YouBike Rental Data Analytics

Group 4

Agenda

- Business Problem
- Data Overview & Visualization
- Multilevel Models
- Gaussian Process Models
- Takeaway

1

Business Problem

Business Problem

- Bike-dispatching is always a problem for YouBike company. There will be excess supplies or demands in some rental spots that need tackling with.
- In this project, our goal is to **estimate rent and return counts of each station by hour**, in order to optimize the dispatching route.
- We achieve this goal by two steps.
    - Collect viable feature data that might influence whether people rent YouBike as a way to move in the city.
    - Use the technique of multi-level modeling as well as Gaussian Process to build models that can effectively predict future behavior.

2

Data Overview & Visualization

Data Overview

- This dataset includes YouBike rental data within January 2016.
- Features included are:
  - Number of stations
  - Date
  - Duration of that journey
  - Costs of that journey
  - Rent & Return Stations
  - Area for the YouBike Station
  - Station Name of the YouBike Station

Top 5 Popular Stations

rank	station_name	station_area	rent_count
1	MRT Gongguan Sta.(Exit 2)	大安	23550
2	Roosevelt & Xinsheng S. Intersection	大安	20111
3	MRT Taipei City Hall Stataion(Exit 3)-2	信義	19021
4	NTU Information Bldg.	大安	15554
5	N.T.U.S.T	大安	14353

Rent Count

rank	station_name	station_area	return_count
1	MRT Gongguan Sta.(Exit 2)	大安	23197
2	Roosevelt & Xinsheng S. Intersection	大安	19938
3	MRT Taipei City Hall Stataion(Exit 3)-2	信義	16896
4	NTU Information Bldg.	大安	16869
5	NTNU Library	大安	15443

Return Count

Top 5 & Last 5 Stations needed dispatching

Top 5

rank	station_name	station_area	net_in
1	Taipei Medical University	信義	2440
2	NTNU Library	大安	1774
3	Huajiang High School	萬華	1347
4	Lanya Park	士林	1324
5	NTU Information Bldg.	大安	1315

Last 5

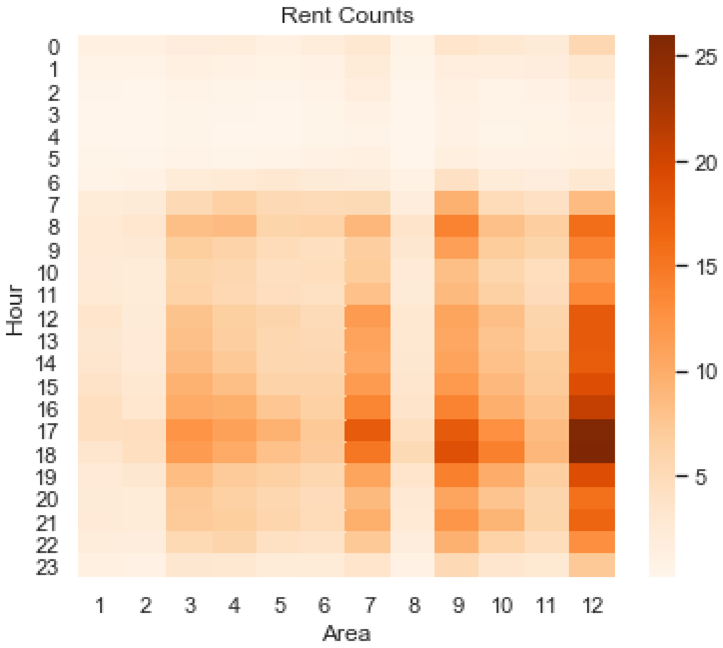
rank	station_name	station_area	net_in
1	MRT Zhishan Sta.(Exit 2)	士林	-2321
2	MRT Taipei City Hall Stataion(Exit 3)-2	信義	-2125
3	Xinyi Square(Taipei 101)	信義	-2081
4	MRT Daan Sta.	大安	-1988
5	MRT Daan Park Sta.	大安	-1632

※ Net in = Return count - Rent count

Heatmap for each area average usage by hour

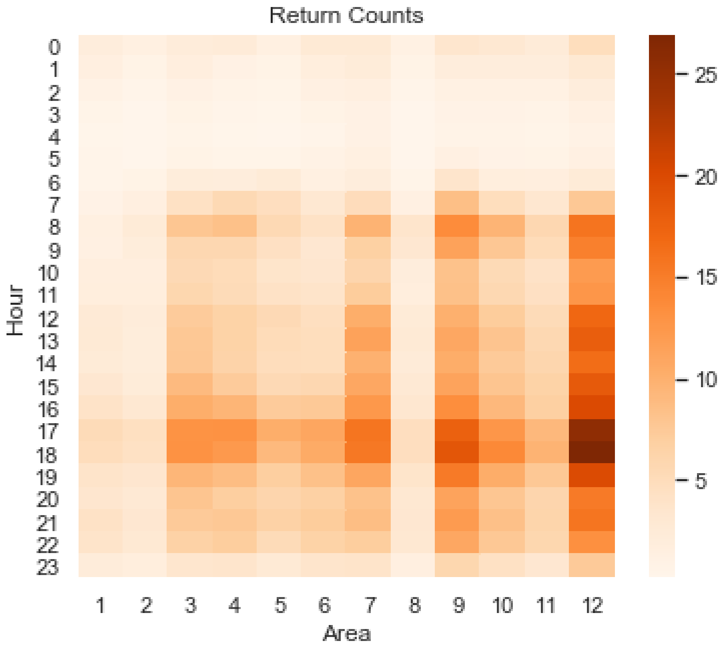
Area with high "Rent Count"

12-大安 9-信義 7-中正



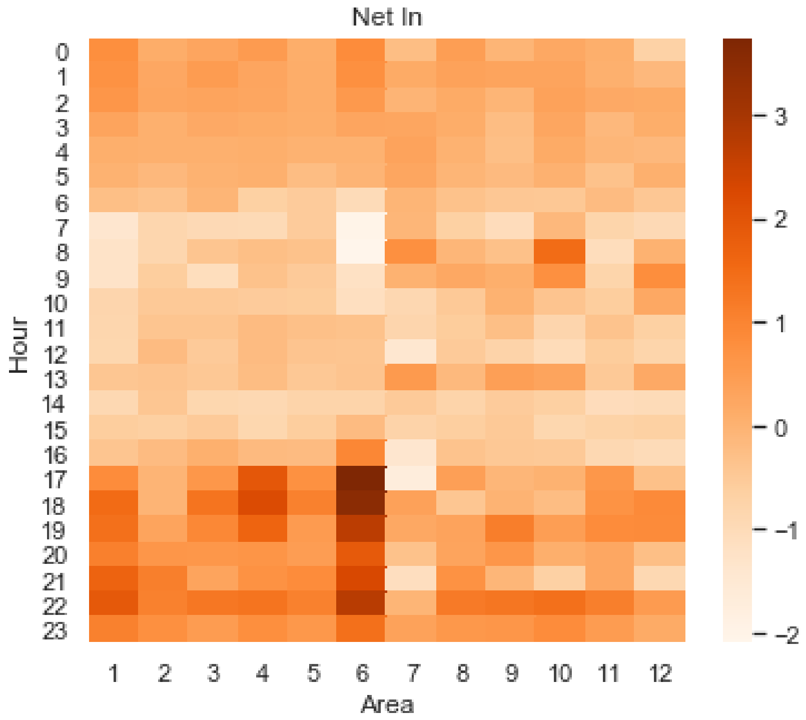
Hour with high "Return Count"

