Zurich Bike Traffic

Mike Krähenbühl

2024-04-22

Data loading

```
# article qualitatively describing the counting concept
# https://www.stadt-zuerich.ch/ted/de/index/taz/verkehr/webartikel/webartikel_velozaehlungen.html

# raw dataset
#https://data.stadt-zuerich.ch/dataset/ted_taz_verkehrszaehlungen_werte_fussgaenger_velo


# load the data for the individual years
data2020 <- read.csv('../../results/df_agg_hourly_2020.csv', header = TRUE)
data2021 <- read.csv('../../results/df_agg_hourly_2021.csv', header = TRUE)
data2022 <- read.csv('../../results/df_agg_hourly_2022.csv', header = TRUE)
data2023 <- read.csv('../../results/df_agg_hourly_2023.csv', header = TRUE)


# combine the years to one dataset
data <- rbind(data2020, data2021, data2022, data2023)

# Create Month column
data$Month <- as.factor(format(as.Date(data$Date), "%m"))

# Create Day column
data$Day <- as.factor(format(as.Date(data$Date), "%d"))

# Create a Weekday column
data$Weekday <- as.factor(weekdays(as.Date(data$Date)))
```

Check structure of dataset

```
head(data)
```

```
##   Standort       Date  Time         Datetime Hr...Hr. RainDur..min.
## 1       20 2020-01-01 00:00 2020-01-01 00:00 90.45667             0
## 2       20 2020-01-01 01:00 2020-01-01 01:00 90.09333             0
## 3       20 2020-01-01 02:00 2020-01-01 02:00 90.52333             0
## 4       20 2020-01-01 03:00 2020-01-01 03:00 91.29000             0
## 5       20 2020-01-01 04:00 2020-01-01 04:00 92.14667             0
## 6       20 2020-01-01 05:00 2020-01-01 05:00 93.05000             0
##   StrGlo..W.m2.     T...C.    WD.... WVs..m.s. WVv..m.s.  p..hPa. Year
```

```
## 1          0.03 -0.2366667 160.48000 0.9966667 0.6700000 982.9467 2020
## 2          0.03 -0.4166667  58.28667 0.8266667 0.5266667 982.4667 2020
## 3          0.03 -0.5966667 167.85000 1.0733333 0.7766667 982.1833 2020
## 4          0.02 -0.8266667 159.85000 1.3733333 1.1700000 981.8133 2020
## 5          0.02 -0.8500000  58.51000 0.9966667 0.6766667 981.8300 2020
## 6          0.02 -0.9133333 169.95333 0.6833333 0.2933333 981.7800 2020
##   AnzBestWir   bezeichnung bike_tot ped_tot Month Day  Weekday
## 1     434736 Militärbrücke        0      61    01  01 Mittwoch
## 2     434736 Militärbrücke        0     135    01  01 Mittwoch
## 3     434736 Militärbrücke        0     114    01  01 Mittwoch
## 4     434736 Militärbrücke        0      38    01  01 Mittwoch
## 5     434736 Militärbrücke        0      37    01  01 Mittwoch
## 6     434736 Militärbrücke        0      36    01  01 Mittwoch
```

```
str(data)
```

```
## 'data.frame':    729483 obs. of  20 variables:
##  $ Standort     : int  20 20 20 20 20 20 20 20 20 20 ...
##  $ Date         : chr  "2020-01-01" "2020-01-01" "2020-01-01" "2020-01-01" ...
##  $ Time         : chr  "00:00" "01:00" "02:00" "03:00" ...
##  $ Datetime     : chr  "2020-01-01 00:00" "2020-01-01 01:00" "2020-01-01 02:00" "2020-01-01 03:00" .
##  $ Hr...Hr.     : num  90.5 90.1 90.5 91.3 92.1 ...
##  $ RainDur..min.: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ StrGlo..W.m2.: num  0.03 0.03 0.03 0.02 0.02 ...
##  $ T...C.       : num  -0.237 -0.417 -0.597 -0.827 -0.85 ...
##  $ WD....       : num  160.5 58.3 167.8 159.8 58.5 ...
##  $ WVs..m.s.    : num  0.997 0.827 1.073 1.373 0.997 ...
##  $ WVv..m.s.    : num  0.67 0.527 0.777 1.17 0.677 ...
##  $ p..hPa.      : num  983 982 982 982 982 ...
##  $ Year         : num  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ AnzBestWir   : num  434736 434736 434736 434736 434736 ...
##  $ bezeichnung  : chr  "Militärbrücke" "Militärbrücke" "Militärbrücke" "Militärbrücke" ...
##  $ bike_tot     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ped_tot      : num  61 135 114 38 37 36 28 11 8 26 ...
##  $ Month        : Factor w/ 12 levels "01","02","03",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Day          : Factor w/ 31 levels "01","02","03",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Weekday      : Factor w/ 7 levels "Dienstag","Donnerstag",..: 4 4 4 4 4 4 4 4 4 4 ...
```

Select only one station for a new dataset: Langstrasse (Unterführung Nord)

```
# check measurement locations
unique(data$bezeichnung)
```

```
##  [1] "Militärbrücke"                       "Ohmstrasse Ost"
##  [3] "Hardbrücke Nord (Seite Altstetten)" "Hardbrücke Süd (Seite HB)"
##  [5] "Kloster-Fahr-Weg (Europabrücke)"    "Mythenquai Unterführung"
##  [7] "Langstrasse (Unterführung Nord)"    "Limmatquai --> Bellevue"
##  [9] "Andreasstrasse"                      "Talstrasse"
## [11] "Scheuchzerstrasse"                   "Katzenbach"
## [13] "Lux-Guyer-Weg"                       "Weinbergfussweg"
## [15] "Baslerstrasse"                       "Limmatquai"
## [17] "Bucheggplatz"                        "Cassiopeiasteg"
## [19] "Langstrasse (Fahrbahn Süd)"          "Langstrasse (Fahrbahn Nord)"
```

```
## [21] "Ohmstrasse West"                    "Dammweg"
## [23] "Langstrasse (Unterführung Süd)"     "Tödistrasse"
## [25] "Binzmühlestrasse"                   "Mythenquai"
## [27] "Letten / Dynamo"                    "Hardeggsteg"
## [29] "Hofwiesenstrasse"                   "Sihlpromenade"
## [31] "Schulstrasse"                       "Bertastrasse neu"
## [33] "Limmatquai --> Central"             "Tannenrauchstrasse"
## [35] "Mühlebachstrasse"
```

```r
table(data$bezeichnung)
```

```
##
##                     Andreasstrasse                       Baslerstrasse
##                              35035                              30284
##                   Bertastrasse neu                   Binzmühlestrasse
##                               5183                               9190
##                      Bucheggplatz                     Cassiopeiasteg
##                              24906                              23253
##                           Dammweg Hardbrücke Nord (Seite Altstetten)
##                              15645                              35035
##          Hardbrücke Süd (Seite HB)                         Hardeggsteg
##                              35011                               2688
##                  Hofwiesenstrasse                          Katzenbach
##                               8301                              21094
##    Kloster-Fahr-Weg (Europabrücke)        Langstrasse (Fahrbahn Nord)
##                              27213                              20253
##         Langstrasse (Fahrbahn Süd)    Langstrasse (Unterführung Nord)
##                              20253                              61696
##     Langstrasse (Unterführung Süd)                    Letten / Dynamo
##                              16484                               1728
##                          Limmatquai            Limmatquai --> Bellevue
##                              35035                              35011
##             Limmatquai --> Central                      Lux-Guyer-Weg
##                               3791                              31555
##                       Militärbrücke                    Mühlebachstrasse
##                              33980                                263
##                          Mythenquai            Mythenquai Unterführung
##                              11062                              28221
##                     Ohmstrasse Ost                    Ohmstrasse West
##                              28053                              12736
##                  Scheuchzerstrasse                         Schulstrasse
##                              33163                               6095
##                      Sihlpromenade                          Talstrasse
##                               8300                              31843
##                 Tannenrauchstrasse                        Tödistrasse
##                               2087                               9310
##                    Weinbergfussweg
##                              25726
```

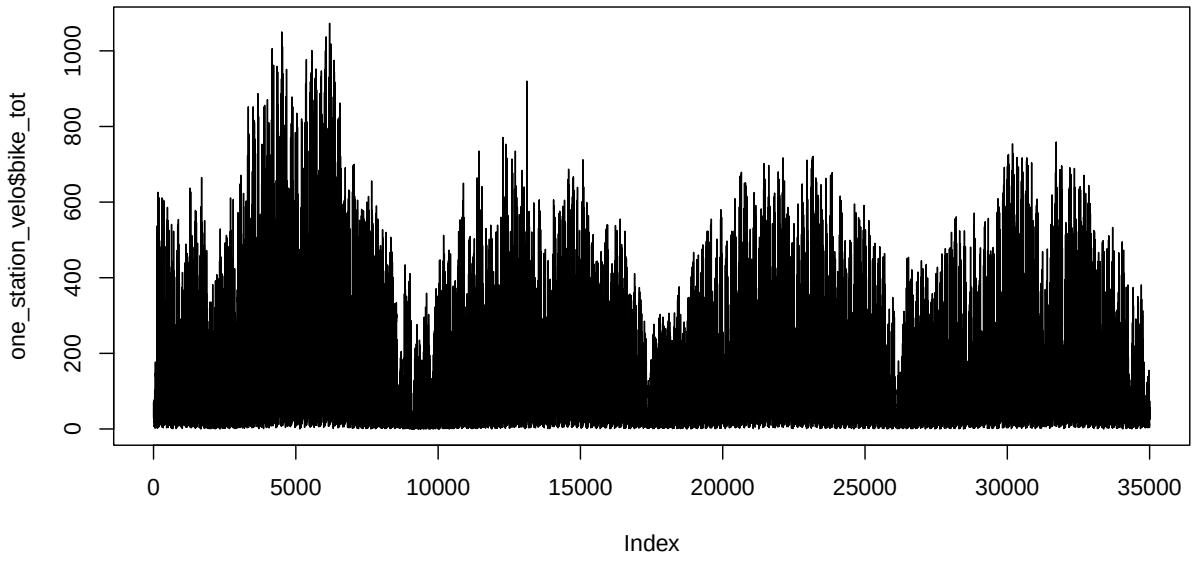
```r
# select a station with only velo traffic
one_station <- data[data$bezeichnung == "Langstrasse (Unterführung Nord)",]
```

```r
# for some Datetimes we have two rows, one for bike and one for foot.
# We only select the Standort 2989 as this one is for velo and the other for foot.
one_station_velo <- one_station[one_station$Standort==2989,]
```

Exploratory data analysis

Plot the velo traffic for the station Langstrasse (Unterführung Nord) for the years 2020-2023.

