school

<https://www.nature.com/articles/s41597-022-01205-9#Sec5>

SWAQ adds to the growing number of urban meteorological networks (UMNs) deployed in the last decade worldwide, with the specific purpose of monitoring city-scale heat and air quality dynamics. Currently deployed UMNs include the Metropolitan Environmental Temperature and Rainfall Observation System (METROS) in Central Tokyo, Japan[6](https://www.nature.com/articles/s41597-022-01205-9#ref-CR6), the Oklahoma City Micronet (OKCNET) in USA[7](https://www.nature.com/articles/s41597-022-01205-9#ref-CR7), the Helsinki Testbed in Finland[8](https://www.nature.com/articles/s41597-022-01205-9#ref-CR8), the Turku Urban Climate Research Project (TURCLIM) in Finland[9](https://www.nature.com/articles/s41597-022-01205-9#ref-CR9), the Olomouc’s Metropolitan Station System in Czech Republic (MESSO)[10](https://www.nature.com/articles/s41597-022-01205-9#ref-CR10), the Birmingham Urban Climate Laboratory (BULC) in the UK[11](https://www.nature.com/articles/s41597-022-01205-9#ref-CR11), the HiSAN network in Tainan City, Taiwan[12](https://www.nature.com/articles/s41597-022-01205-9#ref-CR12), and the MOCCA network in Ghent, Belgium[13](https://www.nature.com/articles/s41597-022-01205-9#ref-CR13). SWAQ aligns with the above UMNs and devotes special attention to site documentation by following a standardized UMN metadata protocol[14](https://www.nature.com/articles/s41597-022-01205-9#ref-CR14) so as to improve site representativeness, maximize comparability across UMNs, and contribute to the buildup of a consistent database.

<https://github.com/giuliaulpiani/SWAQ>

<https://portal.tern.org.au/schools-weather-air-sydney-nsw/22077>

<https://zenodo.org/record/5016296#.YzIMRUzMJD8>

<https://portal.tern.org.au/schools-weather-air-sydney-nsw/22077>

DAIS Japan Environmental Temperature and Rainfall Observation System (METROS) in Central Tokyo, Japan[6](https://www.nature.com/articles/s41597-022-01205-9#ref-CR6)If all else fails, please send e-mail to [dias-office@diasjp.net](mailto:dias-office@diasjp.net).

xts5 <- as.xts(AirPassengers)

theData <- read.table(text="Date Time Value

20090202 9:30 1

20090202 9:31 2

20090202 9:32 3

20090202 9:33 4

20090202 9:34 5

20090202 9:35 6", header=TRUE, as.is=TRUE)

theData$DateTime <- paste(theData$Date, theData$Time)

xts(theData$Value, as.POSIXct(theData$DateTime, format="%Y%m%d %H:%M"))

c(

time <- [ymd\_hms](https://lubridate.tidyverse.org/reference/ymd_hms.html)(c("2010-12-13 15:30:30":"2010-12-14 15:30:30")

# converting to datetime object

data\_frame[['col3']] <- strptime(data\_frame[['col3']],

                                 format = "%Y-%m-%d %H:%M:%S")

start\_2012 <- ymd\_hms("2012-01-01 12:00:00")

file:///C:/Users/STSC/Downloads/v40i03.pdf

seq(from=as.POSIXct("2020-01-01 00:01"),to=as.POSIXct("2020-01-01 01:00"),by="min")

year, month, day, hour, minute, second ymd\_hms()

1st tried to look for data see airrescouces data, but no weather info

2nd find Iowa free airport data. Look at Hayward airport, but in WI. Could not get it into a xts object. Later would realize there were missing datetimes also may na’s. Teacher suggests to using nycflights. Note, this data is clean with all the datetimes. There are a few na’s in the variables. It is only one year 2013. I tried to develop code to look at and select some variables, filled na’s/

