R-Bootcamp Assignment

Joel Luescher, Philipp Gaemperli

Luzern, 2.2.2022

Analysis of Bike Sharing in Washington D.C.

Bicycles are becoming an increasingly popular way to get from A to B quickly in Washington D.C., avoiding traffic chaos and congestion. Instead of buying and maintaining a bike, more and more people are renting bikes so they can use them where they need them when they need them. If potential customers want to rent a bike at a certain location and then all of them are rented at that time, this leaves a bad impression on the customers and they think twice next time if they really want to rent a bike.

To prevent this from happening, it is important that the bike rental system Bike Sharing Washington D.C. can predict demand as accurately as possible and provide bikes accordingly. To predict demand, Bike Sharing Washington D.C. recorded the hourly number of bicycle rentals in 2011 and 2012. In addition, the company also collected data on weather conditions and seasonality, as these parameters are likely to have an impact on the amount of rentals.

As a business intelligence consultant, it is our task to apply methods and processes to analyse systematic data of the organisation at hand in order to identify and quantify opportunities and risks for the business and to present them in observation. Specifically, we want to synthesise information and knowledge from the available data set for a decision-making situation.

Figure 1: Fig. 1: Bike sharing station.

Importing Data

In a first step, we imported the data set, which can be found under the following link on Kaeggle. The data set is in .csv format and separated by semicolns.

```
bike_data <- read.csv("hour.csv", header = TRUE)
```

Getting an overview

To get a rough overview of the data entries and variables, we inspected the first lines of the data set.

```
head(bike_data)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
##   temp  atemp  hum windspeed casual registered cnt
## 1 0.24 0.2879 0.81    0.0000      3         13  16
## 2 0.22 0.2727 0.80    0.0000      8         32  40
## 3 0.22 0.2727 0.80    0.0000      5         27  32
## 4 0.24 0.2879 0.75    0.0000      3         10  13
## 5 0.24 0.2879 0.75    0.0000      0          1   1
## 6 0.24 0.2576 0.75    0.0896      0          1   1
```

Datatypes

Furthermore, we have checked the format of the data types. It can be seen that in the data set we are dealing with numerical values and integers. In addition, for the date we have characters.

```
str(bike_data)
```

```
## 'data.frame':    17379 obs. of  17 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : chr  "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
##  $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
##  $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
##  $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
##  $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
##  $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

Convert values

From the description of the data set we have seen that certain variables have been adjusted (hum, temp, atemp, windspeed). We brought these values back into the correct format with a simple conversion.

```
bike_data$hum <- (bike_data$hum * 100)
bike_data$temp <- (bike_data$temp * (39-(-8)) + (-8))
bike_data$atemp <- (bike_data$atemp * (50-(-16))+ (-16))
bike_data$windspeed <- (bike_data$windspeed * 67)
```

Convert dteday column to date

As already described, the date is shown as a character in the data set. To make it easier to handle this value, we have changed the data type into a Date data type. In addition, the summary shows the smallest value, the first quartile, the median, the mean, the third quartile and the maximum value per column. This summary helps to get a first overview of the data and the values.

```
bike_data$dteday <- as.Date(bike_data$dteday, origin = "1900-01-01")
str(bike_data)
```

```
## 'data.frame':    17379 obs. of  17 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Date, format: "2011-01-01" "2011-01-01" ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  3.28 2.34 2.34 3.28 3.28 3.28 2.34 1.4 3.28 7.04 ...
##  $ atemp     : num  3 2 2 3 3 ...
##  $ hum       : num  81 80 80 75 75 75 80 86 75 76 ...
##  $ windspeed : num  0 0 0 0 0 ...
##  $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
##  $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

After all the data preparation, we would like to give an overview of the data types to see how they have changed.

```
head(bike_data)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
```

```
##   temp  atemp hum windspeed casual registered cnt
## 1 3.28 3.0014  81    0.0000      3         13  16
## 2 2.34 1.9982  80    0.0000      8         32  40
## 3 2.34 1.9982  80    0.0000      5         27  32
## 4 3.28 3.0014  75    0.0000      3         10  13
## 5 3.28 3.0014  75    0.0000      0          1   1
## 6 3.28 1.0016  75    6.0032      0          1   1
```

```
summary(bike_data)
```

```
##     instant          dteday              season           yr
##  Min.   :    1   Min.   :2011-01-01   Min.   :1.000   Min.   :0.0000
##  1st Qu.: 4346   1st Qu.:2011-07-04   1st Qu.:2.000   1st Qu.:0.0000
##  Median : 8690   Median :2012-01-02   Median :3.000   Median :1.0000
##  Mean   : 8690   Mean   :2012-01-02   Mean   :2.502   Mean   :0.5026
##  3rd Qu.:13034   3rd Qu.:2012-07-02   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :17379   Max.   :2012-12-31   Max.   :4.000   Max.   :1.0000
##       mnth              hr           holiday           weekday
##  Min.   : 1.000   Min.   : 0.00   Min.   :0.00000   Min.   :0.000
##  1st Qu.: 4.000   1st Qu.: 6.00   1st Qu.:0.00000   1st Qu.:1.000
##  Median : 7.000   Median :12.00   Median :0.00000   Median :3.000
##  Mean   : 6.538   Mean   :11.55   Mean   :0.02877   Mean   :3.004
##  3rd Qu.:10.000   3rd Qu.:18.00   3rd Qu.:0.00000   3rd Qu.:5.000
##  Max.   :12.000   Max.   :23.00   Max.   :1.00000   Max.   :6.000
##    workingday       weathersit         temp           atemp
##  Min.   :0.0000   Min.   :1.000   Min.   :-7.06   Min.   :-16.000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.: 7.98   1st Qu.:  5.998
##  Median :1.0000   Median :1.000   Median :15.50   Median : 15.997
##  Mean   :0.6827   Mean   :1.425   Mean   :15.36   Mean   : 15.401
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:23.02   3rd Qu.: 24.999
##  Max.   :1.0000   Max.   :4.000   Max.   :39.00   Max.   : 50.000
##       hum           windspeed          casual         registered
##  Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.: 48.00   1st Qu.: 7.002   1st Qu.:  4.00   1st Qu.: 34.0
##  Median : 63.00   Median :12.998   Median : 17.00   Median :115.0
##  Mean   : 62.72   Mean   :12.737   Mean   : 35.68   Mean   :153.8
##  3rd Qu.: 78.00   3rd Qu.:16.998   3rd Qu.: 48.00   3rd Qu.:220.0
##  Max.   :100.00   Max.   :56.997   Max.   :367.00   Max.   :886.0
##       cnt
##  Min.   :  1.0
##  1st Qu.: 40.0
##  Median :142.0
##  Mean   :189.5
##  3rd Qu.:281.0
##  Max.   :977.0
```

Datacleaning

In a next step, we checked whether we had any missing values in the dataset. This showed that there are no missing values in the data set. We also read out the number of rows and the number of columns.

```r
anyNA(bike_data)
```

```
## [1] FALSE
```

```r
nrow(bike_data)
```

```
## [1] 17379
```

