DATA 698 Final Research Project

Data Collection & Exploratory Analysis

Philip Tanofsky

2022-11-17

Data Collection & Preprocessing

Data Collection

Citi Bike provides individual bike trip data on a monthly basis available at https://ride.citibikenyc.com/system-data. The September 2022 bike trip data for New York City was downloaded and unzipped. The dataset contains 13 variables for each bike trip originating at a NYC-based docking station. A note on the system data page indicates trips taken by staff to service or inspect the system have been removed from the dataset. Also, any trips below 60 seconds have also been omitted. With this preprocessing by the data maintainers, the remaining trips are considered to be valid bike trips.

- **ride.id:** Unique identifier of the bike trip
- **rideable.type:** Factor variable - classic, electric, and docked
- **started.at:** Timestamp of trip departure
- **ended.at:** Timestamp of trip arrival
- **start.station.name:** Name of departure docking station
- **start.station.id:** Unique identifier of departure docking station
- **end.station.name:** Name of arrival docking station
- **end.station.id:** Unique identifier of arrival docking station
- **start.lat:** Latitude of departure location
- **start.lng:** Longitude of departure location
- **end.lat:** Latitude of arrival location
- **end.lng:** Longitude of arrival location
- **member.casual:** Factor variable for user type - member or casual

Based on the `started.at` and `ended.at` variables, four variables are derived for each bike trip.

- **day:** Day of the month
- **start.hour:** Hour of the trip departure
- **weekday:** Day of the week for the trip
- **trip.duration:** Duration of bike trip in minutes.

The NYC Open Data (free public data published by New York City agencies and partners) provides a GeoJSON file for the polygons defining each neighborhood in NYC according for the 2010 Neighborhood Tabulation Areas (NTAs). Each NTA is associated with one of the five NYC boroughs. (https://data.cityofnewyork.us/City-Government/2010-Neighborhood-Tabulation-Areas-NTAs-/cpf4-rkhq)

The elevation of each Citi Bike docking station is determined using the R library `elevatr` based on the latitude and longitude of each station. The elevation is defined in meters above sea level.

- **elevation: Units above sea level**
- **elev.units: Unit of measurement for elevation**

Data Preprocessing

Upon initial inspection of the 3,507,123 bike trips in September 2022, a total of 8,012 entries do not contain an `end.station.id` and `end.station.name` listed. Of that count, 3,838 do not contain an `end.lat` and `end.lng` values. These 8,012 without a defined destination docking station will be defined as 'Abandoned' meaning the user did not properly dock the bike. For this purpose, the `end.lat` and `end.lng` will be removed as the abandoned bikes temporarily remove a bike from the bikeshare system. Another rider cannot rent an abandoned bike until the bike is properly docked.

Evaluation of the `end.station.id` for the NYC based bike trips includes docking stations located in New Jersey. The Citi Bike bikeshare system does include docking stations in Jersey City and Hoboken. A number of bike trips end in New Jersey which does remove the bike from the NYC-based docking stations of which this research is focused.

Surplus Calculation

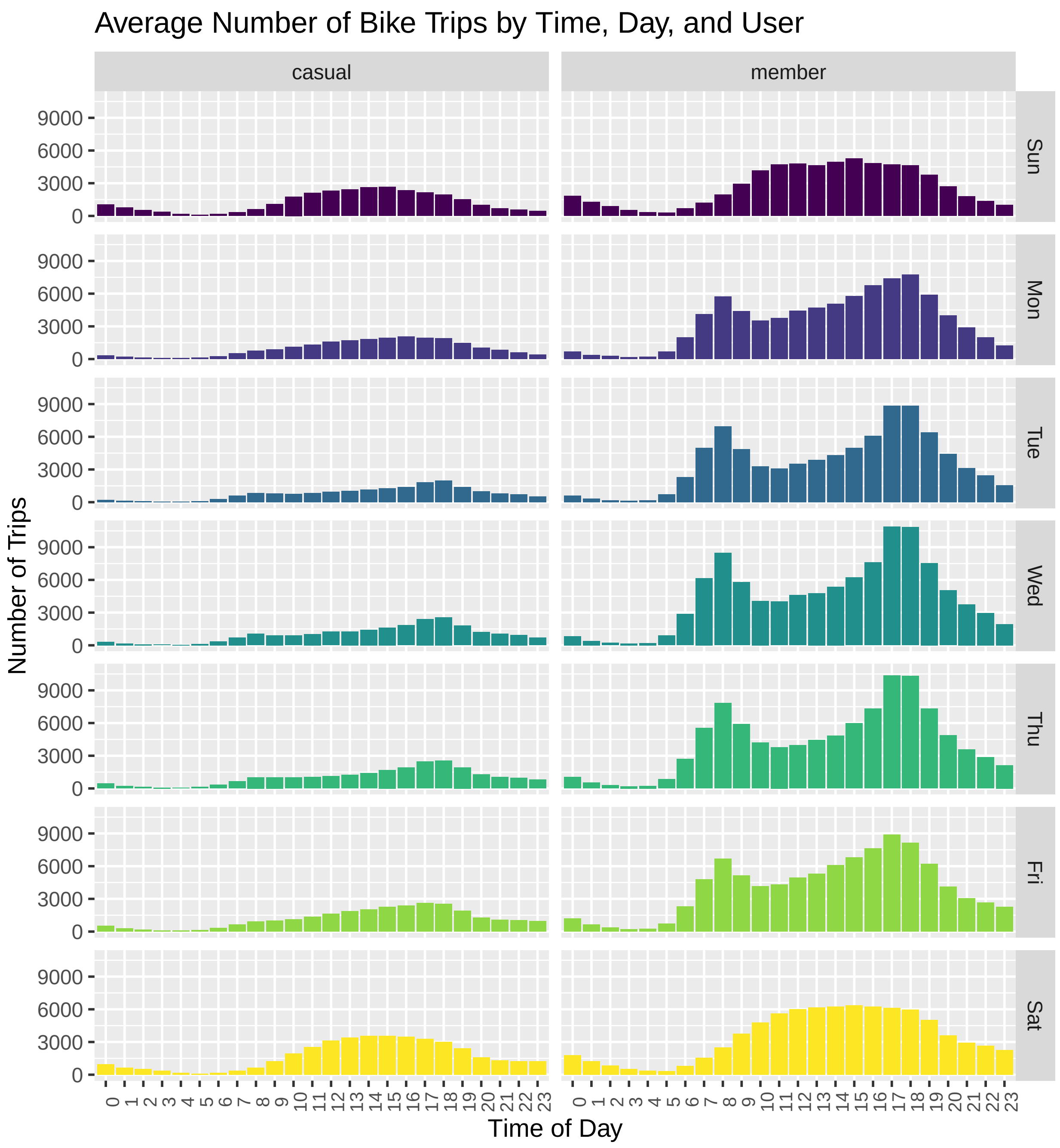
The individual bike trip information was sorted by timestamp and grouped by docking station for each 15-minute interval to count the number of bikes departing and the number bikes arriving. By subtracting the number of departures from the number of arrivals for each station for each interval, we are able to determine the running increase or decrease of bikes at the docking station. This total is defined as the variable `surplus`. A summation of the `surplus` for each docking station is calculated over the course of the month to determine which docking stations are more likely departure stations or arrival stations.

Data Exploration & Analysis

Exploratory data analysis is performed on the Citi Bike trips for September 2022 to evaluate the patterns of bike use and identify docking stations with surplus. First, the count of bike trips are assessed to find the high volume days of the week and time of day. Next, the duration of bike trips are evaluated to assess when bikes are individually likely to be unavailable longer. Finally, the surplus of bikes by docking station and borough are analyzed to determine which areas of the New York City are more prone to having lower bike availability.

Count of Bike Trips

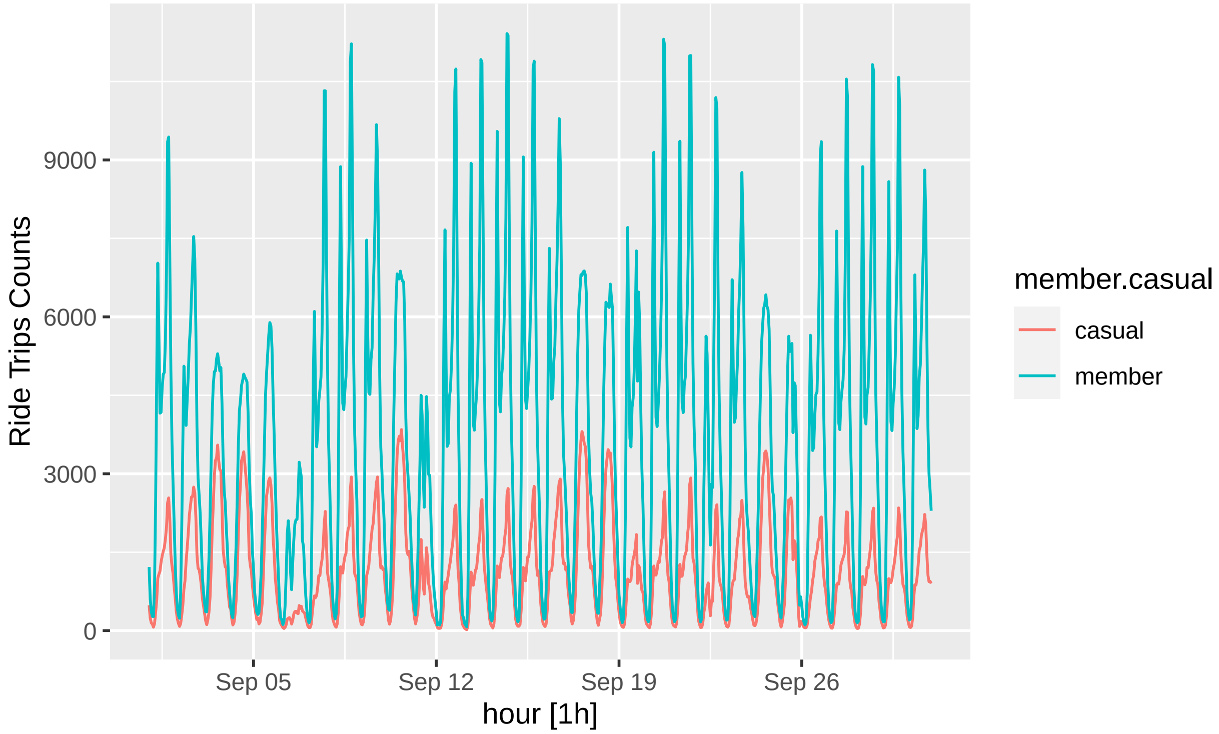
The count of bike trips are separated by user type - member and casual. With the separation by user type, two distinct patterns emerge of bike use over the course of the day and over the course of a week. The member trips are more likely to follow the workday pattern of spikes in the morning and evening as individuals are traveling to work or from work. The casual users show a pattern not necessarily indicative of the workday but instead of tourist or recreational use. The member trips account for an overall higher volume of trips.



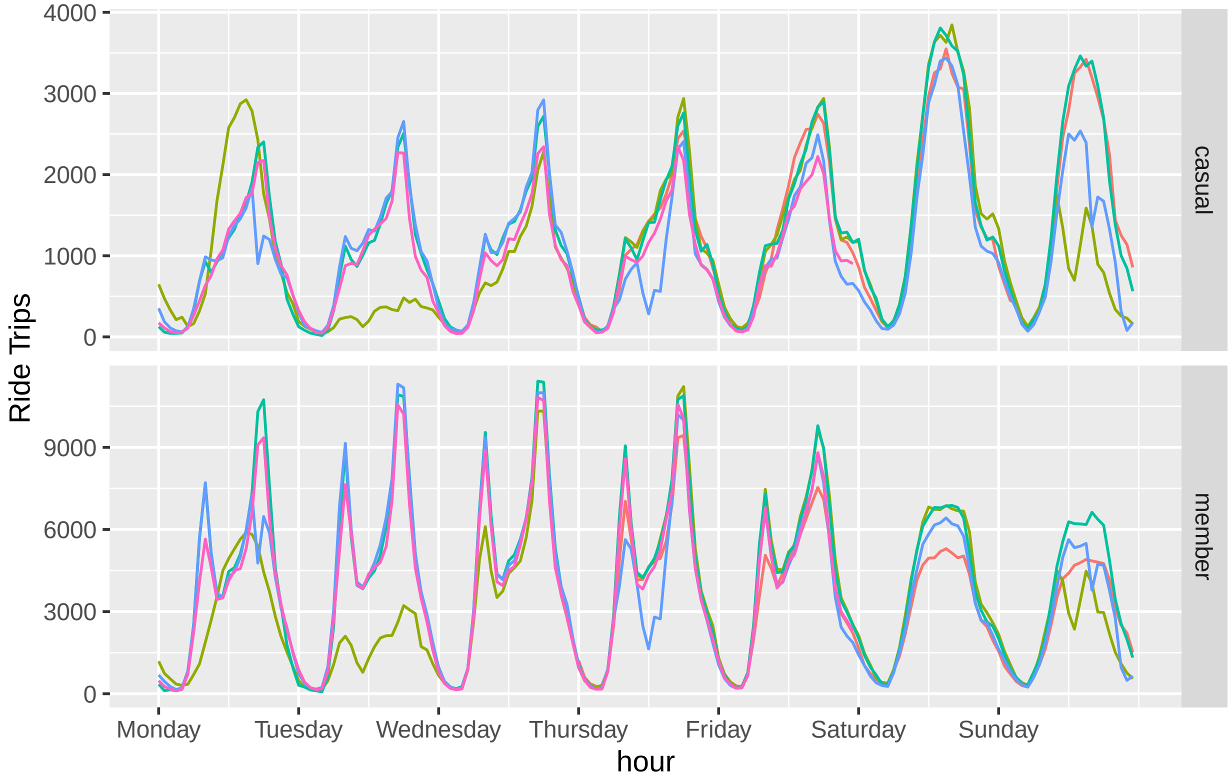
The member user counts show rush-hour spikes for members on weekdays whereas casual users do show higher counts around 5pm and 6pm on weekdays. The comparison of the two user groups also points to greater usage overall by member users. For everyday of the week, the member average member counts per hour are greater than the casual users. On weekend days, both user groups show a pattern indicative of recreational use with plateau use during the middle of the day and without distinct spikes found on the weekdays. Also, the higher usage of by member users throughout the middle of the day may indicate even if someone is a member the primary reason may not be transportation to and from work.

Ride Trips by Hour − Sept. 2022

Citi Bike NYC



Seasonal plot: Weekly Trip Counts for Sept. 2022



The time-series chart ‘Ride Trips by Hour’ confirms the higher usage by members with a clear pattern of weekday volumes corresponding to transportation for work purposes. The casual users follow a daily pattern with increases toward the end of the week - Thursday through Saturday.

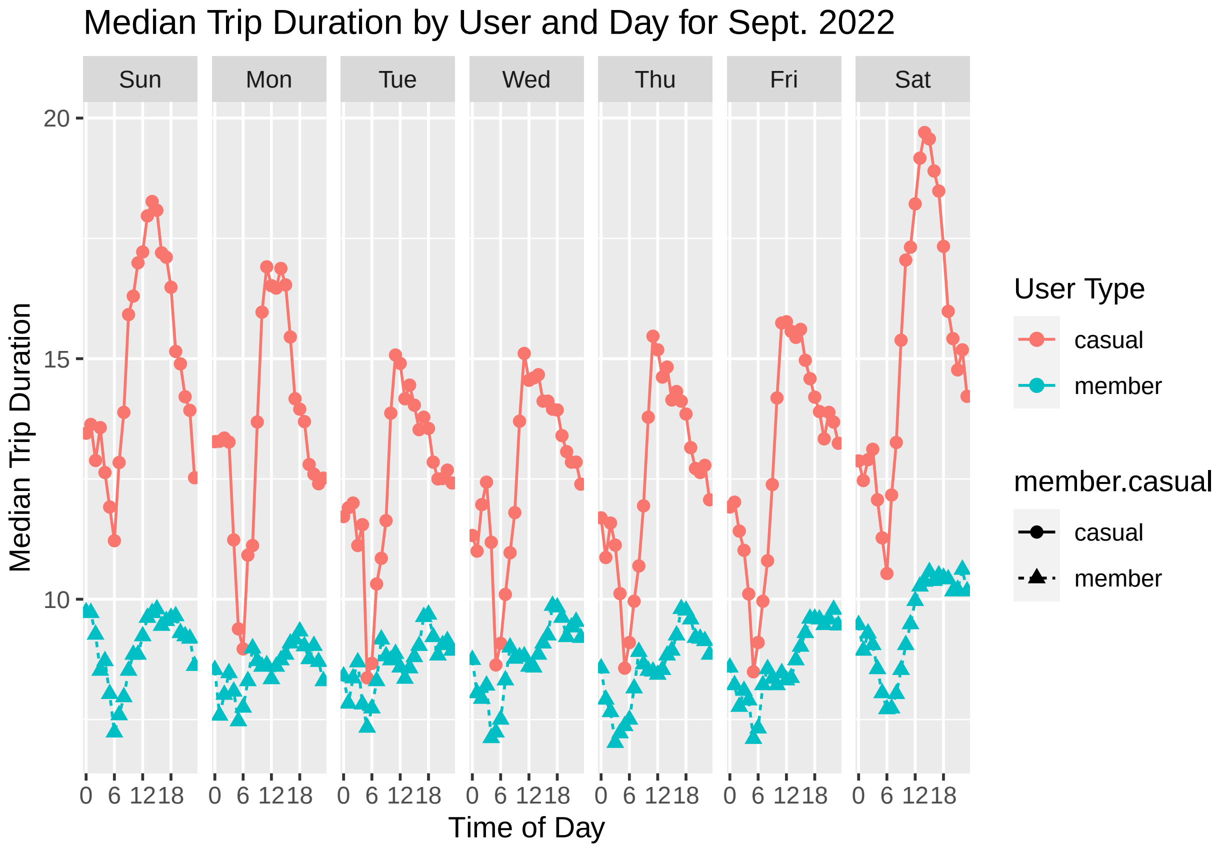
The weekly seasonal plot ‘Seasonal plot: Weekly Trip Counts’ denotes the same pattern week over week. Based on the plot, the bike trips show consistency from week to week based on member users utilizing the bikes for work and casual users renting the bikes for recreational purposes.

By count of trips per day of the week, Wednesday appears the highest volume day. Given the data comes from September 2022, we believe this observation is a result of pandemic return-to-office policies in which individuals more likely to return to office during the middle of the week instead of Monday or Friday for those with hybrid schedules.

Note: On Tuesday, Sept, 6 2022, the weather consisted of rain the entire day and thus a clear drop in use is evident in both user groups.

Duration of Bike Trips

Next, we evaluate the duration of bike trips to better understand longevity of individual bike unavailability along with reason for trip. As expected, the average bike trip for all users and all times are below 30 minutes as the rental defined time is 30 minutes with users incurring additional fees beyond the base time limit.



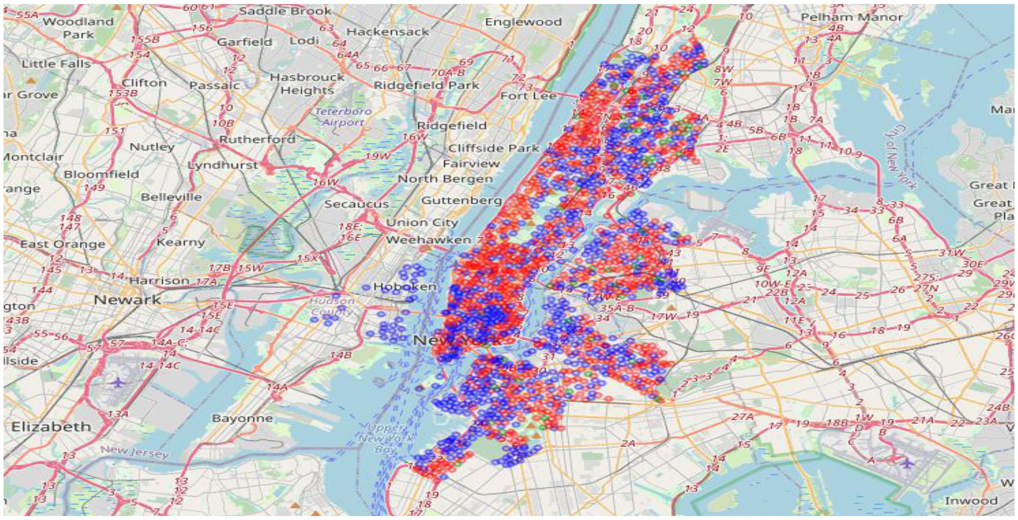
The chart of average trip duration by user type shows a clear distinction between the user types. Member users tend to average bike trips of 10 minutes or less throughout the week, whereas the casual users have a greater variance in average duration based on time of day and day of the week. The member users only average trips greater than 10 minutes during the afternoon and evening on Saturday while casual users typically average even longer trips of greater than 17 minutes on the weekend. Casual users tend to reach of peak of greater than 15 minutes on average everyday of the week, particularly around lunch on weekdays.

We observe that member users tend to have longer average trips during the morning and evening rush hours on weekdays with a slight deviation on Friday evening as the weekend starts. The afternoon and evening hours of Saturday and Sunday show longer trips for members as likely the result of recreational trips.