Tools

Datetime objects xts, ts,

Tools for datetime objects

<https://www.geeksforgeeks.org/how-to-merge-date-and-time-in-r/#:~:text=The%20date%20and%20time%20objects%20in%20R%20programming,The%20POSIXlt%20class%20stores%20date%20and%20time%20information>. POIX

luberdate: <https://lubridate.tidyverse.org/reference/lubridate-package.html#:~:text=Lubridate%27s%20parsing%20functions%20read%20strings%20into%20R%20as,ydm%20%28%29%2C%20dym%20%28%29%2C%20mdy%20%28%29%2C%20ymd_hms%20%28%29%29>.

Daily vs less <https://medium.com/@lopezgrj/date-objects-in-r-using-lubridate-package-59bf4bc21162>

**lubridate::ymd\_hms**("2019-06-04 12:00:00", tz = "America/ America/Tijuana")

grep("America", OlsonNames(), value=TRUE)

Graph

Fill na’s

Collapse from hourly to monthly to year

RDS <http://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata#:~:text=Saving%20data%20into%20R%20data%20formats%20can%20reduce,to%20a%20specified%20file%20%28in%20rds%20file%20format%29>.

Class, str,

<https://www.rstudio.com/blog/interactive-time-series-with-dygraphs/>

Reading in the data

Started with read.csv. This is not a good choice as it seemed at times to read in na’s. Took me weeks to realize that read.csv is NOT a good tool more moderate to large files.

Alternatively, read\_csv comes is in readr, but also more compatible with dplyr

Yet another approach is fread which very fast. **data.table**

<https://stackoverflow.com/questions/51765374/read-csv-faster-than-data-tablefread#:~:text=If%20you%20take%20a%20look%20into%20the%20functions,reading.%20data.table%20is%20incredibly%20faster%20for%20big%20datasets>.

<https://www.r-bloggers.com/2016/12/remember-to-use-the-rds-format/#:~:text=Here%20again%2C%20R%20offers%20a%20superior%20alternative%20to,RDS%20objectstill%20takes%20around%203%C2%A0GB%20of%20%28live%29%20memory>.

Check for all datetimes use merg

Look for na’s

Fill na’s

Storing the dataframe dot.csv, RDS is compressed with qzip.

Also tried jasonlite

Predict

Measures of variability

<https://www.geeksforgeeks.org/variability-in-r-programming/>

* Variance
* Standard Deviation
* Range
* Mean Deviation
* Interquartile Range

Cross validation: <https://www.geeksforgeeks.org/cross-validation-in-r-programming/>

easystats::[install\_suggested](https://easystats.github.io/easystats/reference/install_suggested.html)()

tidymodels

report

rm(list = ls(all.names = TRUE))

library(xts)

library(dygraphs)

library(tidyverse)

library(lubridate)

library(fable)

library(tsibble) # ?weather # see what is in weather

library(nycflights13)

library(readr)

jfk <- weather %>%

filter(origin == "JFK") # get weather data 2013 for JFK

colnames(jfk) # look at variables

head( jfk %>% select(-c(origin, wind\_gust)) -> jfk1) # look at data & remove column origin

typeof(jfk1) # jfk1 is a list

`# Use\_jasonlite to store and retrieve a data frame.

temp<-toJSON(jfk) # store jfk in temp

a<-fromJSON(temp) # retrieve temp into a

head(a) # compare a to jfk

head(jfk)

library(jsonlite)

df %>%

group\_split(jfk) %>%

toJSON

**One minuet Iowa**

[**https://mesonet.agron.iastate.edu/request/asos/1min.phtml**](https://mesonet.agron.iastate.edu/request/asos/1min.phtml)

