

# SCA\_WS2021\_Gruppe105\_HA2

Wilke Klausning und Anh-Thu Tran

## Inhaltsverzeichnis

<b>Vorbereitung der Daten.....</b>	<b>2</b>
<i>Aufgabe 1) Aggregation der Verkaufszahlen je Monat je Region.....</i>	<i>2</i>
<i>Aufgabe 2) Umwandeln von Long-Format in das Wide-Format.....</i>	<i>2</i>
<i>Aufgabe 3) Umwandeln der aggregierten Verkaufszahlen in Datentyp time-series.....</i>	<i>3</i>
<b>Modellierung vorbereiten .....</b>	<b>4</b>
<i>Aufgabe 4) Visualisieren des Nachfrageverlaufs der Region Shanghai .....</i>	<i>4</i>
<i>Aufgabe 5) Stellungnahme und Annahme .....</i>	<i>5</i>
<b>Modellierung .....</b>	<b>5</b>
<i>Aufgabe 6) Zeitreihenanalyse von der Region Shanghai.....</i>	<i>5</i>
<i>Aufgabe 7) Durchschnittliche Abweichung der Modellwerte zu den Originalwerten .....</i>	<i>7</i>
<i>Aufgabe 8) Nachfragevorhersage fuer ein weiteres Jahr .....</i>	<i>7</i>
<i>Aufgabe 9) Bewertung des Modells aus Aufgabe 6 mit Hilfe von 4 verschiedenen Fehler-Kennzahlen.....</i>	<i>10</i>
<i>Aufgabe 10) Modellvergleich mit den Vermutungen aus Aufgabe 5.....</i>	<i>11</i>
<i>Aufgabe 11) Modelle zur Nachfragevorhersage fuer 4 weitere Regionen.....</i>	<i>11</i>
<b>Abschluss.....</b>	<b>13</b>
<i>Aufgabe 12) Untersuchung der Nachfrage nach Limonade in Peking, den Philippinen und Suedkorea fuer den naechsten Monat (Januar 2021) .....</i>	<i>13</i>
<i>Aufgabe 13) Berechnung der Nachfragevorhersage im ersten Quartal von 2021 in Shanghai fuer drei Supermaerkte. ....</i>	<i>14</i>

```
cost = read.csv2("output_cost_8Players_v0010.csv")
prices = read.csv2("output_prices_8Players_v0010.csv")
services = read.csv2("output_services_8Players_v0010.csv")
transactions = read.csv2("output_transactions_8Players_v0010.csv")
```

## Vorbereitung der Daten

### Aufgabe 1) Aggregation der Verkaufszahlen je Monat je Region

```
#Variable Periode (YYYY-MM) erzeugen
transactions$Periode = sprintf("%02d-%02d", transactions$Year,
                                transactions$Month)

#LONG-Format
Demand <- aggregate(list(Sales=transactions$Sales),
                     by=list(region=transactions$region, transactions$Periode), sum)

# Entfernen von Sales, die 0 beinhalten und Duplikate, da Sales nach der
# Aufgabenbeschreibung ueber den Tag aufsummiert sind
transactions = transactions[transactions$Sales != 0,] %>% unique(.)

# Tabellen werden benannt. Die Spalte "Sales" wird in "Demand" umbenannt.
colnames(Demand) = c("region", "Periode", "Demand")

# Der Tabellenkopf wird angezeigt
head(Demand)

##   region Periode Demand
## 1  Japan 2016-01  19118
## 2 Peking 2016-01  15026
## 3 Phlppn 2016-01  16184
## 4 Shangh 2016-01  20039
## 5 Skorea 2016-01  16624
## 6  Japan 2016-02  17172
```

