Homework 5 by Nina Braunmiller 07.11.21 k11923286

Task 1

In [3]:
```
install.packages("languageserver")
```

```
Installing package into ‘/home/c/R/x86_64-pc-linux-gnu-library/3.6’
(as ‘lib’ is unspecified)

also installing the dependencies ‘sys’, ‘diffobj’, ‘askpass’, ‘ps’, ‘lazyeval’, ‘remotes’, ‘brio’, ‘praise’, ‘waldo’, ‘curl’, ‘mime’, ‘openssl’, ‘highr’, ‘yaml’, ‘R.methodsS3’, ‘R.oo’, ‘R.utils’, ‘processx’, ‘rex’, ‘cyclocomp’, ‘testthat’, ‘rstudioapi’, ‘httr’, ‘knitr’, ‘brew’, ‘commonmark’, ‘desc’, ‘pkgload’, ‘stringr’, ‘cpp11’, ‘backports’, ‘R.cache’, ‘rematch2’, ‘rprojroot’, ‘xfun’, ‘callr’, ‘collections’, ‘fs’, ‘lintr’, ‘roxygen2’, ‘stringi’, ‘styler’, ‘xml2’, ‘xmlparsedata’


Warning message in install.packages("languageserver"):
“installation of package ‘curl’ had non-zero exit status”
```

In [99]:
```
install.packages('geometry')
library(geometry)
```

In [102]:
```
sekeleton_no = TRUE

for(x in 10:15)
    {
    skeleton = c(1,x,x^2,x^3,x^4,x^5)
    if(sekeleton_no == FALSE)
        {x_matrix <- rbind(x_matrix, skeleton)}
    else
        {
        x_matrix <- skeleton
        sekeleton_no = FALSE
        }
    }
```

In [104]:
```
f_matrix = matrix(c(25,16,26,19,21,20),nrow=6,ncol=1)
```

In [115]:
```
a_matrix <- solve(x_matrix,f_matrix)
a_matrix
```

A matrix: 6 × 1 of type dbl

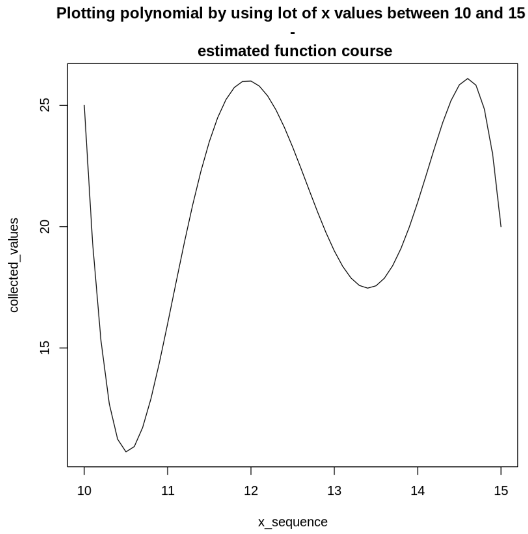
2.536100e+05
-1.025510e+05
1.650092e+04
-1.320667e+03
5.258333e+01
-8.333333e-01

So we have: f_matrix = x_matrix @ transpose(a_matrix);
@ as cross product

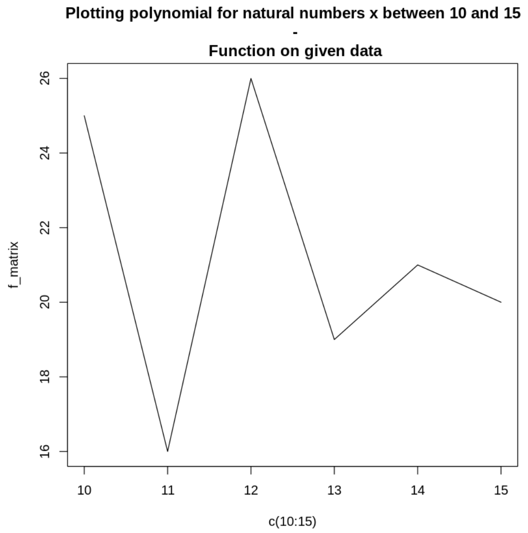
In [150]:
```r
collected_values <- c()

x_sequence <- seq(10,15,0.1)
for(xs in x_sequence)
    {
    inner_prod <- crossprod(c(1,xs,xs^2,xs^3,xs^4,xs^5),a_matrix)
    collected_values <- append(collected_values, inner_prod)

    }
plot(x_sequence,collected_values,type="l",main="Plotting polynomial by using lot of x values between 10 and 15
```



In [152]:
```r
plot(c(10:15),f_matrix, col="black",type="l",main="Plotting polynomial for natural numbers x between 10 and 15\
```



Task 2: Linear regression and weighting

Linear regression general: $y_i = bx_i + a + \epsilon$
here: $y_i = \alpha_i + \alpha_0 + \epsilon_0$
let be: b=1, $x_i = \alpha_i, a = alpha_0, \epsilon_0$ is unknown

$\hat{\alpha}_i = y_i - y_0 = y_i - \alpha_0 - \epsilon_0$

Use expected values because true errors are unkown
Let be: $E(y_0) = \alpha_0; E(y_i) = 1 * \alpha_i + \alpha_0$

In [161]:
```r
y <- c(0,1,2,3)
y <-(y - mean(y))/sd(y)
```

In [73]:
```
alpha_array <- c()
alpha_0_array <- c()
for(i in y)
    {
    alpha_array <- append(alpha_array, (i-y[1]))
    alpha_0_array <- append(alpha_0_array, y[1])
    }
alpha_array[1] <- y[1]
```

