Numerieke wiskunde TinLab casus

eindopdracht

In opdracht van:

Opleiding: Technische Informatica, Hogeschool Rotterdam

Uitgevoerd door: Galvin Bartes

Datum: 08-04-2020

# Doel

Het doel van de opdracht is een lineair regressie model te bouwen waarmee de doorlooptijd kan worden voorspeld.

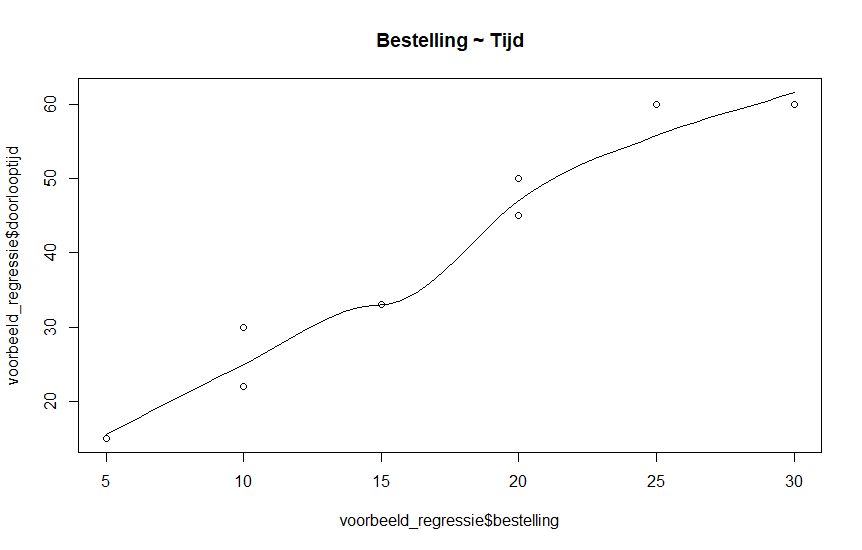
# Opdracht 1

## Voorbeeld 1

Een scatterplot van de dataset voor de response en de predict variabele.

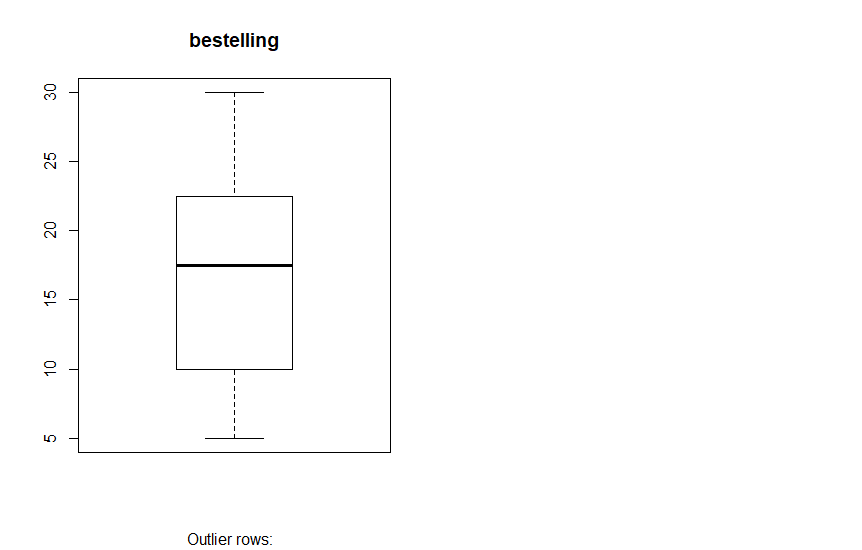
scatter.smooth(x=voorbeeld\_regressie$bestelling, y=voorbeeld\_regressie$doorlooptijd, main="Bestelling ~ Tijd") # scatterplot

geeft:



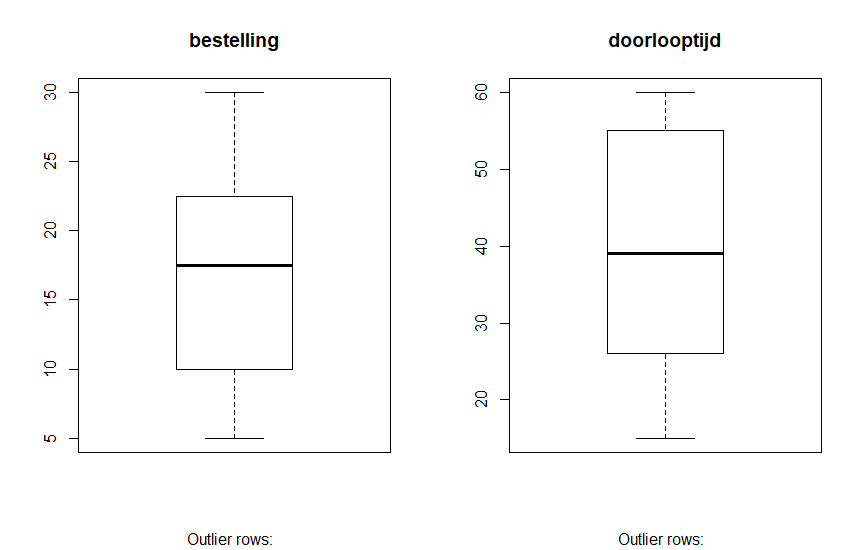
Met de boxplot kunnen we enkele uitschieters binnen de dataset visualiseren.

boxplot(voorbeeld\_regressie$bestelling, main="bestelling", sub=paste("Outlier rows: ", boxplot.stats(voorbeeld\_regressie$bestelling)$out)) # box plot for 'bestelling'

geeft:

boxplot(voorbeeld\_regressie$doorlooptijd, main="doorlooptijd", sub=paste("Outlier rows: ", boxplot.stats(voorbeeld\_regressie$doorlooptijd)$out)) # box plot for 'doorlooptijd'

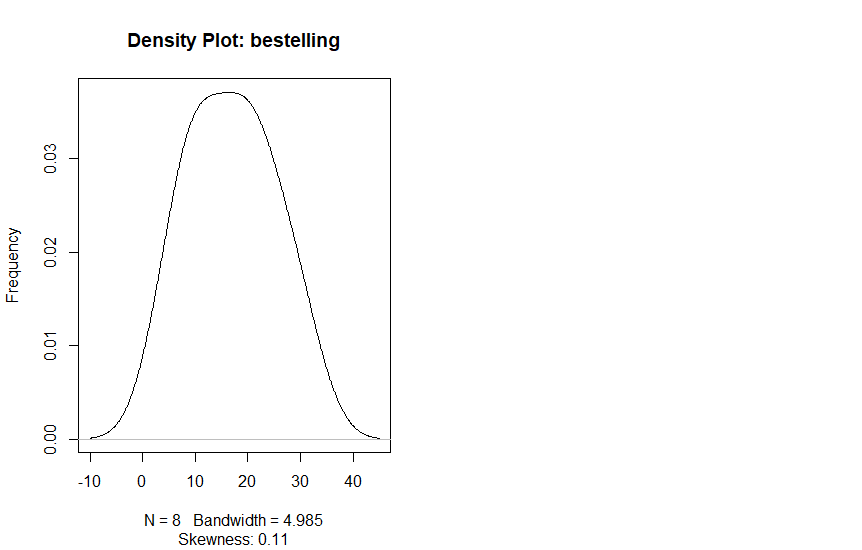
geeft:



De dichtheid is al volgt gevisualiseerd

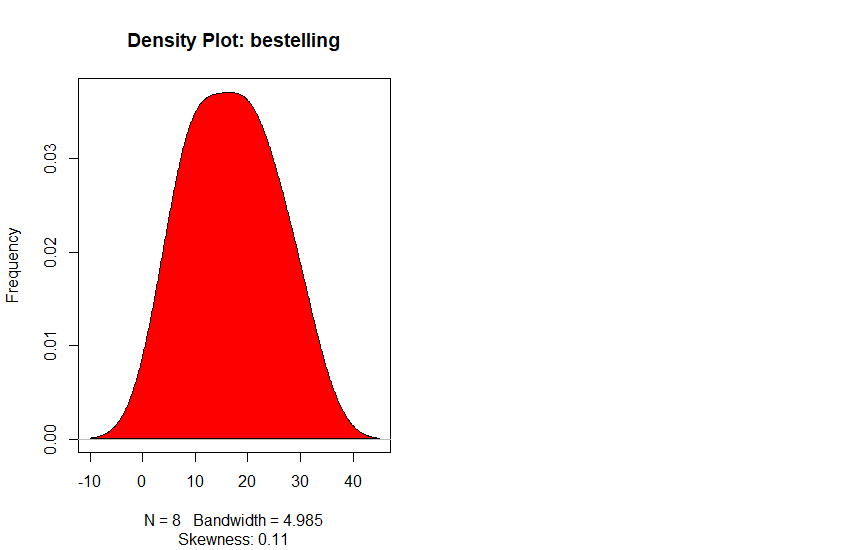
plot(density(voorbeeld\_regressie$bestelling), main="Density Plot: bestelling", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(voorbeeld\_regressie$bestelling), 2))) # density plot for 'bestelling'

geeft:



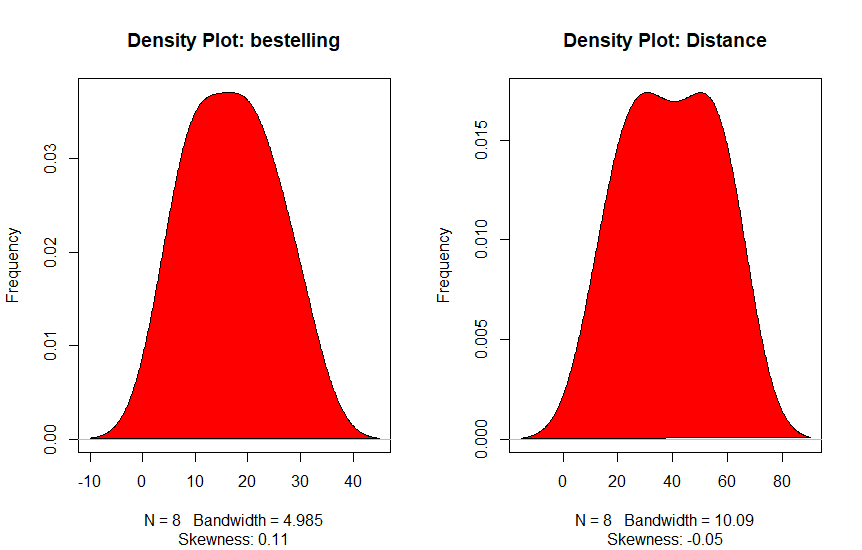
polygon(density(voorbeeld\_regressie$bestelling), col="red")

geeft



plot(density(voorbeeld\_regressie$doorlooptijd), main="Density Plot: Distance", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(voorbeeld\_regressie$doorlooptijd), 2))) # density plot for 'doorlooptijd'

geeft:



Met de coorelatieefficient bepalen we de de sterkte van een lineaire relatie tussen twee continu variabelen.

De correlatie kan waarden innemen tussen -1 en +1.

Als bijvoorbeeld een bepaalde variabele een stijging laat zien tegelijkertijd met een stijging van de andere variabele, dan is het aannemelijk dat er een sterke positieve correlatie is. Dat is een waarde dichtbij -1.

Een waarde dichtbij -1 betekent een sterke negatieve correlatie.

Een waarde dichtbij de 0 beteken dat er een zwakke correlatie is tussen de twee variabelen.

Een hoge correlatie betekent nog geen causaal verband. Correlatie is alleen bedoeld om de relatie tussen twee variabelen te duiden.

cor(voorbeeld\_regressie$bestelling, voorbeeld\_regressie$doorlooptijd) # calculate correlation between bestelling and tijd

geeft:

[1] 0.9726135

Dat is dus een sterke correlatie tussen bestelling en doorlooptijd

linearMod <- lm(doorlooptijd ~ bestelling, data=voorbeeld\_regressie) # build linear regression model on full data

print(linearMod)

geeft:

Call:

lm(formula = doorlooptijd ~ bestelling, data = voorbeeld\_regressie)

Coefficients:

(Intercept) bestelling

6.283 1.961

# capture model summary as an object

modelSummary <- summary(linearMod)

geeft:

Call:

lm(formula = doorlooptijd ~ bestelling, data = voorbeeld\_regressie)

Residuals:

Min 1Q Median 3Q Max

-5.1132 -2.9969 -0.7956 4.2044 4.6918

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.2830 3.5632 1.763 0.128

bestelling 1.9610 0.1913 10.250 5.03e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.265 on 6 degrees of freedom

Multiple R-squared: 0.946, Adjusted R-squared: 0.937

F-statistic: 105.1 on 1 and 6 DF, p-value: 5.03e-05

Een model is statistisch significant wanneer de p-value van het model en de p-value van de individueel voorspelde minder zijn dan de significantie-level van 0.05 gegeven een 95% betrouwbaarheid.

