Data:
* *Registered* * ridership data on **weekdays** in **June** 2014

Step 1 P(s): Starting Station

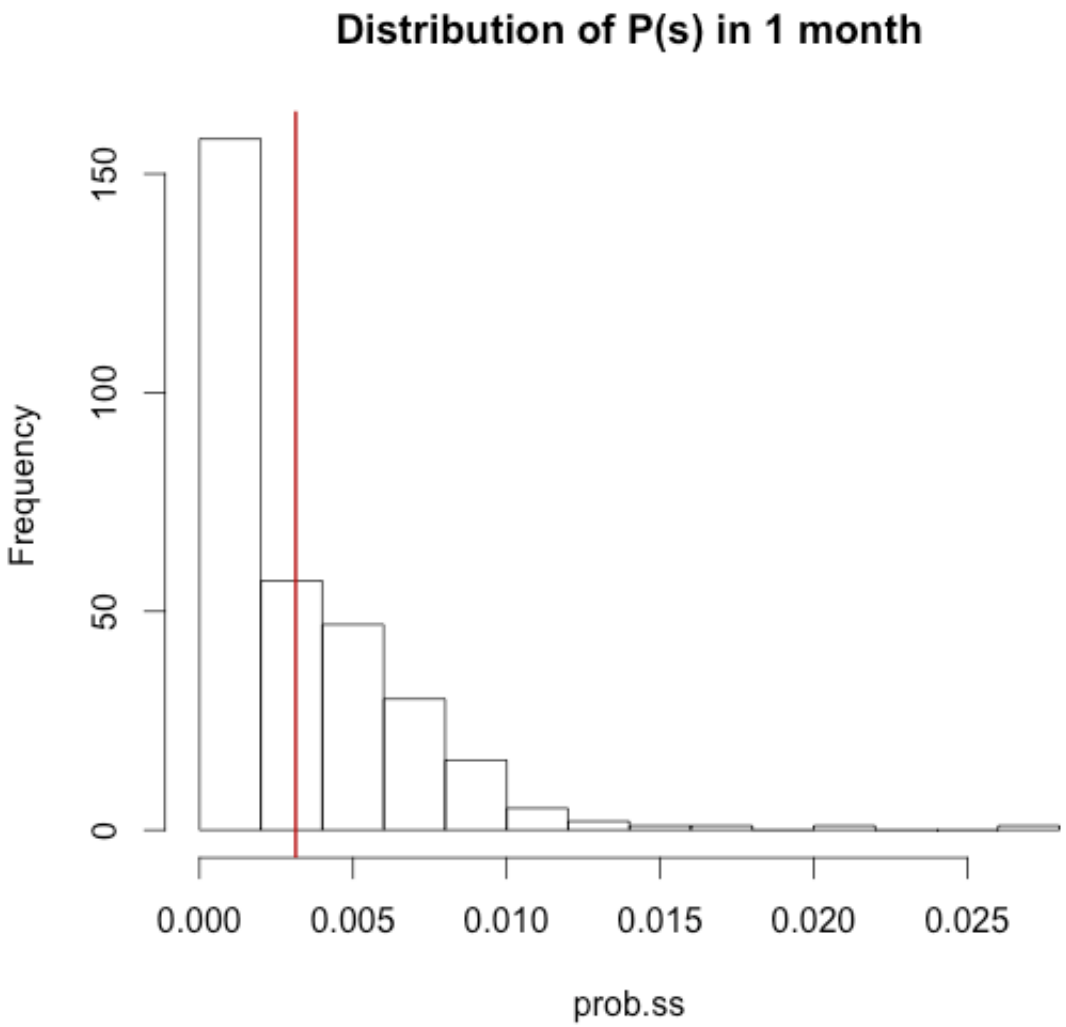
Method 1: empirical sample proportion

I using following formula to calculate the proportion:

$$P(s_i) = \frac{\text{\# of rides starting from station i}}{\text{Total rides}} \text{ for each weekday}$$

$$\hat{P}(s_i) = \frac{\sum_{d=1}^{n} P(s_i)}{n}$$

`n` is the # of days in the observation.

