# Demand forecasting of Bikeshare System

Xinyue Zhang

April 20, 2020

## 1 Introduction

A bike-sharing system is a service which allows the users to rent or use bicycles available for shared use on a short term basis for an affordable price. Currently there are over 500 bike-sharing programs around the world. The number of users may vary from time to time for such programs. Thus, it's essential for data scientists to develop algorithms to predict the number of bike rentals in a given hour given the environmental and seasonal conditions for that hour.

## 2 Data Description

The dataset of interest was collected through Capital Bikeshare System, Washington D.C for two years (2011 and 2012). The dataset contains 17379 observations with each observation corresponding to one particular hour. The dataset contains 17 variables.

The goal of this project is to 1) understand what regression model is the best for predicting the total number of bikes(denoted as variable cnt) rented in a particular hour and 2) interpret the result of chosen model and determine whether it is resonable to view the fitted parameters as causal effects and 3) provide guidance for future studies of bikeshare use based on the form of model.

## 3 Exploratory Data Analysis

The goal of Exploratory Data Analysis is to identify and understand the correlation between bicycle rent count and the other explanatory variables in the dataset. Some initial hypotheses are listed below.

- **Seasonal effect**: In some seasons like summer and fall, people may have higher demand for bicycle rental.
- **Holiday effect**: Causal users may demand more bikes on holidays as opposed to registered users who may demand more bikes on weekdays.
- **Weekday effect**: Registered users may demand more bikes on weekdays as compared to weekend or holiday.
- **Weather effect**: Demand for bikes may be lower on a rainy day as compared to a sunny day. There also might be a correlation between temperature and the demand of bikes.
- **Environmental condition**: Temperature, humidity and windspeed might also correlate with the demand for bicycle rental.

- **User status**: Total demand should have higher contribution of registered user as compared to casual because registered user base would increase over time. Registered users and casual users might have different behavioral patterns of bike usage.

