STAC67 Case Study Project: Analyzing The Factors That Affect Risk of Cervical Cancer Relapse

Project Group #13

Team Members:

Joshua Garvida (1003042520)

Background and Significance, Discussion/Conclusion, References

Jimmy Deng (1006280472)

Exploratory Data Analysis, Graphs/Visuals

Daniyal Iqbal (1006145805)

Data Handling, Model Selection and Checking

Nghia Le (1007523958)

Model Selection, Model Diagnostics, ML Models

Introduction

The initial concept of a bicycle sharing system started in 1965 in Amsterdam (Shaheen, Guzman, & Zhang, 2010), with the intent to better environmental and social welfare. Through systems in which people rent a bicycle at minimal costs at one docking station and return it in another, bicycle sharing as a whole has significant impacts in "increasing the use of transportation, minimizing greenhouse gas emissions, enhancing public health and also traffic troubles." (Park & Cho, 2020) The introduction of rental bikes has shown tremendous growth as there are more than 50 countries implementing this kind of system, informed by the renting of bicycles being cost effective and at certain instances being free of charge. However, a problem arises as the constant rise of users means a higher demand for accessible bicycles. Therefore, there is a need for the supply of rental bikes to match this demand to ensure accessibility and decrease wait times. This is the case for "Ddareungi," or "Seoul Bike," the bicycle sharing system in South Korea started in 2015. Through the "Seoul Bike" data and understanding the significance of various weather information, our research aims to estimate the optimal amount of rental bikes needed at each hour of the day.

EXPLORATORY DATA ANALYSIS

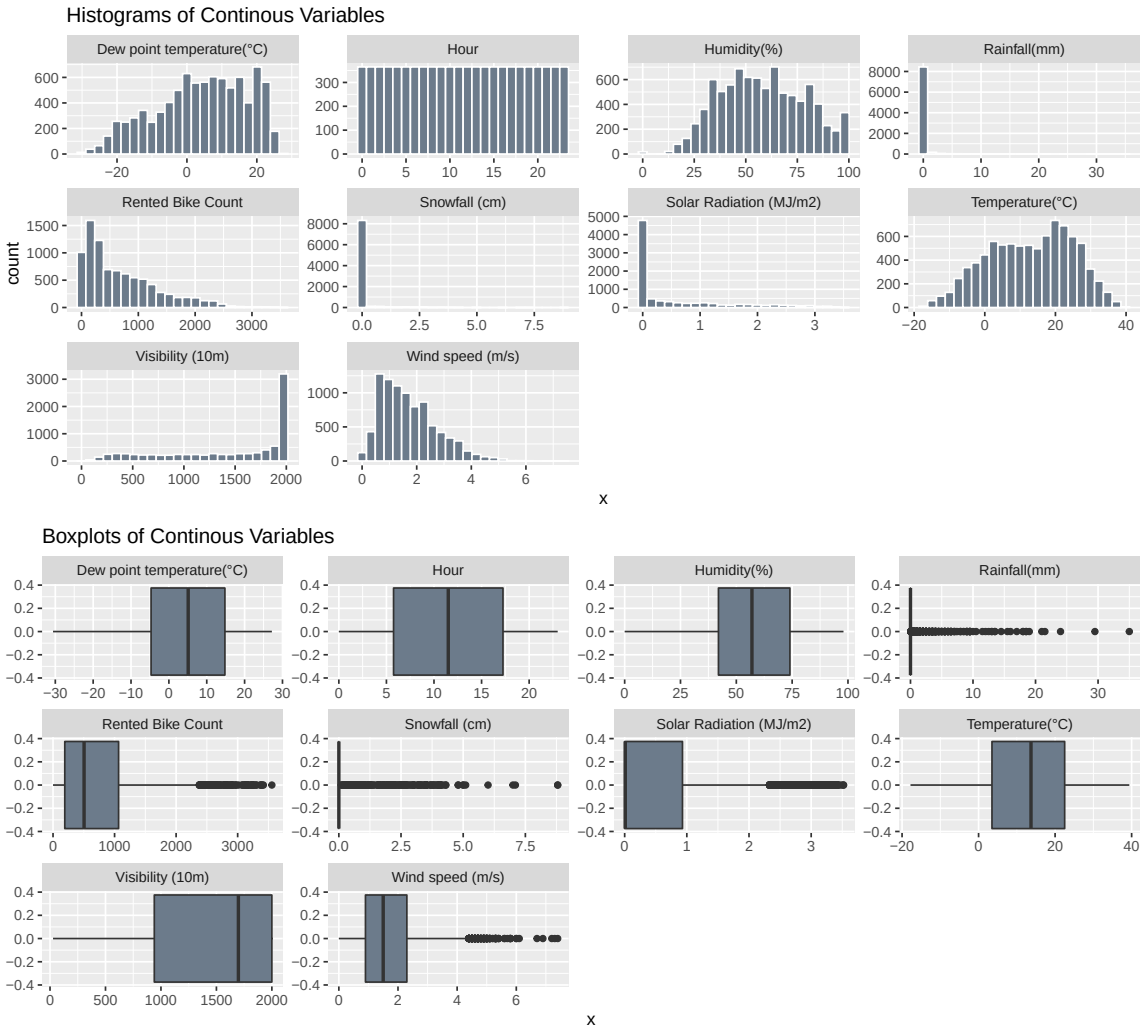
Description of dataset The data set used in this case study comes from data.Seoul.kr as part of data collection done by the Seoul Metropolitan Government between January 2017 and November 2018 on several variables of data points for predicting supply of rental bikes per hour. The data set is comes in an excel file with 14 variables and 8760 observations.

```
bike <- read_excel("./SeoulBikeData.xlsx")    # Reading in the data set
```

