ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Δ.Π.Μ.Σ. ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Εξαμηνιαία Εργασία Μαθήματος

"Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση"

Αντώνιος Μπαροτσάκης
Αριθμός Μητρώου : 03400260
Email: : adoniosbarotsakis@mail.ntua.gr
Μεταπτυχιακός Φοιτητής Ε.ΔΕ.ΜΜ

Ιούνιος 2025

Περιεχόμενα

Κατάλογος σχημάτων

0 Εισαγωγή

Η παρούσα εργασία εκπονήθηκε στα πλαίσια του μαθήματος "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σκοπός της εργασίας είναι η εφαρμογή και η διερεύνηση διαφόρων υπολογιστικών και στοχαστικών μεθόδων που αποτελούν κεντρικά εργαλεία της σύγχρονης στατιστικής ανάλυσης και μοντελοποίησης.

Η εργασία διαρθρώνεται σε τέσσερις κύριες ασκήσεις. Η πρώτη άσκηση επικεντρώνεται στη μη παραμετρική παλινδρόμηση, με έμφαση στην μέθοδο Nadaraya-Watson και την επιλογή του βέλτιστου bandwidth μέσω cross-validation, καθώς και στη μελέτη της κατανομής μιας στατιστικής συνάρτησης με χρήση της μεθόδου Bootstrap, τόσο στην μη παραμετρική όσο και στην παραμετρική της εκδοχή. Η δεύτερη άσκηση διερευνά τεχνικές προσομοίωσης, συγκεκριμένα την "squeezed" μέθοδο της απόρριψης για την παραγωγή δειγμάτων από την τυποποιημένη κανονική κατανομή, και την εκτίμηση ολοκληρωμάτων μέσω κλασικού Monte Carlo και δειγματοληψίας σπουδαιότητας. Η τρίτη άσκηση ασχολείται με την εκτίμηση παραμέτρων σε ένα μοντέλο μίξης κατανομών, αξιοποιώντας τον αλγόριθμο Expectation-Maximization (EM). Τέλος, η τέταρτη άσκηση εστιάζει στο πρόβλημα της επιλογής μεταβλητών σε ένα μοντέλο πολλαπλής γραμμικής παλινδρόμησης για το σύνολο δεδομένων "Boston housing", συγκρίνοντας την πλήρη διερεύνηση του χώρου των μοντέλων με βάση το κριτήριο AIC, την εφαρμογή της μεθόδου LASSO, και την κατασκευή διαστημάτων εμπιστοσύνης με χρήση Residual Bootstrap.

Για την υλοποίηση των αλγορίθμων και την ανάλυση των δεδομένων χρησιμοποιήθηκε η γλώσσα προγραμματισμού `R`. Η θεωρητική θεμελίωση και οι κατευθυντήριες γραμμές για την επίλυση των ασκήσεων αντλήθηκαν σε μεγάλο βαθμό από το υλικό και τις διαλέξεις του μαθήματος, όπως αυτά παρατίθενται στις διαφάνειες που είναι διαθέσιμες στην επίσημη ιστοσελίδα του μαθήματος [1]. Μέσα από τις ασκήσεις αυτές, επιδιώκεται η πρακτική κατανόηση των μεθόδων, η κριτική αξιολόγηση των αποτελεσμάτων τους και η εξαγωγή χρήσιμων συμπερασμάτων.

1 Άσκηση 1

1.1 Ερώτημα (α): Μη Παραμετρική Παλινδρόμηση Nadaraya-Watson

Στο παρόν ερώτημα θα ασχοληθούμε με την εφαρμογή μεθόδων μη παραμετρικής παλινδρόμησης για την ανάλυση ενός συνόλου δεδομένων που αποτελείται από 200 ζεύγη παρατηρήσεων (x_i, y_i), $i = 1, \ldots, 200$, ενός τυχαίου δείγματος από δύο συνεχείς τυχαίες μεταβλητές X και Y. Στόχος μας είναι να εκτιμήσουμε την υποκείμενη σχέση $Y = m(X) + \epsilon$, όπου $m(X) = E[Y|X = x]$ είναι η συνάρτηση παλινδρόμησης και ϵ ο όρος σφάλματος.

Θα εστιάσουμε συγκεκριμένα στη μέθοδο Nadaraya-Watson, η οποία αποτελεί έναν δημοφιλή μη παραμετρικό εκτιμητή πυρήνα (kernel estimator) για τη συνάρτηση παλινδρόμησης. Ως συνάρτηση πυρήνα (kernel) θα χρησιμοποιήσουμε τον Gaussian πυρήνα. Ένα κρίσιμο βήμα στην εφαρμογή αυτής της μεθόδου είναι η επιλογή του βέλτιστου πλάτους h_x, το οποίο ελέγχει τον βαθμό εξομάλυνσης της εκτιμώμενης καμπύλης. Η επιλογή του h_x θα πραγματοποιηθεί μέσω της τεχνικής leave-one-out cross-validation, χρησιμοποιώντας ως κριτήριο την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος πρόβλεψης.

Αρχικά, θα οπτικοποιήσουμε τα δεδομένα και την εκτιμώμενη καμπύλη παλινδρόμησης που προκύπτει από τη μέθοδο Nadaraya-Watson με το βέλτιστο h_x. Στη συνέχεια, θα διερευνήσουμε έναν αποδοτικότερο τρόπο υλοποίησης της διαδικασίας leave-one-out cross-validation, αποφεύγοντας την επαναληπτική προσαρμογή του μοντέλου n φορές. Τέλος, θα εξετάσουμε θεωρητικά τη συμπεριφορά της εκτιμώμενης καμπύλης Nadaraya-Watson στις οριακές περιπτώσεις όπου το εύρος ζώνης h_x τείνει στο μηδέν και στο άπειρο.

Το πρώτο βήμα είναι να φορτώσουμε τα δεδομένα από το αρχείο `"data1pairs.rds"`:

```r
# Load the data
data_pairs <-readRDS("data1pairs.rds")

# Preview the first few rows of the data for confirmation
print(head(data_pairs))

# Check the column names
print(colnames(data_pairs))

# For convenience, let's assign X and Y to separate vectors
X <-data_pairs$X
Y <-data_pairs$Y
n <-length(X)
```

1.1.1 Υποερώτημα i: Επιλογή Βέλτιστου Bandwidth με LOOCV

Για την εκτίμηση της συνάρτησης παλινδρόμησης $m(x) = E[Y|X = x]$ από ένα δείγμα παρατηρήσεων (x_i, y_i), $i = 1, \ldots, n$, όταν δεν θέλουμε να κάνουμε ισχυρές παραδοχές για τη μορφή της $m(x)$, χρησιμοποιούμε μεθόδους μη παραμετρικής παλινδρόμησης. Μια τέτοια μέθοδος είναι ο εκτιμητής Nadaraya-Watson [2], ο οποίος εκτιμά την $m(x)$ ως έναν τοπικά σταθμισμένο μέσο όρο των y_i:

$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_x}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_x}\right)}$$

όπου $K(\cdot)$ είναι μια συνάρτηση πυρήνα (kernel function) και $h_x > 0$ είναι το πλάτος, το οποίο καθορίζει τον βαθμό εξομάλυνσης. Στην παρούσα άσκηση, θα χρησιμοποιήσουμε τον Gaussian πυρήνα, ο οποίος ορίζεται ως:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$

Η συνάρτηση `ksmooth` της R υλοποιεί τον εκτιμητή Nadaraya-Watson. Το όρισμα `kernel="normal"` αντιστοιχεί στον Gaussian πυρήνα, και το `bandwidth` στο h_x.

Η επιλογή του bandwidth h_x είναι κρίσιμη για την απόδοση του εκτιμητή Nadaraya-Watson. Μικρές τιμές του h_x μπορεί να οδηγήσουν σε υπερβολική προσαρμογή (overfitting) στα δεδομένα, με αποτέλεσμα μια καμπύλη με μεγάλη μεταβλητότητα, ενώ μεγάλες τιμές του h_x μπορεί να οδηγήσουν σε υπερβολική εξομάλυνση (underfitting), αποκρύπτοντας σημαντικά χαρακτηριστικά της σχέσης μεταξύ X και Y.

Για την επιλογή του βέλτιστου h_x, θα χρησιμοποιήσουμε τη μέθοδο leave-one-out cross-validation (LOOCV) [3]. Η διαδικασία περιλαμβάνει την αφαίρεση κάθε παρατήρησης (x_i, y_i) μία φορά από το δείγμα, την εκτίμηση της τιμής y_i χρησιμοποιώντας τα υπόλοιπα $n-1$ σημεία (ας την ονομάσουμε $\hat{m}_{h_x,-i}(x_i)$), και τον υπολογισμό του τετραγωνικού σφάλματος πρόβλεψης. Το κριτήριο που θα ελαχιστοποιήσουμε είναι το μέσο τετραγωνικό σφάλμα (Mean Squared Error) του LOOCV:

$$\mathrm{CV}(h_x) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{m}_{h_x,-i}(x_i))^2$$

Θα δοκιμάσουμε ένα εύρος πιθανών τιμών για το h_x και θα επιλέξουμε εκείνη που ελαχιστοποιεί το $\mathrm{CV}(h_x)$.

```r
# Define a sequence of candidate bandwidths (hx_values)
hx_values <-seq(0.1, 2.5, by =0.05)

# Initialize a vector to store the CV(MSE) for each hx
cv_mse <-numeric(length(hx_values))

# Loop through each candidate hx
for (j in 1:length(hx_values)) {
 h_current <-hx_values[j]
 squared_errors <-numeric(n) # To store squared errors for this h_current

 # LOOCV loop: for each data point
 for (i in 1:n) {
  # Data for training (all points except i)
  X_train <-X[-i]
  Y_train <-Y[-i]

  # Point to predict
  x_test_point <-X[i]

  # Use ksmooth to get the prediction
  ksmooth_fit_loocv <-ksmooth(x =X_train, y =Y_train, kernel ="normal", bandwidth =h_current, x.points
      =x_test_point)
  y_pred_loocv <-ksmooth_fit_loocv$y
  squared_errors[i] <-(Y[i] - y_pred_loocv)^2
  }

 # Calculate the mean squared error for the current hx
```

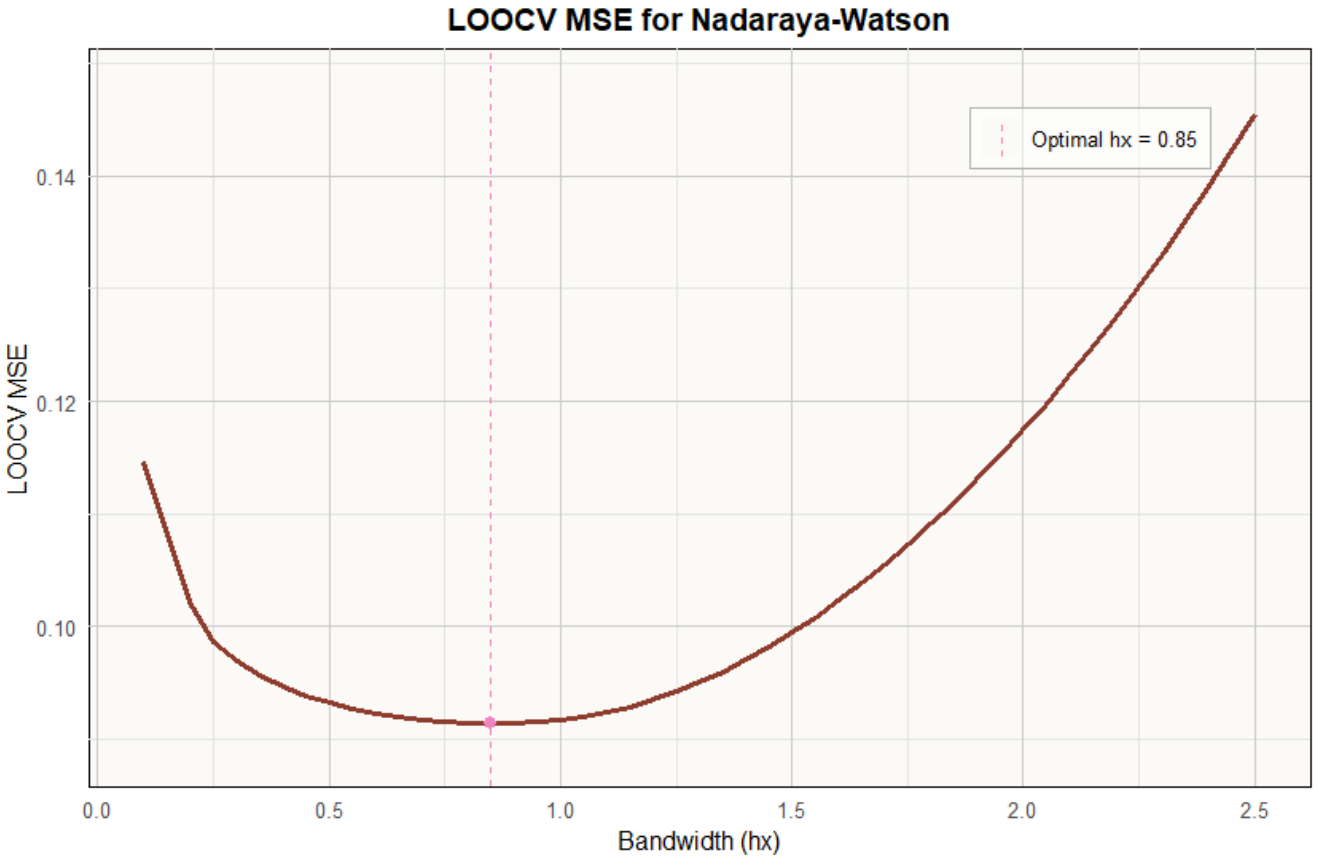
```
  cv_mse[j] <-mean(squared_errors, na.rm =TRUE)
}

# Find the optimal hx that minimizes CV(MSE)
optimal_hx_index <-which.min(cv_mse)
optimal_hx <-hx_values[optimal_hx_index]
min_cv_mse <-cv_mse[optimal_hx_index]
```

Δοκιμάστηκε ένα εύρος τιμών για το h_x από 0.1 έως 2.5 με βήμα 0.05. Στο Σχήμα 1 απεικονίζεται η σχέση μεταξύ του h_x και του αντίστοιχου LOOCV MSE.

```
# Create a data frame for ggplot
plot_data_gg <-data.frame(Bandwidth =hx_values,MSE =cv_mse)

# Calculate y-axis limits to make it "narrower"
# Focus on the range of observed finite MSE values with a little padding
min_mse_for_plot <-min(plot_data_gg$MSE, na.rm =TRUE)
max_mse_for_plot <-max(plot_data_gg$MSE, na.rm =TRUE)

# Dynamic padding: 5% of the range of MSE values shown
y_axis_padding <-(max_mse_for_plot - min_mse_for_plot) * 0.05

y_limit_lower <-min_mse_for_plot - y_axis_padding
y_limit_upper <-max_mse_for_plot + y_axis_padding


# Define the label for the optimal hx line in the legend
optimal_hx_legend_label <-paste("Optimal hx =", format(round(optimal_hx, 2), nsmall =2))

# Create the ggplot
loocv_gg_plot <-ggplot(plot_data_gg, aes(x =Bandwidth, y =MSE)) +
  # Panel background and grid
  theme_minimal(base_size =12) +
  theme(
    plot.title =element_text(hjust =0.5, face ="bold", size =14),
    panel.background =element_rect(fill ="#FBFAF7"),
    plot.background =element_rect(fill ="white", colour =NA),
    panel.grid.major =element_line(colour ="grey80", linewidth =0.4),
    panel.grid.minor =element_line(colour ="grey90", linewidth =0.2),
    legend.position =c(0.82, 0.88),
    legend.background =element_rect(fill =alpha("white", 0.5), colour ="grey70", linewidth=0.5),
    legend.key =element_rect(fill ="#FBFAF7", colour =NA),
    legend.title =element_blank()
  ) +
  # MSE Curve
  geom_line(color ="#8E3C2E", linewidth =1.2) +
  # Vertical line for optimal hx
  geom_vline(aes(xintercept =optimal_hx, linetype ="Optimal"), color ="#F084C1", linewidth =0.7) +
  # Point marking the optimal hx and minimum MSE
  geom_point(aes(x =optimal_hx, y =min_cv_mse), color ="#F084C1", size =2.5, shape =19) +
  # Define the linetype for the legend
  scale_linetype_manual(name ="", values =c("Optimal" ="dashed"),
    labels =c("Optimal" =optimal_hx_legend_label)) +
  # Set y-axis limits for a "narrower" view & x-axis expansion
```

```
coord_cartesian(ylim =c(y_limit_lower, y_limit_upper),
  xlim =range(plot_data_gg$Bandwidth, na.rm =TRUE),
  expand =TRUE
) +
# Labels
labs(title ="LOOCV MSE for Nadaraya-Watson", x ="Bandwidth (hx)", y ="LOOCV MSE")
```

Σχήμα 1: Σχέση του bandwidth h_x με το LOOCV MSE για την Μέθοδο Nadaraya-Watson.



Από τη διαδικασία LOOCV, το βέλτιστο bandwidth που ελαχιστοποιεί το MSE βρέθηκε να είναι $h_x^{opt} = 0.85$, με ελάχιστο LOOCV MSE ίσο με 0.0914.

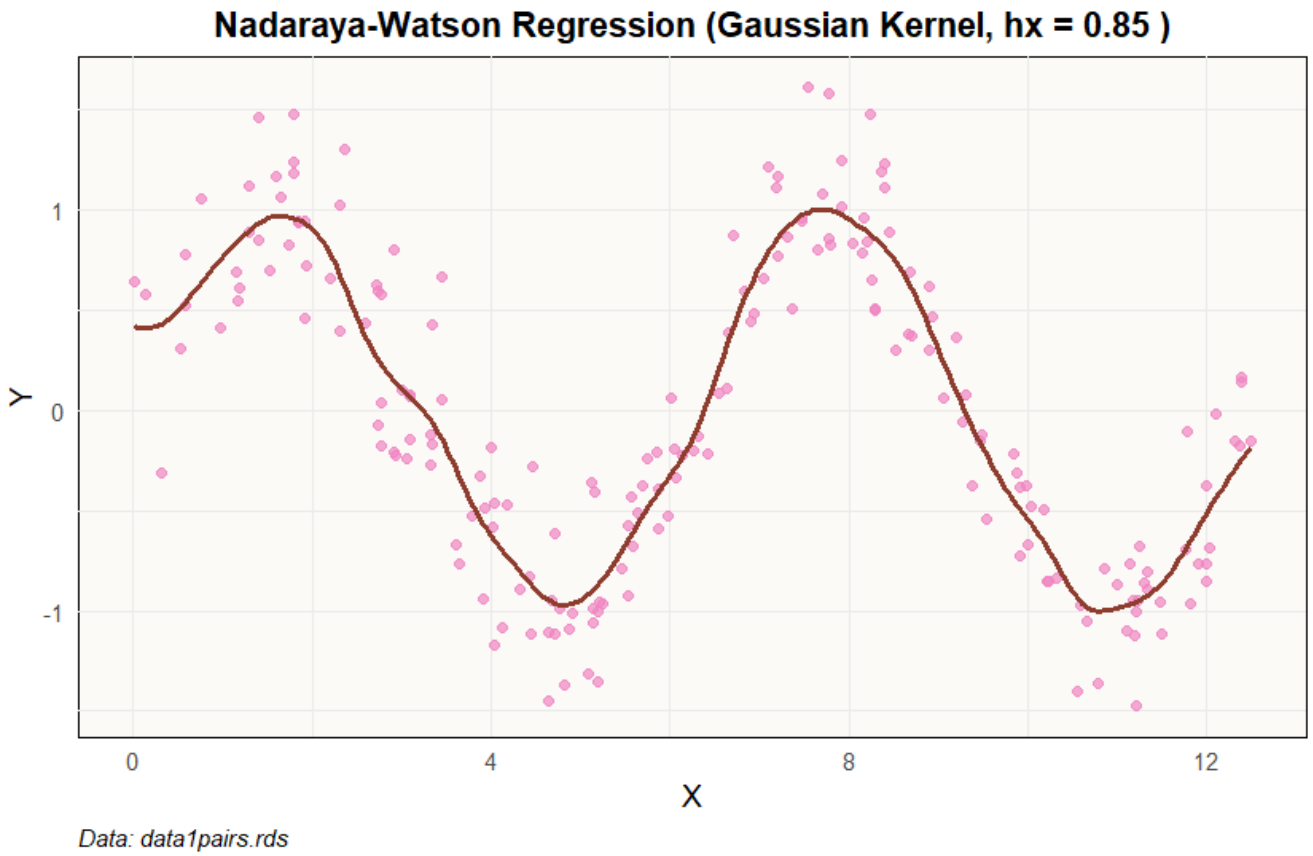
Χρησιμοποιώντας αυτό το βέλτιστο bandwidth, θα υπολογίσουμε την τελική εκτίμηση της καμπύλης παλινδρόμησης $\hat{m}_{h_x^{opt}}(x)$ μέσω του εκτιμητή Nadaraya-Watson, χρησιμοποιώντας το σύνολο των $n = 200$ παρατηρήσεων.

```
# Generate a sequence of x values for plotting the smooth curve
x_grid <-seq(min(X), max(X), length.out =200)

# Get the Nadaraya-Watson estimate using the optimal hx
nw_estimate_optimal <-ksmooth(x =X, y =Y, kernel ="normal", bandwidth =optimal_hx, x.points =x_grid)
```

Για την οπτικοποίηση των αποτελεσμάτων, δημιουργήθηκε το Σχήμα 2 με το πακέτο `ggplot2`, το οποίο παρουσιάζει το διάγραμμα διασποράς των αρχικών παρατηρήσεων (x_i, y_i) μαζί με την εκτιμώμενη καμπύλη Nadaraya-Watson που προέκυψε με πλάτος $h_x = 0.85$.

```r
# Prepare data for ggplot
# Original data
plot_data_scatter <-data.frame(X_orig =X, Y_orig =Y)
# Smoothed curve data
plot_data_line <-data.frame(X_smooth =nw_estimate_optimal$x, Y_smooth =nw_estimate_optimal$y)

# Create the ggplot
final_plot <-ggplot() +
  # Scatter plot of the original data
  geom_point(data=plot_data_scatter, aes(x =X_orig, y =Y_orig), color ="#F084C1", alpha =0.7, size =2) +
  # Add the Nadaraya-Watson smoothed line
  geom_line(data =plot_data_line, aes(x =X_smooth, y =Y_smooth), color ="#8E3C2E", linewidth =1.2) +
  # Add titles and labels
  labs(title =paste("Nadaraya-Watson Regression (Gaussian Kernel, hx =", round(optimal_hx, 2), ")"),
       x ="X", y ="Y", caption ="Data: data1pairs.rds") +
  # Apply a theme
  theme_minimal(base_size =14) +
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
        plot.caption =element_text(hjust =0, face ="italic"),
        panel.background =element_rect(fill ="#FBFAF7"),)
```

Σχήμα 2: Εκτιμώμενη καμπύλη Nadaraya-Watson με $h_x = 0.85$.



Παρατηρούμε ότι η εκτιμώμενη καμπύλη φαίνεται να συλλαμβάνει επιτυχώς την υποκείμενη κυματοειδή (περιοδική) δομή των δεδομένων. Η καμπύλη διέρχεται ομαλά ανάμεσα από τα σημεία, υποδεικνύοντας ότι το επιλεγμένο bandwidth

$h_x = 0.85$ παρέχει μια καλή ισορροπία μεταξύ της προσαρμογής στα δεδομένα (αποφυγή underfitting) και της ομαλότητας της εκτίμησης (αποφυγή overfitting). Δεν παρατηρούνται έντονες περιοχές όπου η καμπύλη είτε αγνοεί εμφανείς τάσεις των δεδομένων είτε ακολουθεί υπερβολικά τον θόρυβο.

Ο Gaussian πυρήνας, ο οποίος δίνει βάρος σε όλες τις παρατηρήσεις (αν και μειώνεται εκθετικά με την απόσταση), συμβάλλει στη δημιουργία μιας συνεχούς και διαφορίσιμης εκτιμώμενης καμπύλης, όπως φαίνεται και στο διάγραμμα. Η ομαλότητα αυτή είναι ένα επιθυμητό χαρακτηριστικό για πολλές εφαρμογές.

Συμπερασματικά, η μέθοδος Nadaraya-Watson, σε συνδυασμό με την αντικειμενική επιλογή του bandwidth μέσω LOOCV, αποδείχθηκε κατάλληλη για την περιγραφή της μη γραμμικής σχέσης που υπάρχει στο συγκεκριμένο σύνολο δεδομένων, παρέχοντας μια οπτικά εύλογη και στατιστικά θεμελιωμένη εκτίμηση της συνάρτησης παλινδρόμησης.

1.1.2 Υποερώτημα ii: Αποδοτική Υλοποίηση και Σύγκριση του LOOCV

Στο προηγούμενο ερώτημα, για τον υπολογισμό του LOOCV MSE, αφαιρούσαμε κάθε παρατήρηση (x_i, y_i) και εκτελούσαμε τη συνάρτηση `ksmooth` στα υπόλοιπα $n-1$ σημεία για να προβλέψουμε το y_i. Αυτή η διαδικασία, ενώ είναι εννοιολογικά απλή, περιλαμβάνει την εκτέλεση της `ksmooth` n φορές για κάθε υποψήφια τιμή του h_x. Για μεγάλα σύνολα δεδομένων, αυτό μπορεί να είναι υπολογιστικά δαπανηρό.

Στην περίπτωση της γραμμικής παλινδρόμησης, υπάρχει ένας γνωστός τύπος [3] που επιτρέπει τον υπολογισμό των προβλέψεων LOOCV $\hat{y}_{-i}$ χωρίς την επαναπροσαρμογή του μοντέλου n φορές, χρησιμοποιώντας το hat matrix A:

$$y_i - \hat{y}_{-i} = \frac{y_i - \hat{y}_i}{1 - \alpha_{ii}}$$

όπου $\hat{y}_i$ είναι η πρόβλεψη για το y_i από το μοντέλο που έχει προσαρμοστεί σε όλα τα δεδομένα, και α_{ii} είναι το i-οστό διαγώνιο στοιχείο του hat matrix.

