# Part 2.

# Data analysis and visualisation

Author:

Filip Bulanda

1. Generate a histogram showing the actual delivery length with 1 minute granularity (rounded up).

The creation of histogram required data showing actual delivery times, which was not included in the received database. To calculate the delivery times for each delivery I needed to make assumption, that delivery time would be calculated as a difference in time between rows representing start and end of segment (which also had an orderID assigned to it). At the beginning I also wanted to try a different approach, which was to calculate the time difference between first 'DRIVE' that followed finished delivery and time of finishing the next delivery. I quickly realised that this overcomplicated solution would generate significantly longer delivery times than predicted values, so I switched to the simpler one.

Using MySQL, I calculated the delivery times, selected all data that I thought would be useful and exported it as .csv file for future use with MS Excel and Python scripts.

Using Microsoft Excel, I analysed the data to check if what I got made sense. That meant calculating parameters like mean, median, minimum, maximum, variance and standard deviation. Based on minimum and maximum values I could eliminate outliers and wrong data. Negative delivery times make no sense, therefore rows with delivery times shorter or equal to 0 were removed. In the data set there was several delivery times longer than 14 000 seconds (almost 4 hours). Rows with those delivery times were also dismissed.

While going through the dataset I noticed that some orderIDs appeared twice, since they were always appearing consecutively next to each other I assumed that rather than choosing one of them to calculate time I will sum their separate times.

After this initial data cleaning and analysis I prepared a histogram (Fig. 1) according to instructions (histogram showing the actual delivery length with 1 minute granularity).

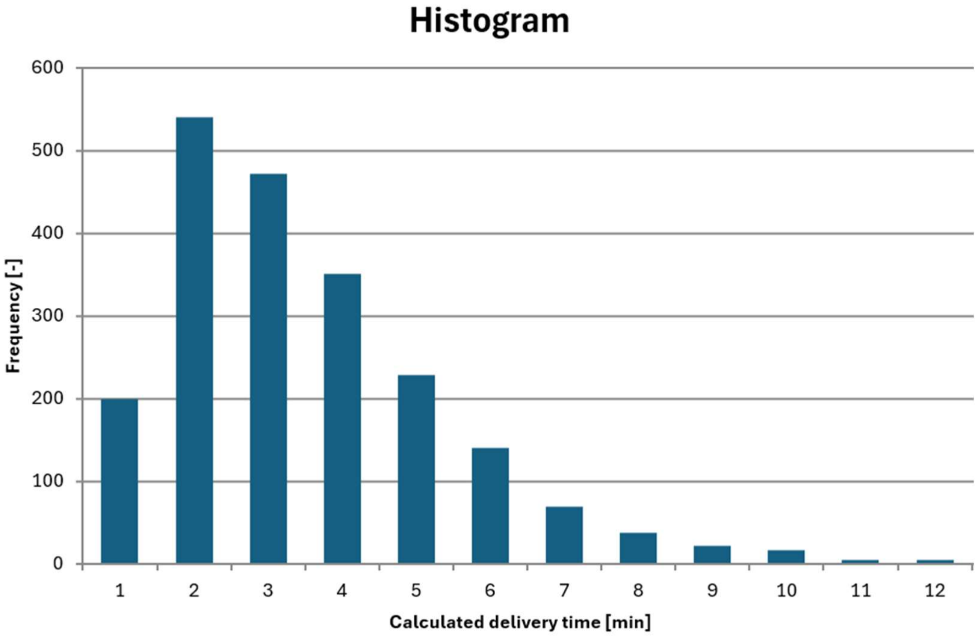


Fig. 1 Histogram for calculated delivery times

2. Generate a histogram showing prediction error (difference between planned and actual delivery times).

Data for the next required histogram was calculated using Excel (using data from exported .csv file), following the formula required by instructions:

$$e = t_p - t_o$$

Where:

$e - prediction\ error\ [s]$

$t_p - predicted\ delivery\ time\ [s]$

$t_o - observerd\ (measured)\ delivery\ time\ [s]$

To match the formatting of the first histogram, calculated errors were converted to minutes (histogram shows the prediction errors with 1 minute granularity, rounded up). Negative error means underestimation, positive values mean overestimation.