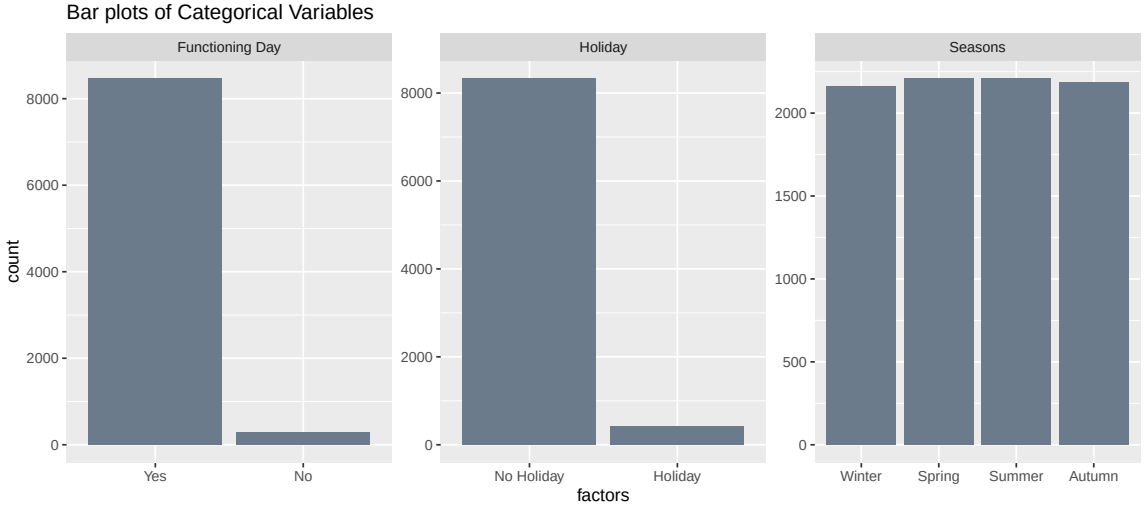
The 14 variables in the data set include dates, quantitative variables, and qualitative variables. The quantitative variables include, rented bike count, hour, temperature, humidity, wind speed, visibility, dew point temperature, solar radiation, rainfall, and snowfall. The qualitative variables include, seasons, holiday, and functioning day, where the latter two are binary variables dictating whether that day is a holiday/functioning day or not and the former indicates the season it is. The identifier for each observation consists of the date and hour. The response variable in this data set is Rented Bike Count and the rest are explanatory variables.

Extracting and cleaning data For this data set, we chose discarded date as the sole identifier and kept hour for our analysis because the hour of day can have a more observable effect on the response compared to specific days in the year (as seen later in graph analysis). Furthermore, we decided to keep hour as a quantitative variable instead of qualitative due to time having a set order. Further analysis may include visualizing and exploring this data set as a times series, but was forgone in this case study for simplicity.

Plotting individual variables With the data imported, we can begin with taking an overview of the data by plotting the individual variables to illustrate the general distributions of our data.



Here, it is seen that most of the quantitative variables are skewed in some fashion. Notably, rainfall, snowfall, solar radiation, wind speed have extreme right skew distributions, and visibility has an extreme left skew. Hour has a uniform distribution, and the response variable has a right skew distribution. Humidity, dew point temperature, and temperature have minor skews in their distribution.

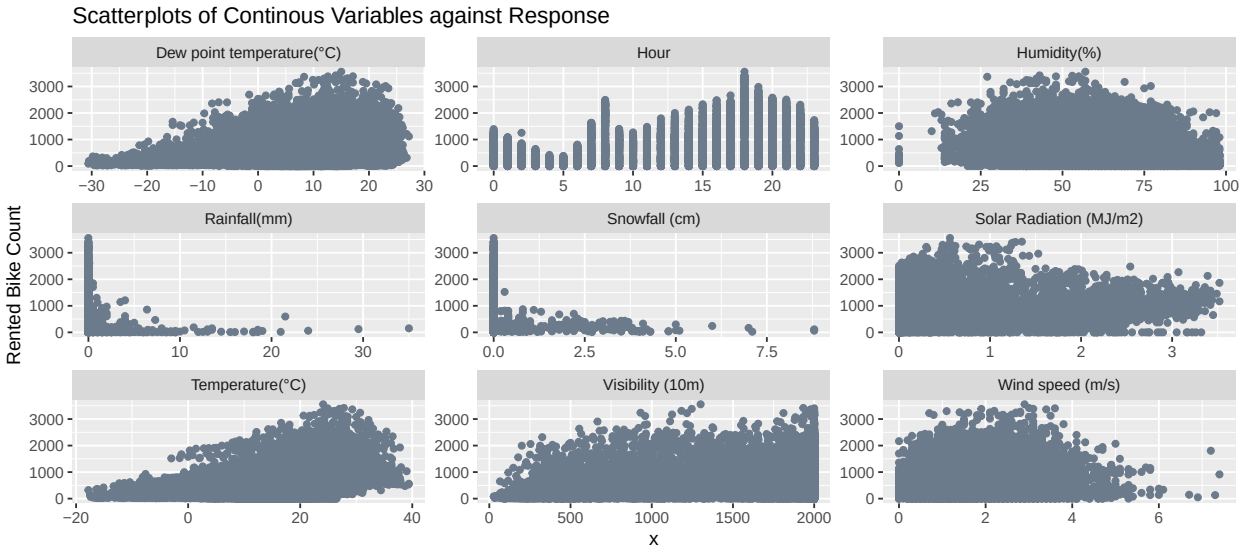


For the categorical variables, season appears to be uniformly distributed and the other two have extreme right skews.