Για τον μη παραμετρικό εκτιμητή Nadaraya-Watson, η κατάσταση είναι λίγο διαφορετική, αλλά η ιδέα της αναζήτησης ενός τρόπου υπολογισμού του $\hat{m}_{h_x,-i}(x_i)$ χωρίς πλήρη επαναπροσαρμογή είναι παρόμοια. Συγκεκριμένα, ο εκτιμητής Nadaraya-Watson για την πρόβλεψη του y_i αφήνοντας έξω την παρατήρηση i είναι:

$$\hat{m}_{h_x,-i}(x_i) = \frac{\sum_{j=1, j\neq i}^{n} K\left(\frac{x_i - x_j}{h_x}\right) y_j}{\sum_{j=1, j\neq i}^{n} K\left(\frac{x_i - x_j}{h_x}\right)}$$

Μπορούμε να υπολογίσουμε αυτόν τον τύπο απευθείας. Εάν ορίσουμε $w_{ij}(h_x) = K\left(\frac{x_i - x_j}{h_x}\right)$ ως τα βάρη, τότε:

$$\hat{m}_{h_x}(x_i) = \frac{\sum_{j=1}^{n} w_{ij}(h_x) y_j}{\sum_{j=1}^{n} w_{ij}(h_x)}$$

και

$$\hat{m}_{h_x,-i}(x_i) = \frac{\sum_{j=1}^{n} w_{ij}(h_x) y_j - w_{ii}(h_x) y_i}{\sum_{j=1}^{n} w_{ij}(h_x) - w_{ii}(h_x)} = \frac{S_i(h_x) - w_{ii}(h_x) y_i}{D_i(h_x) - w_{ii}(h_x)}$$

όπου $S_i(h_x) = \sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h_x}\right) y_j$ είναι ο αριθμητής του $\hat{m}_{h_x}(x_i)$ και $D_i(h_x) = \sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h_x}\right)$ είναι ο παρονομαστής του $\hat{m}_{h_x}(x_i)$. Το $w_{ii}(h_x) = K(0)$ είναι το βάρος που δίνει η παρατήρηση i στον εαυτό της. Αυτός ο τρόπος είναι πιο αποδοτικός γιατί οι $S_i(h_x)$ και $D_i(h_x)$ μπορούν να υπολογιστούν μία φορά για κάθε x_i (για δεδομένο h_x) και μετά να γίνουν απλές αφαιρέσεις.

```r
# Gaussian kernel function
gaussian_kernel <-function(u) {
  (1 / sqrt(2 * pi)) * exp(-0.5 * u^2)
}

# Initialize a vector to store the CV(MSE) for each hx using the efficient method
cv_mse_efficient <-numeric(length(hx_values))

# Value of K(0) for Gaussian kernel
K_0 <-gaussian_kernel(0)
```

Θα δημιουργήσουμε έναν βρόχο που θα διατρέχει όλες τις υποψήφιες τιμές του h_x. Μέσα σε αυτόν τον βρόχο, για κάθε σημείο x_i, θα υπολογίσουμε πρώτα τους όρους $S_i(h_x)$ και $D_i(h_x)$ που περιλαμβάνουν όλες τις παρατηρήσεις. Στη συνέχεια, για την πρόβλεψη $\hat{m}_{h_x,-i}(x_i)$, θα αφαιρέσουμε τη συνεισφορά της ίδιας της παρατήρησης i (δηλαδή τον όρο $K(0)y_i$ από τον αριθμητή και τον όρο $K(0)$ από τον παρονομαστή).

```r
# Gaussian kernel function
gaussian_kernel <-function(u) {
  (1 / sqrt(2 * pi)) * exp(-0.5 * u^2)
}

# Initialize a vector to store the CV(MSE) for each hx using the efficient method
cv_mse_efficient <-numeric(length(hx_values))

# Value of K(0) for Gaussian kernel
K_0 <-gaussian_kernel(0)

# Loop through each candidate hx
for (j in 1:length(hx_values)) {
 h_current <-hx_values[j]
 squared_errors_efficient <-numeric(n) # To store squared errors for this h_current

 # For each data point i (the one to be left out)
 for (i in 1:n) {
  # Calculate weights w_ij =K((x_i - x_j)/h_current) for all j
  u_values <-(X[i] - X) / h_current # Vectorized calculation for (x_i - x_j)/h
  weights_ij <-gaussian_kernel(u_values)

  # Sum of weighted Ys (numerator S_i(h_x) for m_hat(x_i))
  S_i_current_h <-sum(weights_ij * Y)

  # Sum of weights (denominator D_i(h_x) for m_hat(x_i))
  D_i_current_h <-sum(weights_ij)

  # Numerator for LOOCV prediction: S_i(h_x) - w_ii(h_x)*y_i
  # w_ii is K((x_i - x_i)/h_current) =K(0)
  numerator_loocv <-S_i_current_h - (K_0 * Y[i])

  # Denominator for LOOCV prediction: D_i(h_x) - w_ii(h_x)
  denominator_loocv <-D_i_current_h - K_0

  # Calculate the LOOCV prediction for y_i
  y_pred_loocv_efficient <-numerator_loocv / denominator_loocv
```

```
    # Calculate the Squared Errors
    squared_errors_efficient[i] <-(Y[i] - y_pred_loocv_efficient)^2
  }

  # Calculate the mean squared error for the current hx
  cv_mse_efficient[j] <-mean(squared_errors_efficient, na.rm =TRUE)
}

# Find the optimal hx that minimizes CV(MSE) using the efficient method
optimal_hx_index_efficient <-which.min(cv_mse_efficient)
optimal_hx_efficient <-hx_values[optimal_hx_index_efficient]
```

Εκτελώντας τον κώδικα για το ίδιο εύρος υποψήφιων τιμών h_x (από 0.1 έως 2.5 με βήμα 0.05) όπως και στο προηγούμενο υποερώτημα. Το βέλτιστο bandwidth που ελαχιστοποιεί το MSE βρέθηκε να ειναι $h_x^{\text{opt}} = 0.3$, με ελάχιστο CV MSE ίσο με 0.09143024.

Αυτό τα αποτέλεσμα παρουσιάζει μια αξιοσημείωτη διαφορά σε σχέση με την προσέγγιση του προηγούμενου υποερωτήματος, όπου χρησιμοποιήθηκε η έτοιμη συνάρτηση `ksmooth` της R εντός του βρόχου LOOCV. Παρόλο που οι δύο μεθοδολογίες ειναι μαθηματικά ισοδύναμες. Συγκεκριμένα, το βέλτιστο εύρος ζώνης είχε προσδιοριστεί σε $h_x = 0.85$ με ελάχιστο CV(MSE) ίσο με 0.09139248.

Ενώ οι τιμές του ελάχιστου MSE είναι εξαιρετικά κοντινές μεταξύ των δύο προσεγγίσεων (διαφορά της τάξης του 10^{-5}), η βέλτιστη τιμή του h_x που προκύπτει είναι σημαντικά διαφορετική (0.3 έναντι 0.85). Αυτή η απόκλιση υποδηλώνει ότι η έτοιμη συνάρτηση `ksmooth`, όταν καλείται για την πρόβλεψη ενός μόνο σημείου εντός ενός LOOCV βρόχου, ενδέχεται να εφαρμόζει μια ελαφρώς διαφορετική λογική ή να περιλαμβάνει επιπλέον βήματα ομαλοποίησης/διόρθωσης σε σύγκριση με την άμεση, "καθαρή" μαθηματική υλοποίηση του τύπου LOOCV που εφαρμόστηκε εδώ.

Για να διερευνήσουμε περαιτέρω αυτή την παρατήρηση και να επιβεβαιώσουμε την ορθότητα της προσέγγισής μας για τον υπολογισμό του $\hat{m}_{h_x,-i}(x_i)$, προχωρήσαμε στην υλοποίηση και μιας "χειροκίνητης απλοϊκής" (manual naive) μεθόδου LOOCV. Σε αυτή την προσέγγιση, για κάθε παρατήρηση i που αφαιρείται, ο εκτιμητής Nadaraya-Watson $\hat{m}_{h_x,-i}(x_i)$ υπολογίζεται από την αρχή χρησιμοποιώντας μόνο τις υπόλοιπες $n-1$ παρατηρήσεις, χωρίς να αξιοποιείται ο αποδοτικός τύπος που βασίζεται στην αφαίρεση της συνεισφοράς του $K(0)$. Μαθηματικά, αυτή η "χειροκίνητη απλοϊκή" προσέγγιση θα πρέπει να είναι ισοδύναμη με την "αποδοτική" μέθοδο που περιγράφηκε προηγουμένως για το παρών υποερώτημα, καθώς και οι δύο στοχεύουν στον υπολογισμό της ίδιας ποσότητας $\hat{m}_{h_x,-i}(x_i)$.

```
# --- Manual Nadaraya-Watson LOOCV (Naive Approach) ---

# Gaussian kernel function K(u) - the same as used in the efficient method attempt
gaussian_kernel <-function(u) {
  (1 / sqrt(2 * pi)) * exp(-0.5 * u^2)
}

# Initialize a vector to store the CV(MSE) for each hx
cv_mse_manual_naive <-numeric(length(hx_values))

# Loop through each candidate hx
for (j in 1:length(hx_values)) {
  h_current <-hx_values[j]
  squared_errors_for_h <-numeric(n) # Store squared errors for the current h_current

  # LOOCV loop: iterate through each data point i to use as the test point
  for (i in 1:n) {
    # Training data (all points except i)
```

```
  X_train <-X[-i]
  Y_train <-Y[-i]

  # Test point (the one left out)
  x_test_point <-X[i]

  # --- Manual Nadaraya-Watson calculation for x_test_point using (X_train, Y_train) ---
  # Calculate u_kj =(x_test_point - x_k_train) / h_current for all k in training set
  u_values_train <-(x_test_point - X_train) / h_current

  # These are the weights for the training points
  kernel_weights_train <-gaussian_kernel(u_values_train)

  # Calculate the Nadaraya-Watson estimate
  # m_hat_(-i)(x_i) =sum [ kernel_weights_train_k * Y_train_k ] / sum [ kernel_weights_train_k ]
  sum_weighted_Y_train <-sum(kernel_weights_train * Y_train)
  sum_kernel_weights_train <-sum(kernel_weights_train)
  y_pred_loocv_manual_naive <-sum_weighted_Y_train / sum_kernel_weights_train

  # Calculate squared error
  squared_errors_for_h[i] <-(Y[i] - y_pred_loocv_manual_naive)^2
 }

 # Calculate the mean squared error for the current hx
 cv_mse_manual_naive[j] <-mean(squared_errors_for_h, na.rm =TRUE)
}

# Find the optimal hx that minimizes CV(MSE)
optimal_hx_index_manual_naive <-which.min(cv_mse_manual_naive)
optimal_hx_manual_naive <-hx_values[optimal_hx_index_manual_naive]
min_mse_manual_naive <-cv_mse_manual_naive[optimal_hx_index_manual_naive]
```

Πράγματι, η εκτέλεση της "χειροκίνητης απλοϊκής" LOOCV οδήγησε στα ίδια ακριβώς αποτελέσματα με την "αποδοτική" μέθοδο: το βέλτιστο εύρος ζώνης βρέθηκε $h_x^{opt} = 0.3$ με ελάχιστο CV MSE ίσο με 0.09143024. Αυτή η ταύτιση επιβεβαιώνει ότι οι δύο χειροκίνητες υλοποιήσεις του Nadaraya-Watson LOOCV είναι συνεπείς μεταξύ τους και ότι η απόκλιση παρατηρείται μόνο σε σύγκριση με την επαναληπτική χρήση της έτοιμης συνάρτησης `ksmooth`.

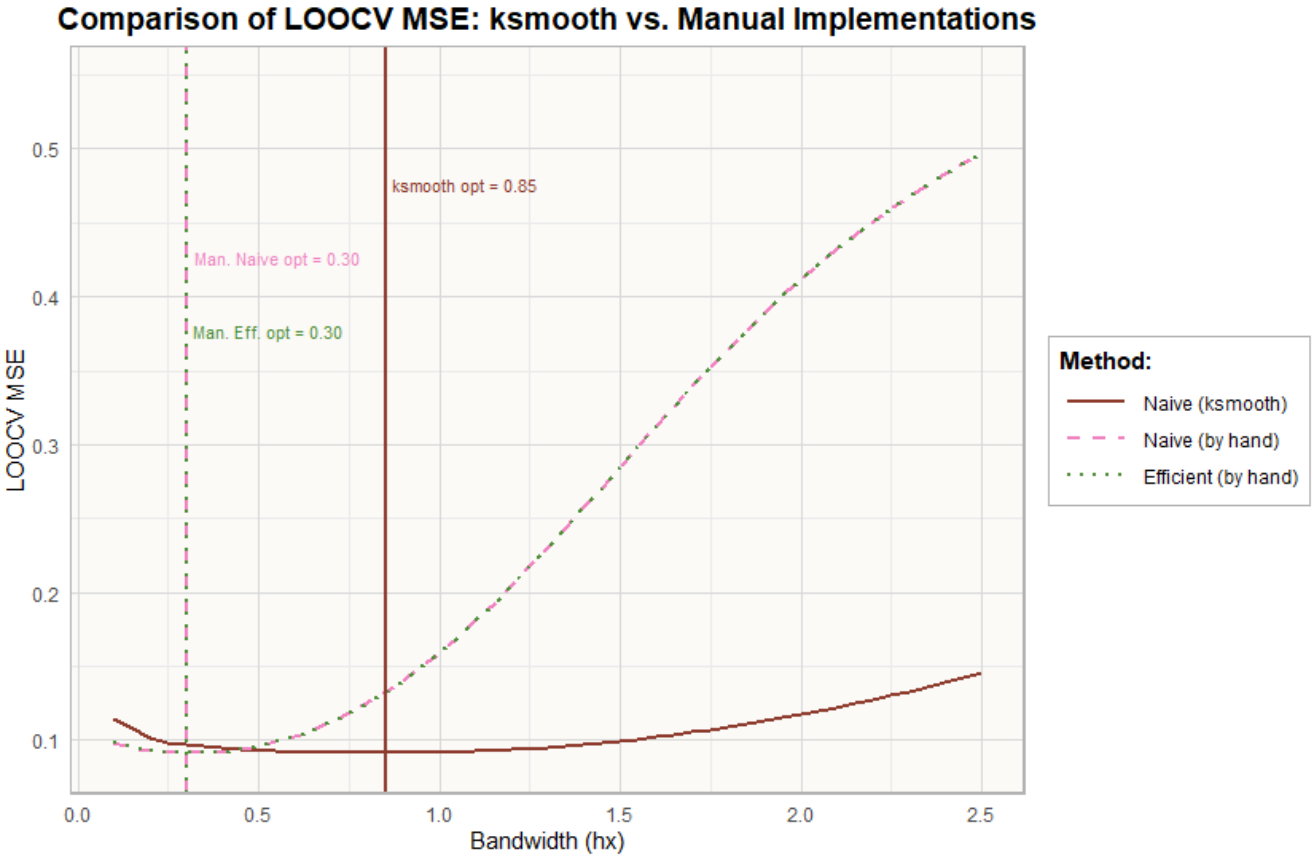
Για μια πληρέστερη οπτική σύγκριση, παραθέτουμε στο Σχήμα 3 τις καμπύλες του LOOCV MSE ως συνάρτηση του εύρους ζώνης h_x και για τις τρεις υλοποιήσεις: (1) την αρχική απλοϊκή προσέγγιση με χρήση της `ksmooth` (από το προηγούμενο υποερώτημα), (2) την αποδοτική υλοποίηση του LOOCV (από το παρών υποερώτημα), και (3) την χειροκίνητη απλοϊκή υλοποίηση του LOOCV.

```
# Naive (ksmooth) values
cv_mse_naive <-cv_mse
optimal_hx_naive <-optimal_hx

# Prepare Data for Combined Plot
plot_comparison_df_all3 <-data.frame(
 Bandwidth =rep(hx_values, 3),
 MSE =c(cv_mse_naive, cv_mse_efficient, cv_mse_manual_naive),
 Method =factor(rep(c("Naive (ksmooth)", "Efficient (by hand)", "Naive (by hand)"),
    each =length(hx_values)), levels =c("Naive (ksmooth)", "Naive (by hand)", "Efficient (by hand)"))
)
```

```r
# Define Customizable Colors and Linetypes
color_palette <-c("Naive (ksmooth)"= "#8E3C2E", "Naive (by hand)"= "#F084C1",
   "Efficient (by hand)"= "#4F8E38")

vline_colors <-color_palette

# Line types
linetype_palette <-c("Naive (ksmooth)"= "solid", "Naive (by hand)"= "dashed",
   "Efficient (by hand)"= "dotted")

# Optimal hx values
optimal_values <-list("Naive (ksmooth)" =optimal_hx_naive, "Naive (by hand)" =optimal_hx_manual_naive,
  "Efficient (by hand)" =optimal_hx_efficient)

# Create the ggplot
mse_comparison_plot_all3 <-ggplot(plot_comparison_df_all3, aes(x =Bandwidth, y =MSE, color =Method,
    linetype =Method)) +

  # MSE lines
  geom_line(linewidth =1) +

  # Vertical lines for optimal hx for each method
  geom_vline(aes(xintercept =optimal_values[["Naive (ksmooth)"]], color ="Naive (ksmooth)",
    linetype ="Naive (ksmooth)"), linewidth =0.8, show.legend =FALSE) +
  geom_vline(aes(xintercept =optimal_values[["Naive (by hand)"]], color ="Naive (by hand)",
    linetype ="Naive (by hand)"), linewidth =0.8, show.legend =FALSE) +
  geom_vline(aes(xintercept =optimal_values[["Efficient (by hand)"]], color ="Efficient (by hand)",
    linetype ="Efficient (by hand)"), linewidth =0.8, show.legend =FALSE) +

  # Annotations for optimal hx values
  annotate("text", x =optimal_values[["Naive (ksmooth)"]],
         y =max(plot_comparison_df_all3$MSE, na.rm =TRUE) * 0.95,
         label =paste("ksmooth opt=", format(round(optimal_values[["Naive (ksmooth)"]],2), nsmall=2)),
         color =vline_colors[["Naive (ksmooth)"]], hjust =-0.05, vjust =0, size =3.0) +
  annotate("text", x =optimal_values[["Naive (by hand)"]],
         y =max(plot_comparison_df_all3$MSE, na.rm =TRUE) * 0.85,
         label =paste("Man. Naive opt=", format(round(optimal_values[["Naive (by hand)"]],2), nsmall=2)),
         color =vline_colors[["Naive (by hand)"]], hjust =-0.05, vjust =0, size =3.0) +
  annotate("text", x =optimal_values[["Efficient (by hand)"]],
         y =max(plot_comparison_df_all3$MSE, na.rm =TRUE) * 0.75,
         label =paste("Man. Eff. opt=", format(round(optimal_values[["Efficient (by hand)"]],2), nsmall=
             2)),
         color =vline_colors[["Efficient (by hand)"]], hjust =-0.05, vjust =0, size =3.0) +

  # Scales for color and linetype
  scale_color_manual(values =color_palette, name ="Method:") +
  scale_linetype_manual(values =linetype_palette, name ="Method:") +

  # Y-axis and X-axis limits
  coord_cartesian(
    ylim =c(min(plot_comparison_df_all3$MSE, na.rm =TRUE) * 0.95,
          max(plot_comparison_df_all3$MSE, na.rm =TRUE) * 1.1),
    xlim =range(plot_comparison_df_all3$Bandwidth, na.rm =TRUE), expand =TRUE) +

  labs(
    title ="Comparison of LOOCV MSE: ksmooth vs. Manual Implementations",
```

```
  x ="Bandwidth (hx)", y ="LOOCV MSE") +
theme_minimal(base_size =11) +
theme(
  plot.title =element_text(hjust =0.5, face ="bold", size =14),
  panel.background =element_rect(fill ="#FBFAF7", colour =NA),
  panel.border =element_rect(colour ="grey70", fill =NA, linewidth =1),
  panel.grid.major =element_line(colour ="grey85", linewidth =0.3),
  panel.grid.minor =element_line(colour ="grey92", linewidth =0.15),
  legend.position ="right",
  legend.background =element_rect(fill =alpha("white", 0.8), colour ="grey70", linewidth =0.5),
  legend.title =element_text(face ="bold"),
  legend.key.width =unit(1.2, "cm"))
```

Σχήμα 3: Σύγκριση Καμπυλών LOOCV MSE ως συνάρτηση του bandwidth : `ksmooth` vs Χειροκίνητες Υλοποιήσεις



Όπως αναμενόταν, οι δύο χειροκίνητες υλοποιήσεις του Nadaraya-Watson LOOCV ("Naive (by hand)" και "Efficient (by hand)") παράγουν ταυτόσημες καμπύλες MSE, οι οποίες συμπίπτουν οπτικά στο σχήμα, επιβεβαιώνοντας τη μαθηματική τους ισοδυναμία και την ορθότητα του αποδοτικού τύπου. Και οι δύο αυτές χειροκίνητες προσεγγίσεις υποδεικνύουν ως βέλτιστο bandwidth $h_x = 0.3$.

Αντιθέτως, η καμπύλη MSE που προκύπτει από την επαναληπτική χρήση της έτοιμης συνάρτησης `ksmooth` ("Naive (`ksmooth`)") είναι ορατά διαφορετική, ιδιαίτερα για τιμές $h_x > 0.5$ όπου εμφανίζεται σημαντικά χαμηλότερη, και οδηγεί σε διαφορετικό βέλτιστο bandwidth $h_x = 0.85$. Αυτό ενισχύει την υπόθεση ότι η `ksmooth` ενσωματώνει επιπλέον λογική ή διαφορετικό χειρισμό οριακών συνθηκών σε σχέση με την άμεση εφαρμογή του βασικού τύπου Nadaraya-Watson για LOOCV.

1.1.3 Υποερώτημα iii: Ανάλυση Οριακών Περιπτώσεων ($h_x \to 0$ και $h_x \to \infty$)

Η συμπεριφορά της εκτιμώμενης καμπύλης $\hat{m}_{h_x}(x)$ από τη μέθοδο Nadaraya-Watson εξαρτάται κρίσιμα από την τιμή του bandwidth h_x. Εξετάζουμε τις δύο οριακές περιπτώσεις:

- **Όταν $h_x \to 0$ (Bandwidth Τείνει στο Μηδέν)**: Καθώς το h_x γίνεται πολύ μικρό, ο πυρήνας $K\left(\frac{x-x_i}{h_x}\right)$ γίνεται εξαιρετικά "στενός" και συγκεντρωμένος γύρω από το x_i. Αυτό σημαίνει ότι για την εκτίμηση $\hat{m}_{h_x}(x)$ σε ένα σημείο x, ουσιαστικά λαμβάνονται υπόψη μόνο οι παρατηρήσεις x_i που βρίσκονται πολύ κοντά (ή ταυτίζονται με) το x.

  Η εκτιμώμενη καμπύλη τείνει να παρεμβάλει τα δεδομένα. Δηλαδή, για κάθε x που είναι ένα από τα παρατηρούμενα x_i, το $\hat{m}_{h_x}(x_i)$ θα τείνει στο αντίστοιχο y_i. Μεταξύ των παρατηρούμενων σημείων, η καμπύλη μπορεί να παρουσιάζει έντονες διακυμάνσεις ή και ασυνέχειες (αν ο πυρήνας έχει πεπερασμένο υποστήριγμα και δεν υπάρχουν παρατηρήσεις εντός του "παραθύρου"). Η καμπύλη γίνεται πολύ "τραχιά" και ακολουθεί τον θόρυβο των δεδομένων.

  Αυτό οδηγεί σε υψηλή μεταβλητότητα (high variance), χαμηλή μεροληψία (low bias) τοπικά στα σημεία των δεδομένων, αλλά μπορεί να είναι πολύ κακή στην πρόβλεψη νέων σημείων. Αυτό αντιστοιχεί σε **overfitting**.

- **Όταν $h_x \to \infty$ (Bandwidth Τείνει στο Άπειρο)**: Καθώς το h_x γίνεται πολύ μεγάλο, το όρισμα του πυρήνα $\frac{x-x_i}{h_x}$ τείνει στο μηδέν για όλα τα x_i. Έτσι, η τιμή του πυρήνα $K\left(\frac{x-x_i}{h_x}\right)$ τείνει στην τιμή $K(0)$ για όλες τις παρατηρήσεις i.

  Ο εκτιμητής Nadaraya-Watson γίνεται:

$$\hat{m}_{h_x}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_x}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_x}\right)} \approx \frac{\sum_{i=1}^{n} K(0) y_i}{\sum_{i=1}^{n} K(0)} = \frac{K(0)\sum_{i=1}^{n} y_i}{nK(0)} = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}$$

  Η εκτιμώμενη καμπύλη τείνει να γίνει μια οριζόντια γραμμή στην τιμή του δειγματικού μέσου όρου των y_i, δηλαδή $\bar{y}$. Η καμπύλη γίνεται εξαιρετικά ομαλή και χάνει κάθε τοπική πληροφορία ή δομή των δεδομένων.

  Αυτό οδηγεί σε χαμηλή μεταβλητότητα (low variance), υψηλή μεροληψία (high bias), καθώς η εκτίμηση δεν προσαρμόζεται καθόλου στις τοπικές αλλαγές των δεδομένων. Αυτό αντιστοιχεί σε **underfitting**.

Συνοπτικά, η επιλογή του h_x αντιπροσωπεύει έναν συμβιβασμό μεταξύ μεροληψίας και μεταβλητότητας. Πολύ μικρά h_x οδηγούν σε χαμηλή μεροληψία αλλά υψηλή μεταβλητότητα, ενώ πολύ μεγάλα h_x οδηγούν σε υψηλή μεροληψία αλλά χαμηλή μεταβλητότητα. Η διαδικασία cross-validation που εφαρμόστηκε στα προηγούμενα υποερωτήματα στοχεύει στην εύρεση ενός h_x που εξισορροπεί αυτούς τους δύο παράγοντες.

1.2 Ερώτημα (β): Μέθοδος Bootstrap για τη Μελέτη του $T = \min(X_1, X_2, \ldots, X_n)$

Στο αυτό το ερώτημα της άσκησης θα εστιάσουμε στη μελέτη της κατανομής μιας συγκεκριμένης στατιστικής συνάρτησης, $T = \min(X_1, X_2, \ldots, X_n)$, η οποία υπολογίζεται από ένα τυχαίο δείγμα $X_1, X_2, \ldots, X_n$ μεγέθους $n = 200$, προερχόμενο από έναν άγνωστο συνεχή πληθυσμό. Δεδομένου ότι η πραγματική κατανομή του πληθυσμού (και συνεπώς η ακριβής θεωρητική κατανομή της T) είναι άγνωστη, θα χρησιμοποιήσουμε υπολογιστικές μεθόδους επαναδειγματοληψίας (resampling) για να προσεγγίσουμε την κατανομή της T.

Συγκεκριμένα, στο πρώτο σκέλος θα υλοποιήσουμε τη μέθοδο Bootstrap "χειροκίνητα", γράφοντας τον δικό μας κώδικα στην R. Θα δημιουργήσουμε $B = 2000$ Bootstrap δείγματα, θα υπολογίσουμε την τιμή της T για καθένα από αυτά, και θα κατασκευάσουμε ένα ιστόγραμμα των εκτιμήσεων αυτών. Θα αναλύσουμε το αποτέλεσμα, εξετάζοντας τους λόγους για

τους οποίους η μέθοδος Bootstrap ενδέχεται να μην αποδίδει ικανοποιητικά για τη συγκεκριμένη στατιστική συνάρτηση (το ελάχιστο του δείγματος).

Στο δεύτερο σκέλος, υποθέτοντας ότι εκ των υστέρων πληροφορούμαστε πως το αρχικό δείγμα προέρχεται από μια κατανομή Student, θα εφαρμόσουμε τη μέθοδο του παραμετρικού Bootstrap. Πάλι με $B = 2000$ προσομοιώσεις, θα κατασκευάσουμε ένα νέο ιστόγραμμα για την T και θα συγκρίνουμε τα χαρακτηριστικά του με το προηγούμενο, σχολιάζοντας τις διαφορές και την επίδραση της παραμετρικής πληροφορίας.

Το πρώτο βήμα είναι να φορτώσουμε τα δεδομένα από το αρχείο `"data1b.rds"`:

```r
# Load the data for question 1.b
data_1b <-readRDS("data1b.rds")

# Inspect the data
print(head(data_1b))
print(summary(data_1b)) # To get a sense of the values
```

1.2.1 Υποερώτημα i: Μη Παραμετρικό Bootstrap και οι Περιορισμοί του

Στο παρόν υποερώτημα, επιδιώκουμε να προσεγγίσουμε την κατανομή δειγματοληψίας της στατιστικής συνάρτησης $T = \min(X_1, X_2, \ldots, X_n)$, όπου $X_1, \ldots, X_n$ είναι ένα τυχαίο δείγμα μεγέθους $n = 200$ από έναν άγνωστο συνεχή πληθυσμό. Η πραγματική κατανομή του πληθυσμού, F, είναι άγνωστη, επομένως δεν μπορούμε να υπολογίσουμε θεωρητικά την κατανομή της T.

