Final Project - Group 14

Andrew Schuck, Charlotte Milard, Hang Tian
UB PERSON IDs: 50469241,50477883,50413372

2022-12-13

Github Repository

This term project paper is saved on GitHub repository for better co-working. Here is the link.
Website: https://github.com/ht6631/STA545_Final_Project

1. Abstract

Various models were applied to bikesharing data, which was very noisy and non-linear to various variables. Notably, linear models were not successful at describing the hour variable, but multiple models said that it was an important factor. Some success what found by reclassifying the hour variable, but it was still difficult to cut through all the noise.

2. Introduction

a. Data description

Read data and simple processing.

```
suppressPackageStartupMessages(library(tidyverse)) # just in case
library(ISLR2)
library(tidyverse)
library(dplyr)
library(naniar)
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(glmnet)) # penalized linear models
suppressPackageStartupMessages(library(glmnetUtils))  # for quality of life functions over glmnet
suppressPackageStartupMessages(library(corrplot)) # correlation plots
suppressPackageStartupMessages(library(pls)) # for pcr
#call data
origin_data=read_csv('Bike-Sharing-Dataset/hour.csv',show_col_types = FALSE)
#Check how many predictors have NAs
origin_data%>%miss_var_summary()%>%filter(n_miss!=0)%>%nrow()%>%print()
```

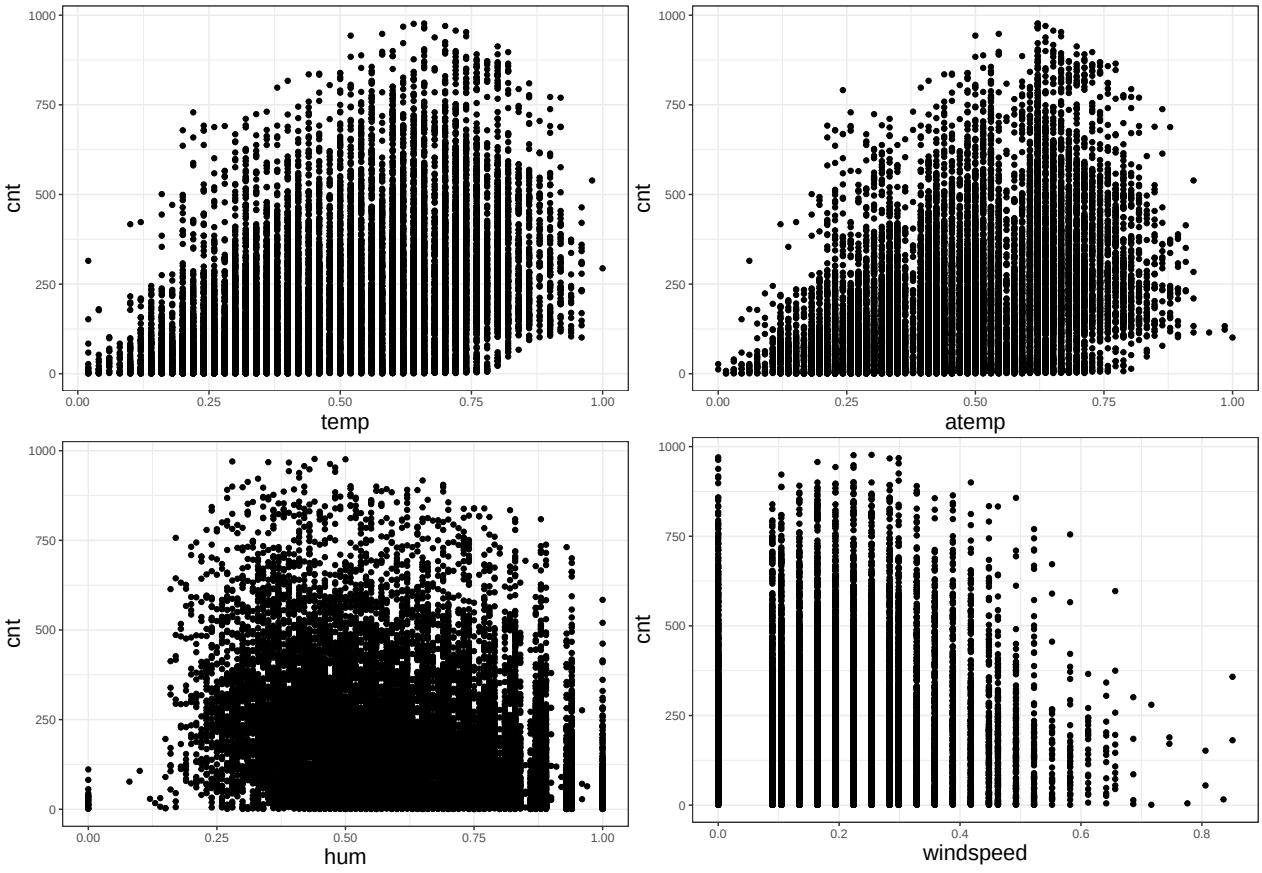
```
## [1] 0
```

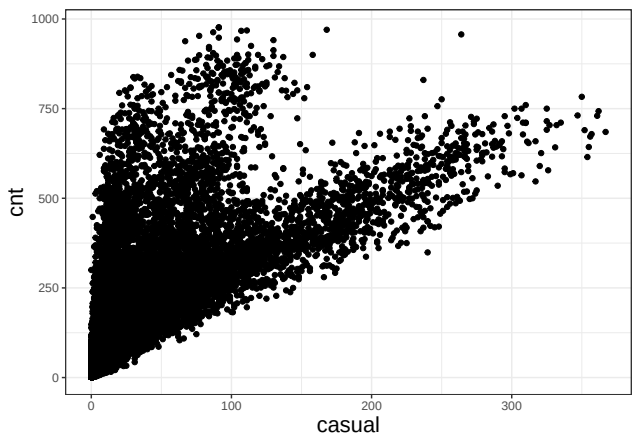
```
#Avoid changing original data
bs_hour=origin_data%>%mutate(dteday=as.Date(dteday))%>%dplyr::select(-instant)
#Add one hourly identifiable column to identify every row
bs_hour=bs_hour%>%mutate(hourly_id=paste(as.character(dteday),as.character(hr)))%>%mutate(hourly_id=ymd_
bs_hour=bs_hour[,c(1:15,17,16)]
bs_hour$windspeed=as.numeric(bs_hour$windspeed)
```

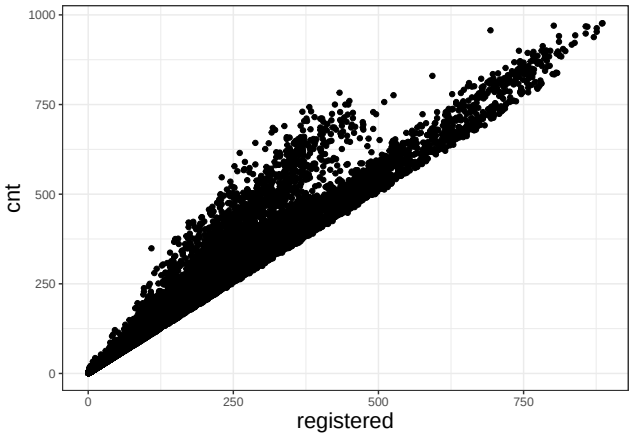
Scatter plots and box plots.

```
col_vec_scatter=colnames(bs_hour)[10:15]
col_vec_box=colnames(bs_hour)[2:9]
for (value in col_vec_scatter) {
    print(ggplot(bs_hour)+geom_point(aes_string(value,'cnt'))+theme_bw()+
            theme(axis.title.y=element_text(size=16),
        axis.title.x=element_text(size=16)))
}
```