Docking Station Surplus

After constructing a matrix of bike arrivals and departures for each NYC-based docking station for every 15-minute interval of September 2022, we calculate the overall surplus or shortage of bikes. The researchers note, the shortage of bikes can logically not fall below zero without the introduction of rebalancing throughout the bikeshare system. The following analysis confirms the rebalancing of bikes across the docking stations occurs to ensure popular departure docking stations have bikes available despite the dearth of bike trip arrivals.

Overall Monthly Surplus/Shortage by Docking Station



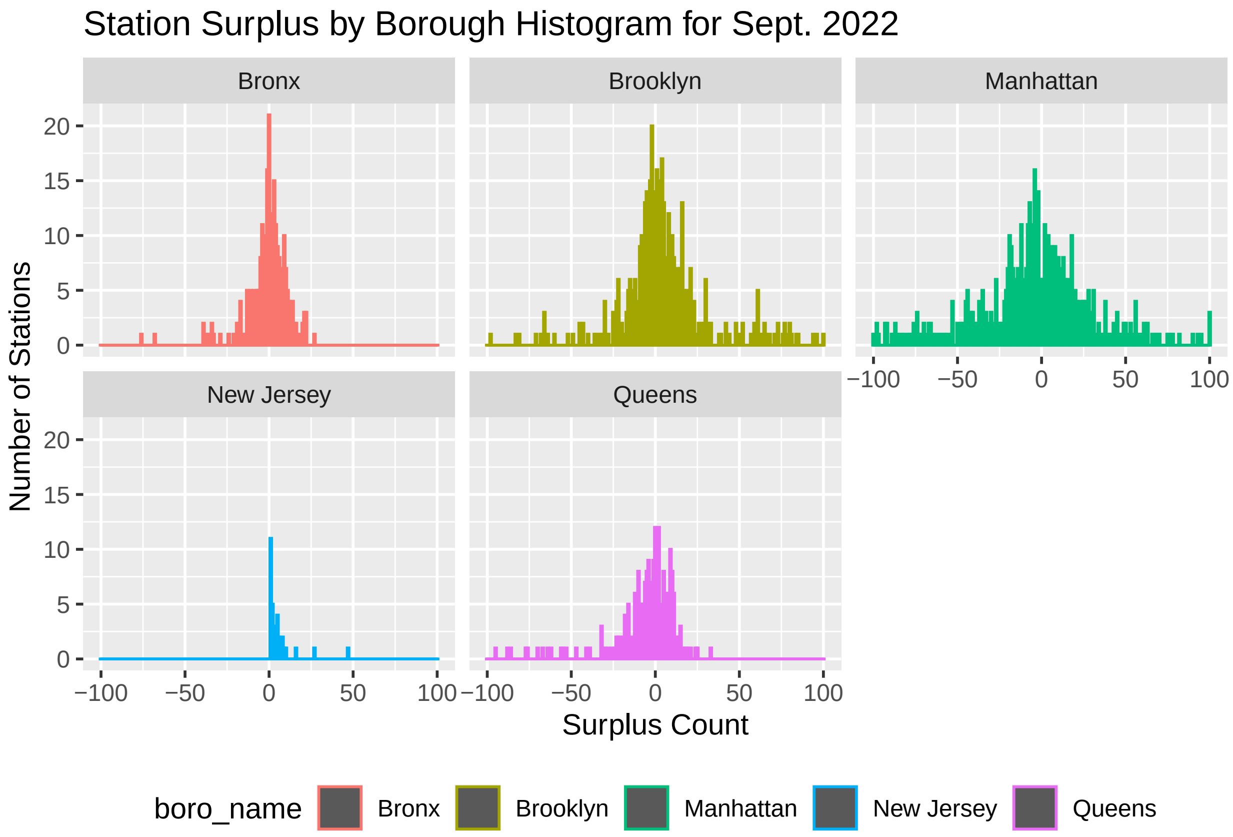
The plot of docking stations across the New York City maps the location of every docking station along with a color to indicate a net positive, negative or even amount of bikes for the given month. The blue stations are net positive, and red stations indicate net negative while green stations are even for the entire month. Of the 1656 docking stations represented, 799 ended with a surplus, 804 with a shortage, and 53 with an even count.

The plot does indicate several blue-colored docking stations in New Jersey. As the dataset does not contain any bike trips originating in NYC, a number of trips have an ending station in Jersey City or Hoboken. The New Jersey based stations are guaranteed to be blue based on that dataset without any departing trips from those docking stations.

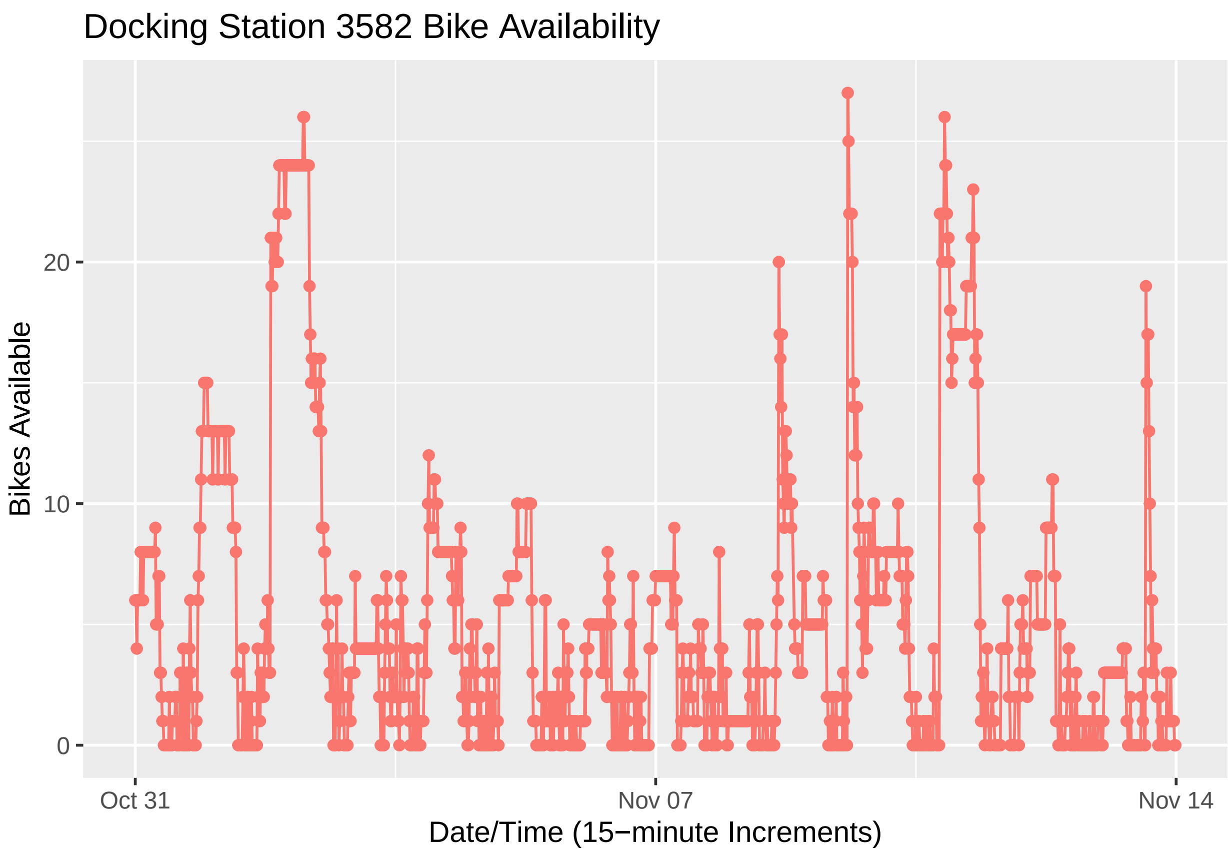
The dataset does encompass 472,920 valid combinations of departure and arrival docking stations. Of the valid combinations 15 of the top 20 combinations are the same departure and arrival docking stations, indicative of recreational bike trips. Of the aforementioned valid combinations, 471,313 denote travel between two different docking stations.

Surplus by Borough

With almost an even number of docking stations with a net shortage and a net surplus, we evaluate the shortage and surplus by borough.



The plot denotes a near normal distribution for the four boroughs included in the dataset - the Bronx, Brooklyn, Manhattan, and Queens. As noted previously, the dataset contains only trips originating in New York City, so the plot for New Jersey only indicates docking stations surpluses. The balanced distribution across the four NYC boroughs likely demonstrates the bike trips are contained within a borough. With average duration less than 10 minutes for member users and less than 20 minutes for casual users, the distance traveled for all bike trips is likely less than three miles. The 30-minute base rental rate dictates a limited travel distance which inhibits users from traveling across boroughs via Citi Bike.



The above shows there is rebalancing at this station but it's not consistent.

```
stations_attrs_trim_bk61 <- stations_attrs_trim %>%
  filter(ntaname == "Crown Heights North")

#(stations_attrs_trim_bk61)
#11 stations in this file
```

```
# Let's try the clustering
# https://www.r-bloggers.com/2019/06/hierarchical-clustering-for-location-based-strategy-using-r-for-e-
library(geosphere)

# Distance matrix for docking stations
# Result is in meters
dist_mat <- distm(stations_attrs_trim_bk61[9:10], stations_attrs_trim_bk61[9:10], fun=distHaversine)
dist_mat <- as.data.frame(dist_mat)
```

```
dist_mat[is.na(dist_mat)] <- 0

dMat <- as.dist(dist_mat)
```

```
dMat[1:10]
```

```
##  [1]  312.8990  367.9586  391.9150  740.9843  659.7183  577.1017  778.0415
##  [8]  941.7302 1643.7342 2536.0525
```

```
# Now for the clustering
hier_clust <- hclust(dMat, method = "complete")
# 400 meters is about a quarter of a mile (0.248548 miles)
stations_attrs_trim_bk61$cluster <- cutree(hier_clust, h=400)
```

```
stations_attrs_trim_bk61
```

```
##    short_name station_id elevation elev_units                        name
## 1     3919.07       3584     41.41     meters Eastern Pkwy & Franklin Ave
## 2     3960.01       3582     43.12     meters    Lincoln Pl & Classon Ave
## 3     3993.03       3579     34.01     meters   Sterling Pl & Bedford Ave
## 4     4033.06       3578     33.24     meters      Park Pl & Franklin Ave
## 5     4066.15       3571     26.48     meters     Bedford Ave & Bergen St
## 6     4074.03       3562     33.95     meters  Classon Ave & St Marks Ave
## 7     4107.05       3569     29.73     meters Franklin Ave & St Marks Ave
## 8     4107.13       3673     26.90     meters      Dean St & Franklin Ave
## 9     4148.07       3546     26.41     meters    Pacific St & Classon Ave
## 10    4205.06       3041     18.40     meters  Kingston Ave & Herkimer St
## 11    4228.02       3042     17.56     meters       Fulton St & Utica Ave
##    capacity             ntaname boro_name       lon      lat cluster
## 1        42 Crown Heights North  Brooklyn -73.95768 40.67078       1
## 2        28 Crown Heights North  Brooklyn -73.96090 40.67217       1
## 3        25 Crown Heights North  Brooklyn -73.95413 40.67269       2
## 4        21 Crown Heights North  Brooklyn -73.95649 40.67418       3
## 5        24 Crown Heights North  Brooklyn -73.95292 40.67637       3
## 6        27 Crown Heights North  Brooklyn -73.95961 40.67652       4
## 7        29 Crown Heights North  Brooklyn -73.95617 40.67583       3
## 8        21 Crown Heights North  Brooklyn -73.95564 40.67759       3
## 9        21 Crown Heights North  Brooklyn -73.95879 40.67919       4
## 10       24 Crown Heights North  Brooklyn -73.94143 40.67891       5
## 11       19 Crown Heights North  Brooklyn -73.92989 40.67943       6
```

```
stations_with_bike_avail <- read.csv('bike_avail_by_station_and_time.csv', row.names = 1, header= TRUE,
```

```
stations_with_bike_avail_long <- stations_with_bike_avail %>%
  pivot_longer(!timestamp, names_to = "station.id", values_to = "bikes.avail.count")
```

```
station_to_cluster <- stations_attrs_trim_bk61 %>%
  dplyr::select(station_id, cluster)

(station_to_cluster)
```

```
##    station_id cluster
## 1        3584       1
## 2        3582       1
## 3        3579       2
## 4        3578       3
## 5        3571       3
## 6        3562       4
## 7        3569       3
## 8        3673       3
## 9        3546       4
## 10       3041       5
## 11       3042       6
```

```
# Merge long df with counts with station.id and cluster to label clusters properly
stations_bike_avail_by_time <-
  merge(stations_with_bike_avail_long,
        station_to_cluster,
        by.x=c('station.id'),
        by.y=c('station_id'), all.x = TRUE)
```

```
stations_bike_avail_by_time_no_na <- stations_bike_avail_by_time %>%
  filter(!is.na(cluster))
```

```
# group by cluster ID for each timestamp
stations_bike_avail_by_time_no_na_group <- aggregate(bikes.avail.count ~ timestamp + cluster, data = sta
```

```
stations_bike_avail_by_time_no_na_group <- stations_bike_avail_by_time_no_na_group[order(stations_bike_a
```

```
# mutate to conver timestamp into day of the week, etc.
stations_bike_avail_by_time_no_na_group <- stations_bike_avail_by_time_no_na_group %>%
  mutate(time=as.ITime(ymd_hms(timestamp)),
         weekday = lubridate::wday(ymd_hms(timestamp),label=TRUE,abbr=TRUE))
```

```
stations_bike_avail_by_time_no_na_group$cluster <- as.factor(stations_bike_avail_by_time_no_na_group$clu
stations_bike_avail_by_time_no_na_group$time <- as.factor(stations_bike_avail_by_time_no_na_group$time)
stations_bike_avail_by_time_no_na_group$weekday <- as.factor(stations_bike_avail_by_time_no_na_group$wee
```

```
tab <- table(stations_bike_avail_by_time_no_na_group$bikes.avail.count)
tab
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
## 609 535 767 518 498 417 353 301 333 231 268 185 180 158 135 146 113 105 101 116
##  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
##  86 143 108 102  96  67  93  60  83  63  66  65  95  37  72  35  62  12  26  19
##  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59
##  13  17   9  17  14  21  23  14  18  32  42   9  17  16  12  10  22   8   4   9
##  60  61  62  63  64  65  66  67  68  69  70  71  72  74  75  76  77  78  79  80
##   8   2   7   9   8  11  10   1   7   6   1  14   3   4   3   3  12   3   4   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  99 101 104
##   2   8   8   3   7   5   8   5   4   6   4   3   7   1   6   2   5   4   2   1
## 105 107 108 109 110 111 113 114 115 116 117
##   4  13   2   5   1   1   3   4   5   4   3
```

```
# Below shows the range of values for `bikes.avail.count` with 35611 at zero and max of 438 with one oc
```

```
stations_bike_avail_by_time_no_timestamp <- subset(stations_bike_avail_by_time_no_na_group, select=-c(ti
```