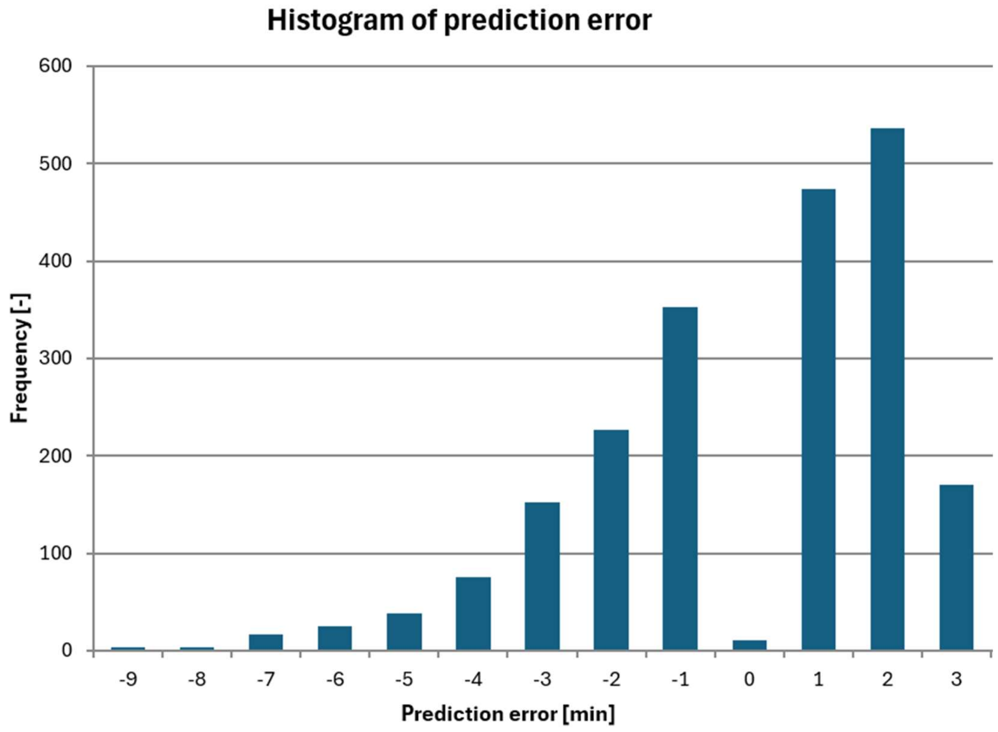


Fig 2. Histogram of prediction error for current model

3. We received insight from our drivers that delivering in one of the sectors is significantly longer than in other sectors. Generate a chart to visualise this hypothesis.

To decide whether or not given hypothesis of delivery time discrepancy between sectors is reasonable, I decided to approach this problem in two steps. First one was to create a simple box plot using Excel (see Fig. 3). Each box was created for single sector with values of calculated delivery time. It seems relatively obvious that sector 1 differs from other sectors. Therefore the provided insight was most likely accurate.
Box plots are helpful for this kind of comparisons because they show parameters like median (line inside the box), distribution of the measured values (whiskers cover all data, except for outliers – dots outside of whiskers)

