Project least squares

File

12/16/2020

β_1 = Season

β_2 = year

β_3 = month

β_4 = hour

β_5 = holiday

β_6 = weekday

##Dataset Info:

Attribute Information:

Both hour.csv and day.csv have the following fields, except hr which is not available in day.csv

- instant: record index
- dteday : date
- season : season (1:winter, 2:spring, 3:summer, 4:fall)
- yr : year (0: 2011, 1:2012)
- mnth : month (1 to 12)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (extracted from [Web Link])
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
- 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 + Fog
- temp : Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
- atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

#function here is ment to convert hour to days..so we are able to use this dataset instead of picking one over the other.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

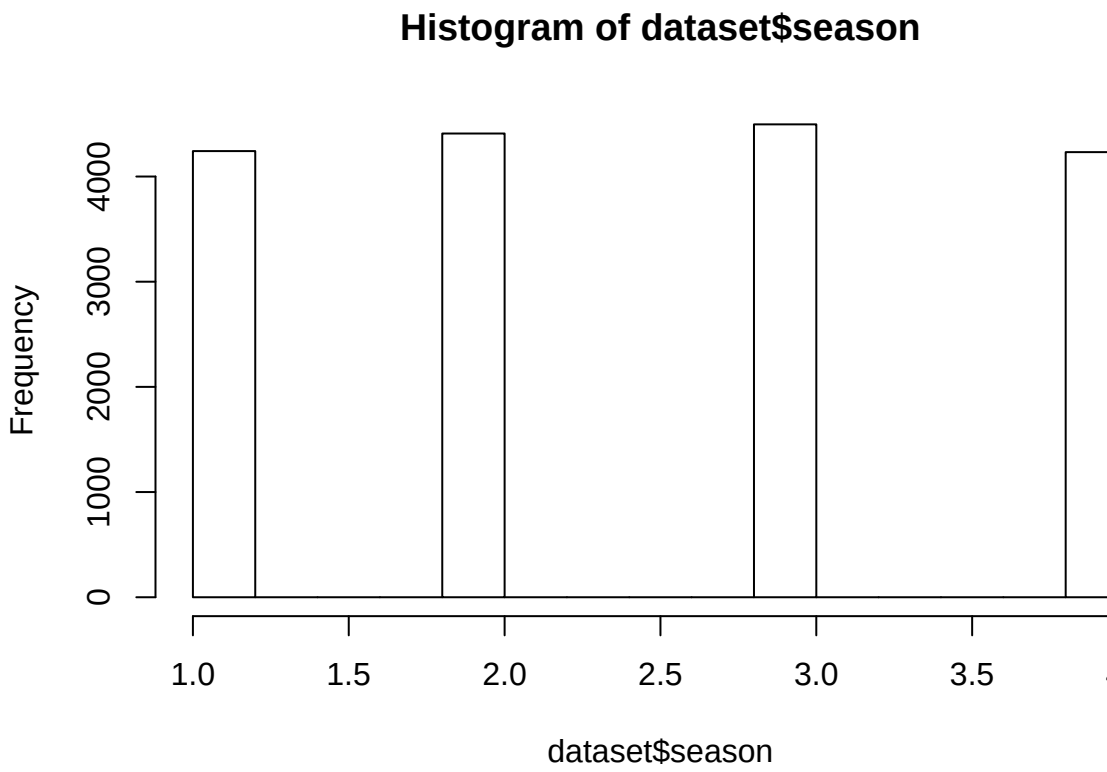
```r
dataset = read.csv("hour.csv") # dataset we are using
convert_hour_to_day <- function(hour){
  require(tidyverse)
  day = hour %>%
    group_by(dteday, season, yr, mnth, holiday, weekday, workingday) %>%
    summarize(weathersit = as.integer(round(mean(weathersit))),
              temp = mean(temp),
              atemp = mean(atemp),
              hum = mean(hum),
              windspeed = mean(windspeed),
              casual = sum(casual),
              registered = sum(registered),
              cnt = sum(cnt))
  #day = tibble::rowid_to_column(day, "instant")
  return(day)
}
```

```
##     instant             dteday          season           yr
##  Min.   :    1   2011-01-01:   24   Min.   :1.000   Min.   :0.0000
##  1st Qu.: 4346   2011-01-08:   24   1st Qu.:2.000   1st Qu.:0.0000
##  Median : 8690   2011-01-09:   24   Median :3.000   Median :1.0000
##  Mean   : 8690   2011-01-10:   24   Mean   :2.502   Mean   :0.5026
##  3rd Qu.:13034   2011-01-13:   24   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :17379   2011-01-15:   24   Max.   :4.000   Max.   :1.0000
##                  (Other)   :17235
##       mnth              hr           holiday           weekday
##  Min.   : 1.000   Min.   : 0.00   Min.   :0.00000   Min.   :0.000
##  1st Qu.: 4.000   1st Qu.: 6.00   1st Qu.:0.00000   1st Qu.:1.000
##  Median : 7.000   Median :12.00   Median :0.00000   Median :3.000
##  Mean   : 6.538   Mean   :11.55   Mean   :0.02877   Mean   :3.004
##  3rd Qu.:10.000   3rd Qu.:18.00   3rd Qu.:0.00000   3rd Qu.:5.000
##  Max.   :12.000   Max.   :23.00   Max.   :1.00000   Max.   :6.000
##
##    workingday       weathersit         temp           atemp
##  Min.   :0.0000   Min.   :1.000   Min.   :0.020   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.340   1st Qu.:0.3333
##  Median :1.0000   Median :1.000   Median :0.500   Median :0.4848
##  Mean   :0.6827   Mean   :1.425   Mean   :0.497   Mean   :0.4758
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.660   3rd Qu.:0.6212
```

