

# Chose Your Own Project - Machine Learning Submission

HarvardX Data Science Capstone - PH125.9x

Simon Gibson

2024-04-05

## Contents

Introduction . . . . .	1
Method . . . . .	1
References . . . . .	10

## Introduction

For the 9th Course in the HarvardX Data Science course we have been asked to create two recommendation systems. The first was a Movie Recommendation System using the MovieLens dataset. The second is a “Choose your Own Project.” For this a we are targetting a Workforce Recommendation System - mixing weather forecasts with Police 911 call information to see if it is possible to predict Police staffing requirements based on weather based trends.

We are using the Seattle Police Department 911 Incident Response data set found here : <https://www.kaggle.com/datasets/sohier/seattle-police-department-911-incident-response>

For Weather data we will use National Oceanic and Atmospheric Administration (NOAA) data. Michael Minns’ tutorial is inciteful for weather analysis. It can be found here: <https://michaelminn.net/tutorials/r-weather/index.html> This weather data does not appear to be available via an api call or similar and is quite a manual download process. Due to download constraints we will be using a locally sourced dataset covering the years 2001 to 2002.

In order to test the results of the recommendation system we are using the root-mean-square error (RMSE) to measure the difference between the values predicted by the model and the observed values.

## Method

The first step is to clear any set variables so we do not introduce anything unexpected into the data we are working with.

Then we install the packages required to manipulate the data.

```
#####  
# This code is divided into the following sections #  
# 1. Install required packages                      #  
# 2. edx code for creating data sets                #  
# 3. Data set exploration                          #  
#####  
  
#####  
# 1. Install required packages and download data    #  
#####
```

```

# Note: this process takes a couple of minutes

if(!require(tidyverse)) install.packages("tidyverse", repos = "https://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "https://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "https://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos = "https://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "https://cran.us.r-project.org")
if(!require(scales)) install.packages("scales", repos = "https://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr", repos = "http://cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(xts)) install.packages("xts", repos = "http://cran.us.r-project.org")
if(!require(tsbox)) install.packages("tsbox", repos = "http://cran.us.r-project.org")
if(!require(forecast)) install.packages("forecast", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(measurements)) install.packages("measurements", repos = "http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(caret)
library(dplyr)
library(kableExtra)
library(lubridate)
library(scales)
library(stringr)
library(readr)
library(xts)
library(tsbox)
library(forecast)
library(data.table)
library(measurements)
library(kableExtra)

```

Following that, the data is downloaded and then divided into 2 sets. The first set is used to train the algorithm and the second set is used to validate the algorithm. By dividing the data the problem of over-training and thus producing skewed results can be avoided.

The creation of the 2 sets involves the following steps. Initially required packages are installed if not installed and then loaded. Next the data is downloaded if the zip files are not found. Column names are set and the data is converted into forms more easily processed. Then the data is joined. Finally the joined data is split into 2 sets - the edx set used to train the algorithm and the final\_holdout\_test set that will be used to validate the algorithm and calculate the final RMSE score.

```

#Seattle Police Department 911 Incident Response
#https://www.kaggle.com/datasets/sohier/seattle-police-department-911-incident-response/download?dataset=

#National Oceanic and Atmospheric Administration (NOAA) data
#https://www.ncei.noaa.gov/orders/cdo/3533326.csv

options(timeout = 120)

dl <- "archive.zip"
if(!file.exists(dl))
  download.file("https://www.kaggle.com/datasets/sohier/seattle-police-department-911-incident-response",
               dl)

dl <- "3533326.csv"