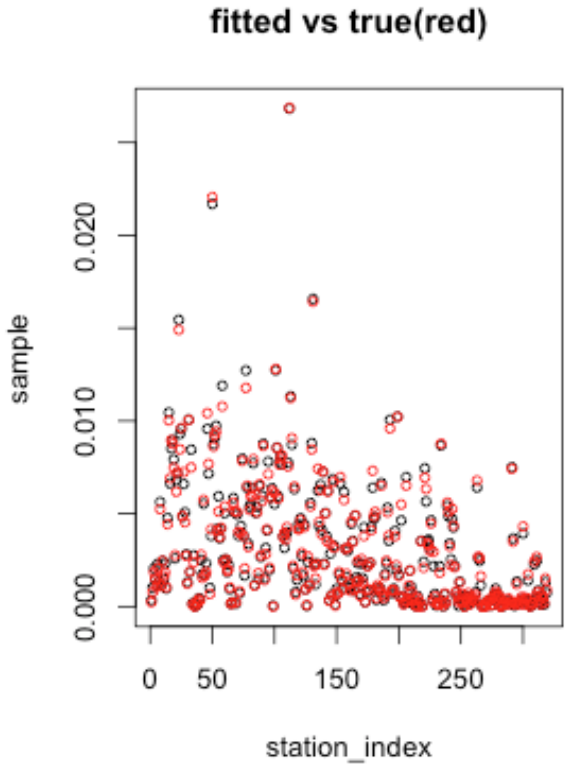


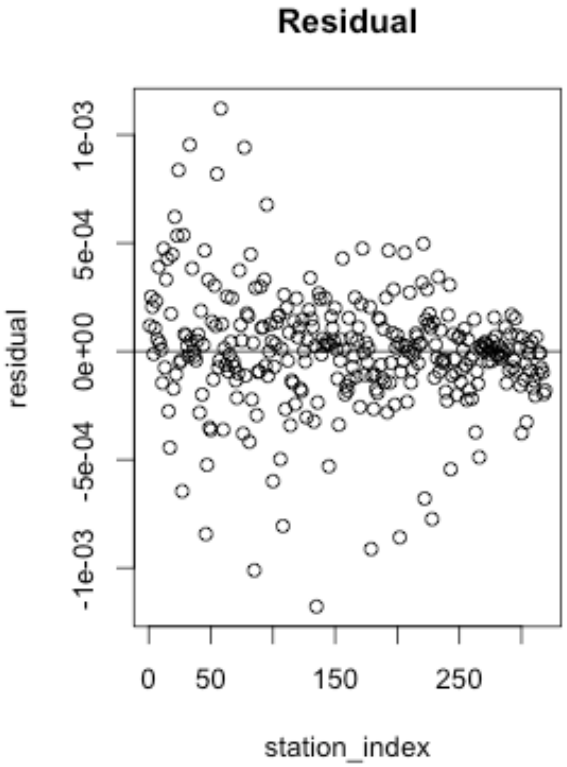
> The red line indicates the probability when rides are equally distributed among stations. We see more than 2/3 of stations' # of starting rides are below the average.

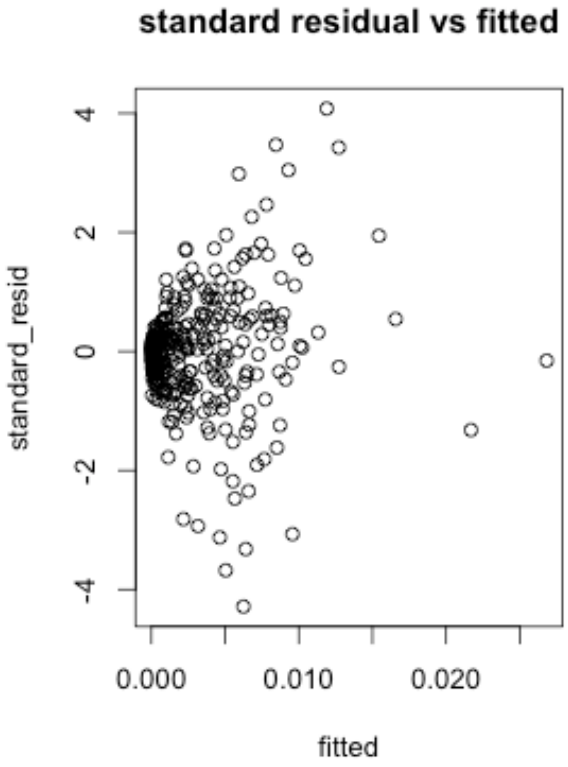
Using the above formula, `mse = 3.987214e-07` when all 21 weekdays in June are included.

To have independent sample and test set, I pick 15 random days as training set and the rest 6 days as testing set.

```
fitted = mean(P(s_i)) when i in range(1,15)
true = mean(P(s_i)) when i in range(16,21)
residual = fitted - true
```



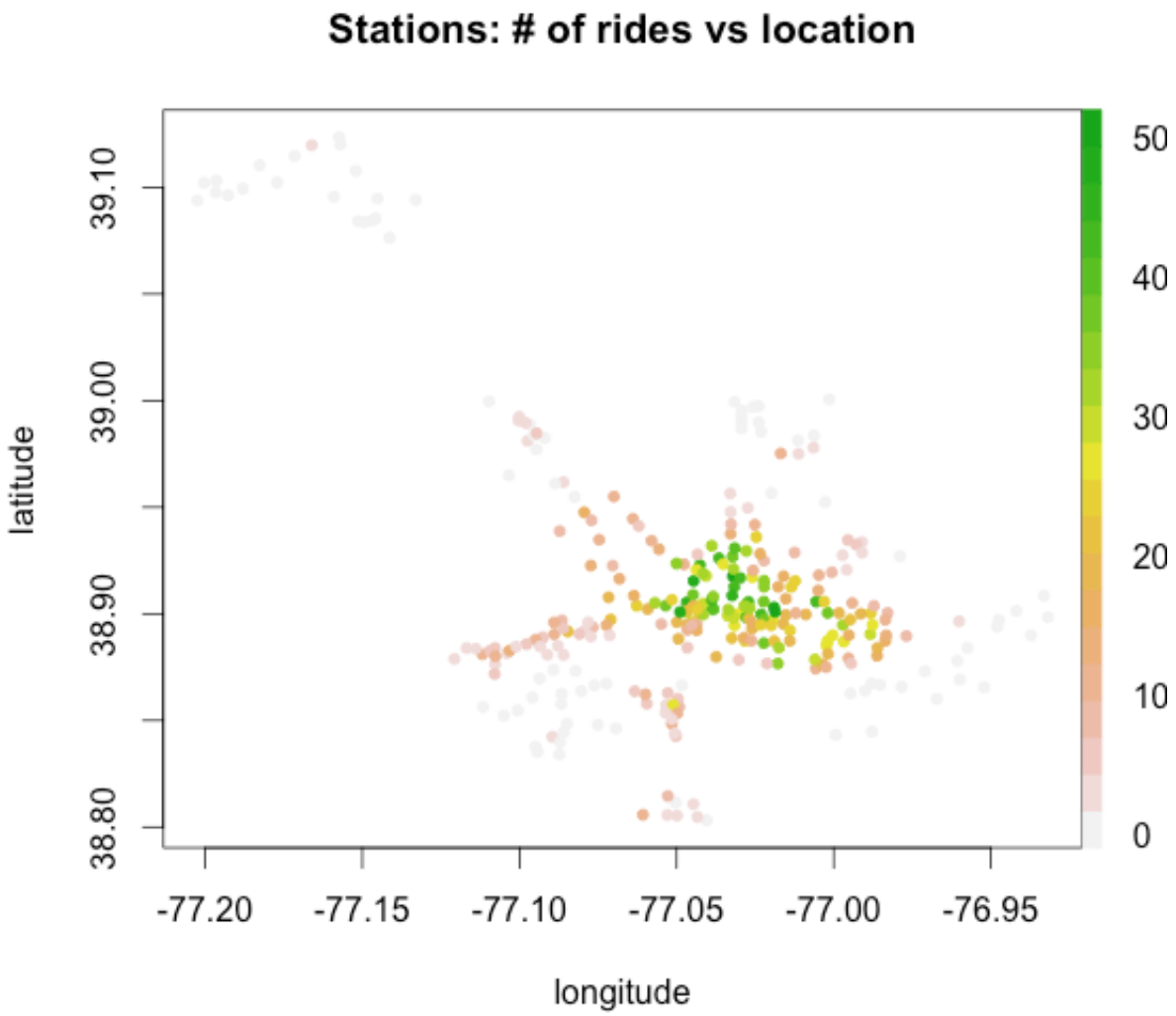





```
MSE = 0.005723973
```

> From *figure fitted vs true*, we see the red dots(true proportion) are close to the black dots(predictions). From *residual*, we see residual is associated with fitted value.