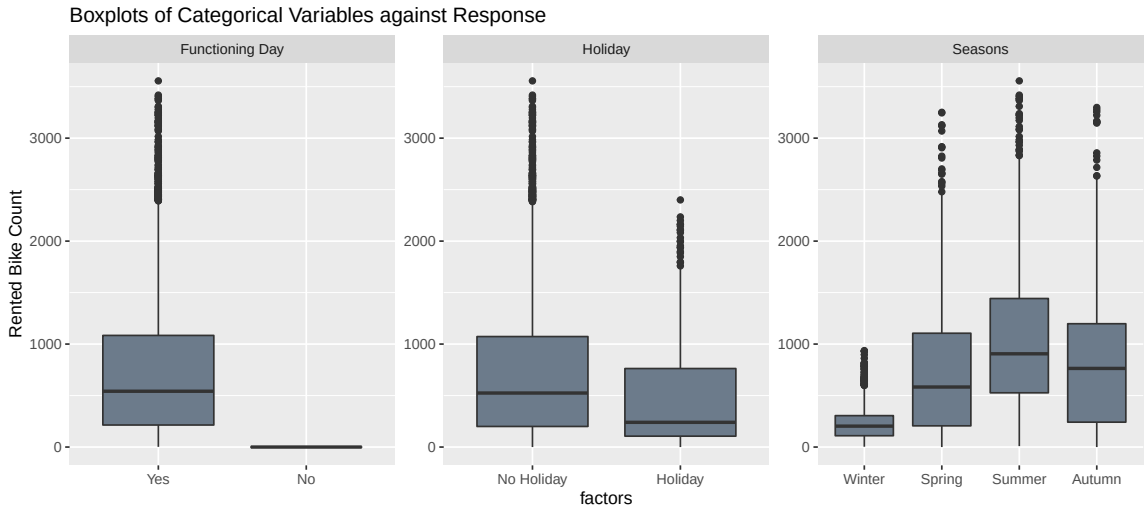
variable	mean	median	mode	sd	var	Q1	Q3	max	min
Dew point temperature(°C)	4.07	5.1	0	13.06	170.6	-4.7	14.8	27.2	-30.6
Hour	11.5	11.5	0	6.92	47.92	5.75	17.25	23	0
Humidity(%)	58.23	57	53	20.36	414.6	42	74	98	0
Rainfall(mm)	0.15	0	0	1.13	1.27	0	0	35	0
Rented Bike Count	704.6	504.5	0	645	416022	191	1065	3556	0
Snowfall (cm)	0.08	0	0	0.44	0.19	0	0	8.8	0
Solar Radiation (MJ/m2)	0.57	0.01	0	0.87	0.75	0	0.93	3.52	0
Temperature(°C)	12.88	13.7	20.5	11.94	142.7	3.5	22.5	39.4	-17.8
Visibility (10m)	1437	1698	2000	608.3	370027	940	2000	2000	27
Wind speed (m/s)	1.72	1.5	1.1	1.04	1.07	0.9	2.3	7.4	0

Here we have generated a summary table of some statistics of the quantitative variables. For some of the extreme skew distributions from before, we can see that some of the quartile values correspond with the minimum such as rainfall/snowfall. These zero values in our data may not be very helpful in predicting our response value and could be considered for removal in our model. Likewise for visibility, most values seem to correspond with the maximum, which means a possible limit in measurement of data. This limit could be obscuring some trends past value of 2000 and may also be considered for removal.

Plotting response variable against others Now that we've made some observations on the individual variables themselves, we will plot these variables against the Rented Bike Count to see how their values relate to the number of bikes rented.



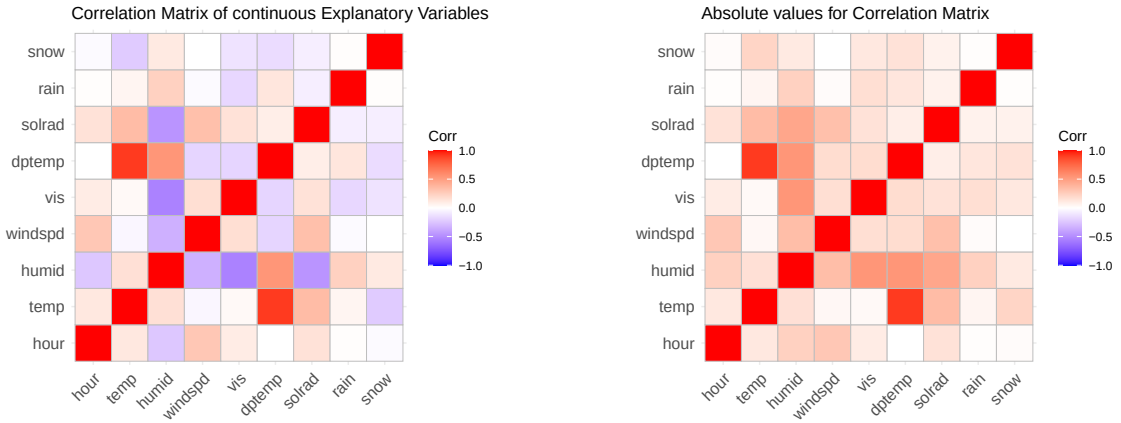
From these plots, we can see that some variables have linear trends such as temperature, and dew point temperature. Other variables have non-linear trends, which may be better fitted with some polynomial or logarithmic methods later on.



