# Gruppe 105 Hausaufgabe 2

Supply Chain Analytics SS23

Cordelia Mena Hernandez, Daniel Glatter

2023-05-24

## Daten vorbereiten

```
# Laden der Daten in die Dataframes 'cost', 'services', 'prices' und 'transactions'

cost <- read.csv("../HA1/data/output_cost_8Players_v0020.csv", sep=";", dec=",")
services <- read.csv("../HA1/data/output_services_8Players_v0020.csv", sep=";", dec=",")
prices <- read.csv("../HA1/data/output_prices_8Players_v0020.csv", sep=";", dec=",")
transactions <- read.csv("../HA1/data/output_transactions_8Players_v0020.csv", sep=";",
↪  dec=",")

# Aufbereiten der Daten (Filtern nach Jahr und Umwandlung ins Datumsformat)
year18_22 <- interval(ymd("2018-01-01"), ymd("2022-12-31"))

cost$Date <- make_date(cost$Year, cost$Month)
cost <- subset(cost, select=-c(Year, Month))
cost <- cost[cost$Date %within% year18_22, ]

services$Date <- make_date(services$Year, services$Month, services$Day)
services <- subset(services, select=-c(Year, Month, Day))
services <- services[services$Date %within% year18_22, ]


transactions$Date <- make_date(transactions$Year, transactions$Month, transactions$Day)
transactions <- subset(transactions, select=-c(Year, Month, Day))
transactions <- transactions[transactions$Date %within% year18_22, ]

# Characters in Factors umwandeln
cost$Product = as.factor(cost$Product)
services$region = as.factor(services$region)
services$storename = as.factor(services$storename)
services$Product = as.factor(services$Product)
services$vendor = as.factor(services$vendor)
services$service = as.factor(services$service)
prices$vendor = as.factor(prices$vendor)
prices$service = as.factor(prices$service)
transactions$region = as.factor(transactions$region)
transactions$storename = as.factor(transactions$storename)
transactions$Product = as.factor(transactions$Product)
```

## Aufgabe 1

```
# Daten nach Monat zusammenfassen
Demand <- transactions %>%
    arrange(Date) %>%
    # Periodenspalte mit Datentyp yearmon aus dem Zoo package für monatliche Daten
    group_by(region, Period = as.yearmon(Date)) %>%
  summarise(Demand = sum(Sales))

head(Demand) %>% kable(caption='Verkaufszahlen aggregiert nach Region und Periode
↪  (Monat/Jahr)')
```

Table 1: Verkaufszahlen aggregiert nach Region und Periode (Monat/Jahr)

| region | Period | Demand |
|---|---|---|
| Japan | Jan 2018 | 15783 |
| Japan | Feb 2018 | 18062 |
| Japan | Mrz 2018 | 19384 |
| Japan | Apr 2018 | 19611 |
| Japan | Mai 2018 | 18676 |
| Japan | Jun 2018 | 16654 |

## Aufgabe 2

```
# reshape() funktioniert nicht mit dem groupby() aus dem Tidyverse-Paket. Wir verwenden
↪  daher stattdessen die im Paket integrierte pivot_wider() Funktion.
# Die Syntax für die reshape() Funktion wäre hier:
# reshape(Demand, timevar = 'region', idvar = 'Period', direction = 'wide')

Demand_wide <- pivot_wider(data = Demand, id_cols = "Period", names_from = "region",
↪  values_from = "Demand", names_prefix = "Demand in ")

head(Demand_wide) %>% kable(caption='Verkaufszahlen aggregiert nach Region und Periode,
↪  umgewandelt in das Wide-Format')
```

Table 2: Verkaufszahlen aggregiert nach Region und Periode, umgewandelt in das Wide-Format

| Period | Demand in Japan | Demand in Peking | Demand in Phlppn | Demand in Shangh | Demand in Skorea |
|---|---|---|---|---|---|
| Jan 2018 | 15783 | 16166 | 15162 | 19973 | 15894 |
| Feb 2018 | 18062 | 16268 | 14337 | 20901 | 20290 |
| Mrz 2018 | 19384 | 16488 | 11234 | 17728 | 17964 |
| Apr 2018 | 19611 | 18674 | 15106 | 17113 | 17459 |
| Mai 2018 | 18676 | 16100 | 13991 | 13889 | 17329 |
| Jun 2018 | 16654 | 15151 | 17371 | 18306 | 15989 |

