HA3\_Markt2

Carlo Schmid, Ronny Georgi

06/03/2021

# Hausaufgabe 3 - Markt 2

Import der CSV-Dateien

rm(list = ls())  
transaktionen = read.csv2("output\_transactions\_8Players\_v0013.csv")  
service = read.csv2("output\_services\_8Players\_v0013.csv")  
preise = read.csv2("output\_prices\_8Players\_v0013.csv")  
kosten = read.csv2("output\_cost\_8Players\_v0013.csv")  
externals = read.csv2("externals13.csv")

Laden von Packages

library("dummies")

## dummies-1.5.6 provided by Decision Patterns

library("ggplot2")  
library("Metrics")  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Aufgabe 2.1

### (1)

Laden Sie die Datensaetze externals und services. Berechnen Sie ueber den gesamten Datensatz services, d.h. fuer jede durchgefuehrte Dienstleistung, den On‐Time‐Delivery (OTD) Status (d.h. 0 oder FALSE, wenn unpuenktlich; 1 oder TRUE wenn puenktlich) sowie die Item Fill Rate (IFR). Stellen Sie an‐ schliessend jeweils die Kennzahlen der durchschnittlichen OTD‐Rate und der durchschnittlichen Item Fill Rate als Kennzahl je Logistikdienstleister aggregiert dar. Geben Sie diese Werte in zwei Tabellen aus. Die Tabellen sollen einen einfachen Vergleich der LDL ermöglichen. Bewertungsrelevant: Output, Code.

Hinweis: Erneut bietet es sich an, eine Variable Periode dem Datensatz hinzu zu fuegen, welche aus Jahr und Monat besteht (im Format YYYYMM, z.B. Februar 2014 –> 201402)

# Laden der Daten:  
service = read.csv2("output\_services\_8Players\_v0013.csv")  
externals = read.csv2("externals13.csv")  
  
# Berechnen der OTD:  
service$OTD = service$DaysExecuted <= service$DaysScheduled  
  
# Berechnen der IFR:  
service$IFR = service$QExecuted / service$QScheduled  
  
# Berechnen der durchschnittlichen aggregierten IFR:  
vendorIFR = aggregate(IFR ~ vendor, data = service, mean)  
  
# Berechnen der durchschnittlichen aggregierten OTD pro Logistikdienstleister  
vendorOTD = aggregate(OTD ~ vendor, data = service, mean)  
  
# Erstellen einer Tabelle fuer IFR in der Kennzahlen absteigend und nur   
# relevante Daten enthalten sind  
vendorIFR\_sorted = subset(vendorIFR[order(vendorIFR$IFR, decreasing = TRUE),], IFR!=1)  
  
# Entfernen der Zeilennummer  
rownames(vendorIFR\_sorted) = NULL  
  
# Erzeugung eines dataframes  
df\_IFR = data.frame(Logistikdienstleister\_Warehousing = vendorIFR\_sorted$vendor,   
 IFR = vendorIFR\_sorted$IFR)  
  
# die OTD-Werte auf 3 Nachkommastellen runden:  
library(dplyr)  
df\_IFR %>%   
 mutate\_if(is.numeric, round, digits=3)

## Logistikdienstleister\_Warehousing IFR  
## 1 CPS Warehousing 0.844  
## 2 Flying Mercury Warehousing 0.838  
## 3 Bange+Hammer Warehousing 0.831  
## 4 JNT Warehousing 0.830  
## 5 Gifter Warehousing 0.827  
## 6 EPD Warehousing 0.827  
## 7 DWL Warehousing 0.824  
## 8 HCX Warehousing 0.822  
## 9 IntEx Warehousing 0.819  
## 10 AHL Express Warehousing 0.816

# Erstellen einer Tabelle fuer IFR in der Kennzahlen absteigend und nur  
# relevante Daten enthalten sind  
vendortOTD\_sort = subset(vendorOTD[order(vendorOTD$OTD, decreasing = TRUE),], OTD!=1)   
  
# Entfernen der Zeilennummer  
rownames(vendortOTD\_sort) = NULL  
  
# Erzeugung eines dataframes  
df\_OTD = data.frame(Logistikdienstleister\_Shipping = vendortOTD\_sort$vendor,   
 OTD\_Rate = vendortOTD\_sort$OTD)  
  
# Runden der OTD-Werte auf 3 Nachkommastellen (da die Werte sehr nah bei einander liegen   
# reicht eine Rundung auf 2 Nachkommastellen nicht für einen Vergleich aus)  
df\_OTD %>%   
 mutate\_if(is.numeric, round, digits=3)

## Logistikdienstleister\_Shipping OTD\_Rate  
## 1 IntEx Shipping 0.481  
## 2 HCX Shipping 0.428  
## 3 CPS Shipping 0.410  
## 4 Gifter Shipping 0.393  
## 5 EPD Shipping 0.386  
## 6 Flying Mercury Shipping 0.359  
## 7 JNT Shipping 0.320  
## 8 DWL Shipping 0.314  
## 9 Bange+Hammer Shipping 0.289  
## 10 AHL Express Shipping 0.282

### (2)

Erzeugen Sie ein neues Dataframe, welches die aggregierte IFR je Warehousing‐Logistikdienstleister enthaelt. Die IFR soll je LDL (nur Warehousing), Region und Periode (eine Periode = ein Monat eines einzelnen Jahres) aggregiert werden. Geben Sie anschliessend den IFR‐Wert (und die entsprechende Periode) aus, den der beste Warehousing‐DL in seiner schlechtesten Periode in der Region Peking erreicht hat. Bewertungsrelevant: Output, Code.

# Erstellen einer Periode  
service$Period = sprintf("%1.0f/%02d",service$Year, service$Month)  
  
# Erstellen eines Dataframes in der die aggregierte IFR je WH-DL über Region und Periode   
# angezeigt wird  
WH\_IFR = aggregate(IFR ~ vendor + region + Period,   
 data = subset(service, service=="Warehousing"), mean)  
  
#WH\_IFR  
  
WH\_IFR\_vendors = aggregate(IFR ~ vendor, data = subset(service, service=="Warehousing"), mean)  
best\_vendor\_IFR = max(WH\_IFR\_vendors[,c("IFR")])  
best\_vendor = WH\_IFR\_vendors[WH\_IFR\_vendors$IFR == best\_vendor\_IFR,c("vendor")]  
  
bv\_IFR\_peking = subset(WH\_IFR, vendor==best\_vendor & region=="Peking")  
bv\_min\_IFR\_peking = min(bv\_IFR\_peking[,c("IFR")])  
bv\_min\_IFR\_peking\_periode = bv\_IFR\_peking[bv\_IFR\_peking$IFR == bv\_min\_IFR\_peking,c("Period")]  
bv\_min\_IFR\_peking\_periode

## [1] "2016/09"

cat(best\_vendor,"hat insgesamt die höchste IFR und ist damit der beste LDL für Warehousing.\n")

## CPS Warehousing hat insgesamt die höchste IFR und ist damit der beste LDL für Warehousing.

cat("Der LDL",best\_vendor,"hat in Peking seine geringste IFR in der Periode",bv\_min\_IFR\_peking\_periode,".\n")

## Der LDL CPS Warehousing hat in Peking seine geringste IFR in der Periode 2016/09 .

cat("Dort beträgt die IFR",round(bv\_min\_IFR\_peking\*100,1),"%.\n")

## Dort beträgt die IFR 82.5 %.

### (3)

Erzeugen Sie ein neues Dataframe, welches die aggregierte OTD je Shipping‐Logistikdienstleister enthaelt. Die OTD soll je LDL (nur Shipping), Region und Periode (eine Periode = ein Monat eines einzelnen Jahres) aggregiert werden. Geben Sie anschliessend den OTD‐Wert (und die entsprechende Periode) aus, den der schlechteste Shipping‐DL in seiner besten Periode in der Region Peking erreicht hat. Bewertungsrelevant: Output, Code.