Method 2: multinomial Regression



> This figure plot all rides in june with its starting station. The stations in the center tend to have more traffic.

There are 319 stations and each station have unique location, therefore, multinomial regression might not work in this case. It is possible that P(s) follows bivariate normal distribution.

Instead, I use linear regression with formula

```
P(s)~latitude + I(latitude ^ 2) +  longitude + I(longitude ^ 2) +
latitude * longitude
```

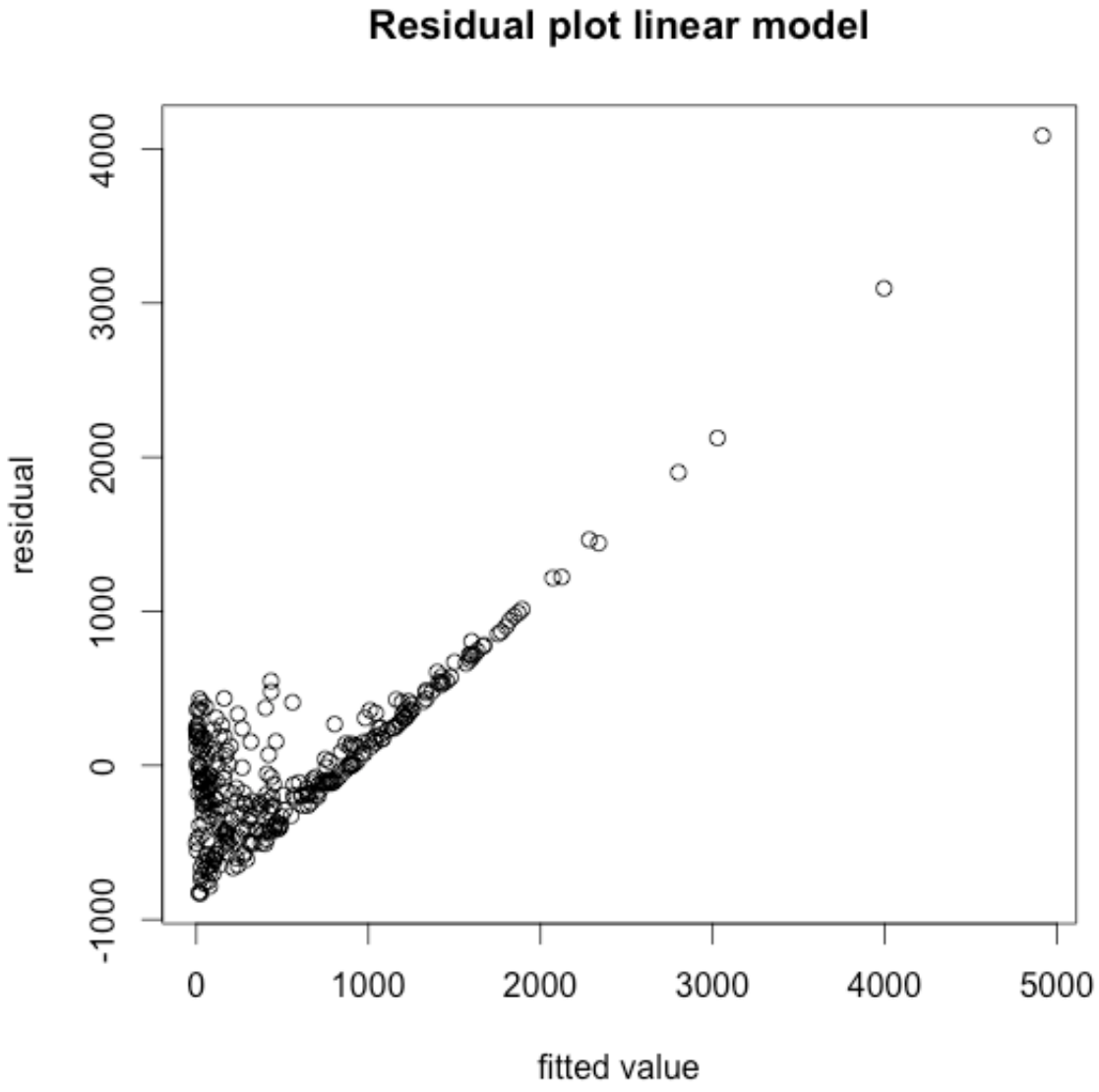
The output is

```
Residuals:
       Min         1Q     Median         3Q        Max
-0.0045241 -0.0017880 -0.0005216  0.0012795  0.0222865

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.946e+03  3.470e+02  -5.608 4.51e-08 ***
latitude           -3.241e+01  5.518e+00  -5.874 1.09e-08 ***
I(latitude^2)      -3.596e-01  5.180e-02  -6.942 2.23e-11 ***
longitude          -6.689e+01  1.041e+01  -6.423 4.97e-10 ***
I(longitude^2)     -6.321e-01  7.869e-02  -8.033 1.95e-14 ***
latitude:longitude -7.839e-01  9.132e-02  -8.584 4.36e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.002918 on 313 degrees of freedom
Multiple R-squared:  0.3201,    Adjusted R-squared:  0.3093
F-statistic: 29.48 on 5 and 313 DF,  p-value: < 2.2e-16
```

All coefficient are significant and a station's P(s) is associated with its location, but it does not work well as it tends to predict higher P(s)(leverage?)



