

# HW1\_141B\_Jewell

Benjamin Jewell

2023-04-18

The attached code runs to unzip all five location climate/solar data into a single folder, /dt/ in this case, and then reads in the WEA, PVSYST and STAT data. From here it compiles the specified tables contained within and formats them into a proper data frame. The STAT hourly data is then recombined and graphed, and the data frames are either returned or printed.

## **Assumptions & Understanding the data:**

During this procedure I made several assumptions about the data. For one, I assumed that the data was correct and properly formatted (no missing values, inconsistent tables, etc) which I checked by opening the files in a text editor and looking at them. As well I assumed and verified that that each table(s) was the same across all five documents, which was true as well. One assumption I made that turned out to be incorrect was that the tables would be the same size for all documents, and that all tables would start on the same line. This turned out to not be the case, and as such the start and end lines of each table had to be calculated using the grep command.

For the WEA and PVSYST tables I mostly looked at the tables visually to understand their structure and how to break them up. For the STAT files I assumed that the Hourly and Monthly data would be structured the same and thus made a function to read in hourly tables and a function to read in monthly tables. As it turns out this wasn't quite correct, as the Wind Speed Monthly table wrote out their minimum and maximum values slightly differently. Upon discovering this I had to write a custom flag to adjust the index of the numeric values when extracting the minimum and maximum from the Wind Speed Monthly table. As it turned out the Monthly Wind Direction was a bit different from the other tables, and thus my assumptions that all monthly tables were the same was wrong. I decided to just write a custom parser for Monthly Wind Direction that was somewhat similar to the Month Parser rather than trying to make the base function more modular as that seemed easier.

## **Verification & Debugging:**

My main procedure was to check the first three and last three entries of each column in a given table for each variable by visually comparing the two values. I was fairly confident in this method as I never performed any arithmetic on these values, and the same operation was performed on all values, meaning that if the first couple were right, they should all be correct. Of note when cutting up strings I would take a more in depth look than just the first and last value, as these were much more sensitive to problems. For those values that needed to be reformatted or cut out a string I would make sure to check these values with lots of print statements to see that the output fit what I wanted. In generally this combination of visual comparison and a hefty use of print statements were my main debugging/verification tools.

As well I also wrote a debug\_stats() function to use on a few tables which checked the unique variables for a given table, along with the min and max values. For the monthly tables in which we needed to verify the minimum and maximum values I wrote a function that checked that these lined up with the data from the tables. A function, vis\_inspc() was also added to allow for visualization of each column of a data frame. This was not used to debug the hourly tables, as they were already being printed out separately. Due to the number of graphs this can create, they will only print if the variable DEBUG == TRUE.

## **The Code:**

The code itself is mainly composed of a function to read each table type, with a few expansions and additions. Ggplot2 was imported for graphs. The functions made were:

*unzip\_fct* Unzips the specified list of zip folders

*debug\_stats* Takes a data frame and prints out the unique variables in each column, along with their min and max for debugging purposes

*vis\_inspc* Takes a data frame and the selected columns. For each of those columns if they contain numbers, it will graph them against their column index.

*asnum\_col* Takes a data frame and a list of columns. Returns the given data frame with the indicated columns transformed into numerics

*posix\_convert* takes column(s) from a dataframe and converts them into POSIXct format. Returns a data frame with the indicated columns transformed.

*wea\_parser* Given a path to a .WEA file it returns a data frame of the data contained within

*pvsyst\_parser* Given a path to a .pvsyst file it returns a data frame of the data contained within.

*month\_parser* Takes data as a list of strings from readLines(), along with a start line, an end line and **wind** Boolean. This data should be from a month table from a stat file. Start line and end line pick what subset of the stat file should be read. The **wind** Boolean is a flag to indicate we are reading from Monthly Wind Speed and allows for some minor corrections to our code to ensure no bugs. Returns a data frame.

*wind\_dir\_parser* A custom modified month\_parser. If using OOP this would be a subclass of month\_parser. Only takes data as a list of string from readLines(). Returns a data frame.

*hourly\_parser* Takes data as a list of strings from readLines() along with a start line and an end line. To be used for data from hourly tables. Data is placed in month, hour order, along with a column for month and hour for each data point. Returns a data frame.

*trim\_time* Takes a data frame and a column. Removes white space from the given column so that as.numeric() can be used on them safely.

*combine\_hours* Takes five data frames from hourly tables. Places month data in first column, hour data in second column and then extracts the data columns from the five given data frames adds them to the table aligned with proper month and time. Returns this new data frame.

*graph\_attrs* Takes a data frame from combine\_hours() and graphs all five attributes for each given month of the data frame. Prints out the resulting scatter plot for visual inspection. Returns NONE.

*stat\_parse* Takes a path to a STAT file to parse. Parses the file into 9 separate data frames as specified in the assignment, then combines the 5 hourly data frames into a singular data frame. The combined data frame is then graphed for each month for a total of 12 graphs. The four hourly data frames and the one combine hourly data frame are then either printed or returned.

### **Code Commentary:**

I chose to write out each function for each of the file types and call them individually for each given location. While it would be rather easy to write a main function that runs through the entire process for a given location, I chose this approach given the fact that Rmarkdowm runs through the entire code anyways I didn't see the need, especially since it was not specified and this code is being used for a single assignment. If we were using this code again in the future to read through the data for every location in California for example then I most certainly would automate it that way, but for now it doesn't seem necessary. I chose to print out the data frames and graphs from STAT file rather than returning them for ease of access for actually reading them for my own analysis and for grading. The data frames can be returned inside a vector if **ReturnDataFrames** is set to TRUE, in case that is needed for grading.