```

```
if(!file.exists(dl))
  download.file("https://www.ncei.noaa.gov/orders/cdo/3533326.csv", dl)
```

```
#Load Seattle 0911 Call data
```

```
Seattle_911 <- read_csv("Seattle_Police_Department_911_Incident_Response.csv")
```

```
#Load weather data
```

```
Weather <- read.csv("3533326.csv", as.is=T)
```

```
##Data Investigation
```

```
head(Weather)
```

```
##      STATION      NAME      DATE PRCP SNOW TAVG TMAX TMIN TSUN WT01
## 1 USC00450872 BREMERTON, WA US 2000-01-01 0.23    0  NA  44  38  NA  NA
## 2 USC00450872 BREMERTON, WA US 2000-01-02 0.00    0  NA  44  31  NA  NA
## 3 USC00450872 BREMERTON, WA US 2000-01-03 0.10    0  NA  45  32  NA  NA
## 4 USC00450872 BREMERTON, WA US 2000-01-04 1.38    0  NA  47  35  NA  NA
## 5 USC00450872 BREMERTON, WA US 2000-01-05 0.02    0  NA  51  30  NA  NA
## 6 USC00450872 BREMERTON, WA US 2000-01-06 0.01    0  NA  44  34  NA  NA
##   WT02 WT03 WT04 WT05 WT06 WT07 WT08 WT09 WT11 WT13 WT14 WT15 WT16 WT17 WT18
## 1   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 4   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   WT19 WT21 WT22 WV01 WV03
## 1   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA
## 3   NA   NA   NA   NA   NA
## 4   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA
```

```
names(Weather)
```

```
## [1] "STATION" "NAME"    "DATE"    "PRCP"    "SNOW"    "TAVG"    "TMAX"
## [8] "TMIN"    "TSUN"    "WT01"    "WT02"    "WT03"    "WT04"    "WT05"
## [15] "WT06"    "WT07"    "WT08"    "WT09"    "WT11"    "WT13"    "WT14"
## [22] "WT15"    "WT16"    "WT17"    "WT18"    "WT19"    "WT21"    "WT22"
## [29] "WV01"    "WV03"
```

```
min(range(Weather$DATE))
```

```
## [1] "2000-01-01"
```

```
max(range(Weather$DATE))
```

```
## [1] "2002-12-31"
```

