Homework

UK Lineare Multivariate Statistik

Cordula Eggerth

Matrikelnummer: 00750881

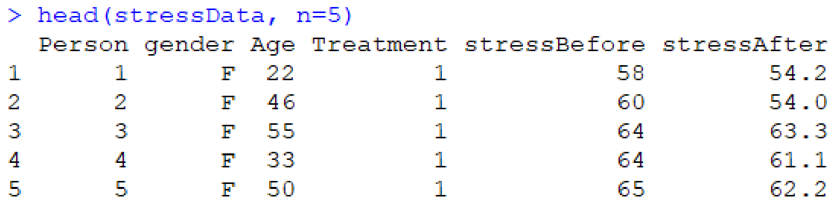
Sommersemester 2019

Aufgabe 1:

Three types of medical treatments for stress reduction were tested on groups of males and females. For each person it is provided which treatment was used ("1", "2", "3") and the stress level before and after the treatment. We are interested whether these treatments are equally effective and whether the gender plays a role. Read the stressData.csv into R and then do the following. + Provide a report (pdf format) which contains your code and its outputs, as well as corresponding plots from R. Summarize the conclusions of your analysis.

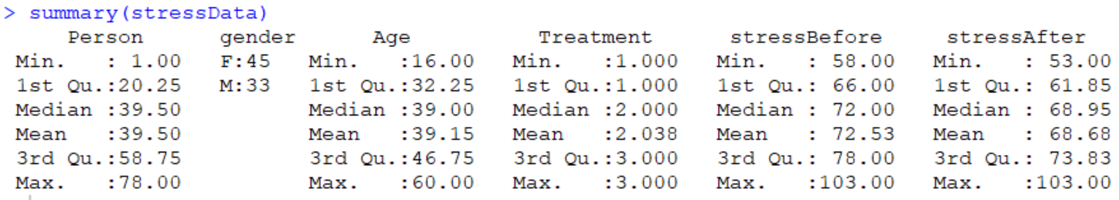
1. Graphically represent the data (for example with box plots), calculate the means and medians of changes in stress levels, taking both factors (treatment type and gender) into consideration.

The dataset has 78 observations (rows) and 6 columns.

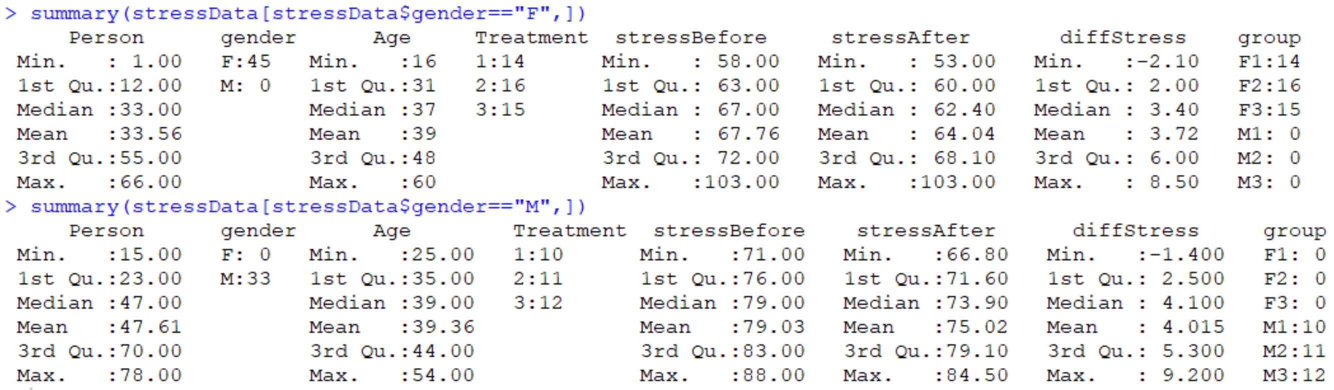


```
> head(stressData, n=5)
  Person gender Age Treatment stressBefore stressAfter
1      1      F  22         1           58        54.2
2      2      F  46         1           60        54.0
3      3      F  55         1           64        63.3
4      4      F  33         1           64        61.1
5      5      F  50         1           65        62.2
```

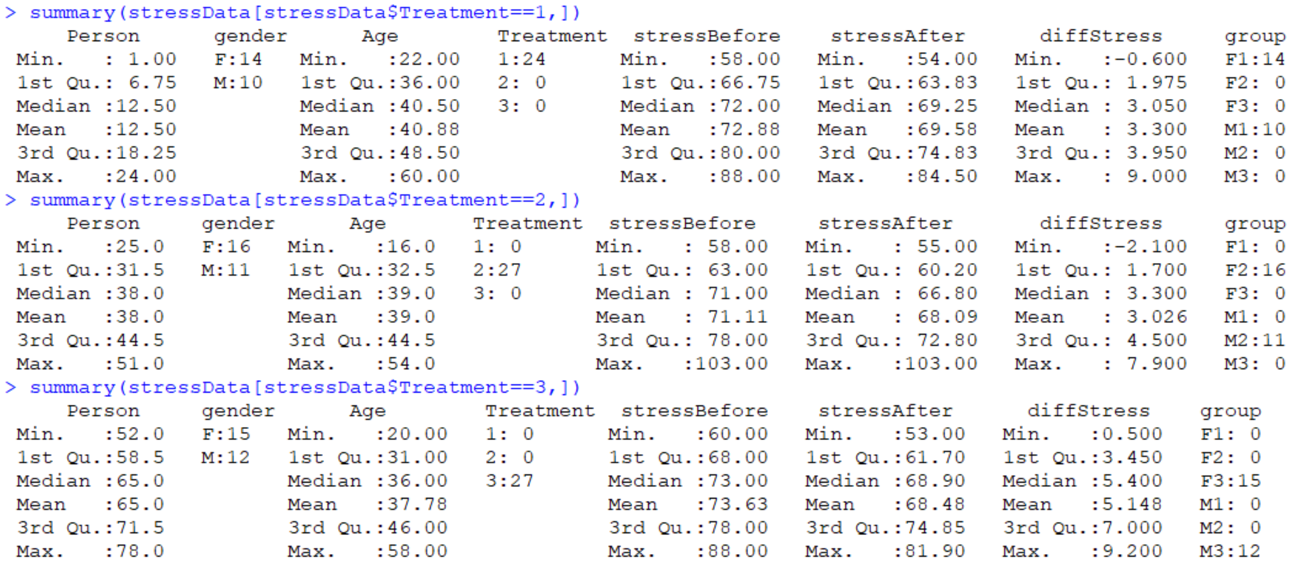
Descriptive statistics:



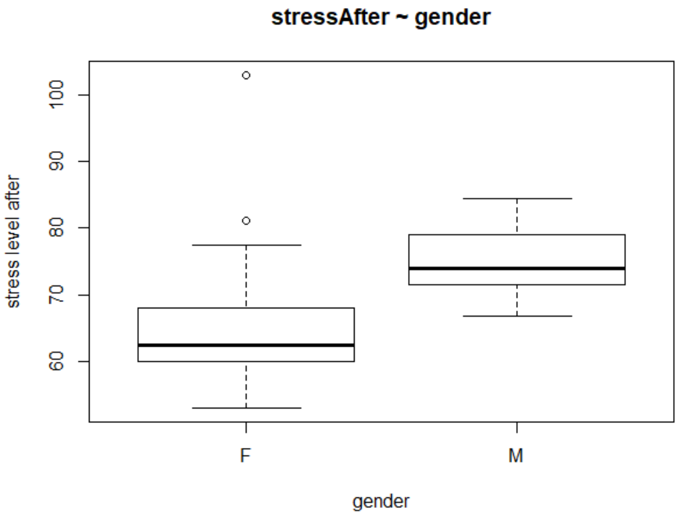
```
> summary(stressData)
     Person      gender      Age          Treatment      stressBefore      stressAfter    
 Min.   : 1.00   F:45   Min.   :16.00   Min.   :1.000   Min.   : 58.00   Min.   : 53.00  
 1st Qu.:20.25   M:33   1st Qu.:32.25   1st Qu.:1.000   1st Qu.: 66.00   1st Qu.: 61.85  
 Median :39.50          Median :39.00   Median :2.000   Median : 72.00   Median : 68.95  
 Mean   :39.50          Mean   :39.15   Mean   :2.038   Mean   : 72.53   Mean   : 68.68  
 3rd Qu.:58.75          3rd Qu.:46.75   3rd Qu.:3.000   3rd Qu.: 78.00   3rd Qu.: 73.83  
 Max.   :78.00          Max.   :60.00   Max.   :3.000   Max.   :103.00   Max.   :103.00  
```

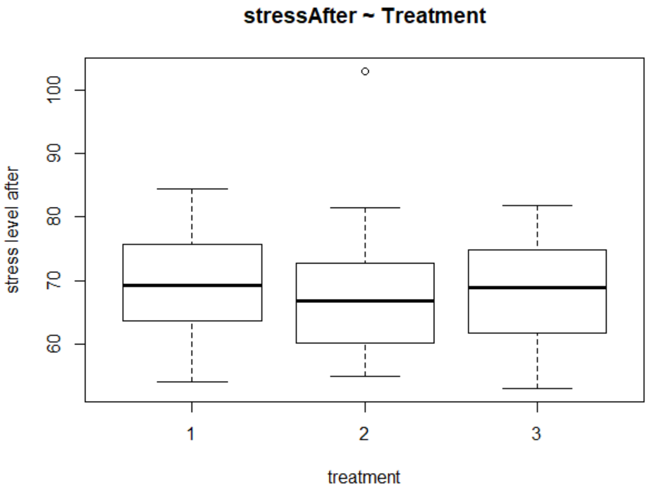


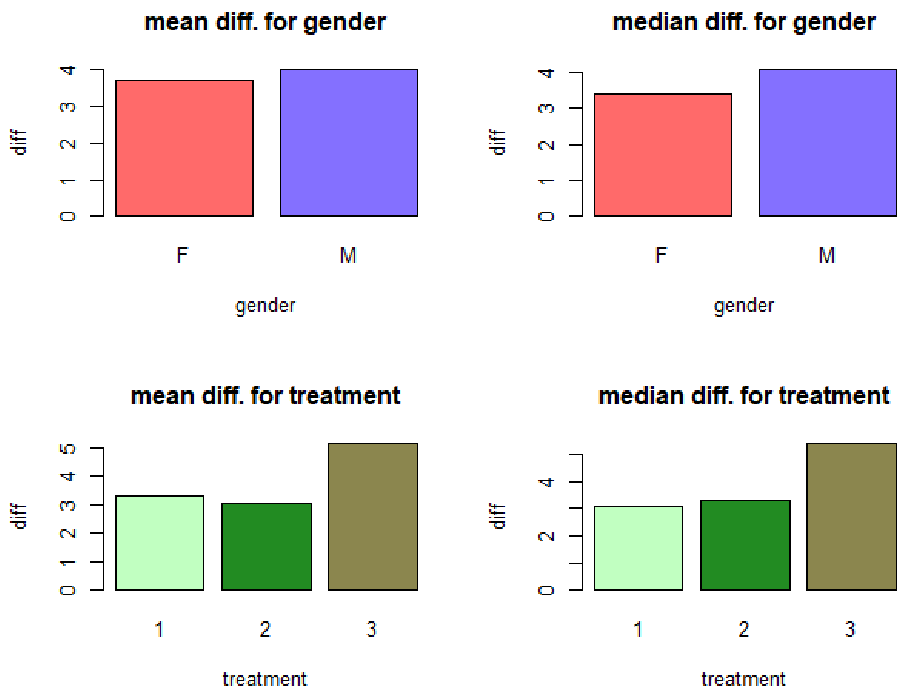
```
> summary(stressData[stressData$gender=="F",])
     Person      gender      Age      Treatment  stressBefore      stressAfter        diffStress      group  
 Min.   : 1.00   F:45   Min.   :16   1:14      Min.   : 58.00   Min.   : 53.00   Min.   :-2.10   F1:14  
 1st Qu.:12.00   M: 0   1st Qu.:31   2:16      1st Qu.: 63.00   1st Qu.: 60.00   1st Qu.: 2.00   F2:16  
 Median :33.00          Median :37   3:15      Median : 67.00   Median : 62.40   Median : 3.40   F3:15  
 Mean   :33.56          Mean   :39             Mean   : 67.76   Mean   : 64.04   Mean   : 3.72   M1: 0  
 3rd Qu.:55.00          3rd Qu.:48             3rd Qu.: 72.00   3rd Qu.: 68.10   3rd Qu.: 6.00   M2: 0  
 Max.   :66.00          Max.   :60             Max.   :103.00   Max.   :103.00   Max.   : 8.50   M3: 0  
> summary(stressData[stressData$gender=="M",])
     Person      gender      Age          Treatment  stressBefore     stressAfter      diffStress        group   
 Min.   :15.00   F: 0   Min.   :25.00   1:10       Min.   :71.00   Min.   :66.80   Min.   :-1.400   F1: 0  
 1st Qu.:23.00   M:33   1st Qu.:35.00   2:11       1st Qu.:76.00   1st Qu.:71.60   1st Qu.: 2.500   F2: 0  
 Median :47.00          Median :39.00   3:12       Median :79.00   Median :73.90   Median : 4.100   F3: 0  
 Mean   :47.61          Mean   :39.36              Mean   :79.03   Mean   :75.02   Mean   : 4.015   M1:10  
 3rd Qu.:70.00          3rd Qu.:44.00              3rd Qu.:83.00   3rd Qu.:79.10   3rd Qu.: 5.300   M2:11  
 Max.   :78.00          Max.   :54.00              Max.   :88.00   Max.   :84.50   Max.   : 9.200   M3:12  
```



```
> summary(stressData[stressData$Treatment==1,])
     Person          gender      Age         Treatment  stressBefore     stressAfter       diffStress        group
 Min.   : 1.00   F:14    Min.   :22.00   1:24       Min.   :58.00   Min.   :54.00   Min.   :-0.600   F1:14
 1st Qu.: 6.75   M:10    1st Qu.:36.00   2: 0       1st Qu.:66.75   1st Qu.:63.83   1st Qu.: 1.975   F2: 0
 Median :12.50           Median :40.50   3: 0       Median :72.00   Median :69.25   Median : 3.050   F3: 0
 Mean   :12.50           Mean   :40.88              Mean   :72.88   Mean   :69.58   Mean   : 3.300   M1:10
 3rd Qu.:18.25           3rd Qu.:48.50              3rd Qu.:80.00   3rd Qu.:74.83   3rd Qu.: 3.950   M2: 0
 Max.   :24.00           Max.   :60.00              Max.   :88.00   Max.   :84.50   Max.   : 9.000   M3: 0
> summary(stressData[stressData$Treatment==2,])
     Person        gender      Age        Treatment  stressBefore      stressAfter        diffStress        group
 Min.   :25.0   F:16    Min.   :16.0   1: 0       Min.   : 58.00   Min.   : 55.00   Min.   :-2.100   F1: 0
 1st Qu.:31.5   M:11    1st Qu.:32.5   2:27       1st Qu.: 63.00   1st Qu.: 60.20   1st Qu.: 1.700   F2:16
 Median :38.0           Median :39.0   3: 0       Median : 71.00   Median : 66.80   Median : 3.300   F3: 0
 Mean   :38.0           Mean   :39.0              Mean   : 71.11   Mean   : 68.09   Mean   : 3.026   M1: 0
 3rd Qu.:44.5           3rd Qu.:44.5              3rd Qu.: 78.00   3rd Qu.: 72.80   3rd Qu.: 4.500   M2:11
 Max.   :51.0           Max.   :54.0              Max.   :103.00   Max.   :103.00   Max.   : 7.900   M3: 0
> summary(stressData[stressData$Treatment==3,])
     Person        gender      Age         Treatment  stressBefore     stressAfter      diffStress      group
 Min.   :52.0   F:15    Min.   :20.00   1: 0       Min.   :60.00   Min.   :53.00   Min.   :0.500   F1: 0
 1st Qu.:58.5   M:12    1st Qu.:31.00   2: 0       1st Qu.:68.00   1st Qu.:61.70   1st Qu.:3.450   F2: 0
 Median :65.0           Median :36.00   3:27       Median :73.00   Median :68.90   Median :5.400   F3:15
 Mean   :65.0           Mean   :37.78              Mean   :73.63   Mean   :68.48   Mean   :5.148   M1: 0
 3rd Qu.:71.5           3rd Qu.:46.00              3rd Qu.:78.00   3rd Qu.:74.85   3rd Qu.:7.000   M2: 0
 Max.   :78.0           Max.   :58.00              Max.   :88.00   Max.   :81.90   Max.   :9.200   M3:12
```





