Project 1, Phase 2

Agnieszka Sitko, Annanina Koster, Rafal Cylwa, Piotr Obarski

Modele liniowe i mieszane

Abstract

We are going to process data to try to get better results in tests for fulfiling assumptions of ANOVA, because in phase 1, we had a big problem with that. Then we will check assumptions of ancova and interactions and we will perform ancova and post hoc tests on the genes which fulfils these assumptions. Then we are going to perform two way anova on the rest of the genes and finally choose the small list of genes which will help us to recognize type of cancer given person can have. And this small list is our main goal.

Data processing

Load data

We load libraries and load data from the 1st. phase of project.

```r
library(agricolae)
library(MASS)
library(nortest)
library(Rcmdr)
library(lmtest)

load("~/Modele liniowe i mieszane/projekt 1/data.RData")
```

n is a paramter for how many genes we want perform whole code. We remove one group with only 12 observations, and we make small corrections to data.

```r
n<-length(colnames(data))
#tapply(data$id, data$CancerType, length)

data<-data[-which(data$CancerType=="TCGA Formalin Fixed Paraffin-Embedded Pilot Phase II"),]
CancerType <- data$CancerType
CancerType<-factor(data$CancerType, levels(CancerType)[-6])
colnames(data)[20]<-"age"
```

Checking assumptions

Introduction

Since in the 1st. phase we had a lot of trouble with assumptions now we have a several approaches how we are going to change data to make it better.

Checking assumptions - main part

We use Shapiro-Wilk test to check the normality of the residuals of the model. Afterwards we adjust received p-values with the Holm-Bonferroni method in order to control Familywise error rate. Very similar things we do with checking homogenity of variance and to do that we used Leven's test. We also checked normality of residuals with lillie test. Beneath is the function which checks assumptions.

```
checking_assumption<-function(data) {
  normality<-NA
  normality1<-NA
  variance_in_groups<-NA
  # length(colnames(data))
  for (i in 30:n) {
    x <- data[,i]
    model = aov(x ~ data$CancerType*data$age*data$gender)
    normality[i]<-lillie.test(rstandard(model))$statistic
    normality1[i]<-shapiro.test(rstandard(model))[[2]]
     variance_in_groups[i] <- leveneTest(x ~ data$gender * data$CancerType)[[3]][1]
  }
  return(cbind(normality, normality1, variance_in_groups))
}
```

There is also one question, what cutoff should we choose for lillie test for D statistic. From a quick look at the data (specifically gene expressions), we see that relevant numbers are those before point and two after it. So let's make quick simulation. We have approximately 3000 patients, so let's do simulation in this way:

```
mean(replicate(1000, lillie.test(round(rnorm(3000),2))$statistic))
```

```
## [1] 0.01283388
```

Hence we chose as a cutoff for a Lillie test 0.015.

```


pvals<-checking_assumption(data)
pvals[,2]<-p.adjust(pvals[,2], method="holm")
pvals[,3]<-p.adjust(pvals[,3], method="holm")
length(which(pvals[,1]<0.015))/(n - 29)
```

Non transformed data

```
## [1] 0.03332211
```

```
length(which(pvals[,2]>0.05))/(n - 29)
```

```
## [1] 0.05048805
```

```
length(which(pvals[,3]>0.05))/(n - 29)
```

```
## [1] 0.0414002
```

```
#qqnorm((rstandard(lm(data[,32]~data$CancerType))))
```

We see that results are not satisfactory. Every assumption is fulfilled in about 3% cases, and plot of normality of residuals for one genes is far from being perfect.

```


data1<-data
a<- 0
for(i in 30:n) {
  x <- data1[,i]
  x<-x+1-min(x[which(!is.na(x))]) # shift data to get only positive
  a<-boxcox(x~data$CancerType*data$gender, plotit = FALSE)
  if (a$x[which.max(a$y)]!= 0) {
    x<-(x^(a$x[which.max(a$y)])-1)/a$x[which.max(a$y)]
  } else {
    x<-log(x)
  }
  data1[,i]<-x
}

pvals1<-checking_assumption(data1)
pvals1[,2]<-p.adjust(pvals1[,2], method="holm")
pvals1[,3]<-p.adjust(pvals1[,3], method="holm")
length(which(pvals1[,1]<0.015))/(n-29)
```