Our data range starts from 2000-01-01 and ends 2002-12-31.

```
#Seattle_Weather <- xts(Weather["Weather$STATION" == 'USC00450872',c("TMAX","TMIN","PRCP")], order.by=as.Date(Weather$DATE))
Seattle_Weather <- xts(Weather[,c("NAME","STATION","DATE","TMAX","TMIN","PRCP")], order.by=as.Date(Weather$DATE))
```

```
Seattle_Weather <- as.data.frame(Seattle_Weather)
```

```
#Seattle_Weather = window(Seattle_Weather, start=as.Date("2000-01-01"), end=as.Date("2002-12-31"))
```

```

class(Seattle_Weather)

## [1] "data.frame"

Seattle_Weather$DATE <- as.Date(Seattle_Weather$DATE)
Seattle_Weather$PRCP <- as.numeric(Seattle_Weather$PRCP)

#Convert Precipitation from Imperial to Metric
Seattle_Weather$PRCP <- conv_unit(Seattle_Weather$PRCP, "inch", "mm")

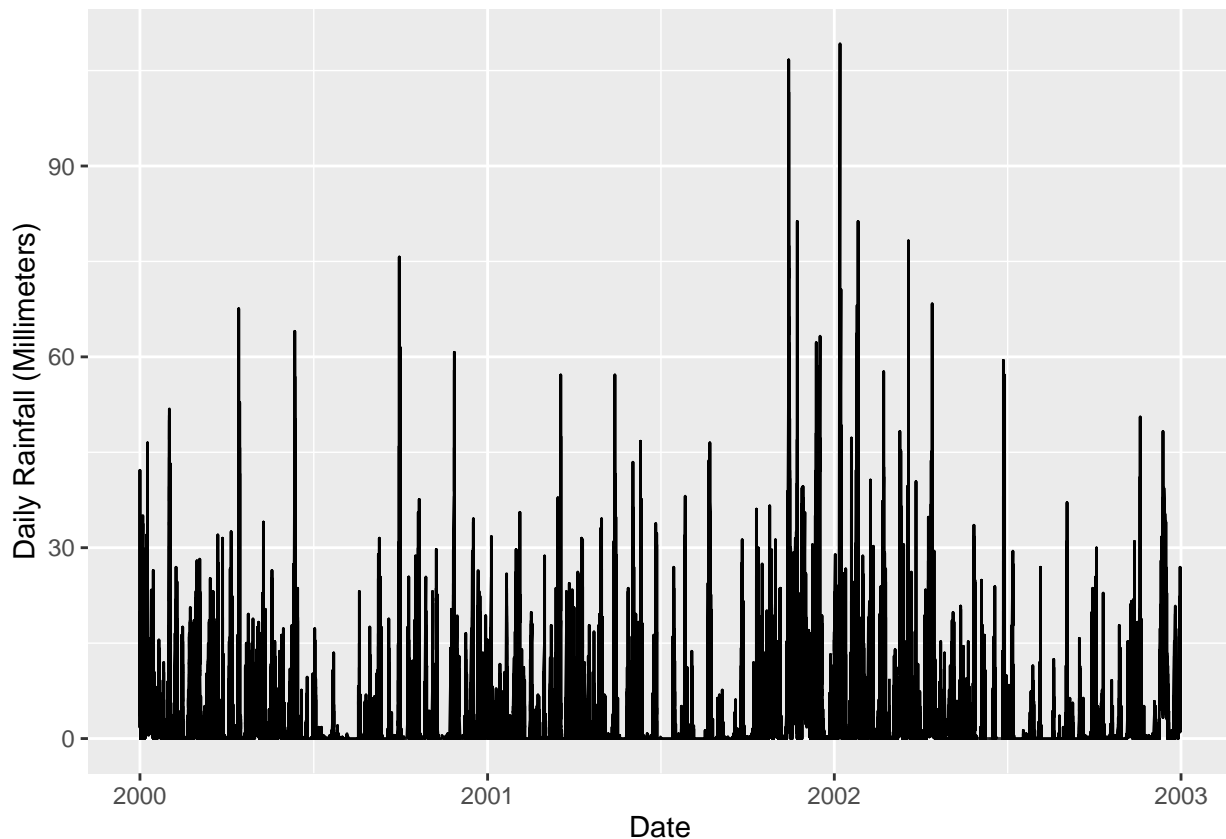
Seattle_Weather$TMAX <- as.numeric(Seattle_Weather$TMAX)
Seattle_Weather$TMIN <- as.numeric(Seattle_Weather$TMIN)
#hist(x = Seattle_Weather$TMIN, xlab = "Precipitation", ylab = 'Frequency of readings', main = #paste("N

#Extract Unique Station Names and Identifiers
Seattle_Stations <- unique(Seattle_Weather[, c('NAME', 'STATION')])

# Remove the index column - otherwise it gets printed even though we asked for only Station and Name
rownames(Seattle_Stations) <- NULL

ggplot(Seattle_Weather, aes(x=Seattle_Weather$DATE,y=Seattle_Weather$PRCP)) +
  geom_line() +
  xlab("Date") +
  ylab("Daily Rainfall (Millimeters)")

```



We have data from 20 stations:

NAME	STATION
BREMERTON, WA US	USC00450872
EVERETT, WA US	USC00452675
MONROE, WA US	USC00455525
TOLT SOUTH FORK RESERVOIR, WA US	USC00458508
RENTON MUNICIPAL AIRPORT, WA US	USW00094248
KENT, WA US	USC00454169
TACOMA NUMBER 1, WA US	USC00458278
LANDSBURG, WA US	USC00454486
CEDAR LAKE, WA US	USC00451233
SNOQUALMIE FALLS, WA US	USC00457773
WAUNA 3 W, WA US	USC00459021
PALMER 3 ESE, WA US	USC00456295
TACOMA NARROWS AIRPORT, WA US	USW00094274
EVERETT SNOHOMISH CO AIRPORT, WA US	USW00024222
SEATTLE TACOMA AIRPORT, WA US	USW00024233
SEATTLE SAND POINT WEATHER FORECAST OFFICE, WA US	USW00094290
SEATTLE BOEING FIELD, WA US	USW00024234
GIG HARBOR 3.4 NW, WA US	US1WAPR0075
OLALLA 1.4 WNW, WA US	US1WAKP0013
WOODINVILLE 0.9 ENE, WA US	US1WAKG0078

Of 17773 rainfall measurements, 7869 recorded rainfall, and 9794 recorded no rainfall. The maximum rainfall during this period was 109.22mm which fell on 2002-01-07. Heavy rainfall is defined by NIWA as rainfall of over 100mm in 24 hours and this occurred 3 times during the period we have data for.

```
Seattle_Weather %>% group_by(Seattle_Weather$STATION)
```

```
## # A tibble: 17,773 x 7
## # Groups:   Seattle_Weather$STATION [20]
##   NAME          STATION DATE      TMAX  TMIN  PRCP Seattle_Weather$STAT~1
##   <chr>          <chr>   <date>    <dbl> <dbl> <dbl> <chr>
##  1 BREMERTON, WA US USC004~ 2000-01-01    44    38  5.84 USC00450872
##  2 EVERETT, WA US   USC004~ 2000-01-01    NA    NA  12.7 USC00452675
##  3 MONROE, WA US    USC004~ 2000-01-01    45    38  4.06 USC00455525
##  4 TOLT SOUTH FORK ~ USC004~ 2000-01-01    NA    NA  42.2 USC00458508
##  5 RENTON MUNICIPAL~ USW000~ 2000-01-01    45    39  6.86 USW00094248
##  6 KENT, WA US      USC004~ 2000-01-01    47    36  8.13 USC00454169
##  7 TACOMA NUMBER 1,~ USC004~ 2000-01-01    47    37  10.9 USC00458278
##  8 LANDSBURG, WA US USC004~ 2000-01-01    43    36  8.13 USC00454486
##  9 CEDAR LAKE, WA US USC004~ 2000-01-01    41    31  19.8 USC00451233
## 10 SNOQUALMIE FALLS~ USC004~ 2000-01-01    44    36  18.3 USC00457773
## # i 17,763 more rows
## # i abbreviated name: 1: `Seattle_Weather$STATION`
```

```
summary(Seattle_911)
```

```
##   CAD CDW ID      CAD Event Number  General Offense Number
## Length:1433853   Min.      :9.000e+09   Min.      :2.011e+04
## Class :character 1st Qu.:1.200e+10   1st Qu.:2.010e+09
## Mode  :character Median :1.400e+10   Median :2.012e+09
##                      Mean  :1.366e+10   Mean  :1.641e+09
##                      3rd Qu.:1.600e+10   3rd Qu.:2.015e+09
##                      Max.   :1.700e+10   Max.   :2.012e+10
```

```
##
## Event Clearance Code Event Clearance Description Event Clearance SubGroup
## Length:1433853      Length:1433853      Length:1433853
## Class :character    Class :character    Class :character
## Mode :character     Mode :character     Mode :character
##
##
##
## Event Clearance Group Event Clearance Date Hundred Block Location
## Length:1433853      Length:1433853      Length:1433853
## Class :character    Class :character    Class :character
## Mode :character     Mode :character     Mode :character
##
##
##
## District/Sector      Zone/Beat      Census Tract      Longitude
## Length:1433853      Length:1433853      Length:1433853      Min.      :-122.4
## Class :character    Class :character    Class :character    1st Qu.:-122.3
## Mode :character     Mode :character     Mode :character     Median :-122.3
##                                     Mean      :-122.3
##                                     3rd Qu.:-122.3
##                                     Max.      :-122.2
##                                     NA's      :1
## Latitude      Incident Location      Initial Type Description
## Min.      :47.45      Length:1433853      Length:1433853
## 1st Qu. :47.59      Class :character    Class :character
## Median :47.61      Mode :character     Mode :character
## Mean    :47.62
## 3rd Qu. :47.66
## Max.    :47.78
## NA's    :1
## Initial Type Subgroup Initial Type Group At Scene Time
## Length:1433853      Length:1433853      Length:1433853
## Class :character    Class :character    Class :character
## Mode :character     Mode :character     Mode :character
##
##
##
##
```