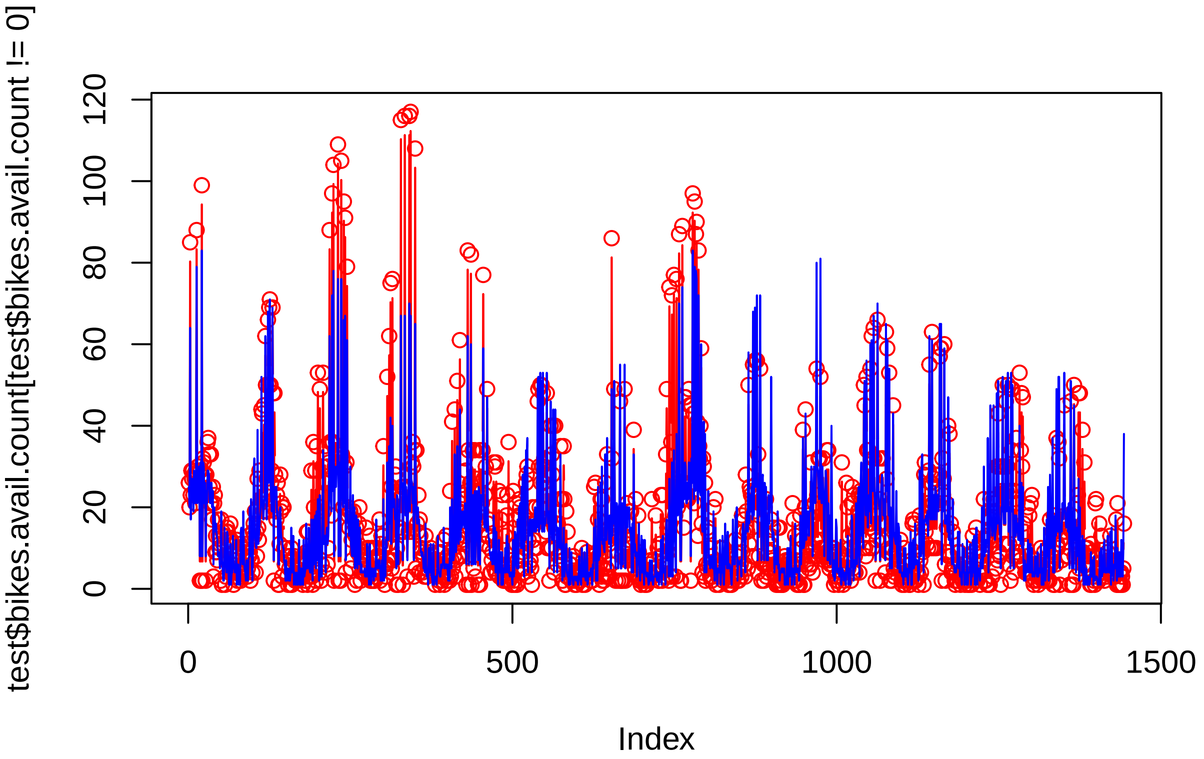
```r
#stations_bike_avail_by_time_no_timestamp
```

```r
# Start with data partition here.
set.seed(8675309)
index <- sample(2, nrow(stations_bike_avail_by_time_no_timestamp), replace = TRUE, p=c(0.8, 0.2))
train <- stations_bike_avail_by_time_no_timestamp[index==1,]
test <- stations_bike_avail_by_time_no_timestamp[index==2,]
```

```r
mod1 <- glm(bikes.avail.count ~ cluster + time + weekday, data = train, family ="poisson")
summary(mod1)
```

```
##
## Call:
## glm(formula = bikes.avail.count ~ cluster + time + weekday, family = "poisson",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.4597  -1.7847  -0.3791   1.1239   8.4367
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.027394   0.026155 115.750  < 2e-16 ***
## cluster2     -0.212459   0.012601 -16.861  < 2e-16 ***
## cluster3      0.952189   0.009949  95.707  < 2e-16 ***
## cluster4     -0.031446   0.011920  -2.638 0.008339 **
## cluster5     -0.354699   0.013160 -26.952  < 2e-16 ***
## cluster6     -1.356997   0.018381 -73.826  < 2e-16 ***
## time00:15:00 -0.034474   0.036464  -0.945 0.344434
## time00:30:00  0.065799   0.035442   1.857 0.063379 .
## time00:45:00  0.068291   0.035104   1.945 0.051726 .
## time01:00:00  0.088188   0.034829   2.532 0.011341 *
## time01:15:00  0.119915   0.035966   3.334 0.000856 ***
## time01:30:00  0.182831   0.035938   5.087 3.63e-07 ***
## time01:45:00  0.192717   0.033802   5.701 1.19e-08 ***
## time02:00:00  0.190189   0.033742   5.637 1.73e-08 ***
## time02:15:00  0.161024   0.034308   4.693 2.69e-06 ***
## time02:30:00  0.178540   0.034639   5.154 2.55e-07 ***
## time02:45:00  0.194827   0.033833   5.758 8.49e-09 ***
## time03:00:00  0.205296   0.033743   6.084 1.17e-09 ***
## time03:15:00  0.221456   0.035544   6.230 4.65e-10 ***
## time03:30:00  0.189802   0.034041   5.576 2.47e-08 ***
## time03:45:00  0.234113   0.034075   6.871 6.40e-12 ***
## time04:00:00  0.176451   0.033905   5.204 1.95e-07 ***
## time04:15:00  0.211681   0.033561   6.307 2.84e-10 ***
## time04:30:00  0.250103   0.032937   7.593 3.11e-14 ***
## time04:45:00  0.235313   0.035199   6.685 2.31e-11 ***
## time05:00:00  0.214041   0.033406   6.407 1.48e-10 ***
## time05:15:00  0.188136   0.034336   5.479 4.27e-08 ***
## time05:30:00  0.231422   0.033839   6.839 7.98e-12 ***
## time05:45:00  0.169399   0.036119   4.690 2.73e-06 ***
## time06:00:00  0.194328   0.033094   5.872 4.31e-09 ***
## time06:15:00  0.147418   0.035109   4.199 2.68e-05 ***
## time06:30:00  0.144038   0.034176   4.215 2.50e-05 ***
## time06:45:00  0.090176   0.035653   2.529 0.011431 *
## time07:00:00  0.036372   0.035495   1.025 0.305506
## time07:15:00  0.037213   0.035741   1.041 0.297792
## time07:30:00 -0.043794   0.036531  -1.199 0.230606
## time07:45:00 -0.086541   0.038090  -2.272 0.023086 *
## time08:00:00 -0.259651   0.039482  -6.577 4.82e-11 ***
## time08:15:00 -0.405077   0.039600 -10.229  < 2e-16 ***
## time08:30:00 -0.469173   0.041166 -11.397  < 2e-16 ***
## time08:45:00 -0.537492   0.041635 -12.910  < 2e-16 ***
## time09:00:00 -0.662938   0.041871 -15.833  < 2e-16 ***
## time09:15:00 -0.763519   0.044325 -17.225  < 2e-16 ***
## time09:30:00 -0.830695   0.045540 -18.241  < 2e-16 ***
## time09:45:00 -1.025892   0.052359 -19.594  < 2e-16 ***
## time10:00:00 -1.184834   0.053911 -21.978  < 2e-16 ***
## time10:15:00 -1.119755   0.052105 -21.490  < 2e-16 ***
## time10:30:00 -1.248594   0.059132 -21.115  < 2e-16 ***
## time10:45:00 -1.242307   0.054098 -22.964  < 2e-16 ***
## time11:00:00 -1.322015   0.058593 -22.563  < 2e-16 ***
## time11:15:00 -1.321669   0.056925 -23.218  < 2e-16 ***
## time11:30:00 -1.469130   0.060216 -24.398  < 2e-16 ***
## time11:45:00 -1.458215   0.060140 -24.247  < 2e-16 ***
## time12:00:00 -1.467967   0.062899 -23.338  < 2e-16 ***
## time12:15:00 -1.534490   0.060824 -25.228  < 2e-16 ***
## time12:30:00 -1.590254   0.060821 -26.146  < 2e-16 ***
## time12:45:00 -1.640441   0.063160 -25.973  < 2e-16 ***
## time13:00:00 -1.722559   0.065984 -26.106  < 2e-16 ***
## time13:15:00 -1.757796   0.071332 -24.642  < 2e-16 ***
## time13:30:00 -1.635196   0.064666 -25.287  < 2e-16 ***
## time13:45:00 -1.716339   0.065465 -26.218  < 2e-16 ***
## time14:00:00 -1.642404   0.063804 -25.741  < 2e-16 ***
## time14:15:00 -1.443973   0.058919 -24.508  < 2e-16 ***
## time14:30:00 -1.788588   0.067296 -26.578  < 2e-16 ***
## time14:45:00 -1.626425   0.064762 -25.114  < 2e-16 ***
## time15:00:00 -1.693216   0.062633 -27.034  < 2e-16 ***
## time15:15:00 -1.581433   0.065367 -24.193  < 2e-16 ***
## time15:30:00 -1.461387   0.060441 -24.179  < 2e-16 ***
## time15:45:00 -1.410174   0.058046 -24.294  < 2e-16 ***
## time16:00:00 -1.561763   0.062378 -25.037  < 2e-16 ***
## time16:15:00 -1.630680   0.063709 -25.596  < 2e-16 ***
## time16:30:00 -1.537787   0.063256 -24.311  < 2e-16 ***
## time16:45:00 -1.461326   0.058182 -25.117  < 2e-16 ***
## time17:00:00 -1.327290   0.057539 -23.068  < 2e-16 ***
## time17:15:00 -1.263322   0.057055 -22.142  < 2e-16 ***
## time17:30:00 -1.400349   0.056803 -24.653  < 2e-16 ***
## time17:45:00 -1.283768   0.054550 -23.534  < 2e-16 ***
## time18:00:00 -1.306517   0.055608 -23.495  < 2e-16 ***
## time18:15:00 -1.215287   0.052658 -23.079  < 2e-16 ***
## time18:30:00 -1.232980   0.053105 -23.218  < 2e-16 ***
## time18:45:00 -1.258770   0.052527 -23.964  < 2e-16 ***
## time19:00:00 -1.206720   0.056453 -21.376  < 2e-16 ***
## time19:15:00 -1.197097   0.052758 -22.690  < 2e-16 ***
## time19:30:00 -1.137081   0.053530 -21.242  < 2e-16 ***
## time19:45:00 -1.081515   0.052410 -20.636  < 2e-16 ***
## time20:00:00 -1.026292   0.050684 -20.249  < 2e-16 ***
## time20:15:00 -0.927748   0.048668 -19.063  < 2e-16 ***
## time20:30:00 -0.936490   0.051311 -18.251  < 2e-16 ***
## time20:45:00 -0.895753   0.048087 -18.628  < 2e-16 ***
## time21:00:00 -0.727182   0.045236 -16.075  < 2e-16 ***
## time21:15:00 -0.660247   0.042545 -15.519  < 2e-16 ***
## time21:30:00 -0.590980   0.042269 -13.981  < 2e-16 ***
## time21:45:00 -0.526844   0.044933 -11.725  < 2e-16 ***
## time22:00:00 -0.484159   0.040326 -12.006  < 2e-16 ***
## time22:15:00 -0.424325   0.040513 -10.474  < 2e-16 ***
## time22:30:00 -0.470320   0.042021 -11.192  < 2e-16 ***
## time22:45:00 -0.325715   0.041014  -7.942 2.00e-15 ***
## time23:00:00 -0.275513   0.037938  -7.262 3.81e-13 ***
## time23:15:00 -0.236609   0.039665  -5.965 2.44e-09 ***
## time23:30:00 -0.132927   0.037359  -3.558 0.000374 ***
## time23:45:00 -0.138450   0.037206  -3.721 0.000198 ***
## weekday.L    -0.111860   0.009594 -11.659  < 2e-16 ***
## weekday.Q    -0.358574   0.009429 -38.029  < 2e-16 ***
## weekday.C     0.098315   0.009126  10.773  < 2e-16 ***
## weekday^4    -0.105129   0.008656 -12.145  < 2e-16 ***
## weekday^5     0.096484   0.008650  11.154  < 2e-16 ***
## weekday^6    -0.113754   0.008367 -13.595  < 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: 108886  on 6463  degrees of freedom
## Residual deviance:  30206  on 6357  degrees of freedom
## AIC: 54262
##
## Number of Fisher Scoring iterations: 5
```

```r
pred <- predict.glm(mod1, newdata=test[test$bikes.avail.count != 0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(pred),col="blue")
```



```
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:fabletools':
##
##     accuracy
```

```
pred <- predict.glm(mod1, newdata = test, type = "response")
rmsemodelp <- ModelMetrics::rmse(test$bikes.avail.count,round(pred))
maemodelp <- mae(test$bikes.avail.count,round(pred))
```

```
rmsemodelp
```

```
## [1] 8.675188
```

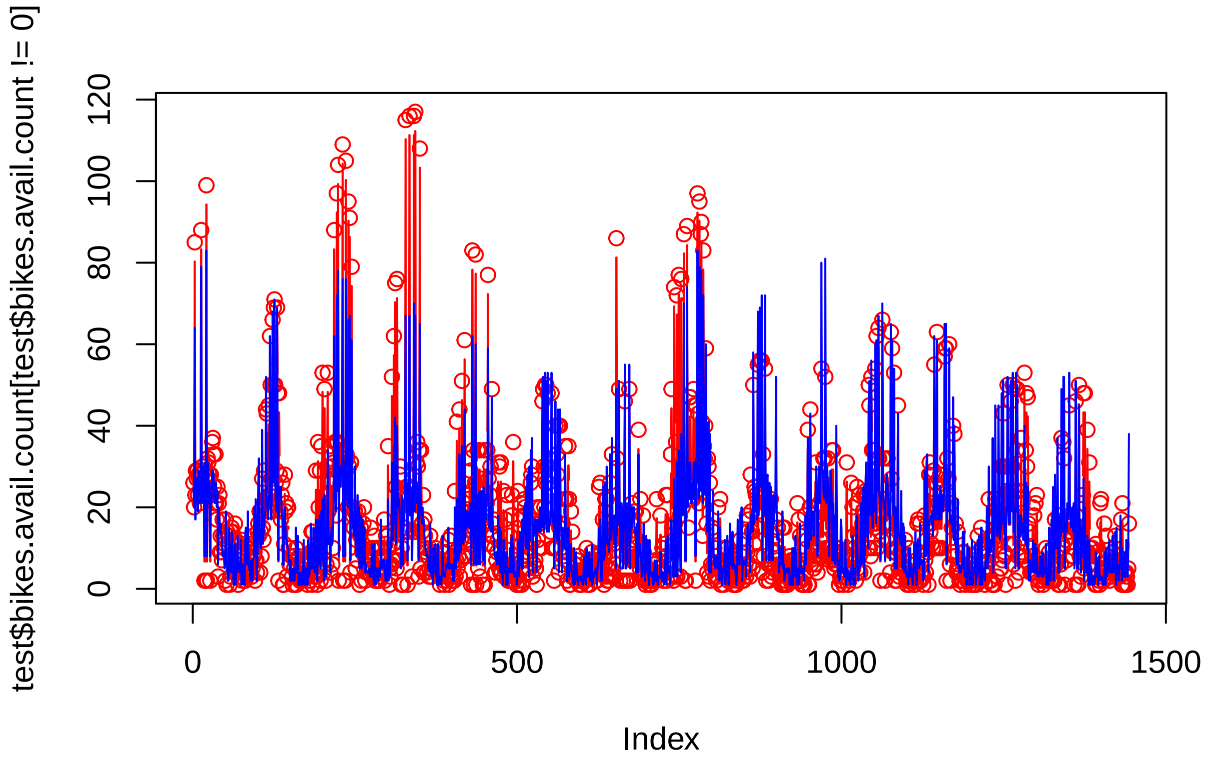
```
maemodelp
```

```
## [1] 5.961929
```

```r
mod2 <- glm(bikes.avail.count ~ cluster + time + weekday, data = train, family ="quasipoisson")
summary(mod2)
```

```
##
## Call:
## glm(formula = bikes.avail.count ~ cluster + time + weekday, family = "quasipoisson",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.4597  -1.7847  -0.3791   1.1239   8.4367
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.02739    0.05991  50.532  < 2e-16 ***
## cluster2     -0.21246    0.02886  -7.361 2.06e-13 ***
## cluster3      0.95219    0.02279  41.781  < 2e-16 ***
## cluster4     -0.03145    0.02731  -1.152 0.249506
## cluster5     -0.35470    0.03015 -11.766  < 2e-16 ***
## cluster6     -1.35700    0.04210 -32.229  < 2e-16 ***
## time00:15:00 -0.03447    0.08353  -0.413 0.679813
## time00:30:00  0.06580    0.08119   0.810 0.417696
## time00:45:00  0.06829    0.08041   0.849 0.395757
## time01:00:00  0.08819    0.07978   1.105 0.269042
## time01:15:00  0.11991    0.08239   1.456 0.145569
## time01:30:00  0.18283    0.08232   2.221 0.026392 *
## time01:45:00  0.19272    0.07743   2.489 0.012837 *
## time02:00:00  0.19019    0.07729   2.461 0.013893 *
## time02:15:00  0.16102    0.07859   2.049 0.040507 *
## time02:30:00  0.17854    0.07935   2.250 0.024474 *
## time02:45:00  0.19483    0.07750   2.514 0.011965 *
## time03:00:00  0.20530    0.07729   2.656 0.007925 **
## time03:15:00  0.22146    0.08142   2.720 0.006547 **
## time03:30:00  0.18980    0.07798   2.434 0.014956 *
## time03:45:00  0.23411    0.07805   2.999 0.002716 **
## time04:00:00  0.17645    0.07767   2.272 0.023123 *
## time04:15:00  0.21168    0.07688   2.753 0.005913 **
## time04:30:00  0.25010    0.07545   3.315 0.000922 ***
## time04:45:00  0.23531    0.08063   2.918 0.003530 **
## time05:00:00  0.21404    0.07652   2.797 0.005171 **
## time05:15:00  0.18814    0.07865   2.392 0.016786 *
## time05:30:00  0.23142    0.07751   2.986 0.002841 **
## time05:45:00  0.16940    0.08274   2.047 0.040654 *
## time06:00:00  0.19433    0.07581   2.563 0.010388 *
## time06:15:00  0.14742    0.08042   1.833 0.066845 .
## time06:30:00  0.14404    0.07828   1.840 0.065825 .
## time06:45:00  0.09018    0.08167   1.104 0.269566
## time07:00:00  0.03637    0.08131   0.447 0.654646
## time07:15:00  0.03721    0.08187   0.455 0.649460
## time07:30:00 -0.04379    0.08368  -0.523 0.600753
## time07:45:00 -0.08654    0.08725  -0.992 0.321305
## time08:00:00 -0.25965    0.09044  -2.871 0.004105 **
## time08:15:00 -0.40508    0.09071  -4.466 8.12e-06 ***
## time08:30:00 -0.46917    0.09430  -4.976 6.68e-07 ***
## time08:45:00 -0.53749    0.09537  -5.636 1.82e-08 ***
## time09:00:00 -0.66294    0.09591  -6.912 5.25e-12 ***
## time09:15:00 -0.76352    0.10153  -7.520 6.24e-14 ***
## time09:30:00 -0.83070    0.10432  -7.963 1.97e-15 ***
## time09:45:00 -1.02589    0.11994  -8.554  < 2e-16 ***
## time10:00:00 -1.18483    0.12349  -9.594  < 2e-16 ***
## time10:15:00 -1.11976    0.11935  -9.382  < 2e-16 ***
## time10:30:00 -1.24859    0.13545  -9.218  < 2e-16 ***
## time10:45:00 -1.24231    0.12392 -10.025  < 2e-16 ***
## time11:00:00 -1.32201    0.13422  -9.850  < 2e-16 ***
## time11:15:00 -1.32167    0.13039 -10.136  < 2e-16 ***
## time11:30:00 -1.46913    0.13793 -10.651  < 2e-16 ***
## time11:45:00 -1.45821    0.13776 -10.585  < 2e-16 ***
## time12:00:00 -1.46797    0.14408 -10.188  < 2e-16 ***
## time12:15:00 -1.53449    0.13933 -11.014  < 2e-16 ***
## time12:30:00 -1.59025    0.13932 -11.414  < 2e-16 ***
## time12:45:00 -1.64044    0.14468 -11.338  < 2e-16 ***
## time13:00:00 -1.72256    0.15115 -11.397  < 2e-16 ***
## time13:15:00 -1.75780    0.16340 -10.758  < 2e-16 ***
## time13:30:00 -1.63520    0.14813 -11.039  < 2e-16 ***
## time13:45:00 -1.71634    0.14996 -11.445  < 2e-16 ***
## time14:00:00 -1.64240    0.14615 -11.237  < 2e-16 ***
## time14:15:00 -1.44397    0.13496 -10.699  < 2e-16 ***
## time14:30:00 -1.78859    0.15415 -11.603  < 2e-16 ***
## time14:45:00 -1.62643    0.14835 -10.964  < 2e-16 ***
## time15:00:00 -1.69322    0.14347 -11.802  < 2e-16 ***
## time15:15:00 -1.58143    0.14973 -10.562  < 2e-16 ***
## time15:30:00 -1.46139    0.13845 -10.555  < 2e-16 ***
## time15:45:00 -1.41017    0.13296 -10.606  < 2e-16 ***
## time16:00:00 -1.56176    0.14289 -10.930  < 2e-16 ***
## time16:15:00 -1.63068    0.14594 -11.174  < 2e-16 ***
## time16:30:00 -1.53779    0.14490 -10.613  < 2e-16 ***
## time16:45:00 -1.46133    0.13327 -10.965  < 2e-16 ***
## time17:00:00 -1.32729    0.13180 -10.070  < 2e-16 ***
## time17:15:00 -1.26332    0.13069  -9.666  < 2e-16 ***
## time17:30:00 -1.40035    0.13012 -10.762  < 2e-16 ***
## time17:45:00 -1.28377    0.12495 -10.274  < 2e-16 ***
## time18:00:00 -1.30652    0.12738 -10.257  < 2e-16 ***
## time18:15:00 -1.21529    0.12062 -10.075  < 2e-16 ***
## time18:30:00 -1.23298    0.12164 -10.136  < 2e-16 ***
## time18:45:00 -1.25877    0.12032 -10.462  < 2e-16 ***
## time19:00:00 -1.20672    0.12931  -9.332  < 2e-16 ***
## time19:15:00 -1.19710    0.12085  -9.906  < 2e-16 ***
## time19:30:00 -1.13708    0.12262  -9.273  < 2e-16 ***
## time19:45:00 -1.08152    0.12005  -9.009  < 2e-16 ***
## time20:00:00 -1.02629    0.11610  -8.840  < 2e-16 ***
## time20:15:00 -0.92775    0.11148  -8.322  < 2e-16 ***
## time20:30:00 -0.93649    0.11754  -7.968 1.90e-15 ***
## time20:45:00 -0.89575    0.11015  -8.132 5.03e-16 ***
## time21:00:00 -0.72718    0.10362  -7.018 2.49e-12 ***
## time21:15:00 -0.66025    0.09746  -6.775 1.36e-11 ***
## time21:30:00 -0.59098    0.09682  -6.104 1.10e-09 ***
## time21:45:00 -0.52684    0.10293  -5.119 3.17e-07 ***
## time22:00:00 -0.48416    0.09237  -5.241 1.65e-07 ***
## time22:15:00 -0.42433    0.09280  -4.572 4.91e-06 ***
## time22:30:00 -0.47032    0.09626  -4.886 1.05e-06 ***
## time22:45:00 -0.32571    0.09395  -3.467 0.000530 ***
## time23:00:00 -0.27551    0.08690  -3.170 0.001530 **
## time23:15:00 -0.23661    0.09086  -2.604 0.009233 **
## time23:30:00 -0.13293    0.08558  -1.553 0.120399
## time23:45:00 -0.13845    0.08523  -1.624 0.104319
## weekday.L    -0.11186    0.02198  -5.090 3.68e-07 ***
## weekday.Q    -0.35857    0.02160 -16.602  < 2e-16 ***
## weekday.C     0.09832    0.02090   4.703 2.62e-06 ***
## weekday^4    -0.10513    0.01983  -5.302 1.18e-07 ***
## weekday^5     0.09648    0.01981   4.870 1.15e-06 ***
## weekday^6    -0.11375    0.01917  -5.935 3.09e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 5.247101)
##
##     Null deviance: 108886  on 6463  degrees of freedom
## Residual deviance:  30206  on 6357  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
predq <- predict.glm(mod2, newdata=test[test$bikes.avail.count != 0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predq),col="blue")
```



```
predq <- predict.glm(mod2,newdata=test, type = "response")
rmsemodelqp <- ModelMetrics::rmse(test$bikes.avail.count,round(predq))
maemodelqp <- mae(test$bikes.avail.count,round(predq))
rmsemodelqp
```

```
## [1] 8.675188
```

```
maemodelqp
```

