

# Statistics 2 Exercises

Pascal Pilz, k12111234

2024-01-29

## Exercise 1

```
x <- c(0,3,3,3,6,0,1,0,0,1,1,3,0,4,3,1,4,6,4,2,0,1,1,0,1,1,0,1,0,3)

cat(sprintf("mean: %.3f, sd: %.3f,", mean(x), sd(x)),
    "quantiles:", sprintf("%.3f", quantile(x)))

## mean: 1.767, sd: 1.794, quantiles: 0.000 0.000 1.000 3.000 6.000
```

## Exercise 2

```
x1 <- c(157,184,162,166,168,163,178,185,190,187,176,174,179)

cat(sprintf("mean: %.3f, sd: %.3f,", mean(x1), sd(x1)),
    "quantiles:", sprintf("%.3f", quantile(x1)))

## mean: 174.538, sd: 10.572, quantiles: 157.000 166.000 176.000 184.000 190.000

x2 <- c(47,85,66,83,62,73,84,94,99,96,76,94,74)

cat(sprintf("mean: %.3f, sd: %.3f,", mean(x2), sd(x2)),
    "quantiles:", sprintf("%.3f", quantile(x2)))

## mean: 79.462, sd: 15.180, quantiles: 47.000 73.000 83.000 94.000 99.000
```

## Exercise 3

For this we realize that the final income is 200% of the initial income, thus we simply take the 10th root of 2.

```
x <- 2 ^ (1/10)

cat(sprintf("The average annual growth rate is %s%%", round((x-1)*100, 3)))

## The average annual growth rate is 7.177%
```

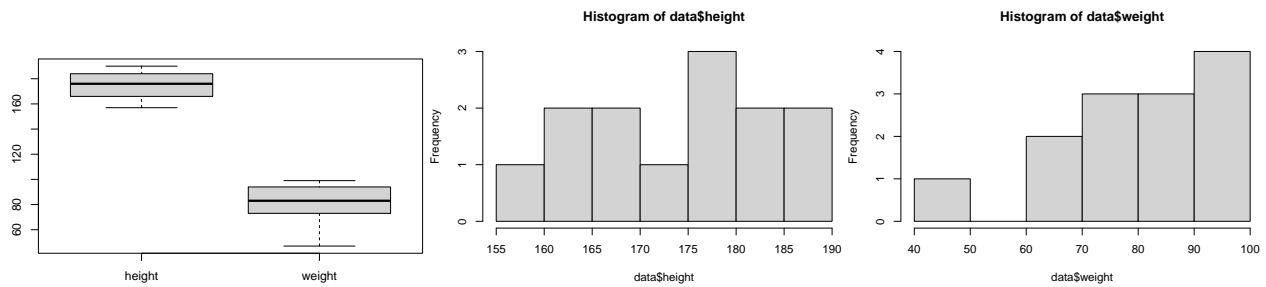
## Exercise 4

We create boxplots of the data from example 2:

```
x1 <- c(157,184,162,166,168,163,178,185,190,187,176,174,179)
x2 <- c(47,85,66,83,62,73,84,94,99,96,76,94,74)
data <- data.frame("height"=x1, "weight"=x2)
```

```
boxplot(data)

hist(data$height)
hist(data$weight)
```



## Exercise 5

We have haemoglobin data of 16 patients:

```
x <- c(7.2, 7.7, 8.0, 8.1, 8.3, 8.4, 8.4, 8.5, 8.6, 8.7, 9.1, 9.1, 9.1, 9.8, 10.1, 10.3)
```

a)

As a point estimator for the true mean and true standard deviation we can calculate the sample mean and sample standard deviation:

```
n <- length(x)

sample_mean <- (1/n) * sum(x)
sample_var <- (1/(n-1)) * sum((x-sample_mean)^2)
sample_std <- sqrt(sample_var)

print(sprintf("sample mean: %.3f, sample standard deviation: %.3f",
              sample_mean, sample_std))
```

```
## [1] "sample mean: 8.713, sample standard deviation: 0.844"
```

b)

Calculating a 95% confidence interval for the true mean value of haemoglobin based on the sample above.

For this we have the following formula:

$$\mu \in \left[ \bar{x} \pm t_{n-1, 1-\alpha/2} \cdot \frac{s}{\sqrt{n}} \right]$$

where  $\alpha$  is the chosen risk level,  $t_{n-1, 1-\alpha/2}$  is corresponding quantile of the  $t$ -distribution, and  $s$  is the sample variance ( $s/\sqrt{n}$  is the standard error).

We have 16 samples and a confidence level of 95% ( $\alpha = 0.05$ ), we find in the table  $t_{15, 0.975} = 2.13$ . To make it more precise and as an exercise in R we also get the value from the `qt` function.

```
alpha <- 0.05
t <- qt(1-(alpha/2), n-1)

val <- t * sample_std / sqrt(n)
```

```

conf_lower <- sample_mean - val
conf_upper <- sample_mean + val

print(sprintf("P(true mean in [%.3f,%.3f]) = %s",
              conf_lower, conf_upper, 1-alpha))

```

```
## [1] "P(true mean in [8.263,9.162]) = 0.95"
```

c)

What would happen with the confidence interval, if the sample size was not 16 patients but 160 patients?

The confidence interval would get smaller, since both the value of the  $t$ -distribution would get smaller ( $t_{159,0.975} = 1.974$ ) and  $n$ , which is in the numerator of the formula, would increase. If the sample mean and sample variance would remain the same we would get the following:

```

n_new <- 159
alpha <- 0.05
t <- qt(1-(alpha/2), n_new-1)

val <- t * sample_std / sqrt(n_new)

conf_lower <- sample_mean - val
conf_upper <- sample_mean + val

print(sprintf("P(true mean in [%.3f,%.3f]) = %s",
              conf_lower, conf_upper, 1-alpha))

```

```
## [1] "P(true mean in [8.580,8.845]) = 0.95"
```

## Exercise 6

We have a sample of 11 persons, 2 of which smoke. We want to calculate two different 90% confidence intervals.

```

alpha <- 0.1
q <- 1 - (alpha/2)
n <- 11
k <- 2
p <- k/n
print(sprintf("p: %.3f, n*p*(1-p): %.3f", p, n*p*(1-p)))

```

```
## [1] "p: 0.182, n*p*(1-p): 1.636"
```

### a) 90% asymptotic confidence interval for the true proportion of smokers

First we get the point estimator  $\hat{\pi}$  given by  $\hat{\pi} = p = \frac{k}{n}$ , then we use the formula found in the slides to calculate the confidence interval for  $\alpha = 0.1$ :

```

val <- qnorm(q) * sqrt(p*(1-p)/n)
conf_lower <- p - val
conf_upper <- p + val
print(sprintf("P(true proportion in [%.3f,%.3f]) = %s",
              conf_lower, conf_upper, 1-alpha))

```

```
## [1] "P(true proportion in [-0.009,0.373]) = 0.9"
```

As we can see, the result is not very convincing.