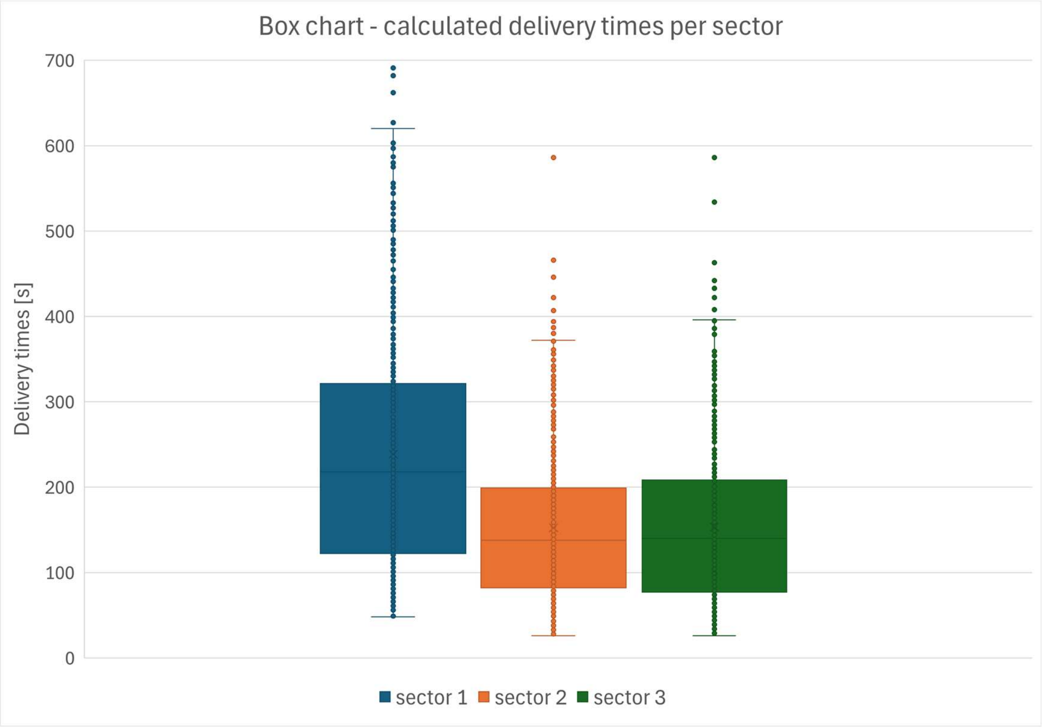
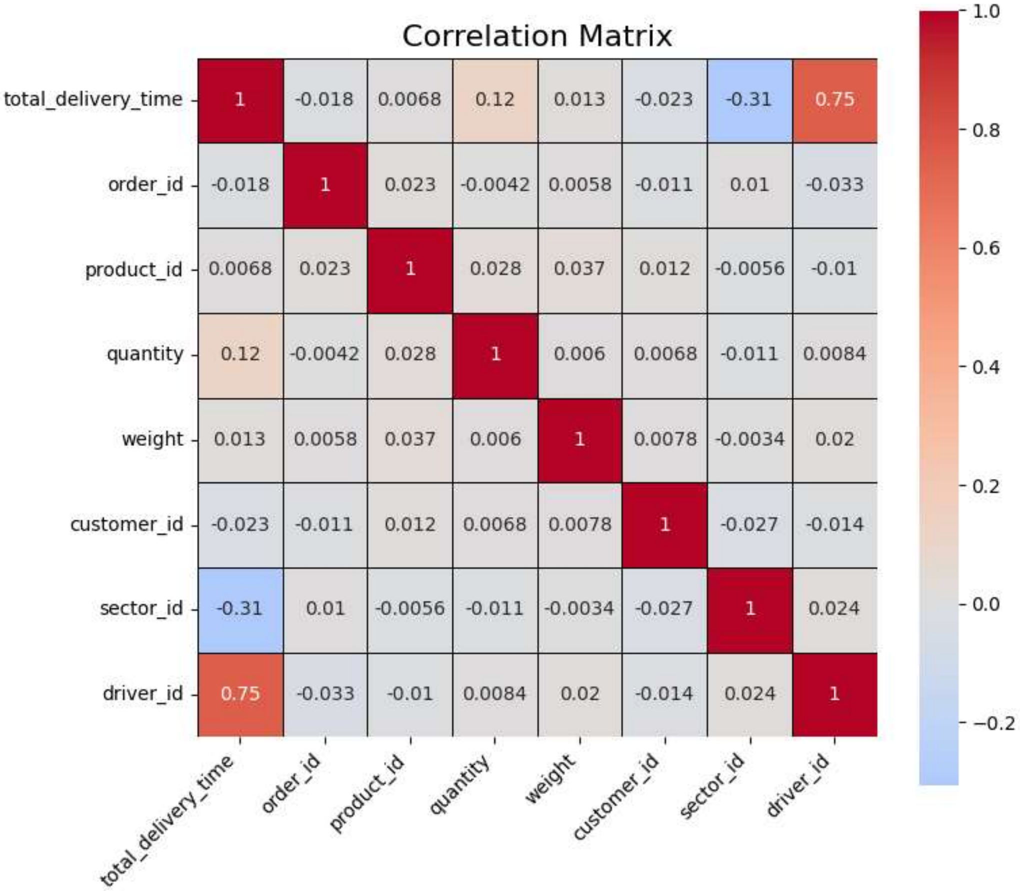


Fig 3. Box plot showing delivery times occurring at each sector

For additional information I carried out a Kruskal-Wallis H-test. Tests was conducted using Python and its chosen libraries (Pandas, SciPy). Said test is one of non-parametric statistical tests used to determine whether two or more data groups are from the same population (are from the same statistical distribution). The test confirmed the hypothesis that delivery times in sector 2 and sector 3 are statistically identical, and times in sector 1 and other are statistically different.

4. Play with the data by grouping, aggregating and remodelling it. Are you able to find any correlations or trends that could be valuable for prediction quality improvement? Describe briefly your findings and visualise them on charts.

