ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF APPLIED MATHEMATICAL AND PHYSICAL SCIENCES

Εισαγωγή στην R και Περιγραφική Στατιστική

Ονοματεπώνυμο φοιτητή: Γεώργιος Λεβής
Αριθμός μητρώου: ge19120
Έτος: 4ο
Εξάμηνο: 8ο
Email: ge19120@mail.ntua.gr / levgiorg@gmail.com

Ακαδημαϊκό Έτος
2022 – 2023

Άσκηση 1

Εισάγουμε τα δοθέντα στοιχεία στην R με την παρακάτω εντολή :

```
gifts <-
read.table("http://www.math.ntua.gr/~fouskakis/Data_Analysis/Exercises/gifts.tx
t", header = TRUE,na.strings="*")
```

Έτσι δηλώνουμε το path του αρχείου .txt, με το argument (header = TRUE) ενημερώνουμε ότι στην πρώτη γραμμή δηλώνονται ονόματα και με το argument (na.strings="*") δηλώνουμε ότι ο χαρακτήρας "*" είναι οι αγνοούμενες τιμές. Η R μας δίνει το πλαίσιο δεδομένων gifts το οποίο είναι ένα data frame. Συνεχίζουμε με τον υπόλοιπο κώδικα.

```
str(gifts)
gifts <- na.omit(gifts)
gifts <- subset(gifts, age >= 18)
```

Χρησιμοποιώ την εντολή (str(gift)) για να δω την δομή του 'gifts'. Έπειτα χρησιμοποιώ την εντολή (gifts <- na.omit(gifts)) για να αφαιρέσω οποιαδήποτε γραμμή περιέχει αγνοουμένη τιμή NA. Η εντολή (gifts <- subset(gifts, age >= 18)) με βοηθάει να αφαιρέσω οποιαδήποτε γραμμή που περιέχει τιμές από μη ενήλικα άτομα. Χρησιμοποιούμε την εντολή subset() γιατί μας επιτρέπει να πάρουμε ένα συγκεκριμένο subset από ένα data frame. Έτσι, στο πρώτο argument επιλέγουμε το data frame που θέλουμε να πάρουμε ('gifts') και στο δεύτερο argument γράφουμε τα conditions, στην περίπτωση μας είναι age >= 18 με το οποίο κρατάμε όλες τις σειρές που έχουν τιμή ίση ή πάνω από 18.

i)

Θα δοθεί μια περιγραφική και γραφική ανάλυση κάθε ξεχωριστής μεταβλητής των δεδομένων. Αρχικά θα ορίσουμε τις 6 μεταβλητές από τις στήλες εξαρχής για δική μας ευκολία.

```
spend <- gifts$spend
age <- gifts$age
holiday <-gifts$holiday
sex <- gifts$sex
time <- gifts$time
salary <- gifts$salary
```

Spend :

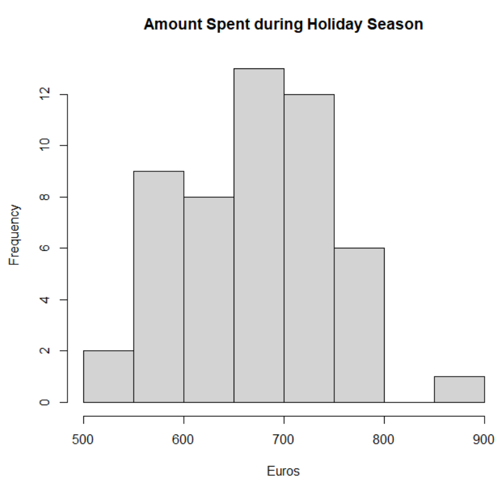
```
class(spend)
summary(spend)
hist(spend, main="Amount Spent during Holiday Season", xlab="Euros")
boxplot(spend, main="Amount Spent during Holiday Season", xlab="Euros")
fivenum(spend)
```

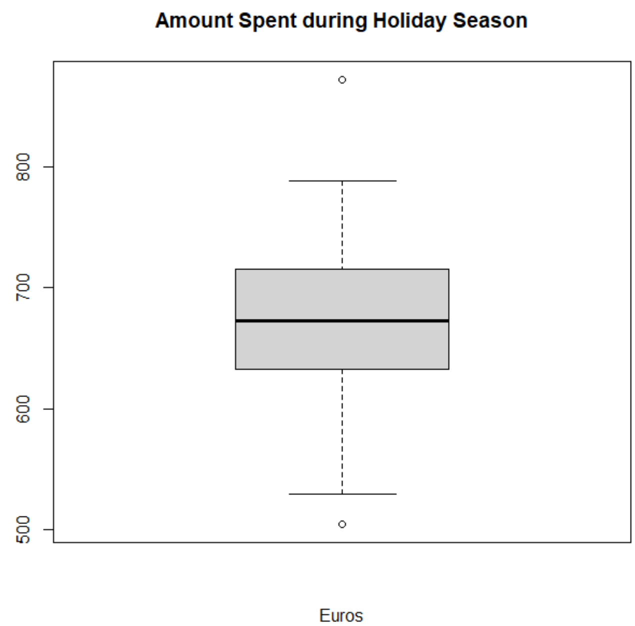
Με την εντολή class `class(spend)` μας δίνεται ο τύπος δεδομένων της μεταβλητής spend ο οποίος είναι 'numeric' δηλαδή αριθμητική κλάση.