Η μέθοδος Bootstrap [4] προσφέρει μια λύση σε τέτοια προβλήματα. Η κεντρική ιδέα είναι να χρησιμοποιήσουμε το παρατηρημένο δείγμα $x = (x_1, \ldots, x_n)$ για να κατασκευάσουμε μια εκτίμηση της άγνωστης πληθυσμιακής κατανομής F. Η πιο συνηθισμένη μη παραμετρική προσέγγιση είναι η χρήση της εμπειρικής συνάρτησης κατανομής (Empirical Distribution Function - EDF), $\hat{F}_n$. Η $\hat{F}_n$ αποδίδει πιθανότητα $1/n$ σε κάθε παρατηρημένη τιμή x_i του αρχικού δείγματος.

Η διαδικασία Bootstrap περιλαμβάνει τα εξής βήματα:

1. **Δημιουργία Bootstrap Δειγμάτων**: Από το αρχικό δείγμα x, δημιουργούμε B νέα δείγματα, $x^{*b} = (x_1^{*b}, \ldots, x_n^{*b})$, $b = 1, \ldots, B$. Κάθε Bootstrap δείγμα x^{*b} δημιουργείται με δειγματοληψία **με επανατοποθέτηση** από το αρχικό δείγμα x, και έχει το ίδιο μέγεθος n με το αρχικό δείγμα. Αυτό ισοδυναμεί με δειγματοληψία από την εμπειρική κατανομή $\hat{F}_n$.

2. **Υπολογισμός της Στατιστικής Συνάρτησης**: Για κάθε Bootstrap δείγμα x^{*b}, υπολογίζουμε την τιμή της στατιστικής συνάρτησης ενδιαφέροντος, $T^{*b} = \min(x_1^{*b}, \ldots, x_n^{*b})$.

3. **Προσέγγιση της Κατανομής**: Η συλλογή των B τιμών $T^{*1}, \ldots, T^{*B}$ αποτελεί μια εμπειρική προσέγγιση της κατανομής δειγματοληψίας της T. Μπορούμε να χρησιμοποιήσουμε αυτές τις τιμές για να κατασκευάσουμε ένα ιστόγραμμα, να υπολογίσουμε τυπικά σφάλματα, διαστήματα εμπιστοσύνης κ.λπ.

Θα υλοποιήσουμε αυτή τη διαδικασία "χειροκίνητα" στην `R`, χρησιμοποιώντας $B = 2000$ Bootstrap δείγματα.

Ο παρακάτω κώδικας υλοποιεί τον βρόχο Bootstrap. Σε κάθε επανάληψη b από 1 έως B:

- Δημιουργείται ένα Bootstrap δείγμα `current_bootstrap_sample` μεγέθους `n_obs` με δειγματοληψία με επανατοποθέτηση από το `data_1b`.

- Υπολογίζεται το ελάχιστο του `current_bootstrap_sample`, το οποίο αποθηκεύεται ως η b-οστή τιμή T_b^* στον πίνακα `T_star_values`.

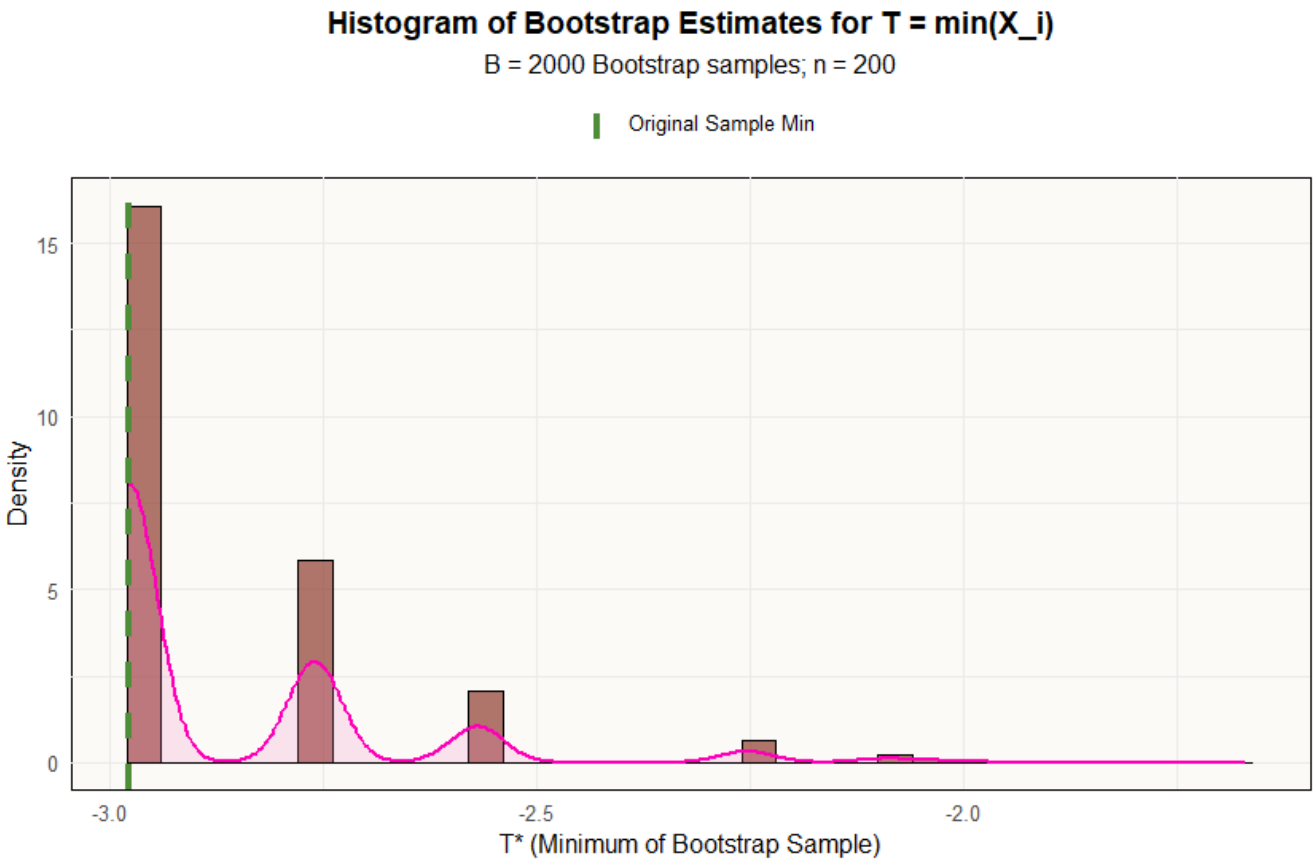
```r
# Parameters for Bootstrap
B <-2000 # Number of Bootstrap samples
n_obs <-length(data_1b)

# Vector to store the T* values (minimum of each Bootstrap sample)
T_star_values <-numeric(B)

# Bootstrap loop
set.seed(123) # For reproducibility
for (b in 1:B) {
  # Step 1: Generate a Bootstrap sample
  # Sample with replacement from the original data_1b
  current_bootstrap_sample <-sample(data_1b, size =n_obs, replace =TRUE)

  # Step 2: Calculate the statistic T* for the current Bootstrap sample
  # T =min(X1, ..., Xn)
  T_star_values[b] <-min(current_bootstrap_sample)
}
```

Αφού υπολογίσαμε τις $B = 2000$ τιμές T_b^*, θα κατασκευάσουμε ένα ιστόγραμμα αυτών των τιμών για να οπτικοποιήσουμε την εκτιμώμενη κατανομή της στατιστικής συνάρτησης T. Θα χρησιμοποιήσουμε τη βιβλιοθήκη `ggplot2` για τη δημιουργία του ιστογράμματος και θα το σχολιάσουμε.

```r
# Create a data frame for ggplot
T_star_df <-data.frame(T_star =T_star_values)

# Calculate the minimum of the original sample for reference
original_sample_min <-min(data_1b)
print(paste("Minimum of the original sample:", original_sample_min))

# Create the histogram
histogram_T_star <-ggplot(T_star_df, aes(x =T_star)) +
  geom_histogram(aes(y =..density..), binwidth =0.04,
                 fill ="#8E3C2E", color ="black", alpha =0.7) +
  geom_density(alpha =0.2, fill ="#F084C1", color ="#FF00B9", linewidth =1) +
  geom_vline(aes(xintercept =original_sample_min,
                 linetype ="Original Sample Min"),
             color ="#4F8E38", linewidth =1.5) +
  scale_linetype_manual(name ="", values =c("Original Sample Min" ="dashed")) +
  labs(title ="Histogram of Bootstrap Estimates for T =min(X_i)",
       subtitle =paste("B =", B, "Bootstrap samples; n =", n_obs),
       x ="T* (Minimum of Bootstrap Sample)",
       y ="Density") +
  theme_minimal(base_size =12) +
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
        plot.subtitle =element_text(hjust =0.5),
        legend.position ="top",
        panel.background =element_rect(fill ="#FBFAF7"))
```

```r
# Print the histogram
print(histogram_T_star)

# Further analysis: How many T_star values are equal to the original sample minimum
count_equal_to_original_min <-sum(T_star_values ==original_sample_min)
percentage_equal_to_original_min <-(count_equal_to_original_min / B) * 100
print(paste("Number of T* values equal to original sample min (", round(original_sample_min,4), "):",
    count_equal_to_original_min))
print(paste("Percentage of T* values equal to original sample min:", round(
    percentage_equal_to_original_min, 2), "%"))
```

Σχήμα 4: Ιστόγραμμα Εκτιμήσεων Boostrap για την $T = \min(X_1, \ldots, X_n)$



Το Σχήμα 4 παρουσιάζει το εν λόγω ιστόγραμμα, μαζί με μια εκτίμηση πυκνότητας και μια κάθετη γραμμή που υποδεικνύει την τιμή του ελαχίστου του αρχικού μας δείγματος, $x_{(1)} = \min(x_1, \ldots, x_n)$.

Από την ανάλυση του ιστογράμματος 4 και των σχετικών υπολογισμών, προκύπτουν τα εξής βασικά συμπεράσματα για τις $B = 2000$ Bootstrap εκτιμήσεις T_b^* της στατιστικής $T = \min(X_i)$. Το ελάχιστο του αρχικού δείγματος, $x_{(1)}$, βρέθηκε ίσο με περίπου -2.9772. Παρατηρείται ότι ένας σημαντικός αριθμός των Bootstrap εκτιμήσεων, συγκεκριμένα 1286 (ποσοστό 64.3%), ταυτίζεται ακριβώς με αυτή την τιμή $x_{(1)}$. Αυτό αντικατοπτρίζεται στο ιστόγραμμα με μια δεσπόζουσα στήλη (μπάρα) στην εν λόγω τιμή, υποδεικνύοντας έντονη συγκέντρωση της εκτιμώμενης κατανομής γύρω από το ελάχιστο του αρχικού δείγματος. Επιπλέον, παρατηρούνται μικρότερες συχνότητες για τιμές T_b^* ελαφρώς μεγαλύτερες του $x_{(1)}$, οι οποίες αντιστοιχούν στις περιπτώσεις όπου άλλες μικρές τιμές του αρχικού δείγματος αποτέλεσαν το ελάχιστο

των αντίστοιχων Bootstrap δειγμάτων (στις περιπτώσεις που το $x_{(1)}$ δεν συμπεριλήφθηκε σε αυτά). Ένα κρίσιμο και **αναμενόμενο** εύρημα είναι ότι καμία Bootstrap εκτίμηση T_b^* δεν έλαβε τιμή μικρότερη από το $x_{(1)}$, γεγονός που αναδεικνύει έναν θεμελιώδη περιορισμό της μεθόδου για τη συγκεκριμένη στατιστική συνάρτηση.

Η παρατηρούμενη συμπεριφορά αναδεικνύει ένα γνωστό όριο της μη παραμετρικής μεθόδου Bootstrap όταν εφαρμόζεται σε στατιστικές συναρτήσεις που εξαρτώνται από τα ακραία σημεία της κατανομής, όπως το ελάχιστο (min) ή το μέγιστο (max), ειδικά όταν η υποκείμενη κατανομή F είναι συνεχής.

Το Bootstrap βασίζεται στην εμπειρική συνάρτηση κατανομής $\hat{F}_n$, η οποία είναι διακριτή και αποδίδει πιθανότητα $1/n$ σε κάθε μία από τις n παρατηρημένες τιμές του αρχικού δείγματος $x_1, \ldots, x_n$. Δεν υπάρχουν τιμές στην $\hat{F}_n$ μικρότερες από το $x_{(1)} = \min(x_i)$ ή μεγαλύτερες από το $x_{(n)} = \max(x_i)$.

Κατά συνέπεια, οποιοδήποτε Bootstrap δείγμα x^{*b} θα αποτελείται αποκλειστικά από τιμές που υπάρχουν στο αρχικό δείγμα. Επομένως, το ελάχιστο ενός Bootstrap δείγματος, $T^{*b} = \min(x_1^{*b}, \ldots, x_n^{*b})$, δεν μπορεί ποτέ να είναι μικρότερο από το ελάχιστο του αρχικού δείγματος, $x_{(1)}$. Αυτό εξηγεί γιατί στο ιστόγραμμα δεν υπάρχουν τιμές αριστερά της πράσινης γραμμής.

Η πιθανότητα μια συγκεκριμένη παρατήρηση του αρχικού δείγματος (όπως το $x_{(1)}$) να μην επιλεγεί σε ένα Bootstrap δείγμα μεγέθους n είναι $(1 - 1/n)^n$. Καθώς το n αυξάνεται, αυτή η πιθανότητα τείνει στο $e^{-1} \approx 0.3679$. Επομένως, η πιθανότητα το $x_{(1)}$ να περιλαμβάνεται τουλάχιστον μία φορά στο Bootstrap δείγμα είναι $1 - (1 - 1/n)^n \approx 1 - e^{-1} \approx 0.6321$. Στην περίπτωσή μας με $n = 200$, αυτή η πιθανότητα είναι πολύ κοντά στο 63.2%. Το γεγονός ότι παρατηρήσαμε το $x_{(1)}$ να είναι το ελάχιστο στο 64.3% των Bootstrap δειγμάτων είναι συνεπές με αυτή τη θεωρητική προσέγγιση. Όταν το $x_{(1)}$ περιλαμβάνεται στο Bootstrap δείγμα, θα είναι σίγουρα και το ελάχιστο αυτού του δείγματος.

Επειδή η Bootstrap κατανομή της T είναι "κομμένη" στο $x_{(1)}$ και δεν μπορεί να εξερευνήσει τιμές μικρότερες από αυτό (ενώ η πραγματική κατανομή της T για μια συνεχή F θα είχε μη μηδενική πιθανότητα για τιμές μικρότερες από ένα συγκεκριμένο $x_{(1)}$ που παρατηρήθηκε), το Bootstrap τείνει να υποεκτιμά τη μεταβλητότητα της T και μπορεί να οδηγήσει σε μεροληπτικές εκτιμήσεις για τα άκρα της κατανομής της T. Η εκτιμώμενη πυκνότητα στο ιστόγραμμα 4 είναι εμφανώς ασύμμετρη και συσσωρευμένη στο $x_{(1)}$.

Συμπερασματικά, για στατιστικές συναρτήσεις όπως το ελάχιστο (ή το μέγιστο), η τυπική μη παραμετρική μέθοδος Bootstrap δεν είναι κατάλληλη για την ακριβή προσέγγιση της κατανομής δειγματοληψίας, καθώς η διακριτή φύση της εμπειρικής κατανομής επιβάλλει τεχνητούς περιορισμούς στις τιμές που μπορεί να πάρει η Bootstrap εκτίμηση.

1.2.2 Υποερώτημα ii: Εφαρμογή Παραμετρικού Bootstrap

Στο προηγούμενο υποερώτημα, παρατηρήσαμε τους περιορισμούς του μη παραμετρικού Bootstrap για την εκτίμηση της κατανομής της στατιστικής $T = \min(X_1, \ldots, X_n)$, οι οποίοι οφείλονται κυρίως στη διακριτή φύση της εμπειρικής συνάρτησης κατανομής.

Στο παρόν υποερώτημα, αλλάζουμε προσέγγιση υποθέτοντας ότι έχουμε επιπλέον πληροφορία: το αρχικό μας δείγμα $x_1, \ldots, x_n$ προέρχεται από μια κατανομή Student $t(\nu, \mu, \sigma)$, όπου ν είναι οι βαθμοί ελευθερίας, μ η παράμετρος θέσης (location) και σ η παράμετρος κλίμακας (scale). Στην τυπική Student t_ν, $\mu = 0$ και $\sigma = 1$. Για μια γενική Student $t(\nu, \mu, \sigma)$, αν $Z \sim t_\nu$, τότε $X = \mu + \sigma Z \sim t(\nu, \mu, \sigma)$.

Η μέθοδος του παραμετρικού Bootstrap [4] λειτουργεί ως εξής:

1. **Εκτίμηση Παραμέτρων**: Από το αρχικό δείγμα $x = (x_1, \ldots, x_n)$, εκτιμούμε τις άγνωστες παραμέτρους της υποτιθέμενης κατανομής. Στην περίπτωσή μας, θα πρέπει να εκτιμήσουμε τις παραμέτρους $\hat{\nu}, \hat{\mu}, \hat{\sigma}$ της κατανομής Student από το `data_1b`. Θα χρησιμοποιήσουμε τη μέθοδο μέγιστης πιθανοφάνειας (Maximum Likelihood Estimation -

MLE) για την εκτίμηση αυτών των παραμέτρων. Οι παράγωγοι της λογαριθμικής πιθανοφάνειας ως προς τις παραμέτρους (που θα έπρεπε να μηδενιστούν για να βρεθούν οι MLEs) οδηγούν σε ένα σύστημα μη γραμμικών εξισώσεων που δεν μπορεί να λυθεί αναλυτικά. Γι' αυτό τον λόγο, η εκτίμηση των παραμέτρων γίνεται **αριθμητικά**. Η συνάρτηση `fitdist` από το πακέτο `fitdistrplus` της `R` μπορεί να χρησιμοποιηθεί για αυτόν τον σκοπό.

2. **Δημιουργία Παραμετρικών Bootstrap Δειγμάτων**: Δημιουργούμε B νέα δείγματα Boostrap, $x^{*b} = (x_1^{*b}, \ldots, x_n^{*b})$, $b = 1, \ldots, B$. Κάθε x_i^{*b} παράγεται από την εκτιμημένη παραμετρική κατανομή, δηλαδή $X_i^{*b} \sim t(\hat{\nu}, \hat{\mu}, \hat{\sigma})$.

3. **Υπολογισμός της Στατιστικής Συνάρτησης**: Για κάθε παραμετρικό Bootstrap δείγμα x^{*b}, υπολογίζουμε την τιμή της στατιστικής συνάρτησης, $T^{*b} = \min(x_1^{*b}, \ldots, x_n^{*b})$.

4. **Προσέγγιση της Κατανομής**: Η συλλογή των B τιμών $T^{*1}, \ldots, T^{*B}$ αποτελεί μια εμπειρική προσέγγιση της κατανομής δειγματοληψίας της T, υπό την παραδοχή ότι η αρχική μας υπόθεση για την κατανομή Student είναι σωστή.

Θα υλοποιήσουμε αυτή τη διαδικασία χρησιμοποιώντας $B = 2000$ παραμετρικά Bootstrap δείγματα.

Θα ξεκινήσουμε με την εκτίμηση των παραμέτρων της κατανομής Student από το αρχικό δείγμα `data_1b`.

```r
# Load necessary library
library(fitdistrplus)

# --- Fit a t-distribution with location and scale using a custom definition ---
# The standard 't' distribution in R is for the non-central t or standard t.
# We need to define the density for a location-scale t-distribution.
# dlst =density of location-scale t (df, mu, sigma)
# plst =distribution function of location-scale t
# qlst =quantile function of location-scale t
# rlst =random generation from location-scale t

# Density function for location-scale t
dlst <-function(x, df, mu, sigma) {
  if (any(sigma <= 0) || any(df <= 0)) return(rep(NaN, length(x))) # sigma and df must be positive
  return(1/sigma * dt((x - mu)/sigma, df))
}

# Distribution function for location-scale t
plst <-function(q, df, mu, sigma) {
  if (any(sigma <= 0) || any(df <= 0)) return(rep(NaN, length(q)))
  return(pt((q - mu)/sigma, df))
}

# Quantile function for location-scale t
qlst <-function(p, df, mu, sigma) {
  if (any(sigma <= 0) || any(df <= 0)) return(rep(NaN, length(p)))
  return(mu + sigma * qt(p, df))
}

# Random generation function (will be needed for Bootstrap)
rlst <-function(n, df, mu, sigma) {
  if (any(sigma <= 0) || any(df <= 0)) return(rep(NaN, n))
  return(mu + sigma * rt(n, df))
}
```

```r
# We need to provide reasonable starting values for df, mu, sigma.
mu_start <-mean(data_1b)
sigma_start <-sd(data_1b)
df_start <-5

# Using tryCatch to handle potential errors during fitting
student_fit <-NULL
tryCatch({
  student_fit <-fitdist(data_1b, "lst", start =list(df =df_start, mu =mu_start, sigma =sigma_start),
      lower =c(0.001, -Inf, 0.001)) # df and sigma > 0
}, error =function(e) {
  print(paste("Error during fitting:", e$message))
})

# Check the fit
print(summary(student_fit))

# Extract estimated parameters
est_params_student <-coef(student_fit)
nu_hat <-est_params_student["df"]
mu_hat <-est_params_student["mu"]
sigma_hat <-est_params_student["sigma"]

print(paste("Estimated nu (df):", nu_hat))
print(paste("Estimated mu:", mu_hat))
print(paste("Estimated sigma:", sigma_hat))
```

Τα αποτελέσματα από την προσαρμογή της κατανομής Student $t(\nu, \mu, \sigma)$ στα δεδομένα `data_1b` μέσω της μεθόδου μέγιστης πιθανοφάνειας, χρησιμοποιώντας τη συνάρτηση `fitdist` του πακέτου `fitdistrplus`, παρουσιάζονται παρακάτω. Οι εκτιμώμενες τιμές των παραμέτρων είναι:

- **Εκτιμώμενοι βαθμοί ελευθερίας ($\hat{\nu}$):** 10.21 (με τυπικό σφάλμα ≈ 1.09)
- **Εκτιμώμενη παράμετρος θέσης ($\hat{\mu}$):** -0.045 (με τυπικό σφάλμα ≈ 0.088)
- **Εκτιμώμενη παράμετρος κλίμακας ($\hat{\sigma}$):** 1.003 (με τυπικό σφάλμα ≈ 0.090)

Οι εκτιμήσεις αυτές φαίνονται λογικές. Η εκτιμώμενη παράμετρος θέσης $\hat{\mu} \approx -0.045$ είναι κοντά στο μηδέν, και η εκτιμώμενη παράμετρος κλίμακας $\hat{\sigma} \approx 1.003$ είναι κοντά στη μονάδα, υποδηλώνοντας ότι τα δεδομένα, αν προέρχονται από κατανομή Student, μπορεί να μην απέχουν πολύ από μια τυπική (ή ελαφρώς μετατοπισμένη/κλιμακωμένη) κατανομή Student. Οι βαθμοί ελευθερίας $\hat{\nu} \approx 10.21$ υποδεικνύουν μια κατανομή με κάπως βαρύτερες ουρές σε σχέση με την κανονική κατανομή (καθώς για $\nu \to \infty$, η κατανομή Student τείνει στην κανονική). Τα τυπικά σφάλματα των εκτιμήσεων μας δίνουν μια ένδειξη της ακρίβειας αυτών των εκτιμήσεων.