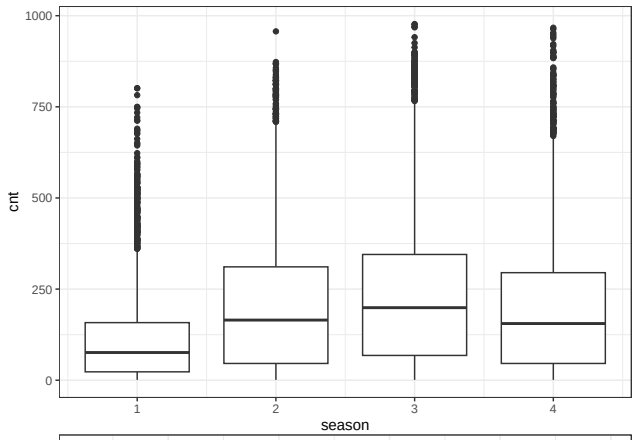
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation ideoms with `aes()`
```

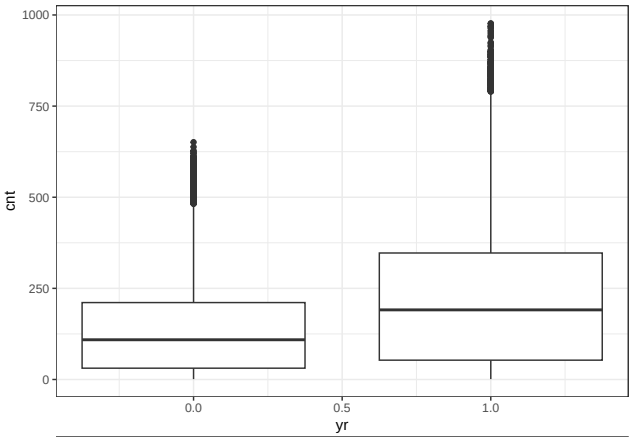


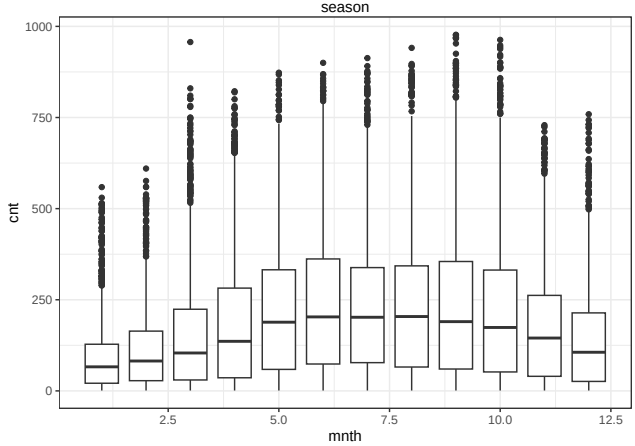


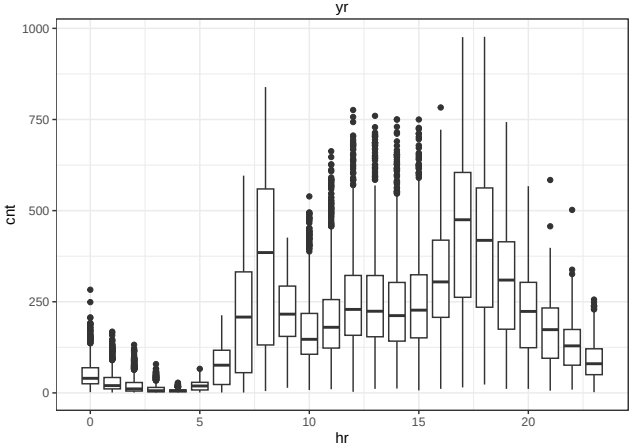


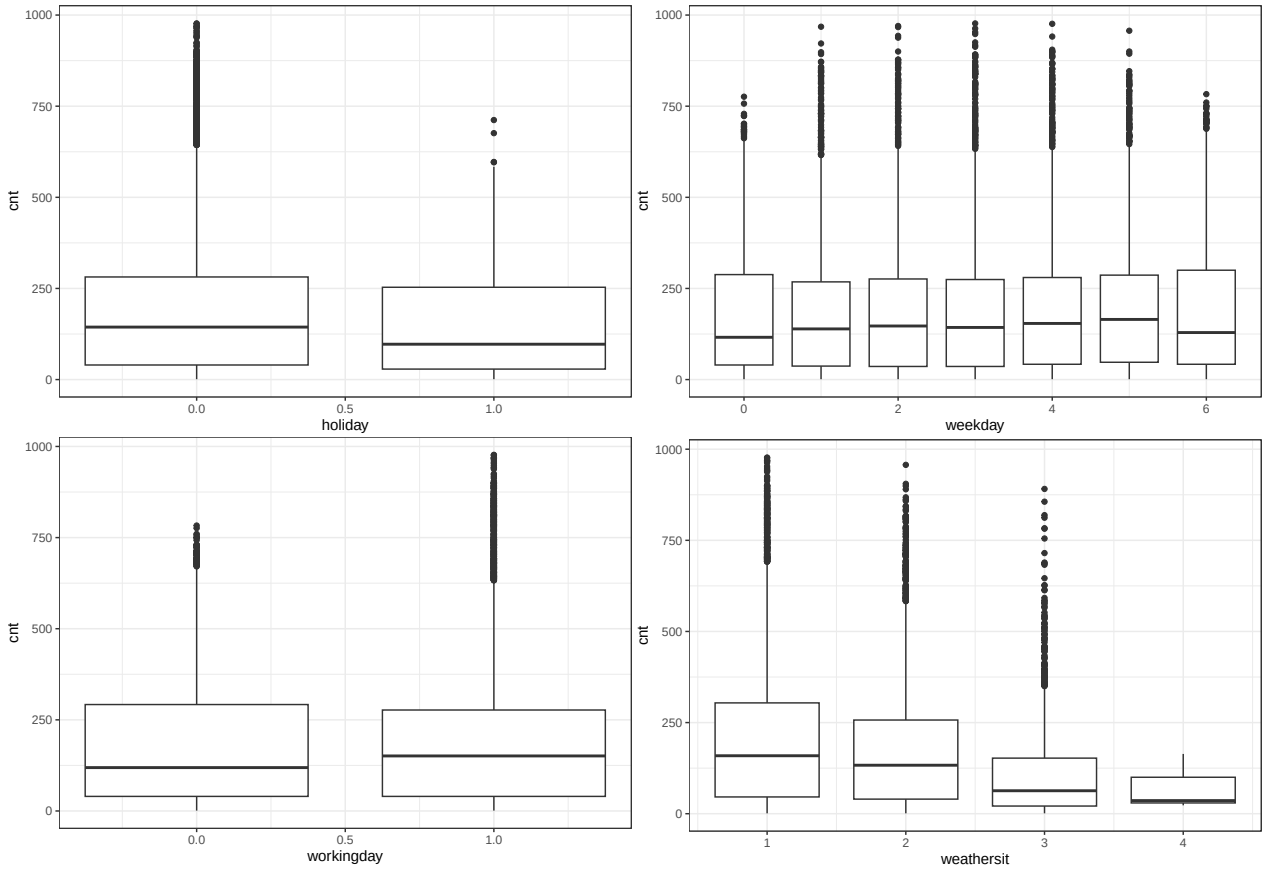

```
for (value in col_vec_box) {
    print(ggplot(bs_hour)+geom_boxplot(aes_string(value,'cnt',group=value))+theme_bw())+
            theme(axis.title.y=element_text(size=16),
        axis.title.x=element_text(size=16))
}
```



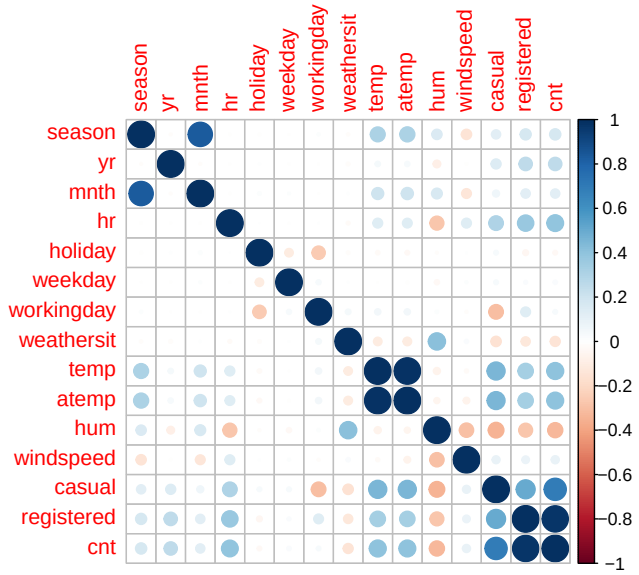









