CYO-project | HarvardX

Marco Schicker

14 5 2021

Contents

1 Executive summary

The current corona crisis and past crisis like in 2008 have shown over and over again that companies that are not able to react fast enough and have too much capital bound along their supply chain have a problem to adapt and an increased risk for bankruptcy. In order to form resilient, effective and efficient supply chains lead times are a very central key performance indicator (KPI) to optimize. Any change you make to a process will take at least one avg lead time to show effect. If your lead times are at 100 working days - which is not so uncommon and can be much longer in many cases - you will have a reaction time of almost half a year for most measures to kick in in case of a crisis. This can be too long for some companies and will make a timely check of measures almost impossible.

The goal of this project is to predict Order-to-Casch (O2C) lead times at the time of ordering by the customer, depending on product, the sales channel, sales organization, the production site and seasonal trends, along with what data we can get out of the ERP. In order to do this we will analyze system logs of all sales orders from the last couple years.

2 Background information

2.1 Lead time

When talking about lead time we make a difference between the length and the volatility/reliability of lead time.

- A short lead time of material throughout the whole supply chain means that material takes the most direct route between producer and customer, not being stocked, reworked, stopped for any reason other than having value added to it. This concept of focusing on lead time is well-known from LEAN Management. Any activity done to the product that doesn´t increase the value from customer perspective is considered waste. The target is to eliminate waste as much as possible and reduce necessary waste as much as possible. A short lead time also means that a company receives the money from the customer fast and doesn´t need to finance a lot of material and processes in between.

- A low volatility of lead time makes a supply chain more predictable, regardless of how short or long it may be. This will help any company to give reliable information to customers, provide a good Level of Service (LoS) and overall increase trust in the company´s performance. Further, a predictable supply chain enables us to reduce safety stocks and hence, reduce the capital employed significantly. Last but not least a low volatility makes it much easier to automate and level production flow. In order to compare very different lead times we use the coefficient of variation (CV), which is the standard deviation divided by the mean. It is also known as the relative standard deviation (RSD). A high CV makes prediction harder if we don´t find the origin of the variation in the data.

2.2 Order tracking

Most companies work with so called order status updates once a concrete order has passed particular milestones, e.g. manufactured, packed, shipped, delivered, etc... Based on the business model and the setup of the supply chain these status differ and are adapted to the company´s needs. The data can be stored across different systems (ERP, CRM, web shop, 3rd party applications) and needs to be stitched together to create a conclusive data set.

2.3 Company description

The company that we will use to showcase ML in lead time prediction is a european supplier of shading solutions. Manufacturing sites and sales organizations exist in multiple countries. The business model follows a mixed approach of B2B2C and B2C, that means running own organizations that provide consulting, installation and service to final customers as well as selling product to retailers. This makes it more challenging as all orders run through the same factories but serve different business models and different supply chains. The final target is the same though: reducing lead time to be agile and reduce the effects of having a lot of capital bound in the supply chain.

3 Methods & Analysis

The process to train the models consists of the following basic steps:

- Data import
- Data exploration, cleaning and visualization
- Creating validation, training and test set
- Training and comparing different models
- Validation of best model

3.1 Data import

The data provided by the company has been widely anonymized and shared on a web site as a CSV file. It consists of two different files:

1. "O2C.csv" - a file showing *one observation per order status change* and including information like timestamp, corresponding project ID, sales data, volumes etc...
2. "project.csv" - a file showing *one observation per projectID* linking projects to direct sales organizations and making it possible to use these as predictors. the project-file is leftjoined to the O2c file to combine all data in one table.

The data is only roughly cleaned and needs to be altered in order to be used.

3.2 Data exploration, cleaning and visualization

In order to work with the data set it is analyzed and altered.

3.2.1 Overall analysis

To get a first overview about the data the following summary is created:

```
##    country             DS.PB            division.x          statusID
##  Length:3973420     Length:3973420     Length:3973420     Length:3973420
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  status_name         created_dt         mod_dt            PROJID
##  Length:3973420     Length:3973420     Mode:logical   Min.   :1000046
##  Class :character   Class :character   NA's:3973420   1st Qu.:6032141
##  Mode  :character   Mode  :character                  Median :6337811
##                                                      Mean   :5231301
##                                                      3rd Qu.:6533461
##                                                      Max.   :6761353
##   corr_project       salesID             itemID             sum_qty
##  Min.   :0.00000   Length:3973420     Length:3973420     Min.   : -1216.00
##  1st Qu.:0.00000   Class :character   Class :character   1st Qu.:     4.00
##  Median :0.00000   Mode  :character   Mode  :character   Median :     8.00
##  Mean   :0.05286                                        Mean   :    72.81
```

```
##  3rd Qu.:0.00000                                          3rd Qu.:    32.00
##  Max.   :1.00000                                          Max.   :250420.00
##      sum_m2         days_on_status     channel_name        division.y
##  Min.   :  -2.081   Min.   :   0.000   Length:3973420     Length:3973420
##  1st Qu.:   0.000   1st Qu.:   0.000   Class :character   Class :character
##  Median :  10.080   Median :   0.000   Mode  :character   Mode  :character
##  Mean   :  54.722   Mean   :   7.679
##  3rd Qu.:  41.200   3rd Qu.:   5.000
##  Max.   :9828.832   Max.   :1457.000
##  divisiongroup
##  Length:3973420
##  Class :character
##  Mode  :character
##
##
##
```

We can see that the O2C data table consists of the following columns:

- country - the country in which the order was sold
- DS.PB - direct sales (=our own shops) or partner business (=retailer)
- statusID - the different stati that any order (=salesID) can go through
- status_name - the corresponding short description of the statusID
- created_dt - timestamp automatically generated when the status change has been made
- corr_project - binary to mark correction projects
- PROJID - project ID. Any project can have 1 or more SalesIDs associated to it. An order cannot be finalized if there are other unfinished orders within a project.
- salesID - an order line identifier. It is created by appending an order number to the projectID
- itemID - The product sold within the order line. Can be empty in some cases if there are multiple items included. This is true only for some sales channels
- sum_qty - how many items are sold
- sum_m2 - how many m^2 have been sold within this order
- days on status - time in days that an order stays in a certain status.
- channel_name - detailed information which organization or planner is selling the order
- divisiongroup - high level differentiation if it is a service, installation or retail project
- division - mid level differentiation between 17 divisions.

The dimensions and the first rows of the data file are as follows:

```
## [1] 3973420      17
```

```
##    country DS.PB division.x statusID                    status_name
## 1 country1    FH         FH     A000                 Auftrag eröffnet
## 2 country1    FH         FH     A100                 Auftrag erstellt
## 3 country1    FH         FH     A300             Produktion gestartet
## 4 country1    FH         FH     A310 Produktion beendet / Eingang Hub
## 5 country1    FH         FH     A400      Rechnungsvorschlag erstellt
## 6 country1    FH         FH     A000                 Auftrag eröffnet
##             created_dt mod_dt  PROJID corr_project      salesID   itemID
## 1 2017-05-23T08:06:39Z     NA 3000002            0 3000002-A001 P402A.01
## 2 2017-05-23T08:09:13Z     NA 3000002            0 3000002-A001 P402A.01
## 3 2017-06-08T06:47:55Z     NA 3000002            0 3000002-A001 P402A.01
## 4 2017-07-21T04:17:51Z     NA 3000002            0 3000002-A001 P402A.01
## 5 2017-07-21T21:02:42Z     NA 3000002            0 3000002-A001 P402A.01
## 6 2017-08-29T14:41:02Z     NA 3000015            0 3000015-A001 P502B.02
##   sum_qty    sum_m2 days_on_status          channel_name division.y
## 1       8  15.16205              0 Zehentleitner JÃ¼rgen Fachhandel
## 2       8  15.16205             15 Zehentleitner JÃ¼rgen Fachhandel
## 3       8  15.16205             42 Zehentleitner JÃ¼rgen Fachhandel
## 4       8  15.16205              0 Zehentleitner JÃ¼rgen Fachhandel
## 5       8  15.16205              2 Zehentleitner JÃ¼rgen Fachhandel
## 6      28 176.07984              0    Monsberger Andreas Fachhandel
##   divisiongroup
## 1        RETAIL
## 2        RETAIL
## 3        RETAIL
## 4        RETAIL
## 5        RETAIL
## 6        RETAIL
```

The data classes are shown in the following table

```
##        country          DS.PB     division.x       statusID    status_name
##    "character"    "character"    "character"    "character"    "character"
##     created_dt         mod_dt         PROJID   corr_project        salesID
##    "character"      "logical"      "numeric"      "integer"    "character"
##         itemID        sum_qty         sum_m2 days_on_status   channel_name
##    "character"      "numeric"      "numeric"      "integer"    "character"
##     division.y  divisiongroup
##    "character"    "character"
```

The Status IDs are especially interesting as they mark different gates that our order passes. We will concentrate on the following statusIDs:

- A000 - an order is being created
- A100 - an order is opened, having defined products, delivery dates, etc.
- A105 - an order is visible to the plant and is being scheduled for production. At this stage the order should ideally flow all the way through the supply chain until we can invoice the customer
- A300 - production is started
- A310 - production is finished an goods are ready to ship
- A320 - re-stocked item shipped.
- A330 - goods are delivered to the customer site and installation is in process
- A340 - installation is finished. Waiting for invoice proposal. In this state an order needs to wait for potential other orders that are included in the same project.

- A400 - invoice proposal finished
- A410 - invoice sent to customer. This status is the last one in the system but unfortunately not part of the data received. However we can calculate it by taking the A400 timestamp and add the duration in A400 in days

For this analysis we will start with A105, because that is the status from which onward the order must flow. In status A000 and A100 the order can be used for prediction but doesn´t cause much trouble if staying there for a long time. We will keep A000, because the data shows that most orders pass through this state (in theory all must pass through this state). We will use the timestamp of the A000 state to create predictors for year, month and day. A100 will not be used for calculation of total lead time but will be kept for analysis purposes and to visualize the production funnel. Further we will drop A320 as it is a not widely used intermediate statusID, there is only 1 occurrence.

Let us define the starting point and see how many unique SalesID (=orderlines) are in the data set.

```
### how many unique sales_ID (=order lines)?
orig_salesID <- n_distinct(O2C$salesID)
orig_salesID
```

```
## [1] 768020
```

3.2.2 Exclude special cases (correction projects, "90x" products, Service orders and Inter company orders)

So called *correction projects* are entered in the system to handle any quality deviations that include manufacturing or delivery for a customer. Since these correction projects follow a different workflow they are excluded from the analysis

Not all products shown in the column *itemID* are real products that need to be manufactured, shipped and installed. Some represent hours of administrative work that can be invoiced to customer. Those itemIDs start with a *90x* and will be also excluded from the analysis.

Orders and projects that are marked as *SERV* in the column *divisiongroup* show service-orders which are an important part of the business model but also follow a separate workflow and therefore are also excluded from this analysis.

Lastly, the data also include *intercompany orders* which need to be excluded from analysis as they have a digital twin in one of the countries.

3.2.3 Find duplicate entries

Investigating the data we can find 0 duplicates.