## b) 90% exact confidence interval for the true proportion of smokers

For this we use formula found on slide 14, which involves the  $F$ -distribution:

```
conf_lower <- k / (k + (n-k+1) * qf(q, 2*(n-k+1), 2*k))
conf_upper <- (k+1) * qf(q, 2*(k+1), 2*(n-k)) /
              ((n-k) + (k+1) * qf(q, 2*(k+1), 2*(n-k)))
print(sprintf("P(true proportion in [%.3f,%.3f]) = %s",
              conf_lower, conf_upper, 1-alpha))
```

```
## [1] "P(true proportion in [0.033,0.470]) = 0.9"
```

## Exercise 7

We have the following number of bacteria per sample taken for 10 different sample:

```
x <- c(13,14,12,11,16,16,17,20,20,21)
n <- length(x)

sample_mean <- mean(x)
sample_std <- sd(x)

print(sprintf("mean %.3f, std %.3f", sample_mean, sample_std))
```

```
## [1] "mean 16.000, std 3.528"
```

## a) 95% confidence interval for the true mean

```
alpha <- 0.05
t <- qt(1-(alpha/2), n-1)

val <- t * sample_std / sqrt(n)

conf_lower <- sample_mean - val
conf_upper <- sample_mean + val

print(sprintf("P(true mean in [%.3f,%.3f]) = %s",
              conf_lower, conf_upper, 1-alpha))
```

```
## [1] "P(true mean in [13.476,18.524]) = 0.95"
```

## b) test if “true mean” differs from $\mu_0 = 14$

We want to construct a test that checks whether the “true mean”  $\mu$  of bacteria count differs from  $\mu_0 = 14$ . We want to use a type I error of 5%, we can assume normality.

For this, let us define the following:

- Null hypothesis  $H_0$ :  $\mu = \mu_0$
- Alternative hypothesis  $H_1$ :  $\mu \neq \mu_0$

What we want to verify is that the true mean differs from 14, thus we want to verify the alternative hypothesis. For this, we show that the null hypothesis can be rejected.

In our example, a one-sample t-test is appropriate:

```
crit_val <- qt(1-(alpha/2), n-1)
test_stat <- abs(sqrt(n)*(sample_mean-14)/sample_std)

print(sprintf("critical value: %.3f, test statistic: %.3f",
              crit_val, test_stat))
```

```
## [1] "critical value: 2.262, test statistic: 1.793"
```

Since the critical value is greater than the test statistic, we cannot reject the null hypothesis.

## Exercise 8

```
A <- c(7.2,7.7,8.0,8.1,8.3,8.4,8.4,8.5,8.6,8.7,9.1,9.1,9.1,9.8,10.1,10.3)
B <- c(8.1,9.2,10.0,10.4,10.6,10.9,11.1,11.9,12.0,12.1)

n <- length(A)
m <- length(B)

alpha <- 0.05

mean_A <- sum(A)/n
mean_B <- sum(B)/m

s_A <- sqrt( sum(A^2)/n - mean_A^2 )
s_B <- sqrt( sum(B^2)/m - mean_B^2 )

cat(sprintf("sample variance A: %.3f, sample variance B: %.3f", s_A, s_B))

## sample variance A: 0.818, sample variance B: 1.218
```

### F-test

We want to perform an  $F$ -test to see if the “true” variances differ.

For this, we formulate the hypothesis:

- $H_0: \sigma_A^2 = \sigma_B^2$
- $H_1: \sigma_A^2 \neq \sigma_B^2$

With the test statistic  $F(x, y) = s_x^2/s_y^2$  for  $s_x^2 > s_y^2$  and the critical value  $F_{n_x-1, n_y-1, 1-\alpha}$ .

We have that that  $s_A < s_B$ , thus we have the test statistic  $F = s_B^2/s_A^2$ .

```
test_statistic <- s_B^2 / s_A^2
critical_value <- qf(1-alpha, m-1, n-1)

cat(sprintf("test statistic: %.3f, critical value: %.3f",
            test_statistic, critical_value))
```

```
## test statistic: 2.220, critical value: 2.588
```

```
cat(sprintf("The null hypothesis %s be rejected. p-value %s",
            if (test_statistic > critical_value) "can" else "cannot",
            df(test_statistic, m-1, n-1)))
```

```
## The null hypothesis cannot be rejected. p-value 0.116071862962748
```

## Levene-test

I choose to use the mean to calculate the spread, just to make things easier.

```
z_A <- abs(A - mean_A)
mean_z_A <- sum(z_A)/n

z_B <- abs(B - mean_B)
mean_z_B <- sum(z_B)/m

z <- sum(c(z_A, z_B)) / (n+m)

test_statistic <- ( (n+m-2) * (n*(mean_z_A - z)^2 + m*(mean_z_B - z)^2) ) /
  ( sum((z_A - mean_z_A)^2) + sum((z_B - mean_z_B)^2) )
critical_value <- qf(1-alpha, 1, n+m-2)

cat(sprintf("test statistic: %.3f, critical value: %.3f",
            test_statistic, critical_value))

## test statistic: 1.594, critical value: 4.260
cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic > critical_value) "can" else "cannot" ))

## The null hypothesis cannot be rejected.
suppressMessages(library(car))
suppressWarnings(
cat(sprintf("p-value of: manual implementation %.3f, package 'car' %.3f",
            1 - pf(test_statistic, 1, n+m-2),
            leveneTest(c(A,B), c(rep(0, n), rep(1, m)), center="mean")$`Pr(>F)`[1]))
)

## p-value of: manual implementation 0.219, package 'car' 0.219
```

## Test to verify if there is a difference between the “true means”

Since both  $F$ -test and Levene-test cannot reject the null hypothesis, we can assume homoscedasticity.

We will perform an independent two-sample t-test:

```
s_p <- sqrt( ((n-1)*(s_A^2) + (m-1)*(s_B^2)) / (n+m-2) )

test_statistic <- abs(mean_A - mean_B) * sqrt(n*m/n+m) / s_p

critical_value <- qt(1-(alpha/2), n+m-2)

cat(sprintf("test statistic: %.3f, critical value: %.3f",
            test_statistic, critical_value))

## test statistic: 8.687, critical value: 2.064
cat(sprintf("The null hypothesis %s be rejected. p-value %s",
            if (test_statistic > critical_value) "can" else "cannot",
            dt(test_statistic, n+m-2)))

## The null hypothesis can be rejected. p-value 7.55120171274825e-09
```

## Confidence interval for difference of “true means”

```
val <- qt(1-(alpha/2), n+m-2) * s_p * sqrt(1/n + 1/m)

cat(sprintf("The true difference of the means is with %.f%% in [%.3f, %.3f]",
            (1-alpha)*100, mean_A-mean_B - val, mean_A-mean_B + val))

## The true difference of the means is with 95% in [-2.739, -1.096]
```

