

# Gruppe 105 Hausaufgabe 1

Supply Chain Analytics SS23

Cordelia Mena Hernandez, Daniel Glatter

2023-05-09

## Übersicht

### Aufgabe 1

```
# Laden der Daten in die Dataframes 'cost', 'services', 'prices' und 'transactions'
cost <- read.csv("data/output_cost_8Players_v0020.csv", sep=";", dec=",")
services <- read.csv("data/output_services_8Players_v0020.csv", sep=";", dec=",")
prices <- read.csv("data/output_prices_8Players_v0020.csv", sep=";", dec=",")
transactions <- read.csv("data/output_transactions_8Players_v0020.csv", sep=";", dec=",")
```

```
# Zusammenfassung der Daten 'cost' anzeigen
summary(cost)
```

```
##      Year      Month      Product      Amount
## Min.   :2018   Min.    : 1.00   Length:480   Min.    :25685
## 1st Qu.:2019   1st Qu.: 3.75   Class :character 1st Qu.:31609
## Median :2020   Median : 6.50   Mode  :character Median :32687
## Mean   :2020   Mean    : 6.50               Mean   :32688
## 3rd Qu.:2021   3rd Qu.: 9.25               3rd Qu.:33785
## Max.    :2022   Max.    :12.00               Max.    :36620
```

```
# Zusammenfassung der Daten 'services' anzeigen
summary(services)
```

```
##      Year      Month      Day      region
## Min.   :2018   Min.    : 1.000   Min.    : 1.0   Length:107058
## 1st Qu.:2019   1st Qu.: 4.000   1st Qu.: 5.0   Class :character
## Median :2020   Median : 7.000   Median :11.0   Mode  :character
## Mean   :2020   Mean    : 6.503   Mean    :10.5
## 3rd Qu.:2021   3rd Qu.: 9.000   3rd Qu.:15.0
## Max.    :2022   Max.    :12.000   Max.    :20.0
## storename      Product      vendor      service
## Length:107058   Length:107058   Length:107058   Length:107058
## Class :character Class :character Class :character Class :character
## Mode  :character Mode  :character Mode  :character Mode  :character
##
##
##
## DaysScheduled DaysExecuted  QScheduled  QExecuted      cost
## Min.   :0      Min.   :0.000   Min.   : 31.0   Min.   : 19.0   Min.   : 36.46
## 1st Qu.:0      1st Qu.:0.000   1st Qu.: 79.0   1st Qu.: 69.0   1st Qu.:103.56
```

```
## Median :1      Median :1.000  Median :108.0  Median : 97.0  Median :139.41
## Mean :1      Mean :1.364  Mean :112.5  Mean :102.8  Mean :146.56
## 3rd Qu.:2      3rd Qu.:3.000  3rd Qu.:144.0  3rd Qu.:134.0  3rd Qu.:188.71
## Max. :2      Max. :4.000  Max. :286.0  Max. :286.0  Max. :395.99
```

```
# Zusammenfassung der Daten 'prices' anzeigen
summary(prices)
```

```
##      vendor      service      PriceSpot      PriceContract
## Length:20      Length:20      Min. :1.211      Min. :0.7268
## Class :character      Class :character      1st Qu.:1.300      1st Qu.:0.7801
## Mode :character      Mode :character      Median :1.312      Median :0.7874
##                                     Mean :1.313      Mean :0.7880
##                                     3rd Qu.:1.338      3rd Qu.:0.8026
##                                     Max. :1.374      Max. :0.8247
## FeeContractPerUnit      PriceChange      UsageLastPeriod
## Min. :0.1938      Min. : -0.065250      Min. : 7178
## 1st Qu.:0.2080      1st Qu.: -0.043548      1st Qu.: 8992
## Median :0.2100      Median : -0.015248      Median :10000
## Mean :0.2101      Mean : -0.014532      Mean : 9821
## 3rd Qu.:0.2140      3rd Qu.: 0.007333      3rd Qu.:10707
## Max. :0.2199      Max. : 0.043387      Max. :11255
```

```
# Zusammenfassung der Daten 'transactions' anzeigen
summary(transactions)
```

```
##      Year      Month      Day      region
## Min. :2018      Min. : 1.000      Min. : 1.00      Length:296577
## 1st Qu.:2019      1st Qu.: 3.000      1st Qu.: 5.00      Class :character
## Median :2020      Median : 6.000      Median :10.00      Mode :character
## Mean :2020      Mean : 6.493      Mean :10.46
## 3rd Qu.:2021      3rd Qu.: 9.000      3rd Qu.:15.00
## Max. :2023      Max. :12.000      Max. :20.00
## storename      Product      Sales      Received
## Length:296577      Length:296577      Min. : 0.00      Min. : 0.00
## Class :character      Class :character      1st Qu.: 0.00      1st Qu.: 0.00
## Mode :character      Mode :character      Median : 16.00      Median : 0.00
##                                     Mean : 17.22      Mean : 16.88
##                                     3rd Qu.: 28.00      3rd Qu.: 0.00
##                                     Max. :209.00      Max. :256.00
```

