Computer exercises on Experimental Design, ANOVA, and the BOOTSTRAP

Exercice 1:

Clear your present workspace by using the command
`rm(list=ls())`
Copy the workspace `ExpDes_1.Rdata` to your local D drive and load it into a R-session.

Perform a sample size calculation for the following situation:

Two group comparison using Affimetrix technology
Array size 20000 genes
500 genes are expected to be changed between both groups with a fold-change of 5
The comparison should produce not more than 10 false positives with a probability of 0.001.
You would like to detect 95% of the true positives.
The gene specific variability is assumed to be $\sigma = 0.7$.

How many arrays per group (same group sizes) are needed if a genewise comparison with a t-test is planned?

Power of a single test is 0.95 and thus beta=0.05

Family error rate (α_F):

> The distribution of false positives is nearly Poisson distributed and the probability that not more than 10 false positives will show up is $1-\sum_{k=0}^{10}\frac{\lambda^k}{k!}e^{-\lambda}$.
> There is a relationship between α_F and λ: $(-1)\cdot\ln(1-\alpha_F) = \lambda$ or $\alpha_F = 1-\exp\{-\lambda\}$
> You have to find α_F by calculating λ and using the above relationship.
>
> Find λ by trial and error using the following commands:
>
> ```
> alpha.F<-0.8
> lambda<-(-1)*log(1-alpha.F)
> 1-sum(dpois(0:10,lambda))
> ```
>
> Find a value `alpha.F` of which gives in the last line a result close to 0.001.
> Start with `alpha.F` = 0.96 and move upwards.

Define the test-specific adjusted alpha (α_0) to be `alpha.F / 19500`

Use the function `sample.size.normal.paralell.rfc` with the right arguments to calculate the study size (2 time the group size):
The arguments of this function are `alpha` (test specific alpha), `beta`, `sigma`, `delta` (the absolute effect which is intended to be discovered).
Save your workspace into `ExpDes_1.Rdata`.

Exercise 2

Clear your present workspace by using the command
`rm(list=ls())`
Copy the workspace `breit.Rdata` to your local D drive and load it into a R-session.

The first column of the matrix `breit` contains the Affimetrics gene IDs.
The matrix `breit.vsn` contains the VSN-normalized values of the 9 arrays.

Use the position number of a gene in the ID list and produce plots and a simple ANOVA with the functions:

```
breit.plot.1.rfc()
breit.plot.2.rfc()
breit.simple.anova.rfc()
```

Try to replicate the *between patients* results for gene 6 by using the formulas discussed in the morning session.

```
x<-breit.vsn[6,]
m.1<-mean(x[1:3])
m.2<-mean(x[4:6])
m.3<-mean(x[7:9])
m.t<-mean(x)
m.ind<-c(m.1,m.2,m.3)
ss.between<-3*sum((m.ind-m.t)^2)
```

Perform the ANOVA analysis of the data by using the function

```
breit.complex.anova.res<-breit.complex.anova.rfc()
```

This object is a list of different components also containing the gene-specific effects a.k (linear) and b.k (quadratic).

```
> names(breit.complex.anova.res)
[1] "ANOVA.table" "resid"       "a"           "b"           "a.k"
[6] "b.k"
```

Make a figure in which a.k is plotted versus b.k and look for interesting genes. Try to find interesting genes and do an individual analysis by using `breit.simple.anova.rfc()` with the appropriate gene number (not ID just position in the list).

```
plot(breit.complex.anova.res$a.k,breit.complex.anova.res$b.k,xlab="linear",
ylab="quadratic")
```

Identify genes with a strong linear effect up or down, identify genes with a mixture of linear and quadratic effect.

Exercise 3:

Clear your present workspace by using the command
`rm(list=ls())`
Copy the workspace `BootRes_1.Rdata` to your local D drive and load it into a R-session.

Create a dataframe which contains informaion on normal distributed observations measured in groups 1 (n_1=23) and 2 (n_2=25).

```
my.data<-
data.frame(group=c(rep(1,23),rep(2,25)),values=c(rnorm(23,1,1),rnorm(25,2.5,1)))
```

Give some descriptive statistics of the observed data:
`group.1.summary<-summary(my.data$value[my.data$group==1])`

What kind of object is `group.1.summary ?` Calculate a `group.1.summary`. Make a table which contains the summaries of the observed data.

```
descrive.stat.table<-rbind(group.1.summary,group.2.summary)
```

Give some nice names to the Row-names of the object `descrive.stat.table`.

```
> dimnames(describ.stat.table)[[1]]<-c("Group 1","Group 2")
> describ.stat.table
           Min.   1st Qu. Median   Mean 3rd Qu.  Max.
Group 1 -0.8754 -0.003584 0.5653 0.7326   1.441 2.483
Group 2  0.3748  1.514000 2.1910 2.1380   2.507 4.333
```