### **WORKS CITED:**

[1] <https://statisticsglobe.com/convert-data-frame-column-to-numeric-in-r>

- [2] <https://www.tutorialspoint.com/how-to-access-elements-of-nested-lists-in-r>
- [3] <https://www.geeksforgeeks.org/converting-a-list-to-vector-in-r-language-unlist-function/>
- [4] <https://statisticsglobe.com/find-elements-list-r>
- [5] <https://www.geeksforgeeks.org/how-to-plot-a-subset-of-a-dataframe-using-ggplot2-in-r/#>
- [6] <https://stackoverflow.com/questions/15678261/ggplot-does-not-work-if-it-is-inside-a-for-loop-although-it-works-outside-of-it>

Global Stuff

```
library(ggplot2)
```

```
#Determines if we print out debug plots & print statements, default FALSE due to how long the output of
DEBUG = TRUE
#Determines if stat_parser() returns the data frames as a vector (TRUE) or just prints them out (FALSE)
ReturnDataframes = TRUE
```

Unzip our files into ./dt

```
zip_paths = c('E:/College/UC Davis/STA141B/HW1/dt/USA_CA_Fairfield-San.Francisco.Bay.Reserve.998011_TMYx.2007-2021.zip',
            'E:/College/UC Davis/STA141B/HW1/dt/USA_CA_Marin.County.AP-Gnoss.Field.720406_TMYx.2007-2021.zip',
            'E:/College/UC Davis/STA141B/HW1/dt/USA_CA_Napa.County.AP.724955_TMYx.2007-2021.zip',
            'E:/College/UC Davis/STA141B/HW1/dt/USA_CA_Point.Reyes.Lighthouse.724959_TMYx.2007-2021.zip',
            'E:/College/UC Davis/STA141B/HW1/dt/USA_CA_UC-Davis-University.AP.720576_TMYx.2007-2021.zip')

unzip_fct <- function(zippaths){
  for (i in 1:length(zippaths)){
    unzip(zippaths[i], exdir = './dt')
  }
}

unzip_fct(zip_paths)
```

## Misc Functions

```
#takes the data frame and # of variables/columns
debug_stats <- function(df, n_var){
  print('DEBUG STATS: # of NAs, # of Unique Vals, Min, Max')
  for (i in 1:n_var){
    print(which(is.na(df[,i])))
    print(length(unique(df[,i])))
    print(c(min(df[,i]), max(df[,i]))) }
}

#rewrites a given column(s) as numeric
asnum_col <- function(df, cols_idx){
  for (i in cols_idx){
    df[,i] = as.numeric(df[,i])
  }
  return(df)
}

#converts a row of a:b time into POSIXct time
```

```

posix_convert <- function(df, time_cols){
  #Add 2023, Month, Time and PDT to data

  #convert to POSIX
  for (i in time_cols){
    for (j in c(1:nrow(df))){
      #print(df[j,i])
      dayhour = strsplit(df[j,i], ':')
      time = paste('2023/', j, '/', dayhour[[1]][[1]], ' ', dayhour[[1]][[2]], ':00', sep='')
      #print(paste('TIME:', as.POSIXct(time)))
      df[j,i] = as.character(as.POSIXct(time))
    }
  }
  return(df)
}

#graph each column for visual inspection
vis_inspc <- function(df, cols_idx){
  for (i in cols_idx){
    xaxis = c(1:nrow(df))
    yaxis = df[,i]
    if (is.numeric(yaxis) == TRUE){
      plot(xaxis, yaxis)
    }
    if (is.integer(yaxis) == TRUE){
      plot(xaxis, yaxis)
    }
    if (is.double(yaxis) == TRUE){
      plot(xaxis, yaxis)
    }
  }
}

```

## WEA Parser

```

wea_parser <- function(wea_path){

  #Read in the data & ignore the header
  wea_lines = readLines(wea_path)
  start = grep("weather_data_", wea_lines) + 1
  wea_txt = wea_lines[start:length(wea_lines)]

  #Split our data and then rearrange it for ease of data.frame assembly
  splt_wea = strsplit(wea_txt, " ")
  monthv = vector()
  dayv = vector()
  sdt_tv = vector()
  dniv = vector()
  dhiv = vector()

  for (i in c(1:length(splt_wea))){
    #[1][2] Used to save as int, & find out how to access nested list values
    monthv = c(monthv, as.integer(splt_wea[[i]][[1]]))
  }
}

