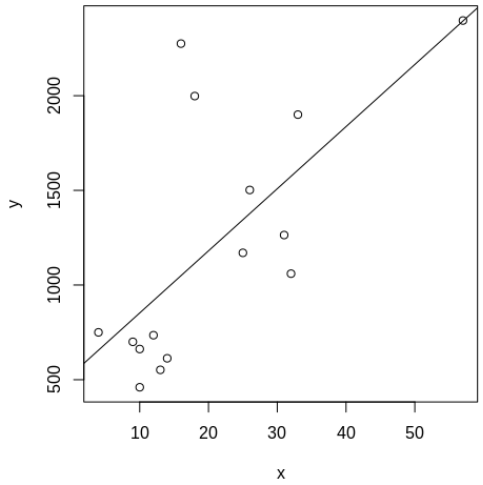
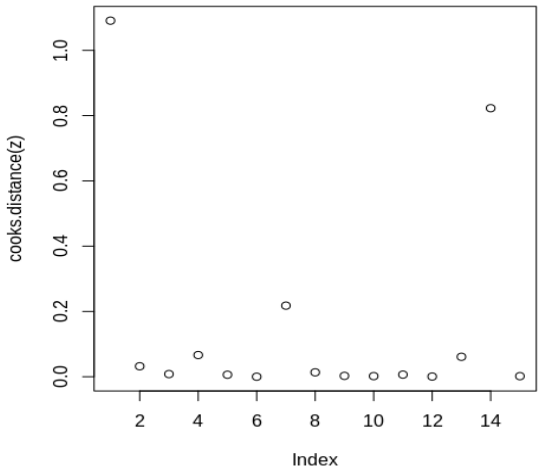
# Task 1
## Cook‘s distance

I didn‘t found the Andes data in the internet. Therefore, I took the data from the lecture which was on a slide after the Andes data was mentioned on the previous slide. I guess that should be the right data then.

```
# Our 2D data:
x<-c(57,31,25,32,33,10,18,9,12,14,10,26,4,16,13)
y<-c(2397,1264,1170,1060,1900,460,1998,700,735,613,662,1502,750,2275,552)

# Question: Which points are influential?
lin_model <-lm(y~x)

plot(x,y)
abline(lin_model)
```

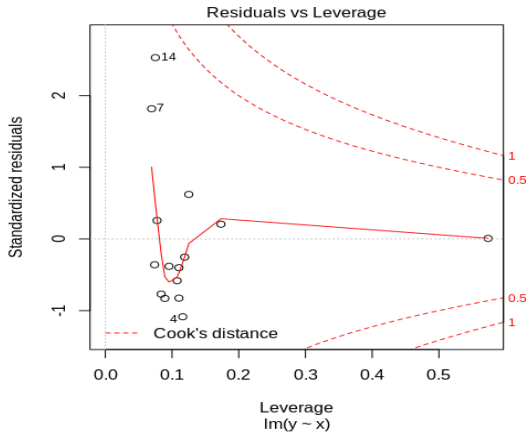



```
# point 1 (57|2397) could be influential because it is far away on the x-axis from the other points, it‘s isolated and (as sidemark) has a high y-value. However, it seems to lie more or less on the regression line by eye.

plot(cooks.distance(lin_model))
```




# To see that point 1 and 14 have both reached a high value for Cook‘s distance. For point it has a
# high potential to be influential because Cook‘s distance > 1. However, also point 1 is a candidate
# (Cook‘s distance > 0.5).

# However, we also have to consider the leverage:
plot(lin_model)




# It shows by considering the leverage no datapoint reached Cook‘s distance.

# Create linear models without point 1 and 14:
# for point 1:
```
x_copy <- sapply(x, function(i) i)  # copy of x
y_copy <- sapply(y, function(i) i)
```

x_copy <-x_copy[-c(1)] # Remove element with index 1 (point 1)

y_copy <-y_copy[-c(1)]

```
lin_model_c <- lm(y_copy~x_copy)
summary(lin_model_c)
```

```
Call:
lm(formula = y_copy ~ x_copy)

Residuals:
   Min     1Q Median     3Q    Max
-513.1 -347.3 -178.7  117.3 1225.6

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   525.75     308.46   1.704   0.1140
x_copy         32.73      15.21   2.152   0.0524 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 524.3 on 12 degrees of freedom
Multiple R-squared:  0.2785,    Adjusted R-squared:  0.2184
F-statistic: 4.633 on 1 and 12 DF,  p-value: 0.05242
```

```
summary(lin_model)
```