## Exercise 9

```
mean_A <- 22.13
mean_B <- 18.68

s_A <- 3.74
s_B <- 1.21

n <- 8
m <- 6
```

## Homoscedasticity

First we conduct an F-test to see whether we can assume homoscedasticity.

```
test_statistic <- s_A^2 / s_B^2
critical_value <- qf(0.95, n-1, m-1)

cat(sprintf("test statistic: %.3f, critical value: %.3f",
            test_statistic, critical_value))

## test statistic: 9.554, critical value: 4.876

cat(sprintf("The null hypothesis %s be rejected. p-value %s",
            if (test_statistic > critical_value) "can" else "cannot",
            df(test_statistic, m-1, n-1)))

## The null hypothesis can be rejected. p-value 0.00150766926950449
```

## Test for difference of means

As we can see, we cannot assume homoscedasticity. Therefore, we use Welch's two-sample t-test:

```
Z_A <- ( ((s_A^2)/n) / ((s_A^2)/n + (s_B^2)/m) )^2 / (n-1)
Z_B <- ( ((s_B^2)/n) / ((s_A^2)/n + (s_B^2)/m) )^2 / (m-1)
df <- 1 / (Z_A + Z_B)

cat(sprintf("Z_A %.5f, Z_B %.5f, df %.2f", Z_A, Z_B, df))

## Z_A 0.11001, Z_B 0.00169, df 8.95

test_statistic <- (mean_A - mean_B) / sqrt(s_A^2/n + s_B^2/m)
critical_value <- qt(1-(0.05/2), df)

cat(sprintf("test statistic: %.3f, critical value: %.3f",
            test_statistic, critical_value))

## test statistic: 2.444, critical value: 2.264
```

```
cat(sprintf("The null hypothesis %s be rejected. p-value %s",
            if (test_statistic > critical_value) "can" else "cannot",
            dt(test_statistic, df)))
```

```
## The null hypothesis can be rejected. p-value 0.0304819517434573
```

## Exercise 10

```
# Relevant difference
d <- 13

# Sample standard deviations
s_A <- 15
s_B <- 18

# Type 1 and type 2 error
a <- 0.025
b <- 0.1
```

Since the size of the two groups will be the same we can get the pooled variance by taking the mean of the two sample variances.

```
# Pooled standard deviation
s_p <- sqrt((s_A^2 + s_B^2) / 2)
cat(sprintf("s_p: %s", s_p), "\n")
```

```
## s_p: 16.5680415257809
```

```
# Quantiles of normal distribution
u_a <- qnorm(1-a)
u_b <- qnorm(1-b)
cat(sprintf("u_a: %s, u_b: %s", u_a, u_b), "\n")
```

```
## u_a: 1.95996398454005, u_b: 1.2815515655446
```

```
# Sample size estimation for one-sided t-test, rounded up for conservative approach
n <- ceiling(2 * (u_a + u_b)^2 * s_p^2 / d^2)
cat(sprintf("We need %s patients total, %s per group.", 2*n, n))
```

```
## We need 70 patients total, 35 per group.
```

## Exercise 11

We do a Kolmogorov-Smirnov test to check for normality.

```
mydata <- c(3,4,5,6,7)

Sn <- c(0.2, 0.4, 0.6, 0.8, 1)
Sn_shift <- c(0, 0.2, 0.4, 0.6, 0.8)

F0 <- pnorm(mydata, mean=3, sd=2)

test_statistic <- max( c(max(abs(Sn - F0)), max(abs(Sn_shift - F0))) )
cat(sprintf("Test statistic sup|Sn(x) - F0(x)| = %s", test_statistic), "\n")
```

```
## Test statistic sup|Sn(x) - F0(x)| = 0.5
```



```
# We get the critical value from the table, for n=5 and alpha=0.1
critical_value <- 0.509
cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic > critical_value) "can" else "cannot"))
```

```
## The null hypothesis cannot be rejected.
```

## Exercise 12

Here we do a Lilliefors test to check for normality.

```
mydata <- c(3,4,5,6,7)

Sn <- c(0.2, 0.4, 0.6, 0.8, 1)
Sn_shift <- c(0, 0.2, 0.4, 0.6, 0.8)

F0 <- pnorm(mydata, mean=mean(mydata), sd=sd(mydata))

test_statistic <- max( c(max(abs(Sn - F0)), max(abs(Sn_shift - F0))) )
cat(sprintf("Test statistic sup|Sn(x) - F0(x)| = %s", test_statistic), "\n")
```

```
## Test statistic sup|Sn(x) - F0(x)| = 0.136455371567231
```

```
# We get the critical value from the table, for n=5 and alpha=0.1
critical_value <- 0.315
cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic > critical_value) "can" else "cannot"))
```

```
## The null hypothesis cannot be rejected.
```

## Exercise 13

```
x <- c(13,14,12,11,16,16,17,20,20,21)
n <- length(x)

m <- mean(x)
s <- sd(x)

print(sprintf("mean %.3f, std %.3f", m, s))
```

```
## [1] "mean 16.000, std 3.528"
```

```
ref_val <- 14
a <- 0.05
b <- 0.2
```

Since we want to know whether there is a difference, we use a two-sided one-sample t-test.

```
ua <- qnorm(1-a/2)
ub <- qnorm(1-b)
cat(sprintf("u(1-a/2) %.3f, u(1-b) %.3f", ua, ub), "\n")
```

```
## u(1-a/2) 1.960, u(1-b) 0.842
```

```
d <- m - ref_val
```

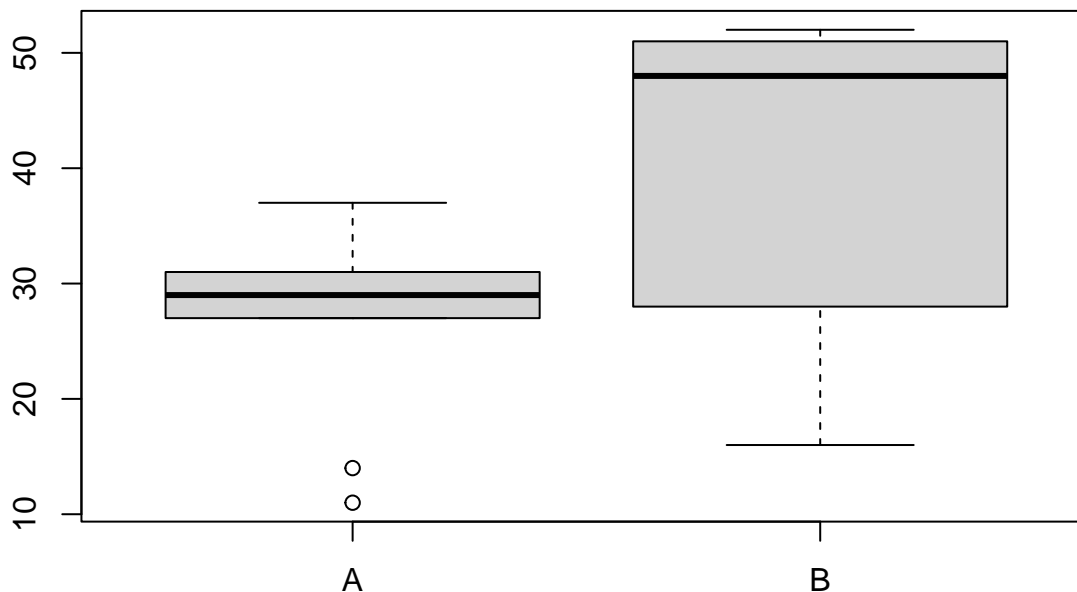
```
n <- (ua+ub)^2 * s^2 / d^2
cat(sprintf("We need %s samples.", ceiling(n)))
```

```
## We need 25 samples.
```

