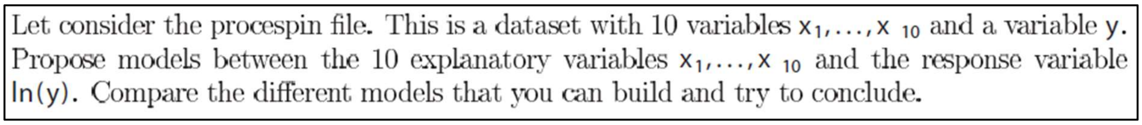
Data ScienceTech Institute
Advanced Statistical Analysis and Machine Learning
Final Project Report

S19 Cohort Motoharu DEI

January 28th, 2020

Contents

1. Exercise 1

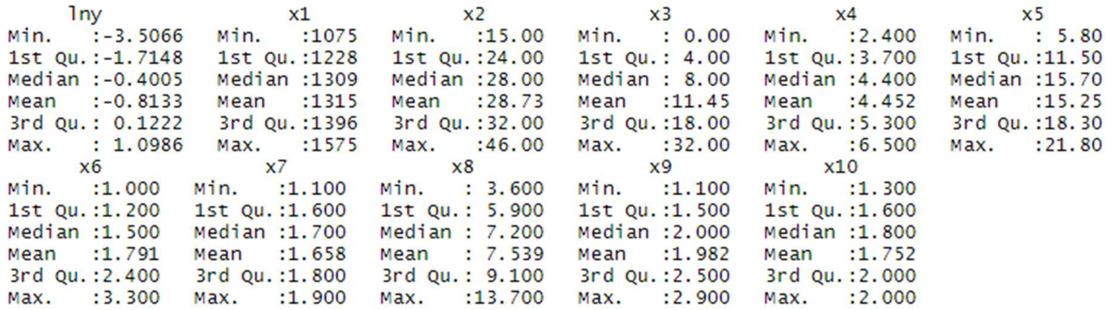

Let consider the procespin file. This is a dataset with 10 variables $x_1, \ldots, x_{10}$ and a variable y. Propose models between the 10 explanatory variables $x_1, \ldots, x_{10}$ and the response variable $\ln(y)$. Compare the different models that you can build and try to conclude.

I used R version 3.6.1 for this exercise and used RStudio version 1.2.1335.

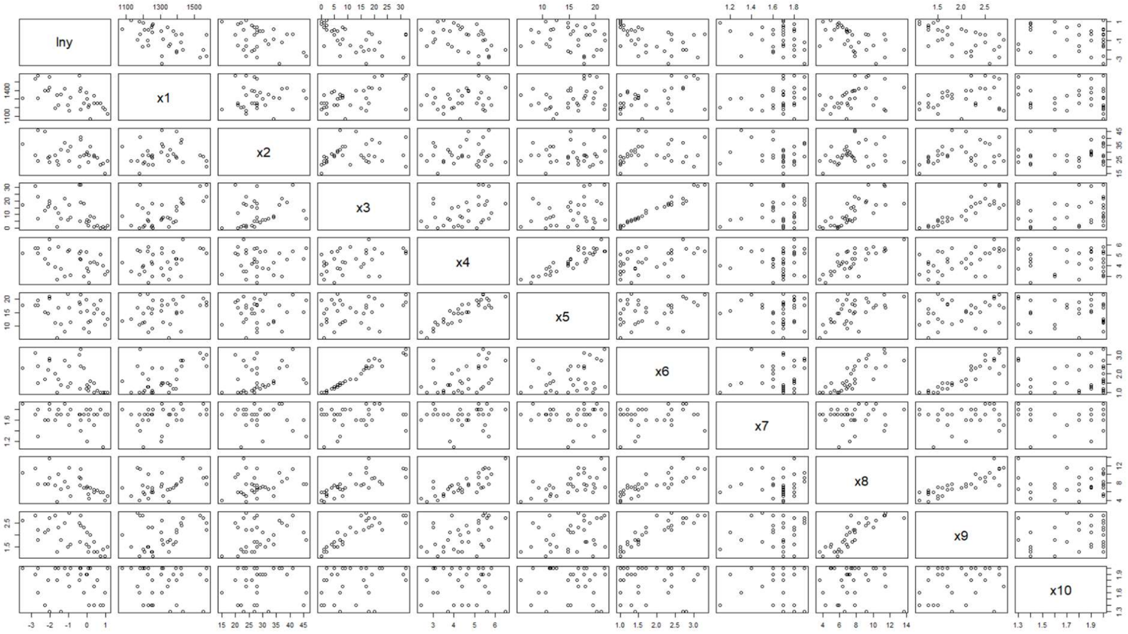
1.1. Initial glance at the data

I added a column as ln(y) in a name 'lny' and removed original y. There are just 33 observations.

Here's the summary of all column:


```
      lny                 x1              x2              x3               x4               x5       
 Min.   :-3.5066   Min.   :1075   Min.   :15.00   Min.   : 0.00   Min.   :2.400   Min.   : 5.80  
 1st Qu.:-1.7148   1st Qu.:1228   1st Qu.:24.00   1st Qu.: 4.00   1st Qu.:3.700   1st Qu.:11.50  
 Median :-0.4005   Median :1309   Median :28.00   Median : 8.00   Median :4.400   Median :15.70  
 Mean   :-0.8133   Mean   :1315   Mean   :28.73   Mean   :11.45   Mean   :4.452   Mean   :15.25  
 3rd Qu.: 0.1222   3rd Qu.:1396   3rd Qu.:32.00   3rd Qu.:18.00   3rd Qu.:5.300   3rd Qu.:18.30  
 Max.   : 1.0986   Max.   :1575   Max.   :46.00   Max.   :32.00   Max.   :6.500   Max.   :21.80  
       x6              x7              x8               x9              x10       
 Min.   :1.000   Min.   :1.100   Min.   : 3.600   Min.   :1.100   Min.   :1.300  
 1st Qu.:1.200   1st Qu.:1.600   1st Qu.: 5.900   1st Qu.:1.500   1st Qu.:1.600  
 Median :1.500   Median :1.700   Median : 7.200   Median :2.000   Median :1.800  
 Mean   :1.791   Mean   :1.658   Mean   : 7.539   Mean   :1.982   Mean   :1.752  
 3rd Qu.:2.400   3rd Qu.:1.800   3rd Qu.: 9.100   3rd Qu.:2.500   3rd Qu.:2.000  
 Max.   :3.300   Max.   :1.900   Max.   :13.700   Max.   :2.900   Max.   :2.000  
```

Also, the pair-wise scatter plots.



Some pairs of variables seemed having higher linear correlations such as (x3,x6), (x3,x9), (x4,x5), (x6, x8), (x6,x9), and (x8,x9).

The data size was smaller for the number of variables (33 data rows vs. 10 variables) and some high linear correlations between variables potentially indicated that in this data set, a special care needed to be taken for train-test split—not to get accidentally biased test set—, and variable selection or dimension reduction could play important role in the modeling.

1.2. Train-test split

Here, I split the data into two sets; training set and test set of 26 rows and 7 rows (equivalent to 80:20 split). Although the test set is commonly used for final model evaluation after the model selection over the validation set, **in this exercise I used test set for model selection because the number of observations was highly limited.**

Training set and test set have to hold the similar distributions in any variables. In order to secure that characteristic, I ran five different train-test splits with separate random seeds, compared the variables and executed t test, and finally chose the best random seed which provided the least biased train-test split.

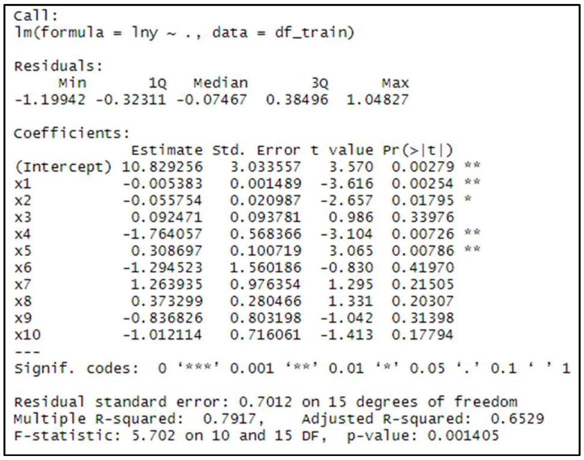
Here are the results.

Random Seed	Variable	Train Set Avg.	Test Set Avg.	Delta	p-value
1	ln(y)	-0.6	-1.62	1.02	12%
	x1	1306.54	1348	-41.46	42%
	x2	28.35	30.14	-1.8	61%
	x3	11.88	9.86	2.03	59%
	x4	4.4	4.64	-0.24	70%
	x5	15.24	15.29	-0.04	99%
	x6	1.79	1.8	-0.01	97%
	x7	1.65	1.7	-0.05	55%
	x8	7.47	7.81	-0.35	81%
	x9	1.98	1.99	0	99%
	x10	1.81	1.54	0.26	4%
2	ln(y)	-0.79	-0.9	0.12	86%
	x1	1306.35	1348.71	-42.37	34%
	x2	28.77	28.57	0.2	95%
	x3	11.69	10.57	1.12	75%
	x4	4.41	4.61	-0.21	66%
	x5	15.07	15.93	-0.86	63%
	x6	1.79	1.79	0.01	98%
	x7	1.65	1.7	-0.05	40%
	x8	7.35	8.23	-0.87	51%
	x9	1.98	2	-0.02	93%
	x10	1.77	1.69	0.08	52%
3	ln(y)	-0.88	-0.56	-0.33	54%
	x1	1323.69	1284.29	39.41	49%
	x2	29.58	25.57	4.01	3%
	x3	12.81	6.43	6.38	10%
	x4	4.63	3.79	0.85	9%
	x5	16.04	12.31	3.73	9%
	x6	1.88	1.46	0.42	18%
	x7	1.67	1.61	0.05	60%
	x8	7.97	5.94	2.03	3%
	x9	2.04	1.76	0.29	34%
	x10	1.79	1.61	0.17	20%
4	ln(y)	-1.02	-0.04	-0.99	4%
	x1	1338.38	1229.71	108.67	4%
	x2	29.08	27.43	1.65	46%
	x3	12.42	7.86	4.57	17%
	x4	4.56	4.04	0.52	15%
	x5	15.57	14.09	1.48	40%
	x6	1.87	1.5	0.37	15%
	x7	1.66	1.64	0.02	83%
	x8	7.78	6.64	1.14	8%
	x9	2	1.93	0.07	78%
	x10	1.73	1.84	-0.12	30%
5	ln(y)	-0.88	-0.57	-0.31	64%
	x1	1325.46	1277.71	47.75	25%
	x2	29.46	26	3.46	40%
	x3	12.88	6.14	6.74	8%
	x4	4.58	3.99	0.59	24%
	x5	15.73	13.46	2.28	31%
	x6	1.89	1.43	0.46	11%
	x7	1.68	1.56	0.13	25%
	x8	7.97	5.96	2.01	2%
	x9	2.08	1.6	0.48	5%
	x10	1.79	1.6	0.19	3%