```
cor(bs_hour[, -c(1,16)]) %>%
  corrplot::corrplot()
```



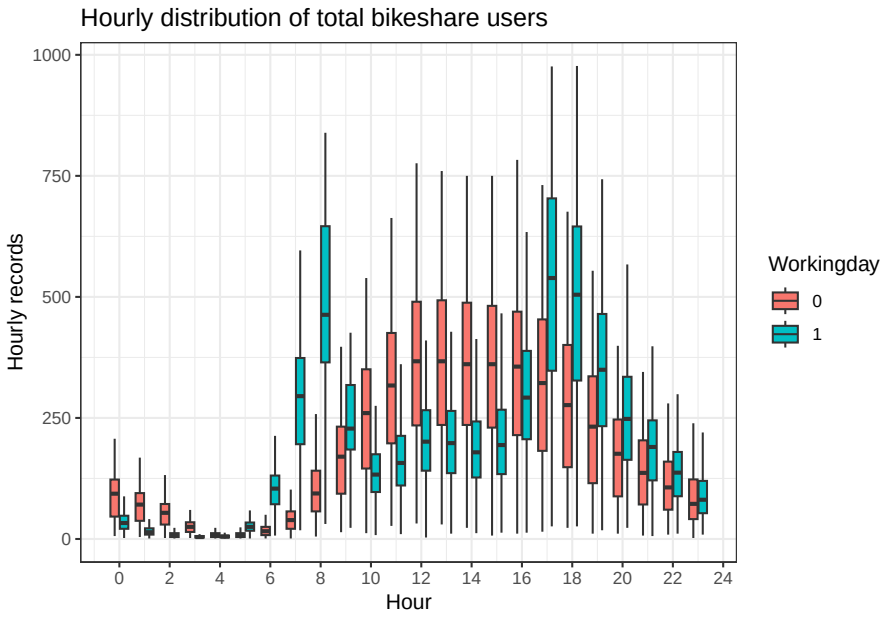
From these plots, it seems counts of total bikeshare users have slight positive correlation with time information like "hr" (hour), similar situations are seen for feeling temperature and temperature. While for "hum" which is humidity, there is a negative correlation.

b. Materials and methods.

First, our group discovered this data set for some useful information.

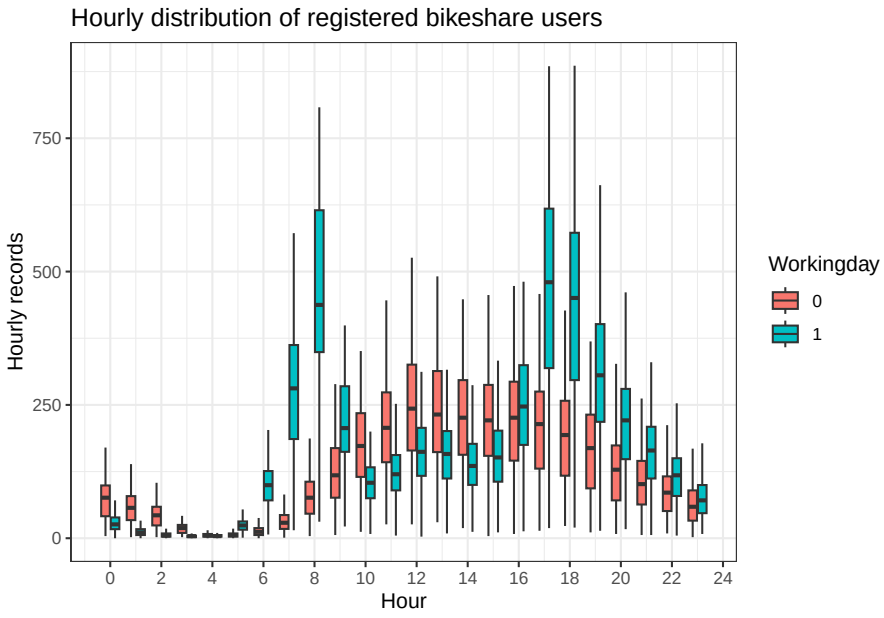
Material question 1. Would the hourly distributions of bikeshare users on working days / non-working days different, and what about casual / registered users?

Total bikeshare users



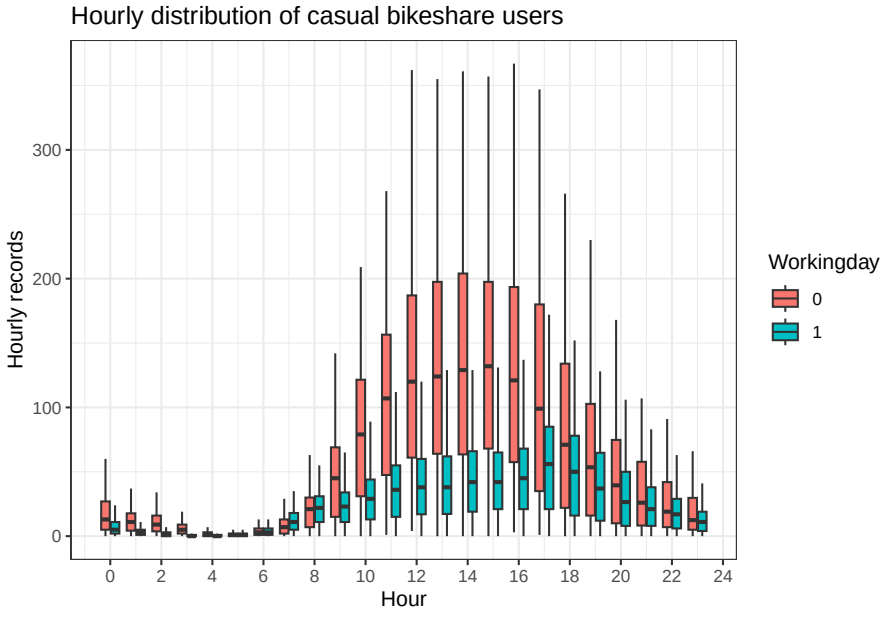
The answer is Yes.
Before 6am, there are a few bike share users for both working-day types while more people on non-working days tend to use bike share from 0am to 2am.
From 6am to 11pm, Two peaks of users are shown around 8am and 5pm on working days, which may reflect commuting during the rush hours. While on non-working days, we saw a smooth increasing then decreasing trend on bike share users.

Registered bikeshare users



The hourly distribution of registered users are quite like that of the total users.

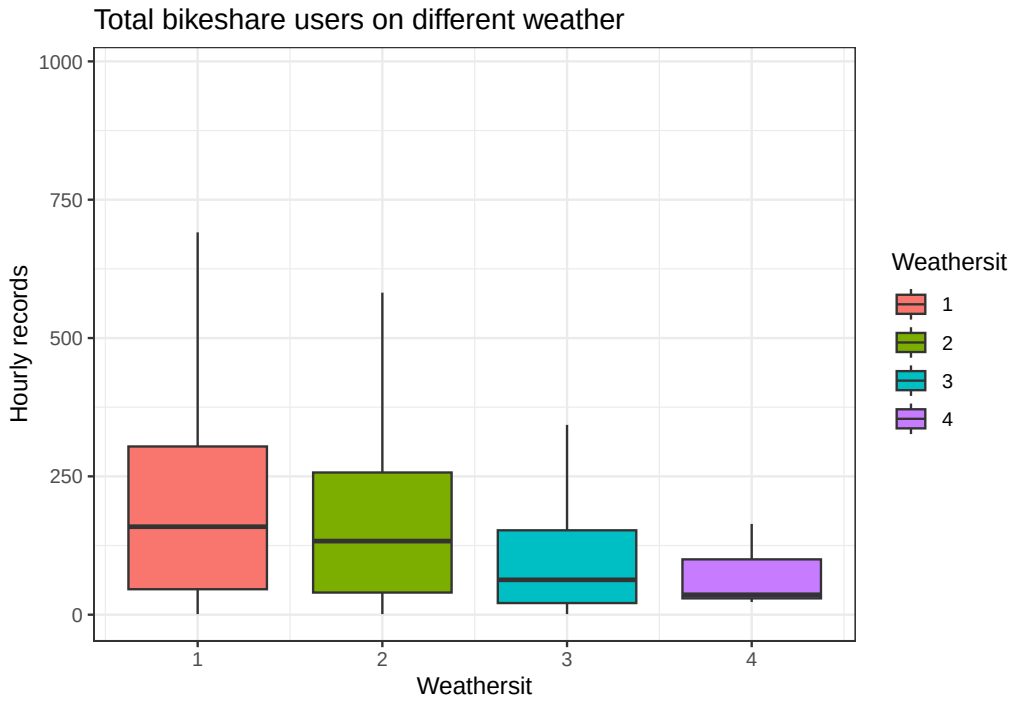
Casual bikeshare users



Casual users tend to use bikeshare more often on non-workding days while there is no strong evidence they would use bikeshare for commuting on rush hours.

Material question 2. Would bikeshare users counts on different weather be different, and what about casual / registered users?

Total bikeshare users



Here's the description for weather type 1 - 4:
weathersit :
- 1: Clear, Few clouds, Partly cloudy, Partly cloudy
- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fogv

From these boxplots, it's obvious that from weather type 1 - 4, people are becoming more unwillingly to use bike share. This makes sense because commonly speaking, ordinary people would see weather type from 1 - 4 as weather getting worse. And for type 4, it's definitely bad weather.

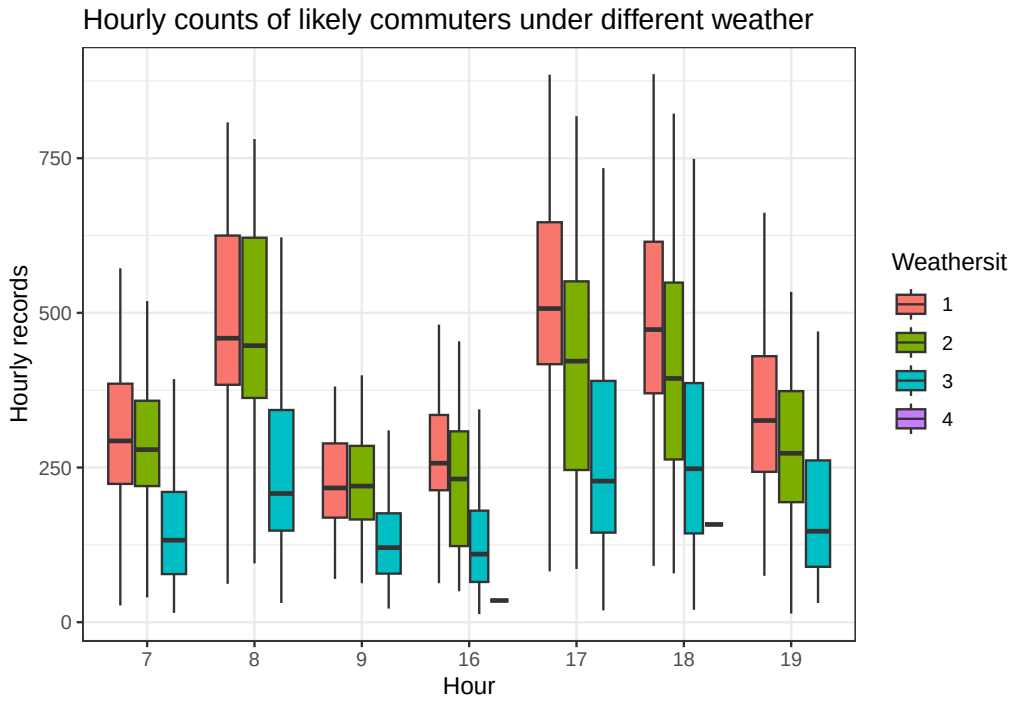
Box plots for registered and casual users show just the same trend.

Material question 3. Influence of weathertype on likely commuters during rush hours

First let's find out likely commuters (registered users during rush hours on workding days)