8~9 & 12~19



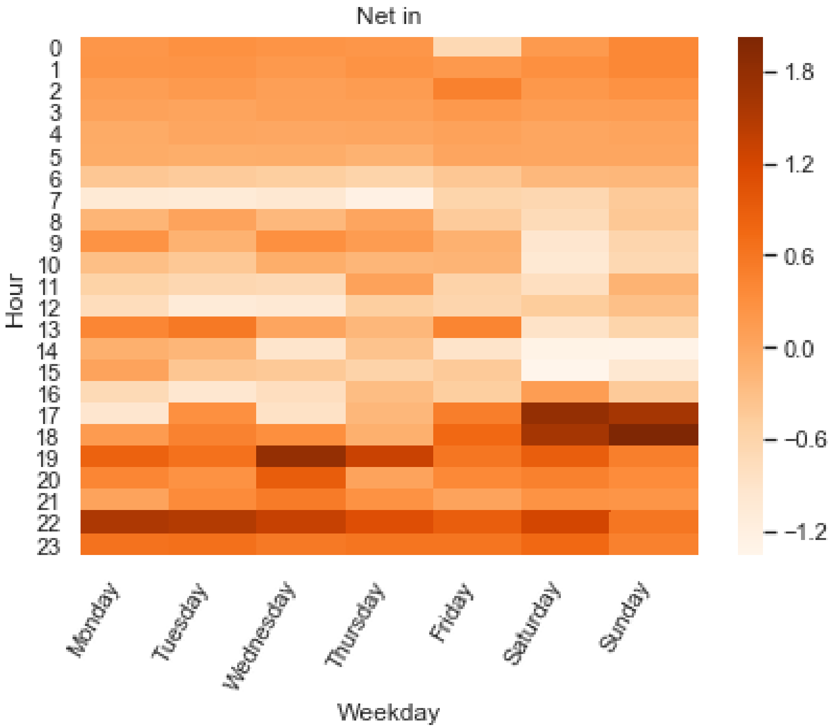
Heatmap for each area average usage by hour

- Dark area represents excessive bikes.
  - 萬華區(6) at 5pm ~ 7pm
- Light area represents shortage.
  - 萬華區(6) at 7am ~ 8am
  - 中正區(7) at 4pm ~ 5pm



Heatmap for each area average usage by hour

- Dark area represents excessive bikes.
  - Evening (around 5pm ~ 7pm)
- Light area represents shortage.
  - Afternoon (around 2pm ~ 4pm)



Data Preprocessing for Visualization

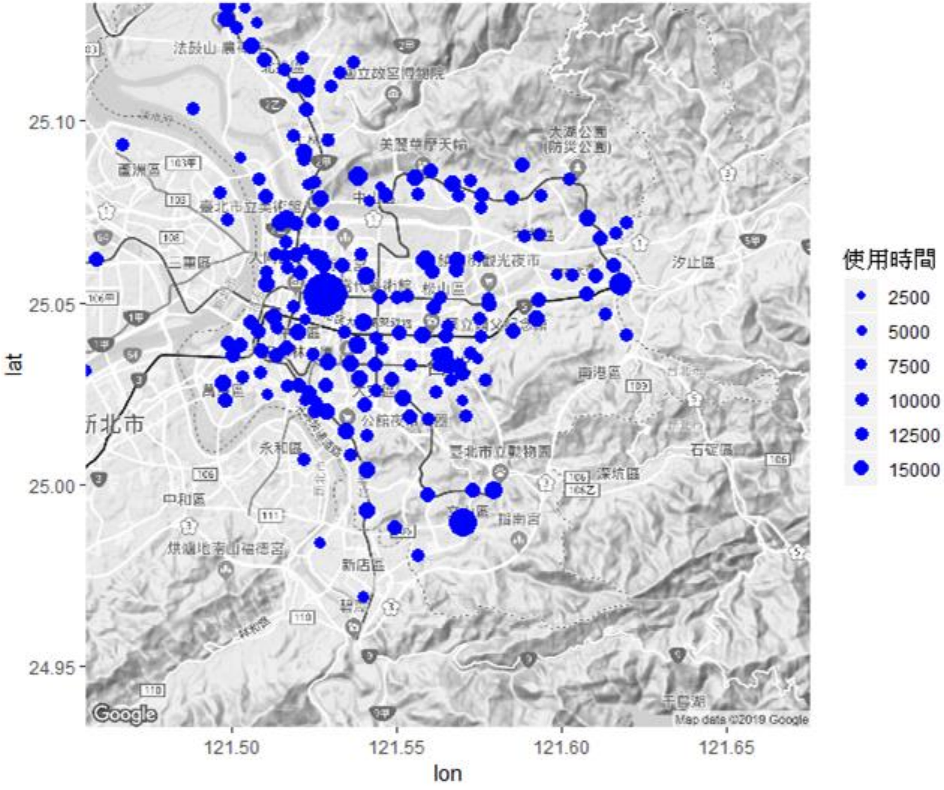
- Before plotting, we should find out the longitude and latitude of those YouBike Stations.

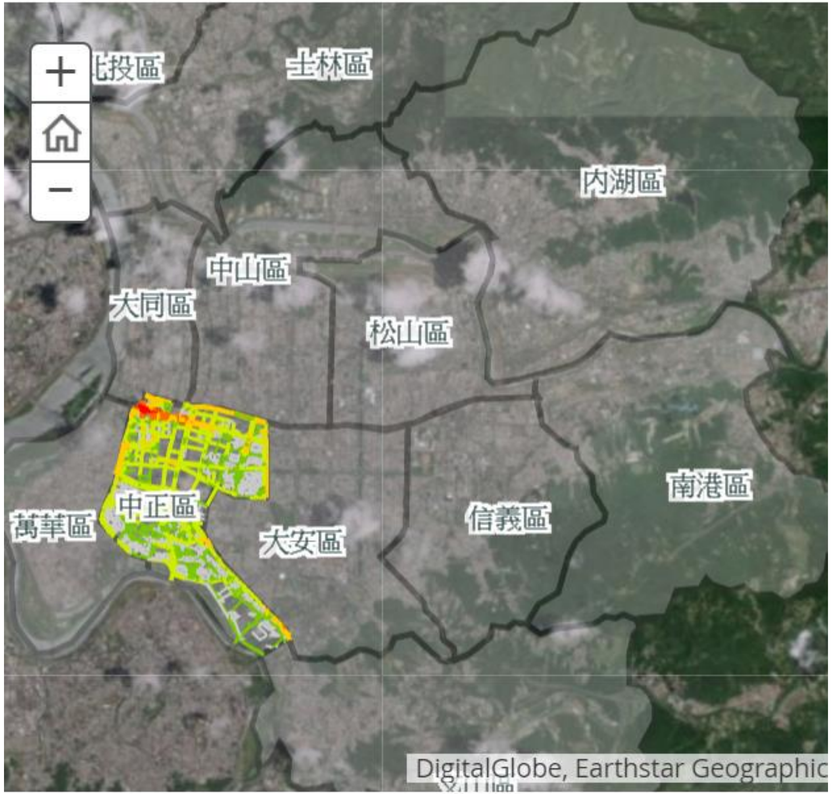
```
df <- read.csv("C:/Users/willy/Desktop/ntu mba/grade 1 semester two/multivariate analysis/QBS group project/20160
               sep="")

#TO MAKE SURE THIS GOOGLEMAP TO FIND THE RIGHT PLACE, I ADD TAIWAN FOR EACH STATION NAME
df2 <- df
df2$rent_sta <- lapply(df2$rent_sta, function(x) paste("Taiwan Taipei", x, sep = " "))
#find all station names in data and its lat and long
station_names <- unique(df2$rent_sta)
#lat_and_long <- geocode('Taiwan Taipei Xingya Jr. High School')
lat_and_long = data.frame()
for (i in station_names){
  lat_and_long <- rbind(lat_and_long,geocode(i))
}
#s_lat_and_long為一個有rent_sta跟其相關之經緯度的dataframe
s_lat_and_long = data.frame(matrix(ncol = 3, nrow = 0))
x <- c("rent_sta","lat","long")
colnames(s_lat_and_long) <- x
```