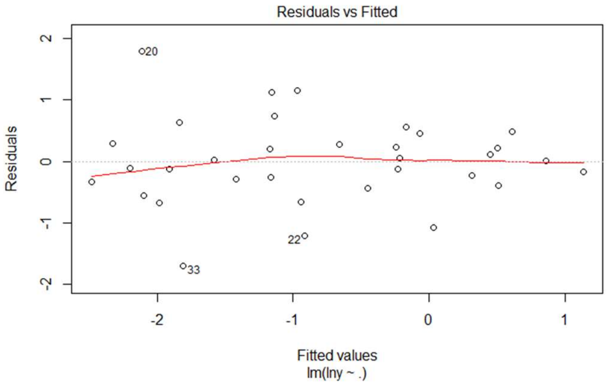
Seed=2 looked producing the most balanced train-test split.

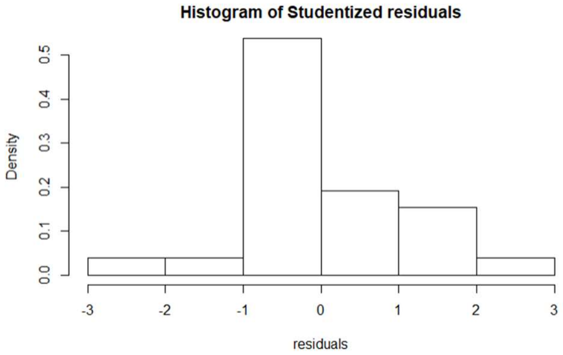
1.3. Linear regression using all variables (Model #1)

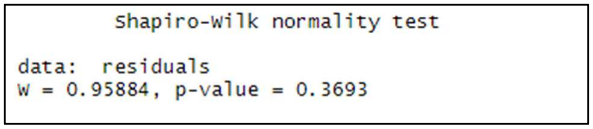
In Model #1, I used the all the variables for modeling without selection.



```
Call:
lm(formula = lny ~ ., data = df_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.19942 -0.32311 -0.07467  0.38496  1.04827 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 10.829256   3.033557   3.570  0.00279 **
x1          -0.005383   0.001489  -3.616  0.00254 **
x2          -0.055754   0.020987  -2.657  0.01795 * 
x3           0.092471   0.093781   0.986  0.33976   
x4          -1.764057   0.568366  -3.104  0.00726 **
x5           0.308697   0.100719   3.065  0.00786 **
x6          -1.294523   1.560186  -0.830  0.41970   
x7           1.263935   0.976354   1.295  0.21505   
x8           0.373299   0.280466   1.331  0.20307   
x9          -0.836826   0.803198  -1.042  0.31398   
x10         -1.012114   0.716061  -1.413  0.17794   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7012 on 15 degrees of freedom
Multiple R-squared:  0.7917,	Adjusted R-squared:  0.6529 
F-statistic: 5.702 on 10 and 15 DF,  p-value: 0.001405
```







```
        Shapiro-Wilk normality test

data:  residuals
W = 0.95884, p-value = 0.3693
```

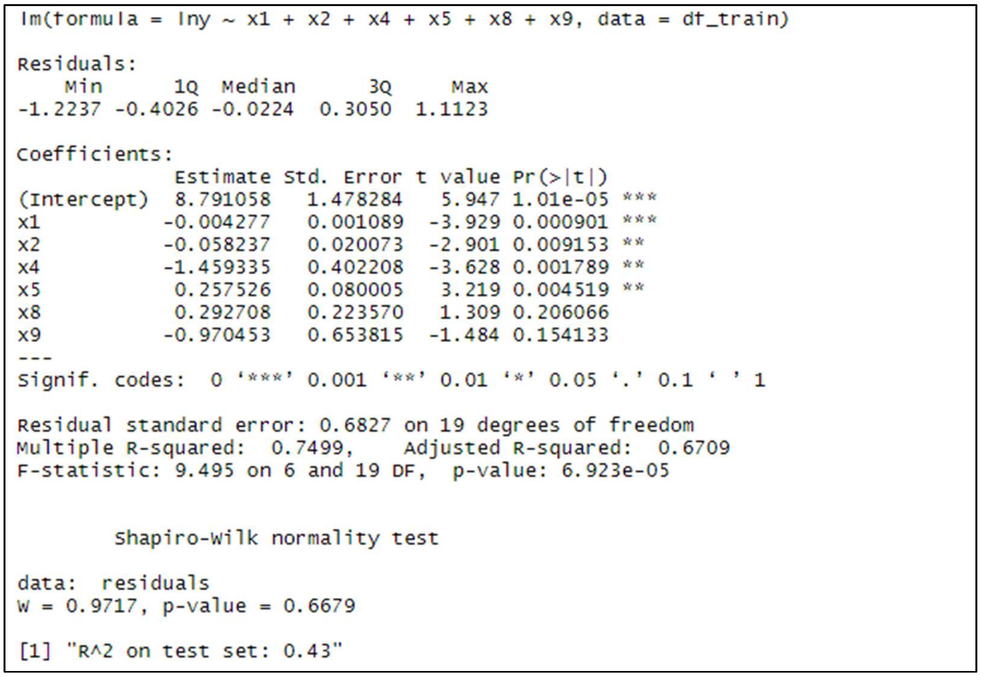
Model fit seemed good though there were many insignificant coefficients. However, **the model score on test set was R^2=0.263**, much lower than the training R^2=0.792, which indicated the sign of overfitting.

1.4. Linear regression with stepwise variable selection (Model #2)

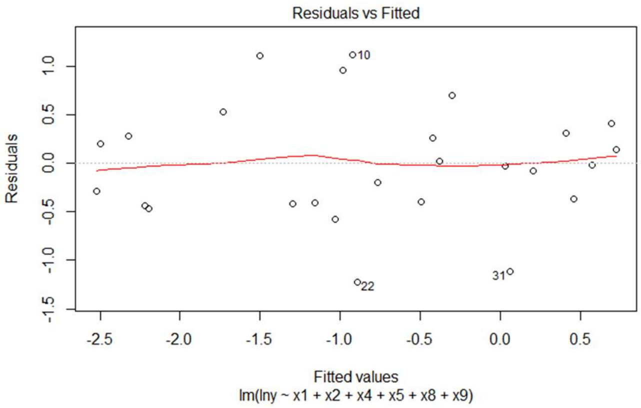
In Model #2, I used the stepwise variable selection over the linear regression model.

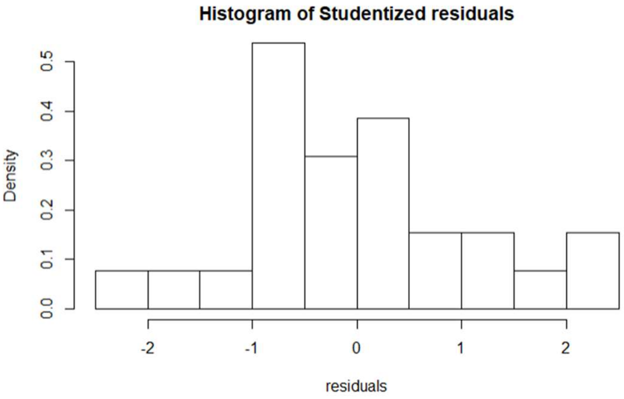
With *stepAIC* function on R, starting from full model, the function selected a model using six variables, x1, x2, x4, x5, x8, and x9.

I got the following coefficients and Shapiro test results on studentized residuals,



```
lm(formula = lny ~ x1 + x2 + x4 + x5 + x8 + x9, data = df_train)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2237 -0.4026 -0.0224  0.3050  1.1123 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.791058   1.478284   5.947 1.01e-05 ***
x1          -0.004277   0.001089  -3.929 0.000901 ***
x2          -0.058237   0.020073  -2.901 0.009153 ** 
x4          -1.459335   0.402208  -3.628 0.001789 ** 
x5           0.257526   0.080005   3.219 0.004519 ** 
x8           0.292708   0.223570   1.309 0.206066    
x9          -0.970453   0.653815  -1.484 0.154133    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6827 on 19 degrees of freedom
Multiple R-squared:  0.7499,	Adjusted R-squared:  0.6709 
F-statistic: 9.495 on 6 and 19 DF,  p-value: 6.923e-05


	Shapiro-Wilk normality test

data:  residuals
W = 0.9717, p-value = 0.6679

[1] "R^2 on test set: 0.43"
```



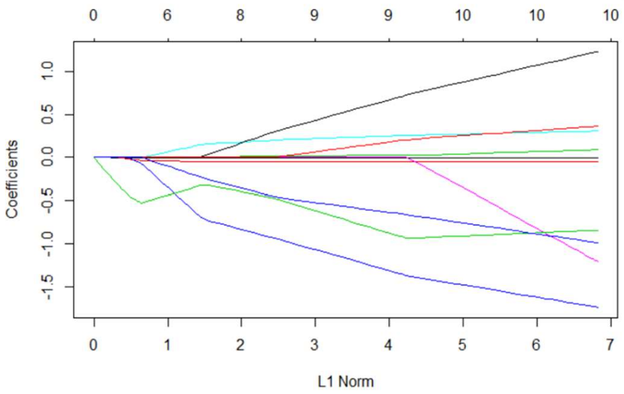


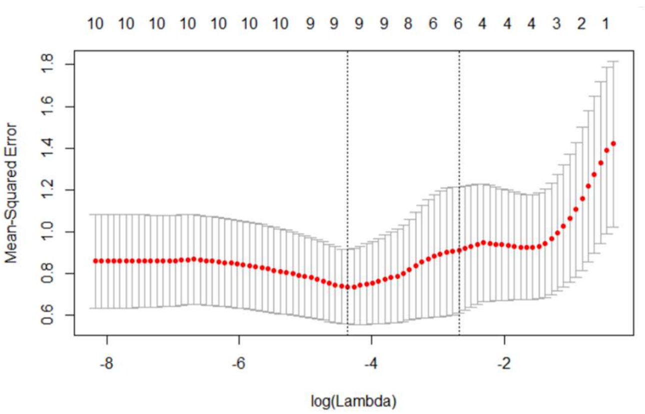
This Model #2 had the **model score on test set is R^2=0.430**, much lower than the training R^2=0.750, which indicated the sign of overfitting but less than Model #1.