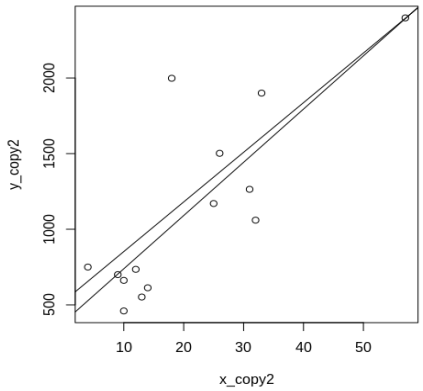
```
> summary(lin_model)

Call:
lm(formula = y ~ x)

Residuals:
   Min     1Q Median     3Q    Max
-514.4 -324.2 -174.7  109.4 1225.6

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  524.489    241.974   2.168   0.0493 *
x             32.809      9.873   3.323   0.0055 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 503.8 on 13 degrees of freedom
Multiple R-squared:  0.4593,    Adjusted R-squared:  0.4177
F-statistic: 11.04 on 1 and 13 DF,  p-value: 0.005498
```

```
# overview of our regression model without any changes in data. As basis of comparison with
# models where single points were deleted.
# In comparison with summary of lin_model_c where point 1 was deleted:
# Only very small change in slope b and intercept a. Point 1 doesn‘t seem to be an influental point.

# same for point 14:
x_copy2 <- sapply(x, function(i) i)  # copy of x
y_copy2 <- sapply(y, function(i) i)


x_copy2 <-x_copy2[-c(14)] # Remove element with index 14 (point 14)

y_copy2 <-y_copy2[-c(14)]

lin_model_c2 <- lm(y_copy2~x_copy2)
summary(lin_model_c2)
```

```
Call:
lm(formula = y_copy2 ~ x_copy2)

Residuals:
     Min       1Q   Median       3Q      Max
-452.95 -253.42  -75.59  151.23  977.62

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  387.070    183.931   2.104 0.057094 .
x_copy2       35.184      7.356   4.783 0.000446 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 373.6 on 12 degrees of freedom
Multiple R-squared:  0.656,     Adjusted R-squared:  0.6273
F-statistic: 22.88 on 1 and 12 DF,  p-value: 0.0004461
```

```
# In comparison to the lin_model, the lin_model2 explains with an R-squared value > 60 the
# underlying data much more better. The intercept a shrinked daramtically. The slope b circa
# increased by 3.
```

```
plot(x_copy2,y_copy2);abline(lin_model);abline(lin_model_c2)
```



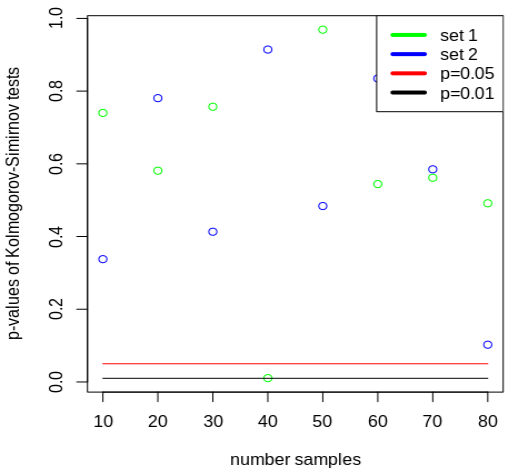

```
# different models but no huge difference. It could be by having some more data points that the
# impact of point 14 vanishes.
```

(Task 2 next page)

# Task 2
## Kolmogorov-Smirnov tests

```
set.seed(1)

# Create the samples:

user_input1_vec<-c()
user_input2_vec<-c()
user_input1<-"not stop"
while(user_input1 != "stop")
        {
        user_input1<-readline(prompt = "Type in size sample1 for [1,8] or stop:")

        user_input2<-readline(prompt = "Type in size sample2 for [1,8] :")

        if(user_input1=="stop"){break}

        user_input1<-as.numeric(user_input1)
        user_input2<-as.numeric(user_input2)

        user_input1_vec<-append(user_input1_vec, user_input1)
        user_input2_vec<-append(user_input2_vec, user_input2)
        }

# Find the p-values:

plot_p1<-c()
plot_p2<-c()
plot_p3<-c()
index_counter<-1 # in R it starts by 1

for(i in user_input1_vec)
        {
        sample1 <-rnorm(i*10, mean=0, sd=1)
        sample2 <-rnorm(user_input2_vec[index_counter]*10, mean=0, sd=1)

        k1<-ks.test(sample1,"pnorm") # test if samples come from normal distribution
        k2<-ks.test(sample2,"pnorm")

        k3<-ks.test(sample1,sample2) # look if both samples come from the same distribution

        p1<-k1$p.value
        p2<-k2$p.value
        p3<-k3$p.value

        plot_p1<-append(plot_p1, p1)
        plot_p2<-append(plot_p2, p2)
        plot_p3<-append(plot_p3, p3)
```