# Erzeugen von dataframe mit OTD für alle Shipping-Dienstleister  
SH\_OTD = aggregate(OTD ~ vendor + region + Period,   
 data = subset(service, service=="Shipping"), mean)  
#SH\_OTD  
  
SH\_OTD\_vendors = aggregate(OTD ~ vendor, data = subset(service, service=="Shipping"), mean)  
worst\_vendor\_OTD = min(SH\_OTD\_vendors[,c("OTD")])  
worst\_vendor = SH\_OTD\_vendors[SH\_OTD\_vendors$OTD == worst\_vendor\_OTD,c("vendor")]  
  
wv\_OTD\_peking = subset(SH\_OTD, vendor==worst\_vendor & region=="Peking")  
wv\_max\_OTD\_peking = max(wv\_OTD\_peking[,c("OTD")])  
wv\_max\_OTD\_peking\_periode = wv\_OTD\_peking[wv\_OTD\_peking$OTD == wv\_max\_OTD\_peking,c("Period")]  
  
cat(worst\_vendor,"hat insgesamt die niedrigste OTD und ist damit der schlechteste LDL für Shipping.\n")

## AHL Express Shipping hat insgesamt die niedrigste OTD und ist damit der schlechteste LDL für Shipping.

cat("Der LDL",worst\_vendor,"hat in Peking seine höchste IFR in der Periode",wv\_max\_OTD\_peking\_periode,".\n")

## Der LDL AHL Express Shipping hat in Peking seine höchste IFR in der Periode 2020/07 .

cat("Dort beträgt die OTD",round(wv\_max\_OTD\_peking\*100,1),"%.\n")

## Dort beträgt die OTD 50 %.

### (4)

Waehlen Sie den Warehousing‐DL “AHL Express Warehousing” aus. Vereinigen Sie das eben erzeugte DataFrame (genauer: Ein Subset dieses Dataframes bezueglich des gewaehlten Warehousing‐DL) mit den externen Faktoren der jeweiligen Periode und Region in einem neuen Dataframe. Zeigen Sie davon den Tabellenkopf. Bewertungsrelevant: Output. Hinweis: In der Funktion merge() koennen mehrere ueberschneidende Spalten genutzt werden, indem dem “by =”‐Parameter ein Vektor der Spalten uebergeben wird. Ihnen steht frei, andere Funktionen zu verwenden.

AHL= subset(WH\_IFR, vendor=="AHL Express Warehousing")  
  
AHL\_externals = merge(AHL,externals[externals$Year!="2021",])  
  
AHL\_externals = AHL\_externals [, c((1:4), (8:29))]  
  
head(AHL\_externals)

## region Period vendor IFR Temperature\_C Rain\_mm  
## 1 Japan 2016/10 AHL Express Warehousing 0.7982881 12.783 37.459  
## 2 Japan 2016/11 AHL Express Warehousing 0.8122462 4.905 48.561  
## 3 Japan 2016/12 AHL Express Warehousing 0.7864903 2.798 50.487  
## 4 Japan 2017/10 AHL Express Warehousing 0.7806593 10.893 24.129  
## 5 Japan 2017/11 AHL Express Warehousing 0.7819056 7.003 47.883  
## 6 Japan 2017/12 AHL Express Warehousing 0.7806554 2.755 59.369  
## Sunshine\_h Humidity Congestion InternetStability PowerGridStability  
## 1 106.599 74.881 47.67 1843 0.21  
## 2 47.986 77.286 39.89 2007 0.25  
## 3 34.939 87.309 53.81 1994 0.22  
## 4 91.627 65.801 39.41 1707 0.20  
## 5 48.764 76.915 43.10 1780 0.25  
## 6 39.549 81.966 41.00 1716 0.20  
## ParkingSpaceAvailability RoadCondition PoliticalStability AvgHealth  
## 1 0.03114353 5.37 2.41 74.82  
## 2 0.02749802 6.18 2.30 74.44  
## 3 0.03144905 5.78 2.37 73.90  
## 4 0.03277340 5.84 2.93 75.89  
## 5 0.03119713 6.29 2.74 75.85  
## 6 0.03142005 6.71 2.42 75.97  
## Criminality AirPollution WaterQuality leisureAndSocialInteractions  
## 1 24.86 236.69 0.88 6.14  
## 2 24.91 209.72 0.95 6.20  
## 3 23.67 187.91 0.98 7.01  
## 4 22.59 226.45 0.98 6.65  
## 5 21.69 238.21 0.85 7.09  
## 6 21.39 218.18 0.98 6.44  
## SkilledLaborAvailability UnskilledLaborAvailability WorkerMotivation Overtime  
## 1 52.17 45.73 5.59 0.159  
## 2 51.97 45.83 5.47 0.046  
## 3 52.20 44.88 6.11 0.179  
## 4 52.71 47.79 6.91 0.000  
## 5 53.37 48.84 7.08 0.162  
## 6 51.58 48.15 7.06 0.022  
## Inflation BusinessConfidence FuelPrice  
## 1 0.00102262 100.6836 2.617  
## 2 0.00095997 100.5842 2.671  
## 3 0.00088866 100.5054 2.531  
## 4 0.00094883 100.0760 2.257  
## 5 0.00097577 100.0014 2.368  
## 6 0.00096550 100.0132 2.442

### (5)

Sie moechten sich eine Uebersicht zu der Korrelation zwischen den externen Faktoren und der IFR des Warehousing‐Dienstleister schaffen. Fuehren Sie dazu die folgenden Schritte aus:

1. Geben Sie eine unsortierte Tabelle aus, in der die externen Effekte und deren Korrelation zur IFR abgebildet sind.

# Korrelation zwischen externen Effekten und IFR herstellen   
Correlation\_Externals = cor(AHL\_externals[, c(4:26)])  
  
Cor\_IFR\_sub = Correlation\_Externals[, c(0:2)]  
  
Cor\_IFR\_sub

## IFR Temperature\_C  
## IFR 1.00000000 -0.43896874  
## Temperature\_C -0.43896874 1.00000000  
## Rain\_mm -0.31300445 0.46622877  
## Sunshine\_h -0.41508242 0.93154237  
## Humidity 0.28444282 -0.54344523  
## Congestion -0.39650984 0.69403898  
## InternetStability -0.02347020 -0.30716867  
## PowerGridStability 0.33690160 0.02021967  
## ParkingSpaceAvailability 0.49240725 -0.68432487  
## RoadCondition -0.43907920 0.27389141  
## PoliticalStability 0.24486488 -0.01674296  
## AvgHealth -0.39008282 0.32557096  
## Criminality 0.26104283 -0.17635433  
## AirPollution -0.37703831 0.04319984  
## WaterQuality 0.03636328 -0.33122269  
## leisureAndSocialInteractions 0.24265455 -0.31336604  
## SkilledLaborAvailability 0.19281619 -0.16400508  
## UnskilledLaborAvailability -0.03014333 0.19817238  
## WorkerMotivation -0.23990206 0.40585696  
## Overtime -0.12226749 0.20981304  
## Inflation 0.40272856 -0.30939398  
## BusinessConfidence 0.06646015 0.16088187  
## FuelPrice 0.06163320 -0.02376250