### Aufgabe 2) Umwandeln von Long-Format in das Wide-Format.

```
# LONG-Format mit drei Spalten
Demand <- aggregate(list(Sales=transactions$Sales),
                     by=list(region=transactions$region,
                             Periode=transactions$Periode), sum)

# Entfernen von Sales, die 0 beinhalten und Duplikate
Demand <- data.frame(data = subset(Demand, Sales != 0)) %>% unique(.)

# Spaltennamen werden uebergeben
colnames(Demand) = c("Region", "Periode", "Demand")

# Umwandeln von aggregierten Demand-Daten vom Long-Format in das Wide-Format.
```

```

# Im WIDE-Format mit sechs Spalten erzeugt mittels der Reshape()-Funktion
Demand = reshape(Demand,
  idvar = "Periode", # Spalte bleibt unverändert
  timevar = "Region", # Spalte teilt sich in mehrere Spalten
  direction = "wide" # Typ
)

# Spaltennamen werden uebergeben
colnames(Demand) = c("Periode", "Demand in Japan", "Demand in Peking",
  "Demand in Phlppn", "Demand in Shangh", "Demand in Skorea")

# Tabellenkopf wird ausgegeben
head(Demand)

##      Periode Demand in Japan Demand in Peking Demand in Phlppn Demand in Shangh
## 1  2016-01          18761          15026          15351          19129
## 6  2016-02          16724          16118          15650          19395
## 11 2016-03          16853          15991          17876          16589
## 16 2016-04          12742          16540          17475          17743
## 21 2016-05          18026          15544          18818          16012
## 26 2016-06          16989          18175          18703          17763
##      Demand in Skorea
## 1              16435
## 6              16621
## 11             17348
## 16             16609
## 21             16496
## 26             16528

```

### Aufgabe 3) Umwandeln der aggregierten Verkaufszahlen in Datentyp time-series

```

# Datentyp umwandeln (typecast)
demand_ts_japan = ts(Demand$`Demand in Japan`, frequency = 12)
demand_ts_peking = ts(Demand$`Demand in Peking`, frequency = 12)
demand_ts_phlppn = ts(Demand$`Demand in Phlppn`, frequency = 12)
demand_ts_shangh = ts(Demand$`Demand in Shangh`, frequency = 12)
demand_ts_skorea = ts(Demand$`Demand in Skorea`, frequency = 12)

```

## Modellierung vorbereiten

### Aufgabe 4) Visualisieren des Nachfrageverlaufs der Region Shanghai

*# Visualisierung der Daten insgesamt 5 Jahre*

```
ggplot(data = Demand, aes(x = Periode, y = `Demand in Shangh`, group=1)) +  
  geom_line() +
```

*# Periodenbeschriftung von x-Achse um 90 Grad gedreht, Schriftgröße ist 7*  
`theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 7)) +`

*# Periode in 2 Jahrsabstaenden wiedergeben*

```
scale_x_discrete(breaks =  
  Demand$Periode[seq(1, length(Demand$Periode), by = 2)])
```