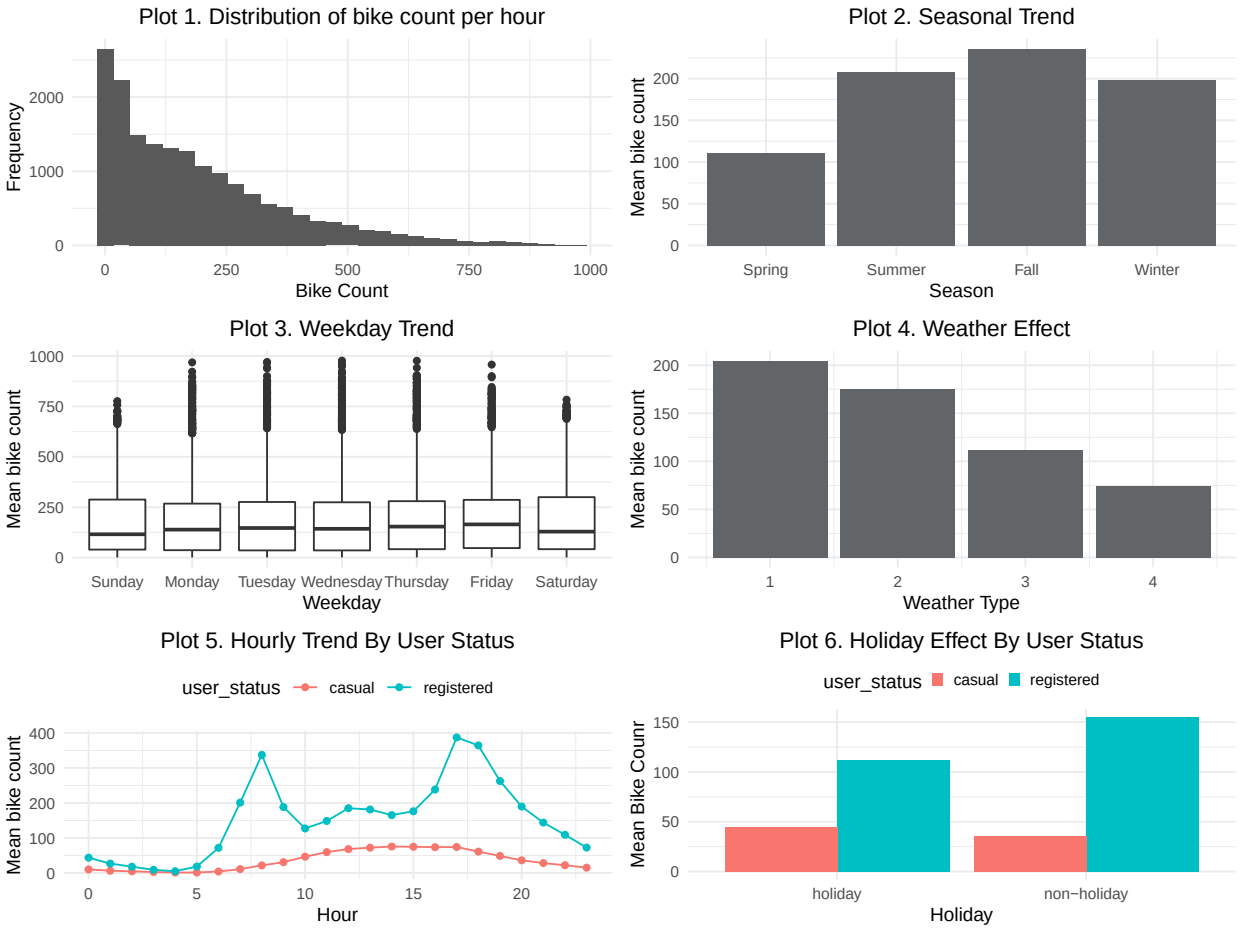


Figure 1: Exploratory Data Analysis Plots

**Summary of EDA plots**:

- The distribution of the dependent variable `cnt` is right-skewed. We will apply transformation to `cnt` to make it approximate to a normal distribution.
- Demand for bikes tends to be lower during `Spring` compared to `Summer`, `Fall` and `Winter`.
- Plot 3 shows demand for bikes tends to be high during weekday than weekends.
- Plot 4 shows how weather affects bike usage. As weather type goes from 1(good weather) to 4(extremely bad weather), the weather conditions become worse and fewer bikes are being demanded.
- For registered users, the hourly effect is stronger as Plot 5 shows that bike usage during 7-9am and 4-7pm is nearly 3 times higher for registered users than the rest of the hours. For casual users, we observe a flatter curve with 2-5 pm as the peak hours that users rent bikes.
- The holiday effect depends on the user status. For registered users, bike usage count is lower on holiday than nonholiday on average. For casual users, the demand is higher on holiday.

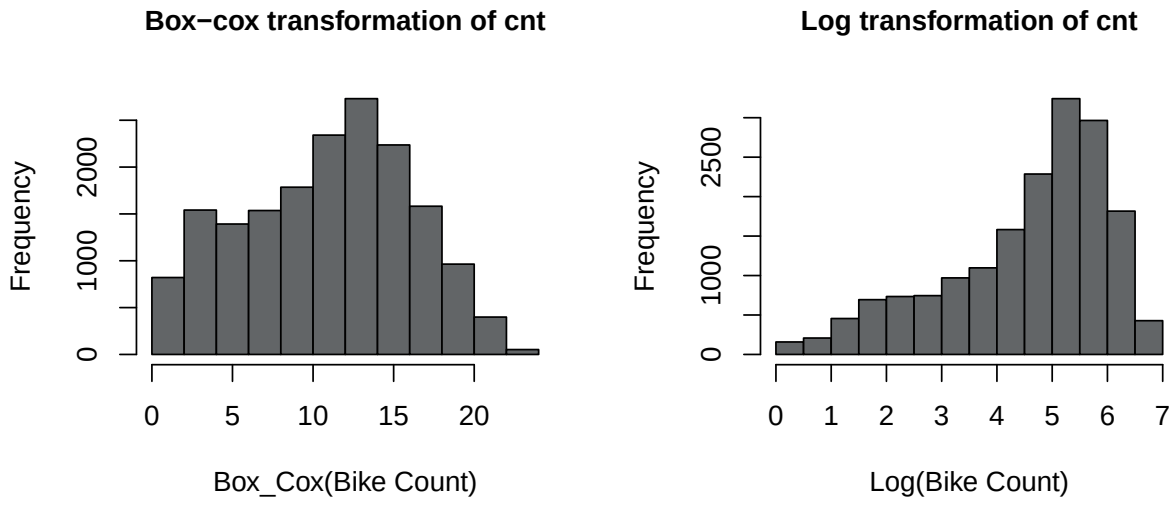


Figure 2: Distributoin of bike count per hour after two types of transformation. We observe that Box-cox transformation approximates the normal distribution better than log transformation so we use it to normalize 'cnt'.

# 4 Model Selection

Before diving into model selection, we performed feature engineering by setting all categorical variables as factors, and turning them into dummy variables and creating the two more variables to improve the prediction power of model.