`summary(Weather)`

```
## STATION      NAME      DATE      PRCP
## Length:17773      Length:17773      Length:17773      Min.      :0.0000
## Class :character    Class :character    Class :character    1st Qu.:0.0000
## Mode :character     Mode :character     Mode :character     Median :0.0000
##                                     Mean    :0.1278
##                                     3rd Qu.:0.1200
##                                     Max.    :4.3000
##                                     NA's    :110
## SNOW      TAVG      TMAX      TMIN
## Min.      : 0.000      Min.      : 0.00      Min.      : 0.00      Min.      : -16.00
## 1st Qu.: 0.000      1st Qu.:44.00      1st Qu.:50.00      1st Qu.: 36.00
```

```

## Median : 0.000    Median :51.00    Median :58.00    Median : 43.00
## Mean   : 0.042    Mean   :52.27    Mean   :59.14    Mean   : 43.21
## 3rd Qu.: 0.000    3rd Qu.:60.00    3rd Qu.:68.00    3rd Qu.: 50.00
## Max.   :24.000    Max.   :82.00    Max.   :99.00    Max.   : 77.00
## NA's   :7233     NA's   :11397    NA's   :2511     NA's   :2537
##      TSUN          WT01          WT02          WT03
## Min.   : 0.00    Min.   :1       Min.   :1       Min.   :1
## 1st Qu.: 0.00    1st Qu.:1       1st Qu.:1       1st Qu.:1
## Median : 0.00    Median :1       Median :1       Median :1
## Mean   : 31.76    Mean   :1       Mean   :1       Mean   :1
## 3rd Qu.: 0.00    3rd Qu.:1       3rd Qu.:1       3rd Qu.:1
## Max.   :931.00    Max.   :1       Max.   :1       Max.   :1
## NA's   :14935    NA's   :16900    NA's   :17707    NA's   :17755
##      WT04          WT05          WT06          WT07          WT08
## Min.   :1       Min.   :1       Min.   :1       Mode:logical    Min.   :1
## 1st Qu.:1       1st Qu.:1       1st Qu.:1       NA's:17773       1st Qu.:1
## Median :1       Median :1       Median :1                                     Median :1
## Mean   :1       Mean   :1       Mean   :1                                     Mean   :1
## 3rd Qu.:1       3rd Qu.:1       3rd Qu.:1                                     3rd Qu.:1
## Max.   :1       Max.   :1       Max.   :1                                     Max.   :1
## NA's   :17766    NA's   :17761    NA's   :17772                                     NA's   :17752
##      WT09          WT11          WT13          WT14          WT15
## Mode:logical    Mode:logical    Min.   :1       Min.   :1       Mode:logical
## NA's:17773       NA's:17773       1st Qu.:1       1st Qu.:1       NA's:17773
##                                     Median :1       Median :1
##                                     Mean   :1       Mean   :1
##                                     3rd Qu.:1       3rd Qu.:1
##                                     Max.   :1       Max.   :1
##                                     NA's   :17286    NA's   :17688
##      WT16          WT17          WT18          WT19
## Min.   :1       Min.   :1       Min.   :1       Min.   :1
## 1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1
## Median :1       Median :1       Median :1       Median :1
## Mean   :1       Mean   :1       Mean   :1       Mean   :1
## 3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1
## Max.   :1       Max.   :1       Max.   :1       Max.   :1
## NA's   :17209    NA's   :17772    NA's   :17748    NA's   :17771
##      WT21          WT22          WV01          WV03
## Min.   :1       Min.   :1       Min.   :1       Min.   :1
## 1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1
## Median :1       Median :1       Median :1       Median :1
## Mean   :1       Mean   :1       Mean   :1       Mean   :1
## 3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1
## Max.   :1       Max.   :1       Max.   :1       Max.   :1
## NA's   :17725    NA's   :17766    NA's   :17767    NA's   :17771

