Data Science with R_Capstone Project

Ankita Gairola

2023-08-22

Project scenario

Your project is to analyze how weather would affect bike-sharing demand in urban areas. To complete this project, you need to first collect and process related weather and bike-sharing demand data from various sources, perform exploratory data analysis on the data, and build predictive models to predict bike-sharing demand. You will combine your results and connect them to a live dashboard displaying an interactive map and associated visualization of the current weather and the estimated bike demand.

Tasks:

1. Collecting and understanding data from multiple sources
2. Performing data wrangling and preparation with regular expressions and Tidyverse
3. Performing exploratory data analysis and visualization using Tidyverse and ggplot2
4. Performing modelling the data with linear regressions using Tidymodels
5. Building an interactive dashboard using R Shiny

Web scrape a Global Bike-Sharing Systems Wiki Page

Import Packages

install.packages("tidyverse")

install.packages("rio")

install.packages("lubridate")

install.packages("ggplot2")

install.packages("tidymodels")

install.packages("ggthemes")

install.packages("ggpubr")

install.packages("glmnet")

install.packages("rvest")

install.packages("fastDummies")

```r
library(rio)

library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr   1.1.2     ✓ readr   2.1.4
## ✓ forcats 1.0.0     ✓ stringr 1.5.0
## ✓ ggplot2 3.4.3     ✓ tibble  3.2.1
## ✓ purrr   1.0.1     ✓ tidyr   1.3.0

## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(ggplot2)

library(ggpubr)

library(glmnet)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-7

library(rvest)

##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(fastDummies)
```

```
## Thank you for using fastDummies!
## To acknowledge our work, please cite the package:
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy
(Binary) Columns and Rows from Categorical Variables. Version 1.7.1. URL:
https://github.com/jacobkap/fastDummies,
https://jacobkap.github.io/fastDummies/.
```

```
library(ggthemes)
```

```
library(tidymodels)
```

```
## ── Attaching packages ─────────────────────────────────────── tidymodels
1.1.0 ──
## ✓ broom        1.0.5     ✓ rsample      1.1.1
## ✓ dials        1.2.0     ✓ tune         1.1.1
## ✓ infer        1.0.4     ✓ workflows    1.1.3
## ✓ modeldata    1.2.0     ✓ workflowsets 1.0.1
## ✓ parsnip      1.1.1     ✓ yardstick    1.2.0
## ✓ recipes      1.0.6
## ── Conflicts ────────────────────────────────────────
tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ Matrix::expand()  masks tidyr::expand()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ Matrix::pack()    masks tidyr::pack()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## ✖ Matrix::unpack()  masks tidyr::unpack()
## ✖ recipes::update() masks Matrix::update(), stats::update()
## • Search for functions across packages at https://www.tidymodels.org/find/
```

```
Sys.setlocale("LC_TIME", "English_Hong Kong")
```

```
## Warning in Sys.setlocale("LC_TIME", "English_Hong Kong"): OS reports
request to
## set locale to "English_Hong Kong" cannot be honored
```

```
## [1] ""
```

```
options(repr.plot.width = 16, repr.plot.height = 9)
```

Task_1: Extract bike sharing systems HTML table from a Wiki page and convert it into a data frame

```
url = "https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems"

data <- read_html(url)
```

```
table_nodes <- html_nodes(data, "table")

bike_df <- data %>%
  html_element("table") %>%
  html_table()
print(df)
```

```
## function (x, df1, df2, ncp, log = FALSE)
## {
##     if (missing(ncp))
##         .Call(C_df, x, df1, df2, log)
##     else .Call(C_dnf, x, df1, df2, ncp, log)
## }
## <bytecode: 0x000001c2ea382f10>
## <environment: namespace:stats>
```

Export to csv file

```
write.csv(bike_df,"E:/Coursera/Data Science with R_Capstone
Project/raw_bike_sharing_system.csv")
```

Task_2: The current weather data for a city using OpenWeather API

install.packages("httr")

```
library(httr)
```

URL for Current Weather API

```
current_weather_url <- 'https://api.openweathermap.org/data/2.5/weather'
```

List to hold URL parameters for current weather API

```
my_api_key <- "63cbacfaa61878d2acc06654c13fd311"

current_query <- list(q="Seoul",appid=my_api_key,units="metric")
```

HTTP request to the current weather API

```
response <- GET(current_weather_url, query=current_query)
http_type(response)
```

```
## [1] "application/json"
```

Read JASON Http data

```
json_result <- content(response, as="parsed")
json_result
```

```
## $coord
## $coord$lon
## [1] 126.9778
##
## $coord$lat
## [1] 37.5683
##
##
```

```
## $weather
## $weather[[1]]
## $weather[[1]]$id
## [1] 804
##
## $weather[[1]]$main
## [1] "Clouds"
##
## $weather[[1]]$description
## [1] "overcast clouds"
##
## $weather[[1]]$icon
## [1] "04d"
##
##
##
## $base
## [1] "stations"
##
## $main
## $main$temp
## [1] 21.89
##
## $main$feels_like
## [1] 22.67
##
## $main$temp_min
## [1] 21.69
##
## $main$temp_max
## [1] 22.66
##
## $main$pressure
## [1] 1007
##
## $main$humidity
## [1] 97
##
##
## $visibility
## [1] 3000
##
## $wind
## $wind$speed
## [1] 2.57
##
## $wind$deg
## [1] 40
##
##
## $clouds
```

```
## $clouds$all
## [1] 100
##
##
## $dt
## [1] 1692827402
##
## $sys
## $sys$type
## [1] 1
##
## $sys$id
## [1] 8105
##
## $sys$country
## [1] "KR"
##
## $sys$sunrise
## [1] 1692824105
##
## $sys$sunset
## [1] 1692872080
##
##
## $timezone
## [1] 32400
##
## $id
## [1] 1835848
##
## $name
## [1] "Seoul"
##
## $cod
## [1] 200
```

```
class(json_result)
```

```
## [1] "list"
```

```
json_result
```

```
## $coord
## $coord$lon
## [1] 126.9778
##
## $coord$lat
## [1] 37.5683
##
##
## $weather
## $weather[[1]]
```

```
## $weather[[1]]$id
## [1] 804
##
## $weather[[1]]$main
## [1] "Clouds"
##
## $weather[[1]]$description
## [1] "overcast clouds"
##
## $weather[[1]]$icon
## [1] "04d"
##
##
##
## $base
## [1] "stations"
##
## $main
## $main$temp
## [1] 21.89
##
## $main$feels_like
## [1] 22.67
##
## $main$temp_min
## [1] 21.69
##
## $main$temp_max
## [1] 22.66
##
## $main$pressure
## [1] 1007
##
## $main$humidity
## [1] 97
##
##
## $visibility
## [1] 3000
##
## $wind
## $wind$speed
## [1] 2.57
##
## $wind$deg
## [1] 40
##
##
## $clouds
## $clouds$all
## [1] 100
```

```
##
##
## $dt
## [1] 1692827402
##
## $sys
## $sys$type
## [1] 1
##
## $sys$id
## [1] 8105
##
## $sys$country
## [1] "KR"
##
## $sys$sunrise
## [1] 1692824105
##
## $sys$sunset
## [1] 1692872080
##
##
## $timezone
## [1] 32400
##
## $id
## [1] 1835848
##
## $name
## [1] "Seoul"
##
## $cod
## [1] 200
```

Create some empty vectors to hold data temporarily

```
city <-c()
weather <- c()
visibility <- c()
temp <- c()
temp_min <- c()
temp_max <- c()
pressure <- c()
humidity <- c()
wind_speed <- c()
wind_deg <- c()
```

Assign the values in the json_result list into different vectors

```
city <- c(city, json_result$name)
weather <- c(weather, json_result$weather[[1]]$main)
visibility <- c(visibility, json_result$visibility)
temp <- c(temp, json_result$main$temp)
```

```
temp_min <- c(temp_min, json_result$main$temp_min)
temp_max <- c(temp_max, json_result$main$temp_max)
pressure <- c(pressure, json_result$main$pressure)
humidity <- c(humidity, json_result$main$humidity)
wind_speed <- c(wind_speed, json_result$wind$speed)
wind_deg <- c(wind_deg, json_result$wind$deg)
```

Combine all vectors as columns of a data frame

```
weather_data_frame <- data.frame(city = city,
                                 weather=weather,
                                 visibility=visibility,
                                 temp=temp,
                                 temp_min=temp_min,
                                 temp_max=temp_max,
                                 pressure=pressure,
                                 humidity=humidity,
                                 wind_speed=wind_speed,
                                 wind_deg=wind_deg)

print(weather_data_frame)
```

```
##    city weather visibility  temp temp_min temp_max pressure humidity
wind_speed
## 1 Seoul  Clouds       3000 21.89    21.69    22.66     1007       97
2.57
##   wind_deg
## 1       40
```

Task_3: 5-day weather forecasts for cities using the OpenWeather API

```
city <- c()
weather <- c()
visibility <- c()
temp <- c()
temp_min <- c()
temp_max <- c()
pressure <- c()
humidity <- c()
wind_speed <- c()
wind_deg <- c()

# Get 5 -day weather forecast for a list of cities
weather_forecast_by_cities <- function(city_names) {
  df <- data.frame()
  for (city_name in city_names) {

    #forecast API URL
    forecast_url <-'https://api.openweathermap.org/data/2.5/weather'

    #create query parameter
    forecast_query <- list(q=city_name,appid=my_api_key, units="metric")

    #make HTTP GET call for the given city
```

```
    response <- GET(forecast_url, query=forecast_query)

    json_result <- content(response, as="parsed")
    results <- json_result$list

    #Loop the json result
    for(result in results) {
      city <- c(city, city_name)
    }

    # Add R lists into a data frame
    city <- c(city, json_result$name)
    weather <- c(weather, json_result$weather[[1]]$main)
    visibility <- c(visibility, json_result$visibility)
    temp <- c(temp, json_result$main$temp)
    temp_min <-c(temp_min, json_result$main$temp_min)
    temp_max <- c(temp_max, json_result$main$temp_max)
    pressure <- c(pressure, json_result$main$pressure)
    humidity <- c(humidity, json_result$main$humidity)
    wind_speed <- c(wind_speed, json_result$wind$speed)
    wind_deg <-c(wind_deg, json_result$wind$deg)

    #Combine all vector into data frame
    df <- data.frame(city = city,
                             weather=weather,
                              visibility=visibility,
                              temp=temp,
                              temp_min=temp_min,
                              temp_max=temp_max,
                              pressure=pressure,
                              humidity=humidity,
                              wind_speed=wind_speed,
                              wind_deg=wind_deg)
  }
  return(df)
}

cities <- c("Seoul", "Washington, D.C.", "Paris", "Suzhou", "New Delhi",
"Kyoto", "Cologne", "London" )
cities_weather_df <- weather_forecast_by_cities(cities)
print(cities_weather_df)
```

```
##         city weather visibility  temp temp_min temp_max pressure humidity
## 1      Seoul  Clouds       3000 21.89    21.69    22.66     1007       97
## 2 Washington  Clouds      10000 23.11    21.85    23.99     1013       35
## 3      Paris  Clouds      10000 25.38    22.22    26.73     1016       69
## 4     Suzhou   Clear      10000 23.29    23.29    23.29     1007       89
## 5  New Delhi    Mist       3500 27.09    27.09    27.09     1000       89
## 6      Kyoto    Rain      10000 26.67    24.98    27.59     1010       87
## 7    Cologne   Clear      10000 19.55    17.64    21.64     1020       72
## 8     London  Clouds      10000 20.93    18.36    22.27     1015       71
```

```
##   wind_speed wind_deg
## 1       2.57       40
## 2       1.15      250
## 3       3.09       60
## 4       1.48       33
## 5       1.54      250
## 6       2.55      102
## 7       0.51        0
## 8       2.06      190
```

Task_4: Download datasets as csv files from cloud storage

some general city information such as name and locations

```
url <- "https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0321EN-
SkillsNetwork/labs/datasets/raw_worldcities.csv"


download.file(url, destfile = "raw_worldcities.csv")
```

specific hourly Seoul bike sharing demand dataset

```
url <- "https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0321EN-
SkillsNetwork/labs/datasets/raw_seoul_bike_sharing.csv"


download.file(url, destfile = "raw_seoul_bike_sharing.csv")
```

Download raw_cities_weather_forecast

```
url <- "https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-RP0321EN-
SkillsNetwork/labs/datasets/raw_cities_weather_forecast.csv"

download.file(url, destfile = "raw_cities_weather_forecast.csv")
```

Data Wrangling with Regular Expressions

```
dataset_list <- c('raw_bike_sharing_systems.csv',
'raw_seoul_bike_sharing.csv', 'raw_cities_weather_forecast.csv',
'raw_worldcities.csv')
```

Task_5: Standardize column names for all collected datasets

Convert iterate over the above datasets and convert their column names

```
for (dataset_name in dataset_list) {
  if (file.exists(dataset_name)) {  # Check if the file exists
    dataset <- read.csv(dataset_name, fileEncoding = "UTF-8")  # Specify
encoding if needed
    names(dataset) <- toupper(names(dataset))
    names(dataset) <- str_replace_all(names(dataset), " ", "_")
    write.csv(dataset, dataset_name, row.names = FALSE)
  } else {
```

```
    cat("File not found:", dataset_name, "\n")
  }
}
```

```
## File not found: raw_bike_sharing_systems.csv
```

Process the web-scraped bike sharing system dataset

Load Dataset

```
bike_sharing_df <- read.csv("E:/Coursera/Data Science with R_Capstone
Project/raw_bike_sharing_system.csv")

head(bike_sharing_df)
```

```
##   X   Country                  City                 Name            System
## 1 1   Albania             Tirana[5]             Ecovolis
## 2 2 Argentina    Buenos Aires[6][7]              Ecobici Serttel Brasil[8]
## 3 3 Argentina           Mendoza[10]            Metrobici
## 4 4 Argentina               Rosario  Mi Bici Tu Bici[11]
## 5 5 Argentina San Lorenzo, Santa Fe             Biciudad          Biciudad
## 6 6 Australia         Melbourne[12] Melbourne Bike Share         PBSC & 8D
##                         Operator         Launched         Discontinued
Stations
## 1                                       March 2011
8
## 2 Bike In Baires Consortium[9]             2010
400
## 3                                          2014
2
## 4                                2 December 2015
47
## 5                               27 November 2016
8
## 6                     Motivate        June 2010 30 November 2019[13]
53
##   Bicycles Daily.ridership
## 1      200
## 2     4000           21917
## 3       40
## 4      480
## 5       80
## 6      676
```

```
sub_bike_sharing_df <- bike_sharing_df %>%
  select(Country, City, System, Bicycles)


sub_bike_sharing_df %>%
  summarize_all(class) %>%
  gather(variable, class)
```

```
##   variable     class
## 1  Country character
## 2     City character
## 3   System character
## 4 Bicycles character
```

```
find_character <- function(strings) grepl("[^0-9]", strings)

sub_bike_sharing_df %>%
  select(Bicycles) %>% #Use the function to check BICYCLES column
  filter(find_character(Bicycles)) %>%
  slice(0:10)
```

```
##                                Bicycles
## 1                      1790 (2019)[21]
## 2                          4200 (2021)
## 3                             4115[25]
## 4  7270 (regular)  2395 (electric)[38]
## 5                              310[65]
## 6                              500[75]
## 7                                 [78]
## 8                              180[79]
## 9                              600[82]
## 10          initially 800 (later 2500)
```

Because there are some values associated with numeric and non-numeric value, BYCICLES was classified as character.

Check if COUNTRY, CITY, SYSTEM have any reference link, such as Melbourne[12]

Create a function to check if there is any reference link in the values

```
ref_pattern <- "\\[[A-z0-9]+\\]"
find_reference_pattern <- function(strings) grepl(ref_pattern, strings)
```

Check whether the CITY column has any reference links

```
sub_bike_sharing_df %>%
  select(City) %>%
  filter(find_reference_pattern(City)) %>%
  slice(1:10)
```

```
##                       City
## 1                Tirana[5]
## 2       Buenos Aires[6][7]
## 3               Mendoza[10]
## 4            Melbourne[12]
## 5            Melbourne[12]
## 6          Brisbane[14][15]
## 7         Lower Austria[16]
## 8  Different locations[19]
## 9              Brussels[24]
## 10                Namur[26]
```

Check whether the System column has any reference links

```
sub_bike_sharing_df %>%
    select(System) %>%
    filter(find_reference_pattern(System)) %>%
    slice(0:10)
```

```
##                               System
## 1                  Serttel Brasil[8]
## 2                       EasyBike[64]
## 3                         4 Gen.[72]
## 4               3 Gen. SmooveKey[135]
## 5 3 Gen. Smoove[162][163][164][160]
## 6                  3 Gen. Smoove[200]
## 7                  3 Gen. Smoove[202]
## 8                  3 Gen. Smoove[204]
```

Task_6: Remove undesired reference links using regular expressions

remove reference link

```
remove_ref <- function(strings) {
  ref_pattern <- "\\[[A-z0-9]+\\]" # Define a pattern matching a reference
link such as [1]
  result <- stringr::str_replace_all(strings,ref_pattern,"")  # Replace all
matched substrings with a white space
  result <-  trimws(result)
    return(result)
}
```

install.packages("magrittr")

install.packages("dplyr")

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

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

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(dplyr)
```

Use the function to remove the reference links

```
sub_bike_sharing_df %<>% #use mutate and remove_ref fcn to remove ref in CITY
and SYSTEM
  mutate(System=remove_ref(System),
         City=remove_ref(City))
```

Check whether all reference links are removed

```
sub_bike_sharing_df %>%
  select(Country, City, System, Bicycles) %>%
  filter (find_reference_pattern(Country) | find_reference_pattern(City) |
find_reference_pattern(City) |find_reference_pattern(Bicycles) )
```