1. Geben Sie eine Tabelle aus, in der die 5 am starksten zur IFR korrelierenden externen Effekten und deren Korrelation zur IFR abgebildet sind.

cor\_Strong = data.frame(Cor\_IFR\_sub[c(rownames(data.frame(sort(abs(Cor\_IFR\_sub[,1]),TRUE)[2:6]))),1])  
colnames(cor\_Strong) = c("Correlation to IFR")  
round(cor\_Strong,3)

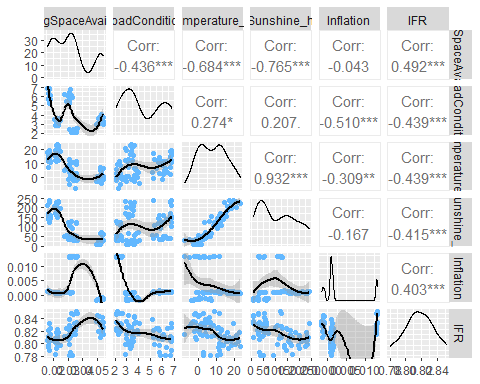
## Correlation to IFR  
## ParkingSpaceAvailability 0.492  
## RoadCondition -0.439  
## Temperature\_C -0.439  
## Sunshine\_h -0.415  
## Inflation 0.403

1. Erstellen Sie ein Korrelations‐Plot fuer diese 5 externen Faktoren. Bewertungsrelevant: Output.

library("GGally")

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

ggpairs(AHL\_externals[, c(rownames(cor\_Strong),"IFR")],  
# ohne Visualisierung des Fortschritts der Erstellung des plots  
progress = FALSE,  
# mit Visualisierung einer Glaettungslinie und Aenderung der Farbe der Punkte,  
#damit Linie erkennbar  
lower = list(continuous = wrap("smooth\_loess", colour = "steelblue1")))



### (6)

Sie moechten nun eine Lineare Regression durchfuehren, um die IFR mit Hilfe der externen Effekte vorherzusagen. Um die Guete Ihrer Modelle vergleichen zu koennen, benoetigen Sie eine geeignete Baseline. Erzeugen Sie eine sinnvolle Baseline in dem DataFrame zu Ihrem gewaehlten Warehousing‐DL in einer Variable Baseline. Begruenden Sie Ihre Wahl. Geben Sie von dem DataFrame den Tabellenkopf aus. Geben Sie Sie nur die Spalten ‘Periode’, ‘Region’, ‘IFR’ und ‘Baseline’ aus. Bewertungsrelevant: Output, Begruendung.

# Baseline durch Mittelwert erzeugen und Ausgabe in einem neuen Dataframe  
AHL\_externals$Baseline = mean(AHL\_externals$IFR)  
  
df\_base = head(AHL\_externals[, c("Period", "region", "IFR","Baseline")])  
  
df\_base %>%   
 mutate\_if(is.numeric, round, digits=3)

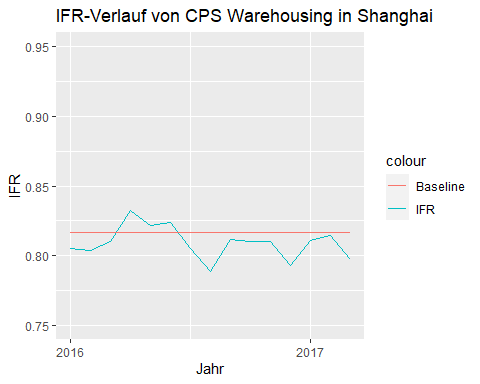
## Period region IFR Baseline  
## 1 2016/10 Japan 0.798 0.817  
## 2 2016/11 Japan 0.812 0.817  
## 3 2016/12 Japan 0.786 0.817  
## 4 2017/10 Japan 0.781 0.817  
## 5 2017/11 Japan 0.782 0.817  
## 6 2017/12 Japan 0.781 0.817

Begründung: Wir können hier den Durchschnitt der IFR als Baseline benutzen, da der Durchschnitt unabhängig von den externen Effekten ist und auch grafisch gut darstellbar ist.

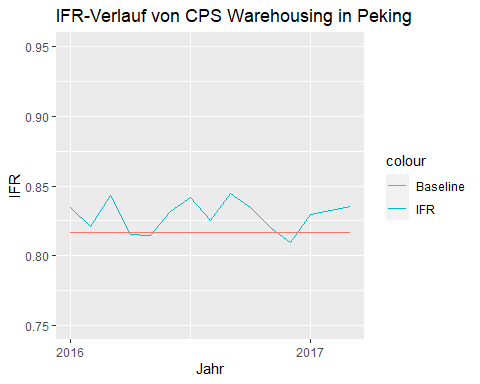
### (7)

Visualisieren Sie die Baseline Ihres gewaehlten LDL fuer den Zeitraum von 2016 bis 2020 sowie die IFR in der Region Peking und die IFR in der Region Shanghai. Bewertungsrelevant: Output.

# Subset fuer die Region Shanghai erstellen  
AHL\_externals\_Shang= subset(AHL\_externals, region=="Shangh")  
  
# Erzeugung Liniendiagramm  
ggplot(data=AHL\_externals\_Shang, aes(ymin = 0.75, ymax = 0.95)) +  
# Hinufügen von IFR und Baseline  
geom\_line(data=AHL\_externals\_Shang, aes(x=seq(1,length(Period),1), y=IFR,  
colour = 'IFR')) +  
# Zweite Linie hinzufuegen (Baseline)  
geom\_line(data=AHL\_externals\_Shang, aes(x=seq(1,length(Period),1), y=Baseline,  
colour = 'Baseline')) +  
labs(x = "Jahr", title="IFR-Verlauf von CPS Warehousing in Shanghai") +  
scale\_x\_continuous(breaks = c(1,13,25,37,49, 61),  
labels = c("2016","2017","2018","2019", "2020", "2021"))



# Subset fuer die Region Peking erstellen  
AHL\_externals\_Peking = subset(AHL\_externals, region=="Peking")  
  
# Erzeugung Liniendiagramm  
ggplot(data=AHL\_externals\_Peking, aes(ymin = 0.75, ymax = 0.95)) +  
# Hinufügen von IFR und Baseline  
geom\_line(data=AHL\_externals\_Peking, aes(x=seq(1,length(Period),1), y=IFR,  
colour = 'IFR')) +  
# Zweite Linie hinzufuegen (Baseline)  
geom\_line(data=AHL\_externals\_Peking, aes(x=seq(1,length(Period),1), y=Baseline,  
colour = 'Baseline')) +  
labs(x = "Jahr", title="IFR-Verlauf von CPS Warehousing in Peking") +  
scale\_x\_continuous(breaks = c(1,13,25,37,49, 61),  
labels = c("2016","2017","2018","2019", "2020", "2021"))



### (8)

Bewerten Sie die Baseline fuer Ihren gewaehlten Warehousing‐Logistikdienstleister nach MAE und MAPE. Bewertungsrelevant: Output. Hinweis: Es bietet sich an, die Werte der Bewertungen in einem Dataframe ‘evaluation’ zu speichern.