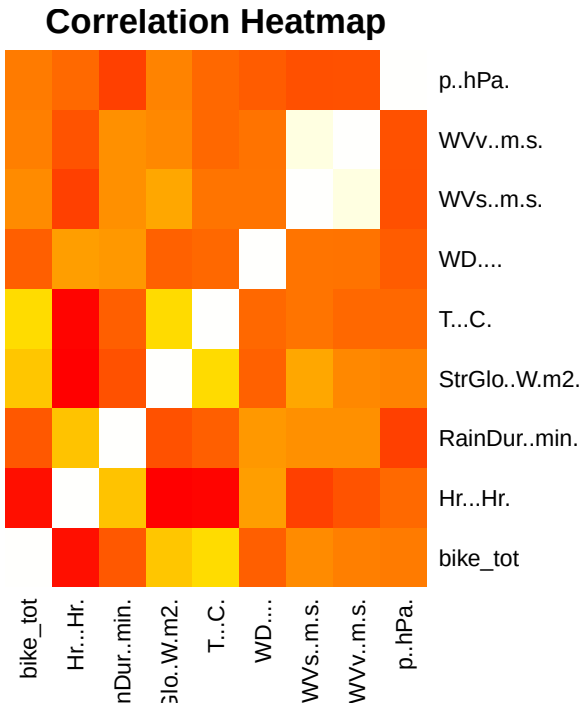
```r
plot(one_station_velo$bike_tot, type="l")
```



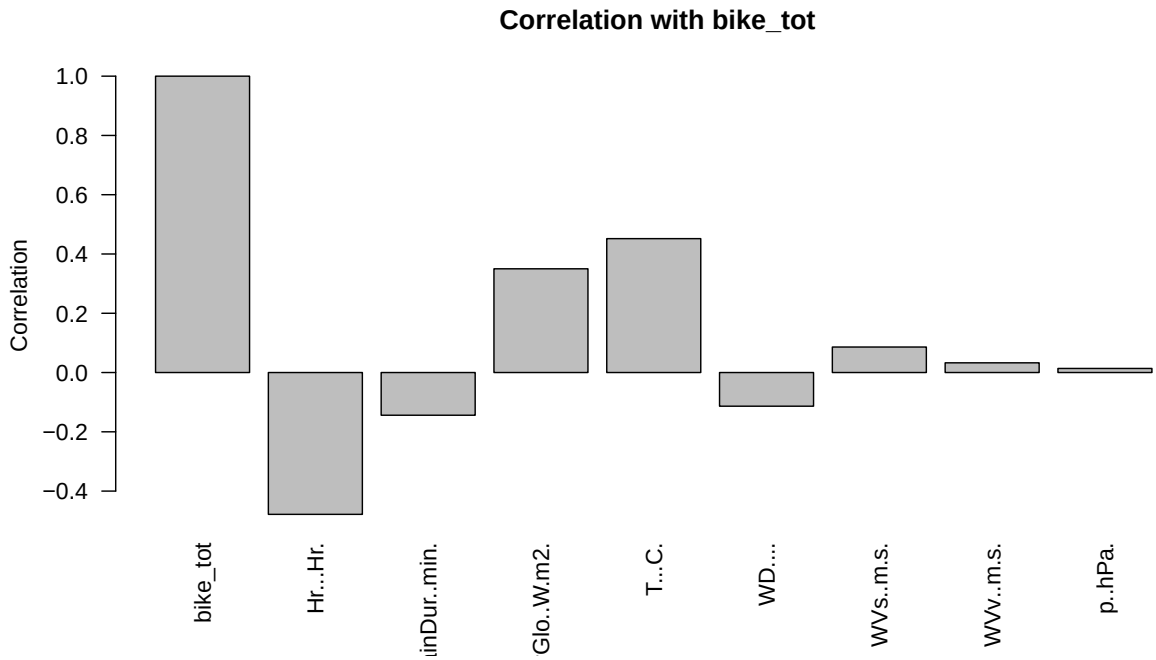

```r
# Calculate correlation matrix

correlation_matrix <- cor(na.omit(subset(one_station_velo, select = c("bike_tot", "Hr...Hr.", "RainDur.
rownames(correlation_matrix) <- colnames(correlation_matrix)

dim(correlation_matrix)
```

```
## [1] 9 9
```

```r
library(heatmaply)
heatmap(correlation_matrix,
        Rowv = NA, Colv = NA,
        col = heat.colors(256),
        scale = "none",
        margins = c(5, 10),
        main = "Correlation Heatmap")
```




```
barplot(correlation_matrix[1,], las=2,
        main= " Correlation with bike_tot",
        ylab="Correlation")
```



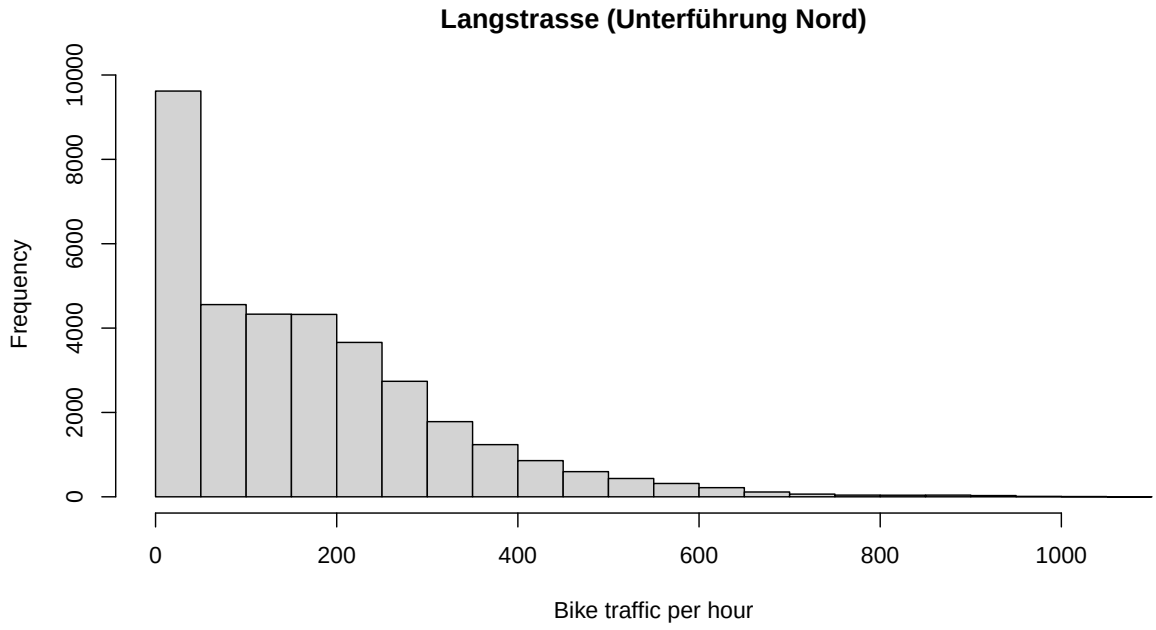
The relative humidity seems to have the highest (negative) correlation with bike_tot. Surprisingly, the rain duration is only weakly correlated. The global radiation and the temperature are positively correlated with bike_tot.

Regression models

First we analyze the outcome variable bike_tot. Since it is count data, a poisson regression might be appropriate. Let's check the assumptions:

- Response follows poisson distribution: Yes, count data per time unit (hour)
- Independence: Yes, the bike_tot depends on time and other variables but not necessarily on previous bike_tot.
- Mean = Variance: this assumption is clearly not met since variance is much larger than mean.

```
# looks like poisson (makes sense because counts per time unit)
hist(one_station_velo$bike_tot, main="Langstrasse (Unterführung Nord)", xlab = "Bike traffic per hour")
```




```
# var > mean --> this means we have overdispersion
mean(one_station_velo$bike_tot)
```

```
## [1] 168.3156
```

```
var(one_station_velo$bike_tot)
```

```
## [1] 22753.93
```

Since we have over-dispersion (variance > mean) a Poisson regression cannot properly model the data because it has only one parameter λ. Therefore the negative binomial distribution seems to be more appropriate. It can be seen as a generalization of the poisson regression that has one additional parameter to model the over-dispersion.

Train/test-split

Use the year 2022 as training set and 2023 as test set.

```
# approximately equal amount of observations in each year
table(one_station_velo$Year)
```

```
##
## 2020 2021 2022 2023
## 8735 8759 8759 8758
```

```
train <- one_station_velo[one_station_velo$Year==2022,]

test <- one_station_velo[one_station_velo$Year==2023,]
```

Model training (on 2022 data)

Negative binomial regression

We train models with different variables and check the RMSE of the training set.

