

R Notebook

[Code ▼](#)

HAUSAUFGABE 2 BEFINDET SICH AB ZEILE 312

```
||||_|||  
|_|||||  
_|||_||||  
||||_|||  
|| || || |_ HIER IST HAUSAUFGABE 1. HAUSAUFGABE 2 IST AB ZEILE 312 BZW SEITE 21 IN PDF
```

**** Teil A) Übersicht verschaffen ****

1. Importieren Sie die zwei Datensätze und speichern Sie diese als Variablen mit passenden Namen.
Geben Sie die Struktur von jeder Variable aus. Bewertungsrelevant: Input, Output.

[Hide](#)

```
## Speichern der Datensätze als Variablen  
services = read.csv2("output_services_v0025.csv")  
transactions = read.csv2("output_transactions_v0025.csv")  
## Ausgabe ihrer Struktur  
print("Struktur der Services")
```

```
[1] "Struktur der Services"
```

[Hide](#)

```
str(services)
```

```
'data.frame':  106674 obs. of  13 variables:  
 $ Year      : int  2019 2019 2019 2019 2019 2019 2019 2019 2019 ...  
 $ Month     : int  1 1 1 1 1 1 1 1 1 ...  
 $ Day       : int  3 3 3 3 3 3 3 3 3 ...  
 $ region    : chr   "Shangh" "Shangh" "Shangh" "Shangh" ...  
 $ storename : chr   "Shangh-1" "Shangh-1" "Shangh-1" "Shangh-1" ...  
 $ Product   : chr   "Gruppe201" "Gruppe201" "Gruppe204" "Gruppe204" ...  
 $ vendor    : chr   "AHL Express Warehousing" "JNT Shipping" "CPS Warehousing" "CPS Shippi  
ng" ...  
 $ service   : chr   "Warehousing" "Shipping" "Warehousing" "Shipping" ...  
 $ DaysScheduled: int  0 2 0 2 0 2 0 2 0 ...  
 $ DaysExecuted : int  0 2 0 3 0 3 0 3 0 ...  
 $ QScheduled  : int  39 39 193 193 116 116 39 39 77 ...  
 $ QExecuted   : int  27 39 166 193 90 116 28 39 60 ...  
 $ cost        : num  46.8 46.8 231.6 231.6 139.2 ...
```

[Hide](#)

```
print("Struktur der Transactions")
```

```
[1] "Struktur der Transactions"
```

[Hide](#)

```
str(transactions)
```

```
'data.frame':  297073 obs. of  8 variables:
 $ Year      : int   2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 ...
 $ Month     : int    1 1 1 1 1 1 1 1 1 1 ...
 $ Day       : int    1 1 1 1 1 1 1 1 1 1 ...
 $ region    : chr   "Shangh" "Shangh" "Shangh" "Shangh" ...
 $ storename : chr   "Shangh-1" "Shangh-1" "Shangh-1" "Shangh-1" ...
 $ Product   : chr   "Gruppe201" "Gruppe202" "Gruppe203" "Gruppe204" ...
 $ Sales     : int    0 0 0 0 0 0 0 0 0 0 ...
 $ Received  : int   125 125 125 125 125 125 125 125 125 125 ...
```

2. Entfernen Sie alle Datenpunkte, die ausserhalb der Zeitspanne von 2019 bis 2023 liegen. Geben Sie anschliessend die Zusammenfassung der Variablen aus. Bewertungsrelevant: Output.

[Hide](#)

```
## Ausfilterung aller Datenpunkte außerhalb 2019 bis 2023
services_2019_to_2023 = subset(services, Year >= 2019 & Year <= 2023)
transactions_2019_to_2023 = subset(transactions, Year >= 2019 & Year <= 2023)
## Ausgabe der Zusammenfassung der Variablen
print("Zusammenfassung der gefilterten Services")
```

```
[1] "Zusammenfassung der gefilterten Services"
```

[Hide](#)

```
summary(services_2019_to_2023)
```

uct	Year	Month	Day	region	storename	Prod
Min.	:2019	Min. : 1.000	Min. : 1.00	Length:106674	Length:106674	Length
h:106674		Length:106674	Length:106674	Min. :0	Min. :0.000	
1st Qu.:2020	1st Qu.: 3.000	1st Qu.: 5.00	Class :character	Class :character	Class	
:character	Class :character	Class :character	1st Qu.:0	1st Qu.:0.000		
Median :2021	Median : 7.000	Median :10.00	Mode :character	Mode :character	Mode	
:character	Mode :character	Mode :character	Median :1	Median :1.000		
Mean :2021	Mean : 6.504	Mean :10.49				
Mean :1	Mean :1.391					
3rd Qu.:2022	3rd Qu.: 9.000	3rd Qu.:15.00				
3rd Qu.:2	3rd Qu.:3.000					
Max. :2023	Max. :12.000	Max. :20.00				
Max. :2	Max. :4.000					
QScheduled	QExecuted	cost				
Min. : 30.0	Min. : 19.0	Min. : 36.34				
1st Qu.: 81.0	1st Qu.: 70.0	1st Qu.:103.91				
Median :101.0	Median : 95.0	Median :131.66				
Mean :112.3	Mean :102.8	Mean :145.78				
3rd Qu.:142.0	3rd Qu.:132.0	3rd Qu.:185.25				
Max. :343.0	Max. :343.0	Max. :486.27				

[Hide](#)

```
print("Zusammenfassung der gefilterten Transactions")
```

```
[1] "Zusammenfassung der gefilterten Transactions"
```

Hide

```
summary(transactions_2019_to_2023)
```

uct	Year	Month	Day	region	storename	Prod
Min. :2019	Min. : 1.000	Min. : 1.00	Length:296945	Length:296945	Length:296945	Length:296945
1st Qu.:2020	1st Qu.: 3.000	1st Qu.: 5.00	Class :character	Class :character	Class :character	Class :character
Median :2021	Median : 6.000	Median :10.00	Mode :character	Mode :character	Mode :character	Mode :character
Mean :2021	Mean : 6.497	Mean :10.47				
3rd Qu.:2022	3rd Qu.: 9.000	3rd Qu.:15.00				
Max. :2023	Max. :12.000	Max. :20.00				

3. Extrahieren Sie aus den Transaktionsdaten eine Liste aller Supermärkte, in der jeder Supermarkt nur einmal enthalten ist. Beachten Sie ggf. diesen Link. Bewertungsrelevant: Input, Output.

Hide

```
# Liste aller Supermärkte
list_supermarkets = unique(transactions_2019_to_2023$storename)
print("Liste der Supermärkte:")
```

```
[1] "Liste der Supermärkte:"
```

Hide

```
list_supermarkets
```

```
[1] "Shangh-1" "Shangh-2" "Shangh-3" "Shangh-4" "Shangh-5" "Peking-1" "Peking-2" "Peking-3"
"Peking-4" "Peking-5" "Japan-1" "Japan-2" "Japan-3" "Japan-4"
[15] "Japan-5" "Skorea-1" "Skorea-2" "Skorea-3" "Skorea-4" "Skorea-5" "Phlppn-1" "Phlppn-2"
"Phlppn-3" "Phlppn-4" "Phlppn-5"
```

4. Extrahieren Sie aus den Servicedaten eine Tabelle aller 20 Logistikdienstleister mitsamt deren Dienstleistungen. Jeder Logistikdienstleister soll in der Liste nur einmal enthalten sein. Sortieren Sie die Tabelle nach Warehousing- Dienstleister und Shipping-Dienstleister. Bewertungsrelevant: Input, Output.

Hide

```
# Extraktion aller Logistikdienstleister mitsamt deren Dienstleistungen
all_vendors = select(services_2019_to_2023, vendor, service)
# Filterung der einzigartigen Dienstleister
list_vendors = distinct(all_vendors, vendor, .keep_all = TRUE)
# Sortieren der Dienstleister
list_vendors[order(list_vendors$service, decreasing = TRUE), ]
```

	vendor <chr>	service <chr>
1	AHL Express Warehousing	Warehousing
3	CPS Warehousing	Warehousing
5	Gifter Warehousing	Warehousing
6	EPD Warehousing	Warehousing
8	IntEx Warehousing	Warehousing
9	DWL Warehousing	Warehousing
12	Flying Mercury Warehousing	Warehousing
15	JNT Warehousing	Warehousing
19	Bange+Hammer Warehousing	Warehousing
20	HCX Warehousing	Warehousing
1-10 of 20 rows		Previous 1 2 Next

5. Extrahieren Sie aus den Transaktionsdaten eine Tabelle aller existierenden Produkte in Südkorea, in der jedes Produkt nur einmal enthalten ist. Bewertungsrelevant: Input, Output.

Hide

```
# Extraktion aller Produkte in Korea
all_products_in_kr = filter(transactions_2019_to_2023, region == "Skorea")
# Ausfilterung der nicht relevanten Spalten
all_products_in_kr = select(all_products_in_kr, Product, region)
# Filterung der einzigartigen Produkte
list_products_in_kr = distinct(all_products_in_kr, Product, .keep_all = TRUE)
list_products_in_kr
```

Product <chr>	region <chr>
Gruppe201	Skorea
Gruppe202	Skorea
Gruppe203	Skorea
Gruppe204	Skorea
Gruppe205	Skorea
Gruppe206	Skorea

Product <chr>	region <chr>
Gruppe207	Skorea
Gruppe208	Skorea
Lost Sales	Skorea
9 rows	

**** Teil B) Marktübersicht **** 6) Erstellen Sie eine Tabelle mit den Marktanteilen (an den tatsächlich verkauften Mengen) der jeweiligen Produkte. Interpretieren Sie diese Tabelle kurz. Wie viele Flaschen Limonade haben Sie im gesamten Zeitraum verkauft? Bewertungsrelevant: Output, Kommentar.

