Exploratory Data Analysis

A Muesli Company

Silke & Trista

What is our goal?

- **To help a Muesli company to understand their delivery process and develop KPIs to improve their service**
- **To discuss our approach with our fellow classmates**

Exploratory Data Analysis - Data Cleaning

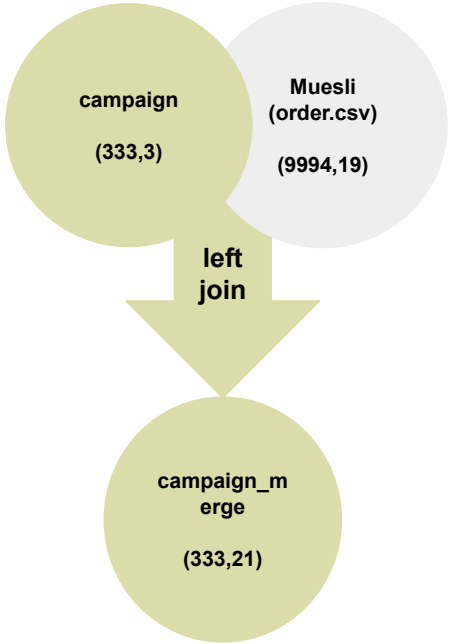
- Are there any null values or outliers? How will you wrangle/handle them?
  - Have a look at the data sets:
    - `.info()`
    - `.describe()`
    - `.shape`
    - `.isnull()`
  - Both Null values and outliers have no impacts on the KPIs we want to look at, leave it as they are

Dataframes	Null values?	Outliers?
muesli	11 null values column“postal_code”	Sales, profits, discount, profits
process data	no	no
ready to ship	no	no
campaign	no	no

Exploratory Data Analysis - Data Cleaning

- Are there any variables that warrant transformations?
  - Change data types: `pd.to_datetime()`
  - Extract ‘year’ and ‘month’ from all ‘date’ for further analysis
    - `muesli['order month'] = muesli['order date'].dt.month`
    - `muesli['order_year'] = muesli['order_date'].dt.year`
  - Clean up column names
    - `.columns.str.replace(" ", "_")`
    - `.columns.str.replace("/", "_")`
    - `.columns.str.replace("-", "_")`
    - `.columns = [x.lower() for x in order_process.columns]`
  - Checked duplicates & decide if we drop them
    - dropped duplicates in ready_to_ship (intern data): `.drop_duplicates(subset= 'order_id', inplace= True)`

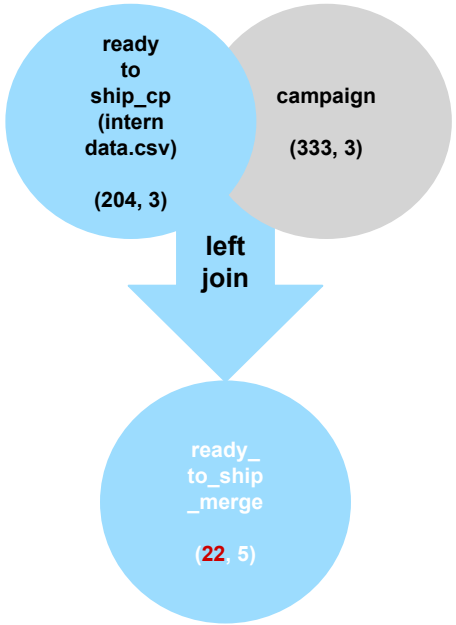
Exploratory Data Analysis - Merging Dataframes




```
campaign_merge = pd.merge(campaign, muesli, on= "order_id")
```

```
Data columns (total 21 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   order_id           333 non-null    object
 1   arrival_scan_date  333 non-null    datetime64[ns]
 2   customer_name_x    333 non-null    object
 3   index              333 non-null    int64
 4   order_date         333 non-null    datetime64[ns]
 5   ship_mode          333 non-null    object
 6   customer_id        333 non-null    object
 7   customer_name_y    333 non-null    object
 8   origin_channel     333 non-null    object
 9   country_region     333 non-null    object
 10  city               333 non-null    object
 11  state              333 non-null    object
 12  postal_code        333 non-null    float64
 13  region             333 non-null    object
 14  category           333 non-null    object
 15  sub_category       333 non-null    object
 16  product_id         333 non-null    object
 17  sales              333 non-null    float64
 18  quantity           333 non-null    int64
 19  discount           333 non-null    float64
 20  profit             333 non-null    float64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
```