## Aufgabe 2

### Vorbereitung der Daten

```
# Neue Spalte 'Date' im 'cost' Dataframe mit den Daten aus den Spalten 'Year' und 'Month'
cost$Date <- make_date(cost$Year, cost$Month)

# Löschen der Spalten 'Year' und 'Month, da sie nicht mehr nötig sind
cost <- subset(cost, select=-c(Year, Month))

# Neue Spalte 'Date' im 'services' Dataframe mit den Daten aus den Spalten 'Year',
↳ 'Month' und 'Day'
services$Date <- make_date(services$Year, services$Month, services$Day)
```

```
# Löschen der Spalten 'Year', 'Month' und 'Day' , da sie nicht mehr nötig sind
services <- subset(services, select=-c(Year, Month, Day))
```

```
# Analog für 'transactions'
transactions$Date <- make_date(transactions$Year, transactions$Month, transactions$Day)
transactions <- subset(transactions, select=-c(Year, Month, Day))
```

```
# Intervallvariable von 2018-2022
year18_22 <- interval(ymd("2018-01-01"), ymd("2022-12-31"))
# Einzelne Dataframes mit Daten in der Zeitspanne von 2018 bis 2022
cost18_22 <- cost[cost$Date %within% year18_22, ]
services18_22 <- services[services$Date %within% year18_22, ]
transactions18_22 <- transactions[transactions$Date %within% year18_22, ]
```

```
# Zusammenfassung der Daten 'cost' zwischen 2018 und 2022 anzeigen
summary(cost18_22)
```

```
##      Product          Amount          Date
## Length:480      Min.   :25685      Min.   :2018-01-01
## Class :character 1st Qu.:31609      1st Qu.:2019-03-24
## Mode  :character Median :32687      Median :2020-06-16
##                      Mean  :32688      Mean  :2020-06-16
##                      3rd Qu.:33785      3rd Qu.:2021-09-08
##                      Max.   :36620      Max.   :2022-12-01
```

```
# Zusammenfassung der Daten 'services' zwischen 2018 und 2022 anzeigen
summary(services18_22)
```

```
##      region          storename          Product          vendor
## Length:107058      Length:107058      Length:107058      Length:107058
## Class :character   Class :character   Class :character   Class :character
## Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##      service          DaysScheduled  DaysExecuted          QScheduled
## Length:107058      Min.   :0      Min.   :0.000      Min.   : 31.0
## Class :character   1st Qu.:0      1st Qu.:0.000      1st Qu.: 79.0
## Mode  :character   Median :1      Median :1.000      Median :108.0
##                      Mean  :1      Mean  :1.364      Mean  :112.5
##                      3rd Qu.:2      3rd Qu.:3.000      3rd Qu.:144.0
##                      Max.   :2      Max.   :4.000      Max.   :286.0
##      QExecuted          cost          Date
## Min.   : 19.0      Min.   : 36.46      Min.   :2018-01-03
## 1st Qu.: 69.0      1st Qu.:103.56      1st Qu.:2019-04-01
## Median : 97.0      Median :139.41      Median :2020-07-01
## Mean   :102.8      Mean   :146.56      Mean   :2020-06-25
## 3rd Qu.:134.0      3rd Qu.:188.71      3rd Qu.:2021-09-20
## Max.   :286.0      Max.   :395.99      Max.   :2022-12-20
```

```
# Zusammenfassung der Daten 'prices' zwischen 2018 und 2022 anzeigen
summary(prices)
```

```
##      vendor          service          PriceSpot          PriceContract
```

```
## Length:20      Length:20      Min.   :1.211   Min.   :0.7268
## Class :character Class :character 1st Qu.:1.300   1st Qu.:0.7801
## Mode  :character Mode  :character Median :1.312   Median :0.7874
##                                     Mean  :1.313   Mean  :0.7880
##                                     3rd Qu.:1.338   3rd Qu.:0.8026
##                                     Max.   :1.374   Max.   :0.8247
## FeeContractPerUnit PriceChange UsageLastPeriod
## Min.   :0.1938   Min.   :-0.065250 Min.   : 7178
## 1st Qu.:0.2080   1st Qu.: -0.043548 1st Qu.: 8992
## Median :0.2100   Median :-0.015248 Median :10000
## Mean   :0.2101   Mean   :-0.014532 Mean   : 9821
## 3rd Qu.:0.2140   3rd Qu.: 0.007333 3rd Qu.:10707
## Max.   :0.2199   Max.   : 0.043387 Max.   :11255
```

```
# Zusammenfassung der Daten 'transactions' zwischen 2018 und 2022 anzeigen
summary(transactions18_22)
```

```
##      region      storename      Product      Sales
## Length:296458 Length:296458 Length:296458 Min.   : 0.00
## Class :character Class :character Class :character 1st Qu.: 0.00
## Mode  :character Mode  :character Mode  :character Median : 16.00
##                                     Mean  : 17.23
##                                     3rd Qu.: 28.00
##                                     Max.   :209.00
##      Received      Date
## Min.   : 0.00   Min.   :2018-01-01
## 1st Qu.: 0.00   1st Qu.:2019-03-20
## Median : 0.00   Median :2020-06-20
## Mean   : 16.85   Mean   :2020-06-25
## 3rd Qu.: 0.00   3rd Qu.:2021-09-20
## Max.   :256.00   Max.   :2022-12-20
```

### Aufgabe 3

```
# Extrahieren der Regionen, in dem Verkauf oder Warenempfang stattgefunden haben
transactions_regions <- data.frame(regions=unique(transactions18_22$region))
# Nutzen der kable() Funktion von Knitr für bessere Tabellendarstellung
transactions_regions %>% kable(row.names = FALSE, caption = 'Absatzregionen')
```

Table 1: Absatzregionen

regions
Shangh
Peking
Japan
Skorea
Phlppn

### Aufgabe 4

```
# Selektion der Spalten 'vendor' und 'service'
service_vendors <- services18_22[, c("vendor", "service")]
```

```

# Liste der einzelnen Warehousing- und Shipping-Dienstleister
service_vendors <- unique(service_vendors)

# Sortieren der Tabelle nach Shipping-DL und Warehousing-DL
service_vendors <- service_vendors[order(service_vendors$service),]

service_vendors %>% kable(row.names = FALSE, caption = 'Dienstleister')

```

Table 2: Dienstleister

vendor	service
EPD Shipping	Shipping
DWL Shipping	Shipping
CPS Shipping	Shipping
AHL Express Shipping	Shipping
IntEx Shipping	Shipping
HCX Shipping	Shipping
Flying Mercury Shipping	Shipping
JNT Shipping	Shipping
Bange+Hammer Shipping	Shipping
Gifter Shipping	Shipping
IntEx Warehousing	Warehousing
Flying Mercury Warehousing	Warehousing
JNT Warehousing	Warehousing
AHL Express Warehousing	Warehousing
DWL Warehousing	Warehousing
CPS Warehousing	Warehousing
HCX Warehousing	Warehousing
EPD Warehousing	Warehousing
Bange+Hammer Warehousing	Warehousing
Gifter Warehousing	Warehousing

## Aufgabe 5

```

# Selektion der Spalten 'region' und 'storename'
supermarkets <- transactions18_22[, c("region", "storename")]

# Selektion der einzelnen Supermärkte
supermarkets <- unique(supermarkets)

# Selektion aller existierenden Supermärkte in Shanghai, Peking und Südkorea
supermarkets <- subset(supermarkets, region %in% c('Shangh', 'Peking', 'Skorea'))

supermarkets %>% kable(row.names = FALSE, caption = 'Supermärkte in Shanghai, Peking und
↳ Südkorea')

```

Table 3: Supermärkte in Shanghai, Peking und Südkorea

region	storename
Shangh	Shangh-1
Shangh	Shangh-2
Shangh	Shangh-3
Shangh	Shangh-4
Shangh	Shangh-5
Peking	Peking-1
Peking	Peking-2
Peking	Peking-3
Peking	Peking-4
Peking	Peking-5
Skorea	Skorea-1
Skorea	Skorea-2
Skorea	Skorea-3
Skorea	Skorea-4
Skorea	Skorea-5

## Marktübersicht

### Aufgabe 6

```
# Aggregation der verkauften Limonaden der Gruppe 105 im Zeitraum 2018-2022
totalsales_105 <- sum(transactions18_22[transactions18_22$Product == "Gruppe105",4])

cat("Insgesamt wurden", totalsales_105, "Flaschen Limonade von Gruppe 105 verkauft.\n")

## Insgesamt wurden 619079 Flaschen Limonade von Gruppe 105 verkauft.

# Aggregation der gesamten verkauften Limonaden im Zeitraum 2018-2022
totalsales <- aggregate(transactions18_22$Sales, by=list(transactions18_22$Product),
  ↪ FUN=sum)

colnames(totalsales) <- c("Group", "Total Sales")

# Summe aller verkauften Limonaden ausgenommen der 'Lost Sales'
total_marketsales <- sum(totalsales[totalsales$Group != "Lost Sales",2])
# Berechnung des Marktanteils der Gruppe 105
marketshare <- totalsales[totalsales$Group=="Gruppe105", 2]/total_marketsales

cat("Bei Betrachtung der tatsächlich verkauften Menge (ohne Lost Sales) ergibt sich ein",
  ↪ "\n", "(gerundeter) Marktanteil von", round(marketshare, digits = 3)*100, "%.")

## Bei Betrachtung der tatsächlich verkauften Menge (ohne Lost Sales) ergibt sich ein
## (gerundeter) Marktanteil von 12.4 %.
```