In [75]:
```
m <-lm(y[-1]~alpha_array[-1]+alpha_0_array[-1]) # to get slope=1, intercept=alpha_0
```

In [155]:
```
summary(m, digits = digits, maxsum = maxsum)
```

```
Warning message in summary.lm(m, digits = digits, maxsum = maxsum):
“essentially perfect fit: summary may be unreliable”


Call:
lm(formula = y[-1] ~ alpha_array[-1] + alpha_0_array[-1])

Residuals:
         1          2          3 
-1.813e-16  3.626e-16 -1.813e-16 

Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error    t value Pr(>|t|)    
(Intercept)       -1.162e+00  6.784e-16 -1.713e+15 3.72e-16 ***
alpha_array[-1]    1.000e+00  4.054e-16  2.467e+15 2.58e-16 ***
alpha_0_array[-1]         NA         NA         NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.441e-16 on 1 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1 
F-statistic: 6.085e+30 on 1 and 1 DF,  p-value: 2.581e-16
```

*

Here you can see:b=1, a=$alpha_0$

In [96]:
```
estimated_residuals_var <- sd(m$residuals)**2
variance_alpha <- vcov(m)[2,2]
print(paste('Variance of estimated residuals times 2: ',estimated_residuals_var*2 ,'vs. Variance alpha: ',varia
```

```
Warning message in summary.lm(object, ...):
“essentially perfect fit: summary may be unreliable”

[1] "Variance of estimated residuals times 2:  1.97215226305253e-31 vs. Variance alpha:  1.64346021921044e-31"
```

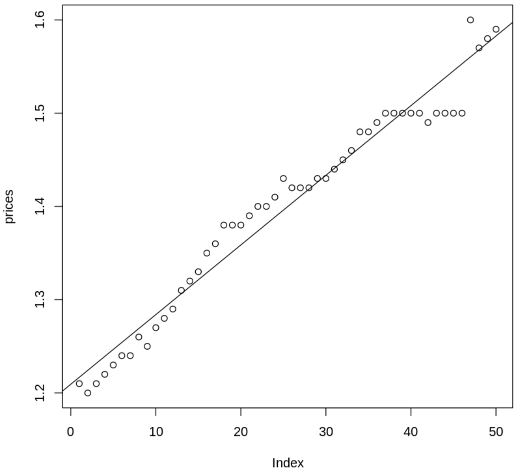
They are almost the same. Because the residuals are only estimated let's assume: variance alpha = Variance of residuals times 2

Task 3: Prices for Super Plus

data retrieved from: https://de.statista.com/statistik/daten/studie/796570/umfrage/durchschnittlicher-preis-fuer-einen-liter-benzin-in-oesterreich/ (https://de.statista.com/statistik/daten/studie/796570/umfrage/durchschnittlicher-preis-fuer-einen-liter-benzin-in-oesterreich/)

In [156]:
```
prices <- c(1.21,1.2,1.21,1.22,1.23,1.24,1.24,1.26,1.25,1.27,1.28,1.29,1.31,1.32,1.33,1.35,1.36,1.38,1.38,1.38,
time <- c(1:length(prices))
Y <- lm(prices~time)
```

In [157]:
```
plot(prices)
abline(Y)
```



In [158]:
```
variance_b <- (summary(Y)$coefficients[2,2])**2 # or: vcov(Y)[2,2]
variance_b
```

4.40515606002304e-08

Calculation: variance_b-est = (Standard_error_of_b-est)**2

In [160]:
```
variance_a <- (summary(Y)$coefficients[1,2])**2 # or: vcov(Y)[1,1]
variance_a
```

3.78182647752978e-05

Calculation: variance_a-est = (Standard_error_of_a-est)**2

In [ ]:
```

```

→ (*) Task 2

Example 2

→ Take given $y_1 \ldots$ or $\alpha_1, \ldots$

→ Weights of 3 wonges

→ library (matrixcalc)

(1)
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

Don't wish it but this systematic error unavoidable

↓ Dc of α_0

$$y = X\alpha + \varepsilon$$ Not measurable

OLS → Orinal Least squares estimate : $X^T y = \underbrace{X^T X}_{} \hat{\alpha}$ errors not visible

~~Rank Matrix~~

det(X) = 1 → Inversion possible

$$(X^T X)^{-1} X^T y = \hat{\alpha}$$

Unknown $V[\varepsilon] = \sigma^2$

$$V[\hat{\alpha}] = V[(X^T X)^{-1} X^T y] = (X^T X)^{-1} \cancel{X^T X (X^T X)^{-1}} V(y)$$

$$= \sigma^2 (X^T X)^{-1}$$

$$(X^T X)^{-1} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 2 & 1 & 1 \\ -1 & 1 & 2 & 1 \\ -1 & 1 & 1 & 2 \end{bmatrix} \qquad V(y) = \sigma^2 I$$

$$\Rightarrow V[\hat{\alpha}] = 2\sigma^2$$

$$V[\hat{\alpha}_1] = V[y_1 - y_0] = V[y_1] + V[y_0] - \underbrace{2\operatorname{cov}(y_1, y_0)}_{=0} = 2\sigma^2$$

$$\hat{\alpha} = (X^T X)^{-1} X^T y$$

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$(X^T X)^{-1} X^T X (X^T X)^{-1} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 2 & 1 & 1 \\ -1 & 1 & 2 & 1 \\ -1 & 1 & 1 & 2 \end{bmatrix}$$