Indien er sprake is van een p-value dan is er ook sprake van een null- en alternatieve hypothese, respectievelijk 0 en 1.

Een grote t-waarde is een indicatie die aangeeft dat het minder aannemelijk is de coefficient alleen door toeval niet gelijk aan nul is.

Een p-value is de waarschijnlijkheid dat de t-waarde even hoog of hoger is dan wanneer de geobserveerde waarde ( coefficient is gelijk aan nul of dat er geen relatie is) bij een null hypothese waar is.

Dus wanneer de p-value l;einder is dan 0.05 ( significance level), is het toegestaan de null hypothese, de hypothese dat de co-efficient gelijk is aan nul, te verwerpen.

Dus we verwerpen de null hypothese en concluderen dat het model statistisch significant is.

# model coefficients

modelCoeffs <- modelSummary$coefficients

geeft:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.283019 3.5631859 1.763315 1.283054e-01

bestelling 1.961006 0.1913163 10.250074 5.030188e-05

# get beta estimate for bestelling

beta.estimate <- modelCoeffs["bestelling", "Estimate"]

geeft:

> beta.estimate

[1] 1.961006

# get std.error for bestelling

std.error <- modelCoeffs["bestelling", "Std. Error"]

geeft:

> std.error

[1] 0.1913163

# calc t statistic

t\_value <- beta.estimate/std.error

geeft:

> t\_value

[1] 10.25007

# calc p Value

p\_value <- 2\*pt(-abs(t\_value), df=nrow(voorbeeld\_regressie)-ncol(voorbeeld\_regressie))

geeft:

> p\_value

[1] 5.030188e-05

# fstatistic

f\_statistic <- linearMod$fstatistic[1]

geeft:

> f\_statistic

NULL

# parameters for model p-value calc

f <- summary(linearMod)$fstatistic

geeft

> f

value numdf dendf

105.064 1.000 6.000

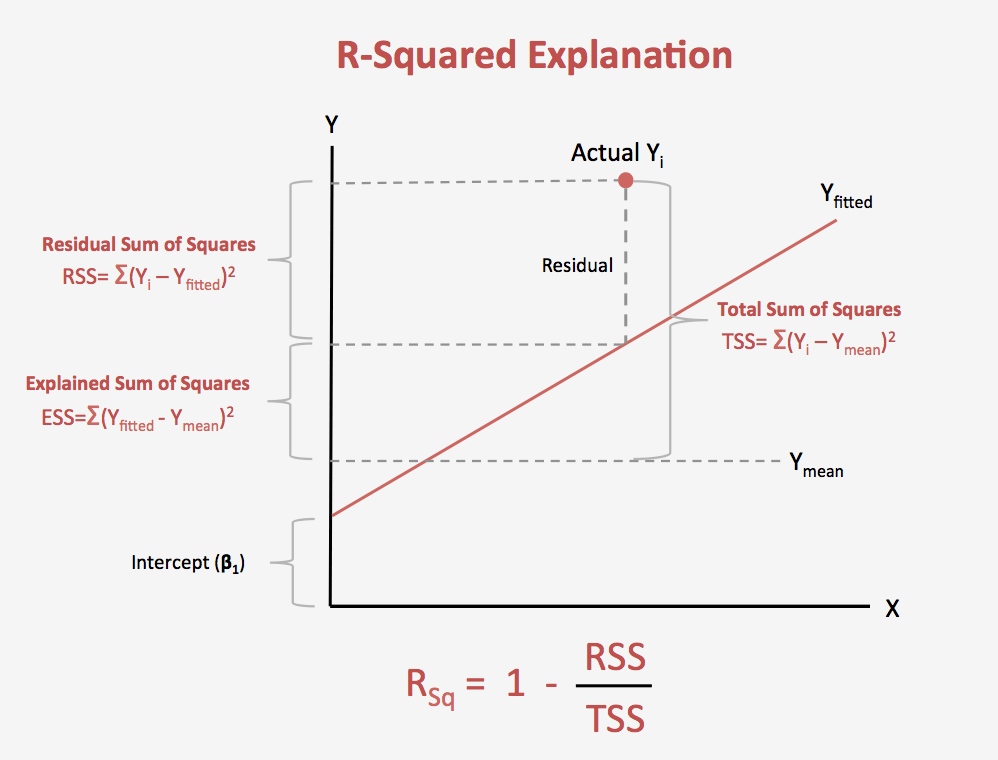
model\_p <- pf(f[1], f[2], f[3], lower=FALSE)

geeft:

> model\_p

value

5.030188e-05

Rsquared geeft aan in hoe verre de variatie van het afhankelijke variabele verklaard wordt door dit model. 