Data transformation - boxcox

```
## [1] 0.06462471
```

```
length(which(pvals1[,2]>0.05))/(n-29)
```

```
## [1] 0.1272299
```

```
length(which(pvals1[,3]>0.05))/(n-29)
```

```
## [1] 0.04678559
```

```
pvals_box_cox<-pvals1
```

Beneath, due to pages limitation some plots are commented. After each transformation of data plots should be made.

```
#par(mfrow=c(1,2))
#hist(data[which(data$CancerType=="TCGA Acute Myeloid Leukemia"),38])
#hist(data1[which(data1$CancerType=="TCGA Acute Myeloid Leukemia"),38])
#boxplot(data[,38]~data$CancerType)
#boxplot(data1[,38]~data$CancerType)
#par(mfrow=c(1,1))
#qqnorm((rstandard(lm(data1[,32]~data$CancerType))))
```

We see that boxcox improved our data slightly. But on plots it is not visible (and it is not because plots are not visible!), and assumptions are still very far from fulfilling assumptions. We can also see that a few outliers were added, which in some sense is good because there is a hope that we can improve more by doing sth with outliers.

Log transformation This transformation made our test worse so it is not worthwhile to put it in the report.

Analyzing outliers Now we are going to deal with outliers. First approach will be to move outliers to the end of the boxplot's whiskers.

```r
data2<-data1

windsor <- function (x) {
      lim <- boxplot.stats(x)$stats[c(1,5)]

      x[x < lim[1] & !is.na(x)] <- lim[1]
      x[x > lim[2] & !is.na(x)] <- lim[2]
      x
}
data1[,30:n] <- apply(data1[,30:n], 2, windsor)

pvals1<-checking_assumption(data1)
pvals1[,2]<-p.adjust(pvals1[,2], method="holm")
pvals1[,3]<-p.adjust(pvals1[,3], method="holm")
length(which(pvals1[,1]<0.015))/(n - 29)
```

```
## [1] 0.08212723
```

```r
length(which(pvals1[,2]>0.05))/(n - 29)
```

```
## [1] 0.2396499
```

```r
length(which(pvals1[,3]>0.05))/(n - 29)
```

```
## [1] 0.03433187
```

Even though we have some improvement in our test of assumptions, looking at the boxplots and histograms it doesn't seem to be the right thing to do (plots were commented due to limitation in number of pages). So maybe it is the better approach to move outliers among the groups, but it has less reasnoable causes.

```r
data1<-data2
for (i in 30:n) {
  for (j in 1:12) {
    x<-data1[which(data$CancerType==levels(data1$CancerType)[j]),i]
    krok<-(quantile(x, na.rm=TRUE)[4]-quantile(x, na.rm=TRUE)[2])*1.5

    x[which(x>quantile(x, na.rm=TRUE)[3]+krok)]<-quantile(x, na.rm=TRUE)[3]+krok
    x[which(x<quantile(x, na.rm=TRUE)[2]-krok)]<-quantile(x, na.rm=TRUE)[2]-krok
    data1[which(data$CancerType==levels(data$CancerType)[j]),i]<-x
  }
}

pvals1<-checking_assumption(data1)
pvals1[,2]<-p.adjust(pvals1[,2], method="holm")
pvals1[,3]<-p.adjust(pvals1[,3], method="holm")
length(which(pvals1[,1]<0.015))/(n - 29)
```

```
## [1] 0.2170986
```

```
length(which(pvals1[,2]>0.05))/(n - 29)
```

```
## [1] 0.1713228
```

```
length(which(pvals1[,3]>0.05))/(n - 29)
```

```
## [1] 0.01918546
```