```
##           Country                                          City
## 1         Belgium                          Different locations
## 2         Belgium                                     Brussels
## 3          Canada                                     Montreal
## 4          Cyprus                    Limassol, Nicosia District
## 5         Czechia                                       Prague
## 6         Czechia                                     Prague 7
## 7         Czechia                                    Prostějov
## 8         Czechia                                      Ostrava
## 9         Denmark                                        Farsø
## 10        Finland                                      Kouvola
## 11        Finland                                       Kuopio
## 12        Finland                                        Lahti
## 13        Finland                                  Lappeenranta
## 14        Finland                                         Pori
## 15        Finland                                     Raseborg
## 16        Finland                                    Riihimäki
## 17        Finland                                      Tampere
## 18        Finland                                        Turku
## 19        Finland                                      Varkaus
## 20        Georgia                                       Batumi
## 21        Germany                                    Darmstadt
## 22         Greece                                        Corfu
## 23        Hungary                                     Budapest
## 24        Hungary                                         Győr
## 25        Hungary                                    Kaposvár
## 26          Italy                                       Milan
## 27      Lithuania                                       Kaunas
## 28    Netherlands Various Locations (especially railway stations)
## 29         Russia                                        Kazan
## 30       Slovakia                                   Bratislava
## 31       Slovakia                                   Bratislava
## 32       Slovakia                                        Košice
## 33       Slovakia                          Moldava nad Bodvou
## 34       Slovakia                                    Terchová
## 35       Slovakia                                      Žilina
## 36    South Korea                                     Changwon
## 37 United Kingdom                           Glasgow, Scotland
## 38 United Kingdom                 Greater Manchester, England
## 39 United Kingdom                         Edinburgh, Scotland
## 40 United Kingdom                          Liverpool, England
##                            System                            Bicycles
## 1                        Blue-bike                  1790 (2019)[21]
## 2                3 Gen. Cyclocity                           4115[25]
## 3                       PBSC & 8D 7270 (regular)  2395 (electric)[38]
```

```
## 4                     3 Gen. Smoove                                310[65]
## 5                                                                  500[75]
## 6                         4 Gen. Ofo                                 [78]
## 7                  3 Gen. nextbike                                 180[79]
## 8                  3 Gen. nextbike                                 600[82]
## 9                            2 Gen                                   [89]
## 10                 Donkey Republic  60(regular) 10(electric)(2022)[96]
## 11                        Freebike                       350(2022) [97]
## 12                        Freebike                     250 (2022) [98]
## 13                 Donkey Republic                       120(2022)[99]
## 14                       Rolanbike                     50 (2022) [103]
## 15                 Donkey Republic                     30 (2022) [104]
## 16                 Donkey Republic                    60 (2022)  [105]
## 17                 CityBike Global                     700 (2022)[106]
## 18                        Nextbike                    700 (2022) [107]
## 19                            Juro                     20 (2022) [110]
## 20                3 Gen. SmooveKey                            370[136]
## 21    3 & 4 Gen. Call a Bike flex                           350 [147]
## 22                    3 Gen. Smoove                             100[64]
## 23                           3 Gen.                           1526[180]
## 24                                                             180[182]
## 25                                    32 (including 6 rollers)  [183]
## 26                 3 Gen. Clear CC                 5430 (1000 E)[195]
## 27                                                          150 E[207]
## 28 OV-Fiets/Nederlandse Spoorwegen                        21700 [210]
## 29                3 Gen. Cyclocity                            120[238]
## 30                                                            400[240]
## 31                                                             80[242]
## 32                                                            500[246]
## 33                                                             25[249]
## 34                                                             60[261]
## 35                        nextbike                            123[266]
## 36                                                           2348[275]
## 37                 3 Gen. nextbike                             400[297]
## 38                                                           1500[299]
## 39                   Urban Sharing                             500[300]
## 40                                                           1000[301]
```

Task_7: Extract the numeric value using regular expressions

```
extract_num <- function(columns) {
  digitals_pattern <- "\\d+" #define a pattern matching digital substring
  str_extract(columns,digitals_pattern) %>%
  as.numeric()
}
```

install.packages("stringr")

```
library(stringr)

sub_bike_sharing_df %<>% #use mutate and to apply function to BICYLCLES
  mutate(Bicycles=extract_num(Bicycles))
```

```
write.csv(sub_bike_sharing_df, "E:/Coursera/Data Science with R_Capstone
Project/bike_sharing_system.csv")
```

Data Wrangling with dplyr

Quick look at the dataset

```
summary(bike_sharing_df)
```

```
##        X           Country              City               Name
##  Min.   :  1.0   Length:556         Length:556         Length:556
##  1st Qu.:139.8   Class :character   Class :character   Class :character
##  Median :278.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :278.5
##  3rd Qu.:417.2
##  Max.   :556.0
##     System            Operator           Launched         Discontinued
##  Length:556         Length:556         Length:556         Length:556
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Stations           Bicycles         Daily.ridership
##  Length:556         Length:556         Length:556
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
dim(bike_sharing_df)
```

```
## [1] 556  11
```

Task_8: Detect and handle missing values

```
dataset_list <- c('bike_sharing_system.csv','raw_seoul_bike_sharing.csv')
for (dataset_name in dataset_list) {
  dataset <- read.csv(dataset_name)
  names(dataset) <- toupper(names(dataset))
  names(dataset) <- str_replace_all(names(dataset), " ", "_")
  write.csv(dataset, dataset_name, row.names = FALSE)
}

bike_sharing_df <- read.csv("raw_seoul_bike_sharing.csv")
summary(bike_sharing_df)
```

```
##      DATE           RENTED_BIKE_COUNT      HOUR        TEMPERATURE
##  Length:8760        Min.   :   2.0    Min.   : 0.00   Min.   :-17.80
##  Class :character   1st Qu.: 214.0    1st Qu.: 5.75   1st Qu.:  3.40
##  Mode  :character   Median : 542.0    Median :11.50   Median : 13.70
##                     Mean   : 729.2    Mean   :11.50   Mean   : 12.87
##                     3rd Qu.:1084.0    3rd Qu.:17.25   3rd Qu.: 22.50
```

```
##                    Max.   :3556.0    Max.   :23.00   Max.   : 39.40
##                    NA's   :295                        NA's   :11
##     HUMIDITY       WIND_SPEED      VISIBILITY   DEW_POINT_TEMPERATURE
##  Min.   : 0.00   Min.   :0.000   Min.   :  27   Min.   :-30.600
##  1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 940   1st Qu.: -4.700
##  Median :57.00   Median :1.500   Median :1698   Median :  5.100
##  Mean   :58.23   Mean   :1.725   Mean   :1437   Mean   :  4.074
##  3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 14.800
##  Max.   :98.00   Max.   :7.400   Max.   :2000   Max.   : 27.200
##
##  SOLAR_RADIATION     RAINFALL          SNOWFALL         SEASONS
##  Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000   Length:8760
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   Class :character
##  Median :0.0100   Median : 0.0000   Median :0.00000   Mode  :character
##  Mean   :0.5691   Mean   : 0.1487   Mean   :0.07507
##  3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :3.5200   Max.   :35.0000   Max.   :8.80000
##
##    HOLIDAY          FUNCTIONING_DAY
##  Length:8760        Length:8760
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

```
dim(bike_sharing_df)
```

```
## [1] 8760   14
```

missing values in the TEMPERATURE column

```
bike_sharing_df %>%
                filter(is.na(TEMPERATURE))
```

```
##          DATE RENTED_BIKE_COUNT HOUR TEMPERATURE HUMIDITY WIND_SPEED
VISIBILITY
## 1  07/06/2018             3221   18          NA       57        2.7
1217
## 2  12/06/2018             1246   14          NA       45        2.2
1961
## 3  13/06/2018             2664   17          NA       57        3.3
919
## 4  17/06/2018             2330   17          NA       58        3.3
865
## 5  20/06/2018             2741   19          NA       61        2.7
1236
## 6  30/06/2018             1144   13          NA       87        1.7
390
## 7  05/07/2018              827   10          NA       75        1.1
1028
## 8  11/07/2018              634    9          NA       96        0.6
```

```
450
## 9  12/07/2018               593    6          NA       93        1.1
852
## 10 21/07/2018               347    4          NA       77        1.2
1203
## 11 21/08/2018              1277   23          NA       75        0.1
1892
##    DEW_POINT_TEMPERATURE SOLAR_RADIATION RAINFALL SNOWFALL SEASONS
HOLIDAY
## 1                   16.4            0.96      0.0        0  Summer No
Holiday
## 2                   12.7            1.39      0.0        0  Summer No
Holiday
## 3                   16.4            0.87      0.0        0  Summer No
Holiday
## 4                   16.7            0.66      0.0        0  Summer No
Holiday
## 5                   17.5            0.60      0.0        0  Summer No
Holiday
## 6                   23.2            0.71      3.5        0  Summer No
Holiday
## 7                   20.8            1.22      0.0        0  Summer No
Holiday
## 8                   24.9            0.41      0.0        0  Summer No
Holiday
## 9                   24.3            0.01      0.0        0  Summer No
Holiday
## 10                  21.2            0.00      0.0        0  Summer No
Holiday
## 11                  20.8            0.00      0.0        0  Summer No
Holiday
##    FUNCTIONING_DAY
## 1              Yes
## 2              Yes
## 3              Yes
## 4              Yes
## 5              Yes
## 6              Yes
## 7              Yes
## 8              Yes
## 9              Yes
## 10             Yes
## 11             Yes
```