Το summary μας δίνει μερικές σημαντικές τιμές για την μεταβλητή spend, δηλαδή το ελάχιστο(min), το πρώτο τεταρτημόριο(1st Qu.), τη δειγματική διάμεσο(Median), το δειγματικό μέσο(Mean), το τρίτο τεταρτημόριο(3rd Qu.) και το μέγιστο(Max).

```
> summary(spend)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  504.2   632.3   672.4   671.1   715.5   872.4 
```

Το πρώτο argument του hist() και box() λέει στην R να σχεδιάσει την στήλη spend που ορίσαμε πριν, το xlab βάζει έναν τίτλο στον άξονα x και το main δίνει έναν τίτλο στο διάγραμμα μας. Παίρνουμε το εξής γράφημα.





Παρατηρούμε ότι οι περισσότεροι σπαταλούν από 550 έως 800 ευρώ στην περίοδο των γιορτών. Από το θηκόγραμμα παρατηρούμε ότι το ελάχιστο ποσό που σπαταλούν κατά την εορταστική περίοδο είναι λίγο περισσότερο από 500 ευρώ. Η μεγαλύτερη τιμή του δείγματος είναι περίπου στα 750 ευρώ . Παρατηρούμε πως δεν υπάρχουν έκτροπες τιμές. Με την εντολή `fivenum(spend)` παίρνουμε τα παρακάτω και μπορούμε να επαληθεύσουμε τα λεγόμενα μας.

```
[1] 504.250 632.305 672.410 715.480 872.420
```

<u>Age :</u>

```
class(age)
summary(age)
hist(age, main="Age Distribution of Survey Participants", xlab="Age")
boxplot(age, main="Age Distribution of Survey Participants", xlab="Age")
```

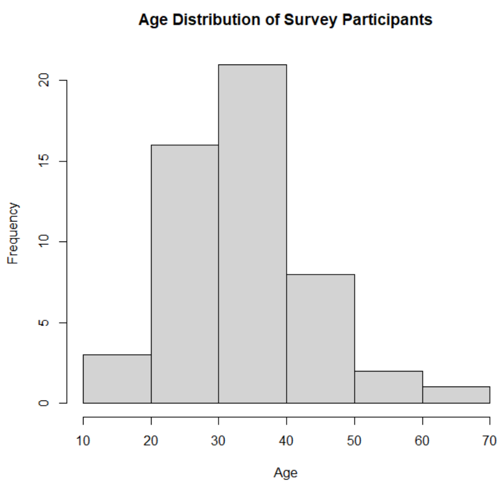
```
fivenum(age)
```

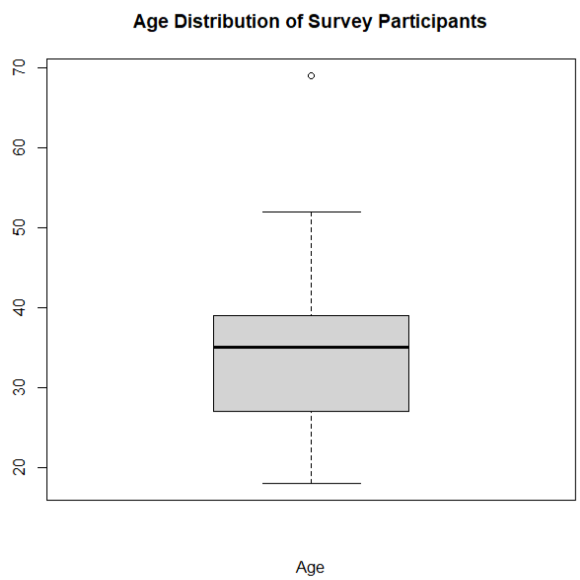
Με την εντολή class `class(age)` μας δίνεται ο τύπος δεδομένων της μεταβλητής spend ο οποίος είναι 'integer' δηλαδή αριθμητική κλάση.

Το summary μας δίνει μερικές σημαντικές τιμές για την μεταβλητή spend, δηλαδή το ελάχιστο(min), το πρώτο τεταρτημόριο(1st Qu.), τη δειγματική διάμεσο(Median), το δειγματικό μέσο(Mean), το τρίτο τεταρτημόριο(3rd Qu.) και το μέγιστο(Max).

```
> summary(age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   27.00   35.00   34.43   39.00   69.00
```

Έχει εξηγηθεί παραπάνω ο τρόπος με τον οποίο φτιάχνουμε τις γραφικές παραστάσεις. Είναι προφανές ότι οι περισσότεροι που ξοδεύουν χρήματα στις εορτές είναι μεταξύ 20-40. Επίσης παρατηρούμε καθώς η ηλικία ανεβαίνει μετά τα 40 τόσο ο κόσμος χαλάει λιγότερα χρήματα.





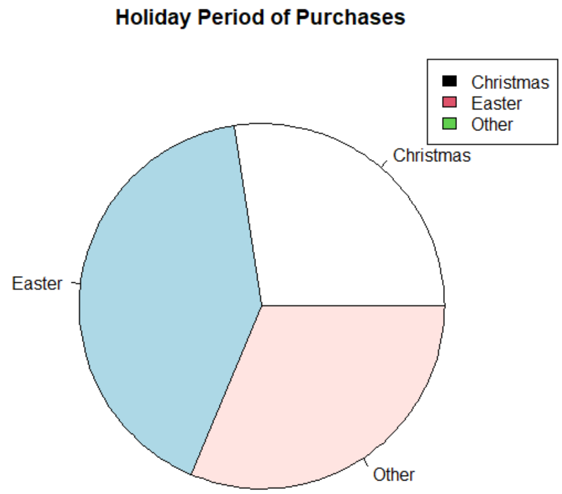
Με την εντολή `fivenum(spend)` παίρνουμε τα παρακάτω και μπορούμε να επαληθεύσουμε τα λεγόμενα μας.

```
[1] 18 27 35 39 69
```