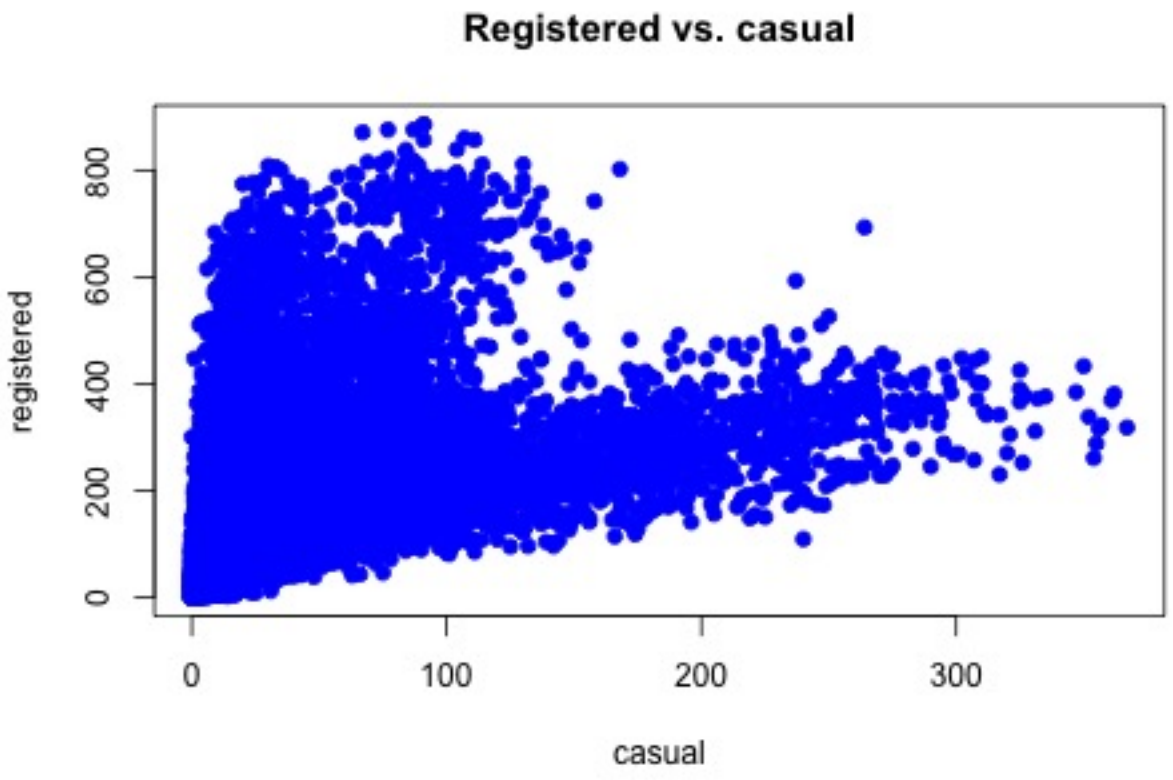
```r
ncol(bike_data)
```

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

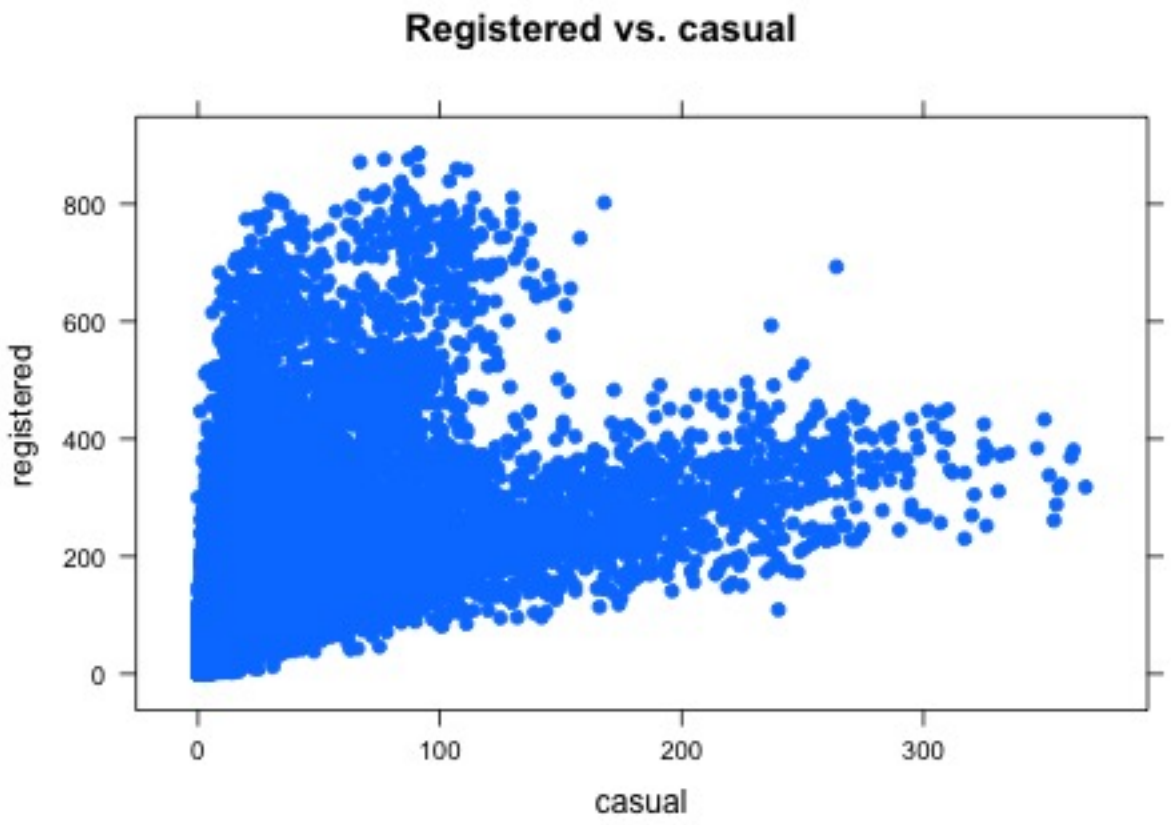
Analysis

In a first step we have plotted the same plot twice, once with the library lattice and once with ggplot. We wanted to get a feel for both libraries and then decided to use the ggplot library for the rest of the plots.

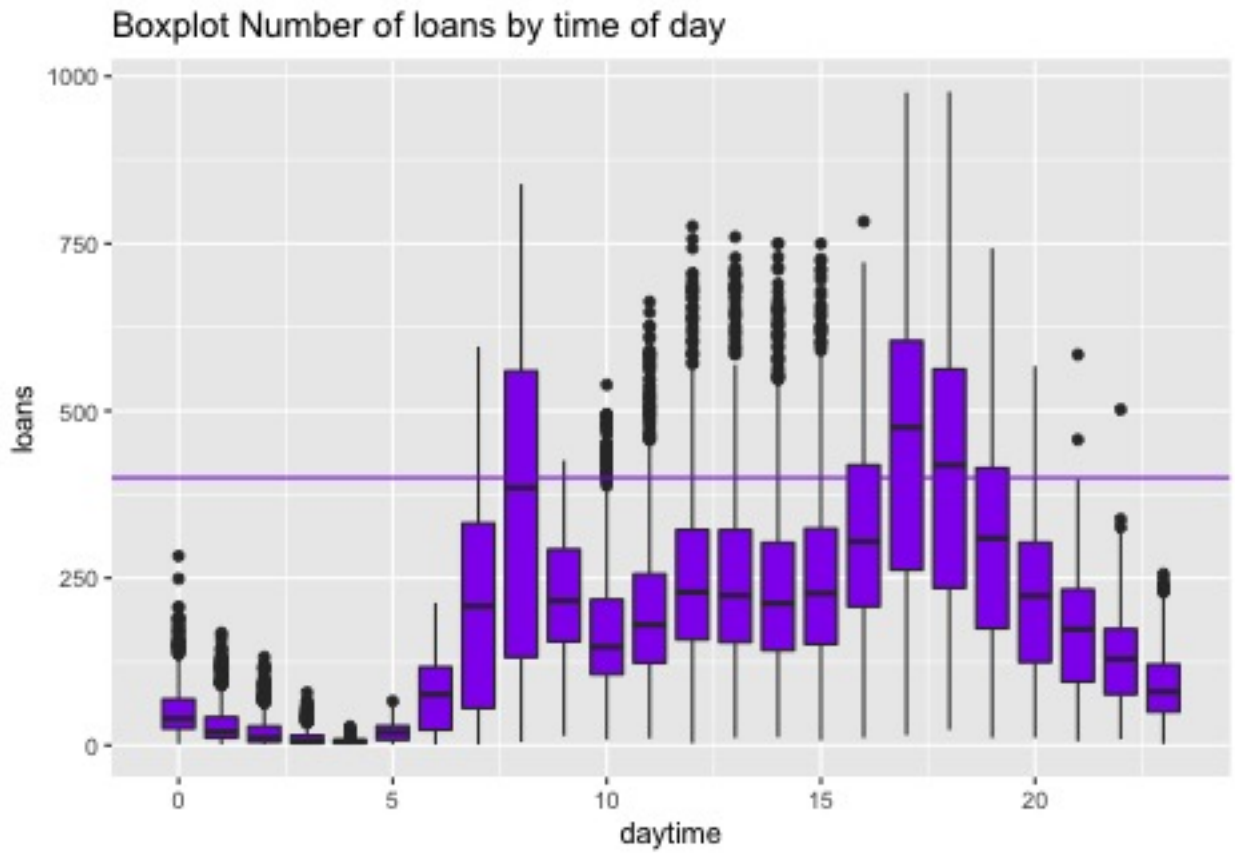
```r
#Fist analysis with plot function
plot(x= bike_data$casual, y = bike_data$registered, main = "Registered vs. casual",
     xlab = "casual",
     ylab = "registered",
     col = "blue",
     pch = 19)
```



```r
#First analysis with xyplot function from library lattice
xyplot (bike_data$registered ~ bike_data$casual, main = "Registered vs. casual",
     xlab = "casual",
     ylab = "registered",
     pch = 19)
```




```r
#Boxplot with total loans by time of day, grouped by time of day
ggplot(data = bike_data, aes(hr, cnt, group=hr)) +
       geom_boxplot(fill="purple2") +
xlab("daytime") +
ylab("loans") +
ggtitle("Boxplot Number of loans by time of day") +
geom_hline(yintercept = 400, color = "purple2", size = 0.5)
```