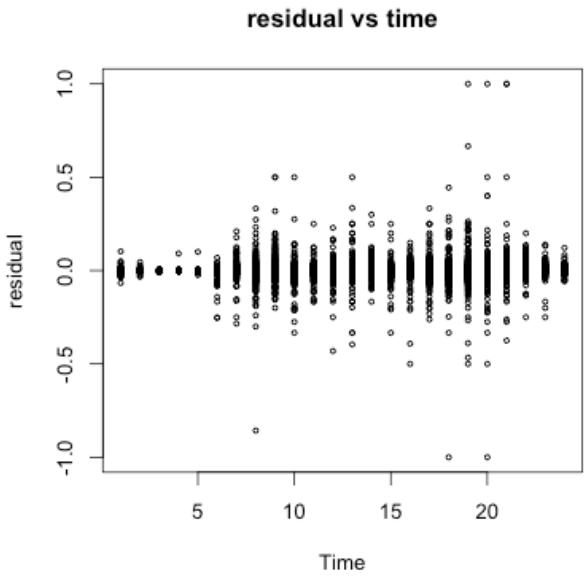
Step 2 P(t|s): Time in the day given stations

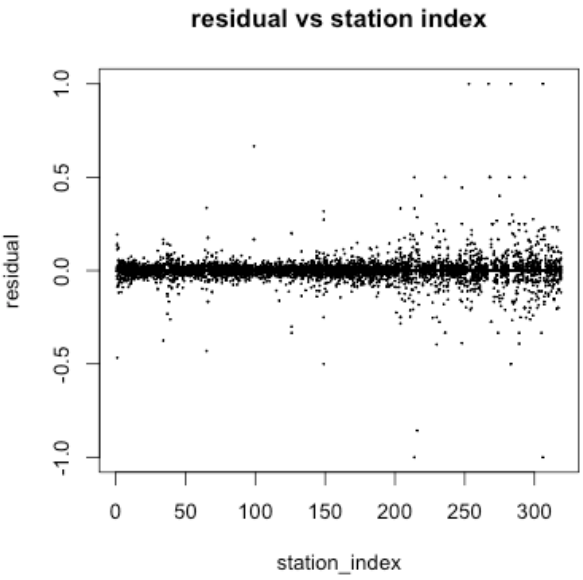
Method 1: empirical sample proportion

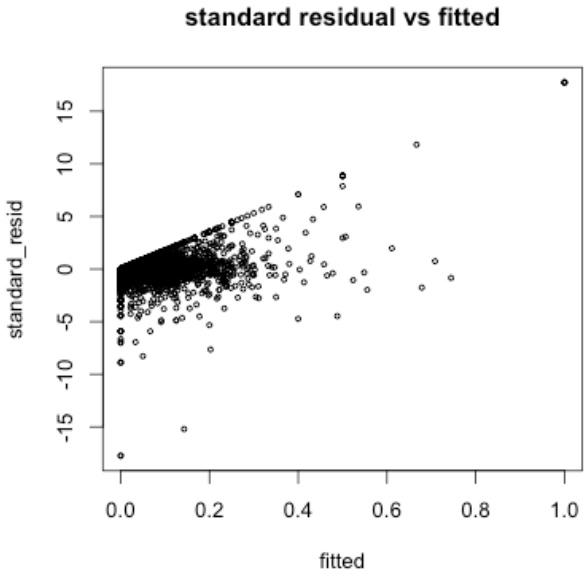
For each station, I calculated the probability for each hour. That is, there are 319(stations) * 24 (hours) in total.

$$P(t_j|s_i) = \frac{\text{\# of rides starting in jth hour}}{\text{Total rides of station i}}$$

