hw5

Asahi (Ash) Kuroki

12/1/2021

```
knitr::opts_chunk$set(cache = TRUE)
set.seed(123)
```

```
# import packages
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(caret)
```

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

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

```
## Loading required package: rpart
```

```
library(gridExtra)
library(labelVector)
library(tidyverse)
```

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

```
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
```

Analysis

Processing and Visualizing data
a) Load the data. Get a summary of the data, report it. Use ggplot to plot a histogram for the distribution of the number of bike-rides.

```
df <- read.csv("bike_day.csv")
summary(df)
```

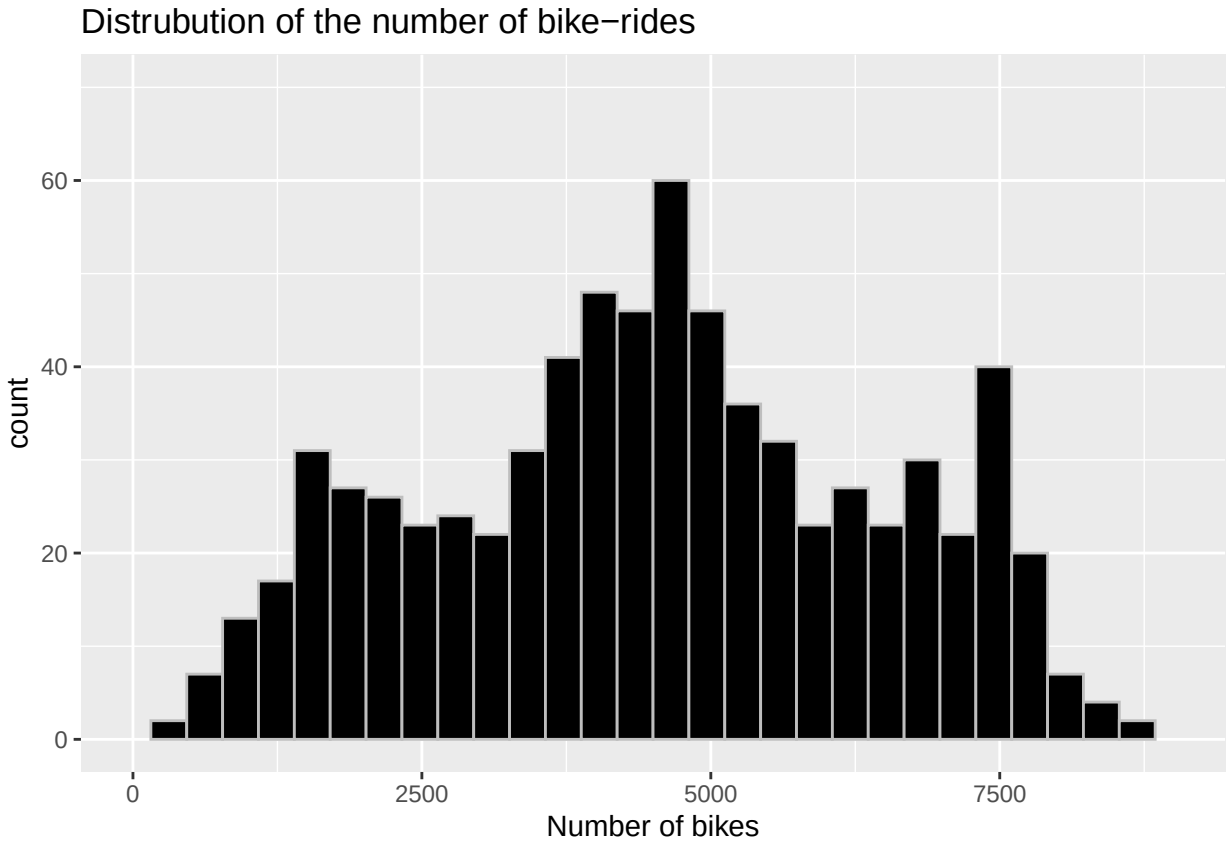
```
##     cnt_bike        atemp            hum          windspeed          temp
##  Min.   :  22   Min.   : 3.95   Min.   : 0.00   Min.   : 1.50   Min.   : 2.42
```

```
##  1st Qu.:3152   1st Qu.:16.89   1st Qu.:52.00   1st Qu.: 9.04   1st Qu.:13.82
##  Median :4548   Median :24.34   Median :62.67   Median :12.13   Median :20.43
##  Mean   :4504   Mean   :23.72   Mean   :62.79   Mean   :12.76   Mean   :20.31
##  3rd Qu.:5956   3rd Qu.:30.43   3rd Qu.:73.02   3rd Qu.:15.62   3rd Qu.:26.88
##  Max.   :8714   Max.   :42.04   Max.   :97.25   Max.   :34.00   Max.   :35.33
##     holiday          workingday
##  Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.00000   Median :1.000
##  Mean   :0.02873   Mean   :0.684
##  3rd Qu.:0.00000   3rd Qu.:1.000
##  Max.   :1.00000   Max.   :1.000
```

```
ggplot(data = df, aes(cnt_bike)) +
  xlim(0, 9000) +
  ylim(0, 70) +
  geom_histogram(colour = "grey", fill = "black") +
  ggtitle("Distrubution of the number of bike-rides") +
  labs(x = "Number of bikes")
```