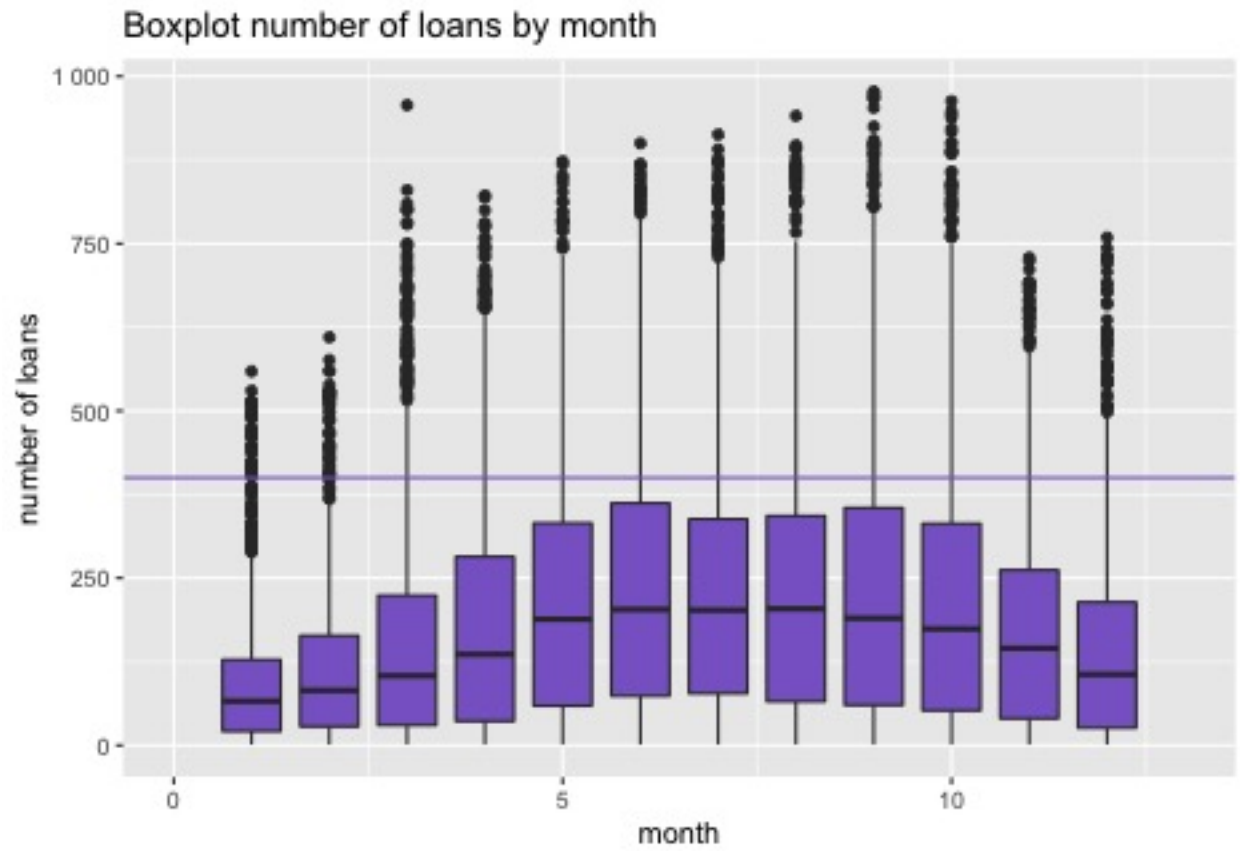


To get a overview of the data, we also created plots with the different values from the data set to see at which time of day, on which day of the week, in which weather conditions etc. the most bikes were lent out.

From this Boxplot it can be read at which times of the day on average the most bicycles are rented. It is also visible how great the dispersion is at the different times of the day.

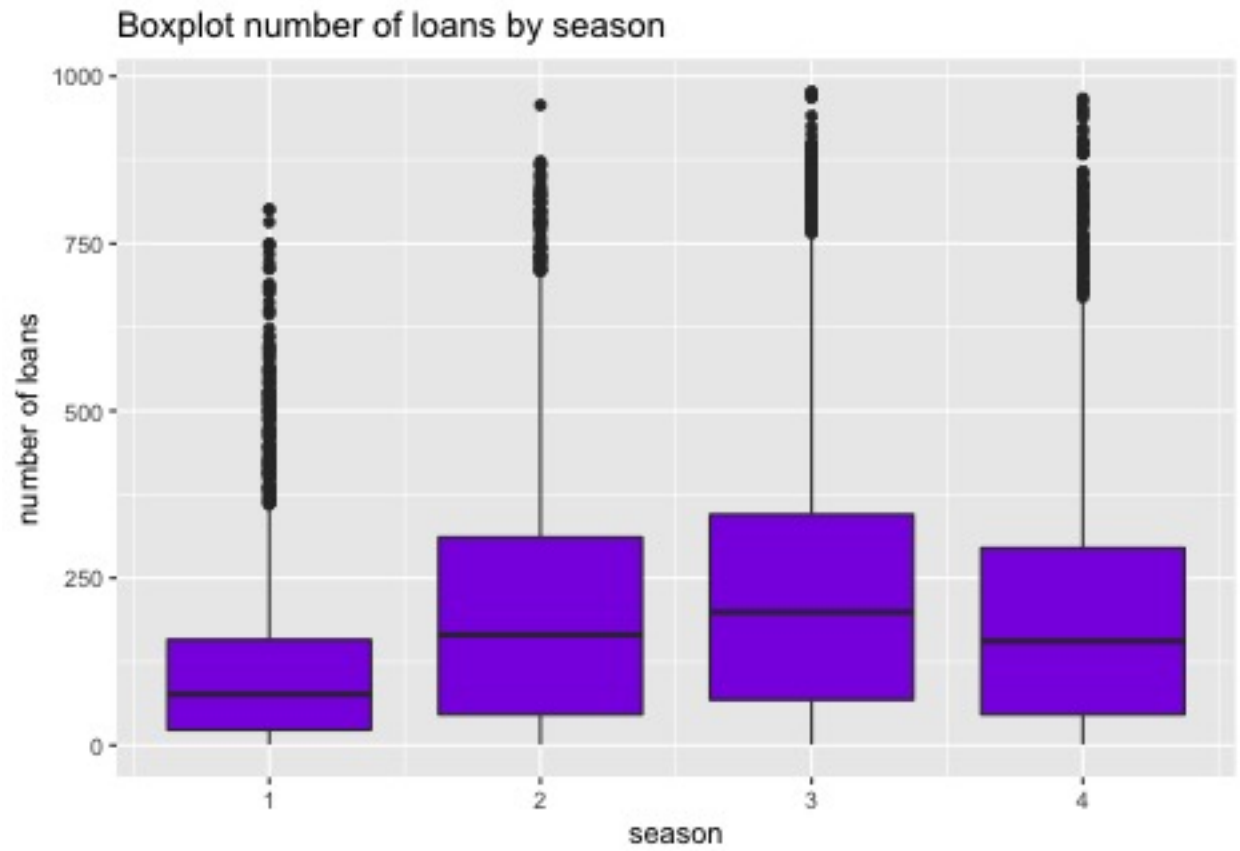
The diagram shows that most bicycles are rented in the morning between 08:00 and 09:00 and in the evening between 17:00 and 19:00.

```
ggplot(data = bike_data, aes(mnth, cnt, group=mnth)) +
       geom_boxplot(fill="mediumpurple3") +
xlab("month") +
ylab("number of loans") +
coord_cartesian(xlim = c(0, 13)) +
scale_y_continuous(labels = number_format()) +
ggtitle("Boxplot number of loans by month") +
geom_hline(yintercept = 400, color = "mediumpurple3", size = 0.5)
```



In this chart, it can be seen that the summer months between May and October have the highest average number of bicycle rentals. The purple line is again at 400 rentals. In no month is the mean or the third quartile at 400 rentals, this is probably due to the fact that there are usually days with bad weather in each month when fewer rentals are made and therefore the monthly mean is lower.