$$\hat{P}(t_j|s_i) = \frac{\sum_{d=1}^{n} P(s_i|t_j)}{n}$$



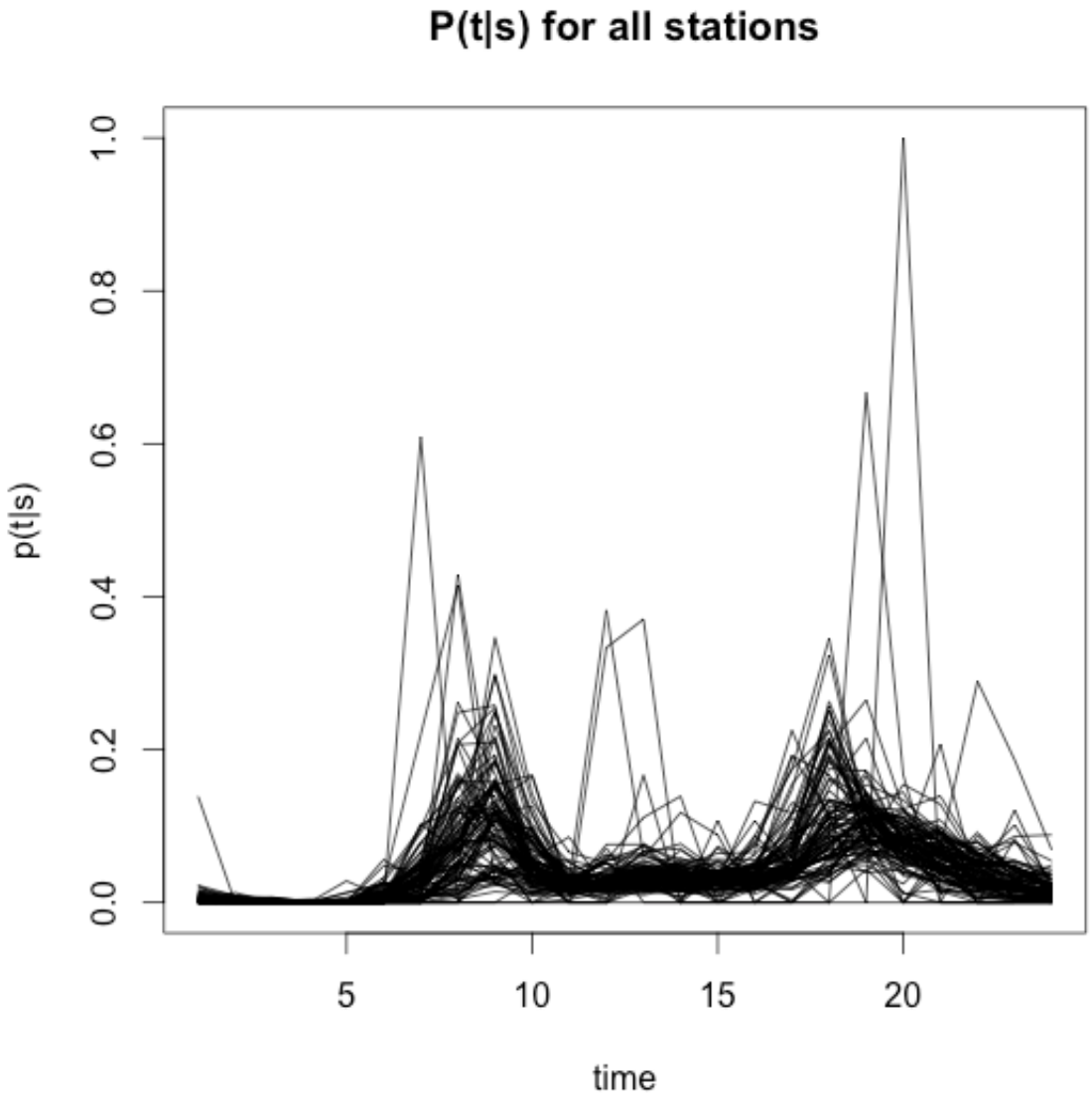




> The residual tends to be bigger in rush hours .
> Stations with higher index have higher residual.

Method 2: Clustering

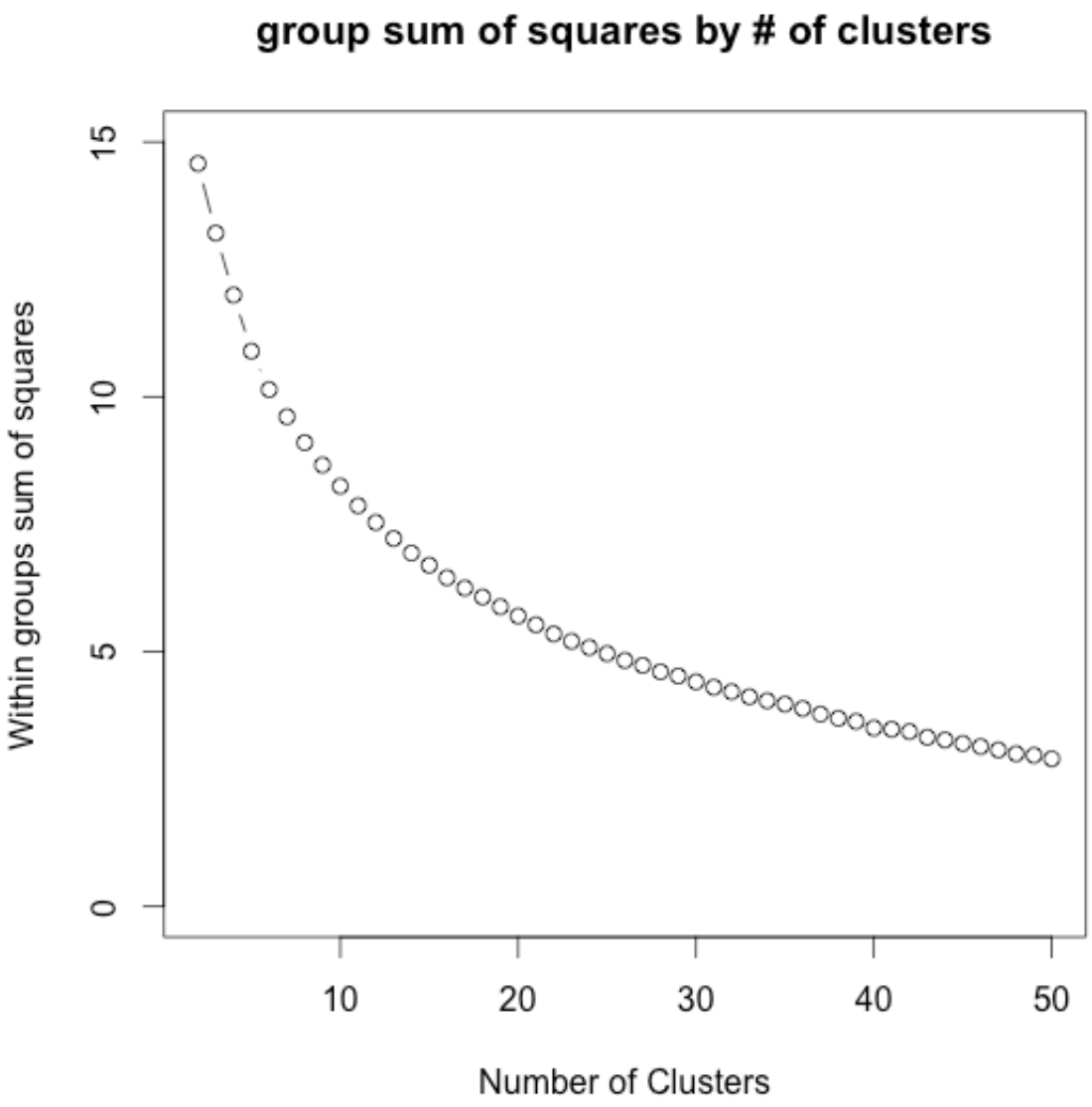
I use the same formula as method 1. Then I have a 24 * 319 matrix.



k-means algorithm

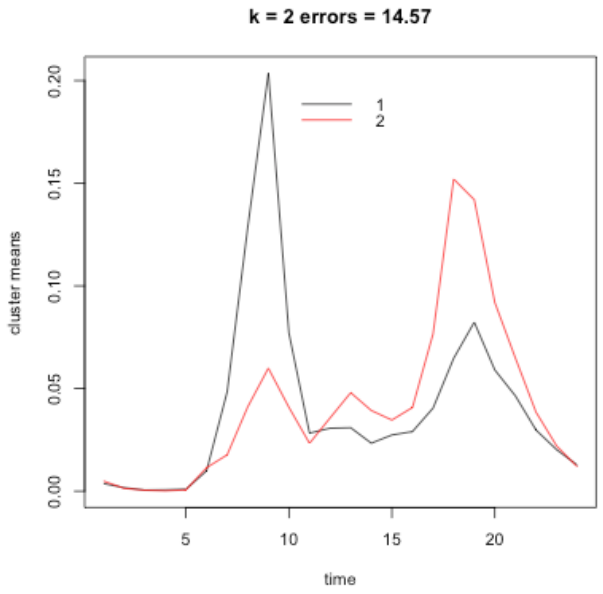
k-means algorithm will return different result with different initialization. To pick the best clustering, I run the algorithm for 1000 times and pick the clustering with smallest total within groups sum of square errors. (According to my experient, 1000 is still not big enough to get the global minimum, but the variance of sse are not big especially for small k).