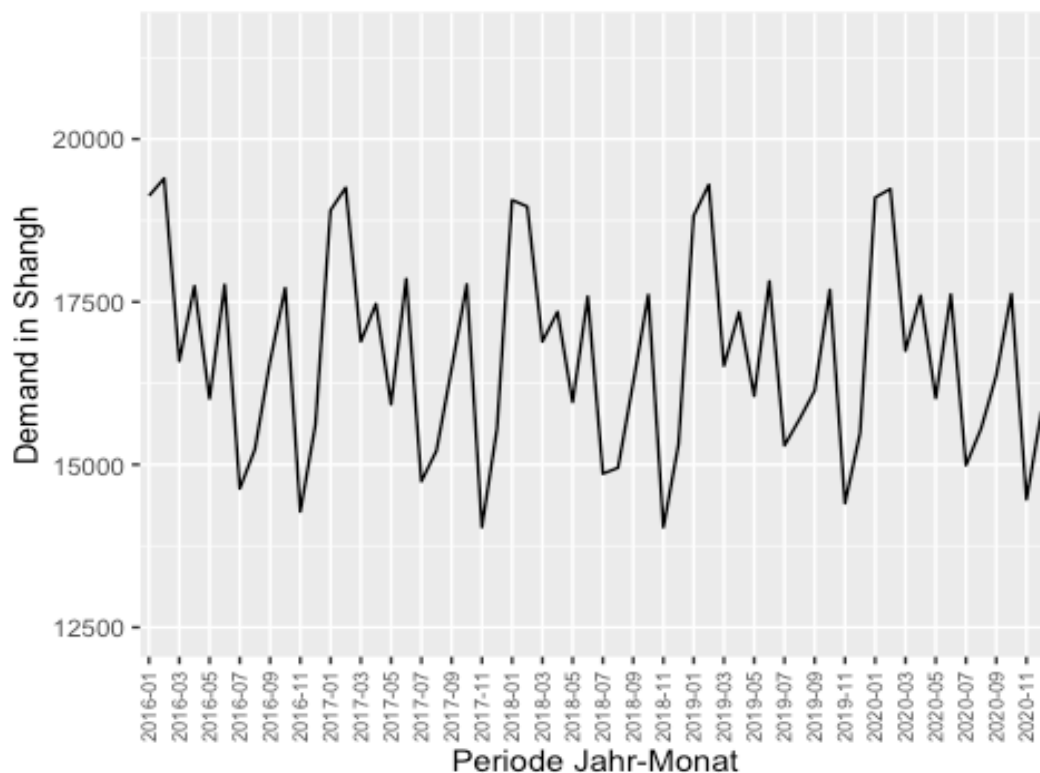
+

*# Titelbeschriftung der x-Achse*

```
xlab("Periode Jahr-Monat") +
```

*# Festgelegte GroeÙe fuer y-Achse*

```
ylim(12500, 21500)
```



*Begründung zu Aufgabe 4:* Wir haben uns fuer einen Liniendiagramm entschieden, da es am deutlichsten den Nachfrageverlauf in Shanghai darstellt. Fuer die x-Achse haben wir die Periode von Anfang 2016 bis Ende 2020 zu Grunde gelegt. Auf der y-Achse haben wie die

Nachfrage von 12500 bis 21500 eingegrenzt, um den Nachfrageverlauf besser zu verdeutlichen. Das Liniendiagramm zu wählen ist hier durchaus sinnvoll, da Trend und Saisonalitäten auf Monatsebene klar zu erkennen sind.

## Aufgabe 5) Stellungnahme und Annahme

*Stellungnahme Aufgabe 5:* In dem visualisierten Liniendiagramm wird ersichtlich, dass kein klarer Trend im Hinblick auf die Nachfrage in Shanghai vorliegt. Über die Saisonalitäten lässt sich ein wiederkehrendes Verhalten aufeinanderfolgender Periodenabschnitten in der x-Achse in den Monaten wiedererkennen. Beispielsweise erkennt man in den Monaten zwischen Januar und Februar, dass die Nachfrage ihren Hochpunkt erreicht und gegen Ende Februar wieder abnimmt. Der Tiefpunkt der Nachfrage befindet sich jedoch im November jeden wiederkehrenden Jahres.

*Annahme:* Um die Zeitreihenanalyse sinnvoll zu modellieren, müssen wir verschiedene Annahmen treffen. Da wir historische Nachfragewerte als Indikator für die zukünftige Nachfrage voraussetzen, sollte darauf geachtet werden, dass keine signifikante Veränderung im Nachfragemuster vorliegen. Außerdem ist es sinnvoll ein Liniendiagramm zu nehmen, da wir uns einen ersten guten Überblick über die Nachfrage verschaffen können. Um die Ergebnisse der Modellierung für unser Produkt sinnvoll zu nutzen, müssen wir annehmen, dass die Informationen quantifizierbar sind. Zudem ist es anzunehmen, dass das Vergangenheitsverhalten andauert. Jedoch müssen wir berücksichtigen, dass Nachfragevorhersagen immer ungenau sind und kurzfristige Vorhersagen genauer sind als Langfristige.