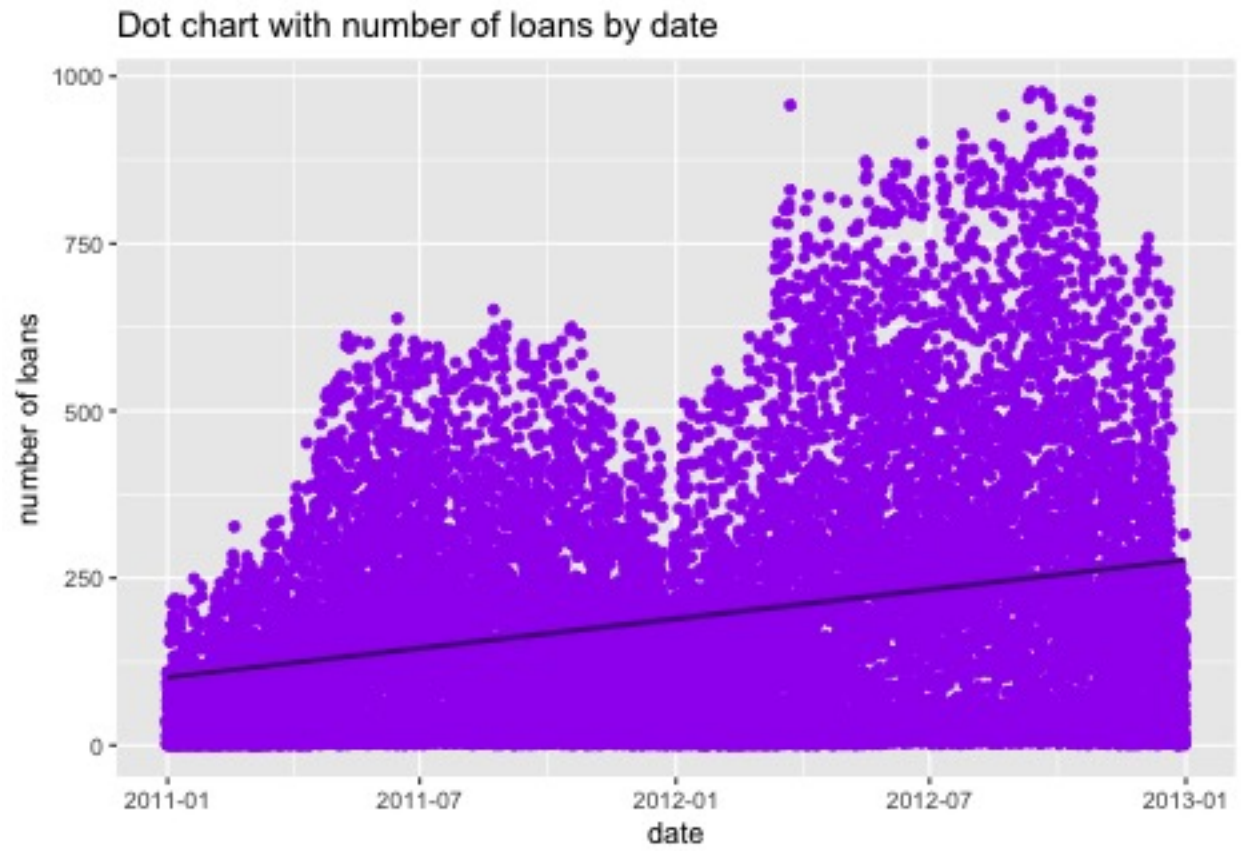
```
#Boxplot with total loans and season, grouped by season
ggplot(data = bike_data, aes(season, cnt, group=season)) +
       geom_boxplot(fill="blueviolet") +
xlab("season") +
ylab("number of loans") +
ggtitle("Boxplot number of loans by season")
```



From this boxplot we can see that the median is largest in the third season, summer, followed by spring, autumn and finally winter. After seeing the evaluation by month, we are no longer surprised by this result.

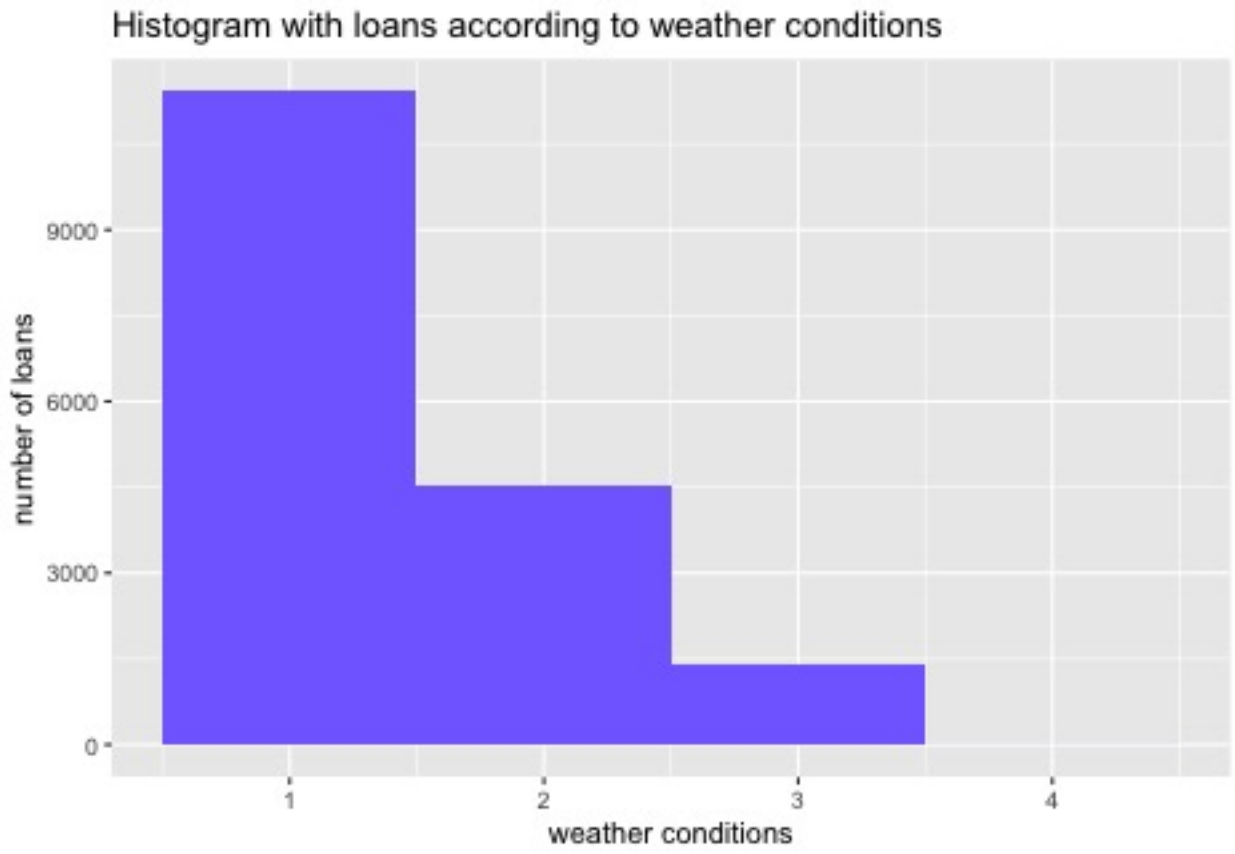
The spread between the lower and upper quartiles is also greatest in summer.

```
#Dot chart with number of loans by date
ggplot(data = bike_data, aes(dteday, cnt)) +
       geom_point(color="purple") +
       geom_smooth(method = "lm", color = "purple4") +
xlab("date") +
ylab("number of loans") +
ggtitle("Dot chart with number of loans by date")
```