1.5. Linear regression with LASSO (Model #3)

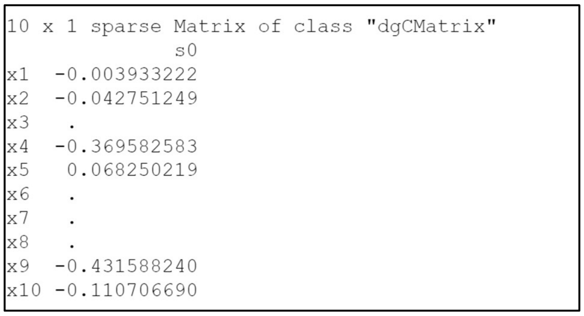
In Model #3, I used LASSO over the linear regression model.

With *glmnet* function on R, I executed LASSO. To find the optimal lambda, which was 0.06817, I used cross-validation with five folds and 1se rule. It gave nonzero coefficients on six variables, x1, x2, x4, x5, x9, and x10.





The followings are the estimated parameters by LASSO. As well known, these are not unbiased estimations. R^2 score on training set was 0.641.



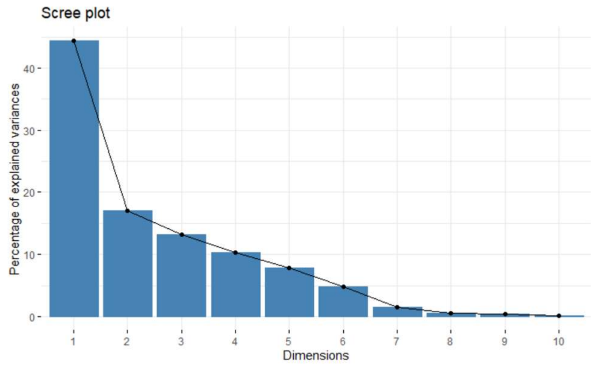
```
10 x 1 sparse Matrix of class "dgCMatrix"
              s0
x1  -0.003933222
x2  -0.042751249
x3   .
x4  -0.369582583
x5   0.068250219
x6   .
x7   .
x8   .
x9  -0.431588240
x10 -0.110706690
```

This Model #3 had the **model score on test set is R^2=0.429**, which was quite close to the score of Model #2. Since the train set score was higher than the test set score by far, it seemed experiencing over-fit of model.

1.6. Linear regression with dimension reduction with PCA (Model #4)

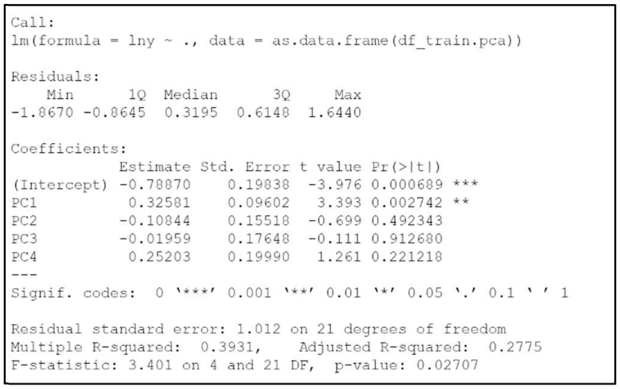
In Model #4, I used PCA for dimension reduction and next the linear regression model by reduced dimension.

With *prcomp* function on R, inputting 10 variables, here is the scree plot of explained variance.

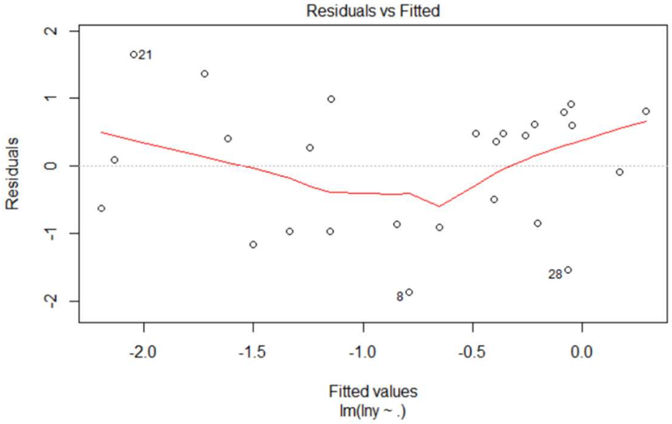


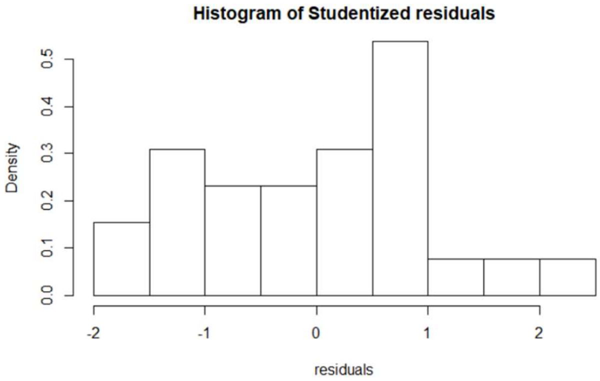
I chose five top principals which 90% of variance is explained by.

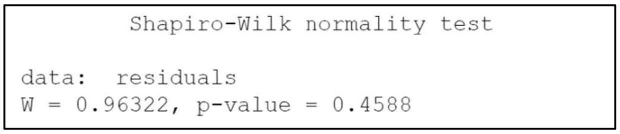
Next, using four principals, I fitted the linear regression.



```
Call:
lm(formula = lny ~ ., data = as.data.frame(df_train.pca))

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8670 -0.8645  0.3195  0.6148  1.6440 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.78870    0.19838  -3.976 0.000689 ***
PC1          0.32581    0.09602   3.393 0.002742 ** 
PC2         -0.10844    0.15518  -0.699 0.492343    
PC3         -0.01959    0.17648  -0.111 0.912680    
PC4          0.25203    0.19990   1.261 0.221218    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.012 on 21 degrees of freedom
Multiple R-squared:  0.3931,	Adjusted R-squared:  0.2775 
F-statistic: 3.401 on 4 and 21 DF,  p-value: 0.02707
```







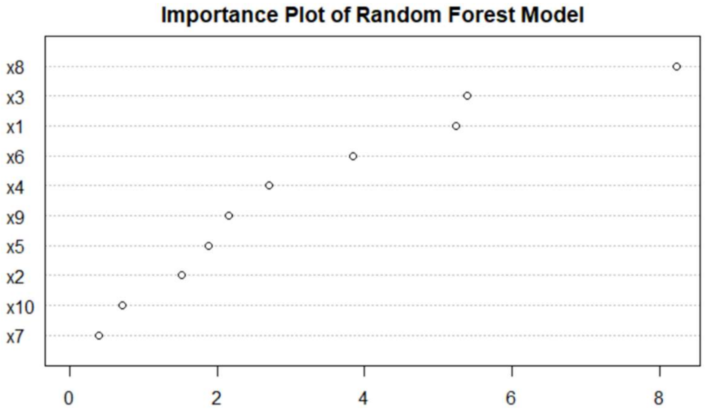
```
        Shapiro-Wilk normality test

data:  residuals
W = 0.96322, p-value = 0.4588
```

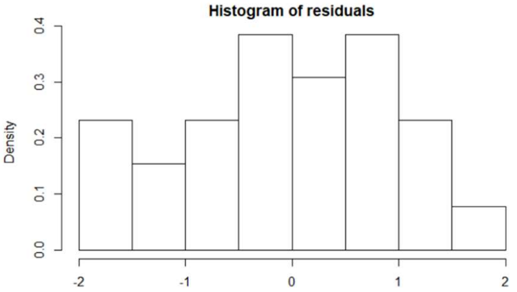
This Model #4 had the **model score on test set is R^2=0.530**, the highest score by now. Since the training R^2 was 0.393, it seemed experiencing under-fit of the model.

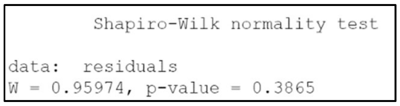
1.7. Random Forest (Model #5)

In Model #5, I used random forest for modeling.

With *randomForest* function in *randomForest* library on R and manual hyper-parameter tuning to find *mtry=3* and *nodesize=5*, the followings were the results.







```
        Shapiro-Wilk normality test

data:  residuals
W = 0.95974, p-value = 0.3865
```

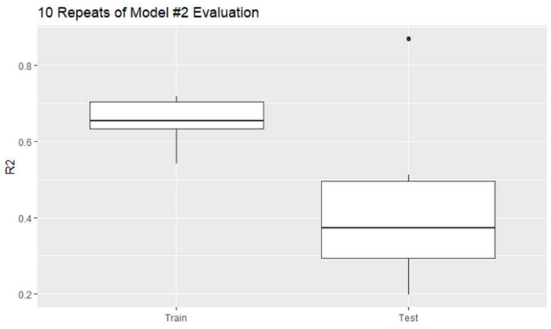
This Model #5 had the model score on train set R^2=0.283 and **test set R^2=0.398.** The score seemed to be not good enough to survive the competitions with other models.