Model with only weekday + rainduration:

```
# weather data
# Luftdruck (p), die Niederschlagsdauer (RainDur), die Globalstrahlung (StrGlo), die Temperatur (T), di

set.seed(123)
glm_nb_1 <-glm.nb(bike_tot ~ Weekday + RainDur..min., data=train)


summary(glm_nb_1)
```

```
##
## Call:
## glm.nb(formula = bike_tot ~ Weekday + RainDur..min., data = train,
##     init.theta = 1.113072727, link = log)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         5.2343528  0.0270472 193.526  < 2e-16 ***
## WeekdayDonnerstag   0.0284362  0.0380682   0.747    0.455
## WeekdayFreitag     -0.0616465  0.0381948  -1.614    0.107
## WeekdayMittwoch     0.0084244  0.0380989   0.221    0.825
## WeekdayMontag      -0.1543060  0.0380745  -4.053 5.06e-05 ***
## WeekdaySamstag     -0.3108984  0.0379223  -8.198 2.44e-16 ***
## WeekdaySonntag     -0.6273148  0.0381294 -16.452  < 2e-16 ***
## RainDur..min.      -0.0118218  0.0008002 -14.773  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1131) family taken to be 1)
##
```

```
##     Null deviance: 10563.8  on 8758  degrees of freedom
## Residual deviance:  9965.9  on 8751  degrees of freedom
## AIC: 105507
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.1131
##           Std. Err.:  0.0153
##
##  2 x log-likelihood:  -105488.5950
```

```
prediction_errors <- (predict(glm_nb_1, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 128.1769
```

Model with addidionally global radiation (slightly better):

```
glm_nb_2<-glm.nb(bike_tot ~ Weekday +  RainDur..min.+ StrGlo..W.m2. , data=train)


summary(glm_nb_2)
```

```
##
## Call:
## glm.nb(formula = bike_tot ~ Weekday + RainDur..min. + StrGlo..W.m2.,
##     data = train, init.theta = 1.185517915, link = log)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         5.020e+00  2.708e-02 185.402  < 2e-16 ***
## WeekdayDonnerstag   2.658e-02  3.690e-02   0.720   0.4713
## WeekdayFreitag     -6.155e-02  3.702e-02  -1.662   0.0964 .
## WeekdayMittwoch    -5.707e-03  3.693e-02  -0.155   0.8772
## WeekdayMontag      -1.738e-01  3.692e-02  -4.707 2.51e-06 ***
## WeekdaySamstag     -3.260e-01  3.676e-02  -8.868  < 2e-16 ***
## WeekdaySonntag     -6.337e-01  3.697e-02 -17.141  < 2e-16 ***
## RainDur..min.      -7.526e-03  7.883e-04  -9.547  < 2e-16 ***
## StrGlo..W.m2.       1.104e-03  4.122e-05  26.793  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1855) family taken to be 1)
##
##     Null deviance: 11233.1  on 8758  degrees of freedom
## Residual deviance:  9908.8  on 8750  degrees of freedom
## AIC: 104842
```

```
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.1855
##           Std. Err.:  0.0164
##
##  2 x log-likelihood:  -104822.1470
```

```
prediction_errors <- (predict(glm_nb_2, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 123.0556
```

Model with additinally hour of the day (RMSE improved a lot):

```
set.seed(123)
glm_nb_3<-glm.nb(bike_tot ~ Time + Weekday + StrGlo..W.m2. + RainDur..min., data=train)


summary(glm_nb_3)
```

```
##
## Call:
## glm.nb(formula = bike_tot ~ Time + Weekday + StrGlo..W.m2. +
##     RainDur..min., data = train, init.theta = 3.013815913, link = log)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         4.254e+00  3.452e-02 123.243  < 2e-16 ***
## Time01:00          -4.971e-01  4.389e-02 -11.325  < 2e-16 ***
## Time02:00          -9.060e-01  4.431e-02 -20.450  < 2e-16 ***
## Time03:00          -1.291e+00  4.481e-02 -28.812  < 2e-16 ***
## Time04:00          -1.664e+00  4.555e-02 -36.533  < 2e-16 ***
## Time05:00          -1.466e+00  4.512e-02 -32.497  < 2e-16 ***
## Time06:00          -3.236e-01  4.381e-02  -7.386 1.51e-13 ***
## Time07:00           8.927e-01  4.362e-02  20.464  < 2e-16 ***
## Time08:00           1.279e+00  4.419e-02  28.935  < 2e-16 ***
## Time09:00           6.931e-01  4.526e-02  15.313  < 2e-16 ***
## Time10:00           3.859e-01  4.634e-02   8.327  < 2e-16 ***
## Time11:00           4.979e-01  4.710e-02  10.572  < 2e-16 ***
## Time12:00           6.140e-01  4.732e-02  12.975  < 2e-16 ***
## Time13:00           6.690e-01  4.691e-02  14.260  < 2e-16 ***
## Time14:00           6.415e-01  4.605e-02  13.931  < 2e-16 ***
## Time15:00           7.520e-01  4.492e-02  16.741  < 2e-16 ***
## Time16:00           1.036e+00  4.396e-02  23.555  < 2e-16 ***
## Time17:00           1.533e+00  4.343e-02  35.298  < 2e-16 ***
## Time18:00           1.615e+00  4.324e-02  37.359  < 2e-16 ***
```

```
## Time19:00          1.304e+00  4.325e-02  30.156  < 2e-16 ***
## Time20:00          1.003e+00  4.329e-02  23.172  < 2e-16 ***
## Time21:00          7.390e-01  4.334e-02  17.051  < 2e-16 ***
## Time22:00          6.442e-01  4.337e-02  14.856  < 2e-16 ***
## Time23:00          3.922e-01  4.343e-02   9.029  < 2e-16 ***
## WeekdayDonnerstag  8.148e-02  2.356e-02   3.459 0.000542 ***
## WeekdayFreitag     7.310e-02  2.364e-02   3.092 0.001987 **
## WeekdayMittwoch    8.851e-03  2.360e-02   0.375 0.707607
## WeekdayMontag     -1.600e-01  2.362e-02  -6.772 1.27e-11 ***
## WeekdaySamstag     8.226e-02  2.346e-02   3.507 0.000453 ***
## WeekdaySonntag    -1.306e-01  2.361e-02  -5.530 3.21e-08 ***
## StrGlo..W.m2.      8.120e-04  3.928e-05  20.671  < 2e-16 ***
## RainDur..min.     -7.267e-03  5.125e-04 -14.181  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.0138) family taken to be 1)
##
##     Null deviance: 27521.5  on 8758  degrees of freedom
## Residual deviance:  9478.9  on 8727  degrees of freedom
## AIC: 95973
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.0138
##           Std. Err.:  0.0471
##
##  2 x log-likelihood:  -95907.3530
```

```
prediction_errors <- (predict(glm_nb_3, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 80.27359
```

Additionally Month, wind speed and temperature (RMSE better again, now around 69):

```
set.seed(123)
glm_nb_4 <-glm.nb(bike_tot ~ Time + Weekday + Month+ StrGlo..W.m2. +WVv..m.s.  + T...C. + RainDur..min.


summary(glm_nb_4)
```

```
##
## Call:
## glm.nb(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min., data = train, init.theta = 3.762197024,
##     link = log)
```

```
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         3.688e+00  3.754e-02  98.244  < 2e-16 ***
## Time01:00          -4.723e-01  3.969e-02 -11.898  < 2e-16 ***
## Time02:00          -8.738e-01  4.019e-02 -21.743  < 2e-16 ***
## Time03:00          -1.246e+00  4.078e-02 -30.543  < 2e-16 ***
## Time04:00          -1.604e+00  4.163e-02 -38.522  < 2e-16 ***
## Time05:00          -1.396e+00  4.116e-02 -33.908  < 2e-16 ***
## Time06:00          -2.080e-01  3.968e-02  -5.243 1.58e-07 ***
## Time07:00           1.081e+00  3.952e-02  27.339  < 2e-16 ***
## Time08:00           1.548e+00  4.025e-02  38.450  < 2e-16 ***
## Time09:00           1.041e+00  4.157e-02  25.038  < 2e-16 ***
## Time10:00           7.969e-01  4.288e-02  18.585  < 2e-16 ***
## Time11:00           9.293e-01  4.377e-02  21.234  < 2e-16 ***
## Time12:00           1.046e+00  4.403e-02  23.756  < 2e-16 ***
## Time13:00           1.068e+00  4.359e-02  24.509  < 2e-16 ***
## Time14:00           9.769e-01  4.259e-02  22.938  < 2e-16 ***
## Time15:00           9.958e-01  4.128e-02  24.123  < 2e-16 ***
## Time16:00           1.180e+00  4.020e-02  29.345  < 2e-16 ***
## Time17:00           1.591e+00  3.959e-02  40.188  < 2e-16 ***
## Time18:00           1.615e+00  3.928e-02  41.121  < 2e-16 ***
## Time19:00           1.273e+00  3.918e-02  32.481  < 2e-16 ***
## Time20:00           9.563e-01  3.912e-02  24.446  < 2e-16 ***
## Time21:00           6.989e-01  3.911e-02  17.870  < 2e-16 ***
## Time22:00           6.262e-01  3.909e-02  16.017  < 2e-16 ***
## Time23:00           3.832e-01  3.915e-02   9.788  < 2e-16 ***
## WeekdayDonnerstag   8.672e-02  2.129e-02   4.073 4.64e-05 ***
## WeekdayFreitag      8.084e-02  2.132e-02   3.792 0.000149 ***
## WeekdayMittwoch     1.846e-03  2.128e-02   0.087 0.930871
## WeekdayMontag      -1.352e-01  2.131e-02  -6.344 2.24e-10 ***
## WeekdaySamstag      1.007e-01  2.118e-02   4.754 1.99e-06 ***
## WeekdaySonntag     -1.328e-01  2.135e-02  -6.221 4.94e-10 ***
## Month02             8.397e-02  2.928e-02   2.868 0.004130 **
## Month03             3.606e-01  2.917e-02  12.363  < 2e-16 ***
## Month04             2.868e-01  3.094e-02   9.269  < 2e-16 ***
## Month05             4.371e-01  3.668e-02  11.918  < 2e-16 ***
## Month06             4.381e-01  4.086e-02  10.723  < 2e-16 ***
## Month07             3.740e-01  4.281e-02   8.736  < 2e-16 ***
## Month08             3.227e-01  4.152e-02   7.772 7.74e-15 ***
## Month09             3.179e-01  3.508e-02   9.062  < 2e-16 ***
## Month10             2.591e-01  3.425e-02   7.565 3.88e-14 ***
## Month11             3.494e-01  2.962e-02  11.796  < 2e-16 ***
## Month12            -2.159e-02  2.809e-02  -0.769 0.442145
## StrGlo..W.m2.      -1.024e-04  4.371e-05  -2.344 0.019101 *
## WVv..m.s.          -5.162e-02  1.193e-02  -4.326 1.52e-05 ***
## T...C.              2.243e-02  1.681e-03  13.344  < 2e-16 ***
## RainDur..min.      -7.601e-03  4.702e-04 -16.166  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.7622) family taken to be 1)
##
##     Null deviance: 33892.1  on 8758  degrees of freedom
```

```
## Residual deviance:  9515.3  on 8714  degrees of freedom
## AIC: 94146
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.7622
##           Std. Err.:  0.0609
##
##  2 x log-likelihood:  -94053.7110
```

```
prediction_errors <- (predict(glm_nb_4, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 69.90891
```