For categorical variables, we can see how different factors affect the rented bike count. Some observations are, rented bike count is always 0 for non-functioning day, and bike count is higher in warmer months.

Plotting correlation between explanatory variables After looking and noticing some observations from plots, we can further explore relationships in our model my looking at correlation between the variables and the response.

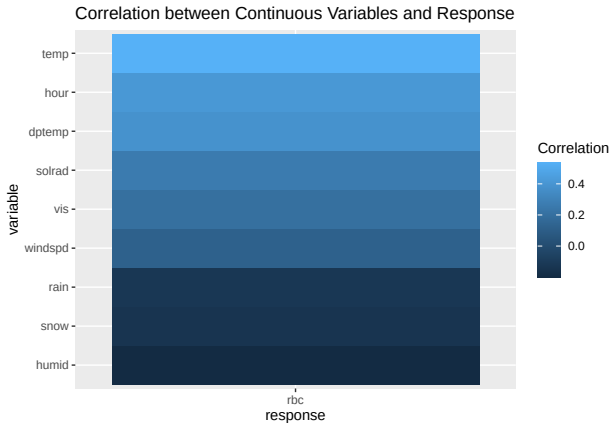
	hour	temp	humid	windspd	vis	dptemp	solrad	rain	snow
hour	1	0.124	-0.242	0.285	0.099	0.003	0.145	0.009	-0.022
temp	0.124	1	0.159	-0.036	0.035	0.913	0.354	0.05	-0.218
humid	-0.242	0.159	1	-0.337	-0.543	0.537	-0.462	0.236	0.108
windspd	0.285	-0.036	-0.337	1	0.172	-0.176	0.332	-0.02	-0.004
vis	0.099	0.035	-0.543	0.172	1	-0.177	0.15	-0.168	-0.122
dptemp	0.003	0.913	0.537	-0.176	-0.177	1	0.094	0.126	-0.151
solrad	0.145	0.354	-0.462	0.332	0.15	0.094	1	-0.074	-0.072
rain	0.009	0.05	0.236	-0.02	-0.168	0.126	-0.074	1	0.008
snow	-0.022	-0.218	0.108	-0.004	-0.122	-0.151	-0.072	0.008	1



From correlation matrix and heat map, we can see signs of multicolinearity from high correlation between dew point temperature and temperature. Note that humidity has noticeably moderate values of correlation with a few other variables, which we should keep in mind although it is not high enough to be worrying.

Additionally, these values for humidity are negative, meaning that humidity appears to inversely proportional these other variables.

We can also look at how quantitative variables correlate to rented bike count.



	term	Correlation
1	temp	0.5385582
2	hour	0.4102573
3	dptemp	0.3797881
4	solrad	0.2618370
5	vis	0.1992803
6	windspd	0.1211084
7	rain	−0.1230740
8	snow	−0.1418036
9	humid	−0.1997802

Correlation between continuous explanatory variables and response shows that temperature is the most correlated value. Lowest correlation is between rain, snow, and humidity. The takeaway here is that the higher the correlation between rented bike count and the variable is, the more positive contribution that variable gives to the value of rented bike count. With this in mind, variables like temperature and hour will help predict the rented bike count best, while variables at the bottom with negative correlation, will predict against the response, or will decrease the value of rented bike count. t(correlation_matrix)

MODEL BUILDING

We start by building a *main effect* model, that includes all the explanatory variables We regress the response variable against the variables to have a sense of how our model behaves intially.

```
##
## Call:
## lm(formula = `Rented Bike Count` ~ . - Date, data = bike)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1215.94  -274.37   -57.39   211.09  2282.82
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -8.205e+01  9.861e+01  -0.832 0.405404
## Hour                         2.753e+01  7.347e-01  37.470  < 2e-16 ***
## `Temperature(°C)`            1.607e+01  3.662e+00   4.388 1.16e-05 ***
## `Humidity(%)`               -1.081e+01  1.030e+00 -10.498  < 2e-16 ***
## `Wind speed (m/s)`           1.920e+01  5.095e+00   3.769 0.000165 ***
## `Visibility (10m)`           1.029e-02  9.881e-03   1.042 0.297635
## `Dew point temperature(°C)`  1.116e+01  3.835e+00   2.911 0.003608 **
## `Solar Radiation (MJ/m2)`   -7.731e+01  7.593e+00 -10.182  < 2e-16 ***
## `Rainfall(mm)`              -5.848e+01  4.270e+00 -13.694  < 2e-16 ***
## `Snowfall (cm)`              3.269e+01  1.121e+01   2.918 0.003534 **
## SeasonsSpring               -1.353e+02  1.388e+01  -9.749  < 2e-16 ***
## SeasonsSummer               -1.545e+02  1.721e+01  -8.976  < 2e-16 ***
## SeasonsWinter               -3.661e+02  1.971e+01 -18.574  < 2e-16 ***
## HolidayNo Holiday            1.176e+02  2.160e+01   5.442 5.40e-08 ***
```

```
## `Functioning Day`Yes         9.321e+02  2.665e+01  34.974  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 432.8 on 8745 degrees of freedom
## Multiple R-squared:  0.5504, Adjusted R-squared:  0.5497
## F-statistic: 764.8 on 14 and 8745 DF,  p-value: < 2.2e-16
```