Hide

```
## Erstellung einer Tabelle mit den Marktanteilen der jeweiligen Produkte
# Berechnung der gesamten Anzahl an verkauften Produkten
total_sold = sum(transactions_2019_to_2023$Sales)
# Berechnung der verkauften Mengen jedes Produktes
sold_products = aggregate(transactions_2019_to_2023$Sales, list(transactions_2019_to_2023$Product), FUN=sum)
names(sold_products) = c("Products", "Sales")
# Berechnung der Marktanteilen der jeweiligen Produkte
sold_products["Market Share"] = sold_products$Sales/total_sold
sold_products
```

Products <chr>	Sales <int>	Market Share <dbl>
Gruppe201	630678	0.12317379
Gruppe202	627842	0.12261991
Gruppe203	629957	0.12303297
Gruppe204	624546	0.12197619
Gruppe205	618342	0.12076452
Gruppe206	616537	0.12041200
Gruppe207	617523	0.12060457
Gruppe208	616436	0.12039227
Lost Sales	138368	0.02702379
9 rows		

Hide

```
## Interpretation der Tabelle
# "Gruppe201" ist das beliebteste Produkt. Jedoch haben alle Produkte ungefähr gleichen Marktanteilen, nämlich jeweils 12% bis 12,3%.
# Erstaunlicherweise sind circa 2,7% der Produkte verlorgen gegangen.
## Wie viele Flaschen Limonade haben im gesamten Zeitraum verkauft? (ist genau die Anzahl an verkauften Produkten)
sprintf("Die gesamte Anzahl an verkauften Flaschen Limonade ist %d", total_sold)
```

```
[1] "Die gesamte Anzahl an verkauften Flaschen Limonade ist 5120229"
```

7. Erstellen Sie eine Tabelle, um einen Überblick über den Absatz Ihres Produktes in den fünf verschiedenen Regionen zu erhalten. Die Tabelle sollte folgende Spalten aufweisen: Region und gesamte Absatzmenge. Wie ist der prozentuale Unterschied zwischen der absatzstärksten und der absatzschwächsten Region? Bewertungsrelevant: Output, Kommentar.

Hide

```
## Erstellung einer Tabelle über den Produktabsatz in den Regionen
# Berechnung der Anzahl an verkauften Produkten in jeder Region
sales_by_region = aggregate(transactions_2019_to_2023$Sales, list(transactions_2019_to_2023$region), FUN=sum)
colnames(sales_by_region) = c("region", "Sales")
# Berechnung des Prozentsatzes
sales_by_region["Percentage"] = sales_by_region$Sales/total_sold
sales_by_region
```

region <chr>	Sales <int>	Percentage <dbl>
Japan	1031894	0.2015328
Peking	1027097	0.2005959
Philppn	1020098	0.1992290
Shangh	1030585	0.2012771
Skorea	1010555	0.1973652
5 rows		

Hide

```
## Unterschied zwischen der stärksten und der schwächsten Region
# Alle 5 Regionen haben gleichmäßigen Prozentsatz, jeweils circa 20%
# Die stärkste Region ist Japan mit 20,15328%, während die schwächste Korea ist mit 19,73652%.
# Somit liegt der prozentuale Unterschied bei 0,41676%.
```

8. Berechnen Sie für jedes Jahr den Marktanteil Ihres Produkts (an der tatsächlich verkauften Menge). Wie hat sich der Marktanteil in den letzten Jahren entwickelt? Bewertungsrelevant: Output, Kommentar.

Hide

```

transactions_2019_to_2023 <- subset(transactions, Year >= 2019 & Year <= 2023)
## Berechnung des Marktanteils des Produktes "Gruppe204" für jedes Jahr
market_share_per_year <- transactions_2019_to_2023 %>%
  group_by(Year) %>%
  summarise(
    total_sales = sum(Sales),
    product_sales = sum(Sales[Product == "Gruppe204"])
  ) %>%
  mutate(market_share = product_sales / total_sales)
print(market_share_per_year)

```

Year <int>	total_sales <int>	product_sales <int>	market_share <dbl>
2019	1032576	124607	0.1206759
2020	1028731	126204	0.1226793
2021	1018562	124987	0.1227093
2022	1018380	124458	0.1222117
2023	1021980	124290	0.1216169

5 rows

Hide

Analyse: Der Marktanteil des Produkts hat sich über die fünf Jahre nur geringfügig verändert.
 #Es gab einen leichten Anstieg von 2019 bis 2020, danach blieb der Marktanteil relativ stabil bis 2021.
 #In den Jahren 2022 und 2023 ist ein kleiner Rückgang zu verzeichnen.
 #Insgesamt zeigt der Marktanteil eine leichte Abwärtsbewegung, bleibt aber über die Jahre relativ stabil bei etwa 12%.

9. Berechnen Sie den durchschnittlichen Absatz Ihres Produkts je Kalendermonat. Nennen Sie daraufhin den Monat mit dem höchsten durchschnittlichen Absatz. Bewertungsrelevant: Output, Kommentar.

Hide

```

## Berechnung des durchschnittlichen Absatzes je Kalendermonates
average_monthly_sales <- transactions_2019_to_2023 %>%
  group_by(Month) %>%
  summarise(average_sales = mean(Sales[Product == "Gruppe204"]))
print(average_monthly_sales)

```

Month <int>	average_sales <dbl>
1	16.47201
2	16.88139
3	17.93589
4	16.72805

Month <int>	average_sales <dbl>
5	16.72462
6	16.24434
7	17.07604
8	18.22258
9	17.91983
10	16.74975

1-10 of 12 rows

Previous 1 2 Next

Hide

```
## Nennung des Monats mit dem höchsten durchschnittlichen Absatzes
max_sales_month <- average_monthly_sales %>%
  filter(average_sales == max(average_sales))
print(max_sales_month)
```

Month <int>	average_sales <dbl>
8	18.22258

1 row

**** Teil C) Umsatz, Kosten und Profit **** 10) Berechnen Sie Ihren Gesamtumsatz, Ihre Gesamtkosten und Ihren Gesamtprofit im Betrachtungszeitraum. Be- wertungsrelevant: Output. Berechnen Sie ihren Umsatz und ihre Gesamtkosten im Betrachtungszeitraum. (Zur Vereinfachung werden nur die Kosten für die Produktion und Transport zu den Großlagern betrachtet). Wie viel Profit konnten Sie erwirtschaften? Bewertungsrelevant: Output, Kommentar.

Hide

```
price_per_bottle <- 3.90
cost_per_bottle <- 0.60
# Berechnung des Umsatzes, der Gesamtkosten und des Gesamtprofites
total_revenue <- sum(transactions_2019_to_2023$Sales[transactions_2019_to_2023$Product == "Grupp
e204"]) * price_per_bottle
total_cost <- sum(transactions_2019_to_2023$Sales[transactions_2019_to_2023$Product == "Grupp
e204"]) * cost_per_bottle
total_profit <- total_revenue - total_cost
# Ausgabe der Ergebnisse
cat("Gesamtumsatz:", total_revenue, "\n")
```

Gesamtumsatz: 2435729

Hide

```
cat("Gesamtkosten:", total_cost, "\n")
```


Gesamtkosten: 374727.6

Hide

```
cat("Gesamtprofit:", total_profit, "\n")
```

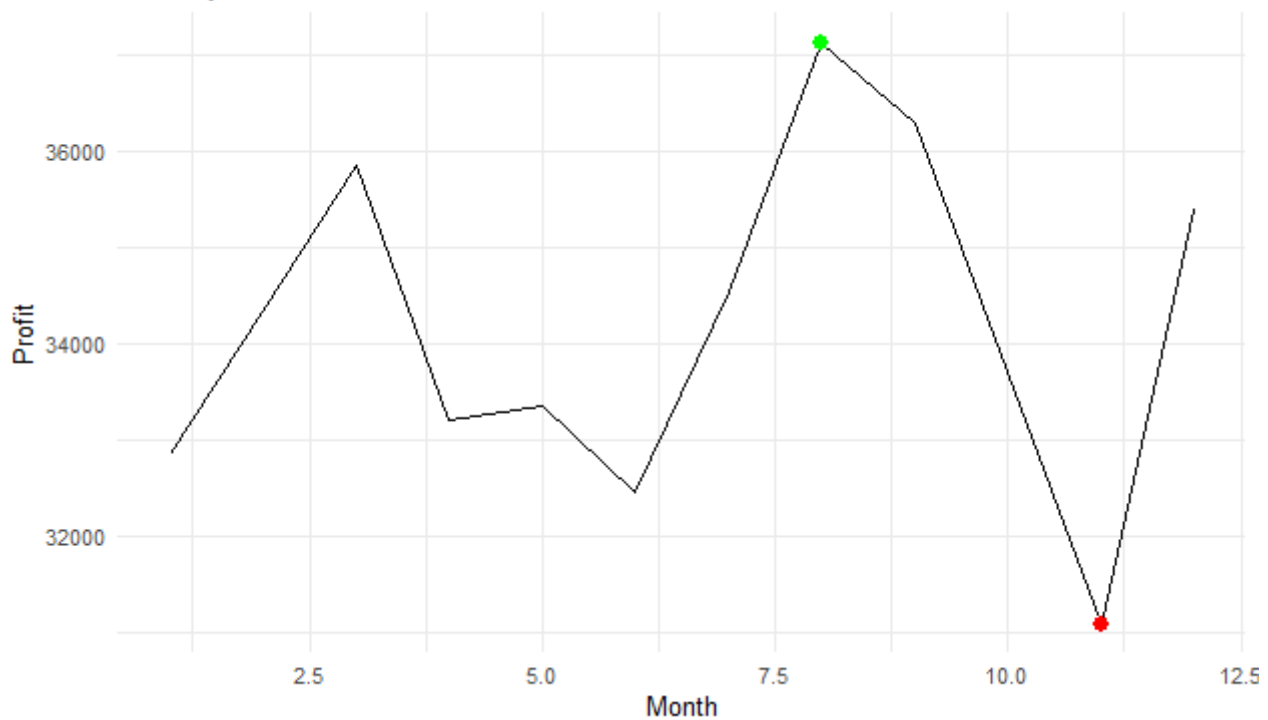
Gesamtprofit: 2061002

11. Erstellen Sie eine Grafik, in der Ihr Profit für das Jahr 2023 auf Monatsebene dargestellt ist. Heben Sie die Monate mit dem grössten und kleinsten Profit passend farbig hervor. Bewertungsrelevant: Output. Hinweis: Es könnte hilfreich sein, zunächst ein Dataframe mit den passenden Daten zu erstellen.

