
# function to perform full enumeration of the model space using BIC
bestLM <- function(xnames, yname, data) {
  # fit null model
  best_model <- lm(paste(yname, "~", 1), data = data)
  min_bic <- BIC(best_model)

  for (m in 1:length(xnames)) {
    combinations <- combn(xnames, m)  # get combinations m at a time
    for (c in 1:ncol(combinations)) {
      # fit a model for each combination
      model <- lm(paste(paste(yname, "~"), paste(combinations[, c], collapse="+")), data = data)
      bic <- BIC(model)

      # check if this model is better
      if (bic < min_bic) {
        min_bic <- bic
        best_model <- model
      }
    }
  }

  return (list(best_model, min_bic))
}

# find best model
results <- bestLM(vnames, 'Y', data)
```

```
results
```

Το μοντέλο που επιλέχθηκε κατά αυτόν τον τρόπο είναι το

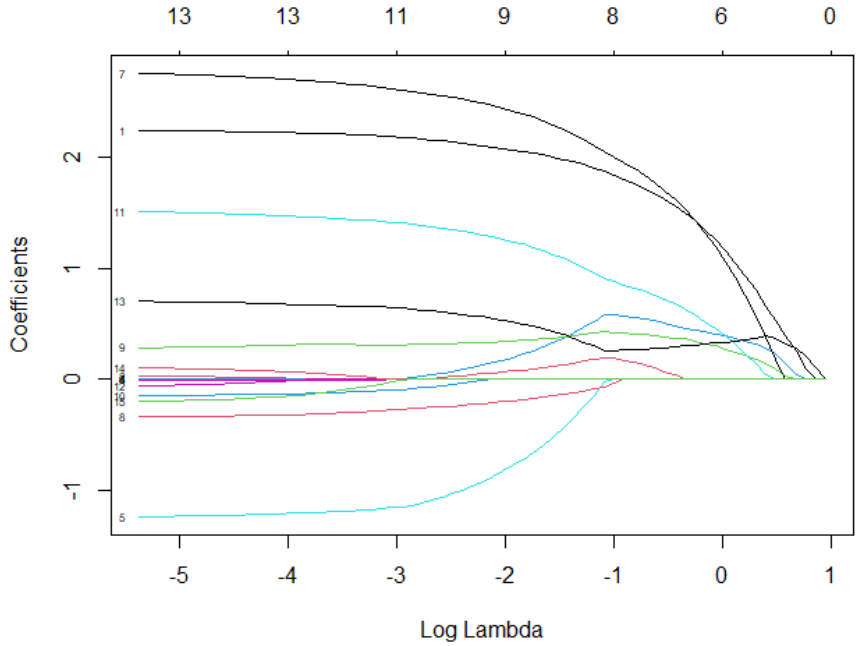
$$Y = 3.92 + 2.19X_1 - 1.46X_5 + 2.74X_7 + 1.5X_{11} + 0.78X_{13}$$

με BIC = 200.14. Παρατηρούμε πως οι μεταβλητές που επιλέχθηκαν είναι αυτές από τις οποίες εξαρτάται η Y, δηλαδή αυτές που χρησιμοποιήσαμε για τον υπολογισμό των μέσων τιμών των κατανομών από τις οποίες προσομοιώσαμε τιμές για την Y. Επιπλέον, βλέπουμε πως και η προσέγγιση των αντίστοιχων συντελεστών είναι αρκετά καλή.

(β) Στη συνέχεια, θα χρησιμοποιήσουμε τη μεθοδολογία Lasso για επιλογή μεταβλητών, αφού με τη χρήση της L1 norm συρρικνώνει έως και μηδενίζει τους συντελεστές των μεταβλητών που δεν είναι αρκετά σημαντικές για την πρόβλεψη της εξαρτημένης μεταβλητής. Θέλουμε, δηλαδή, να βρούμε τους συντελεστές β που ελαχιστοποιούν την ποσότητα

$$(y - Z\beta)^T(y - Z\beta) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

όπου η παράμετρος λ ελέγχει το βαθμό κανονικοποίησης. Εφαρμόζοντας Lasso λαμβάνουμε το σχήμα 5. Από αυτό το διάγραμμα βγάζουμε συμπεράσματα για το ποιοι συντελεστές και κατ' επέκταση ποιες μεταβλητές καταφέρνουν να επιβιώσουν στο μοντέλο, καθώς η τιμή της παραμέτρου λ αυξάνεται. Παρατηρούμε ότι για πολύ μικρό λ σχεδόν όλοι οι συντελεστές περιλαμβάνονται στο μοντέλο, ενώ για πολύ μεγάλο λ όλοι οι συντελεστές μηδενίζονται. Είναι ξεκάθαρο όμως, πως οι μεταβλητές X1, X5, X7, X11 και X13 είναι οι πιο σημαντικές, αφού έχουν τους μεγαλύτερους συντελεστές και πλην της X5 επιβιώνουν και για μεγάλες τιμές λ.



Σχήμα 5: Τιμές των συντελεστών β, καθώς η παράμετρος λ της Lasso αυξάνεται.

Παρακάτω φαίνεται ο κώδικας για την παραγωγή του σχήματος 5:

```
# perform Lasso
lasso <- glmnet(data[, ..vnames], data$Y)
plot(lasso, label=T, xvar='lambda')
```

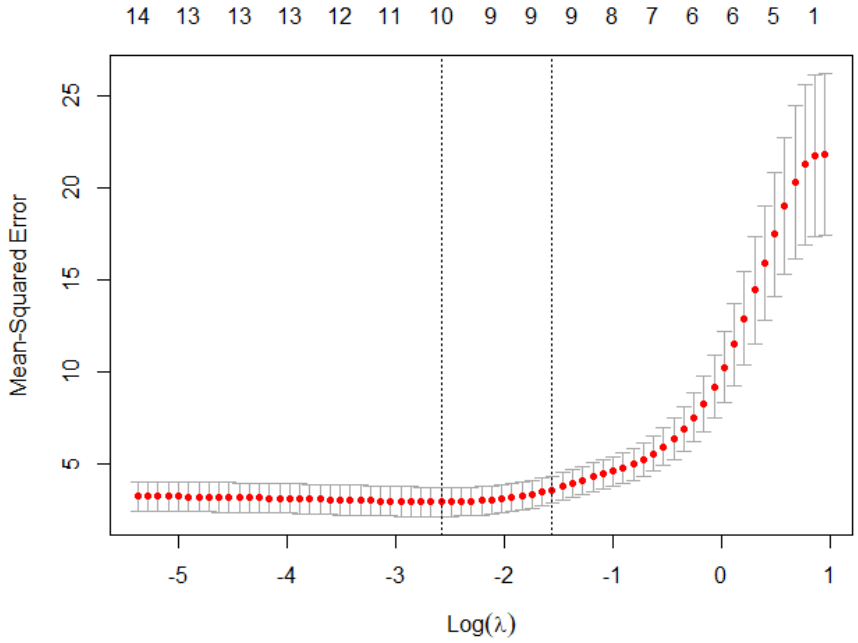
Τίθεται, λοιπόν, το ζήτημα της επιλογής της βέλτιστης τιμής για την παράμετρο λ. Για το σκοπό αυτό θα χρησιμοποιήσουμε 10-fold cross-validation, αξιολογώντας την προβλεπτική ικανότητα κάθε μοντέλου με τη μετρική Mean Squared Error (MSE). Στο σχήμα 6 βλέπουμε την τιμή του MSE καθώς μεγαλώνει το λ, η οποία αυξάνεται, αφού όλο και περισσότεροι συντελεστές μηδενίζονται. Με αυτή τη μέθοδο επιλέχθηκε η τιμή $\lambda = 0.075$ με $MSE = 2.886$ και το αντίστοιχο μοντέλο με 10 από τις 15 επεξηγηματικές μεταβλητές είναι το

$$Y = 3.794 + 2.151X_1 + 0.017X_2 + 0.055X_4 - 1.059X_5 + 2.559X_7 \\ - 0.247X_8 + 0.32X_9 - 0.058X_{10} + 1.365X_{11} + 0.608X_{13}$$

Παρακάτω φαίνεται ο αντίστοιχος κώδικας:

```
# perform cross-validation to find best lambda
lasso_cv <- cv.glmnet(as.matrix(data[, ..vnames]), data$Y, nfolds =
    10, type.measure='mse')
plot(lasso_cv)
lasso_cv

# get coefficients
blasso <- coef(lasso_cv, s="lambda.min")
blasso
```



Σχήμα 6: Cross-validated MSE, καθώς η παράμετρος λ της Lasso αυξάνεται.

Τέλος, υπολογίζουμε την παράμετρο συρρίκνωσης

$$s = \frac{\sum_{j=1}^{p} |\beta_j|}{max \sum_{j=1}^{p} |\beta_j|}, \; s \in [0, 1].$$

Όταν $s = 1$ σημαίνει πως δεν υπάρχει κανονικοποίηση και έτσι έχουμε την Ordinary Least Squares λύση, ενώ όταν $s = 0$ τότε ο βαθμός κανονικοποίησης είναι πολύ μεγάλος και όλοι οι συντελεστές είναι 0. Προκειμένου να υπολογίσουμε την παράμετρο s, κανονικοποιούμε τους συντελεστές πολλαπλασιάζοντάς τους με την τυπική απόκλιση των δεδομένων ανά στήλη, τόσο για το επιλεγμένο μοντέλο με Lasso όσο και για το πλήρες μοντέλο. Τελικά, η παράμετρος s που αντιστοιχεί στο μοντέλο που επιλέχθηκε είναι η

$$s = 0.846.$$

Η τιμή είναι μεγάλη, κάτι το οποίο σημαίνει πως ο βαθμός κανονικοποίησης είναι σχετικά μικρός. Παρακάτω φαίνεται ο αντίστοιχος κώδικας:

```
# full model
mfull <- lm(paste('Y ~', paste(vnames, collapse="+")), data = data)

# standardize
zblasso <-blasso[-1] * apply(as.matrix(data[, ..vnames]), 2, sd)
zbols <- coef(mfull)[-1] * apply(as.matrix(data[, ..vnames]), 2, sd
    )

```

```r
# calculate shrinkage factor
s <- sum(abs(zblasso)) / sum(abs(zbols))
s
```