# DataFrame erzeugen, das mit den werten der Modellbewertung später gefüllt wird  
evaluation = data.frame(Model = "Baseline",  
 MAE = numeric(1),  
 MAPE = numeric(1),  
 R\_Squared = numeric (1),  
 R\_Squared\_adj = numeric(1))  
  
# MAE berechnen  
evaluation[evaluation$Model == "Baseline",]$MAE = mean(abs(AHL\_externals$IFR - AHL\_externals$Baseline))  
  
# MAPE berechnen  
evaluation[evaluation$Model == "Baseline",]$MAPE = mape(AHL\_externals$IFR, AHL\_externals$Baseline)  
  
# R-Squared berechnen  
evaluation[evaluation$Model == "Baseline",]$R\_Squared = ""  
  
# R-Squared\_Adj berechnen  
evaluation[evaluation$Model == "Baseline",]$R\_Squared\_adj = ""  
  
  
evaluation

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.01426048 0.0174863

### (9)

Teilen Sie das Dataframe Ihres gewaehlten Warehousing‐Logistikdienstleisters in ein Trainings‐ (80%) und ein Test‐Set (20%) auf. Geben Sie von beiden den Tabellenkopf aus. Setzen Sie vorher den Seed 4141. Bewer‐ tungsrelevant: Code, Output.

set.seed(4141)  
  
#Zufällig gezogenes Sample erstellen  
zufall = sample(1:nrow(AHL\_externals), nrow(AHL\_externals) \* 0.8)  
  
AHL\_externals\_training = AHL\_externals[zufall, ]  
  
rownames(AHL\_externals\_training) = NULL  
  
head(AHL\_externals\_training)

## region Period vendor IFR Temperature\_C Rain\_mm  
## 1 Shangh 2018/10 AHL Express Warehousing 0.8053654 18.705 89.576  
## 2 Phlppn 2018/12 AHL Express Warehousing 0.8086538 12.249 126.238  
## 3 Peking 2019/10 AHL Express Warehousing 0.8344992 5.489 87.049  
## 4 Skorea 2020/11 AHL Express Warehousing 0.8327696 3.112 44.073  
## 5 Skorea 2019/11 AHL Express Warehousing 0.8215882 0.626 43.696  
## 6 Shangh 2020/12 AHL Express Warehousing 0.7974545 14.264 123.163  
## Sunshine\_h Humidity Congestion InternetStability PowerGridStability  
## 1 233.610 69.968 64.00 274 1.24  
## 2 159.352 73.903 49.31 611 2.64  
## 3 64.092 85.048 31.69 1721 1.60  
## 4 53.091 86.676 40.76 65 3.23  
## 5 51.885 75.099 42.07 70 3.48  
## 6 137.858 78.886 57.59 334 1.32  
## ParkingSpaceAvailability RoadCondition PoliticalStability AvgHealth  
## 1 0.01706371 6.72 3.07 73.98  
## 2 0.02280499 3.49 2.20 79.98  
## 3 0.05191863 3.84 1.47 75.41  
## 4 0.03443367 2.46 3.74 57.06  
## 5 0.03494417 2.13 3.23 57.70  
## 6 0.01660268 5.78 3.06 78.46  
## Criminality AirPollution WaterQuality leisureAndSocialInteractions  
## 1 38.19 67.62 0.89 7.00  
## 2 12.28 66.70 0.66 6.13  
## 3 13.63 12.57 0.94 7.45  
## 4 40.08 58.17 0.73 6.88  
## 5 38.98 55.41 0.71 6.54  
## 6 35.98 68.94 0.85 6.03  
## SkilledLaborAvailability UnskilledLaborAvailability WorkerMotivation Overtime  
## 1 61.79 40.49 6.29 0.176  
## 2 39.76 64.59 8.08 0.176  
## 3 35.57 45.81 3.72 0.085  
## 4 75.47 57.69 8.60 0.034  
## 5 72.49 59.63 7.29 0.000  
## 6 57.45 37.85 6.39 0.204  
## Inflation BusinessConfidence FuelPrice Baseline  
## 1 0.00123952 100.8986 2.069 0.8167054  
## 2 0.00072088 100.9120 2.114 0.8167054  
## 3 -0.00125430 99.0479 2.568 0.8167054  
## 4 0.01315555 101.9247 2.719 0.8167054  
## 5 0.01297422 102.0838 2.642 0.8167054  
## 6 0.00160202 100.7581 2.686 0.8167054

AHL\_externals\_test = AHL\_externals[-zufall, ]  
  
rownames(AHL\_externals\_test) = NULL  
  
# Ausgabe ohne Rundung der Werte, da auch mit den genauen Werten weitergerechnet wird  
head(AHL\_externals\_test)

## region Period vendor IFR Temperature\_C Rain\_mm  
## 1 Japan 2016/10 AHL Express Warehousing 0.7982881 12.783 37.459  
## 2 Japan 2019/10 AHL Express Warehousing 0.8258688 11.071 35.081  
## 3 Peking 2016/10 AHL Express Warehousing 0.8344494 6.595 79.297  
## 4 Peking 2016/12 AHL Express Warehousing 0.8428941 0.173 67.922  
## 5 Peking 2017/10 AHL Express Warehousing 0.8156051 3.752 72.382  
## 6 Phlppn 2016/11 AHL Express Warehousing 0.7922282 16.770 76.560  
## Sunshine\_h Humidity Congestion InternetStability PowerGridStability  
## 1 106.599 74.881 47.67 1843 0.21  
## 2 100.154 68.923 32.58 1679 0.19  
## 3 60.099 77.545 29.51 1923 2.04  
## 4 8.675 81.093 30.96 2117 1.80  
## 5 69.066 85.822 21.19 1617 1.63  
## 6 200.109 76.611 64.63 699 2.53  
## ParkingSpaceAvailability RoadCondition PoliticalStability AvgHealth  
## 1 0.03114353 5.37 2.41 74.82  
## 2 0.03137741 6.76 2.71 82.05  
## 3 0.05400501 3.74 1.96 85.69  
## 4 0.05140936 3.67 1.89 85.68  
## 5 0.05268777 3.97 1.72 81.32  
## 6 0.02242115 3.30 1.67 76.76  
## Criminality AirPollution WaterQuality leisureAndSocialInteractions  
## 1 24.86 236.69 0.88 6.14  
## 2 24.26 203.00 0.88 7.20  
## 3 20.09 14.30 0.95 7.16  
## 4 19.44 12.02 0.98 7.51  
## 5 21.08 12.56 0.98 7.59  
## 6 16.72 72.10 0.63 5.38  
## SkilledLaborAvailability UnskilledLaborAvailability WorkerMotivation Overtime  
## 1 52.17 45.73 5.59 0.159  
## 2 48.35 39.60 4.06 0.302  
## 3 34.49 51.22 5.74 0.003  
## 4 35.54 50.90 5.59 0.175  
## 5 35.74 46.84 5.74 0.008  
## 6 42.01 62.57 7.87 0.339  
## Inflation BusinessConfidence FuelPrice Baseline  
## 1 0.00102262 100.6836 2.617 0.8167054  
## 2 0.00104047 99.6719 2.483 0.8167054  
## 3 -0.00079482 99.1375 2.698 0.8167054  
## 4 -0.00079932 98.9214 2.835 0.8167054  
## 5 -0.00080989 99.2797 2.368 0.8167054  
## 6 0.00071238 100.4893 2.659 0.8167054