```
bs_rush=bs_hour%>%filter(hr %in% c(7,8,9,16,17,18,19))%>%filter(workingday==1)
bs_rush$hr=as.factor(bs_rush$hr)
```

Hourly grouped weather influence



During the rush hours in the morning, weather type 1 and 2 seems have similar pattern, a little cloudy weather won't affect people's choice in the morning. Only when weather getting worse to type 3 or 4, people would not willing to ride a bike.

However during the afternoon rush hours, things are different. As the weather getting worse, even a little cloudy would make people not willing to ride a bike.

c. Regression question

After discovering this dataset, our group agreed to study the pattern of "likely commuters" which are registered users during rush hours on working days.

We will include related variables to predict the counts of likely commuters. There are multiple regression methods we used, including PCR, ridge and lasso regression to identify which predictors contribute more, meanwhile, linear regression, decision tree, and random forest are also included for a general modeling.

Methods group 1: linear regression, decision tree, and random forest

```
library(leaps)
library(glmnet)
library(glmnetUtils)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ISLR2)
library(tidyverse)

library(rpart)
library(rpart.plot)
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

```
# This investigation only concerns the rush hours (7AM,8AM,9AM,4PM,5PM,6PM,7PM)

hour_orig <- read_csv('Bike-Sharing-Dataset/hour.csv',show_col_types = FALSE);

# filtering out irrelevant data
hour.filtered <- subset(hour_orig, workingday >.5)
hour.filtered <- subset(hour.filtered, hr %in% c(7,8,9,16,17,18,19))

hour <- hour.filtered[,c("mnth","hr","atemp","weathersit","hum","windspeed","registered")]
```

```
hour.lm <- lm(registered ~ ., data = hour)
hour.lm.r2 <- summary(hour.lm)$r.squared
hour.lm.r2
```

Performing a linear regression, and getting the R^2 value.

```
## [1] 0.2446437
```

The r-squared value for linear regression is 0.245, meaning 24.5% of the error is explained by the model.

```r
set.seed(1)
train.hour <- sample(1:nrow(hour), nrow(hour) / 2)
tree.hour <- rpart(registered ~ ., data = hour, subset = train.hour)
rpart.plot(tree.hour)
summary(tree.hour)
```

Performing a decision tree analysis and getting the R^2 value.