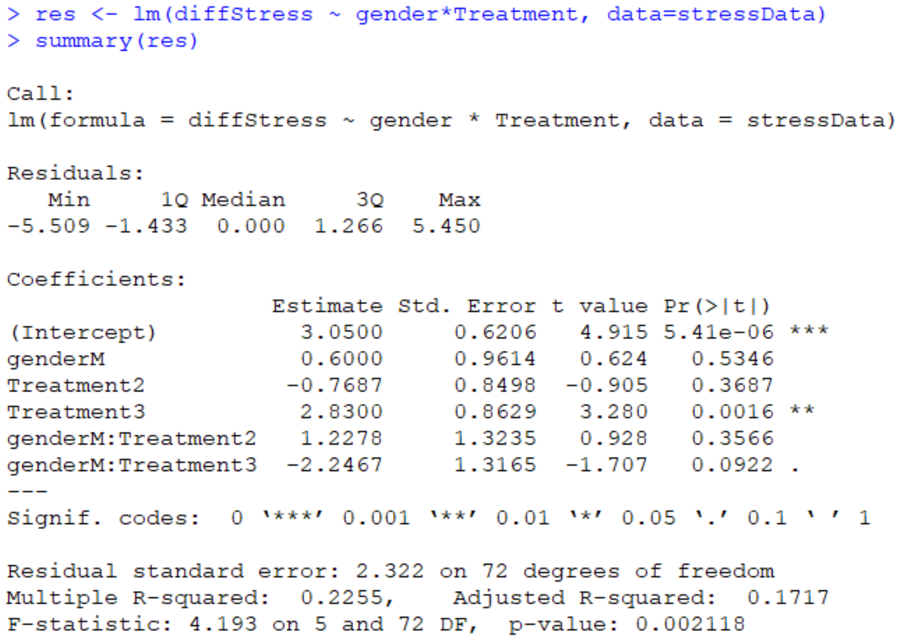


Mean and median differences in stress for all groups:

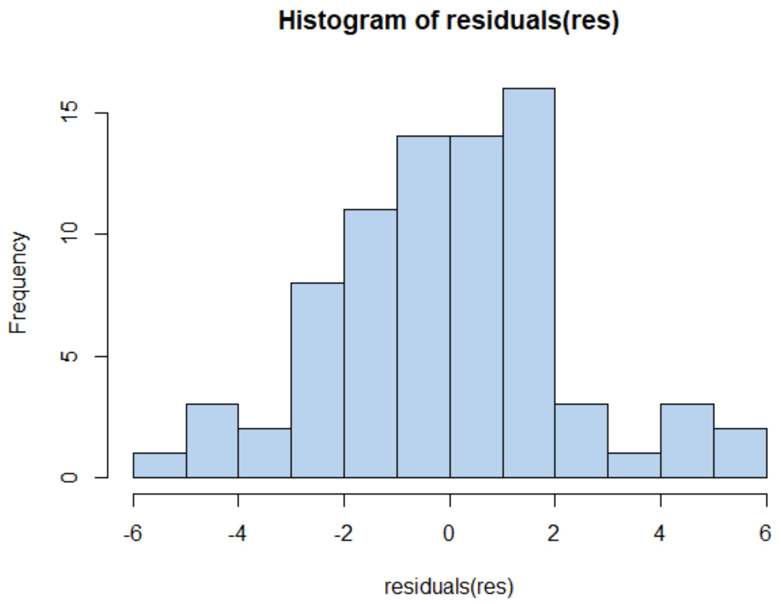
```
> diffs.gt.median
     F   M
1 2.85 3.7
2 2.35 4.3
3 6.80 3.8
> diffs.gt.mean
        F        M
1 3.05000 3.650000
2 2.28125 4.109091
3 5.88000 4.233333
```

2. Using R-built-in functions analyze whether the standard analysis of variance (ANOVA) assumptions are met (normal data and equal variances).

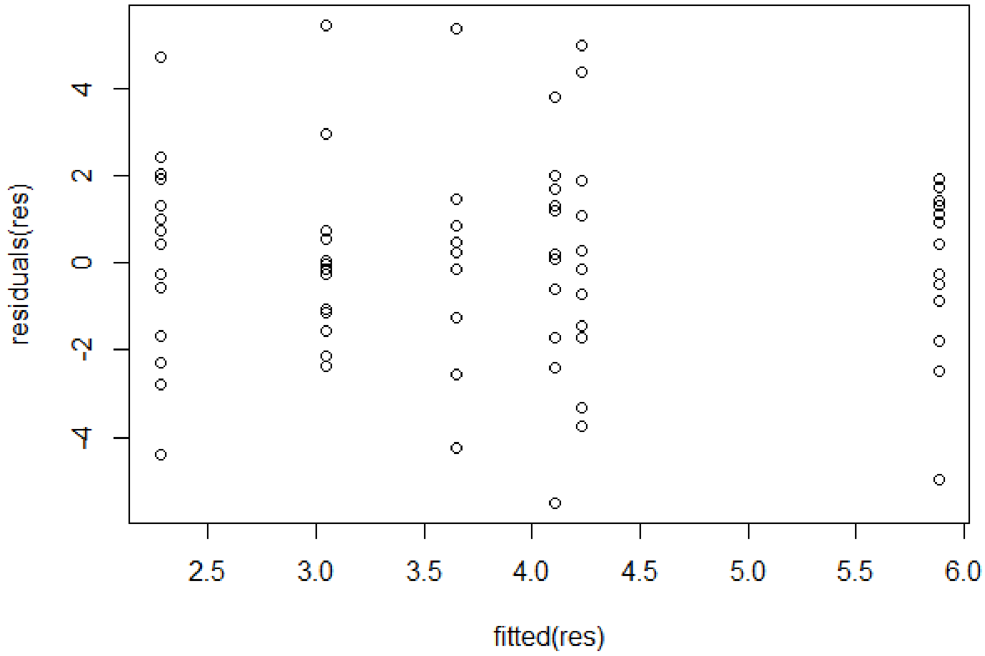
Check if the residuals are normal:



```
> res <- lm(diffStress ~ gender*Treatment, data=stressData)
> summary(res)

Call:
lm(formula = diffStress ~ gender * Treatment, data = stressData)

Residuals:
   Min     1Q Median     3Q    Max 
-5.509 -1.433  0.000  1.266  5.450 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)           3.0500     0.6206   4.915 5.41e-06 ***
genderM               0.6000     0.9614   0.624   0.5346    
Treatment2           -0.7687     0.8498  -0.905   0.3687    
Treatment3            2.8300     0.8629   3.280   0.0016 ** 
genderM:Treatment2    1.2278     1.3235   0.928   0.3566    
genderM:Treatment3   -2.2467     1.3165  -1.707   0.0922 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.322 on 72 degrees of freedom
Multiple R-squared:  0.2255,	Adjusted R-squared:  0.1717 
F-statistic: 4.193 on 5 and 72 DF,  p-value: 0.002118
```

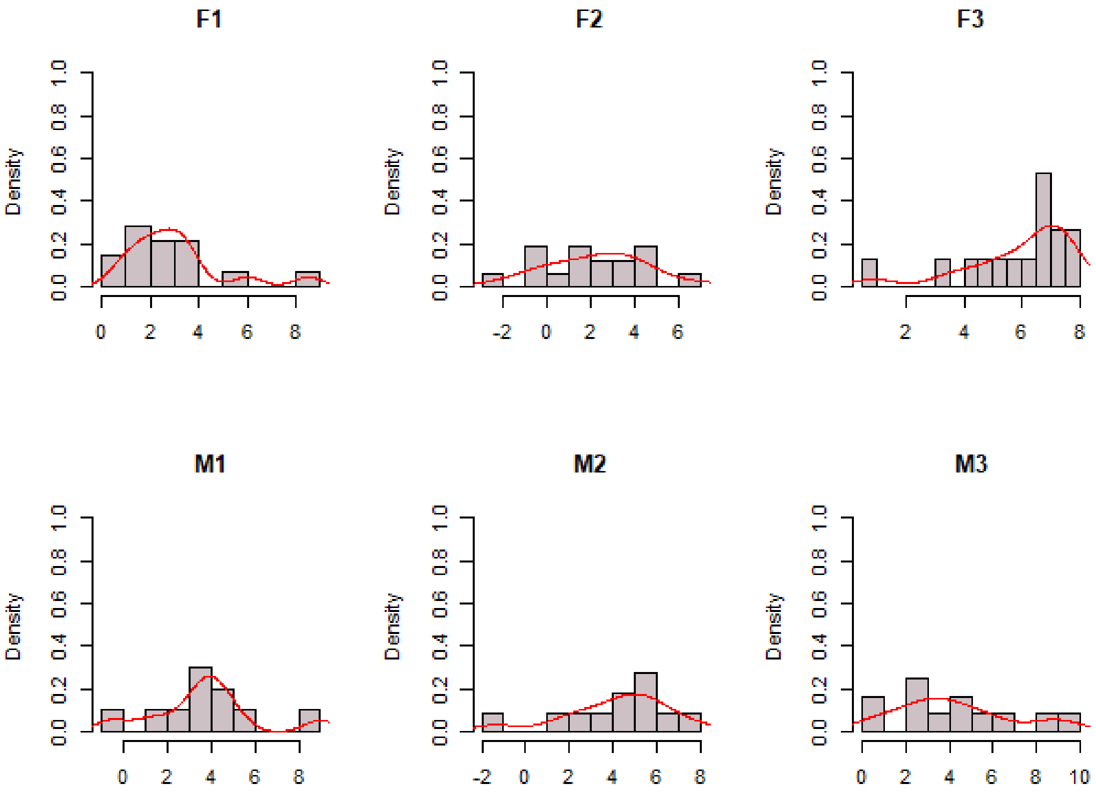


Check if the residuals have equal variance:

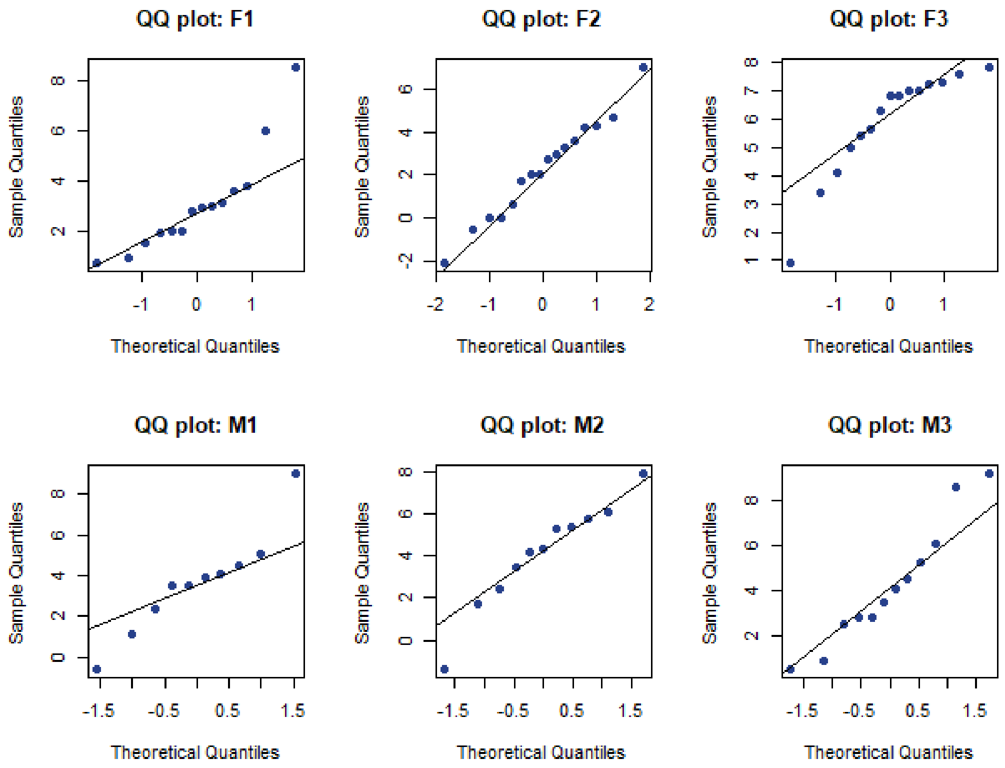


Check (for the groups) if the data is normal:

Histograms for all groups:



QQ-plots for all groups:



Shapiro-Wilk Test for Normality:

```
> shapiro.test(stressData$diffStress[stressData$group=="F1"]) # not normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "F1"]
W = 0.8518, p-value = 0.0235

> shapiro.test(stressData$diffStress[stressData$group=="F2"]) # normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "F2"]
W = 0.98523, p-value = 0.9917

> shapiro.test(stressData$diffStress[stressData$group=="F3"]) # not normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "F3"]
W = 0.8492, p-value = 0.01692

> shapiro.test(stressData$diffStress[stressData$group=="M1"]) # normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "M1"]
W = 0.93667, p-value = 0.5166
```

```
> shapiro.test(stressData$diffStress[stressData$group=="M2"]) # normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "M2"]
W = 0.95022, p-value = 0.6468

> shapiro.test(stressData$diffStress[stressData$group=="M3"]) # normal

        Shapiro-Wilk normality test

data:  stressData$diffStress[stressData$group == "M3"]
W = 0.94143, p-value = 0.5168
```

Anderson-Darling Test for Normality:

```
> ad.test(stressData$diffStress[stressData$group=="F1"]) # not normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "F1"]
A = 0.77577, p-value = 0.03302

> ad.test(stressData$diffStress[stressData$group=="F2"]) # normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "F2"]
A = 0.16682, p-value = 0.9229

> ad.test(stressData$diffStress[stressData$group=="F3"]) # not normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "F3"]
A = 0.82207, p-value = 0.02555

> ad.test(stressData$diffStress[stressData$group=="M1"]) # normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "M1"]
A = 0.39556, p-value = 0.3012

> ad.test(stressData$diffStress[stressData$group=="M2"]) # normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "M2"]
A = 0.29421, p-value = 0.5333

> ad.test(stressData$diffStress[stressData$group=="M3"]) # normal

        Anderson-Darling normality test

data:  stressData$diffStress[stressData$group == "M3"]
A = 0.29933, p-value = 0.5273
```

→ **Result**: The data is not normal; not all groups are normal.

Check if the variances are equal:

Bartlett test of homogeneity of variances (data should ideally be normal):

```
> bartlett.test(diffStress ~ group, data=stressData) # equal variances

	Bartlett test of homogeneity of variances

data:  diffStress by group
Bartlett's K-squared = 2.1994, df = 5, p-value = 0.8209
```

Levene Test (also appropriate for non-normal data):

```
 # levene test (also for non-normal data)
> leveneTest(diffStress ~ group, data=stressData) # equal variances
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  5  0.4351 0.8227
      72
```

Fligner-Killeen Test (also appropriate for non-normal data):

```
> fligner.test(diffStress ~ group, data=stressData) # equal variances

	Fligner-Killeen test of homogeneity of variances

data:  diffStress by group
Fligner-Killeen:med chi-squared = 2.5808, df = 5, p-value = 0.7643
```

→ **Result**: The H0 that the variances are equal is not rejected.

3. Perform the analysis of variance (ANOVA).

4. Can we conclude that all groups perform similarly? If not, use the standard post-hoc tests such as Bonferroni, Tukey, etc. to investigate further.

If the assumptions were met, ANOVA would be appropriate. This would give the result:

```
> res.aov <- aov(diffStress ~ gender * Treatment, data=stressData)
> summary(res.aov)
                 Df Sum Sq Mean Sq F value  Pr(>F)
gender            1    1.7    1.66   0.308 0.58088
Treatment         2   70.5   35.24   6.535 0.00247 **
gender:Treatment  2   40.9   20.46   3.795 0.02712 *
Residuals        72  388.2    5.39
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

With Bonferroni correction:

```
> pairwise.t.test(stressData$diffStress, stressData$group, p.adjust="bonfer
roni")
```

```
        Pairwise comparisons using t tests with pooled SD

data:  stressData$diffStress and stressData$group

   F1      F2      F3      M1      M2
F2 1.00000 -       -       -       -
F3 0.02405 0.00076 -       -       -
M1 1.00000 1.00000 0.32087 -       -
M2 1.00000 0.72312 0.87995 1.00000 -
M3 1.00000 0.46373 1.00000 1.00000 1.00000

P value adjustment method: bonferroni
```

→ **Result**: The H0 that all means are equal is rejected.

But because the data is not normal, the Kruskal-Wallis Test can be used as a replacement for ANOVA. It comes also to the conclusion that the H0 that the means are equal is rejected.

Kruskal-Wallis Test:

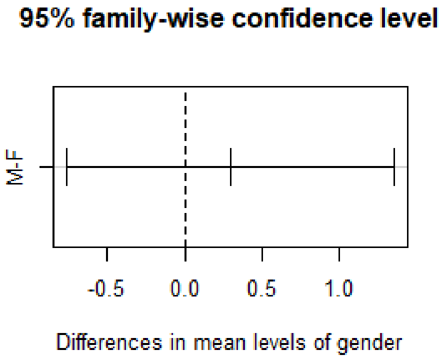
```
> kruskal.test(stressData$diffStress ~ stressData$group,
+              data=stressData) # means are not equal

        Kruskal-Wallis rank sum test

data:  stressData$diffStress by stressData$group
Kruskal-Wallis chi-squared = 18.579, df = 5, p-value = 0.002302
```

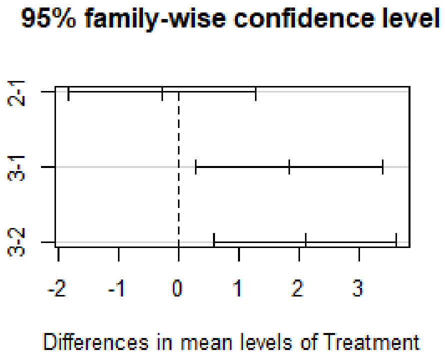
Tukey's Test:

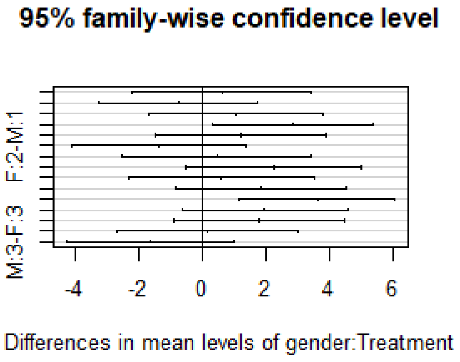
The Tukey test illustrates that the H0 that the means of treatment 3 and 1 and the means of 3 and 2 are equal is rejected. Likewise, there are some significant interactions, which suggest that the means are not equal. Overall, the Tukey test concludes that the means are not equal.

```
> TukeyHSD(res.aov)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = diffStress ~ gender * Treatment, data = stressData)

$`gender`
         diff        lwr      upr     p adj
M-F 0.2951515 -0.7657312 1.356034 0.5808818

$Treatment
          diff        lwr      upr     p adj
2-1 -0.2713412 -1.8303113 1.287629 0.9089274
3-1  1.8399495  0.2809794 3.398920 0.0166593
3-2  2.1112907  0.5988676 3.623714 0.0037527

$`gender:Treatment`
              diff        lwr       upr     p adj
M:1-F:1  0.6000000 -2.2149296 3.4149296 0.9888861
F:2-F:1 -0.7687500 -3.2568198 1.7193198 0.9441073
M:2-F:1  1.0590909 -1.6801838 3.7983656 0.8664203
F:3-F:1  2.8300000  0.3035233 5.3564767 0.0192018
M:3-F:1  1.1833333 -1.4912613 3.8579280 0.7865909
```

```
F:2-M:1 -1.3687500 -4.1093940 1.3718940 0.6889858
M:2-M:1  0.4590909 -2.5114754 3.4296572 0.9975219
F:3-M:1  2.2300000 -0.5455581 5.0055581 0.1872562
M:3-M:1  0.5833333 -2.3276965 3.4943632 0.9916231
M:2-F:2  1.8278409 -0.8350384 4.4907203 0.3471151
F:3-F:2  3.5987500  1.1553129 6.0421871 0.0006968
M:3-F:2  1.9520833 -0.6442135 4.5483802 0.2499542
F:3-M:2  1.7709091 -0.9278906 4.4697087 0.3979393
M:3-M:2  0.1242424 -2.7136955 2.9621804 0.9999950
M:3-F:3 -1.6466667 -4.2797922 0.9864589 0.4528384
```





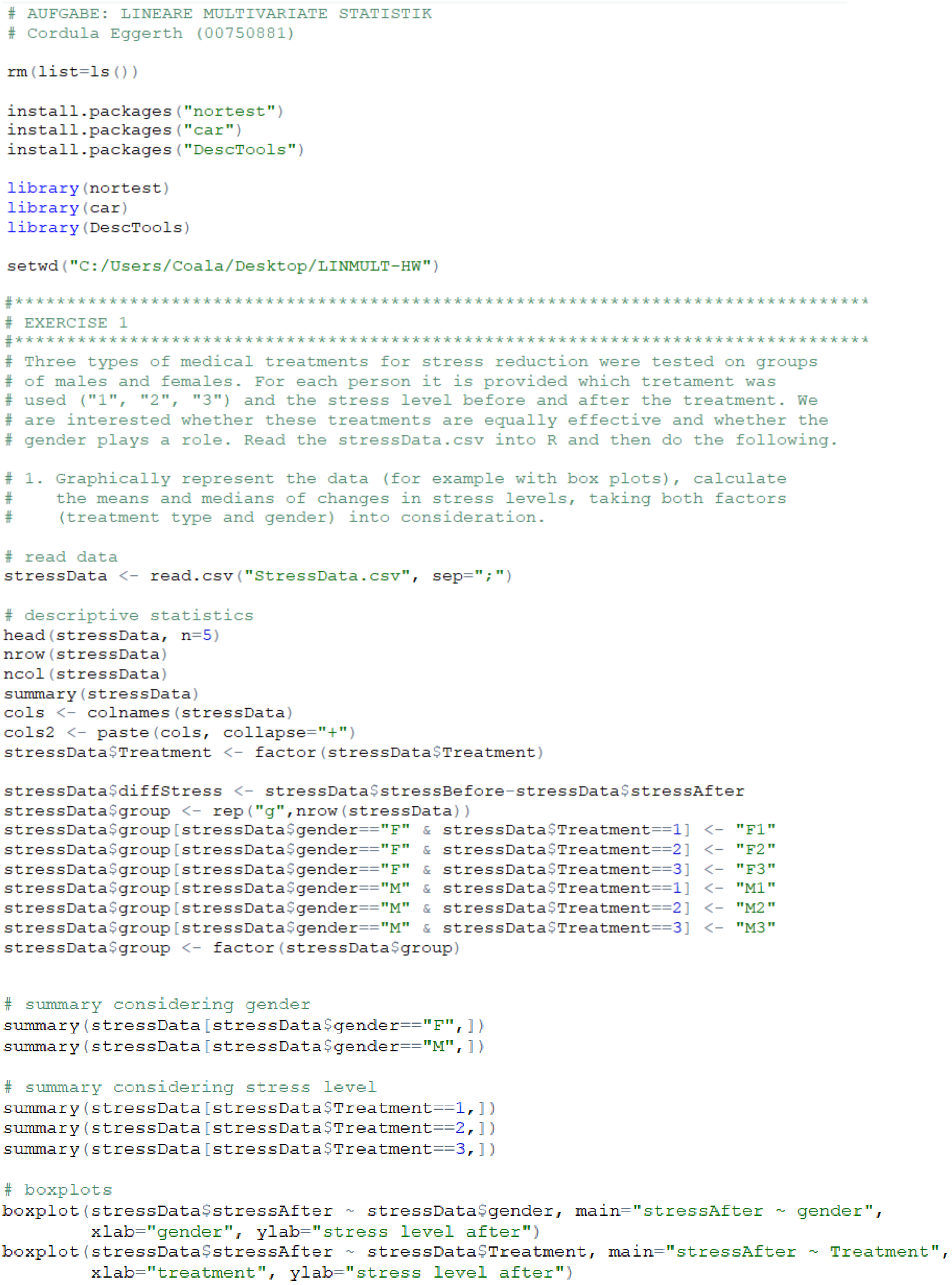


Posthoc Test with Scheffe Method / Scheffe Test:

```
> PostHocTest(aov(diffStress ~ group, data=stressData), method = "scheffe")

  Posthoc multiple comparisons of means : Scheffe Test
    95% family-wise confidence level

$`group`
            diff      lwr.ci   upr.ci   pval
F2-F1 -0.7687500 -3.6766140 2.139114 0.9750
F3-F1  2.8300000 -0.1227510 5.782751 0.0690 .
M1-F1  0.6000000 -2.6898726 3.889873 0.9954
M2-F1  1.0590909 -2.1423620 4.260544 0.9353
M3-F1  1.1833333 -1.9425265 4.309193 0.8898
F3-F2  3.5987500  0.7430492 6.454451 0.0047 **
M1-F2  1.3687500 -1.8343032 4.571803 0.8280
M2-F2  1.8278409 -1.2843270 4.940009 0.5479
M3-F2  1.9520833 -1.0822681 4.986435 0.4425
M1-F3 -2.2300000 -5.4738581 1.013858 0.3645
M2-F3 -1.7709091 -4.9250579 1.383240 0.5973
M3-F3 -1.6466667 -4.7240607 1.430727 0.6471
```