3.2.4 Convert to wide data

The data table shows one observation per status change of any order. Since we want to predict the total lead time for a complete order we need one observation per order (=salesID) and transform our data frame so that we have the different timestamps per status change in columns and add a column for total lead time.

We can find 476999 individual orders in the data set.

3.2.4.1 Prepare data The problem we face is the fact that even though in theory an order cannot pass through the same status twice we can find exactly those cases in the data set. Out of 2602771 rows we can identify 120013 cases of duplicate combinations of salesID and statusID.

In order to handle this we will only keep the earlier status change per order.

After eliminating the problem cases we can create the wide data frame without running into issues.

3.2.4.2 Add new columns To prepare for modeling we will create new columns to differentiate between different kind of orders

- total_lt - total lead time from beginning to end, defined as the difference between the timestamp A400 and A105 plus the duration in status A400. Remember that an invoice sent is status A410, for which we don´t have a timestamp.
- order_complete - a binary variable which is 1 if there is a A400 timestamp and 0 if not. 0 means that the order has dropped out of the system at some point.
- start_year/month - two columns derived from the timestamp of each order passing status A000

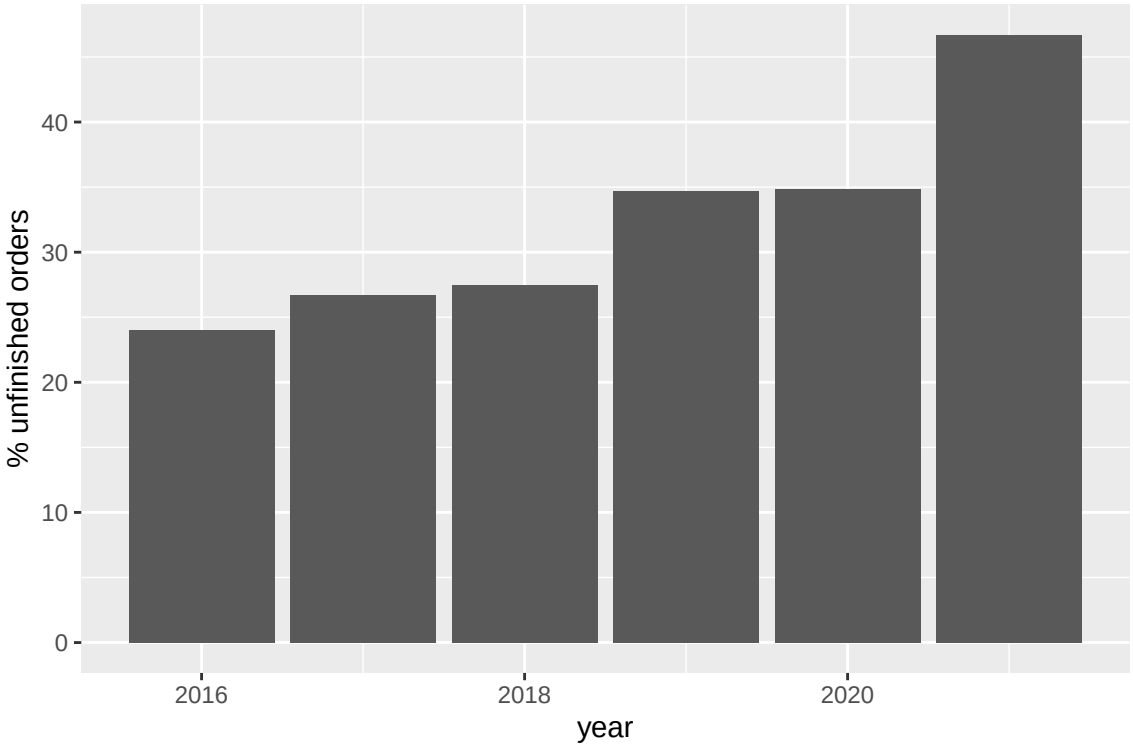
3.2.4.3 Investigate new wide data frame Let us have a look at the summary of the new wide data frame:

```
##    salesID              country       DS.PB          division
##  Length:478188      country1: 18720   DS:120015   R1     :148716
##  Class :character   country2:350835   PB:358173   RV     : 86440
##  Mode  :character   country3: 86440               I1     : 78023
##                     country4: 22193               R3     : 57153
##                                                   I2     : 39046
##                                                   FH     : 18720
##                                                   (Other): 50090
##      PROJID           itemID             sum_qty             sum_m2
##  Min.   :1000046   Length:478188      Min.   : -1216.0   Min.   :  -2.00
##  1st Qu.:3026097   Class :character   1st Qu.:     4.0   1st Qu.:   0.00
##  Median :6327217   Mode  :character   Median :    12.0   Median :  11.00
##  Mean   :5082215                      Mean   :   103.8   Mean   :  56.51
##  3rd Qu.:6522379                      3rd Qu.:    40.0   3rd Qu.:  45.00
##  Max.   :6760303                      Max.   :250420.0   Max.   :9308.00
##
##          channel_name    created_dt_A000
##  FH CH          : 67862   Min.   :2016-05-19 12:52:32
##  MO FR          : 31422   1st Qu.:2017-10-02 15:10:34
##  FH CH Mitte    : 31259   Median :2018-11-23 09:14:32
##  FH CH Ost      : 24630   Mean   :2018-12-01 04:41:52
##  MO IT          : 17629   3rd Qu.:2020-02-13 12:59:48
##  David Monteiro: 10262   Max.   :2021-05-19 09:48:10
##  (Other)       :295124   NA's   :1189
##  created_dt_A100               created_dt_A105
##  Min.   :2016-05-19 12:56:19   Min.   :2016-05-19 13:10:02
##  1st Qu.:2017-09-14 11:05:15   1st Qu.:2017-10-13 12:20:19
##  Median :2018-10-08 13:42:01   Median :2018-12-21 10:41:14
##  Mean   :2018-11-03 11:55:01   Mean   :2018-12-18 08:56:14
##  3rd Qu.:2019-12-12 07:56:27   3rd Qu.:2020-03-03 07:17:26
##  Max.   :2021-05-19 09:51:13   Max.   :2021-05-19 09:33:30
```

```
##  NA's   :98020                 NA's   :75792
##  created_dt_A300               created_dt_A330
##  Min.   :2016-05-19 14:15:03   Min.   :2016-05-19 14:54:54
##  1st Qu.:2017-10-02 06:48:00   1st Qu.:2017-07-27 12:47:12
##  Median :2018-11-26 08:52:33   Median :2018-08-24 12:58:34
##  Mean   :2018-12-05 23:45:51   Mean   :2018-09-19 20:04:48
##  3rd Qu.:2020-02-20 11:51:16   3rd Qu.:2019-10-25 01:31:34
##  Max.   :2021-05-17 04:47:51   Max.   :2021-05-17 06:40:29
##  NA's   :251488                NA's   :398126
##  created_dt_A340               created_dt_A400
##  Min.   :2016-05-20 06:22:25   Min.   :2016-05-20 12:20:06
##  1st Qu.:2017-10-26 12:56:00   1st Qu.:2017-10-20 05:57:54
##  Median :2019-01-21 07:35:48   Median :2018-11-29 09:58:13
##  Mean   :2019-01-03 00:53:03   Mean   :2018-12-17 14:46:15
##  3rd Qu.:2020-03-17 08:47:42   3rd Qu.:2020-02-25 12:31:08
##  Max.   :2021-05-19 10:22:01   Max.   :2021-05-19 10:51:13
##  NA's   :92044                 NA's   :149608
##  created_dt_A310               days_on_status_A400 order_complete
##  Min.   :2016-05-23 09:51:48   Min.   :  0.00      Mode :logical
##  1st Qu.:2017-11-03 15:58:05   1st Qu.:  0.00      FALSE:149608
##  Median :2019-02-19 08:47:18   Median :  0.00      TRUE :328580
##  Mean   :2019-01-17 23:49:41   Mean   :  0.62
##  3rd Qu.:2020-04-17 11:23:36   3rd Qu.:  0.00
##  Max.   :2021-05-19 08:26:26   Max.   :266.00
##  NA's   :223071                NA's   :149608
##      lt_105            lt_000         start_year     start_month
##  Min.   :-212.00   Min.   :   0.00   2016: 46872   6      : 48667
##  1st Qu.:  12.00   1st Qu.:  13.00   2017: 96758   7      : 47753
##  Median :  20.00   Median :  24.00   2018:102340   10     : 46031
##  Mean   :  43.06   Mean   :  54.87   2019:101233   9      : 44082
##  3rd Qu.:  43.00   3rd Qu.:  54.00   2020:100716   11     : 42441
##  Max.   :1390.00   Max.   :1390.00   2021: 29080   (Other):248025
##  NA's   :208571    NA's   :150200    NA's:  1189   NA's   :  1189
```

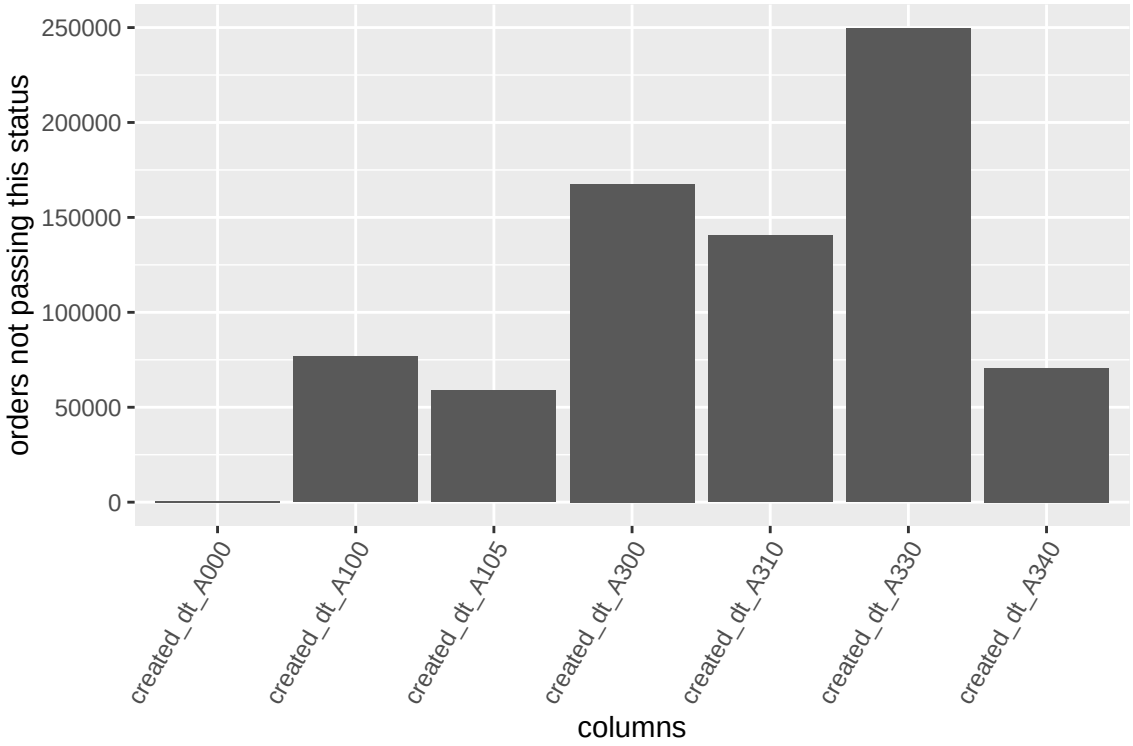
We can see that there are a few NA´s created but overall the data looks good. We need to dig deeper and define the best way to handle these orders.