### Aufgabe 3

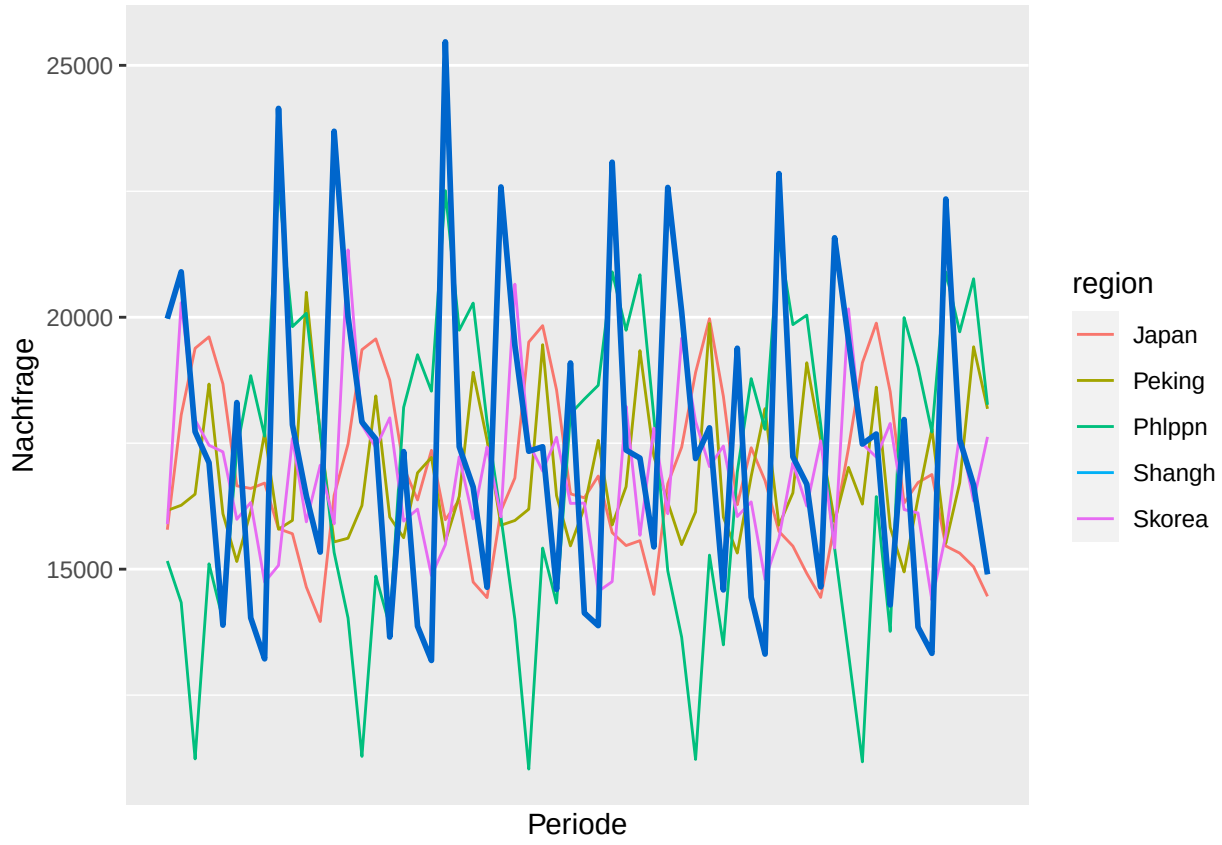
```
ts_Japan = ts(Demand_wide$`Demand in Japan`, frequency = 12)

ts_Peking = ts(Demand_wide$`Demand in Peking`, frequency = 12)
ts_Phlppn = ts(Demand_wide$`Demand in Phlppn`, frequency = 12)
ts_Shangh = ts(Demand_wide$`Demand in Shangh`, frequency = 12)
ts_Skorea = ts(Demand_wide$`Demand in Skorea`, frequency = 12)
```

## Modellierung vorbereiten

### Aufgabe 4

```
ggplot(data=Demand, aes(x=Period, y=Demand)) +
  geom_line(aes(col = region)) +
  geom_line(data = filter(Demand, region == "Shangh"), size = 1, col="#0066CC") +
  xlab("Periode") +
  ylab("Nachfrage") +
  theme(panel.grid.major.x = element_line(color = "red", size = 0.3, linetype = 2)) +
  scale_x_continuous(breaks = seq(12, 60, by = 12))
```