Now additionally air pressure and humidity (model barely improved with this):

```
set.seed(123)
glm_nb_5 <-glm.nb(bike_tot ~ Time + Weekday + Month+ StrGlo..W.m2. +WVv..m.s.  + T...C. + RainDur..min.+


summary(glm_nb_5)
```

```
##
## Call:
## glm.nb(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min. + p..hPa. + Hr...Hr.,
##     data = train, init.theta = 3.847404694, link = log)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -5.220e-02  9.560e-01  -0.055   0.9565
## Time01:00         -4.650e-01  3.930e-02 -11.832  < 2e-16 ***
## Time02:00         -8.540e-01  3.980e-02 -21.455  < 2e-16 ***
## Time03:00         -1.220e+00  4.041e-02 -30.193  < 2e-16 ***
## Time04:00         -1.577e+00  4.127e-02 -38.197  < 2e-16 ***
## Time05:00         -1.367e+00  4.081e-02 -33.499  < 2e-16 ***
## Time06:00         -1.751e-01  3.934e-02  -4.452 8.52e-06 ***
## Time07:00          1.118e+00  3.921e-02  28.512  < 2e-16 ***
## Time08:00          1.589e+00  3.995e-02  39.767  < 2e-16 ***
## Time09:00          1.080e+00  4.125e-02  26.180  < 2e-16 ***
## Time10:00          8.319e-01  4.252e-02  19.567  < 2e-16 ***
## Time11:00          9.537e-01  4.335e-02  21.999  < 2e-16 ***
## Time12:00          1.060e+00  4.360e-02  24.317  < 2e-16 ***
## Time13:00          1.070e+00  4.315e-02  24.787  < 2e-16 ***
## Time14:00          9.618e-01  4.220e-02  22.793  < 2e-16 ***
## Time15:00          9.639e-01  4.098e-02  23.520  < 2e-16 ***
```

```
## Time16:00          1.135e+00  4.000e-02  28.372  < 2e-16 ***
## Time17:00          1.542e+00  3.942e-02  39.125  < 2e-16 ***
## Time18:00          1.569e+00  3.911e-02  40.114  < 2e-16 ***
## Time19:00          1.231e+00  3.896e-02  31.590  < 2e-16 ***
## Time20:00          9.216e-01  3.882e-02  23.739  < 2e-16 ***
## Time21:00          6.720e-01  3.876e-02  17.338  < 2e-16 ***
## Time22:00          6.099e-01  3.871e-02  15.755  < 2e-16 ***
## Time23:00          3.768e-01  3.875e-02   9.723  < 2e-16 ***
## WeekdayDonnerstag  1.026e-01  2.110e-02   4.861 1.17e-06 ***
## WeekdayFreitag     8.351e-02  2.111e-02   3.956 7.61e-05 ***
## WeekdayMittwoch    8.694e-03  2.106e-02   0.413   0.6797
## WeekdayMontag     -1.421e-01  2.110e-02  -6.733 1.66e-11 ***
## WeekdaySamstag     9.833e-02  2.101e-02   4.680 2.87e-06 ***
## WeekdaySonntag    -1.400e-01  2.113e-02  -6.626 3.46e-11 ***
## Month02            7.130e-02  2.917e-02   2.444   0.0145 *
## Month03            2.848e-01  3.034e-02   9.387  < 2e-16 ***
## Month04            3.321e-01  3.253e-02  10.209  < 2e-16 ***
## Month05            5.350e-01  3.729e-02  14.350  < 2e-16 ***
## Month06            5.903e-01  4.225e-02  13.972  < 2e-16 ***
## Month07            4.680e-01  4.303e-02  10.875  < 2e-16 ***
## Month08            4.750e-01  4.279e-02  11.101  < 2e-16 ***
## Month09            4.650e-01  3.686e-02  12.616  < 2e-16 ***
## Month10            4.384e-01  3.641e-02  12.040  < 2e-16 ***
## Month11            4.787e-01  3.127e-02  15.309  < 2e-16 ***
## Month12            7.508e-02  2.955e-02   2.541   0.0111 *
## StrGlo..W.m2.     -2.659e-04  4.521e-05  -5.881 4.07e-09 ***
## WVv..m.s.         -6.002e-02  1.217e-02  -4.933 8.09e-07 ***
## T...C.             1.256e-02  1.862e-03   6.747 1.51e-11 ***
## RainDur..min.     -4.903e-03  5.072e-04  -9.667  < 2e-16 ***
## p..hPa.            4.414e-03  9.721e-04   4.541 5.60e-06 ***
## Hr...Hr.          -7.108e-03  5.812e-04 -12.231  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.8474) family taken to be 1)
##
##     Null deviance: 34607.6  on 8758  degrees of freedom
## Residual deviance:  9530.4  on 8712  degrees of freedom
## AIC: 93976
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.8474
##           Std. Err.:  0.0625
##
##  2 x log-likelihood:  -93880.4690
```

```
prediction_errors <- (predict(glm_nb_5, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 68.60677
```

Poisson regression

Let's see how a poisson regression would perform. First easy model:

```
set.seed(123)
model_pois1<-glm(bike_tot ~ Weekday + RainDur..min., family=poisson(link="log"), data=train)

summary(model_pois1)
```

```
##
## Call:
## glm(formula = bike_tot ~ Weekday + RainDur..min., family = poisson(link = "log"),
##     data = train)
##
## Coefficients:
##                     Estimate Std. Error  z value Pr(>|z|)
## (Intercept)        5.233e+00  2.112e-03 2477.608  < 2e-16 ***
## WeekdayDonnerstag  2.445e-02  2.964e-03    8.250  < 2e-16 ***
## WeekdayFreitag    -5.087e-02  3.049e-03  -16.681  < 2e-16 ***
## WeekdayMittwoch    8.254e-03  2.985e-03    2.765  0.00569 **
## WeekdayMontag     -1.453e-01  3.092e-03  -46.971  < 2e-16 ***
## WeekdaySamstag    -3.019e-01  3.223e-03  -93.659  < 2e-16 ***
## WeekdaySonntag    -6.289e-01  3.555e-03 -176.887  < 2e-16 ***
## RainDur..min.     -1.233e-02  8.468e-05 -145.590  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 981730  on 8758  degrees of freedom
## Residual deviance: 904651  on 8751  degrees of freedom
## AIC: 960196
##
## Number of Fisher Scoring iterations: 5
```

```
prediction_errors <- (predict(model_pois1, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 128.1638
```