### (10)

Wenden Sie die Forward Selection Variante der Wrapper Methode an (siehe Vorlesung). D.h. erstellen Sie zunaechst alle uni‐variaten Modelle, bewerten Sie diese Modelle und waehlen Sie das Modell mit der besten Bewertung aus. Erstellen Sie ‐ basierend auf dem besten Modell der ersten Iteration ‐ alle bi‐variaten Mod‐ elle (das Modell der vorherigen Wrapper‐Iteration wird jeweils um eine Variable erweitert), bewerten Sie diese Modelle und waehlen Sie das Modell mit der besten Bewertung aus. Fuehren Sie dies so lange fort, bis keine Verbesserung mehr erreicht wird. Nutzen Sie zur Modellierung die lineare Regression. Bewerten Sie die Mod‐ elle entsprechend nach MAE und MAPE sowie nach regressionsspezifischen Kennzahlen. Nutzen Sie nur die 5 externen Faktoren als Features, die Sie oben als am staerksten korrelierende externe Faktoren identifiziert haben. Kommentieren Sie Ihr Vorgehen zwischen den Iterationen. Bewertungsrelevant: Output, Vorgehen (einschliesslich Kommentare). Hinweis: Tritt eine starke Multikollinearität (“strong multicollinearity”) auf, so koennen Sie alle Mod‐ ellierungen mit der entsprechenden Variablen‐Kombination unter Bezug auf diesen Hinweis auslassen (siehe Vorlesungsinhalte zu Korrelation). 4 Hinweis 2: Fuer das Erstellen der Modelle reicht es aus, zunaechst die Trainings‐Daten zu nutzen. Ueber‐ pruefen Sie ihr endgueltiges Modell jedoch am Ende auf Overfitting, indem Sie die Test‐Daten nutzen!

#### Iteration 1

# Modelle mit Trainings-Sets erstellen und anzeigen  
model\_1 = lm(IFR ~ ParkingSpaceAvailability, data = AHL\_externals\_training)  
model\_2 = lm(IFR ~ RoadCondition, data = AHL\_externals\_training)  
model\_3 = lm(IFR ~ Temperature\_C, data = AHL\_externals\_training)  
model\_4 = lm(IFR ~ Sunshine\_h, data = AHL\_externals\_training)  
model\_5 = lm(IFR ~ Inflation, data = AHL\_externals\_training)  
  
  
summary(model\_1)

##   
## Call:  
## lm(formula = IFR ~ ParkingSpaceAvailability, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.037091 -0.009277 0.001252 0.007527 0.030844   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.795690 0.005804 137.100 < 2e-16 \*\*\*  
## ParkingSpaceAvailability 0.673123 0.171213 3.931 0.000228 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01603 on 58 degrees of freedom  
## Multiple R-squared: 0.2104, Adjusted R-squared: 0.1968   
## F-statistic: 15.46 on 1 and 58 DF, p-value: 0.0002277

summary(model\_2)

##   
## Call:  
## lm(formula = IFR ~ RoadCondition, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.040673 -0.011407 0.001497 0.011371 0.027841   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.838682 0.005747 145.930 < 2e-16 \*\*\*  
## RoadCondition -0.005027 0.001245 -4.039 0.00016 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01594 on 58 degrees of freedom  
## Multiple R-squared: 0.2195, Adjusted R-squared: 0.2061   
## F-statistic: 16.32 on 1 and 58 DF, p-value: 0.0001595

summary(model\_3)

##   
## Call:  
## lm(formula = IFR ~ Temperature\_C, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.040603 -0.007299 -0.000787 0.009701 0.035112   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.8236967 0.0030715 268.173 <2e-16 \*\*\*  
## Temperature\_C -0.0008852 0.0002888 -3.065 0.0033 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01674 on 58 degrees of freedom  
## Multiple R-squared: 0.1394, Adjusted R-squared: 0.1246   
## F-statistic: 9.396 on 1 and 58 DF, p-value: 0.003299

summary(model\_4)

##   
## Call:  
## lm(formula = IFR ~ Sunshine\_h, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.041722 -0.008228 -0.000765 0.010185 0.034988   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.257e-01 3.795e-03 217.556 < 2e-16 \*\*\*  
## Sunshine\_h -8.322e-05 2.980e-05 -2.793 0.00707 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01694 on 58 degrees of freedom  
## Multiple R-squared: 0.1185, Adjusted R-squared: 0.1033   
## F-statistic: 7.8 on 1 and 58 DF, p-value: 0.007065

summary(model\_5)

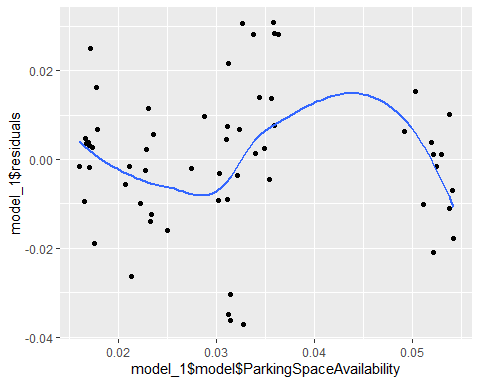
##   
## Call:  
## lm(formula = IFR ~ Inflation, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.033365 -0.009682 -0.001158 0.011636 0.033917   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.812627 0.002462 330.11 < 2e-16 \*\*\*  
## Inflation 1.442483 0.413335 3.49 0.00093 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0164 on 58 degrees of freedom  
## Multiple R-squared: 0.1735, Adjusted R-squared: 0.1593   
## F-statistic: 12.18 on 1 and 58 DF, p-value: 0.00093

**Regressionsbewertung:** Unabhaengige Variablen sind signifikant und haben ein R^2 und adjusted R^2 zwischen 0.1 und 0.22.