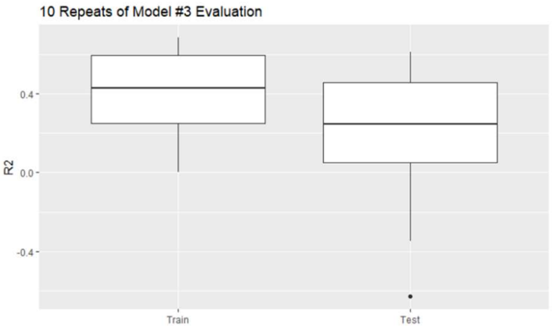
1.8. Conclusion

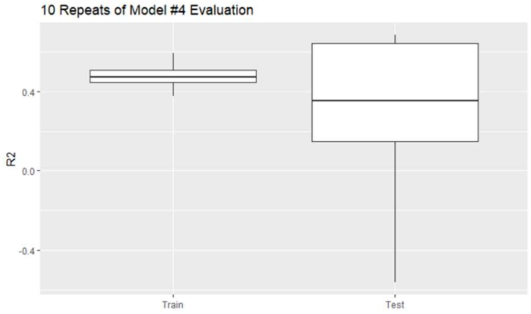
This is the table of score summary:

Model #	Model Description	Train Set R^2 Score	Test Set R^2 Score
1	Linear regression of all variables	0.792	0.263
2	Linear regression with stepwise variable selection	0.750	0.430
3	Linear regression with LASSO	0.641	0.429
4	Linear regression with dimension reduction by PCA	0.393	0.530
5	Random Forest	0.283	0.398

In order to conclude, I reran the Model #2, #3, and #4 with different train-test splits 10 times and compare the R^2 scores between train set vs. test set for each model.

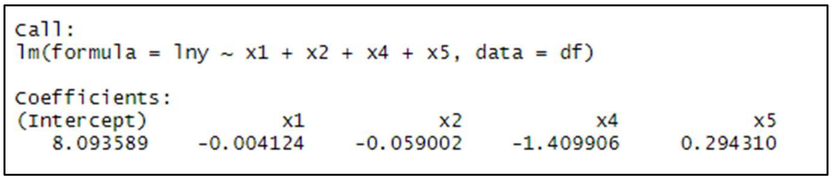






Median on test set was higher on Model #2 and #4 than on Model #3. Some repetition by Model #4 made much lower R^2 score. Therefore, **I concluded Model #2 "linear regression with stepwise variable selection" was the best option** for the higher R^2 score and stability in change per data permutation, though it seemed to cause overfitting of the model because of the lack of data.

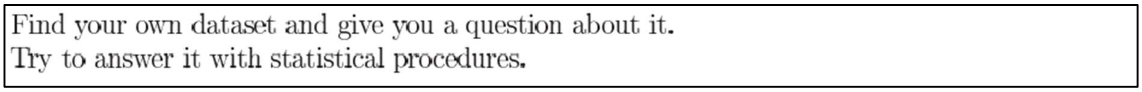
Retraining the model with entire data, the following is the final model coefficients. **This made the R^2 score = 0.650**.



```
Call:
lm(formula = lny ~ x1 + x2 + x4 + x5, data = df)

Coefficients:
(Intercept)           x1           x2           x4           x5
   8.093589    -0.004124    -0.059002    -1.409906     0.294310
```

-- [Exercise 1 ends] --

2. Exercise 2


Find your own dataset and give you a question about it.
Try to answer it with statistical procedures.

2.1. Purpose and Introduction

I chose the goal of my analysis in this exercise being to know “**the influence of weather on the subway use**”. One hypothesis is that when the weather is bad people want to use subway more because they do not want to walk the street in the rain. Another contrary hypothesis is that people do not want to use subway when the weather is bad because they decide to stay at home.

I used R version 3.6.1 for this exercise and used RStudio version 1.2.5019.

2.2. Dataset used in this exercise and first top-down filtering of variables

I will use the data set of “New York City subway station passengers statistics with weather variables”, used in the old Udacity “Intro to Data Science” course (https://www.udacity.com/course/intro-to-data-science--ud359). The same data set is still publicly available in the shared Dropbox repository (https://www.dropbox.com/s/1lpoeh2w6px4diu/improved-dataset.zip?dl=0).

The data set has 42,649 observations with the following 27 variables:

Variable Name	Description
UNIT	Remote unit that collects turnstile information. Can collect from multiple banks of turnstiles. Large subway stations can have more than one unit.
DATEn	Date in "yyyymmdd" (20110521) format.
TIMEn	Time in “hh:mm:ss” (08:05:02) format.
ENTRIESn	Raw reading of cummulative turnstile entries from the remote unit. Occasionally resets to 0.
EXITSn	Raw reading of cummulative turnstile exits from the remote unit. Occasionally resets to 0.
ENTRIESn_hourly	Difference in ENTRIES from the previous REGULAR reading.
EXITSn_hourly	Difference in EXITS from the previous REGULAR reading.
datetime	Date and time in "yyyymmdd hh:mm:ss" format (20110501 00:00:00). Can be parsed into a Pandas datetime object without modifications.
hour	Hour of the timestamp from TIMEn. Truncated rather than rounded.
day_week	Integer (0 - 6 Mon Sun) corresponding to the day of the week.
weekday	Indicator (0 or 1) if the date is a weekday (Mon - Fri).
station	Subway station corresponding to the remote unit.
latitude	Latitude of the subway station corresponding to the remote unit.
longitude	Longitude of the subway station corresponding to the remote unit.
conds	Categorical variable of the weather conditions (Clear, Cloudy etc.) for the time and location.
fog	Indicator (0 or 1) if there was fog at the time and location.
precipi	Precipitation in inches at the time and location.
pressurei	Barometric pressure in inches Hg at the time and location.
rain	Indicator (0 or 1) if rain occurred within the calendar day at the location.

tempi	Temperature in °F at the time and location.
wspdi	Wind speed in mph at the time and location.
meanprecipi	Daily average of precipi for the location.
meanpressurei	Daily average of pressurei for the location.
meantempi	Daily average of tempi for the location.
meanwspdi	Daily average of wspdi for the location.
weather_lat	Latitude of the weather station the weather data is from.
weather_lon	Longitude of the weather station the weather data is from.

To remember, the purpose of my analysis here is to know the influence of weather factors to the number of subway passengers. Therefore, I targeted *ENTRIESn_hourly* which gives the change in cumulative number of turnstile entries that is the number of new entries for the record hour.

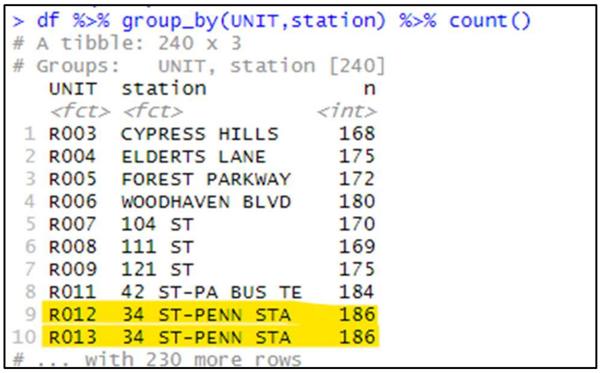
For the purpose, I examined the explanatory variables related to weather such as *fog, precipi, rain, tempi*, and *wspdi*.

Based on my knowledge, I could also assume *UNIT, DATEn, TIMEn, hour, day_week, weekday*, and *station* would affect the *ENTRIESn_hourly*.

2.3. Explanatory data analysis

We will see how much the variables *UNIT, DATEn, TIMEn, hour, day_week, weekday*, and *station* affect the *ENTRIESn_hourly* in the data.

2.3.1. *UNIT* vs. *station*



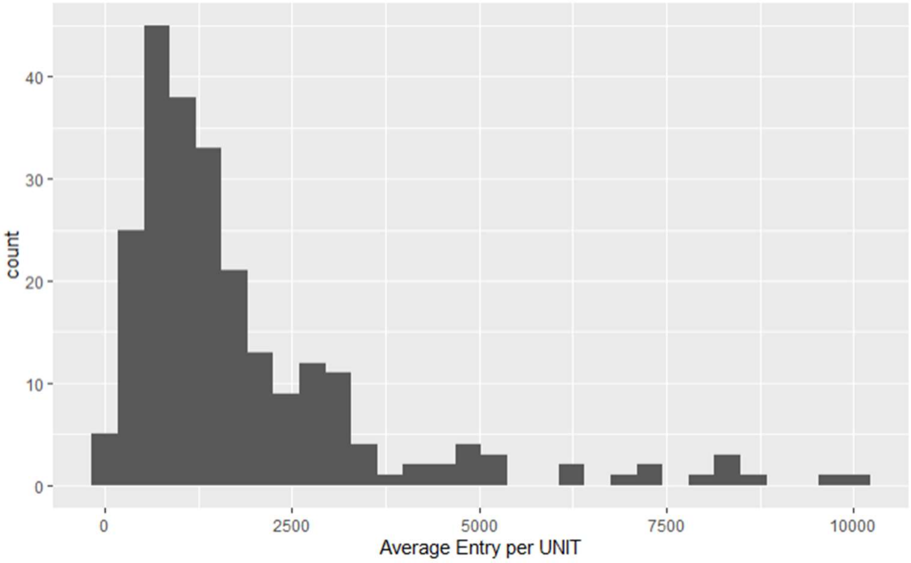
```
> df %>% group_by(UNIT,station) %>% count()
# A tibble: 240 x 3
# Groups:   UNIT, station [240]
   UNIT  station              n
   <fct> <fct>            <int>
 1 R003  CYPRESS HILLS      168
 2 R004  ELDERTS LANE       175
 3 R005  FOREST PARKWAY     172
 4 R006  WOODHAVEN BLVD     180
 5 R007  104 ST             170
 6 R008  111 ST             169
 7 R009  121 ST             175
 8 R011  42 ST-PA BUS TE    184
 9 R012  34 ST-PENN STA     186
10 R013  34 ST-PENN STA     186
# ... with 230 more rows
```

Some larger stations can have multiple UNITs such as “ST 34-PENN STA”, Penn Station on 34th street of New York City, which is a hub station to go outside Manhattan like Long Island or New Jersey state (and the nearest station from my previous workplace :)). Although there aren’t a lot who have multiple UNITs**, I will choose UNIT as the unit of analysis**.