- `Day_type`: indicates whether the day is `holiday`(0), `weekend`(1) or `weekday`(2)
- `Rush_hour`: indicates whether the hour is during rush hour(1) `7-9am, 4-7pm` or not(0)

In this analysis, we focus on the following 6 methods to build a model for predicting the bicycle usage count for a particular hour.

- Shrinkage methods:
  - Lasso regression
  - Ridge regression
- Stepwise forward methods:
  - Forward selection using RSS
  - Forward selection using Adjusted $R^2$
  - Forward selection using BIC
  - Forward selection using Mallow Cp.

# 5 Model Evaluation

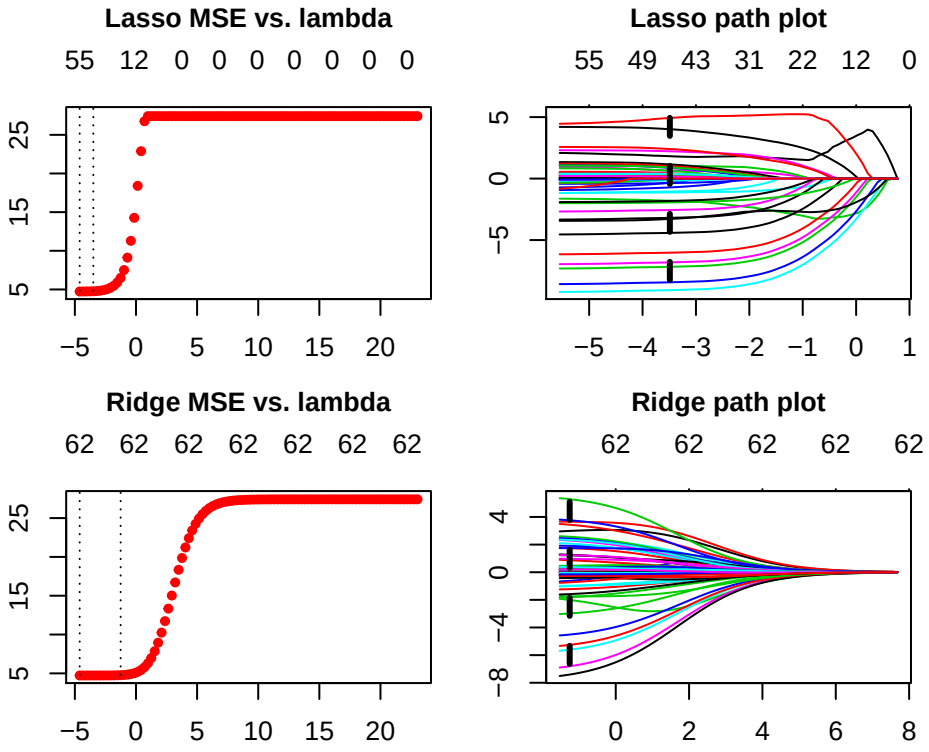


Figure 3: Shrinkage method

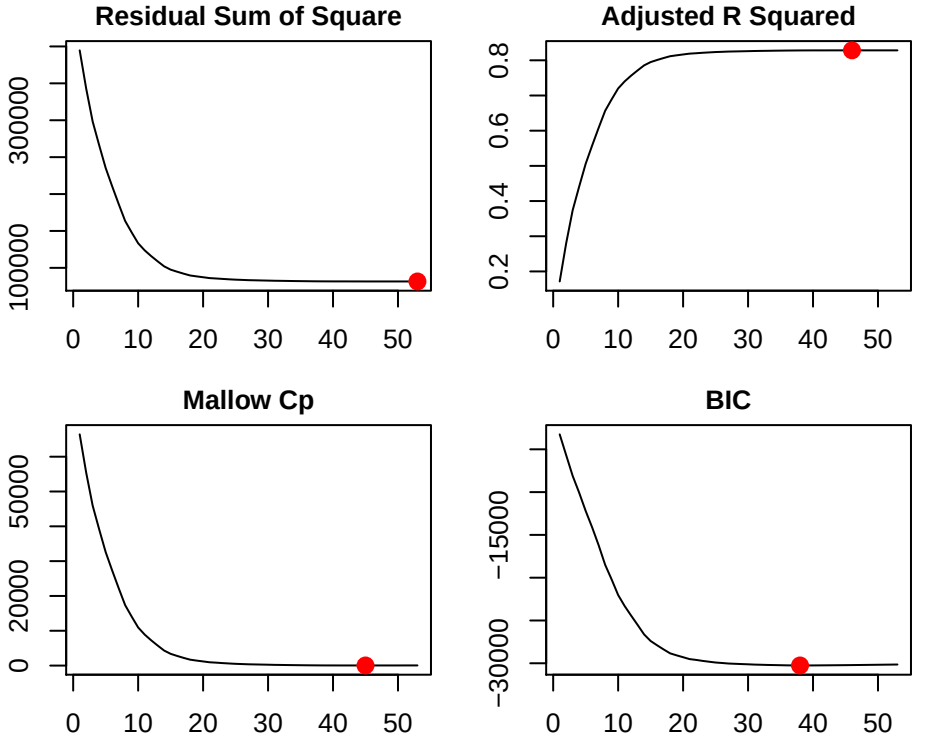


Figure 4: Stepwise forward method

Table 1: Evaluation of various model fits using the cross validation.

| Method | CV Error | # features selected |
|---|---|---|
| Forward Mallow CP | 4.720 | 46 |
| Forward Adjusted R square | 4.721 | 47 |
| Forward RSS | 4.724 | 54 |
| Forward BIC | 4.726 | 39 |
| Ridge | 4.727 | 63 |
| Lasso | 4.732 | 47 |

Table 1 displays the cross-validated error for each of the method. We observe that Forward selection with Marllow CP slightly outperforms the competing regularized linear regression models and other stepwise forward models.

# 6 Diagnostics, Interpretation

In this analysis, we focus on the diagnostics plot of Forward Selection with Mallow's CP to see if it is an appropriate fit for the data and interpret our final model.