Holiday :

```
holiday_freq <- table(gifts$holiday)
pie(holiday_freq, main="Holiday Period of Purchases")
legend("topright", c("Christmas", "Easter", "Other"), fill=1:3)
```

Για αυτή τη μεταβλητή θα χρησιμοποιήσουμε άλλη μορφή διαγράμματος την Pie. Αρχικά χρησιμοποιούμε την εντολή table() για να δούμε την ακριβή συχνότητα με την οποία εμφανίζεται κάθε εορτή στο δείγμα μας. Έπειτα χρησιμοποιώ την εντολή pie() που τις δίνω τίτλο με το argument main. Τέλος χρησιμοποιούμε την εντολη legend για να δώσουμε λεζάντες στην πίτα μας.

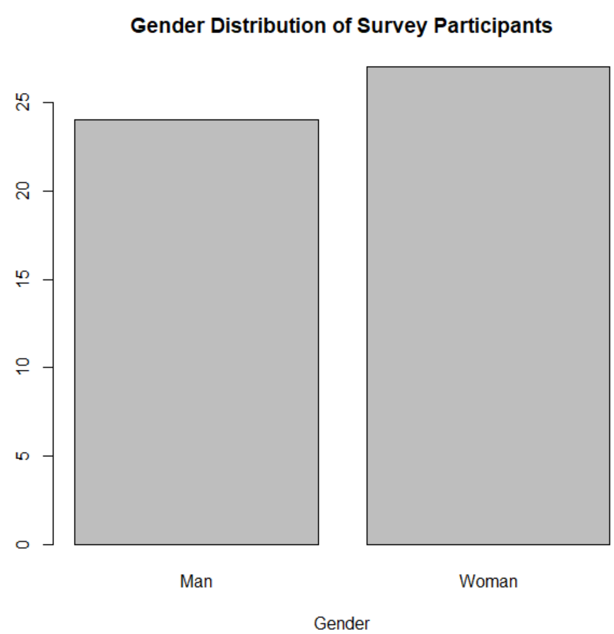


Παρατηρούμε ότι στις γιορτές που πραγματοποιήθηκαν οι περισσότερες αγορές ήταν το Πάσχα.

<u>Sex :</u>

Εδώ θα χρησιμοποιήσω ένα διάγραμμα barplot για την γραφική επεξήγηση της μεταβλητής.

```
barplot(table(sex), main="Gender Distribution of Survey Participants",
xlab="Gender")
```



Από το barplot είναι προφανές ότι οι γυναίκες κατά λίγο ψωνίζουν πιο πολύ από τους άνδρες κατά την περίοδο των γιορτών.

```
table(sex)
prop.table(table(sex))
```

Με την εντολή table() βλέπουμε την ακριβή συχνότητα με την οποία εμφανίζεται το κάθε φύλο (ή ηλικία) στο δείγμα μας. Με τη δεύτερη εντολή παίρνουμε τη σχετική συχνότητα, η οποία είναι προφανές πως αθροίζει στο 1

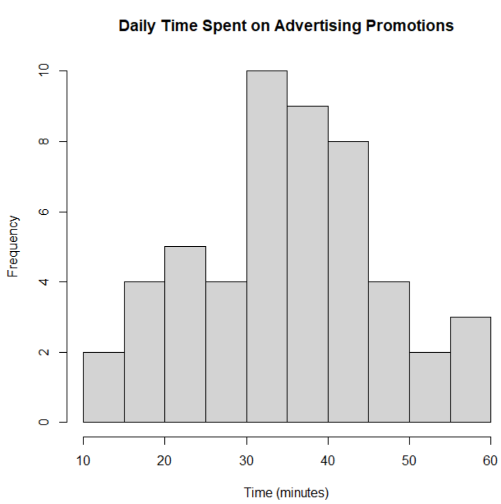
```
> table(sex)
sex
  Man Woman
   24    27
```

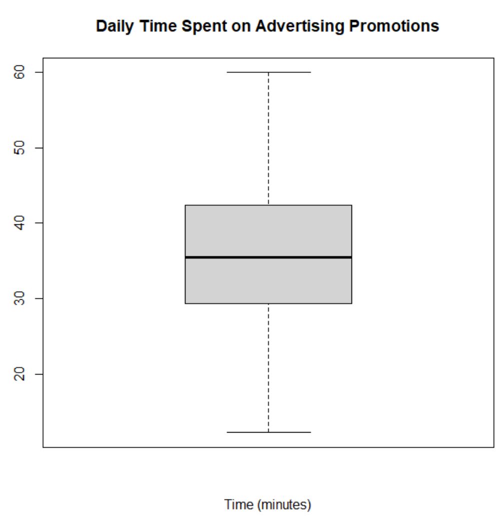
```
> prop.table(table(sex))
sex
      Man     Woman
0.4705882 0.5294118
```

Time :

```
class(time)
summary(time)
hist(time, main="Daily Time Spent on Advertising Promotions", xlab="Time (minutes)")
boxplot(time, main="Daily Time Spent on Advertising Promotions", xlab="Time (minutes)")
fivenum(time)
```

Με την εντολή class `class(spend)` μας δίνεται ο τύπος δεδομένων της μεταβλητής spend ο οποίος είναι ‘numeric’ δηλαδή αριθμητική κλάση.