2.3.2. Deep dive in *UNIT*

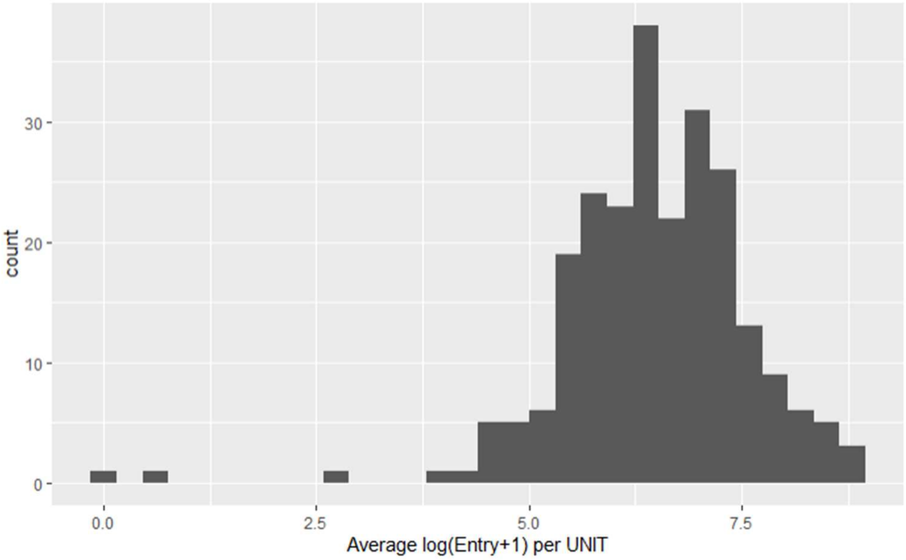
There are 240 unique UNITs.

The distribution of average ENTRIESn_hourly per UNIT is the following:



There is much variety in entries among UNITs, therefore taking the difference of units into account may be crucial to the model performance.

The distribution is skewed to lower values, then I transformed the number of entries by natural logarithm (after adding one to avoid the error when entry=0) before taking the mean. Here's the distribution.



Now the data has a symmetric distribution. Therefore, I will use the log(entry+1) for the target variable instead of the raw entry amount. Also, since they are far lower than others, I will also remove the records with log(Entry+1)<3 as outliers.

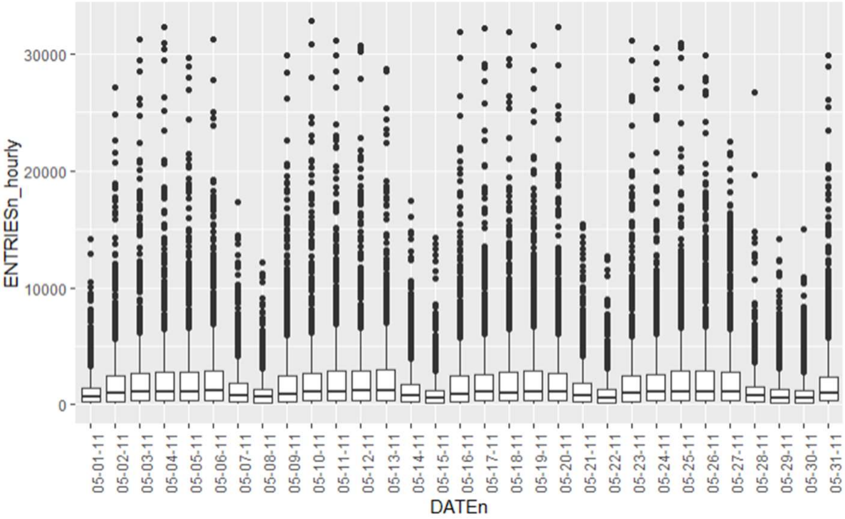
In linear additive model using log transformed entry,

$$\log(Entry_i + 1) = X_i\beta + \varepsilon_i \iff Entry_i = \exp(X_i\beta) \cdot \exp(\varepsilon_i) - 1$$

Therefore, the coefficients imply the marginal multiplicative influence on the raw entry counts.

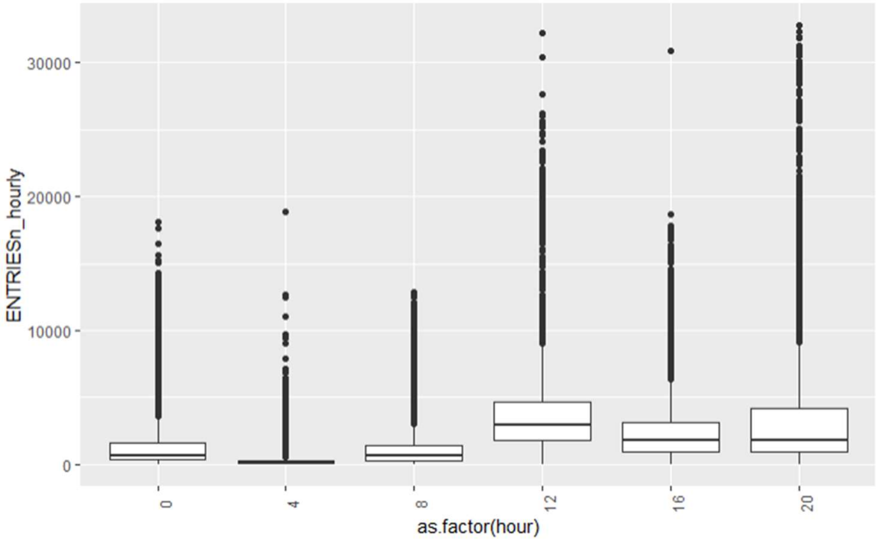
2.3.3. *ENTRIESn_hourly* vs. *DATEn*

Here's the box plot of ENTRIESn_hourly by DATEn. The weekends have fewer entries, but weekdays' entries are rather similar to each other. Since the difference between weekdays and weekends is not relevant to the purpose of analysis, **I will choose only the weekdays between 05-02-2011 and 05-27-2011** for its plenty-ness of the data.



2.3.4. *ENTRIESn_hourly* vs. *hour*

Here's the box plot of ENTRIESn_hourly by hour, only for the data of weekdays.



As a matter of course, the time is highly relevant to the number of entries**. I will choose data only from the time of 16:00PM** because it has enough entries and does not have the influence of commuters. Note removing other time records is still in line with my final purpose because other time records are likely to have passengers who will not change their behaviors by the weather; commuters, early train takers to get other scheduled transportations (e.g. airplane taking off from JFK airport at 7AM etc.).

2.3.5. Summary of data transformation and filtering

One of the weather variables *precipi* basically represents how much rain each date and time had in New York City, but it only held 6 unique variables and not a lot additional information to another explanatory variable *rain*. Therefore, I decided not to use it as explanatory variables.

Here is the list of data transformation and filtering:

- We will log transform ***ENTRIESn_hourly*** after adding 1 and use it as the target variable of the model.
- We will remove the records with **log(*ENTRIESn_hourly*+1)<3**, which is *ENTRIESn_hourly*<exp(3)-1, as outliers.
- We will use ***fog, rain, tempi***, and ***wspdi*** as the explanatory variables to the model.
- ***UNIT*** is a factor which makes high variability in the target variable. It needs to be adjusted in the model.
- To reduce the variability in data which is not necessarily our focus in the analysis, we will filter the data by choosing the rows:

    - **from the weekdays**
    - **at the time of 16:00PM**

Finally, we have a data set to be used in modeling with 5,124 observations and the following 6 variables (1 target variable and 5 explanatory variables.)

- *log(ENTRIESn_hourly+1):* Log transformed number of entries.
- *UNIT:* The identifier of turnstiles. There are 239 unique UNITs. Each UNIT has around 22 records.
- *fog:* 0/1 variable to indicate there was fog at the record date and time.
- *rain:* 0/1 variable to indicate there was rain at the record date and time.
- *tempi:* The continuous variable to represent the temperature in Fahrenheit, ranging from 48 to 86, corresponding from 8.9 to 30 in Celsius.
- *wspdi:* The continuous variable to represent the wind speed in mile-per-hour, ranging from 0 to 23, corresponding from 0 to 37.0 in kilometer-per-hour.

2.3.6. Train-test split

To split the data into training data and testing data, I stratify the split by UNIT which preserves the proportion of UNIT in train data and test data. Otherwise the split was random.

80% of data is for training set and 20% is for testing set.

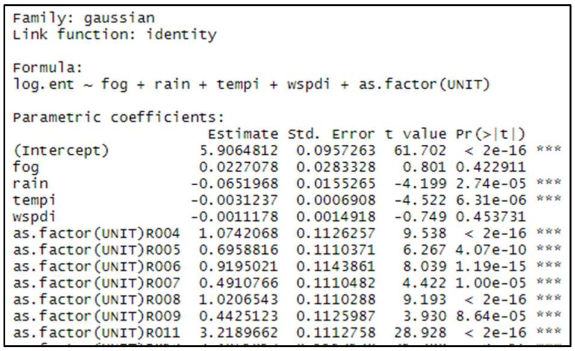
2.4. Modeling

2.4.1. All variables in as linear factors

The first model I checked was the model which includes all variables as terms in linear model except *UNIT* as converted to categorical variable and dummy variables.