YouBike Animation

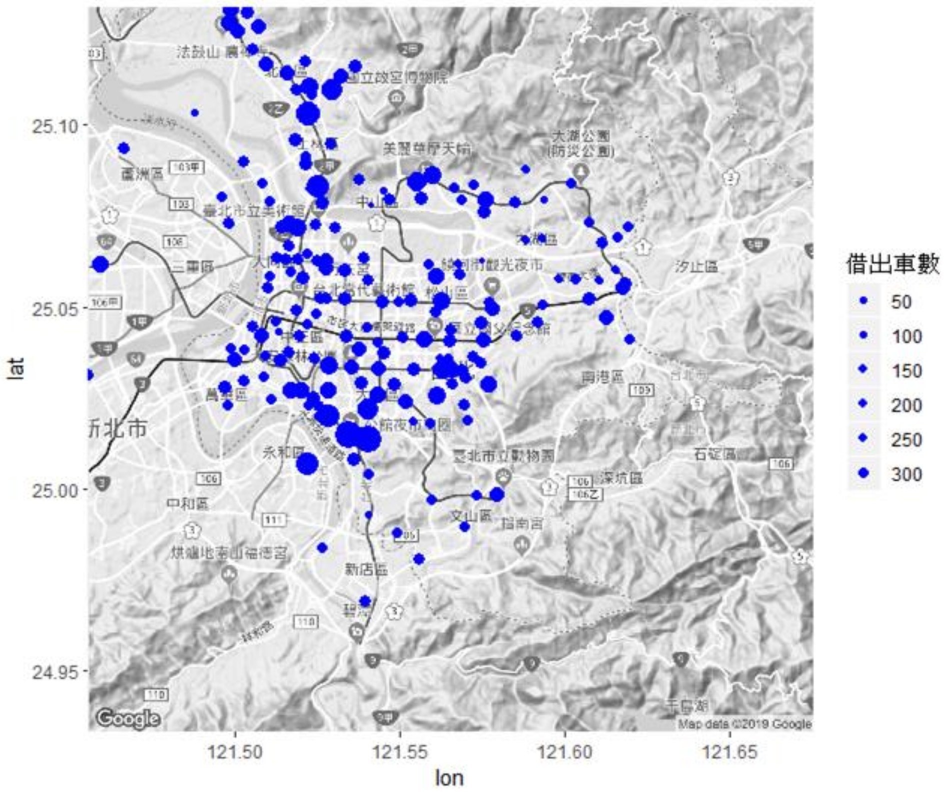
Used Time per YouBike Station within January

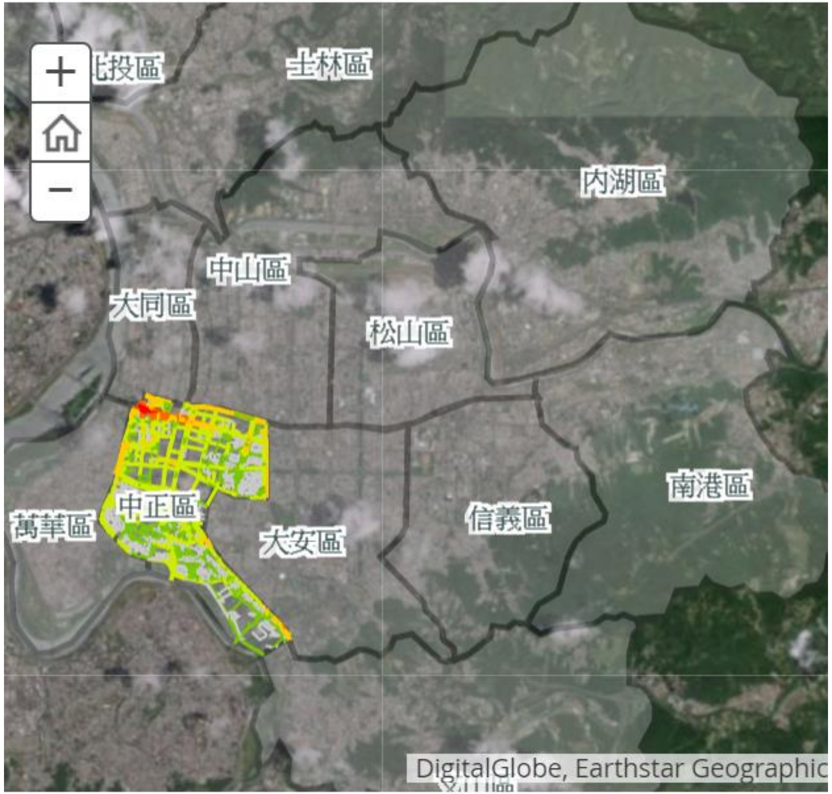




YouBike Animation

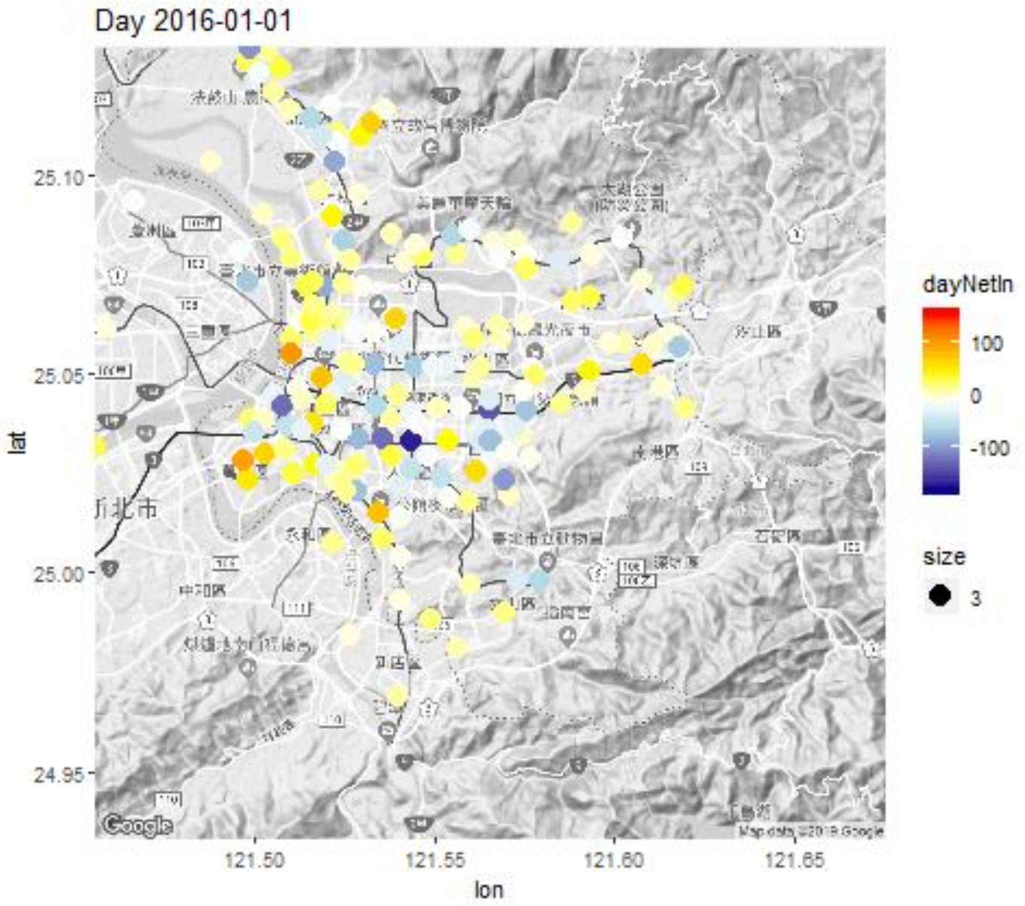
Used Time per YouBike Station within January





YouBike Animation

Net-in of each station by day.