Hide

```
# Berechnung des monatlichen Profites
transactions_2023 <- subset(transactions_2019_to_2023, Year == 2023)
monthly_profit_2023 <- transactions_2023 %>%
  group_by(Month) %>%
  summarise(
    total_sales = sum(Sales[Product == "Gruppe204"]),
    revenue = total_sales * price_per_bottle,
    cost = total_sales * cost_per_bottle,
    profit = revenue - cost
  )
# Bestimmung der Monate mit dem größten und kleinsten Profit
max_profit_month <- monthly_profit_2023 %>% filter(profit == max(profit))
min_profit_month <- monthly_profit_2023 %>% filter(profit == min(profit))
# Darstellung des monatlichen Profites
ggplot(monthly_profit_2023, aes(x = Month, y = profit)) +
  geom_line() +
  geom_point(data = max_profit_month, aes(x = Month, y = profit), color = "green", size = 3)
+
  geom_point(data = min_profit_month, aes(x = Month, y = profit), color = "red", size = 3) +
  ggtitle("Monthly Profit for 2023") +
  xlab("Month") +
  ylab("Profit") +
  theme_minimal()
```

Monthly Profit for 2023



12. Erstellen Sie eine Grafik, in der Ihre Kosten über den gesamten Zeitraum für Transportdienstleistungen abgebildet sind. Die Grafik soll einen Vergleich der Regionen ermöglichen. Für jede Region soll ersichtlich werden:

1. Wie viel wurde für pünktliche Transportdienstleistungen ausgegeben und (2) wie viel wurde für verspätete Transportdienstleistungen ausgegeben. Interpretieren Sie die Grafik. Bewertungsrelevant: Output, Kommentar.

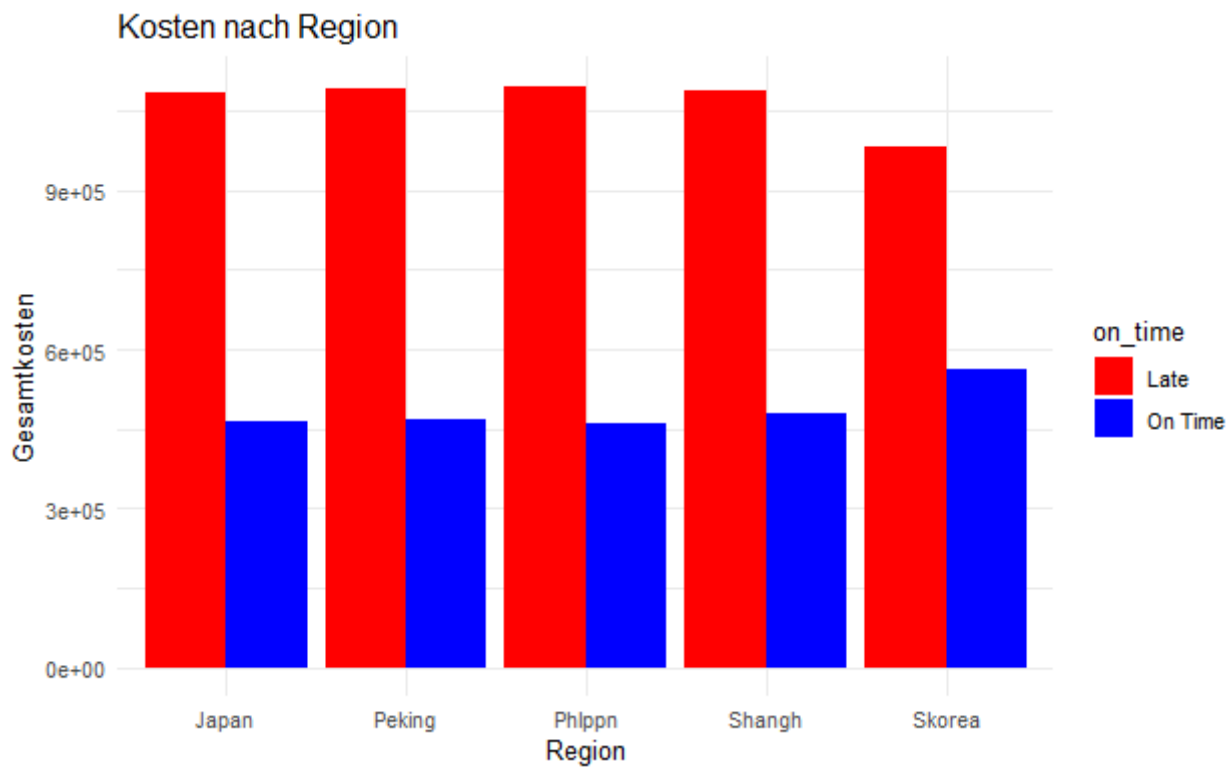
Hide

```
# Berechnung der Kosten für Transportdienstleistungen
transport_costs <- services_2019_to_2023 %>%
  filter(service == "Shipping") %>%
  group_by(region, DaysExecuted <= 2) %>%
  summarise(total_cost = sum(cost))
```

`summarise()` has grouped output by 'region'. You can override using the `.groups` argument.

Hide

```
# Umbenennung der Gruppe
transport_costs <- transport_costs %>%
  mutate(on_time = ifelse(`DaysExecuted` <= 2, "On Time", "Late"))
# Darstellung der Kosten nach Region
ggplot(transport_costs, aes(x = region, y = total_cost, fill = on_time)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Kosten nach Region") +
  xlab("Region") +
  ylab("Gesamtkosten") +
  scale_fill_manual(values = c("On Time" = "blue", "Late" = "red")) +
  theme_minimal()
```



** Teil D) Bewertung der DLI ** 13) Nennen Sie eine geeignete Kennzahl zur Bewertung Ihrer Shipping-Dienstleister. Beachten Sie dabei, was die Qualiteat der Shipping-Dienstleister ausmacht. Begründen Sie die Wahl der Kennzahl kurz. Erstellen Sie ein Sub- set der Services Tabelle mit allen Shipping-Dienstleistungen. Berechnen Sie ihre gewählte Kennzahl für alle Di- enstleistungen als zusätzliche Variable in dieser Tabelle. Berechnen Sie daraufhin die durchschnittliche Kennzahl der Shipping-Dienstleister im Jahr 2023. Geben Sie Ihre Ergebnisse in einer Tabelle aus, in der die Kennzahl-Werte absteigend sortiert sind. Bewertungsrelevant: Kommentar, Code, Output.

Hide

```
# Eine geeignete Kennzahl dafür ist der Prozentsatzes der punktlichen Lieferung von jedem Dienstleister.
# Damit wird festgestellt, welcher Dienstleister am häufigsten die Produkte punktlich liefert
# Berechnung des Prozentsatzes der punktlichen Lieferung von jedem Dienstleister
shipping_evaluation <- services_2019_to_2023 %>%
  filter(service == "Shipping") %>%
  group_by(vendor) %>%
  summarise(
    total_deliveries = n(),
    on_time_deliveries = sum(DaysExecuted <= 2),
    on_time_percentage = on_time_deliveries / total_deliveries * 100
  ) %>%
  arrange(desc(on_time_percentage))
print(shipping_evaluation)
```

vendor <chr>	total_deliveries <int>	on_time_deliveries <int>	on_time_percentage <dbl>
IntEx Shipping	5372	1805	33.60015
DWL Shipping	5299	1776	33.51576
Bange+Hammer Shipping	5318	1751	32.92591
EPD Shipping	5335	1741	32.63355

vendor <chr>	total_deliveries <int>	on_time_deliveries <int>	on_time_percentage <dbl>
Gifter Shipping	5390	1741	32.30056
Flying Mercury Shipping	5405	1714	31.71138
JNT Shipping	5318	1669	31.38398
HCX Shipping	5254	1632	31.06205
CPS Shipping	5344	1429	26.74027
AHL Express Shipping	5302	1400	26.40513

1 10 of 10 rows

14. Nennen Sie eine geeignete Kennzahl zur Bewertung Ihrer Warehousing-Dienstleister. Beachten Sie dabei, was die Qualität der Warehousing-Dienstleister ausmacht. Begründen Sie die Wahl der Kennzahl kurz. Erstellen Sie ein Subset der Services Tabelle mit allen Warehousing-Dienstleistungen. Berechnen Sie ihre ausgewählte Kennzahl für alle Dienstleistungen als zusätzliche Variable in dieser Tabelle. Berechnen Sie anschließend die durchschnittliche Kennzahl für die Warehousing-Dienstleister für die Dienstleistungen an Ihrem Produkt über die gesamte Laufzeit (5 Jahre). Geben Sie Ihre Ergebnisse in einer Tabelle aus, in der die Kennzahl-Werte absteigend sortiert sind. Bewertungsrelevant: Kommentar, Code, Output.