As a first check we need to find out how many orders are complete and exclude all incomplete orders. As check criteria we assume that any complete order has a valid timestamp for statusID=A400 (= invoice proposal created). We can see that only a proportion of 0.6871356 of all orders are complete. Orders are always part of projects and it is possible that a project is being invoiced and closed with one or more orders still open. These orders need to be excluded from the analysis. A deep dive reveals the amount of incomplete orders per year. A sales order is assigned to a year according to its A000-timestamp.



These orders are deleted from the data set.

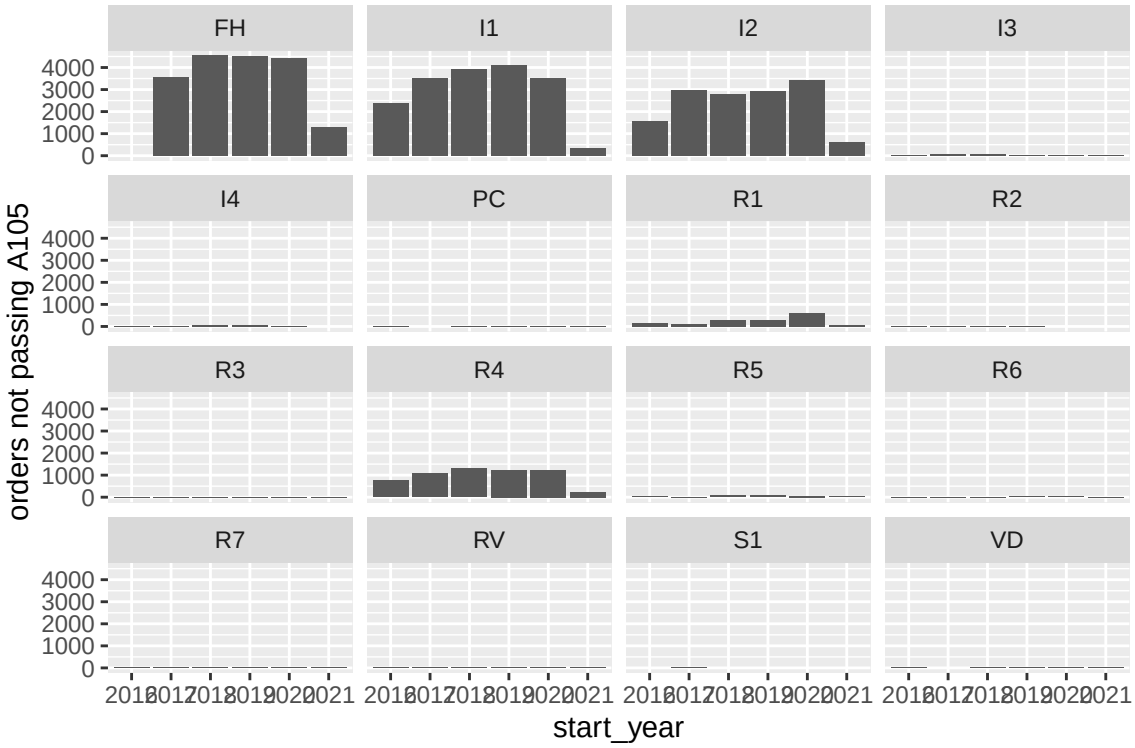
The next check is for NA´s in any of the statusID columns. An *NA* means that the corresponding order has not gone through a particular statusID. That can be ok and normal in some cases as orders from partner business usually don´t pass some of the orderstatus for direct sales (e.g. installation completed because installation is not part of their business model). However, we can see that there are quite a few orders that have never been created or opened in the first place (NA in column A000). These orders might have been opened in previous years, so we need to erase them as well to not bias the data.



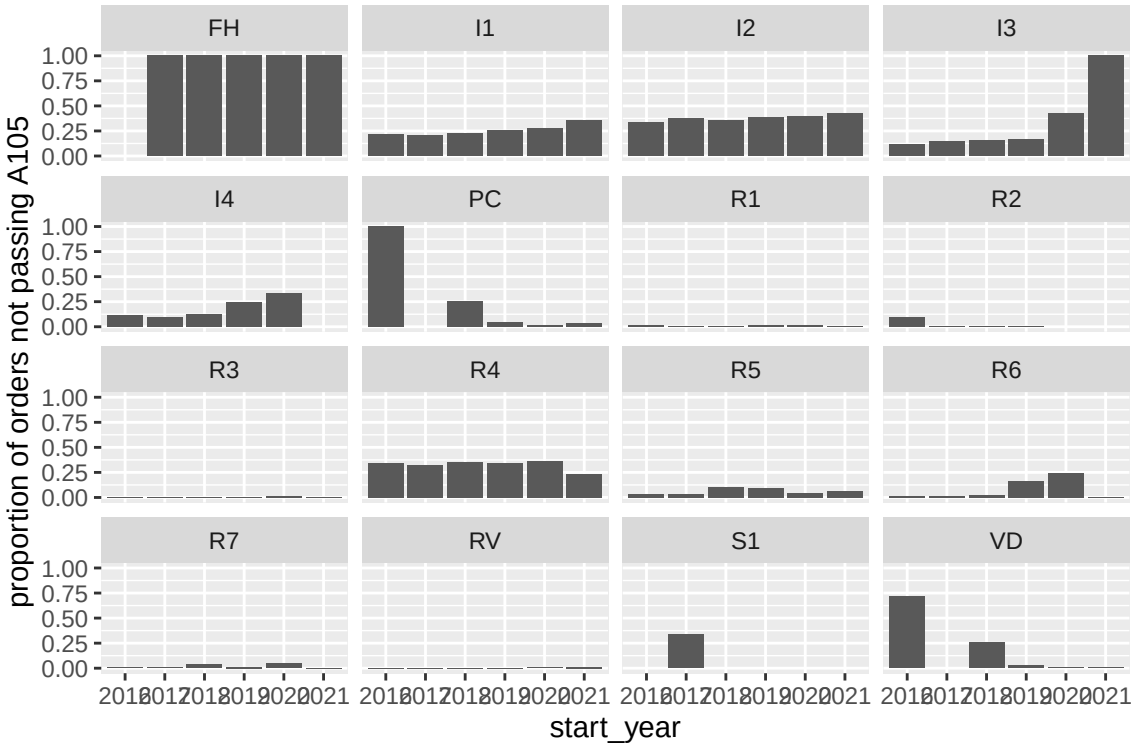
For the next check we need to look back at the purpose of this project and combine it with the insights gained. We want to predict lead times at the time of ordering. Because we don´t know the preferred delivery date we assume our supply chain should deliver according to its capabilities. That is why we need to look at orders that also pass the status A105 which means that the product has been planned in the factory. That is the moment when the supply process needs to flow. As we could see we still have some orders that don´t pass that status (NA in created_dt_A105).

Before making adjustments we need to have a deep dive and check how many order lines don´t pass status A105.

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




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



We can see that the division *FH* (that is partner business in 2 countries) has a proportion of 100%, which means none of their orders passes through status A105. In order to not exclude them completely from

comparison we will copy the timestamp of A000 to A105. Later, when we look at the difference between the two different lead times we will keep this in mind for interpretation. The second effect we can see is a proportion between 15% and 35% of orders that are not passing through status A105. The Hypothesis is that some orders are entered into the system as a dummy order during the sales process. If the customer decides to place an actual order a new system order is created. We will consider all cases other then for division *FH* to be this way and will exclude the rows from the data set.

3.2.5 Remove rows with a lead time <= 0 days

For further cleaning of the data we will erase rows with a lead time of <= 0 days.

3.2.6 Remove predictors with near zero variation

The following table shows all remaining variables checked for zero or near zero variation. The corresponding variables will not be suitable to be used for predictions.

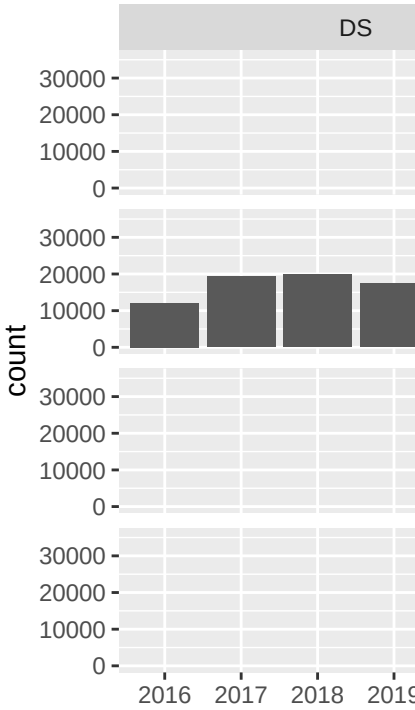
```
##                 zeroVar   nzv
## salesID           FALSE FALSE
## country           FALSE FALSE
## DS.PB             FALSE FALSE
## division          FALSE FALSE
## PROJID            FALSE FALSE
## itemID            FALSE FALSE
## sum_qty           FALSE FALSE
## sum_m2            FALSE FALSE
## channel_name      FALSE FALSE
## created_dt_A000   FALSE FALSE
## created_dt_A105   FALSE FALSE
## created_dt_A400   FALSE FALSE
## lt_105            FALSE FALSE
## lt_000            FALSE FALSE
## start_year        FALSE FALSE
## start_month       FALSE FALSE
```

We can see that no variables have a zero or near zero variation.

3.2.7 Align two tables O2C and O2C_wide

We have analyzed and altered the table *O2C_wide* a lot. For some of the following visualizations we will use the long data *O2C*. However, we need to erase all rows from *O2C* that have SalesIDs that are not part of *O2C_wide*.