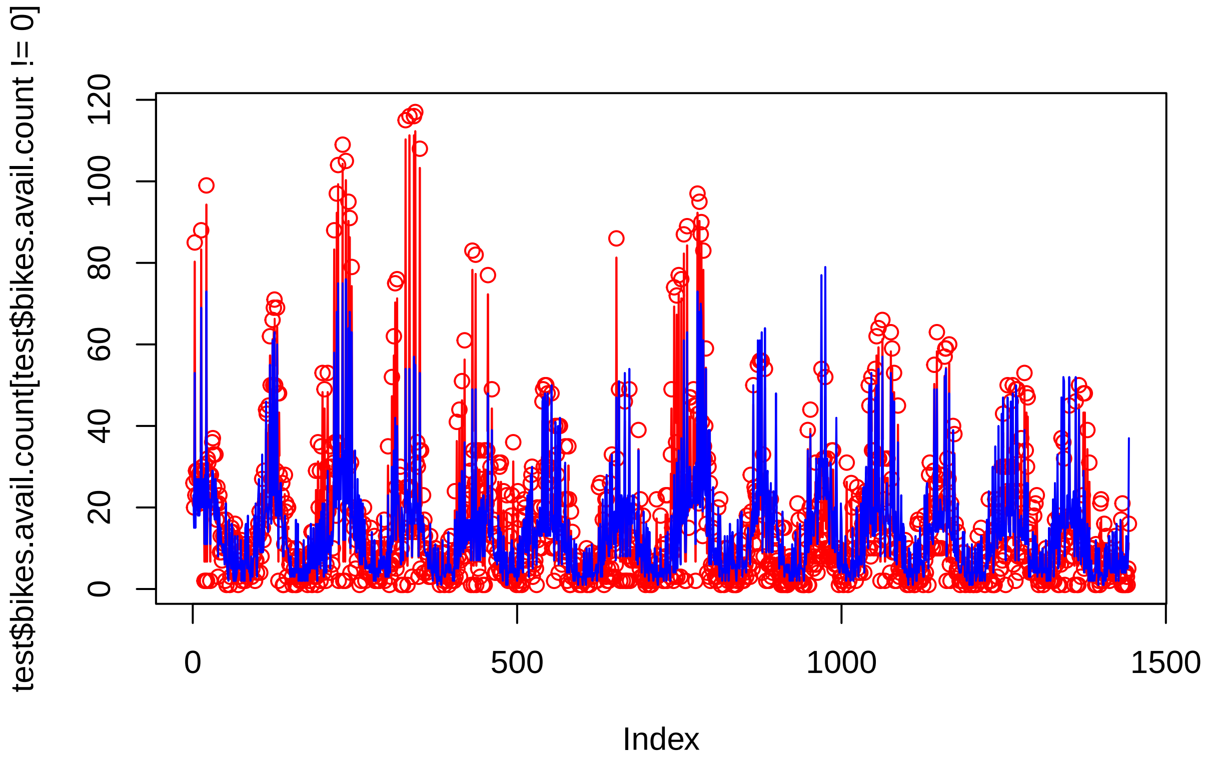
```
## [1] 5.961929
```

```
mod3 <- glm.nb(bikes.avail.count ~ cluster + time + weekday, data=train)
summary(mod3)
```

```
##
## Call:
## glm.nb(formula = bikes.avail.count ~ cluster + time + weekday,
##     data = train, init.theta = 2.367937065, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3310  -0.9605  -0.1628   0.4787   3.6554
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.00671    0.08701  34.555  < 2e-16 ***
## cluster2     -0.40320    0.03198 -12.608  < 2e-16 ***
## cluster3      0.83753    0.03060  27.374  < 2e-16 ***
## cluster4     -0.11517    0.03118  -3.694 0.000221 ***
## cluster5     -0.40381    0.03204 -12.605  < 2e-16 ***
## cluster6     -1.06582    0.03356 -31.761  < 2e-16 ***
## time00:15:00 -0.05266    0.12060  -0.437 0.662357
## time00:30:00  0.04711    0.12064   0.390 0.696192
## time00:45:00  0.08619    0.11959   0.721 0.471090
## time01:00:00  0.10920    0.11966   0.913 0.361460
## time01:15:00  0.11596    0.12155   0.954 0.340105
## time01:30:00  0.19890    0.12267   1.621 0.104925
## time01:45:00  0.20445    0.11788   1.734 0.082843 .
## time02:00:00  0.20732    0.11707   1.771 0.076565 .
## time02:15:00  0.18657    0.11867   1.572 0.115903
## time02:30:00  0.19985    0.11920   1.677 0.093614 .
## time02:45:00  0.19481    0.11902   1.637 0.101684
## time03:00:00  0.21920    0.11822   1.854 0.063708 .
## time03:15:00  0.23955    0.12399   1.932 0.053364 .
## time03:30:00  0.19317    0.11916   1.621 0.104987
## time03:45:00  0.24085    0.11946   2.016 0.043789 *
## time04:00:00  0.20733    0.11706   1.771 0.076539 .
## time04:15:00  0.21196    0.11782   1.799 0.072016 .
## time04:30:00  0.29853    0.11740   2.543 0.010997 *
## time04:45:00  0.25519    0.12094   2.110 0.034856 *
## time05:00:00  0.24225    0.11696   2.071 0.038332 *
## time05:15:00  0.19190    0.11886   1.615 0.106409
## time05:30:00  0.28422    0.11963   2.376 0.017512 *
## time05:45:00  0.22302    0.12696   1.757 0.078975 .
## time06:00:00  0.22720    0.11552   1.967 0.049219 *
## time06:15:00  0.17535    0.11934   1.469 0.141742
## time06:30:00  0.17246    0.11842   1.456 0.145314
## time06:45:00  0.08556    0.12183   0.702 0.482484
## time07:00:00  0.05704    0.11999   0.475 0.634514
## time07:15:00  0.11245    0.12081   0.931 0.351926
## time07:30:00  0.02966    0.12018   0.247 0.805089
## time07:45:00 -0.03846    0.12356  -0.311 0.755609
## time08:00:00 -0.21363    0.12287  -1.739 0.082085 .
## time08:15:00 -0.36263    0.12110  -2.994 0.002750 **
## time08:30:00 -0.37158    0.12047  -3.084 0.002040 **
## time08:45:00 -0.36401    0.12144  -2.997 0.002723 **
## time09:00:00 -0.48045    0.11910  -4.034 5.48e-05 ***
## time09:15:00 -0.57140    0.12097  -4.723 2.32e-06 ***
## time09:30:00 -0.65967    0.12166  -5.422 5.89e-08 ***
## time09:45:00 -0.80280    0.12797  -6.273 3.53e-10 ***
## time10:00:00 -0.96944    0.12752  -7.602 2.91e-14 ***
## time10:15:00 -0.93637    0.12746  -7.346 2.04e-13 ***
## time10:30:00 -1.00513    0.13366  -7.520 5.47e-14 ***
## time10:45:00 -1.01504    0.12589  -8.063 7.44e-16 ***
## time11:00:00 -1.08513    0.12845  -8.448  < 2e-16 ***
## time11:15:00 -1.06086    0.12763  -8.312  < 2e-16 ***
## time11:30:00 -1.24808    0.13045  -9.568  < 2e-16 ***
## time11:45:00 -1.19435    0.12886  -9.269  < 2e-16 ***
## time12:00:00 -1.16179    0.13371  -8.689  < 2e-16 ***
## time12:15:00 -1.27791    0.12801  -9.983  < 2e-16 ***
## time12:30:00 -1.32139    0.12818 -10.309  < 2e-16 ***
## time12:45:00 -1.39311    0.13118 -10.620  < 2e-16 ***
## time13:00:00 -1.49525    0.13166 -11.357  < 2e-16 ***
## time13:15:00 -1.47781    0.13384 -11.041  < 2e-16 ***
## time13:30:00 -1.43330    0.13253 -10.815  < 2e-16 ***
## time13:45:00 -1.55011    0.13199 -11.744  < 2e-16 ***
## time14:00:00 -1.45382    0.13063 -11.130  < 2e-16 ***
## time14:15:00 -1.27172    0.12998  -9.784  < 2e-16 ***
## time14:30:00 -1.64291    0.13343 -12.313  < 2e-16 ***
## time14:45:00 -1.46736    0.13371 -10.974  < 2e-16 ***
## time15:00:00 -1.53759    0.13001 -11.827  < 2e-16 ***
## time15:15:00 -1.39425    0.13388 -10.414  < 2e-16 ***
## time15:30:00 -1.30332    0.12980 -10.041  < 2e-16 ***
## time15:45:00 -1.22043    0.12956  -9.420  < 2e-16 ***
## time16:00:00 -1.37301    0.13165 -10.429  < 2e-16 ***
## time16:15:00 -1.41305    0.13198 -10.706  < 2e-16 ***
## time16:30:00 -1.33272    0.13293 -10.026  < 2e-16 ***
## time16:45:00 -1.26807    0.12677 -10.003  < 2e-16 ***
## time17:00:00 -1.19362    0.13014  -9.172  < 2e-16 ***
## time17:15:00 -1.14022    0.12942  -8.810  < 2e-16 ***
## time17:30:00 -1.30569    0.12888 -10.131  < 2e-16 ***
## time17:45:00 -1.13593    0.12615  -9.004  < 2e-16 ***
## time18:00:00 -1.17342    0.12644  -9.281  < 2e-16 ***
## time18:15:00 -1.11068    0.12573  -8.834  < 2e-16 ***
## time18:30:00 -1.10393    0.12774  -8.642  < 2e-16 ***
## time18:45:00 -1.12872    0.12517  -9.017  < 2e-16 ***
## time19:00:00 -1.12150    0.13347  -8.402  < 2e-16 ***
## time19:15:00 -1.10427    0.12639  -8.737  < 2e-16 ***
## time19:30:00 -1.06503    0.12954  -8.222  < 2e-16 ***
## time19:45:00 -0.98464    0.12713  -7.745 9.56e-15 ***
## time20:00:00 -0.95054    0.12729  -7.468 8.15e-14 ***
## time20:15:00 -0.89290    0.12514  -7.135 9.65e-13 ***
## time20:30:00 -0.87177    0.12867  -6.775 1.24e-11 ***
## time20:45:00 -0.87962    0.12424  -7.080 1.44e-12 ***
## time21:00:00 -0.72149    0.12311  -5.861 4.61e-09 ***
## time21:15:00 -0.65074    0.12111  -5.373 7.74e-08 ***
## time21:30:00 -0.56897    0.12135  -4.689 2.75e-06 ***
## time21:45:00 -0.54053    0.12615  -4.285 1.83e-05 ***
## time22:00:00 -0.46334    0.12111  -3.826 0.000130 ***
## time22:15:00 -0.38230    0.12077  -3.165 0.001548 **
## time22:30:00 -0.41799    0.12336  -3.388 0.000703 ***
## time22:45:00 -0.30783    0.12546  -2.454 0.014143 *
## time23:00:00 -0.25611    0.11957  -2.142 0.032207 *
## time23:15:00 -0.20372    0.12248  -1.663 0.096244 .
## time23:30:00 -0.12613    0.12089  -1.043 0.296802
## time23:45:00 -0.13453    0.11852  -1.135 0.256355
## weekday.L    -0.20230    0.02490  -8.125 4.48e-16 ***
## weekday.Q    -0.29302    0.02468 -11.873  < 2e-16 ***
## weekday.C     0.13893    0.02470   5.625 1.85e-08 ***
## weekday^4     0.02965    0.02436   1.217 0.223565
## weekday^5     0.07167    0.02452   2.924 0.003461 **
## weekday^6    -0.18619    0.02409  -7.730 1.07e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.3679) family taken to be 1)
##
##     Null deviance: 17955.0  on 6463  degrees of freedom
## Residual deviance:  7662.2  on 6357  degrees of freedom
## AIC: 41664
##
## Number of Fisher Scoring iterations: 1
##
##
##                Theta:  2.3679
##            Std. Err.:  0.0590
##
##  2 x log-likelihood:  -41447.5450
```

```
prednb <- predict.glm(mod3,newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(prednb),col="blue")
```



```
prednb <- predict.glm(mod3,newdata=test,type = "response")
rmsemodelnb <- ModelMetrics::rmse(test$bikes.avail.count,round(prednb))
maemodelnb <- mae(test$bikes.avail.count,round(prednb))
rmsemodelnb
```

```
## [1] 9.360348
```

```
maemodelnb
```

```
## [1] 6.381345
```

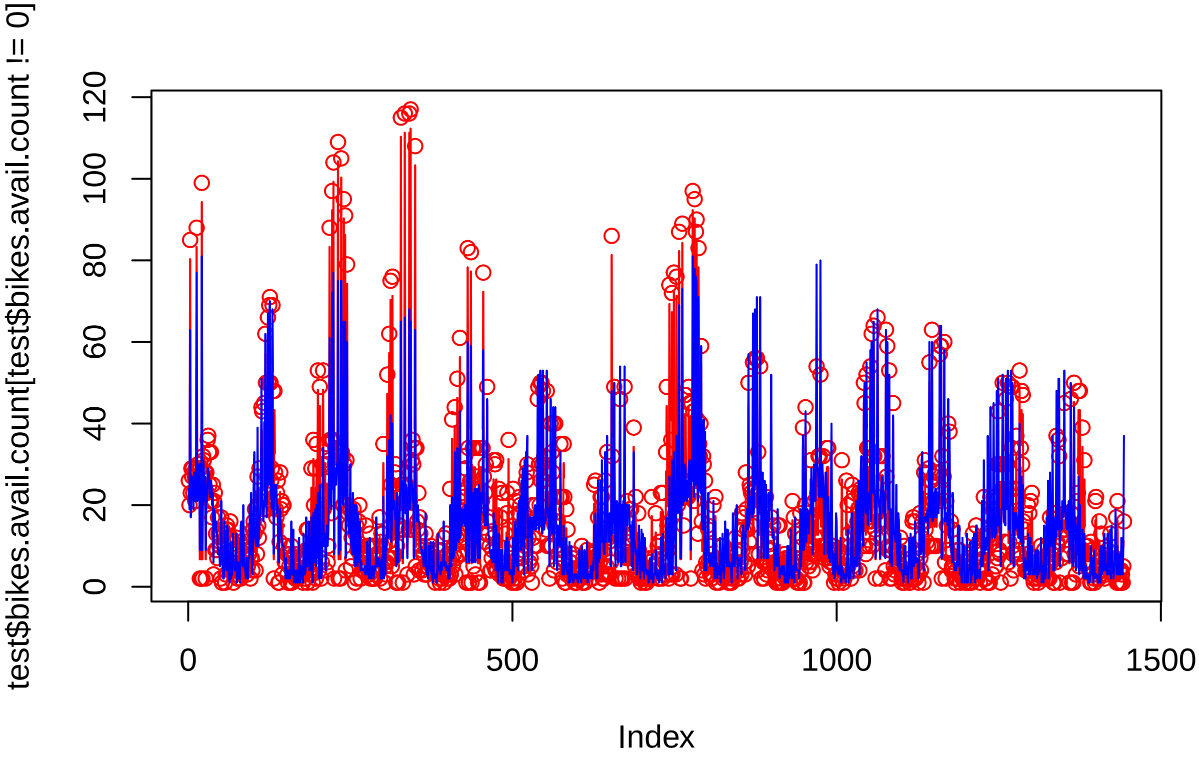
```
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
modelhp <- hurdle(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "poisson")
summary(modelhp)
```

```
## 
## Call:
## hurdle(formula = bikes.avail.count ~ cluster + time + weekday, data = train, 
##     dist = "poisson")
## 
## Pearson residuals:
##     Min      1Q  Median      3Q     Max 
## -5.3085 -1.2915 -0.2897  1.0314 12.0648 
## 
## Count model coefficients (truncated poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   3.032319   0.026165 115.890  < 2e-16 ***
## cluster2     -0.165194   0.012647 -13.062  < 2e-16 ***
## cluster3      0.940973   0.009966  94.421  < 2e-16 ***
## cluster4     -0.038326   0.011954  -3.206 0.001345 ** 
## cluster5     -0.338756   0.013234 -25.597  < 2e-16 ***
## cluster6     -1.275058   0.019381 -65.789  < 2e-16 ***
## time00:15:00 -0.031297   0.036479  -0.858 0.390916    
## time00:30:00  0.064292   0.035453   1.813 0.069765 .  
## time00:45:00  0.064900   0.035115   1.848 0.064574 .  
## time01:00:00  0.092765   0.034837   2.663 0.007750 ** 
## time01:15:00  0.125699   0.035976   3.494 0.000476 ***
## time01:30:00  0.172007   0.035947   4.785 1.71e-06 ***
## time01:45:00  0.194685   0.033809   5.758 8.50e-09 ***
## time02:00:00  0.188835   0.033750   5.595 2.21e-08 ***
## time02:15:00  0.159075   0.034317   4.635 3.56e-06 ***
## time02:30:00  0.181518   0.034647   5.239 1.61e-07 ***
## time02:45:00  0.200381   0.033840   5.921 3.19e-09 ***
## time03:00:00  0.202073   0.033751   5.987 2.14e-09 ***
## time03:15:00  0.214092   0.035552   6.022 1.72e-09 ***
## time03:30:00  0.191785   0.034051   5.632 1.78e-08 ***
## time03:45:00  0.232657   0.034083   6.826 8.72e-12 ***
## time04:00:00  0.176805   0.033912   5.214 1.85e-07 ***
## time04:15:00  0.210765   0.033569   6.279 3.42e-10 ***
## time04:30:00  0.251413   0.032944   7.632 2.32e-14 ***
## time04:45:00  0.229342   0.035205   6.514 7.30e-11 ***
## time05:00:00  0.206510   0.033414   6.180 6.40e-10 ***
## time05:15:00  0.180036   0.034346   5.242 1.59e-07 ***
## time05:30:00  0.223832   0.033846   6.613 3.76e-11 ***
## time05:45:00  0.160781   0.036127   4.450 8.57e-06 ***
## time06:00:00  0.188369   0.033103   5.690 1.27e-08 ***
## time06:15:00  0.134611   0.035120   3.833 0.000127 ***
## time06:30:00  0.134365   0.034186   3.930 8.48e-05 ***
## time06:45:00  0.083030   0.035665   2.328 0.019910 *
## time07:00:00  0.028459   0.035507   0.801 0.422852
## time07:15:00  0.028000   0.035754   0.783 0.433542
## time07:30:00 -0.051208   0.036544  -1.401 0.161132
## time07:45:00 -0.088061   0.038111  -2.311 0.020851 *
## time08:00:00 -0.265787   0.039519  -6.726 1.75e-11 ***
## time08:15:00 -0.407456   0.039650 -10.276  < 2e-16 ***
## time08:30:00 -0.459534   0.041237 -11.144  < 2e-16 ***
## time08:45:00 -0.527014   0.041704 -12.637  < 2e-16 ***
## time09:00:00 -0.655310   0.041982 -15.609  < 2e-16 ***
## time09:15:00 -0.751981   0.044487 -16.904  < 2e-16 ***
## time09:30:00 -0.782964   0.045744 -17.116  < 2e-16 ***
## time09:45:00 -1.014074   0.052848 -19.188  < 2e-16 ***
## time10:00:00 -1.126011   0.054601 -20.623  < 2e-16 ***
## time10:15:00 -1.063669   0.052533 -20.248  < 2e-16 ***
## time10:30:00 -1.136106   0.059931 -18.957  < 2e-16 ***
## time10:45:00 -1.098020   0.054716 -20.068  < 2e-16 ***
## time11:00:00 -1.221261   0.059661 -20.470  < 2e-16 ***
## time11:15:00 -1.203649   0.057867 -20.800  < 2e-16 ***
## time11:30:00 -1.365044   0.061459 -22.211  < 2e-16 ***
## time11:45:00 -1.406999   0.061567 -22.853  < 2e-16 ***
## time12:00:00 -1.358345   0.064343 -21.111  < 2e-16 ***
## time12:15:00 -1.450244   0.062524 -23.195  < 2e-16 ***
## time12:30:00 -1.523665   0.062702 -24.300  < 2e-16 ***
## time12:45:00 -1.581124   0.065238 -24.236  < 2e-16 ***
## time13:00:00 -1.682250   0.068771 -24.462  < 2e-16 ***
## time13:15:00 -1.728023   0.076029 -22.729  < 2e-16 ***
## time13:30:00 -1.589497   0.067065 -23.701  < 2e-16 ***
## time13:45:00 -1.669551   0.068207 -24.478  < 2e-16 ***
## time14:00:00 -1.578777   0.066050 -23.903  < 2e-16 ***
## time14:15:00 -1.367368   0.060173 -22.724  < 2e-16 ***
## time14:30:00 -1.648727   0.069970 -23.563  < 2e-16 ***
## time14:45:00 -1.529494   0.066780 -22.904  < 2e-16 ***
## time15:00:00 -1.611311   0.064723 -24.895  < 2e-16 ***
## time15:15:00 -1.540865   0.067750 -22.743  < 2e-16 ***
## time15:30:00 -1.404501   0.062133 -22.605  < 2e-16 ***
## time15:45:00 -1.399690   0.059361 -23.579  < 2e-16 ***
## time16:00:00 -1.466512   0.064176 -22.852  < 2e-16 ***
## time16:15:00 -1.473808   0.065508 -22.498  < 2e-16 ***
## time16:30:00 -1.482028   0.065061 -22.779  < 2e-16 ***
## time16:45:00 -1.385525   0.059528 -23.275  < 2e-16 ***
## time17:00:00 -1.292935   0.058441 -22.124  < 2e-16 ***
## time17:15:00 -1.151353   0.057822 -19.912  < 2e-16 ***
## time17:30:00 -1.324823   0.057633 -22.987  < 2e-16 ***
## time17:45:00 -1.228444   0.055275 -22.224  < 2e-16 ***
## time18:00:00 -1.251576   0.056518 -22.145  < 2e-16 ***
## time18:15:00 -1.189355   0.053238 -22.340  < 2e-16 ***
## time18:30:00 -1.214026   0.053666 -22.622  < 2e-16 ***
## time18:45:00 -1.230546   0.053112 -23.169  < 2e-16 ***
## time19:00:00 -1.133799   0.056902 -19.925  < 2e-16 ***
## time19:15:00 -1.147835   0.053299 -21.536  < 2e-16 ***
## time19:30:00 -1.098887   0.053977 -20.358  < 2e-16 ***
## time19:45:00 -1.033546   0.052754 -19.592  < 2e-16 ***
## time20:00:00 -0.995699   0.051006 -19.521  < 2e-16 ***
## time20:15:00 -0.876679   0.048935 -17.915  < 2e-16 ***
## time20:30:00 -0.906589   0.051645 -17.554  < 2e-16 ***
## time20:45:00 -0.798821   0.048308 -16.536  < 2e-16 ***
## time21:00:00 -0.661282   0.045330 -14.588  < 2e-16 ***
## time21:15:00 -0.630407   0.042636 -14.786  < 2e-16 ***
## time21:30:00 -0.564331   0.042334 -13.330  < 2e-16 ***
## time21:45:00 -0.487529   0.045010 -10.832  < 2e-16 ***
## time22:00:00 -0.471303   0.040377 -11.673  < 2e-16 ***
## time22:15:00 -0.411399   0.040556 -10.144  < 2e-16 ***
## time22:30:00 -0.469112   0.042068 -11.151  < 2e-16 ***
## time22:45:00 -0.317011   0.041046  -7.723 1.13e-14 ***
## time23:00:00 -0.256608   0.037959  -6.760 1.38e-11 ***
## time23:15:00 -0.241820   0.039701  -6.091 1.12e-09 ***
## time23:30:00 -0.134143   0.037378  -3.589 0.000332 ***
## time23:45:00 -0.142454   0.037227  -3.827 0.000130 ***
## weekday.L    -0.099087   0.009661 -10.256  < 2e-16 ***
## weekday.Q    -0.348296   0.009493 -36.690  < 2e-16 ***
## weekday.C     0.112187   0.009181  12.219  < 2e-16 ***
## weekday^4    -0.098859   0.008700 -11.363  < 2e-16 ***
## weekday^5     0.088399   0.008688  10.175  < 2e-16 ***
## weekday^6    -0.112348   0.008399 -13.376  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.103e+00  7.399e-01   5.545 2.94e-08 ***
## cluster2     -1.457e+00  1.765e-01  -8.254  < 2e-16 ***
## cluster3      1.196e+00  3.101e-01   3.857 0.000115 ***
## cluster4      3.590e-01  2.324e-01   1.545 0.122334
## cluster5     -6.270e-01  1.926e-01  -3.256 0.001130 **
## cluster6     -1.262e+00  1.790e-01  -7.053 1.76e-12 ***
## time00:15:00 -3.650e-01  9.416e-01  -0.388 0.698267
## time00:30:00  8.000e-02  1.026e+00   0.078 0.937874
## time00:45:00  1.609e-01  1.026e+00   0.157 0.875381
## time01:00:00 -4.779e-01  9.419e-01  -0.507 0.611865
## time01:15:00 -6.635e-01  8.968e-01  -0.740 0.459388
## time01:30:00  7.979e-01  1.247e+00   0.640 0.522157
## time01:45:00 -2.810e-01  9.413e-01  -0.299 0.765301
## time02:00:00  1.465e-01  1.025e+00   0.143 0.886332
## time02:15:00  1.549e-02  1.025e+00   0.015 0.987944
## time02:30:00 -5.894e-01  8.962e-01  -0.658 0.510722
## time02:45:00 -6.490e-01  8.966e-01  -0.724 0.469111
## time03:00:00  1.439e-01  1.026e+00   0.140 0.888481
## time03:15:00  6.390e-01  1.247e+00   0.512 0.608324
## time03:30:00 -3.828e-01  9.412e-01  -0.407 0.684182
## time03:45:00  2.637e-02  1.026e+00   0.026 0.979496
## time04:00:00 -3.286e-01  9.403e-01  -0.349 0.726763
## time04:15:00 -2.803e-01  9.405e-01  -0.298 0.765709
## time04:30:00 -3.585e-01  9.414e-01  -0.381 0.703342
## time04:45:00  1.010e-01  1.027e+00   0.098 0.921657
## time05:00:00  8.845e-01  1.245e+00   0.710 0.477513
## time05:15:00  8.609e-01  1.245e+00   0.691 0.489391
## time05:30:00  1.506e+01  1.250e+03   0.012 0.990392
## time05:45:00  1.504e+01  1.402e+03   0.011 0.991443
## time06:00:00  9.248e-01  1.245e+00   0.743 0.457547
## time06:15:00  1.512e+01  1.242e+03   0.012 0.990288
## time06:30:00  1.508e+01  1.215e+03   0.012 0.990092
## time06:45:00  1.495e+01  1.288e+03   0.012 0.990740
## time07:00:00  1.500e+01  1.248e+03   0.012 0.990415
## time07:15:00  1.509e+01  1.261e+03   0.012 0.990450
## time07:30:00  8.383e-01  1.246e+00   0.673 0.501074
## time07:45:00 -3.327e-03  1.027e+00  -0.003 0.997416
## time08:00:00  7.147e-01  1.247e+00   0.573 0.566448
## time08:15:00  2.326e-02  1.026e+00   0.023 0.981910
## time08:30:00  1.451e-01  1.026e+00   0.141 0.887508
## time08:45:00  8.121e-02  1.026e+00   0.079 0.936933
## time09:00:00  1.682e-01  1.025e+00   0.164 0.869636
## time09:15:00  1.282e-01  1.025e+00   0.125 0.900502
## time09:30:00 -8.222e-01  8.671e-01  -0.948 0.343050
## time09:45:00 -4.518e-02  1.027e+00  -0.044 0.964909
## time10:00:00 -1.237e+00  8.352e-01  -1.481 0.138622
## time10:15:00 -1.208e+00  8.501e-01  -1.421 0.155252
## time10:30:00 -1.390e+00  8.528e-01  -1.630 0.103096
## time10:45:00 -1.421e+00  8.238e-01  -1.725 0.084601 .
## time11:00:00 -1.689e+00  8.090e-01  -2.088 0.036800 *
## time11:15:00 -1.644e+00  8.161e-01  -2.015 0.043915 *
## time11:30:00 -1.857e+00  8.100e-01  -2.292 0.021901 *
## time11:45:00 -1.301e+00  8.352e-01  -1.557 0.119390
## time12:00:00 -1.608e+00  8.377e-01  -1.920 0.054880 .
## time12:15:00 -1.685e+00  8.086e-01  -2.084 0.037129 *
## time12:30:00 -1.418e+00  8.237e-01  -1.722 0.085138 .
## time12:45:00 -1.612e+00  8.168e-01  -1.974 0.048424 *
## time13:00:00 -1.536e+00  8.245e-01  -1.863 0.062496 .
## time13:15:00 -1.204e+00  8.342e-01  -1.443 0.149006
## time13:30:00 -1.301e+00  8.360e-01  -1.556 0.119690
## time13:45:00 -1.522e+00  8.241e-01  -1.847 0.064789 .
## time14:00:00 -1.714e+00  8.087e-01  -2.120 0.034016 *
## time14:15:00 -1.439e+00  8.351e-01  -1.723 0.084889 .
## time14:30:00 -2.072e+00  7.957e-01  -2.604 0.009228 **
## time14:45:00 -1.860e+00  8.053e-01  -2.309 0.020921 *
## time15:00:00 -1.685e+00  8.088e-01  -2.083 0.037224 *
## time15:15:00 -1.310e+00  8.351e-01  -1.569 0.116762
## time15:30:00 -1.211e+00  8.341e-01  -1.452 0.146580
## time15:45:00 -3.901e-01  9.427e-01  -0.414 0.679046
## time16:00:00 -1.629e+00  8.165e-01  -1.995 0.046009 *
## time16:15:00 -2.037e+00  8.007e-01  -2.544 0.010947 *
## time16:30:00 -1.355e+00  8.367e-01  -1.619 0.105401
## time16:45:00 -1.526e+00  8.146e-01  -1.874 0.060936 .
## time17:00:00 -1.301e+00  8.359e-01  -1.557 0.119553
## time17:15:00 -1.798e+00  8.093e-01  -2.222 0.026298 *
## time17:30:00 -1.837e+00  8.037e-01  -2.285 0.022284 *
## time17:45:00 -1.472e+00  8.233e-01  -1.788 0.073795 .
## time18:00:00 -1.406e+00  8.227e-01  -1.709 0.087470 .
## time18:15:00 -1.103e+00  8.482e-01  -1.301 0.193368
## time18:30:00 -7.505e-01  8.977e-01  -0.836 0.403175
## time18:45:00 -1.065e+00  8.480e-01  -1.255 0.209320
## time19:00:00 -1.992e+00  8.126e-01  -2.452 0.014223 *
## time19:15:00 -1.464e+00  8.237e-01  -1.777 0.075511 .
## time19:30:00 -1.215e+00  8.508e-01  -1.428 0.153252
## time19:45:00 -1.311e+00  8.352e-01  -1.569 0.116538
## time20:00:00 -9.519e-01  8.684e-01  -1.096 0.272972
## time20:15:00 -1.266e+00  8.346e-01  -1.517 0.129308
## time20:30:00 -7.831e-01  8.966e-01  -0.873 0.382481
## time20:45:00 -1.608e+00  8.078e-01  -1.990 0.046572 *
## time21:00:00 -1.554e+00  8.150e-01  -1.907 0.056484 .
## time21:15:00 -1.000e+00  8.474e-01  -1.181 0.237748
## time21:30:00 -8.520e-01  8.675e-01  -0.982 0.326033
## time21:45:00 -9.380e-01  8.688e-01  -1.080 0.280300
## time22:00:00 -6.949e-01  8.962e-01  -0.775 0.438159
## time22:15:00 -6.162e-01  8.955e-01  -0.688 0.491362
## time22:30:00 -3.511e-01  9.419e-01  -0.373 0.709351
## time22:45:00 -8.198e-01  8.981e-01  -0.913 0.361334
## time23:00:00 -1.131e+00  8.480e-01  -1.334 0.182219
## time23:15:00  1.255e-01  1.026e+00   0.122 0.902669
## time23:30:00  7.004e-02  1.026e+00   0.068 0.945568
## time23:45:00  2.134e-01  1.025e+00   0.208 0.834999
## weekday.L    -3.597e-01  1.309e-01  -2.748 0.006005 **
## weekday.Q    -3.566e-01  1.331e-01  -2.679 0.007391 **
## weekday.C    -3.785e-01  1.340e-01  -2.825 0.004730 **
## weekday^4    -3.751e-02  1.389e-01  -0.270 0.787090
## weekday^5     4.413e-01  1.368e-01   3.225 0.001258 **
## weekday^6    -2.440e-01  1.411e-01  -1.729 0.083847 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Number of iterations in BFGS optimization: 89
## Log-likelihood: -2.598e+04 on 214 Df
```

```
predhp <- predict(modelhp, newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predhp),col="blue")
```



```
predhp <- predict(modelhp, newdata=test, type = "response")
rmsemodelhp<-ModelMetrics::rmse(test$bikes.avail.count,round(predhp))
maemodelhp<-mae(test$bikes.avail.count,round(predhp))
rmsemodelhp
```

```
## [1] 8.733759
```

```
maemodelhp
```