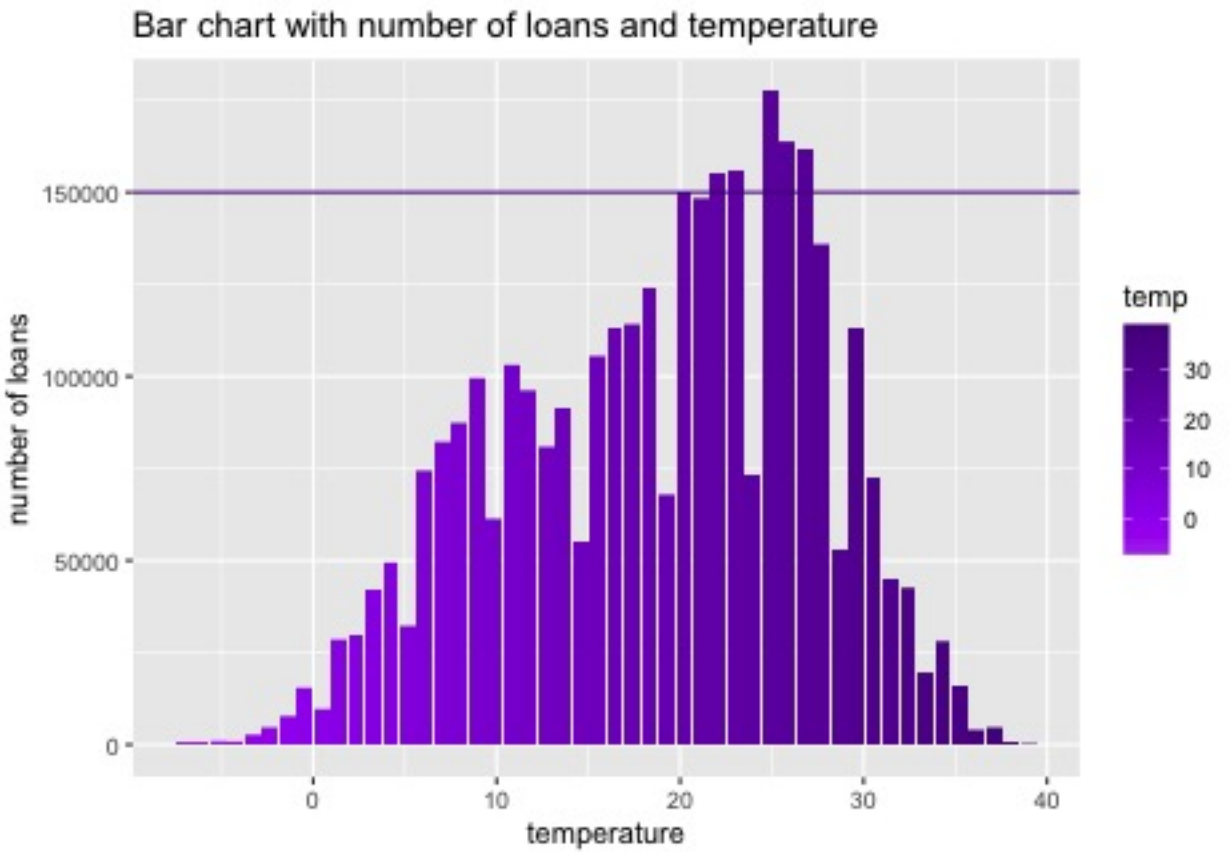
In the chart above you can see the number of loans by date. As can be seen, in the first year, 2011, fewer bicycles were borrowed overall than in 2012. The line shows the trend. The trend is upwards. This means that, according to the trend, more bicycles are borrowed each year.

```
#Histogram with number of loans according to weather conditions
ggplot(data = bike_data, aes(weathersit)) +
       geom_histogram(fill="slateblue1", binwidth=1) +
xlab("weather conditions") +
ylab("number of loans")+
ggtitle("Histogram with loans according to weather conditions")
```



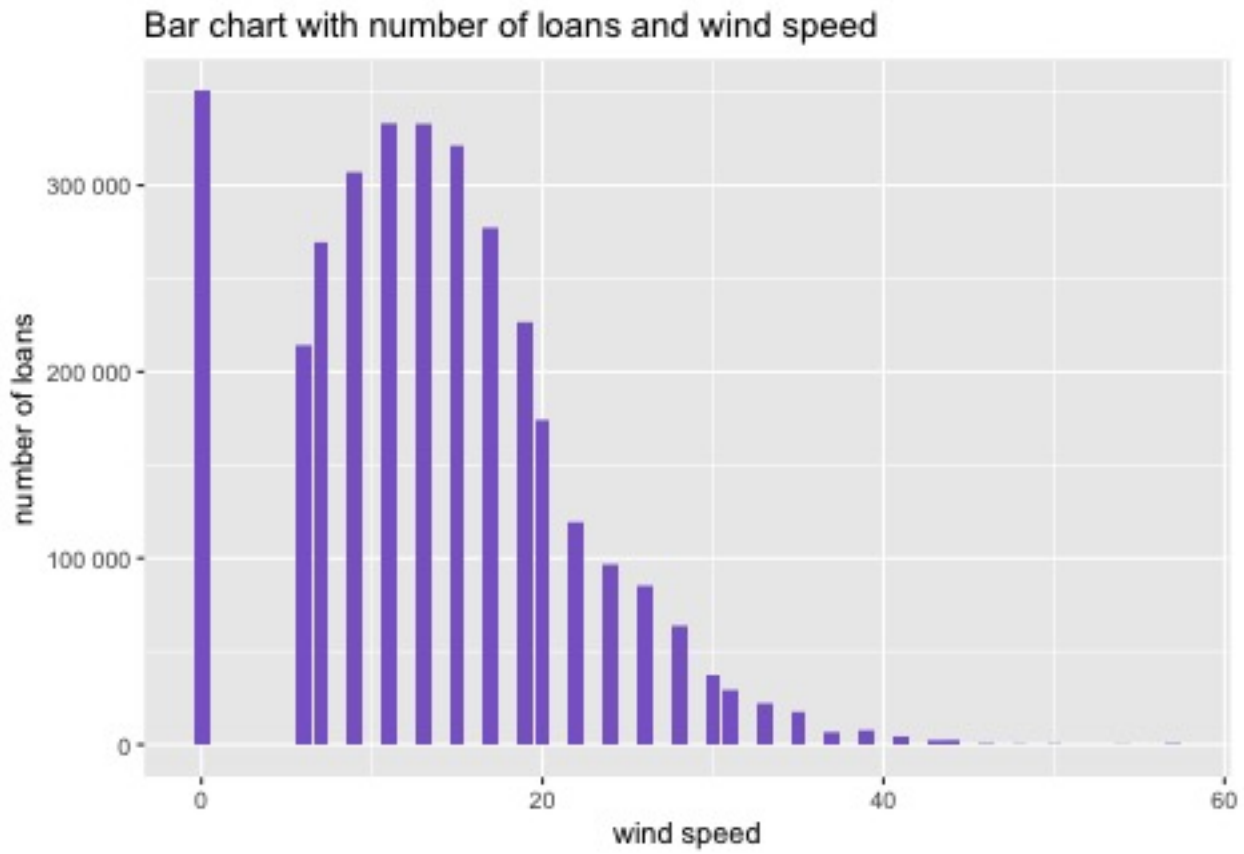
The graph above shows that by far the most bicycles are rented in good weather and in very bad weather, at number 4, so few bicycles are rented that this is not visible on the graph.

```
#Bar chart with number of loans and temperature
ggplot(data = bike_data, aes(x=temp, y=cnt, fill=temp)) +
       geom_col() +
 scale_fill_gradient2(low="plum2", high="purple4", mid="purple") +
xlab("temperature") +
ylab("number of loans") +
geom_hline(yintercept = 150000, color = "purple4", size = 0.5) +
ggtitle("Bar chart with number of loans and temperature")
```



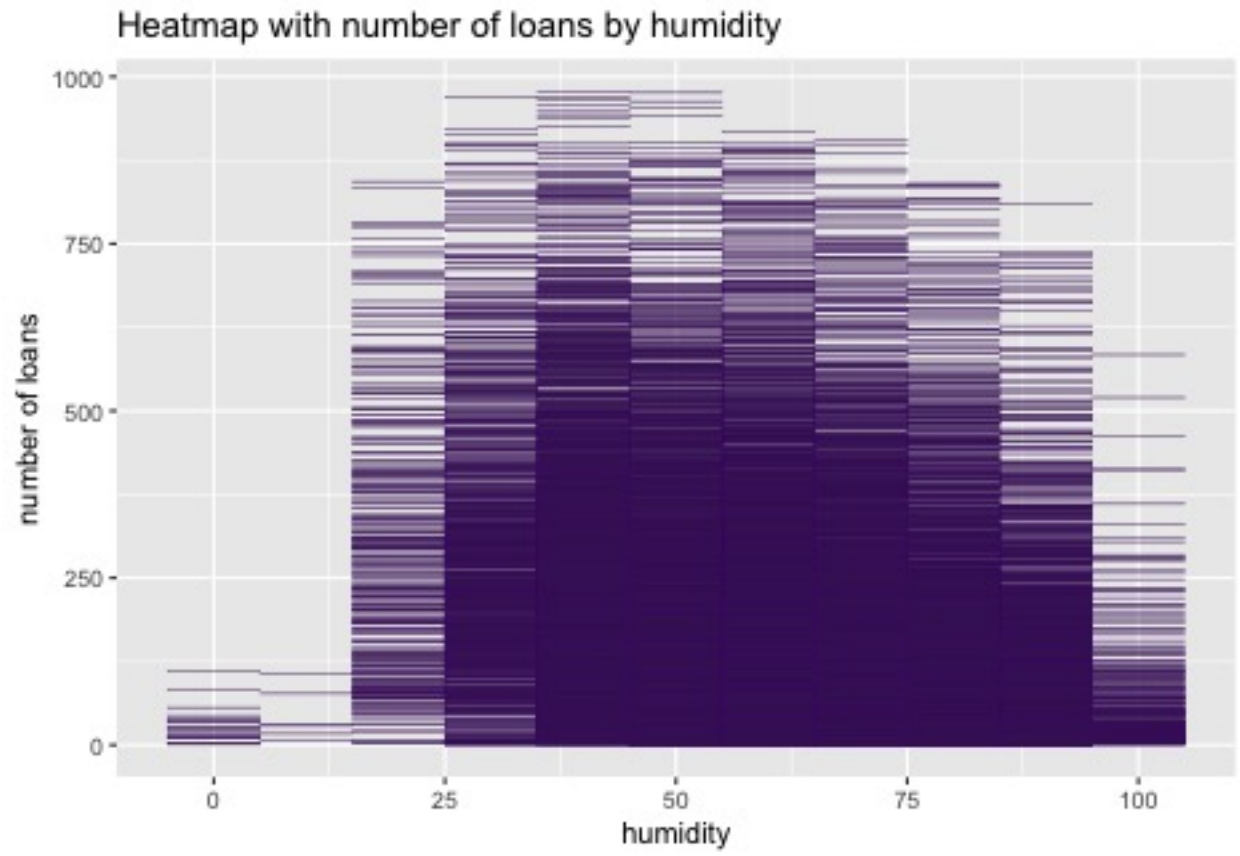
In the diagram above, it can be seen that the number of rentals is highest between 20 and 27 degrees. At temperatures below 5 and above 31 degrees Celsius, the proportion of borrowed bicycles is very small.

```
#Bar chart with number of loans and wind speed
ggplot(data = bike_data, aes(x=windspeed, y=cnt)) +
       geom_col(fill="mediumpurple3") +
xlab("wind speed") +
ylab("number of loans") +
scale_y_continuous(labels = number_format()) +
ggtitle("Bar chart with number of loans and wind speed")
```