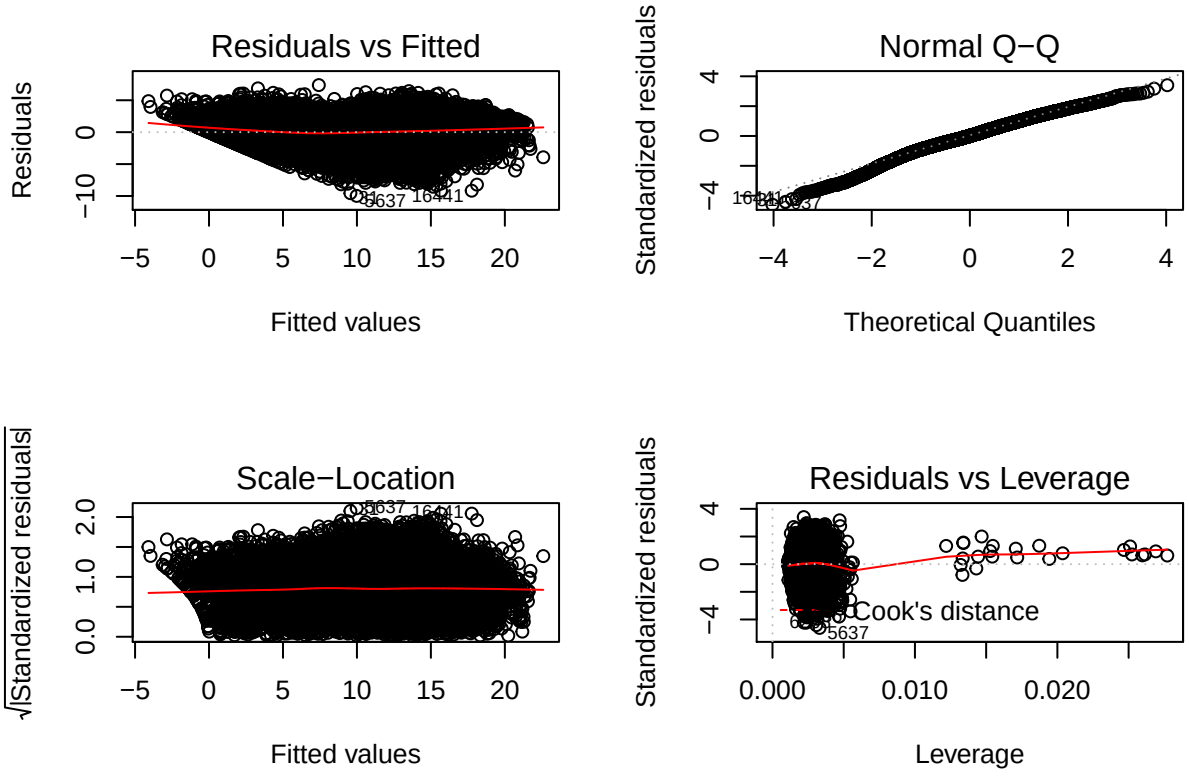


Figure 5: Diagnostics of Forward Selection with Mallow's cp

- Residuals vs fitted values plot shows that most residuals seem to be centered around 0 as the red straight line suggests. The plot does not have a "fanning" pattern and noises are just due to chance. Therefore the residuals have a linear pattern.
- Normal Q-Q plot indicates that the residuals are mostly normally distributed.Thus the normality assumption is also satisfied.
- Scale-location plot shows that the residuals are spread equally along the range of predictors. Since the red line is straight so we can infer that the data follows homoscedasticity.
- The Residuals vs. leverage plot indicates that there are no very high leverage points.

Now, to interpret our results, we examine and compare the selected variables in our models. In particular, we extract the top 20 features as measured by the cofficient magnitude. We observe some overlap among the top selected features between different models. For instance, `hour` is frequently selected by these models since it has large coefficient magnitude and plays an important role in the prediction or of bikeshare use. Future studies can look further into the casual effects of hour on bike usage and come up with a time series model that captures hourly effects and seasonality. We also observe that there's a systematic difference in rental bike usage for causal and registered users, as they have distinct patterns in hourly trend and holiday effect. Again, for future research purpose, one can use random forest regression to differentiate between casual users and registered users. Random forest can handle colinearity in the features really well and it's particularly good at handling high dimensional data.