### Aufgabe 7

```
# Selektion des Absatzes der Gruppe 105 in den 5 verschiedenen Regionen
regions_105 <- subset(transactions18_22, Product == 'Gruppe105', select =
  ↪ c('region', 'storename', 'Sales'))
```

```
# Gesamte Absatzmenge pro Region
regions_summary <- aggregate(regions_105$Sales, by=list(regions_105$region), FUN=sum)
colnames(regions_summary) <- c('Region', 'Total sales')

# Durchschnittliche Absatzmenge pro Supermarkt
regions_summary$`Avg Sales per Supermarket` <- regions_summary$`Total sales`/5

regions_summary %>% kable(row.names = FALSE, caption = 'Absatzdaten nach Region')
```

Table 4: Absatzdaten nach Region

Region	Total sales	Avg Sales per Supermarket
Japan	102513	20502.6
Peking	120743	24148.6
Philppn	155145	31029.0
Shangh	101495	20299.0
Skorea	139183	27836.6

```
cat("Der Unterschied zwischen der absatzstärksten und absatzschwächsten Region beträgt",
    round(1 - min(regions_summary$`Avg Sales per Supermarket`)/max(regions_summary$`Avg
    ↳ Sales per Supermarket`), 3)*100, "%."
)
```

## Der Unterschied zwischen der absatzstärksten und absatzschwächsten Region beträgt 34.6 %.  
Die Philippinen haben den größten Absatz, und Shanghai den niedrigsten. ## Aufgabe 8

```
# Die tatsächlich verkaufte Menge pro Jahr
yearly_marketshare <- transactions18_22 %>% filter(Product != 'Lost Sales') %>%
↳ group_by(year = floor_date(Date, 'year'), Product) %>% summarise(sum_sales =
↳ sum(Sales))

# Prozentualer Anteil der Gruppe105 an der tatsächlich verkauften Menge pro Jahr
marketshare105 <- yearly_marketshare %>% mutate(totalsales_year = sum(sum_sales)) %>%
↳ filter(Product=='Gruppe105') %>% mutate(relative_marketshare =
↳ sum_sales/totalsales_year*100)

marketshare105 %>% kable(row.names = FALSE, caption = 'Jährliche Umsatzdaten', digits =
↳ 2)
```

Table 5: Jährliche Umsatzdaten

year	Product	sum_sales	totalsales_year	relative_marketshare
2018-01-01	Gruppe105	123688	994787	12.43
2019-01-01	Gruppe105	123623	997038	12.40
2020-01-01	Gruppe105	124625	999671	12.47
2021-01-01	Gruppe105	123727	996823	12.41
2022-01-01	Gruppe105	123416	1000349	12.34

Der Marktanteil des Produkts 105 bleibt relativ konstant bei ungefähr 12,4%, dies ändert sich fast nicht über die Jahre.

## Aufgabe 9

```
# Berechnung des durchschnittlichen Umsatzes
monthlymean_sales <- transactions18_22 %>%
  filter(Product == 'Gruppe105') %>%
  # Nach Monat und Jahr aggregieren
  group_by(month_and_year = floor_date(Date, 'month')) %>%
  summarise(sales_per_month= sum(Sales)) %>%
  # Berechnen des Durchschnitts nach Jahren
  group_by(month=month(month_and_year)) %>%
  summarise(mean_sales= mean(sales_per_month))

monthlymean_sales %>% kable(row.names = FALSE, caption = 'Durchschnittliche
↪ Monatsumsätze')
```

Table 6: Durchschnittliche Monatsumsätze

month	mean_sales
1	10139.6
2	10410.0
3	9866.6
4	10366.8
5	9863.0
6	10059.6
7	10126.4
8	9918.8
9	10725.2
10	10930.6
11	11110.6
12	10298.6

```
# Maximum der durchschnittlichen Monatsumsätze
monthlymean_sales[which.max(monthlymean_sales$mean_sales),] %>% kable()
```

month	mean_sales
11	11110.6

Der Monat mit dem höchsten durchschnittlichen Absatz über alle in den Daten vorhandenen Jahre von etwa 11111 Flaschen ist der November. Insgesamt sind die Schwankungen über das Jahr hinweg aber eher gering, die Limonadennachfrage scheint relativ konstant, selbst im Sommer, wo man vielleicht Abweichungen erwarten würde.

## Umsatz, Kosten und Profit

### Aufgabe 10

```
# Gesamtumsatz der Gruppe 105 im Zeitraum 2018-2022
revenue_105 <- totalsales_105*3.9
revenue_105
```



```
## [1] 2414408
```

Der Gesamtumsatz im Zeitraum 2018-2022 beträgt ca. 2.41 Millionen Geldeinheiten.

```
# Gesamtkosten der Gruppe 105 im Zeitraum 2018-2022
```

```
transport_cost_germany_asia <- 0.6
```

```
totalcost_105 <- sum(cost18_22[cost18_22$Product == "Gruppe105", 2]) +  
  ↳ transport_cost_germany_asia * totalsales_105  
totalcost_105
```

```
## [1] 2317635
```

Die Gesamtkosten setzen sich zusammen aus den Transportkosten von Deutschland nach Asien sowie den Kosten für Logistikdienstleister im Warehousing der Großlager in Asien und den Transport von den Großlagern zu den Supermärkten als Endabnehmer. Die Gesamtkosten im Zeitraum 2018-2022 betragen damit ca. 2.32 Millionen Geldeinheiten.

```
# Gesamtprofit der Gruppe 105 im Zeitraum 2018-2022
```

```
profit_105 <- revenue_105 - totalcost_105  
profit_105
```

```
## [1] 96773.4
```

Der Gesamtprofit im Zeitraum 2018-2022 beträgt ca. 97.000 Geldeinheiten.

## Aufgabe 11

```
# Umsatz (revenue) aggregieren
```

```
rev_data105 <- transactions18_22[, c("Product", "Sales", "Date")]
```

```
rev_data105 <- subset(rev_data105, Product %in% c('Gruppe105'))
```

```
rev_data105 <- aggregate(rev_data105$Sales, by=list(floor_date(rev_data105$Date,  
  ↳ 'month')), FUN=sum)
```

```
rev_data105$x <- rev_data105$x*3.9
```

```
# Kosten aggregieren
```

```
cost_data105 <- subset(cost18_22, Product %in% c('Gruppe105'))
```

```
cost_data105 <- aggregate(cost_data105$Amount, by=list(floor_date(cost_data105$Date,  
  ↳ 'month')), FUN=sum)
```

```
cost_data105$x <- rev_data105$x / 3.9 * 0.6 # add transport costs from Germany to Asia
```

```
# Verbinden der DataFrames für monatliche Sales und Kosten
```

```
# Profit berechnen
```

```
profit_data105 <- merge(x = rev_data105, y = cost_data105, by = "Date", all = TRUE)
```

```
colnames(profit_data105) <- c('Date', 'Revenue', 'Cost')
```

```
profit_data105$Profit <- profit_data105$Revenue - profit_data105$Cost
```

```
# Gesamtprofit nur für das Jahr 2020
```

```
profit_data105 <- profit_data105[year(profit_data105$Date) == 2020,]
```