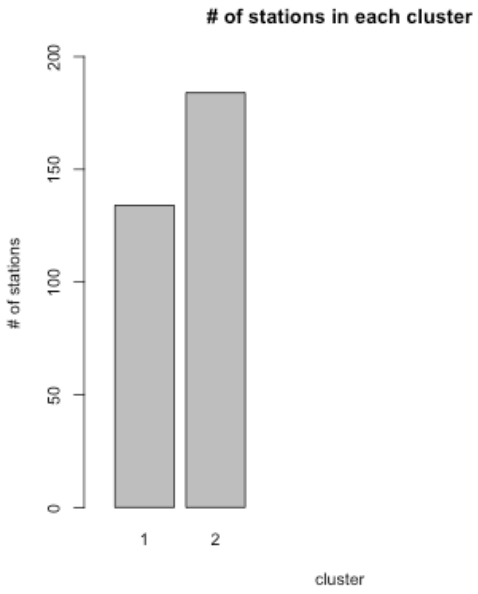
I use the following plot to decide the range of k(# of clusters)

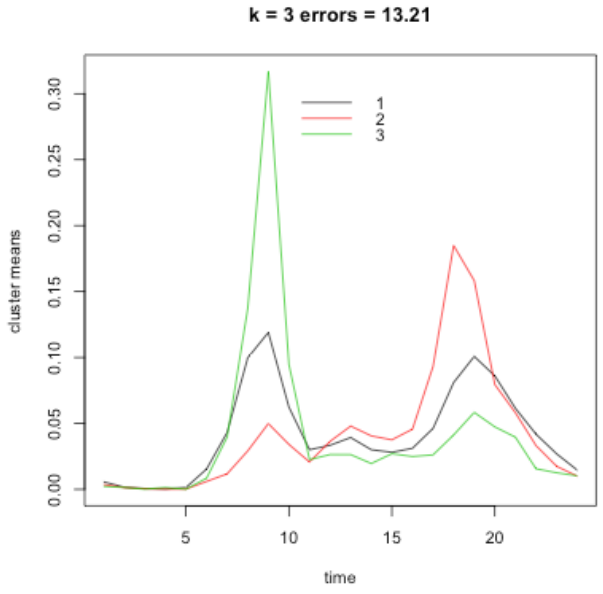


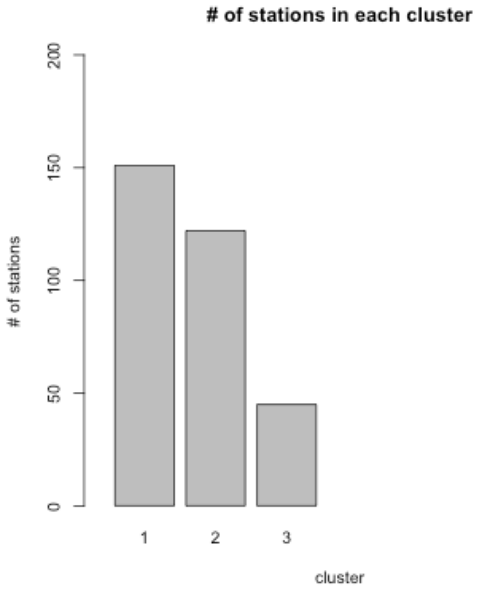
> There is no “elbow points” in the plot. So I will analyze k in range of (2 - 8) manually and choose the best k.