※Net in = Return count - Rent count

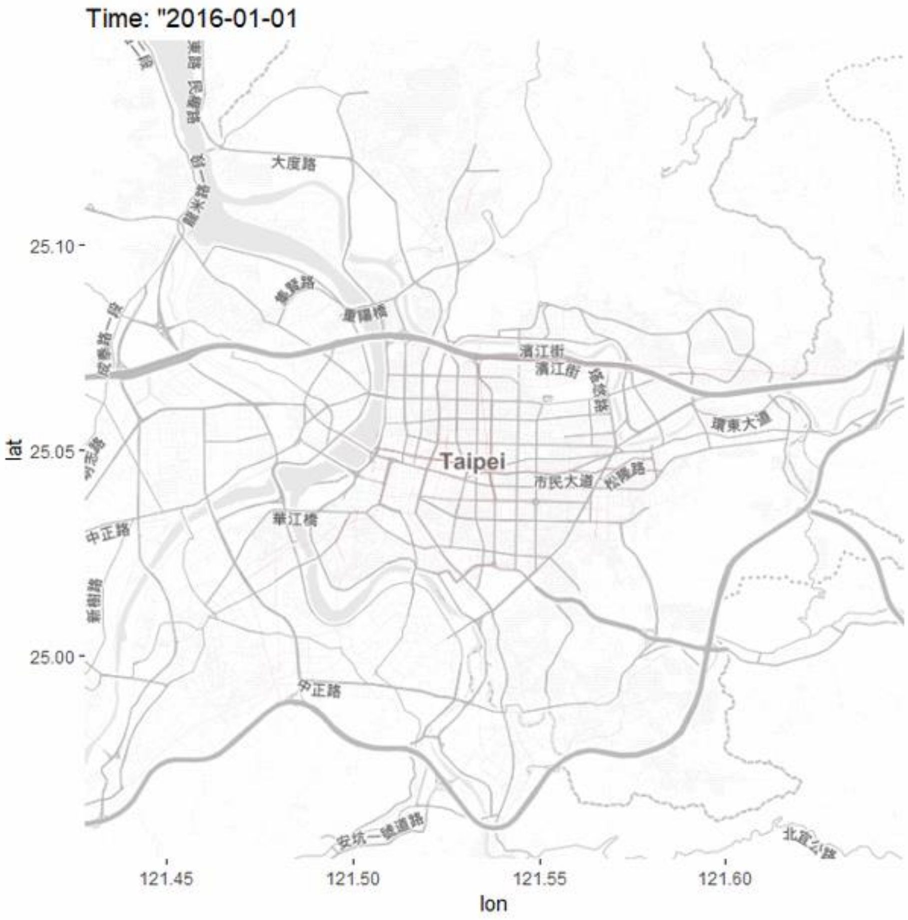
YouBike Animation

Popularity of each route by day.

```
df <- data.frame(rent_sta = d$rent_sta,
                 return_sta = d$return_sta,
                 lon = d$lon,
                 lat = d$lat,
                 group = d$group,
                 rent_time=anytime(d$rent_time),
                 day = anytime::anydate(d$rent_time),
                 dayn = d$day,
                 hour = d$hour)

map <- get_map(location = c(lon = 121.537801 , lat = 25.050139),
               zoom = 12, language = "zh-TW", maptype = "toner-lite")
test <- df  %>%
  dplyr::filter(day <= '2016-01-15')

ggmap(map, darken = c(0.4, 'white'))+
  geom_path(aes(x = lon, y = lat, group = group), color = 'red',
            size = 0.5, alpha = 0.0015, data = test) +
  labs(title = 'Time: "2016-01-15')
```

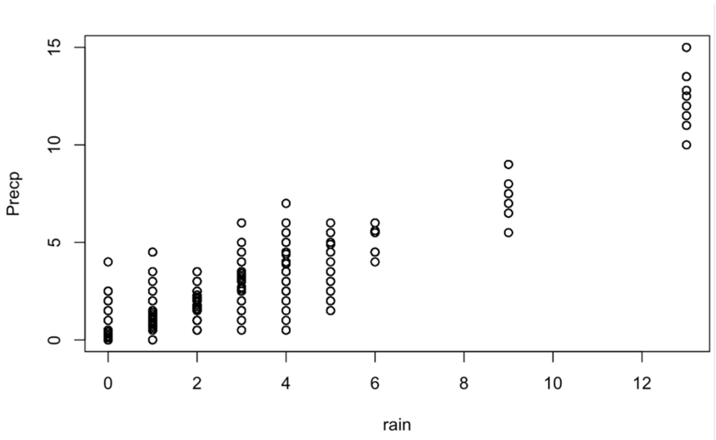


3

Multi-level Models

Model Building Strategy

- Data Sources:
  - Weather Data by "District" from Central Weather Bureau
  - Station_id to GPS from ggmap.
- Data Preprocessing:
  - Calculate net flow (return_count - rent_count) of every station by hour.
  - Append the weather data to the hourly data.
  - Pairs of rent and return for all time in the data.
- Model strategy:
  - Renting is a poisson process
  - Hour, rain are significant factors



Easy models do not work well

Easy_m:

$$rent\_count \sim possion(\lambda)$$
$$log(\lambda) = a + b_{hr}$$

Easy_m_hr_multi:

$$rent\_count \sim possion(\lambda)$$
$$log(\lambda) = a + b_{hr} + bF * Precp$$
$$b_hr[hr] \sim dnorm(\alpha, \sigma)$$
$$\alpha \sim dnorm(0, 10)$$
$$\sigma \sim dcauchy(0, 10)$$

Easy_m_hr:

$$rent\_count \sim possion(\lambda)$$
$$log(\lambda) = a + b_{hr} + bF * Precp$$

Easy_m_hr_inter:

$$rent\_count \sim possion(\lambda)$$
$$log(\lambda) = a + b_{hr} + bF * Precp + bF_{hr} * Precp$$