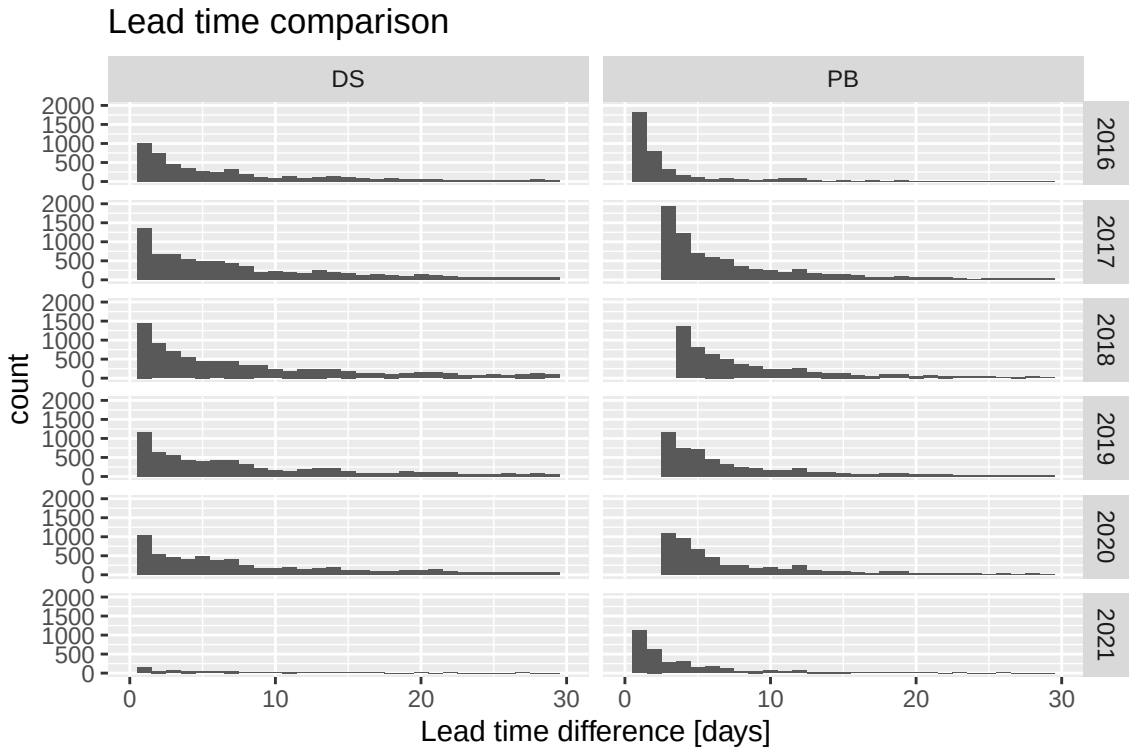
3.2.8 Various visualizations and explorations



3.2.8.1 Amount of orders and distribution among time and business model

We can see that direct sales is only active in country 2. The amount is higher compared to all other countries in total. We can also see that in 2021 there are no started and finished orders in country 3 and hardly any in country 4. Especially in country 3 we can observe a significant drop from 2018 to 2019 which is not plausible. A hypothesis is that the remaining orders have been deleted during cleaning because the practical handling of orders in the system is different compared to the other countries. The deep dive will not be done within this project but will be investigated separately.

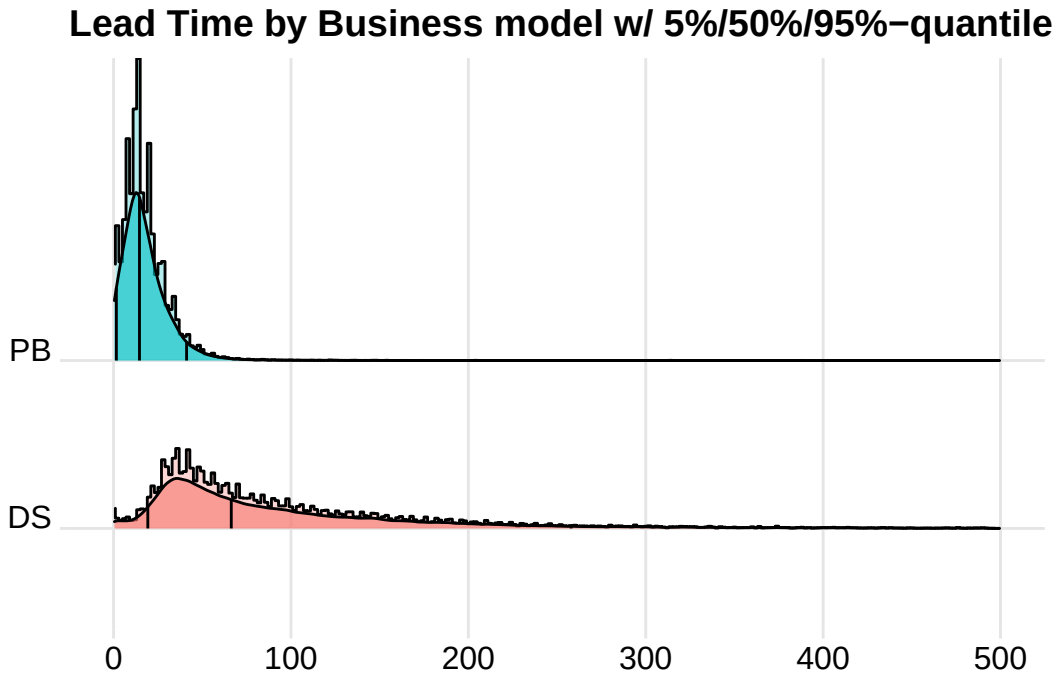
3.2.8.2 Difference between LT__A000 and LT__A105 One hypothesis mentioned above is that orders wait for some time in the status of A000 or A105. The following visualization shows the difference in days between the two lead times by year and Business model



As we can see most of the orders in partner business have a difference between 1 and 7 days with a clear peak around 1-3 days. Especially in 2021 the difference seems to be getting smaller. In direct sales this is not so clear. With a clear peak at 1 day the rest of the distribution is much more evenly distributed up to 20 days. The reason could be that some clarifications and quotations for non-standard items in direct sales need to be done *after* opening the order, in partner business these tasks happen *before* opening the order. Nevertheless, we observe a long right tail in all distributions, showing that there are quite a few orders that have been on status A000 or A100 for many days. We will not consider this in the modeling approach as we will start from status A105. However, this could be a topic in terms of customer response time as one of the possible impacts.

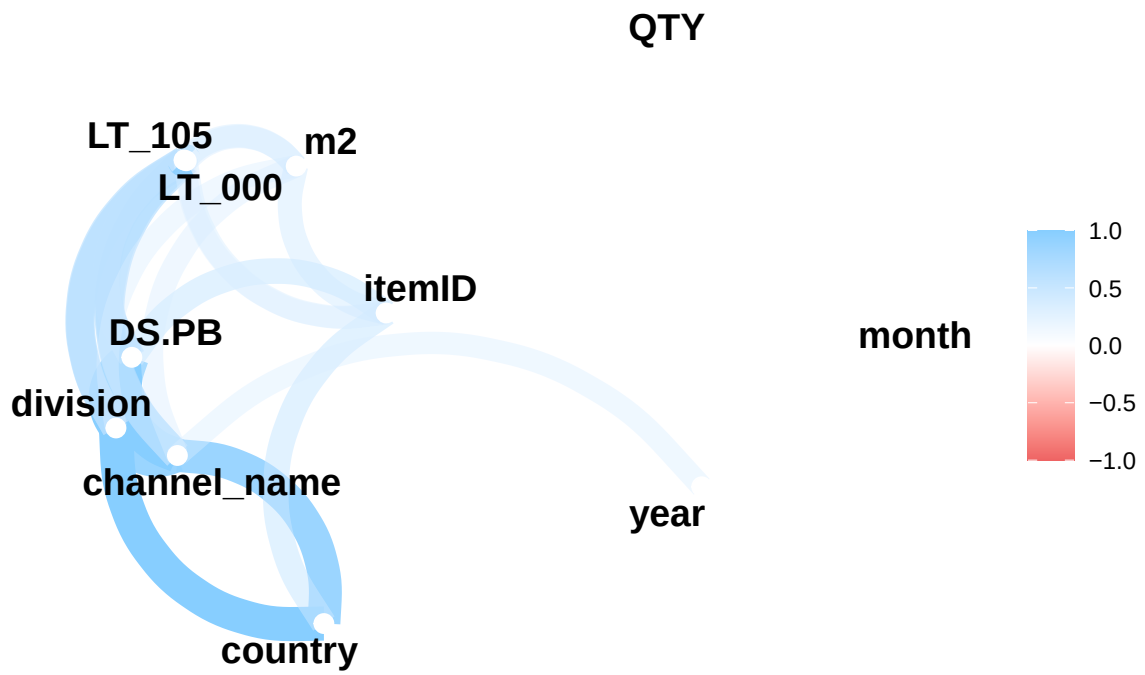
3.2.8.3 Lead time 105 Histogram and density

```
## Picking joint bandwidth of 3.23
```



The distribution shows a much shorter lead time for partner business compared to direct sales. The 50%-percentile (median) in partner business is about the same length as the 5%-percentile in direct sales - around 20 days

3.2.8.4 Correlation Correlation and association is important to look at. Highly correlated predictors require a closer look to see if really both are needed. The following table and network map gives an impression how the various variables are related. Due to the fact that we have different classes we need to follow a differentiated approach, using different measures of association/correlation:
- nominal vs nominal with Chi-square
- numeric vs numeric with Pearson correlation
- nominal vs numeric with ANOVA



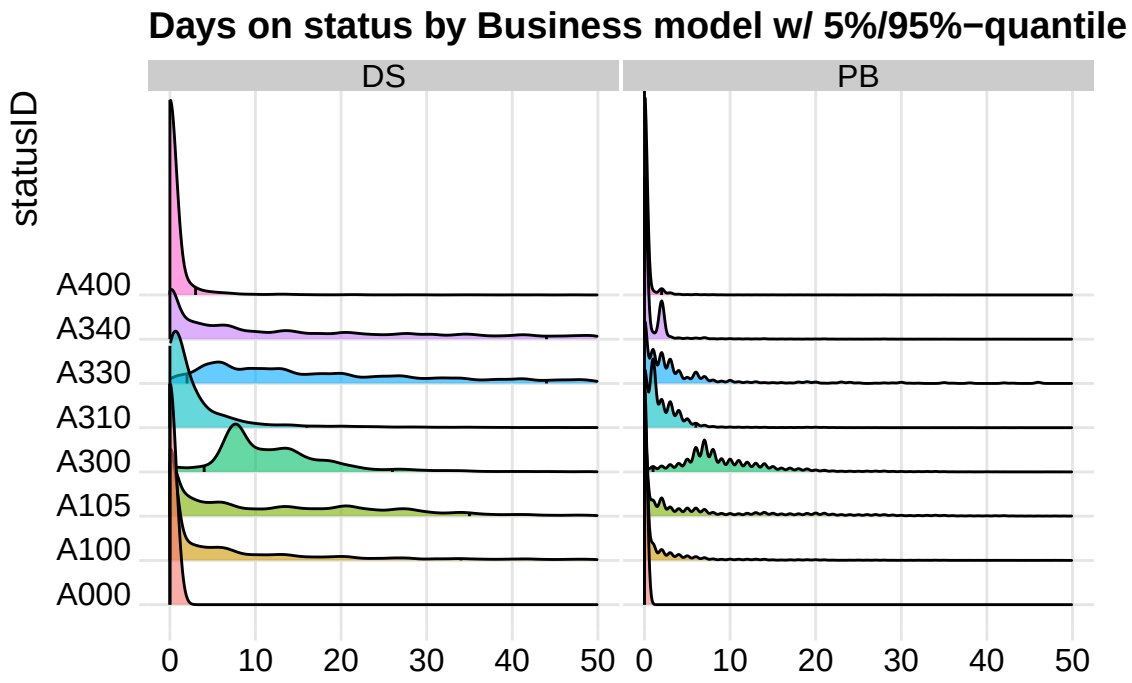
We can see that there is a correlation between DS.PB, division and channel name and country. This is expected as every division can be assigned to either direct sales (DS) or partner business (PB). The channel names are a more detailed level of divisions. We will keep DS.PB because it provides a broader overview. In termes of association with our target variable LT_105 (lead time) we can see that channel name has the highest association of all predictors with .655 before division with 0.635. After that DS.PB (.562), m^2 (.378) and itemID (.320) follow. Country (.117), year (.100), and month (.048) don´t seem to be as associated to the lead time. This result seems plausible in general, except for country which we have learned above doesn´t include orders from country 3, which is the second strongest country. Secondly, the impact of year should have been higher if the company would have improved over the last 5 years.