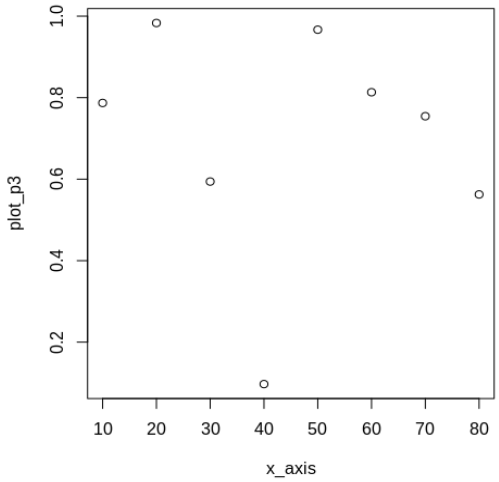
```
        index_counter<-index_counter+1
        }

# Plot the p-values for the two different subsets:

x_axis<-seq(10,(length(plot_p1)*10), by=10)

plot(x_axis, plot_p1, col="green",xlab="number samples",ylab="p-values of Kolmogorov-Simirnov
tests")

lines(x_axis, plot_p2, col="blue",type="p") # add plot_p2 into the graphic, we don‘t want to add a
# line, therefore: type="p"=points

p_line<-rep(0.05, length(plot_p1)) # creating a line for significane level 0.05
lines(x_axis,p_line,col="red")

p_line2<-rep(0.01, length(plot_p1)) # creating a line for high significane level 0.01
lines(x_axis,p_line2)

legend("topright",legend=c("set 1","set
2","p=0.05","p=0.01"),lwd=4,col=c("green","blue","red","black")) # lwd: how thick the lines are
```




```
# In alomost all cases the p-value > 0.05. When this happens we can NOT reject the null-hypothesis
# which says that there is NO difference between a subsample set and a normal distribution. So it‘s
# possible that our sample sets originates from the normal distribution. In this case we know that
# because we created them.

# Furthermore, only for number_samples = 40 set 1 we could reject the nullhypothesis.
# However, randomly drawn samples can be with bad luck also unrepresentative for the underlying
# distribtuion and so below the significance level of 0.05.
# You can see a trend that the more samples we have in our sets the higher the p-values.
```

# Different p-values for sets of same size and same distribution could be a result of randomness.
# Therefore, the two sets also differentiate in p-values. However, the representativity differences of
# the samples could also lead to different p-values.

plot(x_axis, plot_p3)




# Here we get the p-values for the question if set1 comes from the same distribution as set2.
# Only for the case of 40 samples per set we reject the nullhypothesis. That means set1 seems to
# originate from an other distribution than set2 for different sample sizes. However, unlucky
# random sampling could be responsible for this observation because in that case we already know
# that both sets come from the same distribution.

(Task 3 next page)

# Task 3
## Article: Computing the Two-Sided Kolmogorov-Sminov Distribution

n = number of observations
x = border to which the cumulative distribution function G (cdf) shall be computed

Measure the difference between estimated and real cdf G-s:
$D_n = sup_x |G_n - \bar{G}_n|$ (1)

Get the formula for the Komogorov-Sminov Distribution with the given null hypothesis H_0 that the n observations are different:
$F_n(x) = P[D_n \leqslant x | H_0]$ (2)

One more formula with exact values that counts for special cases:
Given: $1 - \frac{1}{n} \leqslant x < 1$ → Here are x which are very near to 1 (in our interest)
Therefore: $F_n(x) = 1 - 2(1-x)^n$ (3)

From (3) we get that when n are very large and x very close to 1, we observe the following:
1 – 2(very_small_value)^(very_large_value). When a very small value has a power to a very large number then it even becomes smaller. So, we have only very small changes. Therefore, it is more difficult to catch up the precision.

(1): Also the D_n from the first two formulas becomes very small when x is near 1 and when n is a large number (the more observations the better the estimation). That‘s the case because the estimation of cdf comes to the real one very close when x goes to 1. Both cdf-s converge to 1. Also here, we have to work with very small value changes.
This small value changes could get lost such that the final p-value won‘t be that precise anymore.

Furthermore, on page 7 of the paper it becomes clear that F_n for n converging to infinity needs to be approximated with a very long formula. Firstly, we only can approximate the p-values. Secondly, that can take a while. The program “Mathematica” is doing a good job but needs for n>10000 days to compute F_n, especially when x > ln(2) (phi/(2n))^0.5. So a circa rounding with less precision is faster but less precise.