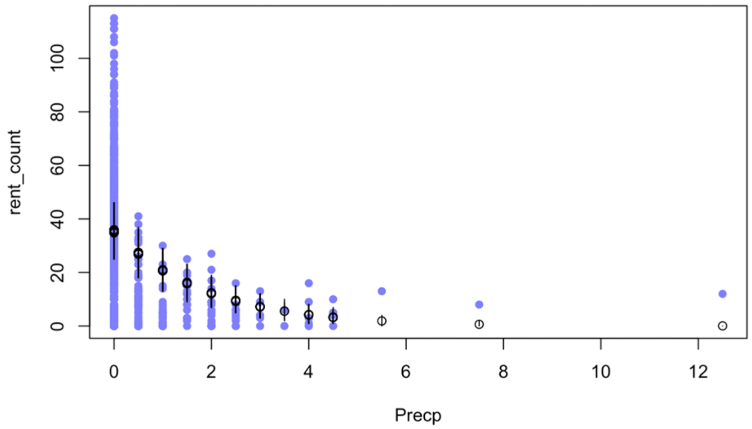
	WAIC <dbl>	pWAIC <dbl>	dWAIC <dbl>	weight <dbl>	SE <dbl>	dSE <dbl>
easy_m_hr_inter	7870.0	302.6	0.0	1	391.39	NA
easy_m_hr	8081.8	198.9	211.8	0	437.28	140.54
easy_m_hr_multi	8088.0	199.5	218.0	0	438.46	143.10
easy_m	18857.2	40.0	10987.2	0	619.38	729.85

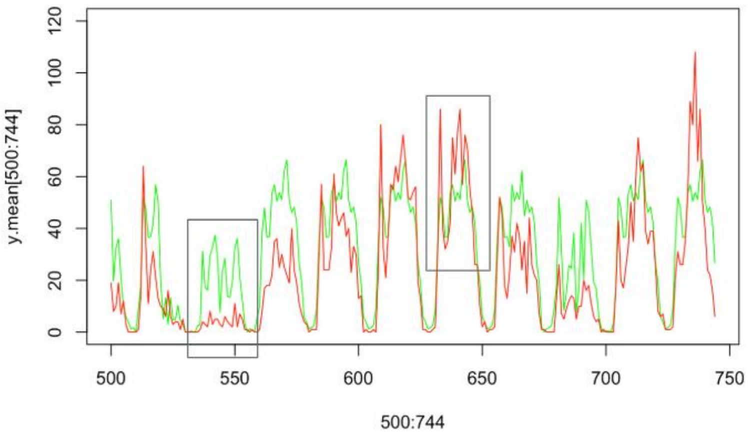
Plot, Plot, Plot

- Within Model Easy_m_hr_inter:
  - Overestimating when there's low rain, yet underestimating when there's much rain.

    ---> Use log to mitigate this effect.
  - There are spikes on raining days.

    ---> Accumulation may work.





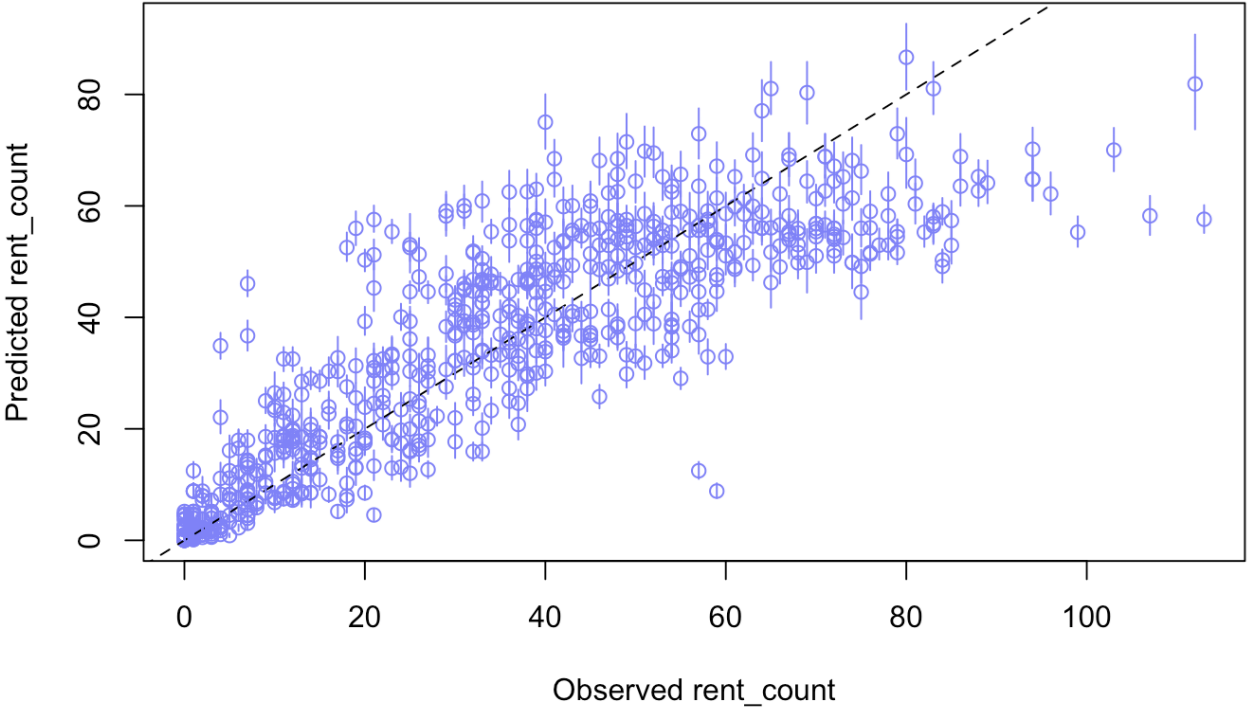
Log and Accumulated Rain work magic.

$$\begin{gathered} rent\_count \sim possion(\lambda) \\ log(\lambda) = a + bF * log(Precp + 1) + bhr + bw \\ +bphr_h * log(Precp + 1) \\ +bA * log(Accum + 1) \\ +bahr_h * log(Accum + 1) \\ +bwhr * bhr \end{gathered}$$

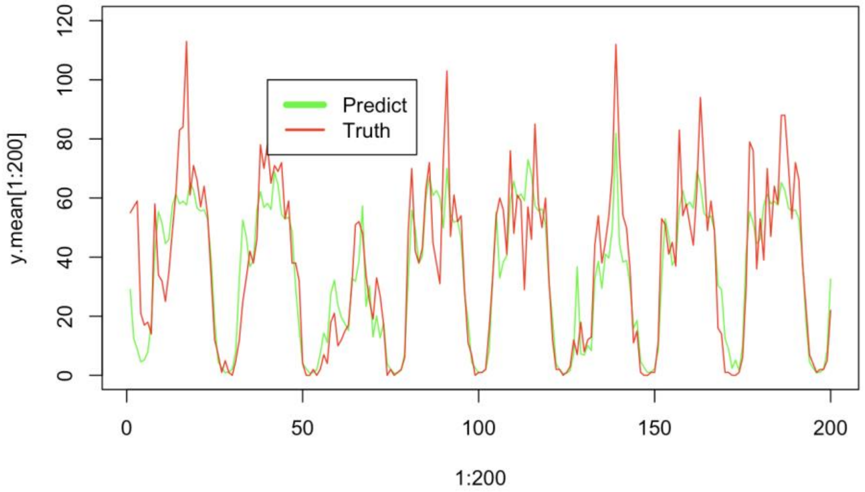
Model improves quite a bit when applying log precipitation as well as accumulative.