AirPortVariables

1 station, Fed airport id, 3 letters

2 station\_name,

3 valid(UTC), date time (year month day hour minuet) json format

4 tmpf, temp F

5 dwpf, dew pt temp F

6 sknt, wind speed knots

7 drct, wind direction

8 gust\_drct, 5 sec gust wind direction

9 gust\_sknt, 5 sec gust wind speed knots

10 vis1\_coeff, visibility 1 coefficient

11 vis1\_nd, visibility 1 night/day

12 vis2\_coeff,

13 vis2\_nd,

14 vis3\_coeff,

15 vis3\_nd,

16 ptype, precipitation type code

17 precip, 1 min precip inches

18 pres1, station 1 pressure inches

19 pres2,

20 pres3,

1 2 3 4 5 6 7 8 9 10 11 16 18 19 20

HYR, HAYWARD, 2005-03-03 10:08, -9,-14, 1, 316, 332,0 ,0.095,N, , , , , NP, ,28.835, 28.833,,

<https://mesonet.agron.iastate.edu/request/asos/1min.phtml>

visiblility

<http://www.moratech.com/aviation/metar-class/metar-pg8-RVR.html>

precipitation

<https://thepointsguy.com/guide/metar-decoding-guide/>

* RA: Rain
* SN: Snow
* DZ: Drizzle
* GR: Hail
* GS: Small hail/Snow pellets
* IC: Ice crystals
* PL: Snow pellets

<https://www.excel-easy.com/examples/import-export-text-files.html>

setwd("C:/Users/Admin/Desktop")

getwd()

**length(list.files())**

**head(list.files())**

**#check if file 'analysis3.R' exists in working directory**

**'analysis3.R' %in% list.files()**

# enter data into R Statology

[How to Use setwd / getwd in R (With Examples) - Statology](https://www.statology.org/setwd-getwd-in-r/#:~:text=You%20can%20use%20the%20following%20functions%20in%20R,2%20setwd%20%28%E2%80%98Path%2FTo%2FSome%2FDirectory%E2%80%99%29%20%E2%80%93%20Set%20current%20working%20directory)

<https://www.statology.org/setwd-getwd-in-r/#:~:text=You%20can%20use%20the%20following%20functions%20in%20R,2%20setwd%20%28‘Path%2FTo%2FSome%2FDirectory’%29%20–%20Set%20current%20working%20directory>

**data1 <- read.csv("C:\\Users\\Bob\\Desktop\\data.csv", header=TRUE, stringsAsFactors=FALSE)**

airport <- read.csv(**"hayward\_sm\_cut.txt")**

**typeof(airport)**

df <-data.frame(airport) # but dropped some rows

**df %>% select(-points)**

**df %>% select(-(position:rebounds))**

**#remove columns that contain the word 'points'**

**df %>% select(-contains('points'))**

**#remove columns that start with 'po'**

**df %>% select(-starts\_with('po'))**

**#remove columns that end with 's'**

**df %>% select(-ends\_with('s'))**

**#remove columns in position 1 and 4**

**df %>% select(-1, -4)**

**df %>% select(-1, -4)**

**#remove columns var1 and var3**

**new\_df <- subset(df, select = -c(var1, var3))**

**new\_df <- subset(df, select = -c(var1, var3))**

**#remove first and fourth columns**

**new\_df <- subset(df, select = -c(1, 4))**

sm\_airport <- subset(airport, select = -c(2, 12, 13, 14, 15, 19,20))

library(dplyr)

**sm\_airport %>% select(-14)**

write.csv(sm\_airport, file = "my\_data.csv")

write.csv(sm\_airport, file = ("D:/my\_data.csv"))

some how can not remove last column?

sm\_airport <- subset(sm\_airport, select = c(14))

shoes

[🔥ON THIS WEEK SALE 70% OFF🔥 Men's Casual Hand Stitching Leather Arch - Tartepl](https://tartepl.com/products/leatherarcshoesbuy?utm_source=mediago&utm_medium=referral&utm_campaign=0929-%E7%89%9B%E7%9A%AE%E9%9E%8B&utm_content=+They%27re+Like+Walking+On+Clouds+-+Now+70%25+Off%21+&utm_term=4649462&a0v5la7bquf89=831b3db47a4c1d6e24527bb485f0e393&uy3ubftvh0u6o8=d085c1239c6b98ff4d211730f8d9e003&cusduxj27i=1420420&xnfrr0ncac=20685&zsmoi87pih9=trace.mediago.io&lzzgnpz8d=fd2c78e45533ef29df88e75fbc16275a)