## Exercise 14

```
A <- sort(c(11,37,32,27,29,29,27,31,14))
B <- sort(c(52,38,51,48,49,52,28,17,16))

boxplot(A, B, names=c("A", "B"))
```



## Check for normality

First we check if our data is normally distributed. For this, we use a Lilliefors test.

```
ma <- mean(A)
sa <- sd(A)
S <- 1:length(A) / length(A)
S_shift <- c(0, 1:(length(A)-1)/length(A))
F0 <- pnorm(A, ma, sa)

test_statistic <- max(max(abs(S-F0)), max(abs(S_shift-F0)))
critical_value <- 0.249
cat(sprintf("Test statistic %s", test_statistic), "\n")
```

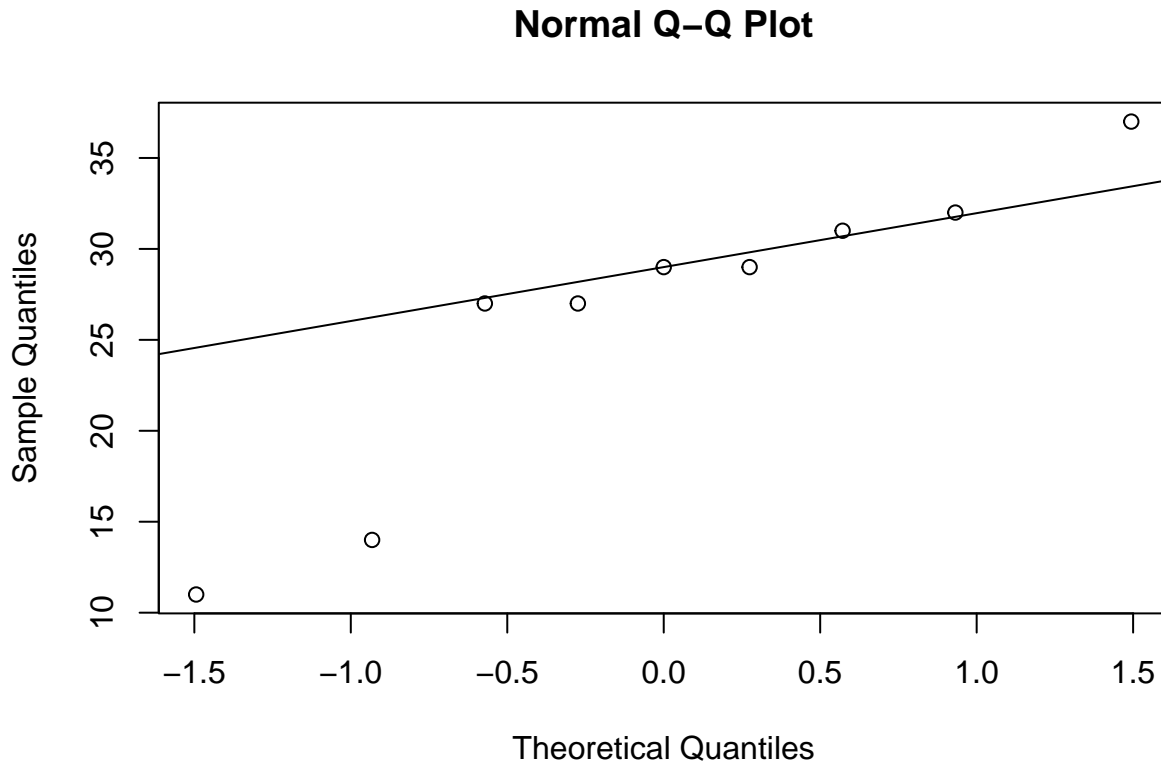
```
## Test statistic 0.309253452744006
```

```
cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic > critical_value) "can" else "cannot"))
```

```
## The null hypothesis can be rejected.
```

Since A is not normally distributed we forgo checking B and immediately go over to using a non parametric test. I am not sure if this is correct, since we do have ties in both groups and therefore the KS test should maybe not be used. We can also a QQ-plot:

```
qqnorm(A)
qqline(A)
```



## Test

We use a Mann-Whitney-U Test. We need to use the exact MWU test because we have ties

```
#install.packages("coin")
library(coin)
```

```
## Loading required package: survival
```

```
##
```

```
## Attaching package: 'coin'
```

```
## The following object is masked _by_ '.GlobalEnv':
```

```
##
```

```
## alpha
```

```
data<-c(A,B)
```

```
group<-as.factor(c(rep("A",length(A)),rep("B",length(B))))
```

```
wilcox_test(data~group,distribution = "exact")
```

```
##
```

```
## Exact Wilcoxon-Mann-Whitney Test
```

```
##
```

```
## data: data by group (A, B)
```

```
## Z = -1.9014, p-value = 0.05895
```

```
## alternative hypothesis: true mu is not equal to 0
```

This indicates that we cannot reject the null hypothesis.

## Exercise 15

```
A <- c(54.5,60.4,85.6,78.2,120.6,121)
B <- c(55.5,69.6,86.7,81.6,116.5,115)
```

We know that normality of the data cannot be assumed. Therefore, we use a Wilcoxon Test for paired data.

### Manual Wilcoxon Test

```
d <- A - B

df <- data.frame(difference=d, rank=rank(abs(d)))
print(df)

##   difference rank
## 1      -1.0    1
## 2     -9.2    6
## 3     -1.1    2
## 4     -3.4    3
## 5      4.1    4
## 6      6.0    5

df <- df[df$difference!=0,]
pos_rank_sum <- sum(df$rank[df$difference>0])
neg_rank_sum <- sum(df$rank[df$difference<0])
cat(sprintf("positive rank sum %s", pos_rank_sum), "\n")

## positive rank sum 9

cat(sprintf("negative rank sum %s", neg_rank_sum), "\n")

## negative rank sum 12

test_statistic <- min(pos_rank_sum, neg_rank_sum)
cat(sprintf("test statistic %s", test_statistic), "\n")

## test statistic 9

critical_value <- 0

cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic <= critical_value) "can" else "cannot"))

## The null hypothesis cannot be rejected.
```

### Implemented Test

```
wilcox.test(A,B,paired=TRUE,correct=FALSE)

##
## Wilcoxon signed rank exact test
##
## data: A and B
## V = 9, p-value = 0.8438
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcoxsign_test(A~B,distribution = "exact",zero.method="Wilcoxon")

##
## Exact Wilcoxon Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -0.31449, p-value = 0.8438
## alternative hypothesis: true mu is not equal to 0
wilcoxsign_test(A~B,distribution = "asymptotic",zero.method="Wilcoxon")

##
## Asymptotic Wilcoxon Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -0.31449, p-value = 0.7532
## alternative hypothesis: true mu is not equal to 0
wilcoxsign_test(A~B,distribution = approximate(nresample = 10000),zero.method="Wilcoxon")

##
## Approximative Wilcoxon Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -0.31449, p-value = 0.8456
## alternative hypothesis: true mu is not equal to 0
```

## Exercise 16

### Assumption

```
n <- 21
k <- 18
p <- k/n

cat(sprintf("n*p*(1-p) = %.2f", n*p*(1-p)))

## n*p*(1-p) = 2.57
```

Since  $n \cdot p \cdot (1 - p) < 9$  we cannot use the asymptotic test. Therefore, we use the exact binomial test.