### Aufgabe 5

Man kann an der Zeitreihe für die Region Shanghai eine Saisonalität erkennen, es lässt sich jedoch kein Trend ausmachen. Da der Verlauf nicht rein stochastisch ist, ist die Zeitreihenanalyse eine sinnvolle Methode

zur Vorhersage der Nachfrage. Wir sollten allerdings von einfachen Methoden wie einem simplen Moving Average-Verfahren absehen, da dies Saisonalität nicht aufreichend berücksichtigen kann. Stattdessen bietet sich das Holt-Winter-Modell für die Saison-korrigierte exponentielle Glättung an. Eine wichtige Annahme zur erfolgreichen Anwendung dieses Modells zur Nachfragevorhersage ist, dass sich die Nachfrage auch in Zukunft so entwickelt wie bislang, dass also keine signifikanten Änderungen der Saisonalität oder Entstehung von Trends auftreten.

# Modellierung

## Aufgabe 6

```
# Modell automatisch erstellen lassen
m_Shangh = ets(ts_Shangh, model = "ZZZ")

# Ausgabe des Modells
m_Shangh
```

```
## ETS(M,N,A)
##
## Call:
##  ets(y = ts_Shangh, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 3e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 17513.7439
##     s = -2550.171 -825.6737 -35.6933 6286.551 -4207.033 -3467.201
##           935.6788 -3281.657 24.7882 45.5573 2465.96 4608.893
##
##   sigma:  0.0337
##
##      AIC     AICc      BIC
## 1023.398 1034.307 1054.813
```

```
#Urspüngliche Zeitreihe
m_Shangh$x
```

```
##     Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1 19973 20901 17728 17113 13889 18306 14034 13220 24145 17866 16455 15339
## 2 23691 20015 17920 17582 13653 17334 13867 13188 25466 17429 16630 14641
## 3 22586 19446 17340 17429 14599 19092 14130 13879 23077 17364 17204 15442
## 4 22576 20109 17196 17805 14588 19385 14442 13314 22851 17230 16686 14648
## 5 21577 19548 17483 17682 14290 17969 13851 13329 22346 17571 16666 14898
```

```
#Residuen anzeigen
m_Shangh$residuals
```

```
##             Jan           Feb           Mar           Apr           May
## 1 -0.0971691184  0.0461501698  0.0096315711 -0.0242425160 -0.0240717391
## 2  0.0709269827  0.0017541972  0.0205322484  0.0024663227 -0.0407048303
## 3  0.0209476172 -0.0267270006 -0.0124927860 -0.0062380484  0.0257974407
## 4  0.0204704023  0.0064352306 -0.0207316185  0.0151631553  0.0249642915
```

```
## 5 -0.0246983842 -0.0216372720 -0.0043674816  0.0081604702  0.0040413105
##             Jun           Jul           Aug           Sep           Oct
## 1 -0.0077401644 -0.0008449316 -0.0064656311  0.0145134437  0.0222308868
## 2 -0.0604616525 -0.0127607628 -0.0088933600  0.0700028804 -0.0028188478
## 3  0.0348375254  0.0059267207  0.0429892956 -0.0304190935 -0.0065395702
## 4  0.0506695168  0.0280726594  0.0004501824 -0.0399429882 -0.0142460489
## 5 -0.0260653406 -0.0139429105  0.0016558144 -0.0611142381  0.0053345285
##             Nov           Dec
## 1 -0.0139392797  0.0251261311
## 2 -0.0034870394 -0.0215689826
## 3  0.0309080078  0.0319485278
## 4 -0.0001762257 -0.0211488848
## 5 -0.0013091659 -0.0043668018
```

## Aufgabe 7

```
# Die durchschnittliche Höhe der Originalwerte im Jahr 2020
mean(m_Shangh$x[25:36])
```

```
## [1] 17632.33
```

```
# Die durchschnittliche Höhe der Modellwerte im Jahr 2020
mean(m_Shangh$fitted[25:36])
```

```
## [1] 17513.88
```

Die Originalwerte liegen im Schnitt bei einer Nachfrage von ca. 17632 Flaschen, die Modellwerte bei ca. 17514, also leicht darunter. Bei einer Verwendung des Modells für das Jahr 2020 hätten wir die tatsächliche Nachfrage unterschätzt. Wenn wir entsprechend weniger produziert hätten (also nicht die Unsicherheit in der Analyse berücksichtigt und einen gewissen Sicherheitsbestand angelegt hätten), wäre die Nachfrage nicht bedienbar gewesen und wir hätten an erzielbarem Umsatz eingebüßt.