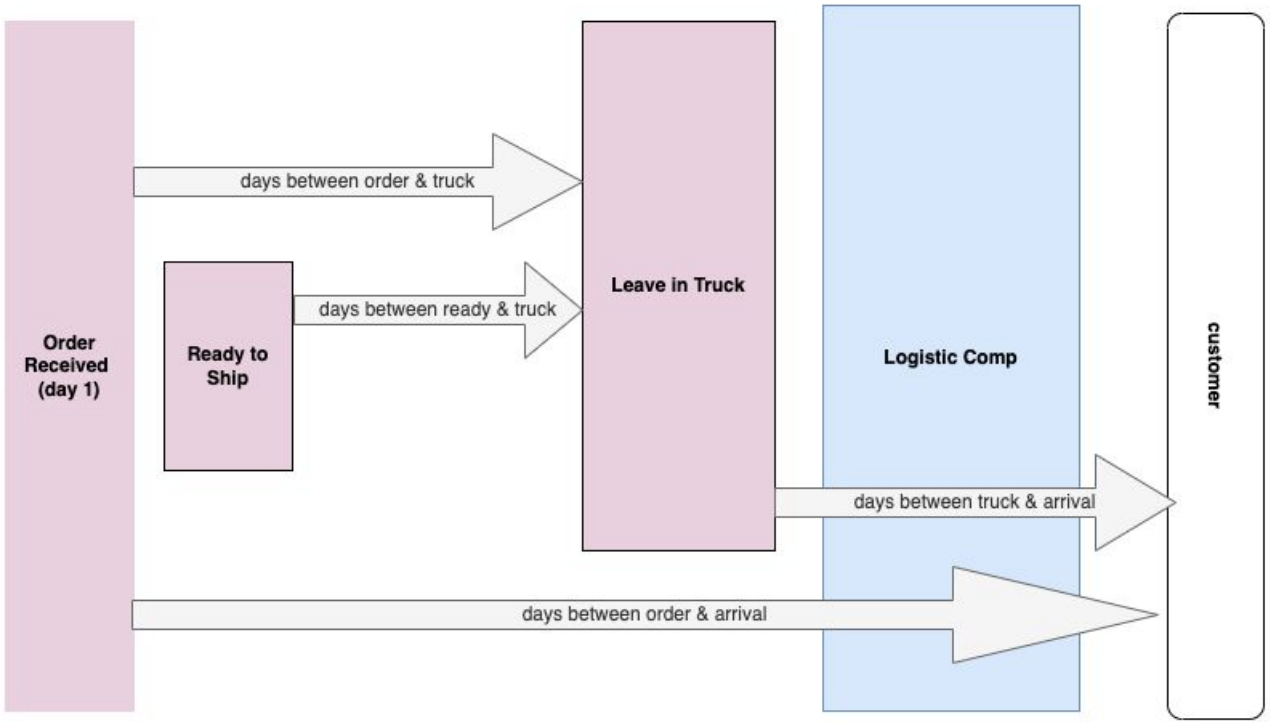
Exploratory Data Analysis - Merging Dataframes




```
# copy ready_to_ship and drop the duplicates
ready_to_ship_cp = ready_to_ship.copy()
ready_to_ship_cp.drop_duplicates(inplace= True)
# merge
ready_to_ship_merge = pd.merge(ready_to_ship_cp, campaign, on= "order_id")
```

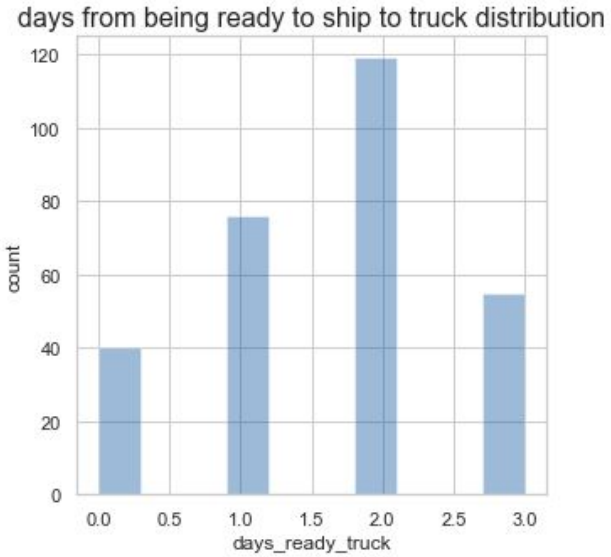
```
Int64Index: 22 entries, 0 to 21
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   order_id            22 non-null     object
 1   ready_to_ship_date  22 non-null     datetime64[ns]
 2   pickup_date         22 non-null     datetime64[ns]
 3   arrival_scan_date   22 non-null     datetime64[ns]
 4   customer_name       22 non-null     object
dtypes: datetime64[ns](3), object(2)
```

How does the delivery process work?