Based on the produced output we can see the relationship with the response and explaintory variables is about **55%**. We begin to test individual variables against the hypothesis of that estimate being irrelavant. We also notice multi-collinearity as shown by the output in the above graphs. We will remove those variables in hopes of improving our relationship and make our model simpler leading to a higher predictive function.

```
##
## Call:
## lm(formula = `Rented Bike Count` ~ . - Date - `Humidity(%)`,
##     data = bike)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1181.75  -282.31   -54.11   213.87  2316.18
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.037e+03  3.824e+01 -27.121  < 2e-16 ***
## Hour                         2.787e+01  7.386e-01  37.734  < 2e-16 ***
## `Temperature(°C)`            5.056e+01  1.627e+00  31.072  < 2e-16 ***
## `Wind speed (m/s)`           1.812e+01  5.126e+00   3.535 0.000410 ***
## `Visibility (10m)`           3.657e-02  9.618e-03   3.802 0.000145 ***
## `Dew point temperature(°C)` -2.643e+01  1.380e+00 -19.154  < 2e-16 ***
## `Solar Radiation (MJ/m2)`   -7.826e+01  7.640e+00 -10.244  < 2e-16 ***
## `Rainfall(mm)`              -6.592e+01  4.237e+00 -15.558  < 2e-16 ***
## `Snowfall (cm)`              1.791e+01  1.119e+01   1.601 0.109449
## SeasonsSpring               -1.361e+02  1.397e+01  -9.745  < 2e-16 ***
## SeasonsSummer               -1.456e+02  1.730e+01  -8.420  < 2e-16 ***
## SeasonsWinter               -3.565e+02  1.981e+01 -17.995  < 2e-16 ***
## HolidayNo Holiday            1.155e+02  2.174e+01   5.312 1.11e-07 ***
## `Functioning Day`Yes         9.272e+02  2.681e+01  34.577  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 435.5 on 8746 degrees of freedom
## Multiple R-squared:  0.5448, Adjusted R-squared:  0.5441
## F-statistic: 805.1 on 13 and 8746 DF,  p-value: < 2.2e-16
```

After one of the highly correlated variables we see a small adjustment in R^2 which is not a significant amount to come to a solid conclusion, so we will keep removing correlated variables till we see an improvement or we run out of correlated variables to remove from our model.

```
## lm(formula = `Rented Bike Count` ~ . - Date - `Humidity(%)` -
##     `Dew point temperature(°C)`, data = bike)
```

```
## [1] 0.5256798
```

```
## [1] 0.5250291
```

Now solrad and snow have become insignificant, and R-squared values dropping by 0.02. We would drop

solrad first, then check and drop snowfall if needed

```
## lm(formula = `Rented Bike Count` ~ . - Date - `Humidity(%)` -
##     `Dew point temperature(°C)` - `Solar Radiation (MJ/m2)`,
##     data = bike)
```

```
## [1] 0.525673
```

```
## [1] 0.5250765
```

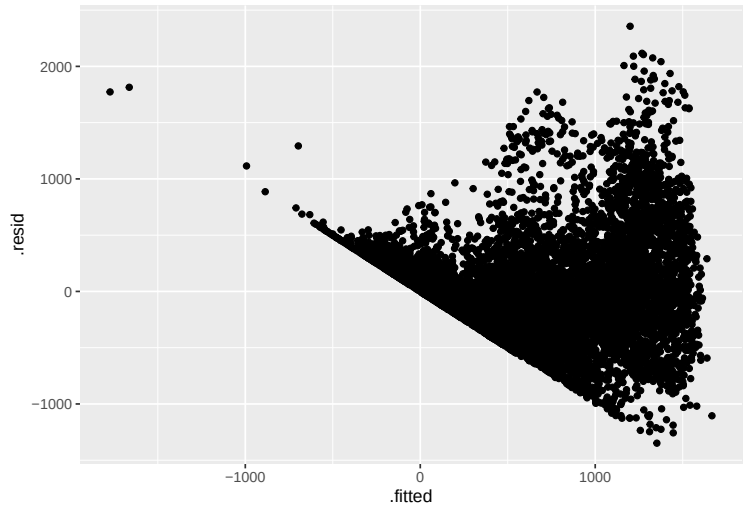
```
## lm(formula = `Rented Bike Count` ~ . - Date - `Humidity(%)` -
##     `Dew point temperature(°C)` - `Solar Radiation (MJ/m2)` -
##     `Snowfall (cm)`, data = bike)
```

```
## [1] 0.525651
```

```
## [1] 0.5251088
```