```

```

# Group Data by weather station
weather_data_grouped <- Seattle_Weather %>%
  group_by(STATION)

# find average maximum temperature
average_max_temp <- weather_data_grouped %>%
  summarise(avg_max_temp = mean(TMAX, na.rm = TRUE))

```

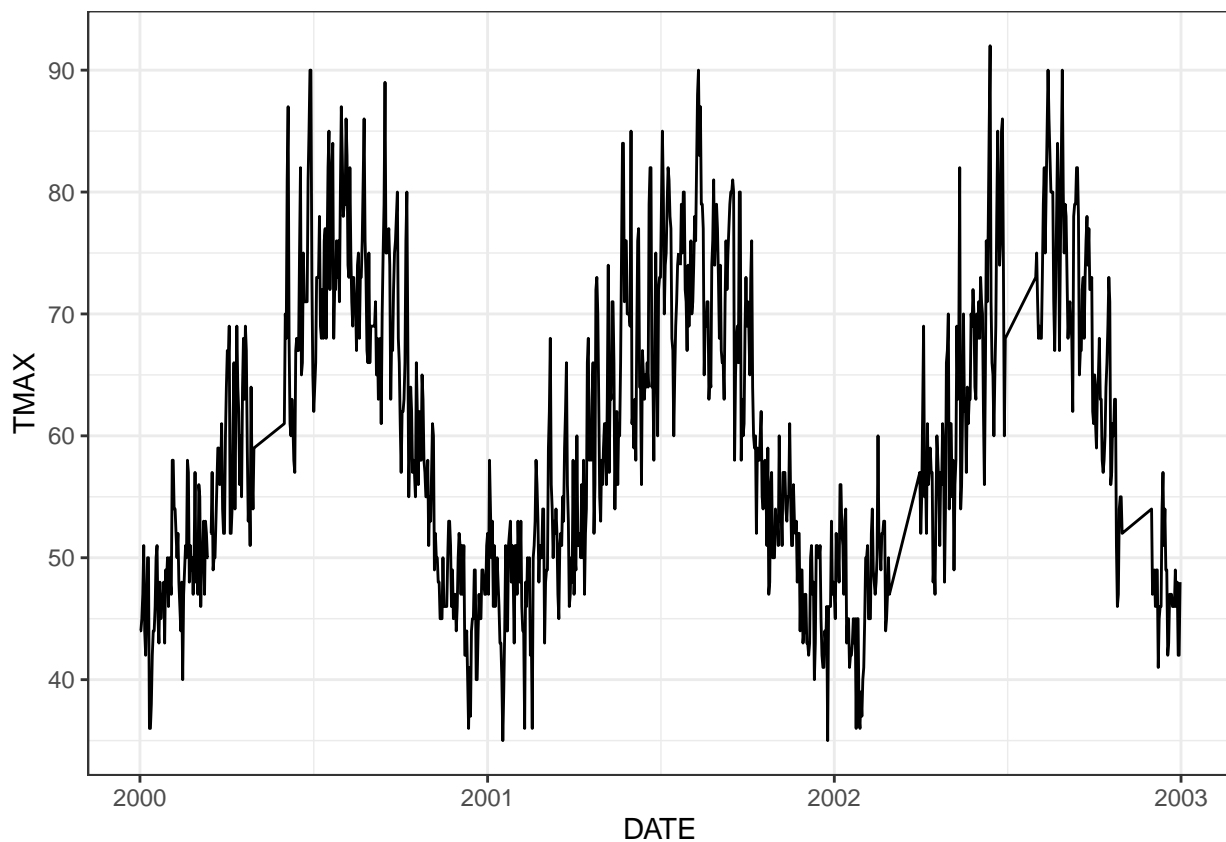
```

# Get unique station codes
station_codes <- unique(Seattle_Weather$STATION)

# Create a list to store data frames for each station
station_data_list <- list()

# Loop through each station code and filter data for that station
for (station_code in station_codes) {
  station_data <- filter(Seattle_Weather, STATION == station_code)
  station_data_list[[station_code]] <- station_data
}
ggplot(station_data_list[["USC00450872"]], aes(x=DATE, y=TMAX)) +
  geom_line() +
  theme_bw()