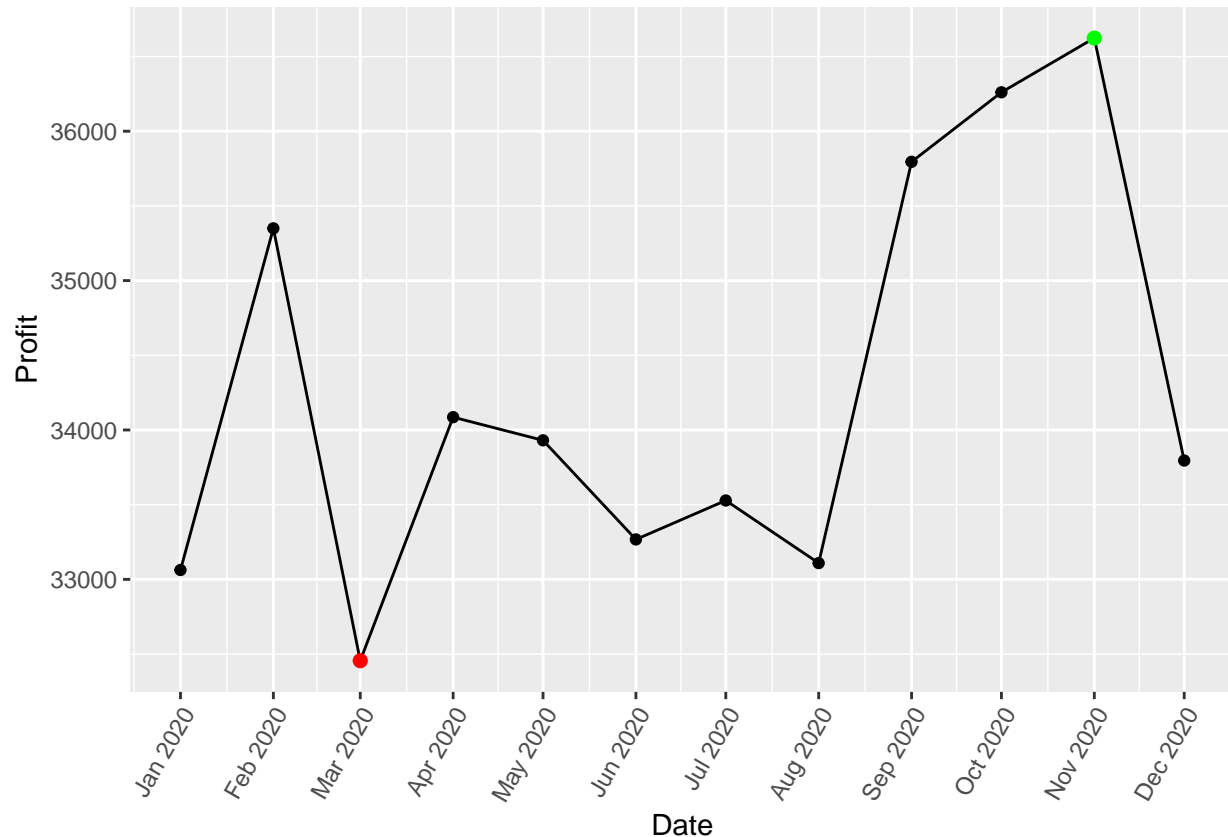
```
# Visualisierung des Gesamtprofits pro Monat im Jahr 2020
```

```
ggplot(data=profit_data105, aes(x=Date, y=Profit)) +  
  geom_line(color="black")+  
  geom_point()+
```

```

geom_point(data = profit_data105[which.min(profit_data105$Profit), ], color="red",
  ↪ size=2) +
geom_point(data = profit_data105[which.max(profit_data105$Profit), ], color="green",
  ↪ size=2)+
scale_x_date(date_labels = "%Y %b")+
scale_x_date(date_breaks = "1 month", date_labels = "%b %Y")+
theme(axis.text.x=element_text(angle=60, hjust=1))

```



Im März 2020 wird der kleinste, im November 2020 der größte Profit erzielt. Dieser enthält alle Kosten wie in Aufgabe 10 beschrieben.

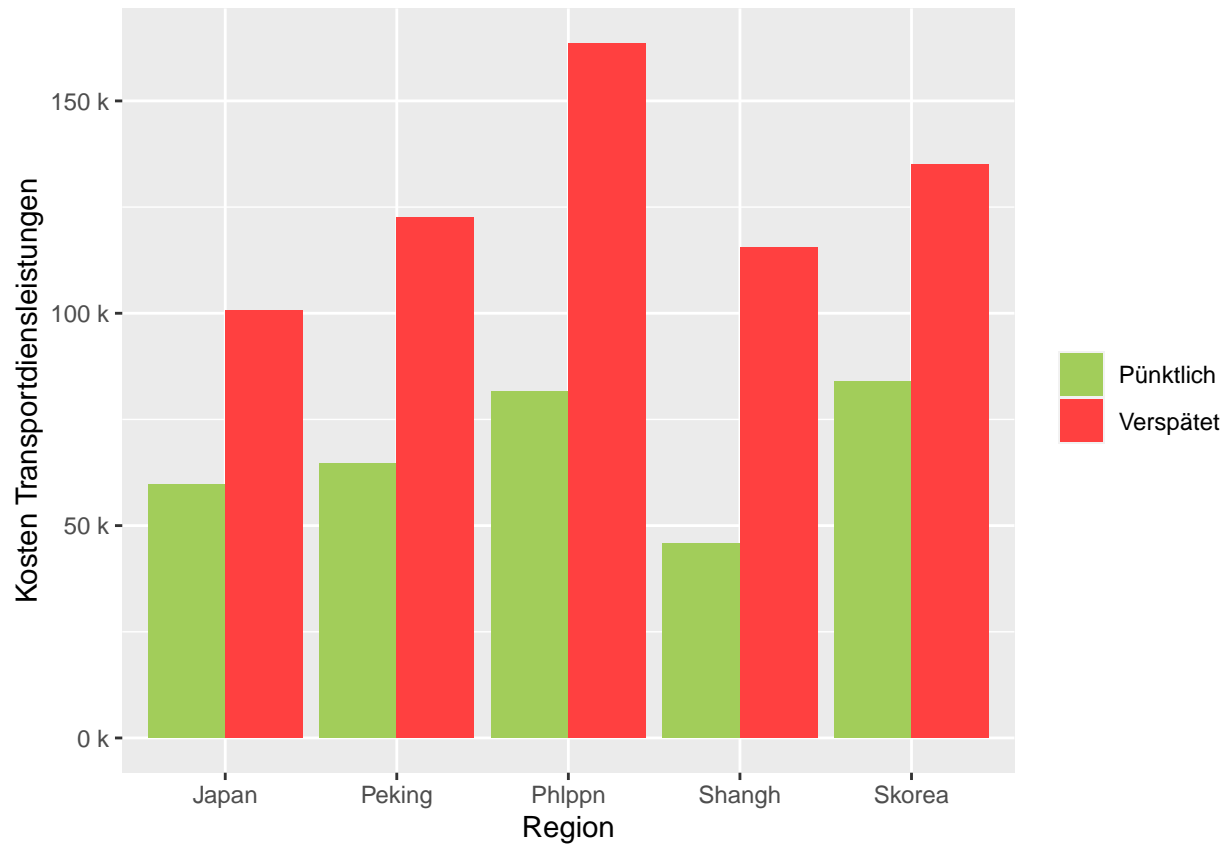
## Aufgabe 12

```

shipping_data105 <- services18_22 %>% filter(Product=="Gruppe105", service=="Shipping")
# Konditionierte Spalte um verspätete Lieferungen zu identifizieren
shipping_data105$late <- ifelse(shipping_data105$DaysExecuted >
  ↪ shipping_data105$DaysScheduled, "Verspätet", "Pünktlich")
# Kosten verspäteter und pünktlicher Lieferungen pro Region
shipping_byregion <- summarise(group_by(shipping_data105, region, late),
  ↪ total_cost=sum(cost))
# Visualisierung der Kosten verspäteter und pünktlicher Lieferungen pro Region
ggplot(data=shipping_byregion, aes(x=region, y=total_cost, fill=late)) +
  geom_bar(position="dodge", stat="identity")+
  xlab("Region") +
  ylab("Kosten Transportdienstleistungen")+