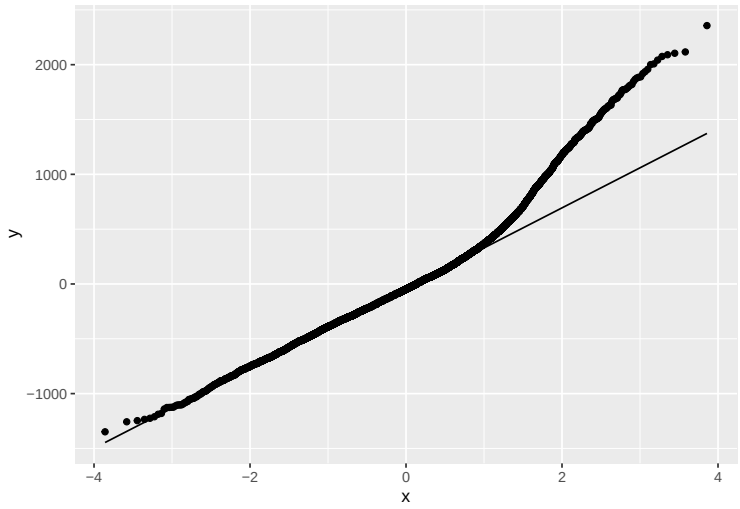
After removing all the correlated variables, we still do not obtain a decent relationship between the response and explanatory variables. We draw residual plots to analyse how the errors are scattered and to further check if our regression assumptions are being met in hopes to create a model that will improve our relationship and hence create a decent predictive model.

```
# Fitted vs residuals
ggplot(model2.3, aes(x = .fitted, y = .resid)) + geom_point()
```




```
# Explanatory vs residuals
#ggplot(model2.3a_long, aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free")

# QQ plot against residuals
ggplot(model2.3, aes(sample = .resid)) + stat_qq() + stat_qq_line()
```




```
bptest(model2.3)    # Breusch-Pagan test, reject constant variance
```

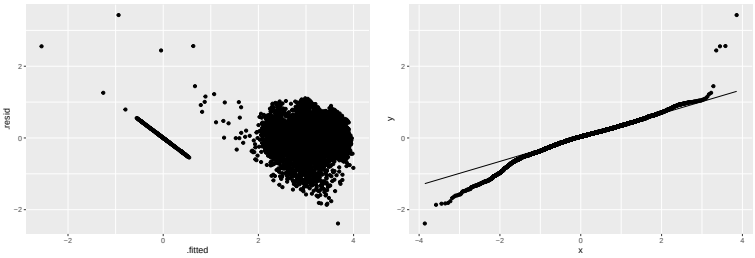
```
##
##  studentized Breusch-Pagan test
##
## data:  model2.3
## BP = 847.4, df = 10, p-value < 2.2e-16
```

The most apparent observation we notice is that our data violates the normality assumption this is further apparent via the **BP** test. Another observation we make is this interesting linear line of residuals in our residual plot, giving us the possible conclusion we are missing some important variable(s) that would bridge the gap of the poor relationship between the response and explanatory variables. Since our normality assumptions have been violated a quick remedy is to apply a box-cox transformation to transform the data to satisfy the normality assumption.

```
## lm(formula = I(`Rented Bike Count`^0.1818) ~ . - Date - `Humidity(%)` -
##     `Dew point temperature(°C)` - `Solar Radiation (MJ/m2)` -
##     `Snowfall (cm)`, data = bike)
```

```
## [1] 0.7703619
```

```
## [1] 0.7700995
```




```
## lm(formula = `Rented Bike Count` ~ . - Date, data = bike, weights = 1/var.s)
```

```
## [1] 0.6063792
```

```
## [1] 0.6057491
```

This transformation seems more inline with a good model, but we still have some outliers and this weird line on the left. Another solution is to you **weighted least squares (WLS)** which normalizes the data

and fixes normality assumptions and constant variance violations. We have tried WLS, but it doesn't help. Doing Box-Cox or Yeo-Johnson yields better results at this moment.

FEATURE IMPORTANCE USING CARET

```
importance <- varImp(model1, scale=FALSE)
print(importance)
```

```
##                                Overall
## Hour                          37.470487
## `Temperature(°C)`              4.388162
## `Humidity(%)`                 10.497744
## `Wind speed (m/s)`             3.769034
## `Visibility (10m)`             1.041580
## `Dew point temperature(°C)`    2.911293
## `Solar Radiation (MJ/m2)`     10.181576
## `Rainfall(mm)`                13.694466
## `Snowfall (cm)`                2.917780
## SeasonsSpring                  9.748721
## SeasonsSummer                  8.976020
## SeasonsWinter                 18.573580
## HolidayNo Holiday              5.442263
## `Functioning Day`Yes          34.974186
```

```
model6 <- lm(I(`Rented Bike Count`^0.1818) ~ .-Date-`Temperature(°C)`-`Dew point temperature(°C)`-`Visi
model6$call
```

```
## lm(formula = I(`Rented Bike Count`^0.1818) ~ . - Date - `Temperature(°C)` -
##     `Dew point temperature(°C)` - `Visibility (10m)` - `Snowfall (cm)` -
##     `Wind speed (m/s)`, data = bike)
```

```
summary(model6)$r.squared
```

```
## [1] 0.7695759
```

```
summary(model6)$adj.r.squared
```

```
## [1] 0.7693389
```

Extension: We used a feature a selection algorithm which gives us possible important features for our model building process. Using the features specified by the algorithm, we include them in our regression and notice a significantly increased relationship of 77%.

We go further and test a model of **Weighted least squares** paired with a **transformation** and notice strong relationship of 83% between our response variable and the selected explanatory variables.