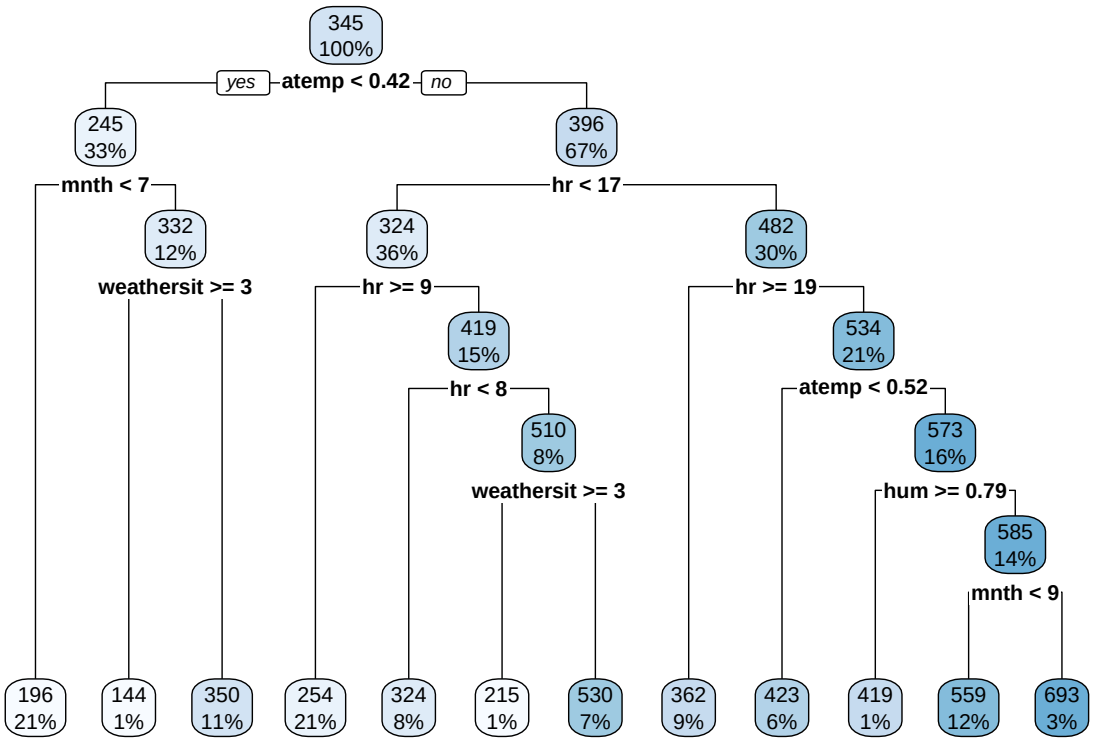
```
## Call:
## rpart(formula = registered ~ ., data = hour, subset = train.hour)
##   n= 1741
##
##            CP nsplit rel error    xerror       xstd
## 1  0.14583146      0 1.0000000 1.0021371 0.03169014
## 2  0.11962735      1 0.8541685 0.8578508 0.02798597
## 3  0.06969803      2 0.7345412 0.7393989 0.02598584
## 4  0.05434260      3 0.6648432 0.6940150 0.02432468
## 5  0.04072062      4 0.6105006 0.6268340 0.02242085
## 6  0.03819510      5 0.5697799 0.5923548 0.02105693
## 7  0.02624955      6 0.5315848 0.5597440 0.01975301
## 8  0.01390194      7 0.5053353 0.5209471 0.01830221
## 9  0.01161917      8 0.4914333 0.4968395 0.01757528
## 10 0.01012186      9 0.4798142 0.4834702 0.01709233
## 11 0.01000000     11 0.4595705 0.4726075 0.01657184
##
## Variable importance
##         hr      atemp       mnth        hum weathersit  windspeed
##         41         26         18          8          4          3
##
## Node number 1: 1741 observations,    complexity param=0.1458315
##   mean=345.3607, MSE=34602.48
##   left son=2 (579 obs) right son=3 (1162 obs)
##   Primary splits:
##       atemp      < 0.41665 to the left,  improve=0.14583150, (0 missing)
##       mnth       < 3.5     to the left,  improve=0.13505750, (0 missing)
##       hr         < 16.5    to the left,  improve=0.10694110, (0 missing)
##       hum        < 0.855   to the right, improve=0.06932247, (0 missing)
##       weathersit < 2.5     to the right, improve=0.06620006, (0 missing)
##   Surrogate splits:
##       mnth       < 3.5     to the left,  agree=0.801, adj=0.402, (0 split)
##       weathersit < 3.5     to the right, agree=0.669, adj=0.003, (0 split)
##       hum        < 0.08    to the left,  agree=0.669, adj=0.003, (0 split)
##       windspeed  < 0.597   to the right, agree=0.668, adj=0.002, (0 split)
##
## Node number 2: 579 observations,    complexity param=0.04072062
##   mean=244.7271, MSE=22746.04
##   left son=4 (372 obs) right son=5 (207 obs)
##   Primary splits:
##       mnth       < 6.5     to the left,  improve=0.18626700, (0 missing)
##       weathersit < 2.5     to the right, improve=0.07038976, (0 missing)
##       atemp      < 0.2803  to the left,  improve=0.06792898, (0 missing)
##       hum        < 0.84    to the right, improve=0.05005623, (0 missing)
```

```
##       hr         < 8.5     to the right, improve=0.04197214, (0 missing)
##
## Node number 3: 1162 observations,    complexity param=0.1196273
##   mean=395.5043, MSE=32949.78
##   left son=6 (635 obs) right son=7 (527 obs)
##   Primary splits:
##       hr         < 16.5    to the left,  improve=0.18822520, (0 missing)
##       hum        < 0.865   to the right, improve=0.07226675, (0 missing)
##       weathersit < 2.5     to the right, improve=0.05892074, (0 missing)
##       atemp      < 0.58335 to the left,  improve=0.04851446, (0 missing)
##       mnth       < 10.5    to the right, improve=0.01398040, (0 missing)
##   Surrogate splits:
##       hum        < 0.575   to the right, agree=0.629, adj=0.182, (0 split)
##       windspeed  < 0.20895 to the left,  agree=0.585, adj=0.085, (0 split)
##       atemp      < 0.70455 to the left,  agree=0.571, adj=0.053, (0 split)
##       mnth       < 4.5     to the right, agree=0.565, adj=0.042, (0 split)
##       weathersit < 1.5     to the right, agree=0.553, adj=0.015, (0 split)
##
## Node number 4: 372 observations
##   mean=196.172, MSE=13675.53
##
## Node number 5: 207 observations,    complexity param=0.01161917
##   mean=331.9855, MSE=27195.8
##   left son=10 (18 obs) right son=11 (189 obs)
##   Primary splits:
##       weathersit < 2.5     to the right, improve=0.12433940, (0 missing)
##       mnth       < 11.5    to the right, improve=0.10471060, (0 missing)
##       hum        < 0.84    to the right, improve=0.07113894, (0 missing)
##       hr         < 8.5     to the right, improve=0.04931323, (0 missing)
##       atemp      < 0.32575 to the left,  improve=0.02578938, (0 missing)
##
## Node number 6: 635 observations,    complexity param=0.06969803
##   mean=323.7606, MSE=23018.43
##   left son=12 (367 obs) right son=13 (268 obs)
##   Primary splits:
##       hr         < 8.5     to the right, improve=0.28726120, (0 missing)
##       weathersit < 2.5     to the right, improve=0.09625179, (0 missing)
##       hum        < 0.895   to the right, improve=0.03723952, (0 missing)
##       windspeed  < 0.20895 to the right, improve=0.03077524, (0 missing)
##       mnth       < 4.5     to the left,  improve=0.02816687, (0 missing)
##   Surrogate splits:
##       hum       < 0.695   to the left,  agree=0.688, adj=0.261, (0 split)
##       windspeed < 0.1194  to the right, agree=0.613, adj=0.082, (0 split)
##       atemp     < 0.5985  to the right, agree=0.583, adj=0.011, (0 split)
##
## Node number 7: 527 observations,    complexity param=0.0543426
##   mean=481.9507, MSE=31241.44
##   left son=14 (159 obs) right son=15 (368 obs)
##   Primary splits:
##       hr         < 18.5    to the right, improve=0.19884040, (0 missing)
##       atemp      < 0.58335 to the left,  improve=0.11959680, (0 missing)
##       hum        < 0.785   to the right, improve=0.10783520, (0 missing)
##       weathersit < 1.5     to the right, improve=0.05547816, (0 missing)
##       mnth       < 10.5    to the right, improve=0.03463376, (0 missing)
```