## Modellierung

### Aufgabe 6) Zeitreihenanalyse von der Region Shanghai

```
ets_shangh = ets(demand_ts_shangh, model = "ZZZ")
```

#### 1. Zusammenfassung des Modells

```
cat("\n Zusammenfassung des Modells: \n")  
  
##  
## Zusammenfassung des Modells:  
  
summary(ets_shangh)  
  
## ETS(A,N,A)  
##  
## Call:  
## ets(y = demand_ts_shangh, model = "ZZZ")
```

```
##
## Smoothing parameters:
##   alpha = 0.1302
##   gamma = 1e-04
##
## Initial states:
##   l = 16701.3532
##   s = -1170.822 -2466.762 1030.339 -290.4633 -1376.123 -1783.887
##       1048.143 -674.5363 793.8407 78.1582 2527.835 2284.278
##
## sigma: 184.7616
##
##      AIC      AICc      BIC
## 886.0064 896.9155 917.4216
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 8.817847 161.7763 128.155 0.04499874 0.7868164 0.6844818
##              ACF1
## Training set -0.001418694
```

## 2. Urspruengliche Zeitreihe

```
cat("\n Urspruengliche Zeitreihe: \n")
##
## Urspruengliche Zeitreihe:
ets_shangh$x
##      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1 19129 19395 16589 17743 16012 17763 14631 15235 16584 17713 14277 15616
## 2 18908 19254 16889 17470 15924 17852 14741 15215 16471 17768 14038 15550
## 3 19063 18963 16890 17346 15959 17582 14860 14951 16252 17608 14036 15315
## 4 18822 19299 16516 17342 16053 17817 15292 15699 16136 17684 14406 15487
## 5 19103 19234 16748 17596 16019 17614 14989 15554 16367 17622 14464 15818
```

## 3. Werte der Residuen

```
cat("\n Residuen: \n")
##
## Residuen:
ets_shangh$residuals
##      Jan      Feb      Mar      Apr      May      Jun
## 1 143.368571 147.146078 -228.335041 239.710579 -54.121601 -18.754558
## 2 -94.185441 20.520126 102.562680 -45.519876 -117.186973 103.387354
## 3 102.571498 -254.351255 155.470081 -124.485963 -26.864871 -123.072129
## 4 -29.507966 207.801120 -152.588648 -22.408861 159.912498 180.397310
## 5 118.439184 -9.537505 -44.594737 93.511719 -27.277485 -151.423448
```

##	Jul	Aug	Sep	Oct	Nov	Dec
## 1	-316.282516	-78.867825	194.740285	-22.416219	41.602672	79.246880
## 2	-189.013159	-98.192099	84.904574	50.069896	-189.354976	51.355089
## 3	3.040588	-314.151950	-57.956613	-15.187939	-88.092592	-93.590500
## 4	463.987089	402.813166	-298.362680	-32.298520	191.031838	-48.806236
## 5	75.339499	222.771822	-78.893334	-134.429686	222.175341	251.306516

## Aufgabe 7) Durchschnittliche Abweichung der Modellwerte zu den Originalwerten

```
# Durchschnittliche Abweichung mit Mean Forecast Error (MFE) berechnen
# Mittelwert(Urspruengliche Zeitreihe - Vorhersagen fuer den gegebenen Zeitraum)
MFE <- round(mean(as.numeric(ets_shangh$x - ets_shangh$fitted)), 4)
MFE
## [1] 8.8178
```

*Kommentar zu Aufgabe 7:* Die Modellwerte weichen durchschnittlich um 8.8178 von den Originalwerten ab. In unserem Unternehmen in der Data Analytics Abteilung rechnen wir mit der Fehlerkennzahl Mean Forecast Error (MFE). Das MFE gibt den durchschnittlichen Fehler unserer Modellwerte zu den Originalwerten an. Diese setzt sich aus drei verschiedenen Schritten. 1. Berechnung der Original-Nachfragewerte. 2. Berechnung der Vorhersage fuer den gegebenen Zeitraum. 3. Differenz aus 1. und 2. berechnen und den Mittelwert daraus bilden. Es gilt je naeher das MFE bei 0 ist, desto genauer ist die Vorhersage.