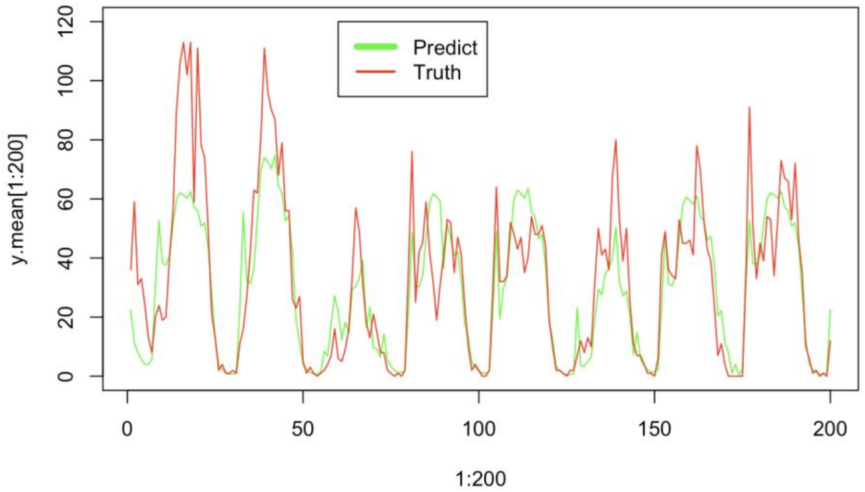
	WAIC <dbl>	pWAIC <dbl>	dWAIC <dbl>	weight <dbl>	SE <dbl>	dSE <dbl>
accum_hr_inter_logP	6892.7	300.3	0.0	1	251.78	NA
easy_m_hr_inter	7870.0	302.6	977.3	0	391.39	198.70
easy_m_hr	8081.8	198.9	1189.1	0	437.28	263.23
easy_m	18857.2	40.0	11964.5	0	619.38	673.33

Plot, Plot, Plot

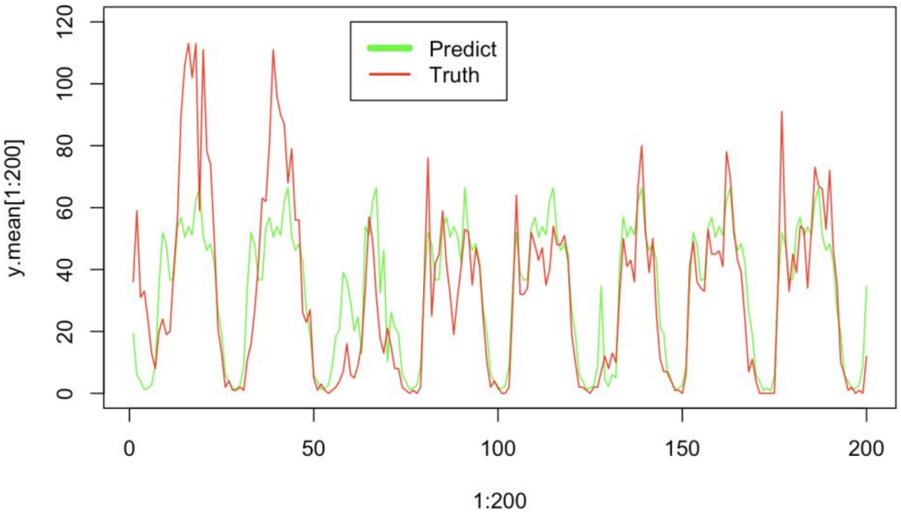


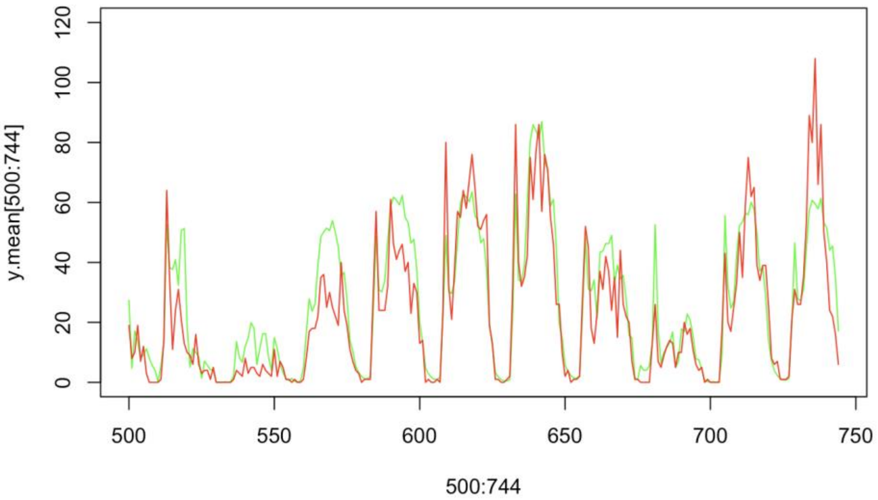
Model seems to fit well for known station - 45





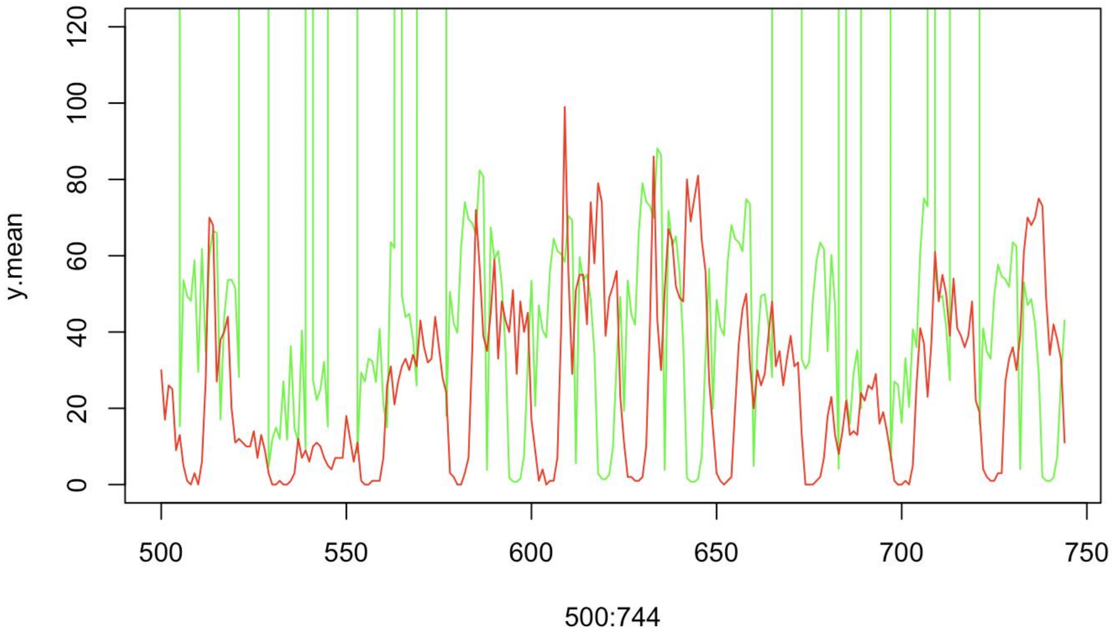
Model seems to fit well for known station - 132





However, when tested by unknown time...