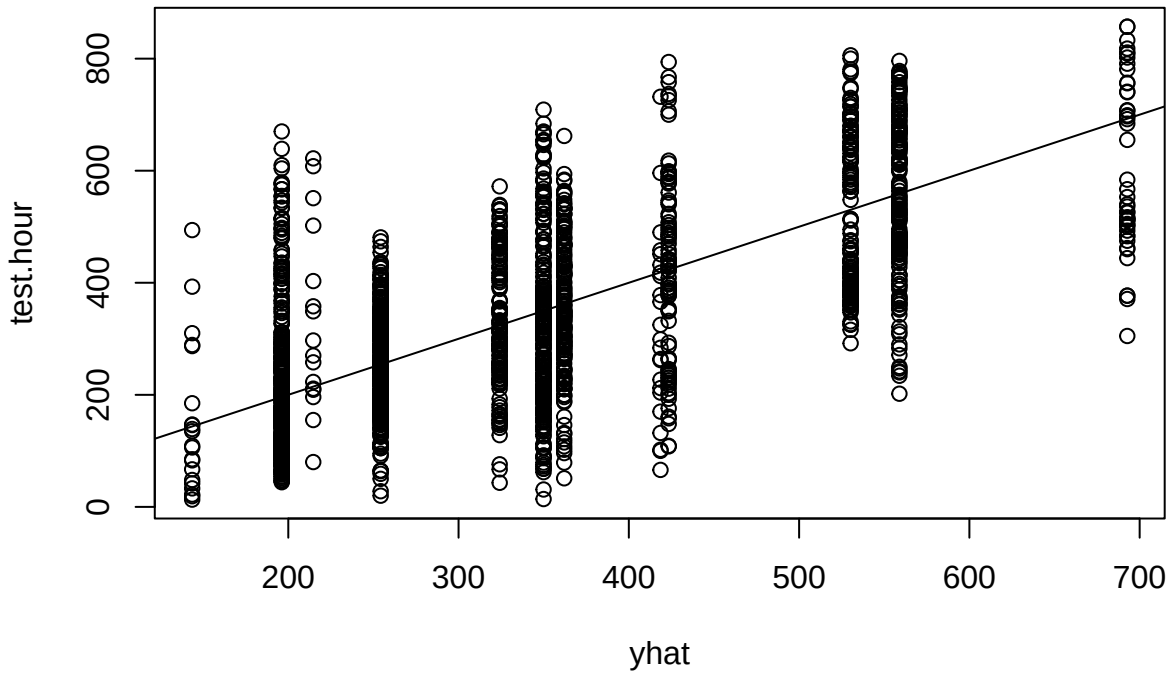
```
##   Surrogate splits:
##       windspeed < 0.55225 to the right, agree=0.7, adj=0.006, (0 split)
##
## Node number 10: 18 observations
##   mean=143.5556, MSE=22109.91
##
## Node number 11: 189 observations
##   mean=349.9312, MSE=23976.61
##
## Node number 12: 367 observations
##   mean=254.2725, MSE=6981.773
##
## Node number 13: 268 observations,    complexity param=0.0381951
##   mean=418.9179, MSE=29311.87
##   left son=26 (131 obs) right son=27 (137 obs)
##   Primary splits:
##       hr         < 7.5     to the left,  improve=0.29291070, (0 missing)
##       weathersit < 2.5     to the right, improve=0.17266530, (0 missing)
##       hum        < 0.865   to the right, improve=0.13538790, (0 missing)
##       mnth       < 4.5     to the left,  improve=0.02985074, (0 missing)
##       atemp      < 0.5682  to the left,  improve=0.02738988, (0 missing)
##   Surrogate splits:
##       hum        < 0.775   to the right, agree=0.582, adj=0.145, (0 split)
##       weathersit < 2.5     to the right, agree=0.537, adj=0.053, (0 split)
##       windspeed  < 0.1194  to the left,  agree=0.537, adj=0.053, (0 split)
##       atemp      < 0.5985  to the left,  agree=0.522, adj=0.023, (0 split)
##       mnth       < 9.5     to the left,  agree=0.519, adj=0.015, (0 split)
##
## Node number 14: 159 observations
##   mean=362.044, MSE=14444.46
##
## Node number 15: 368 observations,    complexity param=0.02624955
##   mean=533.7582, MSE=29602.75
##   left son=30 (96 obs) right son=31 (272 obs)
##   Primary splits:
##       atemp      < 0.52275 to the left,  improve=0.14516030, (0 missing)
##       hum        < 0.825   to the right, improve=0.11708480, (0 missing)
##       weathersit < 1.5     to the right, improve=0.07862619, (0 missing)
##       mnth       < 10.5    to the right, improve=0.05595736, (0 missing)
##       windspeed  < 0.29105 to the right, improve=0.02992277, (0 missing)
##   Surrogate splits:
##       mnth < 10.5    to the right, agree=0.815, adj=0.292, (0 split)
##       hum  < 0.915   to the right, agree=0.761, adj=0.083, (0 split)
##
## Node number 26: 131 observations
##   mean=324.1603, MSE=14276.71
##
## Node number 27: 137 observations,    complexity param=0.01390194
##   mean=509.5255, MSE=26893.05
##   left son=54 (9 obs) right son=55 (128 obs)
##   Primary splits:
##       weathersit < 2.5     to the right, improve=0.22731120, (0 missing)
##       hum        < 0.865   to the right, improve=0.09873164, (0 missing)
##       mnth       < 4.5     to the left,  improve=0.04150213, (0 missing)
```

```
##       atemp      < 0.5682  to the left,  improve=0.02199428, (0 missing)
##       windspeed  < 0.1194  to the right, improve=0.01143980, (0 missing)
##
## Node number 30: 96 observations
##   mean=423.4167, MSE=24868.26
##
## Node number 31: 272 observations,    complexity param=0.01012186
##   mean=572.7022, MSE=25459.97
##   left son=62 (20 obs) right son=63 (252 obs)
##   Primary splits:
##       hum        < 0.785   to the right, improve=0.07402686, (0 missing)
##       weathersit < 2.5     to the right, improve=0.06684452, (0 missing)
##       mnth       < 7.5     to the left,  improve=0.06438072, (0 missing)
##       windspeed  < 0.29105 to the right, improve=0.04835190, (0 missing)
##       atemp      < 0.61365 to the left,  improve=0.03010371, (0 missing)
##
## Node number 54: 9 observations
##   mean=214.6667, MSE=4658.444
##
## Node number 55: 128 observations
##   mean=530.2578, MSE=21913.5
##
## Node number 62: 20 observations
##   mean=418.6, MSE=20223.94
##
## Node number 63: 252 observations,    complexity param=0.01012186
##   mean=584.9325, MSE=23841.22
##   left son=126 (203 obs) right son=127 (49 obs)
##   Primary splits:
##       mnth       < 8.5     to the left,  improve=0.11765930, (0 missing)
##       windspeed  < 0.29105 to the right, improve=0.07215572, (0 missing)
##       atemp      < 0.7803  to the right, improve=0.04465834, (0 missing)
##       weathersit < 2.5     to the right, improve=0.02892134, (0 missing)
##       hum        < 0.525   to the right, improve=0.02487777, (0 missing)
##   Surrogate splits:
##       atemp < 0.58335 to the right, agree=0.821, adj=0.082, (0 split)
##
## Node number 126: 203 observations
##   mean=558.9113, MSE=20899.96
##
## Node number 127: 49 observations
##   mean=692.7347, MSE=21599.99
```

```
best_cp <- tree.hour$cptable %>%
  as_tibble() %>%
  filter(xerror == min(xerror)) %>%
  head(1) %>%
  pull(CP) # note the best CP is 0.01, which corresponds with the most (11) splits
prune.hour <- prune(tree.hour, cp = best_cp)
rpart.plot(prune.hour)
```




```
yhat <- predict(tree.hour, newdata = hour[-train.hour, ])
test.hour <- hour[-train.hour, "registered"]
test.hour <- as.integer(unlist(test.hour))
plot(yhat, test.hour)
abline(0, 1)
```



```r
mean((yhat - test.hour)^2)
```

```
## [1] 17728.39
```

```r
tmp <- printcp(tree.hour)
```

```
##
## Regression tree:
## rpart(formula = registered ~ ., data = hour, subset = train.hour)
##
## Variables actually used in tree construction:
## [1] atemp      hr         hum        mnth       weathersit
##
## Root node error: 60242915/1741 = 34602
##
## n= 1741
##
##          CP nsplit rel error  xerror     xstd
## 1  0.145831      0   1.00000 1.00214 0.031690
## 2  0.119627      1   0.85417 0.85785 0.027986
## 3  0.069698      2   0.73454 0.73940 0.025986
## 4  0.054343      3   0.66484 0.69402 0.024325
## 5  0.040721      4   0.61050 0.62683 0.022421
## 6  0.038195      5   0.56978 0.59235 0.021057
## 7  0.026250      6   0.53158 0.55974 0.019753
## 8  0.013902      7   0.50534 0.52095 0.018302
## 9  0.011619      8   0.49143 0.49684 0.017575
## 10 0.010122      9   0.47981 0.48347 0.017092
## 11 0.010000     11   0.45957 0.47261 0.016572
```

```r
tree.hour.r2.table <- 1-tmp[,c(3,4)]
tree.hour.r2 <- tree.hour.r2.table[11,1]
tree.hour.r2
```

```
## [1] 0.5404295
```

The R^2 value is 0.540, meaning 54.0% of the error is explained by the model.