Now with the same variables as in our fourth negative binomial model (RMSE of 57, surprisingly much better than negative binomial):

```r
model_pois2 <-glm(bike_tot ~ Time + Weekday + Month+ StrGlo..W.m2. +WVv..m.s.  + T...C. + RainDur..min.

summary(model_pois2)
```

```
##
## Call:
## glm(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min., family = poisson(link = "log"),
##     data = train)
##
## Coefficients:
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         3.897e+00  7.710e-03  505.480  < 2e-16 ***
## Time01:00          -5.029e-01  1.034e-02  -48.630  < 2e-16 ***
## Time02:00          -9.009e-01  1.185e-02  -76.000  < 2e-16 ***
## Time03:00          -1.277e+00  1.368e-02  -93.307  < 2e-16 ***
## Time04:00          -1.647e+00  1.596e-02 -103.247  < 2e-16 ***
## Time05:00          -1.436e+00  1.465e-02  -98.023  < 2e-16 ***
## Time06:00          -2.395e-01  9.652e-03  -24.810  < 2e-16 ***
## Time07:00           1.042e+00  7.472e-03  139.452  < 2e-16 ***
## Time08:00           1.494e+00  7.228e-03  206.696  < 2e-16 ***
## Time09:00           9.729e-01  7.804e-03  124.671  < 2e-16 ***
## Time10:00           7.220e-01  8.216e-03   87.883  < 2e-16 ***
## Time11:00           8.554e-01  8.173e-03  104.664  < 2e-16 ***
## Time12:00           9.640e-01  8.090e-03  119.153  < 2e-16 ***
## Time13:00           9.901e-01  8.001e-03  123.744  < 2e-16 ***
## Time14:00           9.034e-01  7.944e-03  113.731  < 2e-16 ***
## Time15:00           9.307e-01  7.739e-03  120.261  < 2e-16 ***
## Time16:00           1.128e+00  7.418e-03  152.032  < 2e-16 ***
## Time17:00           1.549e+00  7.045e-03  219.928  < 2e-16 ***
## Time18:00           1.583e+00  6.977e-03  226.955  < 2e-16 ***
## Time19:00           1.255e+00  7.171e-03  174.958  < 2e-16 ***
## Time20:00           9.575e-01  7.424e-03  128.975  < 2e-16 ***
## Time21:00           7.070e-01  7.705e-03   91.758  < 2e-16 ***
## Time22:00           6.321e-01  7.812e-03   80.908  < 2e-16 ***
## Time23:00           4.026e-01  8.154e-03   49.378  < 2e-16 ***
## WeekdayDonnerstag   2.322e-02  2.978e-03    7.796 6.38e-15 ***
## WeekdayFreitag     -5.267e-02  3.053e-03  -17.253  < 2e-16 ***
## WeekdayMittwoch    -5.813e-03  2.989e-03   -1.945   0.0518 .
## WeekdayMontag      -1.409e-01  3.099e-03  -45.466  < 2e-16 ***
## WeekdaySamstag     -2.729e-01  3.234e-03  -84.403  < 2e-16 ***
## WeekdaySonntag     -6.087e-01  3.563e-03 -170.862  < 2e-16 ***
## Month02             4.936e-02  5.433e-03    9.086  < 2e-16 ***
## Month03             3.320e-01  5.042e-03   65.842  < 2e-16 ***
## Month04             2.316e-01  5.459e-03   42.424  < 2e-16 ***
## Month05             3.895e-01  6.003e-03   64.885  < 2e-16 ***
## Month06             3.705e-01  6.555e-03   56.522  < 2e-16 ***
## Month07             3.104e-01  6.875e-03   45.158  < 2e-16 ***
## Month08             2.747e-01  6.706e-03   40.955  < 2e-16 ***
## Month09             3.176e-01  5.783e-03   54.914  < 2e-16 ***
## Month10             2.537e-01  5.650e-03   44.903  < 2e-16 ***
## Month11             3.366e-01  5.069e-03   66.395  < 2e-16 ***
## Month12            -5.202e-02  5.425e-03   -9.588  < 2e-16 ***
```

```
## StrGlo..W.m2.     -2.773e-05  6.358e-06   -4.361 1.30e-05 ***
## WVv..m.s.         -4.294e-02  1.913e-03  -22.451  < 2e-16 ***
## T...C.             2.123e-02  2.477e-04   85.732  < 2e-16 ***
## RainDur..min.     -7.923e-03  8.744e-05  -90.603  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 981730  on 8758  degrees of freedom
## Residual deviance: 189770  on 8714  degrees of freedom
## AIC: 245389
##
## Number of Fisher Scoring iterations: 5
```

```
prediction_errors <- (predict(model_pois2, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 57.03907
```

Linear model

Just as a benchmark a linear model with the same variables as in the last poisson regression (about equal with our best negative binomial model):

```
lm1 <-lm(bike_tot ~ Time + Weekday + Month+ StrGlo..W.m2. +WVv..m.s.  + T...C. + RainDur..min., data=tra

summary(lm1)
```

```
##
## Call:
## lm(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -312.62  -39.65   -4.23   36.20  332.18
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        35.088326   4.924839   7.125 1.13e-12 ***
## Time01:00         -25.469518   5.179417  -4.917 8.93e-07 ***
## Time02:00         -37.981576   5.184428  -7.326 2.58e-13 ***
## Time03:00         -45.390042   5.182896  -8.758  < 2e-16 ***
## Time04:00         -50.104307   5.185354  -9.663  < 2e-16 ***
## Time05:00         -46.674507   5.188718  -8.995  < 2e-16 ***
## Time06:00         -12.876526   5.202882  -2.475 0.013347 *
```

```
## Time07:00         118.408586   5.251783  22.546  < 2e-16 ***
## Time08:00         223.366352   5.358552  41.684  < 2e-16 ***
## Time09:00          94.514964   5.521999  17.116  < 2e-16 ***
## Time10:00          48.649123   5.687762   8.553  < 2e-16 ***
## Time11:00          65.827781   5.810262  11.330  < 2e-16 ***
## Time12:00          84.135012   5.849645  14.383  < 2e-16 ***
## Time13:00          90.190528   5.790471  15.576  < 2e-16 ***
## Time14:00          76.583699   5.655683  13.541  < 2e-16 ***
## Time15:00          86.747191   5.482562  15.822  < 2e-16 ***
## Time16:00         133.835390   5.344462  25.042  < 2e-16 ***
## Time17:00         259.476453   5.271447  49.223  < 2e-16 ***
## Time18:00         273.683712   5.232197  52.308  < 2e-16 ***
## Time19:00         173.953220   5.213679  33.365  < 2e-16 ***
## Time20:00         108.837770   5.197904  20.939  < 2e-16 ***
## Time21:00          68.637028   5.187966  13.230  < 2e-16 ***
## Time22:00          59.106025   5.182308  11.405  < 2e-16 ***
## Time23:00          32.833042   5.179450   6.339 2.43e-10 ***
## WeekdayDonnerstag   1.555647   2.811083   0.553 0.580005
## WeekdayFreitag    -10.015422   2.813372  -3.560 0.000373 ***
## WeekdayMittwoch    -2.651304   2.806322  -0.945 0.344807
## WeekdayMontag     -24.390978   2.805337  -8.694  < 2e-16 ***
## WeekdaySamstag    -42.510114   2.795116 -15.209  < 2e-16 ***
## WeekdaySonntag    -81.416769   2.810864 -28.965  < 2e-16 ***
## Month02            -3.474437   3.812931  -0.911 0.362202
## Month03            25.469708   3.813287   6.679 2.55e-11 ***
## Month04             7.063451   4.038139   1.749 0.080294 .
## Month05            24.698038   4.811678   5.133 2.91e-07 ***
## Month06            19.216039   5.365303   3.582 0.000343 ***
## Month07             5.851350   5.621051   1.041 0.297918
## Month08             1.960226   5.453109   0.359 0.719253
## Month09            15.020547   4.601675   3.264 0.001102 **
## Month10             8.505638   4.495317   1.892 0.058509 .
## Month11            28.976932   3.881178   7.466 9.06e-14 ***
## Month12            -7.126553   3.651440  -1.952 0.051005 .
## StrGlo..W.m2.       0.048459   0.005821   8.325  < 2e-16 ***
## WVv..m.s.          -3.594518   1.566466  -2.295 0.021776 *
## T...C.              4.449218   0.222009  20.041  < 2e-16 ***
## RainDur..min.      -0.856263   0.061062 -14.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.96 on 8714 degrees of freedom
## Multiple R-squared:  0.725,  Adjusted R-squared:  0.7236
## F-statistic: 522.2 on 44 and 8714 DF,  p-value: < 2.2e-16
```

```
prediction_errors <- (predict(lm1, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 69.77685
```

```r
extractAIC(lm1)
```

```
## [1]    45.0 74459.2
```

The second poisson model was the best one in terms of square-root-MSE, hence we will select this model.

Model validation (on 2023 data)

Compare the model accuracies on the 2023 data. For the accuracy we take the prediction (backtransformed), subtract the true value and square it. The square root of the mean squared error (MSE) is then the accuracy score (average error).

```r
get_pred_error <- function(model){
  prediction_errors <- (predict(model, test, type="response") - test$bike_tot)^2

# sum(length(which(is.na(prediction_errors)) ))

 mean_pred_error <- sqrt(mean(na.omit(prediction_errors)))
 return(mean_pred_error)

}



# negative binomial
get_pred_error(glm_nb_1)
```

```
## [1] 129.7437
```

```r
get_pred_error(glm_nb_2)
```

```
## [1] 123.5692
```

```r
get_pred_error(glm_nb_3)
```

```
## [1] 81.43214
```

```r
get_pred_error(glm_nb_4)
```

```
## [1] 71.43266
```

```r
get_pred_error(glm_nb_5)
```

```
## [1] 69.76264
```

```r
# poisson
get_pred_error(model_pois1)
```

```
## [1] 129.7368
```

```r
get_pred_error(model_pois2)
```

```
## [1] 58.0245
```

```r
# linear
get_pred_error(lm1)
```

```
## [1] 71.26215
```