Hide

```
## Eine geeignete Kennzahl dazu ist der Prozentsatz der Anfragen, bei denen sich QExecuted und
QScheduled übereinstimmen.
# Wenn diese zwei Attribute bei demselben Wert liegen, dann wird die richtige Menge der Produkte
abgefertigt.
# Daher kann mit dieser Kennzahl die Qualität der Dienstleister bewertet werden.
# Filterung der Warehousing-DL und Berechnung des Genauigkeitsprozentsatzes
warehousing_evaluation <- services_2019_to_2023 %>%
  filter(service == "Warehousing") %>%
  group_by(vendor) %>%
  summarise(
    total_requests = n(),
    accurate_fulfillments = sum(QExecuted == QScheduled),
    accuracy_percentage = accurate_fulfillments / total_requests * 100
  ) %>%
  arrange(desc(accuracy_percentage))
# Ausgabe der Ergebnisse
print(warehousing_evaluation)
```

vendor <chr>	total_requests <int>	accurate_fulfillments <int>	accuracy_percentage <dbl>
DWL Warehousing	5267	2	0.0379
Bange+Hammer Warehousing	5351	2	0.0373
HCX Warehousing	5379	2	0.0371
Flying Mercury Warehousing	5348	1	0.0186
JNT Warehousing	5406	1	0.0184

vendor	total_requests	accurate_fulfillments	accuracy_percent
<chr>	<int>	<int>	
AHL Express Warehousing	5332	0	0.0000
CPS Warehousing	5289	0	0.0000
EPD Warehousing	5377	0	0.0000
Gifter Warehousing	5304	0	0.0000
IntEx Warehousing	5284	0	0.0000
1 - 10 of 10 rows			

15. Visualisieren Sie in geeigneter Form die gewählte Qualitätskennzahl der Warehouse-Dienstleister in einem ggplot (bezogen auf alle Produkte) im Jahr 2023. Durch die Visualisierung soll eine differenzierte Vergleichbarkeit der Dienstleister möglich sein. Wie bewerten Sie die Qualität der Warehousing-DL insgesamt? Bewertungsrelevant: Output, Kommentar.

Hide

```
# Filterung der Warehousing-DL im Jahr 2023
warehousing_2023 <- services_2019_to_2023 %>%
  filter(service == "Warehousing" & Year == 2023)
print(warehousing_2023)
```

Y...	Mo...	D..	region	storename	Product	vendor	service	Da
<int>	<int>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	
2023	1	1	Shangh	Shangh-1	Gruppe204	DWL Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-2	Gruppe202	DWL Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-2	Gruppe206	EPD Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-2	Gruppe208	AHL Express Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-3	Gruppe201	EPD Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-3	Gruppe205	Gifter Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-3	Gruppe207	Flying Mercury Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-3	Gruppe208	JNT Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-4	Gruppe202	Flying Mercury Warehousing	Warehousing	
2023	1	1	Shangh	Shangh-4	Gruppe205	JNT Warehousing	Warehousing	
1-10 of 10,633 rows 1-9 of 13 columns								
Previous 1 2 3 4 5 6 ... 100 Next								

Hide

```
# Berechnung des Genauigkeitsprozentsatzes für jeden Dienstleister im Jahr 2023
warehousing_accuracy_2023 <- warehousing_2023 %>%
  group_by(vendor) %>%
  summarise(
    total_requests = n(),
    accurate_fulfillments = sum(QExecuted == QScheduled),
    accuracy_percentage = accurate_fulfillments / total_requests * 100
  )
print(warehousing_accuracy_2023)
```

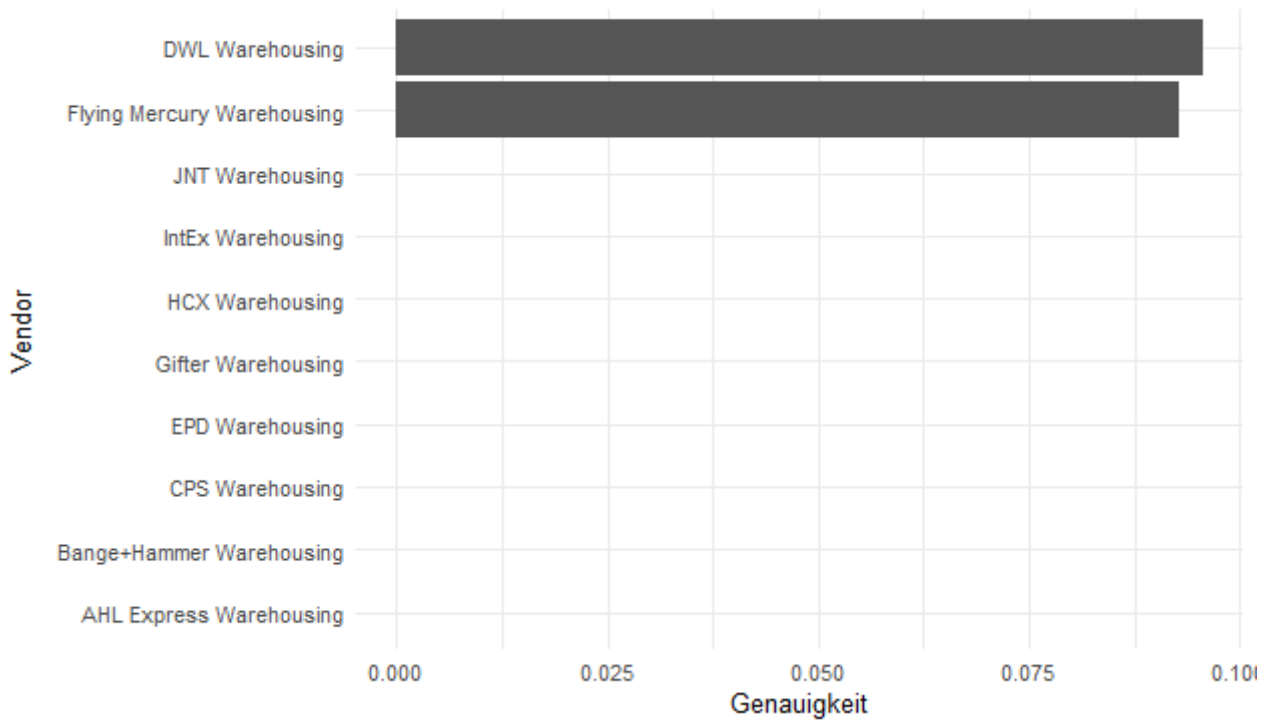
vendor <chr>	total_requests <int>	accurate_fulfillments <int>	accuracy_perce
AHL Express Warehousing	1059	0	0.0000
Bange+Hammer Warehousing	1059	0	0.0000
CPS Warehousing	1069	0	0.0000
DWL Warehousing	1046	1	0.0956
EPD Warehousing	1103	0	0.0000
Flying Mercury Warehousing	1078	1	0.0927
Gifter Warehousing	1041	0	0.0000
HCX Warehousing	1060	0	0.0000
IntEx Warehousing	1029	0	0.0000
JNT Warehousing	1089	0	0.0000

1-10 of 10 rows

Hide

```
# Darstellung des Genauigkeitsprozentsatzes
ggplot(warehousing_accuracy_2023, aes(x = reorder(vendor, accuracy_percentage), y = accuracy_
percentage)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  ggtitle("Genauigkeitsprozentsatzes der Dienstleister") +
  xlab("Vendor") +
  ylab("Genauigkeit") +
  theme_minimal()
```

Genauigkeitsprozentsatzes der Dienstleister



Hide

```
## Von der Analyse wird festgestellt, dass die Genauigkeit sehr niedrig ist bzw. die Qualität
der DL ganz schlecht ist.
# Über den gesamten Zeitraum (2019-2023) ist die Genauigkeit bei der Hälfte der DLn nur zwisc
hen 1,8% und 3,8%.
# Bei den anderen DLn ist die Genauigkeit sogar bei 0, also diese DL haben gar keine Anfrage
genau bearbeitet.
# Im Jahr 2023 ist die Qualität der DL noch schlimmer.
# Bei DWL und Flying Mercury gibt es nur eine Anfrage, die genau bearbeitet wurde wie geplan
t.
# Bei anderen Dienstleistern gibt es gar keine solche Anfrage.
```

HAUSAUFGABE2 AB HIER

```
||||_||_|
|||||_|
_|||||_|
||||_||_|
||||||_|
```

Laden von Packages

Hide

```
library(dplyr)
```

Warnung: Paket 'dplyr' wurde unter R Version 4.2.3 erstellt

Attache Paket: 'dplyr'

Die folgenden Objekte sind maskiert von 'package:stats':

filter, lag

Die folgenden Objekte sind maskiert von 'package:base':

intersect, setdiff, setequal, union

Hide

```
library(ggplot2)
```

Warnung: Paket 'ggplot2' wurde unter R Version 4.2.3 erstellt

Hide

```
library(reshape2)
```

Warnung: Paket 'reshape2' wurde unter R Version 4.2.3 erstellt

Hide

```
library(plotly)
```

Warnung: Paket 'plotly' wurde unter R Version 4.2.3 erstellt

Registered S3 method overwritten by 'data.table':

method	from
print.data.table	

Registered S3 method overwritten by 'htmlwidgets':

method	from
print.htmlwidget	tools:rstudio

Attache Paket: 'plotly'

Das folgende Objekt ist maskiert 'package:ggplot2':

last_plot

Das folgende Objekt ist maskiert 'package:stats':

filter

Das folgende Objekt ist maskiert 'package:graphics':

layout

**** Daten vorbereiten 1)** Aggregieren Sie die Verkaufszahlen so, dass Sie eine Tabelle mit der Nachfrage je Monat je Region erhalten. Die Nachfrage beinhaltet alle Produkte, einschließlich Lost Sales. Speichern Sie das Dataframe in einer neuen Variable mit dem Namen "Demand" und den Spaltennamen "Region", "Period" und

“Demand”. Geben Sie den Tabellenkopf aus. Bewertungsrelevant: Code, Output. Hinweis: Für die folgenden Aufgaben bietet es sich an, eine Variable “Periode” zu erzeugen, die aus dem Jahr und dem Monat (in dieser Reihenfolge und jeweils durch eine Zahl repräsentiert) besteht. Siehe dazu die Dokumentation unter diesem Link und die Dokumentation unter diesem Link.