Note that in our analysis thus far, we have not performed any inference on the estimated parameters, but one could do hypothesis testing on the estimated coefficents by using bootstrap to obtain an estimate of the variance of the parameters and using an appropriate test statistic. It's hard to say whether it is resonable to view all the fitted parameters as causal effects, even though our model seems to be a reasonable fit given the results of the diagnostics, it's likely that we haven't included all the confounding variables in our model. For instance, geographic location of the rentals(whether it's in metropolis area or suburban area), whether there're huge public event in the city that might affect traffic and the demand for bike usage on that day.

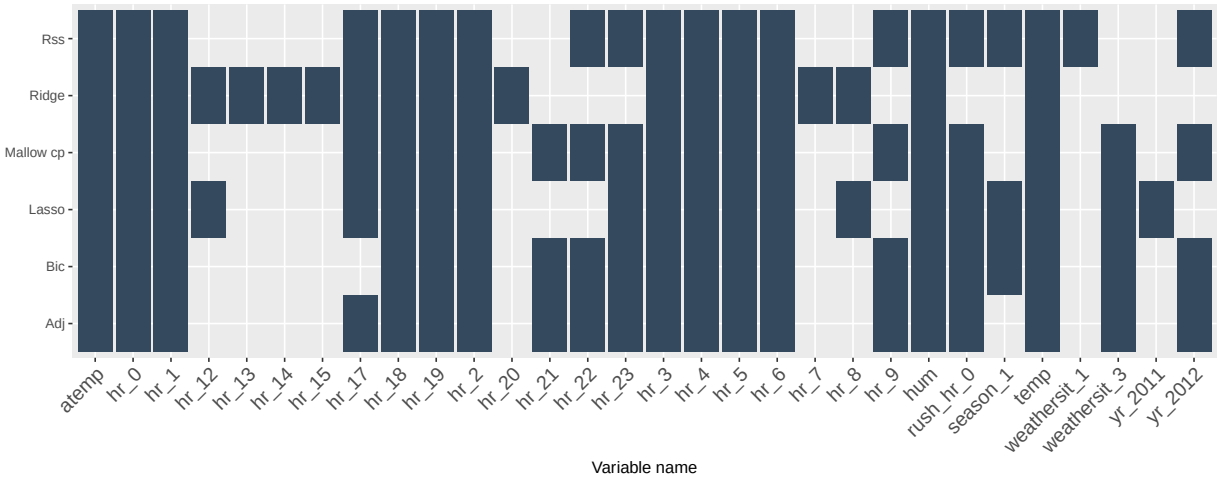


Figure 6: Top 20 features selected by each model

# 7 Conclusion

Overall, the forward selection with mallow cp gave us an OLS model that results in the lowest cross-validated error, compared to other selection methods and regularized linear regression model. Our model also appears

to be an appropriate fit for the dataset given the diagnostic plots. Further analysis is needed to justify the casual inference of the fitted parameters in our model. We also observe evidence in the data as possibilities for using time-series model or random forest regression since they might lead to better performance.