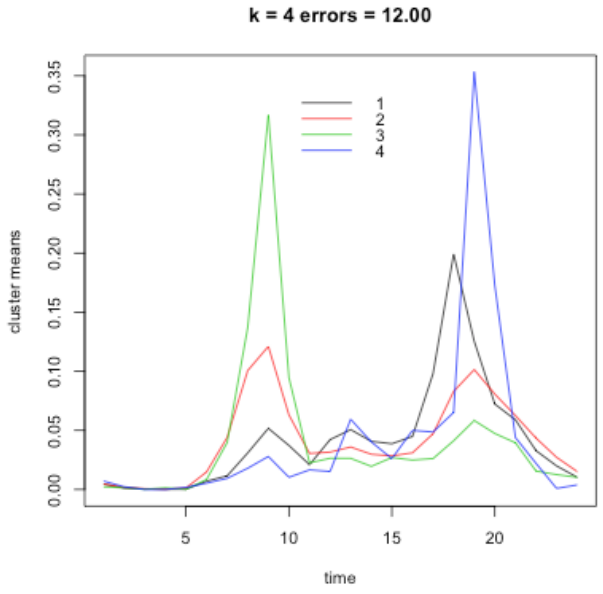
clustering with k = 2,3,4

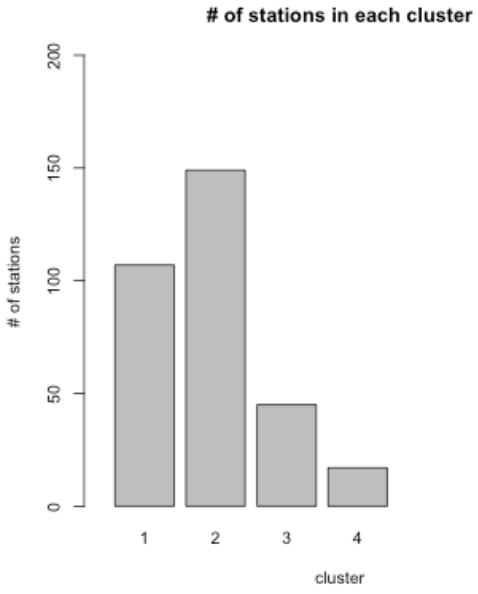




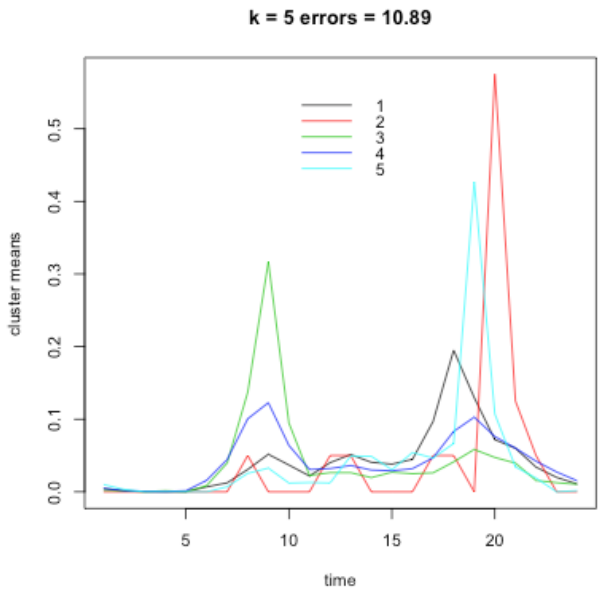


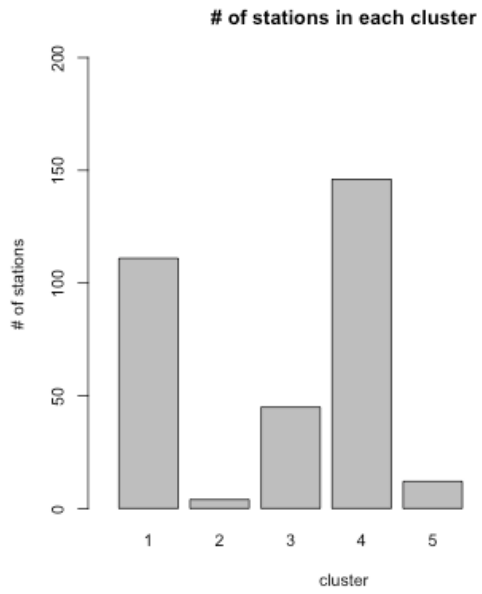


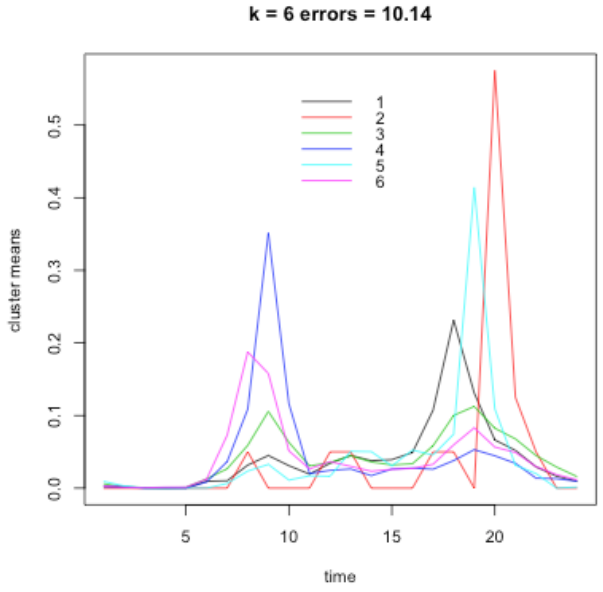


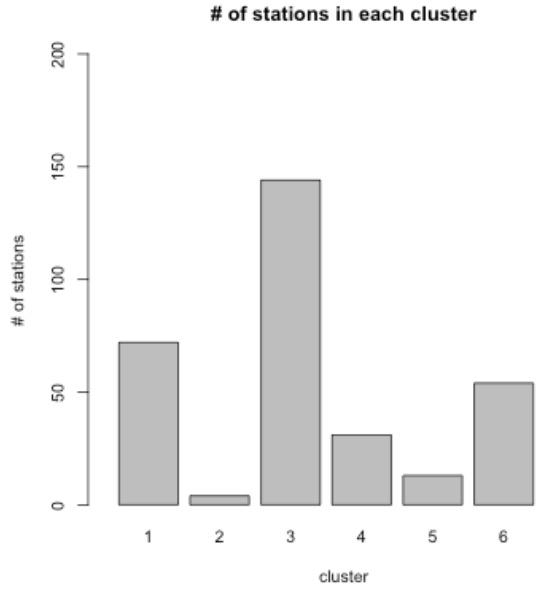


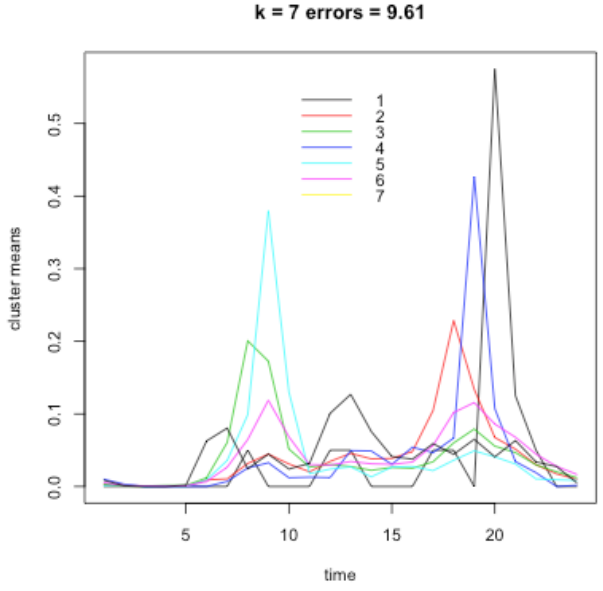
> **k = 2** : (morning)< evening(8pm)) & (morning > evening)
> **k = 3** : (k = 2) + (morning >> evening)
> **k = 4**: (k = 3) + (morning << evening(7pm))