In the bar chart above, it is clear that the higher the wind speed, the fewer bikes are rented.

```
#Heatmap mit Anzahl Ausleihen und Luftfeuchtigkeit
#Round humidity to the 10ths and create a new variable so that the graphic is clearer.
bike_data$humR <- round(bike_data$hum, digits=-1)
ggplot(data = bike_data, aes(x=humR, y=cnt)) +
       geom_tile(color="purple4") +
xlab("humidity") +
ylab("number of loans") +
ggtitle("Heatmap with number of loans by humidity")
```

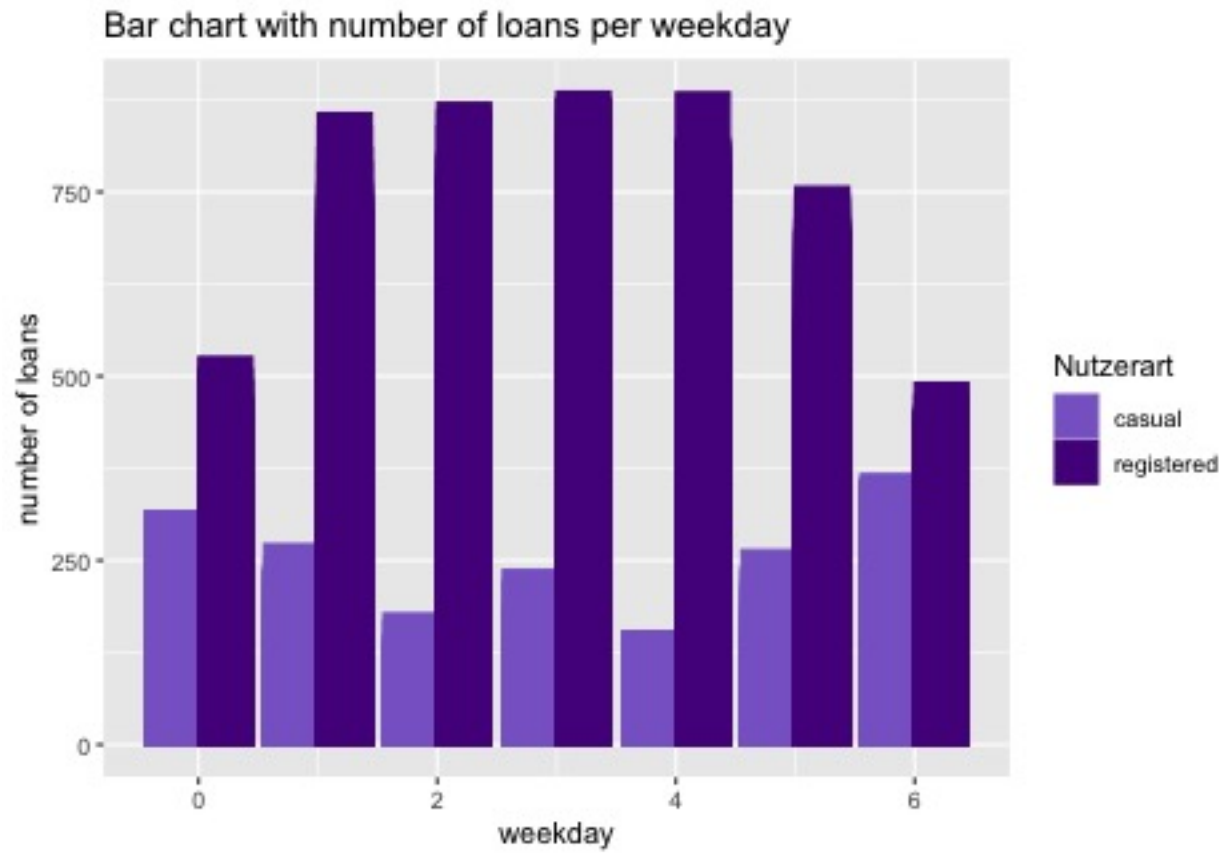



```
#Remove column with rounded values again
bike_data <- bike_data %>%
    select(-contains("humR"))
```

In this graph, it can be seen that when the humidity is around 25-80, the most bicycles are rented. It can also be seen that not many bicycles are rented at low humidity and at very high humidity.

For better readability, the continuous values of humidity have been rounded to the nearest 10.

```
#casual and registered together
bike_data_Day <- bike_data %>% gather(
  `casual`, `registered`, key = "Nutzerart", value = "Wert")
#Bar chart with number of loans registered and not registered and day of the week
ggplot(data=bike_data_Day, aes(x=weekday, y=Wert, color=Nutzerart, fill=Nutzerart)) +
       geom_col(position="dodge") +
xlab("weekday") +
ylab("number of loans") +
scale_fill_manual(values = c("mediumpurple3", "purple4")) +
scale_color_manual(values = c("mediumpurple3", "purple4")) +
ggtitle("Bar chart with number of loans per weekday")
```



First, a note on the above diagram. The week starts at 0 and day 0 is Sunday and day 6 is Saturday.

In the above diagram it can be seen that many more registered users will rent bicycles than non-registered users. It can also be seen that more bicycles are rented by non-registered users on weekends. It can be assumed that more people spontaneously decide to rent a bike at the weekend than during the week. On weekdays, it is the other way round, with more registered users renting a bike.

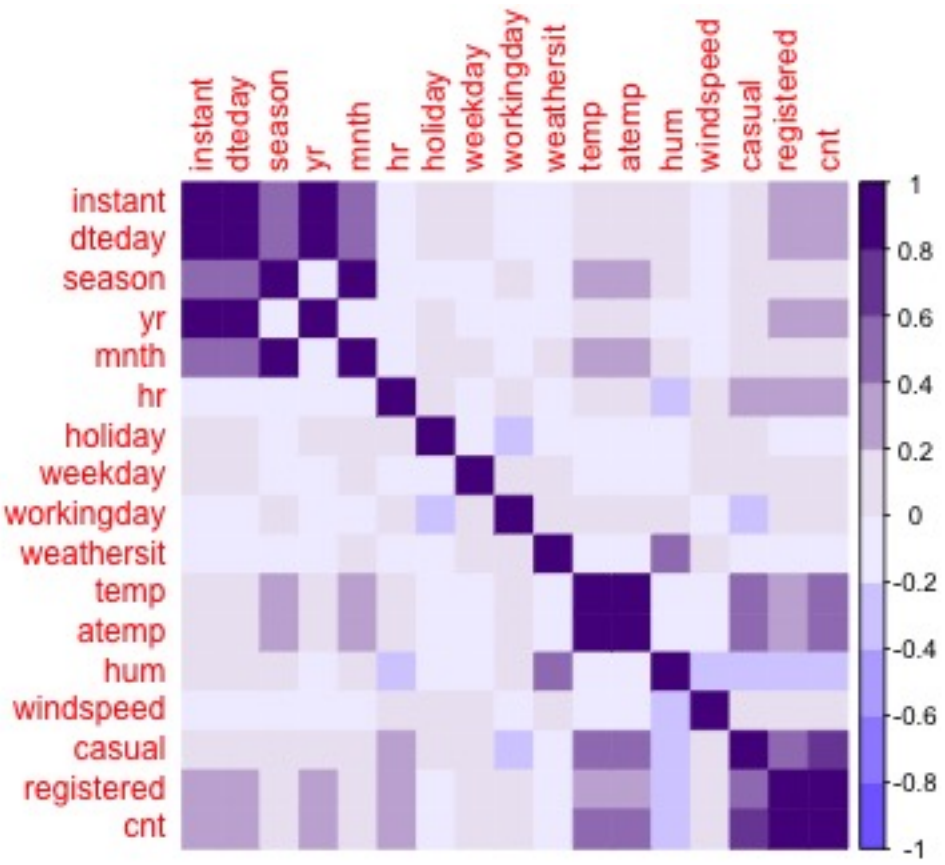
Since more registered users rent bikes during the week, who are also more likely to be regular customers than the non-registered ones, the probability during the week can probably be predicted more accurately.