#install.packages("Rtools") # required for projects

#install.packages("dygraphs")

#install.packages('nycflights13')

#install.packages('Rtools')

## rm(list = ls(all.names = TRUE))

library(xts)

library(dygraphs)

library(tidyverse)

library(lubridate)

library(zoo)

library(fable)

library(tsibble)

library(Rtools) # ?weather # see what is in weather

library(nycflights13)

library(bearer)

library(fpp3)

jfk <- weather %>%

filter(origin == "JFK") # get weather data 2013 for JFK

colnames(jfk) # look at variables

head( jfk %>% select(-c(origin, wind\_gust)) -> jfk1) # look at data & remove column origin

typeof(jfk1) # jfk1 is a list

# here

jfk1$time\_hour <- ymd\_hms(jfk1$time\_hour) # ymd\_h would not work, only ymd\_hms

head(jfk1) # each point is an hour reading

p <- ggplot(jfk1, aes(x=time\_hour, y=temp)) +

geom\_point() + ggtitle("Scatter Plot JFK 2019; Temp vs Date; hourly readings") +

xlab("Date") + ylab("Temperature (**Fahrenheit)**")

p # plots temperature for each hour of each day for the year

# check for NA in each column of jfk1 then fill

pres\_na <- sum(is.na(jfk1$pressure)) # 831

windd\_na <- sum(is.na(jfk1$wind\_dir)) # 51

windsp\_na <- sum(is.na(jfk1$wind\_speed)) # 3

jfk2 = jfk1

jfk2 %>%

fill(pressure,wind\_speed, wind\_dir) -> njfk # fills NA’s down

head(njfk) # NAs still in jfk1, but none in njfk, 8706 rows 1 per hour

### here

njfk %>% relocate(time\_hour) -> njfk1 # move time\_hour to 1st column

colnames(njfk1) # check column names & sequence

# [1] "time\_hour" "temp" "dewp" "humid" "wind\_dir" "wind\_speed"

# "precip" "pressure" "visib"

njfk1 %>% # works reduces to daily data using mean hours

group\_by(year, month, day) %>%

summarise(

meantemp = mean(temp, na.rm = TRUE),

meandewp = mean(dewp, na.rm = TRUE),

meanhumid = mean(humid, na.rm = TRUE),

meanwinddir = mean(wind\_dir, na.rm = TRUE),

meanwindsp = mean(wind\_speed, na.rm = TRUE),

meanprecip = mean(precip, na.rm = TRUE),

meanpressure = mean(pressure, na.rm = TRUE),

meanvisib = mean(visib, na.rm = TRUE)) -> dnjfk

# now only daily mean data, 364 rows in dnjfk

is\_tibble(dnjfk)

jfk <- weather %>%

filter(origin == "JFK") # get weather data 2013 for JFK

colnames(jfk) # look at variables

head( jfk %>% select(-c(origin, wind\_gust)) -> jfk1) # look at data & remove column origin

typeof(jfk1) # jfk1 is a list

atb = jfk

is.tibble(jfk)

[1] "origin" "year" "month" "day" "hour" "temp" "dewp" "humid" "wind\_dir"