# 8 Appendix

```
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  cache = FALSE,
  fig.pos = 'H')

# load in useful packages
library(tidyverse)
library(R.utils)
library(caret)
library(expss)
library(leaps)
library(dplyr)
library(knitr)
library(ggplot2)
library(glmnet)
library(ggpubr)
library(olsrr)
library(fastDummies)
library(StepReg)
bike = read.csv('BikeSharingDataset.csv')
bike = transform(bike, season_name = factor(season,
                                    levels = c(1, 2, 3, 4),
                                    labels = c("Spring", 'Summer', 'Fall', 'Winter')))

bike = transform(bike, weekday_name = factor(weekday,
                                     levels = c(0, 1, 2, 3, 4, 5, 6),
                                     labels = c('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday'

bike = transform(bike, yr = factor(yr,
                                   levels = c(0, 1),
                                   labels = c(2011, 2012)))

bike = transform(bike, holiday_name = factor(holiday,
                                   levels = c(0, 1),
                                   labels = c('non-holiday', 'holiday')))

cnt_status = c(bike$registered, bike$casual)
user_status = c(rep('registered', length(bike$registered)), rep('casual', length(bike$casual)))
hour = c(bike$hr, bike$hr)
bike_status = data.frame(hour, cnt_status, user_status)

bike_general = bike %>% group_by(yr, mnth) %>% summarize(sum_cnt = sum(cnt))
bike_general$date = paste(as.character(bike_general$yr), as.character(bike_general$mnth), rep(1, 24), s
```

```r
bike_general$date = as.Date(bike_general$date)

bike_holiday = bike %>% group_by(holiday_name) %>% summarize(registered_cnt = mean(registered), casual_
cnt = c(bike_holiday$registered_cnt, bike_holiday$casual_cnt)
holiday = rep(c('non-holiday', 'holiday'), 2)
user_status = c(rep('registered', 2), rep('casual', 2))
bike_holiday = data.frame(holiday, cnt, user_status)

plot_bike_hist = bike %>% ggplot(aes(x = cnt)) + geom_histogram() + xlab("Bike Count") +
                      ylab("Frequency") + theme_minimal() + ggtitle('Plot 1. Distribution of bike count

plot_season = bike %>% group_by(season_name) %>% summarize(avg_cnt = mean(cnt)) %>%
              ggplot(aes(x = season_name, y = avg_cnt)) + geom_bar(stat = 'identity', fill = '#626567')
              theme_minimal() + ggtitle('Plot 2. Seasonal Trend') +
              theme(legend.position = "none", plot.title = element_text(hjust = 0.5))

plot_weekday_users = bike  %>%
              ggplot(aes(x = weekday_name, y = cnt)) +
              geom_boxplot() + xlab("Weekday") + ylab("Mean bike count") +
              theme_minimal() + ggtitle('Plot 3. Weekday Trend') +
              theme(legend.position = "top", plot.title = element_text(hjust = 0.5))

plot_weather = bike %>% group_by(weathersit) %>% summarize(avg_cnt = mean(cnt)) %>%
              ggplot(aes(x = weathersit, y = avg_cnt)) + geom_bar(stat = 'identity', fill = '#626567') +
              xlab("Weather Type") + ylab("Mean bike count") +
              theme_minimal() + ggtitle('Plot 4. Weather Effect') +
              theme(legend.position = "none", plot.title = element_text(hjust = 0.5))

plot_hourly_users = bike_status %>% group_by(hour, user_status) %>%
                    summarize(avg_cnt = mean(cnt_status)) %>%
                    ggplot(aes(x = hour, y = avg_cnt, group = user_status)) +
                    geom_line(aes(color = user_status)) + geom_point(aes(color=user_status)) +
                    xlab('Hour') + ylab("Mean bike count") + theme_minimal() +
                    ggtitle('Plot 5. Hourly Trend By User Status') +
                    theme(legend.position="top", plot.title = element_text(hjust = 0.5))