## Aufgabe 8) Nachfragevorhersage fuer ein weiteres Jahr

*Vorbereitung zu Aufgabe 8 + Zusammenfassung der vorherigen Aufgaben*

```
# Datentyp umwandeln (typecast)
demand_ts_shangh = ts(Demand$`Demand in Shangh`, frequency = 12)

# Zeitreihenanalyse aus Aufgabe 6
ets_shangh = ets(demand_ts_shangh, model = "ZZZ")

# Forecast der Variable fcast_Shangh (fuer exponentielle Glaettung) uebergeben
# Der Output zeigt die Forecast-Werte sowie die oberen und unteren Grenzen
# der 80% und 95% Konfidenzintervalle.
fcast_Shangh = forecast(ets_shangh, 12)
```

1. DataFrame Original Nr. 1 erstellen

```
# DataFrame erstellen
df_Shangh_orig = data.frame(
```

```

period = seq(1, length(fcast_Shangh$x), 1),
value = as.numeric(fcast_Shangh$x),
grp = rep("original", length(fcast_Shangh$x)))

```

## 2. DataFrame Vorhersage Nr. 2 erstellen

```

# DataFrame erstellen
df_Shangh_fcast = data.frame(
  period = seq(1, length(fcast_Shangh$fitted) + length(fcast_Shangh$mean), 1),
  value = c(as.numeric(fcast_Shangh$fitted), as.numeric(fcast_Shangh$mean)),
  grp = rep("fcast", length(fcast_Shangh$fitted) + length(fcast_Shangh$mean))
)

```

## 3. Verbinden der DataFrames Nr. 1 und Nr. 2

```

# Verbinden der Data Frames
df_EG = rbind(df_Shangh_orig, df_Shangh_fcast)

# Anzeigen des DataFrame (ausgewählte Beobachtungen)
head(df_EG)

##   period value      grp
## 1      1 19129 original
## 2      2 19395 original
## 3      3 16589 original
## 4      4 17743 original
## 5      5 16012 original
## 6      6 17763 original

```

## Schritt 4 - Visualisierung

```

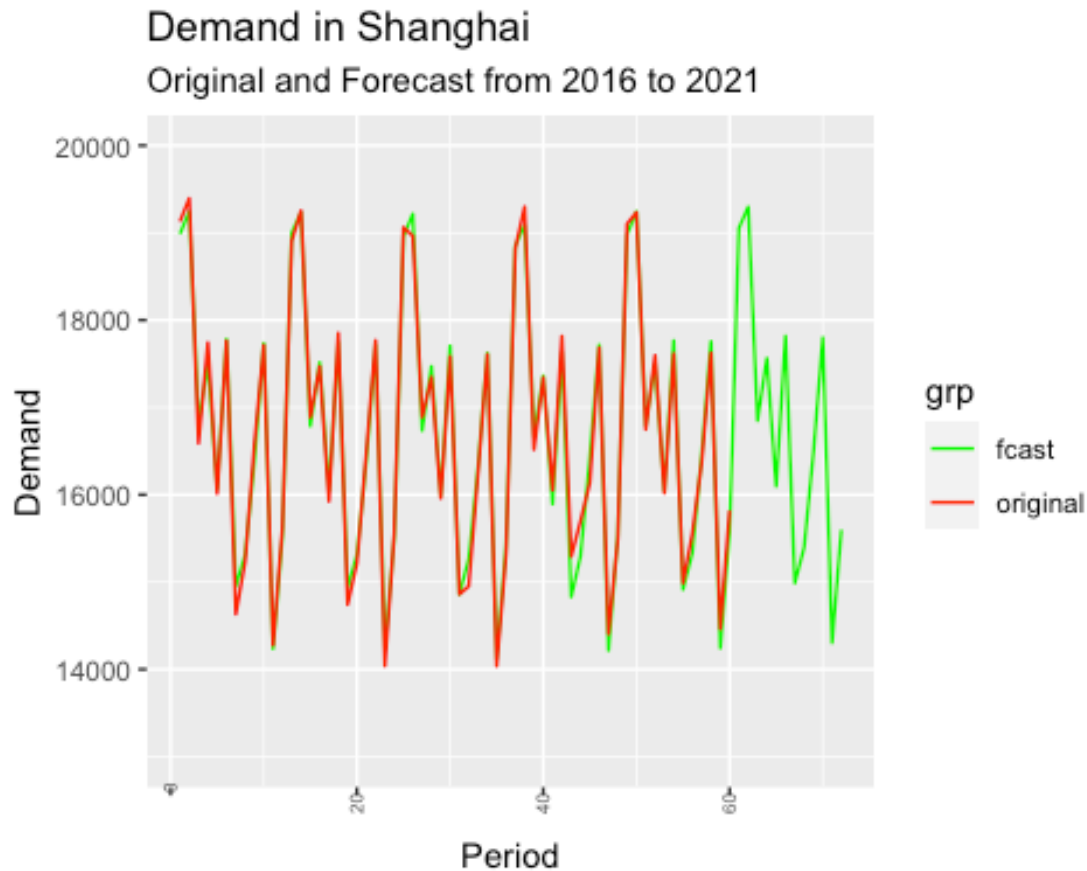
ggplot(data = df_EG, aes(x = period,
                        y = value, colour = grp)) +
  geom_line() +

  # Beschriftung von x-Achse 90 Grad umgedreht
  theme(axis.text.x = element_text(angle=90, vjust = 0.5, hjust=2, size=6)) +
  xlab("Period") +
  ylab("Demand") +
  ylim(13000, 20000) +
  scale_colour_manual(breaks = c("fcast", "original"),
                    values = c("green", "red")) +

  ggtitle(label = "Demand in Shanghai",
          subtitle = "Original and Forecast from 2016 to 2021")