```

```
scale_y_continuous(labels = unit_format(unit = "k", scale = 1e-3)) +
theme(legend.title=element_blank()) +
scale_fill_manual(values=c("darkolivegreen3", "brown1"))
```



Die Transportausgaben (Shipping expenses) liegen in allen Regionen für verspätete Transporte höher als für pünktliche. Für die Region Philippinen sind nicht nur die Gesamtausgaben für Transporte am höchsten, sondern insbesondere auch für verspätete Transporte. Bei den pünktlichen Transporten liegt die Region Südkorea knapp vorne.

### Aufgabe 13

```
warehouse_data <- services18_22 %>% filter(Product=="Gruppe105", service=="Warehousing" )
→ %>% summarise(sum_scheduled= sum(QScheduled), sum_exec=sum(QExecuted),
→ sum_cost=sum(cost))
```

```
# Kosten pro abgefertigter Menge
real_unitcost <- warehouse_data$sum_cost/warehouse_data$sum_exec
real_unitcost
```

```
## [1] 1.573946
```

```
# Kosten pro vereinbarter Menge
scheduled_unitcost <- warehouse_data$sum_cost / warehouse_data$sum_scheduled
scheduled_unitcost
```

```
## [1] 1.302432
```

```
# Kostenvergleich der tatsächlichen abgefertigten Menge mit der vereinbarten Menge
real_unitcost / scheduled_unitcost - 1
```

```
## [1] 0.2084668
```

Die Kosten für die tatsächliche Lagerleistung betragen 1,57 Geldeinheiten pro Stück, die für die geplante Lagerleistung nur 1,30 Geldeinheiten pro Stück. Damit haben wir ca. 21 % mehr gezahlt.

## Bewertung der Logistikdienstleister

### Aufgabe 14

Für eine ganzheitliche Bewertung der Qualität eines Transportdienstleisters verwenden wir das Produkt von On-Time Delivery (OTD) und In-Full Rate (IFR).

$$KPI = IFR * OTD = \frac{\# \text{ of orders delivered complete}}{\# \text{ of customer orders}} * \frac{\# \text{ of orders delivered on or before due date}}{\# \text{ of customer orders}}$$

Beide zusammengenommen beschreiben, wie gut wir in der Lage sind, Kundenaufträge zu erfüllen. Wir haben uns bewusst gegen OTIF (On-Time, In-Full) als KPI entschieden, die nur Lieferungen im Zähler hätte, die sowohl vollständig als auch pünktlich sind. Die Annahme dahinter ist, dass wir auch unvollständige Lieferungen verkaufen können (wo nur einzelne Flaschen fehlen), nur eben weniger Flaschen.

```
# Neue Spalte für IFR und On-time delivery
services18_22$IFR <- services18_22$QExecuted / services18_22$QScheduled
services18_22$On_Time <- services18_22$DaysExecuted <= services18_22$DaysScheduled #
  ↳ true/false

# Berechnung der Qualität KPI pro Dienstleister
shipping_vendor_quality <- services18_22 %>%
  # Filter Shipping-DL nach Gruppe 105
  filter(Product=="Gruppe105", service=="Shipping" ) %>%
  group_by(vendor) %>%

  # Berechnung der OTD (Der boolean Lieferung wird numeric 0 für verspätete Lieferungen
  ↳ und 1 für pünktliche Lieferungen zugeordnet, deswegen können wir den mean()
  ↳ benutzen)
  # Berechnung des durchschnittlichen IFR, durchschnittlichen Kosten pro tatsächlicher
  ↳ abgefertigten Menge und Anzahl der Lieferungen
  summarise(Vendor_Quality=mean(On_Time)*mean(IFR),
             OTD=mean(On_Time),
             IFR=mean(IFR),
             Avg_Cost_per_Qty=mean(cost/QExecuted),
             No_Shipments=n()) %>%

  # Sortieren nach Vendor Quality KPI
  arrange(Vendor_Quality)

shipping_vendor_quality %>% kable(row.names = FALSE, caption = 'Qualitätskennzahlen für
  ↳ Shipping-Dienstleister', digits = 3)
```

Table 8: Qualitätskennzahlen für Shipping-Dienstleister

vendor	Vendor_Quality	OTD	IFR	Avg_Cost_per_Qty	No_Shipments
JNT Shipping	0.291	0.291	1	1.298	660
AHL Express Shipping	0.297	0.297	1	1.308	653
CPS Shipping	0.322	0.322	1	1.306	668
IntEx Shipping	0.327	0.327	1	1.286	655
Flying Mercury Shipping	0.344	0.344	1	1.302	694
HCX Shipping	0.350	0.350	1	1.295	677
DWL Shipping	0.360	0.360	1	1.308	684
EPD Shipping	0.362	0.362	1	1.312	686
Gifter Shipping	0.370	0.370	1	1.293	643
Bange+Hammer Shipping	0.458	0.458	1	1.313	690

Der schlechteste Shipping-Dienstleister nach unserer Qualitätskennzahl ist hier JNT Shipping, die eine deutlich schlechtere Liefertreue bzw. OTD haben als die anderen Dienstleister. Da die IFR für Shipping-Dienstleistungen in diesem Datensatz immer 1 beträgt, ist dies der einzige Einflussfaktor. Der beste Shipping-Dienstleister, Bange+Hammer Shipping, führt für uns 690 Lieferungen aus. Bei JNT Shipping sind es nur unwesentlich weniger (660). Neben der reinen Qualität sollten die Dienstleister auch nach ihren Kosten bewertet werden (siehe Spalte "Avg\_Cost\_per\_Qty", also Kosten pro tatsächlich ausgeführter Einheit). Diese unterscheiden sich allerdings eher gering und wären angesichts der extrem schlechten OTD-Raten wohl erstmal zweitrangig.