missing value in RENTED_BIKE_COUNT column

```
bike_sharing_df %>%
                filter(is.na(RENTED_BIKE_COUNT))

##           DATE RENTED_BIKE_COUNT HOUR TEMPERATURE HUMIDITY WIND_SPEED
## 1   11/04/2018               NA    0       14.40       82        4.6
## 2   11/04/2018               NA    1       13.60       81        3.6
```

```
## 3   11/04/2018                NA     2       12.70        80         3.9
## 4   11/04/2018                NA     3       11.60        81         3.1
## 5   11/04/2018                NA     4       10.20        83         3.5
## 6   11/04/2018                NA     5        9.70        84         1.7
## 7   11/04/2018                NA     6        9.00        86         2.0
## 8   11/04/2018                NA     7        8.80        85         1.1
## 9   11/04/2018                NA     8        9.70        77         1.6
## 10  11/04/2018                NA     9       11.80        59         2.1
## 11  11/04/2018                NA    10       13.30        51         4.1
## 12  11/04/2018                NA    11       14.90        44         4.0
## 13  11/04/2018                NA    12       15.70        46         3.9
## 14  11/04/2018                NA    13       15.60        38         4.7
## 15  11/04/2018                NA    14       16.40        28         3.4
## 16  11/04/2018                NA    15       17.00        28         3.5
## 17  11/04/2018                NA    16       16.90        23         4.4
## 18  11/04/2018                NA    17       16.20        22         4.1
## 19  11/04/2018                NA    18       15.20        21         2.9
## 20  11/04/2018                NA    19       13.20        28         2.9
## 21  11/04/2018                NA    20       11.90        33         2.0
## 22  11/04/2018                NA    21       11.00        34         2.1
## 23  11/04/2018                NA    22       10.30        42         1.0
## 24  11/04/2018                NA    23        9.70        45         2.0
## 25  10/05/2018                NA     0       14.50        63         0.9
## 26  10/05/2018                NA     1       13.90        66         1.0
## 27  10/05/2018                NA     2       13.10        73         1.9
## 28  10/05/2018                NA     3       12.88        74         1.7
## 29  10/05/2018                NA     4       12.00        79         0.6
## 30  10/05/2018                NA     5       11.50        80         0.9
## 31  10/05/2018                NA     6       11.40        80         1.0
## 32  10/05/2018                NA     7       12.10        71         1.6
## 33  10/05/2018                NA     8       13.50        63         2.2
## 34  10/05/2018                NA     9       15.50        55         1.7
## 35  10/05/2018                NA    10       17.30        52         2.3
## 36  10/05/2018                NA    11       18.30        51         3.1
## 37  10/05/2018                NA    12       19.80        49         2.3
## 38  10/05/2018                NA    13       20.70        46         2.9
## 39  10/05/2018                NA    14       21.20        47         3.7
## 40  10/05/2018                NA    15       20.40        51         4.0
## 41  10/05/2018                NA    16       19.60        54         3.9
## 42  10/05/2018                NA    17       19.20        54         3.4
## 43  10/05/2018                NA    18       18.00        61         3.5
## 44  10/05/2018                NA    19       15.50        74         3.6
## 45  10/05/2018                NA    20       13.60        82         2.1
## 46  10/05/2018                NA    21       12.60        85         2.4
## 47  10/05/2018                NA    22       12.30        85         2.3
## 48  10/05/2018                NA    23       12.40        85         1.1
## 49  18/09/2018                NA     0       19.50        73         1.2
## 50  18/09/2018                NA     1       19.30        71         0.4
## 51  18/09/2018                NA     2       18.80        74         0.2
## 52  18/09/2018                NA     3       18.70        76         0.5
## 53  18/09/2018                NA     4       18.30        78         0.4
```

```
## 54  18/09/2018                NA     5       17.90       77        0.5
## 55  18/09/2018                NA     6       17.70       77        0.2
## 56  18/09/2018                NA     7       17.80       76        0.5
## 57  18/09/2018                NA     8       18.70       73        0.9
## 58  18/09/2018                NA     9       20.80       64        1.1
## 59  18/09/2018                NA    10       22.70       54        1.0
## 60  18/09/2018                NA    11       24.10       46        2.1
## 61  18/09/2018                NA    12       24.70       43        1.8
## 62  18/09/2018                NA    13       25.60       40        2.5
## 63  18/09/2018                NA    14       26.10       37        2.3
## 64  18/09/2018                NA    15       26.10       43        2.8
## 65  18/09/2018                NA    16       26.40       34        2.5
## 66  18/09/2018                NA    17       25.50       38        2.9
## 67  18/09/2018                NA    18       24.80       48        1.4
## 68  18/09/2018                NA    19       23.20       57        2.2
## 69  18/09/2018                NA    20       22.60       58        1.4
## 70  18/09/2018                NA    21       22.10       61        1.5
## 71  18/09/2018                NA    22       21.80       65        0.3
## 72  18/09/2018                NA    23       21.20       69        1.3
## 73  19/09/2018                NA     0       21.00       66        0.4
## 74  19/09/2018                NA     1       20.50       64        0.4
## 75  19/09/2018                NA     2       20.00       70        0.2
## 76  19/09/2018                NA     3       19.70       70        0.5
## 77  19/09/2018                NA     4       19.50       70        0.5
## 78  19/09/2018                NA     5       19.30       73        0.9
## 79  19/09/2018                NA     6       19.00       76        0.3
## 80  19/09/2018                NA     7       19.00       74        0.8
## 81  19/09/2018                NA     8       19.80       71        1.1
## 82  19/09/2018                NA     9       21.50       61        0.8
## 83  19/09/2018                NA    10       22.90       59        1.1
## 84  19/09/2018                NA    11       23.60       55        0.6
## 85  19/09/2018                NA    12       24.90       51        1.2
## 86  19/09/2018                NA    13       26.50       39        1.7
## 87  19/09/2018                NA    14       26.10       40        1.2
## 88  19/09/2018                NA    15       25.90       39        1.4
## 89  19/09/2018                NA    16       25.60       41        1.2
## 90  19/09/2018                NA    17       25.00       44        1.3
## 91  19/09/2018                NA    18       24.00       55        2.2
## 92  19/09/2018                NA    19       23.50       58        0.4
## 93  19/09/2018                NA    20       22.60       67        0.3
## 94  19/09/2018                NA    21       21.70       63        1.8
## 95  19/09/2018                NA    22       20.90       71        1.6
## 96  19/09/2018                NA    23       20.40       57        0.9
## 97  28/09/2018                NA     0       17.10       52        1.4
## 98  28/09/2018                NA     1       16.50       53        1.5
## 99  28/09/2018                NA     2       16.00       56        1.5
## 100 28/09/2018                NA     3       15.30       59        1.6
## 101 28/09/2018                NA     4       14.70       62        1.5
## 102 28/09/2018                NA     5       14.50       60        1.4
## 103 28/09/2018                NA     6       14.40       61        1.2
## 104 28/09/2018                NA     7       14.50       62        1.5
```

```
## 105 28/09/2018                NA    8       15.20       58        1.7
## 106 28/09/2018                NA    9       16.50       50        1.2
## 107 28/09/2018                NA   10       17.40       51        1.3
## 108 28/09/2018                NA   11       17.90       54        1.4
## 109 28/09/2018                NA   12       19.00       52        1.2
## 110 28/09/2018                NA   13       19.10       53        1.4
## 111 28/09/2018                NA   14       19.40       53        1.4
## 112 28/09/2018                NA   15       20.30       52        1.3
## 113 28/09/2018                NA   16       21.90       46        1.2
## 114 28/09/2018                NA   17       21.70       52        1.1
## 115 28/09/2018                NA   18       20.40       56        1.7
## 116 28/09/2018                NA   19       19.30       61        0.8
## 117 28/09/2018                NA   20       18.00       67        0.7
## 118 28/09/2018                NA   21       17.80       66        1.1
## 119 28/09/2018                NA   22       17.10       70        1.0
## 120 28/09/2018                NA   23       16.60       69        1.0
## 121 30/09/2018                NA    0       18.10       60        0.2
## 122 30/09/2018                NA    1       17.50       66        0.5
## 123 30/09/2018                NA    2       17.30       68        0.7
## 124 30/09/2018                NA    3       17.00       70        0.9
## 125 30/09/2018                NA    4       16.60       73        0.3
## 126 30/09/2018                NA    5       16.10       78        0.4
## 127 30/09/2018                NA    6       15.50       80        0.4
## 128 30/09/2018                NA    7       15.10       81        0.8
## 129 30/09/2018                NA    8       16.20       69        1.1
## 130 30/09/2018                NA    9       18.50       62        0.5
## 131 30/09/2018                NA   10       20.70       62        1.3
## 132 30/09/2018                NA   11       20.20       63        2.8
## 133 30/09/2018                NA   12       20.80       47        3.1
## 134 30/09/2018                NA   13       21.10       42        4.6
## 135 30/09/2018                NA   14       21.10       32        4.4
## 136 30/09/2018                NA   15       21.30       32        3.8
## 137 30/09/2018                NA   16       21.20       28        3.2
## 138 30/09/2018                NA   17       20.40       31        3.2
## 139 30/09/2018                NA   18       18.30       34        3.3
## 140 30/09/2018                NA   19       17.10       35        2.9
## 141 30/09/2018                NA   20       16.00       39        2.9
## 142 30/09/2018                NA   21       15.20       40        3.0
## 143 30/09/2018                NA   22       14.60       46        2.2
## 144 30/09/2018                NA   23       14.10       53        2.1
## 145 02/10/2018                NA    0       13.00       72        1.8
## 146 02/10/2018                NA    1       12.50       74        1.9
## 147 02/10/2018                NA    2       12.30       75        1.6
## 148 02/10/2018                NA    3       11.80       78        0.3
## 149 02/10/2018                NA    4       11.20       80        0.3
## 150 02/10/2018                NA    5       10.80       82        0.3
## 151 02/10/2018                NA    6       10.60       81        0.0
## 152 02/10/2018                NA    7       10.40       81        0.3
## 153 02/10/2018                NA    8       11.20       69        0.9
## 154 02/10/2018                NA    9       13.90       66        1.6
## 155 02/10/2018                NA   10       16.70       58        2.4
```

```
## 156 02/10/2018                NA   11       18.30       44        3.0
## 157 02/10/2018                NA   12       19.40       41        3.1
## 158 02/10/2018                NA   13       20.40       36        2.6
## 159 02/10/2018                NA   14       21.50       39        2.5
## 160 02/10/2018                NA   15       21.80       38        2.6
## 161 02/10/2018                NA   16       21.70       36        3.2
## 162 02/10/2018                NA   17       21.20       40        2.4
## 163 02/10/2018                NA   18       19.20       48        2.2
## 164 02/10/2018                NA   19       17.60       57        1.7
## 165 02/10/2018                NA   20       16.80       62        2.0
## 166 02/10/2018                NA   21       16.20       65        1.5
## 167 02/10/2018                NA   22       15.50       69        0.8
## 168 02/10/2018                NA   23       14.80       68        0.1
## 169 04/10/2018                NA    0       15.50       65        0.8
## 170 04/10/2018                NA    1       14.70       69        0.6
## 171 04/10/2018                NA    2       14.00       73        0.5
## 172 04/10/2018                NA    3       13.60       75        0.8
## 173 04/10/2018                NA    4       13.30       75        0.4
## 174 04/10/2018                NA    5       13.20       73        0.9
## 175 04/10/2018                NA    6       13.20       73        1.1
## 176 04/10/2018                NA    7       13.20       75        0.8
## 177 04/10/2018                NA    8       13.90       68        0.9
## 178 04/10/2018                NA    9       16.70       58        1.8
## 179 04/10/2018                NA   10       19.00       49        1.7
## 180 04/10/2018                NA   11       21.30       41        1.9
## 181 04/10/2018                NA   12       23.20       38        1.7
## 182 04/10/2018                NA   13       23.40       39        1.4
## 183 04/10/2018                NA   14       24.30       39        1.7
## 184 04/10/2018                NA   15       25.30       39        1.6
## 185 04/10/2018                NA   16       25.10       41        1.0
## 186 04/10/2018                NA   17       24.20       44        1.5
## 187 04/10/2018                NA   18       23.10       48        1.3
## 188 04/10/2018                NA   19       22.30       50        1.4
## 189 04/10/2018                NA   20       21.30       53        1.8
## 190 04/10/2018                NA   21       20.50       57        2.0
## 191 04/10/2018                NA   22       19.80       60        1.6
## 192 04/10/2018                NA   23       19.40       62        1.7
## 193 06/10/2018                NA    0       16.40       94        1.8
## 194 06/10/2018                NA    1       16.70       89        2.4
## 195 06/10/2018                NA    2       16.90       88        2.5
## 196 06/10/2018                NA    3       16.80       90        3.3
## 197 06/10/2018                NA    4       16.90       89        2.7
## 198 06/10/2018                NA    5       16.70       93        2.8
## 199 06/10/2018                NA    6       16.50       96        2.3
## 200 09/10/2018                NA    0       12.60       54        0.2
## 201 09/10/2018                NA    1       12.00       58        0.7
## 202 09/10/2018                NA    2       11.40       65        1.1
## 203 09/10/2018                NA    3       11.00       66        0.9
## 204 09/10/2018                NA    4       11.00       63        0.8
## 205 09/10/2018                NA    5       10.80       64        0.8
## 206 09/10/2018                NA    6       10.80       69        0.8
```

```
## 207 09/10/2018                NA     7       10.90       70        0.7
## 208 09/10/2018                NA     8       11.40       66        0.8
## 209 09/10/2018                NA     9       12.80       57        1.2
## 210 09/10/2018                NA    10       14.80       51        1.0
## 211 09/10/2018                NA    11       16.40       45        0.7
## 212 09/10/2018                NA    12       18.30       40        0.6
## 213 09/10/2018                NA    13       19.20       39        1.0
## 214 09/10/2018                NA    14       18.80       39        1.6
## 215 09/10/2018                NA    15       18.30       40        1.2
## 216 09/10/2018                NA    16       19.00       39        1.4
## 217 09/10/2018                NA    17       18.40       39        2.1
## 218 09/10/2018                NA    18       17.70       43        1.5
## 219 09/10/2018                NA    19       17.30       47        2.2
## 220 09/10/2018                NA    20       17.00       49        1.4
## 221 09/10/2018                NA    21       16.80       52        1.6
## 222 09/10/2018                NA    22       16.70       55        1.9
## 223 09/10/2018                NA    23       16.60       53        1.1
## 224 03/11/2018                NA     0        8.80       63        0.6
## 225 03/11/2018                NA     1        8.50       61        0.2
## 226 03/11/2018                NA     2        7.60       68        1.2
## 227 03/11/2018                NA     3        7.10       67        0.6
## 228 03/11/2018                NA     4        6.30       72        1.0
## 229 03/11/2018                NA     5        6.10       75        1.0
## 230 03/11/2018                NA     6        5.50       78        0.6
## 231 03/11/2018                NA     7        5.20       79        1.3
## 232 03/11/2018                NA     8        5.40       75        0.4
## 233 03/11/2018                NA     9        7.90       58        1.3
## 234 03/11/2018                NA    10       11.00       49        0.7
## 235 03/11/2018                NA    11       14.50       42        1.6
## 236 03/11/2018                NA    12       16.50       37        1.8
## 237 03/11/2018                NA    13       17.80       29        2.0
## 238 03/11/2018                NA    14       18.10       24        2.4
## 239 03/11/2018                NA    15       18.50       25        1.6
## 240 03/11/2018                NA    16       18.70       25        2.1
## 241 03/11/2018                NA    17       16.70       45        2.2
## 242 03/11/2018                NA    18       14.20       51        1.5
## 243 03/11/2018                NA    19       13.40       62        1.6
## 244 03/11/2018                NA    20       12.10       68        0.8
## 245 03/11/2018                NA    21       11.10       69        0.9
## 246 03/11/2018                NA    22       10.00       68        1.1
## 247 03/11/2018                NA    23        9.20       65        1.2
## 248 06/11/2018                NA     0       10.00       73        0.5
## 249 06/11/2018                NA     1        9.80       75        0.5
## 250 06/11/2018                NA     2        9.70       78        0.0
## 251 06/11/2018                NA     3        9.40       81        0.3
## 252 06/11/2018                NA     4        9.10       81        0.0
## 253 06/11/2018                NA     5        8.80       84        0.6
## 254 06/11/2018                NA     6        8.50       83        0.7
## 255 06/11/2018                NA     7        8.30       84        0.9
## 256 06/11/2018                NA     8        8.70       77        0.7
## 257 06/11/2018                NA     9       10.10       68        0.2
```

```
## 258 06/11/2018                NA   10       11.70       62        0.5
## 259 06/11/2018                NA   11       13.00       55        1.0
## 260 06/11/2018                NA   12       14.50       57        0.8
## 261 06/11/2018                NA   13       16.10       56        1.7
## 262 06/11/2018                NA   14       17.40       50        2.4
## 263 06/11/2018                NA   15       16.30       54        1.6
## 264 06/11/2018                NA   16       16.60       55        1.3
## 265 06/11/2018                NA   17       15.30       63        1.2
## 266 06/11/2018                NA   18       14.00       71        1.5
## 267 06/11/2018                NA   19       13.30       72        1.1
## 268 06/11/2018                NA   20       13.20       73        1.1
## 269 06/11/2018                NA   21       13.00       76        0.9
## 270 06/11/2018                NA   22       12.30       73        1.2
## 271 06/11/2018                NA   23       12.20       74        1.1
## 272 09/11/2018                NA    0       12.00       96        3.1
## 273 09/11/2018                NA    1       11.40       89        2.6
## 274 09/11/2018                NA    2       11.30       88        3.7
## 275 09/11/2018                NA    3       10.80       82        2.8
## 276 09/11/2018                NA    4       10.60       75        3.2
## 277 09/11/2018                NA    5       10.90       68        3.1
## 278 09/11/2018                NA    6       11.00       70        3.1
## 279 09/11/2018                NA    7       10.40       81        3.0
## 280 09/11/2018                NA    8       10.10       85        2.4
## 281 09/11/2018                NA    9       10.80       78        2.6
## 282 09/11/2018                NA   10       11.30       68        4.7
## 283 09/11/2018                NA   11       13.30       59        4.1
## 284 09/11/2018                NA   12       13.90       50        4.1
## 285 09/11/2018                NA   13       14.00       46        5.3
## 286 09/11/2018                NA   14       15.00       49        2.8
## 287 09/11/2018                NA   15       15.10       52        3.4
## 288 09/11/2018                NA   16       14.30       59        3.6
## 289 09/11/2018                NA   17       12.88       66        3.6
## 290 09/11/2018                NA   18       12.30       69        2.1
## 291 09/11/2018                NA   19       11.90       71        2.7
## 292 09/11/2018                NA   20       11.90       72        2.5
## 293 09/11/2018                NA   21       11.40       74        1.9
## 294 09/11/2018                NA   22       11.20       75        1.7
## 295 09/11/2018                NA   23       10.90       76        1.2
##     VISIBILITY DEW_POINT_TEMPERATURE SOLAR_RADIATION RAINFALL SNOWFALL
SEASONS
## 1         1041                  11.3            0.00      0.0        0
Spring
## 2          886                  10.3            0.00      0.0        0
Spring
## 3          885                   9.3            0.00      0.0        0
Spring
## 4          687                   8.4            0.00      0.0        0
Spring
## 5          554                   7.4            0.00      0.0        0
Spring
## 6          447                   7.1            0.00      0.0        0
```

```
Spring
## 7          442                    6.7            0.00      0.0        0
Spring
## 8          438                    6.4            0.11      0.0        0
Spring
## 9          519                    5.8            0.68      0.0        0
Spring
## 10         975                    4.0            1.44      0.0        0
Spring
## 11        1487                    3.3            2.17      0.0        0
Spring
## 12        1468                    2.7            2.75      0.0        0
Spring
## 13        1132                    4.0            3.05      0.0        0
Spring
## 14        1558                    1.3            3.32      0.0        0
Spring
## 15        1804                   -2.1            3.16      0.0        0
Spring
## 16        1804                   -1.6            2.90      0.0        0
Spring
## 17        1938                   -4.4            2.39      0.0        0
Spring
## 18        2000                   -5.5            1.73      0.0        0
Spring
## 19        2000                   -7.0            0.94      0.0        0
Spring
## 20        2000                   -4.9            0.22      0.0        0
Spring
## 21        2000                   -3.8            0.00      0.0        0
Spring
## 22        2000                   -4.2            0.00      0.0        0
Spring
## 23        1923                   -2.0            0.00      0.0        0
Spring
## 24        1969                   -1.6            0.00      0.0        0
Spring
## 25        2000                    7.5            0.00      0.0        0
Spring
## 26        1960                    7.6            0.00      0.0        0
Spring
## 27        1895                    8.3            0.00      0.0        0
Spring
## 28        1698                    8.3            0.00      0.0        0
Spring
## 29        1351                    8.4            0.00      0.0        0
Spring
## 30         974                    8.1            0.00      0.0        0
Spring
## 31         885                    8.0            0.02      0.0        0
Spring
```

```
## 32         876                  6.9            0.34      0.0        0
Spring
## 33        1211                  6.5            1.01      0.0        0
Spring
## 34        1318                  6.4            1.76      0.0        0
Spring
## 35        1235                  7.3            2.38      0.0        0
Spring
## 36        1015                  7.9            2.84      0.0        0
Spring
## 37        1320                  8.7            3.17      0.0        0
Spring
## 38        1078                  8.6            3.26      0.0        0
Spring
## 39        1142                  9.4            3.22      0.0        0
Spring
## 40         946                  9.9            2.87      0.0        0
Spring
## 41        1154                 10.0            2.41      0.0        0
Spring
## 42        1075                  9.6            1.83      0.0        0
Spring
## 43         790                 10.3            1.05      0.0        0
Spring
## 44         675                 10.8            0.35      0.0        0
Spring
## 45         520                 10.5            0.02      0.0        0
Spring
## 46         460                 10.1            0.00      0.0        0
Spring
## 47         452                  9.8            0.00      0.0        0
Spring
## 48         369                  9.9            0.00      0.0        0
Spring
## 49        1916                 14.5            0.00      0.0        0
Autumn
## 50        1912                 13.9            0.00      0.0        0
Autumn
## 51        1866                 14.0            0.00      0.0        0
Autumn
## 52        1916                 14.3            0.00      0.0        0
Autumn
## 53        1906                 14.3            0.00      0.0        0
Autumn
## 54        1658                 13.8            0.00      0.0        0
Autumn
## 55        1757                 13.6            0.00      0.0        0
Autumn
## 56        1729                 13.5            0.08      0.0        0
Autumn
## 57        1501                 13.7            0.34      0.0        0
```

```
Autumn
## 58        1205                  13.7            1.13      0.0        0
Autumn
## 59        1619                  12.9            1.74      0.0        0
Autumn
## 60        1678                  11.7            2.03      0.0        0
Autumn
## 61        1809                  11.2            1.75      0.0        0
Autumn
## 62        1704                  10.9            2.18      0.0        0
Autumn
## 63        1749                  10.2            1.92      0.0        0
Autumn
## 64        1785                  12.5            1.62      0.0        0
Autumn
## 65        1953                   9.2            1.60      0.0        0
Autumn
## 66        1997                  10.1            0.92      0.0        0
Autumn
## 67        2000                  13.0            0.51      0.0        0
Autumn
## 68        2000                  14.2            0.03      0.0        0
Autumn
## 69        2000                  13.9            0.00      0.0        0
Autumn
## 70        2000                  14.2            0.00      0.0        0
Autumn
## 71        2000                  14.9            0.00      0.0        0
Autumn