How many days between ready to ship and truck?

```
#Compute new KPI: days between ready to ship and package being loaded on the truck
ready_to_ship['days_ready_truck'] = (ready_to_ship['pickup_date'] - ready_to_ship['ready_to_ship_date']).dt.days
ready_to_ship.head()

# Get to know the new KPI "ready to ship to truck" # ... by plotting it
plt.figure(figsize=(5,5))
sns.distplot(ready_to_ship['days_ready_truck'], kde=False, hist=True, bins=10)
plt.title('days from being ready to ship to truck distribution', size=16)
plt.ylabel('count')
# ... by describing it
ready_to_ship.describe()['days_ready_truck']
# ... by looking at the distribution in a table
ready_to_ship.groupby('days_ready_truck').count()['order_id']
```

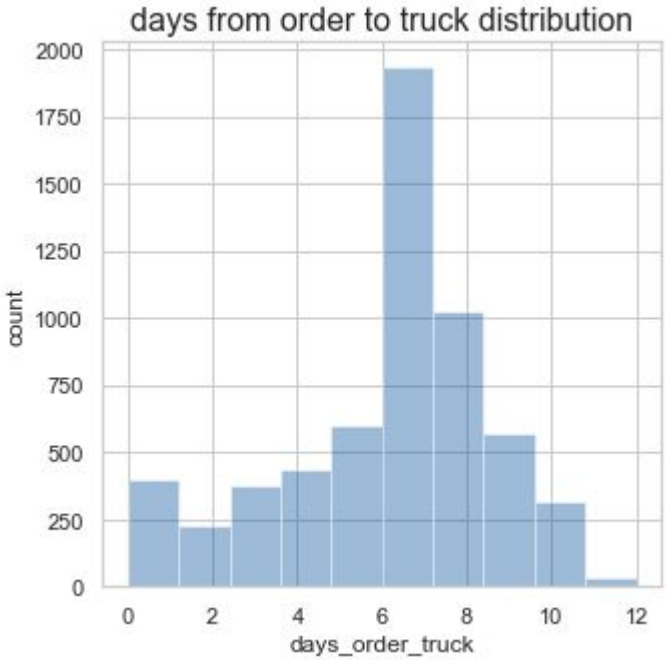


Sample Size: 290

→ on average it takes 1.65 days

How many days does it take from order to truck?

```
# Get to know the new KPI "order to truck"
# ... by plotting it
plt.figure(figsize=(5,5))
sns.distplot(order_process['days_order_truck'], kde=False,
hist=True, bins=10)
plt.title('days from order to truck distribution', size=16)
plt.ylabel('count')

# ... by describing it
order_process.describe()['days_order_truck']

# ... by looking at the distribution in a table
order_process.groupby('days_order_truck').count()['order_id']
```