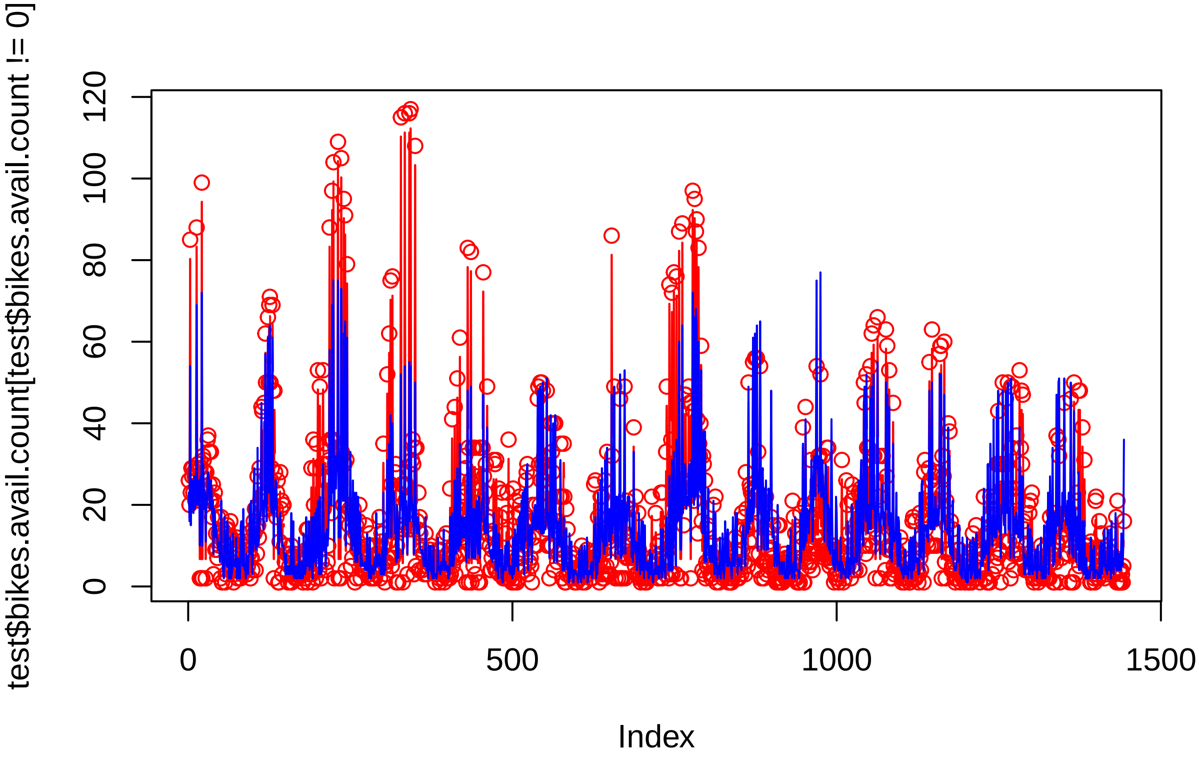
```
## [1] 5.965102
```

```
modelhnb <- hurdle(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "negbin")
summary(modelhnb)
```

```
##
## Call:
## hurdle(formula = bikes.avail.count ~ cluster + time + weekday, data = train,
##     dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.5700 -0.7792 -0.1689  0.5510  7.2111
##
## Count model coefficients (truncated negbin with log link):
```

```
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   3.0174993  0.0804254  37.519  < 2e-16 ***
## cluster2     -0.3047108  0.0311197  -9.792  < 2e-16 ***
## cluster3      0.8330154  0.0286641  29.061  < 2e-16 ***
## cluster4     -0.1276793  0.0295442  -4.322 1.55e-05 ***
## cluster5     -0.3879340  0.0308451 -12.577  < 2e-16 ***
## cluster6     -1.0720501  0.0353344 -30.340  < 2e-16 ***
## time00:15:00 -0.0428373  0.1116628  -0.384 0.701252    
## time00:30:00  0.0454833  0.1112047   0.409 0.682536    
## time00:45:00  0.0758959  0.1103040   0.688 0.491414    
## time01:00:00  0.1221074  0.1107341   1.103 0.270154    
## time01:15:00  0.1378947  0.1129609   1.221 0.222189    
## time01:30:00  0.1735002  0.1129833   1.536 0.124630    
## time01:45:00  0.2132685  0.1089771   1.957 0.050347 .  
## time02:00:00  0.1997754  0.1079231   1.851 0.064157 .  
## time02:15:00  0.1799692  0.1094890   1.644 0.100234    
## time02:30:00  0.2152050  0.1106563   1.945 0.051798 .  
## time02:45:00  0.2142472  0.1103979   1.941 0.052297 .  
## time03:00:00  0.2110402  0.1089695   1.937 0.052783 .  
## time03:15:00  0.2195392  0.1140629   1.925 0.054264 .  
## time03:30:00  0.2026845  0.1101899   1.839 0.065855 .  
## time03:45:00  0.2379523  0.1101206   2.161 0.030708 *  
## time04:00:00  0.2147585  0.1082176   1.985 0.047199 *  
## time04:15:00  0.2133109  0.1088792   1.959 0.050095 .  
## time04:30:00  0.3050520  0.1085299   2.811 0.004942 ** 
## time04:45:00  0.2456450  0.1116079   2.201 0.027739 *  
## time05:00:00  0.2181216  0.1075462   2.028 0.042543 *  
## time05:15:00  0.1703844  0.1092604   1.559 0.118894    
## time05:30:00  0.2516759  0.1096724   2.295 0.021745 *  
## time05:45:00  0.1881834  0.1163259   1.618 0.105722    
## time06:00:00  0.2051248  0.1062425   1.931 0.053517 .  
## time06:15:00  0.1369908  0.1095460   1.251 0.211105    
## time06:30:00  0.1347056  0.1086527   1.240 0.215056    
## time06:45:00  0.0595408  0.1117258   0.533 0.594090    
## time07:00:00  0.0271617  0.1100732   0.247 0.805094    
## time07:15:00  0.0735691  0.1109933   0.663 0.507443    
## time07:30:00 -0.0002171  0.1108561  -0.002 0.998437    
## time07:45:00 -0.0485497  0.1145374  -0.424 0.671656    
## time08:00:00 -0.2346776  0.1136346  -2.065 0.038905 *  
## time08:15:00 -0.3750561  0.1124459  -3.335 0.000852 ***
## time08:30:00 -0.3857672  0.1123635  -3.433 0.000596 ***
## time08:45:00 -0.3793241  0.1135464  -3.341 0.000836 ***
## time09:00:00 -0.5029641  0.1116288  -4.506 6.62e-06 ***
## time09:15:00 -0.5975968  0.1138268  -5.250 1.52e-07 ***
## time09:30:00 -0.6405830  0.1157429  -5.535 3.12e-08 ***
## time09:45:00 -0.8388757  0.1224236  -6.852 7.27e-12 ***
## time10:00:00 -0.9411324  0.1246457  -7.550 4.34e-14 ***
## time10:15:00 -0.9095876  0.1233676  -7.373 1.67e-13 ***
## time10:30:00 -0.9613116  0.1312287  -7.325 2.38e-13 ***
## time10:45:00 -0.9604354  0.1230958  -7.802 6.08e-15 ***
## time11:00:00 -1.0066723  0.1282735  -7.848 4.23e-15 ***
## time11:15:00 -0.9897119  0.1269114  -7.798 6.27e-15 ***
## time11:30:00 -1.1738404  0.1307269  -8.979  < 2e-16 ***
## time11:45:00 -1.1874938  0.1282351  -9.260  < 2e-16 ***
## time12:00:00 -1.1122780  0.1344146  -8.275  < 2e-16 ***
## time12:15:00 -1.2294673  0.1288001  -9.546  < 2e-16 ***
## time12:30:00 -1.3162001  0.1287483 -10.223  < 2e-16 ***
## time12:45:00 -1.3774347  0.1323188 -10.410  < 2e-16 ***
## time13:00:00 -1.5145283  0.1328709 -11.398  < 2e-16 ***
## time13:15:00 -1.5343161  0.1375909 -11.151  < 2e-16 ***
## time13:30:00 -1.4587412  0.1330556 -10.963  < 2e-16 ***
## time13:45:00 -1.5742266  0.1326898 -11.864  < 2e-16 ***
## time14:00:00 -1.4337943  0.1315777 -10.897  < 2e-16 ***
## time14:15:00 -1.2495249  0.1277123  -9.784  < 2e-16 ***
## time14:30:00 -1.5787654  0.1353557 -11.664  < 2e-16 ***
## time14:45:00 -1.4088620  0.1348369 -10.449  < 2e-16 ***
## time15:00:00 -1.5210957  0.1292724 -11.767  < 2e-16 ***
## time15:15:00 -1.4157825  0.1338129 -10.580  < 2e-16 ***
## time15:30:00 -1.3088825  0.1282410 -10.206  < 2e-16 ***
## time15:45:00 -1.2760903  0.1258795 -10.137  < 2e-16 ***
## time16:00:00 -1.3378451  0.1313315 -10.187  < 2e-16 ***
## time16:15:00 -1.3137854  0.1333173  -9.855  < 2e-16 ***
## time16:30:00 -1.3321103  0.1325342 -10.051  < 2e-16 ***
## time16:45:00 -1.2403237  0.1250482  -9.919  < 2e-16 ***
## time17:00:00 -1.1855238  0.1268361  -9.347  < 2e-16 ***
## time17:15:00 -1.0625443  0.1274009  -8.340  < 2e-16 ***
## time17:30:00 -1.2409322  0.1265136  -9.809  < 2e-16 ***
## time17:45:00 -1.1051757  0.1224997  -9.022  < 2e-16 ***
## time18:00:00 -1.1507484  0.1232642  -9.336  < 2e-16 ***
## time18:15:00 -1.1125833  0.1208212  -9.209  < 2e-16 ***
## time18:30:00 -1.1248238  0.1220078  -9.219  < 2e-16 ***
## time18:45:00 -1.1353104  0.1201082  -9.452  < 2e-16 ***
## time19:00:00 -1.0275972  0.1303932  -7.881 3.25e-15 ***
## time19:15:00 -1.0714930  0.1220223  -8.781  < 2e-16 ***
## time19:30:00 -1.0467711  0.1239690  -8.444  < 2e-16 ***
## time19:45:00 -0.9516510  0.1219818  -7.802 6.11e-15 ***
## time20:00:00 -0.9427072  0.1209951  -7.791 6.63e-15 ***
## time20:15:00 -0.8495783  0.1192497  -7.124 1.05e-12 ***
## time20:30:00 -0.8663095  0.1219976  -7.101 1.24e-12 ***
## time20:45:00 -0.7909688  0.1190788  -6.642 3.09e-11 ***
## time21:00:00 -0.6428509  0.1169700  -5.496 3.89e-08 ***
## time21:15:00 -0.6211723  0.1138570  -5.456 4.88e-08 ***
## time21:30:00 -0.5449063  0.1137494  -4.790 1.66e-06 ***
## time21:45:00 -0.5031392  0.1187262  -4.238 2.26e-05 ***
## time22:00:00 -0.4420675  0.1130360  -3.911 9.20e-05 ***
## time22:15:00 -0.3652784  0.1127051  -3.241 0.001191 **
## time22:30:00 -0.4179336  0.1148726  -3.638 0.000275 ***
## time22:45:00 -0.2808685  0.1171273  -2.398 0.016486 *
## time23:00:00 -0.2110636  0.1118694  -1.887 0.059201 .
## time23:15:00 -0.2163419  0.1134501  -1.907 0.056529 .
## time23:30:00 -0.1313750  0.1117413  -1.176 0.239712
## time23:45:00 -0.1444534  0.1095277  -1.319 0.187211
## weekday.L    -0.1798574  0.0243823  -7.377 1.62e-13 ***
## weekday.Q    -0.2827574  0.0241809 -11.693  < 2e-16 ***
## weekday.C     0.1728147  0.0239214   7.224 5.04e-13 ***
## weekday^4     0.0316614  0.0235286   1.346 0.178413
## weekday^5     0.0500145  0.0235400   2.125 0.033615 *
## weekday^6    -0.1776784  0.0231876  -7.663 1.82e-14 ***
## Log(theta)    1.0660469  0.0286289  37.237  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.103e+00  7.399e-01   5.545 2.94e-08 ***
## cluster2     -1.457e+00  1.765e-01  -8.254  < 2e-16 ***
## cluster3      1.196e+00  3.101e-01   3.857 0.000115 ***
## cluster4      3.590e-01  2.324e-01   1.545 0.122334
## cluster5     -6.270e-01  1.926e-01  -3.256 0.001130 **
## cluster6     -1.262e+00  1.790e-01  -7.053 1.76e-12 ***
## time00:15:00 -3.650e-01  9.416e-01  -0.388 0.698267
## time00:30:00  8.000e-02  1.026e+00   0.078 0.937874
## time00:45:00  1.609e-01  1.026e+00   0.157 0.875381
## time01:00:00 -4.779e-01  9.419e-01  -0.507 0.611865
## time01:15:00 -6.635e-01  8.968e-01  -0.740 0.459388
## time01:30:00  7.979e-01  1.247e+00   0.640 0.522157
## time01:45:00 -2.810e-01  9.413e-01  -0.299 0.765301
## time02:00:00  1.465e-01  1.025e+00   0.143 0.886332
## time02:15:00  1.549e-02  1.025e+00   0.015 0.987944
## time02:30:00 -5.894e-01  8.962e-01  -0.658 0.510722
## time02:45:00 -6.490e-01  8.966e-01  -0.724 0.469111
## time03:00:00  1.439e-01  1.026e+00   0.140 0.888481
## time03:15:00  6.390e-01  1.247e+00   0.512 0.608324
## time03:30:00 -3.828e-01  9.412e-01  -0.407 0.684182
## time03:45:00  2.637e-02  1.026e+00   0.026 0.979496
## time04:00:00 -3.286e-01  9.403e-01  -0.349 0.726763
## time04:15:00 -2.803e-01  9.405e-01  -0.298 0.765709
## time04:30:00 -3.585e-01  9.414e-01  -0.381 0.703342
## time04:45:00  1.010e-01  1.027e+00   0.098 0.921657
## time05:00:00  8.845e-01  1.245e+00   0.710 0.477513
## time05:15:00  8.609e-01  1.245e+00   0.691 0.489391
## time05:30:00  1.506e+01  1.250e+03   0.012 0.990392
## time05:45:00  1.504e+01  1.402e+03   0.011 0.991443
## time06:00:00  9.248e-01  1.245e+00   0.743 0.457547
## time06:15:00  1.512e+01  1.242e+03   0.012 0.990288
## time06:30:00  1.508e+01  1.215e+03   0.012 0.990092
## time06:45:00  1.495e+01  1.288e+03   0.012 0.990740
## time07:00:00  1.500e+01  1.248e+03   0.012 0.990415
## time07:15:00  1.509e+01  1.261e+03   0.012 0.990450
## time07:30:00  8.383e-01  1.246e+00   0.673 0.501074
## time07:45:00 -3.327e-03  1.027e+00  -0.003 0.997416
## time08:00:00  7.147e-01  1.247e+00   0.573 0.566448
## time08:15:00  2.326e-02  1.026e+00   0.023 0.981910
## time08:30:00  1.451e-01  1.026e+00   0.141 0.887508
## time08:45:00  8.121e-02  1.026e+00   0.079 0.936933
## time09:00:00  1.682e-01  1.025e+00   0.164 0.869636
## time09:15:00  1.282e-01  1.025e+00   0.125 0.900502
## time09:30:00 -8.222e-01  8.671e-01  -0.948 0.343050
## time09:45:00 -4.518e-02  1.027e+00  -0.044 0.964909
## time10:00:00 -1.237e+00  8.352e-01  -1.481 0.138622
## time10:15:00 -1.208e+00  8.501e-01  -1.421 0.155252
## time10:30:00 -1.390e+00  8.528e-01  -1.630 0.103096
## time10:45:00 -1.421e+00  8.238e-01  -1.725 0.084601 .
## time11:00:00 -1.689e+00  8.090e-01  -2.088 0.036800 *
## time11:15:00 -1.644e+00  8.161e-01  -2.015 0.043915 *
## time11:30:00 -1.857e+00  8.100e-01  -2.292 0.021901 *
## time11:45:00 -1.301e+00  8.352e-01  -1.557 0.119390
## time12:00:00 -1.608e+00  8.377e-01  -1.920 0.054880 .
## time12:15:00 -1.685e+00  8.086e-01  -2.084 0.037129 *
## time12:30:00 -1.418e+00  8.237e-01  -1.722 0.085138 .
## time12:45:00 -1.612e+00  8.168e-01  -1.974 0.048424 *
## time13:00:00 -1.536e+00  8.245e-01  -1.863 0.062496 .
## time13:15:00 -1.204e+00  8.342e-01  -1.443 0.149006
## time13:30:00 -1.301e+00  8.360e-01  -1.556 0.119690
## time13:45:00 -1.522e+00  8.241e-01  -1.847 0.064789 .
## time14:00:00 -1.714e+00  8.087e-01  -2.120 0.034016 *
## time14:15:00 -1.439e+00  8.351e-01  -1.723 0.084889 .
## time14:30:00 -2.072e+00  7.957e-01  -2.604 0.009228 **
## time14:45:00 -1.860e+00  8.053e-01  -2.309 0.020921 *
## time15:00:00 -1.685e+00  8.088e-01  -2.083 0.037224 *
## time15:15:00 -1.310e+00  8.351e-01  -1.569 0.116762
## time15:30:00 -1.211e+00  8.341e-01  -1.452 0.146580
## time15:45:00 -3.901e-01  9.427e-01  -0.414 0.679046
## time16:00:00 -1.629e+00  8.165e-01  -1.995 0.046009 *
## time16:15:00 -2.037e+00  8.007e-01  -2.544 0.010947 *
## time16:30:00 -1.355e+00  8.367e-01  -1.619 0.105401
## time16:45:00 -1.526e+00  8.146e-01  -1.874 0.060936 .
## time17:00:00 -1.301e+00  8.359e-01  -1.557 0.119553
## time17:15:00 -1.798e+00  8.093e-01  -2.222 0.026298 *
## time17:30:00 -1.837e+00  8.037e-01  -2.285 0.022284 *
## time17:45:00 -1.472e+00  8.233e-01  -1.788 0.073795 .
## time18:00:00 -1.406e+00  8.227e-01  -1.709 0.087470 .
## time18:15:00 -1.103e+00  8.482e-01  -1.301 0.193368
## time18:30:00 -7.505e-01  8.977e-01  -0.836 0.403175
## time18:45:00 -1.065e+00  8.480e-01  -1.255 0.209320
## time19:00:00 -1.992e+00  8.126e-01  -2.452 0.014223 *
## time19:15:00 -1.464e+00  8.237e-01  -1.777 0.075511 .
## time19:30:00 -1.215e+00  8.508e-01  -1.428 0.153252
## time19:45:00 -1.311e+00  8.352e-01  -1.569 0.116538
## time20:00:00 -9.519e-01  8.684e-01  -1.096 0.272972
## time20:15:00 -1.266e+00  8.346e-01  -1.517 0.129308
## time20:30:00 -7.831e-01  8.966e-01  -0.873 0.382481
## time20:45:00 -1.608e+00  8.078e-01  -1.990 0.046572 *
## time21:00:00 -1.554e+00  8.150e-01  -1.907 0.056484 .
## time21:15:00 -1.000e+00  8.474e-01  -1.181 0.237748
## time21:30:00 -8.520e-01  8.675e-01  -0.982 0.326033
## time21:45:00 -9.380e-01  8.688e-01  -1.080 0.280300
## time22:00:00 -6.949e-01  8.962e-01  -0.775 0.438159
## time22:15:00 -6.162e-01  8.955e-01  -0.688 0.491362
## time22:30:00 -3.511e-01  9.419e-01  -0.373 0.709351
## time22:45:00 -8.198e-01  8.981e-01  -0.913 0.361334
## time23:00:00 -1.131e+00  8.480e-01  -1.334 0.182219
## time23:15:00  1.255e-01  1.026e+00   0.122 0.902669
## time23:30:00  7.004e-02  1.026e+00   0.068 0.945568
## time23:45:00  2.134e-01  1.025e+00   0.208 0.834999
## weekday.L    -3.597e-01  1.309e-01  -2.748 0.006005 **
## weekday.Q    -3.566e-01  1.331e-01  -2.679 0.007391 **
## weekday.C    -3.785e-01  1.340e-01  -2.825 0.004730 **
## weekday^4    -3.751e-02  1.389e-01  -0.270 0.787090
## weekday^5      4.413e-01  1.368e-01   3.225 0.001258 **
## weekday^6     -2.440e-01  1.411e-01  -1.729 0.083847 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 2.9039
## Number of iterations in BFGS optimization: 57
## Log-likelihood: -2.054e+04 on 215 Df
```

```
predhnb <- predict(modelhnb, newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predhnb),col="blue")
```



```
predhnb<- predict(modelhnb, newdata=test,type = "response")
rmsemodelhnb<-ModelMetrics::rmse(test$bikes.avail.count,round(predhnb))
maemodelhnb<-mae(test$bikes.avail.count,round(predhnb))