As in the training set, the the model model_pois2 perfomrmed best. Following we discuss this model:

```r
summary(model_pois2)
```

```
##
## Call:
## glm(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min., family = poisson(link = "log"),
##     data = train)
##
## Coefficients:
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         3.897e+00  7.710e-03  505.480  < 2e-16 ***
## Time01:00          -5.029e-01  1.034e-02  -48.630  < 2e-16 ***
## Time02:00          -9.009e-01  1.185e-02  -76.000  < 2e-16 ***
## Time03:00          -1.277e+00  1.368e-02  -93.307  < 2e-16 ***
## Time04:00          -1.647e+00  1.596e-02 -103.247  < 2e-16 ***
## Time05:00          -1.436e+00  1.465e-02  -98.023  < 2e-16 ***
## Time06:00          -2.395e-01  9.652e-03  -24.810  < 2e-16 ***
## Time07:00           1.042e+00  7.472e-03  139.452  < 2e-16 ***
## Time08:00           1.494e+00  7.228e-03  206.696  < 2e-16 ***
## Time09:00           9.729e-01  7.804e-03  124.671  < 2e-16 ***
## Time10:00           7.220e-01  8.216e-03   87.883  < 2e-16 ***
## Time11:00           8.554e-01  8.173e-03  104.664  < 2e-16 ***
## Time12:00           9.640e-01  8.090e-03  119.153  < 2e-16 ***
## Time13:00           9.901e-01  8.001e-03  123.744  < 2e-16 ***
## Time14:00           9.034e-01  7.944e-03  113.731  < 2e-16 ***
## Time15:00           9.307e-01  7.739e-03  120.261  < 2e-16 ***
## Time16:00           1.128e+00  7.418e-03  152.032  < 2e-16 ***
## Time17:00           1.549e+00  7.045e-03  219.928  < 2e-16 ***
## Time18:00           1.583e+00  6.977e-03  226.955  < 2e-16 ***
## Time19:00           1.255e+00  7.171e-03  174.958  < 2e-16 ***
## Time20:00           9.575e-01  7.424e-03  128.975  < 2e-16 ***
## Time21:00           7.070e-01  7.705e-03   91.758  < 2e-16 ***
## Time22:00           6.321e-01  7.812e-03   80.908  < 2e-16 ***
## Time23:00           4.026e-01  8.154e-03   49.378  < 2e-16 ***
## WeekdayDonnerstag   2.322e-02  2.978e-03    7.796 6.38e-15 ***
## WeekdayFreitag     -5.267e-02  3.053e-03  -17.253  < 2e-16 ***
## WeekdayMittwoch    -5.813e-03  2.989e-03   -1.945   0.0518 .
## WeekdayMontag      -1.409e-01  3.099e-03  -45.466  < 2e-16 ***
## WeekdaySamstag     -2.729e-01  3.234e-03  -84.403  < 2e-16 ***
## WeekdaySonntag     -6.087e-01  3.563e-03 -170.862  < 2e-16 ***
```

```
## Month02            4.936e-02  5.433e-03    9.086  < 2e-16 ***
## Month03            3.320e-01  5.042e-03   65.842  < 2e-16 ***
## Month04            2.316e-01  5.459e-03   42.424  < 2e-16 ***
## Month05            3.895e-01  6.003e-03   64.885  < 2e-16 ***
## Month06            3.705e-01  6.555e-03   56.522  < 2e-16 ***
## Month07            3.104e-01  6.875e-03   45.158  < 2e-16 ***
## Month08            2.747e-01  6.706e-03   40.955  < 2e-16 ***
## Month09            3.176e-01  5.783e-03   54.914  < 2e-16 ***
## Month10            2.537e-01  5.650e-03   44.903  < 2e-16 ***
## Month11            3.366e-01  5.069e-03   66.395  < 2e-16 ***
## Month12           -5.202e-02  5.425e-03   -9.588  < 2e-16 ***
## StrGlo..W.m2.     -2.773e-05  6.358e-06   -4.361 1.30e-05 ***
## WVv..m.s.         -4.294e-02  1.913e-03  -22.451  < 2e-16 ***
## T...C.             2.123e-02  2.477e-04   85.732  < 2e-16 ***
## RainDur..min.     -7.923e-03  8.744e-05  -90.603  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 981730  on 8758  degrees of freedom
## Residual deviance: 189770  on 8714  degrees of freedom
## AIC: 245389
##
## Number of Fisher Scoring iterations: 5
```

The model is a generalized linear model, or more specifically a poisson regression. It was fitted using the factor variables Time, Weekday and Month and the numerical variables Global radiation (W/m2), Temperature (celsius) and the rain duration (minutes). Most parameters in the model are significant at a 5% significance level. The only exception, although only barely, is the weekday Wednesday.

```
coefficients(model_pois2)["T...C."]
```

```
##     T...C.
## 0.02123179
```

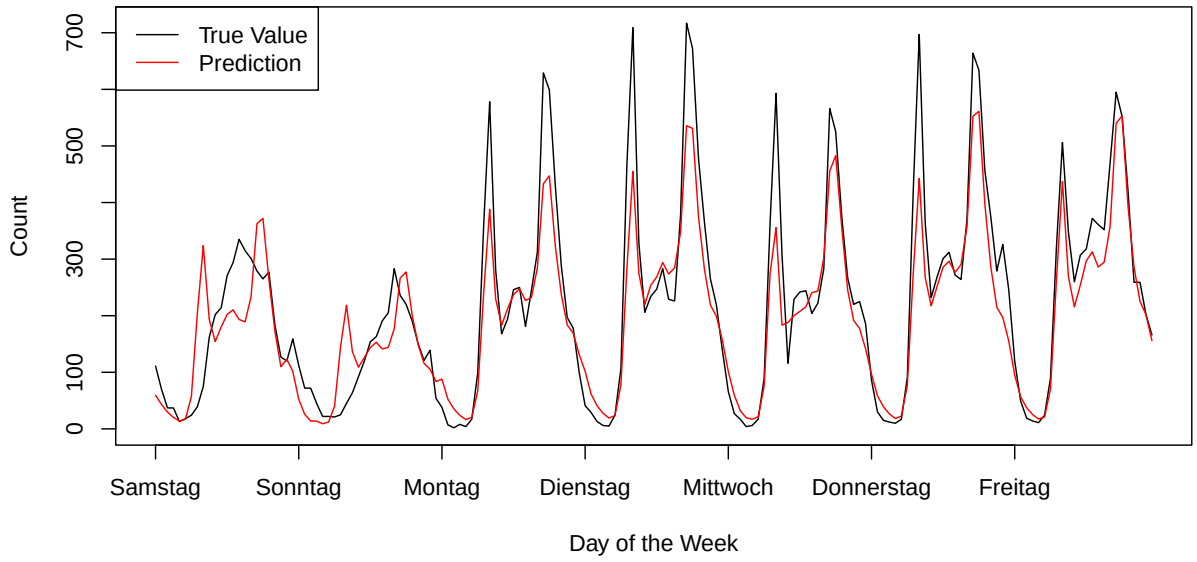
```
exp(coefficients(model_pois2)["T...C."])
```

```
##   T...C.
## 1.021459
```

The coefficient for the variable temperature is 0.021 . This means that for a one-unit increase in temperature, the expected log count of bike traffic in one hour increases by 0.021 (if all other variables remain unchanged).

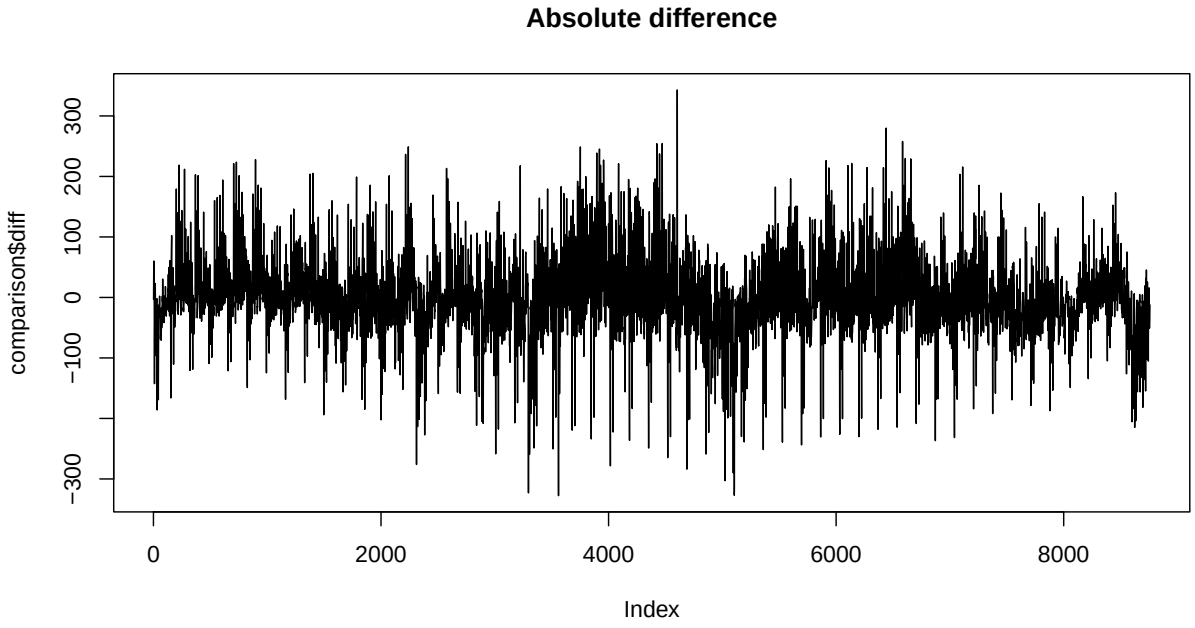
```
# For visualization prediction vs true value of first week July 2023

# Subset data for the month of July
july_data <- test[test$Month== "07", ]

# Further subset to show only the first week of July
first_week_july <- july_data[july_data$Datetime <= as.POSIXct("2023-07-07 23:00"), ]
```

```r
# Create comparison dataframe for the first week of July
comparison_first_week <- data.frame(true_value = first_week_july$bike_tot,
                                    prediction = predict(model_pois2, first_week_july, type="response"))

# Extract unique dates (start of each day)
unique_dates <- c("2023-07-01 00:00", "2023-07-02 00:00", "2023-07-03 00:00",
                  "2023-07-04 00:00", "2023-07-05 00:00", "2023-07-06 00:00",
                  "2023-07-07 00:00")

# Plot
plot(1:nrow(first_week_july), comparison_first_week$true_value, type="l", xlab="Day of the Week", ylab="
lines(1:nrow(first_week_july), comparison_first_week$prediction, col="red")

# Add legend
legend("topleft", legend=c("True Value", "Prediction"), col=c("black", "red"), lty=1)

# Add x-axis labels for weekdays
axis(1, at=which(first_week_july$Datetime %in% unique_dates), labels=format(as.Date(unique_dates), "%A"))
```