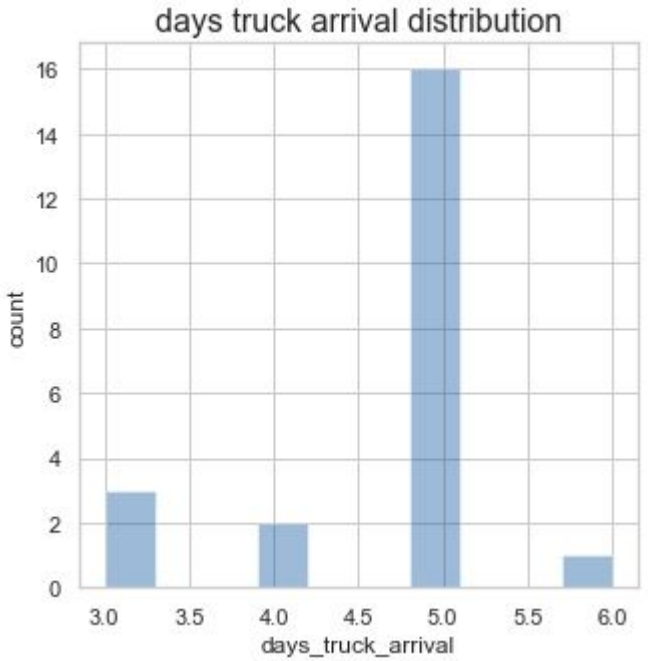
→ on average it takes 6.12 days

Sample Size: 5899

How many days between truck and arrival?

```
# To compute the new KPI 'days between loading on truck and arrival' we have to merge
the dataframe Campaign and Ready_to_ship
ready_to_ship_merge = pd.merge(ready_to_ship_unique, campaign, on='order_id')
ready_to_ship_merge['days_truck_arrival'] = (ready_to_ship_merge['arrival_scan_date'] -
ready_to_ship_merge['pickup_date']).dt.days
ready_to_ship_merge.head(22)
ready_to_ship_merge.days_truck_arrival.mean()

# Plot a Histogram
plt.figure(figsize=(5,5))
sns.distplot(ready_to_ship_merge['days_truck_arrival'], kde=False, hist=True, bins=10)
plt.title('days from truck to arrival distribution', size=16)
plt.ylabel('count')

# Look at the distribution in a table.
ready_to_ship_merge['days_truck_arrival'].value_counts()
ready_to_ship_merge.groupby('days_truck_arrival').count()['order_id']

ready_to_ship_merge.describe()
```

→ on average it takes 4.68 days