## Aufgabe 8

```
# Nachfragevorhersage für ein weiteres Jahr (12 Monate) der Variable m_Shangh
fcast_Shangh = forecast(m_Shangh, 12)

# Forecast ausgeben
fcast_Shangh
```

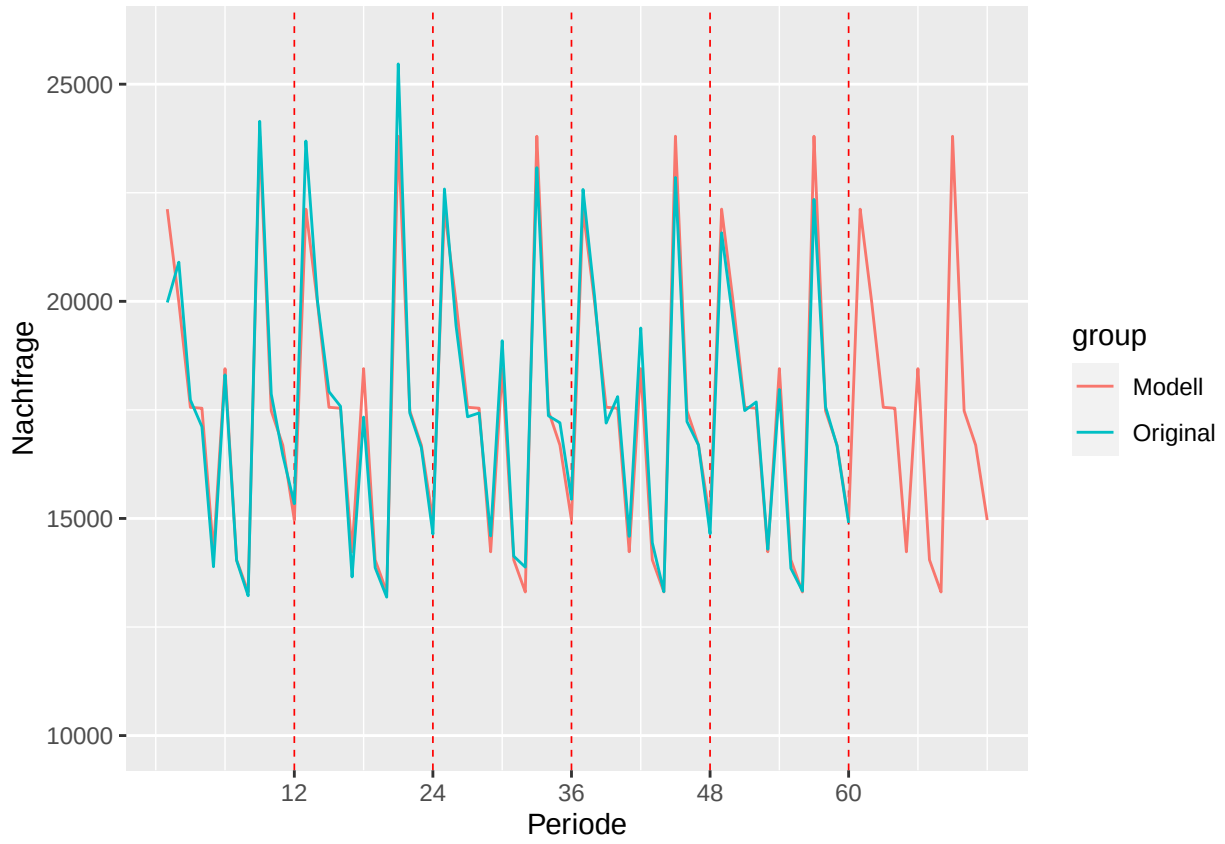
```
##       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 6       22122.33 21168.14 23076.52 20663.03 23581.63
## Feb 6       19979.44 19117.68 20841.20 18661.49 21297.38
## Mar 6       17559.00 16801.64 18316.36 16400.72 18717.29
## Apr 6       17538.24 16781.78 18294.70 16381.33 18695.15
## May 6       14231.79 13617.94 14845.64 13292.99 15170.59
## Jun 6       18449.12 17653.37 19244.87 17232.12 19666.12
## Jul 6       14046.27 13440.43 14652.12 13119.71 14972.84
## Aug 6       13306.48 12732.54 13880.42 12428.72 14184.24
## Sep 6       23799.87 22773.33 24826.41 22229.91 25369.83
## Oct 6       17477.78 16723.92 18231.63 16324.85 18630.70
## Nov 6       16687.81 15968.03 17407.60 15587.00 17788.63
## Dec 6       14963.31 14317.91 15608.71 13976.25 15950.37
```

```r
# DataFrame für Originaldaten erstellen
df_Shangh_orig = data.frame(
  period = seq(1, length(fcast_Shangh$x), 1),
  demand = as.numeric(fcast_Shangh$x),
  group = rep("Original", length(fcast_Shangh$x)))

# DataFrame für Forecast erstellen (sowohl Vorhersage für Zeitreihenwerte als auch ein
↪  Jahr in die Zukunft)
df_Shangh_fcast = data.frame(
  period = seq(1, length(fcast_Shangh$fitted)+length(fcast_Shangh$mean), 1),
  demand = c(as.numeric(fcast_Shangh$fitted), as.numeric(fcast_Shangh$mean)),
  group = rep("Modell", length(fcast_Shangh$fitted)+length(fcast_Shangh$mean)))

# In ein großes DataFrame zusammenführen
df_Shangh = rbind(df_Shangh_orig, df_Shangh_fcast)

# Plot
ggplot(df_Shangh, aes(x = period, y = demand, ymin=10000, ymax=26000, colour = group)) +
  geom_line()+
  xlab("Periode") +
  ylab("Nachfrage")+
  theme(panel.grid.major.x = element_line(color = "red", size = 0.3, linetype = 2))+
  scale_x_continuous(breaks = seq(12, 60, by = 12))
```