## 72        1999                  15.2            0.00      0.0        0
Autumn
## 73        2000                  14.3            0.00      0.0        0
Autumn
## 74        2000                  13.4            0.00      0.0        0
Autumn
## 75        2000                  14.3            0.00      0.0        0
Autumn
## 76        2000                  14.0            0.00      0.0        0
Autumn
## 77        2000                  13.8            0.00      0.0        0
Autumn
## 78        1782                  14.3            0.00      0.0        0
Autumn
## 79        1488                  14.6            0.00      0.0        0
Autumn
## 80        1650                  14.2            0.03      0.0        0
Autumn
## 81        1534                  14.3            0.26      0.0        0
Autumn
## 82        1774                  13.6            0.92      0.0        0
Autumn
```

```
## 83        1819                14.4            1.49      0.0        0
Autumn
## 84        1723                14.0            1.01      0.0        0
Autumn
## 85        1619                14.0            1.56      0.0        0
Autumn
## 86        1560                11.4            2.15      0.0        0
Autumn
## 87        1840                11.4            1.21      0.0        0
Autumn
## 88        1836                10.8            0.90      0.0        0
Autumn
## 89        1594                11.3            0.65      0.0        0
Autumn
## 90        1883                11.8            0.36      0.0        0
Autumn
## 91        1691                14.3            0.12      0.0        0
Autumn
## 92        1836                14.7            0.00      0.0        0
Autumn
## 93        1737                16.1            0.00      0.0        0
Autumn
## 94        1078                14.3            0.00      0.1        0
Autumn
## 95        1057                15.4            0.00      0.1        0
Autumn
## 96        2000                11.5            0.00      0.0        0
Autumn
## 97        2000                 7.1            0.00      0.0        0
Autumn
## 98        2000                 6.8            0.00      0.0        0
Autumn
## 99        2000                 7.2            0.00      0.0        0
Autumn
## 100       2000                 7.3            0.00      0.0        0
Autumn
## 101       2000                 7.4            0.00      0.0        0
Autumn
## 102       2000                 6.8            0.00      0.0        0
Autumn
## 103       2000                 6.9            0.00      0.0        0
Autumn
## 104       2000                 7.2            0.03      0.0        0
Autumn
## 105       2000                 6.9            0.29      0.0        0
Autumn
## 106       2000                 6.0            0.66      0.0        0
Autumn
## 107       1989                 7.1            0.81      0.0        0
Autumn
## 108       2000                 8.4            0.73      0.0        0
```

```
Autumn
## 109       2000                  8.9            0.98      0.0        0
Autumn
## 110       2000                  9.2            0.76      0.0        0
Autumn
## 111       2000                  9.5            0.68      0.0        0
Autumn
## 112       2000                 10.1            1.15      0.0        0
Autumn
## 113       1995                  9.7            1.67      0.0        0
Autumn
## 114       2000                 11.4            0.79      0.0        0
Autumn
## 115       2000                 11.3            0.20      0.0        0
Autumn
## 116       2000                 11.5            0.01      0.0        0
Autumn
## 117       2000                 11.7            0.00      0.0        0
Autumn
## 118       2000                 11.3            0.00      0.0        0
Autumn
## 119       2000                 11.5            0.00      0.0        0
Autumn
## 120       2000                 10.8            0.00      0.0        0
Autumn
## 121       2000                 10.2            0.00      0.0        0
Autumn
## 122       2000                 11.0            0.00      0.0        0
Autumn
## 123       2000                 11.3            0.00      0.0        0
Autumn
## 124       2000                 11.4            0.00      0.0        0
Autumn
## 125       2000                 11.7            0.00      0.0        0
Autumn
## 126       2000                 12.2            0.00      0.0        0
Autumn
## 127       2000                 12.0            0.00      0.0        0
Autumn
## 128       1939                 11.8            0.03      0.0        0
Autumn
## 129       2000                 10.4            0.54      0.0        0
Autumn
## 130       1847                 11.0            1.00      0.0        0
Autumn
## 131       1679                 13.1            1.41      0.0        0
Autumn
## 132       1195                 12.9            1.25      0.0        0
Autumn
## 133       1665                  9.0            2.00      0.0        0
Autumn
```

```
## 134       1954                 7.6            2.74      0.0        0
Autumn
## 135       1999                 3.7            2.48      0.0        0
Autumn
## 136       2000                 3.9            2.31      0.0        0
Autumn
## 137       2000                 1.9            1.77      0.0        0
Autumn
## 138       2000                 2.6            1.06      0.0        0
Autumn
## 139       2000                 2.1            0.36      0.0        0
Autumn
## 140       2000                 1.4            0.00      0.0        0
Autumn
## 141       2000                 2.0            0.00      0.0        0
Autumn
## 142       2000                 1.6            0.00      0.0        0
Autumn
## 143       2000                 3.0            0.00      0.0        0
Autumn
## 144       2000                 4.6            0.00      0.0        0
Autumn
## 145       1987                 8.0            0.00      0.0        0
Autumn
## 146       1992                 7.9            0.00      0.0        0
Autumn
## 147       1840                 7.9            0.00      0.0        0
Autumn
## 148       1843                 8.0            0.00      0.0        0
Autumn
## 149       1236                 7.8            0.00      0.0        0
Autumn
## 150       1778                 7.8            0.00      0.0        0
Autumn
## 151       1798                 7.4            0.00      0.0        0
Autumn
## 152       1956                 7.2            0.02      0.0        0
Autumn
## 153       1714                 5.7            0.49      0.0        0
Autumn
## 154       1427                 7.6            1.11      0.0        0
Autumn
## 155       1554                 8.3            1.81      0.0        0
Autumn
## 156       1878                 5.8            2.35      0.0        0
Autumn
## 157       1956                 5.7            2.67      0.0        0
Autumn
## 158       1776                 4.8            2.71      0.0        0
Autumn
## 159       1691                 6.9            2.56      0.0        0
```

```
Autumn
## 160       1913                  6.8            2.22      0.0        0
Autumn
## 161       1929                  5.9            1.64      0.0        0
Autumn
## 162       2000                  7.0            0.98      0.0        0
Autumn
## 163       2000                  7.9            0.28      0.0        0
Autumn
## 164       2000                  8.9            0.00      0.0        0
Autumn
## 165       2000                  9.4            0.00      0.0        0
Autumn
## 166       2000                  9.6            0.00      0.0        0
Autumn
## 167       1987                  9.8            0.00      0.0        0
Autumn
## 168       1994                  8.9            0.00      0.0        0
Autumn
## 169       2000                  8.9            0.00      0.0        0
Autumn
## 170       2000                  9.0            0.00      0.0        0
Autumn
## 171       1992                  9.2            0.00      0.0        0
Autumn
## 172       1960                  9.2            0.00      0.0        0
Autumn
## 173       1792                  8.9            0.00      0.0        0
Autumn
## 174       1857                  8.4            0.00      0.0        0
Autumn
## 175       1900                  8.4            0.00      0.0        0
Autumn
## 176       1780                  8.8            0.01      0.0        0
Autumn
## 177       1912                  8.0            0.34      0.0        0
Autumn
## 178       1821                  8.3            1.22      0.0        0
Autumn
## 179       1980                  8.0            1.73      0.0        0
Autumn
## 180       1963                  7.5            2.36      0.0        0
Autumn
## 181       1990                  8.0            2.44      0.0        0
Autumn
## 182       2000                  8.6            1.72      0.0        0
Autumn
## 183       2000                  9.4            2.18      0.0        0
Autumn
## 184       2000                 10.3            1.88      0.0        0
Autumn
```

```
## 185       2000                 10.9            1.22      0.0        0
Autumn
## 186       2000                 11.1            0.58      0.0        0
Autumn
## 187       2000                 11.4            0.20      0.0        0
Autumn
## 188       2000                 11.3            0.00      0.0        0
Autumn
## 189       2000                 11.3            0.00      0.0        0
Autumn
## 190       2000                 11.6            0.00      0.0        0
Autumn
## 191       2000                 11.8            0.00      0.0        0
Autumn
## 192       2000                 11.9            0.00      0.0        0
Autumn
## 193       1744                 15.4            0.00      1.0        0
Autumn
## 194       1814                 14.8            0.00      1.0        0
Autumn
## 195       2000                 14.8            0.00      1.5        0
Autumn
## 196       1601                 15.1            0.00      2.0        0
Autumn
## 197       1577                 15.0            0.00      2.5        0
Autumn
## 198        360                 15.5            0.00      4.5        0
Autumn
## 199        417                 15.8            0.00      8.5        0
Autumn
## 200       2000                  3.5            0.00      0.0        0
Autumn
## 201       2000                  3.9            0.00      0.0        0
Autumn
## 202       2000                  5.0            0.00      0.0        0
Autumn
## 203       2000                  4.8            0.00      0.0        0
Autumn
## 204       2000                  4.2            0.00      0.0        0
Autumn
## 205       2000                  4.2            0.00      0.0        0
Autumn
## 206       2000                  5.3            0.00      0.0        0
Autumn
## 207       2000                  5.6            0.00      0.0        0
Autumn
## 208       1991                  5.2            0.18      0.0        0
Autumn
## 209       1838                  4.4            0.55      0.0        0
Autumn
## 210       1998                  4.7            1.07      0.0        0
```

```
Autumn
## 211       2000                 4.4            1.22      0.0        0
Autumn
## 212       1995                 4.4            1.82      0.0        0
Autumn
## 213       2000                 4.8            1.66      0.0        0
Autumn
## 214       2000                 4.5            0.67      0.0        0
Autumn
## 215       2000                 4.4            0.33      0.0        0
Autumn
## 216       2000                 4.7            0.89      0.0        0
Autumn
## 217       2000                 4.1            0.45      0.0        0
Autumn
## 218       2000                 4.9            0.05      0.0        0
Autumn
## 219       2000                 5.8            0.00      0.0        0
Autumn
## 220       2000                 6.1            0.00      0.0        0
Autumn
## 221       2000                 6.8            0.00      0.0        0
Autumn
## 222       2000                 7.6            0.00      0.0        0
Autumn
## 223       2000                 6.9            0.00      0.0        0
Autumn
## 224       1314                 2.1            0.00      0.0        0
Autumn
## 225       1273                 1.3            0.00      0.0        0
Autumn
## 226       1170                 2.0            0.00      0.0        0
Autumn
## 227       1229                 1.3            0.00      0.0        0
Autumn
## 228       1104                 1.6            0.00      0.0        0
Autumn
## 229       1053                 1.9            0.00      0.0        0
Autumn
## 230        983                 1.9            0.00      0.0        0
Autumn
## 231        894                 1.8            0.00      0.0        0
Autumn
## 232        905                 1.3            0.11      0.0        0
Autumn
## 233       1033                 0.1            0.80      0.0        0
Autumn
## 234        957                 0.6            1.40      0.0        0
Autumn
## 235        890                 1.7            1.85      0.0        0
Autumn
```

```
## 236       1060                1.7            2.14      0.0        0
Autumn
## 237       1328               -0.4            2.17      0.0        0
Autumn
## 238       1594               -2.8            1.97      0.0        0
Autumn
## 239       1724               -1.9            1.63      0.0        0
Autumn
## 240       1605               -1.7            1.07      0.0        0
Autumn
## 241        772                4.6            0.43      0.0        0
Autumn
## 242        990                4.1            0.02      0.0        0
Autumn
## 243        740                6.2            0.00      0.0        0
Autumn
## 244        592                6.3            0.00      0.0        0
Autumn
## 245        619                5.6            0.00      0.0        0
Autumn
## 246        791                4.3            0.00      0.0        0
Autumn
## 247       1064                2.9            0.00      0.0        0
Autumn
## 248        700                5.3            0.00      0.0        0
Autumn
## 249        668                5.5            0.00      0.0        0
Autumn
## 250        619                6.0            0.00      0.0        0
Autumn
## 251        500                6.2            0.00      0.0        0
Autumn
## 252        493                6.0            0.00      0.0        0
Autumn
## 253        420                6.2            0.00      0.0        0
Autumn
## 254        409                5.7            0.00      0.0        0
Autumn
## 255        422                5.7            0.00      0.0        0
Autumn
## 256        529                4.8            0.11      0.0        0
Autumn
## 257        578                4.4            0.33      0.0        0
Autumn
## 258        521                4.6            0.77      0.0        0
Autumn
## 259        488                4.1            0.84      0.0        0
Autumn
## 260        373                6.0            1.08      0.0        0
Autumn
## 261        328                7.3            1.54      0.0        0
```

```
Autumn
## 262        336                  6.8            1.52      0.0        0
Autumn
## 263        374                  6.9            0.76      0.0        0
Autumn
## 264        334                  7.5            0.66      0.0        0
Autumn
## 265        260                  8.2            0.23      0.0        0
Autumn
## 266        201                  8.8            0.01      0.0        0
Autumn
## 267        234                  8.3            0.00      0.0        0
Autumn
## 268        246                  8.4            0.00      0.0        0
Autumn
## 269        231                  8.8            0.00      0.0        0
Autumn
## 270        277                  7.5            0.00      0.0        0
Autumn
## 271        291                  7.6            0.00      0.0        0
Autumn
## 272       1185                 11.3            0.00     18.0        0
Autumn
## 273       1376                  9.6            0.00      0.0        0
Autumn
## 274       1680                  9.3            0.00      0.0        0
Autumn
## 275       1712                  7.8            0.00      0.5        0
Autumn
## 276        913                  6.3            0.00      0.0        0
Autumn
## 277       1034                  5.2            0.00      0.0        0
Autumn
## 278        900                  5.7            0.00      0.0        0
Autumn
## 279       1741                  7.2            0.00      0.0        0
Autumn
## 280       1809                  7.6            0.05      0.0        0
Autumn
## 281       1985                  7.1            0.13      0.5        0
Autumn
## 282       1984                  5.5            0.44      0.0        0
Autumn
## 283       1970                  5.4            1.43      0.0        0
Autumn
## 284       1964                  3.6            2.14      0.0        0
Autumn
## 285        927                  2.5            1.68      0.0        0
Autumn
## 286        518                  4.3            1.53      0.0        0
Autumn
```

```
## 287        622                    5.3            1.30      0.0        0
Autumn
## 288        858                    6.3            0.72      0.0        0
Autumn
## 289        682                    6.6            0.32      0.0        0
Autumn
## 290        623                    6.7            0.01      0.0        0
Autumn
## 291        589                    6.7            0.00      0.0        0
Autumn
## 292        526                    7.0            0.00      0.0        0
Autumn
## 293        498                    6.9            0.00      0.0        0
Autumn
## 294        478                    6.9            0.00      0.0        0
Autumn
## 295        456                    6.8            0.00      0.0        0
Autumn
##        HOLIDAY FUNCTIONING_DAY
## 1   No Holiday              No
## 2   No Holiday              No
## 3   No Holiday              No
## 4   No Holiday              No
## 5   No Holiday              No
## 6   No Holiday              No
## 7   No Holiday              No
## 8   No Holiday              No
## 9   No Holiday              No
## 10  No Holiday              No
## 11  No Holiday              No
## 12  No Holiday              No
## 13  No Holiday              No
## 14  No Holiday              No
## 15  No Holiday              No
## 16  No Holiday              No
## 17  No Holiday              No
## 18  No Holiday              No
## 19  No Holiday              No
## 20  No Holiday              No
## 21  No Holiday              No
## 22  No Holiday              No
## 23  No Holiday              No
## 24  No Holiday              No
## 25  No Holiday              No
## 26  No Holiday              No
## 27  No Holiday              No
## 28  No Holiday              No
## 29  No Holiday              No
## 30  No Holiday              No
## 31  No Holiday              No
## 32  No Holiday              No
```

```
## 33  No Holiday             No
## 34  No Holiday             No
## 35  No Holiday             No
## 36  No Holiday             No
## 37  No Holiday             No
## 38  No Holiday             No
## 39  No Holiday             No
## 40  No Holiday             No
## 41  No Holiday             No
## 42  No Holiday             No
## 43  No Holiday             No
## 44  No Holiday             No
## 45  No Holiday             No
## 46  No Holiday             No
## 47  No Holiday             No
## 48  No Holiday             No
## 49  No Holiday             No
## 50  No Holiday             No
## 51  No Holiday             No
## 52  No Holiday             No
## 53  No Holiday             No
## 54  No Holiday             No
## 55  No Holiday             No
## 56  No Holiday             No
## 57  No Holiday             No
## 58  No Holiday             No
## 59  No Holiday             No
## 60  No Holiday             No
## 61  No Holiday             No
## 62  No Holiday             No
## 63  No Holiday             No
## 64  No Holiday             No
## 65  No Holiday             No
## 66  No Holiday             No
## 67  No Holiday             No
## 68  No Holiday             No
## 69  No Holiday             No
## 70  No Holiday             No
## 71  No Holiday             No
## 72  No Holiday             No
## 73  No Holiday             No
## 74  No Holiday             No
## 75  No Holiday             No
## 76  No Holiday             No
## 77  No Holiday             No
## 78  No Holiday             No
## 79  No Holiday             No
## 80  No Holiday             No
## 81  No Holiday             No
## 82  No Holiday             No
## 83  No Holiday             No
```

```
## 84  No Holiday             No
## 85  No Holiday             No
## 86  No Holiday             No
## 87  No Holiday             No
## 88  No Holiday             No
## 89  No Holiday             No
## 90  No Holiday             No
## 91  No Holiday             No
## 92  No Holiday             No
## 93  No Holiday             No
## 94  No Holiday             No
## 95  No Holiday             No
## 96  No Holiday             No
## 97  No Holiday             No
## 98  No Holiday             No
## 99  No Holiday             No
## 100 No Holiday             No
## 101 No Holiday             No
## 102 No Holiday             No
## 103 No Holiday             No
## 104 No Holiday             No
## 105 No Holiday             No
## 106 No Holiday             No
## 107 No Holiday             No
## 108 No Holiday             No
## 109 No Holiday             No
## 110 No Holiday             No
## 111 No Holiday             No
## 112 No Holiday             No
## 113 No Holiday             No
## 114 No Holiday             No
## 115 No Holiday             No
## 116 No Holiday             No
## 117 No Holiday             No
## 118 No Holiday             No
## 119 No Holiday             No
## 120 No Holiday             No
## 121 No Holiday             No
## 122 No Holiday             No
## 123 No Holiday             No
## 124 No Holiday             No
## 125 No Holiday             No
## 126 No Holiday             No
## 127 No Holiday             No
## 128 No Holiday             No
## 129 No Holiday             No
## 130 No Holiday             No
## 131 No Holiday             No
## 132 No Holiday             No
## 133 No Holiday             No
## 134 No Holiday             No
```

```
## 135 No Holiday             No
## 136 No Holiday             No
## 137 No Holiday             No
## 138 No Holiday             No
## 139 No Holiday             No
## 140 No Holiday             No
## 141 No Holiday             No
## 142 No Holiday             No
## 143 No Holiday             No
## 144 No Holiday             No
## 145 No Holiday             No
## 146 No Holiday             No
## 147 No Holiday             No
## 148 No Holiday             No
## 149 No Holiday             No
## 150 No Holiday             No
## 151 No Holiday             No
## 152 No Holiday             No
## 153 No Holiday             No
## 154 No Holiday             No
## 155 No Holiday             No
## 156 No Holiday             No
## 157 No Holiday             No
## 158 No Holiday             No
## 159 No Holiday             No
## 160 No Holiday             No
## 161 No Holiday             No
## 162 No Holiday             No
## 163 No Holiday             No
## 164 No Holiday             No
## 165 No Holiday             No
## 166 No Holiday             No
## 167 No Holiday             No
## 168 No Holiday             No
## 169 No Holiday             No
## 170 No Holiday             No
## 171 No Holiday             No
## 172 No Holiday             No
## 173 No Holiday             No
## 174 No Holiday             No
## 175 No Holiday             No
## 176 No Holiday             No
## 177 No Holiday             No
## 178 No Holiday             No
## 179 No Holiday             No
## 180 No Holiday             No
## 181 No Holiday             No
## 182 No Holiday             No
## 183 No Holiday             No
## 184 No Holiday             No
## 185 No Holiday             No
```

```
## 186 No Holiday              No
## 187 No Holiday              No
## 188 No Holiday              No
## 189 No Holiday              No
## 190 No Holiday              No
## 191 No Holiday              No
## 192 No Holiday              No
## 193 No Holiday              No
## 194 No Holiday              No
## 195 No Holiday              No
## 196 No Holiday              No
## 197 No Holiday              No
## 198 No Holiday              No
## 199 No Holiday              No
## 200    Holiday              No
## 201    Holiday              No
## 202    Holiday              No
## 203    Holiday              No
## 204    Holiday              No
## 205    Holiday              No
## 206    Holiday              No
## 207    Holiday              No
## 208    Holiday              No
## 209    Holiday              No
## 210    Holiday              No
## 211    Holiday              No
## 212    Holiday              No
## 213    Holiday              No
## 214    Holiday              No
## 215    Holiday              No
## 216    Holiday              No
## 217    Holiday              No
## 218    Holiday              No
## 219    Holiday              No
## 220    Holiday              No
## 221    Holiday              No
## 222    Holiday              No
## 223    Holiday              No
## 224 No Holiday              No
## 225 No Holiday              No
## 226 No Holiday              No
## 227 No Holiday              No
## 228 No Holiday              No
## 229 No Holiday              No
## 230 No Holiday              No
## 231 No Holiday              No
## 232 No Holiday              No
## 233 No Holiday              No
## 234 No Holiday              No
## 235 No Holiday              No
## 236 No Holiday              No
```

```
## 237 No Holiday             No
## 238 No Holiday             No
## 239 No Holiday             No
## 240 No Holiday             No
## 241 No Holiday             No
## 242 No Holiday             No
## 243 No Holiday             No
## 244 No Holiday             No
## 245 No Holiday             No
## 246 No Holiday             No
## 247 No Holiday             No
## 248 No Holiday             No
## 249 No Holiday             No
## 250 No Holiday             No
## 251 No Holiday             No
## 252 No Holiday             No
## 253 No Holiday             No
## 254 No Holiday             No
## 255 No Holiday             No
## 256 No Holiday             No
## 257 No Holiday             No
## 258 No Holiday             No
## 259 No Holiday             No
## 260 No Holiday             No
## 261 No Holiday             No
## 262 No Holiday             No
## 263 No Holiday             No
## 264 No Holiday             No
## 265 No Holiday             No
## 266 No Holiday             No
## 267 No Holiday             No
## 268 No Holiday             No
## 269 No Holiday             No
## 270 No Holiday             No
## 271 No Holiday             No
## 272 No Holiday             No
## 273 No Holiday             No
## 274 No Holiday             No
## 275 No Holiday             No
## 276 No Holiday             No
## 277 No Holiday             No
## 278 No Holiday             No
## 279 No Holiday             No
## 280 No Holiday             No
## 281 No Holiday             No
## 282 No Holiday             No
## 283 No Holiday             No
## 284 No Holiday             No
## 285 No Holiday             No
## 286 No Holiday             No
## 287 No Holiday             No
```

```
## 288 No Holiday              No
## 289 No Holiday              No
## 290 No Holiday              No
## 291 No Holiday              No
## 292 No Holiday              No
## 293 No Holiday              No
## 294 No Holiday              No
## 295 No Holiday              No
```