[10] "wind\_speed" "wind\_gust" "precip" "pressure" "visib" "time\_hour"

is\_tibble(as\_tibble(atb))

prison <- readr::read\_csv("https://OTexts.com/fpp3/extrafiles/prison\_population.csv")

prison <- prison %>%

mutate(Quarter = yearquarter(Date)) %>%

select(-Date) %>%

as\_tsibble(key = c(State, Gender, Legal, Indigenous),

index = Quarter)

mytb <- jfk %>%

select(-time\_hour) %>%

as\_tibble(key = c(year,month,day,hour,temp,dewp,humid,wind\_speed,precip,pressure,visib), index = Minuet)

prison

autoplot(atb, temp) +

labs(title = "Ansett airlines economy class",

subtitle = "Melbourne-Sydney",

y = "temperature")

# here

jfk1$time\_hour <- ymd\_hms(jfk1$time\_hour) # ymd\_h would not work, only ymd\_hms

head(jfk1) # each point is an hour reading

p <- ggplot(jfk1, aes(x=time\_hour, y=temp)) +

geom\_point() + ggtitle("Scatter Plot JFK 2019; Temp vs Date; hourly readings") +

xlab("Date") + ylab("Temperature (**Fahrenheit)**")

p # plots temperature for each hour of each day for the year

# check for NA in each column of jfk1 then fill

pres\_na <- sum(is.na(jfk1$pressure)) # 831

windd\_na <- sum(is.na(jfk1$wind\_dir)) # 51

windsp\_na <- sum(is.na(jfk1$wind\_speed)) # 3

jfk2 = jfk1

jfk2 %>%

fill(pressure,wind\_speed, wind\_dir) -> njfk # fills NA’s down

head(njfk) # NAs still in jfk1, but none in njfk, 8706 rows 1 per hour

njfk %>% relocate(time\_hour) -> njfk1 # move time\_hour to 1st column

colnames(njfk1) # check column names & sequence

# [1] "time\_hour" "temp" "dewp" "humid" "wind\_dir" "wind\_speed"

# "precip" "pressure" "visib"

njfk1 %>% # works reduces to daily data using mean hours

group\_by(year, month, day) %>%

summarise(

meantemp = mean(temp, na.rm = TRUE),

meandewp = mean(dewp, na.rm = TRUE),

meanhumid = mean(humid, na.rm = TRUE),

meanwinddir = mean(wind\_dir, na.rm = TRUE),

meanwindsp = mean(wind\_speed, na.rm = TRUE),

meanprecip = mean(precip, na.rm = TRUE),

meanpressure = mean(pressure, na.rm = TRUE),

meanvisib = mean(visib, na.rm = TRUE)) -> dnjfk

# now only daily mean data, 364 rows

dnjfk %>% select(meantemp, meandewp) -> test # make ts small

myts1 <- ts(test)

# make digraph

dygraph(myts1, main = "JFK Temperatures") %>%

dyRangeSelector()

dygraph(myts1, main = "JFK temps") %>%

dyRangeSelector(dateWindow = c("2013-01-01", "2013-12-3"))

dygraph(myts1, main = "JFK Temperatures") %>%

dyRangeSelector(height = 9, strokeColor = "") #20

# by month reduce size to 12 rows by averaging

njfk1 %>%

group\_by(year, month) %>%

summarise(

meantemp = mean(temp, na.rm = TRUE),

meandewp = mean(dewp, na.rm = TRUE),

meanhumid = mean(humid, na.rm = TRUE),

meanwinddir = mean(wind\_dir, na.rm = TRUE),

meanwindsp = mean(wind\_speed, na.rm = TRUE),

meanprecip = mean(precip, na.rm = TRUE),

meanpressure = mean(pressure, na.rm = TRUE),

meanvisib = mean(visib, na.rm = TRUE)) -> mjfk # works reduces to monthly data

nrow(mjfk)

mjfk

mjfk$meantemp

temperature <- round(mjfk$meantemp, digits = 1)

rainfall <- round(mjfk$meanprecip, digits = 1)