Hide

```
# Erstellung der Spalte "Periode" durch die Zusammenführung von Jahr und Monat
transactions_2019_to_2023$Periode = paste(transactions_2019_to_2023$Year, ifelse(transactions_2019_to_2023$Month < 10, paste("0", transactions_2019_to_2023$Month, sep=""), transactions_2019_to_2023$Month), sep="")
transactions_2019_to_2023
```

	Year <int>	Month <int>	D... <int>	region <chr>	storename <chr>	Product <chr>	Sales <int>	Received <int>	Periode <chr>						
1	2019	1	1	Shangh	Shangh-1	Gruppe201	0	125	201901						
2	2019	1	1	Shangh	Shangh-1	Gruppe202	0	125	201901						
3	2019	1	1	Shangh	Shangh-1	Gruppe203	0	125	201901						
4	2019	1	1	Shangh	Shangh-1	Gruppe204	0	125	201901						
5	2019	1	1	Shangh	Shangh-1	Gruppe205	0	125	201901						
6	2019	1	1	Shangh	Shangh-1	Gruppe206	0	125	201901						
7	2019	1	1	Shangh	Shangh-1	Gruppe207	0	125	201901						
8	2019	1	1	Shangh	Shangh-1	Gruppe208	0	125	201901						
9	2019	1	1	Shangh	Shangh-2	Gruppe201	0	125	201901						
10	2019	1	1	Shangh	Shangh-2	Gruppe202	0	125	201901						
1-10 of 296,945 rows						Previous	1	2	3	4	5	6	...	100	Next

Hide

```
# Aggregation der Verkaufszahlen und Gruppierung nach Monat und Region
Demand = transactions_2019_to_2023 %>%
  group_by(region, Periode) %>%
  summarise(Demand = sum(Sales))
```

`summarise()` has grouped output by 'region'. You can override using the `.groups` argument.

Hide

```
# Elimination der irrelevanten Spalten
Demand = select(Demand, c(region, Periode, Demand))

# Umbenennung der Spalten
colnames(Demand) = c("Region", "Period", "Demand")

# Ausgabe des Tabellenkopfs
head(Demand)
```

Region <chr>	Period <chr>	Demand <int>
Japan	201901	18147
Japan	201902	21957
Japan	201903	20283
Japan	201904	19533
Japan	201905	20215
Japan	201906	16777
6 rows		

2. Wandeln Sie die aggregierten Demand-Daten vom Long-Format in das Wide-Format um. (Für den vorliegenden Fall gibt es im Long-Format drei Spalten: Periode, Region und Demand. Im Wide-Format gibt es im vorliegenden Fall sechs Spalten: Periode, Demand in Japan, Demand in Peking, Demand in Phlppn, Demand in Shangh, und Demand in Skorea.) Hierfür können Sie die Reshape()-Funktion nutzen. Infos unter diesem Link (<– bitte anklicken). Geben Sie nur den Tabellenkopf aus. Bewertungsrelevant: Code, Output.

Hide

```
# Umwandlung der Daten in das Wide-Format
Demand_wide = dcast(Demand, Period ~ Region, value.var = "Demand")
head(Demand_wide)
```

Period <chr>	Japan <int>	Peking <int>	Phlppn <int>	Shangh <int>	Skorea <int>
1 201901	18147	16574	15494	15471	17240
2 201902	21957	19168	16660	15256	18196
3 201903	20283	17605	21004	14185	20636
4 201904	19533	17029	19042	12571	17007
5 201905	20215	16497	17607	13532	15477
6 201906	16777	15825	15221	19643	13916
6 rows					

3. Wandeln Sie die aggregierten Verkaufszahlen in den Datentyp time-series mit Frequenz = 12 um. Jede Demand-Spalte (jeder Vektor) soll dabei in eine Time Series umgewandelt werden. Dafür benötigt es nur fünf Befehle. Für diese Aufgabe brauchen Sie nichts ausgeben. Bewertungsrelevant: Code.

Hide

```
# Umwandlung der Demand-Spalten in Time Series mit Frequenz = 12
ts_Demand_jp = ts(Demand_wide$Japan, frequency = 12)
ts_Demand_pe = ts(Demand_wide$Peking, frequency = 12)
ts_Demand_sh = ts(Demand_wide$Shangh, frequency = 12)
ts_Demand_ph = ts(Demand_wide$Phlppn, frequency = 12)
ts_Demand_kr = ts(Demand_wide$Skorea, frequency = 12)
```

**** Modellierung vorbereiten 4)** Visualisieren Sie den Nachfrageverlauf aller Regionen in einem einzigen ggplot. Achten Sie auf übersichtliche Achsenbeschriftungen und Legenden. Heben Sie exemplarisch die Region Japan hervor, damit man den Nachfrageverlauf dort gut erkennt (die anderen Regionen müssen nicht unbedingt gut erkennbar, jedoch theoretisch unterscheidbar sein sein). Bewertungsrelevant: Output.

Hide

```
# Umwandlung der ganzen Daten in Time Series für die Visualisierung des Nachfrageverlaufs
ts_Demand = ts(Demand_wide, frequency = 12)
ts_Demand
```

	Period	Japan	Peking	Phlppn	Shangh	Skorea
Jan 1	1	18147	16574	15494	15471	17240
Feb 1	2	21957	19168	16660	15256	18196
Mar 1	3	20283	17605	21004	14185	20636
Apr 1	4	19533	17029	19042	12571	17007
May 1	5	20215	16497	17607	13532	15477
Jun 1	6	16777	15825	15221	19643	13916
Jul 1	7	14563	17038	16623	17608	19889
Aug 1	8	16636	18676	16136	21391	16998
Sep 1	9	16056	17163	16035	21072	18223
Oct 1	10	12751	16112	17103	19064	15466
Nov 1	11	11043	15815	19271	17306	16368
Dec 1	12	19296	20077	17196	18288	15545
Jan 2	13	20328	16010	15058	15763	17195
Feb 2	14	21390	19680	15406	15082	17169
Mar 2	15	19651	17104	18224	13992	19847
Apr 2	16	18312	16669	18621	12838	17387
May 2	17	20975	16341	17620	14081	15800
Jun 2	18	16965	15760	15040	18446	13966
Jul 2	19	13895	17000	16636	18651	19551
Aug 2	20	16749	18928	16109	21108	16410
Sep 2	21	15867	17764	16600	20748	19086
Oct 2	22	13278	16289	17348	19402	15033
Nov 2	23	11130	16206	19697	17604	16464
Dec 2	24	19790	19436	17405	18078	15749
Jan 3	25	19333	15170	14933	14980	17178
Feb 3	26	21424	18583	15990	14441	17197
Mar 3	27	20367	17555	17381	14682	20149
Apr 3	28	18183	16520	18579	12543	17338
May 3	29	20006	16010	17811	14114	15101
Jun 3	30	17366	15801	15912	19489	13760
Jul 3	31	14326	16918	16469	18184	18679
Aug 3	32	17396	19386	15875	20494	16384
Sep 3	33	15739	17260	16172	20721	18497
Oct 3	34	12883	16108	17450	18992	15076
Nov 3	35	11038	16339	18408	17189	16146
Dec 3	36	19956	17808	17491	18067	15215
Jan 4	37	18636	15061	14833	15429	17029
Feb 4	38	20607	18643	15208	14675	16804
Mar 4	39	20087	18718	19004	14889	20149
Apr 4	40	18682	16596	18584	12367	17560
May 4	41	19499	16201	17590	14211	15255
Jun 4	42	17037	15474	15376	18228	13709
Jul 4	43	13928	18341	16184	17632	19099
Aug 4	44	16339	19304	15595	21013	16655
Sep 4	45	16473	17590	15784	22044	18043
Oct 4	46	13132	16409	17044	18967	15310
Nov 4	47	11232	16472	18815	17023	16455
Dec 4	48	17851	18195	18136	18103	15071
Jan 5	49	19002	14802	14812	15954	17588
Feb 5	50	20550	17772	16252	16750	17099
Mar 5	51	20726	17344	17755	14809	19613
Apr 5	52	18361	16693	19372	12924	16620
May 5	53	21217	16110	18137	14179	15708
Jun 5	54	16716	15539	15180	19826	14158

Jul 5	55	14740	17252	16141	18420	18466
Aug 5	56	16797	18507	15651	21147	16312
Sep 5	57	15526	17123	15664	20624	17871
Oct 5	58	12663	15792	17682	19557	14986
Nov 5	59	10712	15771	19989	16969	16147
Dec 5	60	17777	19164	17683	19769	15510

[Hide](#)

```
#Konvertieren von Time Series zu einem Data Frame
df_ts_Demand <- as.data.frame(ts_Demand)

# Hinzufügen der Period-Spalte
df_ts_Demand$Period <- rownames(df_ts_Demand)

# Umwandlung in das long-Format
df_ts_Demand_long <- df_ts_Demand %>%
  pivot_longer(cols = -Period, names_to = "Region", values_to = "Demand")

# Ausgabe der umgewandelten Daten
df_ts_Demand_long
```