```

```

dayv = c(dayv, as.integer(splt_wea[[i]][[2]]))
sdt_tv = c(sdt_tv, as.integer(splt_wea[[i]][[3]]))
dniv = c(dniv, as.integer(splt_wea[[i]][[4]]))
dhiv = c(dhiv, as.integer(splt_wea[[i]][[5]]))
}
wea_df = data.frame(monthv, dayv, sdt_tv, dniv, dhiv)

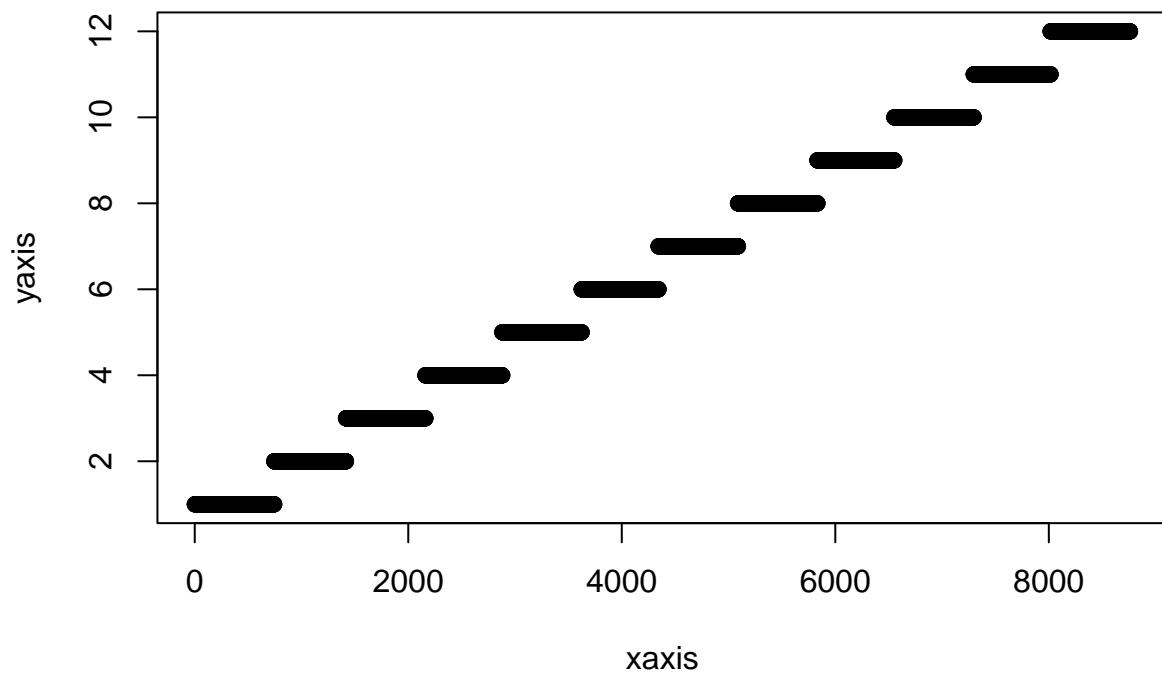
#Define columns names as defined by professor on Piazza
colnames(wea_df) = c("Month", "Day", "Standard_Time", "Direct Normal Irradiance", "Diffuse Horizontal Irradiance")