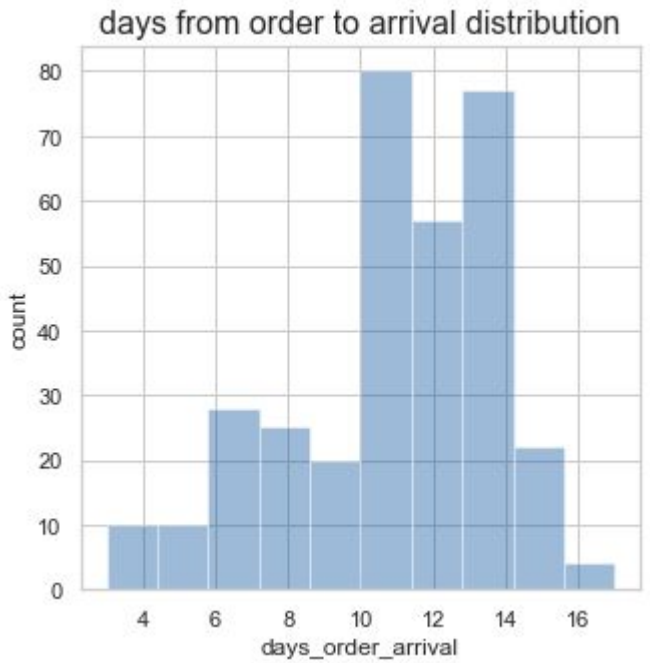


Sample Size: 22

How many days does it take from order to arrival?

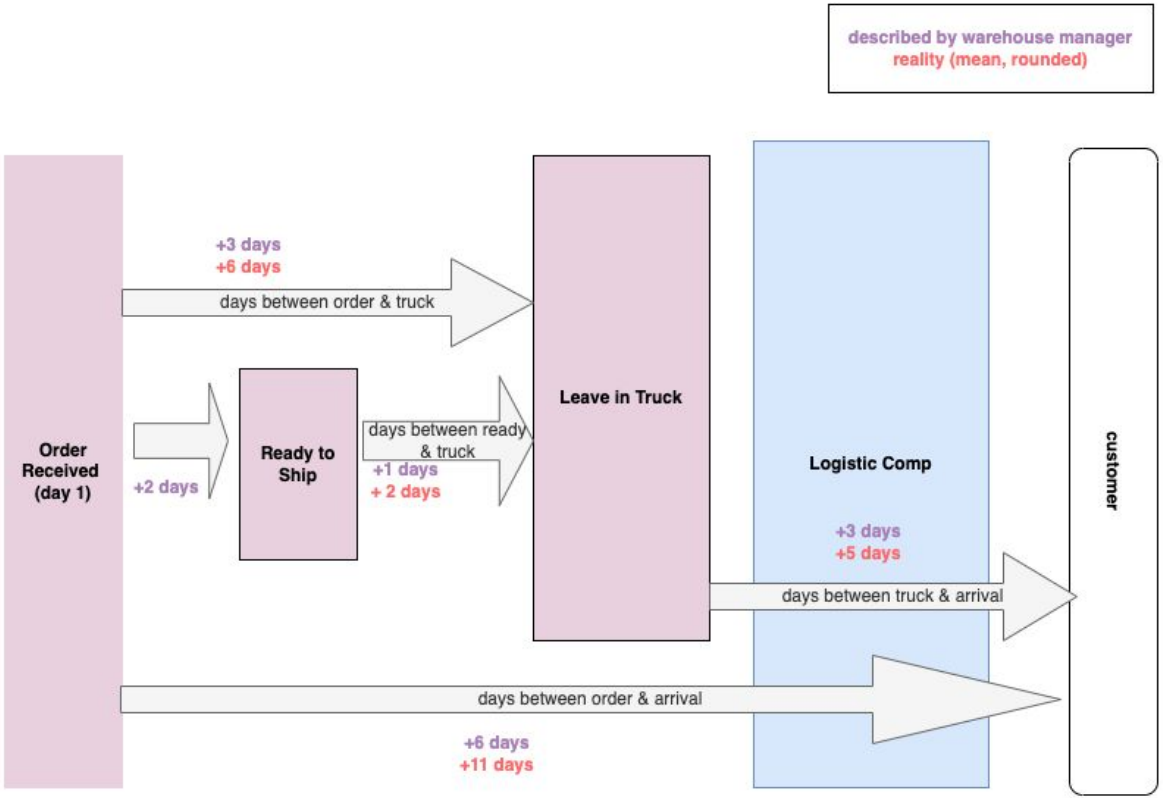
```
# To compute the KPI 'days between order and arrival at customer' we have to merge
dataframe Campaign and Muesli
campaign_merge = pd.merge(campaign, muesli, on='order_id')

# Drop all duplicates
campaign_merge = campaign_merge.drop_duplicates(subset=['order_id'])

# Add new KPI 'days_order_arrival' as a new column
campaign_merge['days_order_arrival'] = (campaign_merge['arrival_scan_date'] -
campaign_merge['order_date']).dt.days

# Get to know the new KPI
# ... by plotting it
plt.figure(figsize=(5,5))
sns.distplot(campaign_merge['days_order_arrival'], kde=False, hist=True, bins=10)
plt.title('days from order to arrival distribution', size=16)
plt.ylabel('count')
# ... by describing it
campaign_merge.describe()
# ... by looking at the distribution in a table
campaign_merge.groupby('days_order_arrival').count()['order_id']
```