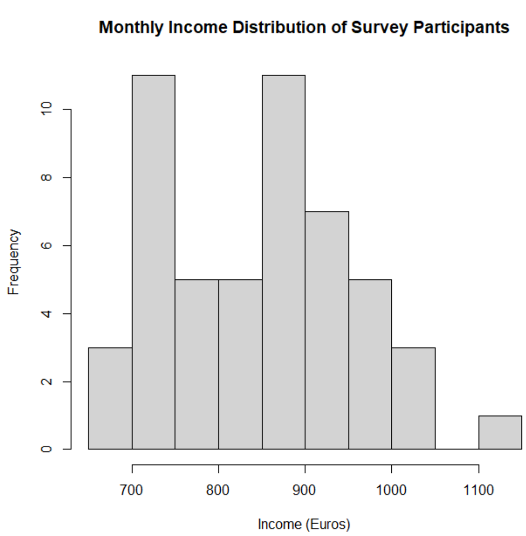
Από τα διαγράμματα και το παρακάτω output από το summary() μπορούμε να δούμε ότι ο μέσος όρος και ο διάμεσος είναι σχετικά κοντά μεταξύ τους, υποδεικνύοντας ότι τα δεδομένα δεν είναι πολύ . Ωστόσο, μπορούμε να δούμε ότι το εύρος των τιμών είναι αρκετά μεγάλο, με την ελάχιστη τιμή να είναι 12,20 και τη μέγιστη τιμή να είναι 60,00.

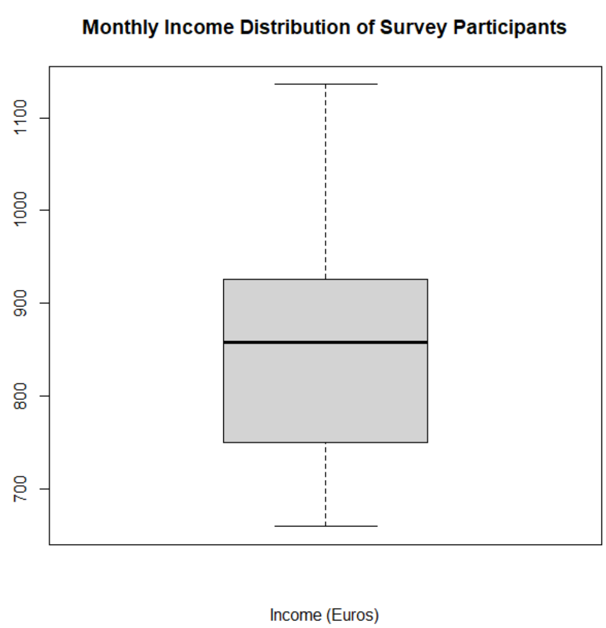
```
> summary(time)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  12.20   29.35   35.50   35.24   42.45   60.00
```

Salary :

Όμοια θα ασχοληθω και με το salary

```
class(salary)
summary(salary)
hist(salary, main="Monthly Income Distribution of Survey Participants",
xlab="Income (Euros)")
boxplot(salary, main="Monthly Income Distribution of Survey Participants",
xlab="Income (Euros)")
fivenum(salary)
```





Με βάση τα διαγράμματα και το παρακάτω output από το summary() μπορούμε να δούμε ότι το ελάχιστο μηνιαίο εισόδημα των συμμετεχόντων στην έρευνα είναι 659 ευρώ, ενώ το μέγιστο είναι 1136 ευρώ. Το μεσαίο εισόδημα είναι 858 ευρώ, που σημαίνει ότι οι μισοί από τους συμμετέχοντες έχουν εισόδημα κάτω από αυτήν την αξία και οι μισοί πάνω από αυτήν. Το μέσο μηνιαίο εισόδημα είναι 847,5 ευρώ, το οποίο είναι ελαφρώς χαμηλότερο από το διάμεσο, γεγονός που δείχνει ότι η κατανομή του εισοδήματος είναι ελαφρώς ασυμμτρο προς τα αριστερά. Το πρώτο τεταρτημόριο (1ο τρίμηνο) είναι 749,5 ευρώ, που σημαίνει ότι το 25% των συμμετεχόντων έχει μηνιαίο εισόδημα κάτω από αυτήν την τιμή, ενώ το τρίτο τεταρτημόριο (3ο τρίμηνο) είναι 926 ευρώ, υποδηλώνοντας ότι το 75% των συμμετεχόντων έχει μηνιαίο εισόδημα κάτω από αυτό. Συνολικά, η κατανομή του μηνιαίου εισοδήματος φαίνεται να είναι σχετικά περιορισμένη, με τους περισσότερους συμμετέχοντες να δουλεύουν μεταξύ 750 και 950 ευρώ το μήνα. Ωστόσο, υπάρχουν κάποιες ακραίες τιμές με εισοδήματα άνω των 1000 ευρώ το μήνα.