# define mts with distinct y-axis scales

temperature <- ts(frequency = 12, start = c(2013, 1),

data = c(35.4, 34.2, 39.5, 50.1, 59.3, 70.0, 78.7, 73.8 ,66.9, 59.8 ,45.1 ,38.6))

rainfall <- ts(frequency = 12, start = c(2013, 1),

data = c(0.0033, 0.0041, 0.0030, 0.0025 ,0.0044 ,0.0110, 0.0030, 0.0037 ,0.0027 ,0.0004 ,0.0036, 0.0063))

weather <- cbind(rainfall, temperature)

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

dyOptions(stackedGraph = TRUE) %>%

dyRangeSelector(height = 20)

weather.updateOptions({

dyOptions(stackedGraph = TRUE) %>%

dyRangeSelector(height = 20, strokeColor = "")

#new\_option2: value2

});

dyRangeSelector(height = 20, strokeColor = "")

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

#### here ################### https://stackoverflow.com/questions/23224142/converting-data-frame-to-xts-order-by-requires-an-appropriate-time-based-object

#df$Date <- as.Date(paste(df$year, df$month, sep="-"), "%Y-%M")

Range selector: <https://rstudio.github.io/dygraphs/gallery-range-selector.html>

<https://rstudio.github.io/dygraphs/gallery-range-selector.html>

https://rstudio.github.io/dygraphs/gallery-axis-options.html

typeof(mjfk$meantemp)

# define mts with distinct y-axis scales

mtemp <- xts(frequency = 12, start = c(2013, 1),

data = mjfk$meantemp)

mrain <- xts(frequency = 12, start = c(2013, 1),

data = mjfk$meanprecip)

weather <- cbind(mrain, mtemp)

weather

# assign the "mrain" series to the y2 axis

dygraph(weather) %>%

dySeries("rain", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rain", axis = 'y2')

# define mts with distinct y-axis scales

temperature <- ts(frequency = 12, start = c(1980, 1),

data = c(7.0, 6.9, 9.5, 14.5, 18.2, 21.5,

25.2, 26.5, 23.3, 18.3, 13.9, 9.6))

rainfall <- ts(frequency = 12, start = c(1980, 1),

data = c(49.9, 71.5, 106.4, 129.2, 144.0, 176.0,

135.6, 148.5, 216.4, 194.1, 95.6, 54.4))

weather <- cbind(rainfall, temperature)

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

Calculate the yearly mean

period.apply(xts5,INDEX=ep2,FUN=mean)

# head(dnjfk)

# myts <-ts(dnjfk)

# use : <https://rstudio.github.io/dygraphs/gallery-axis-options.html>

# <https://rpubs.com/chrisbrunsdon/pipelines>

<https://www.neonscience.org/resources/learning-hub/tutorials/dc-time-series-subset-dplyr-r>

<https://www.datacamp.com/cheat-sheet/xts-cheat-sheet-time-series-in-r>

# make digraph

dygraph(myts, main = "JFK Temperatures") %>%

dyRangeSelector()

dygraph(myts, main = "JFK temps") %>%

dyRangeSelector(dateWindow = c("2013-01-01", "2013-12-3"))

dygraph(myts, main = "JFK Temperatures") %>%

dyRangeSelector(height = 20, strokeColor = "")

## rm(list = ls(all.names = TRUE))

#install.packages("dygraphs")

library(xts)

library(dygraphs)

library(tidyverse)

library(lubridate)

#install.packages('nycflights13')

library(nycflights13)

library(zoo)

library(fable)

library(tsibble)

install.packages("Rtools") # required for projects

library(Rtools)