## Aufgabe 15

Wir verwenden hier analog und mit der gleichen Begründung wie bei den Transportdienstleistern eine kombinierte Rate aus OTD und IFR. Die On-Time Delivery bezieht sich hier nicht auf eine Auslieferung, sondern ob der Warehousing-Auftrag in der ursprünglich geplanten Zeit durchgeführt wurde.

```
# Neue Spalte für IFR und On-time delivery --> siehe Aufgabe 16

# Berechnung der Qualität KPI pro Dienstleister
# Analog zu Aufgabe 16
warehouse_vendor_quality <- services18_22 %>%
  # Filter Warehouse-DL nach Gruppe 105
  filter(Product=="Gruppe105", service=="Warehousing" ) %>%
  group_by(vendor) %>%

  # Berechnung der OTD und IFR analog zu A16
  summarise(Vendor_Quality=mean(On_Time)*mean(IFR),
            OTD=mean(On_Time),
            IFR=mean(IFR),
            Avg_Cost_per_Qty=mean(cost/QExecuted),
            No_WH_Services=n()) %>%

  # Sortieren nach Vendor Quality KPI
  arrange(desc(Vendor_Quality))

warehouse_vendor_quality %>% kable(row.names = FALSE, caption = 'Qualitätskennzahlen für
↳ Warehouse-Dienstleister', digits = 3)
```

Table 9: Qualitätskennzahlen für Warehouse-Dienstleister

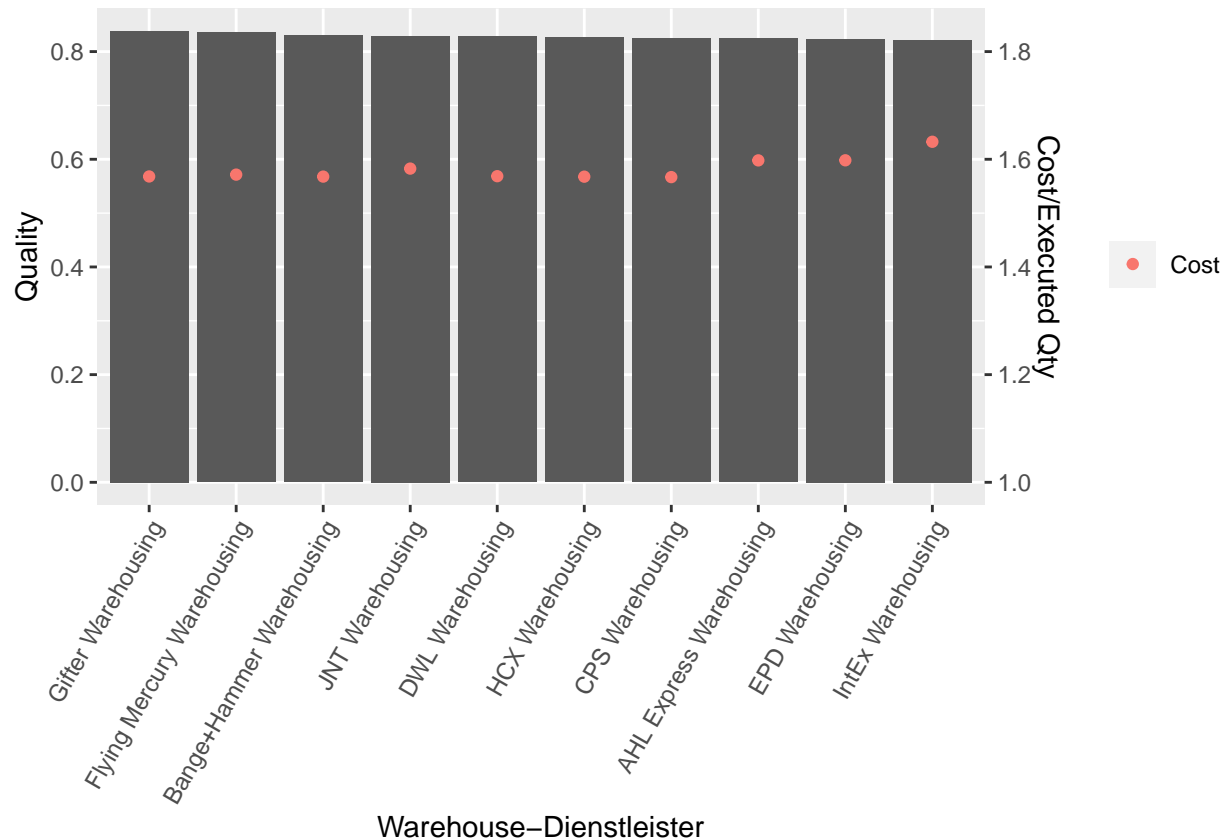
vendor	Vendor_Quality	OTD	IFR	Avg_Cost_per_Qty	No_WH_Services
Gifter Warehousing	0.838	1	0.838	1.565	683
Flying Mercury Warehousing	0.836	1	0.836	1.554	708
JNT Warehousing	0.831	1	0.831	1.579	681
HCX Warehousing	0.829	1	0.829	1.561	572
CPS Warehousing	0.826	1	0.826	1.600	720
AHL Express Warehousing	0.825	1	0.825	1.581	652
DWL Warehousing	0.823	1	0.823	1.585	685
Bange+Hammer Warehousing	0.822	1	0.822	1.581	712
EPD Warehousing	0.820	1	0.820	1.590	620
IntEx Warehousing	0.819	1	0.819	1.601	677

Der schlechteste Warehouse-Dienstleister nach unserer Qualitätskennzahl ist hier IntEx Warehousing, welcher die schlechteste IFR aufweist. Da die IFR für Shipping-Dienstleistungen in diesem Datensatz immer 1 beträgt, ist dies der einzige Einflussfaktor. Insgesamt liegen die Dienstleister jedoch sehr nah beieinander (weniger als 2 Prozentpunkte Unterschied zwischen dem besten und schlechtesten). Auch die Anzahl der erbrachten Services (“No\_WH\_Services”) ist sehr ähnlich. Neben der reinen Qualität sollten die Dienstleister auch nach ihren Kosten bewertet werden (siehe Spalte “Avg\_Cost\_per\_Qty”, also Kosten pro tatsächlich ausgeführter Einheit). Flying Mercury Warehousing ist hier interessant, da es die zweitbeste Qualität bei niedrigsten Preise aufweist.