Im dargestellen Plot sehen wir den Nachfrageverlauf (Original in Blau) ab Periode 1 (Januar 2018) und die Modellvorhersage (in rot) ab Periode 1 sowie von der letzten Periode noch ein Jahr in die Zukunft fortgesetzt.

In einem Linienchart lassen sich beide Kurven einfach miteinander vergleichen. sowie die Fortsetzung der Vorhersage im gleichen Plot mit darstellen.

## Aufgabe 9

Wir bewerten unser Modell mithilfe von vier wichtigen Kennzahlen: MFE, MAE, MSE und MAPE.

```
# Mean Forecast Error (MFE)
mean(as.numeric(fcast_Shangh$x - fcast_Shangh$fitted))
```

```
## [1] -13.66972
```

Der mittlere Vorhersagefehler beträgt ca. -13,7 [Flaschen].

```
# Mean Absolute Error (MAE)
mean(abs(as.numeric(fcast_Shangh$x - fcast_Shangh$fitted)))
```

```
## [1] 408.7134
```

Der mittlere absolute Vorhersagefehler beträgt ca. 409 [Flaschen].

```
# Mean Squared Error (MSE)
mean((as.numeric(fcast_Shangh$x - fcast_Shangh$fitted)^2))
```

```
## [1] 357221.5
```

Der mittlere quadrtierte Fehler beträgt ca. 357.221 und ist in der Einheit Flaschen$^2$, also hier nicht direkt, sondern nur im Vergleich mit anderen, auf den gleichen Daten trainierten Modellen interpretierbar. Über die Wurzel könnten wir den RMSE (Root Mean Squared Error) erhalten (beträgt hier ca. 598 [Flaschen]), der Ausreißer stärker gewichtet als der MAE.

```
# Mean Absolute Percentage Error (MAPE)
MAPE_Shangh <- mean(abs((as.numeric(fcast_Shangh$x -
↪  fcast_Shangh$fitted)/as.numeric(fcast_Shangh$x))*100))
MAPE_Shangh
```

```
## [1] 2.18124
```

Die mittlere absolute prozentuale Abweichung beträgt ca. 2,18 %.