rmsemodelhnb
```

```
## [1] 9.389823
```

```
maemodelhnb
```

```
## [1] 6.35533
```

```
modelzp <- zeroinfl(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "poisson")
summary(modelzp)
```

```
##
## Call:
## zeroinfl(formula = bikes.avail.count ~ cluster + time + weekday, data = train,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -5.3904 -1.3039 -0.2913  1.0276 11.9897
##
## Count model coefficients (poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.030981   0.026172 115.811  < 2e-16 ***
## cluster2     -0.165605   0.012666 -13.075  < 2e-16 ***
## cluster3      0.941204   0.009968  94.424  < 2e-16 ***
## cluster4     -0.039038   0.011960  -3.264 0.001098 **
## cluster5     -0.338526   0.013237 -25.573  < 2e-16 ***
## cluster6     -1.234850   0.018834 -65.563  < 2e-16 ***
## time00:15:00 -0.031515   0.036488  -0.864 0.387755
## time00:30:00  0.064112   0.035464   1.808 0.070639 .
## time00:45:00  0.064663   0.035127   1.841 0.065649 .
## time01:00:00  0.093244   0.034847   2.676 0.007454 **
## time01:15:00  0.125499   0.035986   3.487 0.000488 ***
## time01:30:00  0.171229   0.035964   4.761 1.93e-06 ***
## time01:45:00  0.194546   0.033817   5.753 8.77e-09 ***
## time02:00:00  0.189097   0.033755   5.602 2.12e-08 ***
## time02:15:00  0.159081   0.034326   4.634 3.58e-06 ***
## time02:30:00  0.181466   0.034657   5.236 1.64e-07 ***
## time02:45:00  0.200658   0.033850   5.928 3.07e-09 ***
## time03:00:00  0.201878   0.033764   5.979 2.24e-09 ***
## time03:15:00  0.213730   0.035565   6.010 1.86e-09 ***
## time03:30:00  0.191675   0.034061   5.627 1.83e-08 ***
## time03:45:00  0.232465   0.034092   6.819 9.18e-12 ***
## time04:00:00  0.176710   0.033923   5.209 1.90e-07 ***
## time04:15:00  0.210521   0.033579   6.269 3.62e-10 ***
## time04:30:00  0.251926   0.032952   7.645 2.09e-14 ***
## time04:45:00  0.228824   0.035219   6.497 8.19e-11 ***
## time05:00:00  0.206114   0.033426   6.166 6.99e-10 ***
## time05:15:00  0.179182   0.034359   5.215 1.84e-07 ***
## time05:30:00  0.224068   0.033849   6.620 3.60e-11 ***
## time05:45:00  0.160767   0.036129   4.450 8.59e-06 ***
## time06:00:00  0.188122   0.033108   5.682 1.33e-08 ***
## time06:15:00  0.134083   0.035120   3.818 0.000135 ***
## time06:30:00  0.133946   0.034187   3.918 8.93e-05 ***
## time06:45:00  0.082781   0.035664   2.321 0.020280 *
## time07:00:00  0.028471   0.035505   0.802 0.422613
## time07:15:00  0.027873   0.035751   0.780 0.435614
## time07:30:00 -0.051738   0.036577  -1.414 0.157218
## time07:45:00 -0.088533   0.038134  -2.322 0.020255 *
## time08:00:00 -0.267679   0.039579  -6.763 1.35e-11 ***
## time08:15:00 -0.409793   0.039734 -10.313  < 2e-16 ***
## time08:30:00 -0.458618   0.041180 -11.137  < 2e-16 ***
## time08:45:00 -0.525836   0.041651 -12.625  < 2e-16 ***
## time09:00:00 -0.653579   0.041901 -15.598  < 2e-16 ***
## time09:15:00 -0.749146   0.044345 -16.894  < 2e-16 ***
## time09:30:00 -0.779694   0.045559 -17.114  < 2e-16 ***
## time09:45:00 -1.006604   0.052402 -19.209  < 2e-16 ***
## time10:00:00 -1.118203   0.054243 -20.615  < 2e-16 ***
## time10:15:00 -1.060115   0.052459 -20.208  < 2e-16 ***
## time10:30:00 -1.129386   0.059615 -18.945  < 2e-16 ***
## time10:45:00 -1.089984   0.054256 -20.090  < 2e-16 ***
## time11:00:00 -1.212909   0.059426 -20.410  < 2e-16 ***
## time11:15:00 -1.193810   0.057419 -20.791  < 2e-16 ***
## time11:30:00 -1.354553   0.061065 -22.182  < 2e-16 ***
## time11:45:00 -1.396584   0.061077 -22.866  < 2e-16 ***
## time12:00:00 -1.341885   0.063414 -21.161  < 2e-16 ***
## time12:15:00 -1.439092   0.062135 -23.161  < 2e-16 ***
## time12:30:00 -1.503946   0.061760 -24.351  < 2e-16 ***
## time12:45:00 -1.569804   0.064547 -24.320  < 2e-16 ***
## time13:00:00 -1.684786   0.068322 -24.660  < 2e-16 ***
## time13:15:00 -1.697083   0.074357 -22.824  < 2e-16 ***
## time13:30:00 -1.573795   0.066193 -23.776  < 2e-16 ***
## time13:45:00 -1.654409   0.067428 -24.536  < 2e-16 ***
## time14:00:00 -1.563915   0.065276 -23.959  < 2e-16 ***
## time14:15:00 -1.353860   0.059491 -22.757  < 2e-16 ***
## time14:30:00 -1.634390   0.068928 -23.712  < 2e-16 ***
## time14:45:00 -1.517206   0.066175 -22.927  < 2e-16 ***
## time15:00:00 -1.601686   0.064244 -24.931  < 2e-16 ***
## time15:15:00 -1.519233   0.066274 -22.924  < 2e-16 ***
## time15:30:00 -1.385615   0.061146 -22.661  < 2e-16 ***
## time15:45:00 -1.386093   0.058626 -23.643  < 2e-16 ***
## time16:00:00 -1.455450   0.063690 -22.852  < 2e-16 ***
## time16:15:00 -1.458628   0.064860 -22.489  < 2e-16 ***
## time16:30:00 -1.462776   0.064018 -22.850  < 2e-16 ***
## time16:45:00 -1.374412   0.058985 -23.301  < 2e-16 ***
## time17:00:00 -1.289932   0.058166 -22.177  < 2e-16 ***
## time17:15:00 -1.148682   0.057665 -19.920  < 2e-16 ***
## time17:30:00 -1.323597   0.057529 -23.008  < 2e-16 ***
## time17:45:00 -1.225014   0.055142 -22.216  < 2e-16 ***
## time18:00:00 -1.247331   0.056259 -22.171  < 2e-16 ***
## time18:15:00 -1.189270   0.053190 -22.359  < 2e-16 ***
## time18:30:00 -1.209393   0.053413 -22.642  < 2e-16 ***
## time18:45:00 -1.230011   0.052995 -23.210  < 2e-16 ***
## time19:00:00 -1.137468   0.057038 -19.942  < 2e-16 ***
## time19:15:00 -1.147165   0.053178 -21.572  < 2e-16 ***
## time19:30:00 -1.099466   0.054065 -20.336  < 2e-16 ***
## time19:45:00 -1.035786   0.052892 -19.583  < 2e-16 ***
## time20:00:00 -0.996463   0.051035 -19.525  < 2e-16 ***
## time20:15:00 -0.874095   0.048848 -17.894  < 2e-16 ***
## time20:30:00 -0.902852   0.051520 -17.524  < 2e-16 ***
## time20:45:00 -0.798762   0.048278 -16.545  < 2e-16 ***
## time21:00:00 -0.662005   0.045354 -14.596  < 2e-16 ***
## time21:15:00 -0.631592   0.042665 -14.803  < 2e-16 ***
## time21:30:00 -0.565099   0.042352 -13.343  < 2e-16 ***
## time21:45:00 -0.488812   0.045034 -10.854  < 2e-16 ***
## time22:00:00 -0.471830   0.040421 -11.673  < 2e-16 ***
## time22:15:00 -0.411815   0.040606 -10.142  < 2e-16 ***
## time22:30:00 -0.471152   0.042225 -11.158  < 2e-16 ***
## time22:45:00 -0.317964   0.041124  -7.732 1.06e-14 ***
## time23:00:00 -0.256759   0.037994  -6.758 1.40e-11 ***
## time23:15:00 -0.243136   0.039752  -6.116 9.57e-10 ***
## time23:30:00 -0.134498   0.037396  -3.597 0.000322 ***
## time23:45:00 -0.142859   0.037247  -3.835 0.000125 ***
## weekday.L    -0.097944   0.009648 -10.152  < 2e-16 ***
## weekday.Q    -0.346955   0.009480 -36.600  < 2e-16 ***
## weekday.C     0.112747   0.009173  12.291  < 2e-16 ***
## weekday^4    -0.097797   0.008693 -11.250  < 2e-16 ***
## weekday^5     0.088627   0.008681  10.209  < 2e-16 ***
## weekday^6    -0.112452   0.008390 -13.403  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.06258    0.75698  -5.367 8.01e-08 ***
## cluster2        1.48225    0.18998   7.802 6.08e-15 ***
## cluster3       -1.10381    0.31683  -3.484 0.000494 ***
## cluster4       -0.55373    0.27857  -1.988 0.046835 *
## cluster5        0.54694    0.21438   2.551 0.010735 *
## cluster6        1.07800    0.20136   5.354 8.62e-08 ***
## time00:15:00    0.37976    0.96009   0.396 0.692437
## time00:30:00   -0.07805    1.05218  -0.074 0.940866
## time00:45:00   -0.17692    1.05327  -0.168 0.866603
## time01:00:00    0.49189    0.95783   0.514 0.607569
## time01:15:00    0.68701    0.91277   0.753 0.451649
## time01:30:00   -0.87384    1.32734  -0.658 0.510318
## time01:45:00    0.29914    0.95708   0.313 0.754618
## time02:00:00   -0.12765    1.03855  -0.123 0.902176
## time02:15:00   -0.01438    1.04719  -0.014 0.989046
## time02:30:00    0.59114    0.91211   0.648 0.516922
## time02:45:00    0.65167    0.91192   0.715 0.474848
## time03:00:00   -0.16248    1.05507  -0.154 0.877612
## time03:15:00   -0.66629    1.30103  -0.512 0.608563
## time03:30:00    0.39425    0.95894   0.411 0.680975
## time03:45:00   -0.01928    1.04461  -0.018 0.985274
## time04:00:00    0.34275    0.95887   0.357 0.720754
## time04:15:00    0.28446    0.95825   0.297 0.766577
## time04:30:00    0.36206    0.95753   0.378 0.705339
## time04:45:00   -0.11388    1.05242  -0.108 0.913833
## time05:00:00   -0.94422    1.31670  -0.717 0.473305
## time05:15:00   -0.92253    1.32587  -0.696 0.486556
## time05:30:00  -15.05676 1258.02513  -0.012 0.990451
## time05:45:00  -15.03819 1411.60642  -0.011 0.991500
## time06:00:00   -0.92224    1.26598  -0.728 0.466319
## time06:15:00  -15.12172 1254.79059  -0.012 0.990385
## time06:30:00  -15.08434 1225.22348  -0.012 0.990177
## time06:45:00  -14.95198 1314.66024  -0.011 0.990926
## time07:00:00  -14.99653 1267.75436  -0.012 0.990562
## time07:15:00  -15.08823 1272.41253  -0.012 0.990539
## time07:30:00   -1.02571    1.43509  -0.715 0.474774
## time07:45:00   -0.03301    1.07296  -0.031 0.975457
## time08:00:00   -1.49456    2.39820  -0.623 0.533152
## time08:15:00   -0.57267    1.47501  -0.388 0.697834
## time08:30:00   -0.12463    1.03805  -0.120 0.904437
## time08:45:00   -0.06809    1.03961  -0.065 0.947776
## time09:00:00   -0.16281    1.04237  -0.156 0.875881
## time09:15:00   -0.11174    1.03803  -0.108 0.914275
## time09:30:00    0.84447    0.88064   0.959 0.337596
## time09:45:00    0.06340    1.04224   0.061 0.951490
## time10:00:00    1.19930    0.85933   1.396 0.162829
## time10:15:00    1.15041    0.87651   1.312 0.189354
## time10:30:00    1.34762    0.87858   1.534 0.125064
## time10:45:00    1.41748    0.83939   1.689 0.091277 .
## time11:00:00    1.64163    0.83345   1.970 0.048875 *
## time11:15:00    1.63217    0.83677   1.951 0.051109 .
## time11:30:00    1.79004    0.83477   2.144 0.032005 *
## time11:45:00    1.13573    0.88134   1.289 0.197522
## time12:00:00    1.60276    0.85693   1.870 0.061436 .
## time12:15:00    1.55029    0.84497   1.835 0.066547 .
## time12:30:00    1.32435    0.85805   1.543 0.122724
## time12:45:00    1.37159    0.86883   1.579 0.114415
## time13:00:00    0.77216    1.00432   0.769 0.441989
## time13:15:00    0.88058    0.92420   0.953 0.340691
## time13:30:00    1.05625    0.89937   1.174 0.240221
## time13:45:00    1.20308    0.90405   1.331 0.183267
## time14:00:00    1.52291    0.85460   1.782 0.074747 .
## time14:15:00    1.38929    0.86348   1.609 0.107629
## time14:30:00    1.84705    0.83504   2.212 0.026971 *
## time14:45:00    1.68032    0.84484   1.989 0.046709 *
## time15:00:00    1.40241    0.86612   1.619 0.105405
## time15:15:00    1.12457    0.88510   1.271 0.203889
## time15:30:00    1.15778    0.86176   1.344 0.179107
## time15:45:00    0.18614    1.03916   0.179 0.857839
## time16:00:00    1.47443    0.85698   1.720 0.085342 .
## time16:15:00    1.96993    0.82552   2.386 0.017019 *
## time16:30:00    1.28037    0.86782   1.475 0.140108
## time16:45:00    1.36515    0.85352   1.599 0.109725
## time17:00:00    0.96213    0.91779   1.048 0.294498
## time17:15:00    1.68825    0.84331   2.002 0.045291 *
## time17:30:00    1.61510    0.84639   1.908 0.056364 .
## time17:45:00    1.30768    0.86744   1.508 0.131676
## time18:00:00    1.20209    0.87311   1.377 0.168579
## time18:15:00    0.74256    0.94173   0.789 0.430400
## time18:30:00    0.49358    0.98041   0.503 0.614650
## time18:45:00    0.65045    0.94080   0.691 0.489326
## time19:00:00    1.73510    0.86334   2.010 0.044456 *
## time19:15:00    1.22573    0.87890   1.395 0.163132
## time19:30:00    0.96523    0.92047   1.049 0.294352
## time19:45:00    1.06138    0.89523   1.186 0.235782
## time20:00:00    0.66591    0.95272   0.699 0.484578
## time20:15:00    1.23631    0.85824   1.441 0.149719
## time20:30:00    0.76291    0.92172   0.828 0.407836
## time20:45:00    1.53995    0.83286   1.849 0.064459 .
## time21:00:00    1.48876    0.84061   1.771 0.076553 .
## time21:15:00    0.85801    0.89174   0.962 0.335960
## time21:30:00    0.71988    0.91167   0.790 0.429745
## time21:45:00    0.84371    0.90619   0.931 0.351822
## time22:00:00    0.62639    0.93939   0.667 0.504895
## time22:15:00    0.54345    0.93349   0.582 0.560448
## time22:30:00    0.06060    1.08477   0.056 0.955452
## time22:45:00    0.73525    0.94309   0.780 0.435613
## time23:00:00    1.11145    0.87030   1.277 0.201569
## time23:15:00   -0.30349    1.14644  -0.265 0.791225
## time23:30:00   -0.09512    1.06210  -0.090 0.928636
## time23:45:00   -0.25730    1.06690  -0.241 0.809424
## weekday.L       0.44539    0.15481   2.877 0.004015 **
## weekday.Q       0.23054    0.15467   1.491 0.136085
## weekday.C       0.50639    0.15506   3.266 0.001091 **
## weekday^4       0.04312    0.15402   0.280 0.779527
## weekday^5      -0.41501    0.15434  -2.689 0.007168 **
## weekday^6       0.15376    0.15389   0.999 0.317719
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Number of iterations in BFGS optimization: 151
## Log-likelihood: -2.606e+04 on 214 Df
```

```
predzp<- predict(modelzp,newdata=test,type = "response")
rmsemodelzp<-ModelMetrics::rmse(test$bikes.avail.count,round(predzp))
maemodelzp<-mae(test$bikes.avail.count,round(predzp))
rmsemodelzp
```

```
## [1] 8.747335
```

```
maemodelzp
```

```
## [1] 5.996827
```

```
modelznb <- zeroinfl(bikes.avail.count ~ cluster + time + weekday, data=train,dist = "negbin")
summary(modelznb)
```

```