```





*Begründung zu Aufgabe 8:* Um den Nachfrageverlauf sowie die Vorhersage am übersichtlichsten zu visualisieren, haben wir uns fuer einen Liniendiagramm entschieden. Die Linie der urspruenglichen Nachfrage haben wir in Kombination mit der Linie der Vorhersage in visualisierter Form uebereinander gelegt. Dies ermoeeglicht uns den direkten Vergleich beider Modelle. Auf der x-Achse umfasst die urspruengliche Nachfrage die Perioden 0 bis 60, wobei die Vorhersage der Nachfrage zusaetzlich ein weiteres Jahr umfasst, also die Perioden 0 bis 72. Auf der y-Achse findet sich die Anzahl der Nachfrage wieder, die wir jedoch von 13000 und 20000 eingeschreankt haben, um die Linien besser zu lesen. Hierbei wird der direkte Vergleich am deutlichsten indem die Farbe rot fuer die Vorhersage steht und blau fur die urspruengliche Nachfrage. Zudem koennen wir die Abstaende zwischen Tief- und Hochpunkt anhand der eingefaeerbten Punkte auf einem Liniendiagramm besser vergleichen als mit anderen Diagrammtypen.

## Aufgabe 9) Bewertung des Modells aus Aufgabe 6 mit Hilfe von 4 verschiedenen Fehler-Kennzahlen

### EG: (1) MFE

```
# MFE
MFE <- mean(as.numeric(ets_shangh$x - ets_shangh$fitted))
```

### EG: (2) MAE

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

### EG: (3) MSE

```
# MSE
MSE <- mean((as.numeric(ets_shangh$x - ets_shangh$fitted)^2))
```

### EG: (4) MAPE

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

cat("Es liegen folgende vier Fehlerkennzahlen vor. MFE betraegt", MFE,
    ", MAE betraegt", MAE, ", MAE betraegt", MSE,
    "und MAPE betraegt", MAPE)