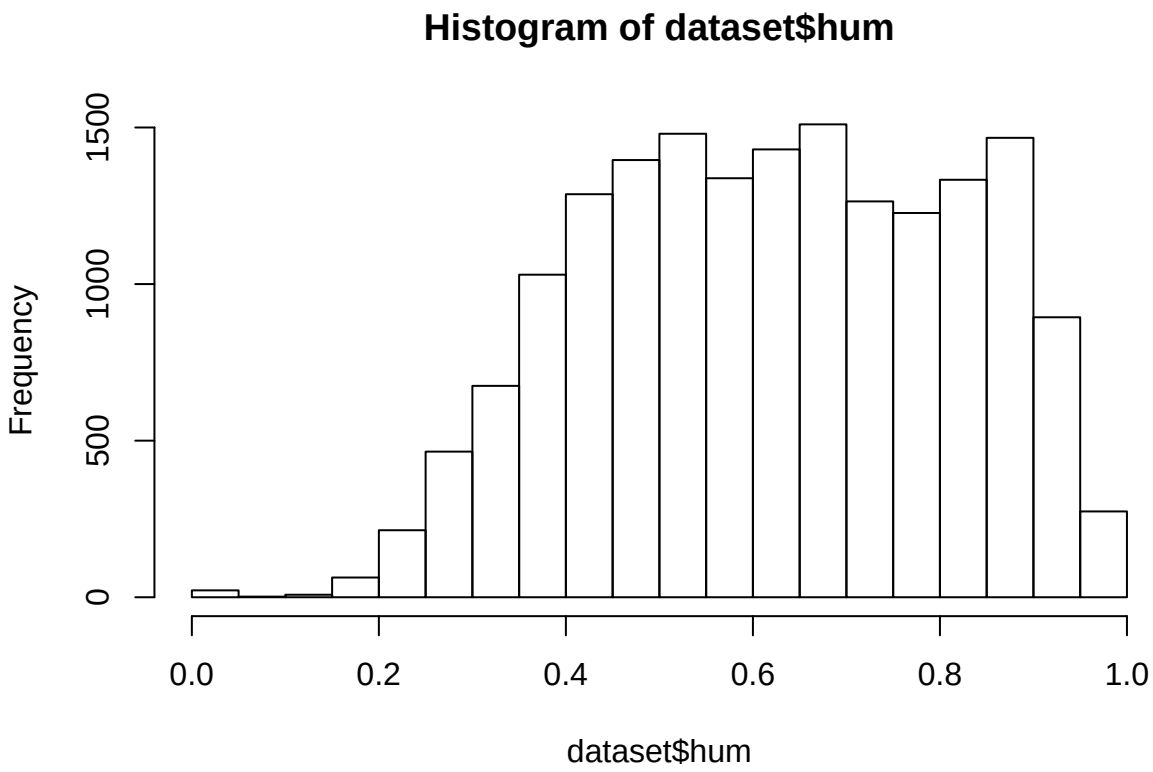
```
##  Max.   :1.0000   Max.   :4.000   Max.   :1.000   Max.   :1.0000
##
##       hum           windspeed          casual         registered
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 34.0
##  Median :0.6300   Median :0.1940   Median : 17.00   Median :115.0
##  Mean   :0.6272   Mean   :0.1901   Mean   : 35.68   Mean   :153.8
##  3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 48.00   3rd Qu.:220.0
##  Max.   :1.0000   Max.   :0.8507   Max.   :367.00   Max.   :886.0
##
##       cnt
##  Min.   :  1.0
##  1st Qu.: 40.0
##  Median :142.0
##  Mean   :189.5
##  3rd Qu.:281.0
##  Max.   :977.0
##
```