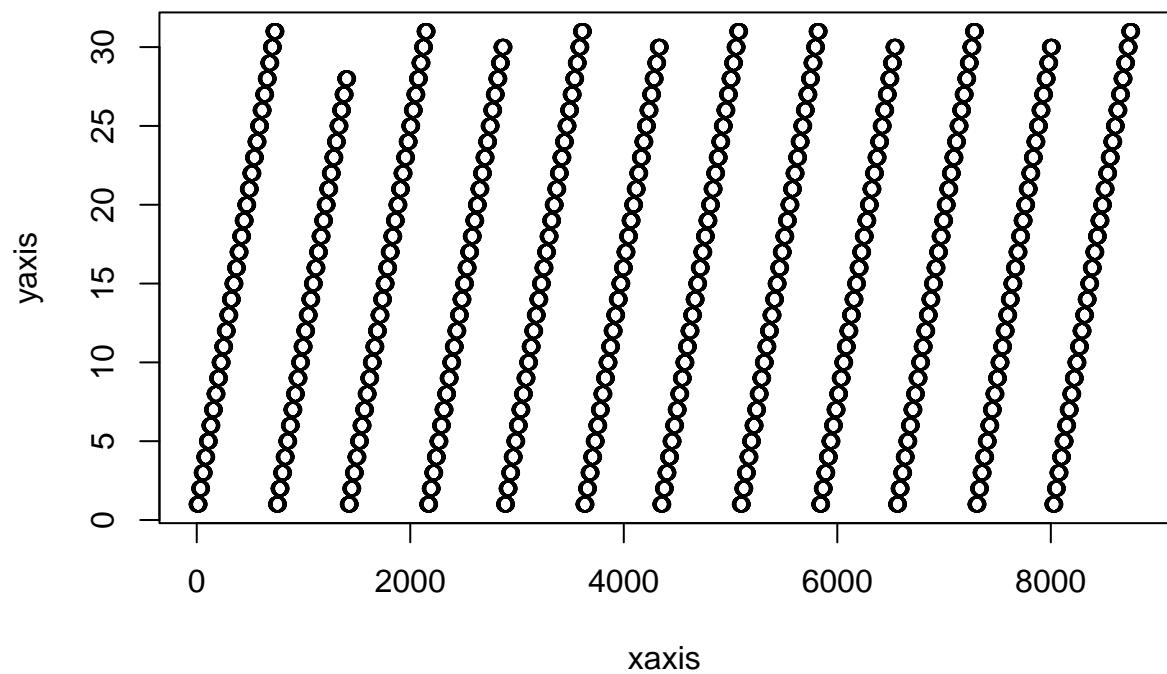
if (DEBUG == TRUE){
  debug_stats(wea_df, 5)
  vis_inspc(wea_df, c(1:ncol(wea_df)))
}
return(wea_df)
}

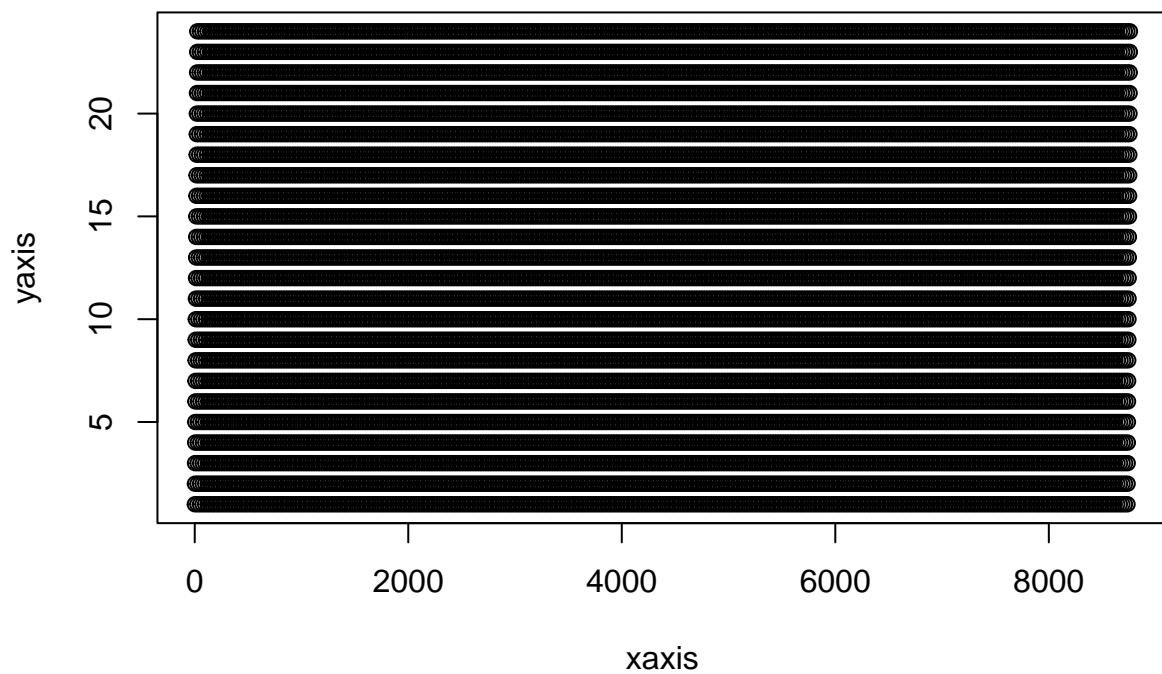
wea_marin = wea_parser('dt/USA_CA_Marin.County.AP-Gnoss.Field.720406_TMYx.2007-2021.wea')

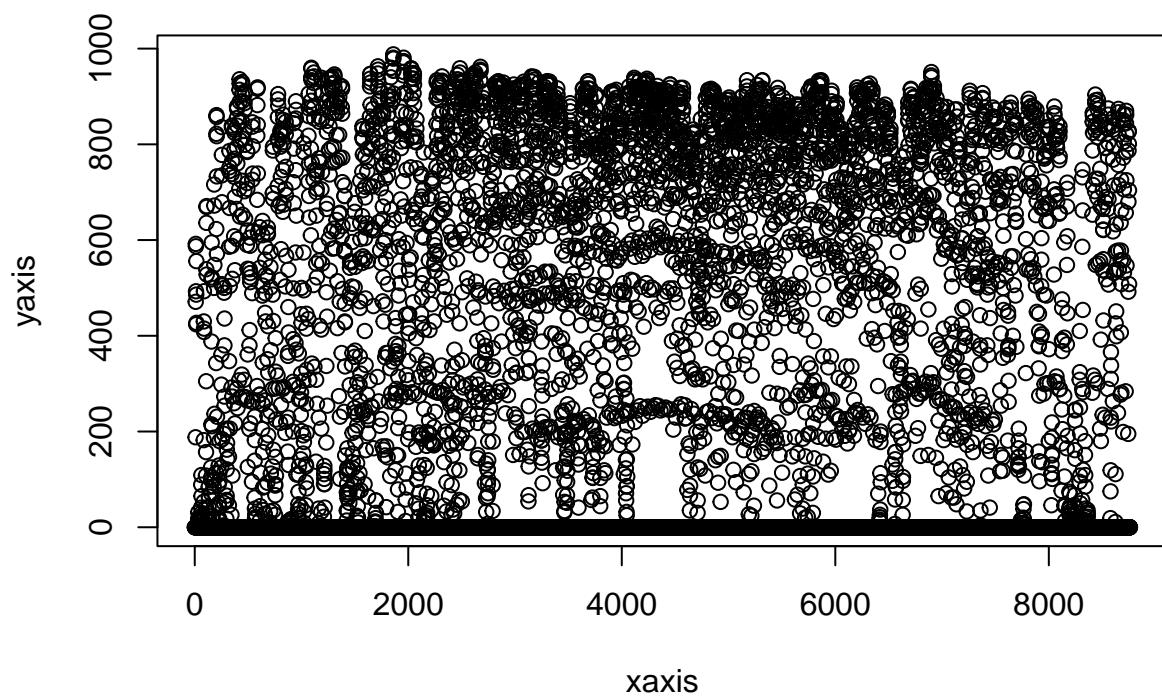
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 930
## [1] 0 988
## integer(0)
## [1] 328
## [1] 0 452

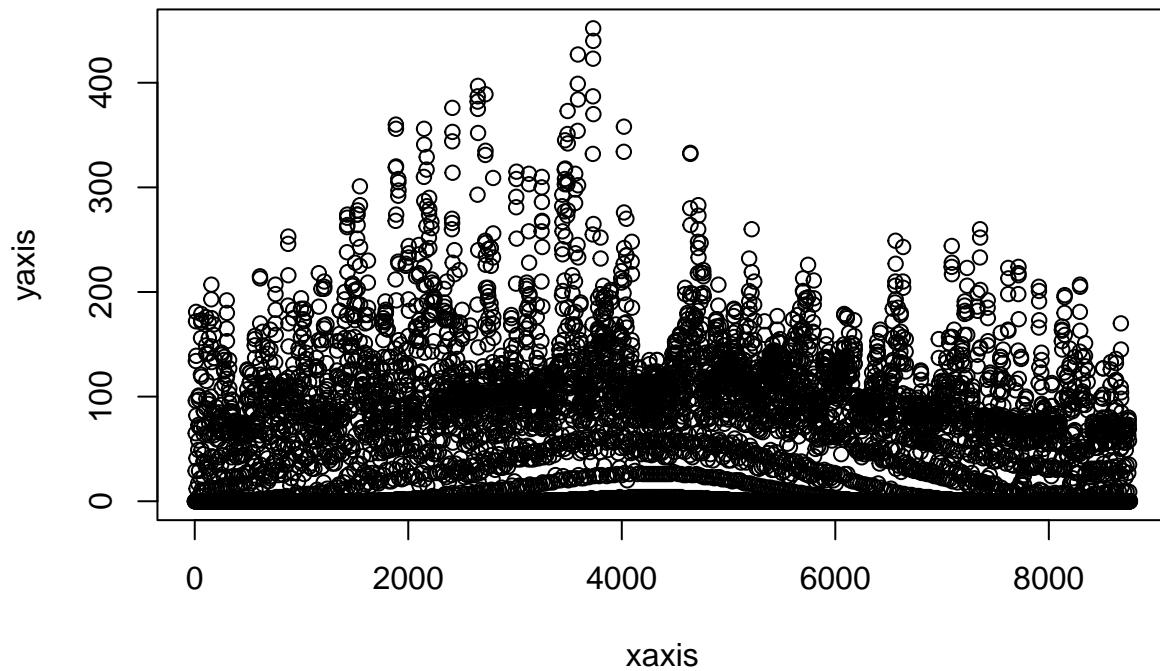
```