→ on average it takes 10.83 days



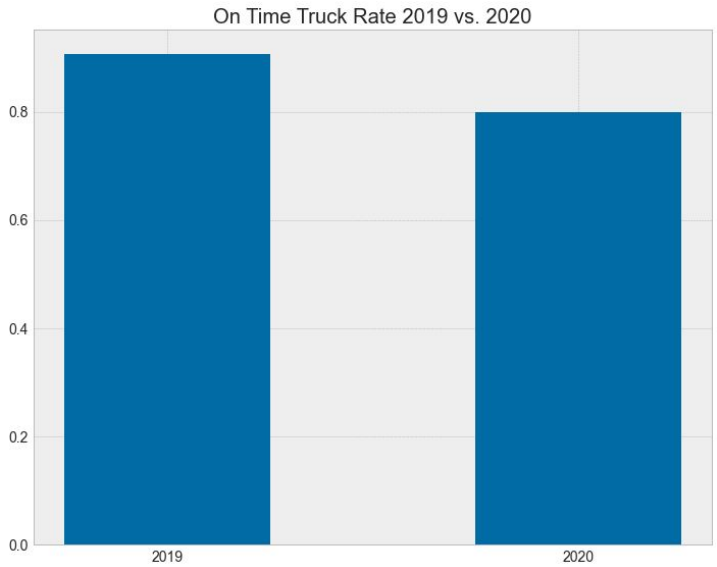
Sample Size: 333

Reality vs. Warehouse Manager' plans



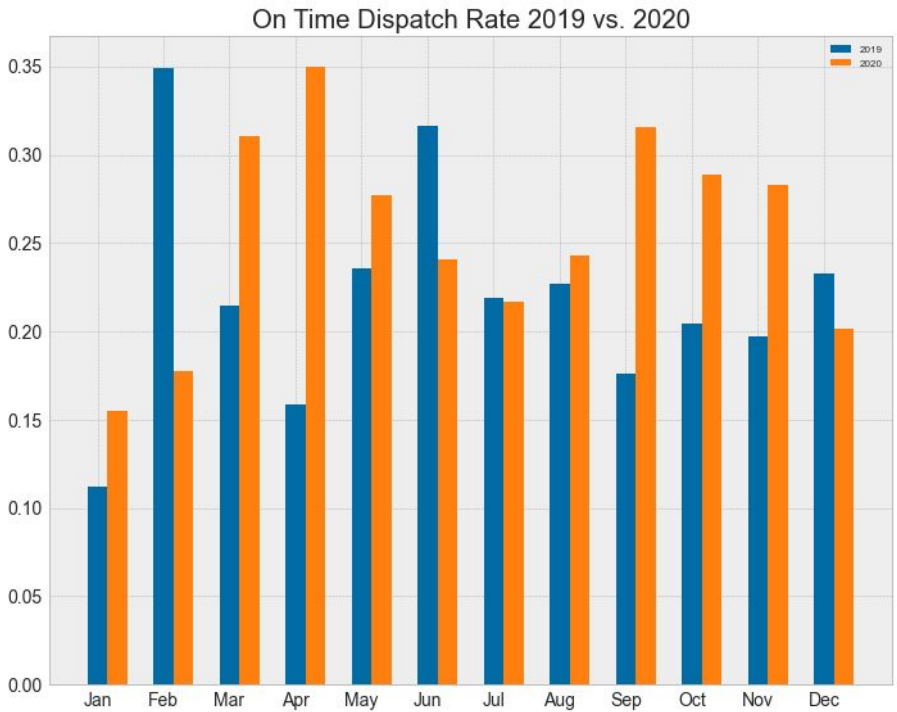
More than 80% of all orders loaded on truck within 2 days.

```
#Create 2 new columns that say if
ready_to_ship_cp['days_ready_truck_T'] =
np.where(ready_to_ship_cp['days_ready_truck'] <= 2, 1, 0)

ready_to_ship_cp['days_ready_truck_F'] =
np.where(ready_to_ship_cp['days_ready_truck'] > 2, 1, 0)

# compute the KPI "on time truck rate"

ready_to_ship_year = ready_to_ship_cp.groupby(['order_year']).sum()
ready_to_ship_year['on_time_truck_rate'] =
ready_to_ship_year['days_ready_truck_T'] /
(ready_to_ship_year['days_ready_truck_T'] +
ready_to_ship_year['days_ready_truck_F'])
```



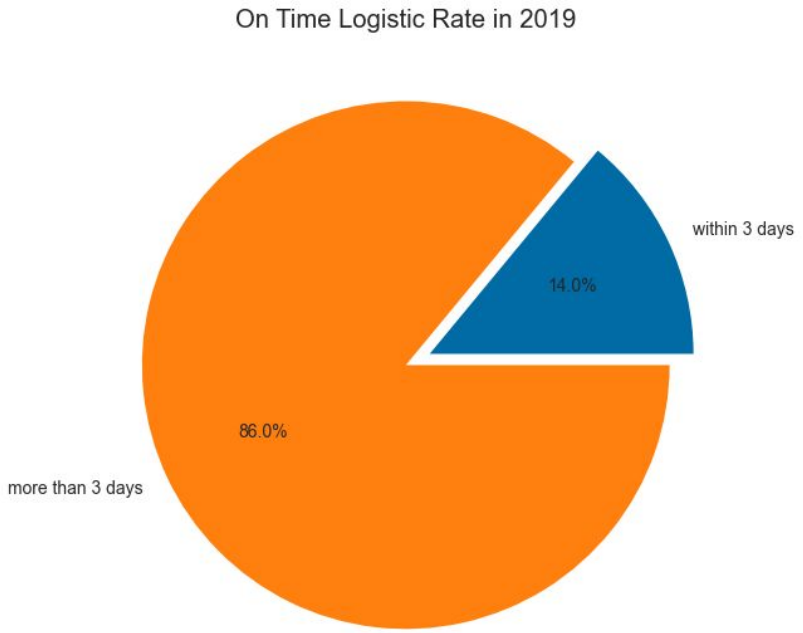
In 2020 only 26% of orders are dispatched on time.

```
#On-time dispatch rate: How many percent of the orders are loaded on the truck
4 days later?
#Add month & year as a column
order_process['order_month'] = order_process['order_date'].dt.month
order_process['order_year'] = order_process['order_date'].dt.year

#Create 2 columns: One that prints 1 if package is dispatched in time and one
if not.
order_process['days_order_truck_T'] =
np.where(order_process['days_order_truck'] <= 4, 1, 0)
order_process['days_order_truck_F'] =
np.where(order_process['days_order_truck'] > 4, 1, 0)

#Create new dataframe that shows the sum of the new columns grouped by year
and month
order_process_month = order_process.groupby(['order_year',
'order_month']).sum()
order_process_month['on_time_dispatch'] =
order_process_month['days_order_truck_T'] /
(order_process_month['days_order_truck_T'] +
order_process_month['days_order_truck_F'])
```