```
M2-M1  0.4590909 -3.0126777 3.930860 0.9990
M3-M1  0.5833333 -2.8188537 3.985520 0.9966
M3-M2  0.1242424 -3.1925205 3.441005 1.0000

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


> ScheffeTest(aov(diffStress ~ group, data=stressData))

  Posthoc multiple comparisons of means : Scheffe Test
    95% family-wise confidence level

$`group`
            diff     lwr.ci   upr.ci   pval
F2-F1 -0.7687500 -3.6766140 2.139114 0.9750
F3-F1  2.8300000 -0.1227510 5.782751 0.0690 .
M1-F1  0.6000000 -2.6898726 3.889873 0.9954
M2-F1  1.0590909 -2.1423620 4.260544 0.9353
M3-F1  1.1833333 -1.9425265 4.309193 0.8898
F3-F2  3.5987500  0.7430492 6.454451 0.0047 **
M1-F2  1.3687500 -1.8343032 4.571803 0.8280
M2-F2  1.8278409 -1.2843270 4.940009 0.5479
M3-F2  1.9520833 -1.0822681 4.986435 0.4425
M1-F3 -2.2300000 -5.4738581 1.013858 0.3645
M2-F3 -1.7709091 -4.9250579 1.383240 0.5973
M3-F3 -1.6466667 -4.7240607 1.430727 0.6471
M2-M1  0.4590909 -3.0126777 3.930860 0.9990
M3-M1  0.5833333 -2.8188537 3.985520 0.9966
M3-M2  0.1242424 -3.1925205 3.441005 1.0000

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R-Code for Exercise 1:



```
# AUFGABE: LINEARE MULTIVARIATE STATISTIK
# Cordula Eggerth (00750881)

rm(list=ls())

install.packages("nortest")
install.packages("car")
install.packages("DescTools")

library(nortest)
library(car)
library(DescTools)

setwd("C:/Users/Coala/Desktop/LINMULT-HW")

#**********************************************************************************
# EXERCISE 1
#**********************************************************************************
# Three types of medical treatments for stress reduction were tested on groups
# of males and females. For each person it is provided which tretament was
# used ("1", "2", "3") and the stress level before and after the treatment. We
# are interested whether these treatments are equally effective and whether the
# gender plays a role. Read the stressData.csv into R and then do the following.

# 1. Graphically represent the data (for example with box plots), calculate
#    the means and medians of changes in stress levels, taking both factors
#    (treatment type and gender) into consideration.

# read data
stressData <- read.csv("StressData.csv", sep=";")

# descriptive statistics
head(stressData, n=5)
nrow(stressData)
ncol(stressData)
summary(stressData)
cols <- colnames(stressData)
cols2 <- paste(cols, collapse="+")
stressData$Treatment <- factor(stressData$Treatment)

stressData$diffStress <- stressData$stressBefore-stressData$stressAfter
stressData$group <- rep("g",nrow(stressData))
stressData$group[stressData$gender=="F" & stressData$Treatment==1] <- "F1"
stressData$group[stressData$gender=="F" & stressData$Treatment==2] <- "F2"
stressData$group[stressData$gender=="F" & stressData$Treatment==3] <- "F3"
stressData$group[stressData$gender=="M" & stressData$Treatment==1] <- "M1"
stressData$group[stressData$gender=="M" & stressData$Treatment==2] <- "M2"
stressData$group[stressData$gender=="M" & stressData$Treatment==3] <- "M3"
stressData$group <- factor(stressData$group)


# summary considering gender
summary(stressData[stressData$gender=="F",])
summary(stressData[stressData$gender=="M",])

# summary considering stress level
summary(stressData[stressData$Treatment==1,])
summary(stressData[stressData$Treatment==2,])
summary(stressData[stressData$Treatment==3,])

# boxplots
boxplot(stressData$stressAfter ~ stressData$gender, main="stressAfter ~ gender",
        xlab="gender", ylab="stress level after")
boxplot(stressData$stressAfter ~ stressData$Treatment, main="stressAfter ~ Treatment",
        xlab="treatment", ylab="stress level after")
```

```r
# means and medians of changes in stress levels
# (taking both factors (treatment and gender) into consideration)
factor.treatment <- factor(stressData$Treatment)
is.factor(factor.treatment)

# gender diffs
 # methode 1
diffs.gender.mean <- tapply(stressData$stressBefore-stressData$stressAfter, stressData$gender, mean)
diffs.gender.median <- tapply(stressData$stressBefore-stressData$stressAfter, stressData$gender, median)

 # methode 2 (probe: ok)
diff.stress.m <- stressData[stressData$gender=="M", ]$stressBefore -
                 stressData[stressData$gender=="M", ]$stressAfter
mean(diff.stress.m)
median(diff.stress.m)
diff.stress.f <- stressData[stressData$gender=="F", ]$stressBefore -
                 stressData[stressData$gender=="F", ]$stressAfter
mean(diff.stress.f)
median(diff.stress.f)

# treatment diffs
diffs.treat.mean <- tapply(stressData$stressBefore-stressData$stressAfter, stressData$Treatment, mean)
diffs.treat.median <- tapply(stressData$stressBefore-stressData$stressAfter, stressData$Treatment, median)

par(mfrow=c(2,2))
barplot(diffs.gender.mean, main="mean diff. for gender", col=c("indianred1","lightslateblue"),
        xlab="gender", ylab="diff")
barplot(diffs.gender.median, main="median diff. for gender", col=c("indianred1","lightslateblue"),
        xlab="gender", ylab="diff")
barplot(diffs.treat.mean, main="mean diff. for treatment", col=c("darkseagreen1","forestgreen","khaki4"),
        xlab="treatment", ylab="diff")
barplot(diffs.treat.median, main="median diff. for treatment", col=c("darkseagreen1","forestgreen","khaki4"),
        xlab="treatment", ylab="diff")

# sample diffs (considering gender and treatment at the same time)
diffs.gt.mean <- tapply(stressData$stressBefore-stressData$stressAfter,
                        list(stressData$Treatment, stressData$gender), mean)
diffs.gt.median <- tapply(stressData$stressBefore-stressData$stressAfter,
                        list(stressData$Treatment, stressData$gender), median)


# 2. Using R-built-in functions analyze whether the standard analysis of vari-
#    ance (ANOVA) assumptions are met (normal data and equal variances).

# check if residuals normal:
res <- lm(diffStress ~ gender*Treatment, data=stressData)
summary(res)

par(mfrow=c(1,1))
hist(residuals(res), breaks=15, col="darkgray")

# check if residuals equal variance:
plot(fitted(res), residuals(res))

# check (for groups): normal data >> result: not all groups are normal
  # histogram
par(mfrow=c(2,3))
hist(stressData$diffStress[stressData$group=="F1"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="F1")
lines(density(stressData$diffStress[stressData$group=="F1"]),col=2)
hist(stressData$diffStress[stressData$group=="F2"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="F2")
lines(density(stressData$diffStress[stressData$group=="F2"]),col=2)
hist(stressData$diffStress[stressData$group=="F3"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="F3")
lines(density(stressData$diffStress[stressData$group=="F3"]),col=2)
hist(stressData$diffStress[stressData$group=="M1"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="M1")
lines(density(stressData$diffStress[stressData$group=="M1"]),col=2)
hist(stressData$diffStress[stressData$group=="M2"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="M2")
lines(density(stressData$diffStress[stressData$group=="M2"]),col=2)
hist(stressData$diffStress[stressData$group=="M3"], probability=TRUE,
     breaks=11, col="lavenderblush3", ylim=c(0,1), main="M3")
lines(density(stressData$diffStress[stressData$group=="M3"]),col=2)
```

```
  # qqplot and qqline
par(mfrow=c(2,3))
qqnorm(stressData$diffStress[stressData$group=="F1"],main="QQ plot: F1",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="F1"])
qqnorm(stressData$diffStress[stressData$group=="F2"],main="QQ plot: F2",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="F2"])
qqnorm(stressData$diffStress[stressData$group=="F3"],main="QQ plot: F3",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="F3"])
qqnorm(stressData$diffStress[stressData$group=="M1"],main="QQ plot: M1",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="M1"])
qqnorm(stressData$diffStress[stressData$group=="M2"],main="QQ plot: M2",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="M2"])
qqnorm(stressData$diffStress[stressData$group=="M3"],main="QQ plot: M3",pch=19,col="royalblue4")
qqline(stressData$diffStress[stressData$group=="M3"])

  # shapiro-wilk test for normality (at 0.05 level)
shapiro.test(stressData$diffStress[stressData$group=="F1"]) # not normal
shapiro.test(stressData$diffStress[stressData$group=="F2"]) # normal
shapiro.test(stressData$diffStress[stressData$group=="F3"]) # not normal
shapiro.test(stressData$diffStress[stressData$group=="M1"]) # normal
shapiro.test(stressData$diffStress[stressData$group=="M2"]) # normal
shapiro.test(stressData$diffStress[stressData$group=="M3"]) # normal

  # anderson-darling test for normality (at 0.05 level)
ad.test(stressData$diffStress[stressData$group=="F1"]) # not normal
ad.test(stressData$diffStress[stressData$group=="F2"]) # normal
ad.test(stressData$diffStress[stressData$group=="F3"]) # not normal
ad.test(stressData$diffStress[stressData$group=="M1"]) # normal
ad.test(stressData$diffStress[stressData$group=="M2"]) # normal
ad.test(stressData$diffStress[stressData$group=="M3"]) # normal

# check: equal variances
  # bartlett test of homogeneity of variances
  # (data should ideally be normal)
bartlett.test(diffStress ~ group, data=stressData) # equal variances

  # levene test (also for non-normal data)
leveneTest(diffStress ~ group, data=stressData) # equal variances

  # fligner-killeen test (also for non-normal data)
fligner.test(diffStress ~ group, data=stressData) # equal variances


# 3. Perform the analysis of variance (ANOVA).
# verified assumptions: non-normal data, but equal variances,
# therefore, a replacement for ANOVA is chosen

# if assumptions met, ANOVA result would be:
res.aov <- aov(diffStress ~ gender * Treatment, data=stressData)
summary(res.aov)
# bonferroni correction:
pairwise.t.test(stressData$diffStress, stressData$group, p.adjust="bonferroni")


# 4. Can we conclude that all groups perform similarly? If not, use the stan-
#    dard post-hoc tests such as Bonferroni, Tukey, etc. to investigate further.

# kruskal-wallis test
kruskal.test(stressData$diffStress ~ stressData$group,
             data=stressData) # means are not equal

# tukey's test
par(mfrow=c(2,2))
TukeyHSD(res.aov)
plot(TukeyHSD(res.aov))
abline(v=0)

# posthoc test with scheffe method/scheffe test
PostHocTest(aov(diffStress ~ group, data=stressData), method = "scheffe")
ScheffeTest(aov(diffStress ~ group, data=stressData))
```

Aufgabe 2:

1. ***In a few sentences compare the following classification methods: linear discriminant analysis, quadratic discriminant analysis and logistic regression. Comment on the basic idea behind the method, as well as the assumptions and performance. Consult the literature for this, if needed.***

LINEAR DISCRIMINANT ANALYSIS (LDA):

LDA was initiated by Fisher in the 1930s for situations comprising two classes. It was later extended to the general case by Rao to the multi-class case. The objective of LDA is to simplify a dataset in terms of dimensions. The method tries to find a straight line through the origin that leads to the best (i.e. maximum) possible split or separation of the data between groups. It is a supervised algorithm.
The assumptions for LDA are that the data is normal and the variances are equal.

Step 1:
Calculate the separation between classes (i.e. between class scatter), which is some form of variance:

$$S_b = \sum_{i=1}^{g} N_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

Step 2:
Calculate the within class scatter, which is again some form of a variance.

$$S_w = \sum_{i=1}^{g} (N_i - 1)S_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

Step 3:
Build the space that maximizes the S_B and minimizes S_W, i.e. maximizing the Fisher Discriminant (that we called J(v) in class, and is called P_{lda} here).

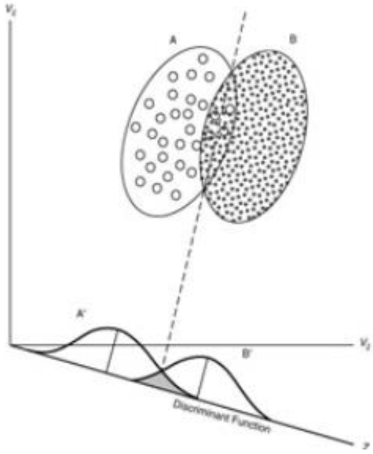
$$P_{lda} = \arg\max_P \frac{|P^T S_b P|}{|P^T S_w P|}$$

Prediction with LDA:
For prediction, the LDA estimates the probability that new unseen data belongs to each one of the classes. The new, unseen data points are then assigned to the class that has the highest probability estimation.
Using LDA appears to be beneficial (or say more beneficial than QDA) in situations where there are few training data points and variance reduction is important.
However, if the training set is very large (and therefore the variance is smaller), LDA is less beneficial as e.g. QDA.

Split between training and test data:

In practices, the dataset is split between training data that is used to fit the classifier, and test data that with which predictions are made and evaluated. The test data is new unseen data and helps to evaluate the accuracy and error rate of the classifier more realistically (on an out-of-sample dataset). Typical split are for training : test either 60% : 40%, 70% : 30%, or 80% : 20%.