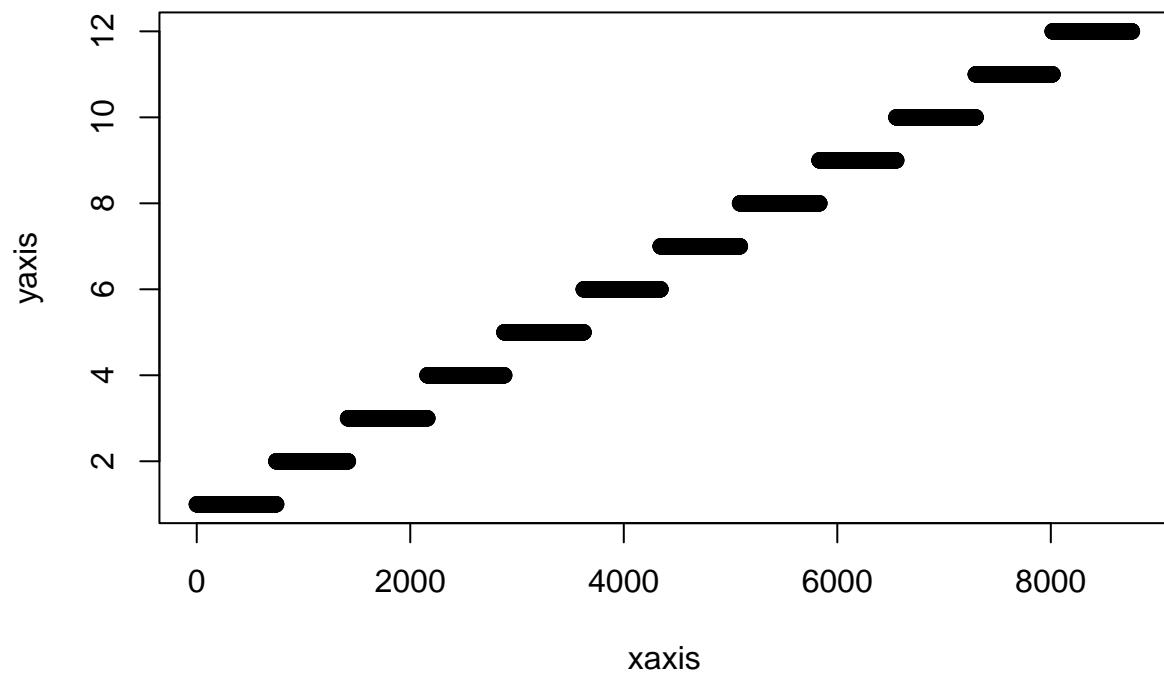


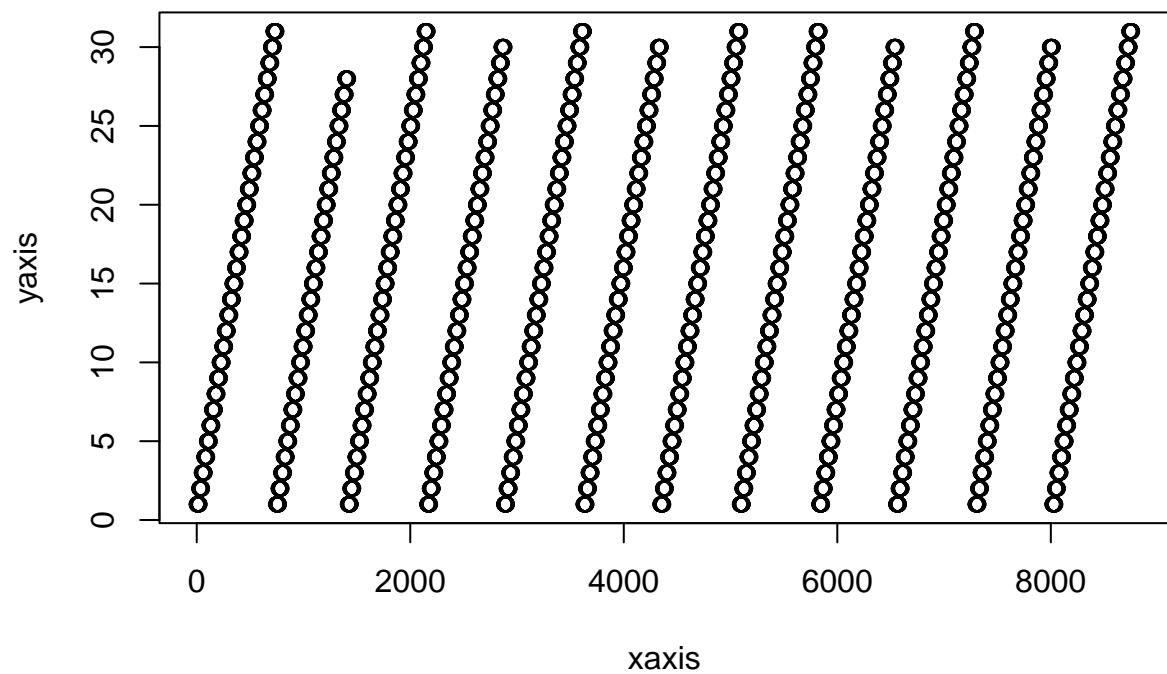


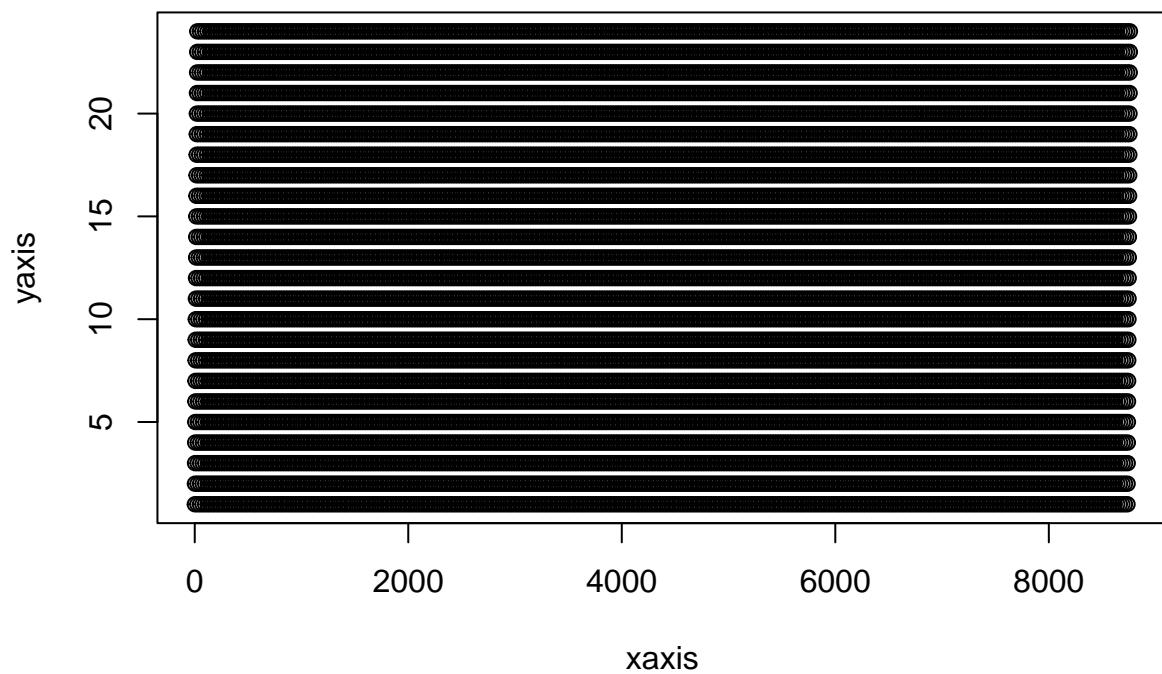