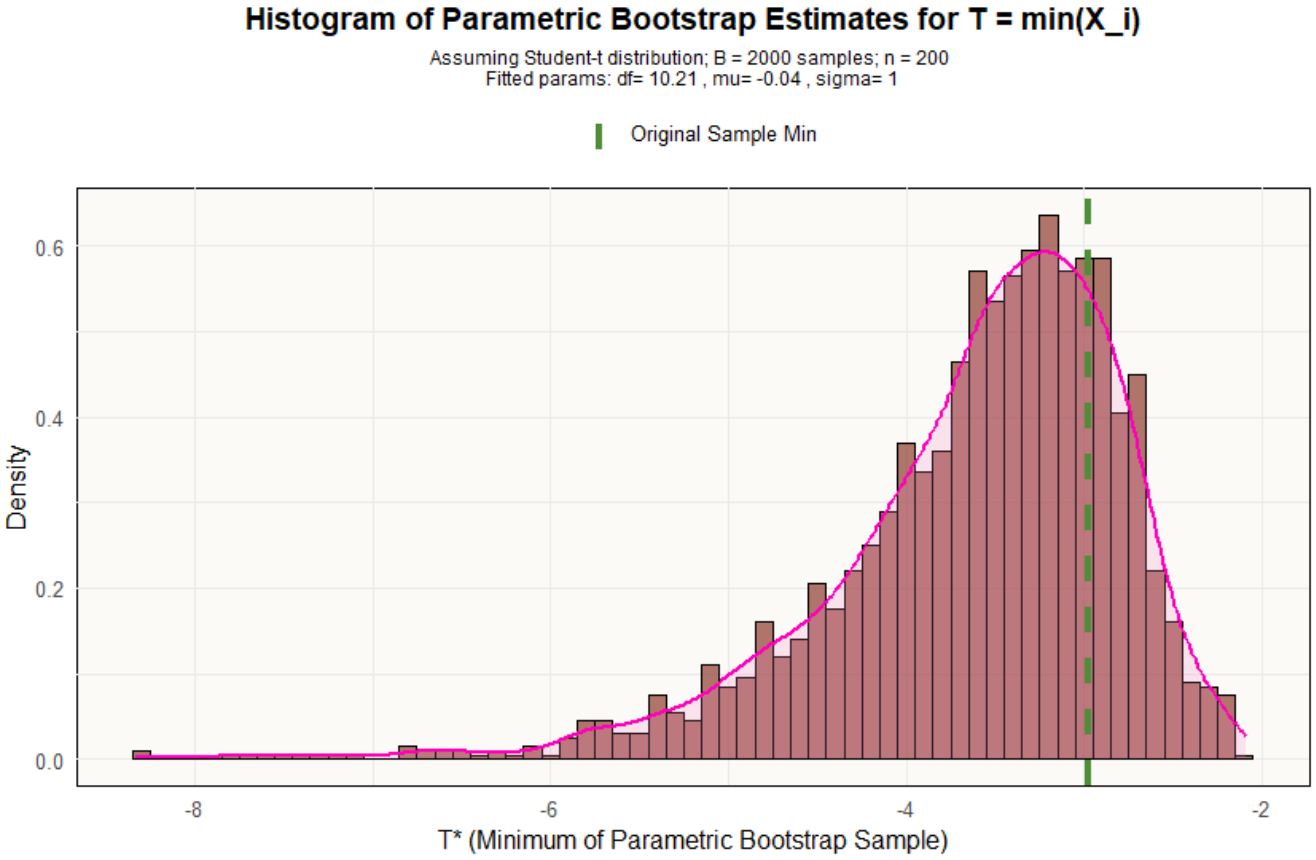
Αφού έχουμε εκτιμήσει τις παραμέτρους $\hat{\nu}, \hat{\mu}, \hat{\sigma}$ της κατανομής Student, το επόμενο βήμα είναι η παραγωγή $B = 2000$ παραμετρικών Bootstrap δειγμάτων. Κάθε δείγμα x^{*b} θα έχει μέγεθος $n = 200$ (όσο και το αρχικό δείγμα) και οι τιμές του θα προέρχονται από την εκτιμηθείσα κατανομή $t(\hat{\nu}, \hat{\mu}, \hat{\sigma})$. Για τη δημιουργία τυχαίων αριθμών από αυτή την κλιμακωμένη και μετατοπισμένη κατανομή Student, θα χρησιμοποιήσουμε τη συνάρτηση `rlst` που ορίσαμε προηγουμένως, η οποία με τη σειρά της καλεί την `rt` της `R` για την τυπική Student και εφαρμόζει τον κατάλληλο μετασχηματισμό. Για κάθε παραγόμενο Bootstrap δείγμα, θα υπολογίσουμε τη στατιστική συνάρτηση $T^{*b} = \min(x_1^{*b}, \ldots, x_n^{*b})$.

```r
# Parameters for Bootstrap
B <-2000
n_obs <-length(data_1b)

# Vector to store the T values from parametric bootstrap
T_star_parametric_values <-numeric(B)


set.seed(456) # For reproducibility of parametric bootstrap
# Parametric Bootstrap loop
for (b in 1:B) {
  # Step 1: Generate a parametric Bootstrap sample
  # Sample from the fitted Student-t distribution t(nu_hat, mu_hat, sigma_hat)
  current_parametric_bootstrap_sample <-rlst(n =n_obs, df =nu_hat, mu =mu_hat, sigma =sigma_hat)

  # Step 2: Calculate the statistic T* for the current sample
  T_star_parametric_values[b] <-min(current_parametric_bootstrap_sample)
}
```

Με τις $B = 2000$ τιμές T^{*b} που προέκυψαν από τη διαδικασία του παραμετρικού Bootstrap, θα κατασκευάσουμε ένα ιστόγραμμα για να οπτικοποιήσουμε την εκτιμώμενη κατανομή της στατιστικής $T = \min(X_i)$.

```r
# Create a data frame for ggplot
T_star_parametric_df <-data.frame(T_star_param =T_star_parametric_values)

# Minimum of the original sample
original_sample_min <-min(data_1b)

# Create the histogram for parametric bootstrap estimates
histogram_T_star_parametric <-ggplot(T_star_parametric_df, aes(x =T_star_param)) +
  geom_histogram(aes(y =..density..), binwidth =0.1,
                 fill ="#8E3C2E", color ="black", alpha =0.7) +
  geom_density(alpha =0.2, fill ="#F084C1", color ="#FF00B9", linewidth =1) +
  geom_vline(aes(xintercept =original_sample_min,
                 linetype ="Original Sample Min"),
             color ="#4F8E38", linewidth =1.5) +
  scale_linetype_manual(name ="", values =c("Original Sample Min" ="dashed")) +
  labs(title ="Histogram of Parametric Bootstrap Estimates for T =min(X_i)",
       subtitle =paste("Assuming Student-t distribution; B =", B, "samples; n =", n_obs,
                       "\nFitted params: df=", round(nu_hat,2), ", mu=", round(mu_hat,2), ", sigma=", round(
                           sigma_hat,2)),
       x ="T* (Minimum of Parametric Bootstrap Sample)",
       y ="Density") +
  theme_minimal(base_size =12) +
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
       plot.subtitle =element_text(hjust =0.5, size=9),
       legend.position ="top",
       panel.background =element_rect(fill ="#FBFAF7"))

# Print the histogram
print(histogram_T_star_parametric)
```

Σχήμα 5: Ιστόγραμμα Εκτιμήσεων Boostrap για την $T = \min(X_1, \ldots, X_n)$ με την υπόθεση ότι $X_i \sim t(\hat{\nu} \approx 10.21, \hat{\mu} \approx -0.045, \hat{\sigma} \approx 1.003)$



Το Σχήμα 5 παρουσιάζει το ιστόγραμμα των $B = 2000$ τιμών $T^{*b} = \min(x_1^{*b}, \ldots, x_n^{*b})$ που προέκυψαν από τη διαδικασία του παραμετρικού Bootstrap. Για τη δημιουργία των Bootstrap δειγμάτων, χρησιμοποιήθηκε η εκτιμηθείσα κατανομή Student $t(\hat{\nu} \approx 10.21, \hat{\mu} \approx -0.045, \hat{\sigma} \approx 1.003)$. Η πράσινη διακεκομμένη γραμμή στο σχήμα υποδεικνύει την τιμή του ελαχίστου του αρχικού δείγματος, $x_{(1)} \approx -2.9772$.

Σε πλήρη αντίθεση με το ιστόγραμμα του μη παραμετρικού Bootstrap (Σχήμα 4), η κατανομή των παραμετρικών Bootstrap εκτιμήσεων T^* είναι εμφανώς πιο ομαλή και συνεχής, όπως θα αναμενόταν από δειγματοληψία από μια συνεχή παραμετρική κατανομή. Δεν παρατηρούνται οι έντονες "αιχμές" και η συσσώρευση σε μεμονωμένες τιμές του αρχικού δείγματος που χαρακτήριζαν το μη παραμετρικό αποτέλεσμα.

Το πιο σημαντικό εύρημα είναι ότι η παραμετρική Bootstrap κατανομή της T εκτείνεται και σε τιμές μικρότερες από το ελάχιστο του αρχικού δείγματος $x_{(1)}$. Αυτό είναι κρίσιμο, καθώς η στατιστική $\min(X_i)$ από μια συνεχή υποκείμενη κατανομή μπορεί πράγματι να λάβει τιμές μικρότερες από οποιοδήποτε παρατηρημένο ελάχιστο σε ένα συγκεκριμένο δείγμα. Το παραμετρικό Bootstrap, δειγματοληπτώντας από μια συνεχή εκτιμώμενη κατανομή, είναι σε θέση να συλλάβει αυτή τη συμπεριφορά.

Η συνολική μορφή της κατανομής είναι ασύμμετρη προς τα αριστερά (left-skewed), όπως θα περίμενε κανείς για την κατανομή του ελαχίστου. Η διασπορά των τιμών T^* φαίνεται να είναι πιο ρεαλιστική και λιγότερο τεχνητά περιορισμένη σε σύγκριση με το μη παραμετρικό Bootstrap.

Η σύγκριση των δύο προσεγγίσεων (μη παραμετρικής και παραμετρικής Bootstrap) για την εκτίμηση της κατανομής

της $T = \min(X_i)$ είναι σημαντική. Ενώ το μη παραμετρικό Bootstrap απέτυχε να παράγει μια χρήσιμη προσέγγιση λόγω των περιορισμών που επιβάλλει η διακριτή φύση της εμπειρικής κατανομής, το παραμετρικό Bootstrap, υπό την προϋπόθεση ότι η παραδοχή για την κατανομή Student είναι εύλογη (ή τουλάχιστον μια καλύτερη προσέγγιση από την EDF για τα άκρα), παρέχει μια σημαντικά βελτιωμένη και πιο ρεαλιστική εικόνα.

Η ικανότητα του παραμετρικού Bootstrap να παράγει τιμές T^* μικρότερες από το $x_{(1)}$ είναι το βασικό του πλεονέκτημα εδώ, καθώς επιτρέπει μια πιο ολοκληρωμένη εξερεύνηση της πιθανής συμπεριφοράς της στατιστικής $\min(X_i)$. Αυτό αναδεικνύει τη σημασία της ενσωμάτωσης εξωτερικής πληροφορίας ή παραδοχών (όπως η μορφή της υποκείμενης κατανομής) όταν η τυπική μη παραμετρική προσέγγιση δεν είναι κατάλληλη, ειδικά για στατιστικές συναρτήσεις που εξαρτώνται από τα άκρα της κατανομής. Φυσικά, η εγκυρότητα των αποτελεσμάτων του παραμετρικού Bootstrap εξαρτάται άμεσα από την ορθότητα της παραδοχής για την κατανομή Student. Αν αυτή η παραδοχή είναι λανθασμένη, τότε και τα συμπεράσματα από το παραμετρικό Bootstrap μπορεί να είναι παραπλανητικά.

2 Άσκηση 2

2.1 Ερώτημα (α): Μέθοδος Rejection Sampling με Squeezing

Στο παρών ερώτημα, θα εξετάσουμε την προσομοίωση τυχαίων τιμών από την τυποποιημένη κανονική κατανομή $N(0,1)$, την οποία συμβολίζουμε με $f(x)$. Επειδή η άμεση προσομοίωση μέσω της μεθόδου της αντιστροφής δεν είναι πρακτική για την κανονική κατανομή, θα εστιάσουμε στην εφαρμογή της μεθόδου της απόρριψης (Rejection Sampling), και πιο συγκεκριμένα, σε μια βελτιωμένη εκδοχή της, τη "squeezed" μέθοδο της απόρριψης.

Ως κατανομή εισήγησης (proposal distribution), $g(x)$, από την οποία μπορούμε εύκολα να δειγματοληπτήσουμε, θα χρησιμοποιήσουμε τη διπλή εκθετική (Laplace) κατανομή, $g(x) = \frac{1}{2}e^{-|x|}$. Η "squeezed" μέθοδος εισάγει επιπλέον μια συνάρτηση συμπίεσης $s(x)$ (προκύπτουσα από την ανισότητα $e^{-x^2/2} \geq 1 - x^2/2$ για $|x| \leq \sqrt{2}$), η οποία επιτρέπει την ταχύτερη αποδοχή δειγμάτων υπό ορισμένες συνθήκες, βελτιώνοντας την αποδοτικότητα του αλγορίθμου σε σχέση με την απλή μέθοδο απόρριψης. Θα υλοποιήσουμε αυτή την τεχνική για να παράγουμε 10000 τιμές από την $N(0,1)$, θα αναλύσουμε την πιθανότητα αποδοχής και θα συζητήσουμε τα χαρακτηριστικά μιας καλής κατανομής εισήγησης.

2.1.1 Υποερώτημα i: Προσομοίωση από $N(0,1)$ με Squeezed Rejection Sampling

Στο παρόν υποερώτημα, ο στόχος μας είναι η προσομοίωση τυχαίων τιμών από την **τυποποιημένη κανονική κατανομή**, $N(0,1)$. Η συνάρτηση πυκνότητας πιθανότητας (pdf) αυτής της κατανομής, την οποία θα συμβολίζουμε με $f(x)$, δίνεται από τον τύπο:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Στην R, αυτή η συνάρτηση είναι διαθέσιμη μέσω της εντολής `dnorm(x, mean = 0, sd = 1)`.

```r
# Target density function f(x): Standard Normal N(0,1)
f_target <-function(x) {
  dnorm(x, mean =0, sd =1)
}
```

Για την εφαρμογή της μεθόδου απόρριψης, χρειαζόμαστε μια συνάρτηση εισήγησης (proposal distribution), $g(x)$, από την οποία μπορούμε εύκολα να παράγουμε δείγματα. Σύμφωνα με την εκφώνηση, ως κατανομή εισήγησης θα χρησιμοποιήσουμε τη διπλή εκθετική (Laplace) κατανομή με παράμετρο θέσης 0 και παράμετρο κλίμακας 1. Η συνάρτηση πυκνότητας πιθανότητας της $g(x)$ είναι:

$$g(x) = \frac{1}{2}e^{-|x|}$$

Η προσομοίωση τιμών από την $g(x)$ θα γίνει με τη μέθοδο της αντιστροφής (inversion method) [5]. Για να το κάνουμε αυτό, χρειαζόμαστε πρώτα την αθροιστική συνάρτηση κατανομής (CDF) της $g(x)$, την $F_g(x)$.

- **Για** $x < 0$:

$$F_g(x) = \int_{-\infty}^{x} \frac{1}{2}e^{t}dt = \left[\frac{1}{2}e^{t}\right]_{-\infty}^{x} = \frac{1}{2}e^{x}$$

- **Για** $x \geq 0$:

$$F_g(x) = \int_{-\infty}^{0} \frac{1}{2}e^{t}dt + \int_{0}^{x} \frac{1}{2}e^{-t}dt = \frac{1}{2} + \left[-\frac{1}{2}e^{-t}\right]_{0}^{x} = \frac{1}{2} - \frac{1}{2}e^{-x} + \frac{1}{2} = 1 - \frac{1}{2}e^{-x}$$

Επομένως, η CDF είναι:

$$F_g(x) = \begin{cases} \frac{1}{2}e^x & \text{if } x < 0 \\ 1 - \frac{1}{2}e^{-x} & \text{if } x \geq 0 \end{cases}$$

Για να εφαρμόσουμε τη μέθοδο της αντιστροφής, θέτουμε $U = F_g(X)$ όπου $U \sim U(0,1)$ και λύνουμε ως προς X:

- **Περίπτωση 1**: Αν $0 < U < 0.5$ (αντιστοιχεί σε $X < 0$)

$$U = \frac{1}{2}e^X \implies 2U = e^X \implies X = \ln(2U)$$

- **Περίπτωση 2**: Αν $0.5 \leq U < 1$ (αντιστοιχεί σε $X \geq 0$)

$$U = 1 - \frac{1}{2}e^{-X} \implies \frac{1}{2}e^{-X} = 1 - U \implies e^{-X} = 2(1-U) \implies X = -\ln(2(1-U))$$

```r
# Proposal density function g(x): Laplace(0,1) or Double Exponential(0,1)
g_proposal <-function(x) {
  0.5 * exp(-abs(x))
}

# Function to generate a sample from g_proposal(x) using the inversion method
generate_from_g_inversion <-function() {
  u <-runif(1) # Generate a U ~ U(0,1)
  if (u < 0.5) {
    # Corresponds to x < 0
    x_sample <-log(2 * u)
  } else {
    # Corresponds to x >= 0
    x_sample <--log(2 * (1 - u))
  }
  return(x_sample)
}
```

Για την εφαρμογή της μεθόδου απόρριψης, πρέπει να βρούμε μια σταθερά $M > 0$ τέτοια ώστε $f(x) \leq M \cdot g(x)$ για κάθε x. Η τιμή αυτής της σταθεράς δίνεται στην εκφώνηση ως $M = \sqrt{2e/\pi}$. Η συνάρτηση φακέλου (envelope function) ορίζεται τότε ως $G(x) = M \cdot g(x)$.

```r
M_constant <-sqrt(2 * exp(1) / pi)

# Envelope function G(x) =M * g_proposal(x)
G_envelope <-function(x) {
  M_constant * g_proposal(x)
}
```

Η "squeezed" μέθοδος της απόρριψης [5] χρησιμοποιεί μια συνάρτηση συμπίεσης (squeezing function), $s(x)$, η οποία πρέπει να ικανοποιεί τη συνθήκη $0 \leq s(x) \leq f(x)$ για κάθε x και να είναι υπολογιστικά "φθηνότερη" από την $f(x)$.

Η εκφώνηση αναφέρεται στην ανισότητα $e^{-y} \geq 1 - y$ για $y \geq 0$. Αν θέσουμε $y = x^2/2$ (το οποίο είναι πάντα ≥ 0), τότε $e^{-x^2/2} \geq 1 - x^2/2$. Πολλαπλασιάζοντας με $1/\sqrt{2\pi}$, παίρνουμε ένα κάτω φράγμα για την $f(x)$:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \geq \frac{1}{\sqrt{2\pi}}(1 - x^2/2)$$

Ωστόσο, ο όρος $1 - x^2/2$ γίνεται αρνητικός όταν $x^2/2 > 1$, δηλαδή όταν $|x| > \sqrt{2}$. Επειδή η συνάρτηση συμπίεσης $s(x)$ πρέπει να είναι μικρότερη ή ίση της $f(x)$ (η οποία είναι πάντα μη αρνητική), και ιδανικά μη αρνητική η ίδια για να έχει νόημα ο λόγος $s(x)/G(x)$, ορίζουμε τη συνάρτηση συμπίεσης ως:

$$s(x) = \frac{1}{\sqrt{2\pi}} \max\{0, 1 - x^2/2\}$$

Αυτή η συνάρτηση $s(x)$ θα χρησιμοποιηθεί στον "φθηνό" έλεγχο του αλγορίθμου.

```r
# Squeezing function s(x)
s <-function(x) {
  term_in_max <-1 - (x^2 / 2)
  (1 / sqrt(2 * pi)) * pmax(0, term_in_max) # pmax ensures element-wise maximum with 0
}
```

Ο αλγόριθμος για την παραγωγή μιας τυχαίας τιμής X από την συνάρτηση πυκνότητας πιθανότητας στόχου $f(x)$, χρησιμοποιώντας την συνάρτηση εισήγησης $g(x)$, την σταθερά M, την συνάρτηση φακέλου $G(x) = M \cdot g(x)$ και την συνάρτηση συμπίεσης $s(x) \leq f(x)$, έχει τα παρακάτω βήματα, τα οποία επαναλαμβάνονται μέχρι να γίνει αποδεκτή μια τιμή:

1. **Δημιουργία από την Συνάρτηση Εισήγησης**: Δημιουργούμε μια υποψήφια τιμή $Y \sim g(y)$ και θέτουμε $y_{cand} = Y$.
2. **Δημιουργία Τυχαίου Αριθμού**: Δημιουργούμε έναν τυχαίο αριθμό $U \sim U(0, 1)$ και θέτουμε $u = U$.
3. **Squeeze Test - "Φθηνός" Έλεγχος**: Υπολογίζουμε τον λόγο $r_s = \frac{s(y_{cand})}{G(y_{cand})}$. **Αν** $u \leq r_s$, τότε αποδεχόμαστε την y_{cand} ως τιμή από την $f(x)$ (δηλαδή, $X = y_{cand}$) και **σταματάμε** την τρέχουσα επανάληψη του αλγορίθμου. **Αλλιώς συνεχίζουμε** στο επόμενο βήμα.
4. **Full Test - "Ακριβός" Έλεγχος, αν ο Squeeze Test απέτυχε**: Αν το Squeeze Test στο Βήμα 3 απέτυχε (δηλαδή $u > r_s$), τότε υπολογίζουμε τον λόγο $r_f = \frac{f(y_{cand})}{G(y_{cand})}$. **Αν** $u \leq r_f$, τότε αποδεχόμαστε την y_{cand} ως τιμή από την $f(x)$ (δηλαδή, $X = y_{cand}$) και **σταματάμε** την τρέχουσα επανάληψη. **Αλλιώς συνεχίζουμε** στο επόμενο βήμα και τελευταίο βήμα.
5. **Απόρριψη**: Αν και το Full Test στο Βήμα 4 απέτυχε (δηλαδή $u > r_s$ και $u > r_f$), τότε απορρίπτουμε την y_{cand} και επαναλαμβάνουμε τον αλγόριθμο από το Βήμα 1 για να δημιουργήσουμε μια νέα υποψήφια τιμή.

Αυτή η διαδικασία επαναλαμβάνεται μέχρι να συλλέξουμε τον επιθυμητό αριθμό δειγμάτων (στην περίπτωσή μας 10000) από την κατανομή $f(x)$.

```r
# Number of samples to generate from the target distribution f(x)
num_samples_target <-10000
generated_samples_from_f <-numeric(num_samples_target) # Vector to store accepted samples

# Counters for analysis
count_accepted_f <-0 # Number of samples accepted so far
```

```r
total_attempts_f <-0 # Total number of proposals generated from g(x)
f_calculations_avoided_f <-0 # Number of times f_target(x) was not calculated due to successful squeeze

set.seed(12345) # Seed for reproducibility

# Main loop to generate the required number of samples
while (count_accepted_f < num_samples_target) {
 total_attempts_f <-total_attempts_f + 1

 # Step 1: Generate a candidate sample Y_cand from g_proposal(x)
 Y_cand <-generate_from_g_inversion()

 # Step 2: Generate a U_rand from U(0,1)
 U_rand <-runif(1)

 # Calculate G_envelope(Y_cand) =M_constant * g_proposal(Y_cand)
 G_Y_cand <-G_envelope(Y_cand) # M*g(y)

 # Step 3: Squeeze Test
 s_Y_cand <-s(Y_cand) # s(y)
 ratio_squeeze_test <-s_Y_cand / G_Y_cand # This is s(y) / (M*g(y))

 if (U_rand <= ratio_squeeze_test) {
  # Accept Y_cand based on the squeeze test
  count_accepted_f <-count_accepted_f + 1
  generated_samples_from_f[count_accepted_f] <-Y_cand
  f_calculations_avoided_f <-f_calculations_avoided_f + 1
 } else {
  # Step 4: Full Test (Squeeze Test failed)
  f_Y_cand <-f_target(Y_cand) # f(y) - Calculate f(y) only if squeeze test fails
  ratio_full_test <-f_Y_cand / G_Y_cand # This is f(y) / (M*g(y))

  if (U_rand <= ratio_full_test) {
   # Accept Y_cand based on the full test
   count_accepted_f <-count_accepted_f + 1
   generated_samples_from_f[count_accepted_f] <-Y_cand
  }
  # Else (U_rand > ratio_full_test): Reject Y_cand, and loop to try again
 }
}

# Print summary statistics of the sampling process
cat("\n--- Squeezed Rejection Sampling Process Summary ---\n")
cat("Target number of samples from f(x):", num_samples_target, "\n")
cat("Constant M used:", M_constant, "\n")
cat("Total proposals generated from g(x) (Total Attempts):", total_attempts_f, "\n")
cat("Total samples accepted for f(x):", count_accepted_f, "\n")
cat("Empirical overall acceptance probability:", count_accepted_f / total_attempts_f, "\n")
cat("Theoretical overall acceptance probability (1/M for simple rejection):", 1 / M_constant, "\n")
cat("Number of times f_target(x) calculation was avoided (successful squeeze):",
    f_calculations_avoided_f, "\n")
cat("Proportion of attempts that were successful squeezes:", f_calculations_avoided_f /
    total_attempts_f, "\n")
cat("Proportion of accepted samples that came from successful squeezes:", f_calculations_avoided_f /
    count_accepted_f, "\n")
```

Για την παραγωγή των 10000 επιθυμητών δειγμάτων από την $f(x)$, χρειάστηκαν συνολικά 13140 προσπάθειες (δηλαδή, δημιουργήθηκαν 13140 υποψήφιες τιμές από την $g(x)$).

Η εμπειρική συνολική πιθανότητα αποδοχής υπολογίστηκε σε $10000/13140 \approx 0.7610$. Αυτή η τιμή είναι, όπως αναμενόταν, πολύ κοντά στη θεωρητική συνολική πιθανότητα αποδοχής για την απλή μέθοδο απόρριψης, η οποία είναι $1/M \approx 1/1.3155 \approx 0.7602$. Η μικρή διαφορά οφείλεται στην τυχαιότητα της δειγματοληψίας.

Ένα σημαντικό εύρημα είναι η αποτελεσματικότητα της συνάρτησης συμπίεσης: από τις 10000 φορές που ένα δείγμα έγινε αποδεκτό, στις 7569 περιπτώσεις (75.69%) η αποδοχή έγινε μέσω του "φθηνού" squeeze test. Αυτό σημαίνει ότι ο υπολογισμός της (πιο ακριβής υπολογιστικά) συνάρτησης $f(x)$ αποφεύχθηκε για ένα μεγάλο ποσοστό των αποδεκτών δειγμάτων, επιβεβαιώνοντας την χρησιμότητα της "squeezed" τεχνικής στη μείωση του υπολογιστικού κόστους.

Η αναλογία των επιτυχών squeeze tests επί του συνόλου των προσπαθειών ήταν $7569/13140 \approx 0.5760$.

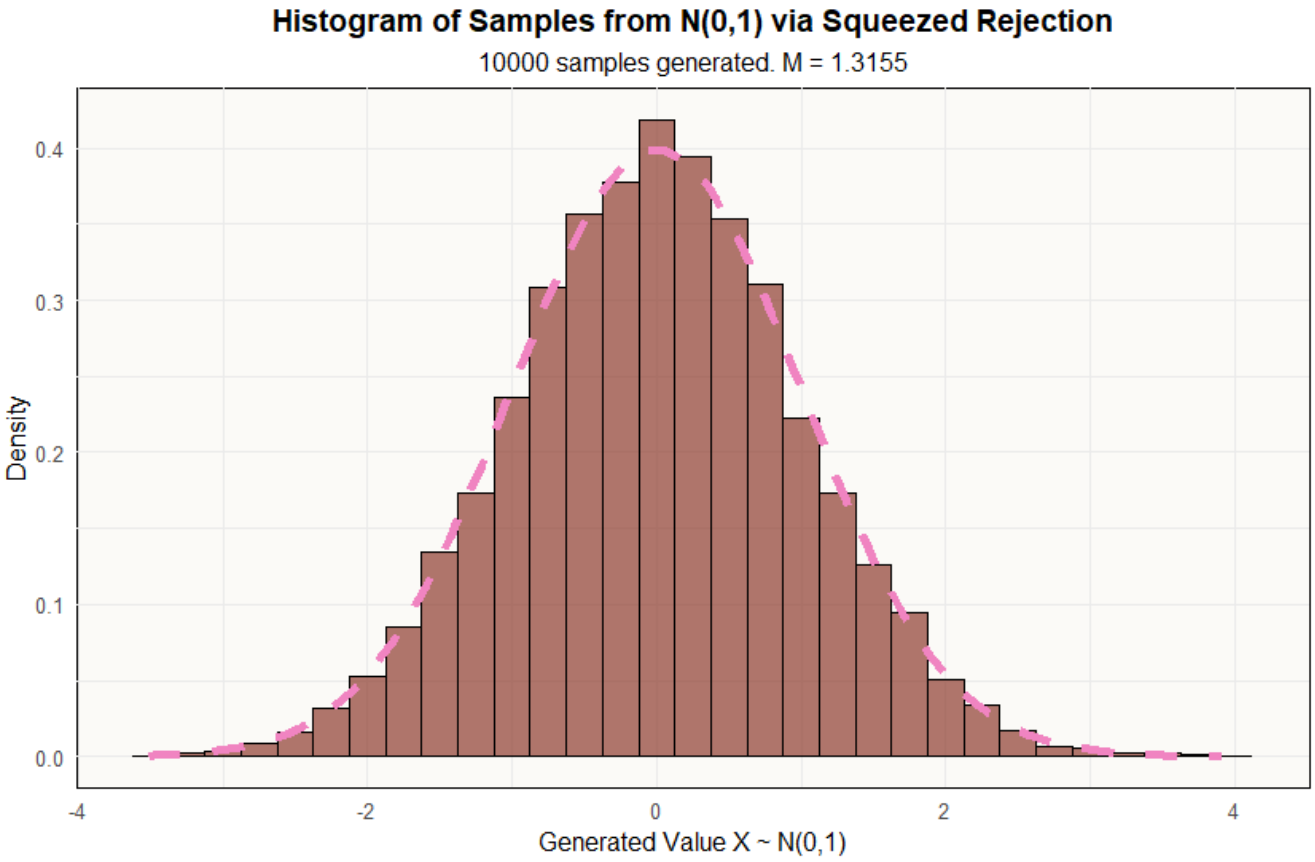
Αυτά τα αποτελέσματα υποδεικνύουν ότι η επιλεγμένη συνάρτηση εισήγησης $g(x)$ και η συνάρτηση συμπίεσης $s(x)$ λειτούργησαν ικανοποιητικά. Η εμπειρική πιθανότητα αποδοχής είναι κοντά στη θεωρητική μέγιστη δυνατή (που θα είχαμε αν δεν υπήρχε καθόλου "squeeze" όφελος, δηλαδή αν $s(x) = 0$ παντού), και ένα σημαντικό ποσοστό των υπολογισμών της $f(x)$ αποφεύχθηκε, καθιστώντας τον αλγόριθμο πιο αποδοτικό.