## Es liegen folgende vier Fehlerkennzahlen vor. MFE betraegt 8.817847 , MAE
betraegt 128.155 , MAE betraegt 26171.57 und MAPE betraegt 0.7868164
```

*Kommentar zu Aufgabe 9:* Wir halten die Fehlerkennzahl MFE und MAPE als geeignete Fehlerkennzahlen. MFE wuerde Sinn machen, da der durchschnittliche Fehler ersichtlich wird. Man erkennt in der Berechnung, wie sich die Abweichung von der vorigen Periode zur naechsten Periode verhealt. MAPE koennte man ebenfalls in Betracht ziehen, da uns die prozentuale Fehler- zahl bei der Auswertung helfen koennte. Hierbei ist darauf zu achten, dass keine große Abweichung des y-Wertes bzw. Nachfrage zur vorigen Periode vorliegt. Zudem fokussieren sich viele Organisationen sich primaer auf MAPE, wenn es darum geht die Vorhersage Genauigkeit zu bestimmen.

## Aufgabe 10) Modellvergleich mit den Vermutungen aus Aufgabe 5

*Kommentar zu Aufgabe 10:* Beim Vergleich unseres vorherigen Modells und den Vermutungen aus Aufgabe 5 koennen wir feststellen, dass sich Saisonalitaeten wiedererkennen laesst. Ein klarer Trend liegt nicht eindeutig vor. Man kann erkennen, dass die hoechste Nachfrage zwischen Januar und Februar eintritt, und Ende Februar wieder abnimmt. Außerdem erkennt man, dass die niedrigste Nachfrage im Monat November vorkommt und sich jedes Jahr wiederholt. Die Vorhersage sagt aus, dass wir keine signifikante Abweichung zu den Vorjahren haben und somit genuegend Nachfrage herrscht. Zudem erkennt man Saisonalitaeten bzw. ein wiederkehrendes Verhaltensmuster im urspruenglichen Modell und Vorhersage Modell. Es gibt kleine Schwankungen zwischen beiden Modellen. Allgemein erachten wir die Zeitreihenanalyse hier durchaus als sinnvoll, da wir durch die historische Nachfrage als Indikator eine stabile zukuenftige Nachfrage identifiziert haben. Jedoch gilt es immer zu beruecksichtigen, dass die Vorhersage immer eine gewissen Ungenauigkeit hat. Zudem sind die Fehlerwerte gering. Somit kann die Vorhersage mit hoher Genauigkeit aussagen, welche Nachfrage in der Zukunft vorliegen wird.

## Aufgabe 11) Modelle zur Nachfragevorhersage fuer 4 weitere Regionen

*#Modelle für die übrigen Regionen erstellen.*

```
fcast_japan = forecast(ets(demand_ts_japan, model = "ZZZ"), 12)
fcast_peking = forecast(ets(demand_ts_peking, model = "ZZZ"), 12)
fcast_phlppn = forecast(ets(demand_ts_phlppn, model = "ZZZ"), 12)
fcast_skorea = forecast(ets(demand_ts_skorea, model = "ZZZ"), 12)
```

*#Funktion um MAPE zu berechnen*

```
mape <- function(actual,pred){
  mape <- round(mean(abs((actual - pred)/actual))*100, 4)
  return (mape)
}
```

*#MAPE ausgeben*

```
cat("Modell für Japan hat einen Wert für MAPE von",
    mape(demand_ts_japan, fcast_japan$fitted),".\n")
```

```
## Modell für Japan hat einen Wert für MAPE von 0.9121 .
```

```
cat("Modell für Peking hat einen Wert für MAPE von",
    mape(demand_ts_peking, fcast_peking$fitted),".\n")
```

```
## Modell für Peking hat einen Wert für MAPE von 0.9981 .
```

```
cat("Modell für Phillipinen hat einen Wert für MAPE von",  
    mape(demand_ts_phlppn, fcast_phlppn$fitted),".\n")  
## Modell für Phillipinen hat einen Wert für MAPE von 0.8036 .  
  
cat("Modell für Suedkorea hat einen Wert für MAPE von",  
    mape(demand_ts_skorea, fcast_skorea$fitted),".\n")  
## Modell für Suedkorea hat einen Wert für MAPE von 0.99 .
```

*Stellungnahme zu Aufgabe 11:* Der Mean Absolute Percentage Error (MAPE) zeigt den durchschnittlichen prozentualen Fehler im Hinblick auf die Vorhersagegenauigkeit der Modelle an. Allgemein gilt, dass MAPE die Vergleichbarkeit bei unterschiedlicher Skalierung ermöglicht und dass kleine Abweichungen bei kleinem  $y$  stark gewertet werden. Wenn wir die vier Regionen Japan, Peking, Phillipinen und Suedkorea miteinander vergleichen, erkennen wir, dass die Region Phillipinen den kleinsten MAPE-Wert von 0.8036 hat. Beim MAPE-Vergleich gilt, dass je kleiner der prozentuale Fehler desto besser ist die Vorhersage. Dies trifft auf die Region Phillipinen zu, welches also das beste Modell im Vergleich zu den anderen Regionen ist.

## Abschluss

### Aufgabe 12) Untersuchung der Nachfrage nach Limonade in Peking, den Philippinen und Suedkorea fuer den naechsten Monat (Januar 2021)