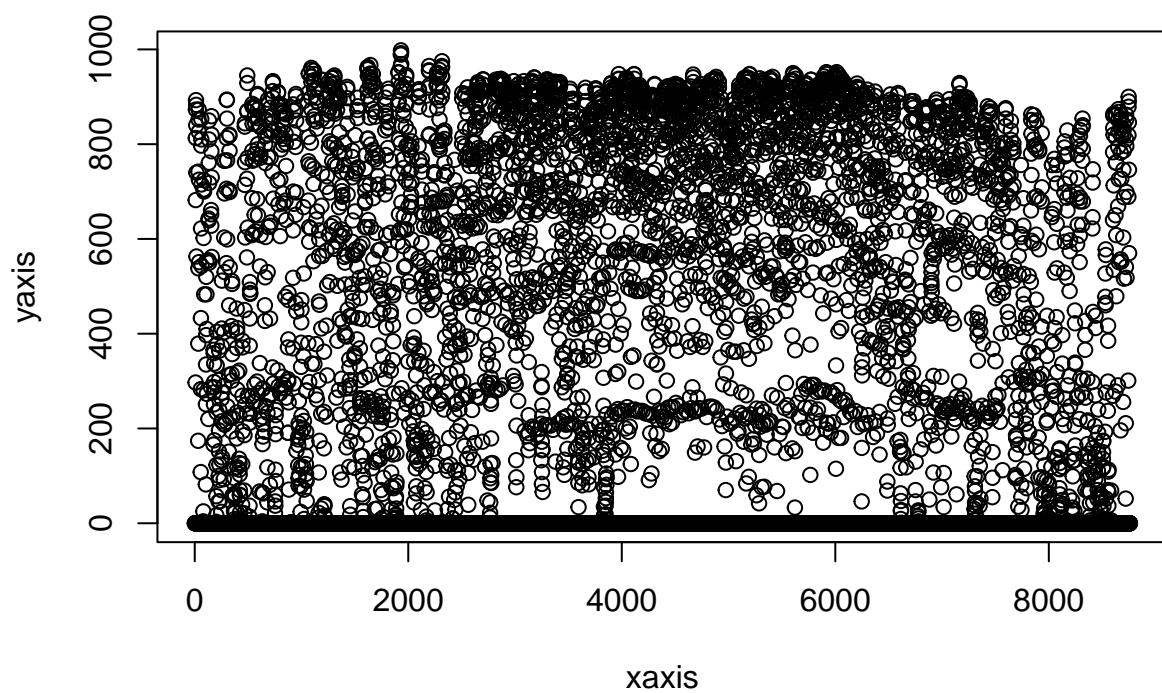
```
wea_SF = wea_parser('dt/USA_CA_Fairfield-San.Francisco.Bay.Reserve.998011_TMYx.2007-2021.wea')
```