Period <chr>	Region <chr>	Demand <int>
1	Japan	18147
1	Peking	16574
1	Phlppn	15494
1	Shangh	15471
1	Skorea	17240
2	Japan	21957
2	Peking	19168
2	Phlppn	16660
2	Shangh	15256
2	Skorea	18196

1-10 of 300 rows

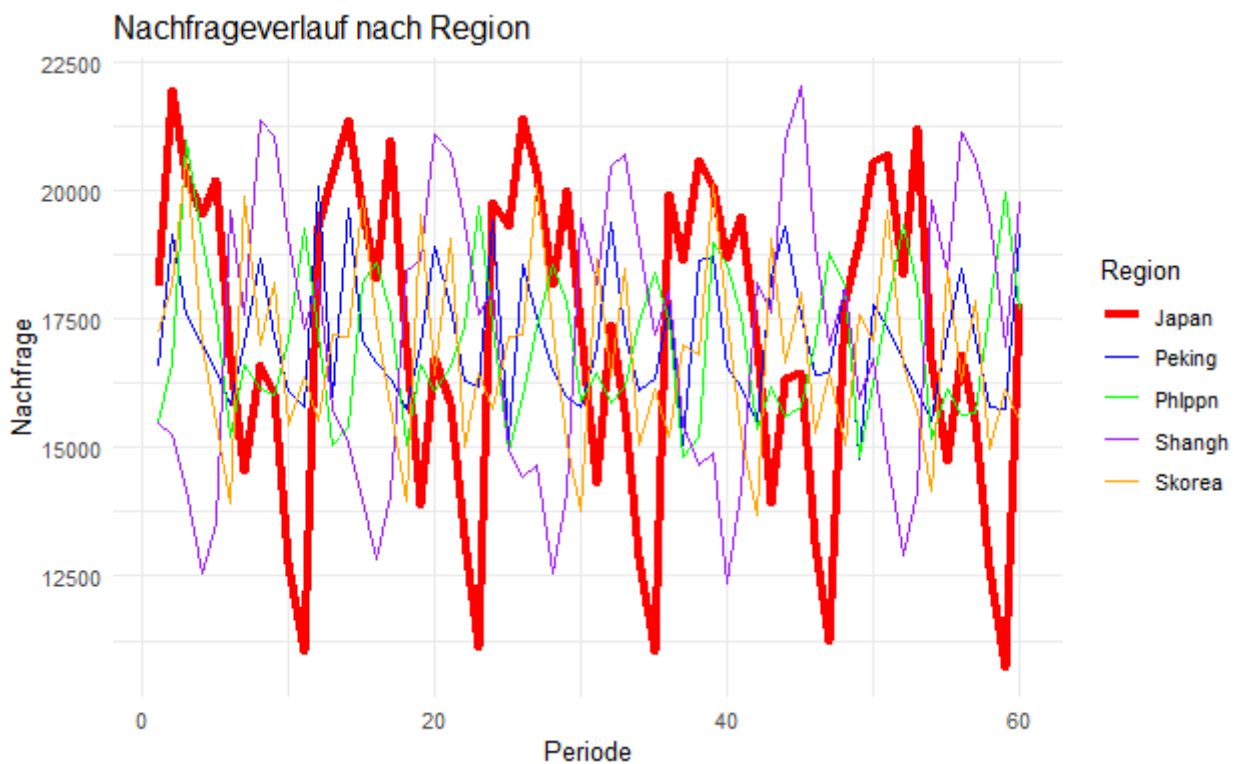
Previous
1
2
3
4
5
6
...
30
Next

[Hide](#)

```

ggplot() +
  geom_line(data = df_ts_Demand_long %>% filter(Region == "Japan"),
            aes(x = as.numeric(Period), y = Demand, color = "Japan"), size = 1.5) +
  geom_line(data = df_ts_Demand_long %>% filter(Region != "Japan"),
            aes(x = as.numeric(Period), y = Demand, color = Region), alpha = 1) +
  scale_color_manual(values = c("Japan" = "red", "Peking" = "blue", "Phlppn" = "green", "Shan
gh" = "purple", "Skorea" = "orange")) +
  labs(title = "Nachfrageverlauf nach Region",
        x = "Periode",
        y = "Nachfrage",
        color = "Region") +
  theme_minimal()

```



Hide

Hier unten ist der ursprünglichen Code, der irgendwie ab heute Abend (04.06) nicht mehr funktioniert. Daher musste ich den Code von Zeile 369 bis Zeile 392 hinzufügen, der ziemlich kompliziert ist.

Visualisierung des Nachfrageverlaufs aller Regionen

```
ggplot(ts_Demand, aes(x = Period)) +  
  geom_line(aes(y = Japan, color = "Japan"), size = 1.5) + # Japan  
  geom_line(aes(y = Peking, color = "Peking"), alpha = 1) + # Peking  
  geom_line(aes(y = Phlppn, color = "Phlppn"), alpha = 1) + # Phlppn  
  geom_line(aes(y = Shangh, color = "Shangh"), alpha = 1) + # Shangh  
  geom_line(aes(y = Skorea, color = "Skorea"), alpha = 1) + # Skorea  
  labs(title = "Nachfrageverlauf nach Region",  
        x = "Periode",  
        y = "Nachfrage",  
        color = "Region") +  
  scale_color_manual(values = c("Japan" = "red", "Peking" = "blue", "Phlppn" = "green", "Shangh" = "purple", "Skorea" = "orange")) +  
  theme_minimal()
```

Error in `geom_line()`:

! Problem while computing aesthetics.

! Error occurred in the 1st layer.

Caused by error in `FUN()`:

! Objekt 'Japan' nicht gefunden

Backtrace:

1. base (local) ``(x)
2. ggplot2::print.ggplot(x)
4. ggplot2::ggplot_build.ggplot(x)
5. ggplot2::by_layer(...)
12. ggplot2 (local) f(l = layers[[i]], d = data[[i]])
13. l\$compute_aesthetics(d, plot)
14. ggplot2 (local) compute_aesthetics(..., self = self)
15. base::lapply(aesthetics, eval_tidy, data = data, env = env)
16. rlang (local) FUN(X[[i]], ...)

5. Betrachten Sie die Visualisierung und nehmen Sie zu möglichen Trends und Saisonalitäten für die Region Japan Stellung. Erachten Sie eine Zeitreihenanalyse in diesem Fall als sinnvolle Methode, um den Nachfrageverlauf zu modellieren? Angenommen, dies ist der Fall: Welche Annahme müssen Sie dann treffen, um die Ergebnisse der Modellierung für Ihr Produkt nutzen zu können? Bewertungsrelevant: Kommentar.

Hide

```
## Es wird festgestellt, dass die Nachfrage von Januar bis Mai besonders hoch zu sein scheint. Währenddessen steigt die Nachfrage ab Juni deutlich ab und erreicht im November ihren Tiefpunkt. Danach erhöht sie sich wieder im Dezember. Dies könnte auf saisonale Faktoren wie Wetterbedingungen zurückzuführen sein.
```

```
## Ja, eine Zeitreihenanalyse wäre eine sinnvolle Methode, um den Nachfrageverlauf zu modellieren. Dadurch wird ermöglicht, die saisonalen Muster und Trends in den Daten zu identifizieren und zukünftige Nachfrageprognosen zu erstellen.
```

```
## Um die Ergebnisse der Modellierung nutzen zu können, müssen Annahmen getroffen werden, dass die identifizierten Trends und saisonalen Muster stabil bleiben und weiterhin relevant sind. Das heißt, man geht davon aus, dass sich die historischen Muster in der Zukunft fortsetzen werden.
```

**** Modellierung 6)** Nutzen Sie nun Ihre exemplarische Region Japan weiter. Führen Sie mit Hilfe der `ets()`-Funktion eine Zeitreihenanalyse durch. Erstellen Sie ein Modell, indem Sie die automatische Festlegung der Modellparameter nutzen (`model = "ZZZ"`). Geben Sie (1) eine Zusammenfassung des Modells, (2) die ursprüngliche Zeitreihe und (3) die Werte der Residuen aus. Bewertungsrelevant: Code, Output.