3.2.8.5 Duration on different status The total lead time is the sum of the durations on every order step along the way. That is why we take a closer look at the order status. The following visualizations show the how long orders stay in which status, divided by different variables.

The first chart shows the order flow on the y axis bottom to top. The x axis shows the amount of days an order stays on that status. The ridgeline shows the distribution, including the 5% and 95% quantile. This chart is done for direct sales and partner business separately.

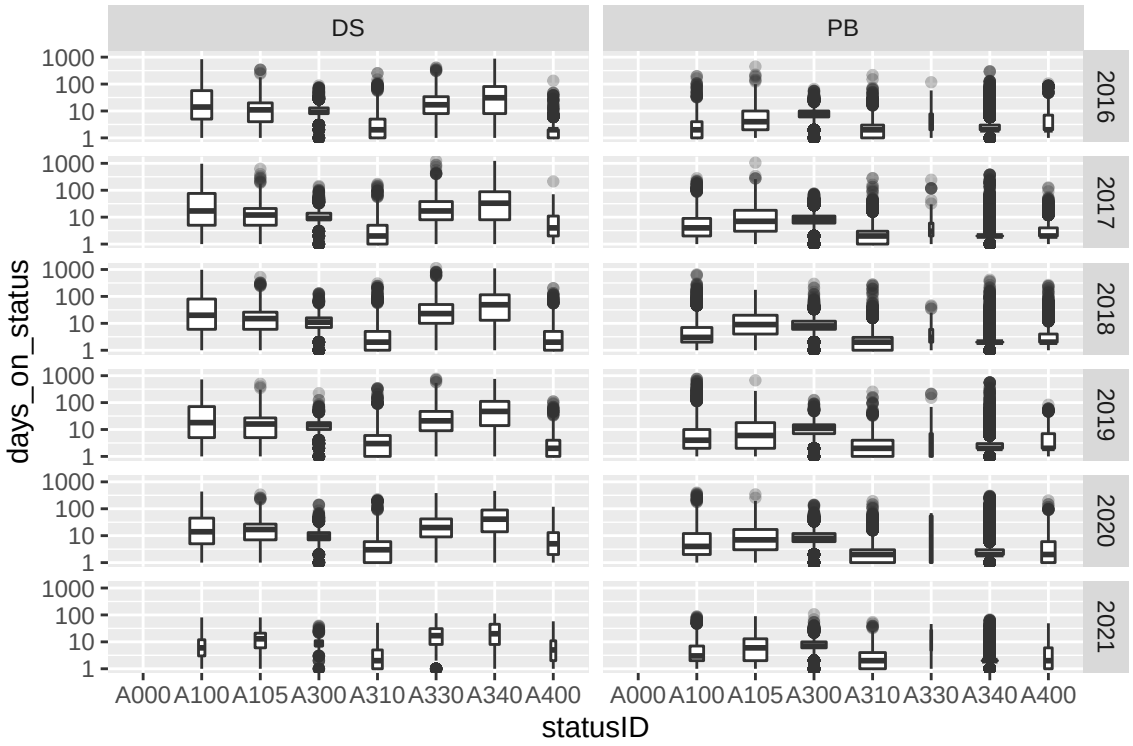
```
## Picking joint bandwidth of 0.754
```

```
## Picking joint bandwidth of 0.308
```



There is a lot of information we can derive from this chart. Firstly we see that the distribution in direct sales is much smoother compared to partner business. The reason could be some kind of scheduling. We will see later if this reflects also in the total lead time. A300 (production) seems to be close to gaussian with an avg of around 7-8 days. Remarkable is the distribution of A330 and A340 in direct sales which shows no clear peak (A330) and a long, even right tail. The 95% percentile is the highest with around 45 days. Also A105 has a long right tail in direct sales which indicates long term plant planning or order buffering and rescheduling.

The second chart is a classic boxplot chart that shows the order flow from left to right on the x axis and the days_on_status on the y axis (watch out: the Y-axis is logarithmic). The data is seperated for direct sales and partner sales and additionally by start-year of the order. The boxplot shows the mean-value as a thick line, the box boundaries show the 25/75-percentiles, the lines show the standard error and the dots visualize outliers, i.e. points with high deviation from standard. The width of the box shows the amount of orders provided to calculate the figures, i.e. a very narrow box means there are only few orders.



In general we see hardly any change over the years, meaning the performance of the supply chain has not changed much. This goes hand in hand with the correlation analysis which showed that the starting year is not much correlated with lead time. the only difference we see is in status A400 with less outliers. That means the company obviously improved in sending out final invoices. We also find very big standard errors, which indicates a very unstable and unaligned process overall. we can see that we have a a very thin box at A300 (production started) which means that the middle 50% of all orders are within a very well predicted range. At the same time we see a lot of outliers to the top and bottom, which means that out of the remaining 50% of orders some are sped up and some delayed while they are in production. The consequence for a production is a loss of productivity as this means *juggling* orders. On status A310 (logistics hub) we see the lowest median but also many outliers which indicates that many orders are being buffered at this stage. 75% of all order are below 10 days.

In direct sales the longest time is spend on A330 and A340 with the biggest uncertainty and the biggest standard error.

In partner business especially status A330 and A340 are of interest, namely because they shouldn´t exist. Due to some reason the status are used to park orders, we can see huge amounts of outliers on A340. This requires a thorough deep dive.

All in all it seems that along the supply chain every step is involved in *juggling and buffering* orders.

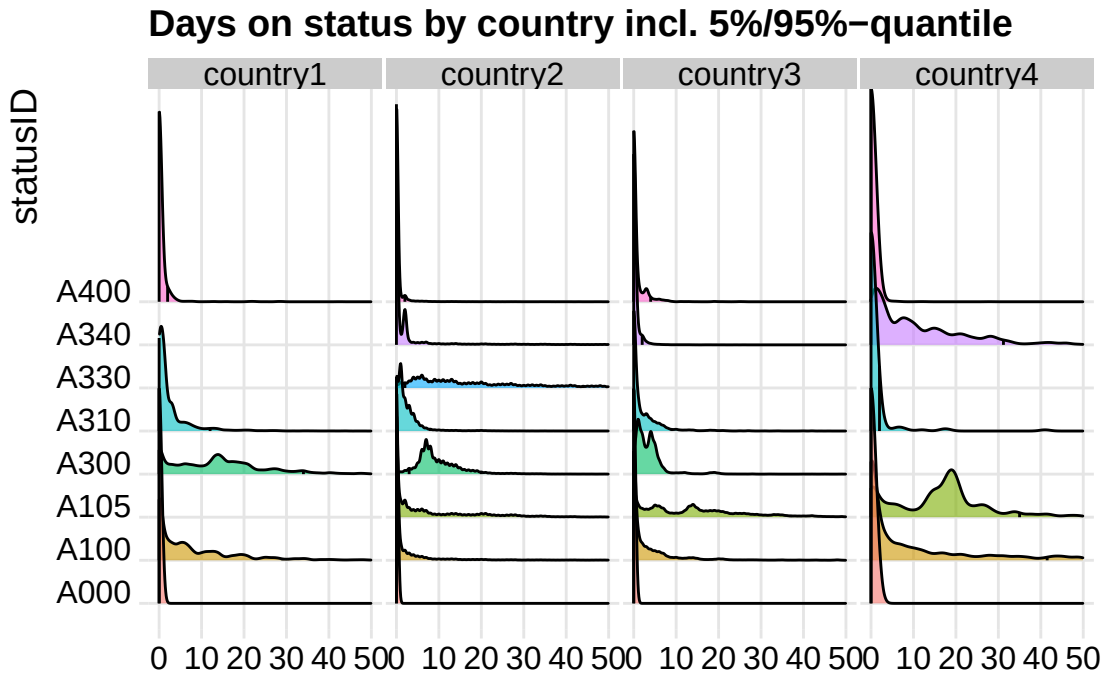
The third visualization is a ridgeline chart, this time divided by country:

```
## Picking joint bandwidth of 0.639
```

```
## Picking joint bandwidth of 0.377
```

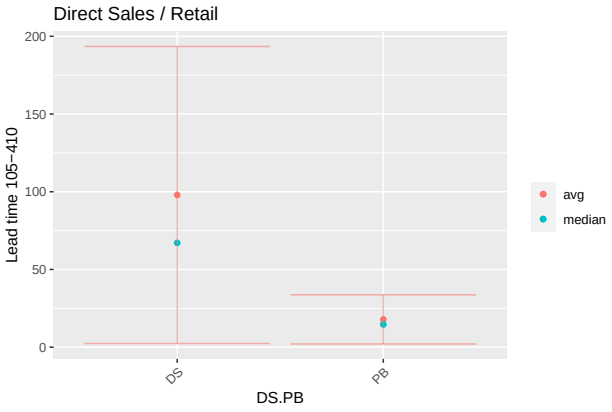
```
## Picking joint bandwidth of 0.456
```

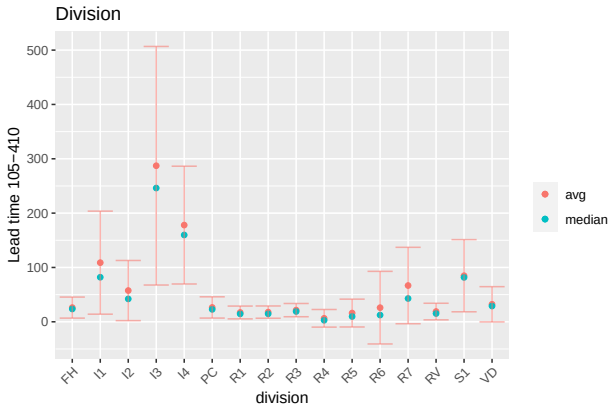
```
## Picking joint bandwidth of 1.25
```