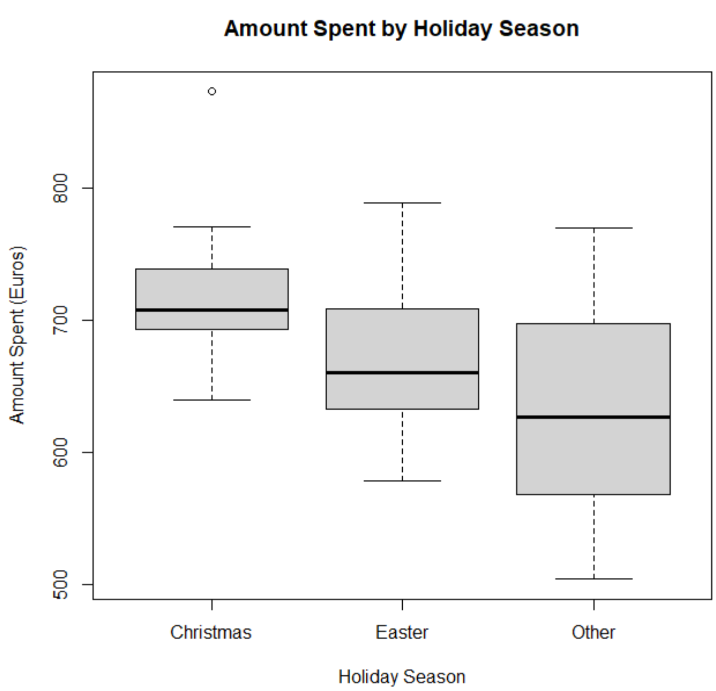
```
> summary(salary)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  659.0   749.5   858.0   847.5   926.0  1136.0
```

ii)

Spend ~ Holiday

Θα χρησιμοποιήσω το παρακάτω κώδικα για να φτιάξω το διάγραμμα (spend ~ holiday).

```
boxplot(spend ~ holiday,
        main = "Amount Spent by Holiday Season",
        xlab = "Holiday Season",
        ylab = "Amount Spent (Euros)")
```

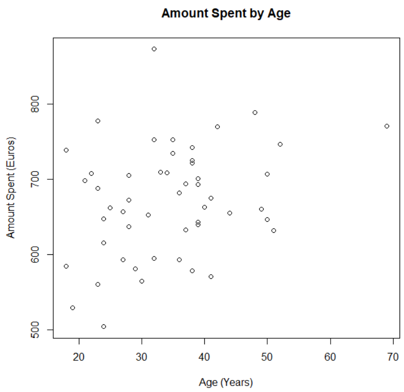


Το boxplot(spend ~ holiday) δείχνει την κατανομή του ποσού που δαπανήθηκε από τους καταναλωτές κατά τη διάρκεια διαφορετικών περιόδων εορτών. Το median ποσό που δαπανήθηκε κατά τη διάρκεια του Πάσχα και άλλων εορτών είναι παρόμοιο, ενώ το median ποσό που δαπανήθηκε κατά τη διάρκεια των Χριστουγέννων είναι ελαφρώς υψηλότερο. Το box για το Πάσχα είναι ελαφρώς μεγαλύτερο από τα άλλα δύο, υποδηλώνοντας μεγαλύτερη μεταβλητότητα στις δαπάνες κατά τη διάρκεια του Πάσχα.

Spend ~ Age

Θα χρησιμοποιήσω το παρακάτω κώδικα για να φτιάξω το διάγραμμα (spend ~ age):

```
plot(spend ~ age, data = gifts, main = "Amount Spent by Age",
     xlab = "Age (Years)", ylab = "Amount Spent (Euros)")
```

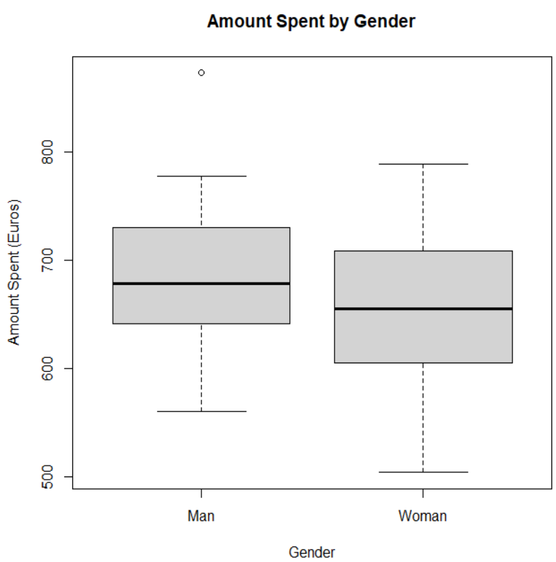


Το παρακάτω διάγραμμα δείχνει τη σχέση μεταξύ της ηλικίας των συμμετεχόντων στην έρευνα και του ποσού που ξόδεψαν κατά την περίοδο των διακοπών. Κάθε κουκκίδα στο διάγραμμα αντιπροσωπεύει έναν συμμετέχοντα στην έρευνα και η θέση της κουκκίδας στον άξονα x υποδεικνύει την ηλικία του συμμετέχοντος, ενώ η θέση στον άξονα y υποδεικνύει το ποσό που δαπανήθηκε. Το γράφημα φαίνεται να δείχνει μια ελαφρά θετική σχέση μεταξύ της ηλικίας και του ποσού που δαπανήθηκε, με τους μεγαλύτερους συμμετέχοντες να ξοδεύουν γενικά περισσότερα από τους νεότερους. Ωστόσο, υπάρχει μεγάλη μεταβλητότητα στα δεδομένα και η σχέση δεν είναι ιδιαίτερα ισχυρή.