### Exact Binomial Test

```
e <- 0.6 # Expected cure rate

p1 <- 0
for (i in 0:k) {
  p1 <- p1 + choose(n, i) * e^i * (1-e)^(n-i)
}
cat(sprintf("p1: %.3f", p1), "\n")

## p1: 0.998
```

```

p2 <- 0
for (i in k:n) {
  p2 <- p2 + choose(n, i) * e^i * (1-e)^(n-i)
}
cat(sprintf("p2: %.3f", p2), "\n")

## p2: 0.011

p_val <- 2*min(p1, p2)
a <- 0.05 # Type 1 error
cat(sprintf("The null hypothesis %s be rejected. p.value %.3f",
            if (p_val < a) {"can"} else {"can not"}, p_val))

## The null hypothesis can be rejected. p.value 0.022

```

## Exercise 17

Since our data is ordinal we can perform a Mann-Whitney-U test to check if the two groups differ significantly.

```

A <- c(5,7,6,4,6,0)
B <- c(3,10,4,5,6,0)

library(coin)
data<-c(A,B)
group<-as.factor(c(rep("A",length(A)),rep("B",length(B))))
wilcox_test(data~group,distribution = "exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: data by group (A, B)
## Z = 0.40531, p-value = 0.7359
## alternative hypothesis: true mu is not equal to 0

```

And so we can see that the null hypothesis cannot be reject, the data are likely from the same distribution.

## Exercise 18

Since we have  $n = 300 > 40$  and  $e_{ij} \geq 5$  we use a Chi-square test.

### Expected data

```

A <- c(70,7,13,10)
B <- c(150,20,20,10)

A_p <- A / sum(A) * 100
A_e <- c(A+B) * sum(A)/(sum(A)+sum(B))
A_e_p <- A_e / sum(A_e) * 100

B_p <- B / sum(B) * 100
B_e <- c(A+B) * sum(B)/(sum(A)+sum(B))
B_e_p <- B_e / sum(B_e) * 100

cat(" Expected A:          ", sprintf("%.2f", A_e), "\n",

```

```

"Observed A:          ", sprintf("%.2f", A), "\n",
"Expected A percent: ", sprintf("%.1f%%", B_e_p), "\n",
"Observed A percent: ", sprintf("%.1f%%", A_p), "\n",
"\n",
"Expected B:          ", sprintf("%.2f", B_e), "\n",
"Observed B:          ", sprintf("%.2f", B), "\n",
"Expected B percent: ", sprintf("%.1f%%", B_e_p), "\n",
"Observed B percent: ", sprintf("%.1f%%", B_p), "\n")

```

```

## Expected A:          73.33   9.00  11.00   6.67
## Observed A:          70.00   7.00  13.00  10.00
## Expected A percent:  73.3%   9.0%  11.0%   6.7%
## Observed A percent:   70%    7%   13%   10%
##
## Expected B:          146.67  18.00  22.00  13.33
## Observed B:          150.00  20.00  20.00  10.00
## Expected B percent:  73.3%   9.0%  11.0%   6.7%
## Observed B percent:   75%   10%   10%    5%

```

## Measure of divergence and test

```

# Type 1 error
a <- 0.05

test_statistic <- sum( ( c(A, B) - c(A_e, B_e) )^2 / c(A_e, B_e) )
critical_value <- qchisq(1-a, 1)

cat(sprintf("The null hypothesis %s be rejected.",
            if (test_statistic > critical_value) {"can"} else {"can not"}))

## The null hypothesis can be rejected.

```

## Exercise 19

Since we have 25 samples and we have one of fields is less than 5, we use Fisher's exact test.

```

data <- matrix(c(4, 5, 13, 3), 2, 2)
fisher.test(data)

##
## Fisher's Exact Test for Count Data
##
## data: data
## p-value = 0.087
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.02040914 1.54909085
## sample estimates:
## odds ratio
##  0.1998047

```

We can see that there is no significant difference.

## Exercise 20

```
data <- matrix(c(5, 1, 2, 6), 2, 2)
```

We want to compare the asymptotic Chi-square test, Chi-square with Yate's correction, and exact Fisher test. Since we have a sample size of XX and cells with values of less than 5 we already know that the assumption of the Chi-square test are violated.

### Chi-square

```
chisq.test(data, correct=FALSE)
```

```
## Warning in chisq.test(data, correct = FALSE): Chi-squared approximation may be
## incorrect
##
## Pearson's Chi-squared test
##
## data:  data
## X-squared = 4.6667, df = 1, p-value = 0.03075
```

### Chi-square with Yate's correction

```
chisq.test(data, correct=TRUE)
```

```
## Warning in chisq.test(data, correct = TRUE): Chi-squared approximation may be
## incorrect
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  data
## X-squared = 2.625, df = 1, p-value = 0.1052
```

### Fisher's exact test

```
fisher.test(data)
```

```
##
## Fisher's Exact Test for Count Data
##
## data:  data
## p-value = 0.1026
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0.7203391 817.9820107
## sample estimates:
## odds ratio
##    11.71745
```

### Conclusion

As we can, the p-value for the corrected Chi-square test and Fisher's exact test are quite close, whereas the uncorrected Chi-square test produces a drastically different result.