Modelle bewerten: (1) Residuenplots

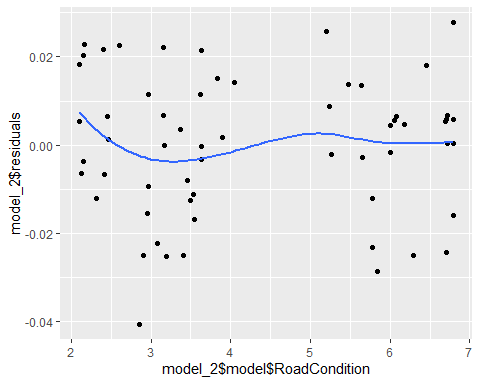
# Residuenplot 1  
ggplot(data = NULL, aes(x = model\_1$model$ParkingSpaceAvailability, y = model\_1$residuals)) +   
 geom\_point() +   
 geom\_smooth(se = FALSE, method = loess)

## `geom\_smooth()` using formula 'y ~ x'



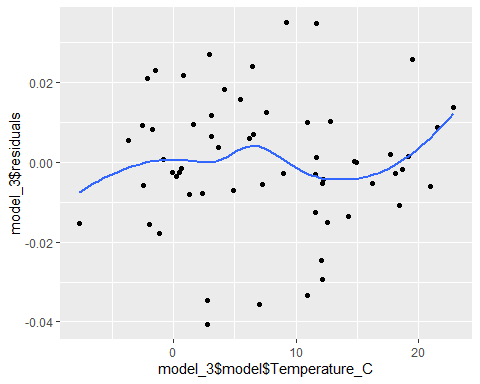
# Residuenplot 2  
ggplot(data = NULL, aes(x = model\_2$model$RoadCondition, y = model\_2$residuals)) +   
 geom\_point() +   
 geom\_smooth(se = FALSE, method = loess)

## `geom\_smooth()` using formula 'y ~ x'



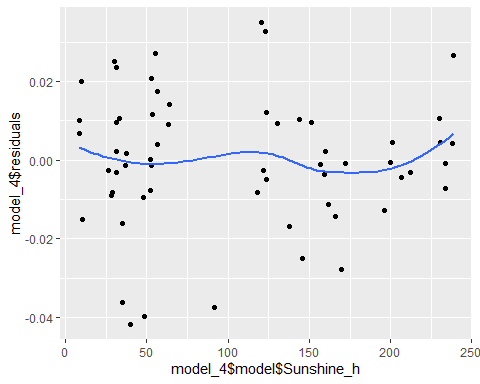
# Residuenplot 3  
ggplot(data = NULL, aes(x = model\_3$model$Temperature\_C, y = model\_3$residuals)) +   
 geom\_point() +   
 geom\_smooth(se = FALSE, method = loess)

## `geom\_smooth()` using formula 'y ~ x'



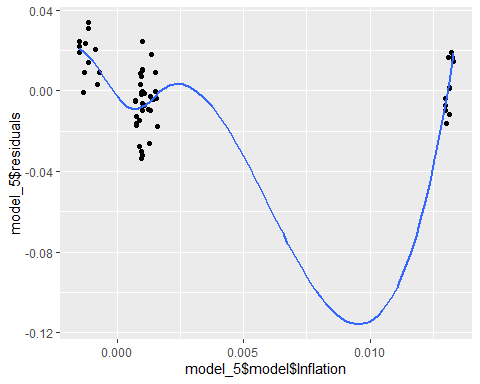
# Residuenplot 4  
ggplot(data = NULL, aes(x = model\_4$model$Sunshine\_h, y = model\_4$residuals)) +   
 geom\_point() +   
 geom\_smooth(se = FALSE, method = loess)

## `geom\_smooth()` using formula 'y ~ x'



# Residuenplot 5  
ggplot(data = NULL, aes(x = model\_5$model$Inflation, y = model\_5$residuals)) +   
 geom\_point() +   
 geom\_smooth(se = FALSE, method = loess)

## `geom\_smooth()` using formula 'y ~ x'

 Residuenplots sehen soweit gut aus.

1. Fehlerkennzahlen (fuer Trainings-Set):

# Data Frame erweitern  
evaluation = rbind(evaluation, data.frame(Model = c("Model1","Model2","Model3","Model4","Model5"),  
 MAE = numeric(1),  
 MAPE = numeric(1),  
 R\_Squared = numeric (1),  
 R\_Squared\_adj = numeric(1)))  
  
# MAE berechnen  
evaluation[evaluation$Model == "Model1",]$MAE = mean(abs(model\_1$residuals))  
evaluation[evaluation$Model == "Model2",]$MAE = mean(abs(model\_2$residuals))  
evaluation[evaluation$Model == "Model3",]$MAE = mean(abs(model\_3$residuals))  
evaluation[evaluation$Model == "Model4",]$MAE = mean(abs(model\_4$residuals))  
evaluation[evaluation$Model == "Model5",]$MAE = mean(abs(model\_5$residuals))  
  
# MAPE berechnen  
evaluation[evaluation$Model == "Model1",]$MAPE = mape(model\_1$model$IFR, model\_1$fitted.values)  
evaluation[evaluation$Model == "Model2",]$MAPE = mape(model\_2$model$IFR, model\_2$fitted.values)  
evaluation[evaluation$Model == "Model3",]$MAPE = mape(model\_3$model$IFR, model\_3$fitted.values)  
evaluation[evaluation$Model == "Model4",]$MAPE = mape(model\_4$model$IFR, model\_4$fitted.values)  
evaluation[evaluation$Model == "Model5",]$MAPE = mape(model\_5$model$IFR, model\_5$fitted.values)  
  
  
# R-Squared berechnen  
evaluation[evaluation$Model == "Model1",]$R\_Squared = summary(model\_1)$r.squared  
evaluation[evaluation$Model == "Model2",]$R\_Squared = summary(model\_2)$r.squared  
evaluation[evaluation$Model == "Model3",]$R\_Squared = summary(model\_3)$r.squared  
evaluation[evaluation$Model == "Model4",]$R\_Squared = summary(model\_4)$r.squared  
evaluation[evaluation$Model == "Model5",]$R\_Squared = summary(model\_5)$r.squared  
  
# R-Squared\_Adj berechnen  
evaluation[evaluation$Model == "Model1",]$R\_Squared\_adj = summary(model\_1)$adj.r.squared  
evaluation[evaluation$Model == "Model2",]$R\_Squared\_adj = summary(model\_2)$adj.r.squared  
evaluation[evaluation$Model == "Model3",]$R\_Squared\_adj = summary(model\_3)$adj.r.squared  
evaluation[evaluation$Model == "Model4",]$R\_Squared\_adj = summary(model\_4)$adj.r.squared  
evaluation[evaluation$Model == "Model5",]$R\_Squared\_adj = summary(model\_5)$adj.r.squared  
  
# Fehler anzeigen  
evaluation

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.01426048 0.01748630   
## 2 Model1 0.01194009 0.01464587 0.21041778092297 0.196804294387159  
## 3 Model2 0.01266202 0.01554046 0.219549710290505 0.206093670812755  
## 4 Model3 0.01251803 0.01534934 0.139410926633964 0.124573183989722  
## 5 Model4 0.01249369 0.01533179 0.118543236061212 0.103345705648475  
## 6 Model5 0.01293505 0.01586560 0.173543889806034 0.159294646526827

Alle Modelle weisen verbesserte Fehlerkennzahlen im Vergleich zur Baseline auf. Modell 1 schneidet bei den Fehlerkennzahlen am besten ab, während Modell 2 in den R^2 und R^2 adjusted Werten dominiert. Wir machen mit Modell 1 weiter.

#### Iteration 2

Bivariate Modelle erstellen: Die Variablen Temperature\_C und Sunshine\_h haben eine starke Multikollinearität mit der Variable ParkingSpaceAvailability und IFR. Daher werden diese Variablenkombination nicht weiter betrachtet.