Projection of the data on lower-dimensional space:[1]



QUADRATIC DISCRIMINANT ANALYSIS (QDA):

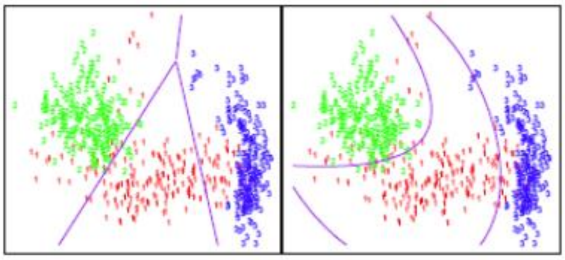
Assumptions:

- Observations of each class of the dependent variable (y) are normally distributed.
- Each class has its own VC-matrix (in contrast to LDA, which assumes the VC-matrix to be the same for all classes). The variances are therefore not assumed to be the same.

An observation is assigned to the class, for which the following formula is maximized (while looking at k different levels of y:

$$\hat{\delta}_k(x) = -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\hat{\mu}_k - \frac{1}{2}\hat{\mu}_k^T\mathbf{\Sigma}_k^{-1}\hat{\mu}_k - \frac{1}{2}log|\mathbf{\Sigma}_k| + log(\hat{\pi}_k)$$

If the VC-matrices are not the same, QDA can prove to be beneficial as it can better accommodate the separation of the classes in a non-linear form. QDA uses a quadratic (instead of a linear) function of the predictor variables. The quadratic boundaries can look for instance like the violet lines below: [2]

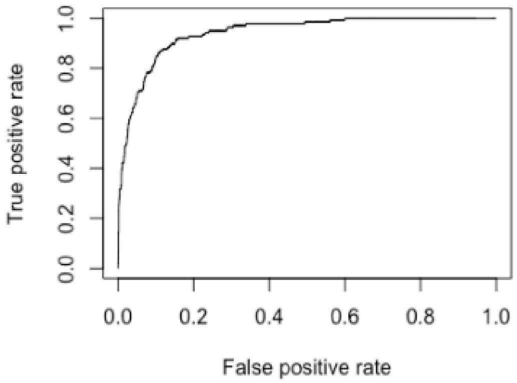


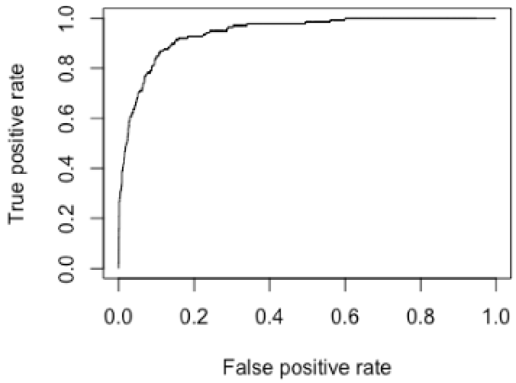
The prediction process works in the same way as for LDA.

[1] http://uc-r.github.io/discriminant_analysis

[2] http://uc-r.github.io/discriminant_analysis

Evaluation of predictions for LDA, QDA and logistic regression:
For all methods, the predictions can be evaluated by comparing them to the actual classes. In doing so, one can calculate the ***confusion matrix***, which shows the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). One can also calculate the ***misclassification rate (or error rate)***, which means the rate how many observations were assigned the wrong class label. Also, the ***ROC curve*** (i.e. Receiver Operating Characteristic curve) can be calculated by looking at different scenarios of the TP and FP rates. The nearer the ROC curve is to the left upper end of the square, the better it is. The aim is to maximize the ***AUC*** (i.e. Area Under the Curve). The ROC and AUC are for instance visible in the screenshots below: [3]



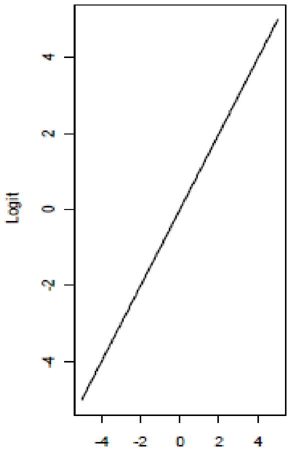


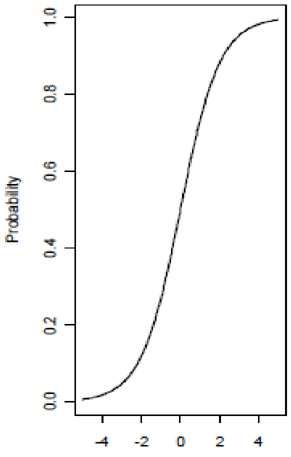
LOGISTIC REGRESSION:

In logistic regression, the dependent variable (y) is a dichotomous variable, i.e. it has 2 classes. The regression equation is modelled by the logit function[4], with the logarithmic expression being called the log-odds:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X + \dots + \beta_n X$$

The probability p can be expressed by rearranging the formula to express p. The logit function maps as follows to the unit interval, firstly in terms of logit and secondly in terms of probability:





[3] http://uc-r.github.io/discriminant_analysis

[4] But it can alternatively be modelled by the probit function or the cloglog function.

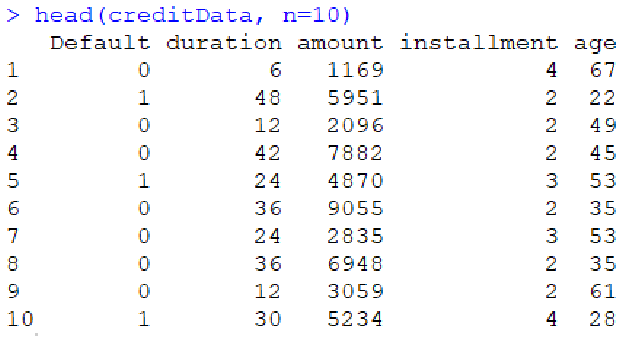
Calculating the β_i coefficients gives the multiplicative factor by which the log-odds change, if x changes by one unit. When it comes to model formulation, the logistic regression can be specified as a GLM (Generalized Linear Model) with the "binomial" family and the "logit" link function.

Unlike LDA and QDA, which can be used for more than 2-class problems, logistic regression is usually only used for 2-class problems (even though it can be extended to multi-class settings). The results of logistic regression can get unstable if the classes are separated well and if there are only few data points to train the classifier. However, in such situations, LDA is beneficial.

LDA and QDA assume that the observations are normally distributed. Logistic regression does not make this assumption. Therefore, it is beneficial if the data is not normally distributed. LDA and logistic regression result in linear boundaries to separate the classes, whereas QDA produces quadratic (i.e. non-linear) boundaries between the classes.

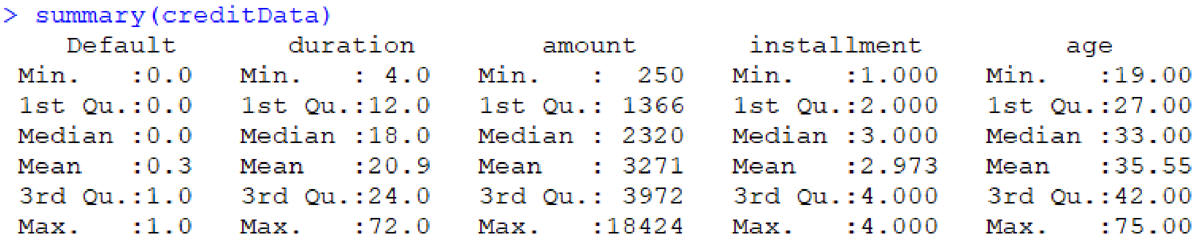
2. *Look at the creditdata.csv and focus on the following variables: default (whether a person defaults on a loan or not), duration (loan duration in months), amount (credit amount), instalment (as a percentage of disposable income) and age.*

Overview of the creditData, which comprises 1000 rows (observations) and 5 columns (variables):

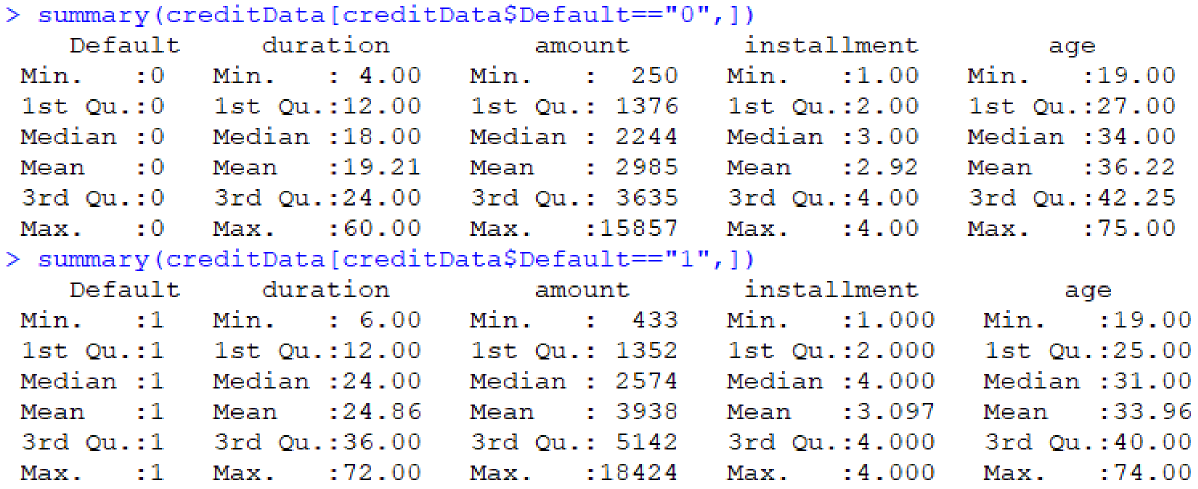


```
> head(creditData, n=10)
   Default duration amount installment age
1        0        6   1169           4  67
2        1       48   5951           2  22
3        0       12   2096           2  49
4        0       42   7882           2  45
5        1       24   4870           3  53
6        0       36   9055           2  35
7        0       24   2835           3  53
8        0       36   6948           2  35
9        0       12   3059           2  61
10       1       30   5234           4  28
```

3. *Summarize the data (averages, normality, etc.)*

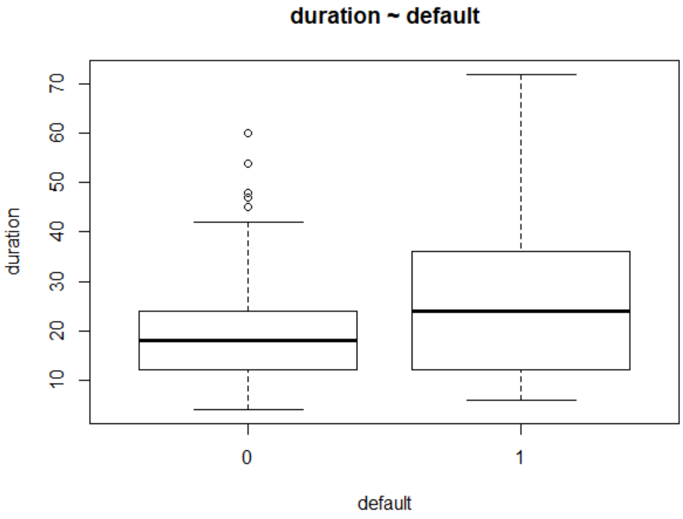


```
> summary(creditData)
    Default         duration          amount        installment          age       
 Min.   :0.0   Min.   : 4.0   Min.   :  250   Min.   :1.000   Min.   :19.00  
 1st Qu.:0.0   1st Qu.:12.0   1st Qu.: 1366   1st Qu.:2.000   1st Qu.:27.00  
 Median :0.0   Median :18.0   Median : 2320   Median :3.000   Median :33.00  
 Mean   :0.3   Mean   :20.9   Mean   : 3271   Mean   :2.973   Mean   :35.55  
 3rd Qu.:1.0   3rd Qu.:24.0   3rd Qu.: 3972   3rd Qu.:4.000   3rd Qu.:42.00  
 Max.   :1.0   Max.   :72.0   Max.   :18424   Max.   :4.000   Max.   :75.00  
```

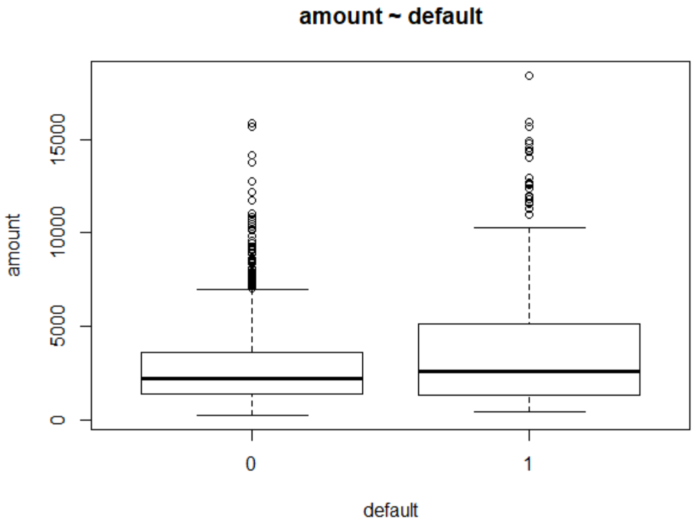


```
> summary(creditData[creditData$Default=="0",])
    Default        duration          amount        installment          age       
 Min.   :0    Min.   : 4.00   Min.   :  250   Min.   :1.00    Min.   :19.00  
 1st Qu.:0    1st Qu.:12.00   1st Qu.: 1376   1st Qu.:2.00    1st Qu.:27.00  
 Median :0    Median :18.00   Median : 2244   Median :3.00    Median :34.00  
 Mean   :0    Mean   :19.21   Mean   : 2985   Mean   :2.92    Mean   :36.22  
 3rd Qu.:0    3rd Qu.:24.00   3rd Qu.: 3635   3rd Qu.:4.00    3rd Qu.:42.25  
 Max.   :0    Max.   :60.00   Max.   :15857   Max.   :4.00    Max.   :75.00  
> summary(creditData[creditData$Default=="1",])
    Default        duration          amount        installment          age       
 Min.   :1    Min.   : 6.00   Min.   :  433   Min.   :1.000    Min.   :19.00  
 1st Qu.:1    1st Qu.:12.00   1st Qu.: 1352   1st Qu.:2.000    1st Qu.:25.00  
 Median :1    Median :24.00   Median : 2574   Median :4.000    Median :31.00  
 Mean   :1    Mean   :24.86   Mean   : 3938   Mean   :3.097    Mean   :33.96  
 3rd Qu.:1    3rd Qu.:36.00   3rd Qu.: 5142   3rd Qu.:4.000    3rd Qu.:40.00  
 Max.   :1    Max.   :72.00   Max.   :18424   Max.   :4.000    Max.   :74.00  
```