Στη συνέχεια, θα οπτικοποιήσουμε την κατανομή των 10000 προσομοιωμένων δειγμάτων μέσω ενός ιστογράμματος, το οποίο θα συγκρίνουμε με τη θεωρητική καμπύλη της τυποποιημένης κανονικής κατανομής για να αξιολογήσουμε την ποιότητα της προσομοίωσης.

```
# Create a data frame for ggplot from the generated samples
samples_df_f <-data.frame(x_values =generated_samples_from_f)

# Create the histogram with the theoretical N(0,1) density overlaid
histogram_f_squeezed <-ggplot(samples_df_f, aes(x =x_values)) +
  geom_histogram(aes(y =..density..),
                 binwidth =0.25,
                 fill ="#8E3C2E",
                 color ="black",
                 alpha =0.7) +
  stat_function(fun =dnorm, args =list(mean =0, sd =1),
                color ="#F084C1",
                linewidth =1.1,
                linetype ="dashed") +
  labs(title ="Histogram of Samples from N(0,1) via Squeezed Rejection",
       subtitle =paste(num_samples_target, "samples generated. M =", round(M_constant, 4)),
       x ="Generated Value X ~ N(0,1)",
       y ="Density") +
  theme_minimal(base_size =12) +
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
        plot.subtitle =element_text(hjust =0.5),
        panel.background =element_rect(fill ="#FBFAF7"))

# Print the histogram
print(histogram_f_squeezed)
```

Το Σχήμα 6 παρουσιάζει το ιστόγραμμα των 10000 τιμών που προσομοιώθηκαν από την τυποποιημένη κανονική κατανομή $f(x)$ χρησιμοποιώντας την "squeezed" μέθοδο της απόρριψης. Στο ίδιο σχήμα, έχει σχηματισθεί η θεωρητική καμπύλη πυκνότητας της $N(0, 1)$ (με διακεκομμένη γραμμή) για άμεση οπτική σύγκριση.

Σχήμα 6: Ιστόγραμμα Προσομοιώσεων από την $N(0,1)$ μέσω της Μεθόδου Squeezed Rejection



Από το σχήμα, είναι εμφανές ότι τα προσομοιωμένα δείγματα ακολουθούν πολύ πιστά τη μορφή της τυποποιημένης κανονικής κατανομής. Το ιστόγραμμα παρουσιάζει τη χαρακτηριστική καμπανοειδή συμμετρία γύρω από το μηδέν και η εμπειρική κατανομή που αναπαριστά προσεγγίζει ικανοποιητικά τη θεωρητική καμπύλη.

Συνολικά, η υλοποίηση της "squeezed" μεθόδου απόρριψης, με την κατάλληλη επιλογή της κατανομής εισήγησης $g(x)$ (διπλή εκθετική), της σταθεράς M, και της συνάρτησης συμπίεσης $s(x)$, επέτρεψε την επιτυχή παραγωγή ενός μεγάλου αριθμού δειγμάτων που προσεγγίζουν με ακρίβεια την επιθυμητή κατανομή στόχο $N(0,1)$. Η υψηλή αναλογία αποφυγής υπολογισμού της $f(x)$ (75.69% των αποδεκτών δειγμάτων) καταδεικνύει την πρακτική αξία της "squeezed" τεχνικής στη βελτίωση της υπολογιστικής απόδοσης.

2.1.2 Υποερώτημα ii: Θεωρητική και Εμπειρική Ανάλυση Απόδοσης

Σε αυτό το υποερώτημα, θα χρησιμοποιήσουμε τα αποτελέσματα της προσομοίωσης από το προηγούμενο υποερώτημα για να εκτιμήσουμε ορισμένα χαρακτηριστικά απόδοσης του αλγορίθμου και θα τα συγκρίνουμε με θεωρητικές τιμές.

Πρώτον, εξετάζουμε την **ολική πιθανότητα αποδοχής** ενός υποψήφιου δείγματος. Από την προσομοίωση είχαμε `count_accepted_f` (που είναι ίσο με `num_samples_target` = 10000) αποδεκτές τιμές, και απαιτήθηκαν συνολικά `total_attempts_f` = 13140 προσπάθειες. Επομένως, η εμπειρική συνολική πιθανότητα αποδοχής, $\hat{P}(\text{accept})$, υπολο-

γίζεται ως ο λόγος των αποδεκτών δειγμάτων προς τις συνολικές προσπάθειες:

$$\hat{P}(\text{accept}) = \frac{\texttt{count\_accepted\_f}}{\texttt{total\_attempts\_f}} = \frac{10000}{13140} \approx 0.761035$$

Για την απλή μέθοδο απόρριψης, η θεωρητική πιθανότητα αποδοχής σε κάθε προσπάθεια είναι $1/M$ [5]. Δεδομένου ότι $M = \sqrt{2e/\pi} \approx 1.315489$, η θεωρητική πιθανότητα αποδοχής είναι $P(\text{accept}) = 1/M \approx 0.7601735$. Είναι σημαντικό να σημειωθεί ότι η "squeezed" μέθοδος έχει την ίδια συνολική πιθανότητα αποδοχής με την απλή μέθοδο απόρριψης, καθώς οι συνθήκες αποδοχής της $U \leq s(Y)/G(Y)$ ή ($U > s(Y)/G(Y)$ ΚΑΙ $U \leq f(Y)/G(Y)$) είναι λογικά ισοδύναμες με τη συνθήκη $U \leq f(Y)/G(Y)$ (αφού $s(Y) \leq f(Y)$). Η διαφορά έγκειται στον αριθμό των υπολογισμών της $f(Y)$ που απαιτούνται. Η εμπειρική μας εκτίμηση (0.761035) βρίσκεται πολύ κοντά στην θεωρητική τιμή (0.7601735), επιβεβαιώνοντας την ορθότητα της προσομοίωσης.

Δεύτερον, αναλύουμε τον **μέσο αριθμό προσπαθειών που απαιτούνται για μία αποδοχή**. Θεωρητικά, ο αριθμός των προσπαθειών μέχρι την πρώτη αποδοχή ακολουθεί γεωμετρική κατανομή με πιθανότητα επιτυχίας $p = P(\text{accept})$. Ο μέσος όρος αυτής της κατανομής, και άρα ο θεωρητικός μέσος αριθμός προσπαθειών, είναι $1/p = M \approx 1.315489$. Εμπειρικά, από την προσομοίωσή μας, αυτός ο μέσος αριθμός εκτιμάται ως:

$$\widehat{E}[\text{attempts per acceptance}] = \frac{\texttt{total\_attempts\_f}}{\texttt{count\_accepted\_f}} = \frac{13140}{10000} = 1.314$$

Και πάλι, παρατηρούμε μια πολύ καλή συμφωνία μεταξύ της εμπειρικής (1.314) και της θεωρητικής (1.315489) τιμής.

Τρίτον, και πιο σημαντικό για την αξιολόγηση της "squeezed" τεχνικής, είναι η **πιθανότητα αποφυγής υπολογισμού της** $f(x)$, δηλαδή η πιθανότητα ένα δείγμα να γίνει αποδεκτό ήδη από τον "φθηνό" Squeeze Test (Βήμα 3 του αλγορίθμου). Εμπειρικά, από τις `count_accepted_f` = 10000 αποδεκτές τιμές, οι `f_calculations_avoided_f` = 7569 προήλθαν από επιτυχή squeeze test. Αυτό αντιστοιχεί σε ποσοστό

$$\hat{P}(\text{f avoided} \mid \text{accepted}) = \frac{\texttt{f\_calculations\_avoided\_f}}{\texttt{count\_accepted\_f}} = \frac{7569}{10000} = 0.7569$$

των αποδεκτών δειγμάτων. Η πιθανότητα μια **τυχαία προσπάθεια** να οδηγήσει σε αποδοχή μέσω του squeeze test ήταν

$$\hat{P}(\text{successful squeeze attempt}) = \frac{\texttt{f\_calculations\_avoided\_f}}{\texttt{total\_attempts\_f}} = \frac{7569}{13140} \approx 0.5760274$$

Η θεωρητική πιθανότητα ένα τυχαίο υποψήφιο $Y \sim g(y)$ (και $U \sim U(0,1)$) να γίνει αποδεκτό ήδη από τον squeeze test δίνεται από την $P(U \leq s(Y)/G(Y))$. Χρησιμοποιώντας τον νόμο της ολικής πιθανότητας (δεσμεύοντας στην τιμή της Y):

$$P(U \leq \frac{s(Y)}{G(Y)}) = \int_{-\infty}^{\infty} P(U \leq \frac{s(Y)}{G(Y)} | Y = y) g(y) dy$$

Επειδή οι U και Y είναι ανεξάρτητες, η $P(U \leq \frac{s(Y)}{G(Y)} | Y = y)$ είναι απλά $P(U \leq \frac{s(y)}{G(y)})$ (η γνώση της τιμής y της Y δεν αλλάζει την κατανομή της U, αλλά καθορίζει την τιμή $\frac{s(y)}{G(y)}$ με την οποία συγκρίνουμε την U). Άρα,

$$P(U \leq \frac{s(Y)}{G(Y)}) = \int_{-\infty}^{\infty} P(U \leq \frac{s(y)}{G(y)}) g(y) dy$$

Τώρα, πρέπει να υπολογίσουμε την $P(U \leq \frac{s(y)}{G(y)})$ για μια σταθερή τιμή $\frac{s(y)}{G(y)}$ (δεδομένου ότι έχουμε δεσμεύσει $Y = y$). Επειδή $U \sim U(0,1)$, έχουμε ότι η αθροιστική συνάρτηση κατανομής της (CDF) είναι $F_U(u) = u$ για $u \in (0,1)$. Επίσης, γνωρίζουμε ότι $0 \leq s(y) \leq f(y)$ και $f(y) \leq G(y)$, άρα $0 \leq s(y) \leq G(y)$. Αυτό σημαίνει ότι ο λόγος $\frac{s(y)}{G(y)}$ ικανοποιεί $0 \leq \frac{s(y)}{G(y)} \leq 1$. Επομένως, για μια δεδομένη τιμή y:

$$P(U \leq \frac{s(y)}{G(y)}) = F_U(\frac{s(y)}{G(y)}) = \frac{s(y)}{G(y)}$$

Αντικαθιστώντας αυτό πίσω στο ολοκλήρωμα:

$$P(U \leq \frac{s(Y)}{G(Y)}) = \int_{-\infty}^{\infty} \frac{s(y)}{G(y)} g(y) dy$$

Αυτό το ολοκλήρωμα είναι εξ ορισμού η αναμενόμενη τιμή της τυχαίας μεταβλητής $\frac{s(Y)}{G(Y)}$ όταν η Y ακολουθεί την κατανομή $g(y)$. Συνολικά έχουμε:

$$P(U \leq \frac{s(Y)}{G(Y)}) = E_Y\left[\frac{s(Y)}{G(Y)}\right] = \int_{-\infty}^{\infty} \frac{s(y)}{Mg(y)} g(y) dy = \frac{1}{M}\int_{-\infty}^{\infty} s(y) dy$$

Υπολογίζοντας το ολοκλήρωμα της συνάρτησης συμπίεσης $s(y) = \frac{1}{\sqrt{2\pi}} \max\{0, 1 - y^2/2\}$:

$$\int_{-\infty}^{\infty} s(y) dy = \int_{-\sqrt{2}}^{\sqrt{2}} \frac{1}{\sqrt{2\pi}}(1 - y^2/2) dy = \frac{1}{\sqrt{2\pi}} \cdot \left[y - y^3/6\right]_{-\sqrt{2}}^{\sqrt{2}} = \frac{4}{3\sqrt{\pi}}$$

Η θεωρητική πιθανότητα επιτυχούς squeeze test είναι τότε:

$$P(\text{successful squeeze}) = \frac{1}{M}\frac{4}{3\sqrt{\pi}} = \frac{\sqrt{\pi}}{\sqrt{2e}}\frac{4}{3\sqrt{\pi}} = \frac{4}{3\sqrt{2e}} \approx 0.5718$$

Η εμπειρική πιθανότητα επιτυχούς squeeze test (≈ 0.5760) είναι και πάλι πολύ κοντά στην αντίστοιχη θεωρητική τιμή (≈ 0.5718), επιβεβαιώνοντας την ορθότητα της υλοποίησης και την αποτελεσματικότητα της μεθόδου.

2.1.3 Υποερώτημα iii: Επιθυμητά Χαρακτηριστικά Κατανομής Εισήγησης

Η επιλογή μιας κατάλληλης κατανομής εισήγησης, $g(x)$, είναι καθοριστικής σημασίας για την απόδοση του αλγορίθμου απόρριψης (τόσο της απλής όσο και της "squeezed" εκδοχής). Μια καλά επιλεγμένη $g(x)$ μπορεί να οδηγήσει σε υψηλή πιθανότητα αποδοχής, μειώνοντας έτσι τον συνολικό αριθμό των απαιτούμενων προσπαθειών και, κατά συνέπεια, το υπολογιστικό κόστος. Τα κυριότερα επιθυμητά χαρακτηριστικά μιας κατανομής εισήγησης $g(x)$ είναι τα εξής:

Πρώτον, η κατανομή εισήγησης $g(x)$ **πρέπει να είναι εύκολο να δειγματοληπτηθεί**. Αυτό είναι θεμελιώδες, καθώς ο αλγόριθμος βασίζεται στην παραγωγή μεγάλου αριθμού υποψήφιων δειγμάτων από την $g(x)$. Αν η ίδια η $g(x)$ είναι δύσκολο να προσομοιωθεί, τότε το όποιο όφελος από τη μέθοδο απόρριψης αναιρείται.

Δεύτερον, το υποστήριγμα (support) της $g(x)$ **πρέπει να καλύπτει το υποστήριγμα της συνάρτησης στόχου** $f(x)$. Δηλαδή, για κάθε x όπου $f(x) > 0$, πρέπει να ισχύει και $g(x) > 0$. Αν υπάρχουν περιοχές όπου η $f(x)$ είναι θετική αλλά

η $g(x)$ είναι μηδέν, τότε ο αλγόριθμος δεν θα μπορέσει ποτέ να παράγει δείγματα από αυτές τις περιοχές της $f(x)$.

Τρίτον, η μορφή της $g(x)$ (ή, ακριβέστερα, της συνάρτησης φακέλου $G(x) = M \cdot g(x)$) **θα πρέπει να προσεγγίζει όσο το δυνατόν καλύτερα τη μορφή της** $f(x)$. Όσο πιο "κοντά" είναι η $g(x)$ στην $f(x)$, τόσο μικρότερη θα είναι η σταθερά M που απαιτείται (ιδανικά $M \to 1$), και κατά συνέπεια, τόσο υψηλότερη θα είναι η πιθανότητα αποδοχής $1/M$. Μια κατανομή εισήγησης που "μιμείται" καλά την $f(x)$ θα οδηγήσει σε λιγότερες απορρίψεις.

Τέλος, ειδικά για την "squeezed" μέθοδο απόρριψης, είναι επιθυμητό να μπορούμε να βρούμε μια **αποτελεσματική συνάρτηση συμπίεσης** $s(x)$ τέτοια ώστε $s(x) \leq f(x)$ και ο λόγος $s(x)/G(x)$ να είναι όσο το δυνατόν μεγαλύτερος, ενώ ο υπολογισμός της $s(x)$ παραμένει σημαντικά "φθηνότερος" από τον υπολογισμό της $f(x)$. Η ύπαρξη μιας τέτοιας "καλής" συνάρτησης συμπίεσης μεγιστοποιεί το όφελος της "squeezed" τεχνικής, μειώνοντας περαιτέρω τον αριθμό των φορών που χρειάζεται να υπολογιστεί η $f(x)$.

Στην περίπτωση της προσομοίωσης από την $N(0,1)$ με πρόταση την διπλή εκθετική, η $g(x)$ είναι εύκολο να δειγματοληπτηθεί, καλύπτει όλο το $\mathbb{R}$ όπως και η $N(0,1)$. Επιπλέον, η ανισότητα που χρησιμοποιήθηκε για την $s(x)$ παρείχε μια χρήσιμη συνάρτηση συμπίεσης.

2.2 Ερώτημα (β): Εκτίμηση Μέσης Τιμής με Στοχαστικές Μεθόδους

Σε αυτό το ερώτημα, θα εστιάσουμε στην εκτίμηση της μέσης τιμής $E[\phi(X)]$ μιας συνάρτησης $\phi(X) = 4/(1+X^2)$, όπου η τυχαία μεταβλητή X ακολουθεί την ομοιόμορφη κατανομή στο διάστημα $(0,1)$, δηλαδή $X \sim U(0,1)$.

Θα συγκρίνουμε δύο υπολογιστικές μεθόδους για την εκτίμηση αυτής της μέσης τιμής:

- Τον κλασικό εκτιμητή Monte Carlo, όπου παράγουμε ένα δείγμα απευθείας από την κατανομή $U(0,1)$.
- Έναν εναλλακτικό εκτιμητή που βασίζεται στη μέθοδο της δειγματοληψίας σπουδαιότητας (Importance Sampling), χρησιμοποιώντας μια κατάλληλα επιλεγμένη συνάρτηση πυκνότητας σπουδαιότητας $g(x) = \frac{1}{3}(4-2x)$ στο διάστημα $[0,1]$.

Για κάθε μέθοδο, θα προσομοιώσουμε 1000 τιμές του αντίστοιχου εκτιμητή (με μέγεθος δείγματος $n = 200$ για κάθε εκτίμηση) και θα υπολογίσουμε το τυπικό σφάλμα των εκτιμήσεων αυτών, προκειμένου να αξιολογήσουμε την αποτελεσματικότητα της δειγματοληψίας σπουδαιότητας στη μείωση της διασποράς του εκτιμητή σε σύγκριση με την απλή Monte Carlo προσέγγιση.

2.2.1 Υποερώτημα i: Κλασικός Εκτιμητής Monte Carlo

Στο παρόν υποερώτημα, επιδιώκουμε να εκτιμήσουμε τη μέση τιμή $E[\phi(X)]$, με $\phi(X) = 4/(1+X^2)$ και η τυχαία μεταβλητή X ακολουθεί την ομοιόμορφη κατανομή στο διάστημα $(0,1)$, δηλαδή $X \sim U(0,1)$. Η συνάρτηση πυκνότητας πιθανότητας (pdf) της X είναι $f(x) = 1$ για $x \in (0,1)$ και 0 αλλού.

Η ζητούμενη μέση τιμή, ας την ονομάσουμε θ, ορίζεται ως το ολοκλήρωμα:

$$\theta = E[\phi(X)] = \int_0^1 \phi(x) f(x) dx = \int_0^1 \frac{4}{1+x^2} \cdot 1 \, dx = 4 \int_0^1 \frac{1}{1+x^2} dx$$

Αυτό το ολοκλήρωμα είναι γνωστό και ισούται με $4[\arctan(x)]_0^1 = 4(\arctan(1) - \arctan(0)) = 4(\pi/4 - 0) = \pi \approx 3.14159$.

Η μέθοδος Monte Carlo [5] παρέχει έναν τρόπο εκτίμησης αυτής της μέσης τιμής μέσω προσομοίωσης. Αν $X_1, X_2, \ldots, X_n$ είναι ένα τυχαίο δείγμα n ανεξάρτητων και ισόνομων παρατηρήσεων από την κατανομή $f(x)$, τότε ο κλασικός Monte Carlo εκτιμητής για το θ, τον οποίο συμβολίζουμε με $\hat{\theta}_1$, δίνεται από τον δειγματικό μέσο όρο των τιμών $\phi(X_i)$:

$$\hat{\theta}_1 = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i)$$

Αυτός ο εκτιμητής είναι αμερόληπτος, δηλαδή $E[\hat{\theta}_1] = \theta$, και η διασπορά του είναι $V[\hat{\theta}_1] = \frac{V[\phi(X)]}{n}$.

Σύμφωνα με την εκφώνηση, θα χρησιμοποιήσουμε μέγεθος δείγματος $n = 200$. Για να εκτιμήσουμε το τυπικό σφάλμα του εκτιμητή $\hat{\theta}_1$, θα προσομοιώσουμε τη διαδικασία εκτίμησης $N_{sim} = 1000$ φορές. Δηλαδή, θα δημιουργήσουμε 1000 διαφορετικά δείγματα μεγέθους $n = 200$ το καθένα, θα υπολογίσουμε την εκτίμηση $\hat{\theta}_{1,k}$ για κάθε ένα από αυτά τα δείγματα ($k = 1, \ldots, 1000$), και στη συνέχεια θα υπολογίσουμε την τυπική απόκλιση αυτών των 1000 εκτιμήσεων. Αυτή η τυπική απόκλιση θα αποτελεί μια εκτίμηση του τυπικού σφάλματος του $\hat{\theta}_1$.

```r
# Define the function phi(x)
phi_x <-function(x) {
  4 / (1 + x^2)
}

# Parameters for the simulation
n_sample_size <-200 # Sample size for each Monte Carlo estimate
N_simulations <-1000 # Number of Monte Carlo estimates to generate

# Vector to store the N_simulations estimates of theta_1
theta_1_estimates <-numeric(N_simulations)
```

Ο παρακάτω κώδικας υλοποιεί τον βρόχο προσομοίωσης. Σε κάθε επανάληψη του βρόχου (από 1 έως $N_{\text{simulations}}$):

1. Δημιουργείται ένα δείγμα $X_1, \ldots, X_n$ μεγέθους `n_sample_size` από την κατανομή $U(0, 1)$.
2. Υπολογίζεται η τιμή της $\phi(X_i)$ για κάθε παρατήρηση στο δείγμα.
3. Υπολογίζεται ο μέσος όρος αυτών των τιμών, ο οποίος είναι η εκτίμηση $\hat{\theta}_{1,k}$ για τη συγκεκριμένη προσομοίωση, και αποθηκεύεται.

```r
set.seed(2024) # For reproducibility

# Loop to generate N_simulations estimates of theta_1
for (k in 1:N_simulations) {
  # Step 1: Generate a sample of size n_sample_size from U(0,1)
  X_sample <-runif(n_sample_size, min =0, max =1)

  # Step 2: Calculate phi(X_i) for each observation in the sample
  phi_X_values <-phi_x(X_sample)

  # Step 3: Calculate the Monte Carlo estimate for this simulation
  theta_1_estimates[k] <-mean(phi_X_values)
}
```

Αφού έχουμε τις $N_{simulations}$ εκτιμήσεις $(\hat{\theta}_{1,1}, \hat{\theta}_{1,2}, \ldots, \hat{\theta}_{1,N_{simulations}})$, το τυπικό σφάλμα του εκτιμητή $\hat{\theta}_1$ μπορεί να εκτιμηθεί από την τυπική απόκλιση αυτών των τιμών:

$$\widehat{SE}(\hat{\theta}_1) = \text{sd}(\hat{\theta}_{1,1}, \ldots, \hat{\theta}_{1,N_{simulations}}) = \sqrt{\frac{1}{N_{simulations} - 1} \sum_{k=1}^{N_{simulations}} (\hat{\theta}_{1,k} - \bar{\hat{\theta}}_1)^2}$$

όπου $\bar{\hat{\theta}}_1 = \frac{1}{N_{simulations}} \sum_{k=1}^{N_{simulations}} \hat{\theta}_{1,k}$ είναι ο μέσος όρος των προσομοιωμένων εκτιμήσεων.

```
# Calculate the standard deviation of the N_simulations estimates
se_theta_1_hat <-sd(theta_1_estimates)

# Print the estimated standard error
print(paste("Estimated Standard Error of theta_1_hat (classic MC):", round(se_theta_1_hat, 6)))

# For comparison, we also print the mean of our estimates, it should be close to the true value pi
print(paste("Mean of theta_1_hat estimates:", round(mean(theta_1_estimates), 6)))
print(paste("True value (pi):", round(pi, 6)))
```

Η εκτέλεση του παραπάνω κώδικα R, με $n = 200$ παρατηρήσεις ανά εκτίμηση Monte Carlo και $N_{sim} = 1000$ επαναλήψεις της προσομοίωσης, παρήγαγε τα ακόλουθα αποτελέσματα:

- **Εκτιμώμενο Τυπικό Σφάλμα** του $\hat{\theta}_1$: 0.044458
- **Μέσος όρος** των 1000 εκτιμήσεων $\hat{\theta}_1$: 3.142628

Ο μέσος όρος των εκτιμήσεων (3.142628) είναι, όπως αναμενόταν λόγω της αμεροληψίας του εκτιμητή Monte Carlo, πολύ κοντά στην πραγματική τιμή του ολοκληρώματος, $\theta = \pi \approx 3.141593$. Το εκτιμώμενο τυπικό σφάλμα, 0.044458, ποσοτικοποιεί την τυπική απόκλιση των εκτιμήσεων $\hat{\theta}_1$ γύρω από την πραγματική τιμή θ για δείγματα μεγέθους $n = 200$. Αυτή η τιμή τυπικού σφάλματος θα χρησιμεύσει ως βάση σύγκρισης με το τυπικό σφάλμα που θα προκύψει από τη μέθοδο της δειγματοληψίας σπουδαιότητας στο επόμενο υποερώτημα.

2.2.2 Υποερώτημα ii: Εκτίμηση με Δειγματοληψία Σπουδαιότητας

Στο προηγούμενο υποερώτημα, εκτιμήσαμε το $\theta = E_f[\phi(X)] = \int \phi(x) f(x) dx$ χρησιμοποιώντας τον κλασικό Monte Carlo εκτιμητή, όπου $X \sim f(x) \equiv U(0,1)$. Η μέθοδος της Δειγματοληψίας Σπουδαιότητας (Importance Sampling) [5] προσφέρει έναν εναλλακτικό τρόπο εκτίμησης του θ, ο οποίος μπορεί, υπό κατάλληλες συνθήκες, να μειώσει σημαντικά τη διασπορά του εκτιμητή.

Η ιδέα είναι να ξαναγράψουμε το ολοκλήρωμα ως προς μια άλλη συνάρτηση πυκνότητας πιθανότητας $g(x)$, την οποία ονομάζουμε **συνάρτηση σπουδαιότητας (importance function)**. Η $g(x)$ πρέπει να είναι τέτοια ώστε $g(x) > 0$ όποτε $f(x)\phi(x) \neq 0$.

$$\theta = \int \phi(x) f(x) dx = \int \frac{\phi(x) f(x)}{g(x)} g(x) dx = E_g \left[\frac{\phi(X) f(X)}{g(X)} \right]$$

Έτσι, αν $X_1, X_2, \ldots, X_n$ είναι ένα τυχαίο δείγμα n ανεξάρτητων και ισόνομων παρατηρήσεων από την κατανομή $g(x)$, τότε ένας νέος εκτιμητής για το θ, τον οποίο συμβολίζουμε με $\hat{\theta}_2$, δίνεται από:

$$\hat{\theta}_2 = \frac{1}{n}\sum_{i=1}^{n} w(X_i)\phi(X_i) \quad \text{όπου} \quad w(X_i) = \frac{f(X_i)}{g(X_i)}$$

Τα $w(X_i)$ ονομάζονται **βάρη σπουδαιότητας (importance weights)**. Ο εκτιμητής $\hat{\theta}_2$ είναι επίσης αμερόληπτος, $E[\hat{\theta}_2] = \theta$. Η διασπορά του είναι $V[\hat{\theta}_2] = \frac{1}{n}V_g\left[\frac{\phi(X)f(X)}{g(X)}\right]$. Η επιλογή της $g(x)$ γίνεται με στόχο τη μείωση αυτής της διασποράς. Ιδανικά, θα θέλαμε $g(x) \propto |\phi(x)f(x)|$.