Chapter of choice

We want to show the correlation of the individual variables with a correlation matrix. With the help of the density, you can easily see which variables are related and to what extent.

In addition to the packages shown in the lessons, we have chosen the package corrplot to show the coorelation. The package is a CRAN package and more information can be found under the following link.

```
#Convert column dteday for correlation matrix to numeric
bike_data$dteday <- as.numeric(bike_data$dteday)
#Correlation matrix
corrplot((cor(bike_data)), method="color",
         col= colorRampPalette(c("slateblue1","white", "purple4"))(10))
```



Correlation tells how strong the linear relationship is between two variables. This correlation can be either positive or negative. The value of the correlation is always between -1 and 1. In the graph, the positive correlation is shown in purple and the negative correlation in blue. The darker the fields are coloured, the stronger the correlation.

In the top left corner there are several dark squares. However, since the correlation is not suitable for finding correlations between two categorical variables, we will disregard these correlations.

Otherwise, we have found that there are no very strong correlations. The strongest ones are at most 0.4 to 0.6, which are in the correlation between temperature and unregistered and total loans. We attribute this result to the fact that there is no strong linear correlation, since people rent more bicycles, for example, the nicer and warmer the weather is. However, if the weather is hotter than 27°C, as we have seen above, then it is too hot for people to ride bicycles and they rent fewer bicycles again.

Since none of the values are strongly correlated with each other, a multiple linear regression could be suitable for the forecast model, since too strong correlations would threaten numerical instability and the model would be difficult to interpret.

Modelling

We also tried to model the data and worked out a polynomial regression.

Predicting the rentals on a specific day is too specific and not what the client wants. It is better to predict how many bikes will be rented on a nice Saturday in June with 25°C and no wind. To implement this, we first had to convert the values with data type int into numeric so that we could group the data set. We grouped the dataset by year, month, day of the week and hour. This grouping was chosen deliberately because, as we saw in the descriptive analytics, the number of loans varies greatly depending on the month, day of the week and hour.

```r
#Convert data types from int to numeric to later convert these values into factors
bike_data$season <- as.numeric(bike_data$season)
bike_data$yr <- as.numeric(bike_data$yr)
bike_data$mnth <- as.numeric(bike_data$mnth)
bike_data$hr <- as.numeric(bike_data$hr)
bike_data$weekday <- as.numeric(bike_data$weekday)
bike_data$weathersit <- as.numeric(bike_data$weathersit)

#Group data set by year, month, weekday and hour
bike_data_grouped <- bike_data %>% group_by(
  yr, mnth, weekday, hr) %>% summarise(
    season = mean(season),
    weathersit = mean(weathersit),
    temp = mean(temp),
    atemp = mean(atemp),
    hum = mean(hum),
    windspeed = mean(windspeed),
    cnt = mean(cnt))

#Round discrete values to whole numbers.
#(Decimal places have been created by grouping and calculating the mean value)
bike_data_grouped$season <- round(bike_data_grouped$season, digits=0)
bike_data_grouped$weathersit <- round(bike_data_grouped$weathersit, digits=0)

#Convert categorical variables into factor
bike_data_grouped$season <-as.factor(bike_data_grouped$season)
bike_data_grouped$yr <-as.factor(bike_data_grouped$yr)
bike_data_grouped$mnth <-as.factor(bike_data_grouped$mnth)
bike_data_grouped$hr <-as.factor(bike_data_grouped$hr)
bike_data_grouped$weekday <-as.factor(bike_data_grouped$weekday)
bike_data_grouped$weathersit <-as.factor(bike_data_grouped$weathersit)

#Splitting the data into training and test data
set.seed(101)
sample <- sample.int(n = nrow(bike_data_grouped),
                     size = floor(0.8 * nrow(bike_data_grouped)), replace = F)
#Create training dataset
bike_train <- bike_data_grouped[ sample, ]
#Create test data set
bike_test <- bike_data_grouped[ -sample, ]
```