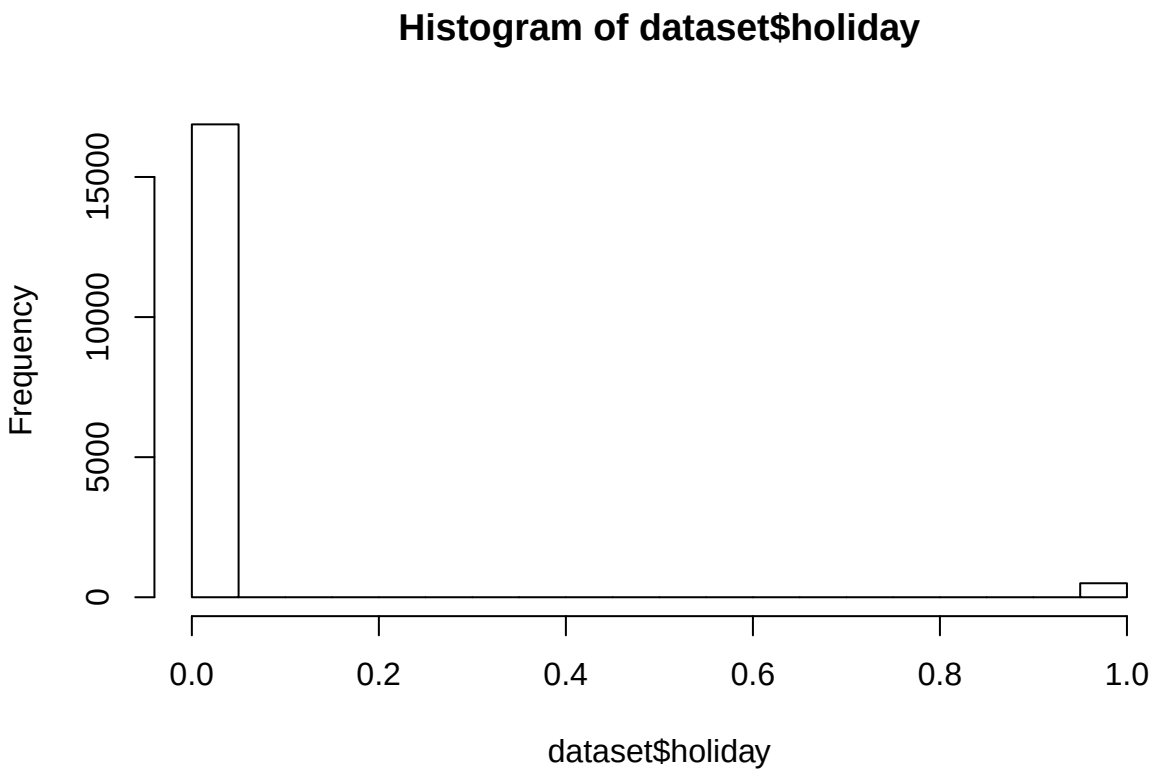
```
## 'data.frame':    17379 obs. of  17 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
##  $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
##  $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
##  $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
##  $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
##  $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

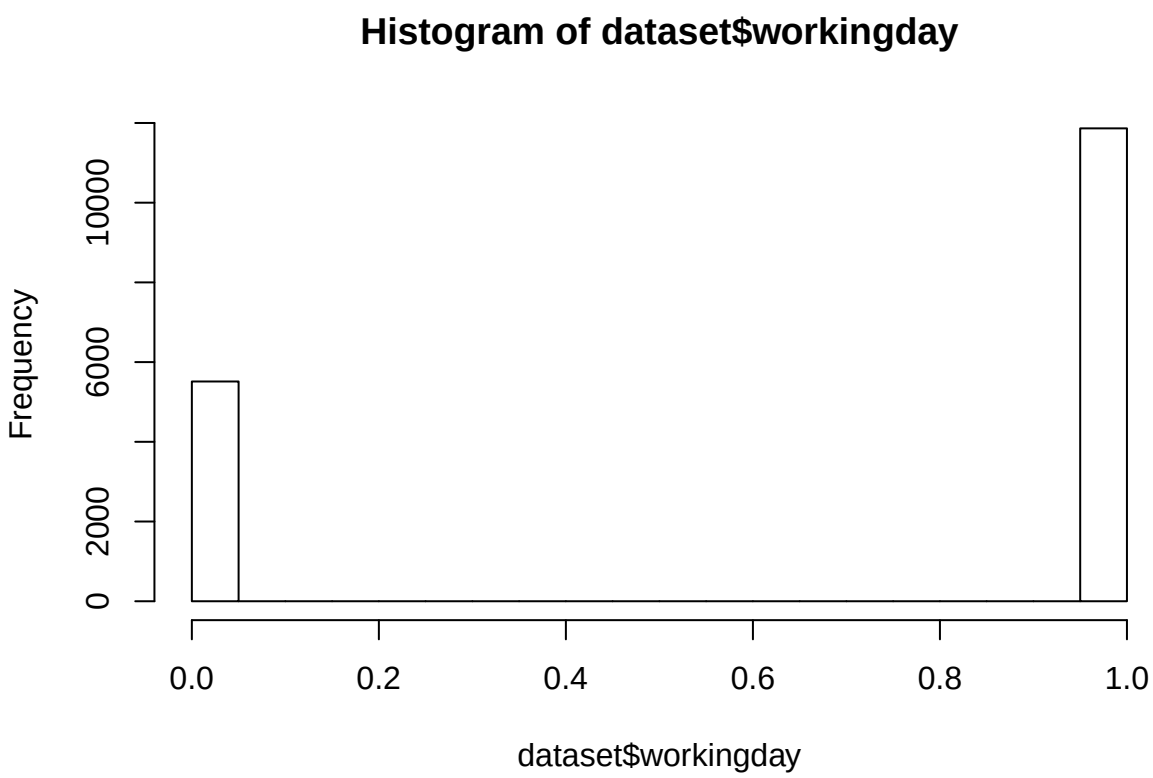
```
##
##  FALSE
## 295443
```

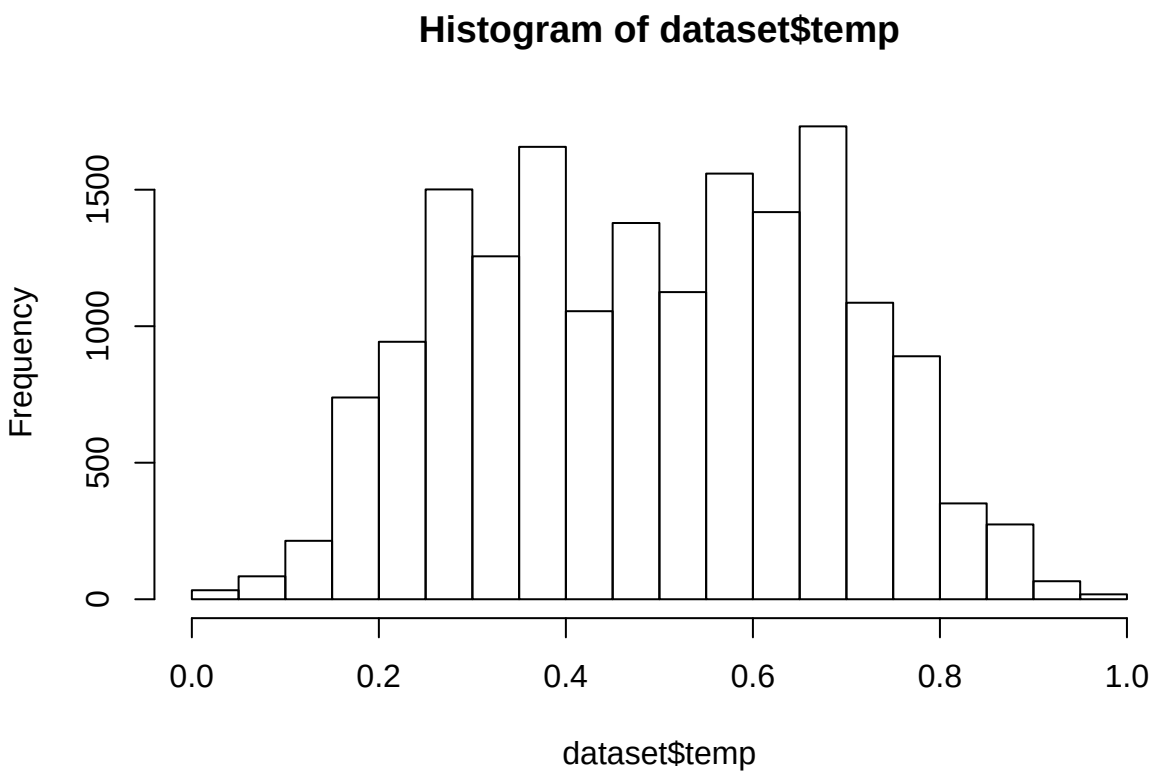


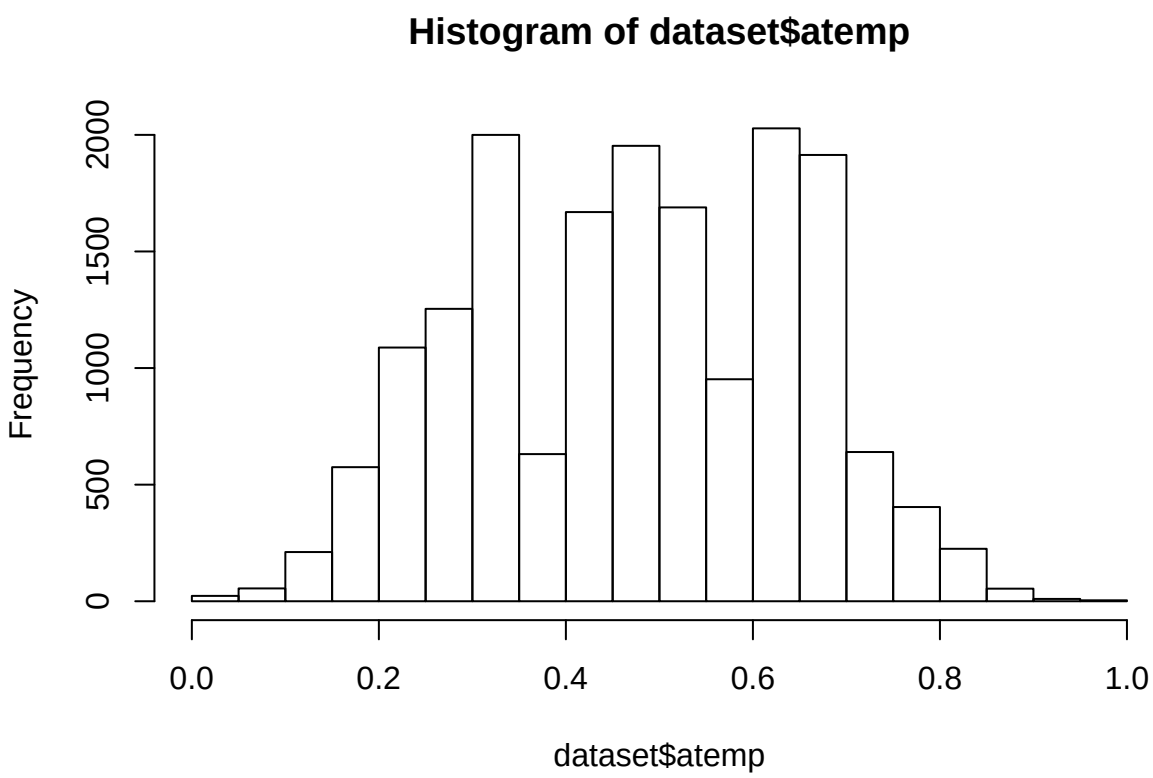
#Histrogram of the dataset:

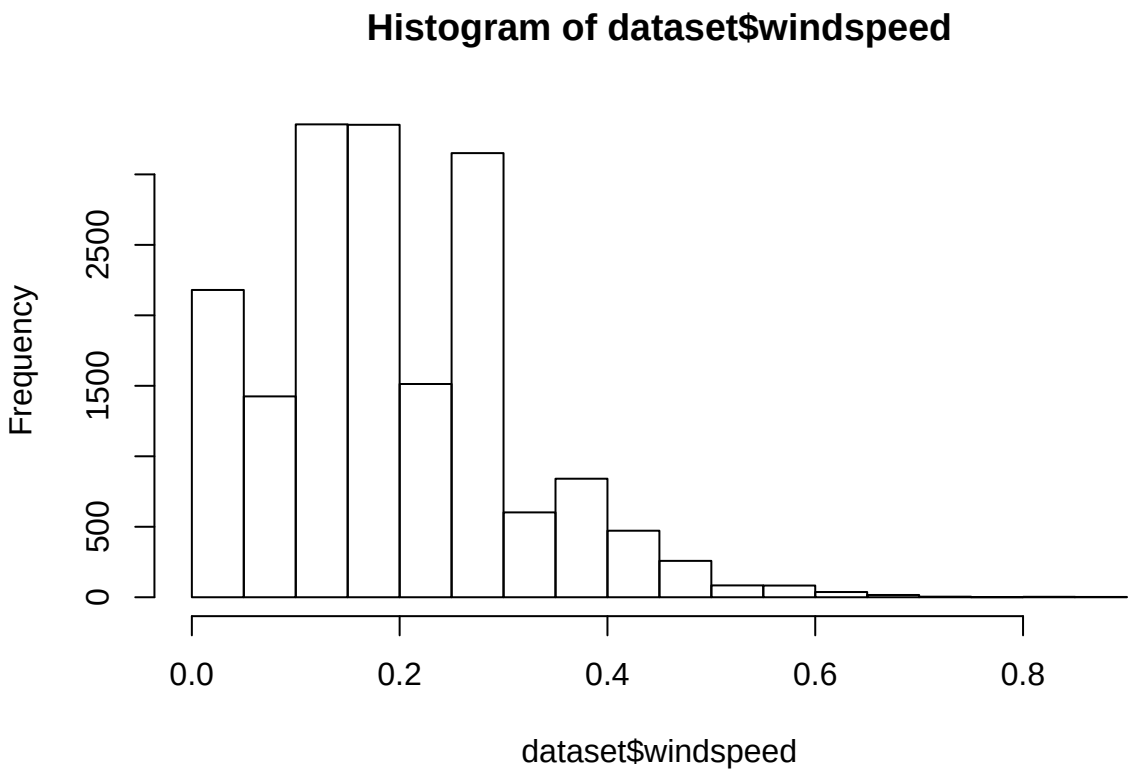