For this part, first thing I did with the data was to create a correlation matrix for all main features that could possibly have any impact on delivery times. I exported all those features from MySQL database to .csv and created the matrix with Python script (Fig. 4). Correlation matrix shows how features (variables) of dataset change compared to each other. If both variables grow at the same ratio the correlation will be high (approaching 1). If both variables change their values at the same ratio, but one grows and the other falls the correlation will approach -1. If the changes of variables change in completely different ratios the correlation will approach 0. (For instance, if taller buildings consistently lead to longer delivery times, we'd see a high positive correlation between building height and delivery time)



| | total_delivery_time | order_id | product_id | quantity | weight | customer_id | sector_id | driver_id |
|---|---|---|---|---|---|---|---|---|
| total_delivery_time | 1 | -0.018 | 0.0068 | 0.12 | 0.013 | -0.023 | -0.31 | 0.75 |
| order_id | -0.018 | 1 | 0.023 | -0.0042 | 0.0058 | -0.011 | 0.01 | -0.033 |
| product_id | 0.0068 | 0.023 | 1 | 0.028 | 0.037 | 0.012 | -0.0056 | -0.01 |
| quantity | 0.12 | -0.0042 | 0.028 | 1 | 0.006 | 0.0068 | -0.011 | 0.0084 |
| weight | 0.013 | 0.0058 | 0.037 | 0.006 | 1 | 0.0078 | -0.0034 | 0.02 |
| customer_id | -0.023 | -0.011 | 0.012 | 0.0068 | 0.0078 | 1 | -0.027 | -0.014 |
| sector_id | -0.31 | 0.01 | -0.0056 | -0.011 | -0.0034 | -0.027 | 1 | 0.024 |
| driver_id | 0.75 | -0.033 | -0.01 | 0.0084 | 0.02 | -0.014 | 0.024 | 1 |

Fig 4. Correlation matrix for chosen data set features

Based on provided correlation matrix we can see three features that are somewhat correlated with our real delivery time (named as ‘total_delivery_time’ in matrix). Those variables are driverID, sectorID, and quantity. We just covered the sector differences, and this correlation matrix is another way to prove that sectors had an impact on delivery time. While going through the data, I could not find any trend that would imply that quantity has any significant impact on delivery time. It's worth noting that correlation of two dataset features does not inherently mean that they are in any way connected to each other. After all, the correlation was still relatively weak at 0,12. So the result is not unexpected.

The third variable correlated with the delivery time was driverID, and (unlike quantity) its correlation was significant, at 0,75. Using provided database, in MySQL, I calculated average delivery time per driver. And represented the results on bar chart (Fig. 5). Additionally, I created a box plot with delivery times per driver (Fig 6.)