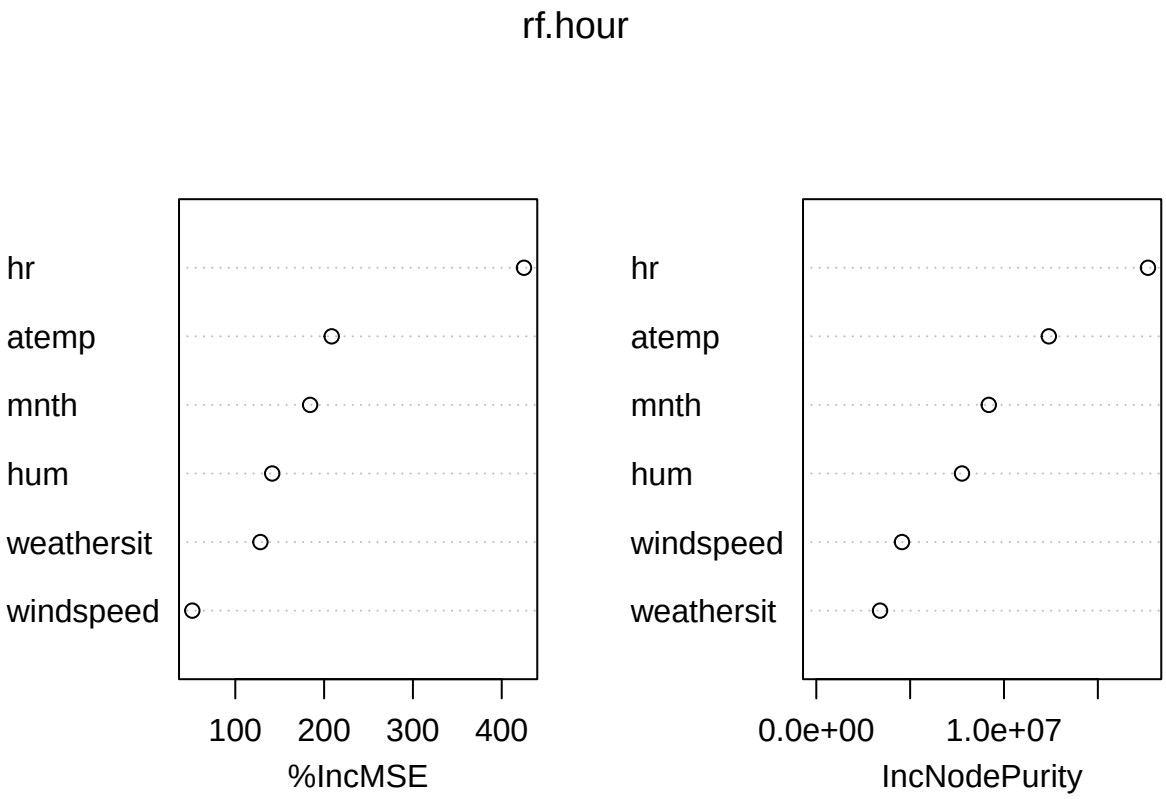
```r
set.seed(2)

rf.hour <- randomForest(
  registered ~ .,
  data = hour,
  subset = train.hour,
  # mtry = 5,
  importance = TRUE,
  ntree = 5000
```

```
)

# predict.rf.hour <- predict(rf.hour, test_data, type = "class")
###
importance(rf.hour)
```

Performing a random forest and getting the R^2 value.

```
##             %IncMSE IncNodePurity
## mnth       184.2348       9189447
## hr         425.0930      17671592
## atemp      208.5309      12387794
## weathersit 128.2614       3396040
## hum        141.5659       7761094
## windspeed   51.5651       4564993
```

```
###
varImpPlot(rf.hour)
```




```
rf.hour.rsq.list  <- rf.hour$rsq
rf.hour.rsq <- rf.hour.rsq.list[5000]
rf.hour.rsq
```

```
## [1] 0.6583721
```

The R^2 value is 0.658, meaning 65.8% of the error is explained by the model.

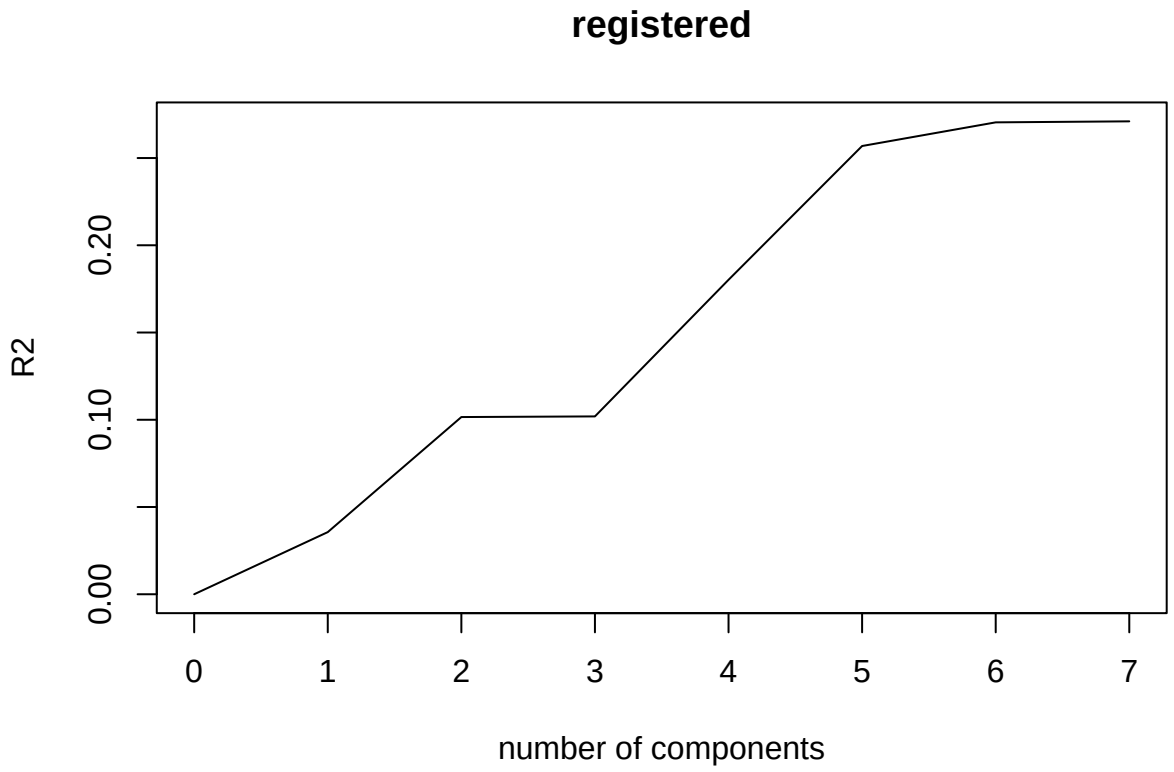
Methods group 2: PCR, ridge and lasso regression.

```
set.seed(1)
regression_model=bs_hour%>%filter(hr %in% c(7,8,9,16,17,18,19))%>%filter(workingday==1)
regression_model=regression_model[,-which(names(regression_model) == "dteday")]
regression_model=regression_model[,-which(names(regression_model) == "season")]
regression_model=regression_model[,-which(names(regression_model) == "yr")]
regression_model=regression_model[,-which(names(regression_model) == "holiday")]
regression_model=regression_model[,-which(names(regression_model) == "workingday")]
regression_model=regression_model[,-which(names(regression_model) == "temp")]
regression_model=regression_model[,-which(names(regression_model) == "casual")]
regression_model=regression_model[,-which(names(regression_model) == "hourly_id")]
regression_model=regression_model[,-which(names(regression_model) == "cnt")]

indice_train = sample(seq_len(nrow(regression_model)),size = 0.2*nrow(regression_model))
indice_test =  seq_len(nrow(regression_model)) %>%
setdiff(indice_train)
```

data processing

```
# fit pcr on training set
pcr = pcr(registered ~ ., data = regression_model,subset = indice_train)
y_prediction_pcr = predict(pcr, regression_model[indice_test, ])

## residual sum of squares
validationplot(pcr, val.type = "R2")
```

Performing a Principal Component Regression and getting the R^2 value



We fit the principal component regression model with all possible values of components on the training dataset ; in order to find the number of components that maximize the value of R^2. We found that if we have a number of components egals to 7 which means we keep all our components, we have an optimal r-squared value. For the PCR regression model, the r-squared value is around 0.25 which means that the model describes around 25% of the variance.

```
pred_error_all = sapply(
  1:6,
  function(i) mean((y_prediction_pcr[, , i] - regression_model[indice_test ,]$registered)^2))
# find which M minimizes prediction error
pcr$coefficients[, , which.min(pred_error_all)]
```

```
##       mnth         hr    weekday weathersit      atemp        hum  windspeed
##  11.117908   2.169323  -4.078933 -55.552636 291.650947 -92.003416  -9.571441
```