Στην περίπτωσή μας:

- $\phi(x) = 4/(1+x^2)$
- $f(x) = 1$ για $x \in (0,1)$ (δηλαδή $U(0,1)$)
- $g(x) = \frac{1}{3}(4-2x)$ για $x \in [0,1]$. (Η οποία είναι μια έγκυρη pdf στο $[0,1]$).

Τα βάρη σπουδαιότητας θα είναι:

$$w(x) = \frac{f(x)}{g(x)} = \frac{1}{\frac{1}{3}(4-2x)} = \frac{3}{4-2x}$$

Και ο εκτιμητής $\hat{\theta}_2$ θα είναι:

$$\hat{\theta}_2 = \frac{1}{n}\sum_{i=1}^{n} \frac{3}{4-2X_i}\cdot\frac{4}{1+X_i^2} = \frac{1}{n}\sum_{i=1}^{n}\frac{12}{(4-2X_i)(1+X_i^2)} \quad \text{όπου } X_i \sim g(x)$$

Για την προσομοίωση δειγμάτων από την $g(x)$, θα χρειαστεί να χρησιμοποιήσουμε τη μέθοδο της αντιστροφής. Η CDF της $g(x)$ είναι:

$$G(x) = \int_0^x \frac{1}{3}(4-2t)dt = \frac{1}{3}(4x - x^2) \quad \text{για } x \in [0,1]$$

Θέτουμε $U = G(X) = \frac{1}{3}(4X - X^2)$, όπου $U \sim U(0,1)$ και λύνουμε ως προς X:

$$3U = 4X - X^2 \implies X^2 - 4X + 3U = 0 \quad \text{για } X \in [0,1]$$

Αυτή είναι μια δευτεροβάθμια εξίσωση ως προς X: $aX^2 + bX + c = 0$ με $a = 1, b = -4, c = 3U$.

Οι λύσεις είναι $X = \frac{-b \pm \sqrt{b^2-4ac}}{2a} = \frac{4 \pm \sqrt{16-4(1)(3U)}}{2} = \frac{4 \pm \sqrt{16-12U}}{2} = 2 \pm \sqrt{4-3U}$. Επειδή θέλουμε $X \in [0,1]$, και γνωρίζοντας ότι $U \in [0,1]$ κρατάμε την λύση που είναι συμβατή:

$$X = G^{-1}(U) = 2 - \sqrt{4-3U}$$

```r
# Importance sampling density g(x) =(1/3)*(4-2x) for x in [0,1]
g_importance <-function(x) {
  ifelse(x >= 0 & x <= 1, (1/3) * (4 - 2*x), 0)
}

# Function to generate samples from g_importance(x) using inversion method
generate_from_g_importance <-function() {
  u <-runif(1) # U ~ U(0,1)
  x_sample <-2 - sqrt(4 - 3*u)
  return(x_sample)
}

# Importance weight function w(x) =f(x)/g(x)
importance_weight <-function(x) {
  g_val <-g_importance(x)
  return(1 / g_val)
}

# The term to average in Importance Sampling is psi(x) =phi(x) * f(x) / g(x) =phi(x) * w(x)
psi_x <-function(x) {
  phi_x(x) * importance_weight(x)
}

# Parameters for the simulation (same as before for comparison)
n_sample_size <-200
N_simulations <-1000

# Vector to store the N_simulations estimates of theta_2
theta_2_estimates <-numeric(N_simulations)
```

Ο παρακάτω κώδικας υλοποιεί τον βρόχο προσομοίωσης για τον εκτιμητή $\hat{\theta}_2$. Σε κάθε επανάληψη:

1. Δημιουργείται ένα δείγμα $X_1, \ldots, X_n$ μεγέθους `n_sample_size` από την κατανομή σπουδαιότητας $g(x)$ χρησιμοποιώντας τη μέθοδο της αντιστροφής.
2. Για κάθε X_i στο δείγμα, υπολογίζεται η τιμή $\psi(X_i) = \phi(X_i)\frac{f(X_i)}{g(X_i)}$.
3. Υπολογίζεται ο μέσος όρος αυτών των τιμών $\psi(X_i)$, ο οποίος είναι η εκτίμηση $\hat{\theta}_{2,k}$ για τη συγκεκριμένη προσομοίωση.

```r
set.seed(29) # For reproducibility

# Loop to generate N_simulations estimates of theta_2
for (k in 1:N_simulations) {
  # Step 1: Generate a sample of size n_sample_size from g_importance(x)
  X_sample_g <-replicate(n_sample_size, generate_from_g_importance())

  # Step 2: Calculate psi(X_i) for each observation in the sample from g
  psi_X_values <-psi_x(X_sample_g)

  # Step 3: Calculate the Importance Sampling estimate for this simulation
  theta_2_estimates[k] <-mean(psi_X_values)
}
```

Αφού έχουμε τις $N_{simulations}$ εκτιμήσεις $(\hat{\theta}_{2,1}, \ldots, \hat{\theta}_{2,N_{simulations}})$, το τυπικό σφάλμα του εκτιμητή $\hat{\theta}_2$ εκτιμάται από την τυπική απόκλιση αυτών των τιμών:

$$\widehat{SE}(\hat{\theta}_2) = \text{sd}(\hat{\theta}_{2,1}, \ldots, \hat{\theta}_{2,N_{simulations}})$$

Θα συγκρίνουμε αυτό το τυπικό σφάλμα με το $\widehat{SE}(\hat{\theta}_1)$ που υπολογίστηκε στο προηγούμενο υποερώτημα για να αξιολογήσουμε αν η χρήση της συγκεκριμένης συνάρτησης σπουδαιότητας $g(x)$ οδήγησε σε βελτίωση (μείωση) της διασποράς του εκτιμητή.

```r
# Calculate the standard deviation of the N_simulations IS estimates
se_theta_2_hat <-sd(theta_2_estimates)
print(paste("Estimated Standard Error of theta_2_hat (Importance Sampling):", round(se_theta_2_hat, 6)))

# For comparison, also print the mean of our IS estimates
print(paste("Mean of theta_2_hat estimates:", round(mean(theta_2_estimates), 6)))

# Compare the standard errors
if (se_theta_2_hat < se_theta_1_hat) {
  reduction_percentage <-(1 - (se_theta_2_hat / se_theta_1_hat)) * 100
  print(paste("Importance Sampling resulted in a reduction of SE by approx.",
              round(reduction_percentage, 2), "% compared to classic MC."))
  print("This suggests the chosen g(x) was helpful in reducing variance.")
} else {
  print("Importance Sampling did NOT result in a reduction of SE compared to classic MC for this g(x).")
}
```

Η εκτέλεση του κώδικα R για τον εκτιμητή Δειγματοληψίας Σπουδαιότητας παρήγαγε τα ακόλουθα αποτελέσματα:

- **Εκτιμώμενο Τυπικό Σφάλμα** του $\hat{\theta}_2$: 0.005729
- **Μέσος όρος** των 1000 εκτιμήσεων $\hat{\theta}_2$: 3.141547

Όπως και με τον κλασικό Monte Carlo εκτιμητή, ο μέσος όρος των εκτιμήσεων μέσω Δειγματοληψίας Σπουδαιότητας (3.141547) είναι πολύ κοντά στην πραγματική τιμή $\theta = \pi \approx 3.141593$, επιβεβαιώνοντας την αμεροληψία και αυτής της μεθόδου.

Το πιο σημαντικό εύρημα, ωστόσο, αφορά τη σύγκριση των τυπικών σφαλμάτων. Το τυπικό σφάλμα του εκτιμητή Δειγματοληψίας Σπουδαιότητας ($\widehat{SE}(\hat{\theta}_2) \approx 0.005729$) είναι σημαντικά μικρότερο από το τυπικό σφάλμα του κλασικού Monte Carlo εκτιμητή ($\widehat{SE}(\hat{\theta}_1) \approx 0.044458$) που υπολογίστηκε στο προηγούμενο υποερώτημα (2.β.i). Συγκεκριμένα, η χρήση της Δειγματοληψίας Σπουδαιότητας με την προτεινόμενη $g(x)$ οδήγησε σε μια **μείωση του τυπικού σφάλματος κατά περίπου** 87.11% σε σύγκριση με την κλασική μέθοδο Monte Carlo.

Αυτή η αξιοσημείωτη μείωση του τυπικού σφάλματος του εκτιμητή υποδηλώνει ότι η επιλεγμένη συνάρτηση σπουδαιότητας $g(x) = \frac{1}{3}(4 - 2x)$ είναι κατάλληλη για το συγκεκριμένο πρόβλημα. Η $g(x)$ φαίνεται να "μοιάζει" περισσότερο με τη μορφή της συνάρτησης $\phi(x) = \frac{4}{1+x^2}$ στο διάστημα $[0, 1]$ σε σχέση με την ομοιόμορφη κατανομή $f(x)$ που χρησιμοποιήθηκε στην κλασική Monte Carlo. Τοποθετώντας περισσότερα σημεία δειγματοληψίας σε περιοχές όπου η συνάρτηση $\phi(x)$ έχει μεγαλύτερη "σπουδαιότητα" (δηλαδή, μεγαλύτερη συνεισφορά στην τιμή του ολοκληρώματος) και ταυτόχρονα σταθμίζοντας το αποτέλεσμα με τα βάρη σπουδαιότητας $w(x) = \frac{f(x)}{g(x)}$, η Δειγματοληψία Σπουδαιότητας επιτυγχάνει μια πιο ακριβή εκτίμηση με την ίδια συνολική υπολογιστική προσπάθεια (ίδιο n). Η μεγάλη μείωση της διασποράς σημαίνει ότι για να επιτύχουμε την ίδια ακρίβεια εκτίμησης, η Δειγματοληψία Σπουδαιότητας θα απαιτούσε σημαντικά μικρότερο μέγεθος δείγματος n σε σχέση με την κλασική Monte Carlo.

3 Άσκηση 3

3.1 Εκτίμηση Πιθανότητας Μίξης Εκθετικών Κατανομών με τον Αλγόριθμο ΕΜ

Στην παρούσα άσκηση, θα ασχοληθούμε με ένα πρόβλημα εκτίμησης παραμέτρου σε ένα μοντέλο μίξης κατανομών. Συγκεκριμένα, θεωρούμε μια τυχαία μεταβλητή X η οποία προκύπτει από μια μίξη δύο εκθετικών κατανομών, με την επιλογή της συγκεκριμένης εκθετικής κατανομής να καθορίζεται από μια διωνυμική τυχαία μεταβλητή $Z \sim \text{Bernoulli}(p)$. Η παράμετρος p, η πιθανότητα επιτυχίας της Bernoulli, είναι άγνωστη και αποτελεί το αντικείμενο της εκτίμησης. Διαθέτουμε παρατηρήσεις μόνο από την τυχαία μεταβλητή X, ενώ η τιμή της Z (ποια εκθετική κατανομή "ενεργοποιήθηκε" για κάθε X_i) παραμένει μη παρατηρήσιμη (λανθάνουσα μεταβλητή).

Για την εκτίμηση της άγνωστης παραμέτρου p σε αυτό το πλαίσιο με λανθάνουσες μεταβλητές, θα χρησιμοποιήσουμε τον αλγόριθμο **Expectation-Maximization (EM)** [6]. Ο αλγόριθμος ΕΜ είναι μια επαναληπτική διαδικασία σχεδιασμένη για την εύρεση εκτιμήσεων μέγιστης πιθανοφάνειας όταν το μοντέλο περιλαμβάνει μη παρατηρήσιμα δεδομένα. Ο αλγόριθμος εναλλάσσεται μεταξύ δύο βημάτων: του βήματος Ε (Expectation), όπου υπολογίζεται η αναμενόμενη τιμή της λογαριθμικής πιθανοφάνειας των πλήρων δεδομένων (παρατηρούμενων και λανθάνοντων) ως προς την τρέχουσα εκτίμηση των παραμέτρων, και του βήματος Μ (Maximization), όπου οι παράμετροι επανεκτιμώνται μεγιστοποιώντας αυτή την αναμενόμενη λογαριθμική πιθανοφάνεια.

Θα αναπτύξουμε θεωρητικά τα βήματα του αλγορίθμου ΕΜ για το συγκεκριμένο πρόβλημα μίξης και στη συνέχεια θα υλοποιήσουμε τον αλγόριθμο στην `R` για να εκτιμήσουμε την p από ένα δοθέν σύνολο δεδομένων.

Ξεκινάμε θεωρώντας ένα σύνολο n ανεξάρτητων και ισόνομων παρατηρήσεων $X_1, X_2, \ldots, X_n$. Κάθε παρατήρηση X_i προέρχεται από μια μίξη δύο εκθετικών κατανομών. Εισάγουμε μια λανθάνουσα διωνυμική μεταβλητή $Z_i \sim \text{Bernoulli}(p)$ για κάθε X_i, όπου p είναι η άγνωστη πιθανότητα επιτυχίας που θέλουμε να εκτιμήσουμε. Η σχέση μεταξύ X_i και Z_i ορίζεται ως εξής:

- Αν $Z_i = 1$ (με πιθανότητα p), τότε $X_i \sim \text{Exp}(\lambda_1)$, όπου $\lambda_1 = 1$.
- Αν $Z_i = 0$ (με πιθανότητα $1 - p$), τότε $X_i \sim \text{Exp}(\lambda_0)$, όπου $\lambda_0 = 5$.

Οι συναρτήσεις πυκνότητας πιθανότητας των δύο εκθετικών κατανομών είναι:

- $f_1(x) = \lambda_1 e^{-\lambda_1 x} = e^{-x}$ για $x \geq 0$ (όταν $Z_i = 1$)
- $f_0(x) = \lambda_0 e^{-\lambda_0 x} = 5e^{-5x}$ για $x \geq 0$ (όταν $Z_i = 0$)

Τα παρατηρημένα δεδομένα είναι $\mathbf{X} = (X_1, \ldots, X_n)$ και οι λανθάνουσες μεταβλητές είναι $\mathbf{Z} = (Z_1, \ldots, Z_n)$. Επομένως τα πλήρη δεδομένα είναι $(\mathbf{X}, \mathbf{Z})$. Λόγω της ανεξαρτησίας των (X_i, Z_i), η συνάρτηση πιθανοφάνειας των πλήρων δεδομένων $(\mathbf{X}, \mathbf{Z})$ για την παράμετρο p είναι:

$$L(p; \mathbf{X}, \mathbf{Z}) = P(\mathbf{X}, \mathbf{Z}|p) = \prod_{i=1}^{n} P(X_i, Z_i|p) = \prod_{i=1}^{n} P(X_i|Z_i, p)P(Z_i|p)$$

Δεδομένου ότι η $P(X_i|Z_i, p)$ δεν εξαρτάται από το p (εξαρτάται μόνο από τις γνωστές λ_1, λ_0), και $P(Z_i|p) = p^{Z_i}(1-p)^{1-Z_i}$, έχουμε:

$$L(p; \mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} [f_1(X_i)p]^{Z_i} [f_0(X_i)(1-p)]^{1-Z_i} \implies$$

$$L(p; \mathbf{X}, \mathbf{Z}) = \left(\prod_{i=1}^{n} f_1(X_i)^{Z_i} f_0(X_i)^{1-Z_i} \right) p^{\sum Z_i} (1-p)^{n-\sum Z_i}$$

Η λογαριθμική πιθανοφάνεια των πλήρων δεδομένων είναι:

$$\ell(p; \mathbf{X}, \mathbf{Z}) = \log L(p; \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{n} [Z_i \log f_1(X_i) + (1 - Z_i) \log f_0(X_i)] + \left(\sum_{i=1}^{n} Z_i \right) \log p + \left(n - \sum_{i=1}^{n} Z_i \right) \log(1-p)$$

Ο αλγόριθμος ΕΜ αποτελείται από δύο βήματα που επαναλαμβάνονται:

1. **Βήμα Ε (Expectation Step):** Στο $(r+1)$-οστό βήμα, υπολογίζουμε την αναμενόμενη τιμή της λογαριθμικής πιθανοφάνειας των πλήρων δεδομένων, $\ell(p; \mathbf{X}, \mathbf{Z})$, ως προς την υπό συνθήκη κατανομή των λανθανουσών μεταβλητών $\mathbf{Z}$ δεδομένων των παρατηρημένων δεδομένων $\mathbf{X}$ και της τρέχουσας εκτίμησης της παραμέτρου $p^{(r)}$ από το προηγούμενο βήμα. Ορίζουμε τη συνάρτηση $Q(p|p^{(r)})$:

$$Q(p|p^{(r)}) = E_{\mathbf{Z}|\mathbf{X}, p^{(r)}}[\ell(p; \mathbf{X}, \mathbf{Z})]$$

Λόγω της γραμμικότητας της αναμενόμενης τιμής και της ανεξαρτησίας των (X_i, Z_i), αρκεί να υπολογίσουμε την $E[Z_i|X_i, p^{(r)}]$.

$$E[Z_i|X_i, p^{(r)}] = 1 \cdot P(Z_i = 1|X_i, p^{(r)}) + 0 \cdot P(Z_i = 0|X_i, p^{(r)}) = P(Z_i = 1|X_i, p^{(r)})$$

Χρησιμοποιώντας τον κανόνα του Bayes:

$$P(Z_i = 1|X_i, p^{(r)}) = \frac{P(X_i|Z_i = 1, p^{(r)}) P(Z_i = 1|p^{(r)})}{P(X_i|Z_i = 1, p^{(r)}) P(Z_i = 1|p^{(r)}) + P(X_i|Z_i = 0, p^{(r)}) P(Z_i = 0|p^{(r)})} \implies$$

$$P(Z_i = 1|X_i, p^{(r)}) = \frac{f_1(X_i) p^{(r)}}{f_1(X_i) p^{(r)} + f_0(X_i)(1 - p^{(r)})}$$

Ας ονομάσουμε αυτή την υπό συνθήκη αναμενόμενη τιμή (ή πιθανότητα) $\gamma_i^{(r+1)}$:

$$\gamma_i^{(r+1)} = E[Z_i|X_i, p^{(r)}] = \frac{p^{(r)} f_1(X_i)}{p^{(r)} f_1(X_i) + (1 - p^{(r)}) f_0(X_i)}$$

Επίσης, $E[1 - Z_i|X_i, p^{(r)}] = 1 - \gamma_i^{(r+1)}$. Αντικαθιστώντας τις αναμενόμενες τιμές των Z_i στην $\ell(p; \mathbf{X}, \mathbf{Z})$:

$$Q(p|p^{(r)}) = \sum_{i=1}^{n} [\gamma_i^{(r+1)} \log f_1(X_i) + (1 - \gamma_i^{(r+1)}) \log f_0(X_i)] + \left(\sum_{i=1}^{n} \gamma_i^{(r+1)} \right) \log p + \left(n - \sum_{i=1}^{n} \gamma_i^{(r+1)} \right) \log(1-p)$$

2. **Βήμα Μ (Maximization Step)**: Στο βήμα Μ, βρίσκουμε την τιμή $p^{(r+1)}$ που μεγιστοποιεί την $Q(p|p^{(r)})$ ως προς p. Για να το κάνουμε αυτό, παραγωγίζουμε την $Q(p|p^{(r)})$ ως προς p και εξισώνουμε την παράγωγο με το μηδέν. Μόνο οι δύο τελευταίοι όροι της $Q(p|p^{(r)})$ εξαρτώνται από το p:

$$\frac{\partial Q(p|p^{(r)})}{\partial p} = \frac{\sum_{i=1}^{n} \gamma_i^{(r+1)}}{p} - \frac{n - \sum_{i=1}^{n} \gamma_i^{(r+1)}}{1-p}$$

Θέτοντας την παράγωγο ίση με μηδέν:

$$\frac{\sum_{i=1}^{n} \gamma_i^{(r+1)}}{p} = \frac{n - \sum_{i=1}^{n} \gamma_i^{(r+1)}}{1-p} \implies$$

$$(1-p)\sum_{i=1}^{n} \gamma_i^{(r+1)} = p\left(n - \sum_{i=1}^{n} \gamma_i^{(r+1)}\right) \implies$$

$$\sum_{i=1}^{n} \gamma_i^{(r+1)} - p\sum_{i=1}^{n} \gamma_i^{(r+1)} = np - p\sum_{i=1}^{n} \gamma_i^{(r+1)} \implies$$

$$\sum_{i=1}^{n} \gamma_i^{(r+1)} = np$$

Λύνοντας ως προς p, παίρνουμε την ενημερωμένη εκτίμηση $p^{(r+1)}$:

$$p^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i^{(r+1)}$$

Δηλαδή, η νέα εκτίμηση $p^{(r+1)}$ είναι ο μέσος όρος των υπό συνθήκη πιθανοτήτων $\gamma_i^{(r+1)}$ (οι οποίες υπολογίστηκαν χρησιμοποιώντας την $p^{(r)}$).

Θα κάνουμε τώρα μια σύνοψη του αλγορίθμου ΕΜ.

Σύνοψη Αλγορίθμου ΕΜ:

1. **Αρχικοποίηση**: Επιλέγουμε μια αρχική τιμή $p^{(0)}$ (π.χ., $p^{(0)} = 0.5$).
2. **Επανάληψη (για $r = 0, 1, 2, \ldots$ μέχρι σύγκλισης)**:
   - **Βήμα Ε**: Για κάθε παρατήρηση $i = 1, \ldots, n$, υπολογίζουμε:

$$\gamma_i^{(r+1)} = \frac{p^{(r)} f_1(X_i)}{p^{(r)} f_1(X_i) + (1-p^{(r)}) f_0(X_i)}$$

   - **Βήμα Μ**: Ενημερώνουμε την εκτίμηση της παραμέτρου:

$$p^{(r+1)} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i^{(r+1)}$$

3. **Κριτήριο Τερματισμού**: Σταματάμε όταν η διαφορά μεταξύ διαδοχικών εκτιμήσεων $p^{(r+1)}$ και $p^{(r)}$ είναι αρκετά μικρή. Η εκφώνηση ζητά:
$$\left|p^{(r+1)} - p^{(r)}\right| \leq 10^{-10}$$

Θα προχωρήσουμε τώρα στην υλοποίηση του αλγορίθμου ΕΜ στην R.

```
# Load Data
data_em <-readRDS("data3em.rds")

# Define the density functions f0(x) and f1(x)
f1_x <-function(x_val) {
  dexp(x, rate =1)
}

f0_x <-function(x_val) {
  dexp(x, rate =5)
}

# For convenience, assign data to X_obs and get n
X_obs <-data_em
n_em <-length(X_obs)
```

Θα αρχικοποιήσουμε την παράμετρο p (π.χ. $p^{(0)} = 0.5$) και θα ξεκινήσουμε τον επαναληπτικό αλγόριθμο ΕΜ. Σε κάθε επανάληψη, θα υπολογίζουμε τις τιμές $\gamma_i^{(r+1)}$ (Βήμα Ε) και στη συνέχεια θα ενημερώνουμε την εκτίμηση $p^{(r+1)}$ (Βήμα Μ). Ο βρόχος θα συνεχίζεται μέχρι να ικανοποιηθεί το κριτήριο σύγκλισης $|p^{(r+1)} - p^{(r)}| \leq 10^{-10}$.

```
# EM Algorithm Implementation

# Initialization
p_current <-0.5 # Initial guess for p
max_iterations <-100000 # Maximum number of iterations to prevent infinite loops
tolerance <-1e-10 # Convergence tolerance |p_new - p_old|
iterations <-0
converged <-FALSE
p_history <-numeric(max_iterations + 1) # To store the history of p estimates
p_history[1] <-p_current

# EM Iterations
for (r in 1:max_iterations) {
  iterations <-r
  p_old <-p_current

  # --- E-Step: Calculate gamma_i values ---

  f1_X_obs <-f1_x(X_obs)
  f0_X_obs <-f0_x(X_obs)

  numerator_gamma <-p_old * f1_X_obs
  denominator_gamma <-(p_old * f1_X_obs) + ((1 - p_old) * f0_X_obs)
  gamma_i <-numerator_gamma / denominator_gamma
```

```
  # --- M-Step: Update p ---
  p_current <-sum(gamma_i) / n_em
  p_history[r + 1] <-p_current

  # Check for convergence
  if (abs(p_current - p_old) < tolerance) {
    converged <-TRUE
    break # Exit loop
  }
}

# Store the final history of p up to the point of convergence (or max_iterations)
p_history <-p_history[1:(iterations + 1)]

# Output Results
if (converged) {
  cat(paste("EM algorithm converged in", iterations, "iterations.\n"))
  cat(paste("Final estimate for p (p_hat_EM):", round(p_current, 12), "\n"))
} else {
  cat(paste("EM algorithm did NOT converge after", max_iterations, "iterations.\n"))
  cat(paste("Last estimate for p:", round(p_current, 12), "\n"))
}
```

Η εκτέλεση του αλγορίθμου οδήγησε στα ακόλουθα αποτελέσματα:

- Ο αλγόριθμος ΕΜ σύγκλινε μετά από 45 επαναλήψεις
- Η τελική εκτίμηση για την παράμετρο p (εκτίμηση μέγιστης πιθανοφάνειας μέσω ΕΜ) βρέθηκε να είναι $\hat{p}_{EM} \approx 0.201536038633$.