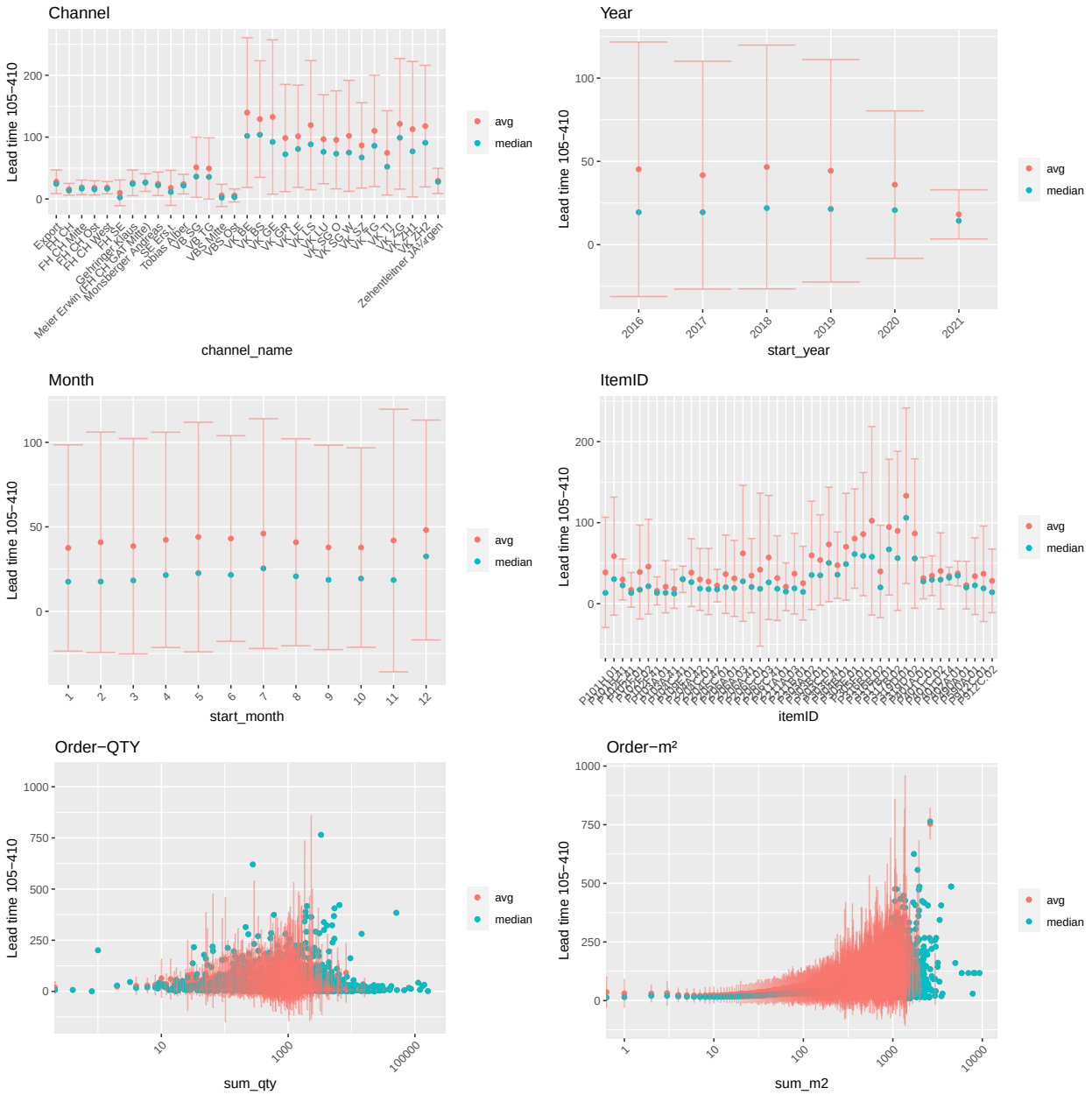


We can immediately see that country 4 is much different compared to the others. Obviously they are handling orders in a very different way compared to the others. Country 1 has a significantly longer production time. country 3 requires a closer look as we have seen that they have a very little amount of orders, so we cannot make a founded interpretation of the outcomes for now.

3.2.8.6 Lead time in relation to predictors The following charts show the leadtime in relation to various predictors. It shows the avg and the standard error in red and the median in green.





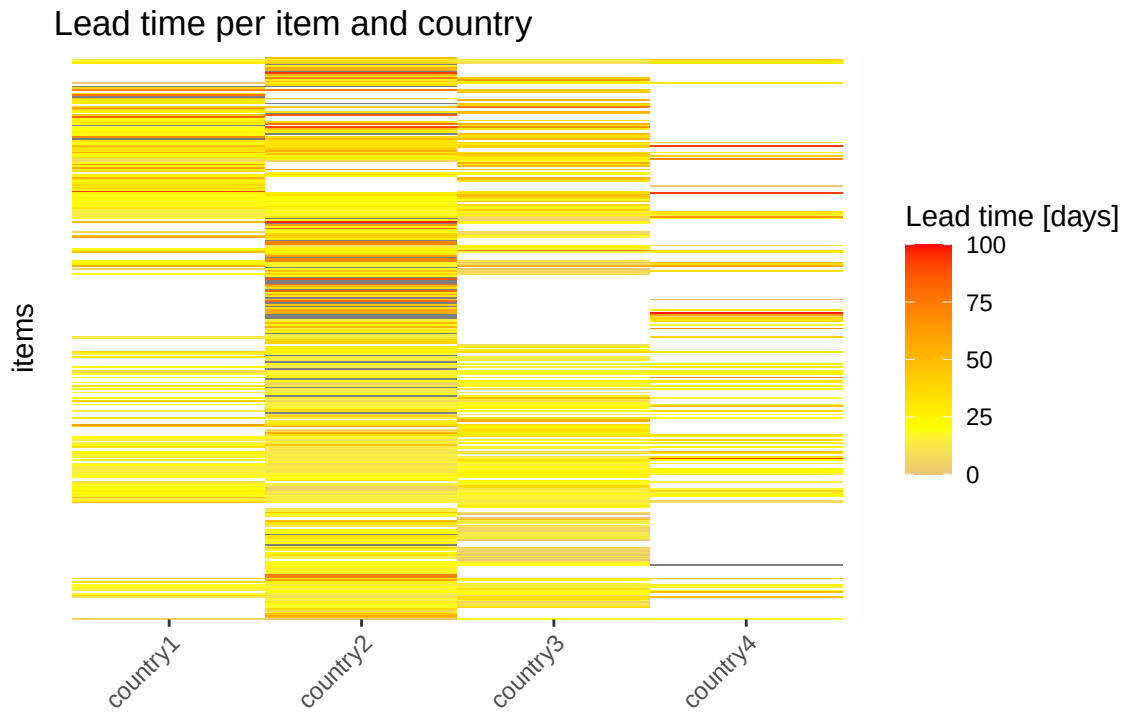


The interpretation is pretty straight forward.

- In general we see that the median is lower than the average, which means that we have a distribution with a right tail
- The standard error reaches partly below zero which means that we dont have a gaussian distribution and due to the size of the standard deviation a very widely spread distribution.
- There is a big difference between direct sales and Partner business
- In more detail we can see the split by division
- Even more detailed the split by sales channel name shows different categories of lead times
- Lead time over year shows a lower error and average in 2020 and 2021. However, to finally evaluate all orders started in those year we need to wait at least 1 or two years. So most likely there are still many open orders that have been excluded from this analysis that will influence the values negatively
- differentiation by month doesn´t show a big difference, a slightly bigger standard error in November and the longest lead time in december, which can be explained by christmas holidays.

- looking at itemID, we are looking at all items sold more than 500 times in the last 5 years. We can see quite a few differences depending on which product is sold. In avg/median as well as in standard error.
- Order QTY shows a non-structured picture. The X-axis is switched to logarithmic to better show the differences between quantities of 1 and 10,000. No clear correlation is visible.
- Ordered m^2 show a correlation as the lead time increases for larger m^2 ordered

3.2.8.7 Items, countries and Lead times The following visualization shows a tile chart with items sold in particular countries. The color indicates the lead time with red and grey being long lead times and light yellow indicating shorter lead times.



We can observe that the longest lead times are in country 2. We clearly see that not all products are sold in every country. It confirms that the lead time depends on the item number even more than on the country.

3.3 Insights gained

A first insight is that the system data contains a very big amount of orders that obviously didn´t follow the standard process. We started out with 768020 unique orders and deleted the following:
1. 155873 Service orders
2. 50761 P90x orders
3. 26455 correction projects
4. 57932 Inter company orders
5. 149608 unfinished orders (no A400 timestamp)
6. 592 orders started before the start data start cut off (no A000 timestamp)
7. 40394 dummy orders (no A105 timestamp [except country 1]) 8. 17 orders with a lead time of <=0 days

That leaves us with a proportion of 0.3744395 kept order lines. All others have been excluded from the data set.

An obviously different handling of orders in France causes most of the orders in France to be excluded. Possibly they don´t use the A400 status and go directly to A410.

The detailed deep dive confirms mostly the assumptions from the correlation analysis, so the following predictors are the most promising ones:

- channel_name
- division
- DS.PB (priority 2 as highly correlated with division)
- m^2
- itemID
- country
- year
- month

3.4 Modeling approach

In order to find the best approach to predict lead times we will try out different models, some of low complexity and some more sophisticated. Since we are predicting a continuous value (lead time) some of the models do not apply as they are build to predict categorical data.

0. Primitive models, e.g. averages
1. Linear models, e.g. lm, Bayes_glm
2. Non-linear models, e.g. kNN, SVM
3. Trees & Rules, e.g. CART, random forest

The process is to train models on data from a train data set and measure and compare the performance on a test set. Finally the best models will be validated on a validation data set. All models run on a reduced tryout data set. However, some models like kNN and SVM take lots of time to calculate on the full train data set, sometimes over 24h and had to be terminated. To achieve an acceptable balance between performance and run time we chose to change standard parameters in those cases and only used a reduced training set and 3-fold cross validation instead of 25 bootstraps.

3.4.1 Initialization

The data set is first finalized for prediction methods by deleting and changing columns that are not used as predictors or changing them to factors. This data set is called *O2C_wide_final*. O2C_wide final is split into a validation set (10% - called *O2C_val*) and one for testing and training (90% - called *O2C_wide_tt*) O2C_wide_tt is then finally split into a training set (90% - called *O2C_train*) and a test set (10% - called *O2C_test*).

Due to the fact that running the models takes a long time we have saved the models as created in the code online and will load the pre-trained models to safe time.

Secondly the target function is defined. As we are aiming to minimize the rooted mean square error (RMSE) the function is defined as follows:

```
#DEFINE RMSE-FUNCTION
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

The last two initialization procedures include calculating the mean rating (as it is used for quite a few of the model calculations) and setting up a dataframe to compare the results of the different models.