calculate the average temperature in summer

```
summer_temp <- bike_sharing_df[bike_sharing_df$SEASONS == "Summer", ]
summer_avg_temp <- mean(summer_temp$TEMPERATURE, na.rm=TRUE)
print(summer_avg_temp)
```

```
## [1] 26.58771
```

Impute missing values for TEMPERATURE column with summer average temperature

```
bike_sharing_df["TEMPERATURE"][is.na(bike_sharing_df["TEMPERATURE"])] <-
summer_avg_temp
```

summary of the dataset

```
summary(bike_sharing_df)
```

```
##      DATE           RENTED_BIKE_COUNT      HOUR        TEMPERATURE
##  Length:8760        Min.   :   2.0    Min.   : 0.00   Min.   :-17.80
##  Class :character   1st Qu.: 214.0    1st Qu.: 5.75   1st Qu.:  3.50
##  Mode  :character   Median : 542.0    Median :11.50   Median : 13.70
##                     Mean   : 729.2    Mean   :11.50   Mean   : 12.88
##                     3rd Qu.:1084.0    3rd Qu.:17.25   3rd Qu.: 22.50
##                     Max.   :3556.0    Max.   :23.00   Max.   : 39.40
##                     NA's   :295
##     HUMIDITY       WIND_SPEED      VISIBILITY   DEW_POINT_TEMPERATURE
##  Min.   : 0.00   Min.   :0.000   Min.   :  27   Min.   :-30.600
##  1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 940   1st Qu.: -4.700
##  Median :57.00   Median :1.500   Median :1698   Median :  5.100
##  Mean   :58.23   Mean   :1.725   Mean   :1437   Mean   :  4.074
##  3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 14.800
##  Max.   :98.00   Max.   :7.400   Max.   :2000   Max.   : 27.200
##
##  SOLAR_RADIATION     RAINFALL          SNOWFALL         SEASONS
##  Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000   Length:8760
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   Class :character
##  Median :0.0100   Median : 0.0000   Median :0.00000   Mode  :character
##  Mean   :0.5691   Mean   : 0.1487   Mean   :0.07507
##  3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :3.5200   Max.   :35.0000   Max.   :8.80000
##
##    HOLIDAY          FUNCTIONING_DAY
##  Length:8760        Length:8760
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
```

```
##
##
##
```

Save the dataset as `seoul_bike_sharing.csv

```
write.csv(bike_sharing_df, "E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing.csv")
```

Task_9: Create indicator (dummy) variables for categorical variables

```
bike_sharing_df <- read.csv("E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing.csv")
```

Using mutate() function to convert HOUR column into character type

```
bike_sharing_df %>%
  mutate(HOUR=as.character(HOUR)) %>%  #convert HOUR to character because
it's from 0 to 23
head(10)
```

```
##     X       DATE RENTED_BIKE_COUNT HOUR TEMPERATURE HUMIDITY WIND_SPEED
## 1   1 01/12/2017               254    0        -5.2       37        2.2
## 2   2 01/12/2017               204    1        -5.5       38        0.8
## 3   3 01/12/2017               173    2        -6.0       39        1.0
## 4   4 01/12/2017               107    3        -6.2       40        0.9
## 5   5 01/12/2017                78    4        -6.0       36        2.3
## 6   6 01/12/2017               100    5        -6.4       37        1.5
## 7   7 01/12/2017               181    6        -6.6       35        1.3
## 8   8 01/12/2017               460    7        -7.4       38        0.9
## 9   9 01/12/2017               930    8        -7.6       37        1.1
## 10 10 01/12/2017               490    9        -6.5       27        0.5
##    VISIBILITY DEW_POINT_TEMPERATURE SOLAR_RADIATION RAINFALL SNOWFALL
SEASONS
## 1        2000                -17.6            0.00        0        0
Winter
## 2        2000                -17.6            0.00        0        0
Winter
## 3        2000                -17.7            0.00        0        0
Winter
## 4        2000                -17.6            0.00        0        0
Winter
## 5        2000                -18.6            0.00        0        0
Winter
## 6        2000                -18.7            0.00        0        0
Winter
## 7        2000                -19.5            0.00        0        0
Winter
## 8        2000                -19.3            0.00        0        0
Winter
## 9        2000                -19.8            0.01        0        0
Winter
## 10       1928                -22.4            0.23        0        0
Winter
##       HOLIDAY FUNCTIONING_DAY
```

```
## 1  No Holiday             Yes
## 2  No Holiday             Yes
## 3  No Holiday             Yes
## 4  No Holiday             Yes
## 5  No Holiday             Yes
## 6  No Holiday             Yes
## 7  No Holiday             Yes
## 8  No Holiday             Yes
## 9  No Holiday             Yes
## 10 No Holiday             Yes
```

install.packages("pacman")

```
library(pacman)

pacman:: p_load(fastDummies)
```

Convert SEASONS, HOLIDAY, FUNCTIONING_DAY, and HOUR columns into indicator columns.

```
bike_sharing_df <- dummy_cols(bike_sharing_df, select_columns = "HOUR")
bike_sharing_df <- dummy_cols(bike_sharing_df, select_columns = "HOLIDAY")
bike_sharing_df <- dummy_cols(bike_sharing_df, select_columns = "SEASONS")

#Change the colnames for shorterning
colnames(bike_sharing_df)[c(40,41,42,43,44,45)] <- c("HOLIDAY", "NO HOLIDAY",
"AUTUMN", "SPRING", "SUMMER","WINTER")
```

Save the dataset as seoul_bike_sharing_converted.csv

```
write.csv(bike_sharing_df, "E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing_converted.csv")
```

Task_10: Normalize data using Min-Max normalization

```
minmax_norm <- function(x){
  (x-min(x))/(max(x)-min(x))}

bike_sharing_df <- read.csv("E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing_converted.csv")

#Apply min-max normalization function to numerical columns in df

bike_sharing_df %<>%
  mutate(TEMPERATURE = minmax_norm(TEMPERATURE),
         HUMIDITY = minmax_norm(HUMIDITY),
         WIND_SPEED = minmax_norm(WIND_SPEED),
         VISIBILITY = minmax_norm(VISIBILITY),
         DEW_POINT_TEMPERATURE = minmax_norm(DEW_POINT_TEMPERATURE),
         SOLAR_RADIATION = minmax_norm(SOLAR_RADIATION),
         RAINFALL = minmax_norm(RAINFALL),
         SNOWFALL = minmax_norm(SNOWFALL))
head(bike_sharing_df)
```

```
##   X.1 X       DATE RENTED_BIKE_COUNT HOUR TEMPERATURE  HUMIDITY WIND_SPEED
## 1   1 1 01/12/2017               254    0   0.2202797 0.3775510  0.2972973
```

```
## 2   2 2 01/12/2017               204    1   0.2150350 0.3877551  0.1081081
## 3   3 3 01/12/2017               173    2   0.2062937 0.3979592  0.1351351
## 4   4 4 01/12/2017               107    3   0.2027972 0.4081633  0.1216216
## 5   5 5 01/12/2017                78    4   0.2062937 0.3673469  0.3108108
## 6   6 6 01/12/2017               100    5   0.1993007 0.3775510  0.2027027
##   VISIBILITY DEW_POINT_TEMPERATURE SOLAR_RADIATION RAINFALL SNOWFALL
SEASONS
## 1          1             0.2249135              0        0        0
Winter
## 2          1             0.2249135              0        0        0
Winter
## 3          1             0.2231834              0        0        0
Winter
## 4          1             0.2249135              0        0        0
Winter
## 5          1             0.2076125              0        0        0
Winter
## 6          1             0.2058824              0        0        0
Winter
##      HOLIDAY FUNCTIONING_DAY HOUR_0 HOUR_1 HOUR_2 HOUR_3 HOUR_4 HOUR_5
HOUR_6
## 1 No Holiday             Yes      1      0      0      0      0      0
0
## 2 No Holiday             Yes      0      1      0      0      0      0
0
## 3 No Holiday             Yes      0      0      1      0      0      0
0
## 4 No Holiday             Yes      0      0      0      1      0      0
0
## 5 No Holiday             Yes      0      0      0      0      1      0
0
## 6 No Holiday             Yes      0      0      0      0      0      1
0
##   HOUR_7 HOUR_8 HOUR_9 HOUR_10 HOUR_11 HOUR_12 HOUR_13 HOUR_14 HOUR_15
HOUR_16
## 1      0      0      0       0       0       0       0       0       0
0
## 2      0      0      0       0       0       0       0       0       0
0
## 3      0      0      0       0       0       0       0       0       0
0
## 4      0      0      0       0       0       0       0       0       0
0
## 5      0      0      0       0       0       0       0       0       0
0
## 6      0      0      0       0       0       0       0       0       0
0
##   HOUR_17 HOUR_18 HOUR_19 HOUR_20 HOUR_21 HOUR_22 HOUR_23 HOLIDAY.1
NO.HOLIDAY
## 1       0       0       0       0       0       0       0         0
1
```

```
## 2       0       0       0       0       0       0       0         0
1
## 3       0       0       0       0       0       0       0         0
1
## 4       0       0       0       0       0       0       0         0
1
## 5       0       0       0       0       0       0       0         0
1
## 6       0       0       0       0       0       0       0         0
1
##   AUTUMN SPRING SUMMER WINTER
## 1      0      0      0      1
## 2      0      0      0      1
## 3      0      0      0      1
## 4      0      0      0      1
## 5      0      0      0      1
## 6      0      0      0      1
```

Save the dataset as `seoul_bike_sharing_converted_normalized.csv

```
write.csv(bike_sharing_df,"E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing_converted_normalized.csv")
```

Standardize the column names again for the new datasets

```
dataset_list <- c('seoul_bike_sharing.csv',
'seoul_bike_sharing_converted.csv',
'seoul_bike_sharing_converted_normalized.csv')

for (dataset_name in dataset_list) {
  dataset <- read.csv(dataset_name)
  names(dataset) <- toupper(names(dataset))
  names(dataset) <- str_replace_all(names(dataset), " ", "_")
  write.csv(dataset, dataset_name, row.names = FALSE)
}
```

Exploratory Data Analysis with tidyverse and ggplot2

Task_11: Load the dataset

```
seoul_bike_sharing <- read.csv("E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing.csv")

str(seoul_bike_sharing)
```

```
## 'data.frame':    8760 obs. of  15 variables:
##  $ X                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ DATE                 : chr  "01/12/2017" "01/12/2017" "01/12/2017"
"01/12/2017" ...
##  $ RENTED_BIKE_COUNT    : int  254 204 173 107 78 100 181 460 930 490 ...
##  $ HOUR                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ TEMPERATURE          : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -7.6 -
6.5 ...
##  $ HUMIDITY             : int  37 38 39 40 36 37 35 38 37 27 ...
```

```
##  $ WIND_SPEED           : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5 ...
##  $ VISIBILITY           : int  2000 2000 2000 2000 2000 2000 2000 2000
2000 1928 ...
##  $ DEW_POINT_TEMPERATURE: num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -19.5 -
19.3 -19.8 -22.4 ...
##  $ SOLAR_RADIATION      : num  0 0 0 0 0 0 0 0.01 0.23 ...
##  $ RAINFALL             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SNOWFALL             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SEASONS              : chr  "Winter" "Winter" "Winter" "Winter" ...
##  $ HOLIDAY              : chr  "No Holiday" "No Holiday" "No Holiday" "No
Holiday" ...
##  $ FUNCTIONING_DAY      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

Task_12: Recast DATE as a date

```
seoul_bike_sharing$DATE = as.Date(seoul_bike_sharing$DATE,format="%d/%m/%Y")
#recast date as a date format

seoul_bike_sharing$HOUR <- factor(seoul_bike_sharing$HOUR, levels = 0:23,
ordered = TRUE) #cast the HOUR as categorical variables

seoul_bike_sharing$SEASONS <- factor(seoul_bike_sharing$SEASONS,
levels=c("Winter", "Spring", "Summer","Autumn"))

class(seoul_bike_sharing$HOUR)

## [1] "ordered" "factor"
```

Task_13 - Cast HOURS as a categorical variable

```
class(seoul_bike_sharing$DATE)

## [1] "Date"

class(seoul_bike_sharing$SEASONS)

## [1] "factor"

sum(is.na(seoul_bike_sharing))

## [1] 295
```

Descriptive Statistics

Task_14: Dataset Summary

```
summary(seoul_bike_sharing)

##        X             DATE            RENTED_BIKE_COUNT      HOUR
##  Min.   :   1   Min.   :2017-12-01   Min.   :   2.0    0      : 365
##  1st Qu.:2191   1st Qu.:2018-03-02   1st Qu.: 214.0    1      : 365
##  Median :4380   Median :2018-06-01   Median : 542.0    2      : 365
##  Mean   :4380   Mean   :2018-06-01   Mean   : 729.2    3      : 365
##  3rd Qu.:6570   3rd Qu.:2018-08-31   3rd Qu.:1084.0    4      : 365
##  Max.   :8760   Max.   :2018-11-30   Max.   :3556.0    5      : 365
##                                      NA's   :295       (Other):6570
```

```
##   TEMPERATURE        HUMIDITY       WIND_SPEED      VISIBILITY
##  Min.   :-17.80   Min.   : 0.00   Min.   :0.000   Min.   :  27
##  1st Qu.:  3.50   1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 940
##  Median : 13.70   Median :57.00   Median :1.500   Median :1698
##  Mean   : 12.88   Mean   :58.23   Mean   :1.725   Mean   :1437
##  3rd Qu.: 22.50   3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000
##  Max.   : 39.40   Max.   :98.00   Max.   :7.400   Max.   :2000
##
##  DEW_POINT_TEMPERATURE SOLAR_RADIATION     RAINFALL          SNOWFALL
##  Min.   :-30.600       Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000
##  1st Qu.: -4.700       1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000
##  Median :  5.100       Median :0.0100   Median : 0.0000   Median :0.00000
##  Mean   :  4.074       Mean   :0.5691   Mean   : 0.1487   Mean   :0.07507
##  3rd Qu.: 14.800       3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   : 27.200       Max.   :3.5200   Max.   :35.0000   Max.   :8.80000
##
##    SEASONS       HOLIDAY          FUNCTIONING_DAY
##  Winter:2160   Length:8760        Length:8760
##  Spring:2208   Class :character   Class :character
##  Summer:2208   Mode  :character   Mode  :character
##  Autumn:2184
##
##
##
```

Task_15: Based on the above stats, calculate how many Holidays there are.

```
holiday_count <- table(seoul_bike_sharing$HOLIDAY)
num_holiday <- holiday_count['Holiday']
num_holiday
```

```
## Holiday
##     432
```

Task_16: Calculate the percentage of records that fall on a holiday.

```
num_holiday/(num_holiday +holiday_count['No Holiday'])
```

```
##    Holiday
## 0.04931507
```

Task_17: Given there is exactly a full year of data, determine how many records we expect to have.

```
# Define the frequency of data recording (e.g., daily)
data_frequency <- "daily"

# Define the number of days in a year
days_in_year <- 365

# Calculate the number of records in a year
if (data_frequency == "daily") {
  num_records_in_year <- days_in_year
} else if (data_frequency == "hourly") {
```

```
  # If recorded hourly, calculate records per day and multiply by days in a
year
  records_per_day <- 24  # 24 records per day if hourly
  num_records_in_year <- records_per_day * days_in_year
} else {
  # Add additional conditions for other data frequencies if needed
  print("Unsupported data frequency")
}

# Print the result
print(paste("Number of records in a year:", num_records_in_year))
```

```
## [1] "Number of records in a year: 365"
```

Task_18: Given the observations for the 'FUNCTIONING_DAY' how many records must there be?

```
# Assuming you have a data frame named 'your_data' with a 'FUNCTIONING_DAY'
column
# Count the unique values of 'FUNCTIONING_DAY'
unique_functioning_days <- unique(seoul_bike_sharing$FUNCTIONING_DAY)

# Calculate the number of records based on unique functioning days
num_records <- length(unique_functioning_days)

# Print the result
print(paste("Number of records based on 'FUNCTIONING_DAY':", num_records))
```

```
## [1] "Number of records based on 'FUNCTIONING_DAY': 2"
```

Task_19: Calculate the seasonal total rainfall and snowfall.

```
seasonal_total <- seoul_bike_sharing %>%
  group_by(SEASONS) %>%
  summarize(total_rainfall=sum(RAINFALL), total_snowfall=sum(SNOWFALL))
seasonal_total
```

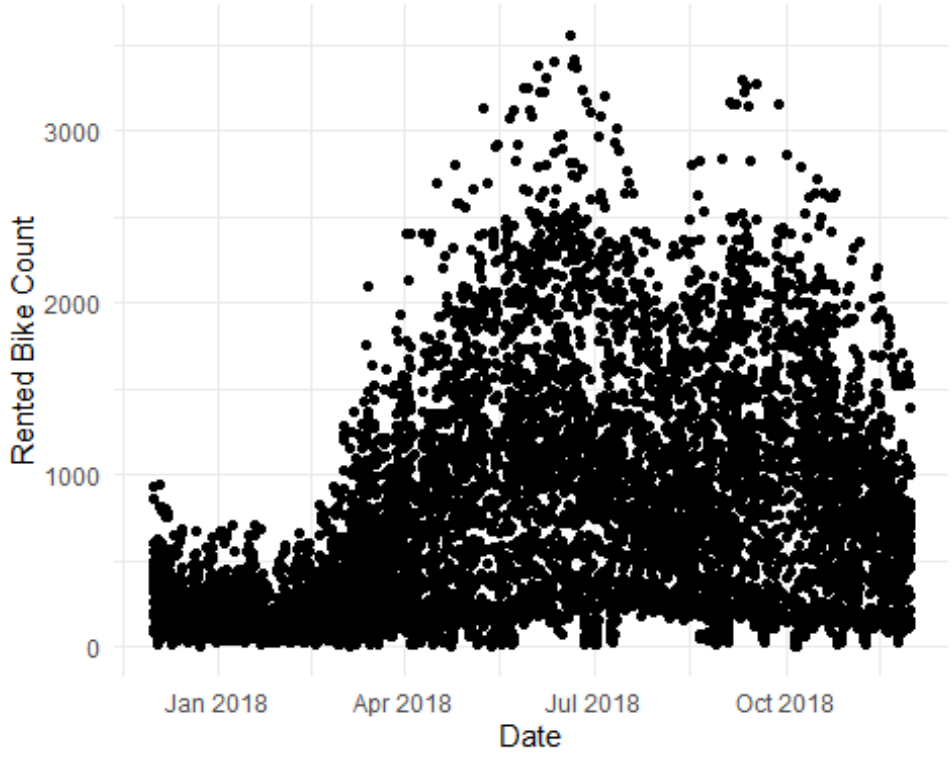
```
## # A tibble: 4 × 3
##   SEASONS total_rainfall total_snowfall
##   <fct>            <dbl>          <dbl>
## 1 Winter            70.9           535.
## 2 Spring           404.              0
## 3 Summer           560.              0
## 4 Autumn           268.            123
```

Data Visualization

Task_20: Create a scatter plot of RENTED_BIKE_COUNT vs DATE