# Modelle mit Trainings-Sets erstellen und anzeigen  
model\_12 = lm(IFR ~ ParkingSpaceAvailability + RoadCondition, data = AHL\_externals\_training)  
model\_15 = lm(IFR ~ ParkingSpaceAvailability + Inflation, data = AHL\_externals\_training)  
  
  
summary(model\_12)

##   
## Call:  
## lm(formula = IFR ~ ParkingSpaceAvailability + RoadCondition,   
## data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.033802 -0.008297 0.002503 0.008753 0.030560   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.817790 0.010152 80.553 <2e-16 \*\*\*  
## ParkingSpaceAvailability 0.451586 0.184255 2.451 0.0173 \*   
## RoadCondition -0.003499 0.001347 -2.597 0.0119 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0153 on 57 degrees of freedom  
## Multiple R-squared: 0.294, Adjusted R-squared: 0.2692   
## F-statistic: 11.87 on 2 and 57 DF, p-value: 4.917e-05

summary(model\_15)

##   
## Call:  
## lm(formula = IFR ~ ParkingSpaceAvailability + Inflation, data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.033974 -0.007662 0.001300 0.009505 0.027977   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.79033 0.00526 150.244 < 2e-16 \*\*\*  
## ParkingSpaceAvailability 0.69800 0.15074 4.631 2.16e-05 \*\*\*  
## Inflation 1.50665 0.35569 4.236 8.40e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01411 on 57 degrees of freedom  
## Multiple R-squared: 0.3995, Adjusted R-squared: 0.3784   
## F-statistic: 18.96 on 2 and 57 DF, p-value: 4.881e-07

**Regressionsbewertung:** Unabhaengige Variablen von Modell 12 sind nicht signifikant, Multiple R-squared: 0.294, Adjusted R-squared: 0.2692. Unabhaengige Variablen von Modell 15 sind signifikant, Multiple R-squared: 0.3995, Adjusted R-squared: 0.3784.

Modelle bewerten: (1) Residuenplots (die Variablen wurden schon ueberprueft)

1. Fehlerkennzahlen (fuer Trainings-Set):

# Data Frame erweitern  
evaluation = rbind(evaluation, data.frame(Model = c("Model12","Model15"),  
 MAE = numeric(1),  
 MAPE = numeric(1),  
 R\_Squared = numeric (1),  
 R\_Squared\_adj = numeric(1)))  
  
# MAE berechnen  
evaluation[evaluation$Model == "Model12",]$MAE = mean(abs(model\_12$residuals))  
evaluation[evaluation$Model == "Model15",]$MAE = mean(abs(model\_15$residuals))  
  
# MAPE berechnen  
evaluation[evaluation$Model == "Model12",]$MAPE = mape(model\_12$model$IFR, model\_12$fitted.values)  
evaluation[evaluation$Model == "Model15",]$MAPE = mape(model\_15$model$IFR, model\_15$fitted.values)  
  
  
  
# R-Squared berechnen  
evaluation[evaluation$Model == "Model12",]$R\_Squared = summary(model\_12)$r.squared  
evaluation[evaluation$Model == "Model15",]$R\_Squared = summary(model\_15)$r.squared  
  
  
# R-Squared\_Adj berechnen  
evaluation[evaluation$Model == "Model12",]$R\_Squared\_adj = summary(model\_12)$adj.r.squared  
evaluation[evaluation$Model == "Model15",]$R\_Squared\_adj = summary(model\_15)$adj.r.squared  
  
  
# Fehler anzeigen  
evaluation

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.01426048 0.01748630   
## 2 Model1 0.01194009 0.01464587 0.21041778092297 0.196804294387159  
## 3 Model2 0.01266202 0.01554046 0.219549710290505 0.206093670812755  
## 4 Model3 0.01251803 0.01534934 0.139410926633964 0.124573183989722  
## 5 Model4 0.01249369 0.01533179 0.118543236061212 0.103345705648475  
## 6 Model5 0.01293505 0.01586560 0.173543889806034 0.159294646526827  
## 7 Model12 0.01199177 0.01472630 0.293954611751277 0.269181089356585  
## 8 Model15 0.01094432 0.01344503 0.399456709068252 0.378385014649594

Alle Modelle weisen verbesserte Fehlerkennzahlen im Vergleich zu Modell 1 auf. Modell 15 dominiert in allen 4 Bewertungskriterien. Es wird mit diesem Modell fortgefahren.

#### Iteration 3

Trivariate Modelle erstellen:

#Modelle mit Trainings-Sets erstellen  
model\_152 = lm(IFR ~ ParkingSpaceAvailability + Inflation + RoadCondition, data = AHL\_externals\_training)  
  
#Modelle anzeigen  
summary(model\_152)

##   
## Call:  
## lm(formula = IFR ~ ParkingSpaceAvailability + Inflation + RoadCondition,   
## data = AHL\_externals\_training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.033313 -0.007657 0.001376 0.009761 0.028636   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.7940121 0.0120721 65.773 < 2e-16 \*\*\*  
## ParkingSpaceAvailability 0.6628201 0.1838636 3.605 0.000665 \*\*\*  
## Inflation 1.4153020 0.4481502 3.158 0.002559 \*\*   
## RoadCondition -0.0005317 0.0015655 -0.340 0.735394   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01422 on 56 degrees of freedom  
## Multiple R-squared: 0.4007, Adjusted R-squared: 0.3686   
## F-statistic: 12.48 on 3 and 56 DF, p-value: 2.335e-06

**Regressionsbewertung:** Bei Modell 152 hat RoadCondition keine Signifikanz mehr. Die R^2 und R^2 adjusted Werte sind Multiple R-squared: 0.4007, Adjusted R-squared: 0.3686.

Modelle bewerten: (1) Residuenplots (Variablen wurden schon ueberprueft)

1. Fehlerkennzahlen (fuer Trainings-Set):

# Data Frame erweitern  
evaluation = rbind(evaluation, data.frame(Model = c("Model152"),  
 MAE = numeric(1),  
 MAPE = numeric(1),  
 R\_Squared = numeric (1),  
 R\_Squared\_adj = numeric(1)))  
  
# MAE berechnen  
evaluation[evaluation$Model == "Model152",]$MAE = mean(abs(model\_152$residuals))  
  
# MAPE berechnen  
evaluation[evaluation$Model == "Model152",]$MAPE = mape(model\_152$model$IFR, model\_12$fitted.values)  
  
  
  
# R-Squared berechnen  
evaluation[evaluation$Model == "Model152",]$R\_Squared = summary(model\_152)$r.squared  
  
  
# R-Squared\_Adj berechnen  
evaluation[evaluation$Model == "Model152",]$R\_Squared\_adj = summary(model\_152)$adj.r.squared  
  
  
# Fehler anzeigen  
evaluation

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.01426048 0.01748630   
## 2 Model1 0.01194009 0.01464587 0.21041778092297 0.196804294387159  
## 3 Model2 0.01266202 0.01554046 0.219549710290505 0.206093670812755  
## 4 Model3 0.01251803 0.01534934 0.139410926633964 0.124573183989722  
## 5 Model4 0.01249369 0.01533179 0.118543236061212 0.103345705648475  
## 6 Model5 0.01293505 0.01586560 0.173543889806034 0.159294646526827  
## 7 Model12 0.01199177 0.01472630 0.293954611751277 0.269181089356585  
## 8 Model15 0.01094432 0.01344503 0.399456709068252 0.378385014649594  
## 9 Model152 0.01099502 0.01472630 0.400691277361969 0.368585452934932

Modell 152 ist in den Fehlerkennzahlen und R^2 adjusted leicht schlechter als Model 15. Nur in dem Bewertungskriterium R^2 ist das Modell marginal besser als Model 15. Daher wird sich für das Model 15 entschieden und die Wrappermethode damit beendet.