MODEL SELECTION

We have made many models using various ways, however there is still an automated test we can use for model selection which tests different subsets of models, comparing each with different values, which are R^2, R^2_{adj}, cp,AIC. The aim of this automation is to pick a value that **maximizes** the R^2 and R^2_{adj} and **minimizes** AIC. These thresholds usually tend to provide the best possible model.

After we run our algorithm we finally obtain the *best possible predictors* however as we can see the R^2 has not improved much..

Another automated testing alogrithm is **backward elimination** which starts of with a complex model and reduces model complexity by testing relevant features the R output shows the same result as the fully

automated test above.

```
step2 = stepAIC(model3, direction="backward")
```

```
## Start:  AIC=-16441.27
## I(`Rented Bike Count`^0.1818) ~ (Date + Hour + `Temperature(°C)` +
##     `Humidity(%)` + `Wind speed (m/s)` + `Visibility (10m)` +
##     `Dew point temperature(°C)` + `Solar Radiation (MJ/m2)` +
##     `Rainfall(mm)` + `Snowfall (cm)` + Seasons + Holiday + `Functioning Day`) -
##     Date - `Humidity(%)` - `Dew point temperature(°C)` - `Solar Radiation (MJ/m2)` -
##     `Snowfall (cm)`
##
##                       Df Sum of Sq    RSS      AIC
## <none>                             1337.5 -16441.3
## - `Wind speed (m/s)`   1      3.09 1340.6 -16423.1
## - Holiday              1     11.10 1348.6 -16370.9
## - `Visibility (10m)`   1     59.03 1396.6 -16065.0
## - Seasons              3     98.89 1436.4 -15822.4
## - `Rainfall(mm)`       1    158.80 1496.3 -15460.5
## - `Temperature(°C)`    1    169.08 1506.6 -15400.5
## - Hour                 1    258.42 1596.0 -14895.8
## - `Functioning Day`    1   2915.22 4252.8  -6310.2
```

```
pprime2 = step2$rank
n = dim(bike)[1]
aic = step2$anova[6][1]
bic = aic + log(n) * pprime2 - 2 * pprime2
bic
```

```
##         AIC
## 1 -16363.41
```

```
PRESS(model3)
```

```
## [1] 1346.677
```

```
#DFFITS
arr = dffits(model3)
print(length(which(arr > 1)))
```

```
## [1] 4
```

```
#DFBETAS
arr1 = dfbetas(model3)
print(length(which(arr > 1)))
```

```
## [1] 4
```

```
#COOK'S DISTANCE
library(car)
arr2 = cooks.distance(model3)
print(length(which(arr > qf(0.2, pprime2, n-pprime2))))
```

```
## [1] 5
```

TRAIN TEST SPLIT

We need to validate our model, and to do so we need to create a training set and a test set. We split our initial dataset into a training set and a test set with proportion 80:20.

```
data_train <- create_train_test(bike, size = 0.8, train = TRUE)
data_test <- create_train_test(bike, size = 0.8, train = FALSE)
```

We now run the model validation for our best performing multiple regression model. First, we would fit the model on the training set, and use the model to make the prediction. We would then compute the MSE and MSPR of the model to evaluate the predictive power. Here, we can observe a huge discrepancy between the MSE and MSPR, which shows that there might be an important feature missing in the dataset. We then calculate RMSE and R^2 for our prediction to compare with other models.

```
model6_training <- lm(I(`Rented Bike Count`^0.1818) ~ .-`Humidity(%)`-`Dew point temperature(°C)`-`Sola
#MSE
mean(model6_training$residuals^2)
```

```
## [1] 0.1506399
```

```
#summary(model6_training)
model6_pred <- predict.lm(model6_training, data_test[,-c(1)]) # excluding date, rented bike
mspr = mean((data_test$`Rented Bike Count` - model6_pred)^2)
print(mspr)
```

```
## [1] 944384.9
```

```
print(PRESS(model6_training))
```

```
## [1] 1064.498
```

```
#This is to compare with ML models
baseline_rmse = RMSE(data_test$`Rented Bike Count`, model6_pred)
baseline_r2 = cor(data_test$`Rented Bike Count`, model6_pred) ^ 2
```

Machine Learning Models

Decision Tree

We would first fit the decision tree regression to our data. The final R^2 is 0.4055, which is not so good.

```
decisiontree <- rpart(`Rented Bike Count`~.-Date-`Dew point temperature(°C)`-`Solar Radiation (MJ/m2)`-
pred = predict(decisiontree, data_test, method = "anova")

#RMSE
dt_rmse = RMSE(pred = pred, obs = data_test$`Rented Bike Count`)
#R^2
dt_r2 = cor(data_test$`Rented Bike Count`, pred) ^ 2
```

Random Forest

Next, we would try to fit the random forest model. This model is an ensemble of decision tree, and we would expect the result to be better than decision tree. Indeed, we can observe that the R^2 is 0.74 and the RMSE is quite small.

```
random_forest <- randomForest(`Rented.Bike.Count`~.-Date-`Dew.point.temperature..C.`-`Solar.Radiation..
p2 <- predict(random_forest, data_test2)
rf_rmse = RMSE(pred = p2, obs = data_test2$`Rented.Bike.Count`)
#R^2
rf_r2 = cor(data_test2$`Rented.Bike.Count`, p2) ^ 2
```