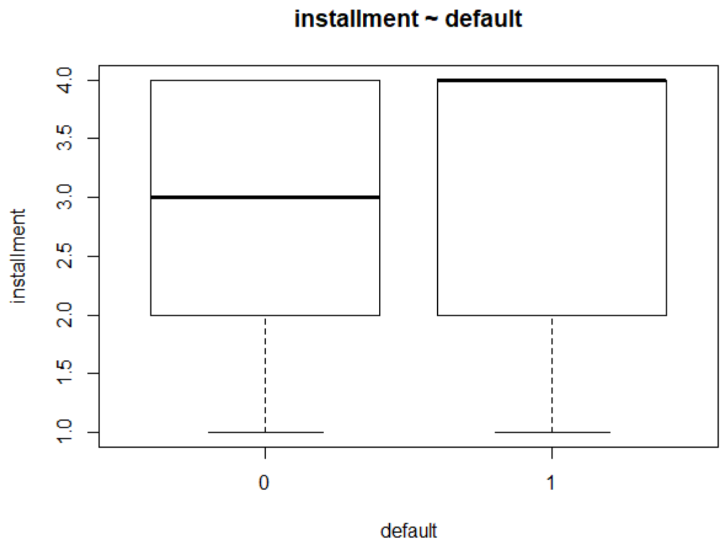
Boxplot of duration for no default vs. default:



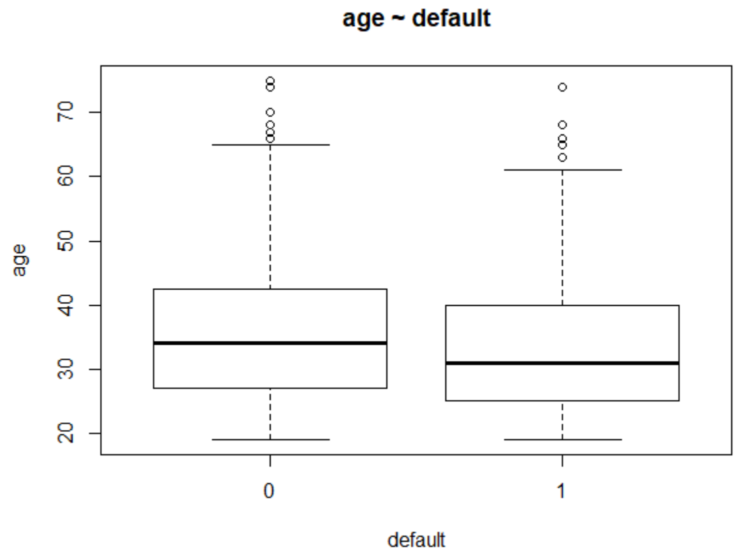
Boxplot of amount for no default vs. default:

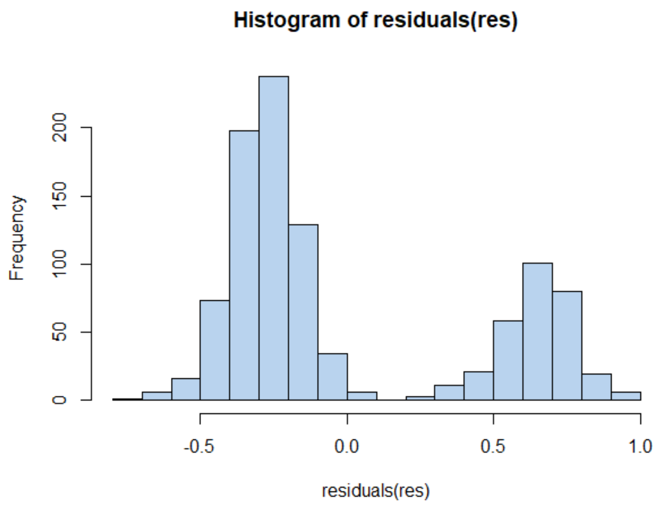


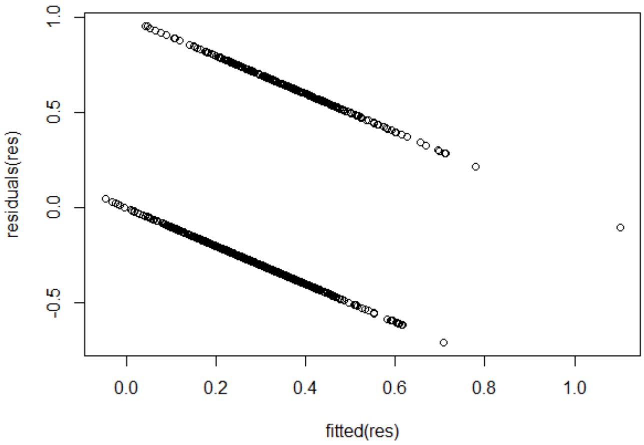
Boxplot of installment for no default vs. default:



Boxplot of age for no default vs. default:







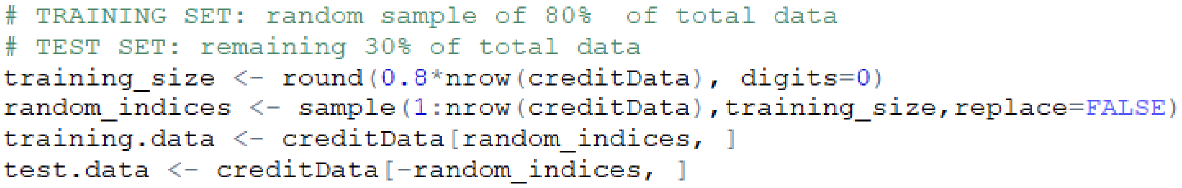
Tests on normality of the data:
According to the tests, the data is not normal.

```
        Shapiro-Wilk normality test

data:  creditData$Default
W = 0.57561, p-value < 2.2e-16

        Anderson-Darling normality test

data:  creditData$Default
A = 218.09, p-value < 2.2e-16
```

Tests on equality of variances of the data:
According to the tests, the data does not have equal variances.

```
        Fligner-Killeen test of homogeneity of variances

data:  amount by factor.Default
Fligner-Killeen:med chi-squared = 36.933, df = 1, p-value = 1.223e-09

Levene's Test for Homogeneity of Variance (center = median)
       Df F value    Pr(>F)
group   1   31.06 3.219e-08 ***
      998
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

4. *Split the data set randomly into a training set (80%) and a testing set (20%).*



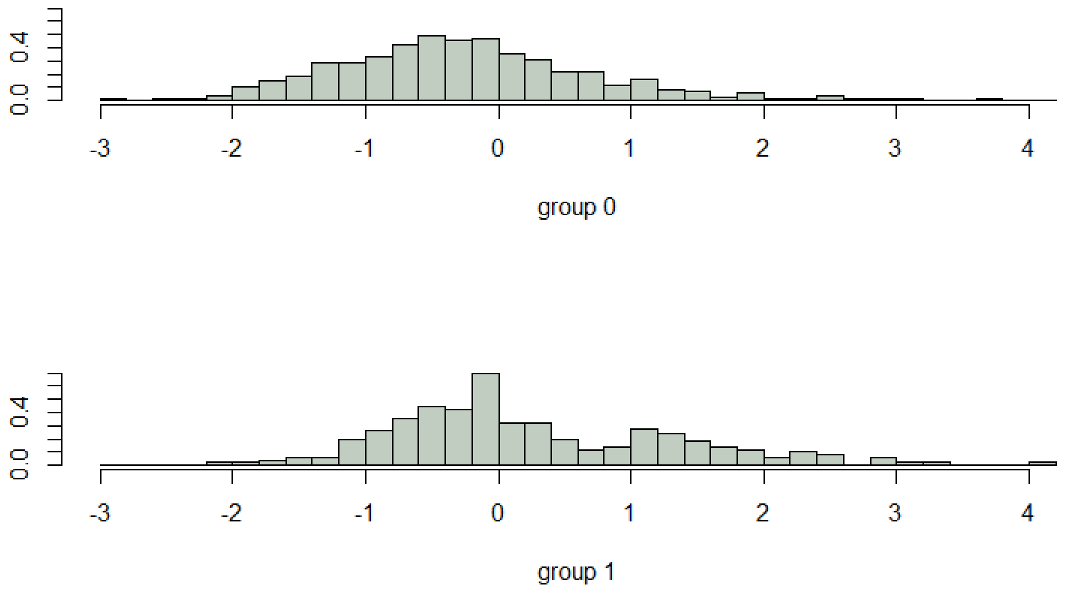
```
# TRAINING SET: random sample of 80%  of total data
# TEST SET: remaining 30% of total data
training_size <- round(0.8*nrow(creditData), digits=0)
random_indices <- sample(1:nrow(creditData),training_size,replace=FALSE)
training.data <- creditData[random_indices, ]
test.data <- creditData[-random_indices, ]
```

5. ***Using built-in-functions in R, train all three classification methods on the training data.***
6. ***To describe the classification performance, create confusion matrices on the test data for all three methods. Calculate the prediction accuracy and misclassification rate in terms of true positives and true negatives. Which model performs the best?***

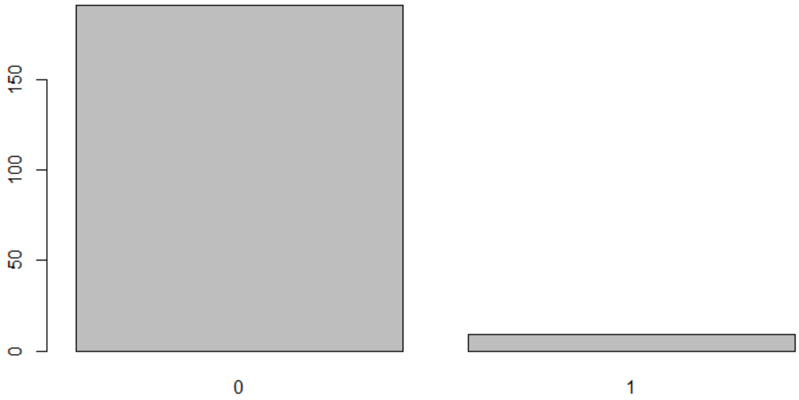
LINEAR DISCRIMINANT ANALYSIS (LDA):

```
> lda.fit
Call:
lda(Default ~ duration + amount + installment + age, data = training.data)

Prior probabilities of groups:
     0      1 
0.6875 0.3125 

Group means:
  duration   amount installment      age
0 19.36909 3008.584    2.921818 36.24727
1 24.70400 4075.352    3.088000 34.33600

Coefficients of linear discriminants:
                      LD1
duration     0.0399208125
amount       0.0001832178
installment  0.3791140264
age         -0.0286021419

> lda.predict <- predict(lda.fit, test.data)$class # on test.data
> lda.predict
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 [45] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
 [89] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
[133] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
[177] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Levels: 0 1
```

Plot lda.fit:



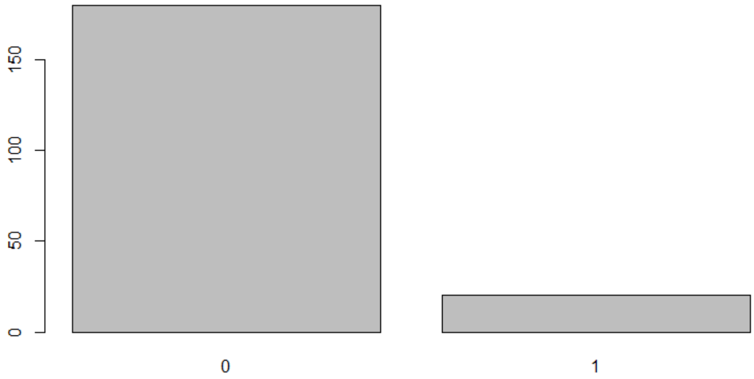
Plot lda.predict:



QUADRATIC DISCRIMINANT ANALYSIS (QDA):

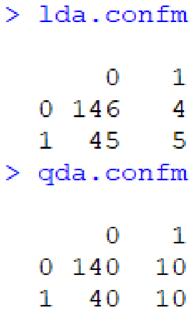
```
> qda.fit
Call:
qda(Default ~ duration + amount + installment + age, data = training.data)

Prior probabilities of groups:
     0      1 
0.6875 0.3125 

Group means:
  duration   amount installment      age
0 19.36909 3008.584    2.921818 36.24727
1 24.70400 4075.352    3.088000 34.33600
```

Plot lda.predict:



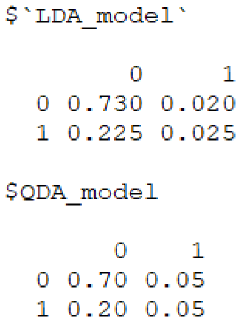
Confusion matrix for LDA and QDA:

Absolute:



```
> lda.confm

      0   1
  0 146   4
  1  45   5
> qda.confm

      0   1
  0 140  10
  1  40  10
```

Relative:



```
$`LDA_model`

        0     1
  0 0.730 0.020
  1 0.225 0.025

$QDA_model

       0    1
  0 0.70 0.05
  1 0.20 0.05
```

Accuracy for LDA:

```
> model_acc
[1] 0.7537688
```

(calculated as (TP+TN)/(TP+TN+FP+FN))

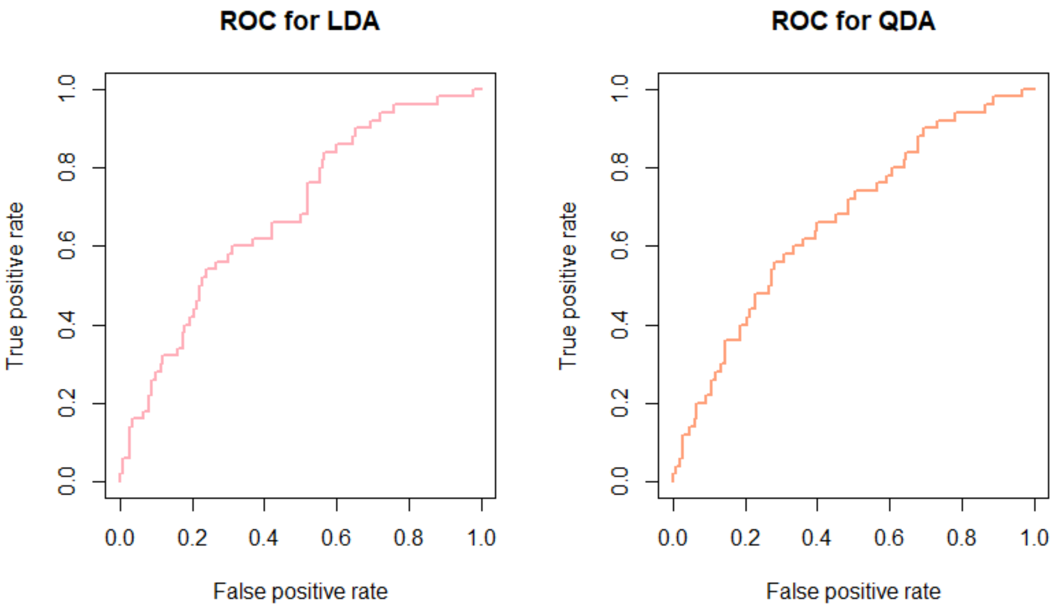
Accuracy for QDA:

```
> model_accQ
[1] 0.8717949
```

Error rates for LDA & QDA:

```
> test.data %>%
+   mutate(lda.predict = (test.predicted.lda$class),
+          qda.predict = (test.predicted.qda$class)) %>%
+   summarise(lda.error = mean(Default != lda.predict),
+             qda.error = mean(Default != qda.predict))
  lda.error qda.error
1     0.245      0.25
```

ROC curves for LDA & QDA:



AUC for LDA (area under curve)

```
> prediction(test.predicted.lda$posterior[,2], test.data$Default) %>%
+   performance(measure = "auc") %>%
+   .@y.values
[[1]]
[1] 0.6814667
```

AUC for QDA (area under curve)

```
> prediction(test.predicted.qda$posterior[,2], test.data$Default) %>%
+   performance(measure = "auc") %>%
+   .@y.values
[[1]]
[1] 0.6649333
```

LOGISTIC REGRESSION:

Fitted model:

```
Call:
glm(formula = factor(training.data$Default) ~ duration + amount +
    installment + age, family = binomial, data = training.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6524  -0.8661  -0.7236   1.2510   1.9962

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.593e+00  3.735e-01  -4.264    2e-05 ***
duration     1.974e-02  8.355e-03   2.362  0.01817 *
amount       8.945e-05  3.604e-05   2.482  0.01307 *
installment  2.077e-01  7.992e-02   2.599  0.00936 **
age         -1.602e-02  7.330e-03  -2.185  0.02886 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 993.74  on 799  degrees of freedom
Residual deviance: 948.87  on 795  degrees of freedom
AIC: 958.87

Number of Fisher Scoring iterations: 4
```

Predictions for test.data:

```
glm.prob <- predict(glm.fit, test.data, type="response")
```

Confusion matrix:

```
> table(test.data$Default, ifelse(glm.prob>0.5,"1","0"))

      0   1
  0 146   4
  1  46   4

> mean(ifelse(glm.prob>0.5,"1","0")==test.data$Default)
[1] 0.75
```

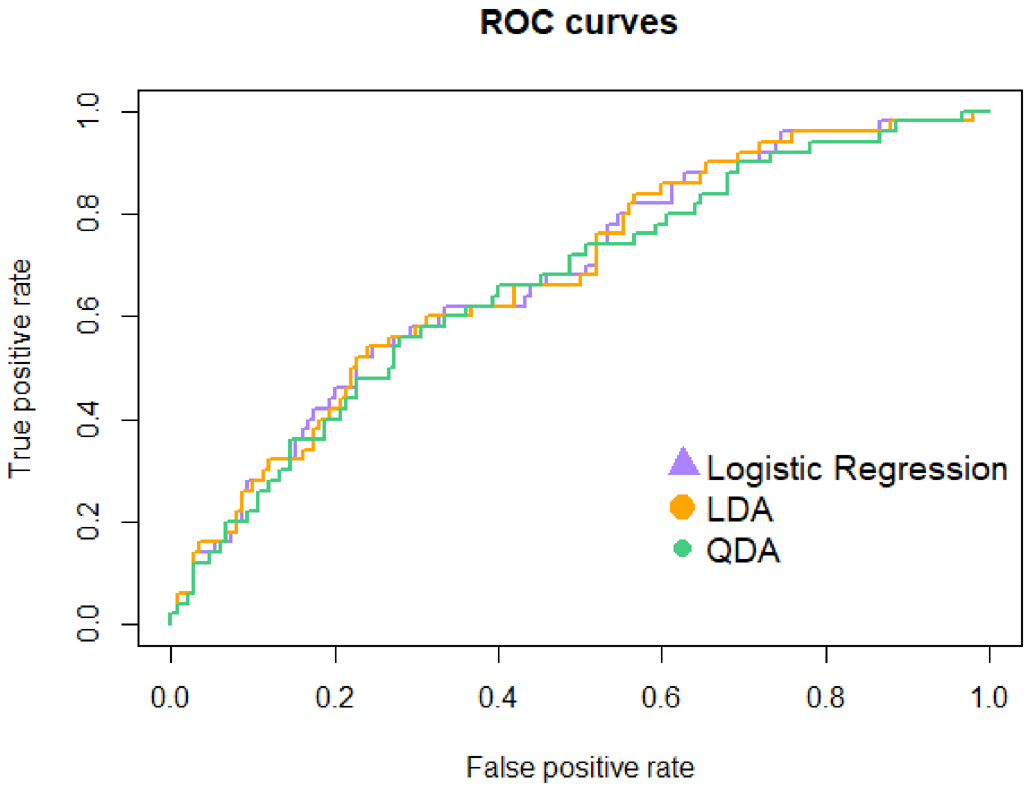
Error rate:

```
> mean(ifelse(glm.prob>0.5,"1","0")!=test.data$Default)
[1] 0.25
```

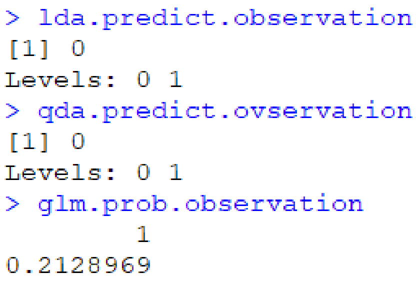
AUC for logistic regression:

```
> prediction(glm.prob, test.data$Default) %>%
+   performance(measure = "auc") %>%
+   .@y.values
[[1]]
[1] 0.6824
```

COMPARING ALL 3 METHODS:



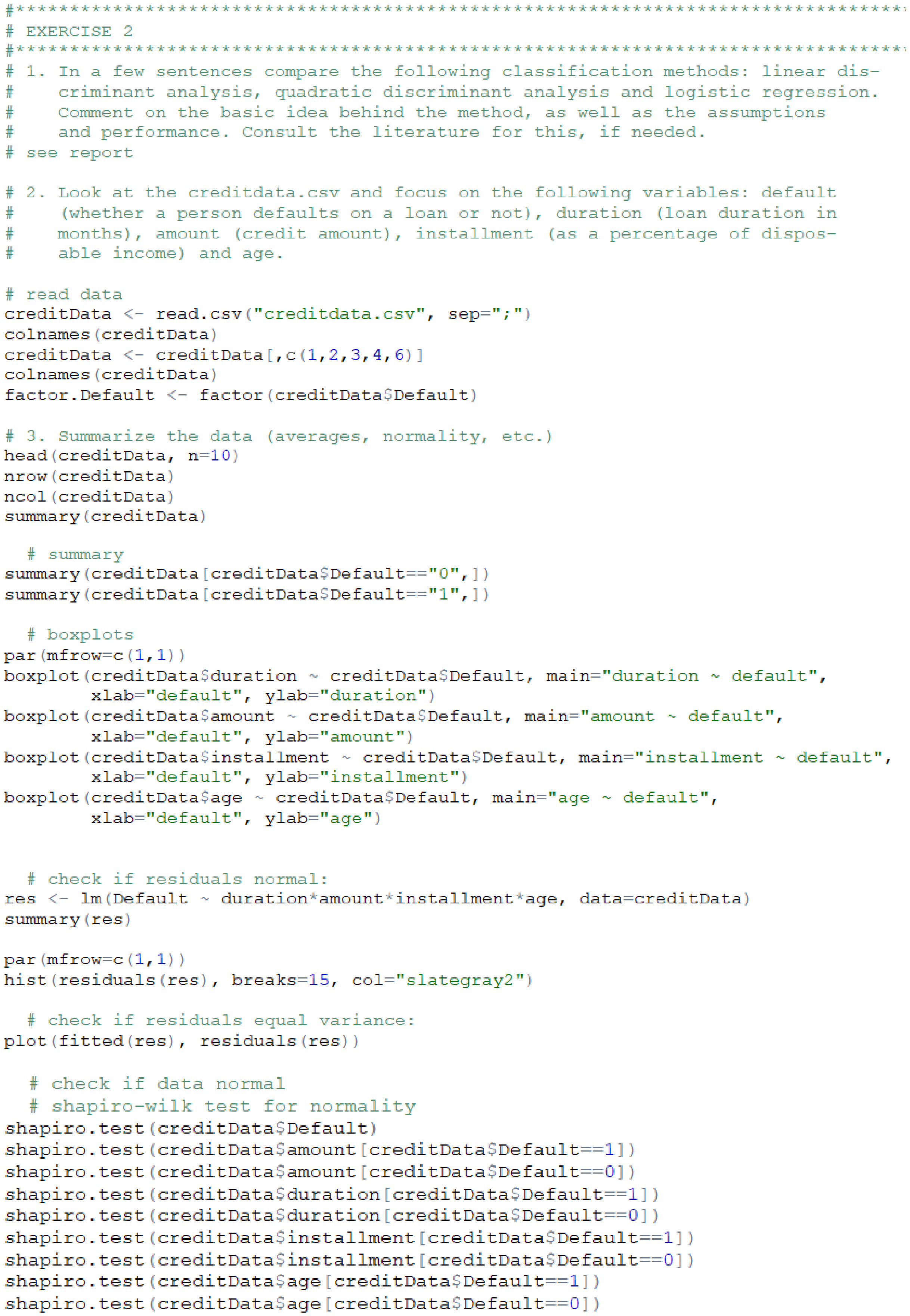
7. Predict whether a person with the following data defaults or not duration=12, amount=2000, instalment=4 and age=60.