# ?weather

jfk <- weather %>%

filter(origin == "JFK") # get weather data 2013 for JFK

colnames(jfk) # look at variables

head(jfk1 <- jfk %>% select(-c(origin, wind\_gust))) # look at data

jfk1$time\_hour <- ymd\_hms(jfk1$time\_hour) # ymd\_h would not work, only ymd\_hms

head(jfk1) # each point is an hour reading

p <- ggplot(jfk1, aes(x=time\_hour, y=temp)) +

geom\_point() + ggtitle("Scatter Plot JFK 2019; Temp vs Date; hourly readings") +

xlab("Date") + ylab("Temperature (**Fahrenheit)**")

p # plots temperature for each hour of each day for the year

# check for NA in each column of jfk1 then fill

pres\_na <- sum(is.na(jfk1$pressure)) # 831

windd\_na <- sum(is.na(jfk1$wind\_dir)) # 51

windsp\_na <- sum(is.na(jfk1$wind\_speed)) # 3

jfk2 = jfk1

jfk2 %>%

fill(pressure,wind\_speed, wind\_dir) -> njfk # fills NA’s down

head(njfk) # NAs still in jfk1, but none in njfk, 8706 rows 1 per hour

njfk %>% relocate(time\_hour) -> njfk1 # move time\_hour to 1st column

colnames(njfk1) # check column names & sequence

# [1] "time\_hour" "temp" "dewp" "humid" "wind\_dir" "wind\_speed"

# "precip" "pressure" "visib"

njfk1 %>% # works reduces to daily data using mean hours

group\_by(year, month, day) %>%

summarise(

meantemp = mean(temp, na.rm = TRUE),

meandewp = mean(dewp, na.rm = TRUE),

meanhumid = mean(humid, na.rm = TRUE),

meanwinddir = mean(wind\_dir, na.rm = TRUE),

meanwindsp = mean(wind\_speed, na.rm = TRUE),

meanprecip = mean(precip, na.rm = TRUE),

meanpressure = mean(pressure, na.rm = TRUE),

meanvisib = mean(visib, na.rm = TRUE)) -> dnjfk

# now only daily mean data, 364 rows

dnjfk %>% select(meantemp, meandewp) -> test # make ts small

myts1 <- ts(test)

# make digraph

dygraph(myts1, main = "JFK Temperatures") %>%

dyRangeSelector()

dygraph(myts1, main = "JFK temps") %>%

dyRangeSelector(dateWindow = c("2013-01-01", "2013-12-3"))

dygraph(myts1, main = "JFK Temperatures") %>%

dyRangeSelector(height = 50, strokeColor = "") #20

# by month reduce size to 12 rows by averaging

njfk1 %>%

group\_by(year, month) %>%

summarise(

meantemp = mean(temp, na.rm = TRUE),

meandewp = mean(dewp, na.rm = TRUE),

meanhumid = mean(humid, na.rm = TRUE),

meanwinddir = mean(wind\_dir, na.rm = TRUE),

meanwindsp = mean(wind\_speed, na.rm = TRUE),

meanprecip = mean(precip, na.rm = TRUE),

meanpressure = mean(pressure, na.rm = TRUE),

meanvisib = mean(visib, na.rm = TRUE)) -> mjfk # works reduces to monthly data

nrow(mjfk)

mjfk

mjfk$meantemp

temperature <- round(mjfk$meantemp, digits = 1)

rainfall <- round(mjfk$meanprecip, digits = 1)

# <http://datacamp-community-prod.s3.amazonaws.com/72771032-0653-4d87-a798-4a83879e99c8>

data(weather)

data(weather)

<https://r-lang.com/as-date-function-in-r-with-example/#:~:text=as.Date%20in%20R%20The%20as.Date%20%28%29%20is%20a,since%201970-01-01%2C%20with%20negative%20values%20for%20earlier%20dates>.

xts5 <- as.xts(weather)