Since certain values do not run linearly, as seen above in the descriptive analytics, we also tried polynomial regression. For values such as temperature and humidity, the number of rentals is not linear. Users rent more bikes the warmer it is. However, if it is warmer than 27°C, then it is too warm for people to ride a bike and they rent fewer bikes again. We then refined the model accordingly and were finally able to create a prediction model that predicts with a 78% probability how many bicycles will be rented at a given time.

```r
#Prediction model (polynomial regression)
pr_bike <- lm(cnt ~  poly(temp, 3, raw=TRUE) + poly(atemp, 3, raw=TRUE) +
                poly(hum, 3, raw=TRUE) + windspeed +
              yr + season + mnth + hr + weekday + weathersit + (temp * hr),
              data=bike_train)
summary(pr_bike)
```

```
##
## Call:
## lm(formula = cnt ~ poly(temp, 3, raw = TRUE) + poly(atemp, 3,
##     raw = TRUE) + poly(hum, 3, raw = TRUE) + windspeed + yr +
##     season + mnth + hr + weekday + weathersit + (temp * hr),
##     data = bike_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -366.16  -46.16   -3.31   39.65  337.27
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.423e+02  9.409e+01  -1.512 0.130647
## poly(temp, 3, raw = TRUE)1  -1.129e+01  8.531e+00  -1.323 0.185982
## poly(temp, 3, raw = TRUE)2   1.435e-01  5.773e-01   0.249 0.803651
## poly(temp, 3, raw = TRUE)3  -5.293e-03  1.164e-02  -0.455 0.649274
## poly(atemp, 3, raw = TRUE)1  3.557e+00  4.404e+00   0.808 0.419291
## poly(atemp, 3, raw = TRUE)2  5.296e-01  2.895e-01   1.829 0.067474 .
## poly(atemp, 3, raw = TRUE)3 -1.227e-02  5.864e-03  -2.092 0.036543 *
## poly(hum, 3, raw = TRUE)1    1.269e+01  4.681e+00   2.710 0.006765 **
## poly(hum, 3, raw = TRUE)2   -2.737e-01  7.900e-02  -3.464 0.000538 ***
## poly(hum, 3, raw = TRUE)3    1.682e-03  4.333e-04   3.881 0.000106 ***
## windspeed                   -9.419e-02  4.492e-01  -0.210 0.833952
## yr1                          8.664e+01  3.124e+00  27.738  < 2e-16 ***
## season2                     -7.909e+00  1.387e+01  -0.570 0.568686
## season3                     -1.587e+01  2.058e+01  -0.771 0.440797
## season4                     -2.693e+01  5.179e+01  -0.520 0.603030
## mnth2                        5.385e+00  7.943e+00   0.678 0.497805
## mnth3                        3.508e+01  1.007e+01   3.483 0.000502 ***
## mnth4                        4.408e+01  1.792e+01   2.460 0.013951 *
## mnth5                        5.473e+01  2.001e+01   2.735 0.006269 **
## mnth6                        6.220e+01  2.221e+01   2.801 0.005131 **
## mnth7                        7.883e+01  3.018e+01   2.612 0.009057 **
## mnth8                        7.863e+01  2.808e+01   2.801 0.005133 **
## mnth9                        8.095e+01  2.582e+01   3.135 0.001733 **
## mnth10                       1.106e+02  5.341e+01   2.070 0.038535 *
## mnth11                       9.467e+01  5.285e+01   1.791 0.073352 .
## mnth12                       7.315e+01  2.119e+01   3.452 0.000564 ***
## hr1                         -7.032e+00  2.033e+01  -0.346 0.729502
## hr2                         -4.533e+00  1.982e+01  -0.229 0.819141
## hr3                         -2.073e+01  2.050e+01  -1.011 0.312112
## hr4                         -8.995e+00  1.995e+01  -0.451 0.652060
## hr5                         -9.514e+00  1.929e+01  -0.493 0.621974
## hr6                          1.404e+01  1.934e+01   0.726 0.468046
## hr7                          8.755e+01  1.984e+01   4.413 1.06e-05 ***
## hr8                          2.131e+02  1.964e+01  10.849  < 2e-16 ***
## hr9                          1.243e+02  2.054e+01   6.052 1.60e-09 ***
## hr10                         6.950e+01  2.019e+01   3.442 0.000585 ***
## hr11                         7.598e+01  2.090e+01   3.634 0.000283 ***
## hr12                         9.476e+01  2.208e+01   4.293 1.82e-05 ***
## hr13                         8.822e+01  2.304e+01   3.829 0.000131 ***
## hr14                         6.541e+01  2.351e+01   2.783 0.005426 **
## hr15                         7.036e+01  2.396e+01   2.936 0.003349 **
```

```
## hr16                         8.423e+01  2.409e+01   3.496 0.000478 ***
## hr17                         1.179e+02  2.387e+01   4.939 8.25e-07 ***
## hr18                         1.113e+02  2.246e+01   4.956 7.57e-07 ***
## hr19                         5.651e+01  2.181e+01   2.591 0.009608 **
## hr20                         2.984e+01  2.209e+01   1.351 0.176831
## hr21                         1.915e+01  2.115e+01   0.905 0.365351
## hr22                         1.984e+01  2.016e+01   0.984 0.325143
## hr23                         7.827e+00  2.044e+01   0.383 0.701756
## weekday1                    -4.499e-01  5.447e+00  -0.083 0.934172
## weekday2                     3.347e+00  5.487e+00   0.610 0.541951
## weekday3                     6.465e+00  5.473e+00   1.181 0.237565
## weekday4                     8.021e+00  5.405e+00   1.484 0.137901
## weekday5                     1.826e+01  5.550e+00   3.290 0.001012 **
## weekday6                     1.464e+01  5.431e+00   2.696 0.007057 **
## weathersit2                 -1.073e+01  3.271e+00  -3.280 0.001050 **
## weathersit3                 -9.000e+01  8.209e+01  -1.096 0.273038
## temp                               NA         NA      NA       NA
## hr1:temp                    -6.741e-01  1.297e+00  -0.520 0.603338
## hr2:temp                    -1.770e+00  1.282e+00  -1.381 0.167323
## hr3:temp                    -1.877e+00  1.308e+00  -1.435 0.151377
## hr4:temp                    -2.756e+00  1.296e+00  -2.126 0.033582 *
## hr5:temp                    -1.775e+00  1.284e+00  -1.382 0.167217
## hr6:temp                     9.394e-01  1.285e+00   0.731 0.464878
## hr7:temp                     6.043e+00  1.288e+00   4.690 2.84e-06 ***
## hr8:temp                     7.236e+00  1.243e+00   5.820 6.46e-09 ***
## hr9:temp                     3.097e+00  1.243e+00   2.491 0.012801 *
## hr10:temp                    3.231e+00  1.236e+00   2.615 0.008968 **
## hr11:temp                    4.222e+00  1.261e+00   3.349 0.000820 ***
## hr12:temp                    5.435e+00  1.302e+00   4.175 3.06e-05 ***
## hr13:temp                    5.768e+00  1.336e+00   4.317 1.63e-05 ***
## hr14:temp                    6.286e+00  1.356e+00   4.637 3.68e-06 ***
## hr15:temp                    6.293e+00  1.381e+00   4.557 5.39e-06 ***
## hr16:temp                    8.846e+00  1.365e+00   6.480 1.06e-10 ***
## hr17:temp                    1.578e+01  1.366e+00  11.549  < 2e-16 ***
## hr18:temp                    1.441e+01  1.309e+00  11.011  < 2e-16 ***
## hr19:temp                    1.175e+01  1.280e+00   9.178  < 2e-16 ***
## hr20:temp                    8.513e+00  1.308e+00   6.510 8.70e-11 ***
## hr21:temp                    6.214e+00  1.271e+00   4.887 1.07e-06 ***
## hr22:temp                    3.683e+00  1.244e+00   2.962 0.003081 **
## hr23:temp                    1.784e+00  1.256e+00   1.420 0.155844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.24 on 3144 degrees of freedom
## Multiple R-squared:  0.7859, Adjusted R-squared:  0.7805
## F-statistic: 146.1 on 79 and 3144 DF,  p-value: < 2.2e-16
```

Quality measurement

```r
#Calculation of residuals of the training data
bike_train$cnt_pr_tr <- predict(pr_bike, bike_train)
bike_train$cnt_resid_tr <- bike_train$cnt - bike_train$cnt_pr_tr
```

```r
summary(bike_train$cnt_resid_tr)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -366.164  -46.162   -3.311    0.000   39.648  337.269
```

```r
#Calculation of the residual standard error of the training data
sqrt(sum(bike_train$cnt_resid_tr^2)/(nrow(bike_train)-2))
```

```
## [1] 80.24764
```

```r
#Calculation of residuals of the test data
bike_test$cnt_pr_ts <- predict(pr_bike, bike_test)
bike_test$cnt_resid_ts <- bike_test$cnt - bike_test$cnt_pr_ts
summary(bike_test$cnt_resid_ts)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -375.225  -52.929   -5.477   -4.501   35.863  296.016
```

```r
#Calculation of the residual standard error of the test data
sqrt(sum(bike_test$cnt_resid_ts^2)/(nrow(bike_test)-2))
```

```
## [1] 81.47584
```

Interpretation of the quality measurement

The scatter of the test data is somewhat smaller than that of the training data. The mean deviates by -4.501 for the test data set. The standard error of 81.47584 for the test data is slightly higher than that of 80.24764 for the training data.

Other Functions

Here we want to try other functions we have learned in the R-Bootcamp.

Matrix

We have created a matrix with 5 rows and 5 columns.

```r
#Matrix
matrix(data = bike_data$windspeed, ncol = 9, nrow = 9)
```

```
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]   [,8]    [,9]
##  [1,] 0.0000  0.0000 16.9979 12.9980  8.9981 19.9995 26.0027 7.0015  6.0032
##  [2,] 0.0000 16.9979 16.9979 12.9980 12.9980 11.0014 16.9979 0.0000  7.0015
##  [3,] 0.0000 19.0012 16.9979 19.9995 11.0014 23.9994 22.0028 7.0015  7.0015
##  [4,] 0.0000 19.0012 12.9980 12.9980 11.0014 27.9993 19.9995 8.9981  8.9981
##  [5,] 0.0000 19.9995 15.0013 15.0013 12.9980 26.0027 19.0012 8.9981 11.0014
##  [6,] 6.0032 19.0012 19.9995 15.0013 22.0028 19.0012 19.0012 7.0015 15.0013
##  [7,] 0.0000 19.9995 19.9995 15.0013 30.0026 26.0027 16.9979 7.0015 22.0028
##  [8,] 0.0000 19.9995 16.9979 16.9979 23.9994 12.9980 16.9979 7.0015 19.9995
##  [9,] 0.0000 19.0012 19.0012 19.9995 22.0028 19.0012 15.0013 8.9981 11.0014
```

Data Frame

In addition, we have created a dataframe from the dataset, which contains the variables temp, casual, registered and cnt.

```
#Data Frame
d.bike_data <- data.frame(bike_data$temp, bike_data$casual, bike_data$registered, bike_data$cnt)
head(d.bike_data)
```

```
##   bike_data.temp bike_data.casual bike_data.registered bike_data.cnt
## 1           3.28                3                   13            16
## 2           2.34                8                   32            40
## 3           2.34                5                   27            32
## 4           3.28                3                   10            13
## 5           3.28                0                    1             1
## 6           3.28                0                    1             1
```

Conclusion

Our analyses have shown that the number of bicycle rentals depends on various factors. More bicycles are rented in good weather than in bad weather. In summer, the weather is often better and it is warmer, which is why most bicycles are rented in the summer months. Demand is also higher in the morning and evening rush hours than during midday. In the middle of the night, the demand for bicycles is very low. There are also differences in the days of the week. More bicycles are rented during the week than at weekends.

It was also noted that more bicycles were rented in 2012 than in 2011. There is a clear upward trend. It is assumed that the bike sharing service was only established in 2011 or the trend for bike rental is increasing. It is assumed that the number of rentals will continue to increase in the coming years.

With the forecast model created, it is possible to predict with a certainty of 78% when and how many bicycles will be rented. The forecast is month-, weekday- and hour-specific.