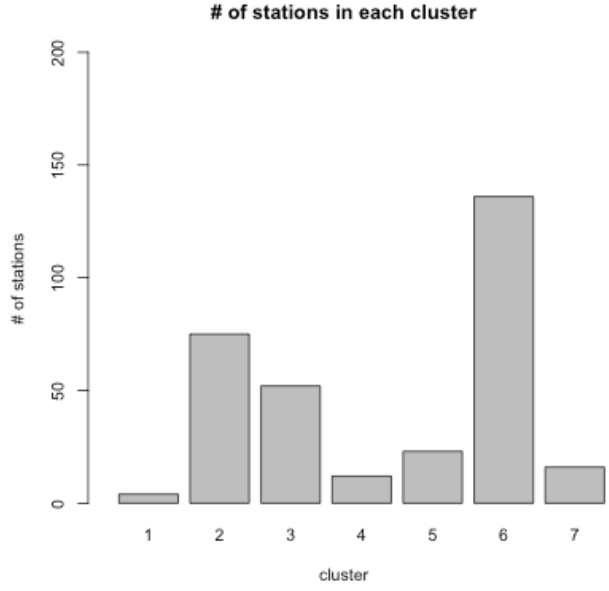

k = 5 errors = 10.89
cluster means
time
1
2
3
4
5


# of stations in each cluster
# of stations
cluster


k = 6 errors = 10.14
cluster means
time
1
2
3
4
5
6


# of stations in each cluster
# of stations
cluster


k = 7 errors = 9.61
cluster means
time
1
2
3
4
5
6
7

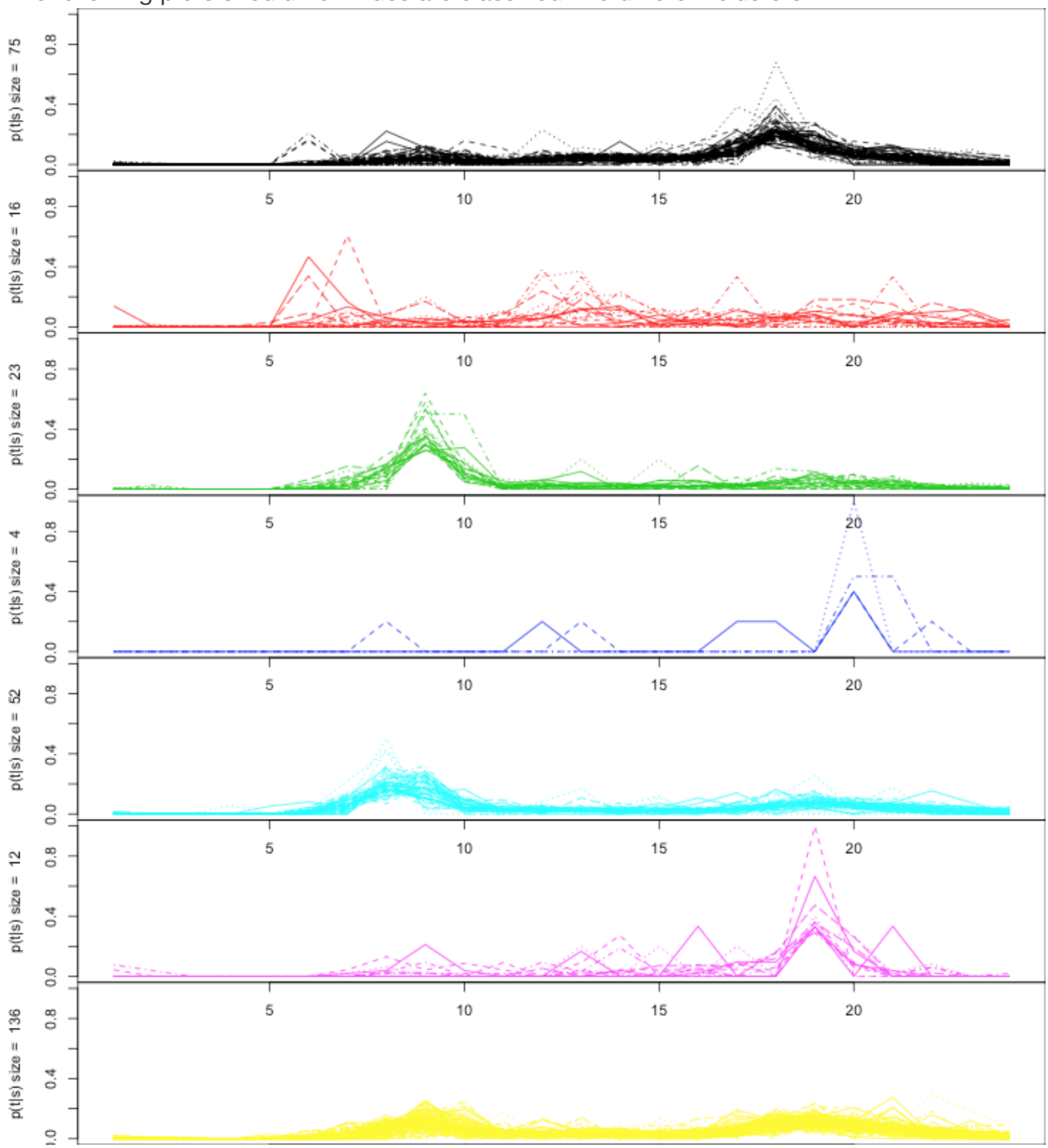

# of stations in each cluster
# of stations
cluster

> **k = 5**: (k = 4) + (peak at 8pm)(only 4 stations)
> **k = 6**: (k = 5) + (peak at 7pm)
> **k = 7** : (k = 6) + (peak at noon)

8 - 10
11 - 13

I pick `k = 7` as when k > 8, the improvement of sse by increasing k is smaller than 0.5.

The following plots should how rides are classified into different clusters.



The mse of each cluster

```
[1]0.005386887 [2]0.005149162 [3]0.002550790 [4]0.001933962
[5]0.003869108
[6]0.003617043 [7]0.007708812
```

Combining **P(s)**(from step 1.method 1)and **P(t|s)**(from step2.method 2), I use the following formula to calculate P(t,s):

$$P(s,t) = P(s) \times P(t|s)$$

We get `mse = 2.662469e-08` when include all rides in training set.
If I use the same training set and data set as in Step 1, `mse = 2.662529e-08`

Next Step:
P(e|t,s) ~ time + s_station(longitude, latitude) + e_station(longitude, latitude)