```
> lda.predict.observation
[1] 0
Levels: 0 1
> qda.predict.ovservation
[1] 0
Levels: 0 1
> glm.prob.observation
        1
0.2128969
```

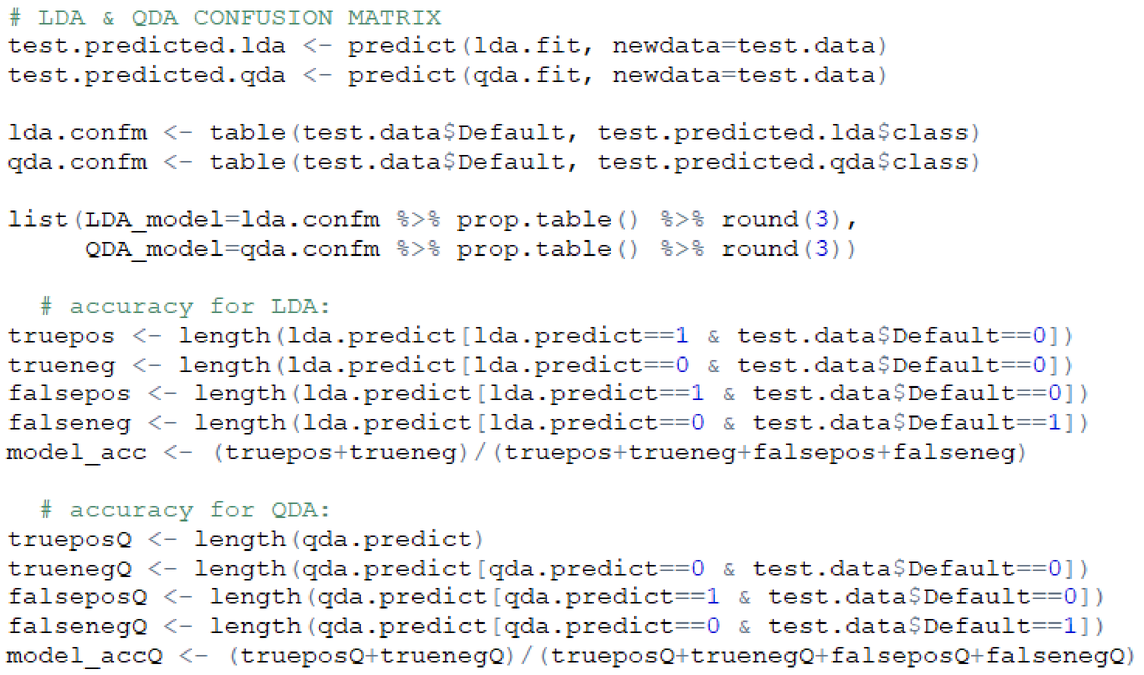
According to LDA, QDA as well as logistic regression, the considered observation will ont default (0).

R-Code for Exercise 2:

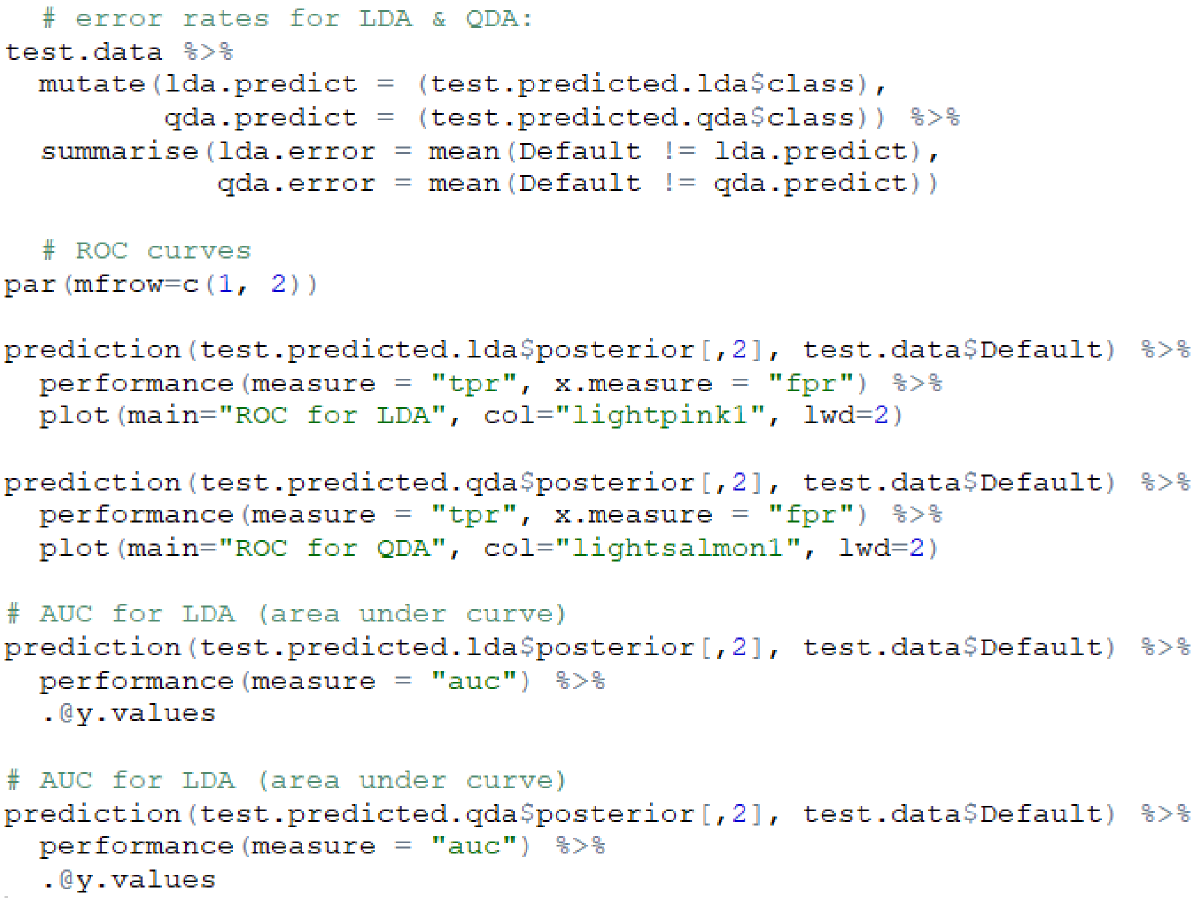


```
#*********************************************************************************
# EXERCISE 2
#*********************************************************************************
# 1. In a few sentences compare the following classification methods: linear dis-
#    criminant analysis, quadratic discriminant analysis and logistic regression.
#    Comment on the basic idea behind the method, as well as the assumptions
#    and performance. Consult the literature for this, if needed.
# see report

# 2. Look at the creditdata.csv and focus on the following variables: default
#     (whether a person defaults on a loan or not), duration (loan duration in
#    months), amount (credit amount), installment (as a percentage of dispos-
#    able income) and age.

# read data
creditData <- read.csv("creditdata.csv", sep=";")
colnames(creditData)
creditData <- creditData[,c(1,2,3,4,6)]
colnames(creditData)
factor.Default <- factor(creditData$Default)

# 3. Summarize the data (averages, normality, etc.)
head(creditData, n=10)
nrow(creditData)
ncol(creditData)
summary(creditData)

  # summary
summary(creditData[creditData$Default=="0",])
summary(creditData[creditData$Default=="1",])

  # boxplots
par(mfrow=c(1,1))
boxplot(creditData$duration ~ creditData$Default, main="duration ~ default",
        xlab="default", ylab="duration")
boxplot(creditData$amount ~ creditData$Default, main="amount ~ default",
        xlab="default", ylab="amount")
boxplot(creditData$installment ~ creditData$Default, main="installment ~ default",
        xlab="default", ylab="installment")
boxplot(creditData$age ~ creditData$Default, main="age ~ default",
        xlab="default", ylab="age")


  # check if residuals normal:
res <- lm(Default ~ duration*amount*installment*age, data=creditData)
summary(res)

par(mfrow=c(1,1))
hist(residuals(res), breaks=15, col="slategray2")

  # check if residuals equal variance:
plot(fitted(res), residuals(res))

  # check if data normal
  # shapiro-wilk test for normality
shapiro.test(creditData$Default)
shapiro.test(creditData$amount[creditData$Default==1])
shapiro.test(creditData$amount[creditData$Default==0])
shapiro.test(creditData$duration[creditData$Default==1])
shapiro.test(creditData$duration[creditData$Default==0])
shapiro.test(creditData$installment[creditData$Default==1])
shapiro.test(creditData$installment[creditData$Default==0])
shapiro.test(creditData$age[creditData$Default==1])
shapiro.test(creditData$age[creditData$Default==0])
```

```r
  # anderson-darling test for normality
ad.test(creditData$Default)
ad.test(creditData$amount[creditData$Default==1])
ad.test(creditData$amount[creditData$Default==0])
ad.test(creditData$duration[creditData$Default==1])
ad.test(creditData$duration[creditData$Default==0])
ad.test(creditData$installment[creditData$Default==1])
ad.test(creditData$installment[creditData$Default==0])
ad.test(creditData$age[creditData$Default==1])
ad.test(creditData$age[creditData$Default==0])

  # check if variances equal
  # bartlett test of homogeneity of variances
  # (data should ideally be normal)
bartlett.test(amount ~ factor.Default, data=creditData)
  # levene test (also for non-normal data)
leveneTest(amount ~ factor.Default, data=creditData)
  # fligner-killeen test (also for non-normal data)
fligner.test(amount ~ factor.Default, data=creditData)

# 4. Split the data set randomly into a training set (80%) and a testing set
#    (20%).

# TRAINING SET: random sample of 80%  of total data
# TEST SET: remaining 30% of total data
training_size <- round(0.8*nrow(creditData), digits=0)
random_indices <- sample(1:nrow(creditData),training_size,replace=FALSE)
training.data <- creditData[random_indices, ]
test.data <- creditData[-random_indices, ]

# 1 ... training sample
# 2 ... test sample
splitting <- rep(2,nrow(creditData))
splitting[random_indices] <- 1 # 80% data is training sample

# 5. Using built-in-functions in R, train all three classification methods on the
#    training data.
# 6. To describe the classification performance, create confusion matrices on
#    the test data for all three methods. Calculate the prediction accuracy and
#    misclassification rate in terms of true positives and true negatives. Which
#    model performs the best?

# LINEAR DISCRIMINANT ANALYSIS (LDA)
lda.fit <- lda(Default ~ duration+amount+installment+age,
                         training.data) # on training.data
lda.fit
lda.predict <- predict(lda.fit, test.data)$class # on test.data
lda.predict

plot(lda.fit, col=c("honeydew3"))
plot(lda.predict)


# QUADRATIC DISCRIMINANT ANALYSIS (QDA)
qda.fit <- qda(Default ~ duration+amount+installment+age,
                         data=training.data)
qda.fit
qda.predict <- predict(qda.fit, test.data)$class
plot(qda.predict)
```



```
# LDA & QDA CONFUSION MATRIX
test.predicted.lda <- predict(lda.fit, newdata=test.data)
test.predicted.qda <- predict(qda.fit, newdata=test.data)

lda.confm <- table(test.data$Default, test.predicted.lda$class)
qda.confm <- table(test.data$Default, test.predicted.qda$class)

list(LDA_model=lda.confm %>% prop.table() %>% round(3),
     QDA_model=qda.confm %>% prop.table() %>% round(3))

  # accuracy for LDA:
truepos <- length(lda.predict[lda.predict==1 & test.data$Default==0])
trueneg <- length(lda.predict[lda.predict==0 & test.data$Default==0])
falsepos <- length(lda.predict[lda.predict==1 & test.data$Default==0])
falseneg <- length(lda.predict[lda.predict==0 & test.data$Default==1])
model_acc <- (truepos+trueneg)/(truepos+trueneg+falsepos+falseneg)

  # accuracy for QDA:
trueposQ <- length(qda.predict)
truenegQ <- length(qda.predict[qda.predict==0 & test.data$Default==0])
falseposQ <- length(qda.predict[qda.predict==1 & test.data$Default==0])
falsenegQ <- length(qda.predict[qda.predict==0 & test.data$Default==1])
model_accQ <- (trueposQ+truenegQ)/(trueposQ+truenegQ+falseposQ+falsenegQ)
```



```
  # error rates for LDA & QDA:
test.data %>%
  mutate(lda.predict = (test.predicted.lda$class),
         qda.predict = (test.predicted.qda$class)) %>%
  summarise(lda.error = mean(Default != lda.predict),
            qda.error = mean(Default != qda.predict))

  # ROC curves
par(mfrow=c(1, 2))

prediction(test.predicted.lda$posterior[,2], test.data$Default) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot(main="ROC for LDA", col="lightpink1", lwd=2)

prediction(test.predicted.qda$posterior[,2], test.data$Default) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot(main="ROC for QDA", col="lightsalmon1", lwd=2)

# AUC for LDA (area under curve)
prediction(test.predicted.lda$posterior[,2], test.data$Default) %>%
  performance(measure = "auc") %>%
  .@y.values

# AUC for LDA (area under curve)
prediction(test.predicted.qda$posterior[,2], test.data$Default) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
# LOGISTIC REGRESSION:
# fit:
glm.fit <- glm(factor.Default ~ duration+amount+installment+age,
               data=training.data,
               family=binomial)
summary(glm.fit)

# predict:
glm.prob <- predict(glm.fit, test.data, type="response")

# confusion matrix:
table(test.data$Default, ifelse(glm.prob>0.5,"1","0"))

# accuracy:
mean(ifelse(glm.prob>0.5,"1","0")==test.data$Default)

# error rate
mean(ifelse(glm.prob>0.5,"1","0")!=test.data$Default)

# AUC for logistic regression (area under curve)
prediction(glm.prob, test.data$Default) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
# COMPARE ROC CURVES OF ALL 3 METHODS:
# LOGISTIC REGRESSION
pred.log <- prediction(glm.prob, test.data$Default) %>%
  performance(measure = "tpr", x.measure = "fpr")

# LDA
pred.lda <- prediction(test.predicted.lda$posterior[,2], test.data$Default) %>%
  performance(measure = "tpr", x.measure = "fpr")

# QDA
pred.qda <- prediction(test.predicted.qda$posterior[,2], test.data$Default) %>%
  performance(measure = "tpr", x.measure = "fpr")

# plots for pred.log, pred.lda, pred.qda
par(mfrow=c(1,1))
plot(pred.log, col = "mediumpurple1", lwd=2, main="ROC curves")
plot(pred.lda, add = TRUE, col = "orange", lwd=2)
plot(pred.qda, add = TRUE, col = "seagreen3", lwd=2)
legend("bottomright",
       legend = c("Logistic Regression", "LDA", "QDA"),
       col = c("mediumpurple1", "orange", "seagreen3"),
       pch = c(17,19,20),
       bty = "n",
       pt.cex = 2,
       cex = 1.2,
       text.col = "black",
       inset = c(0.1, 0.1))

# 7. Predict whether a person with the following data defaults or not dura-
#    tion=12, amount=2000, installment=4 and age=60.
df.observation <- data.frame(Default=0,duration=12,amount=2000,
                             installment=4,age=60)

# LDA:
lda.predict.observation <- predict(lda.fit, df.observation)$class

# QDA:
qda.predict.ovservation <- predict(qda.fit, df.observation)$class

# LOGISTIC REGRESSION:
glm.prob.observation <- predict(glm.fit, df.observation, type="response")
```

Bibliography:

- https://onlinecourses.science.psu.edu/stat501/print/book/export/html/300/
- https://www.sheffield.ac.uk/polopoly_fs/1.579191!/file/stcp-karadimitriou-normalR.pdf
- https://cran.r-project.org/web/packages/nortest/nortest.pdf
- http://www.instantr.com/2012/12/12/performing-bartletts-test-in-r/
- http://www.sthda.com/english/wiki/compare-multiple-sample-variances-in-r#compute-bartletts-test-in-r
- https://www.sheffield.ac.uk/polopoly_fs/1.714578!/file/stcp-marquier-FriedmanR.pdf
- https://www.digitalvidya.com/blog/linear-discriminant-analysis/
- http://uc-r.github.io/discriminant_analysis
- Marcus Hudec (2019). Folien “UK Erweiterungen des linearen Modells“ (Sommersemester 2019, Prof. Marcus Hudec), Universität Wien.
- James Gareth & Robert Tibshirani (2013): An Introduction to Statistical Learning with Applications in Springer Texts in Statistics 2013.