##
## Call:
## zeroinfl(formula = bikes.avail.count ~ cluster + time + weekday, data = train,
##     dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.5517 -0.7862 -0.1675  0.5623  6.8249
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.017687   0.081814  36.885  < 2e-16 ***
## cluster2     -0.344190   0.035797  -9.615  < 2e-16 ***
## cluster3      0.837422   0.029115  28.763  < 2e-16 ***
## cluster4     -0.124187   0.029750  -4.174 2.99e-05 ***
## cluster5     -0.399459   0.031115 -12.838  < 2e-16 ***
## cluster6     -0.977135   0.037143 -26.308  < 2e-16 ***
## time00:15:00 -0.039295   0.113574  -0.346 0.729351    
## time00:30:00  0.041632   0.113196   0.368 0.713033    
## time00:45:00  0.075517   0.112359   0.672 0.501517    
## time01:00:00  0.122905   0.112835   1.089 0.276047    
## time01:15:00  0.140470   0.115094   1.220 0.222284    
## time01:30:00  0.178559   0.115059   1.552 0.120687    
## time01:45:00  0.217067   0.110981   1.956 0.050479 .  
## time02:00:00  0.206752   0.109918   1.881 0.059976 .  
## time02:15:00  0.186019   0.111514   1.668 0.095291 .  
## time02:30:00  0.220379   0.112741   1.955 0.050615 .  
## time02:45:00  0.219154   0.112482   1.948 0.051373 .  
## time03:00:00  0.215547   0.110983   1.942 0.052117 .  
## time03:15:00  0.223108   0.116131   1.921 0.054710 .  
## time03:30:00  0.208227   0.112204   1.856 0.063484 .  
## time03:45:00  0.241938   0.112122   2.158 0.030942 *  
## time04:00:00  0.218267   0.110184   1.981 0.047600 *  
## time04:15:00  0.217642   0.110891   1.963 0.049686 *  
## time04:30:00  0.310714   0.110572   2.810 0.004953 ** 
## time04:45:00  0.250740   0.113681   2.206 0.027409 *  
## time05:00:00  0.223506   0.109534   2.041 0.041299 *  
## time05:15:00  0.173146   0.111220   1.557 0.119520    
## time05:30:00  0.257980   0.111613   2.311 0.020813 *  
## time05:45:00  0.196339   0.118356   1.659 0.097140 .  
## time06:00:00  0.208992   0.108172   1.932 0.053354 .  
## time06:15:00  0.144215   0.111389   1.295 0.195427    
## time06:30:00  0.142367   0.110514   1.288 0.197665    
## time06:45:00  0.063019   0.113554   0.555 0.578918    
## time07:00:00  0.032374   0.111877   0.289 0.772296    
## time07:15:00  0.080350   0.112826   0.712 0.476364    
## time07:30:00 -0.002099   0.112333  -0.019 0.985093    
## time07:45:00 -0.051854   0.116543  -0.445 0.656369    
## time08:00:00 -0.234077   0.115378  -2.029 0.042479 *  
## time08:15:00 -0.382854   0.113973  -3.359 0.000782 ***
## time08:30:00 -0.403148   0.113048  -3.566 0.000362 ***
## time08:45:00 -0.398085   0.114234  -3.485 0.000492 ***
## time09:00:00 -0.520583   0.112234  -4.638 3.51e-06 ***
## time09:15:00 -0.612236   0.114203  -5.361 8.28e-08 ***
## time09:30:00 -0.645466   0.118137  -5.464 4.66e-08 ***
## time09:45:00 -0.822387   0.123108  -6.680 2.39e-11 ***
## time10:00:00 -0.975188   0.127083  -7.674 1.67e-14 ***
## time10:15:00 -0.956952   0.122721  -7.798 6.30e-15 ***
## time10:30:00 -1.009146   0.131015  -7.703 1.33e-14 ***
## time10:45:00 -0.996133   0.124701  -7.988 1.37e-15 ***
## time11:00:00 -1.068581   0.132933  -8.039 9.09e-16 ***
## time11:15:00 -1.096774   0.128471  -8.537  < 2e-16 ***
## time11:30:00 -1.253882   0.135834  -9.231  < 2e-16 ***
## time11:45:00 -1.230754   0.128729  -9.561  < 2e-16 ***
## time12:00:00 -1.165391   0.137331  -8.486  < 2e-16 ***
## time12:15:00 -1.272709   0.128858  -9.877  < 2e-16 ***
## time12:30:00 -1.371954   0.123865 -11.076  < 2e-16 ***
## time12:45:00 -1.382131   0.130607 -10.582  < 2e-16 ***
## time13:00:00 -1.514410   0.129222 -11.719  < 2e-16 ***
## time13:15:00 -1.496351   0.133521 -11.207  < 2e-16 ***
## time13:30:00 -1.427254   0.130640 -10.925  < 2e-16 ***
## time13:45:00 -1.587740   0.126422 -12.559  < 2e-16 ***
## time14:00:00 -1.435361   0.130228 -11.022  < 2e-16 ***
## time14:15:00 -1.241185   0.127409  -9.742  < 2e-16 ***
## time14:30:00 -1.581488   0.134589 -11.750  < 2e-16 ***
## time14:45:00 -1.436706   0.134865 -10.653  < 2e-16 ***
## time15:00:00 -1.534386   0.129376 -11.860  < 2e-16 ***
## time15:15:00 -1.387185   0.133613 -10.382  < 2e-16 ***
## time15:30:00 -1.311005   0.128530 -10.200  < 2e-16 ***
## time15:45:00 -1.250911   0.126307  -9.904  < 2e-16 ***
## time16:00:00 -1.376340   0.130317 -10.562  < 2e-16 ***
## time16:15:00 -1.392083   0.146193  -9.522  < 2e-16 ***
## time16:30:00 -1.328059   0.133690  -9.934  < 2e-16 ***
## time16:45:00 -1.270026   0.125239 -10.141  < 2e-16 ***
## time17:00:00 -1.189112   0.126407  -9.407  < 2e-16 ***
## time17:15:00 -1.088463   0.127883  -8.511  < 2e-16 ***
## time17:30:00 -1.256479   0.127647  -9.843  < 2e-16 ***
## time17:45:00 -1.153098   0.122576  -9.407  < 2e-16 ***
## time18:00:00 -1.157831   0.122880  -9.422  < 2e-16 ***
## time18:15:00 -1.132532   0.120127  -9.428  < 2e-16 ***
## time18:30:00 -1.122475   0.121901  -9.208  < 2e-16 ***
## time18:45:00 -1.157185   0.118123  -9.796  < 2e-16 ***
## time19:00:00 -1.060928   0.133197  -7.965 1.65e-15 ***
## time19:15:00 -1.088087   0.121919  -8.925  < 2e-16 ***
## time19:30:00 -1.091623   0.121595  -8.978  < 2e-16 ***
## time19:45:00 -1.001826   0.121860  -8.221  < 2e-16 ***
## time20:00:00 -0.962181   0.122039  -7.884 3.17e-15 ***
## time20:15:00 -0.858788   0.120363  -7.135 9.68e-13 ***
## time20:30:00 -0.893764   0.120743  -7.402 1.34e-13 ***
## time20:45:00 -0.808731   0.121192  -6.673 2.50e-11 ***
## time21:00:00 -0.646076   0.118894  -5.434 5.51e-08 ***
## time21:15:00 -0.631496   0.115383  -5.473 4.42e-08 ***
## time21:30:00 -0.548314   0.115540  -4.746 2.08e-06 ***
## time21:45:00 -0.506781   0.120569  -4.203 2.63e-05 ***
## time22:00:00 -0.437623   0.114681  -3.816 0.000136 ***
## time22:15:00 -0.361881   0.114636  -3.157 0.001595 **
## time22:30:00 -0.421190   0.116434  -3.617 0.000298 ***
## time22:45:00 -0.287662   0.119131  -2.415 0.015749 *
## time23:00:00 -0.213023   0.113990  -1.869 0.061654 .
## time23:15:00 -0.212956   0.115329  -1.847 0.064818 .
## time23:30:00 -0.127907   0.113630  -1.126 0.260316
## time23:45:00 -0.139489   0.111394  -1.252 0.210492
## weekday.L    -0.175447   0.024509  -7.158 8.16e-13 ***
## weekday.Q    -0.280735   0.024390 -11.510  < 2e-16 ***
## weekday.C     0.194173   0.024718   7.856 3.98e-15 ***
## weekday^4     0.036587   0.024316   1.505 0.132420
## weekday^5     0.055954   0.024178   2.314 0.020654 *
## weekday^6    -0.178077   0.023356  -7.625 2.45e-14 ***
## Log(theta)    1.019888   0.033597  30.357  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.116e+00  1.233e+00  -4.151 3.31e-05 ***
## cluster2      1.962e+00  4.878e-01   4.022 5.78e-05 ***
## cluster3     -5.786e-01  6.196e-01  -0.934 0.350394
## cluster4     -1.513e+01  9.871e+02  -0.015 0.987775
## cluster5      1.942e-01  6.815e-01   0.285 0.775705
## cluster6      2.187e+00  6.615e-01   3.307 0.000944 ***
## time00:15:00  3.907e-01  1.105e+00   0.353 0.723738
## time00:30:00 -5.164e-01  1.425e+00  -0.362 0.717112
## time00:45:00 -5.830e-01  1.423e+00  -0.410 0.682138
## time01:00:00  5.920e-01  1.133e+00   0.523 0.601214
## time01:15:00  7.463e-01  1.065e+00   0.701 0.483312
## time01:30:00 -1.011e+00  1.595e+00  -0.634 0.526115
## time01:45:00  2.968e-01  1.093e+00   0.272 0.785925
## time02:00:00 -1.275e-02  1.190e+00  -0.011 0.991449
## time02:15:00  1.392e-01  1.194e+00   0.117 0.907162
## time02:30:00  5.893e-01  1.040e+00   0.567 0.570811
## time02:45:00  6.572e-01  1.041e+00   0.631 0.527948
## time03:00:00 -1.621e-01  1.199e+00  -0.135 0.892462
## time03:15:00 -5.541e-01  1.489e+00  -0.372 0.709757
## time03:30:00  4.427e-01  1.089e+00   0.407 0.684361
## time03:45:00  6.830e-02  1.180e+00   0.058 0.953824
## time04:00:00  4.289e-01  1.088e+00   0.394 0.693566
## time04:15:00  3.009e-01  1.088e+00   0.277 0.782071
## time04:30:00  5.164e-01  1.085e+00   0.476 0.634249
## time04:45:00 -1.406e-01  1.197e+00  -0.117 0.906505
## time05:00:00 -9.606e-01  1.557e+00  -0.617 0.537339
## time05:15:00 -1.132e+00  1.627e+00  -0.696 0.486368
## time05:30:00 -1.506e+01  1.304e+03  -0.012 0.990783
## time05:45:00 -1.504e+01  1.517e+03  -0.010 0.992088
## time06:00:00 -1.048e+00  1.509e+00  -0.694 0.487584
## time06:15:00 -1.512e+01  1.324e+03  -0.011 0.990884
## time06:30:00 -1.509e+01  1.245e+03  -0.012 0.990330
## time06:45:00 -1.495e+01  1.334e+03  -0.011 0.991055
## time07:00:00 -1.500e+01  1.328e+03  -0.011 0.990989
## time07:15:00 -1.509e+01  1.288e+03  -0.012 0.990653
## time07:30:00 -1.396e+01  9.795e+02  -0.014 0.988627
## time07:45:00 -4.389e-01  1.452e+00  -0.302 0.762447
## time08:00:00 -1.461e+00  2.296e+00  -0.636 0.524589
## time08:15:00 -1.189e+00  2.264e+00  -0.525 0.599439
## time08:30:00 -1.435e+01  1.341e+03  -0.011 0.991460
## time08:45:00 -1.476e+01  1.314e+03  -0.011 0.991039
## time09:00:00 -1.505e+01  1.290e+03  -0.012 0.990692
## time09:15:00 -1.591e+01  2.059e+03  -0.008 0.993834
## time09:30:00  6.836e-01  1.124e+00   0.608 0.543033
## time09:45:00 -1.690e-01  1.332e+00  -0.127 0.899030
## time10:00:00  3.632e-01  1.318e+00   0.276 0.782904
## time10:15:00 -2.750e-01  1.484e+00  -0.185 0.852999
## time10:30:00  3.622e-01  1.312e+00   0.276 0.782449
## time10:45:00  7.565e-01  1.172e+00   0.646 0.518592
## time11:00:00  9.038e-01  1.292e+00   0.700 0.484098
## time11:15:00 -5.759e-01  2.519e+00  -0.229 0.819142
## time11:30:00  4.818e-01  1.782e+00   0.270 0.786895
## time11:45:00 -8.425e-01  2.652e+00  -0.318 0.750756
## time12:00:00  6.467e-01  1.444e+00   0.448 0.654183
## time12:15:00  7.123e-01  1.252e+00   0.569 0.569426
## time12:30:00 -1.148e+01  1.160e+03  -0.010 0.992104
## time12:45:00  7.786e-01  1.183e+00   0.658 0.510473
## time13:00:00 -1.879e-01  1.649e+00  -0.114 0.909263
## time13:15:00 -1.325e-02  1.401e+00  -0.009 0.992454
## time13:30:00  5.167e-01  1.207e+00   0.428 0.668573
## time13:45:00 -1.345e+01  1.240e+03  -0.011 0.991349
## time14:00:00  9.691e-01  1.208e+00   0.802 0.422593
## time14:15:00  9.133e-01  1.150e+00   0.794 0.427122
## time14:30:00  1.307e+00  1.118e+00   1.169 0.242227
## time14:45:00  8.534e-01  1.235e+00   0.691 0.489700
## time15:00:00  9.107e-02  1.789e+00   0.051 0.959392
## time15:15:00  5.557e-01  1.342e+00   0.414 0.678784
## time15:30:00 -8.916e-02  1.687e+00  -0.053 0.957845
## time15:45:00 -2.059e+00  6.437e+00  -0.320 0.749078
## time16:00:00  2.612e-03  1.628e+00   0.002 0.998720
## time16:15:00  6.981e-01  1.929e+00   0.362 0.717359
## time16:30:00  4.459e-01  1.402e+00   0.318 0.750519
## time16:45:00  2.656e-01  1.440e+00   0.184 0.853699
## time17:00:00  4.415e-01  1.242e+00   0.355 0.722228
## time17:15:00  1.486e+00  1.071e+00   1.387 0.165496
## time17:30:00  1.428e+00  1.120e+00   1.275 0.202347
## time17:45:00 -3.915e-01  2.271e+00  -0.172 0.863094
## time18:00:00  6.944e-01  1.171e+00   0.593 0.553245
## time18:15:00 -1.267e+00  2.987e+00  -0.424 0.671551
## time18:30:00 -1.157e+00  2.865e+00  -0.404 0.686221
## time18:45:00 -1.521e+01  1.744e+03  -0.009 0.993045
## time19:00:00  1.595e+00  1.115e+00   1.431 0.152485
## time19:15:00  7.098e-01  1.200e+00   0.591 0.554259
## time19:30:00 -1.280e+01  1.082e+03  -0.012 0.990561
## time19:45:00 -8.025e-01  2.436e+00  -0.329 0.741883
## time20:00:00 -6.191e-01  2.271e+00  -0.273 0.785158
## time20:15:00  9.355e-01  1.109e+00   0.844 0.398851
## time20:30:00 -1.344e+01  1.215e+03  -0.011 0.991174
## time20:45:00  1.372e+00  1.039e+00   1.320 0.186722
## time21:00:00  1.668e+00  1.006e+00   1.658 0.097265 .
## time21:15:00  5.486e-01  1.132e+00   0.485 0.627915
## time21:30:00  5.914e-01  1.137e+00   0.520 0.603028
## time21:45:00  7.870e-01  1.112e+00   0.708 0.478954
## time22:00:00  7.471e-01  1.080e+00   0.692 0.489036
## time22:15:00  4.801e-01  1.103e+00   0.435 0.663317
## time22:30:00 -2.570e-01  1.407e+00  -0.183 0.855014
## time22:45:00  5.895e-01  1.170e+00   0.504 0.614517
## time23:00:00  1.161e+00  1.035e+00   1.121 0.262104
## time23:15:00 -4.176e-01  1.363e+00  -0.306 0.759352
## time23:30:00 -1.627e-01  1.262e+00  -0.129 0.897388
## time23:45:00 -3.399e-01  1.289e+00  -0.264 0.792084
## weekday.L     6.032e-01  8.016e-01   0.753 0.451743
## weekday.Q     1.304e-01  8.620e-01   0.151 0.879787
## weekday.C     2.732e+00  1.859e+00   1.470 0.141658
## weekday^4     1.180e-01  7.666e-01   0.154 0.877650
## weekday^5    -9.881e-01  1.577e+00  -0.627 0.530888
## weekday^6    -5.516e-01  1.407e+00  -0.392 0.695105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Theta = 2.7729
## Number of iterations in BFGS optimization: 195
## Log-likelihood: -2.057e+04 on 215 Df
```

```
predznb<- predict(modelznb,newdata=test,type = "response")
rmsemodelznb<-ModelMetrics::rmse(test$bikes.avail.count,round(predznb))
maemodelznb<-mae(test$bikes.avail.count,round(predznb))
rmsemodelznb
```

```
## [1] 9.41988
```

```
maemodelznb
```

```
## [1] 6.421954
```

```
rmse<-c(rmsemodelp,rmsemodelqp,rmsemodelnb,rmsemodelhp,rmsemodelhnb,
           rmsemodelzp,rmsemodelznb)
mae<-c(maemodelp,maemodelqp,maemodelnb,maemodelhp,maemodelhnb,
           maemodelzp,maemodelznb)
models<-c("pois","q_pois","nb","h_pois","h_nb","zer_pois","zer_nb")
data.frame(models,rmse,mae)%>%
  arrange(rmse)
```

```
##     models     rmse      mae
## 1     pois 8.675188 5.961929
## 2   q_pois 8.675188 5.961929
## 3   h_pois 8.733759 5.965102
## 4 zer_pois 8.747335 5.996827
## 5       nb 9.360348 6.381345
## 6     h_nb 9.389823 6.355330
## 7   zer_nb 9.419880 6.421954
```