For this model, the variables atemp, month and hour have the highest coefficient values.

```
# cross validation based tuning of lambda
#cv glmnet implement already 10 cross validation
ridge = cv.glmnet(
registered ~ .,
data = regression_model,
alpha = 0
)
# ridge with optimal lambda
```

```
ridgefit = glmnet(
registered ~ .,
data = regression_model,
alpha = 0,
lambda = ridge$lambda.1se # optimal lambda
)

y_pred_ridge = predict(ridgefit, newdata = regression_model)

sst = sum((regression_model$registered - mean(regression_model$registered))^2)
sse =sum((y_pred_ridge - regression_model$registered)^2)

r2 = 1 - (sse/sst)
r2
```

Performing a ridge regression and getting the R^2 value

```
## [1] 0.2284835
```

We fit the ridge model on the training data with an optimally tuned ridge shrinkage parameter lambda. We also used the rstudio function cv.glmet which use cross validation to perform the tuning. The R^2 value is 0.22, meaning 22.0% of the error is explained by the model.

```
coef(ridgefit)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) 267.176504
## mnth          6.728842
## hr            1.299012
## weekday      -3.017109
## weathersit  -36.042493
## atemp       253.278138
## hum         -59.112727
## windspeed   -52.727988
```

The ridge model use all the predictors ; the variable atemp has the highest coefficient value.

```
# cross validation based tuning of lambda
lasso=  cv.glmnet(
registered ~ .,
data = regression_model,
alpha = 1
)
# lasso with optimal lambda
lassofit = glmnet(
registered ~ .,
data = regression_model,
alpha = 1,
```

```
lambda = lasso$lambda.1se # optimal lambda
)
# predict on test set
pred_lasso = predict(lassofit, newdata = regression_model)


sst = sum((regression_model$registered - mean(regression_model$registered))^2)
sse =sum((pred_lasso - regression_model$registered)^2)

r2 = 1 - (sse/sst)
r2
```

Performing a lasso regression and getting the R^2 value

```
## [1] 0.2269
```

we fit the ridge model on the training data with an optimally tuned ridge shrinkage parameter lambda. We also used the rstudio function cv.glmet which use cross validation to perform the tuning. The R^2 value is 0.22, meaning 22.0% of the error is explained by the model.

```
coef(lassofit)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) 216.970587
## mnth          5.173980
## hr            .
## weekday       .
## weathersit  -39.482607
## atemp       317.596204
## hum          -7.359954
## windspeed     .
```

The lasso model does not use the variable hour, weekday and windspeed. the variable atemp has the highest coefficient value.

Attempting a categorical subset of the hours. We will try to approach a way to make linear models more effective by classifying the hours better. It is clear based on previous variable importance tests that "hours" is a very important variable, but based on how it is coded (incrementing one for every subsequent hour) instead of grouped in a more reasonable way.

Regrouping hour such that:

7AM to 9AM and 4PM to 7PM is rush hour, coded as 3. 10 AM to 3PM is midday, coded as 2. 5 AM to 6 AM and 8PM to 11PM is odd hours, coded as 1. 12AM to 4 AM is night, coded as 0.

This is still only concerned with working days.

```
hour_orig <- read_csv('Bike-Sharing-Dataset/hour.csv',show_col_types = FALSE);

hour.filtered <- subset(hour_orig, workingday >.5)
hour.filtered1 <- hour.filtered %>% mutate(hr_type = case_when(hr < 4.5 ~ 0,
  hr < 6.5 ~ 1,
```

```
  hr < 9.5 ~ 3,
  hr < 15.5 ~ 2,
  hr < 19.5 ~ 3,
  hr < 23.5 ~ 1))

hour <- hour.filtered1[,c("mnth","hr_type","atemp","weathersit","hum","windspeed","registered")]
```

Now, to run a simple linear model.

```
hour.lm <- lm(registered ~ ., data = hour)
hour.lm.r2 <- summary(hour.lm)$r.squared
hour.lm.r2
```

```
## [1] 0.5593897
```

With SLR giving an R^2 value of 55.9, that's significantly better than the original model, though not quite as good as random forest, implying that the non-linearity of the data is still significant.

4. Conclusion

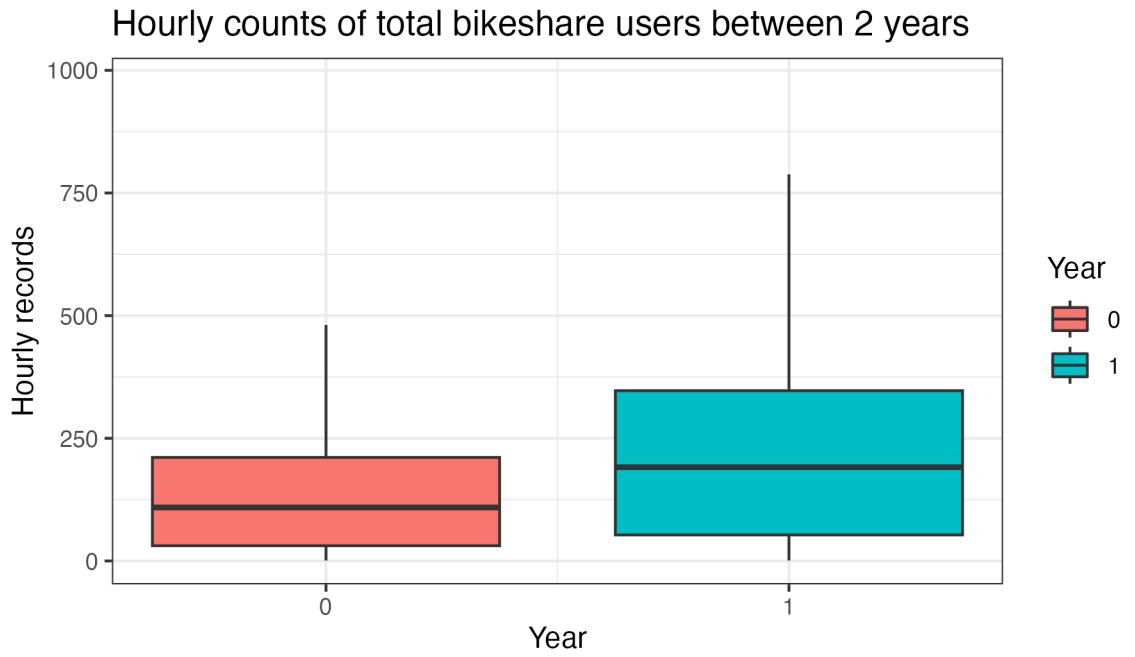
a. Which predictor affects dependent variable more?

Though it's shown to have little influence on the count of registered users in linear regression, Ridge, PCR and Lasso, "hr" which is the hour predictor actually explains much of the variation of dependent variable under a non-linear method like tree-based regression and random forest. Using "hr" as an factor to fit multiple parallel linear regression model with different intercept would be a solution for a better fit.

Not surprisingly, the feeling temperature does contribute a lot on likely commuters' choice, which is seen by all the linear models and tree-based methods. And followed are predictors like humidity, month and weather type. Wind speed shows little influence on dependent variable estimation in random forest modeling while it's been excluded in Lasso with "weekday"

b. Explanation to low variation explained.

In the regression part, our group used 7 predictors for a general model to explain which predictor affects the number of likely commuters, excluding time information like "yr" (year) and redundant information like temperature (because we included feeling temperature). However, an increasing of total bikeshare users could be seen from the first year to the second, as shown in next boxplots.



Thus, combined with the fact that relations between independent and dependent variables are quite non-linear, it's understandable that our regression modeling not performing well.

5. Author contributions

Hang:

Project report: structure building; data description, plots part;
Github repository creating;

Andrew:

SLR, Decision Tree, and Random Forest, as well as reclassifying the hour into new classes, and all the analysis therein.

Charlotte

PCR Analysis, Ridge, LASSO, and the analysis therein.