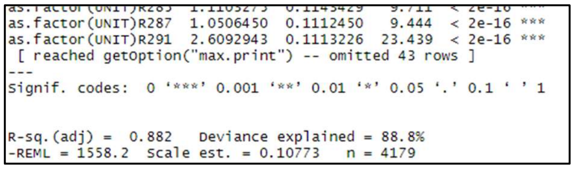
$$ln(ENTRIESn_{hour\quad i} + 1) \sim Normal(\mu_i, \sigma)$$

$$\mu_i = \exp(\,\beta_0 + \beta_1 fog_i + \beta_2 rain_i + \beta_3 tempi_i + \beta_4 wspdi_i + \beta_5 factor(UNIT_i))$$

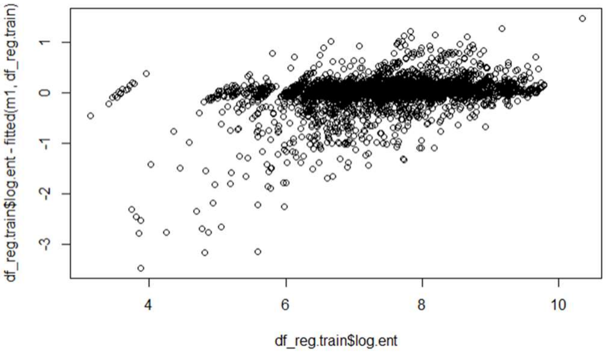


```
Family: gaussian
Link function: identity

Formula:
log.ent ~ fog + rain + tempi + wspdi + as.factor(UNIT)

Parametric coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         5.9064812  0.0957263  61.702  < 2e-16 ***
fog                 0.0227078  0.0283328   0.801 0.422911
rain               -0.0651968  0.0155265  -4.199 2.74e-05 ***
tempi              -0.0031237  0.0006908  -4.522 6.31e-06 ***
wspdi              -0.0011178  0.0014918  -0.749 0.453731
as.factor(UNIT)R004  1.0742068  0.1126257   9.538  < 2e-16 ***
as.factor(UNIT)R005  0.6958816  0.1110371   6.267 4.07e-10 ***
as.factor(UNIT)R006  0.9195021  0.1143861   8.039 1.19e-15 ***
as.factor(UNIT)R007  0.4910766  0.1110482   4.422 1.00e-05 ***
as.factor(UNIT)R008  1.0206543  0.1110288   9.193  < 2e-16 ***
as.factor(UNIT)R009  0.4425123  0.1125987   3.930 8.64e-05 ***
as.factor(UNIT)R011  3.2189662  0.1112758  28.928  < 2e-16 ***
```

.... (results omitted to save green!)



```
as.factor(UNIT)R283  1.1103273  0.1143429   9.711  < 2e-16 ***
as.factor(UNIT)R287  1.0506450  0.1112450   9.444  < 2e-16 ***
as.factor(UNIT)R291  2.6092943  0.1113226  23.439  < 2e-16 ***
 [ reached getOption("max.print") -- omitted 43 rows ]
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


R-sq.(adj) =  0.882   Deviance explained = 88.8%
-REML = 1558.2  Scale est. = 0.10773   n = 4179
```



To interpret the results:

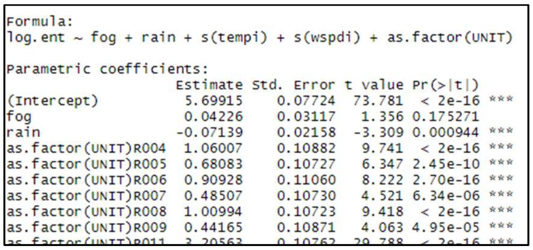
- The *fog* and *wspdi* only have small and insignificant coefficients.
- The *rain* and *tempi* have significant and negative coefficients, which represents their negative influence on the number of subway passengers.
- Completely every UNIT has significant coefficient, which represents the difference in turnstile (almost equivalent to the difference in subway station) is highly relevant to the number of subway passengers.
- Train data R^2 score (=*deviance explained* in GLM with Gaussian distribution and identity link function) = 88.8%
- Test data R^2 score=88.0%

2.4.2. *tempi* and *wspdi* as Generalized Additive Model (GAM) smoother terms

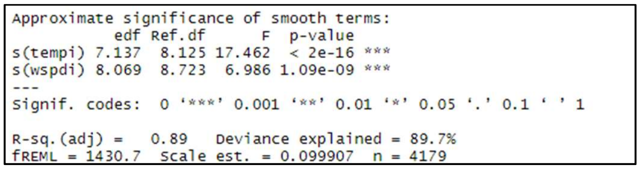
The next model includes all variables as terms in linear model except *UNIT* as converted to categorical variable and dummy variables (just as we did in the model in 2.4.1.) and set the two explanatory variables *tempi* and *wspdi* with GAM smoother terms, to see if they have non-linear associations to the dependent variable. With this setup, the coefficients and the smoother terms on *tempi* and *wspdi* are identified at the same time.

$$ln(ENTRIESn_{hourly_i} + 1) \sim Normal(\mu_i, \sigma)$$

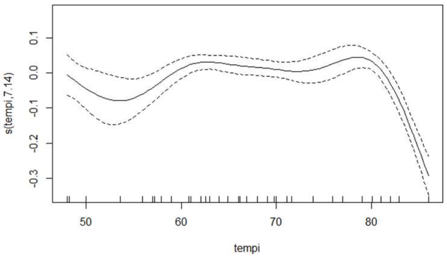
$$\mu_i = \exp(\,\beta_0 + \beta_1 fog_i + \beta_2 rain_i + \beta_3 g_1(tempi_i) + \beta_4 g_2(wspdi_i) + \beta_5 factor(UNIT_i))$$

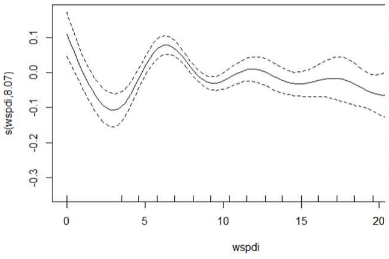


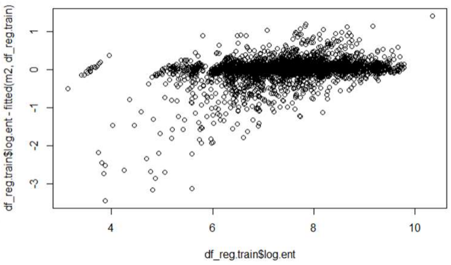
```
Formula:
log.ent ~ fog + rain + s(tempi) + s(wspdi) + as.factor(UNIT)

Parametric coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)            5.69915    0.07724  73.781  < 2e-16 ***
fog                    0.04226    0.03117   1.356 0.175271    
rain                  -0.07139    0.02158  -3.309 0.000944 ***
as.factor(UNIT)R004    1.06007    0.10882   9.741  < 2e-16 ***
as.factor(UNIT)R005    0.68083    0.10727   6.347 2.45e-10 ***
as.factor(UNIT)R006    0.90928    0.11060   8.222 2.70e-16 ***
as.factor(UNIT)R007    0.48507    0.10730   4.521 6.34e-06 ***
as.factor(UNIT)R008    1.00994    0.10723   9.418  < 2e-16 ***
as.factor(UNIT)R009    0.44165    0.10871   4.063 4.95e-05 ***
as.factor(UNIT)R011    3.20563    0.10762  29.788  < 2e-16 ***
```



```
Approximate significance of smooth terms:
            edf Ref.df      F  p-value    
s(tempi) 7.137  8.125 17.462  < 2e-16 ***
s(wspdi) 8.069  8.723  6.986 1.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =   0.89   Deviance explained = 89.7%
fREML = 1430.7  Scale est. = 0.099907  n = 4179
```







To interpret the results:

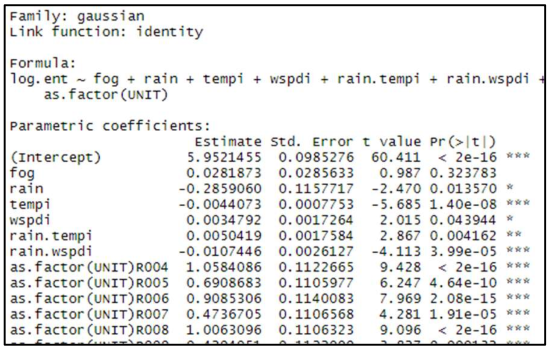
- The *fog* only has a small and insignificant coefficient.
- The *rain* has a significant and negative coefficient, which represents its negative influence on the number of subway passengers.
- Completely every UNIT has significant coefficient, which represents the difference in turnstile (almost equivalent to the difference in subway station) is highly relevant to the number of subway passengers.
- The smoother term estimations are rather stable (narrow Confidence Intervals) and they show rather neutral influences on dependent variables from both of two variables, except the case when the temperature is higher than 80 Fahrenheit degrees = 26.7 Celsius degrees which has negative impact.
- Train data R^2 score = 89.7%
- Test data R^2 score=88.7%

2.4.3. Interaction term of *rain*tempi* and *rain*wspdi*

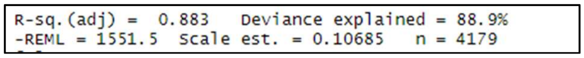
The next model tries to take the interaction between rain and temperature, and rain and wind speed, as *rain*tempi* and *rain*wspdi*, respectively. This is meant to segregate the influence of rain and temperature or wind speed, because the rainy days tend to have low temperature and stronger wind and we can assume they are not independent event.

$$ln(ENTRIESn_{hourl\ \ i} + 1) \sim Normal(\mu_i, \sigma)$$

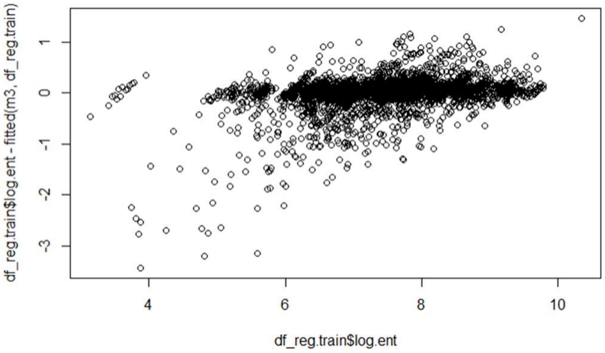
$$\mu_i = \exp(\ \beta_0 + \beta_1 fog_i + \beta_2 rain_i + \beta_3 tempi_i + \beta_4 wspdi_i + \beta_5 factor(UNIT_i) + \beta_6 rain_i * tempi_i + \beta_7 rain_i * wspid_i)$$



```
Family: gaussian
Link function: identity

Formula:
log.ent ~ fog + rain + tempi + wspdi + rain.tempi + rain.wspdi +
    as.factor(UNIT)

Parametric coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          5.9521455  0.0985276  60.411  < 2e-16 ***
fog                  0.0281873  0.0285633   0.987 0.323783    
rain                -0.2859060  0.1157717  -2.470 0.013570 *  
tempi               -0.0044073  0.0007753  -5.685 1.40e-08 ***
wspdi                0.0034792  0.0017264   2.015 0.043944 *  
rain.tempi           0.0050419  0.0017584   2.867 0.004162 ** 
rain.wspdi          -0.0107446  0.0026127  -4.113 3.99e-05 ***
as.factor(UNIT)R004  1.0584086  0.1122665   9.428  < 2e-16 ***
as.factor(UNIT)R005  0.6908683  0.1105977   6.247 4.64e-10 ***
as.factor(UNIT)R006  0.9085306  0.1140083   7.969 2.08e-15 ***
as.factor(UNIT)R007  0.4736705  0.1106568   4.281 1.91e-05 ***
as.factor(UNIT)R008  1.0063096  0.1106323   9.096  < 2e-16 ***
```



```
R-sq.(adj) =  0.883   Deviance explained = 88.9%
-REML = 1551.5  Scale est. = 0.10685   n = 4179
```