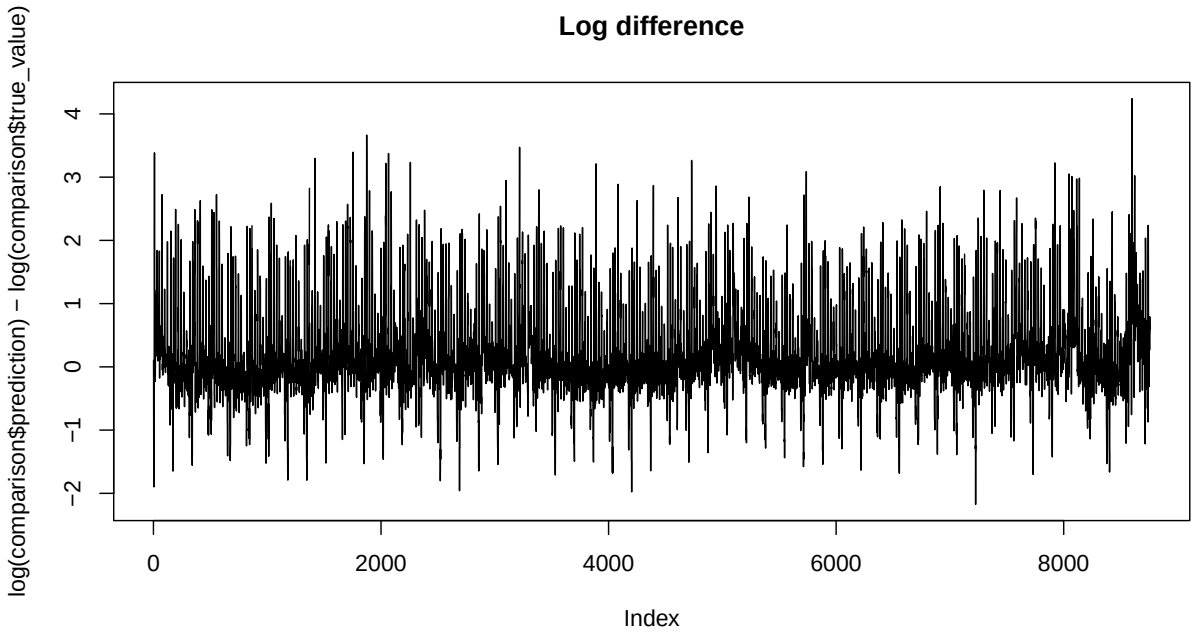


Check how we over- or underestimated:

```r
# make a dataframe of 2023 with true value and predictions
comparison <- data.frame(true_value = test$bike_tot, prediction = predict(model_pois2, test, type="respo

# calculate the differences
comparison$difference <- comparison$true_value -
  comparison$prediction

plot(comparison$diff, type="l", main="Absolute difference")
```




```
plot(log(comparison$prediction)-log(comparison$true_value), type="l", main="Log difference")
```




```
# we over estimated more than we underestimated (true - pred)
sum(comparison$diff>0)
```

```
## [1] 3847
```

```r
sum(comparison$diff<0)
```

```
## [1] 4911
```

Train poisson model on 2020-2022 (3 years)

Train on 2020-2022

```r
# train-test split
train <- one_station_velo[one_station_velo$Year!=2023,]

test <- one_station_velo[one_station_velo$Year==2023,]
```

```r
model_pois_3Y <-glm(bike_tot ~ Time + Weekday + Month+ StrGlo..W.m2. +WVv..m.s.  + T...C. + RainDur..mi

summary(model_pois_3Y)
```

```
##
## Call:
## glm(formula = bike_tot ~ Time + Weekday + Month + StrGlo..W.m2. +
##     WVv..m.s. + T...C. + RainDur..min., family = poisson(link = "log"),
##     data = train)
##
## Coefficients:
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         3.882e+00  4.392e-03  883.926  < 2e-16 ***
## Time01:00          -5.431e-01  6.033e-03  -90.022  < 2e-16 ***
## Time02:00          -9.528e-01  6.957e-03 -136.959  < 2e-16 ***
## Time03:00          -1.362e+00  8.157e-03 -166.941  < 2e-16 ***
## Time04:00          -1.701e+00  9.402e-03 -180.970  < 2e-16 ***
## Time05:00          -1.346e+00  8.140e-03 -165.351  < 2e-16 ***
## Time06:00          -1.102e-01  5.362e-03  -20.557  < 2e-16 ***
## Time07:00           1.113e+00  4.255e-03  261.543  < 2e-16 ***
## Time08:00           1.564e+00  4.114e-03  380.051  < 2e-16 ***
## Time09:00           1.041e+00  4.411e-03  236.004  < 2e-16 ***
## Time10:00           8.234e-01  4.604e-03  178.853  < 2e-16 ***
## Time11:00           9.573e-01  4.573e-03  209.321  < 2e-16 ***
## Time12:00           1.089e+00  4.518e-03  241.124  < 2e-16 ***
## Time13:00           1.127e+00  4.461e-03  252.695  < 2e-16 ***
## Time14:00           1.040e+00  4.427e-03  234.927  < 2e-16 ***
## Time15:00           1.064e+00  4.330e-03  245.735  < 2e-16 ***
## Time16:00           1.247e+00  4.180e-03  298.217  < 2e-16 ***
## Time17:00           1.627e+00  4.008e-03  405.844  < 2e-16 ***
## Time18:00           1.668e+00  3.978e-03  419.354  < 2e-16 ***
## Time19:00           1.345e+00  4.082e-03  329.459  < 2e-16 ***
## Time20:00           1.035e+00  4.225e-03  245.013  < 2e-16 ***
## Time21:00           7.693e-01  4.390e-03  175.238  < 2e-16 ***
## Time22:00           6.638e-01  4.474e-03  148.377  < 2e-16 ***
## Time23:00           4.196e-01  4.684e-03   89.582  < 2e-16 ***
## WeekdayDonnerstag   5.955e-03  1.653e-03    3.603 0.000314 ***
## WeekdayFreitag     -1.225e-02  1.660e-03   -7.378 1.61e-13 ***
```

```
## WeekdayMittwoch    1.150e-04  1.654e-03    0.070 0.944560
## WeekdayMontag     -1.280e-01  1.711e-03  -74.796  < 2e-16 ***
## WeekdaySamstag    -2.398e-01  1.768e-03 -135.641  < 2e-16 ***
## WeekdaySonntag    -5.849e-01  1.958e-03 -298.738  < 2e-16 ***
## Month02            1.392e-02  2.966e-03    4.692 2.71e-06 ***
## Month03            2.078e-01  2.778e-03   74.804  < 2e-16 ***
## Month04            1.711e-01  2.944e-03   58.110  < 2e-16 ***
## Month05            2.926e-01  3.039e-03   96.268  < 2e-16 ***
## Month06            3.083e-01  3.318e-03   92.896  < 2e-16 ***
## Month07            2.387e-01  3.417e-03   69.845  < 2e-16 ***
## Month08            2.222e-01  3.376e-03   65.818  < 2e-16 ***
## Month09            3.410e-01  3.083e-03  110.618  < 2e-16 ***
## Month10            2.354e-01  2.867e-03   82.089  < 2e-16 ***
## Month11            3.078e-01  2.719e-03  113.205  < 2e-16 ***
## Month12           -3.382e-02  2.936e-03  -11.519  < 2e-16 ***
## StrGlo..W.m2.     -4.611e-05  3.453e-06  -13.352  < 2e-16 ***
## WVv..m.s.         -2.615e-02  9.823e-04  -26.617  < 2e-16 ***
## T...C.             2.665e-02  1.242e-04  214.494  < 2e-16 ***
## RainDur..min.     -8.897e-03  4.686e-05 -189.847  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3556252  on 26252  degrees of freedom
## Residual deviance:  775867  on 26208  degrees of freedom
## AIC: 943559
##
## Number of Fisher Scoring iterations: 5
```

```
prediction_errors <- (predict(model_pois_3Y, train, type="response") - train$bike_tot)^2
sum(length(which(is.na(prediction_errors)) ))
```

```
## [1] 0
```

```
sqrt(mean(na.omit(prediction_errors)))
```

```
## [1] 75.9211
```

The RMSE of the training set is much worse here (2020-2022) compared to when we only use 2022 as training data. The reason is probably that during 2020 the bike traffic was quite different from the other years, probably caused by the beginning of covid, that changed the behaviour. This is visible in the plot at the beginning of this document.