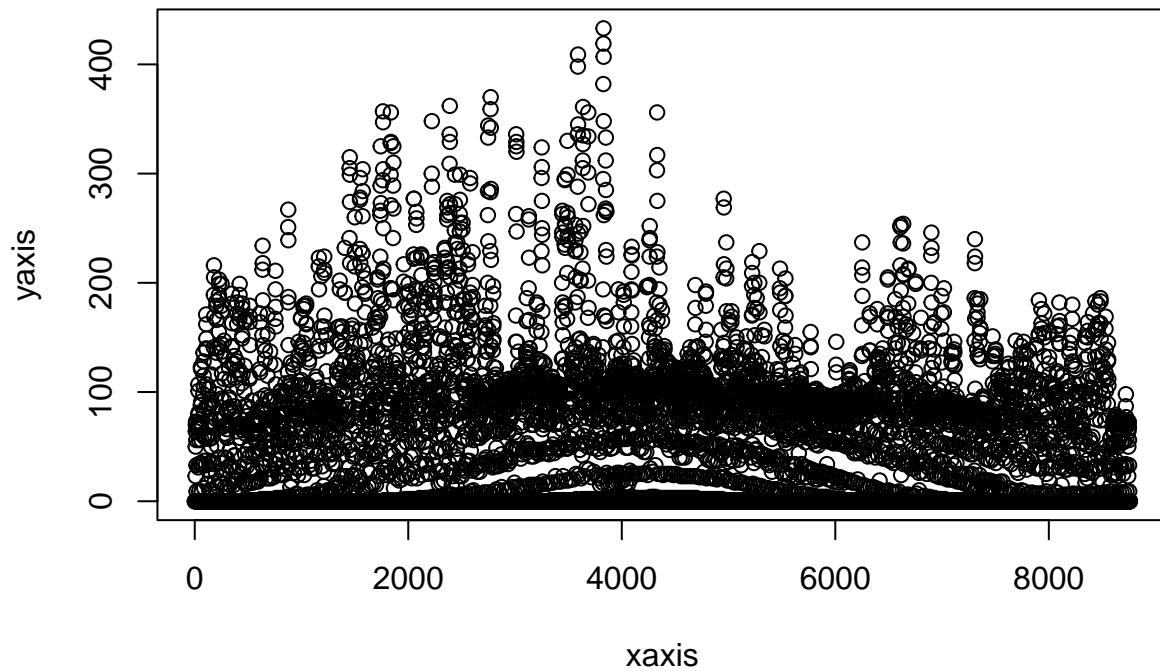
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 934
## [1] 0 998
## integer(0)
## [1] 326
## [1] 0 433
```