```
ggplot(seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT)) +
  geom_point() +
  labs(x = "Date", y = "Rented Bike Count") +
  theme_minimal()
```

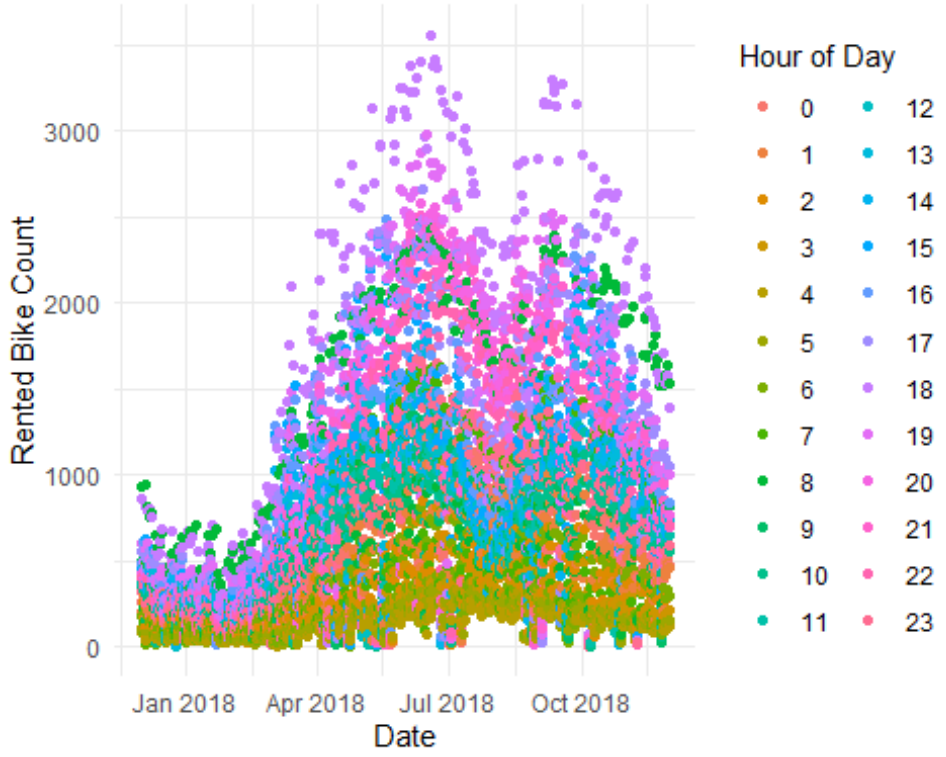
```
## Warning: Removed 295 rows containing missing values (`geom_point()`).
```



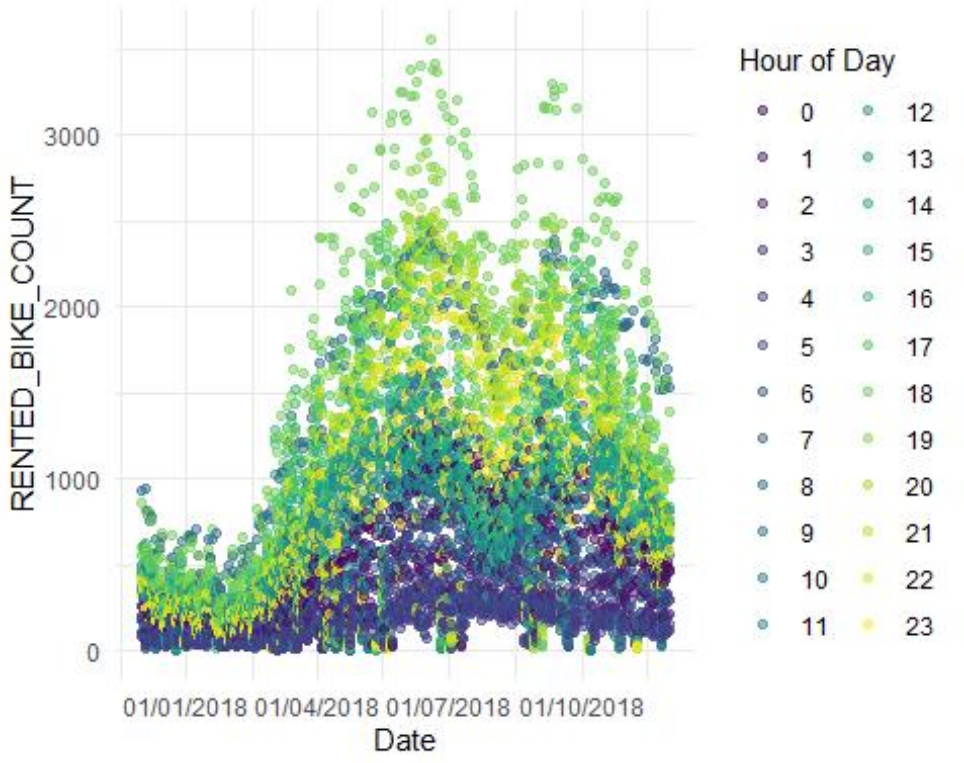
Task_21: Create the same plot of the RENTED_BIKE_COUNT time series, but now add HOURS as the colour.

```
seoul_bike_sharing %>%
  ggplot(aes(x = DATE, y = RENTED_BIKE_COUNT, color = factor(HOUR))) +
  geom_point() +
  labs(x = "Date", y = "Rented Bike Count", color = "Hour of Day") +
  scale_color_discrete() +
  theme_minimal()
```

```
## Warning: Removed 295 rows containing missing values (`geom_point()`).
```



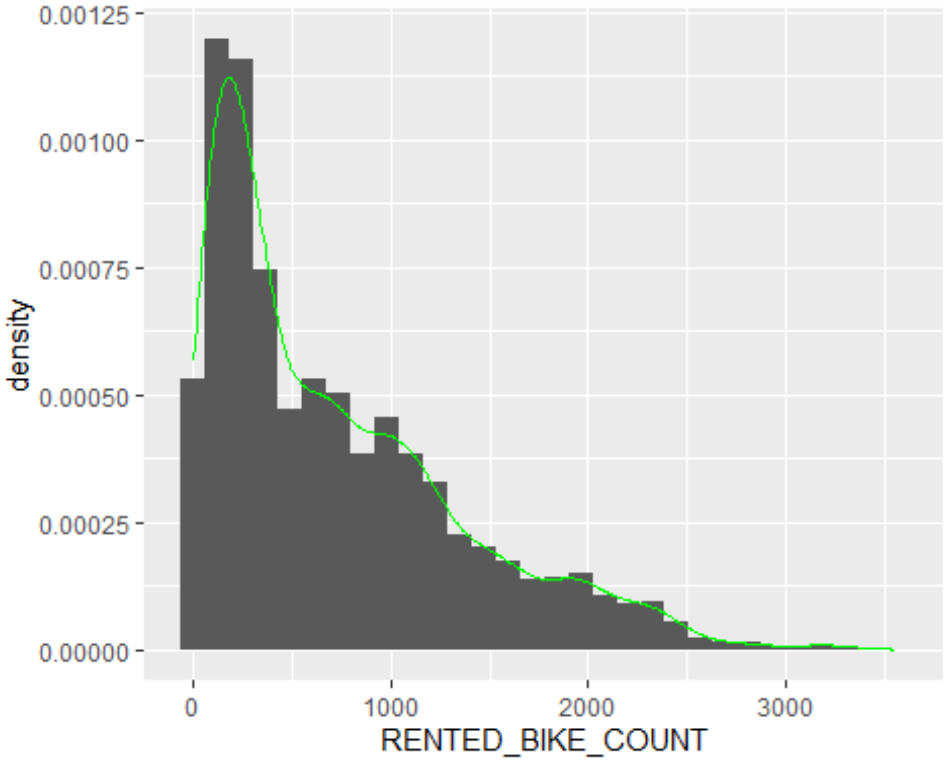

```
seoul_bike_sharing %>%
  mutate(DATE = as.Date(DATE, format = "%d/%m/%Y")) %>%
  ggplot(aes(x = DATE, y = RENTED_BIKE_COUNT, color = factor(HOUR))) +
  geom_point(alpha = 0.5) +
  scale_x_date(date_labels = "%d/%m/%Y") +
  labs(x = "Date", color = "Hour of Day") +
  theme_minimal()

## Warning: Removed 295 rows containing missing values (`geom_point()`).
```



Task_22: Create a histogram overlaid with a kernel density curve

```
ggplot(seoul_bike_sharing, aes(RENTED_BIKE_COUNT)) +
  geom_histogram(aes(y=..density..))+
  geom_density(col="green")
```

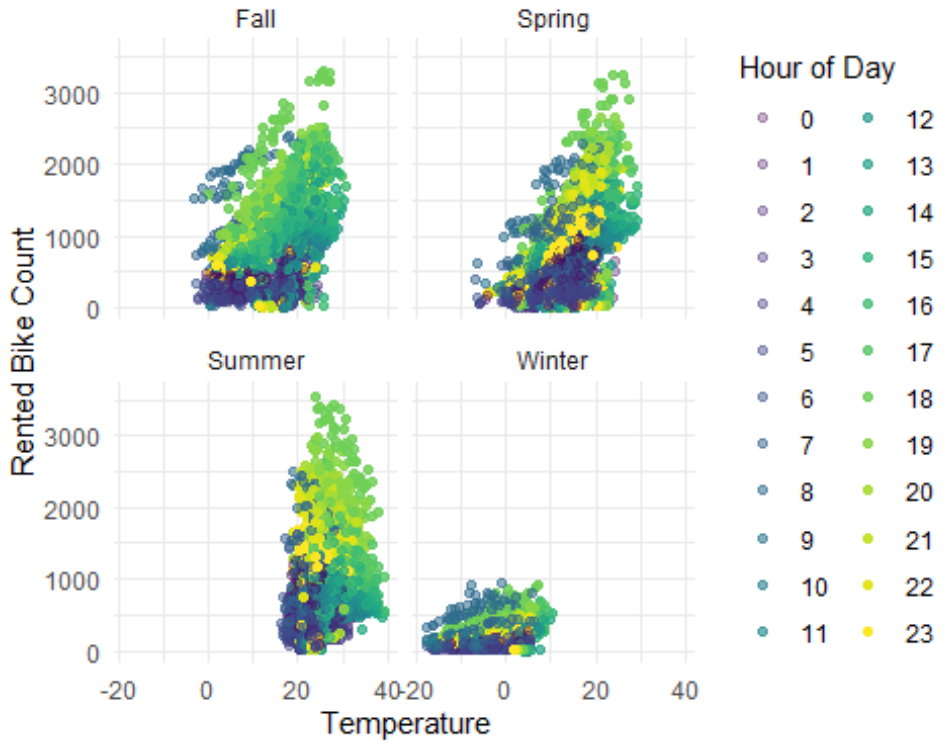
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2
3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

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

## Warning: Removed 295 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 295 rows containing non-finite values (`stat_density()`).
```



Task_23: Use a scatter plot to visualize the correlation between RENTED_BIKE_COUNT and TEMPERATURE by SEASONS

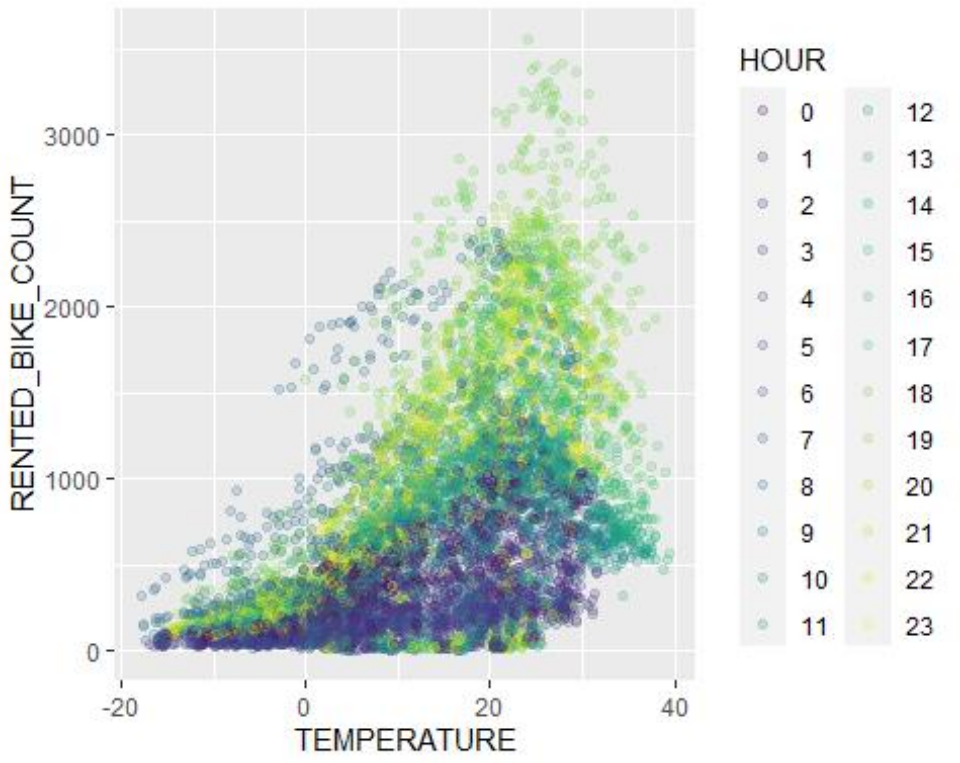
```
get_season <- function(date) {
  month <- as.POSIXlt(date)$mon + 1  # Extract the month from the date
  case_when(
    month %in% c(3, 4, 5) ~ "Spring",
    month %in% c(6, 7, 8) ~ "Summer",
    month %in% c(9, 10, 11) ~ "Fall",
    TRUE ~ "Winter"
  )
}

# Create a new column "SEASON" based on the date
seoul_bike_sharing <- seoul_bike_sharing %>%
  mutate(SEASON = get_season(DATE))

# Create the scatter plot with facet_wrap
ggplot(seoul_bike_sharing, aes(x = TEMPERATURE, y = RENTED_BIKE_COUNT, color
= as.factor(HOUR), alpha = as.factor(HOUR))) +
  geom_point() +
  facet_wrap(~SEASON, nrow = 2) +
  labs(x = "Temperature", y = "Rented Bike Count", color = "Hour of Day",
alpha = "Hour of Day") +
  scale_alpha_discrete(range = c(0.3, 1)) +  # Adjust opacity range for
discrete scale
  theme_minimal()
```

```
## Warning: Using alpha for a discrete variable is not advised.

## Warning: Removed 295 rows containing missing values (`geom_point()`).
```



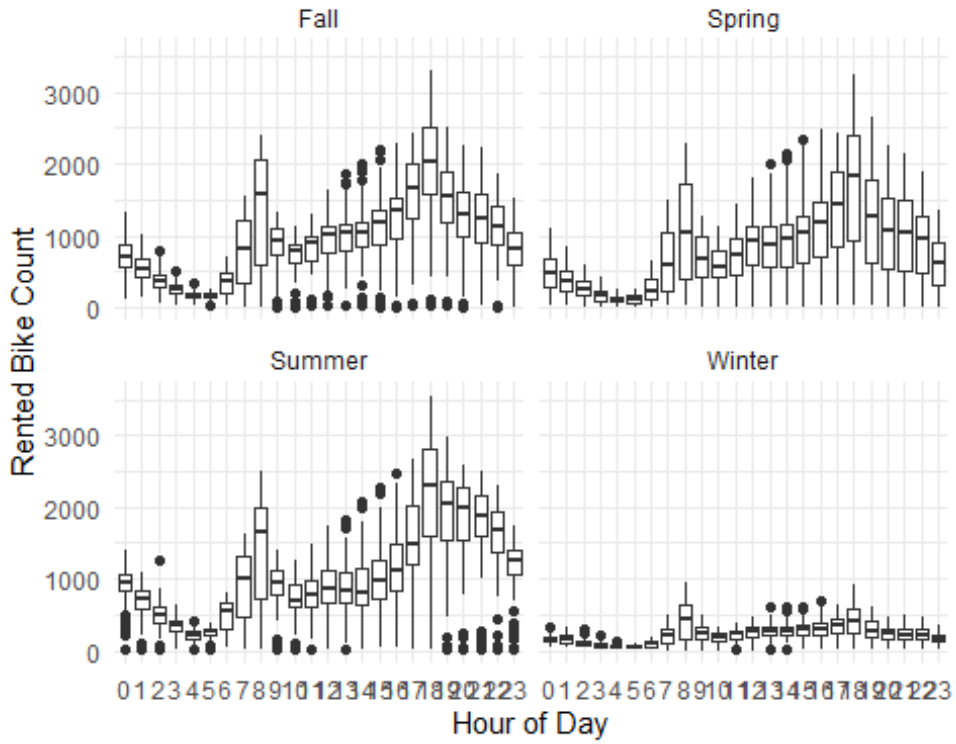
Comparing this plot to the same plot below, but without grouping by SEASONS, shows how important seasonality is in explaining bike rental counts.

```
ggplot(seoul_bike_sharing) +
   geom_point(aes(x=TEMPERATURE,y=RENTED_BIKE_COUNT,colour=HOUR),alpha=1/5)

## Warning: Removed 295 rows containing missing values (`geom_point()`).
```



Task_24: Create a display of four boxplots of RENTED_BIKE_COUNT vs. HOUR grouped by SEASONS

```
# Create a new column "SEASON" based on the date
seoul_bike_sharing <- seoul_bike_sharing %>%
  mutate(MONTH = as.integer(format(DATE, "%m"))) %>%
  mutate(SEASON = case_when(
    MONTH %in% 3:5 ~ "Spring",
    MONTH %in% 6:8 ~ "Summer",
    MONTH %in% 9:11 ~ "Fall",
    TRUE ~ "Winter"
  )) %>%
  select(-MONTH)  # Remove the temporary MONTH column

# Create boxplots grouped by SEASON and faceted by SEASON
ggplot(seoul_bike_sharing, aes(x = HOUR, y = RENTED_BIKE_COUNT)) +
  geom_boxplot() +
  labs(x = "Hour of Day", y = "Rented Bike Count") +
  facet_wrap(~SEASON, nrow = 2) +
  theme_minimal()

## Warning: Removed 295 rows containing non-finite values (`stat_boxplot()`).
```



Task_25: Group the data by DATE, and use the summarize() function to calculate the daily total rainfall and snowfall.