Trained with hour 1:500, test with 500:744



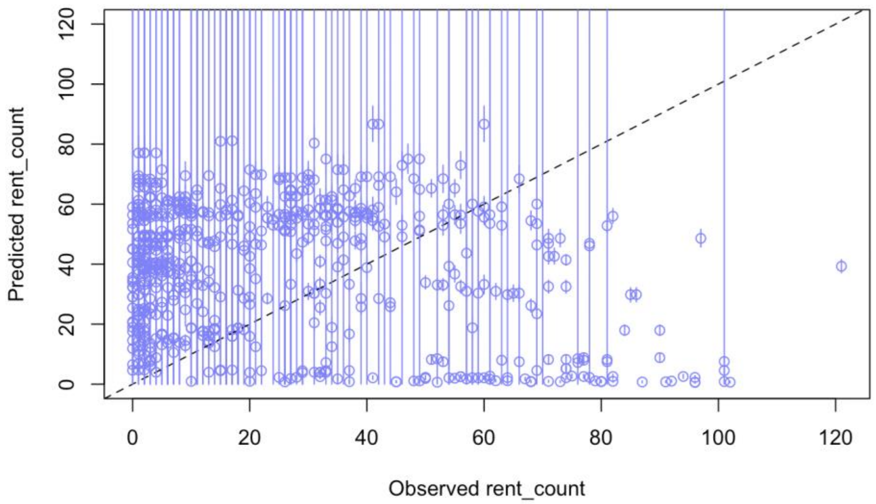
Why the model sucks?

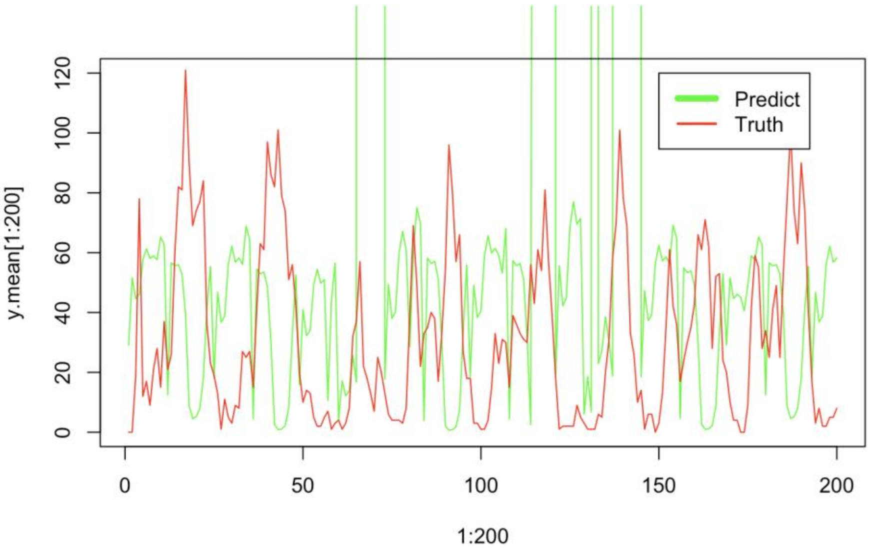
$$
\begin{gathered}
rent\_count \sim possion(\lambda) \\
log(\lambda) = a + bF * log(Precp + 1) + bhr + bw \\
+bphr_h * log(Precp + 1) \\
\boxed{+bA * log(Accum + 1)} \\
+bahr_h * log(Accum + 1) \\
+bwhr * bhr
\end{gathered}
$$

Some combinations are rare, some correlation should be in a prior of hr_r and bahr_h!

However, when tested by unknown time...

It seems to cause some extreme values.





Where's the hope?

- Challenges
  - No clue for predicting holiday.
  - Cross-Station Prediction
  - Future Prediction

- Adaptive Predictions
  - Pre-defined correlation between stations
  - More dummy variables
  - More levels included in a multilevel model
  - **Time series!**

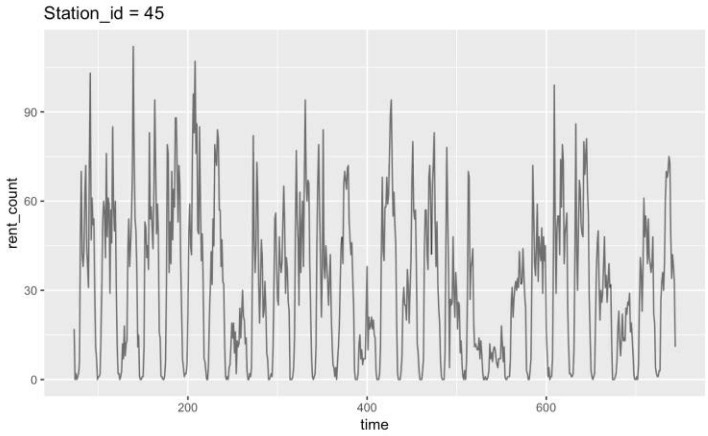
4

Gaussian Process Model

Model Building Strategy

- Time series data
  - Hourly rent and return count of bikes in January 2017
  - Non-stationary mean and variance
  - Clear weekly periodic component

- Strategy
  - Application of a time series forecasting using Gaussian process to predict the future bikes rent counts.
  - Use the data a station for model training and test to avoid the time consuming problems.



Method: Gaussian Process

- Implement time series forecasting using Gaussian process
- Define the kernel = ***kernel1 + kernel2***

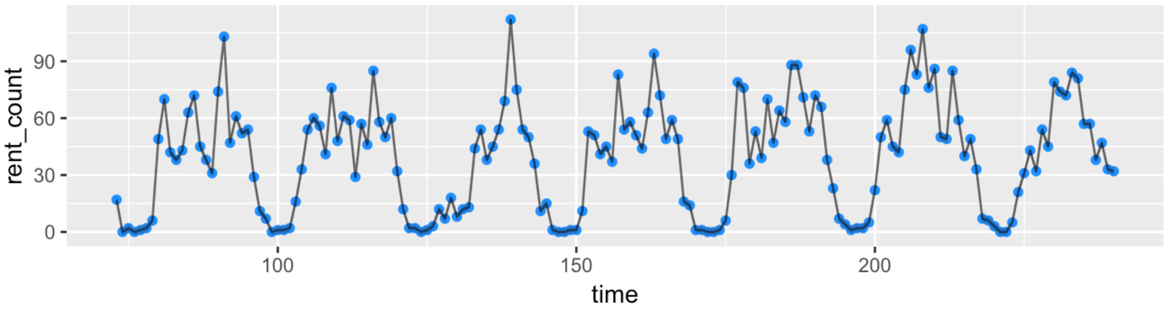
Kernel 1 $$k(t,t') = \sigma^2 exp\Big(-\frac{(t-t')^2}{2l_1^2}\Big)$$ Capture Near-by Relations

Kernel 2 $$k(t,t') = \sigma_2^2 exp\Big(-\frac{2sin^2(\pi(t-t')*1)}{l_2^2}\Big) exp\Big(-\frac{(t-t')^2}{2l_3^2}\Big)$$ Capture Periodicity

Strategy 1: Simple Time Series Techniques

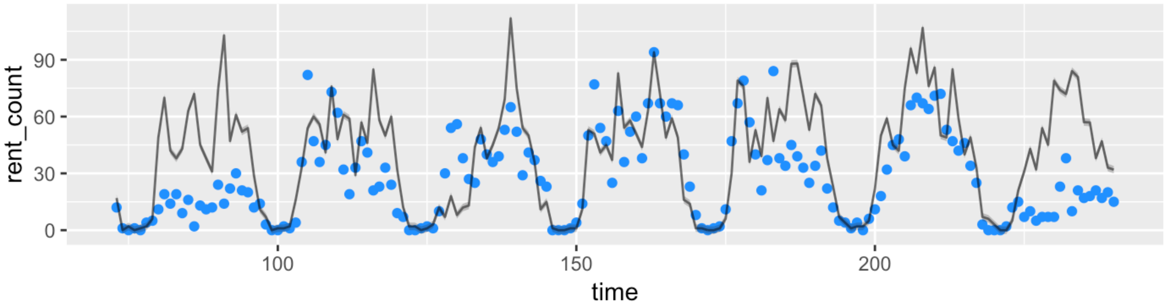
Use similar "hour of day" for predicting.