Spend ~ Sex

Θα χρησιμοποιήσω το παρακάτω κώδικα για να φτιάξω το διάγραμμα (spend ~ sex).

```
boxplot(spend ~ sex,
        main = "Amount Spent by Gender",
        xlab = "Gender",
        ylab = "Amount Spent (Euros)")
```

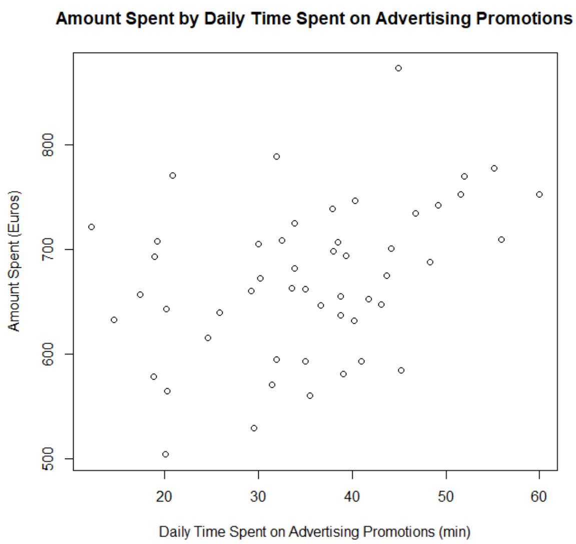


Το διάγραμμα δείχνει την κατανομή του ποσού που δαπανάται από άνδρες και γυναίκες καταναλωτές. Στη μέση του κουτιού δείχνει πού βρίσκονται τα περισσότερα σημεία δεδομένων, με τη γραμμή μέσα να δείχνει τη διάμεσο. Το γράφημα υποδηλώνει ότι υπάρχει μια μικρή διαφορά στο ποσό που δαπανάται μεταξύ ανδρών και γυναικών καταναλωτών, με τους άνδρες να τείνουν να ξοδεύουν ελαφρώς περισσότερα κατά μέσο όρο.

Spend ~ Time

Θα χρησιμοποιήσω το παρακάτω κώδικα για να φτιάξω το διάγραμμα (spend ~ time).

```
boxplot(spend ~ time,
        main = "Amount Spent by Daily Time Spent on Advertising Promotions",
        xlab = "Daily Time Spent on Advertising Promotions(min)",
        ylab = "Amount Spent (Euros)")
```

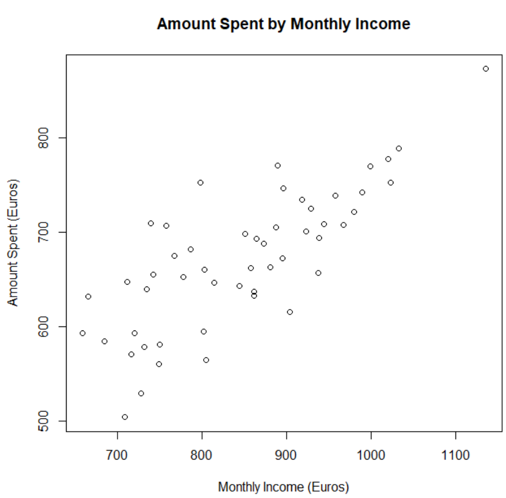


Αυτή η γραφική παράσταση εμφανίζει τη σχέση μεταξύ του ποσού που δαπανάται για δώρα και του ημερήσιου χρόνου που αφιερώνεται σε διαφημίσεις. Κάθε σημείο δεδομένων αντιπροσωπεύει μια αγορά δώρου που έγινε από έναν καταναλωτή. Από το γράφημα, φαίνεται ότι δεν υπάρχει σαφής γραμμική σχέση μεταξύ του ημερήσιου χρόνου που αφιερώνεται σε διαφημίσεις και του ποσού που δαπανάται για δώρα.

Spend ~ Salary

Θα χρησιμοποιήσω το παρακάτω κώδικα για να φτιάξω το διάγραμμα (spend ~ salary).

```
plot(spend ~ salary,
    main = "Amount Spent by Monthly Income",
    xlab = "Monthly Income (Euros)",
    ylab = "Amount Spent (Euros)")
```



Η γραφική παράσταση δείχνει τη σχέση μεταξύ του ποσού που ξοδεύουν οι καταναλωτές και του μηνιαίου εισοδήματός τους. Καθώς το μηνιαίο εισόδημα αυξάνεται, υπάρχει μια τάση οι καταναλωτές να ξοδεύουν περισσότερα χρήματα. Ωστόσο, εξακολουθούν να υπάρχουν ορισμένοι καταναλωτές με σχετικά χαμηλά μηνιαία εισοδήματα που ξοδεύουν πολλά χρήματα. Συνολικά, η γραφική υποδηλώνει ότι το μηνιαίο εισόδημα συσχετίζεται θετικά με το ποσό που δαπανάται, αλλά υπάρχει μεγάλη ποικιλία στις συνήθειες δαπανών ακόμη και μεταξύ των ατόμων με παρόμοια εισοδήματα.

iii)

Για να δημιουργήσουμε την μεταβλητή f_age με 4 κατηγορίες [18-25), [25-38), [38-50), and [50 και άνω), θα χρησιμοποιήσω την εντολή cut():

```
f_age <- cut(age, c(18, 25, 38, 50, Inf), right = FALSE,
             labels = c("[18-25)", "[25-38)", "[38-50)", "[50 and over)"))
```

Τώρα χρειαζόμαστε να δημιουργήσουμε την μεταβλητή f_age με 4 κατηγορίες που θα βασίζονται στα τεταρτημόρια της μεταβλητής χρόνου. Θα χρησιμοποιήσουμε την εντολή quantile() για να υπολογίζουμε τα τεταρτημόρια και την εντολή cut() για να δημιουργήσουμε τις κατηγορίες:

```
q <- quantile(time, probs = c(0, 0.25, 0.5, 0.75, 1))
f_time <- cut(time, q, include.lowest = TRUE,
            labels = c("[0,q1)", "[q1,q2)", "[q2,q3)", "[q3 and above)"))
```

Θα κατασκευάσουμε τον πίνακα συχνοτήτων με την εντολή table() και τον πίνακα σχετικών συχνοτήτων με την εντολή prop.table():

```
table(f_age)
prop.table(table(f_age))
```