## Aufgabe 16

```
# Visualisierung der KPI für Warehouse-Dienstleister
WH_vendor_qual_all_2022 <- services18_22 %>%
  filter(service=="Warehousing", year(Date)==2022) %>%
  group_by(vendor) %>%
  summarise(Vendor_Quality=mean(On_Time)*mean(IFR),
            OTD=mean(On_Time),
            IFR=mean(IFR),
            Avg_Cost_per_Qty=mean(cost/QExecuted),
            No_WH_Services=n())

ggplot(data=WH_vendor_qual_all_2022, aes(x=reorder(vendor, Vendor_Quality,
  ↪ decreasing=TRUE), y=Vendor_Quality)) +
  geom_col() +
  geom_point(aes(x=reorder(vendor, Vendor_Quality, decreasing=TRUE),
  ↪ y=Avg_Cost_per_Qty-1, color="Cost")) +
  xlab("Warehouse-Dienstleister") +
  ylab("Quality") +
  theme(legend.title=element_blank()) +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(name="Quality", sec.axis = sec_axis(~.+1, name="Cost/Executed Qty"))
```



Beim Gesamtvergleich über alle Produkte im Jahr 2022 sehen wir bei den Warehousing-Dienstleistern eher geringe Unterschiede bei der Qualität, bei allerdings unterschiedlichen Kosten. Gifter Warehousing weist die höchste Qualität (hier aufgrund der Daten ausschließlich bestimmt durch die IFR) bei eher geringen Kosten auf. IntEx Warehousing weist die niedrigste IFR bei höchsten Kosten auf.

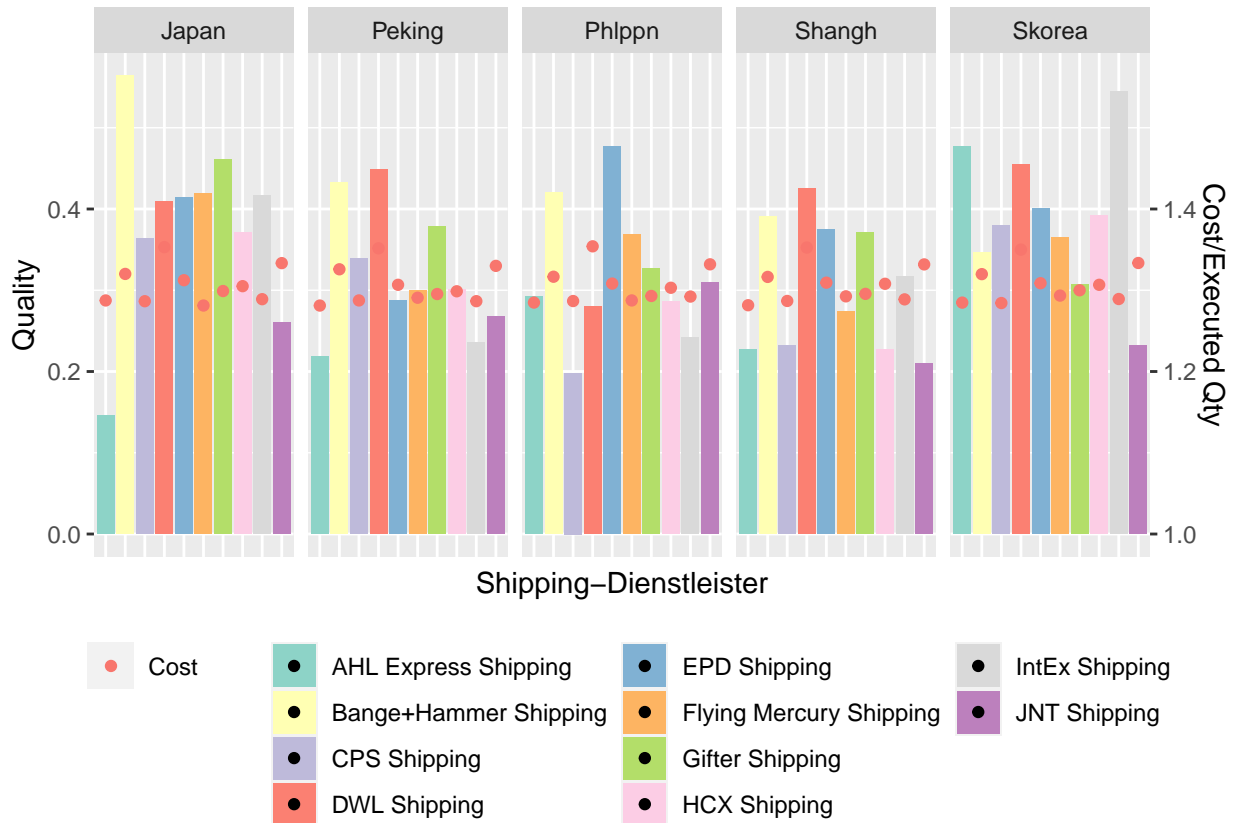
Insgesamt ist die Qualität der Warehousing-Dienstleister mit einer Qualität (hier ausschließlich bestimmt durch die IFR) von >80% als nicht schlecht, aber verbesserungswürdig zu bewerten

## Aufgabe 17

```
# Visualisierung der KPI für Shipping-Dienstleister pro Region
SH_vendor_qual_2022 <- services18_22 %>%
  filter(service=="Shipping", year(Date)==2022) %>%
  group_by(vendor, region) %>%
  summarise(Vendor_Quality=mean(On_Time)*mean(IFR),
            OTD=mean(On_Time),
            IFR=mean(IFR),
            Avg_Cost_per_Qty=mean(cost/QExecuted),
            No_WH_Services=n())

ggplot(data=SH_vendor_qual_2022, aes(x=vendor, y=Vendor_Quality, fill=vendor)) +
  facet_wrap(~region, nrow=1) +
  geom_col() +
  geom_point(aes(x=reorder(vendor, Vendor_Quality, decreasing=TRUE),
    ↪ y=Avg_Cost_per_Qty-1, color="Cost")) +
  xlab("Shipping-Dienstleister") +
  ylab("Quality") +
```

```
theme(legend.title = element_blank(), legend.position = "bottom") +
guides(fill=guide_legend(nrow=4)) +
theme(axis.text.x=element_blank(), axis.ticks.x = element_blank()) +
scale_y_continuous(name="Quality", sec.axis = sec_axis(~.+1, name="Cost/Executed
Qty")) +
scale_color_brewer(palette="Set3", aesthetics = "fill")
```



Beim Gesamtvergleich über alle Produkte im Jahr 2022 und differenziert nach Regionen sehen wir bei den Shipping-Dienstleistern große Unterschiede bei der Qualität bei zumeist vergleichbaren Kosten. Dabei gibt es keinen eindeutig besten Anbieter. In Japan hat beispielsweise Bange+Hammer Shipping mit Abstand die höchste Qualität (On-Time Delivery), ist dafür aber in Südkorea unter den schlechteren Dienstleistern. AHL Express Shipping hat dagegen in den meisten Regionen eine eher niedrige Performance (in Japan und Peking sogar der schlechteste Anbieter), liegt in Südkorea aber auf Platz 2. Eine Dienstleisterauswahl muss also je nach Region erfolgen.

Insgesamt ist die Qualität der Shipping-Dienstleister als schlecht zu bewerten. Mit OTD-Raten zwischen 15% bis etwa 57% können wir unseren Kunden nur sehr selten pünktlich ihre Produkte liefern.

## Projektbeschreibung

Im Folgenden soll die in Aufgabe 2.1 durchgeführte Analyse in ein Projekt überführt werden, das systematisch Mehrwert für Gruppe105 generieren soll. Dazu beschreiben wir die geplante Projektdurchführung, zu verwendende Datenquellen, die Nutzer:innen der Analysen, sowie Empfehlungen aus den bisherigen Erkenntnissen.