## Exercise 21

```
p <- 100
n <- 200

pre <- p / (p + n)

tp <- 90
fp <- 30
tn <- 170
fn <- 10

se <- tp / p
sp <- tn / n

ppv_1 <- tp / (tp + fp)
ppv_2 <- (se * pre) / (se * pre + (1 - sp) * (1 - pre))

npv_1 <- tn / (tn + fn)
npv_2 <- (sp * (1 - pre)) / (sp * (1 - pre) + (1 - se) * pre)

cat(sprintf(" sensitivity: %.3f\n", se),
      sprintf("specificity: %.3f\n", sp),
      "\n",
      sprintf("positive predictive value (excluding prevelance): %.3f\n", ppv_1),
      sprintf("positive predictive value (including prevelance): %.3f\n", ppv_2),
      "\n",
      sprintf("negative predictive value (excluding prevelance): %.3f\n", npv_1),
      sprintf("negative predictive value (including prevelance): %.3f\n", npv_2))

## sensitivity: 0.900
## specificity: 0.850
##
## positive predictive value (excluding prevelance): 0.750
## positive predictive value (including prevelance): 0.750
##
## negative predictive value (excluding prevelance): 0.944
## negative predictive value (including prevelance): 0.944
```

## Exercise 22

```
a <- 5
b <- 3
c <- 2
d <- 10

P1 <- a / (a+b)
P0 <- c / (c+d)

cat(sprintf("p(ill | fish) = %.3f, p(ill | !fish) = %.3f", P1, P0))

## p(ill | fish) = 0.625, p(ill | !fish) = 0.167
```

## Odds-ratio and confidence interval

### Odds

```
odds_fish <- P1 / (1 - P1)
odds_nfish <- P0 / (1 - P0)

cat(sprintf(" odds(fish)  = %.3f, odds(!fish) = %.3f", odds_fish, odds_nfish))

## odds(fish)  = 1.667, odds(!fish) = 0.200
```

### Odds-ratio

```
OR <- odds_fish / odds_nfish
SE <- (1/a + 1/b + 1/c + 1/d) ^ (0.5)

cat(sprintf("OR(fish) = %.3f, 95%% confidence interval: [%.3f, %.3f]",
            OR, OR*exp(-qnorm(0.975) * SE), OR*exp(+qnorm(0.975) * SE)))

## OR(fish) = 8.333, 95% confidence interval: [1.034, 67.142]
```

### Statistical test for odds-ratio

```
z <- log(OR)/SE

cat(sprintf("z = %.3f, significant difference from 1: %s", z, z > qnorm(0.975)))

## z = 1.992, significant difference from 1: TRUE
```

## Risk-ratio and confidence interval

```
RR <- P1 / P0
SE <- ( ( b / (a*(a+b)) ) + ( d / (c*(c+d)) ) ) ^ (0.5)

cat(sprintf("RR(fish) = %.3f, 95%% confidence interval: [%.3f, %.3f]",
            RR, RR*exp(-qnorm(0.975) * SE), RR*exp(+qnorm(0.975) * SE)))

## RR(fish) = 3.750, 95% confidence interval: [0.949, 14.821]
```

## Exercise 23

```
a <- 965
b <- 2691
c <- 957
d <- 2855

P1 <- a / (a+b)
P0 <- c / (c+d)

cat(sprintf("p(s | m) = %.3f, p(s | f) = %.3f", P1, P0))

## p(s | m) = 0.264, p(s | f) = 0.251
```

## Odds ration

### Odds

```
odds_m <- P1 / (1 - P1)
odds_f <- P0 / (1 - P0)

cat(sprintf(" odds(male)  = %.3f, odds(female) = %.3f", odds_m, odds_f))

## odds(male)  = 0.359, odds(female) = 0.335
```

### Odds-ratio

```
OR <- odds_m / odds_f
SE <- (1/a + 1/b + 1/c + 1/d) ^ (0.5)

cat(sprintf("OR(male) = %.3f, 95%% confidence interval: [%.3f, %.3f]",
            OR, OR*exp(-qnorm(0.975) * SE), OR*exp(+qnorm(0.975) * SE)))

## OR(male) = 1.070, 95% confidence interval: [0.964, 1.187]
```

## Risk ratio

```
RR <- P1 / P0
SE <- ( ( b / (a*(a+b)) ) + ( d / (c*(c+d)) ) ) ^ (0.5)

cat(sprintf("RR(male) = %.3f, 95%% confidence interval: [%.3f, %.3f]",
            RR, RR*exp(-qnorm(0.975) * SE), RR*exp(+qnorm(0.975) * SE)))

## RR(male) = 1.051, 95% confidence interval: [0.973, 1.136]
```

## Exercise 24

```
a <- 15
b <- 10
c <- 5
d <- 10

n <- a + b + c + d
```

### a) Measure of agreement

We calculate Cohen's Kappa.

```
p_e <- (a+b)/n * (a+c)/n + (c+d)/n * (b+d)/n
p_a <- (a+d)/n
kappa <- (p_a-p_e) / (1-p_e)
cat(sprintf("p_e: %.3f, p_a: %.3f, kappa: %.3f", p_e, p_a, kappa))

## p_e: 0.500, p_a: 0.625, kappa: 0.250
```

This means we have “fair agreement”.

## b) Statistically relevant difference

We use the exact McNemar test.

```
r <- min(b,c)
p <- 2 * sum(choose(b+c, 0:r) * 0.5^(b+c))

cat(sprintf("p-value: %.3f", p))
```

```
## p-value: 0.302
```

We can see that there is no significant difference.

## Exercise 25

```
X <- c(10,8,3,4,1,5,9,2,6,6)
Y <- c(7,9,5,3,2,6,10,1,4,8)

n <- length(X)
```

### a) Bravais-Pearson correlation coefficient

```
r <- cov(X,Y) / (sd(X) * sd(Y))
cat(sprintf("r: %.3f", r))
```

```
## r: 0.835
```

### Test to verify statistical significance

#### b) Test for statistical significance

Since we only want to know whether there is a significant correlation, we can use the “Exact Test for No Linear Correlation” test.

```
test_stat <- r * sqrt((n-2)/(1-r^2))
cat(sprintf("test statistic: %.3f, significant value: %.3f, ",
            test_stat, qt(0.975, n-2)))
```

```
## test statistic: 4.291, significant value: 2.306,
```

We can see that there is a significant difference from 0.

## Exercise 26

We use the “Tests of Correlation Coefficients I” test. For this, we first perform a Fisher z-transformation.

```
n <- 100
r1 <- 0.62
r2 <- 0.36

z1 <- 0.5 * log((1+r1) / (1-r1))
z2 <- 0.5 * log((1+r2) / (1-r2))

test_stat <- (z1 - z2) / sqrt(2/(n-3))
```

```
cat(sprintf("z1: %.3f, z2: %.3f, test statistic: %.3f, critical value: %.3f",
           z1, z2, test_stat, qnorm(0.975)))
```

```
## z1: 0.725, z2: 0.377, test statistic: 2.424, critical value: 1.960
```

This means that we can reject the null hypothesis.