> table(f_age)	> prop.table(table(f_age))
[18-25) [25-38) [38-50) [50 and over) 11 20 15 5	[18-25) [25-38) [38-50) [50 and over) 0.21568627 0.39215686 0.29411765 0.09803922

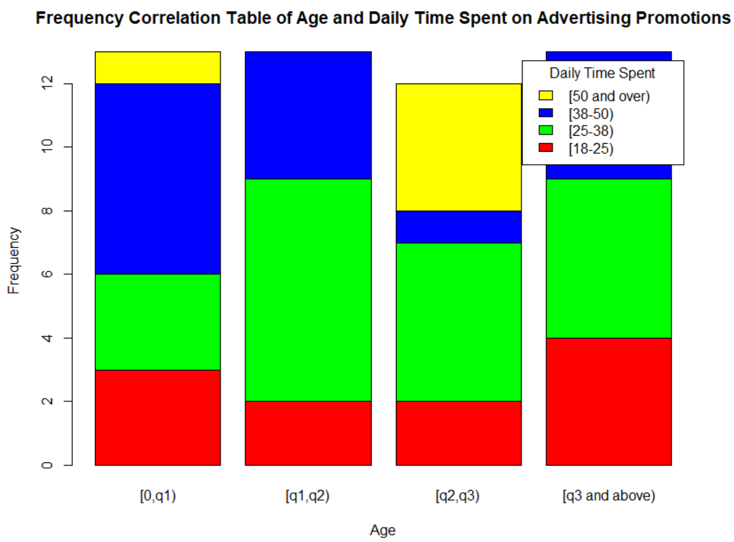
θα κατασκευάσουμε τον πίνακα συνάφειας συχνοτητων των ζητουμενων μεταβλητων ετσι:

```
freq_table <- table(f_age, f_time)
freq_table
```

```
> freq_table
          f_time
f_age          [0,q1) [q1,q2) [q2,q3) [q3 and above)
  [18-25)          3       2       2       4
  [25-38)          3       7       5       5
  [38-50)          6       4       1       4
  [50 and over)    1       0       4       0
```

Για να φτιάξουμε το στοιβαγμένο ραβδόγραμμα θα χρησιμοποιήσουμε την ακόλουθη εντολή:

```
barplot(freq_table, col = c("red","green","blue","yellow"),
        xlab = "Age", ylab = "Frequency",
        main = "Frequency Correlation Table of Age and Daily Time Spent on
Advertising Promotions",
        legend.text = TRUE, args.legend = list(title = "Daily Time Spent"))
```



Το γράφημα στα αριστερά δείχνει την κατανομή των συνδυασμών f_age και f_time. Οι ράβδοι στοιβάζονται έτσι ώστε κάθε ράβδος να αντιπροσωπεύει ένα συγκεκριμένο εύρος ηλικιών και τα διαφορετικά χρώματα αντιπροσωπεύουν το διαφορετικό χρόνο που αφιερώνεται σε εύρη διαφημιστικών προωθητικών ενεργειών. Συνολικά, ο πίνακας και το γράφημα δείχνουν την κατανομή των συνδυασμών του εύρους ηλικιών και του χρόνου που δαπανάται για διαφημιστικές προωθητικές ενέργειες στο δείγμα. Μπορούμε να δούμε ότι δεν υπάρχει σαφές μοτίβο ή τάση μεταξύ του εύρους ηλικιών και του χρόνου που αφιερώνεται σε διαφημιστικές προωθητικές ενέργειες.

Δυστυχώς λόγω του περιορισμού των σελίδων δεν μπόρεσα να γίνω όσο περιγραφικός ήθελα, ειδικά στην επεξήγηση κάποιων εντολών!!!

Άσκηση 2

α)

Δίνουμε τον κώδικα και μετά θα τον περιγράψουμε

```
imitono <- function(n, x) {
  if(!is.numeric(n) || !is.integer(n) || n <= 0) {
    stop("n must be a positive integer")
  }

  sign <- c(1, rep(-1, n))
  terms <- sign * (x^(2 * (0:n) + 1) / factorial(2 * (0:n) + 1))

  return(sum(terms[1:n]))
}
```

Αρχικά θα ορίσω μια νέα συνάρτηση που θα παίρνει δύο ορίσματα, το n(φυσικός αριθμός) και x με την εντολή function. Έπειτα ελέγχουμε αν το n είναι αριθμός, αν είναι μεγαλύτερο ή ίσο του μηδενός και αν η στρογγυλοποίηση του ταυτίζεται με το ίδιο το n(δηλαδή αν το n ανήκει στους φυσικούς αριθμούς). Αν κάτι από αυτά δεν ισχύει τότε θα εμφανιστεί στον χρήστη ένα μήνυμα λάθους. Αν πληρούνται όλα τα παραπάνω κριτήρια τότε θα προσεγγίσουμε το sin(x) με τον τύπο που μας δόθηκε. Αρχικά ορίζουμε την μεταβλητή sign που είναι ένα διάνυσμα με εναλλασσόμενα 1 και -1, ξεκινώντας από 1. Στη συνέχεια, η συνάρτηση υπολογίζει την επέκταση της σειράς Taylor μέχρι τον n-οστό όρο υπολογίζοντας τις δυνάμεις του x για κάθε όρο (x^(2 * (0:n) + 1)) και το παραγοντικά των παρονομαστών κάθε όρου (factorial(2 * (0:n) + 1)). Στη συνέχεια, αυτά τα τρεία διανύσματα συνδυάζονται βάσει στοιχείων και αθροίζονται για να παράγουν την προσέγγιση του sin(x).