Histogram of dataset$hum
Frequency
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
1.0
dataset$hum


Histogram of dataset$holiday
Frequency
0
5000
10000
15000
0.0
0.2
0.4
0.6
0.8
1.0
dataset$holiday

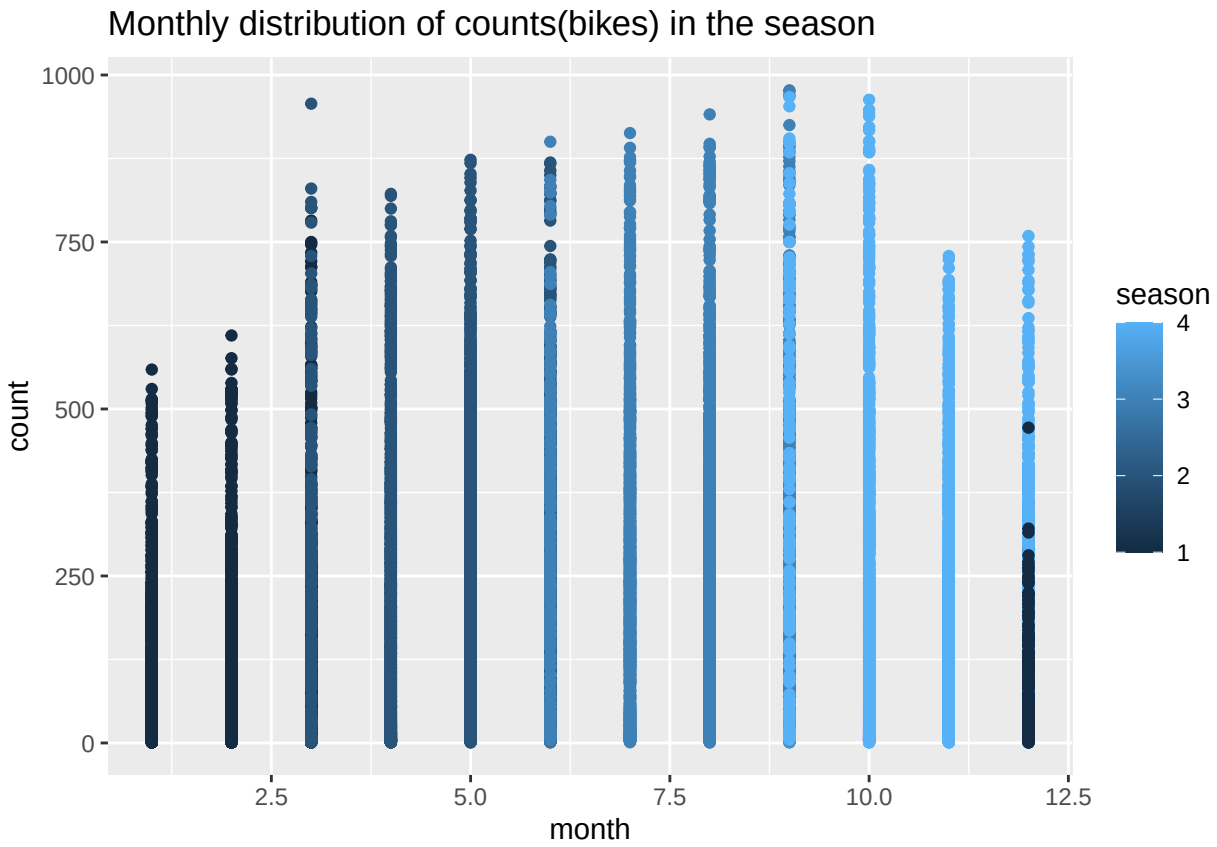

Histogram of dataset$workingday
Frequency
0
2000
6000
10000
0.0
0.2