## Projektdurchführung

Bei der Durchführung des Projektes orientieren wir uns am Standard-Vorgehen nach dem Cross-Industry Standard Process for Data Mining (CRISP-DM), der aus sechs Schritten besteht.

### 1. Problemidentifizierung

In der ersten Phase betrachten wir das wirtschaftliche Problem, noch ohne Blick auf die Daten. Dazu machen wir eine Lageanalyse, sprechen mit relevanten internen Stakeholdern wie der Logistik- oder Vertriebsabteilung, und legen abgestimmte Analyseziele fest. Ein mögliches Ziel der Gruppe 105 ist die Auswahl der besten Liefer- und Lager-Dienstleister anhand von Qualität KPIs. Dies kann in Verbindung mit Nachfragevorhersage, dazu führen Logistikdienstleistungen nicht mehr direkt vom Spotmarket zu beziehen, sondern sie durch preiswertere und langfristig planbare Verträge zu ersetzen. Ein weiteres Ziel kann die Marktanalyse in Bezug auf Konkurrenz in den fünf Regionen sein, um den Marktanteil zu vergrößern und die Kundenkenntnis zu verbessern und damit Kundentreue zu erhöhen. Alle diese Ziele können zu einer Erhöhung der Profitabilität der Gruppe 105 beitragen.

### 2. Datenbeschaffung

Als nächstes mappen und beschreiben wir existierende Datenquellen. Basierend auf dieser Übersicht können wir relevante Daten systematisch auswählen oder wo nötig weitere Daten beschaffen. Grundlage bilden die Datensätze der Jahre 2018 bis 2022, die bereits von den Supermärkten sowie von den Logistikdienstleistern bereitgestellt wurden. Sie beinhalten auch Informationen über die Konkurrenz in den fünf Regionen. Weitere Datenquellen sind unter Datenquellen genannt.

### 3. Datenaufbereitung

Hier extrahieren wir zunächst die zuvor festgelegten Daten aus internen Quellsystemen wie SAP oder externen Datenbanken. Im nächsten Schritt transformieren wir die Daten, indem wir ungültige Daten bereinigen, Filtern setzen (etwa nach Zeiträumen), fehlende Daten entweder ergänzen oder löschen, und Daten in das richtige Format bringen (z. B. Datumskonvertierung). Schließlich laden wir die Daten in das ausgewählte Analyseprogramm (hier R).

### 4. Ausführen der Analyse

Basierend auf den in Schritt 1 festgelegten Zielen verfolgen wir hier verschiedene Analyserichtungen, zum Beispiel:

- Analyse der Profitabilität und möglichen Treibern, sowie Einfluss der Distribution
- Vergleich der Nachfrage zwischen und in Märkten
- Identifizierung der besten und schlechtesten Logistikdienstleister
- Analyse der Lost Sales nach Zeit, Region, und Zusammenhang zur Performance der Logistikdienstleister

In der Analyse verwenden wir eine Reihe von Methoden, um unser Analyseziel zu erreichen. Zunächst deskriptive Methoden wie Aggregationen nach verschiedenen Dimensionen wie Märkten oder Dienstleistern, dann die Berechnung verschiedener KPIs wie In-Full Rate oder On-Time Delivery Rate. Daneben verwenden wir visuelle Methoden, um die Daten intuitiv darzustellen und leichte Analysen zu ermöglichen, z. B. Bar Charts zum Vergleich verschiedener LDL. Darüber hinaus könnten sich je nach Projektverlauf und -zielen weiterführende Methoden aus den Bereichen Predictive/Prescriptive Analytics anbieten, wie etwa die Vorhersage der Nachfrage oder Zuverlässigkeit einzelner LDL.

### 5. Bewertung der Ergebnisse

Nach abgeschlossener Analyse bereiten wir die Ergebnisse in geeigneter Form auf, etwa in Form eines Reports mit Executive Summary. Diese präsentieren wir vor den relevanten internen Stakeholdern und diskutieren Ergebnisse und nächste Schritte. Zudem nehmen wir gemeinsam einen Abgleich mit den ursprünglichen Projektzielen vor und bewerten, ob diese erreicht wurden.

## 6. Bereitstellung und Nutzengenerierung

Im letzten Schritt geht es darum, die aus der Analyse und der Bewertung der Ergebnisse abgeleiteten Maßnahmen umzusetzen.

Weil sich das Geschäftsumfeld und damit die Datengrundlage ständig ändert, ist es erstrebenswert, von einer Einmal-Analyse wegzukommen hin zu kontinuierlichem Monitoring und Verbesserung. Dazu sollte die Analyse replizierbar gemacht werden, etwa durch Definition von festen KPIs, Erstellung von Dashboards und einer zumindest in Teilen automatisierten Dateneinspeisung.

### Datenquellen

Als Datenquelle bieten sich zunächst firmeninterne Quellsysteme wie das ERP-System an. Im SAP-Bereich sind für die gegebene Aufgabenstellung besonders die Module Sales & Distribution (SD; für Vertriebsdaten) und Finance (FI; für Kosten und Zahlungsverkehr mit LDL) relevant. Uns interessieren Tabellen wie etwa:

- VBAK und VBAP: Für Bestellungen und Bestellpositionen
- BSEG: Für Rechnungsinformationen
- LFA1: Für Informationen zu Lieferanten
- KNA1: Für Kundeninformationen

Sollte kein ERP-System verwendet werden, liegen die Daten ggf. als Mischung aus Excel-Tabellen und E-Mails mit Dienstleistern vor. Diese wären in einem ersten Schritt systematisch zu erfassen und dann in eine nutzbare Form in Tabellenformat zu bringen.

Neben internen Datenquellen sind auch externe Quellen potenziell von Interesse, zum Beispiel:

- Tagesaktuelle Spotmarket-Preise für Logistikdienstleistungen
- Informationen zum Gesamtmarkt (z. B. insgesamt verkaufte Limonaden in Supermärkten, über alle Anbieter hinweg) in den gegebenen Regionen

### Nutzer:innen

Die wichtigsten Analyseergebnisse sind relevant für Führungskräfte im Bereich Supply Chain (z. B. Chief Supply Chain Officer). Daneben sind zeitnahe, relevante Ergebnisse wichtig für Angestellte, die direkt mit den LDL zusammenarbeiten bzw. diese beauftragen.

### Empfehlungen basierend auf Teil 2.1

Aus unseren Analysen in Teil 2.1 sehen wir für Logistikdienstleister folgende Schlüsse:

- Shipping-Dienstleister: In Anbetracht der sehr schlechten OTD-Rate von unter 35% sind hier dringend Verbesserungen nötig. Als erste Empfehlung sollten für Gruppe 105 Verträge mit Bange+Hammer Shipping priorisiert werden, da diese mit Abstand die beste OTD Performance aufzeigen, bei durchschnittlichen Kosten. Dies sollte allerdings nach Region nochmal differenziert betrachtet werden.
- Warehousing-Dienstleister: Hier besteht ein etwas geringerer Handlungszwang. Konkret für Gruppe 105 sollten nach Möglichkeit Verträge mit Flying Mercury Warehousing ausgeweitet werden, da diese bei hoher IFR (Rang 2) die geringsten Kosten aufweisen.