To interpret the results:

- The *fog* only has a small and insignificant coefficient.
- The *rain* has a significant and negative coefficient, which represents its negative influence on the number of subway passengers.
- The *tempi, wspdi*, and *rain*tempi* have significant coefficients, but their absolute values are smaller than other coefficients.
- The rain*wspdi has a significant and negative coefficient, which indicates the rainy and windy day, there are less numbers of subway use.
- Completely every UNIT has significant coefficient, which represents the difference in turnstile (almost equivalent to the difference in subway station) is highly relevant to the number of subway passengers.
- Train data R^2 score = 88.9%
- Test data R^2 score=88.1%

2.4.4. Random effect on *UNIT*

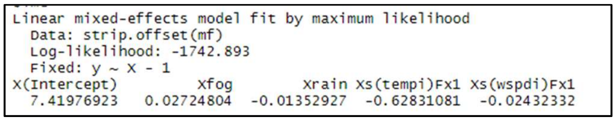
The final model studied is taking the *UNIT* into a random effect term, instead of fixed effect terms as we have done till the previous model. I keep the *tempi* and *wspdi* in the smoother terms as we saw in the model of 2.4.2.

$$ln(ENTRIESn_{hourly_i} + 1) \sim Normal(\mu_i, \sigma)$$

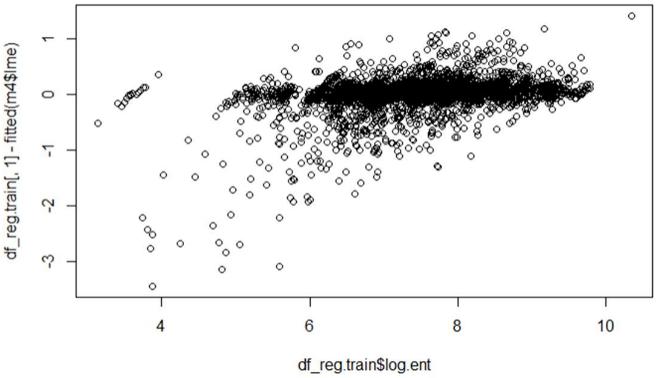
$$\mu_i = \exp(\ \beta_0 + \beta_1 fog_i + \beta_2 rain_i + \beta_3 g_1(tempi_i) + \beta_4 g_2(wspdi_i) + \beta_{5,u}$$

$$\beta_{5,u} \sim Normal(0, G)$$

Where, $G \in \mathbb{R}^{U \times U}$, a covariance matrix of $\beta_{5,u}$ with U as a number of unique *UNIT*s, 239 in our data.



```
Linear mixed-effects model fit by maximum likelihood
  Data: strip.offset(mf)
  Log-likelihood: -1742.893
  Fixed: y ~ X - 1
X(Intercept)         Xfog        Xrain Xs(tempi)Fx1 Xs(wspdi)Fx1
  7.41976923   0.02724804  -0.01352927  -0.62831081  -0.02432332
```



Train set R^2 score = 89.5%.

We did not see any improvement from the fixed effect model for its additional complexity to the model, which was not surprising since we already saw significant estimates of coefficients on every *UNIT* with fixed effect model.

Therefore, **we will not examine this model further**.

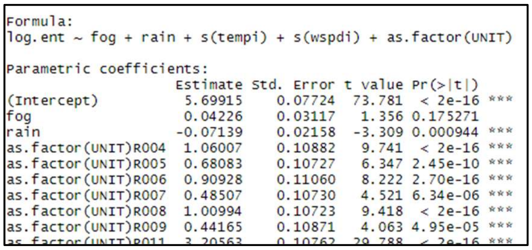
2.5. Final model and conclusion

The exhibit below represents the AICs of each model in 2.4.1 to 2.4.3:

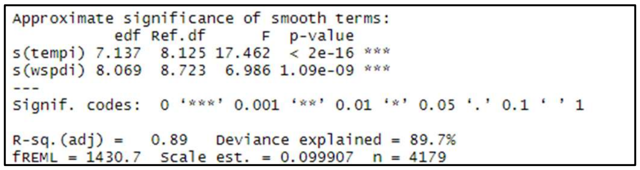
Model	AIC
Model in 2.4.1.	2,785.798
Model in 2.4.2.	2,484.379
Model in 2.4.3.	2,753.327

The smallest AIC is by Model in 2.4.2. and now we say it is the best model out of our exploration.

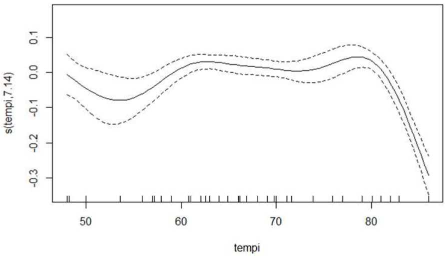
Revisiting the model results by Model in 2.4.2.,

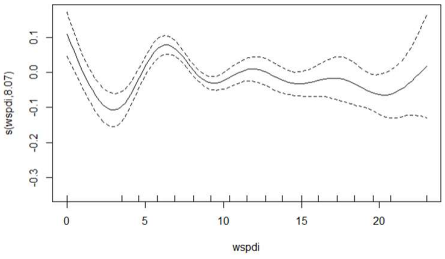


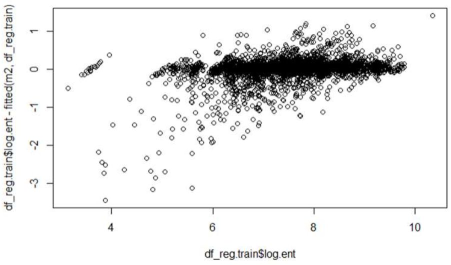
```
Formula:
log.ent ~ fog + rain + s(tempi) + s(wspdi) + as.factor(UNIT)

Parametric coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          5.69915    0.07724  73.781  < 2e-16 ***
fog                  0.04226    0.03117   1.356 0.175271    
rain                -0.07139    0.02158  -3.309 0.000944 ***
as.factor(UNIT)R004  1.06007    0.10882   9.741  < 2e-16 ***
as.factor(UNIT)R005  0.68083    0.10727   6.347 2.45e-10 ***
as.factor(UNIT)R006  0.90928    0.11060   8.222 2.70e-16 ***
as.factor(UNIT)R007  0.48507    0.10730   4.521 6.34e-06 ***
as.factor(UNIT)R008  1.00994    0.10723   9.418  < 2e-16 ***
as.factor(UNIT)R009  0.44165    0.10871   4.063 4.95e-05 ***
as.factor(UNIT)R011  3.20563    0.10762  29.788  < 2e-16 ***
```



```
Approximate significance of smooth terms:
            edf Ref.df      F  p-value    
s(tempi) 7.137  8.125 17.462  < 2e-16 ***
s(wspdi) 8.069  8.723  6.986 1.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =   0.89   Deviance explained = 89.7%
fREML = 1430.7  Scale est. = 0.099907  n = 4179
```







We can conclude:

- **We cannot say fog has any influence** on the number of subway entries.
- **Rain has negative impact** to the number of entries to the extent of minus 7% (= 1 – exp(-0.07139)).
- **The influence of temperature and wind are mostly neutral** on the number of subway entries, **except the case when the temperature is higher than 80 Fahrenheit degrees** = 26.7 Celsius degrees, where we see the drop in subway entries to the extent of minus 25% (= 1 – exp(-0.3)) within the next 5 Fahrenheit degrees = 3 Celsius degrees range.
- **The number of subway use is highly distinctive for the subway stations**.

2.6. Replication study by another GLM model with non-Gaussian distribution

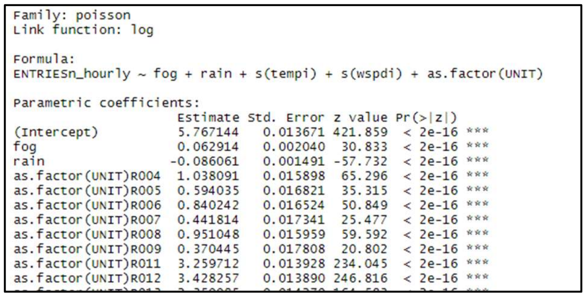
Here I am going to examine the validity of the results using another approach; instead of linear model after log transformation, I will use the GAM with Poisson distribution and log link function. Here is the model formula, assigning the raw *ENTRIESn_hourly*, not the log-transformed one of it:

$$ENTRIESn\_hourly_i \sim Poisson(\lambda_i)$$

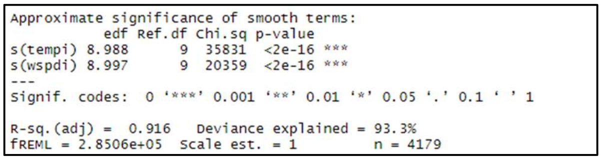
$$\lambda_i = \exp(\,\beta_0 + \beta_1 fog_i + \beta_2 rain_i + \beta_3 g_1(tempi_i) + \beta_4 g_2(wspdi_i) + \beta_5 factor(UNIT_i))$$

Since the link function is log, the marginal influence by coefficients are multiplicative with exponential transformation, just as the linear models we have discussed above.