0.4
0.6
0.8
1.0
dataset$workingday

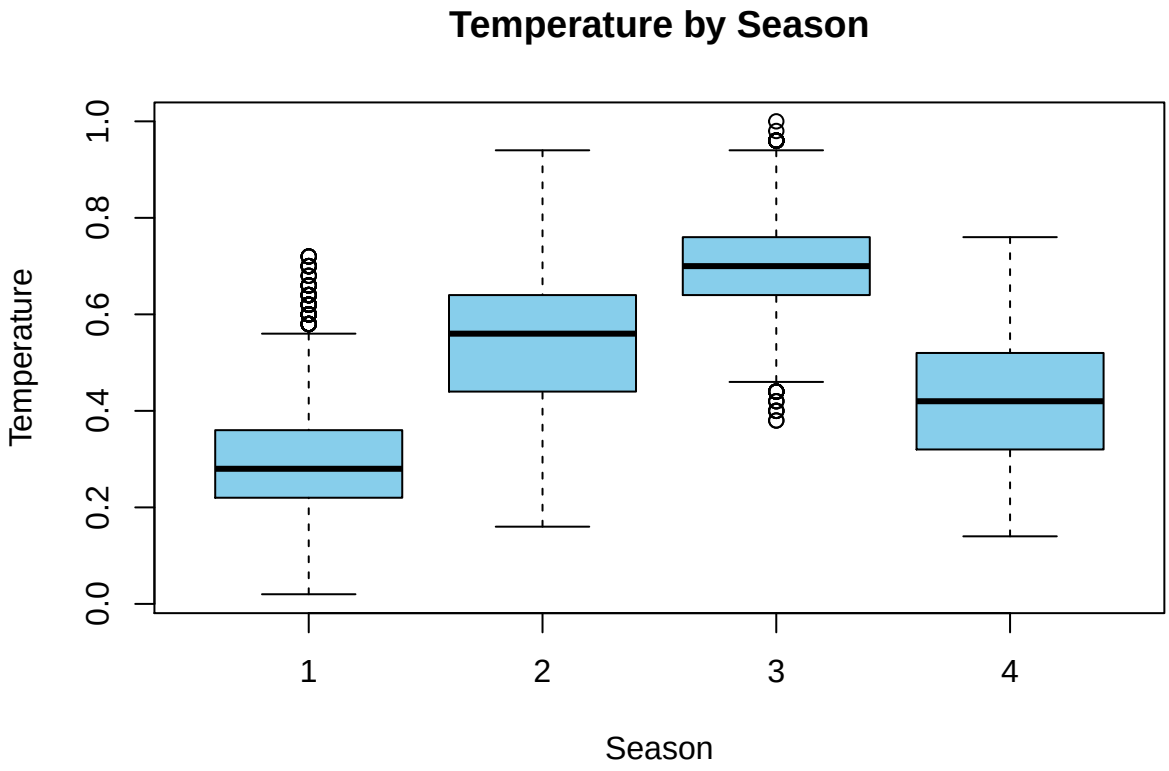

Histogram of dataset$temp
Frequency
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
1.0
dataset$temp


Histogram of dataset$atemp
Frequency
0
500
1000
1500
2000
0.0
0.2
0.4
0.6
0.8
1.0
dataset$atemp



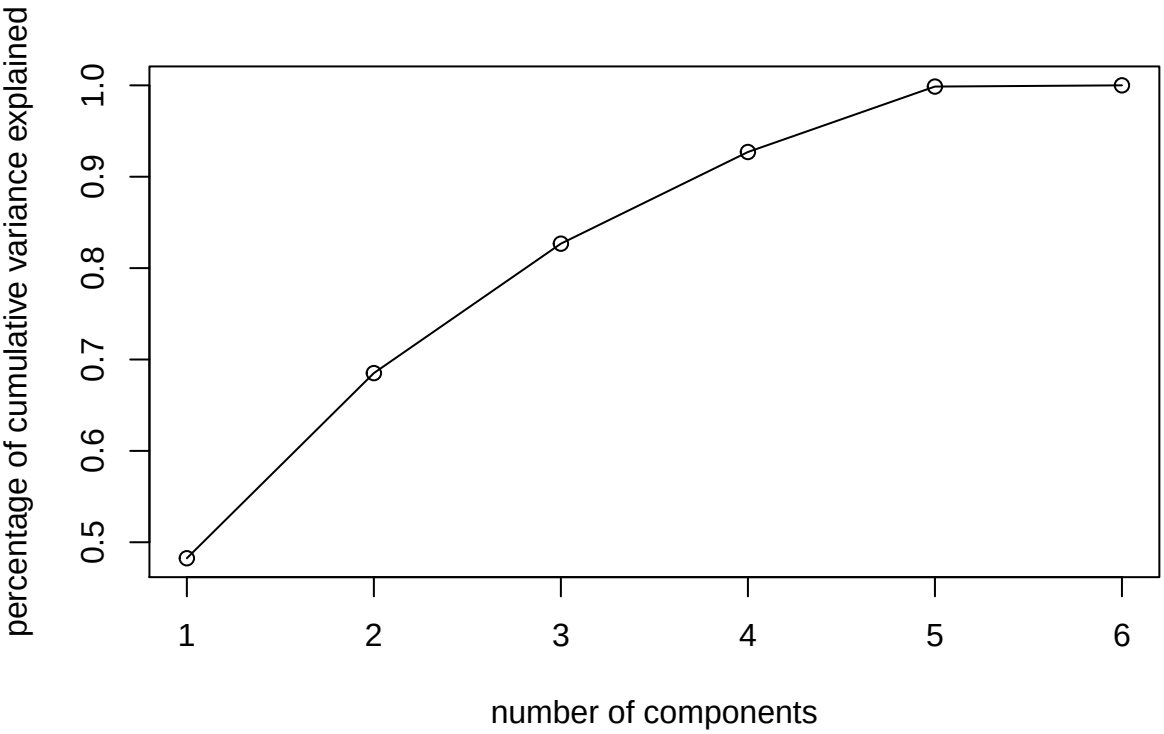