## Exercise 27

```
X <- c(10, 8, 3, 4, 1, 5, 9, 2, 6, 6)
Y <- c(7, 9, 5, 3, 2, 6, 10, 1, 4, 8)
```

### Spearman-rank correlation coefficient

```
X_rank <- c(10, 8, 3, 4, 1, 5, 9, 2, 6.5, 6.5)
Y_rank <- c(7, 9, 5, 3, 2, 6, 10, 1, 4, 8)

X_rank_sd <- sd(X_rank)
Y_rank_sd <- sd(Y_rank)

XY_rank_cov <- cov(X_rank, Y_rank)

rho_XY <- XY_rank_cov / (X_rank_sd * Y_rank_sd)

cat(sprintf("SD X rank: %.3f; SD Y rank: %.3f; COV X,Y rank: %.3f; Rho X,Y: %.3f",
           X_rank_sd, Y_rank_sd, XY_rank_cov, rho_XY))
```

```
## SD X rank: 3.018; SD Y rank: 3.028; COV X,Y rank: 7.611; Rho X,Y: 0.833
```

### Test if the relationship is statistically significant

We perform the “Exact Test for No Linear Correlation”, which just tests the correlation coefficient against 0.

```
n <- length(X)

test_statistic <- rho_XY * sqrt( (n-2) / (1 - rho_XY^2) )
critical_value <- qt(0.975, n-2)

cat(sprintf("test statistic: %.3f; critical value: %.3f"
           , test_statistic, critical_value), "\n",
    sprintf("We can see that we %s reject the null hypothesis.",
            if (abs(test_statistic) > critical_value) "can" else "cannot"))
```

```
## test statistic: 4.256; critical value: 2.306
```

```
## We can see that we can reject the null hypothesis.
```

## Exercise 28

In this examples we have correlated overlapping correlations coefficients, as such we use “Tests of Correlation Coefficients II”.

```
r12 <- -0.205 # family social support - stress
r13 <- 0.285 # loneliness - stress
r23 <- -0.495 # family social support - loneliness
```

```

n <- 405

R_abs <- abs( 1 - r12^2 - r13^2 - r23^2 + 2 * r12 * r13 * r23 )
r_bar <- (r12 + r13) / 2
test_statistic <- (r12-r13) *
  sqrt( ((n-1) * (1+r23)) /
        ( 2 * ((n-1)/(n-3)) * R_abs + r_bar^2 * (1-r23)^3 ) )
critical_value <- qt(0.975, n-3)

cat(sprintf("test statistic: %.3f; critical value: %.3f"
            , test_statistic, critical_value), "\n",
    sprintf("We can see that we %s reject the null hypothesis.",
            if (abs(test_statistic) > critical_value) "can" else "cannot"))

## test statistic: -5.934; critical value: 1.966
## We can see that we can reject the null hypothesis.

```

## Exercise 29

```

A <- c(2, 3, 5, 2)
B <- c(1, 4, 5, 3)

```

### Kendall's Tau-b

```

A_sort <- c(2, 2, 3, 5)
B_sort <- c(1, 3, 4, 5)
n <- length(A)

# 13 14 15, 34 35, 45
# + + + + + +
# 31 34 35, 14 15, 45
# - + + + + +

nc <- 5
nd <- 0

T <- 2 * (2-1) / 2
W <- 0

tau_b <- (nc - nd) /
  ( sqrt(0.5 * n * (n-1) - T) * sqrt(0.5 * n * (n-1) - W) )

cat(sprintf("Kendall Tau-b: %.3f", tau_b))

## Kendall Tau-b: 0.913

```

### Test if correlation is significant

```

v0 <- n * (n-1) * (2*n+5)
vt <- 2 * (2-1) * (2*2+5)
vw <- 0
v1 <- 0

```

```

v2 <- 0
v <- (v0 - vt - vw) / (18 + v1 + v2)

test_statistic <- (nc - nd) / sqrt(v)
critical_value <- qnorm(0.975)

cat(sprintf("test statistic: %.3f; critical value: %.3f"
            , test_statistic, critical_value), "\n",
      sprintf("We can see that we %s reject the null hypothesis.",
              if (abs(test_statistic) > critical_value) "can" else "cannot"))

## test statistic: 1.806; critical value: 1.960
## We can see that we cannot reject the null hypothesis.

```

## Exercise 30

### Partial correlation

```

n_crimes <- c(210, 234, 279, 179, 160, 136)
n_police <- c(20, 21, 23, 19, 18, 15)
population <- c(30000, 40000, 50000, 20000, 10000, 5000)

r_xy <- cor(n_crimes, n_police)
r_xu <- cor(n_crimes, population)
r_yu <- cor(n_police, population)

r_xy.u <- (r_xy - r_xu * r_yu) / sqrt((1 - r_xu^2) * (1 - r_yu^2))
cat(sprintf("r_xy: %.3f, r_xu: %.3f, r_yu: %.3f, r_xy.u: %.3f",
            r_xy, r_xu, r_yu, r_xy.u))

## r_xy: 0.968, r_xu: 0.993, r_yu: 0.958, r_xy.u: 0.490

```

### Statistical test

```

n <- length(n_crimes)
test_statistic <- r_xy.u * sqrt((n - 3) / (1 - r_xy.u^2))
critical_value <- qt(0.975, n-3)

cat(sprintf("test statistic: %.3f; critical value: %.3f"
            , test_statistic, critical_value), "\n",
      sprintf("We can see that we %s reject the null hypothesis.",
              if (abs(test_statistic) > critical_value) "can" else "cannot"))

## test statistic: 0.974; critical value: 3.182
## We can see that we cannot reject the null hypothesis.

```

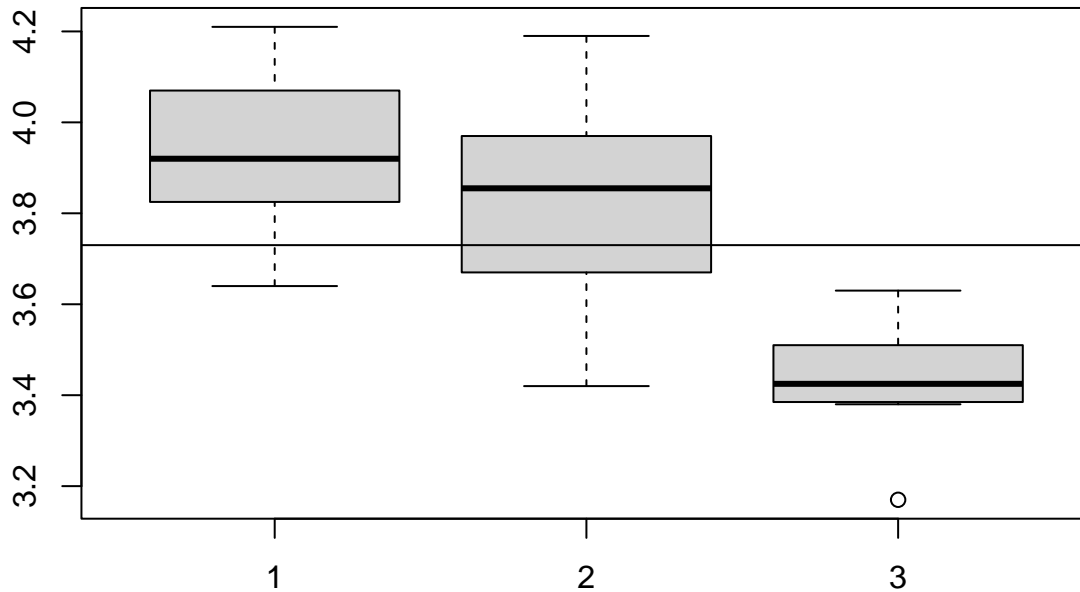
## Exercise 31

```

A <- c(3.64, 3.77, 4.18, 4.21, 3.88, 3.93, 3.91, 3.96)
B <- c(3.42, 3.96, 3.87, 4.19, 3.58, 3.76, 3.84, 3.98)
C <- c(3.17, 3.63, 3.38, 3.47, 3.39, 3.41, 3.55, 3.44)

```

```
boxplot(A, B, C)
abline(h=mean(c(A, B, C)))
```



## Formulation of a statistical hypothesis

Since we are comparing three different groups, we choose the following:

- $H_0: \mu_A = \mu_B = \mu_C$
- $H_1: \mu_A \neq \mu_B \text{ or } \mu_A \neq \mu_C \text{ or } \mu_B \neq \mu_C$

## Statistical test

The obvious choice would be an analysis of variance (ANOVA). But first, we check the assumptions.