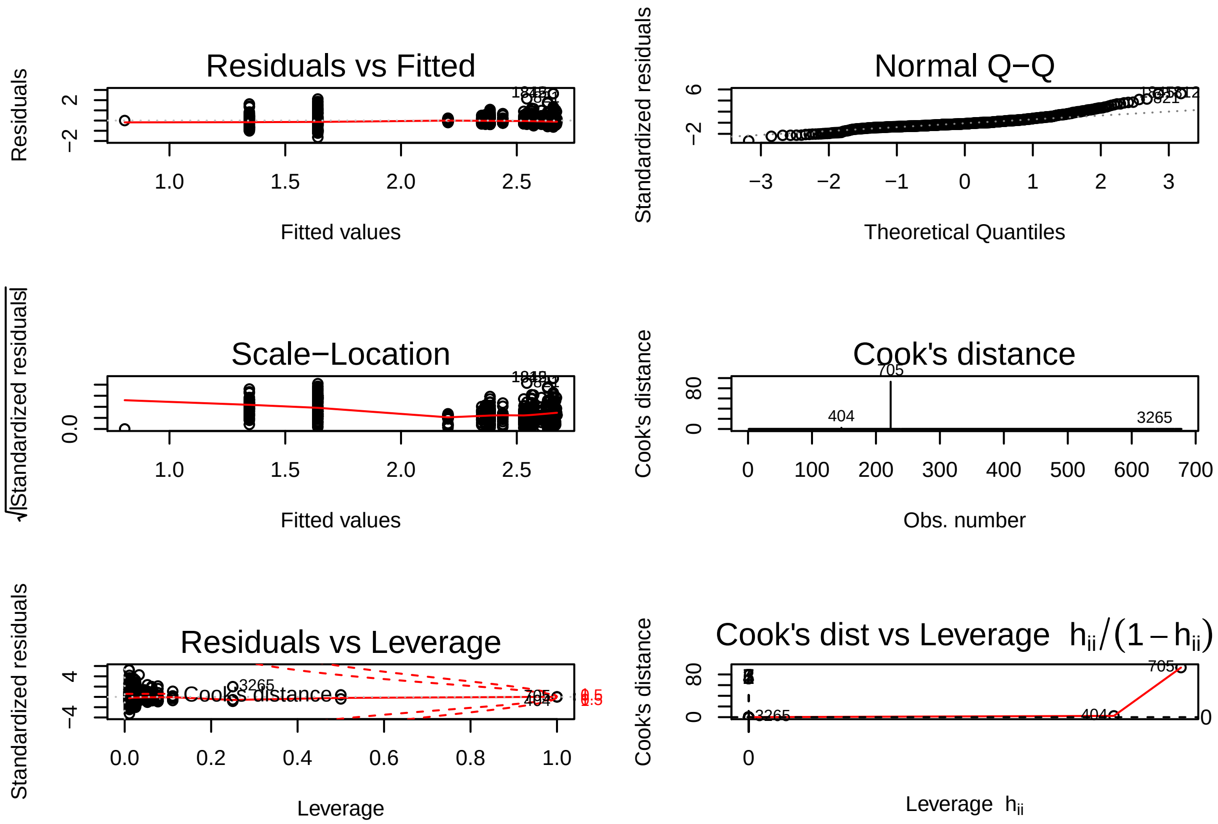
```
data3<-data2
```

As previously we have a little worse outcome in the test for variance, but better in the tests for normility. Plots look also nicer.