```
ets_peking = ets(demand_ts_peking, model = "ZZZ")
ets_phlppn = ets(demand_ts_phlppn, model = "ZZZ")
ets_skorea = ets(demand_ts_skorea, model = "ZZZ")

fcast_peking = data.frame(forecast(ets_peking, 12))
fcast_phlppn = data.frame(forecast(ets_phlppn, 12))
fcast_skorea = data.frame(forecast(ets_skorea, 12))

# Dataframe kombinieren und Summe von Sales für Peking, Philippinen und Südko
rea
SumRegion <- rbind(fcast_peking + fcast_phlppn + fcast_skorea)

# Tabellenkopf erste Zeile bzw. Januar 2021 von forecast ausgeben
head(SumRegion, 1)

##      Point.Forecast  Lo.80   Hi.80   Lo.95   Hi.95
## Jan 6          47127.27 46280.4 47974.14 45832.09 48422.44
```

*Kommentar zu Aufgabe 12:* Der Output zeigt die Forecast-Werte sowie die oberen und unteren Grenzen der 80% und 95% Konfidenzintervalle an. Der wahre Wert liegt mit 95%iger Sicherheit zwischen den Nachfragewerten 45832.09 und 48422.44. Wir gehen die Wette ein, da wir mit 95%iger Sicherheit sagen koennen, dass die Nachfrage nach Limonade in Peking, den Philippinen und Suedkorea im naechsten Monat in der Summe ca. bei 47127 liegen wird. Somit uebersteigt die Nachfrage nach Limonade um 2127 Limonadeflaschen und wir wuerden die Wette mit 95%iger Sicherheit gewinnen. Es gilt jedoch immer eine gewisse Unersicherheit, da wir nicht 100% sagen koennen, ob die Nachfrage nach ueber 45.000 Limoadenflaschen uebersteigt.

## Aufgabe 13) Berechnung der Nachfragevorhersage im ersten Quartal von 2021 in Shanghai fuer drei Supermaerkte.

```
# (1) Datenbereinigen: Nur Produkt Gruppe105 in der Region Shanghai
DemandShanghai = subset(transactions, Product == "Gruppe105" &
                          region=="Shangh" &
                          Sales > 0)
DemandShanghai <- unique(DemandShanghai)

# (2) Aggregieren
DemandShanghai = aggregate(Sales ~ Periode, data = DemandShanghai, sum)

# (3) Erstellen der Zeitreihe
ts_Shanghai = ts(DemandShanghai$Sales, frequency = 12)

# (4) Modell erstellen
m1_Shanghai = ets(ts_Shanghai, model="ZZZ")

# Die Vorhersage Nachfrage bezieht sich auf 5 Gescheafte.
fcast1_Shanghai = data.frame(forecast(m1_Shanghai, 12))

# Da Olaf nur 3 Geschaefte hat, muessen wir die Anzahl anpassen und runden
fcast1_Shanghai <- round((fcast1_Shanghai[1]/5)*3,2)

# Spaltenname aendern
colnames(fcast1_Shanghai) = c("Demand Forecast Q1")

# 3 Monate bzw. ein Quartal anzeigen
head(fcast1_Shanghai, 3)[1]

##          Demand Forecast Q1
## Jan 6          1681.10
## Feb 6          1699.12
## Mar 6          1510.63
```

*Kommentar zu Aufgabe 13:* Anhand unseres Vorhersagemodells basierend auf den Vergangenheitsdaten der Nachfrage in Shanghai empfehlen wir unseren Freund Olaf im ersten Quartal in Januar 1681 Limonaden-Flaschen zu kaufen sowie im Februar 1699 und im Maerz 1511 Limonaden-Flaschen. Hier gilt es insbesondere zu beruecksichtigen, dass die alte Vorhersage alle fuenf Geschaefte und ihre Nach- fragemenge umfasse, daher haben wir die Anzahl der Nachfrage auf die Anzahl der drei Supermaerkte reduziert, die Olaf besitzt.