Die verschiedenen Kennzahlen treffen unterschiedliche Aussagen über unser Modell. MFE berücksichtigt nicht das Vorzeichen des Fehlers und lässt damit kaum Aussagen über die Güte des Modells zu (große positive und große negative Abweichungen könnten sich ausgleichen). Der Wert von -13,7 zeigt uns jedoch, dass das Modell im Schnitt die Nachfrage eher unterschätzt (siehe auch Aufgabe 7). MSE ist ohne Anwendung der Wurzelfunktion erstmal in einer unpassenden Einheit und gewichtet Ausreißer, also besonders große Abweichungen, höher. Das kann für manche Problemstellungen sehr relevant sein, hier konzentrieren wir uns aber auf MAE, welcher uns eine Kennzahl in einer direkt nützlichen Einheit liefert. Darüber hinaus liefert uns MAPE eine prozentuale Kennzahl unabhängig von der Größenordnung der Nachfrage, was besonders dann nützlich ist, wenn sich die Größenordnung der Nachfrage in verschiedenen Regionen sehr stark unterscheidet. Zur Bewertung der Güte unseres Modells eignen sich also vor allem MAE und MAPE. MAE

## Aufgabe 10

Wir hatten vermutet, dass die Nachfrage saisonal schwankt und kein Trend vorliegt. Das gefundene Modell liefert uns die Parameter alpha = 3e-04 und gamma = 1e-04.

Alpha ist der Parameter für die exponentielle Glättung, Gamma für die Saisonalität, die hier durch das Modell identifiziert wurde. Der Parameter Beta fehlt allerdings (bzw. ist 0), das Modell hat also wie vermutet keinen Trend ausfindig gemacht.

## Aufgabe 11

```
# Erstellung des Modells nach Aussage von Matthias
m_ana_Shangh = ets(ts_Shangh, model = "ANA")

# Ausgabe des Modells
m_ana_Shangh
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ts_Shangh, model = "ANA")
##
##   Smoothing parameters:
##     alpha = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 17515.0437
##     s = -2566.409 -834.8293 -49.7546 6310.159 -4175.847 -3469.876
##           905.2245 -3260.531 82.4492 29.2639 2444.744 4585.405
##
##   sigma:  683.7514
##
##      AIC     AICc      BIC
## 1043.030 1053.939 1074.445
```

```
# Nachfragevorhersage für ein weiteres Jahr der Variable m_Shangh (Exponentielles
↪  Glättungsmodell)
fcast_Shangh1 = forecast(m_ana_Shangh, 12)

# Forecast ausgeben
fcast_Shangh1
```

```
##       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 6       22100.35 21224.09 22976.61 20760.22 23440.48
## Feb 6       19959.72 19083.46 20835.98 18619.59 21299.85
## Mar 6       17544.21 16667.95 18420.48 16204.08 18884.34
## Apr 6       17597.37 16721.10 18473.63 16257.24 18937.49
## May 6       14254.40 13378.14 15130.66 12914.27 15594.53
## Jun 6       18420.18 17543.91 19296.44 17080.05 19760.31
## Jul 6       14045.09 13168.83 14921.35 12704.96 15385.22
## Aug 6       13339.13 12462.87 14215.39 11999.00 14679.26
## Sep 6       23824.99 22948.73 24701.25 22484.86 25165.12
## Oct 6       17465.21 16588.95 18341.48 16125.08 18805.34
## Nov 6       16680.15 15803.89 17556.41 15340.02 18020.28
## Dec 6       14948.57 14072.31 15824.83 13608.44 16288.70
```

```
# MAE
mean(abs(as.numeric(fcast_Shangh1$x - fcast_Shangh1$fitted)))
```

```
## [1] 408.2071
```

```
# MAPE
mean(abs((as.numeric(fcast_Shangh1$x -
↪  fcast_Shangh1$fitted)/as.numeric(fcast_Shangh1$x))*100))
```

```
## [1] 2.177524
```

Im neuen Modell "ANA" liegt sowohl MAE mit ca. 408,21 (vs. 408,71 beim alten Modell) als auch MAPE mit 2,1775 % (vs. 2,1812 %) etwas niedriger, also besser. Matthias hat also Recht, dass "ANA" besser performt als "MNA" (das von R gefundene optimale Modell, in dem der Fehler multiplikativ statt additiv eingeht). Die Abweichungen sind allerdings sehr gering. Grund für die Abweichungen ist, dass das forecast-Paket standardmäßig MSE minimiert, nicht MAE oder MAPE.