Hide


```
# Zeitreihenanalyse für die Region Japan mit der ets()-Funktion
japan_demand <- subset(Demand, Region == "Japan")
japan_ts <- ts(japan_demand$Demand, frequency = 12, start = c(2019, 1))
# Modell erstellen mit automatischer Parameterfestlegung
model_japan <- ets(japan_ts, model = "ZZZ")

# Ausgabe der Modellzusammenfassung
summary(model_japan)
```

ETS(M,N,M)

Call:

```
ets(y = japan_ts, model = "ZZZ")
```

Smoothing parameters:

```
alpha = 0.0437
```

```
gamma = 1e-04
```

Initial states:

```
l = 17303.8777
```

```
s = 1.1061 0.6417 0.7512 0.9234 0.9734 0.8302
```

```
0.9843 1.1872 1.0816 1.1776 1.2317 1.1115
```

```
sigma: 0.031
```

```
AIC      AICc      BIC
```

```
1011.483 1022.392 1042.899
```

Training set error measures:

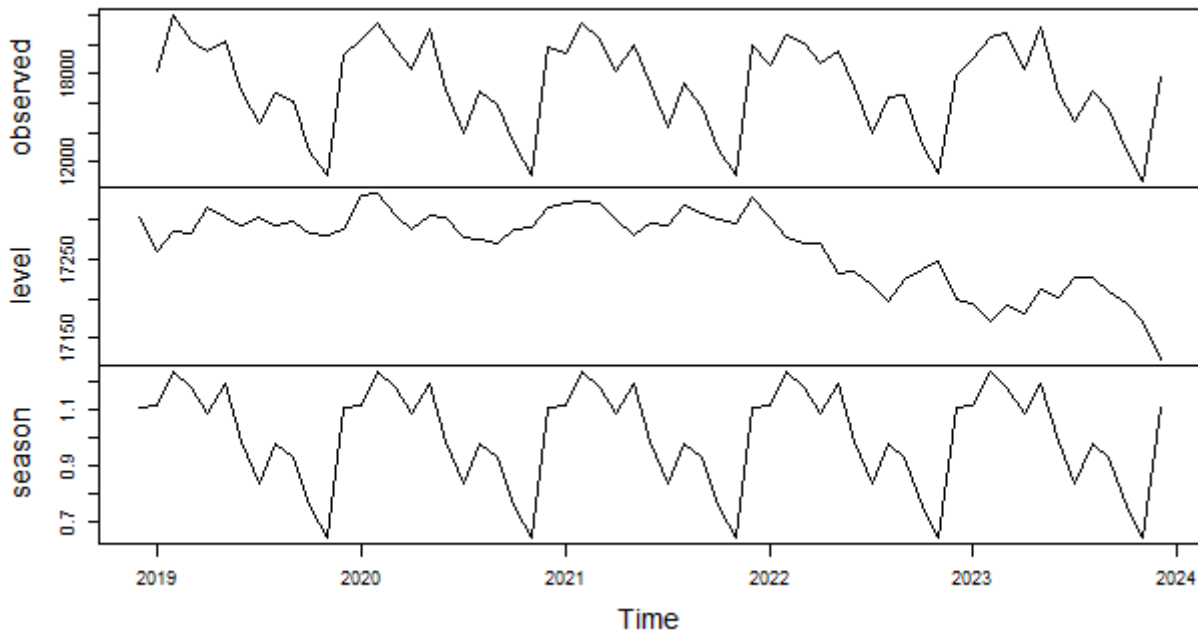
```
ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
```

```
Training set -72.39122 503.0464 384.9499 -0.4715475 2.17211 0.6768846 -0.08299308
```

Hide

```
# Anzeige Ursprüngliche Zeitreihe und Residuen
plot(model_japan)
```

Decomposition by ETS(M,N,M) method



7. Geben Sie fuer das Jahr 2023 die durchschnittliche Höhe der Originalwerte sowie die durchschnittliche Höhe der Modellwerte an. Nutzen Sie dafür nur die eben erstellte Modellvariable. Vergleichen Sie anschliessend die Werte. Welche Auswirkungen hätte es gehabt, wenn das Unternehmen im Jahr 2023 das Modell bereits genutzt hätte, um Bestellmengen zu prognostizieren? Bewertungsrelevant: Code, Output, Kommentar.

Hide

```
# Durchschnittliche Höhe der Original- und Modellwerte für 2023
japan_2023 <- window(japan_ts, start = c(2023, 1), end = c(2023, 12))
model_2023 <- window(fitted(model_japan), start = c(2023, 1), end = c(2023, 12))

mean_original_2023 <- mean(japan_2023)
mean_model_2023 <- mean(model_2023)

cat("Durchschnittliche Höhe der Originalwerte 2023:", mean_original_2023, "\n")
```

Durchschnittliche Höhe der Originalwerte 2023: 17065.58

Hide

```
cat("Durchschnittliche Höhe der Modellwerte 2023:", mean_model_2023, "\n")
```

Durchschnittliche Höhe der Modellwerte 2023: 17197.3

Hide

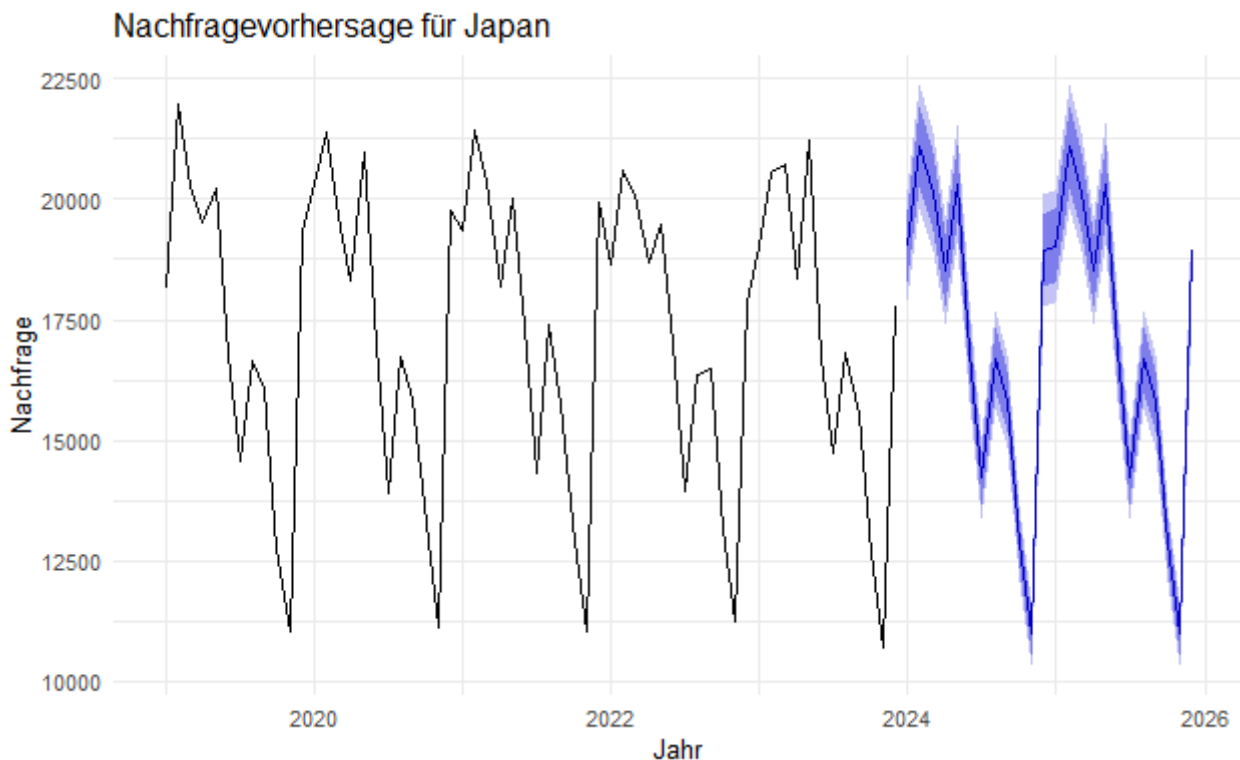
```
## Wenn wir das Modell im Jahr 2023 genutzt hätten, hätten wir präzisere Bestellmengen prognostizieren können, was die Effizienz der Bestandsverwaltung verbessert hätte.
```

8. Erstellen Sie eine Nachfragevorhersage für zwei weitere Jahre. Visualisieren Sie den Nachfrageverlauf sowie den Verlauf Ihres Prognosemodells in einem einzigen ggplot. Begründen Sie die Wahl der

Hide

```
# Nachfragevorhersage für zwei Jahre erstellen
forecast_japan <- forecast(model_japan, h = 24)

# Visualisierung der Nachfrageverlauf und des Prognosemodells
autoplot(forecast_japan) +
  ggtitle("Nachfragevorhersage für Japan") +
  xlab("Jahr") + ylab("Nachfrage") +
  theme_minimal()
```



Hide

```
# Wahl der Visualisierung begründen
cat("Die Wahl der Visualisierung zeigt den Verlauf der tatsächlichen Nachfrage und die prognostizierten Werte klar und deutlich, was die Nachvollziehbarkeit und Entscheidungsfindung erleichtert.\n")
```

Die Wahl der Visualisierung zeigt den Verlauf der tatsächlichen Nachfrage und die prognostizierten Werte klar und deutlich, was die Nachvollziehbarkeit und Entscheidungsfindung erleichtert.

9. Bewerten Sie Ihr Modell aus Aufgabe 6 mit Hilfe von vier verschiedenen Fehler-Kennzahlen, die Sie aus der Übung kennen. Welche der Fehler-Kennzahlen halten Sie für geeignet, um die Güte des Modells zu bewerten? Bewertungsrelevant: Output, Kommentar.

Hide

```
# Berechnung der Fehler-Kennzahlen
mfe = mean(residuals(model_japan))
mae <- mean(abs(residuals(model_japan)))
mse <- mean(residuals(model_japan)^2)
rmse <- sqrt(mse)
mape <- mean(abs(residuals(model_japan) / japan_ts)) * 100

cat("MFE:", mfe, "\n")
```

MFE: -0.00397015

Hide

```
cat("MAE:", mae, "\n")
```

MAE: 0.02154029

Hide

```
cat("MSE:", mse, "\n")
```

MSE: 0.0007356662

Hide

```
cat("RMSE:", rmse, "\n")
```

RMSE: 0.02712317

Hide

```
cat("MAPE:", mape, "%\n")
```

MAPE: 0.0001251326 %

Hide

MAPE ist besonders geeignet, da sie die durchschnittliche prozentuale Abweichung der Prognosewerte von den tatsächlichen Werten zeigt, was für die Bewertung der Modellgüte entscheidend ist.

10. Vergleichen Sie Ihr Modell insgesamt mit Ihren Vermutungen aus Aufgabe 5. Was stellen Sie fest?
Bewertungsrelevant: Kommentar.