> core\_data <- coredata(xts2)  
> index(xts1)

temp2 <- xts(frequency = 12, start = c(2013, 1),

data = c(35.4, 34.2, 39.5, 50.1, 59.3, 70.0, 78.7, 73.8 ,66.9, 59.8 ,45.1 ,38.6))

print(temp2)

xts1 <- xts(x=1:10, order.by=Sys.Date()-1:10)  
> data <- rnorm(5)  
dates <- seq(as.Date("2013-01"),length=11,by="months")

dates <- c("11/20/80", "11/20/91", "11/20/1993", "09/10/93")

dt <- as.Date(dates, "%m/%d/%y")

dt

data(sample\_matrix)

x <- as.xts(sample\_matrix, myattr=100)

coredata(x)

xcoredata(x)

coredata(temp2)

coredata(temperature)

dat <- xts(x = rnorm(n = 5),

order.by = seq(as.Date("2016-01-01"),

length = 5,

by = "days"),

hayek\_bday = as.POSIXct("1899-05-08"))

dat <- xts(x = rnorm(n = 5),

order.by = seq(as.Date("2016-01-01"),

length = 5,

by = "days"),

hayek\_bday = as.POSIXct("1899-05-08"))

dates

`?`(strptime)

\# define mts with distinct y-axis scales

temperature <- ts(frequency = 12, start = c(2013, 1),

data = c(35.4, 34.2, 39.5, 50.1, 59.3, 70.0, 78.7, 73.8 ,66.9, 59.8 ,45.1 ,38.6))

rainfall <- ts(frequency = 12, start = c(2013, 1),

data = c(0.0033, 0.0041, 0.0030, 0.0025 ,0.0044 ,0.0110, 0.0030, 0.0037 ,0.0027 ,0.0004 ,0.0036, 0.0063))

weather <- cbind(rainfall, temperature)

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

#### here ###################

https://stackoverflow.com/questions/23224142/converting-data-frame-to-xts-order-by-requires-an-appropriate-time-based-object

#df$Date <- as.Date(paste(df$year, df$month, sep="-"), "%Y-%M")

https://rstudio.github.io/dygraphs/gallery-axis-options.html

typeof(mjfk$meantemp)

# define mts with distinct y-axis scales

mtemp <- xts(frequency = 12, start = c(2013, 1),

data = mjfk$meantemp)

mrain <- xts(frequency = 12, start = c(2013, 1),

data = mjfk$meanprecip)

weather <- cbind(mrain, mtemp)

weather

# assign the "mrain" series to the y2 axis

dygraph(weather) %>%

dySeries("rain", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rain", axis = 'y2')

# define mts with distinct y-axis scales

temperature <- ts(frequency = 12, start = c(1980, 1),

data = c(7.0, 6.9, 9.5, 14.5, 18.2, 21.5,

25.2, 26.5, 23.3, 18.3, 13.9, 9.6))

rainfall <- ts(frequency = 12, start = c(1980, 1),

data = c(49.9, 71.5, 106.4, 129.2, 144.0, 176.0,

135.6, 148.5, 216.4, 194.1, 95.6, 54.4))

weather <- cbind(rainfall, temperature)

# assign the "rainfall" series to the y2 axis

dygraph(weather) %>%

dySeries("rainfall", axis = 'y2')

dygraph(weather) %>%

dyAxis("y", label = "Temperature (C)") %>%

dyAxis("y2", label = "Rainfall", independentTicks = TRUE) %>%

dySeries("rainfall", axis = 'y2')

# head(dnjfk)

# myts <-ts(dnjfk)

# use : <https://rstudio.github.io/dygraphs/gallery-axis-options.html>

# <https://rpubs.com/chrisbrunsdon/pipelines>

<https://www.neonscience.org/resources/learning-hub/tutorials/dc-time-series-subset-dplyr-r>

<https://www.datacamp.com/cheat-sheet/xts-cheat-sheet-time-series-in-r>

# make digraph

dygraph(myts, main = "JFK Temperatures") %>%

dyRangeSelector()

dygraph(myts, main = "JFK temps") %>%

dyRangeSelector(dateWindow = c("2013-01-01", "2013-12-3"))

dygraph(myts, main = "JFK Temperatures") %>%

dyRangeSelector(height = 20, strokeColor = "")