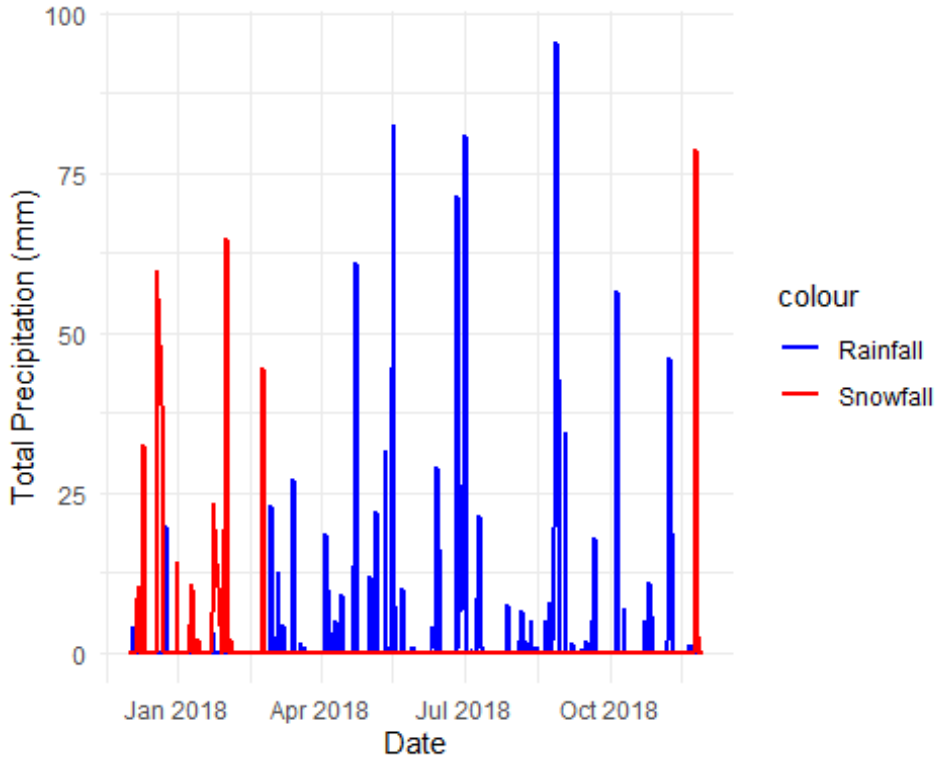
```
# Group data by DATE and calculate daily total rainfall and snowfall
daily_weather_totals <- seoul_bike_sharing %>%
  group_by(DATE) %>%
  summarize(TotalRainfall = sum(RAINFALL, na.rm = TRUE),
            TotalSnowfall = sum(SNOWFALL, na.rm = TRUE))

# Print the resulting daily weather totals
print(daily_weather_totals)
```

```
## # A tibble: 365 × 3
##    DATE       TotalRainfall TotalSnowfall
##    <date>             <dbl>         <dbl>
##  1 2017-12-01           0             0
##  2 2017-12-02           0             0
##  3 2017-12-03           4             0
##  4 2017-12-04           0.1           0
##  5 2017-12-05           0             0
##  6 2017-12-06           1.3           8.6
##  7 2017-12-07           0            10.4
##  8 2017-12-08           0             0
##  9 2017-12-09           0             0
## 10 2017-12-10           4.1          32.5
## # ℹ 355 more rows
```

```
# Create a plot of daily total rainfall and snowfall
ggplot(daily_weather_totals, aes(x = DATE)) +
```

```
  geom_line(aes(y = TotalRainfall, color = "Rainfall"), size = 1) +
  geom_line(aes(y = TotalSnowfall, color = "Snowfall"), size = 1) +
  labs(x = "Date", y = "Total Precipitation (mm)") +
  scale_color_manual(values = c("Rainfall" = "blue", "Snowfall" = "red")) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Task_26: Determine how many days had snowfall

```
# Count the number of days with snowfall
days_with_snowfall <- sum(seoul_bike_sharing$SNOWFALL > 0)

# Print the result
print(paste("Number of days with snowfall:", days_with_snowfall))
```

```
## [1] "Number of days with snowfall: 443"
```

Predict Hourly Rented Bike Count using Basic Linear Regression Models

install.packages("rlang")

install.packages("stringr")

install.packages("broom")

```
library(rlang)

##
## Attaching package: 'rlang'

## The following object is masked from 'package:magrittr':
##
##     set_names

## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice

library(stringr)

library(broom)
```

The seoul_bike_sharing_converted_normalized.csv will be our main dataset

```
bike_sharing_df <- read.csv("E:/Coursera/Data Science with R_Capstone
Project/seoul_bike_sharing_converted_normalized.csv")
```

We won't be using the DATE column, because 'as is', it basically acts like an data entry index. (However, given more time, we could use the DATE colum to create a 'day of week' or 'isWeekend' column, which we might expect has an affect on preferred bike rental times.) We also do not need the FUNCTIONAL DAY column because it only has one distinct value remaining (YES) after missing value processing.

```
bike_sharing_df <- bike_sharing_df %>%
  select(-DATE, -FUNCTIONING_DAY, -X.2, -X.1, -X, -HOUR, -SEASONS, -HOLIDAY)

colnames(bike_sharing_df)[c(34,35)] <- c("HOLIDAY", "NO_HOLIDAY")
```

Task_27: Split training and testing data

```
set.seed(1234)
data_split <- initial_split(bike_sharing_df, prop = 3/4) #set the training
dataset with 75% of the original dataset
bike_train <- training(data_split)
bike_test <- testing(data_split)
```

Task_28: Build a linear regression model using weather variables only

```
lm_model_weather <- lm(RENTED_BIKE_COUNT ~ TEMPERATURE + HUMIDITY +
WIND_SPEED + VISIBILITY + DEW_POINT_TEMPERATURE + SOLAR_RADIATION + RAINFALL
+ SNOWFALL,
                       data=bike_train)
summary(lm_model_weather)

##
## Call:
## lm(formula = RENTED_BIKE_COUNT ~ TEMPERATURE + HUMIDITY + WIND_SPEED +
##     VISIBILITY + DEW_POINT_TEMPERATURE + SOLAR_RADIATION + RAINFALL +
##     SNOWFALL, data = bike_train)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1337.71  -297.09   -59.12   211.28  2334.57
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                180.67      59.16   3.054  0.00227 **
## TEMPERATURE               2328.91     268.89   8.661  < 2e-16 ***
## HUMIDITY                  -967.29     130.37  -7.419 1.33e-13 ***
## WIND_SPEED                 391.29      47.76   8.193 3.05e-16 ***
## VISIBILITY                  19.69      24.82   0.793  0.42768
## DEW_POINT_TEMPERATURE     -250.91     286.97  -0.874  0.38196
## SOLAR_RADIATION           -444.43      34.54 -12.867  < 2e-16 ***
## RAINFALL                 -1824.26     193.90  -9.408  < 2e-16 ***
## SNOWFALL                   321.63     124.08   2.592  0.00956 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 485.2 on 6342 degrees of freedom
##   (219 observations deleted due to missingness)
## Multiple R-squared:  0.4319, Adjusted R-squared:  0.4312
## F-statistic: 602.7 on 8 and 6342 DF,  p-value: < 2.2e-16
```

Task_29:TASK: Build a linear regression model using all variables

```
lm_model_all <- lm(RENTED_BIKE_COUNT ~ .,
                       data=bike_train)
summary(lm_model_all)
```

```
##
## Call:
## lm(formula = RENTED_BIKE_COUNT ~ ., data = bike_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1345.07  -219.83   -11.43   200.62  1810.03
##
## Coefficients: (3 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               336.1282    52.6423   6.385 1.83e-10 ***
## TEMPERATURE               675.8427   217.1794   3.112 0.001867 **
## HUMIDITY                 -962.5891   102.1449  -9.424  < 2e-16 ***
## WIND_SPEED                 -0.1969    39.9048  -0.005 0.996063
## VISIBILITY                 35.2766    20.0750   1.757 0.078925 .
## DEW_POINT_TEMPERATURE     764.1568   227.3129   3.362 0.000779 ***
## SOLAR_RADIATION           267.0582    41.3751   6.455 1.17e-10 ***
## RAINFALL                 -2093.4367   152.0538 -13.768  < 2e-16 ***
## SNOWFALL                  269.9654    97.2871   2.775 0.005538 **
## HOUR_0                   -132.2530    32.5927  -4.058 5.01e-05 ***
## HOUR_1                   -230.6053    32.5206  -7.091 1.48e-12 ***
## HOUR_2                   -357.3816    32.7219 -10.922  < 2e-16 ***
```

```
## HOUR_3                  -436.8099    32.7899 -13.321  < 2e-16 ***
## HOUR_4                  -498.0917    32.7066 -15.229  < 2e-16 ***
## HOUR_5                  -469.3047    32.8406 -14.290  < 2e-16 ***
## HOUR_6                  -320.1232    32.8418  -9.747  < 2e-16 ***
## HOUR_7                    -7.5412    33.2445  -0.227 0.820554
## HOUR_8                   350.0599    33.1858  10.549  < 2e-16 ***
## HOUR_9                  -100.8611    33.4686  -3.014 0.002592 **
## HOUR_10                 -344.0115    34.9835  -9.834  < 2e-16 ***
## HOUR_11                 -348.8339    36.7342  -9.496  < 2e-16 ***
## HOUR_12                 -312.4168    36.9706  -8.450  < 2e-16 ***
## HOUR_13                 -304.0978    38.7014  -7.858 4.57e-15 ***
## HOUR_14                 -299.7209    36.9777  -8.105 6.26e-16 ***
## HOUR_15                 -216.5520    36.4353  -5.943 2.94e-09 ***
## HOUR_16                  -73.5134    34.9760  -2.102 0.035609 *
## HOUR_17                  213.5063    34.1330   6.255 4.23e-10 ***
## HOUR_18                  682.0979    33.0801  20.620  < 2e-16 ***
## HOUR_19                  402.5790    32.7585  12.289  < 2e-16 ***
## HOUR_20                  327.9847    32.6779  10.037  < 2e-16 ***
## HOUR_21                  322.5391    32.8277   9.825  < 2e-16 ***
## HOUR_22                  202.1670    32.7169   6.179 6.84e-10 ***
## HOUR_23                        NA         NA      NA       NA
## HOLIDAY                 -105.8741    22.3364  -4.740 2.18e-06 ***
## NO_HOLIDAY                     NA         NA      NA       NA
## AUTUMN                   342.0001    19.9677  17.128  < 2e-16 ***
## SPRING                   186.2172    19.1625   9.718  < 2e-16 ***
## SUMMER                   167.0077    28.9383   5.771 8.25e-09 ***
## WINTER                         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 375.3 on 6315 degrees of freedom
##   (219 observations deleted due to missingness)
## Multiple R-squared:  0.6616, Adjusted R-squared:  0.6597
## F-statistic: 352.7 on 35 and 6315 DF,  p-value: < 2.2e-16
```

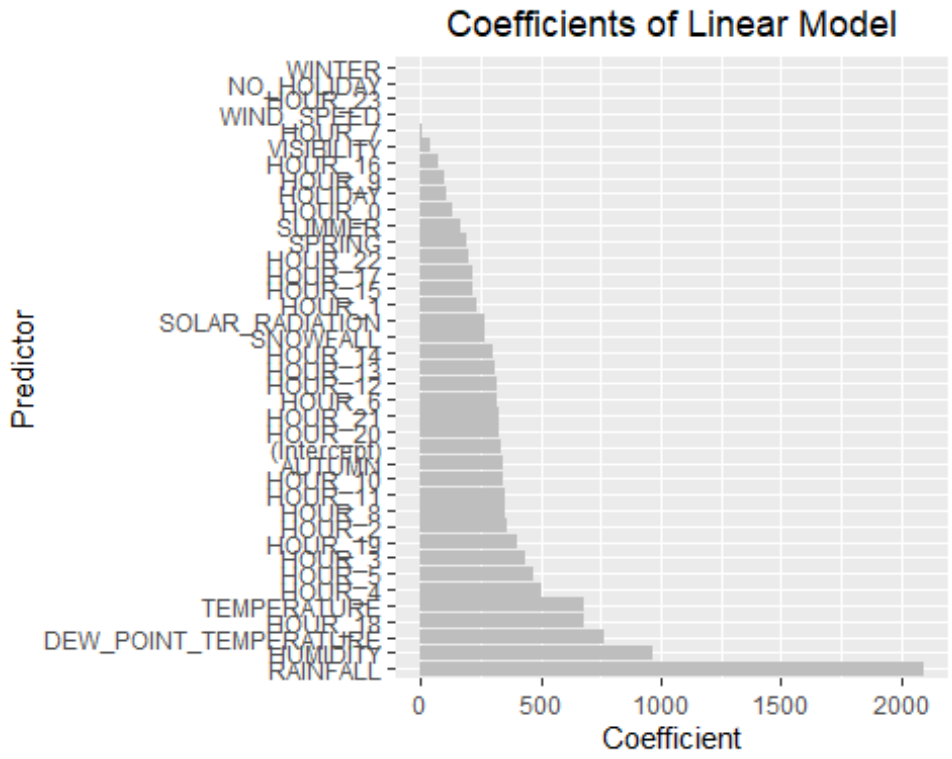
```
lm_model_all <- lm(RENTED_BIKE_COUNT ~ .,
                   data=bike_train)
summary(lm_model_all$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1659.0   337.0   723.4   730.0  1120.4  2243.9
```

Task_30: Model evaluation and identification of important variables

```
# Use model to make prediction
lm_model_weather_pred <- predict(lm_model_weather, newdata = bike_test)
test_results_weather <- data.frame(PREDICTION=lm_model_weather_pred, TRUTH =
bike_test$RENTED_BIKE_COUNT)

lm_model_all_pred <- predict(lm_model_all, newdata = bike_test)
```

```
## Warning in predict.lm(lm_model_all, newdata = bike_test): prediction from
a
## rank-deficient fit may be misleading

test_results_all <- data.frame(PREDICTION = lm_model_all_pred, TRUTH =
bike_test$RENTED_BIKE_COUNT)

summary(lm_model_weather)$r.squared #0.4303

## [1] 0.4318878

summary(lm_model_all)$r.squared #0.6589

## [1] 0.6615725

rmse_weather <- sqrt(mean((test_results_weather$TRUTH-
test_results_weather$PREDICTION)^2))
rmse_all <- sqrt(mean((test_results_all$TRUTH-
test_results_all$PREDICTION)^2))

print(rmse_weather) #474.6247

## [1] NA

print(rmse_all) #361.9543

## [1] NA

# create a data frame of coefficients
coef_df <- tidy(lm_model_all)

# plot the coefficients in a bar chart (coef plot.png)
ggplot(coef_df, aes(x = reorder(term, desc(abs(estimate))), y =
abs(estimate))) +
  geom_bar(stat = "identity", fill = "grey") +
  coord_flip() +
  xlab("Predictor") +
  ylab("Coefficient") +
  ggtitle("Coefficients of Linear Model") +
  theme(plot.title = element_text(hjust = 0.5))

## Warning: Removed 3 rows containing missing values (`position_stack()`).
```

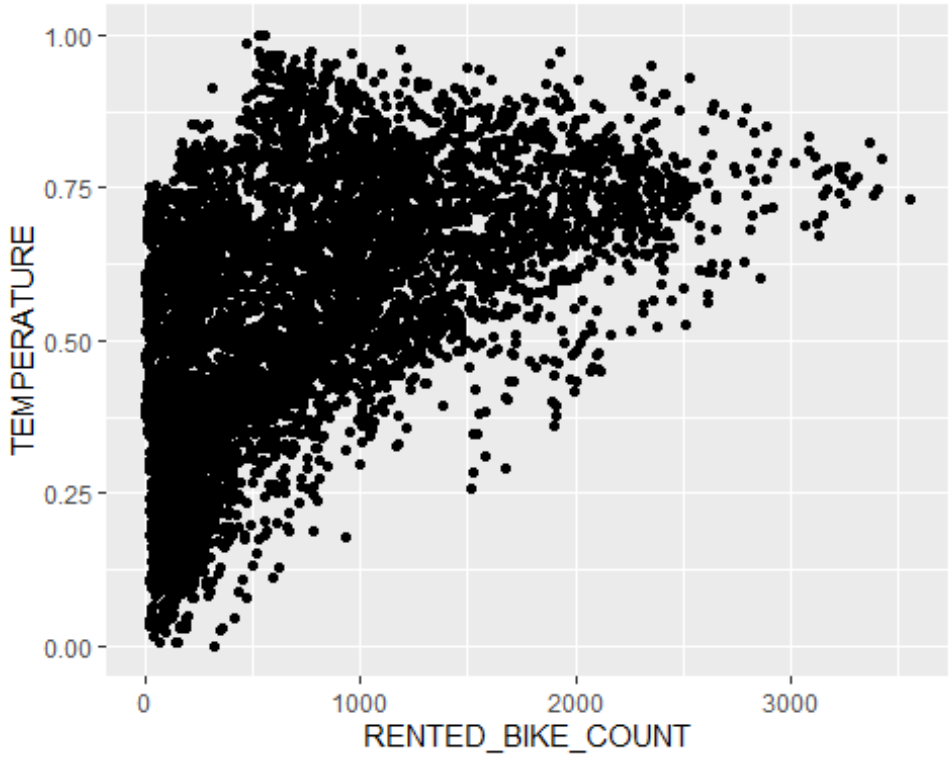


Refine the Baseline Regression Models

```
#Define a linear regression model specification.
lm_spec <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

#Split the data into training and testing datasets.
set.seed(1234)
data_split <- initial_split(bike_sharing_df, prop = 4/5)
train_data <- training(data_split)
test_data <- testing(data_split)
```

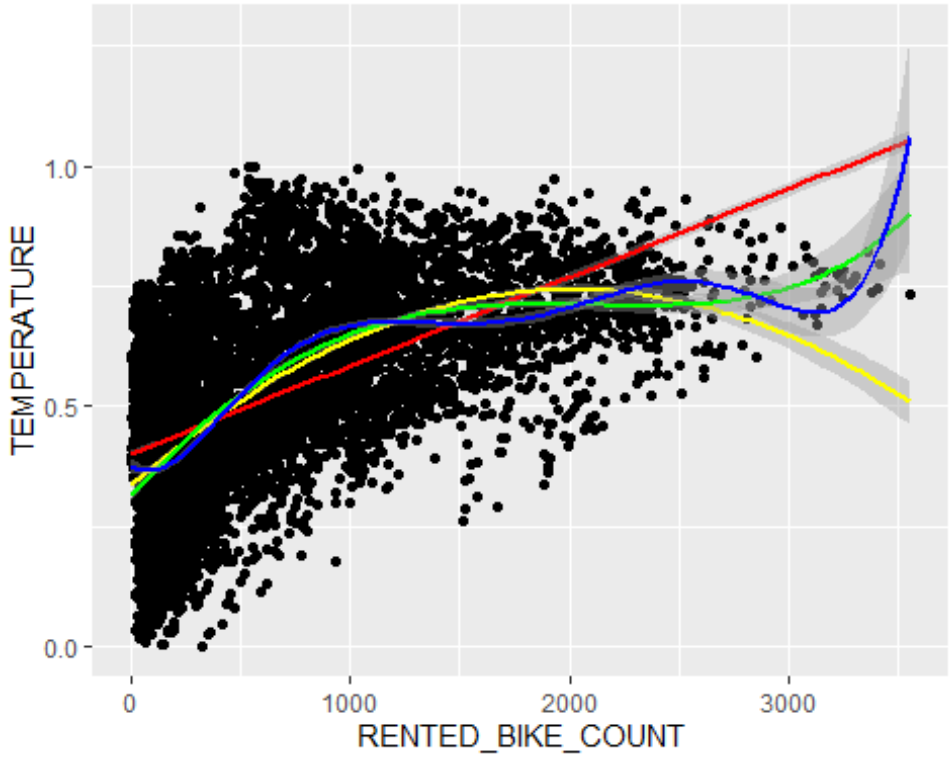
Task_31: Add polynomial terms

```
#(poly1.png)
ggplot(data=bike_train, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
  geom_point() #nonlinearity -> polynomial regression

## Warning: Removed 219 rows containing missing values (`geom_point()`).
```




```
# Plot the higher order polynomial fits
ggplot(data=train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
    geom_point() +
    geom_smooth(method = "lm", formula = y ~ x, color="red") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 2), color="yellow") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 4), color="green") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 6), color="blue")
```

```
## Warning: Removed 230 rows containing non-finite values (`stat_smooth()`).
## Removed 230 rows containing non-finite values (`stat_smooth()`).
## Removed 230 rows containing non-finite values (`stat_smooth()`).
## Removed 230 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 230 rows containing missing values (`geom_point()`).
```




```
# Fit a linear model with higher order polynomial on some important variables
lm_poly <- lm(RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) +
                poly(HUMIDITY, 4)+
                poly(RAINFALL,2), data = bike_train)
summary(lm_poly$fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -820.8   356.1   744.4   730.0  1122.8  1475.3
```

```
lm_poly_pred <- predict(lm_poly, newdata = bike_test) #predict
test_results_poly = data.frame(PREDICTION = lm_poly_pred, TRUTH =
bike_test$RENTED_BIKE_COUNT) #create df for test results

#convert all negative prediction to 0 (RENTED_BIKE_COUNT can't be negative)
test_results_poly <- test_results_poly %>%
  mutate(PREDICTION = ifelse(PREDICTION <0, 0, PREDICTION))

#calculate R_squared and RMSE (better than lm_weather but worse than lm_all)
summary(lm_poly)$r.squared #0.4861
```

```
## [1] 0.4826368
```

```
rmse_poly <- sqrt(mean ( (test_results_poly$TRUTH -
test_results_poly$PREDICTION)^2) )
rmse_poly #451.7091
```

```
## [1] NA
```

Task_32: Add interaction terms

The effect of predictor variable TEMPERATURE on RENTED_BIKE_COUNT may also depend on other variables such as HUMIDITY, RAINFALL, or both (they interact) and the effect of SEASON on RENTED_BIKE_COUNT may also depend on HOLIDAY, HOUR, or both.