Test the model on 2023

```
get_pred_error_2023 <- function(model){
  prediction_errors <- (predict(model, test, type="response") - test$bike_tot)^2
  mean_pred_error <- sqrt(mean(na.omit(prediction_errors)))
  return(mean_pred_error)

}
```

```r
# negative binomial

get_pred_error_2023(model_pois_3Y)
```

```
## [1] 65.90899
```

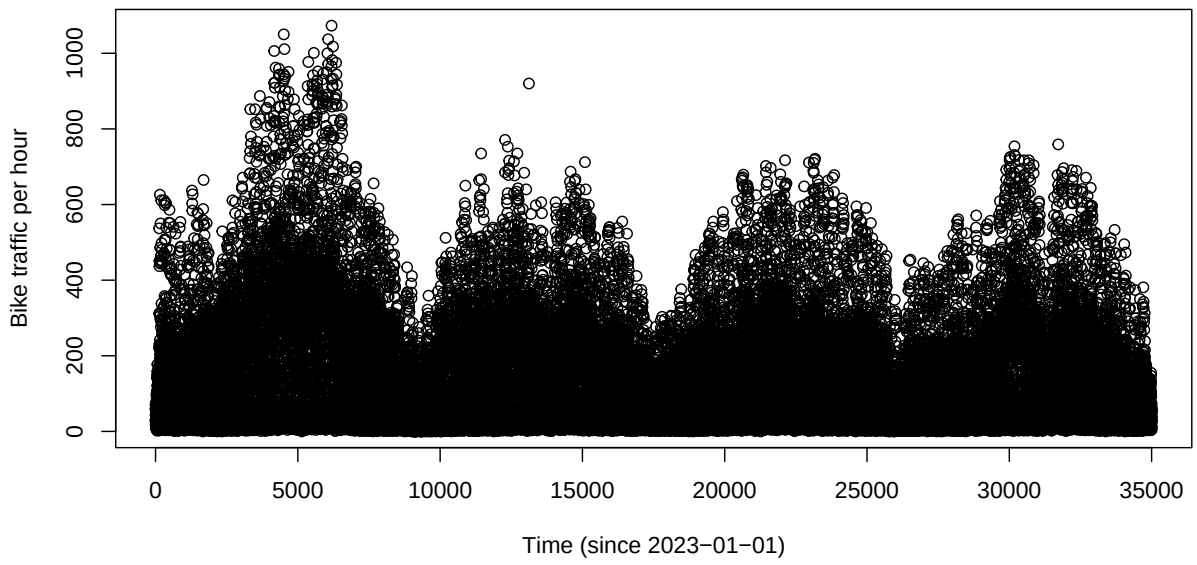
The test prediction is not too bad with an RMSE of 65.9. However, the performance when we only used the 2022 data for training, the test performance was better.

Extreme value theory

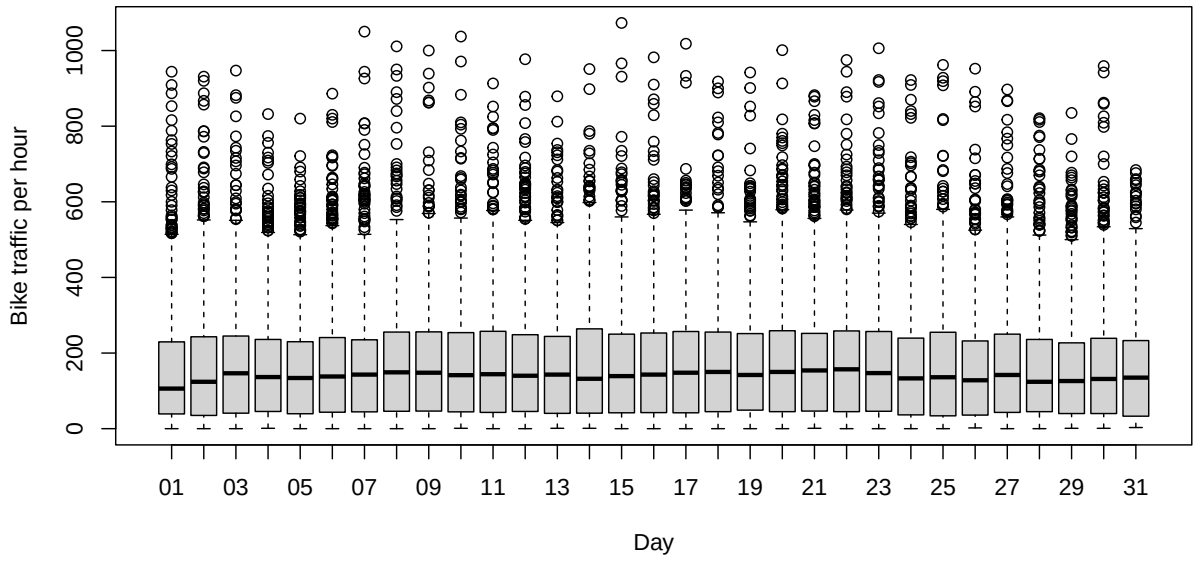
(not finalized, not used in project)

We are interested on the maximum amount of bicycles that are likely to pass the measurement station in one hour within the next 10 years. We are not interested in predicting the mean but in the maximum. This might be an important

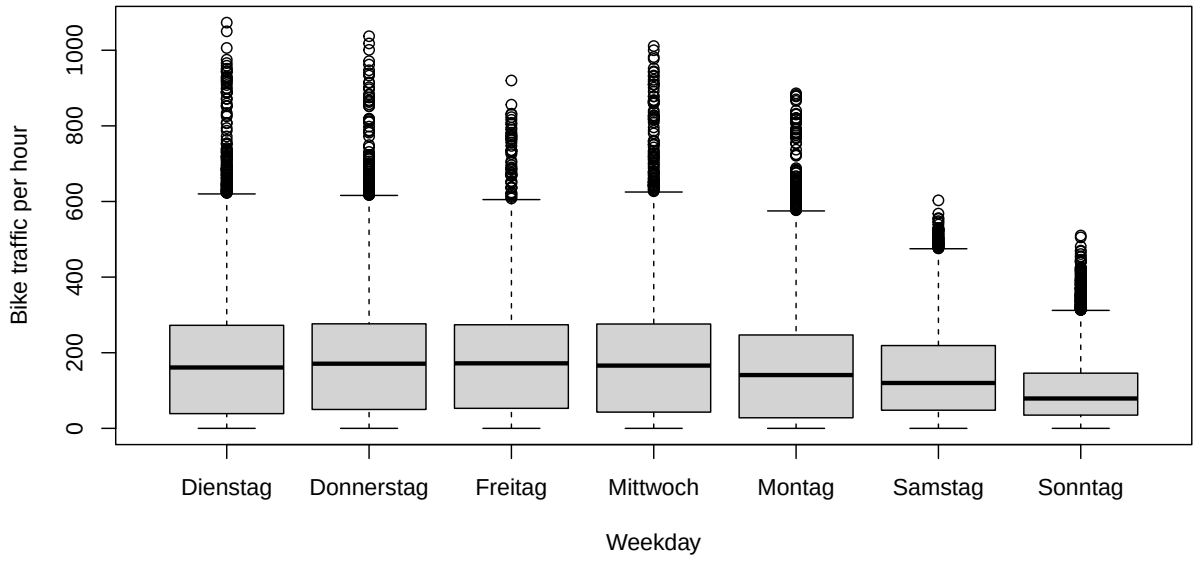
```r
plot(one_station_velo$bike_tot, xlab="Time (since 2023-01-01)", ylab="Bike traffic per hour")
```



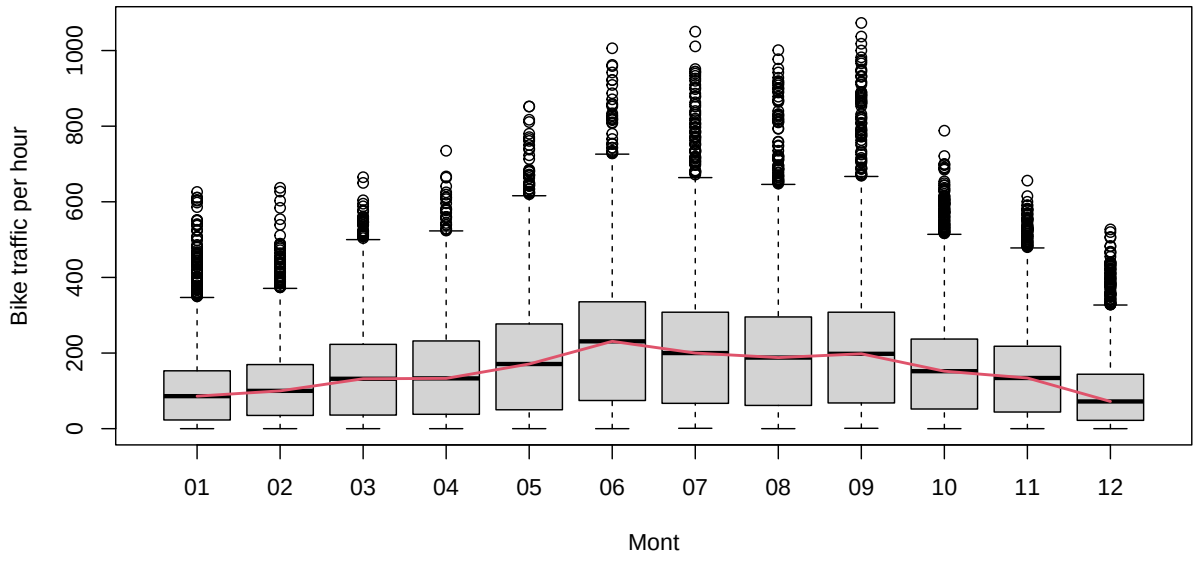

```r
plot(bike_tot ~ Day, data=one_station_velo, xlab="Day", ylab="Bike traffic per hour")
```




```
boxplot(bike_tot ~ Weekday, data=one_station_velo, xlab="Weekday", ylab="Bike traffic per hour")
```




```
plot(bike_tot ~ Month, data=one_station_velo, xlab="Mont", ylab="Bike traffic per hour")

lines(1:12, c(by(one_station_velo$bike_tot, one_station_velo$Month, median)), col=2, lwd=2)
```




```
# day and hour with the max traffic per hour in 2023
one_station_velo[which.max(one_station_velo$bike_tot),]
```

```
##       Standort       Date  Time         Datetime Hr...Hr. RainDur..min.
## 83961     2989 2020-09-15 18:00 2020-09-15 18:00    46.84             0
##       StrGlo..W.m2. T...C.   WD.... WVs..m.s. WVv..m.s.  p..hPa. Year
## 83961         12.49  26.34 141.5733      0.78 0.6633333 970.0467 2020
##       AnzBestWir                      bezeichnung bike_tot ped_tot Month Day
## 83961     434736 Langstrasse (Unterführung Nord)     1073       0    09  15
##        Weekday
## 83961 Dienstag
```