```
#Calculate Mu (average)
mu <- mean(O2C_train$lt_105)
mu
```

```
## [1] 41.3893
```

```
#set up results dataframe
rmse_results <- data.frame(method = character(),
                           RMSE = numeric())

#str(rmse_results)
```

3.4.2 Primitive models

As a first baseline we use the overall average as a prediction. As we have seen in the analysis that average and median are quite different between direct sales and partner business we will also try a primitive prediction using two averages, one for direct sales and one for partner business

method	RMSE
0.1 - Average	66.54011
0.2 - Average by DS/PB	55.22992

As we can see the baseline is an RMSE of approx. 66.5. By using two averages the RMSE can already be reduced to 55 which is a reduction of 11 points or approx 15%. However the RMSE is still approximately double the size of the average lead time for partner business.

3.4.3 Linear models

As representatives of linear models we will use the multilinear regression and Bayes-glm. The following tables show the RMSE´s for the two methods. We see that the results improve using LM by over 5 points (>10%). Bayes GLM takes a long time to run so we decided to reduce the time needed by only using half the data and instead of using 25 resamples as caret standard we use cross-validation with 3 folds. The result is slightly below 50 which is approximately on the same level as lm.

method	RMSE
0.1 - Average	66.54011
0.2 - Average by DS/PB	55.22992
1.1 - LM	49.82371
1.2 - bayes GLM	49.82366

3.4.4 Non-linear models

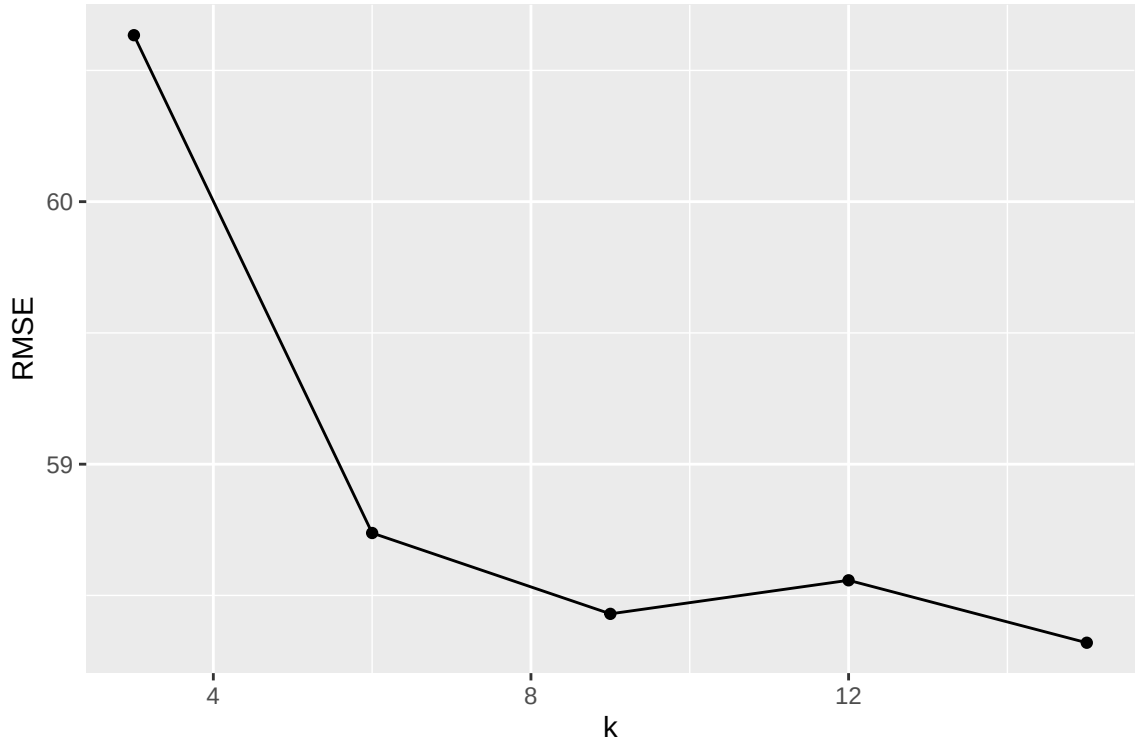
As the next group of models to use we will work with non-linear models, namely k-nearest neighbors (knn) and support vector machines (svm). Both turn out to take a long time to run and cause problems using the full amount of training data.

Using kNN requires to choose the right k, meaning to define the size of the groups of observations that will be combined as one prediction. First tries resulted in aborted simulations due to too many neighbors (>1000). The reason is most likely that we have mainly categorical predictors and more than 200,000 observations, that can lead to the same distance for many cases. We use a first model to find out which k provides the best RMSE. You can see the comparison in the graph below.

Running SVM on the full training data set resulted in a crash of R.

In order to have a basic model available we reduced the amount of data to only 5% of the training data set.

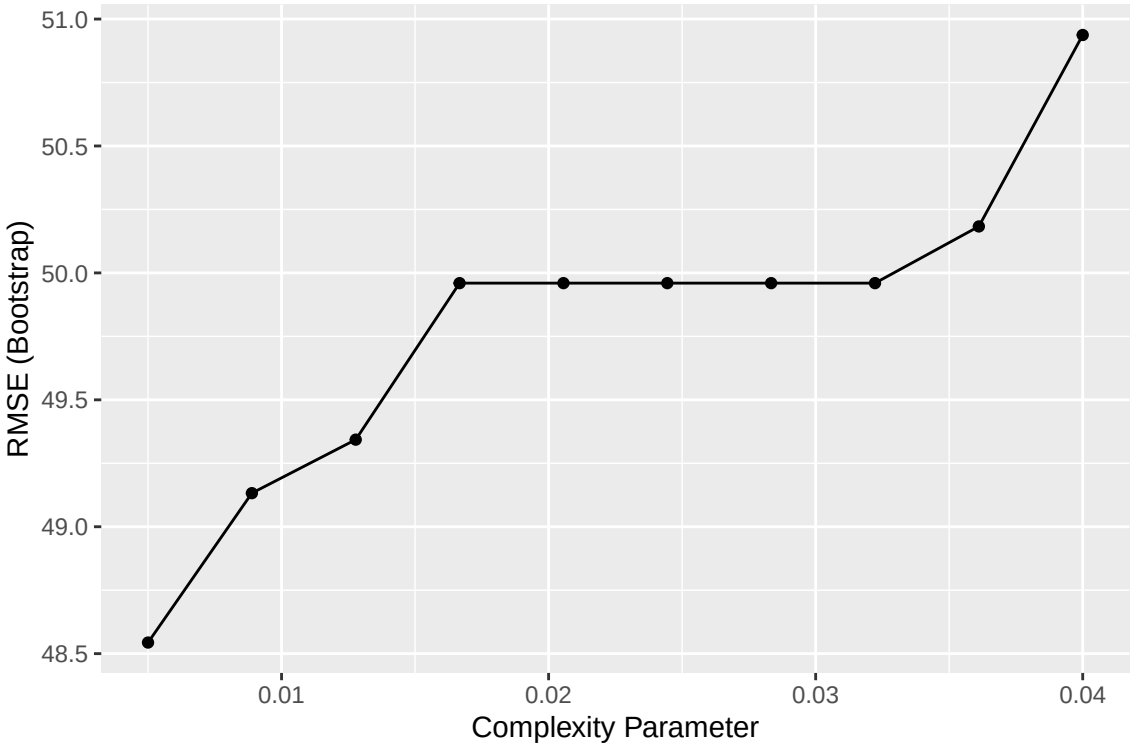
The results are worse than taking the split average (model 0.2). You can see the result for kNN below, SVM didn´t run successfully and will be left out for this project.

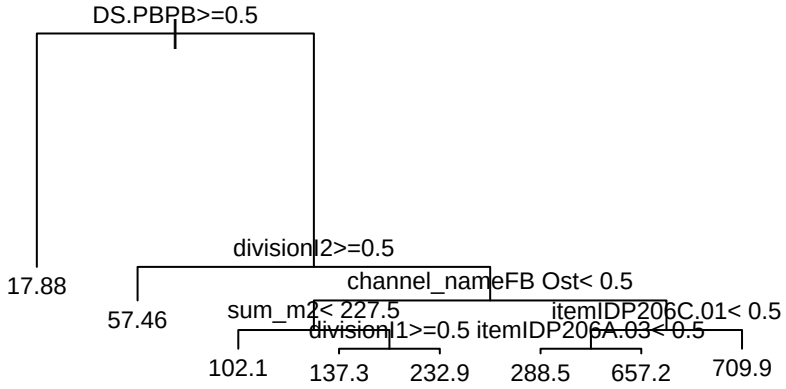


method	RMSE
0.1 - Average	66.54011
0.2 - Average by DS/PB	55.22992
1.1 - LM	49.82371
1.2 - bayes GLM	49.82366
2.1 - knn	55.98609

3.4.5 Trees & Rules

Last but not least We will try regression trees and random forests as prediction methods. Using the rpart-library we will first run a model to tune for the right parameter of cp and run create a tree based on the best performing model. The results can be seen below. Firstly see the tunegrid to identify the best complexity factor cp. As a second visualization you can see the decision tree that leads to the best RMSE. However, the RMSE achieved with the decision tree is above 53 and therefore significantly worse compared to linear models.

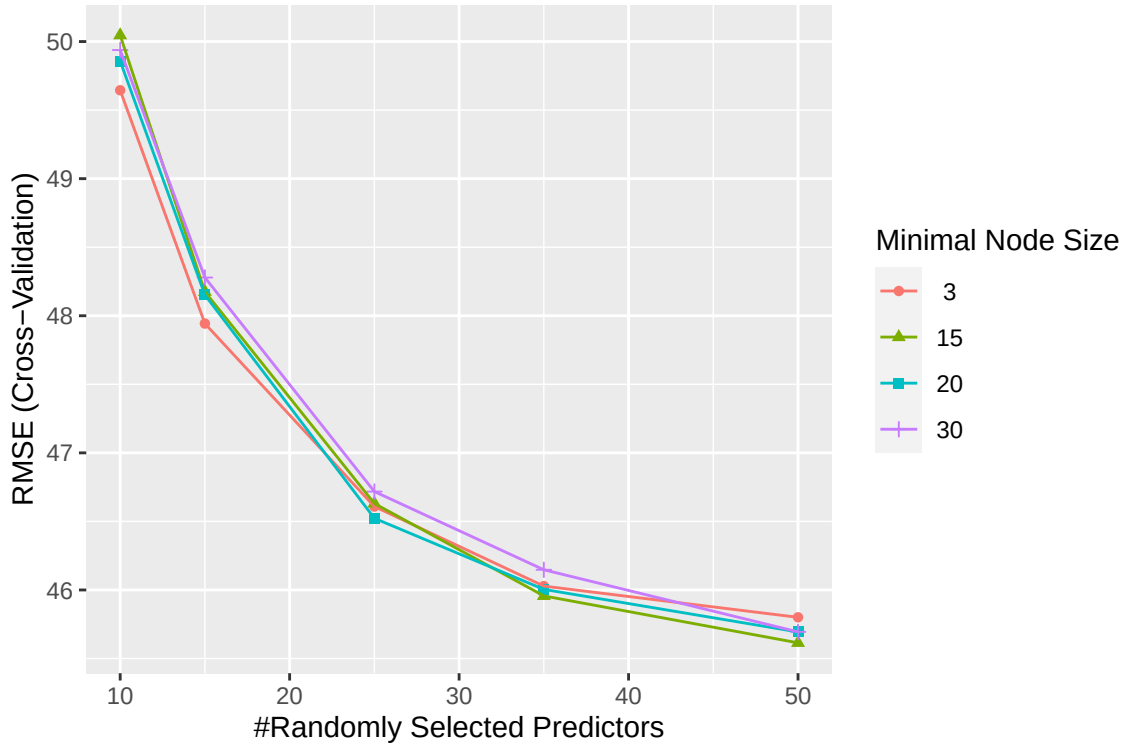




method	RMSE
0.1 - Average	66.54011
0.2 - Average by DS/PB	55.22992
1.1 - LM	49.82371

method	RMSE
1.2 - bayes GLM	49.82366
2.1 - knn	55.98609
3.1 - CART	53.41663

To find the best random forest we also use a tuning run, using the rborist package and finding the lowest RMSE for amount of minimal nodes and amount of randomly selected predictors. Below you can see the performance comparison for different values as well as the results for the best performing model. The final RMSE is added to the comparison table.



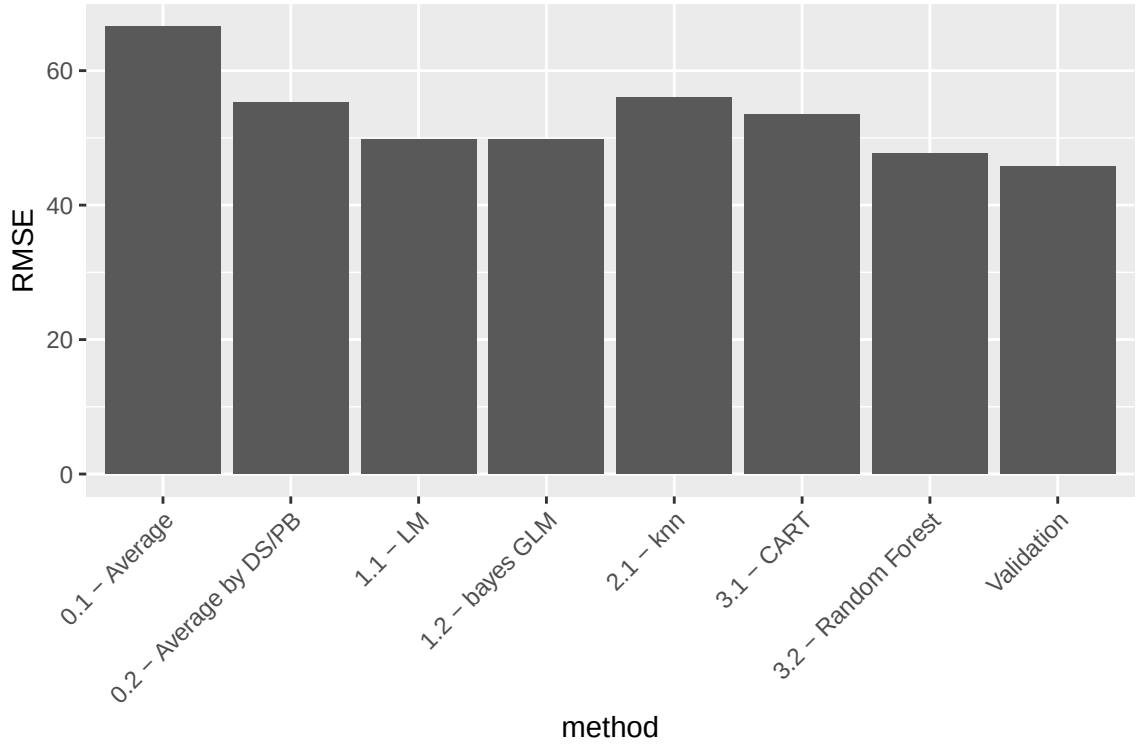