plot_holiday = bike_holiday %>% ggplot(aes(x = holiday, y = cnt, fill = user_status)) +
                                geom_bar(position="dodge", stat = 'identity') + theme_minimal() +
                                ggtitle('Plot 6. Holiday Effect By User Status') + xlab('Holiday') +
                                ylab('Mean Bike Counr') +
                                theme(legend.position="top",
                                      legend.box.spacing = unit(0.2, "cm"),
                                      legend.key.size = unit(0.5, "lines"),
                                      plot.title = element_text(hjust = 0.5))


ggarrange(plot_bike_hist, plot_season, plot_weekday_users, plot_weather, plot_hourly_users, plot_holida
box.cox.cnt = predict(BoxCoxTrans(bike$cnt), bike$cnt)
par(mfrow = c(1, 2))
hist(box.cox.cnt, main = NULL, xlab = 'Box_Cox(Bike Count)', col = '#626567')
title(main = 'Box-cox transformation of cnt', cex.main = 1)

log.cnt = log(bike$cnt)
```

```r
hist(log.cnt, main = NULL, xlab = 'Log(Bike Count)', col = '#626567')
title(main = 'Log transformation of cnt', cex.main = 1)

bike$box.cox.cnt = box.cox.cnt
bike$yr = as.factor(bike$yr)
bike$mnth = as.factor(bike$mnth)
bike$hr = as.factor(bike$hr)
bike$season = as.factor(bike$season)
bike$day_type[bike$holiday == 1] = 0
bike$day_type[bike$holiday == 0 & bike$workingday == 0] = 1
bike$day_type[bike$holiday == 0  & bike$workingday ==  1] = 2
bike$day_type = as.factor(bike$day_type)
bike$rush_hr = 0
bike$rush_hr[bike$hr == 8 & bike$workingday == 1 |
               bike$hr == 17 & bike$workingday == 1 |
               bike$hr == 9 & bike$workingday == 1 |
               bike$hr == 16 & bike$workingday == 1] = 1
bike$rush_hr = as.factor(bike$rush_hr)
bike$weekday = as.factor(bike$weekday)
bike$weathersit= as.factor(bike$weathersit)

df_dummy = dummy_cols(bike[, c('season', 'hr', 'rush_hr',  'yr','mnth', 'weekday', 'day_type', 'weathers

exclude_var = c('season', 'hr', 'yr','mnth','rush_hr', 'weekday', 'weathersit', 'day_type')
df = dplyr::select(df_dummy, -one_of(exclude_var))

X = model.matrix(box.cox.cnt ~., df)
y = df$box.cox.cnt
regfit <- regsubsets(box.cox.cnt ~ ., data = df, nvmax = 60, method = "forward")
reg.summary = summary(regfit)

rss_forward <- reg.summary$which[which.min(reg.summary$rss),]
rss_var = colnames(df[, rss_forward])

adjr2_forward <- reg.summary$which[which.max(reg.summary$adjr2),]
adjr2_var = colnames(df[, adjr2_forward])

bic_forward <- reg.summary$which[which.min(reg.summary$bic),]
bic_var = colnames(df[, bic_forward])

mallows_forward <- reg.summary$which[which.min(reg.summary$cp),]
mallows_var = colnames(df[, mallows_forward])
lambda.grid = 10^seq(10, -2, length=100)
# LASSO model
lasso.mod = glmnet(X, y, alpha=1)
lasso.cv <- cv.glmnet(X, y, alpha=1, lambda = lambda.grid, standardize = TRUE, nfolds=10)
best.lasso.lam = lasso.cv$lambda.1se
best.lasso.coefs <- predict(lasso.mod, type = 'coefficients', s = best.lasso.lam)

# Ridge model
ridge.mod = glmnet(X, y, alpha=0)
ridge.cv <- cv.glmnet(y=y, x=X, alpha = 0, lambda = lambda.grid, standardize = TRUE, nfolds = 10)
best.ridge.lam = ridge.cv$lambda.1se
```

```r
best.ridge.coefs <- predict(ridge.mod, type = 'coefficients', s = best.ridge.lam)

par(mfrow = c(2, 2), mai=c(0.4,0.3,0.6,0.3))

plot(lasso.cv)
title('Lasso MSE vs. lambda', cex.main = 1, line = 2.5)

plot(lasso.mod, xvar = 'lambda')
lines(c(log(best.lasso.lam), log(best.lasso.lam)), c(-1000, 1000), lty = "dashed", lwd = 3)
title('Lasso path plot', cex.main = 1, line = 2.5)

plot(ridge.cv, xvar = 'lambda')
title('Ridge MSE vs. lambda', cex.main = 1, line = 2.5)

plot(ridge.mod, xvar = 'lambda')
lines(c(log(best.ridge.lam), log(best.ridge.lam)), c(-1000, 1000), lty = "dashed", lwd = 3)
title('Ridge path plot', cex.main = 1, line = 2.5)