Now we take another approach. We check whether outlier has big impact on the model, we will do it via cook's distances, and if not we will remove it. Unfortunately, we removed not only outliers but also the code, because of the limitation of the report, but the inferences are here:

We see that this approach boost very significantly our tests for normality as well as for test for variance (but still we don't have satisfactory outcome). Plots look worse comapring to the previous ones. So now we need to choose with which approach we are going to continue our project. Of course project is desinged in this way, that it can be easily changed. So we decided to proceed with unmoved outliers. We think that there are quite many of them, so probably they are not mistakes and it is difficult to choose the best approach and not only the best but also the one who would be reasnable, and would give good results. Now we will also check diagnosis plot for one with maximal p.value in lillie test.

```
data<-data3
x<-data[,which(pvals_box_cox[,1]==max(pvals_box_cox[,1], na.rm = TRUE))]
par(mfrow=c(3,2))
plot(lm(x~data$CancerType*data$gender), which=1:6)
```



We see that 1, 3, 4 are quite ok, and the rest are pretty bad. # Experiment Now we will carry out a small experiment. Since our data doesn't fulfill assumptions, we do sth like this: take gene, check mean in the first group, and then we even means in other groups. Then we add to each observation number from $\mathcal{N}(0, 1)$ and check in how many cases anova gives us false result. Since we have many gens, and we are going to choose small number of them, it is safer to have gens which will predict type of cancer then have gene which has no power of prediction. So we are hoping that in this test we will not get too much false outcomes, and that means that subsequently we will be prone to reject gens that may have be good predict with them, but we will not accept genes which are useless for prediction.

```
data2<-data
for(i in 30:n) {
  x<-data2[,i]
  gene <- split(x, CancerType)
  m<-sapply(gene, mean, na.rm = TRUE)[1]
  for (j in 2:12) {
  x[which(CancerType==levels(CancerType)[j])]<-x[which(CancerType==levels(CancerType)[j])]+m-sapply(gene
  }
  data2[,i]<-x+rnorm(length(x))
}
test<-0
for(i in 30:n){
            x <- data2[,i]
            if(!is.na(mean(x))) {
              model <- lm(x~CancerType*data2$gender)
              test[i] <- summary(aov(model))[[1]][[5]][1]
            }
```

```r
}
test<-p.adjust(test, method="holm")
table(test<0.05)
```

```
##
## FALSE  TRUE
##  1415     1
```

So we have result which pleases us.

Assumptions of ancova and ancova

Now let's check two vital assumptions of ANCOVA i.e. homogeneity of regression slopes and linearity of dependent variable with contiunous one.

```r
wektor<-matrix(rep(0,8*n), nrow=n, ncol=8)
for (i in 30:n) {
  x <- data[,i]
  model = lm(x ~ data$CancerType*data$age*data$gender)
  wektor[i,]<-anova(model)$"Pr(>F)"
}

length(which(wektor[,4]>0.05))/(n-29)
```

```
## [1] 0.4436217
```

```r
length(which(wektor[,6]>0.05))/(n-29)
```

```
## [1] 0.9495119
```

```r
length(which(wektor[,7]>0.05))/(n-29)
```

```
## [1] 0.9491754
```

```r
interakcja<-p.adjust(wektor[,4], method="holm")
length(which(interakcja>0.05))/(i-29)
```

```
## [1] 0.9229216
```

```r
# checking linearity
linearity<-0
for (i in 30:n) {
  x <- data[,i]
  linearity[i]<-resettest(lm(x~data$age), power=2, type="regressor")$p.value
}
linearity<-p.adjust(linearity, method="holm")
length(which(linearity>0.05))
```

```
## [1] 2693
```

```
ancov<-intersect(which(linearity>0.05), which(interakcja>0.05))
```