```
ggplot(dataset,
       aes(x = mnth, y = cnt, color = season)) + geom_point() +
  labs(x = "month", y = "count", title='Monthly distribution of counts(bikes) in the season')
```


Monthly distribution of counts(bikes) in the season
count
1000
750
500
250
0
2.5
5.0
7.5
10.0
12.5
month
season
4
3
2
1



#PCA:

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.7015 1.1025 0.9219 0.7758 0.65553 0.08902
## Proportion of Variance 0.4825 0.2026 0.1416 0.1003 0.07162 0.00132
## Cumulative Proportion  0.4825 0.6851 0.8267 0.9271 0.99868 1.00000
```

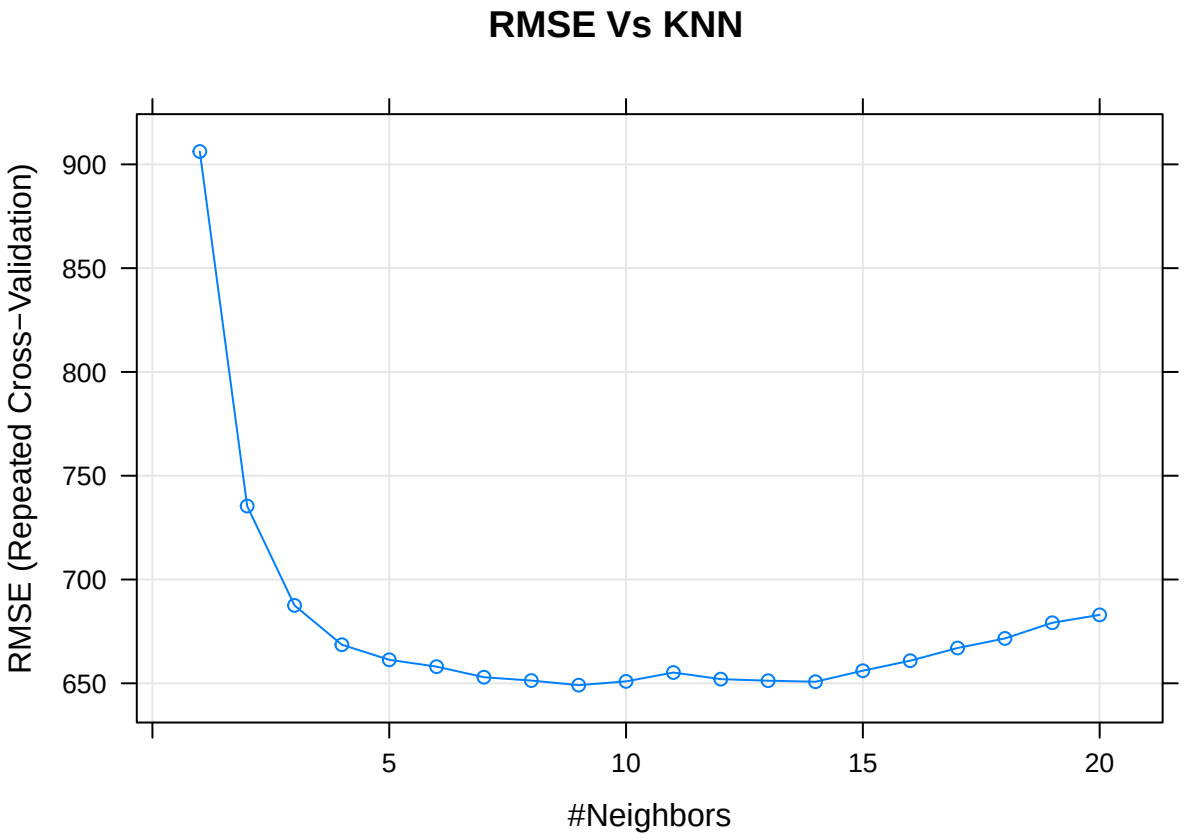


Since 3 components explain 82% of the variance, therefore, we can select 3 components and do the KNN with the three component based coordinates and other categorical variables. #KNN

```
## Loading required package: lattice
```