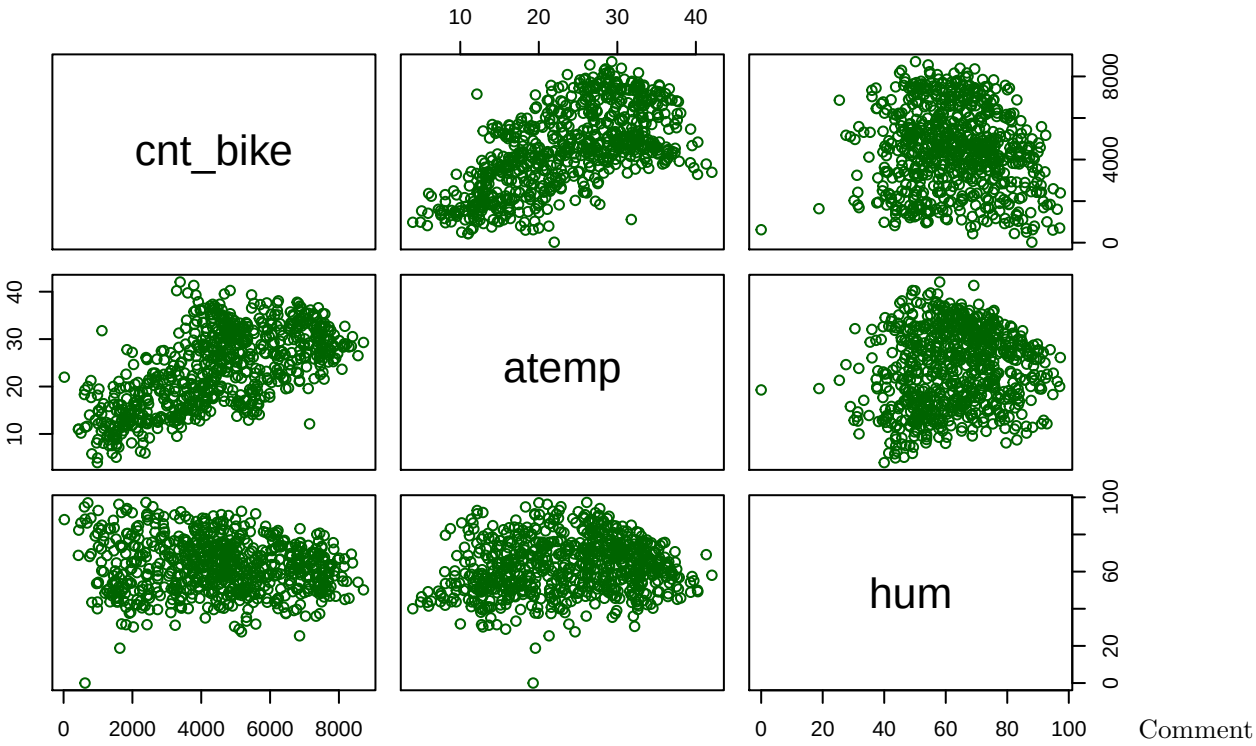
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



b) Use the function pairs() to produce a plot of the relationships among count, atemp and hum.

```
pairs(df[,1:3], col = "darkgreen")
```



Comment on distribution

c) (0.2) Split the data into 80% training and 20% testing.

```
trainRows <- createDataPartition(y = df$cnt_bike, p = 0.8, list = FALSE)
train_set <- df[trainRows,]
test_set <- df[-trainRows,]
```

Train a K-NN model
a) Decide whether you need to standardize the data or not
Yes. We need to standardize the data in K-NN. We will standardize all the attributes besides cnt_bike

```
train_set_stand <- train_set
test_set_stand <- test_set
library(standardize)
```

```
##
##  ************************************************************
##           Loading standardize package version 0.2.2
##      Call standardize.news() to see new features/changes
##  ************************************************************
```

```
#Apply the standardization
train_set_stand[,2:7] <- apply(train_set_stand[,2:7], MARGIN = 2, FUN = scale)
test_set_stand[,2:7] <- apply(test_set_stand[,2:7], MARGIN = 2, FUN = scale)
```

b) Train a k-NN model on the appropriate attributes.

```
knn_model <- train(cnt_bike~., train_set_stand, method = "knn")
knn_model
```

```
## k-Nearest Neighbors
##
## 587 samples
```

```
##   6 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 587, 587, 587, 587, 587, 587, ...
## Resampling results across tuning parameters:
##
##   k  RMSE      Rsquared   MAE
##   5  1481.073  0.4400876  1193.958
##   7  1428.306  0.4704230  1162.359
##   9  1407.003  0.4828142  1149.806
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```