XGBoost

We would now try extreme gradient boosting regression, a famous model using in many competitions in Kaggle. The final result is comparable to random forest, with $R^2 = 0.71$ and RMSE is also acceptable.

```
xgbr = xgboost(data=xgb_train, max.depth = 2, nrounds=50)
pred2 = predict(xgbr, xgb_test)
xgbr_rmse = RMSE(pred2, data_test$`Rented Bike Count`)
# R^2
xgbr_r2 = cor(data_test2$`Rented.Bike.Count`, pred2) ^ 2
```

Support Vector Regression

We would now fit the support vector regression model. The final result is not good with $R^2 = 0.58$. One rationale for this problem is that we are using the default kernel for SVM. We might need to find another kernel that works well for our particular dataset.

```
modelsvm <- svm(`Rented Bike Count`~.-Date-`Dew point temperature(°C)`-`Solar Radiation (MJ/m2)`-`Snowfa
#Predict using SVM regression
predictions = predict(modelsvm, subset(n_data_test, select = -c(Seasons, `Rented Bike Count`)))

svr_rmse = RMSE(predictions * (max(data_test$`Rented Bike Count`) - min(data_test$`Rented Bike Count`))
#R^2
svr_r2 = cor(data_test$`Rented Bike Count`, predictions* (max(data_test$`Rented Bike Count`) - min(data_
```

LightGBM

LightGBM seems to be our best performing ML model with $R^2 = 0.76$ and $RMSE = 324$. This is the highest R^2 value and smallest $RMSE$ value for all of our models. It seems like all the tree-based algorithm, such as random forest, XGBoost or LightGBM works really well for our data.

```
model <- lightgbm(params = list(objective = "regression", metric = "l2"), data = dtrain)
lgbmpred = predict(model, X_test)
```

```
lgbm_rmse = RMSE(pred = lgbmpred, obs = data_test$`Rented Bike Count`)
#R^2
lgbm_r2 = cor(data_test$`Rented Bike Count`, lgbmpred) ^ 2
```

KNN

In the following model, we use K-Nearest Neighbors to predict the bike count. We can see that the result isn't stellar ($R^2 = 0.53$), and RMSE seems to have a fairly large value. One reason is that our data has many features, which means it has high dimensional structure, and KNN does not work well under this setting.

```
knnmodel = knnreg(X2, y2)
pred_y = predict(knnmodel, X_test2)
knn_rmse = RMSE(data_test$`Rented Bike Count`, pred_y * (max(data_test$`Rented Bike Count`) - min(data_
#R^2
knn_r2 = cor(data_test$`Rented Bike Count`, pred_y * (max(data_test$`Rented Bike Count`) - min(data_tes
```

Neural Network

We use a one-layer neural network on our dataset. The result is not good for the neural network, which is quite surprising for us. In general, neural network does not work well for sparse data, since we have some columns with a lot of zeros. Moreover, it seems like neural network is a better fit for classification tasks instead of regression tasks.

```
nn <- nnet(`Rented.Bike.Count`~.-`Dew.point.temperature..C.`-`Solar.Radiation..MJ.m2.`-`Snowfall..cm.`,

nnpred = predict(nn, data_test3)
```

```
nn_rmse = RMSE(data_test$`Rented Bike Count`, nnpred * (max(data_test$`Rented Bike Count`) - min(data_t
#R^2
nn_r2 = cor(data_test$`Rented Bike Count`, nnpred * (max(data_test$`Rented Bike Count`) - min(data_test
```

Comparision for the predictive performance of our models:

Model	RMSE	R-Squared
Baseline model	971.8	0.3759
Decision Tree	490.2	0.4055
Random Forest	331.7	0.7364
Support Vector Regression	481	0.5846
LightGBM Regression	324.2	0.7648
XGBoost Regression	371.9	0.7138
KNN Regression	453.1	0.5329
One-layer Neural Network	609.2	0.4971

We can observe that the predictive power of the machine learning models is much greater than that of our baseline multiple linear regression model. Especially, tree-based algorithm such as random forest and gradient boosting algorithm works especially well for our particular dataset.

Discussion/Conclusion

The optimal model we ended up was a simple additive model with a transformed response variable. With it we found that Wind Speed, Holiday, Visibility, Seasons, Rainfall, Temperature, Hour, and Functioning Day are significant and good predictors. The other variables that we dropped do not help with prediction with our model. There are limitations because of the discrepancy between the low MSRP and MSE of our model. Furthermore, we see with how the machine learning models only hovering around a 0.8 R-squared, there may be missing data features. Going further we may explore time series given the rented bike count plot, as well as ridge regression since it helps solve the problems of multicollinearity in a general sense for regression. Additionally, we could also explore cyclic transformations on the Hour variable or take it as a categorical variable.

References

E, S. V., Park, J., & Cho, Y. (2020). Using data mining techniques for bike sharing demand prediction in Metropolitan City. Computer Communications, 153, 353–366. https://doi.org/10.1016/j.comcom.2020.02.007

E, S. V., & Cho, Y. (2020). A rule-based model for Seoul bike sharing demand prediction using weather data. European Journal of Remote Sensing, 53(sup1), 166–183. https://doi.org/10.1080/22797254.2020.1725789

E, S. V., Park, J., & Cho, Y. (2020). Seoul bike trip duration prediction using data mining techniques. IET Intelligent Transport Systems, 14(11), 1465–1474. https://doi.org/10.1049/iet-its.2019.0796