Now we saved in vector ancov genes which fulfills assumptions of ancova. For them we are going to perform ancova and then we will perform scheffe test and choose genes with interactions and high number of significantly different groups (here we took more than 6, we chose this number to get small list at the end).

```
interactions <- matrix(rep(NA, times = 2*n), n, 2)
colnames(interactions) <- c("cohort:age", "cohort:gender:age")
for(i in ancov){
            x <- data[,i]
            model <- lm(x ~ data$CancerType * data$gender * data$age)
            interactions[i,1:2] <- summary(aov(model))[[1]][[5]][c(5,7)]
}

table(interactions[,1]<0.05)
```

```
##
## FALSE  TRUE
##  1226  1270
```

```
#colnames(data[which(interactions[,1]<0.05)])
table(interactions[,2]<0.05)
```

```
##
## FALSE  TRUE
##  2372   124
```

```
#colnames(data[which(interactions[,2]<0.05)])
doubleinter<-intersect(which(interactions[,1]<0.05),which(interactions[,2]<0.05))
length(doubleinter)
```

```
## [1] 57
```

```
#colnames(data[,doubleinter])



nrofgroups<-0
for (i in ancov) {
   x <- data[,i]
   model <- lm(x ~ data$CancerType * data$gender * data$age)
   model1 <- anova(lm(x ~ data$CancerType * data$gender * data$age))
      if(model1$"Pr(>F)"[1]<=0.05){
            nrofgroups[i]<-match(tail(scheffe.test(aov(model), "data$CancerType", console = FALSE)$group
      }
}
length(intersect(which(nrofgroups>6), doubleinter))
```

```
## [1] 18
```

```
genes1<-colnames(data[,intersect(which(nrofgroups>6), doubleinter)])
#genes1
```

Anova

If there is no linear relation between X and Y, then the analysis of covariance offers no improvement over the one-way analysis of variance in detecting differences between the group means (in our case it will be two-way anova). If both X and Y depend on the group (treatment), then the analysis of covariance can be misleading. So we are going to perform two-way anova for the genes for which we haven't performed ancova.

```
genes<-30:n
genes<-genes[-(ancov-29)]

interactions2<-0
for(i in genes){
            x <- data[,i]
            model <- lm(x ~ data$CancerType * data$gender)
            interactions2[i] <- summary(aov(model))[[1]][[5]][3]
}
table(interactions2<0.05)
```

```
##
## FALSE  TRUE
##   369   107
```

```
inter<-colnames(data[which(interactions2<0.05)])
#inter

nrofgroups2<-0
for (i in genes) {
   x <- data[,i]
   model <- lm(x ~ data$CancerType * data$gender)
   model1 <- anova(lm(x ~ data$CancerType * data$gender))
      if(model1$"Pr(>F)"[1]<=0.05){
            nrofgroups2[i]<-match(tail(scheffe.test(aov(model), "data$CancerType", console = FALSE)$grou
      }
}
```

Now we decided to choose genes which has intercation with age and many different groups in Scheffe test (by many here we mean 7, we chose this number to get at the end small list, it is higher than in ancova because here we don't have interaction with age, so genes are less informative).

```
length(intersect(colnames(data[,which(nrofgroups2>7)]),inter))
```

```
## [1] 13
```

```
genes2<-intersect(colnames(data[,which(nrofgroups2>7)]),inter)
#genes2
```

Conclusion

The final list of genes.

```
genes<-c(genes1, genes2)
genes
```

```
##  [1] "ACAP2"    "AHNAK"    "ANXA2P3"  "APOOL"    "ARHGAP4"  "ARL3"    
##  [7] "ARMC8"    "ATXN2"    "BEND6"    "BEX2"     "BMS1"     "C11orf49"
## [13] "C11orf9"  "C12orf57" "C19orf10" "CD99"     "CENPV"    "CEP63"   
## [19] "ADM"      "ALDH2"    "APOL2"    "ARRB1"    "ASRGL1"   "BAG3"    
## [25] "BSCL2"    "BTG3"     "C9orf41"  "CADM1"    "CBR1"     "CEBPA"   
## [31] "CELF2"
```

Graphical presentation

```
genesnr<-0
for(i in 1:length(genes)) {
  genesnr[i]<-which(colnames(data)==genes[i])
}
x<-data[,genesnr[1]]
y<-CancerType
z<-data$gender
v<-data$age
#par(mfrow=c(3,2))
#boxplot(x~y*z)
#interaction.plot(y,z,x)
#interaction.plot(y,v,x)
#x<-data[,genesnr[length(genesnr-2)]]
#boxplot(x~y)
#interaction.plot(y,z,x)
#interaction.plot(y,v,x)
```

We have also done another approach how to choose genes. But due to volume limitation, we were forced to choose one, but the other one can be found in our Rmd file which we attached.