β)

Δίνουμε τον κώδικα και μετά θα τον περιγράψουμε

```
imitono2 <- function(n, x) {

  if(!is.numeric(n) || n < 0 || n != round(n)) {
    stop("n must be a positive integer")
  }

  calc_sum <- function(i) {
    if(i == 0) {
      return(x)
    } else {
      term <- (-1)^(i) * x^(2*i+1) / factorial(2*i+1)
      return(term + calc_sum(i-1))
    }
  }

  result <- calc_sum(n)

  return(result)
}
```

Το πρώτο κομμάτι είναι με τους περιορισμούς παραμένει ίδιο. Σε αυτή την περίπτωση χρησιμοποιούμε το function calc_sum() που χρησιμοποιεί αναδρομή. Η συνάρτηση calc_sum παίρνει μια παράμετρο εισόδου i, η οποία αντιπροσωπεύει την τρέχουσα επανάληψη. Εάν το i είναι μηδέν, επιστρέφει την τιμή του x. Διαφορετικά, υπολογίζει τον i-οστό όρο χρησιμοποιώντας τον τύπο που μας δόθηκε, προσθέτει αυτόν τον όρο στο άθροισμα των προηγούμενων όρων (υπολογίζεται καλώντας το calc_sum(i-1) αναδρομικά) και επιστρέφει το αποτέλεσμα. Τέλος, η συνάρτηση καλεί το calc_sum(n) για να υπολογίσει προσεγγιστικά το sin(x) μέχρι τον n-οστό όρο και επιστρέφει το αποτέλεσμα στον χρήστη.

γ)

Δίνουμε τον κώδικα και μετά θα τον περιγράψουμε:

```
imitono3 <- function(x) {
  tol <- 0.001
  counter <- 0
  n <- 0
  imitono_old <- 0
  imitono3 <- x
  err <- abs(sin(x) - imitono3)

  while (err > tol) {
    n <- n + 1
    imitono_old <- imitono3
    imitono3 <- imitono_old + ((-1)^n * x^(2*n+1)) / factorial(2*n+1)
    err <- abs(sin(x) - imitono3)
    counter <- counter + 1
  }

  cat("The approximation of sin(", x, ") using the Taylor series with a maximum
error tolerance of", tol, "is:", imitono3, "\n")
  cat("The approximation took", counter, "iterations to achieve the specified
error tolerance of", tol, "\n")
}
```

Και σε αυτή περίπτωση ορίζουμε μια συνάρτηση με μια μεταβλητή x με την εντολή function(). Έπειτα ορίζουμε το μέγιστο όριο σε 0.001, ορίζουμε την μεταβλητή που θα μετράει τις επαναλήψεις ίσο με 0 και το imitono_old συμβολίζει την τιμή της προηγούμενης προσέγγισης, και ίσο με 0. Χρησιμοποιούμε την εντολή while για να ενημερώνουμε σε κάθε επανάληψη του βρόγχου το n στον τρέχοντα αριθμό επανάληψης, το imitono_old στην προηγουμένη προσέγγιση χρησιμοποιώντας τον δοσμένο τύπο. Ενημερώνεται επίσης και το err στο τρέχων σφάλμα, ενημερώνουμε το counter για να παρακολουθούμε τον αριθμό των επαναλήψεων. Τέλος προσθέτουμε τις εντολές για να εκτυπώνουμε την τελική τιμή προσέγγισης του imitono(x) και τον αριθμό των επαναλήψεων που χρειάστηκαν για να επιτευχθεί η καθορισμένη ανοχή σφάλματος.

δ) Θα σχεδιάσω στο ίδιο διάγραμμα τη συνάρτηση imitono2 και την sin(x). Παρακάτω θα δώσω μόνο τον εξτρά κώδικα για την δημιουργία των γραφικών παραστάσεων, τον κώδικα για το imitono2 είναι ακριβώς από πάνω.

```
x <- seq(0, 2*pi, length.out=1000)

plot(x, sin(x), type="l", col="blue", xlab="x", ylab="y")

for(n in 1:10) {
  lines(x, imitono2(n, x), col=rainbow(10)[n], lty=2)

}
```

```
legend("topright", legend=c("sin(x)", paste0("imitono2(", 1:10, ",x)")),
       col=c("blue", rainbow(10)), lty=c(1, rep(2, 10)))
```

Η πρώτη γραμμή του κώδικα δημιουργεί μια ακολουθία 1000 τιμών με ιδίες διαμερίσεις μεταξύ του 0 και 2π, οι οποίες θα χρησιμοποιηθούν ως τιμές του άξονα x για το plot. Η τρίτη γραμμή του κώδικα δημιουργεί το διάγραμμα της συνάρτησης sin(x) με τις τιμές του άξονα x να παίρνονται από την παραπάνω ακολουθία και τις τιμές του y από την συνάρτηση sin(x). Επίσης έχουμε ορίσει το χρώμα της sin(x) να είναι μπλε καθώς όρισα και ονόματα στους άξονες. Στην πέμπτη και έκτη γραμμή του κώδικα δημιουργώ 10 καμπύλες, από μία έως δέκα επαναλήψεις της προσέγγισης του imitono2, για να δείξω ότι για n≥10 η imitono2(n,x) ταυτίζεται με την sin(x). Τέλος στην ένατη γραμμή κώδικα βάζω ένα μικρό πλαίσιο για τις γραφικές παραστάσεις υποδεικνύοντας ποια καμπύλη αντιστοιχεί σε ποιο n.