```



```

USC00450872 <- station_data_list[["USC00450872"]]

historical = xts(USC00450872[,c("TMAX", "TMIN", "PRCP")], order.by=as.Date(USC00450872$DATE))

historical = ts_regular(historical)

historical = suppressWarnings(na.fill(historical, "extend"))

historical = window(historical, start=as.Date("2000-01-01"), end=as.Date("2020-12-31"))

```



```

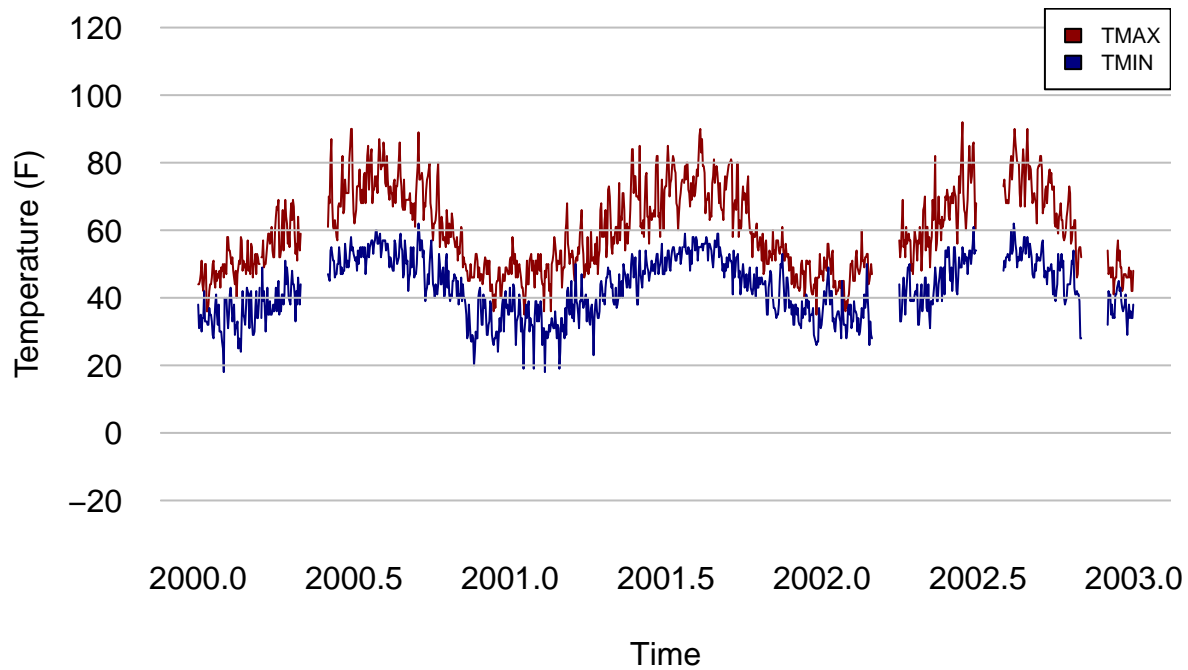
plot(ts_ts(historical$TMAX), col="darkred", bty="n", las=1, fg=NA,
     ylim=c(-20, 120), ylab="Temperature (F)")

lines(ts_ts(historical$TMIN), col="navy")

grid(nx=NA, ny=NULL, lty=1, col="gray")

legend("topright", fill=c("darkred", "navy"), cex=0.7,
      legend=c("TMAX", "TMIN"), bg="white")