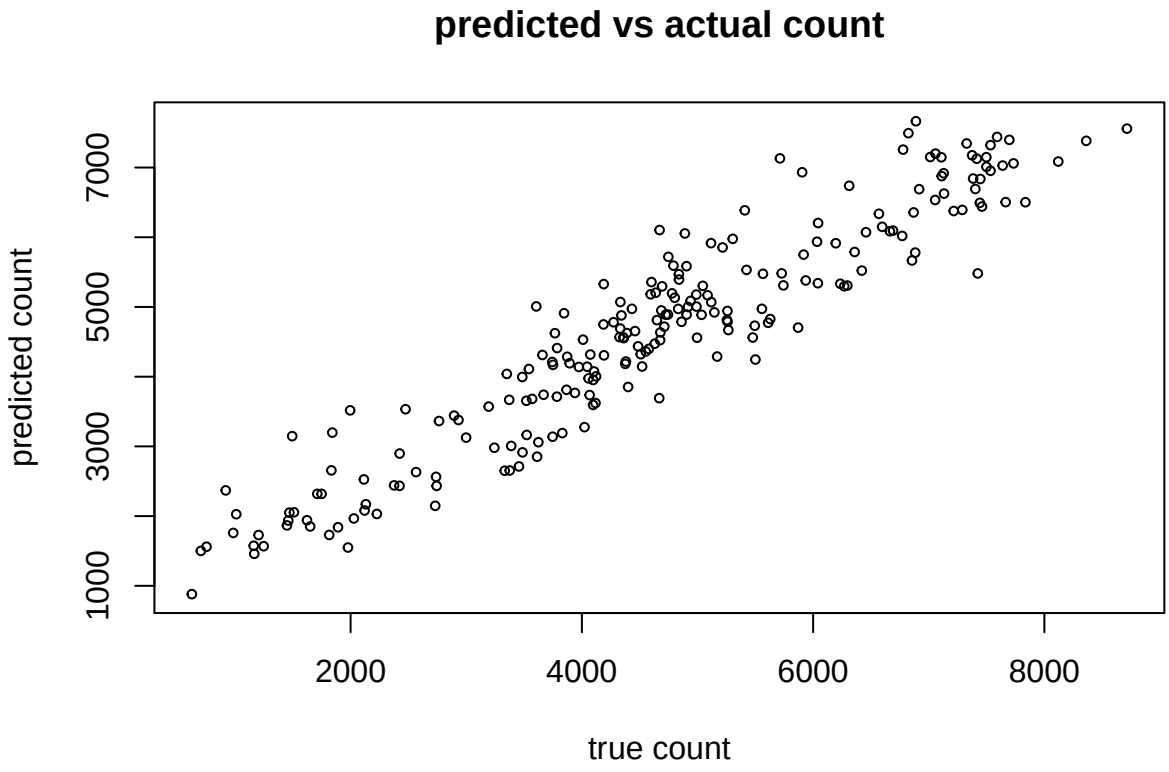
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```




```
## k-Nearest Neighbors
##
## 512 samples
##  10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 461, 460, 460, 462, 461, 460, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    1  906.1895  0.7995117  660.5843
##    2  735.3849  0.8631309  558.4770
##    3  687.5237  0.8809150  530.3115
##    4  668.5704  0.8901826  515.7483
##    5  661.3252  0.8937389  517.6089
##    6  658.0253  0.8951853  515.5019
##    7  652.9275  0.8974504  511.8914
##    8  651.3258  0.8983793  513.1548
##    9  649.1033  0.9002024  513.1542
##   10  650.9264  0.9004055  515.6410
##   11  655.1907  0.8997497  518.7398
##   12  652.0082  0.9018401  517.0262
##   13  651.2473  0.9031175  517.7716
##   14  650.7603  0.9044768  519.7946
##   15  656.0573  0.9037467  527.0363
```

```
##   16  660.8852  0.9025000  530.8019
##   17  666.9928  0.9016157  535.8224
##   18  671.6060  0.9009325  540.6077
##   19  679.1593  0.8991188  546.6749
##   20  682.9495  0.8989091  550.3399
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```




```
## [1] "The R-squared is  0.890038358816099"
```

Principal Componenet Analysis

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of a data set.

Importance of components: PC1 PC2 PC3 PC4 PC5 PC6 Standard deviation 1.7015 1.1025 0.9219 0.7758 0.65553 0.08902 Proportion of Variance 0.4825 0.2026 0.1416 0.1003 0.07162 0.00132 Cumulative Proportion 0.4825 0.6851 0.8267 0.9271 0.99868 1.00000