De totale informate die een variabele geeft is de hoeveelheid aan variantie de variabele in zich heeft.

Een rsquared van een groter model zal altijd groter zijn dan de rsquared van een kleiner model vanwege de grootte van de subset. Want als de variabele vertegenwoordigd is in de originele dataset dan is deze ook aanwezig in de subset. De aangepaste (adjusted) rsquared past het aantal termen van de voorspelde variabelen aan.

Zowel de “standard error” en de “f-statistic” duiden de “goodness of fit”. Of het model goed past bij de dataset. Hoe hoger de f-statistic hoe beter.

De AIC en de BIC zijn twee functies waarmee de “goodness of fit” kan worden berekend van een lineair model, and kan dus gebruikt worden om het juiste model te kiezen.

AIC(linearMod)

geeft:

> AIC(linearMod)

[1] 49.60703

BIC(linearMod)

geeft:

[1] 49.84536

## Voorbeeld van test en training set

Een test van de gegeven dataset door middel van een test en een training set.

Door het berekenen van de “accuracy measures” en “error rates” is het mogelijk de voorspelde accuratie te van het model vast te stellen.

# Create Training and Test data -

set.seed(100) # setting seed to reproduce results of random sampling

trainingRowIndex <- sample(1:nrow(voorbeeld\_regressie), 0.8\*nrow(voorbeeld\_regressie)) # row indices for training data

trainingData <- voorbeeld\_regressie[trainingRowIndex, ] # model training data

testData <- voorbeeld\_regressie[-trainingRowIndex, ] # test data

# Build the model on training data

lmMod <- lm(doorlooptijd ~ bestelling, data=trainingData) # build the model

doorlooptijdPred <- predict(lmMod, testData) # predict doorlooptijdance

summary (lmMod) # model summary

geeft:

Call:

lm(formula = doorlooptijd ~ bestelling, data = trainingData)

Residuals:

1 2 3 4 5 6

2.231 3.677 -0.600 -2.769 -4.046 1.508

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.8769 3.5121 1.389 0.237272

bestelling 2.1446 0.2042 10.503 0.000465 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.36 on 4 degrees of freedom

Multiple R-squared: 0.965, Adjusted R-squared: 0.9563

F-statistic: 110.3 on 1 and 4 DF, p-value: 0.0004646

Een hoge correlatie impliceert dat de werkelijke en de voorspelde waarden een vergelijkbare richting laten zien. Ofwel wanneer de werkelijke waarden toenemen dan nemen de verwachte waarden ook toe, en andersom is dit ook het geval.

actuals\_preds <- data.frame(cbind(actuals=testData$doorlooptijd, predicteds=doorlooptijdPred)) # make actuals\_predicteds dataframe.

correlation\_accuracy <- cor(actuals\_preds)

geeft:

actuals predicteds

actuals 1 1

predicteds 1 1

head(actuals\_preds)

geeft:

> head(actuals\_preds)

actuals predicteds

1 22 26.32308

2 60 69.21538

# Min-Max Accuracy Calculation

min\_max\_accuracy <- mean(apply(actuals\_preds, 1, min) / apply(actuals\_preds, 1, max))

geeft:

> min\_max\_accuracy

[1] 0.8513139

# MAPE Calculation

mape <- mean(abs((actuals\_preds$predicteds - actuals\_preds$actuals))/actuals\_preds$actuals)

geeft:

> mape

[1] 0.1750466

DMwR::regr.eval(actuals\_preds$actuals, actuals\_preds$predicteds)

geeft:

> DMwR::regr.eval(actuals\_preds$actuals, actuals\_preds$predicteds)

mae mse rmse mape

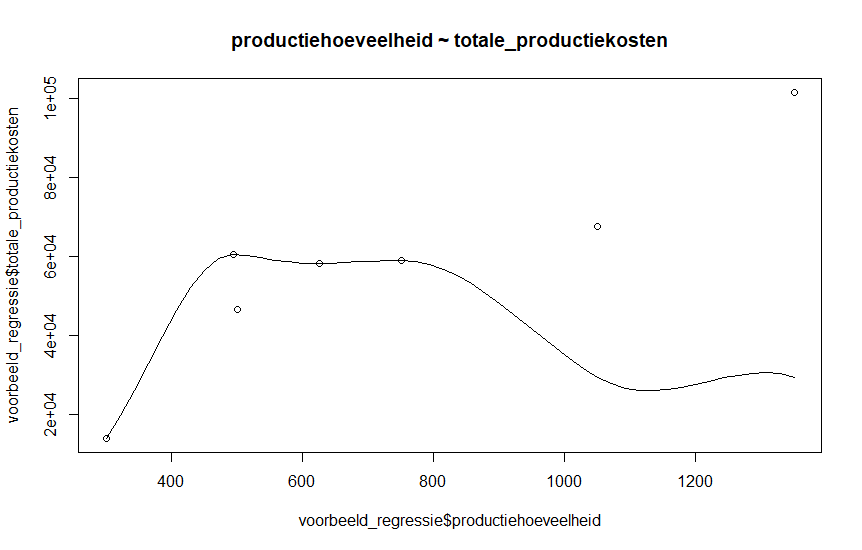
6.7692308 51.8061538 7.1976492 0.1750466

## Voorbeeld 2

Een regressie voor de voorspelling van de totale\_productiekosten

scatter.smooth(x=voorbeeld\_regressie$productiehoeveelheid, y=voorbeeld\_regressie$totale\_productiekosten, main="productiehoeveelheid ~ totale\_productiekosten") # scatterplot

Geeft:

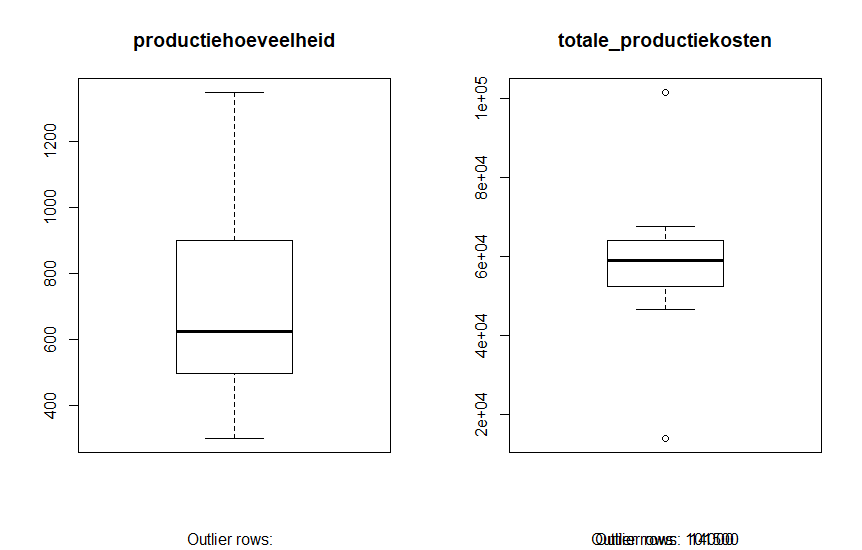


par(mfrow=c(1, 2)) # divide graph area in 2 columns

boxplot(voorbeeld\_regressie$productiehoeveelheid, main="productiehoeveelheid", sub=paste("Outlier rows: ", boxplot.stats(voorbeeld\_regressie$productiehoeveelheid)$out)) # box plot for 'productiehoeveelheid'

boxplot(voorbeeld\_regressie$totale\_productiekosten, main="totale\_productiekosten", sub=paste("Outlier rows: ", boxplot.stats(voorbeeld\_regressie$totale\_productiekosten)$out)) # box plot for 'totale\_productiekosten'

Geeft:



par(mfrow=c(1, 2)) # divide graph area in 2 columns

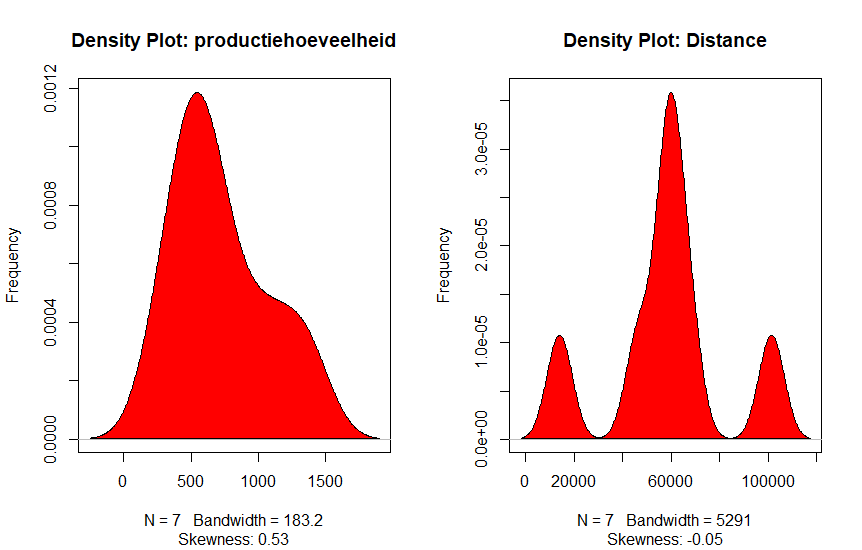
plot(density(voorbeeld\_regressie$productiehoeveelheid), main="Density Plot: productiehoeveelheid", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(voorbeeld\_regressie$productiehoeveelheid), 2))) # density plot for 'productiehoeveelheid'

polygon(density(voorbeeld\_regressie$productiehoeveelheid), col="red")

plot(density(voorbeeld\_regressie$totale\_productiekosten), main="Density Plot: Distance", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(voorbeeld\_regressie$totale\_productiekosten), 2))) # density plot for 'totale\_productiekosten'

polygon(density(voorbeeld\_regressie$totale\_productiekosten), col="red")

Geeft:



cor(voorbeeld\_regressie$productiehoeveelheid, voorbeeld\_regressie$totale\_productiekosten) # calculate correlation between productiehoeveelheid and tijd

Geeft:

[1] 0.8999736

linearMod <- lm(totale\_productiekosten ~ productiehoeveelheid, data=voorbeeld\_regressie) # build linear regression model on full data

print(linearMod)

Geeft:

Call:

lm(formula = totale\_productiekosten ~ productiehoeveelheid, data = voorbeeld\_regressie)

Coefficients:

(Intercept) productiehoeveelheid

11491.89 64.46

# capture model summary as an object

modelSummary <- summary(linearMod)

Geeft:

Call:

lm(formula = totale\_productiekosten ~ productiehoeveelheid, data = voorbeeld\_regressie)

Residuals:

1 2 3 4 5 6 7

6471.3 17100.9 2778.6 -11673.8 2988.6 -836.1 -16829.6

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 11491.89 11149.90 1.031 0.34995

productiehoeveelheid 64.46 13.96 4.616 0.00576 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12420 on 5 degrees of freedom

Multiple R-squared: 0.81, Adjusted R-squared: 0.7719

F-statistic: 21.31 on 1 and 5 DF, p-value: 0.005755

# model coefficients

modelCoeffs <- modelSummary$coefficients

Geeft:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 11491.89114 11149.90290 1.030672 0.349947427

productiehoeveelheid 64.45893 13.96366 4.616190 0.005755225

# get beta estimate for productiehoeveelheid

beta.estimate <- modelCoeffs["productiehoeveelheid", "Estimate"]

Geeft:

[1] 64.45893

# get std.error for productiehoeveelheid

std.error <- modelCoeffs["productiehoeveelheid", "Std. Error"]

Geeft:

[1] 13.96366

# calc t statistic

t\_value <- beta.estimate/std.error

Geeft:

[1] 4.61619

# calc p Value

p\_value <- 2\*pt(-abs(t\_value), df=nrow(voorbeeld\_regressie)-ncol(voorbeeld\_regressie))

Geeft:

|  |
| --- |
| [1] 0.005755225 |
|  |
| |  | | --- | |  | |

# fstatistic

f\_statistic <- linearMod$fstatistic[1]

Geeft:

NULL

# parameters for model p-value calc

f <- summary(linearMod)$fstatistic

value numdf dendf

21.30921 1.00000 5.00000

model\_p <- pf(f[1], f[2], f[3], lower=FALSE)

Geeft:

value

0.005755225

AIC(linearMod)

Geeft:

[1] 155.4878

BIC(linearMod)

Geeft:

[1] 155.3255

# Create Training and Test data -

set.seed(100) # setting seed to reproduce results of random sampling

trainingRowIndex <- sample(1:nrow(voorbeeld\_regressie), 0.8\*nrow(voorbeeld\_regressie)) # row indices for training data

trainingData <- voorbeeld\_regressie[trainingRowIndex, ] # model training data

testData <- voorbeeld\_regressie[-trainingRowIndex, ] # test data

# Build the model on training data

lmMod <- lm(totale\_productiekosten ~ productiehoeveelheid, data=trainingData) # build the model

totale\_productiekostenPred <- predict(lmMod, testData) # predict totale\_productiekostenance

summary (lmMod) # model summary

Geeft:

Call:

lm(formula = totale\_productiekosten ~ productiehoeveelheid, data = trainingData)

Residuals:

1 2 3 4 5

2126 16608 -11101 1831 -9464

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -3807.56 21285.99 -0.179 0.8694

productiehoeveelheid 96.36 38.38 2.511 0.0869 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12860 on 3 degrees of freedom

Multiple R-squared: 0.6776, Adjusted R-squared: 0.5701

F-statistic: 6.305 on 1 and 3 DF, p-value: 0.08687

actuals\_preds <- data.frame(cbind(actuals=testData$totale\_productiekosten, predicteds=totale\_productiekostenPred)) # make actuals\_predicteds dataframe.

correlation\_accuracy <- cor(actuals\_preds)

head(actuals\_preds)

Geeft:

actuals predicteds

1 67500 97373.03

2 101500 126281.77

# Min-Max Accuracy Calculation

min\_max\_accuracy <- mean(apply(actuals\_preds, 1, min) / apply(actuals\_preds, 1, max))

Geeft:

|  |
| --- |
| [1] 0.7484843 |
|  |
| |  | | --- | |  | |

# MAPE Calculation = mean absolute percentage deviation

mape <- mean(abs((actuals\_preds$predicteds - actuals\_preds$actuals))/actuals\_preds$actuals)

Geeft:

|  |
| --- |
| [1] 0.3433594 |
|  |
| |  | | --- | |  | |

DMwR::regr.eval(actuals\_preds$actuals, actuals\_preds$predicteds)

Geeft:

mae mse rmse mape

2.732740e+04 7.532671e+08 2.744571e+04 3.433594e-01

# Opdracht 2

Resultaat:

De dataset waarin de bestellingen worden uitgezet tegeonover de doorlooptijd levert de volgende data:

DMwR::regr.eval(actuals\_preds$actuals, actuals\_preds$predicteds)

mae mse rmse mape

6.7692308 51.8061538 7.1976492 0.1750466

De dataset waarin de productiehoeveelheid wordt uitgezet tegen de totale\_productiekosten levert de volgende statistieken:

min\_max\_accuracy =

Mape = 0.3433594

regr.eval=

mae mse rmse mape

2.732740e+04 7.532671e+08 2.744571e+04 3.433594e-01

Hieronder een overzicht

|  |  |
| --- | --- |
| Bestelling/doorlooptijd | Productiehoeveelheid/productiekosten |
| Correlatie |  |
| 0.8999736 | .09726135 |
| Beta-estimate |  |
| 64.45893 | 1.961006 |
| Std.error | Standaardafwijking steekproefgemiddelde |
| 13.96366 | 0.191363 |
| T\_value | Verhouding tussen geschatte waarde van een param van een veronderstelde waarde tot de standaardfout |
| 4.61619 | 10.25007 |
| P\_value | De kans dat in de verdeling gegeven een nulhypothese de waarde van de toetsingsgrootheid wordt overschreden. |
| 0.005755225 | 5.030188e-05 |
| F\_statistic |  |
| Null | Null |
| Model\_p |  |
| 0.005755225 | 5.030188 |
| Min\_max\_accuracy | Afwijking van het model |
| 0.7484843 | 0.8513139 |
| Mape | Gemiddelde abslute procentuele afwijking |
| 0.3433594 | 0.1750466 |

Conclusie:

Het zijn twee verschillende datasets toch is komt naar voren dat de null-hypothese bij de eerste dataset kan worden verworpen omdat de p-value onder de significantiegraad van 0.05 valt.

Bij de tweede dataset is de p-value ver onder de 0.05 dus wordt de null hypothese ook verworpen. Gezien de min\_max\_accuracy is het aannemelijk om te stellen dat er een correlatie is, met een lage procentuele afwijking en een lage standaardafwijking van het steekproefgemiddelde. Bijv H0 er is geen verband tussen tussen productiehoeveelheid en productiekosten. Bij H1 is er wel een verband tussen de productiehoeveelheid en productiehoeveelheid.