Closing Remarks

The exploratory analysis of the Citi Bike dataset for NYC-based trips in September 2022 shows a consistent pattern of use dependent on time of day and day of the week. The pattern persists week over week for the given month. The majority of users are member which indicates the availability of bikes will be dependent on work schedules during weekdays. The surplus and shortage count of bikes by docking station denotes the uneven direction of bikes in some sections of New York City. The shortage (and surplus) counts confirm the practice of rebalancing by Citi Bike to ensure availability of bikes. The duration of bike trips may not play as large a factor in bike availability compared to time of day, day of week, and directional flow of bikes throughout the city.

Appendix with Code

```
# Required packages
library(tidyverse)
library(ggplot2)
library(skimr)
library(lubridate)
library(fpp3)
library(assertthat)
library(igraph)
library(ggraph)
library(ggmap)
library(leaflet)
library(rgdal)
library(RColorBrewer)
library(jpeg)
library(data.table)
library(MASS)

# Create additional columns of pertinence
# weekday, day of the month, trip duration in minutes, start hour
citibike <- fread("data/202209-citibike-tripdata.csv", data.table=FALSE, check.names=TRUE) %>%
#citibike <- read.csv("data/202209-citibike-tripdata.csv", check.names=TRUE) %>%
  mutate(day = factor(mday(ymd_hms(started_at))),
         start.hour=factor(hour(ymd_hms(started_at))),
         weekday = lubridate::wday(ymd_hms(started_at), label=TRUE, abbr=TRUE),
         trip.duration = as.numeric(difftime(ended_at,started_at,units="mins")),
         member_casual = factor(member_casual)) %>%
  rename(ride.id=ride_id, rideable.type=rideable_type, started.at=started_at,
         ended.at=ended_at, start.station.name=start_station_name, start.station.id=start_station_id,
         end.station.name=end_station_name, end.station.id=end_station_id, start.lat=start_lat,
         start.lng=start_lng, end.lat=end_lat, end.lng=end_lng, member.casual=member_casual)

# Figure out abandoned and label them
citibike$end.station.name[citibike$end.station.name == ''] <- "Abandoned"
citibike$end.station.id[citibike$end.station.id  == ''] <- "ABAN"

# Output DF if needed
head(citibike)

# Trip by weekday by segment by time of day
citibike %>%
  group_by(day, member.casual, start.hour) %>%
  summarize(n=n(),
            weekday=weekday[1]) %>%
  group_by(weekday, member.casual, start.hour) %>%
  summarize(n.m=mean(n)) %>%
  ggplot(aes(x=start.hour, y=n.m, fill=weekday)) +
  geom_bar(stat='identity') +
  labs(x='Time of Day',
       y='Number of Trips',
       title='Average Number of Bike Trips by Time, Day, and User') +
  facet_grid(weekday~member.casual) +
  theme(axis.text.x = element_text(size=8, angle=90),
        legend.position = 'none')
```

```r
citibike$started.at.ts <- as_datetime(as.character(citibike$started.at))

citibike_by_hour <- citibike %>%
  mutate(hour=lubridate::floor_date(started.at.ts, "1 hour")) %>%
  group_by(hour, member.casual) %>%
  summarize(cnt=n())
#citibike_by_hour

citibike_by_hour_ts <- citibike_by_hour %>%
  as_tsibble(index=hour, key=c(member.casual))
#citibike_by_hour_ts

autoplot(citibike_by_hour_ts, cnt) +
  labs(title = "Ride Trips by Hour - Sept. 2022",
       subtitle = "Citi Bike NYC",
       y = "Ride Trips Counts") +
  guides(fill=guide_legend(title="New Legend Title"))
citibike_by_hour_ts %>%
  gg_season(cnt, period = "week") +
  labs(y = "Ride Trips",
       title = "Seasonal plot: Weekly Trip Counts for Sept. 2022")
citibike %>%
  filter(member.casual %in% c('member', 'casual')) %>%
  group_by(weekday, start.hour, member.casual) %>%
  summarize(med.duration=median(trip.duration)) %>%
  ggplot(aes(x=start.hour, y=med.duration, group=member.casual,
             color=member.casual, linetype=member.casual, shape=member.casual)) +
  geom_point(size=2) +
  geom_line(size=0.5) +
  facet_wrap(~weekday, nrow=1) +
  labs(x='Time of Day',
       y='Median Trip Duration',
       title = "Median Trip Duration by User and Day for Sept. 2022",
       color='User Type') +
  scale_x_discrete(breaks=c(0,6,12,18))
# Create table of start to end station IDs
stations_cols <- citibike %>%
  dplyr::select(start.station.id, end.station.id)
stations_table <- as.data.frame((table(stations_cols)))

stations_table <- stations_table %>%
  filter(Freq > 0)

stations_table_order <- stations_table[order(-stations_table$Freq),]

stations_table_dif <- stations_table_order %>%
  filter(as.character(start.station.id) != as.character(end.station.id))

# Create table of start to end station IDs
stations_cols <- citibike %>%
  dplyr::select(start.station.id, end.station.id)
stations_table <- as.data.frame((table(stations_cols)))
stations_table <- stations_table %>%
  filter(Freq > 0)

stations_table_order <- stations_table[order(-stations_table$Freq),]

stations_table_dif <- stations_table_order %>%
  filter(as.character(start.station.id) != as.character(end.station.id))

stations_table_dif1 <- stations_table_dif
stations_table_dif2 <- stations_table_dif

# Added all=TRUE to account for one-sided counts
stations_table_dif_merge <-
  merge(stations_table_dif1,
      stations_table_dif1,
      by.x=c('start.station.id','end.station.id'),
      by.y=c('end.station.id','start.station.id'), all = TRUE)

# Set the NA (one-sided trips) to count of 0
stations_table_dif_merge[is.na(stations_table_dif_merge)] <- 0

stations_table_dif_merge$surplus <- stations_table_dif_merge$Freq.y - stations_table_dif_merge$Freq.x
stations_table_dif_merge <- stations_table_dif_merge[order(-stations_table_dif_merge$surplus),]

# Remove rows with start station id equal to ABAN for abandoned, those are a result of the merge all=TR
stations_table_dif_merge <- stations_table_dif_merge %>% filter(start.station.id != 'ABAN')

# Want to identify the surplus (or not) by station for the month
station_surplus_count <- stations_table_dif_merge %>%
  group_by(start.station.id) %>%
  summarize(surplus.sum=sum(surplus))


station_surplus_count <- station_surplus_count[order(-station_surplus_count$surplus.sum),]
colnames(station_surplus_count)[1] <- "station.id"

# Extract just the end station Id, lat, log
# because this had the higher count from the initial dataset, going with end_station_id
end_station_info <- citibike %>%
  dplyr::select(end.station.id, end.lng, end.lat)

end_station_info <- end_station_info[!duplicated(end_station_info$end.station.id),]

# Now I want the coordinates of all those station_ids
station_surplus_count_coords <- merge(x = station_surplus_count, y = end_station_info, by.x = 'station.

colnames(station_surplus_count_coords)[3] <- "lng"
colnames(station_surplus_count_coords)[4] <- "lat"

station_surplus_count_coords <- station_surplus_count_coords %>%
  add_row(station.id = "ABAN", surplus.sum=1363, lng=-73.99, lat=40.67)

# Using basemaps for NYC
```

```
m <- leaflet(data=station_surplus_count_coords) %>%
#  setView(zoom=12) %>%
  addTiles() %>%
  addCircleMarkers(
    ~lng, ~lat,
    popup=~as.character(station.id),
    label=~as.character(station.id),
    radius=.5,
    color = ~ifelse(surplus.sum >= 1, 'blue',
                    ifelse(surplus.sum == 0, 'green', 'red'))
  )

# Display map
#m
img <- readJPEG("stations_surplus_sum.jpg")
plot(1:10,ty="n", axes = 0, xlab='', ylab='', main='Overall Monthly Surplus/Shortage by Docking Station
rasterImage(img,-1,-1,12,12)
stations_with_elevation <- read.csv('stations_with_elevation.csv', row.names = 1, header= TRUE)

stations_with_boro_hood <- read.csv('stations_with_boro_and_hood.csv', row.names = 1, header= TRUE)

# Combine the elevation, borough, neighborhood, and September surplus
# First let's trim the DF for elevation
stations_with_elevation_trim <- stations_with_elevation %>%
  dplyr::select(short_name, station_id, elevation, elev_units)

stations_with_boro_hood_trim <- stations_with_boro_hood %>%
  dplyr::select(short_name, name, station_id, capacity, ntaname, boro_name, lon, lat)

stations_attrs_trim <-
  merge(stations_with_elevation_trim,
        stations_with_boro_hood_trim,
        by.x=c('short_name'),
        by.y=c('short_name'), all = TRUE)

stations_attrs_trim <- stations_attrs_trim %>%
  dplyr::select(-station_id.y)

colnames(stations_attrs_trim)[colnames(stations_attrs_trim) == 'station_id.x'] <- 'station_id'

stations_attrs_trim[c("boro_name")][is.na(stations_attrs_trim[c("boro_name")])] <- "New Jersey"

stations_attrs_trim <-
  stations_attrs_trim %>%
  mutate(ntaname = ifelse(startsWith(short_name, "JC"), "Jersey City", ntaname))

stations_attrs_trim <-
  stations_attrs_trim %>%
  mutate(ntaname = ifelse(startsWith(short_name, "HB"), "Hoboken", ntaname))


stations_attrs_trim_sur <-
  merge(stations_attrs_trim,
        station_surplus_count,
        by.x=c('short_name'),
        by.y=c('station.id'), all.x = TRUE)

stations_attrs_trim_sur <- stations_attrs_trim_sur %>%
  filter(!is.na(surplus.sum))

stations_attrs_trim_sur %>%
  filter(surplus.sum >= -100 & surplus.sum <= 100) %>%
  ggplot(aes(x=surplus.sum, color=boro_name), ) +
  theme(legend.position="bottom") +
  geom_histogram(bins=201) +
  facet_wrap(~boro_name) +
  labs(x = 'Surplus Count',
       y = 'Number of Stations',
       title = "Station Surplus by Borough Histogram for Sept. 2022",
       fill = "Borough")
# This removed the duplicate rows by transposing the start and end station ids
stations_table_dif_merge_temp <- stations_table_dif_merge %>% select(start.station.id, end.station.id)


stations_for_graph <- stations_table_dif_merge_temp[!duplicated(lapply(as.data.frame(t(stations_table_di

g_stations <- graph_from_data_frame(stations_for_graph, directed=FALSE, vertices=station_surplus_count_

edges_for_plot <- stations_for_graph %>%
  inner_join(station_surplus_count_coords %>% select(station.id, lng, lat), by=c('start.station.id' = '
  rename(x=lng, y=lat) %>%
  inner_join(station_surplus_count_coords %>% select(station.id, lng, lat), by=c('end.station.id' = 'sta
  rename(xend=lng, yend=lat)

#assert_that(nrow(edges_for_plot) == nrow(stations_for_graph))

citibike_bike_avail <- fread("bike_avail_by_station_and_time.csv", data.table=FALSE, check.names=FALSE)
citibike_bike_avail <- citibike_bike_avail %>%
  dplyr::select(-V1)
citibike_bike_avail_long <- citibike_bike_avail %>%
  pivot_longer(!timestamp, names_to = "station.id", values_to = "bikes.avail")

#head(citibike_bike_avail_long)
citibike_bike_avail_long_trim <- citibike_bike_avail_long %>%
  filter(as.integer(station.id) == 3582)
p<-ggplot(citibike_bike_avail_long_trim, aes(x=timestamp, y=bikes.avail, group=station.id)) +
  geom_line(aes(color=station.id), show.legend = FALSE) +
  geom_point(aes(color=station.id), show.legend = FALSE) +
  labs(x = 'Date/Time (15-minute Increments)',
       y = 'Bikes Available',
       title = "Docking Station 3582 Bike Availability")
p
stations_attrs_trim_bk61 <- stations_attrs_trim %>%
  filter(ntaname == "Crown Heights North")

#(stations_attrs_trim_bk61)
```

```
#11 stations in this file
# Let's try the clustering
# https://www.r-bloggers.com/2019/06/hierarchical-clustering-for-location-based-strategy-using-r-for-e-
library(geosphere)

# Distance matrix for docking stations
# Result is in meters
dist_mat <- distm(stations_attrs_trim_bk61[9:10], stations_attrs_trim_bk61[9:10], fun=distHaversine)
dist_mat <- as.data.frame(dist_mat)
dist_mat[is.na(dist_mat)] <- 0

dMat <- as.dist(dist_mat)
dMat[1:10]
# Now for the clustering
hier_clust <- hclust(dMat, method = "complete")
# 400 meters is about a quarter of a mile (0.248548 miles)
stations_attrs_trim_bk61$cluster <- cutree(hier_clust, h=400)
stations_attrs_trim_bk61
stations_with_bike_avail <- read.csv('bike_avail_by_station_and_time.csv', row.names = 1, header= TRUE,
stations_with_bike_avail_long <- stations_with_bike_avail %>%
  pivot_longer(!timestamp, names_to = "station.id", values_to = "bikes.avail.count")
station_to_cluster <- stations_attrs_trim_bk61 %>%
  dplyr::select(station_id, cluster)

(station_to_cluster)
# Merge long df with counts with station.id and cluster to label clusters properly
stations_bike_avail_by_time <-
  merge(stations_with_bike_avail_long,
        station_to_cluster,
        by.x=c('station.id'),
        by.y=c('station_id'), all.x = TRUE)
stations_bike_avail_by_time_no_na <- stations_bike_avail_by_time %>%
  filter(!is.na(cluster))
# group by cluster ID for each timestamp
stations_bike_avail_by_time_no_na_group <- aggregate(bikes.avail.count ~ timestamp + cluster, data = sta
stations_bike_avail_by_time_no_na_group <- stations_bike_avail_by_time_no_na_group[order(stations_bike_a
# mutate to conver timestamp into day of the week, etc.
stations_bike_avail_by_time_no_na_group <- stations_bike_avail_by_time_no_na_group %>%
  mutate(time=as.ITime(ymd_hms(timestamp)),
         weekday = lubridate::wday(ymd_hms(timestamp),label=TRUE,abbr=TRUE))
stations_bike_avail_by_time_no_na_group$cluster <- as.factor(stations_bike_avail_by_time_no_na_group$clu
stations_bike_avail_by_time_no_na_group$time <- as.factor(stations_bike_avail_by_time_no_na_group$time)
stations_bike_avail_by_time_no_na_group$weekday <- as.factor(stations_bike_avail_by_time_no_na_group$wee
tab <- table(stations_bike_avail_by_time_no_na_group$bikes.avail.count)
tab
# Below shows the range of values for `bikes.avail.count` with 35611 at zero and max of 438 with one oc
stations_bike_avail_by_time_no_timestamp <- subset(stations_bike_avail_by_time_no_na_group, select=-c(ti
#stations_bike_avail_by_time_no_timestamp
# Start with data partition here.
set.seed(8675309)
index <- sample(2, nrow(stations_bike_avail_by_time_no_timestamp), replace = TRUE, p=c(0.8, 0.2))
train <- stations_bike_avail_by_time_no_timestamp[index==1,]
test <- stations_bike_avail_by_time_no_timestamp[index==2,]
```

```r
mod1 <- glm(bikes.avail.count ~ cluster + time + weekday, data = train, family ="poisson")
summary(mod1)
pred <- predict.glm(mod1, newdata=test[test$bikes.avail.count != 0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(pred),col="blue")
library(Metrics)
pred <- predict.glm(mod1, newdata = test, type = "response")
rmsemodelp <- ModelMetrics::rmse(test$bikes.avail.count,round(pred))
maemodelp <- mae(test$bikes.avail.count,round(pred))
rmsemodelp
maemodelp
mod2 <- glm(bikes.avail.count ~ cluster + time + weekday, data = train, family ="quasipoisson")
summary(mod2)
predq <- predict.glm(mod2, newdata=test[test$bikes.avail.count != 0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predq),col="blue")
predq <- predict.glm(mod2,newdata=test, type = "response")
rmsemodelqp <- ModelMetrics::rmse(test$bikes.avail.count,round(predq))
maemodelqp <- mae(test$bikes.avail.count,round(predq))
rmsemodelqp
maemodelqp
mod3 <- glm.nb(bikes.avail.count ~ cluster + time + weekday, data=train)
summary(mod3)
prednb <- predict.glm(mod3,newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(prednb),col="blue")
prednb <- predict.glm(mod3,newdata=test,type = "response")
rmsemodelnb <- ModelMetrics::rmse(test$bikes.avail.count,round(prednb))
maemodelnb <- mae(test$bikes.avail.count,round(prednb))
rmsemodelnb
maemodelnb
library(pscl)
modelhp <- hurdle(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "poisson")
summary(modelhp)
predhp <- predict(modelhp, newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predhp),col="blue")
predhp <- predict(modelhp, newdata=test, type = "response")
rmsemodelhp<-ModelMetrics::rmse(test$bikes.avail.count,round(predhp))
maemodelhp<-mae(test$bikes.avail.count,round(predhp))
rmsemodelhp
maemodelhp
modelhnb <- hurdle(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "negbin")
summary(modelhnb)
predhnb <- predict(modelhnb, newdata=test[test$bikes.avail.count!=0,],type = "response")
plot(test$bikes.avail.count[test$bikes.avail.count!=0],type = "b",col="red")
lines(round(predhnb),col="blue")
predhnb<- predict(modelhnb, newdata=test,type = "response")
rmsemodelhnb<-ModelMetrics::rmse(test$bikes.avail.count,round(predhnb))
maemodelhnb<-mae(test$bikes.avail.count,round(predhnb))
rmsemodelhnb
maemodelhnb
modelzp <- zeroinfl(bikes.avail.count ~ cluster + time + weekday, data=train, dist = "poisson")
```

```
summary(modelzp)
predzp<- predict(modelzp,newdata=test,type = "response")
rmsemodelzp<-ModelMetrics::rmse(test$bikes.avail.count,round(predzp))
maemodelzp<-mae(test$bikes.avail.count,round(predzp))
rmsemodelzp
maemodelzp
modelznb <- zeroinfl(bikes.avail.count ~ cluster + time + weekday, data=train,dist = "negbin")
summary(modelznb)
predznb<- predict(modelznb,newdata=test,type = "response")
rmsemodelznb<-ModelMetrics::rmse(test$bikes.avail.count,round(predznb))
maemodelznb<-mae(test$bikes.avail.count,round(predznb))
rmsemodelznb
maemodelznb
rmse<-c(rmsemodelp,rmsemodelqp,rmsemodelnb,rmsemodelhp,rmsemodelhnb,
           rmsemodelzp,rmsemodelznb)
mae<-c(maemodelp,maemodelqp,maemodelnb,maemodelhp,maemodelhnb,
           maemodelzp,maemodelznb)
models<-c("pois","q_pois","nb","h_pois","h_nb","zer_pois","zer_nb")
data.frame(models,rmse,mae)%>%
  arrange(rmse)
# https://yihui.org/en/2018/09/code-appendix/
```

LaTex help https://tex.stackexchange.com/questions/10684/vertical-space-in-lists