```
##    predFixed minNode
## 10        50      15
```

method	RMSE
0.1 - Average	66.54011
0.2 - Average by DS/PB	55.22992
1.1 - LM	49.82371
1.2 - bayes GLM	49.82366
2.1 - knn	55.98609
3.1 - CART	53.41663
3.2 - Random Forest	47.71466

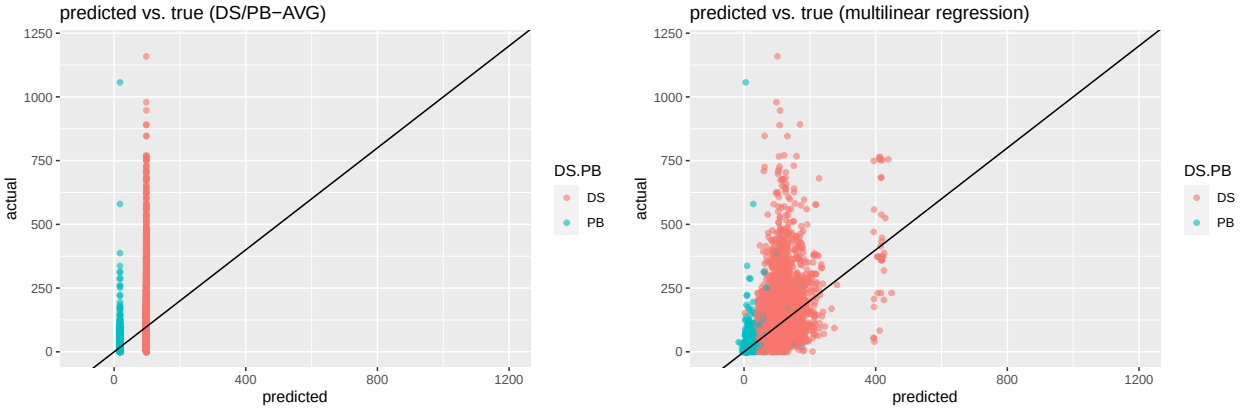
4 Results

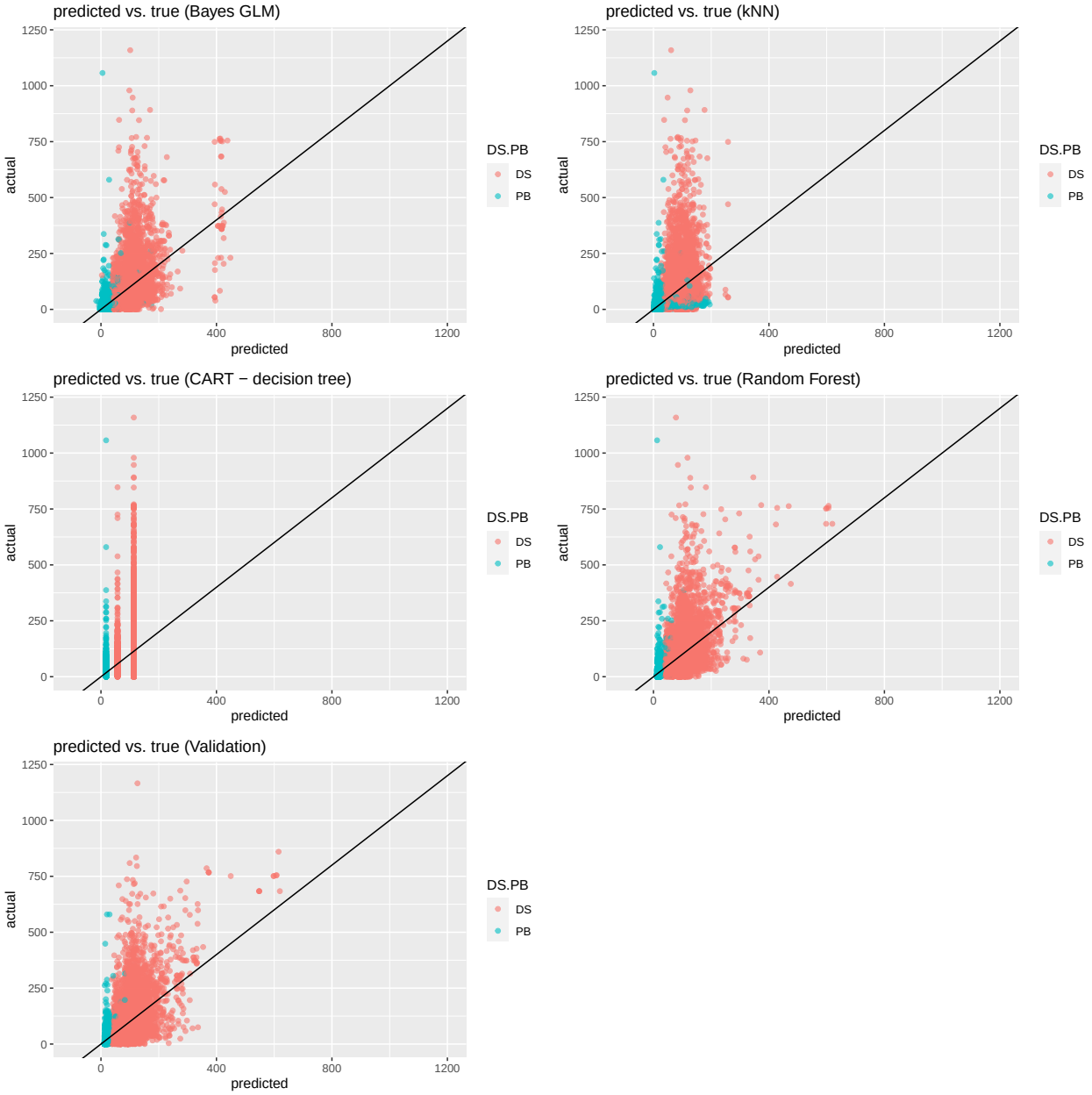
The results show that the RMSE is quite high and doesn´t allow for a precise prediction at all. The main reason lies in variation that cannot be explained by the provided predictors.

We use the best performing model (random forest) on the validation set and add the results to the overview:



The following graphs show the comparison of predicted values vs. actual values of lead time. Overall, no model is sufficiently capable of predicting orders with a high lead time correctly.




predicted vs. true (Bayes GLM)
predicted vs. true (kNN)
predicted vs. true (CART − decision tree)
predicted vs. true (Random Forest)
predicted vs. true (Validation)
actual
predicted
0
250
500
750
1000
1250
0
400
800
1200
DS.PB
DS
PB

5 Conclusion

Once we test the best performing model on the validation set we come up with a total RMSE of 45.7806997 We can see that the RMSE is bigger then the avg lead time for partner business. We conclude that based on the used data and methods we *do not reach the target* of predicting lead time with a satisfying RMSE.

5.1 Limitations

Limitations of this project are the consistency of data, e.g. the different handling of orders in different countries, which makes it hard to find common rules for data preparation. Secondly, data wrangling could be improved by diving deeper into the specifics of how orders are being kept and treated in the system. The system doesn´t recognize orders that are part of big projects. SOme orders need to wait for all other orders within a projects until it gets invoiced.

5.2 Next steps

Improving the algorithm would be possible by categorizing lead time into bins. That would enable us to use more methods like QDA, LDA, GLM, and alike.

Talking about practical next steps within the company all suggested measures can be categorized in two buckets:

1. Improvement based on the outcome
   Based on the outcomes and the conclusion the project results will be integrated in focused task force improvements in the Sales department.

2. System adaptation to integrate data analysis based on this report into daily business and leadership processes
   System adaptations could involve tracking particular key performance indicator (KPI) within shopfloor management. Further a prediction system could be improved by excluding outliers from data and therefore reduce variation of data. As the company is currently implementing process updates along with a new Core ERP the focus on aligned, cross-departmental processes is given and the results of this project can be used to specify process and system requirements.