## Aufgabe 12

```
# Modell automatisch erstellen lassen für Japan
m_Japan = ets(ts_Japan, model = "ZZZ")
# Nachfragevorhersage für ein weiteres Jahr der Variable m_Shangh (Exponentielles
↪  Glättungsmodell)
fcast_Japan = forecast(m_Japan, 12)

# Modell und Forecast für Peking
m_Peking = ets(ts_Peking, model = "ZZZ")
fcast_Peking = forecast(m_Peking, 12)

# Modell und Forecast für Philippinen
m_Phlppn = ets(ts_Phlppn, model = "ZZZ")
fcast_Phlppn = forecast(m_Phlppn, 12)

# Modell und Forecast für Südkorea
m_Skorea = ets(ts_Skorea, model = "ZZZ")
fcast_Skorea = forecast(m_Skorea, 12)

# MAPE für Japan
MAPE_Japan <- mean(abs((as.numeric(fcast_Japan$x -
↪  fcast_Japan$fitted)/as.numeric(fcast_Japan$x))*100))
# MAPE für Peking
MAPE_Peking <- mean(abs((as.numeric(fcast_Peking$x -
↪  fcast_Peking$fitted)/as.numeric(fcast_Peking$x))*100))
# MAPE für Philippinen
MAPE_Phlppn <- mean(abs((as.numeric(fcast_Phlppn$x -
↪  fcast_Phlppn$fitted)/as.numeric(fcast_Phlppn$x))*100))
# MAPE für Südkorea
MAPE_Skorea <- mean(abs((as.numeric(fcast_Skorea$x -
↪  fcast_Skorea$fitted)/as.numeric(fcast_Skorea$x))*100))

MAPE_table <- data.frame(MAPE=rbind(MAPE_Japan, MAPE_Peking, MAPE_Phlppn, MAPE_Skorea,
↪  MAPE_Shangh), row.names=c("Japan", "Peking", "Phlppn", "Skorea", "Shangh"))

MAPE_table %>% kable(caption='Mean Absolute Percentage Error (MAPE) für die verschiendene
↪  Regionen', digits=2)
```

Table 3: Mean Absolute Percentage Error (MAPE) für die verschiendene Regionen

| | MAPE |
|---|---|
| Japan | 1.33 |
| Peking | 1.55 |
| Phlppn | 1.88 |
| Skorea | 1.33 |
| Shangh | 2.18 |

Japan und Südkorea haben den niedrigsten MAPE-Wert, laut dem Bewertungsmaß sind diese die "besten" Modelle. Allerdings sind die Modelle nur bedingt vergleichbar, da sie jeweils auf unterschiedlichen Daten trainiert wurden. Ggf. ist etwa die Nachfrage in Shanghai stochastischer als in Japan, also prinzipiell schwieriger vorherzusagen. Wir können aber daraus folgern, dass wir uns in Japan oder Südkorea eher auf das Vorhersagemodell verlassen würden als beispielsweise in Shanghai oder den Philippinen.

# Abschluss

## Aufgabe 13

```
# Summe der Vorhersagen für April, Mai, Juni 2023 über alle Regionen
sum(c(fcast_Japan$mean[4:6], fcast_Peking$mean[4:6], fcast_Phlppn$mean[4:6],
↪  fcast_Shangh$mean[4:6], fcast_Skorea$mean[4:6]))
```

```
## [1] 254133.6
```

Das Modell sagt für das zweite Quartal 2023 (Monate April, Mai, Juni) über alle fünf Regionen hinweg eine Nachfrage von ca. 254.134 Flaschen voraus. Die Anzahl benötigter Flaschen könnte davon abweichen, beispielsweise wenn wir uns dazu entscheiden, einen Sicherheitsbestand anzulegen.

## Aufgabe 14

```
# Durchschnittl. Nachfrage in Peking im Juli 2018-2021
mean(fcast_Peking$x[seq(7, 43, 12)])
```

```
## [1] 16531.25
```

```
# Nachfrage in Peking im Juli 2022
fcast_Peking$x[55]
```

```
## [1] 16407
```

```
# Vorhergesagte Nachfrage in Peking im Juli 2023
fcast_Peking$mean[7]
```

```
## [1] 16507.47
```

Frank hat Recht damit, dass im Juli in Peking üblicherweise mehr verkauft wurde (im Schnitt ca. 16531 Flaschen) als im Juli 2022 (nur 16407, also über 120 Flaschen weniger). Unser Modell sagt jedoch für den Juli 2023 wieder eine Nachfrage von ca. 16507 voraus, also fast wieder auf dem vorherigen Niveau.