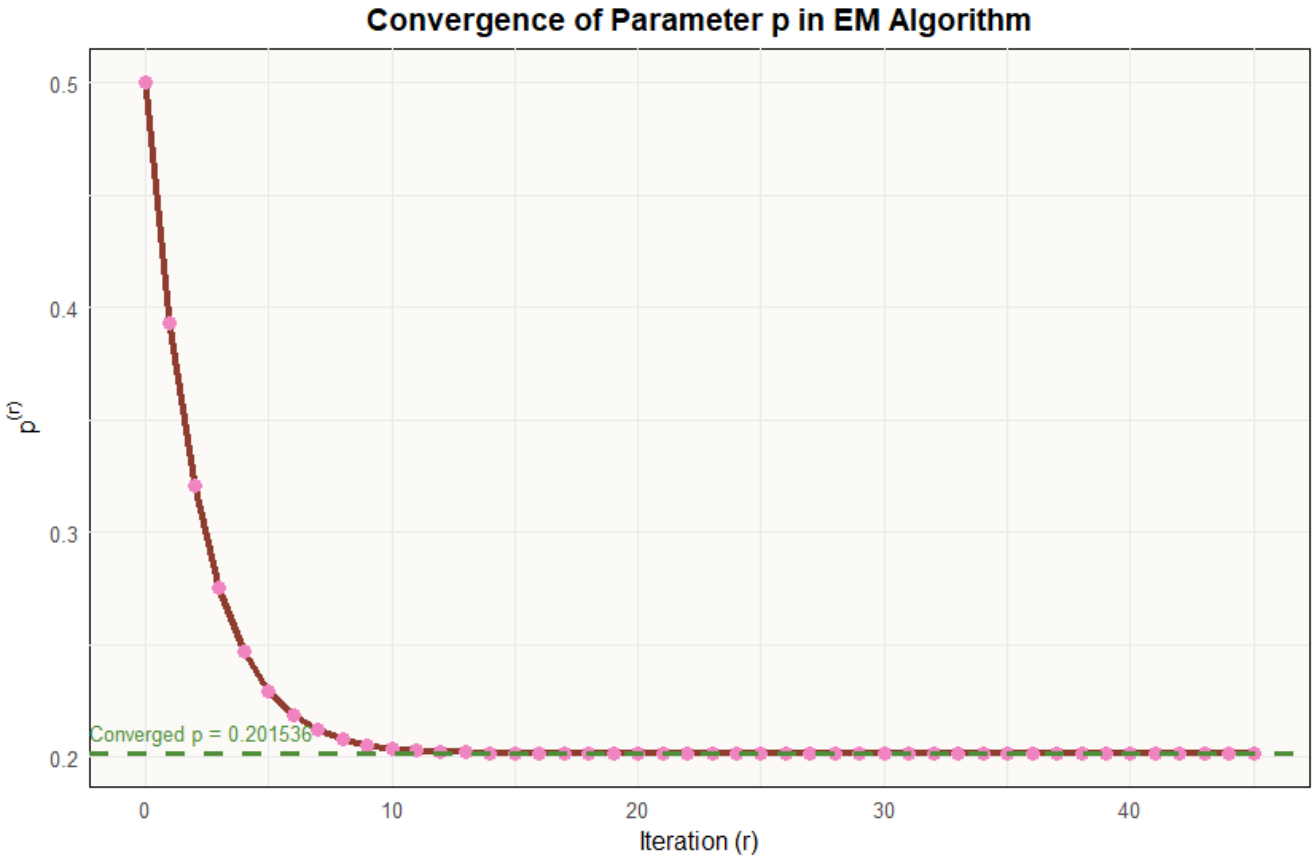
Η σχετικά γρήγορη σύγκλιση (σε 45 επαναλήψεις) υποδηλώνει ότι ο αλγόριθμος λειτούργησε αποτελεσματικά για το συγκεκριμένο πρόβλημα και σύνολο δεδομένων. Η τελική εκτιμώμενη τιμή $\hat{p}_{EM} \approx 0.2015$ υποδεικνύει ότι, με βάση τα παρατηρημένα δεδομένα X_i, είναι πιο πιθανό μια παρατήρηση να προέρχεται από την εκθετική κατανομή με παράμετρο $\lambda_0 = 5$ (που αντιστοιχεί σε $Z_i = 0$, με πιθανότητα $1-p \approx 0.7985$) παρά από την εκθετική κατανομή με παράμετρο $\lambda_1 = 1$ (που αντιστοιχεί σε $Z_i = 1$, με πιθανότητα $p \approx 0.2015$).

Για την οπτική επιβεβαίωση της σύγκλισης, κατασκευάστηκε το Σχήμα 7, το οποίο απεικονίζει την τιμή της εκτιμώμενης παραμέτρου $p^{(r)}$ σε κάθε επανάληψη r του αλγορίθμου ΕΜ.

```
# Create a data frame for plotting
convergence_df <-data.frame(Iteration =0:iterations, P_Estimate =p_history)

# Create the convergence plot using ggplot2
convergence_plot_em <-ggplot(convergence_df, aes(x =Iteration, y =P_Estimate)) +
  geom_line(color ="#8E3C2E", linewidth =1.5) +
  geom_point(color ="#F084C1", size =2.8, shape =16) +
  geom_hline(yintercept =p_current, linetype ="dashed", color ="#4F8E38", linewidth =1.2) +
  annotate("text", x =6.8, y =p_current * 1.05,
           label =paste("Converged p =", round(p_current, 6)),
           color ="#4F8E38", size =3.5, hjust =1) +
  labs(title ="Convergence of Parameter p in EM Algorithm",
       x ="Iteration (r)",
       y =expression(p^"(r)")) +
  theme_minimal(base_size =12) +
```

```
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
        panel.background =element_rect(fill ="#FBFAF7"))

# Print the plot
print(convergence_plot_em)
```

Σχήμα 7: Σύγκλιση της εκτίμησης $p^{(r)}$ του Αλγορίθμου ΕΜ για την πιθανότητα p



Από το Σχήμα 7, παρατηρούμε ότι η τιμή της $p^{(r)}$ ξεκινά από την αρχική τιμή 0.5 και μειώνεται γρήγορα κατά τις πρώτες $10-15$ επαναλήψεις. Στη συνέχεια, η σύγκλιση επιβραδύνεται, με την τιμή της $p^{(r)}$ να σταθεροποιείται προοδευτικά γύρω από την τελική εκτιμώμενη τιμή ≈ 0.201536, επιβεβαιώνοντας την ομαλή προσέγγιση του αλγορίθμου προς το σημείο (τοπικού) μεγίστου της πιθανοφάνειας.

4 Άσκηση 4

Η τέταρτη άσκηση της παρούσας εργασίας εστιάζει στο κρίσιμο πρόβλημα της επιλογής των βέλτιστων επεξηγηματικών μεταβλητών στο πλαίσιο ενός μοντέλου πολλαπλής γραμμικής παλινδρόμησης [7]. Για τον σκοπό αυτό, θα αξιοποιήσουμε το σύνολο δεδομένων "Boston housing", το οποίο παρέχει αναλυτικές πληροφορίες για $n = 506$ προάστια της Βοστώνης. Κεντρικό αντικείμενο της ανάλυσής μας αποτελεί η μεταβλητή απόκρισης `medv`, η οποία αντικατοπτρίζει τη μέση αξία των ιδιοκατοικούμενων κατοικιών (σε χιλιάδες δολάρια). Για την ερμηνεία της `medv`, διαθέτουμε ένα σύνολο $p = 13$ πιθανών επεξηγηματικών μεταβλητών, οι οποίες περιγράφουν ποικίλα κοινωνικοοικονομικά και περιβαλλοντικά χαρακτηριστικά των εν λόγω προαστίων.

Πρωταρχικός στόχος της άσκησης είναι ο εντοπισμός του συγκεκριμένου υποσυνόλου των δεκατριών επεξηγηματικών μεταβλητών που συνθέτει το "καλύτερο" δυνατό μοντέλο παλινδρόμησης. Η προσέγγισή μας για την επίτευξη αυτού του στόχου θα είναι πολυεπίπεδη. Αρχικά, θα προβούμε σε μια εξαντλητική διερεύνηση του συνόλου όλων των 2^{13} πιθανών υπομοντέλων, με σκοπό την ταυτοποίηση εκείνου που ελαχιστοποιεί το κριτήριο πληροφορίας Akaike (AIC). Αυτή η διαδικασία θα υλοποιηθεί μέσω της ανάπτυξης μιας εξειδικευμένης συνάρτησης στην R.

Στη συνέχεια, θα στραφούμε στην εφαρμογή της μεθόδου LASSO (Least Absolute Shrinkage and Selection Operator). Πρόκειται για μια τεχνική κανονικοποίησης που επιτυγχάνει ταυτόχρονα την προσαρμογή του μοντέλου και την αυτόματη επιλογή μεταβλητών, συρρικνώνοντας τους συντελεστές ορισμένων λιγότερο σημαντικών μεταβλητών ακριβώς στο μηδέν. Για την αντικειμενική επιλογή της βέλτιστης παραμέτρου κανονικοποίησης λ, η οποία ελέγχει την ένταση της συρρίκνωσης, θα εφαρμοστεί η τεχνική της διασταυρούμενης επικύρωσης (cross-validation).

Τέλος, θα επικεντρωθούμε στο μοντέλο M1, το οποίο θα έχει προκύψει από την αρχική πλήρη διερεύνηση ως το βέλτιστο βάσει του κριτηρίου AIC. Θα εφαρμόσουμε τη μέθοδο Residual Bootstrap στο μοντέλο M1 με στόχο την κατασκευή ενός 95% διαστήματος εμπιστοσύνης για τον συντελεστή της επεξηγηματικής μεταβλητής `rm` (μέσος αριθμός δωματίων ανά κατοικία). Αυτό θα μας επιτρέψει να εξάγουμε στατιστικά τεκμηριωμένα συμπεράσματα αναφορικά με την επίδραση του αριθμού των δωματίων στην αξία των κατοικιών.

Αρχικά θα φορτώσουμε τα δεδομένα:

```r
# Load the MASS package which contains the Boston dataset
library(MASS)

# Load the Boston dataset
data(Boston)

# Separate predictors (X) and response (y)
response_variable_name <-"medv"
predictor_names <-setdiff(colnames(Boston), response_variable_name)

n_total_predictors <-length(predictor_names)
n_observations <-nrow(Boston)
```

4.1 Ερώτημα (α): Εξαντλητική Αναζήτηση Βέλτιστου Μοντέλου με AIC

Στο παρόν ερώτημα, ο στόχος είναι η επιλογή του "βέλτιστου" υποσυνόλου επεξηγηματικών μεταβλητών για την πρόβλεψη της μεταβλητής απόκρισης `medv` στο σύνολο δεδομένων "Boston". Δεδομένου ότι διαθέτουμε $p = 13$ πιθανές επεξηγηματικές μεταβλητές, ο συνολικός αριθμός των πιθανών μοντέλων γραμμικής παλινδρόμησης που μπορούν να κατασκευαστούν (συμπεριλαμβανομένου του μοντέλου μόνο με τον σταθερό όρο και του πλήρους μοντέλου) είναι $2^{13} = 8192$.

Αυτός ο αριθμός, αν και μεγάλος, είναι διαχειρίσιμος για μια πλήρη διερεύνηση του χώρου των μοντέλων.

Για κάθε πιθανό μοντέλο m, που χαρακτηρίζεται από ένα υποσύνολο των p επεξηγηματικών μεταβλητών, θα προσαρμόσουμε ένα μοντέλο γραμμικής παλινδρόμησης της μορφής:

$$Y = \beta_0 + \sum_{j \in S_m} \beta_j X_j + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

όπου S_m είναι το σύνολο των δεικτών των μεταβλητών που περιλαμβάνονται στο μοντέλο m.

Για την αξιολόγηση και σύγκριση των διαφορετικών μοντέλων, θα χρησιμοποιήσουμε το **Κριτήριο Πληροφορίας Akaike (Akaike Information Criterion - AIC)** . Το AIC προσφέρει έναν τρόπο εξισορρόπησης μεταξύ της καλής προσαρμογής ενός μοντέλου στα δεδομένα (που τείνει να βελτιώνεται με την προσθήκη περισσότερων μεταβλητών) και της πολυπλοκότητάς του (αριθμός παραμέτρων). Η τιμή του AIC για ένα μοντέλο γραμμικής παλινδρόμησης ορίζεται ως [8]:

$$\text{AIC} = N \log\left(\frac{\text{RSS}_m}{N}\right) + 2(k_m)$$

όπου:

- N είναι το μέγεθος του δείγματος (αριθμός παρατηρήσεων, εδώ $N = 506$).
- RSS_m είναι το Άθροισμα των Τετραγώνων των Υπολοίπων (Residual Sum of Squares) για το μοντέλο m.
- k_m είναι ο συνολικός αριθμός των εκτιμώμενων παραμέτρων στο μοντέλο m. Για ένα μοντέλο γραμμικής παλινδρόμησης με s_m επεξηγηματικές μεταβλητές (πέραν του σταθερού όρου), ο αριθμός των παραμέτρων που αφορούν τους συντελεστές είναι $s_m + 1$ (για τις s_m μεταβλητές συν τον σταθερό όρο). Το AIC, λαμβάνει υπόψη και την εκτίμηση της διασποράς των σφαλμάτων σ^2 ως μια επιπλέον παράμετρο. Επομένως, ο όρος πολυπλοκότητας γίνεται $2(s_m + 1 + 1) = 2(s_m + 2)$.

Εναλλακτικά, και πιο συχνά στην πράξη όταν χρησιμοποιούμε έτοιμες συναρτήσεις όπως η `AIC()` στην `R`, το AIC υπολογίζεται (μέχρι προσθετικής σταθεράς) από τη μεγιστοποιημένη λογαριθμική πιθανοφάνεια $\hat{\ell}_m$ του μοντέλου:

$$\text{AIC} = -2\hat{\ell}_m + 2k_m$$

Το μοντέλο με τη **μικρότερη** τιμή AIC θεωρείται το "καλύτερο".

Θα δημιουργήσουμε μια συνάρτηση στην R που θα κατασκευάζει όλα τα δυνατά υποσύνολα των 13 επεξηγηματικών μεταβλητών, θα προσαρμόζει το αντίστοιχο μοντέλο γραμμικής παλινδρόμησης για καθένα από αυτά, θα υπολογίζει το AIC του, και τέλος θα εντοπίζει το μοντέλο με το ελάχιστο AIC.

Ο παρακάτω κώδικας θα υλοποιήσει τη λογική της πλήρους διερεύνησης.

1. Θα δημιουργήσουμε όλους τους δυνατούς συνδυασμούς των $p = 13$ επεξηγηματικών μεταβλητών. Κάθε συνδυασμός αντιστοιχεί σε ένα μοντέλο.
2. Για κάθε συνδυασμό (μοντέλο):
    - Θα κατασκευάσουμε τη formula του μοντέλου.
    - Θα προσαρμόσουμε το μοντέλο γραμμικής παλινδρόμησης (`lm`).
    - Θα υπολογίσουμε το AIC του μοντέλου (χρησιμοποιώντας την έτοιμη συνάρτηση `AIC()`).

3. Θα αποθηκεύσουμε το AIC και τις μεταβλητές για κάθε μοντέλο.

4. Θα εντοπίσουμε το μοντέλο με το ελάχιστο AIC.

```
# Initialize variables to store results
all_models_aic <-numeric(2^n_total_predictors)
all_models_formulas <-character(2^n_total_predictors)
model_counter <-0

# Loop through all possible combinations of predictors
# i will go from 0 (no predictors, only intercept) to 2^p - 1
# We use binary representation of i to select predictors

# --- For loop for all possible models ---

# Create a list to store model results (formula, AIC, number of predictors)
model_results_list <-list()

for (i in 0:(2^n_total_predictors - 1)) {
 model_counter <-model_counter + 1

 # Convert i to a binary vector of length n_total_predictors
 # This binary vector indicates which predictors are included
 binary_selection <-as.integer(intToBits(i))[1:n_total_predictors]

 selected_predictors <-predictor_names[binary_selection ==1]

 # Construct the formula string for the current model
 if (length(selected_predictors) ==0) {
  # Model with intercept only
  current_formula_str <-paste(response_variable_name, "~ 1")
 } else {
  current_formula_str <-paste(response_variable_name, "~", paste(selected_predictors, collapse =" + "))
 }
 current_formula <-as.formula(current_formula_str)

 # Fit the linear model
 current_lm <-lm(current_formula, data =Boston)

 # Calculate AIC for the current model
 current_aic <-AIC(current_lm)

 # Store results
 all_models_aic[model_counter] <-current_aic
 all_models_formulas[model_counter] <-current_formula_str

 # Store in the list for easier retrieval later
 model_results_list[[model_counter]] <-list(
  formula_str =current_formula_str,
  predictors =selected_predictors,
  num_predictors =length(selected_predictors), # s_m
  k_params =length(coef(current_lm)) + 1, # s_m + 1 (coeffs) + 1 (sigma^2)
  aic =current_aic,
  model_object =current_lm
 )
```

```
}

# Find the model with the minimum AIC
min_aic_index <-which.min(all_models_aic)
best_model_info <-model_results_list[[min_aic_index]]

# Output the best model
cat("\n--- Full Enumeration Results (AIC Criterion) ---\n")
cat("Total models evaluated:", 2^n_total_predictors, "\n")
cat("Best model formula based on AIC:\n", best_model_info$formula_str, "\n")
cat("Minimum AIC value:", best_model_info$aic, "\n")
cat("Number of predictors in the best model (excluding intercept):", best_model_info$num_predictors, "\n
    ")
cat("Predictors in the best model:", paste(best_model_info$predictors, collapse=", "), "\n\n")

# Summary of the best model
print(summary(best_model_info$model_object))
```

Τα αποτελέσματα της ανάλυσης έδειξαν τα εξής:

- **Συνολικός αριθμός αξιολογηθέντων μοντέλων**: 8192
- **Ελάχιστη τιμή AIC που επιτεύχθηκε**: 3023.726
- **Επεξηγηματικές μεταβλητές που περιλαμβάνονται (11)**:
  - **`crim`**: per capita crime rate by town
  - **`zn`**: proportion of residential land zoned for lots over 25,000 sq.ft.
  - **`chas`**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  - **`nox`**: nitric oxides concentration (parts per 10 million)
  - **`rm`**: average number of rooms per dwelling
  - **`dis`**: weighted distances to five Boston employment centres
  - **`rad`**: index of accessibility to radial highways
  - **`tax`**: full-value property-tax rate per $10,000$
  - **`ptratio`**: pupil-teacher ratio by town
  - **`black`**: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - **`lstat`**: % lower status of the population

Από τις 13 αρχικά διαθέσιμες επεξηγηματικές μεταβλητές, το κριτήριο AIC υποδεικνύει ότι ένα μοντέλο με 11 από αυτές επιτυγχάνει την καλύτερη ισορροπία μεταξύ καλής προσαρμογής στα δεδομένα και πολυπλοκότητας. Οι μεταβλητές που **δεν** επιλέχθηκαν από το κριτήριο AIC για το μοντέλο M1 είναι οι `indus` (proportion of non-retail business acres per town) και `age` (proportion of owner-occupied units built prior to 1940).

Η αναλυτική σύνοψη του μοντέλου M1, το οποίο προέκυψε ως βέλτιστο από την πλήρη διερεύνηση βάσει του κριτηρίου AIC, αποκαλύπτει σημαντικά ευρήματα σχετικά με την προσαρμοστική του ικανότητα και τη στατιστική σημασία των παραμέτρων του. Καταρχάς, παρατηρείται ότι όλοι οι συντελεστές που αντιστοιχούν στις 11 επιλεγμένες επεξηγηματικές μεταβλητές, καθώς και ο σταθερός όρος (Intercept), επιδεικνύουν υψηλή στατιστική σημαντικότητα, με τα περισσότερα p-values να είναι μικρότερα του 0.001. Αυτό υποδηλώνει ότι κάθε μία από αυτές τις μεταβλητές συνεισφέρει σημαντικά στην ερμηνεία της μεταβλητής απόκρισης, `medv`.

Επιπλέον, η συνολική στατιστική σημαντικότητα του μοντέλου M1 είναι αδιαμφισβήτητη, όπως καταδεικνύεται από την τιμή του F-statistic (128.2 σε 11 και 494 βαθμούς ελευθερίας) και το αντίστοιχο, πρακτικά μηδενικό, p-value (< 2.2e-16). Όσον αφορά την ερμηνευτική του δύναμη, ο συντελεστής προσδιορισμού (Multiple R-squared) ανέρχεται σε 0.7406, γεγονός που σημαίνει ότι περίπου το 74.06% της συνολικής μεταβλητότητας της μέσης τιμής των κατοικιών (`medv`) μπορεί να αποδοθεί στις 11 επιλεγμένες επεξηγηματικές μεταβλητές. Ενώ ο προσαρμοσμένος συντελεστής προσδιορισμού (Adjusted R-squared), λαμβάνοντας υπόψη τον αριθμό των μεταβλητών, διαμορφώνεται στο 0.7348, επιβεβαιώνοντας την καλή προσαρμογή του μοντέλου.

Συνεπώς, η εξαντλητική διερεύνηση όλων των πιθανών υπομοντέλων, καθοδηγούμενη από το κριτήριο πληροφορίας AIC, μας οδήγησε στην επιλογή του μοντέλου M1 ως την βέλτιστη αναπαράσταση της σχέσης μεταξύ των χαρακτηριστικών των προαστίων και της μέσης τιμής των κατοικιών, εντός του πλαισίου του συνόλου δεδομένων "Boston". Το μοντέλο M1 είναι το εξής:

$$\begin{aligned}\hat{\mathbf{medv}} = 36.341 &- 0.108 \cdot \mathbf{crim} + 0.046 \cdot \mathbf{zn} + 2.719 \cdot \mathbf{chas} - 17.376 \cdot \mathbf{nox} + 3.802 \cdot \mathbf{rm} - 1.493 \cdot \mathbf{dis} + 0.300 \cdot \mathbf{rad} \\ &- 0.012 \cdot \mathbf{tax} - 0.947 \cdot \mathbf{ptratio} + 0.009 \cdot \mathbf{black} - 0.523 \cdot \mathbf{lstat}\end{aligned}$$

4.2 Ερώτημα (β): Επιλογή Μεταβλητών με Παλινδρόμηση Lasso

Στο προηγούμενο ερώτημα, χρησιμοποιήσαμε την πλήρη διερεύνηση του χώρου των μοντέλων για την επιλογή του βέλτιστου υποσυνόλου επεξηγηματικών μεταβλητών βάσει του κριτηρίου AIC. Ωστόσο, για προβλήματα με μεγαλύτερο αριθμό επεξηγηματικών μεταβλητών p, η πλήρης διερεύνηση (2^p μοντέλα) είναι υπολογιστικά απαγορευτική.

Η μέθοδος **LASSO (Least Absolute Shrinkage and Selection Operator)** προσφέρει μια εναλλακτική προσέγγιση που συνδυάζει την προσαρμογή του μοντέλου γραμμικής παλινδρόμησης με την αυτόματη επιλογή μεταβλητών [7]. Αυτό επιτυγχάνεται μέσω της ελαχιστοποίησης του αθροίσματος των τετραγώνων των υπολοίπων (RSS) υπό τον περιορισμό ότι το άθροισμα των απόλυτων τιμών των συντελεστών είναι μικρότερο από μια σταθερά t:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right\} \quad \text{υπό τον περιορισμό} \quad \sum_{j=1}^{p} |\beta_j| \leq t$$

Ισοδύναμα, το LASSO ελαχιστοποιεί μια "ποινικοποιημένη" έκδοση του RSS:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Η παράμετρος $\lambda \geq 0$ είναι μια παράμετρος που ελέγχει την ένταση της ποινής L_1 (το άθροισμα των απόλυτων τιμών των συντελεστών). Καθώς το λ αυξάνεται (ή ισοδύναμα, το t μειώνεται), οι συντελεστές β_j συρρικνώνονται προς το μηδέν. Μια κρίσιμη ιδιότητα του LASSO είναι ότι, λόγω της φύσης της ποινής L_1, για επαρκώς μεγάλες τιμές του λ, ορισμένοι συντελεστές μηδενίζονται ακριβώς, επιτυγχάνοντας έτσι την επιλογή μεταβλητών.

Στην πράξη, η βέλτιστη τιμή της παραμέτρου λ επιλέγεται συνήθως μέσω cross-validation. Στο παρόν ερώτημα, θα χρησιμοποιήσουμε τη συνάρτηση `cv.glmnet` από το πακέτο `glmnet` της R, η οποία υλοποιεί το LASSO και επιλέγει το λ μέσω k-fold cross-validation, ελαχιστοποιώντας το μέσο τετραγωνικό σφάλμα πρόβλεψης (CV-MSE).

Συγκεκριμένα, θα εντοπίσουμε δύο τιμές του λ:

- `lambda.min`: Η τιμή του λ που δίνει το ελάχιστο CV-MSE.
- `lambda.1se` Η μεγαλύτερη τιμή του λ για την οποία το CV-MSE βρίσκεται εντός ενός τυπικού σφάλματος από το ελάχιστο CV-MSE. Αυτή η επιλογή οδηγεί συχνά σε πιο λιτά (parsimonious) μοντέλα με παρόμοια προβλεπτική ικανότητα.

Θα εφαρμόσουμε τη μέθοδο LASSO στο σύνολο δεδομένων "Boston" και θα αναλύσουμε τα αποτελέσματα, εστιάζοντας στο μοντέλο που αντιστοιχεί στο `lambda.1se`.

```
library(glmnet)

# We need to remove the intercept added by model.matrix for glmnet's X.
x_matrix_full <-model.matrix(medv ~ ., data =Boston)[, -1]
y_vector <-Boston$medv
```

Ο παρακάτω κώδικας χρησιμοποιεί τη συνάρτηση `cv.glmnet` για να εκτελέσει k-fold cross-validation (εδώ 10-fold) και να βρει τις βέλτιστες τιμές `lambda.min` και `lambda.1se`.

```
# Perform cross-validation to find the optimal lambda
set.seed(123) # For reproducibility of cross-validation folds
cv_lasso_fit <-cv.glmnet(x =x_matrix_full, y =y_vector, alpha =1, nfolds =10)

# Plot the cross-validation results (MSE vs. log(lambda))
plot(cv_lasso_fit)
title("LASSO Cross-Validation: MSE vs. log(Lambda)", line =2.5)

# Optimal lambda values
lambda_min <-cv_lasso_fit$lambda.min
lambda_1se <-cv_lasso_fit$lambda.1se

cat("\n--- LASSO Cross-Validation Results ---\n")
cat("Lambda min (value of lambda that gives minimum CV-MSE):", lambda_min, "\n")
cat("Lambda 1se (largest lambda within 1 SE of minimum CV-MSE):", lambda_1se, "\n")
```

Το Σχήμα 8 απεικονίζει την καμπύλη του CV-MSE ως συνάρτηση του $\log(\lambda)$. Οι κάθετες διακεκομμένες γραμμές υποδεικνύουν τις θέσεις των `lambda.min` και `lambda.1se`, ενώ οι αριθμοί στην κορυφή του διαγράμματος αντιστοιχούν στον αριθμό των μη μηδενικών συντελεστών (επιλεγμένων μεταβλητών) για κάθε τιμή του λ.