#### Test auf Overfitting

# Vorhersage treffen  
pred15 = predict(model\_15, newdata = AHL\_externals\_test)  
  
# Data Frame erweitern  
evaluation = rbind(evaluation, data.frame(Model = "Model15\_test",  
 MAE = numeric(1),  
 MAPE = numeric(1),  
 R\_Squared = numeric (1),  
 R\_Squared\_adj = numeric(1)))  
  
# MAE berechnen  
evaluation[evaluation$Model == "Model15\_test",]$MAE = mean(abs(AHL\_externals\_test$IFR -pred15))  
  
# MAPE berechnen  
evaluation[evaluation$Model == "Model15\_test",]$MAPE = mape(AHL\_externals\_test$IFR, pred15)  
  
# R-Squared berechnen  
evaluation[evaluation$Model == "Model15\_test",]$R\_Squared = ""  
  
# R-Squared\_Adj berechnen  
evaluation[evaluation$Model == "Model15\_test",]$R\_Squared\_adj = ""  
  
evaluation$MAE = round(evaluation$MAE, 3)  
evaluation$MAPE = round(evaluation$MAPE, 3)  
evaluation$R\_Squared = round(as.numeric(evaluation$R\_Squared), 3)  
evaluation$R\_Squared\_adj = round(as.numeric(evaluation$R\_Squared\_adj), 3)  
  
# Fehler anzeigen  
evaluation[c(1,2,8,10),]

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.014 0.017 NA NA  
## 2 Model1 0.012 0.015 0.210 0.197  
## 8 Model15 0.011 0.013 0.399 0.378  
## 10 Model15\_test 0.010 0.012 NA NA

Da die Fehlerkennzahlen sich nicht verschlechtert haben, scheint kein Overfitting vorzuliegen.

**Fazit:** Nach 3 Iterationen ist das Modell 15 als bestes Modell hervorgegangen. Wir konnten unsere Fehlerkennzahlen gegenueber der Baseline stark verbessern und konnten ein Bestimmtheitsmass von ca. 0.4 erreichen.

### (11)

Bewerten Sie ihr Modell quantitativ im Vergleich mit der Baseline. Bewertungsrelevant: Output, Kommentar.

#Nur die Bewertung der Baseline und Model 15 ausgeben:  
final\_evaluation = evaluation[c(1,8),]  
final\_evaluation

## Model MAE MAPE R\_Squared R\_Squared\_adj  
## 1 Baseline 0.014 0.017 NA NA  
## 8 Model15 0.011 0.013 0.399 0.378

cat("Das Model 15 reduziert den MAE um",round((1-final\_evaluation[2,2]/final\_evaluation[1,2])\*100,1),"% gegenüber der Baseline.\n")

## Das Model 15 reduziert den MAE um 21.4 % gegenüber der Baseline.

cat("Der MAPE wird um",round((1-final\_evaluation[2,3]/final\_evaluation[1,3])\*100,1),"% gegenüber der Baseline verringert.\n")

## Der MAPE wird um 23.5 % gegenüber der Baseline verringert.

cat("Mit einem R^2-Wert von",round(final\_evaluation[2,4], 2),"ist die Aussagekraft des Modells außerdem akzeptabel.")

## Mit einem R^2-Wert von 0.4 ist die Aussagekraft des Modells außerdem akzeptabel.

Kommentar: Die Fehlerkennzahlen MAE und MAPE konnten

### (12)

Ihre Chefin kommt auf der Firmenfeier zu Ihnen und schlaegt Ihnen eine Wette vor. Sie sagt: “Ich wette mit Ihnen um 100 Euro, dass die durchschnittliche IFR des oben betrachteten WH‐DL im Dezember 2021 in jeder Region ueber 0.85 sein wird.” Sollten Sie die Wette eingehen? Bewertungsrelevant: Output, Kommentar.

#Dataframe anlegen:  
IFR\_Dez12 = data.frame(Region = "",  
 ParkingSpaceAvailability = numeric(1),  
 Inflation = numeric(1),  
 predicted\_IFR = numeric(1))  
#ParkingSpaceAvailability für jede Region in Periode 2021/12 ins Dataframe hinzufügen:  
IFR\_Dez12[1:5,1:2] = aggregate(ParkingSpaceAvailability ~ region, data = externals[externals$Period=="2021/12",],mean)  
#Inflation für jede Region in Periode 2021/12 ins Dataframe hinzufügen:  
IFR\_Dez12[,3] = aggregate(Inflation ~ region, data = externals[externals$Period=="2021/12",],mean)[,2]  
#Vorhersage basierend auf Model 15 berechnen:  
IFR\_Dez12[,4] = predict(model\_15,IFR\_Dez12[,2:3])  
#Dataframe formatieren:  
rownames(IFR\_Dez12) = NULL  
IFR\_Dez12$predicted\_IFR = round(IFR\_Dez12$predicted\_IFR,3)  
IFR\_Dez12

## Region ParkingSpaceAvailability Inflation predicted\_IFR  
## 1 Japan 0.03147873 0.00106059 0.814  
## 2 Peking 0.05114552 -0.00140027 0.824  
## 3 Phlppn 0.02178779 0.00082554 0.807  
## 4 Shangh 0.01768709 0.00136760 0.805  
## 5 Skorea 0.03223893 0.01320449 0.833

Kommentar: Laut des von uns erstellten Modells, ist die IFR in allen Regionen in der Periode 2021/12 unter 0,85. Deshalb würden wir die Wette nicht eingehen.

### (13)

Ihr Regressionsmodell soll im kommenden Jahr implementiert und langfristig in die Unternehmensprozesse integriert werden. Beschreiben Sie, welche Nutzer und Prozesse davon profitieren koennten und in welcher Form die Loesung bereitgestellt werden koennte. Nehmen Sie ausserdem ausfuehrlich zur Phase der Datenbeschaffung Stellung. Bewertungsrelevant: Kommentar.

Kommentar: Es könnte ein Dashboard geben, auf dem Live die Vorhersage für die IFR angezeigt wird. Mithilfe der Vorhersagen könnte die IFR aus Perspektive des letzten Glieds der Lieferkette gegen 1 gebracht werden, indem am Anfang der Lieferkette mehr losgeschickt wird als bestellt wird. Für die Vorhersage der IFR werden Vorhersagen der Inflation und Parkplatzverfügbarkeit benötigt. Für eine Prognose der Inflation gibt es viele gut zugänglichen Quellen im Internet (siehe: <https://de.statista.com/statistik/daten/studie/5851/umfrage/prognose-zur-entwicklung-der-inflationsrate-in-deutschland/>). Für die Prognose der Parkplatzverfügbarkeit sieht das jedoch schwieriger aus. Es gibt bereits Anbieter die mithilfe von Künstlicher Intelligenz diesen Einfluss vorhersagen (siehe: <https://gobeta.de/projekte/aipark-artificial-intelligence-based-parking/>). Es wäre eine Kooperation in Form der Bereitstellung einer API denkbar.