### Normality

```
library(nortest)
lillie.test(A)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  A
## D = 0.19781, p-value = 0.4714

lillie.test(B)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  B
## D = 0.14983, p-value = 0.861

lillie.test(C)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
```



```
##
## data: C
## D = 0.23089, p-value = 0.238
```

As we can see, the data is normally distributed.

## Homoscedasticity

For this we perform a Levene test.

```
library(car)

dv <- c(A, B, C)
iv <- as.factor(c(rep("A", length(A)), rep("B", length(B)), rep("C", length(C))))

leveneTest(dv~iv, center="mean")

## Levene's Test for Homogeneity of Variance (center = "mean")
##      Df F value Pr(>F)
## group  2  0.9894 0.3885
##      21
```

As we can see, the data can be assumed to be homoscedastic. Thus, we can proceed with ANOVA.

## ANOVA

```
m_grand <- mean(c(A, B, C))

m_A <- mean(A)
l_A <- length(A)

m_B <- mean(B)
l_B <- length(B)

m_C <- mean(C)
l_C <- length(C)

SS_between <- l_A*(m_A-m_grand)^2 + l_B*(m_B-m_grand)^2 + l_C*(m_C-m_grand)^2
SS_within  <- sum((A - m_A)^2 + (B - m_B)^2 + (C - m_C)^2)
SS_total   <- SS_between + SS_within

cat(sprintf("SS_between: %.3f, SS_within: %.3f, SS_total: %.3f",
            SS_between, SS_within, SS_total), "\n")

## SS_between: 1.128, SS_within: 0.788, SS_total: 1.917

k <- 3
N <- length(c(A, B, C))

MS_between <- SS_between / (k-1)
MS_within  <- SS_within / (N-k)
MS_total   <- SS_total / (N-1)

cat(sprintf("MS_between: %.3f, MS_within: %.3f, MS_total: %.3f",
            MS_between, MS_within, MS_total), "\n")

## MS_between: 0.564, MS_within: 0.038, MS_total: 0.083
```

```
test_statistic <- MS_between / MS_within
critical_value <- qf(0.95, k-1, N-k)

cat(sprintf("test statistic: %.3f; critical value: %.3f"
            , test_statistic, critical_value), "\n",
    sprintf("We can see that we %s reject the null hypothesis.",
            if (abs(test_statistic) > critical_value) "can" else "cannot"))

## test statistic: 15.028; critical value: 3.467
## We can see that we can reject the null hypothesis.
```

## Verifying

```
model <- aov(dv~iv)
summary(model)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## iv              2  1.1284   0.5642    15.03 8.89e-05 ***
## Residuals      21  0.7884   0.0375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Exercise 32

```
A <- c(80, 77, 71, 72, 75)
B <- c(84, 74, 81, 80, 79)
C <- c(76, 76, 74, 73, 74)

n <- 5
k <- 3
```

## Hypothesis

Since we want to know if there is a global difference over time, we formulate the hypothesis as such:

- $H_0: \mu_A = \mu_B = \mu_C$
- $H_1: \mu_A \neq \mu_B$  or  $\mu_A \neq \mu_C$  or  $\mu_B \neq \mu_C$

## Statistical test

We perform a one-way repeated measures ANOVA (rANOVA). For this, we first check the assumption of sphericity (normality can already be assumed).

### Sphericity

To test for sphericity via a Mauchly's test.

```
# Building an appropriate Data-Frame for Mauchly's Test
patid <- as.factor(c(seq(1:5), seq(1:5), seq(1:5)))
time  <- c(rep("A", length(A)), rep("B", length(B)), rep("C", length(C)))
bpm   <- c(A, B, C)
df    <- data.frame(patid, time, bpm)
View(df)
```

```

# Mauchly's Test of Sphericity
bpm_matrix <- matrix(bpm, nrow=5, ncol=3)
mauchly.test(lm(bpm_matrix~1), X=~ 1)

##
## Mauchly's test of sphericity
## Contrasts orthogonal to
## ~1
##
## data: SSD matrix from lm(formula = bpm_matrix ~ 1)
## W = 0.57903, p-value = 0.4406

```

As we can see, we can assume sphericity.

## rANOVA

```

m_grand <- mean(c(A, B, C))

# group means
m_A <- mean(A)
m_B <- mean(B)
m_C <- mean(C)

# patient means
m <- numeric(5)
for(i in 1:n) {
  m[i] <- mean(c(A[i], B[i], C[i]))
}

cat(sprintf("m_A: %.3f, m_B: %.3f, m_C: %.3f", m_A, m_B, m_C), "\n",
  "m:", sprintf("%.3f", m), "\n\n")

## m_A: 75.000, m_B: 79.600, m_C: 74.600
## m: 80.000 75.667 75.333 75.000 76.000

SS_cond <- n * sum((m_A-m_grand)^2 + (m_B-m_grand)^2 + (m_C-m_grand)^2)
SS_object <- k * sum((m-m_grand)^2)
SS_within <- sum((A-m_A)^2 + (B-m_B)^2 + (C-m_C)^2)
SS_error <- SS_within - SS_object

cat(sprintf("SS_cond: %.3f, SS_object: %.3f, SS_within: %.3f, SS_error: %.3f",
  SS_cond, SS_object, SS_within, SS_error), "\n")

## SS_cond: 77.200, SS_object: 50.267, SS_within: 114.400, SS_error: 64.133

MS_cond <- SS_cond / (k-1)
MS_object <- SS_object / (n-1)
MS_within <- SS_within / (k*(n-1))
MS_error <- SS_error / ((n-1)*(k-1))

cat(sprintf("MS_cond: %.3f, MS_object: %.3f, MS_within: %.3f, MS_error: %.3f",
  MS_cond, MS_object, MS_within, MS_error), "\n\n")

## MS_cond: 38.600, MS_object: 12.567, MS_within: 9.533, MS_error: 8.017

```

```

test_statistic <- MS_cond / MS_error
critical_value <- qf(0.95, k-1, (n-1)*(k-1))

cat(sprintf("test statistic: %.3f; critical value: %.3f" ,
            test_statistic, critical_value), "\n",
    sprintf("We can see that we %s reject the null hypothesis.",
            if (abs(test_statistic) > critical_value) "can" else "cannot"))

## test statistic: 4.815; critical value: 4.459
## We can see that we can reject the null hypothesis.

```