Station_id = 45 using Gaussian process to fit the training data



RMSE
0.0667

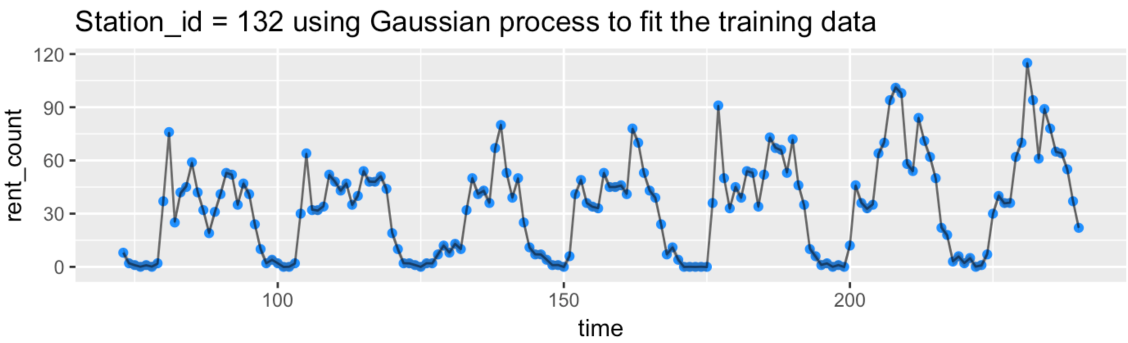
Station_id = 45 using Gaussian process to fit the test data



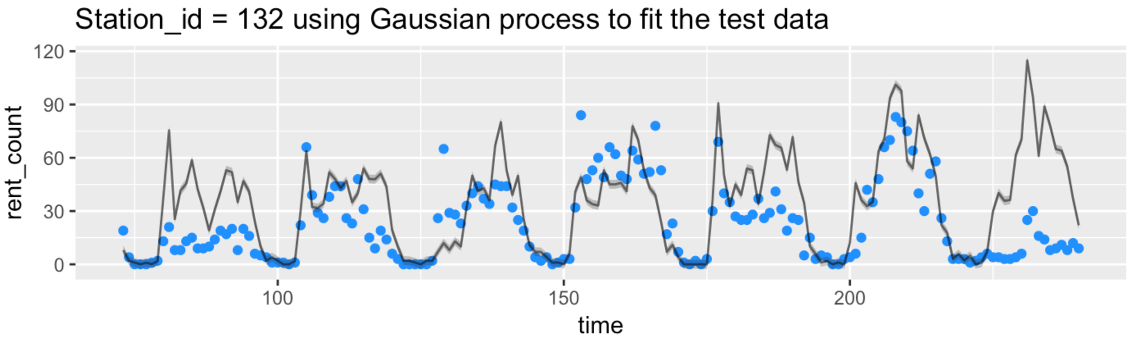
RMSE
25.1455

Strategy 1: Simple Time Series Techniques

Use similar "hour of day" for predicting.



RMSE
0.1654



RMSE
24.36

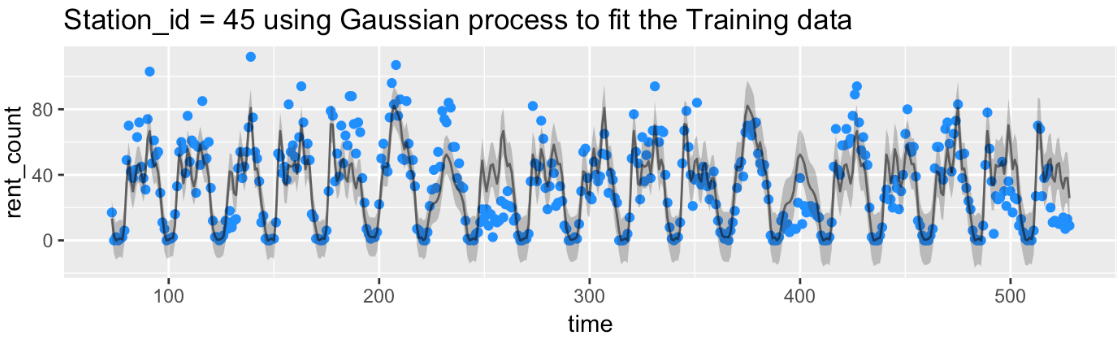
Strategy 2:

- Data Preprocessing
  - Derived variable weekhr: the sequence number by total hours of a week (168hr/ week)

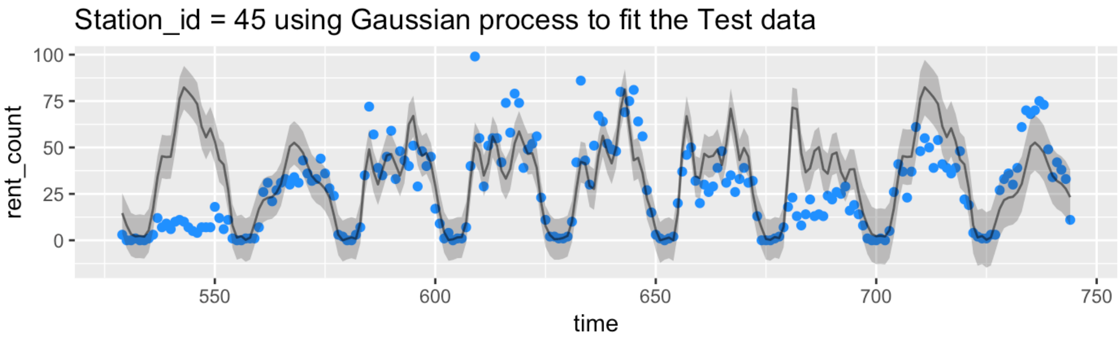
- Train-Test Split
  - First 3 weeks for training.
  - The last week for testing.

	date	weekhr
1	2016-01-04	1
2	2016-01-04	2
3	2016-01-04	3
4	2016-01-04	4
5	2016-01-04	5
6	2016-01-04	6
7	2016-01-04	7
8	2016-01-04	8

To Avoid overfitting problem...



RMSE
14.7352



RMSE
20.6208

Recommendations for the model

- Time series
  - Try using multivariate time series to train the model, since time series with a single value in a period may lose important information, such as raining feature.
  - Consider more on time series effect. For example, derive time-lagged variables.
  - Get more data to capture the trend effect of the time series.
- Gaussian Process
  - Fine tune hyper-parameters of the kernel.
  - Try other non-Gaussian distribution to train the model. For instance, Poisson distribution
  - Add white noise to the model to capture more uncertainty.

5

Takeaway

Takeaway

- There are certain patterns with regard to the trend of the bike's net inflow. Predicting the number of inflow can help YouBike company form better dispatching plan.

- With the help of multilevel models, compared to simple single-level models, we can get a smaller WAIC.

- Models are flexible and objective; as long as we can figure out how to improve the model, either by adding new predictor or changing the type of distribution, we can predict better.

Takeaway

- Time series can greatly improve the model, yet there is still plenty of rooms for improvement.

- To improve the performance of the model, here lists some potential ways.
    - We may improve our model using distance between stops as a covariance matrix. This matrix can be used as a predictor for the model.
    - The nature for each station might serve as a critical predictor. For example, whether or not a station is near busy bus stations, or whether the station is in a residential can be important to the pattern of net inflow.
    - Add white noise, time lagged variables, etc. to improve model performances.

Thank you for listening.