Hide

Das Modell bestätigt die Trends und Saisonalitäten, die in der Aufgabe 5 angenommen wurden. Die Zeitreihenanalyse hat sich als sinnvoll erwiesen, um den Nachfrageverlauf zu modellieren.

11. Ihr Kollege Doug und Ihre Kollegin Carrie diskutieren. Doug ist der Meinung, er könnte möglicherweise ein besseres Modell entwickeln als Sie. "Ich habe schonmal bei anderen Modellen die Modellparameter `model = "ANA"` genutzt, statt `model = "ZZZ"`. Damit hatte ich damals bessere MAPE-Werte erzielt. Vielleicht ist das hier wieder der Fall?", überlegt er in der Mittagspause. "Unsinn!", wirft Carrie bestimmt ein. "Die Modellparameter `model = "ZZZ"` ergeben immer das optimale Modell. Das ZZZ-Modell ist auf jeden Fall das Beste." Vergleichen Sie die Modelle. Wer hat Recht? Bewertungsrelevant: Output, Kommentar.

Hide

```
# Erstellung des Modells mit "ANA" und Berechnung der Fehler-Kennzahlen
model_japan_ana <- ets(japan_ts, model = "ANA")
mfe_ana = mean(residuals(model_japan_ana))
mae_ana <- mean(abs(residuals(model_japan_ana)))
mse_ana <- mean(residuals(model_japan_ana)^2)
rmse_ana <- sqrt(mse)
mape_ana <- mean(abs(residuals(model_japan_ana) / japan_ts)) * 100

# Erstellung einer Tabelle zum Vergleichen der Modelle
c(mfe, mae, mse, rmse, mape)
```

```
[1] -0.0039701501  0.0215402892  0.0007356662  0.0271231680  0.0001251326
```

Hide

```
c(mfe_ana, mae_ana, mse_ana, rmse_ana, mape_ana)
```

```
[1] -8.722656e+01  4.018456e+02  2.559315e+05  2.712317e-02  2.295935e+00
```

Hide

```
vergleich <- matrix(c(mfe, mae, mse, rmse, mape, mfe_ana, mae_ana, mse_ana, rmse_ana, mape_ana), ncol=5, byrow=TRUE)
colnames(vergleich) <- c('MFE', 'MAE', 'MSE', 'RMSE', 'MAPE')
rownames(vergleich) <- c('ZZZ', 'ANA')
vergleich <- as.table(vergleich)
vergleich
```

	MFE	MAE	MSE	RMSE	MAPE
ZZZ	-3.970150e-03	2.154029e-02	7.356662e-04	2.712317e-02	1.251326e-04
ANA	-8.722656e+01	4.018456e+02	2.559315e+05	2.712317e-02	2.295935e+00

Hide

```
# Aus der obigen Tabelle wird festgestellt, dass das Modell 'ZZZ' optimal ist. Damit hat Carrie Recht.
```

12. Hinweis: Gehen Sie unabhängig der Ergebnisse der letzten Aufgabe davon aus, dass das ZZZ-Modell das beste Modell ist und weiterhin genutzt wird. Erstellen Sie für die übrigen Regionen ebenso Modelle zur Nachfragevorhersage (d.h. 4 weitere Modelle). Nutzen Sie erneut die automatische Festlegung der Modellparameter (`model = "ZZZ"`). Berechnen Sie zudem für jedes Modell den MAPE, um den Vergleich der Modelle zu ermöglichen. Nehmen Sie dazu Stellung, welches Modell laut der bewertenden Information das "beste" der fünf Modelle sei. Bewertungsrelevant: Output, Kommentar.

[Hide](#)

```
# Erstellung von Modellen für andere Regionen
regions <- unique(Demand$Region)
regions <- regions[regions != "Japan"]

mape_values <- c()

for(region in regions) {
  region_demand <- subset(Demand, Region == region)
  region_ts <- ts(region_demand$Demand, frequency = 12, start = c(2019, 1))

  model_region <- ets(region_ts, model = "ZZZ")
  mape_region <- mean(abs(residuals(model_region) / region_ts)) * 100
  mape_values <- c(mape_values, mape_region)

  cat("MAPE für", region, ":", mape_region, "%\n")
}
```

```
MAPE für Peking : 0.0001157814 %
MAPE für Phlppn : 0.0001172917 %
MAPE für Shangh : 0.0001165291 %
MAPE für Skorea : 8.467106e-05 %
```

[Hide](#)

```
best_model_region <- regions[which.min(mape_values)]
cat("Das beste Modell ist für die Region:", best_model_region, "\n")
```

```
Das beste Modell ist für die Region: Skorea
```

[Hide](#)

```
## Das beste Modell ist das von Südkorea, weil seine MAPE am niedrigsten ist.
```

**** Abschluss 13)** Ihre Chefin ist begeistert von Ihren fünf Vorhersagemodellen. “Das bedeutet ja, dass wir genau voraussagen können, wie viele Flaschen Limonade insgesamt für das zweite Quartal von 2024 benötigt werden!”, sagt sie freudestrahlend. “Können Sie mir dafür eine Zahl nennen?”. Helfen Sie Ihrer Chefin.
Bewertungsrelevant: Output, Kommentar.

[Hide](#)

```
# Vorhersage für das zweite Quartal 2024
forecast_total <- 0
for(region in regions) {
  region_demand <- subset(Demand, Region == region)
  region_ts <- ts(region_demand$Demand, frequency = 12, start = c(2019, 1))

  model_region <- ets(region_ts, model = "ZZZ")
  forecast_region <- forecast(model_region, h = 6) # Vorhersage für die ersten 6 Monate 2024
  forecast_total <- forecast_total + sum(forecast_region$mean[4:6]) # Zweites Quartal
}

# Ausgabe der vorgegesagten Zahl an Flaschen Limonade
cat("Der vorhergesagte Bedarf für das zweite Quartal 2024 beträgt:", forecast_total, "Flasche
n.\n")
```

Der vorhergesagte Bedarf für das zweite Quartal 2024 beträgt: 192546.2 Flaschen.

14. Ihr guter Freund Arthur wohnt in Shanghai und leitet dort die fünf Supermärkte, die Ihre Limonaden-Marke führen. "Ich habe das Gefühl, im Oktober 2022 haben wir deutlich weniger Flaschen Limonade verkauft, als für den Monat üblich ist!", sagt er. "Kannst du mir sagen, ob das im nächsten Jahr auch so sein wird?" Helfen Sie Arthur. Bewertungsrelevant: Output, Kommentar.

Hide

```
# Nachfragevorhersage für Oktober 2023 und Vergleich mit Oktober 2022
shanghai_demand <- subset(Demand, Region == "Shangh")
shanghai_ts <- ts(shanghai_demand$Demand, frequency = 12, start = c(2019, 1))
# Ausgabe der Zeitreihe von Shanghai zum Ermöglichen der Bewertung unten
print("Zeitreihen von Shanghai")
```

```
[1] "Zeitreihen von Shanghai"
```

Hide

```
shanghai_ts
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
2019 15471 15256 14185 12571 13532 19643 17608 21391 21072 19064 17306 18288
2020 15763 15082 13992 12838 14081 18446 18651 21108 20748 19402 17604 18078
2021 14980 14441 14682 12543 14114 19489 18184 20494 20721 18992 17189 18067
2022 15429 14675 14889 12367 14211 18228 17632 21013 22044 18967 17023 18103
2023 15954 16750 14809 12924 14179 19826 18420 21147 20624 19557 16969 19769
```

Hide

```
# Vorhersage für zwei Jahre
model_shanghai <- ets(shanghai_ts, model = "ZZZ")
forecast_shanghai <- forecast(model_shanghai, h = 24)
# Ausgabe der Nachfrage im Oktober 2022 und Oktober 2023
october_2022 = window(shanghai_ts, start = c(2022, 10), end = c(2022,10))
october_2023 <- window(shanghai_ts, start = c(2023, 10), end = c(2023,10))
#october_2023 <- window(forecast_shanghai$mean, start = c(2023, 10), end = c(2023,10))
cat("Verkaufte Flaschen im Oktober 2022: ", sum(october_2022), "\n")
```

Verkaufte Flaschen im Oktober 2022: 18967

Hide

```
cat("Verkaufte Flaschen im Oktober 2023: ", sum(october_2023), "\n")
```

Verkaufte Flaschen im Oktober 2023: 19557

Hide

```
# Ausgabe der vorhergesagten Nachfrage
cat("Vorhergesagte (nächstes Fenster)"#, forecast_shanghai, "\n")
```

Vorhergesagte (nächstes Fenster)

Hide

```
print(forecast_shanghai)
```

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
Jan 2024	15671.51	15042.37	16300.66	14709.32	16633.71
Feb 2024	15467.63	14843.64	16091.63	14513.32	16421.95
Mar 2024	14839.68	14237.71	15441.66	13919.04	15760.33
Apr 2024	12890.82	12362.94	13418.70	12083.49	13698.15
May 2024	14366.11	13777.88	14954.35	13466.48	15265.75
Jun 2024	19225.31	18442.83	20007.79	18028.62	20422.00
Jul 2024	18333.31	17582.34	19084.28	17184.80	19481.82
Aug 2024	21296.63	20425.78	22167.49	19964.77	22628.50
Sep 2024	21457.55	20576.38	22338.72	20109.92	22805.18
Oct 2024	19364.77	18561.12	20168.41	18135.70	20593.84

1-10 of 24 rows

Previous 1 2 3 Next

Hide

Die Vorhersage zeigt, dass die Nachfrage im Oktober 2024 voraussichtlich ähnlich wie sein sollte. Auch wenn man die Time-Series von Shanghai betrachtet, kann man auch feststellen, dass die Nachfragen in jedem Oktober 2019 bis 2023 immer ungefähr gleich beträgt.