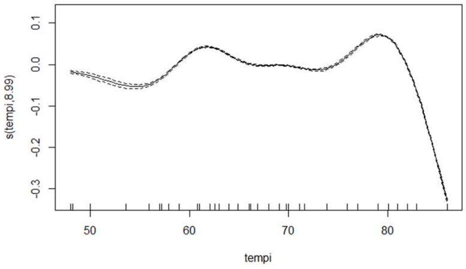
The model results are the followings:

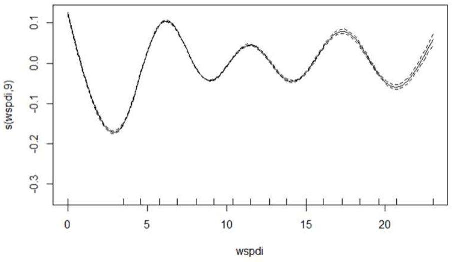


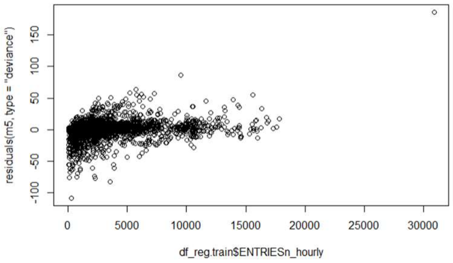
```
Family: poisson
Link function: log

Formula:
ENTRIESn_hourly ~ fog + rain + s(tempi) + s(wspdi) + as.factor(UNIT)

Parametric coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)          5.767144   0.013671 421.859  < 2e-16 ***
fog                  0.062914   0.002040  30.833  < 2e-16 ***
rain                -0.086061   0.001491 -57.732  < 2e-16 ***
as.factor(UNIT)R004  1.038091   0.015898  65.296  < 2e-16 ***
as.factor(UNIT)R005  0.594035   0.016821  35.315  < 2e-16 ***
as.factor(UNIT)R006  0.840242   0.016524  50.849  < 2e-16 ***
as.factor(UNIT)R007  0.441814   0.017341  25.477  < 2e-16 ***
as.factor(UNIT)R008  0.951048   0.015959  59.592  < 2e-16 ***
as.factor(UNIT)R009  0.370445   0.017808  20.802  < 2e-16 ***
as.factor(UNIT)R011  3.259712   0.013928 234.045  < 2e-16 ***
as.factor(UNIT)R012  3.428257   0.013890 246.816  < 2e-16 ***
```



```
Approximate significance of smooth terms:
            edf Ref.df Chi.sq p-value    
s(tempi) 8.988      9  35831  <2e-16 ***
s(wspdi) 8.997      9  20359  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.916   Deviance explained = 93.3%
fREML = 2.8506e+05  Scale est. = 1          n = 4179
```







The overall tendency was consistent with the best linear model we chose above, including the amounts of coefficients. The variances of coefficients are narrower than the ones in linear model. The deviance residuals are slightly increasing over the *ENTRIESn_hourly* and have one apparent outlier, but it is fair to say the fit is good and even better than the best linear model.

Then, now **we were able to double check the conclusion was consistent enough with the data trend**.

--- *[Exercise 2 ends]* ---

3. Exercise 3

Make the summary of the article MTGAUE.

3.1. Overview

Assuming we recorded the spikes on two neurons over a certain period, we would like to know if (or more specifically, would like to statistically-test if) the spikes in those two neurons happen independently or not.

Unitary Event (UE) method proposed by Grün in 1996 was based on binning procedures upon the observation period and comparing the simultaneous bins of two neurons. The article MTGAUE claims the UE method (Grün, 1996) has many defects in the methodology.

Multiple Shift (MS) method (Grün et al, 1999) provided some corrections to UE method and gave a test for the delayed coincidences of spikes from two neurons, which allows to detect the association of two neurons even when their firing has time gap. The article MTGAUE claims that the test given by MS method has methodological flaw.

It also states that the Gaussian approximated distribution with parameter estimates the article provides are more statistically precise and reliable than previous methodologies.

The article also tries to adjust the problem of false discovery caused by repeated tests to test local dependence in case of non-stationary neuron train, which is conventionally done in UE method with no adjustment, by controlling False Discovery Rate (FDR) through Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995).

The article also demonstrates the benefit of the proposed approach over previous methods upon the artificially produced data and actual data.

3.2. Summary of assumptions and null/alternative hypothesis discussed

I listed the assumptions and null/alternative hypothesis to be discussed in the following individual sections for the ease of reference.

- *Assumption 1*: The processes N_1 and N_2 are both Poisson processes.
- *Assumption 2*: The processes N_1 and N_2 are stationary on the time period $[a, b]$.
- *Assumption 3*: The delay δ is less than half the window size, i.e. $\delta \leq (b - a)/2$.
- *Null Hypothesis* (H_0): N_1 is independent of N_2 on $[a, b]$.
- *Alternative Hypothesis* (H_1): N_1 is dependent of N_2 on $[a, b]$.

3.3. Unitary Event (UE) method (Grün, 1996)

In UE method proposed by Grün in 1996, they claimed that under the *Null Hypothesis* (H_0) and *Assumptions 1 and 2* the following proposition holds:

$$m_0 \coloneqq \mathbb{E}(Y) = k(1 - \exp(-\lambda_1 \delta))(1 - \exp(-\lambda_2 \delta))$$

Where,

$$Y = \sum_{i=1}^{k} \mathbf{1}_{\{N_1(I_i) \geq 1, N_2(I_i) \geq 1\}}$$

$N_j(I_i)$ denotes the number of spikes of neuron j on binned interval I_i within $[a, b]$, and δ denotes the length of single bin which is $\delta = (b - a)/k$. To paraphrase in natural language, Y denotes the number of bins within $[a, b]$ which have spikes from both of neuron 1 and neuron 2.

When λ_i and δ are small enough, $m_0 \simeq m_{UE} = k\lambda_1\lambda_2\delta^2$.

The article MTGAUE points out that the primal drawback of this approach is λ_i in this formula is the unknown parameters and we need to estimate them through estimators $\widehat{\lambda}_i$. Using the estimates $\widehat{m}_{UE} = k\,\widehat{\lambda}_1\,\widehat{\lambda}_2\delta^2$ in place of the true parameters in m_{UE} may sometimes make the law illegitimate (known as "*plug-in problem*".)

Also, the article states that the binning approach inherently suffers a loss of information by rolling up the information within the same bins.

3.4. Multiple Shifts (MS) procedure (Grün et al, 1999)

In MS procedure, the coincidences count does not require the spikes to be simultaneous and represents the coincidences count X with delay δ as:

$$X = \int_{[a,b]^2} \mathbf{1}_{|x-y| \leq \delta} dN_1(x) dN_2(y)$$

Where, dN_1 and dN_2 are the point measures associated with each spike train.

Under the *Null Hypothesis* (H_0) and *Assumptions 1, 2 and 3,* the expectation and variance of X are:

$$m_0 \coloneqq \mathbb{E}(X) = \lambda_1\lambda_2[2\delta(b - a) - \delta^2]$$

And,

$$\mathrm{Var}(X) = \lambda_1\lambda_2\left[2\delta(b-a)-\delta^2\right] + \left[\lambda_1^2\lambda_2 + \lambda_1\lambda_2^2\right]\left[4\delta^2(b-a) - \frac{10}{3}\delta^3\right]$$

In original article by Grün et al in 1999, m_0 is approximated by Poisson distribution with $m_0 \simeq m_g = 2\lambda_1\lambda_2\delta(b-a)$, but the article MTGAUE claims that since $\mathbb{E}(X) \neq Var(X)$, this approximation is not correct, particularly when δ is large.

3.5. Gaussian approximation proposed by MTGAUE

Here, the observed average coincidences count is:

$$\bar{m} = \frac{1}{M}\sum_{i=1}^{M} X^{(i)}$$

Where, M represents the number of experiments with observations of two neurons $\left(N_1^{(1)}, N_2^{(1)}\right), \dots, \left(N_1^{(M)}, N_2^{(M)}\right)$, and $X^{(i)}$ is the coincidences count with delay δ for the ith trial.

The estimate of m_0 is $\hat{m}_0 = \hat{\lambda}_1\hat{\lambda}_2[2\delta(b-a)-\delta^2]$. When $\hat{m}_0$ and $\bar{m}$ are far enough, the *Null Hypothesis* (H_0) is rejected and the dependence between N_1 and N_2 holds.

Thanks to Central Limit Theorem,

$$\sqrt{M}\frac{\bar{m}-\hat{m}_0}{\sqrt{\hat{\sigma}^2}} \xrightarrow{\mathcal{L}} \mathrm{N}(0,1)$$

Where, $\hat{\sigma}^2$ is the estimate of $Var(X)$ with replacement of λ_i with $\hat{\lambda}_i$ in the definition noted above, and $\xrightarrow{\mathcal{L}}$ represents the convergence in law when M goes infinite.

The article MTGAUE claims that this Gaussian approximation is more statistically rigorous than UE with MS correction using Poisson approximation with $m_g = 2\lambda_1\lambda_2\delta(b-a)$ and demonstrates it by detection trials over simulated data.

3.6. Local dependence detection in case of non-stational train and False Discovery Rate (FDR) control by Benjamini and Hocheberg approach

When the non-stationary in neuron trains is assumed, it is better to test local time window independence and repeat it over many local windows instead of taking the mean over the entire time range as discussed in the previous section.

If we repeat the statistical test multiple times, we must be aware of the risk of falling in the wrong 'discovery' with finding accidentally at least one rejection of H_0 even when H_0 holds in reality.

Bonferroni's method (Holm, 1979) is a common approach to adjust the multiple tests problem controlling Family-wise Error Rate (FWER) with simple procedure, but is also known that it brings higher possibility capturing false negative.

The article MTGAUE adopts the Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995) which controls the False Discovery Rate (FDR) to deal with the multiple tests problem.

3.7. Demonstration upon simulated data and actual data

Finally, the article demonstrates the superiority of the proposed approach comparing with UE method upon simulated data and actual data with various dependence and δ.

-- *[Exercise 3 ends]* --