The standard deviation of the measurement is not given in the table. Please add it to the information (with the same amount of decimals as the other numbers).

```
STD<-sqrt(c(var(my.data$value[my.data$group==1]),
var(my.data$value[my.data$group==2])))

describ.stat.table<-cbind(describ.stat.table,round(STD,3))
```

Exercise 4: Bootstrap t-test

Use the function `twosam` to calculate the t-statistics for the data in your dataframe `my.data`.

```
t.obs<-twosam(my.data$value[my.data$group==1], my.data$value[my.data$group==2])
```

Write the following function to imitate the unknown random process:

```
> boot.two.sample <- function (i,obs,gr)
      {
       l<-length(gr)
       ss<-sample(l,l,replace=T)
       gr.new<-gr[ss]
       x.1<-obs[gr.new==1]
       x.2<-obs[gr.new==2]
       return(twosam(x.1,x.2))
      }
```

Run a bootstrap with 999 samples:

```
boot.res<-unlist( lapply(1:999,boot.two.sample,obs=my.data$value,gr=my.data$group))
```

Perform a two-sided bootstrap t-test by calculating the p-value:

```
p.boot<-sum(ifelse(abs(boot.res)>abs(t.obs),1,0))/999
```

Interpret the result.
Compare the bootstrap result to a standard t-test which can be calculated by

```
t.test(my.data$value[my.data$group==1],my.data$value[my.data$group==2])
```

Exercise 5: Bootstrap confidence intervals

Read from the result of the standard t-test in R the estimate for the difference of the group means and its 95% confidence interval.

```
t.test.res<-
t.test(my.data$value[my.data$group==1],my.data$value[my.data$group==2])

diff.obs<- t.test.res$estimate[1]<- t.test.res$estimate[2]
```

Calculate a bootstrap sample of mean differences by using the function

```
> boot.mean <- function (i,obs,gr)
      {
        l<-length(gr)
        ss<-sample(l,l,replace=T)
        gr.new<-gr[ss]
        x.1<-obs[gr.new==1]
        x.2<-obs[gr.new==2]
        return(mean(x.1)-mean(x.2))
      }
```

Make a bootstrap sample of mean differences with 999 replicates.

```
mean.res<-unlist(lapply(1:999,boot.mean,obs=my.data$value,gr=my.data$group))
```

Use two approaches to calculate a 95% bootstrap confidence interval for the mean difference.

1.) calculate the variance of the bootstrap sample v^2 [`var(mean.res)`] and use
[diff.obs – 1.96·v; diff.obs + 1.96·v]

2.) order the bootstrap sample and take the 25^{th} element, $diff_{0.025}$, and the 975^{th} element, $diff_{0.975}$ to get
[2·diff.obs – $diff_{0.975}$; 2·diff.obs – $diff_{0.025}$]

```
mean.res<-sort(mean.res)
diff.975<- mean.res[975]
diff.025<- mean.res[25]
```

Compare the bootstrap confidence intervals with the exact 95% confidence intervals. Is the true mean difference of 1 – 2.5 = - 1.5 included in the confidence intervals?

Exercise 6: Balanced 2 by 2 ANOVA

Perform for the two.by.two data an ANOVA to study the Mutation by treatment effect.
The data looks as follows and gives VSN transformed values of signal intensities:

```
> two.by.two[1,]
  W.NT.1   W.TR.1   W.NT.2   W.TR.2   W.NT.3   W.TR.3   W.NT.4   W.TR.4
12.64218 12.92974 12.97218 13.28092 12.71829 13.25633 13.17850 12.89417
  W.NT.5   W.TR.5   M.NT.6   M.TR.6   M.NT.7   M.TR.7   M.NT.8   M.TR.8
13.33704 12.57546 13.10308 12.75008 12.57069 12.51956 12.83289 12.47562
  M.NT.9   M.TR.9  M.NT.10  M.TR.10
13.02821 12.71112 12.28821 12.37065
```

W – wild type / M – mutation
NT – not treated / TR – treated
There are 10 treated and 10 non-treated samples, as well as 10 wild type and 10 mutations.
The function `two.by.two.anova.rfc` performs the analysis and returns a list with two components: the residuals and the interesting `ge.wt.nt` values.

```
two.by.two.anova.res<-two.by.two.anova.rfc()
```

Plot a histogram of the residuals:

```
hist(two.by.two.anova.res[[1]],main="Histogram of residuals")
```

Calculate the standard deviation of the residuals. `sqrt(var(two.by.two.anova.res[[1]]))`
Look at the differential gene expression with respect to treatment between wild type and mutations. What is the minimum, what the maximum log fold change?
The fold-change for a differential expression for gene i is calculated by

```
exp(2 x two.by.two.anova.res[[2]][i])
```

How many genes would be differentially expressed if one uses the 95% CI based on 1.96*SE?

```
table(2*abs(two.by.two.anova.res[[2]])>1.96*sqrt(var(two.by.two.anova.res$res
id)))
```