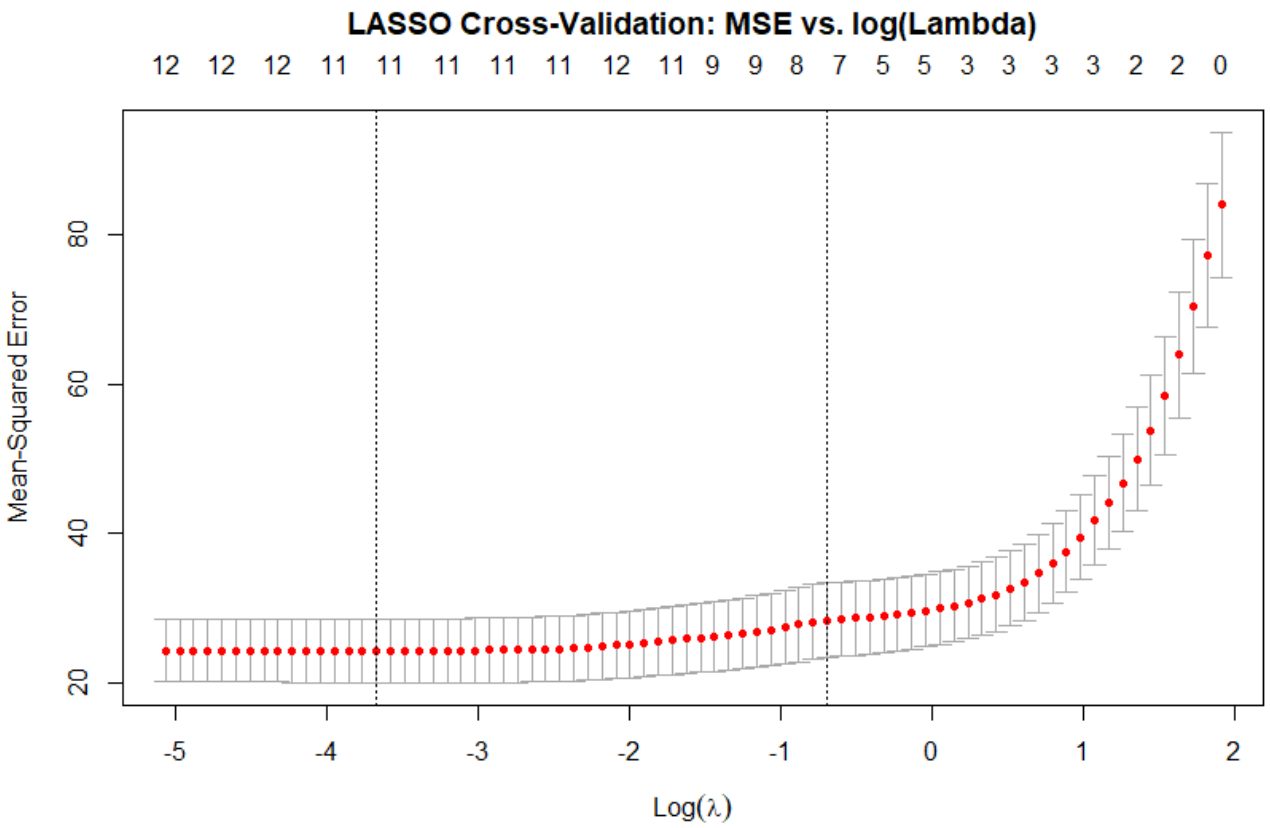
Από την εκτέλεση της `cv.glmnet`, προέκυψαν οι ακόλουθες τιμές για τις παραμέτρους λ:

- **`lambda.min`** (η τιμή του λ που ελαχιστοποιεί το CV-MSE): 0.02551743
- **`lambda.1se`** (η μεγαλύτερη τιμή του λ για την οποία το CV-MSE βρίσκεται εντός ενός τυπικού σφάλματος από το ελάχιστο CV-MSE): 0.5009175

Παρατηρούμε ότι το `lambda.1se` είναι σημαντικά μεγαλύτερο από το `lambda.min`. Αυτό είναι αναμενόμενο, καθώς το κριτήριο "1se" επιλέγει ένα πιο λιτό μοντέλο (μεγαλύτερη τιμή λ οδηγεί σε μεγαλύτερη συρρίκνωση και περισσότερους μηδενικούς συντελεστές) το οποίο όμως έχει προβλεπτική απόδοση στατιστικά ισοδύναμη (εντός ενός τυπικού σφάλματος) με το μοντέλο που αντιστοιχεί στο `lambda.min`. Όπως φαίνεται και από τους αριθμούς στην κορυφή του διαγράμματος

8, το μοντέλο που αντιστοιχεί στο $\log(\lambda_{1se})$ (περίπου $\log(0.5009) \approx -0.69$) έχει 7 μη μηδενικούς συντελεστές, ενώ το μοντέλο που αντιστοιχεί στο $\log(\lambda_{min})$ (περίπου $\log(0.0255) \approx -3.67$) έχει 11 μη μηδενικούς συντελεστές.

Σχήμα 8: LASSO Cross-Validation: MSE vs. log(λ)



Θα προχωρήσουμε στην ανάλυση του μοντέλου που αντιστοιχεί στο `lambda.1se`, καθώς αυτό μας δίνει ένα πιο "φειδωλό" (parsimony) μοντέλο χωρίς σημαντική απώλεια στην προβλεπτική ακρίβεια. Θα ανακτήσουμε και θα εξετάσουμε τους συντελεστές αυτού του μοντέλου.

```
# Get coefficients for lambda.1se
coefficients_lambda_1se <-coef(cv_lasso_fit, s ="lambda.1se")

cat("\nCoefficients for the LASSO model with lambda.1se:\n")
print(coefficients_lambda_1se)

# Identify selected predictors (those with non-zero coefficients)
selected_predictors_lasso <-rownames(coefficients_lambda_1se)[which(coefficients_lambda_1se !=0)]
# Remove "(Intercept)", as we are interested in explanatory variables
selected_predictors_lasso <-setdiff(selected_predictors_lasso, "(Intercept)")

cat("\nPredictors selected by LASSO (lambda.1se):\n")
cat(paste(selected_predictors_lasso, collapse =", "), "\n")
```

```
cat("Number of predictors selected (excluding intercept):", length(selected_predictors_lasso), "\n")
```

Οι συντελεστές του μοντέλου LASSO που αντιστοιχούν σε `lambda.1se` ≈ 0.5009175, παρουσιάζονται στον πίνακα 1:

Πίνακας 1: Συντελεστές Μοντέλου LASSO με $\lambda = \lambda_{1se} \approx 0.5009175$

Μεταβλητή	Συντελεστής
(Intercept)	14.161113866
crim	-0.013309174
zn	0.000000000
indus	0.000000000
chas	1.562819145
nox	0.000000000
rm	4.237264636
age	0.000000000
dis	-0.080074543
rad	0.000000000
tax	0.000000000
ptratio	-0.738845222
black	0.005949482
lstat	-0.513735694

Παρατηρούμε ότι η μέθοδος LASSO, με την επιλογή `lambda.1se`, έχει μηδενίσει τους συντελεστές για τις μεταβλητές: `zn`, `indus`, `nox`, `age`, `rad`, και `tax`. Αυτό σημαίνει ότι αυτές οι μεταβλητές δεν κρίνονται ως σημαντικές για την πρόβλεψη της `medv` από το συγκεκριμένο μοντέλο LASSO.

Οι επεξηγηματικές μεταβλητές που επιλέχθηκαν (δηλαδή, αυτές με μη μηδενικούς συντελεστές) είναι: `crim`, `chas`, `rm`, `dis`, `ptratio`, `black`, και `lstat`. Συνολικά, επιλέχθηκαν 7 από τις 13 αρχικά διαθέσιμες επεξηγηματικές μεταβλητές.

Το μοντέλο γραμμικής παλινδρόμησης που προκύπτει από τη μέθοδο LASSO με `lambda.1se`, στρογγυλοποιώντας τους συντελεστές σε 3 δεκαδικά ψηφία, είναι:

$$\widehat{\textbf{medv}} = 14.161 - 0.013 \cdot \textbf{crim} + 1.563 \cdot \textbf{chas} + 4.237 \cdot \textbf{rm} - 0.080 \cdot \textbf{dis} - 0.739 \cdot \textbf{ptratio} + 0.006 \cdot \textbf{black} - 0.514 \cdot \textbf{lstat}$$

Συγκριτικά με το μοντέλο M1 που προέκυψε από την πλήρη διερεύνηση με κριτήριο AIC (το οποίο περιελάμβανε 11 μεταβλητές), το μοντέλο LASSO με `lambda.1se` είναι πιο λιτό, διατηρώντας μόνο 7 επεξηγηματικές μεταβλητές. Οι μεταβλητές `zn`, `nox`, `rad`, και `tax`, οι οποίες περιλαμβάνονταν στο μοντέλο M1, έχουν αποκλειστεί από το μοντέλο LASSO. Αυτό είναι αναμενόμενο, καθώς το `lambda.1se` τείνει να επιλέγει πιο απλά μοντέλα, θυσιάζοντας ενδεχομένως μια πολύ μικρή ποσότητα προβλεπτικής ακρίβειας (όπως φαίνεται από την καμπύλη CV-MSE) προς όφελος της ερμηνευσιμότητας και της γενίκευσης.

4.3 Ερώτημα (γ): Διάστημα Εμπιστοσύνης Συντελεστή με Residual Bootstrap

Στο παρόν υποερώτημα, θα εφαρμόσουμε τη μέθοδο **Residual Bootstrap** στο μοντέλο M1 (το οποίο προσδιορίστηκε στο υποερώτημα α ως το βέλτιστο μοντέλο βάσει του κριτηρίου AIC) για να κατασκευάσουμε ένα 95% διάστημα εμπιστοσύνης για τον συντελεστή της επεξηγηματικής μεταβλητής `rm` (average number of rooms per dwelling).

Το μοντέλο M1 είναι:

$$\widehat{\textbf{medv}} = 36.341 - 0.108 \cdot \textbf{crim} + 0.046 \cdot \textbf{zn} + 2.719 \cdot \textbf{chas} - 17.376 \cdot \textbf{nox} + 3.802 \cdot \textbf{rm} - 1.493 \cdot \textbf{dis} + 0.300 \cdot \textbf{rad} - 0.012 \cdot \textbf{tax} - 0.947 \cdot \textbf{ptratio} + 0.009 \cdot \textbf{black} - 0.523 \cdot \textbf{lstat}$$

Η μέθοδος Residual Bootstrap [4] είναι κατάλληλη όταν υποθέτουμε ότι οι επεξηγηματικές μεταβλητές X είναι σταθερές (fixed design) και ότι τα σφάλματα ϵ_i είναι ανεξάρτητα και ισόνομα κατανεμημένα (i.i.d.) με μέσο μηδέν, αλλά χωρίς να κάνουμε την παραδοχή της κανονικότητας των σφαλμάτων. Η διαδικασία έχει ως εξής:

1. **Προσαρμογή Αρχικού Μοντέλου**: Προσαρμόζουμε το μοντέλο M1 στα αρχικά δεδομένα και λαμβάνουμε τις εκτιμήσεις των συντελεστών $\hat{\boldsymbol{\beta}}_{M1}$ (συμπεριλαμβανομένου του $\hat{\beta}_{rm}$) και τα υπόλοιπα $\hat{e}_i = y_i - \hat{y}_i$.

2. **Κεντροποίηση Υπολοίπων**: Τα υπόλοιπα συνήθως [9] $\hat{e}_i$ κεντροποιούνται αφαιρώντας τον μέσο όρο τους, $\tilde{e}_i = \hat{e}_i - \bar{\hat{e}}$, ώστε το νέο σύνολο υπολοίπων να έχει ακριβώς μέσο μηδέν. Αυτό είναι σημαντικό για να αποφευχθεί η εισαγωγή μεροληψίας στις Bootstrap προσομοιώσεις.

3. **Δημιουργία Bootstrap Δειγμάτων Απόκρισης**: Για $b = 1, \ldots, B$ επαναλήψεις (εδώ $B = 2000$):

   - Δημιουργούμε ένα Bootstrap δείγμα υπολοίπων $e_1^{*b}, \ldots, e_N^{*b}$ με δειγματοληψία με επανατοποθέτηση από τα (κεντροποιημένα) υπόλοιπα $\tilde{e}_i$.
   - Δημιουργούμε ένα νέο (Bootstrap) δείγμα της μεταβλητής απόκρισης $y_i^{*b} = \hat{y}_i + e_i^{*b}$, όπου $\hat{y}_i$ είναι οι προβλεπόμενες τιμές από το αρχικό μοντέλο M1.

4. **Επαναπροσαρμογή Μοντέλου**: Για κάθε σύνολο δεδομένων (X, y^{*b}), προσαρμόζουμε ξανά το μοντέλο M1 (δηλαδή, παλινδρομούμε το y^{*b} στις ίδιες επεξηγηματικές μεταβλητές που περιέχει το μοντέλο M1) και λαμβάνουμε την Bootstrap εκτίμηση του συντελεστή για τη μεταβλητή `rm`, έστω $\hat{\beta}_{rm}^{*b}$.

5. **Κατασκευή Διαστήματος Εμπιστοσύνης**: Χρησιμοποιώντας τις B εκτιμήσεις $\hat{\beta}_{rm}^{*1}, \ldots, \hat{\beta}_{rm}^{*B}$, κατασκευάζουμε ένα 95% διάστημα εμπιστοσύνης χρησιμοποιώντας τη μέθοδο των ποσοστημορίων (percentile method). Αυτό περιλαμβάνει την ταξινόμηση των $\hat{\beta}_{rm}^{*b}$ και την επιλογή του 2.5% και του 97.5% ποσοστημορίου ως τα όρια του διαστήματος.

```
# Formula for Model M1
formula_M1_str <-"medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat"
formula_M1 <-as.formula(formula_M1_str)

# Fit the original Model M1
model_M1_fit <-lm(formula_M1, data =Boston)

# Extract the original estimate for the coefficient of 'rm'
original_beta_rm_hat <-coef(model_M1_fit)["rm"]

# Get residuals from the original model M1
residuals_M1 <-residuals(model_M1_fit)
```

```r
# Center the residuals
centered_residuals_M1 <-residuals_M1 - mean(residuals_M1)
n_obs_boston <-length(centered_residuals_M1)

# Get fitted values from the original model M1
fitted_values_M1 <-fitted(model_M1_fit)

# Parameters for Bootstrap
B_bootstrap <-2000
bootstrap_beta_rm_estimates <-numeric(B_bootstrap) # To store bootstrap estimates of beta_rm
```

Ο παρακάτω κώδικας R υλοποιεί τον βρόχο Residual Bootstrap. Σε κάθε επανάληψη, δημιουργείται ένα νέο σύνολο τιμών για τη μεταβλητή απόκρισης, το μοντέλο M1 επαναπροσαρμόζεται, και ο συντελεστής για τη μεταβλητή rm αποθηκεύεται.

```r
set.seed(4567) # For reproducibility

for (b in 1:B_bootstrap) {
 # Generate a Bootstrap sample of residuals
 bootstrap_residuals_sample <-sample(centered_residuals_M1, size =n_obs_boston, replace =TRUE)

 # Create a new bootstrap response variable y_star
 y_star_bootstrap <-fitted_values_M1 + bootstrap_residuals_sample

 # Create a temporary bootstrap data frame
 # We need to ensure the original predictors are used with the new y_star
 bootstrap_data <-Boston # Start with original data
 bootstrap_data$medv <-y_star_bootstrap # Replace response with y_star

 # Refit Model M1 using the bootstrap data
 # The formula_M1 already defines which X's are used.
 refitted_model_M1_bootstrap <-lm(formula_M1, data =bootstrap_data)

 # Store the coefficient for 'rm' from this bootstrap iteration
 bootstrap_beta_rm_estimates[b] <-coef(refitted_model_M1_bootstrap)["rm"]
}
```

Αφού συλλέξουμε τις $B = 2000$ Bootstrap εκτιμήσεις $\hat{\beta}_{rm}^{*b}$, θα υπολογίσουμε το 95% διάστημα εμπιστοσύνης για τον β_{rm} χρησιμοποιώντας τη μέθοδο των ποσοστημορίων. Αυτό περιλαμβάνει την εύρεση του 2.5% και του 97.5% ποσοστημορίου της εμπειρικής κατανομής των $\hat{\beta}_{rm}^{*b}$.

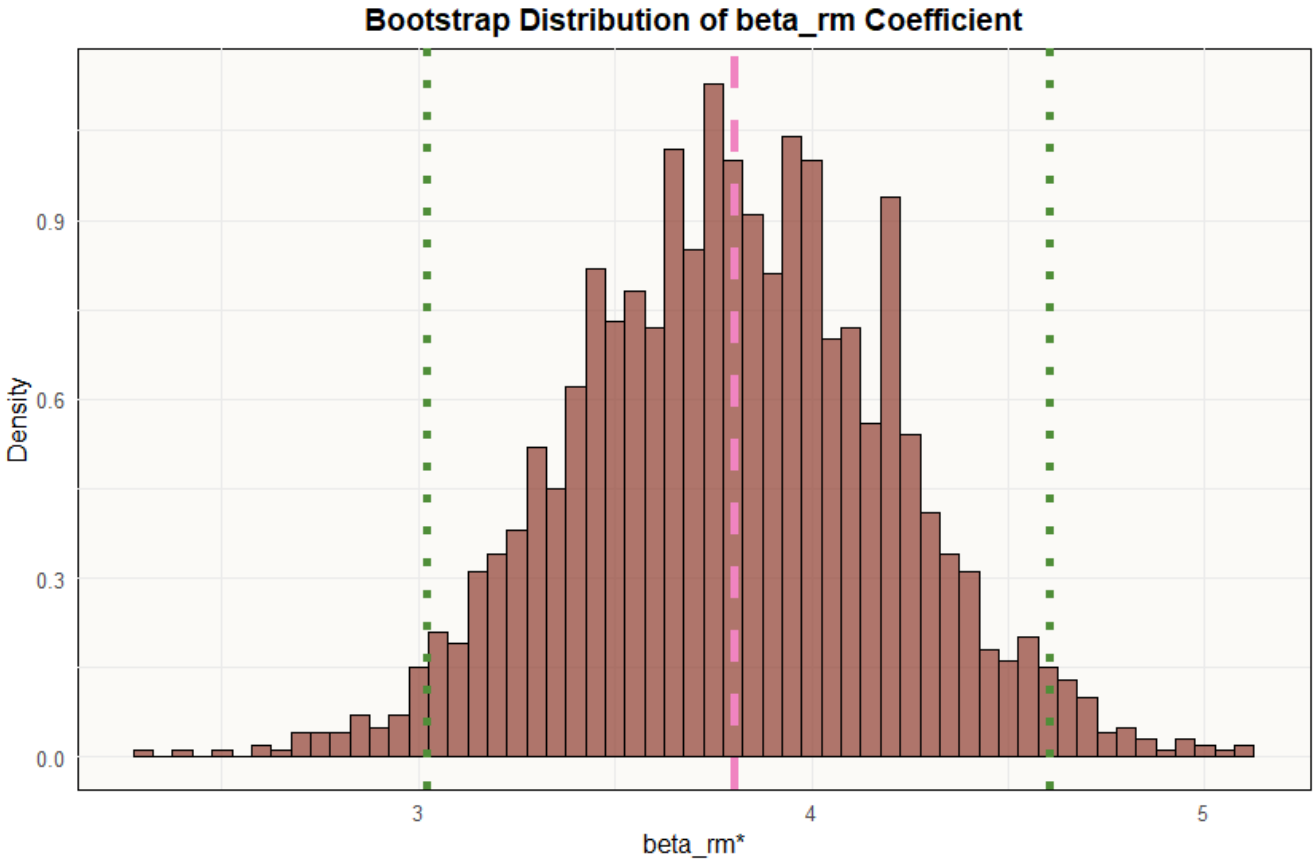
```r
# Construct the 95% percentile bootstrap confidence interval for beta_rm
alpha_CI <-0.05
percentile_CI_beta_rm <-quantile(bootstrap_beta_rm_estimates,
                         probs =c(alpha_CI / 2, 1 - (alpha_CI / 2)),
                         na.rm =TRUE)

cat("\n--- Residual Bootstrap Results for beta_rm (Coefficient of 'rm') ---\n")
cat("Original OLS estimate for beta_rm (from Model M1):", original_beta_rm_hat, "\n")
cat("Number of Bootstrap Samples (B):", B_bootstrap, "\n")
cat("95% Percentile Bootstrap Confidence Interval for beta_rm: [",
```

```
    round(percentile_CI_beta_rm[1], 4), ", ",
    round(percentile_CI_beta_rm[2], 4), "]\n")

#Histogram of bootstrap_beta_rm_estimates
library(ggplot2)
hist_df_beta_rm <-data.frame(beta_rm_star =bootstrap_beta_rm_estimates)
plot_hist_beta_rm <-ggplot(hist_df_beta_rm, aes(x =beta_rm_star)) +
  geom_histogram(aes(y=..density..), binwidth =0.05, fill="#8E3C2E", color="black", alpha=0.7) +
  geom_vline(xintercept =original_beta_rm_hat, color="#F084C1", linetype="dashed", linewidth=2) +
  geom_vline(xintercept =percentile_CI_beta_rm[1], color="#4F8E38", linetype="dotted", linewidth=2) +
  geom_vline(xintercept =percentile_CI_beta_rm[2], color="#4F8E38", linetype="dotted", linewidth=2) +
  labs(title="Bootstrap Distribution of beta_rm Coefficient", x="beta_rm*", y="Density") +
  theme_minimal(base_size =12) +
  theme(plot.title =element_text(hjust =0.5, face ="bold"),
    panel.background =element_rect(fill ="#FBFAF7"))
print(plot_hist_beta_rm)
```

Η εφαρμογή της μεθόδου Residual Bootstrap, με $B = 2000$ επαναλήψεις, στο μοντέλο M1 για την κατασκευή διαστήματος εμπιστοσύνης για τον συντελεστή β_{rm} της μεταβλητής $\texttt{rm}$ οδήγησε στα ακόλουθα αποτελέσματα:

- **Αρχική Εκτίμηση OLS για τον β_{rm} (από το μοντέλο M1):** $\hat{\beta}_{rm} \approx 3.8016$
- 95% **Διάστημα Εμπιστοσύνης Percentile Bootstrap για τον β_{rm}:** $[3.0225, 4.6036]$

Σχήμα 9: Κατανομή των 2000 Bootstrap εκτιμήσεων $\hat{\beta}_{rm}^{*b}$ του συντελεστή β_{rm} της μεταβλητής $\texttt{rm}$



Το Σχήμα 9 απεικονίζει το ιστόγραμμα των 2000 Bootstrap εκτιμήσεων $\hat{\beta}_{rm}^{*b}$. Η κατανομή των εκτιμήσεων φαίνεται προσεγγιστικά συμμετρική και κεντραρισμένη κοντά στην αρχική εκτίμηση OLS (που υποδεικνύεται από την κάθετη ροζ διακεκομμένη γραμμή). Οι δύο πράσινες διακεκομμένες γραμμές αντιστοιχούν στα όρια του 95% διαστήματος εμπιστοσύνης που υπολογίστηκε με τη μέθοδο των ποσοστημορίων.

Συμπεράσματα σχετικά με την επίδραση της μεταβλητής `rm`:

Το 95% διάστημα εμπιστοσύνης για τον συντελεστή β_{rm} είναι $[3.0225, 4.6036]$. Δεδομένου ότι ολόκληρο το διάστημα βρίσκεται **αυστηρά πάνω από το μηδέν** (δηλαδή, δεν περιλαμβάνει την τιμή 0), μπορούμε να συμπεράνουμε με 95% βεβαιότητα ότι η μεταβλητή `rm` έχει στατιστικά σημαντική θετική επίδραση στη μεταβλητή `medv` (μέση τιμή κατοικιών), διατηρώντας σταθερές τις υπόλοιπες μεταβλητές του μοντέλου M1.

Αυτό σημαίνει ότι, κατά μέσο όρο, μια αύξηση στον αριθμό των δωματίων ανά κατοικία (`rm`) σχετίζεται με μια αύξηση στη μέση τιμή της κατοικίας (`medv`). Η εκτίμηση σημείου για αυτή την αύξηση είναι περίπου 3.8016 χιλιάδες δολάρια για κάθε επιπλέον δωμάτιο, και το διάστημα εμπιστοσύνης μας δίνει ένα εύρος πιθανών τιμών για αυτή την επίδραση από περίπου 3.02 έως 4.60 χιλιάδες δολάρια. Η μέθοδος Residual Bootstrap μας επέτρεψε να κατασκευάσουμε αυτό το διάστημα χωρίς να βασιστούμε στην παραδοχή της κανονικότητας των σφαλμάτων του μοντέλου παλινδρόμησης, προσφέροντας έτσι μια πιο στιβαρή εκτίμηση της αβεβαιότητας γύρω από τον συντελεστή β_{rm}.

5 Συμπεράσματα

Η παρούσα εργασία αποτέλεσε μια πρακτική και σε βάθος εξερεύνηση θεμελιωδών μεθόδων της Υπολογιστικής Στατιστικής και της Στοχαστικής Βελτιστοποίησης. Μέσα από τέσσερις κύριες ασκήσεις, εφαρμόστηκε, αναλύθηκε και αξιολογήθηκε κριτικά ένα ευρύ φάσμα τεχνικών, από τη μη παραμετρική παλινδρόμηση και τις μεθόδους επαναδειγματοληψίας, έως τις τεχνικές προσομοίωσης Monte Carlo και την αλγοριθμική εκτίμηση παραμέτρων.

Ένα κεντρικό σημείο που χαρακτηρίζει το σύνολο της εργασίας ήταν η **συνεχής σύνδεση μεταξύ της στατιστικής θεωρίας και της πρακτικής υλοποίησης στην R**. Σε ασκήσεις όπως η εφαρμογή του εκτιμητή Nadaraya-Watson (1α) και η ανάπτυξη του αλγορίθμου EM (3), η αυστηρή μαθηματική θεμελίωση μεταφράστηκε απευθείας σε λειτουργικό και αποδοτικό κώδικα, επιβεβαιώνοντας τη στενή σχέση μεταξύ της θεωρίας και της εφαρμογής.

Εξίσου σημαντική ήταν η ανάδειξη της **κρισιμότητας των υποκείμενων υποθέσεων** κάθε μεθόδου. Η αντίθεση μεταξύ της αποτυχίας του μη παραμετρικού Bootstrap για την εκτίμηση του ελαχίστου (1β.i) και της επιτυχίας της παραμετρικής του εκδοχής (1β.ii) κατέδειξε με τον πιο σαφή τρόπο το κόστος και το όφελος της υιοθέτησης μιας παραμετρικής παραδοχής. Αντίστοιχα, η απόδοση των μεθόδων απόρριψης (2α) και δειγματοληψίας σπουδαιότητας (2β) αποδείχθηκε άρρηκτα συνδεδεμένη με την κατάλληλη επιλογή της κατανομής εισήγησης, η οποία πρέπει να "μιμείται" αποτελεσματικά την κατανομή-στόχο.

Τέλος, η εργασία ανέδειξε την **σημασία της υπολογιστικής αποδοτικότητας, της στατιστικής ακρίβειας και της ερμηνευσιμότητας του μοντέλου**. Η αποδοτική υλοποίηση του LOOCV (1α.ii), η χρήση της τεχνικής "squeezing" για τη μείωση των υπολογισμών (2α.ii), η δραματική μείωση της διασποράς μέσω της δειγματοληψίας σπουδαιότητας (2β.ii) και η επιλογή ενός πιο λιτού μοντέλου μέσω του Lasso (4β) αποτελούν χαρακτηριστικά παραδείγματα. Σε κάθε περίπτωση, η επιλογή της "βέλτιστης" μεθόδου εξαρτάται από τις συγκεκριμένες απαιτήσεις του προβλήματος, τους διαθέσιμους υπολογιστικούς πόρους και τον τελικό στόχο της ανάλυσης.

Συνοψίζοντας, η παρούσα εργασία λειτούργησε ως ένα δομημένο πλαίσιο για τη σύνδεση της στατιστικής θεωρίας με την πρακτική εφαρμογή. Απαίτησε τη σύνθεση ποικίλων δεξιοτήτων, από την θεωρητική ανάπτυξη και αλγοριθμική διατύπωση των μεθόδων, τη συγγραφή αποδοτικού κώδικα, έως την παραγωγή προσομοιώσεων και την κριτική ερμηνεία των παραγόμενων αποτελεσμάτων, διαδικασία που αποτελεί τον ακρογωνιαίο λίθο της σύγχρονης εφαρμοσμένης στατιστικής.

Αναφορές

[1] Dimitris Fouskakis. *Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση - Παρουσιάσεις Μαθήματος*. Ιστοσελίδα μαθήματος. Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/cs_slides.html.

[2] Dimitris Fouskakis. *Density estimation*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/1.density_estimation.pdf.

[3] Dimitris Fouskakis. *Cross-Validation*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/4.cross-validation.pdf.

[4] Dimitris Fouskakis. *Resampling Methods: Jackknife-Bootstrap*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/3.Jackknife-Bootstrap.pdf.

[5] Dimitris Fouskakis. *Stochastic Simulation*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/2.%20stochastic_simulation.pdf.

[6] Dimitris Fouskakis. *Expectation-Maximization Algorithm*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/5.EM.pdf.

[7] Dimitris Fouskakis. *Variable Selection*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/7.variable_selection.pdf.

[8] Dimitris Fouskakis. *Stochastic Optimisation Methods*. Διαφάνειες διάλεξης για το μάθημα "Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση". Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Εθνικό Μετσόβιο Πολυτεχνείο, 2022. URL: http://www.math.ntua.gr/~fouskakis/Computational_Stats/Slides/6.stochastic-optimisation.pdf.

[9] Anthony C. Davison και David V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, 1997. ISBN: 9780521574716.