# Plot stepwise forward plot
par(mfrow=c(2,2), mai=c(0.4,0.3,0.3,0.3))
plot(reg.summary$rss ,xlab="Number of Variables ",ylab="RSS",type="l")
title('Residual Sum of Square', cex.main = 1, line = 0.5)
points(which.min(reg.summary$rss), reg.summary$rss[which.min(reg.summary$rss)], col="red",cex=2,pch=20)

plot(reg.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l")
title('Adjusted R Squared', cex.main = 1, line = 0.5)
points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)], col="red",cex=2,pch

plot(reg.summary$cp ,xlab="Number of Variables ",ylab="Cp", type='l')
title('Mallow Cp', cex.main = 1,  line = 0.5)
points(which.min(reg.summary$cp),reg.summary$cp [which.min(reg.summary$cp)],col="red",cex=2,pch=20)

plot(reg.summary$bic ,xlab="Number of Variables ",ylab="BIC",type='l')
title('BIC', cex.main = 1,  line = 0.5)
points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)],col="red",cex=2,pch=20)
# Cross validation
set.seed(5)
folds = createFolds(bike$cnt , k = 10)
model_candidates = list(rss_var, adjr2_var, bic_var, mallows_var)

get_CV_error <- function(vars){

folds_MSE<- rep(NA, length(folds))
  for (i in 1:length(folds)){
  model<- lm(paste('box.cox.cnt ~ ', paste(vars, collapse = "+")), data = df[-folds[[i]],])
  prediction<- predict(model, df[folds[[i]],])
  true_y<- df[folds[[i]], "box.cox.cnt"]
  folds_MSE[i] = 1/length(folds[[i]]) * sum((prediction-true_y)^2)
  }
  MSE_avg = mean(folds_MSE)
  return(MSE_avg)
}

lasso_cv = min(lasso.cv$cvm)
```

```
ridge_cv = min(ridge.cv$cvm)

ols_error_list = unlist(lapply(model_candidates, get_CV_error))

Method = c('Lasso', 'Ridge', 'Forward RSS', 'Forward Adjusted R square', 'Forward BIC', 'Forward Mallow

cv_error = c(lasso_cv, ridge_cv, ols_error_list)

num_variables = c(length(which(best.lasso.coefs != 0)), length(which(best.ridge.coefs != 0)),
                  length(rss_var), length(adjr2_var),
                  length(bic_var), length(mallows_var))

total_evals = data.frame(Method, cv_error, num_variables) %>%
              arrange(cv_error)  %>%
              rename(`CV Error` = cv_error,
                     `# features selected` = num_variables)

kable(total_evals, digits = 3, caption = "Evaluation of various model fits using the cross validation.")
# Plot diagnostics
mallow_model = lm(paste('box.cox.cnt ~ ', paste(mallows_var, collapse = "+")), data = df)
adj_model = lm(paste('box.cox.cnt ~ ', paste(adjr2_var, collapse = "+")), data = df)
rss_model = lm(paste('box.cox.cnt ~ ', paste(rss_var, collapse = "+")), data = df)
bic_model = lm(paste('box.cox.cnt ~ ', paste(bic_var, collapse = "+")), data = df)

par(mfrow=c(2,2))
plot(mallow_model)
# Model inference
lasso.top.coefs = c(rownames(as.matrix(best.lasso.coefs[order(abs(best.lasso.coefs), decreasing = TRUE)
ridge.top.coefs = c(rownames(as.matrix(best.ridge.coefs[order(abs(best.ridge.coefs), decreasing = TRUE)
mcp.top.coefs = c(rownames(as.matrix(mallow_model$coefficients[order(abs(mallow_model$coefficients), de
rss.top.coefs = c(rownames(as.matrix(rss_model$coefficients[order(abs(rss_model$coefficients), decreasi
bic.top.coefs = c(rownames(as.matrix(bic_model$coefficients[order(abs(bic_model$coefficients), decreasi
adj.top.coefs = c(rownames(as.matrix(adj_model$coefficients[order(abs(adj_model$coefficients), decreasi

models = c(rep('Mallow cp', 20), rep('Adj',20), rep('Rss',20), rep('Bic',20), rep('Lasso',20),rep('Ridg

top.coefs = c(mcp.top.coefs, adj.top.coefs, rss.top.coefs, bic.top.coefs,lasso.top.coefs, ridge.top.coe

# Plot heatmap
data.frame(models, top.coefs) %>% ggplot() + aes(x = as.factor(top.coefs), y = models) + geom_tile(fill
  theme(axis.text.x = element_text(angle = 45, hjust = 1,
                                   size = rel(1.5)),
        axis.text.y = element_text(size = rel(1)))
```