```

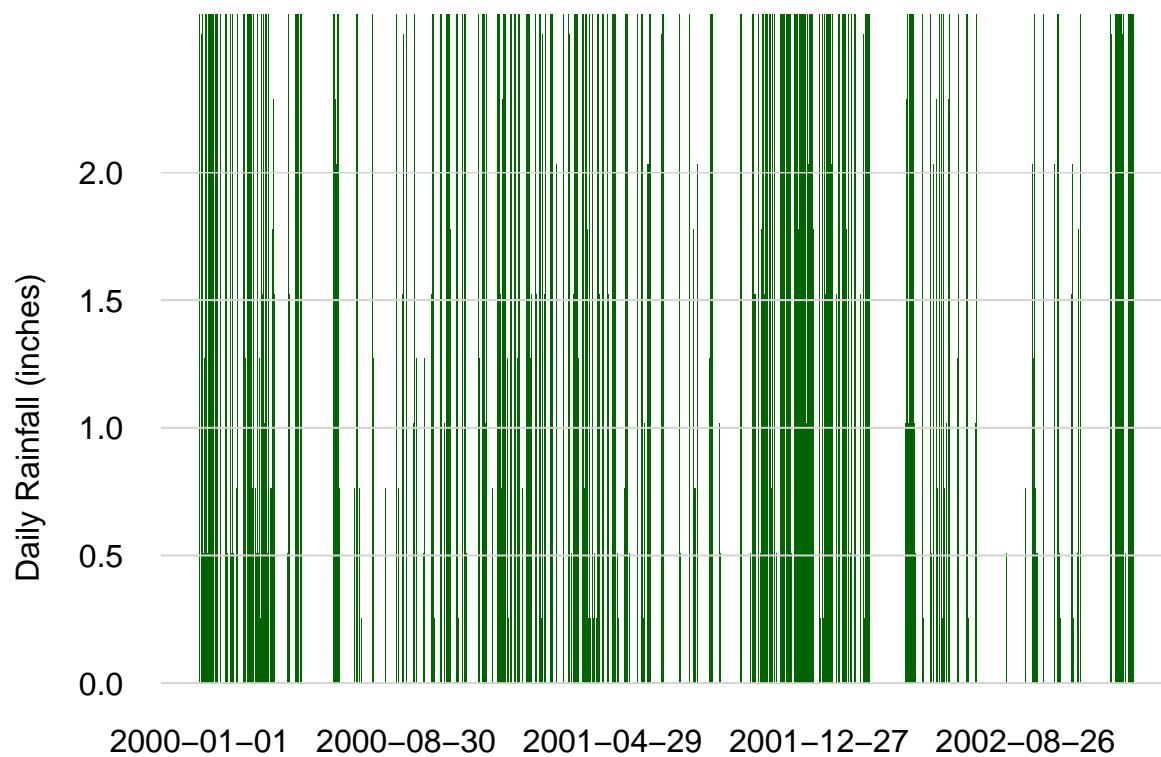


```

barplot(historical$PRCP, border=NA, col="darkgreen", ylim=c(0, 2),
        space=0, bty="n", las=1, fg=NA, ylab="Daily Rainfall (inches)")

grid(nx=NA, ny=NULL, lty=1)

```



## References

- 1.
- 2.
- 3.
4. <https://www.neonscience.org/resources/learning-hub/tutorials/da-viz-coop-precip-data-r>