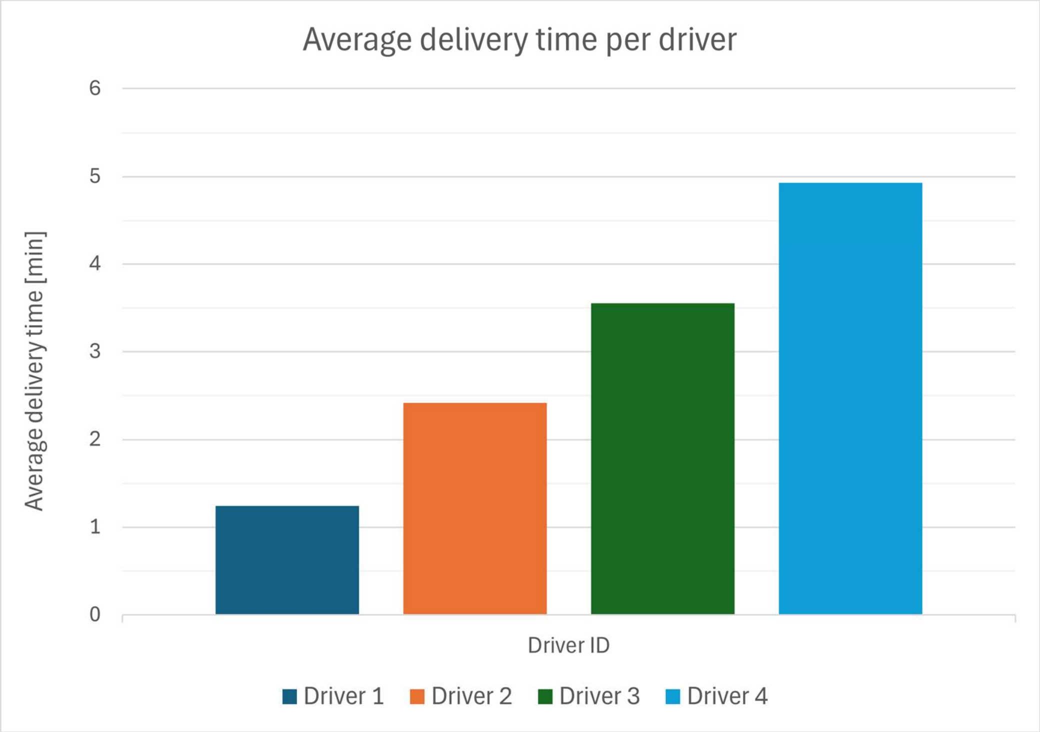


Fig. 5 Bar chart representing average delivery time of each driver

Based on the facts, that I removed outlier values, each driver had almost the same amount of deliveries and covered almost the same amount of distinct customers, I assumed that the average value would be representative enough. It is an important assumption, because average is not always the most descriptive parameter.

What we can see from the chart is that all drivers took different amounts of time to deliver the order. Driver with id=1, seemed to work most efficiently, while driver with id=4 seemed to be the least efficient.

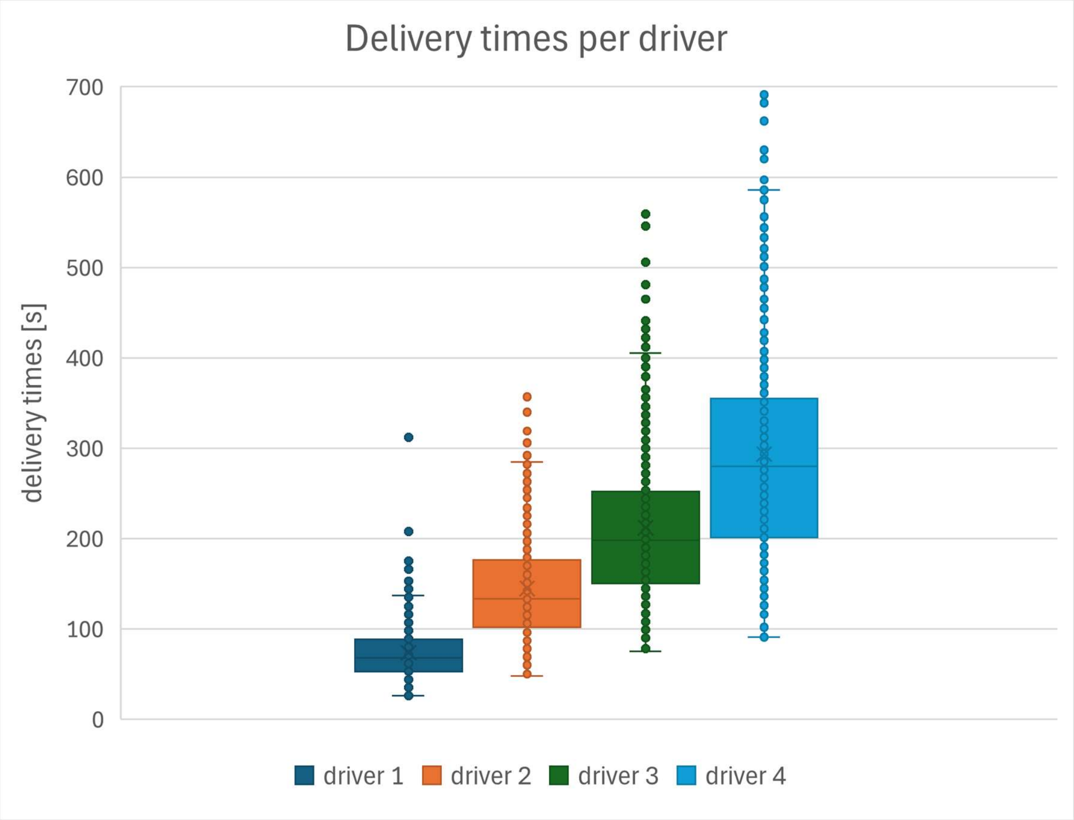


Fig. 6 Box chart showing delivery times for each driver

As seen on the box chart (Fig 6), drivers had significant differences in delivery times. The difference was not only longer delivery times. For example, delivery times of driver 4, had significantly higher variance than for other drivers. Delivery times for driver 1 were contained in range from about 25 to 140 seconds, while for driver 4, they covered a range from about 100 to 600 seconds.