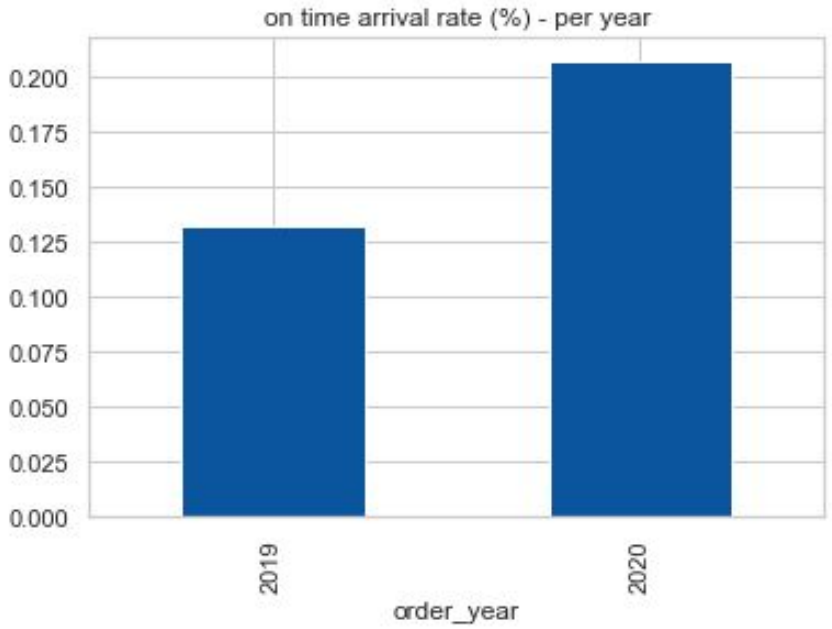
Only 14% of all packages delivered within the promised time

```
# Create 2 new columns that say if package arrives within
3 days at the customer from warehouse or not:
ready_to_ship_merge[ 'days_truck_arrival_T' ] =
np.where(ready_to_ship_merge[ 'days_truck_arrival' ] <= 3, 1, 0)

ready_to_ship_merge[ 'days_truck_arrival_F' ] =
np.where(ready_to_ship_merge[ 'days_truck_arrival' ] > 3, 1, 0)

# compute the KPI "on time logistic rate"

ready_to_ship_merge_year =
ready_to_ship_merge.groupby( 'order_year').sum()
ready_to_ship_merge_year[ 'on_time_logistic_rate' ] =
ready_to_ship_merge_year[ 'days_truck_arrival_T' ] /
(ready_to_ship_merge_year[ 'days_truck_arrival_T' ] +
ready_to_ship_merge_year[ 'days_truck_arrival_F' ])
```



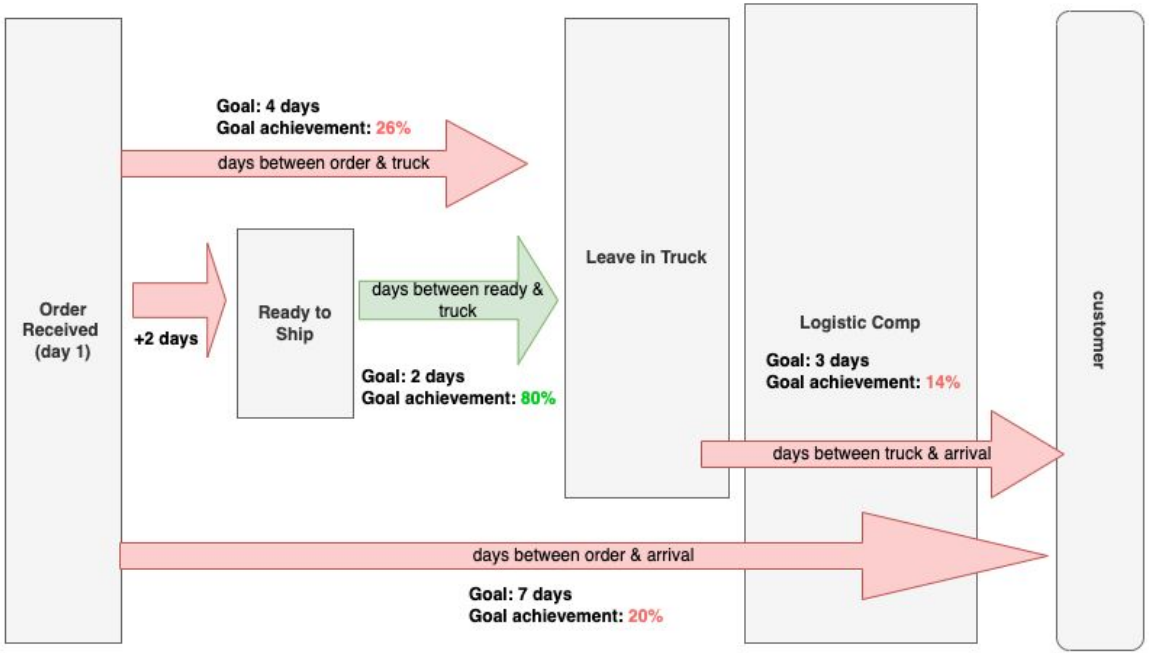
Only 20% of all orders reach the customer within 7 days.

```
#Create 2 new columns that say if package arrives within 7
days at the customer or not:
campaign_merge['days_order_arrival_T'] =
np.where(campaign_merge['days_order_arrival'] <= 7, 1, 0)
campaign_merge['days_order_arrival_F'] =
np.where(campaign_merge['days_order_arrival'] > 7, 1, 0)

#We decided there is not enough data to look at on a month to
month basis, so we looked at the yearly data instead
campaign_merge_year =
campaign_merge.groupby('order_year').sum()
campaign_merge_year['on_time_arrival'] =
campaign_merge_year['days_order_arrival_T'] /
(campaign_merge_year['days_order_arrival_T'] +
campaign_merge_year['days_order_arrival_F'])
display(campaign_merge_year)
```