```
#Task: Add Interaction Terms
lm_poly_interaction <- lm(RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) +
poly(HUMIDITY, 4)+poly(RAINFALL,2)+
                          RAINFALL*HUMIDITY + TEMPERATURE*HUMIDITY,
                          data = bike_train)
summary(lm_poly_interaction)

##
## Call:
## lm(formula = RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) + poly(HUMIDITY,
##     4) + poly(RAINFALL, 2) + RAINFALL * HUMIDITY + TEMPERATURE *
##     HUMIDITY, data = bike_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1323.41  -252.47   -64.05   169.12  2222.50
##
## Coefficients: (3 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1607.55      62.86  25.572  < 2e-16 ***
## poly(TEMPERATURE, 6)1    58060.30    1581.04  36.723  < 2e-16 ***
## poly(TEMPERATURE, 6)2    -4804.65     485.56  -9.895  < 2e-16 ***
## poly(TEMPERATURE, 6)3   -12193.35     489.73 -24.898  < 2e-16 ***
## poly(TEMPERATURE, 6)4    -4497.44     463.78  -9.697  < 2e-16 ***
## poly(TEMPERATURE, 6)5    -1387.89     458.64  -3.026  0.00249 **
## poly(TEMPERATURE, 6)6      250.70     461.76   0.543  0.58721
## poly(HUMIDITY, 4)1        8787.66    1556.96   5.644 1.73e-08 ***
## poly(HUMIDITY, 4)2       -7209.18     500.34 -14.409  < 2e-16 ***
## poly(HUMIDITY, 4)3         429.96     486.32   0.884  0.37667
## poly(HUMIDITY, 4)4       -2505.15     478.09  -5.240 1.66e-07 ***
## poly(RAINFALL, 2)1      -57013.33   20738.97  -2.749  0.00599 **
## poly(RAINFALL, 2)2        1165.78     517.39   2.253  0.02428 *
## RAINFALL                       NA         NA      NA       NA
## HUMIDITY                       NA         NA      NA       NA
## TEMPERATURE                    NA         NA      NA       NA
## RAINFALL:HUMIDITY        21033.90    7951.31   2.645  0.00818 **
## HUMIDITY:TEMPERATURE     -2932.53     165.66 -17.702  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 452 on 6336 degrees of freedom
##   (219 observations deleted due to missingness)
## Multiple R-squared:  0.5073, Adjusted R-squared:  0.5062
## F-statistic:   466 on 14 and 6336 DF,  p-value: < 2.2e-16

lm_poly_interaction_pred <- predict(lm_poly_interaction, newdata = bike_test)
```

```
## Warning in predict.lm(lm_poly_interaction, newdata = bike_test):
prediction
## from a rank-deficient fit may be misleading

test_results_poly_interaction <- data.frame(PREDICTION =
lm_poly_interaction_pred, TRUTH=bike_test$RENTED_BIKE_COUNT)

#model performance (improved model)
summary(lm_poly_interaction)$r.squared #0.5086

## [1] 0.5073318

rmse_poly_interaction <- rmse(test_results_poly_interaction, TRUTH,
PREDICTION )
rmse_poly_interaction #442

## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        445.
```

Task_33: Add regularization

install.packages("glmnet")

install.packages("yardstick")

```
library(glmnet)

library(yardstick)

#prediction function
model_prediction <- function(lm_model, test_data) {
  results <- lm_model %>%
    predict(new_data=test_data) %>%
    mutate(TRUTH=test_data$RENTED_BIKE_COUNT)
  results[results<0] <-0
  return(results)
}

#model evaluation function
model_evaluation <- function(results) {
  rmse = rmse(results, truth=TRUTH, estimate=.pred)
  rsq = rsq(results, truth=TRUTH, estimate=.pred)
  print(rmse)
  print(rsq)
}

#Use grid to define the best penalty (lambda)
penalty_value <- 10^seq(-4,4, by = 0.5) #penalty values ranging from 10^-4 to
10^4
x = as.matrix(bike_train[,-1]) #define a matrix for CV
y= bike_train$RENTED_BIKE_COUNT
```

```
#We can use cross-validation to define the lambda with 10-fold validation

# Impute missing values with mean
x[is.na(x)] <- mean(x, na.rm = TRUE)
y[is.na(y)] <- mean(y, na.rm = TRUE)

# Run cross-validation for Ridge, Lasso, and Elastic Net
cv_ridge <- cv.glmnet(x, y, alpha = 0, lambda = penalty_value, nfolds = 10)
cv_lasso <- cv.glmnet(x, y, alpha = 1, lambda = penalty_value, nfolds = 10)
cv_elasticnet <- cv.glmnet(x, y, alpha = 0.5, lambda = penalty_value, nfolds
= 10)
```

Task_34: Experiment to search for improved models

```
library(dplyr)

library(tidymodels)

#glmnet spec (using CV above, best optimal is 0.3 and 0.5)
glmnet_spec <- linear_reg(penalty = 0.3, mixture=0.5) %>%
  set_engine("glmnet") %>%
  set_mode("regression")
```

The performance requirements for your best model:

1. The RMSE should be less than 330 (rougly 10% of the max value in test dataset)
2. R-squared should be greater than 0.72

```
#Fit the model (best model)
glmnet_best <- glmnet_spec %>%
  fit(RENTED_BIKE_COUNT ~ RAINFALL*HUMIDITY*TEMPERATURE +
SPRING*SUMMER*HOLIDAY*HOUR_18* HOUR_19* HOUR_8* HOUR_21* HOUR_20* HOUR_4 +
        poly(RAINFALL, 8) + poly(HUMIDITY, 5) +  poly(TEMPERATURE, 5) +
poly(DEW_POINT_TEMPERATURE, 5) + poly(SOLAR_RADIATION, 5) + poly(SNOWFALL,5)
+
        SPRING + SUMMER  + HOLIDAY + WIND_SPEED + VISIBILITY +
        HOUR_18+ HOUR_4 + HOUR_5 + HOUR_3 + HOUR_19 + HOUR_11 + HOUR_8 +
HOUR_21 + HOUR_10 + HOUR_2 + HOUR_20,
      data = bike_train)

glmnet_best_pred <- model_prediction(glmnet_best, bike_test)
model_evaluation(glmnet_best_pred) #rsq = 0.783, rmse = 296
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        324.
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.744
```

```r
glmnet_best_rsq = rsq(glmnet_best_pred, truth = TRUTH, estimate = .pred)
glmnet_best_rmse = rmse(glmnet_best_pred, truth = TRUTH, estimate = .pred)

# Fit the model (with top 10 coeficients)
glmnet_top10 <- glmnet_spec %>%
  fit(RENTED_BIKE_COUNT ~ RAINFALL*HUMIDITY*TEMPERATURE + SPRING*SUMMER +
SUMMER +
        poly(RAINFALL, 6) + poly(HUMIDITY, 5) +  poly(TEMPERATURE, 5) +
poly(DEW_POINT_TEMPERATURE,5) +
        HOUR_18 + HOUR_4 + HOUR_5 +HOUR_3,
      data=bike_train
        )
glmnet_top10_pred <- model_prediction(glmnet_top10, bike_test)
model_evaluation(glmnet_top10_pred) #rsq = 0.640, rmse = 381 (not good)

## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        395.
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.619

glmnet_top10_rsq = rsq(glmnet_top10_pred, truth = TRUTH, estimate = .pred)
glmnet_top10_rmse = rmse(glmnet_top10_pred, truth = TRUTH, estimate = .pred)

# Fit Ridge Regression
glmnet_ridge <- glmnet(x,y, alpha=0)
glmnet_ridge_pred <- predict(glmnet_ridge, s=cv_ridge$lambda.min,
                             newx = as.matrix(bike_test[,-1]))

ridge_rmse = sqrt(mean( (bike_test[,1] - glmnet_ridge_pred)^2))
ridge_rmse #365.06

## [1] NA

ridge_mse = mean( (bike_test[,1] - glmnet_ridge_pred)^2)
ridge_rsq = 1 - ridge_mse / var(bike_test[,1])
ridge_rsq #0.667

## [1] NA

# Fit Lasso
glmnet_lasso <- glmnet(x,y,alpha=1)
glm_lasso_pred <- predict(glmnet_lasso, s=cv_lasso$lambda.min,
                          newx=as.matrix(bike_test[,-1]))

lasso_rmse = sqrt(mean( (bike_test[,1] - glm_lasso_pred)^2))
lasso_rmse #364.0492

## [1] NA
```

```
lasso_mse = mean( (bike_test[,1] - glm_lasso_pred)^2)
lasso_rsq = 1 - lasso_mse/var(bike_test[,1])
lasso_rsq #0.6693
```

```
## [1] NA
```

```
# Fit Elastic Net
glmnet_elasticnet <- glmnet(x,y,alpha=0.7)
glm_elasticnet_pred <- predict(glmnet_elasticnet, s=cv_elasticnet$lambda.min,
                          newx=as.matrix(bike_test[,-1]))

elasticnet_rmse = sqrt(mean( (bike_test[,1] - glm_elasticnet_pred)^2))
elasticnet_rmse #364.0468
```

```
## [1] NA
```

```
elasticnet_mse = mean( (bike_test[,1] - glm_elasticnet_pred)^2)
elasticnet_rsq = 1 - elasticnet_mse/var(bike_test[,1])
elasticnet_rsq #0.6693
```

```
## [1] NA
```

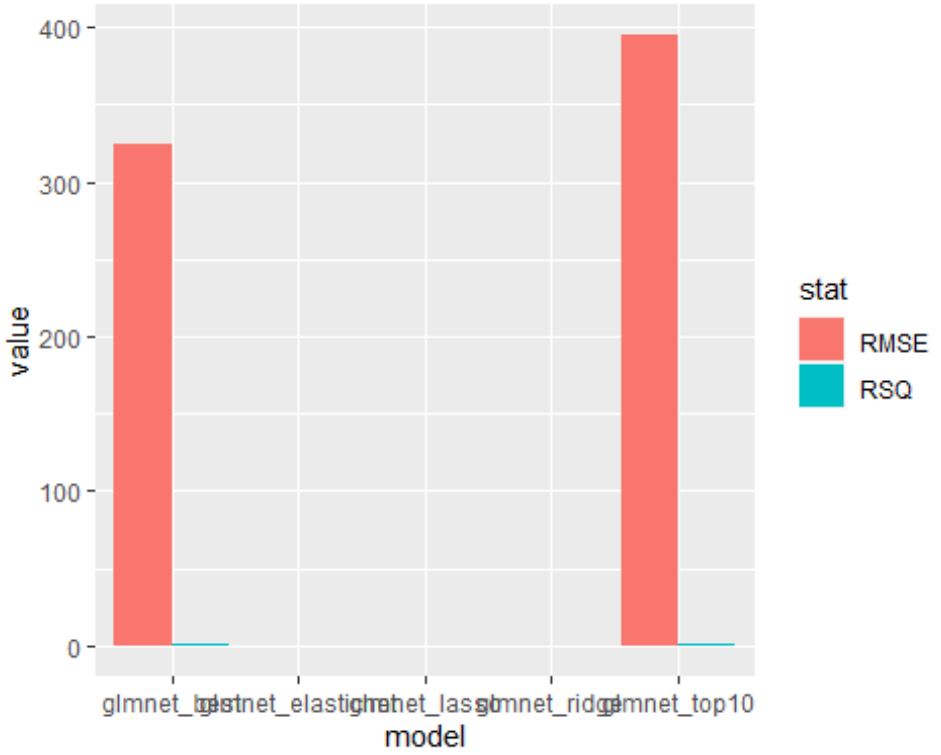
Visualize the saved RMSE and R-squared values using a grouped barchart

```
#Create data frame for group bar chart
model <- c(rep("glmnet_best",2), rep("glmnet_top10",2),
           rep("glmnet_ridge",2), rep("glmnet_lasso",2),
rep("glmnet_elasticnet",2))
stat <- rep(c("RSQ", "RMSE"),5)
value <- c(glmnet_best_rsq$.estimate, glmnet_best_rmse$.estimate,
           glmnet_top10_rsq$.estimate, glmnet_top10_rmse$.estimate,
           ridge_rsq, ridge_rmse,
           lasso_rsq,lasso_rmse,
           elasticnet_rsq, elasticnet_rmse)
model_df <-  data.frame(model, stat, value)
print(model_df)
```

```
##                model stat       value
## 1        glmnet_best  RSQ   0.7442236
## 2        glmnet_best RMSE 323.9034890
## 3       glmnet_top10  RSQ   0.6194298
## 4       glmnet_top10 RMSE 394.9987282
## 5       glmnet_ridge  RSQ          NA
## 6       glmnet_ridge RMSE          NA
## 7       glmnet_lasso  RSQ          NA
## 8       glmnet_lasso RMSE          NA
## 9  glmnet_elasticnet  RSQ          NA
## 10 glmnet_elasticnet RMSE          NA
```

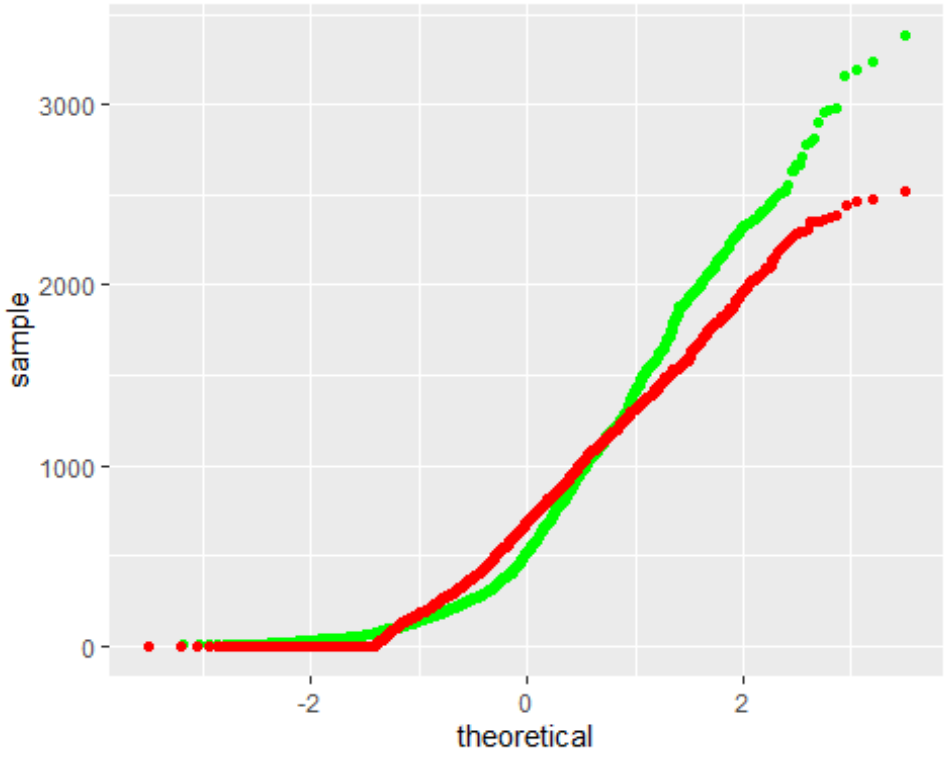
```
# Create group bar chart for rsq and rmse (model evaluation.png)
model_df %>%
  ggplot(aes(fill=stat, x=model, y=value)) +
  geom_bar(position="dodge", stat="identity")
```

```
## Warning: Removed 6 rows containing missing values (`geom_bar()`).
```



Q-Q plot by plotting the distribution difference between the predictions generated by your best model and the true values on the test dataset.

```
# Create a Q-Q chart for best model: glmnet_best (Q-Q chart.png)
glmnet_best_pred %>%
  ggplot() +
  stat_qq(aes(sample=TRUTH), color='green') +
  stat_qq(aes(sample=.pred), color='red')

## Warning: Removed 76 rows containing non-finite values (`stat_qq()`).
```



Conclusion

From the above analysis result, Fall, Spring, Summer and Temperatur are the 4 most important factors for bike rental number in Seoul city.