Sample Size: 290

Dashboard


Order Received (day 1)
Goal: 4 days
Goal achievement: 26%
days between order & truck
+2 days
Ready to Ship
days between ready & truck
Goal: 2 days
Goal achievement: 80%
Leave in Truck
Logistic Comp
Goal: 3 days
Goal achievement: 14%
days between truck & arrival
customer
days between order & arrival
Goal: 7 days
Goal achievement: 20%

Conclusion

- Delivery process operates worse than warehouse manager plans

  → Start using the dashboard to monitor KPIs

- Internal logistic process needs to be improved (from order received to ready to ship)

  → more manpower on the weekend ;)

- Better performance from the logistic company is needed

Next Step

- Collect more data for each points to have a deeper and solid understanding
  - from order received to ready to ship
  - from truck to customers' door
- Do further analysis on what the bottlenecks are. Why is the process slower than anticipated? (Regression analysis f.e. on weekdays,...)

Reference

Feedback from the class

- to have a structure page
- to have a page listing out the metrics and the KPIs
- clearer graph for on time dispatch rate (the title and the graph doesnt match, too much info)
- to coordinate better on who’s sharing the screen and how to control (can use zoom function - sharing screen control!!)

Section 02: Statistical Analysis

Explain how we computed KPIs and merge

KPI:

how many days from order received to be loaded on truck?

+ order_process[‘days_order_truck’]

how many days between ready to ship and truck pickup

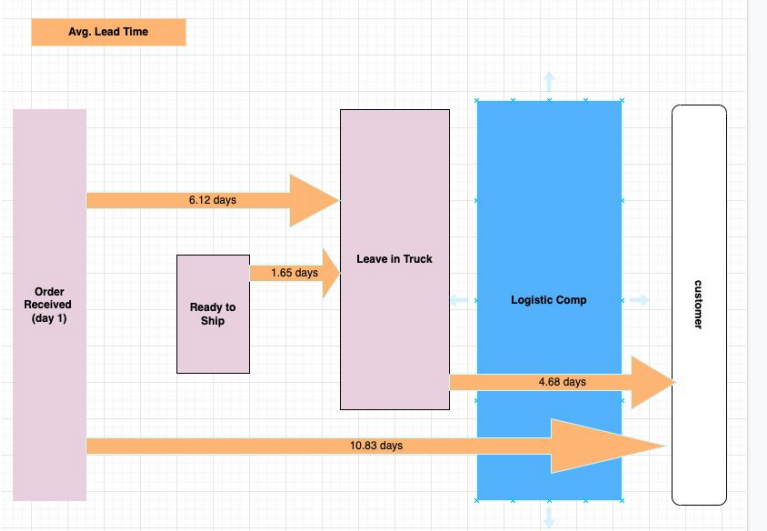
+ order_process[‘days_ready_truck’]

how many days from order to arrival

+ campaign_merge[‘days_order_arrival’]

how many days between pickup date and arrival

+ ready_to_ship_merge[‘days_truck_arrival’]



- What KPIs should we look at?
  - How many days does a package take from order to arrival at customer?
  - Does the logistic company keep their promise of a 3 day delivery?
  - Are there any steps in the process that take up more time than we think they should?
  - Is there a significant difference in the duration of the process depending on the weekday?

- How are these KPIs performing? What are our goals? (need to discuss)
  - Overview of the delivery process
  - Develop a dashboard for monitor KPIs

outline

- Overall Goal
- Exploratory Data Analysis
  - Data Cleaning
  - Data Merging
  - KPI Development
- Data Visualization
- Conclusion (action plans and next steps) & Limitation