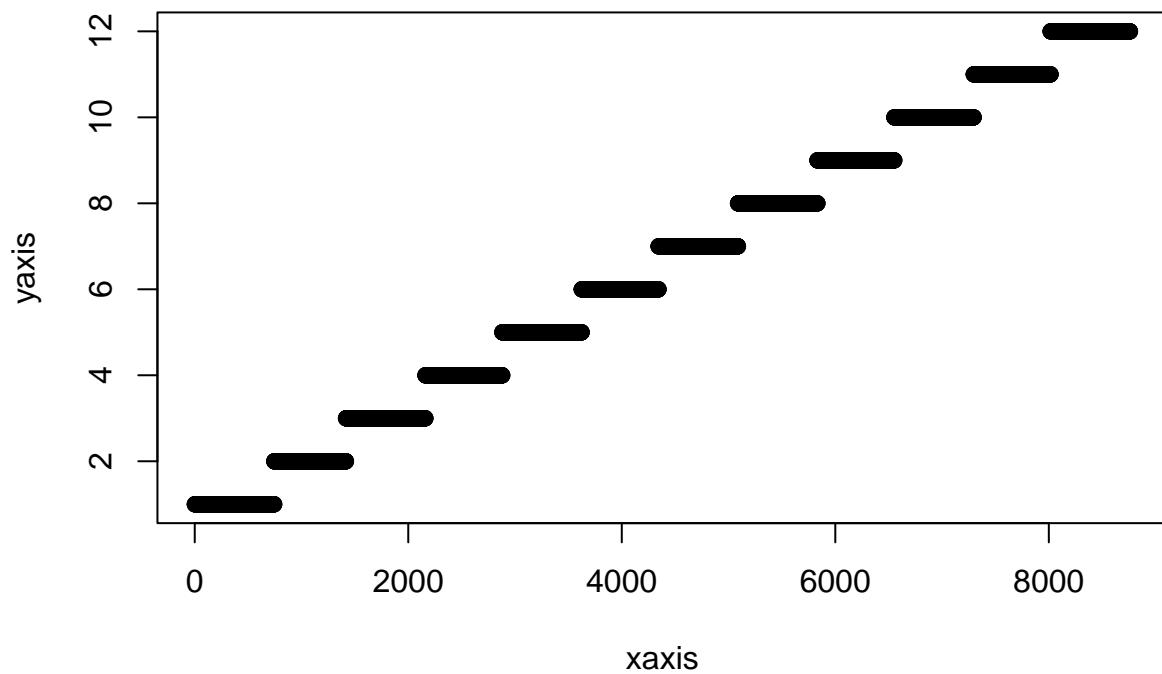


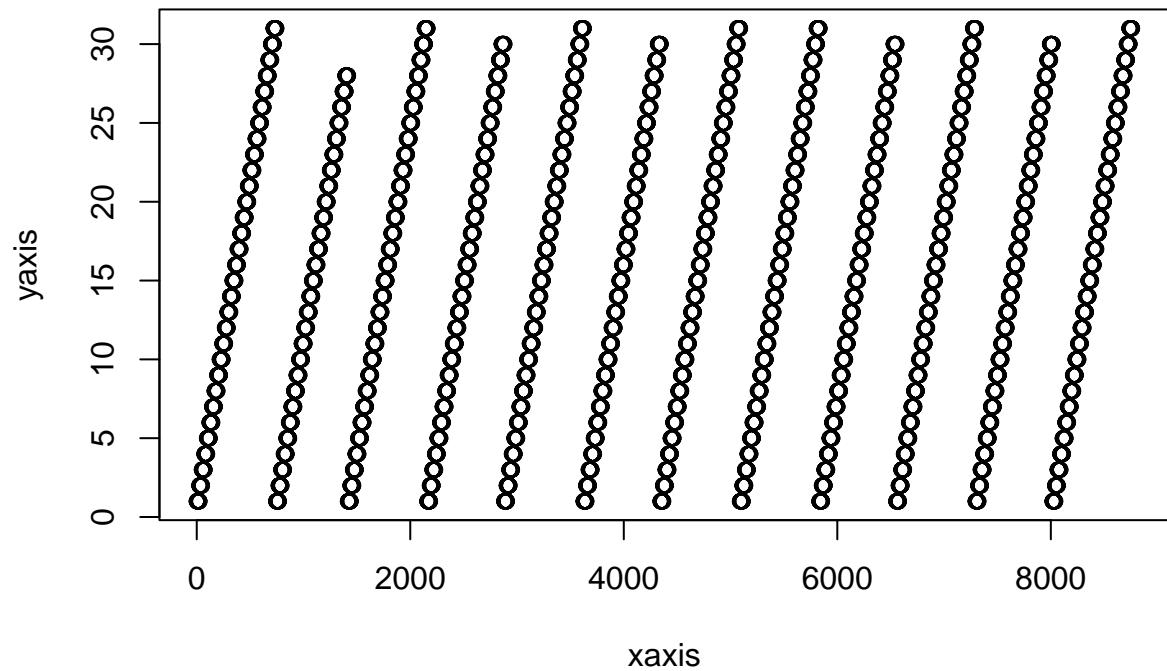


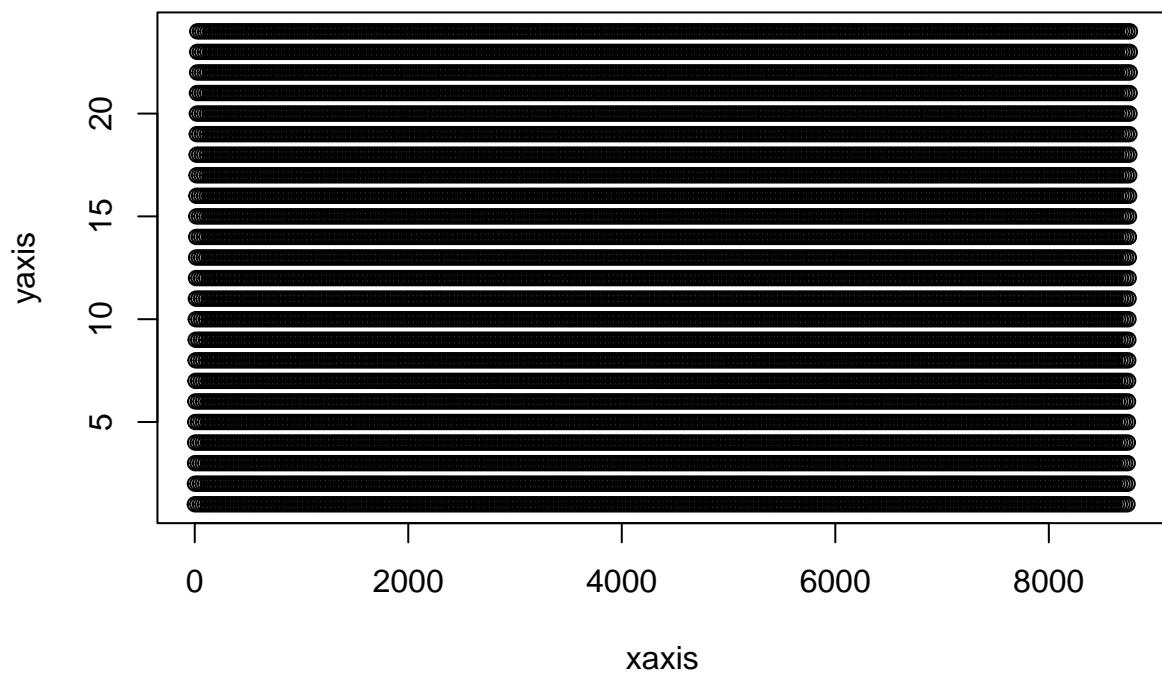


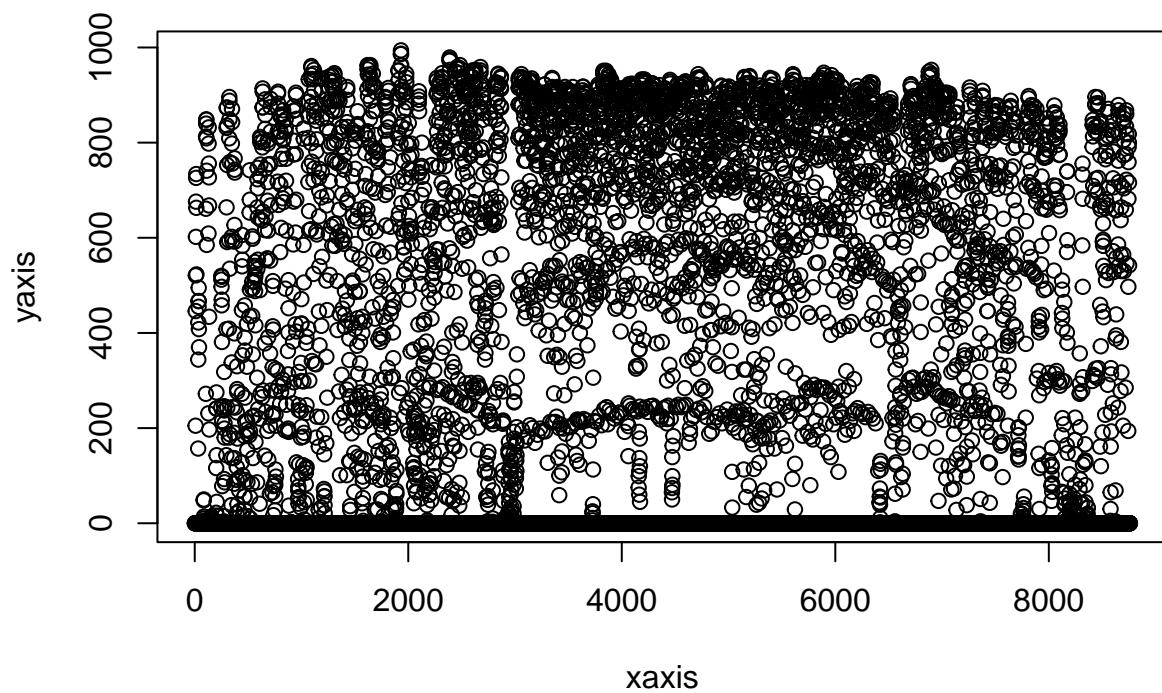
```
wea_napa = wea_parser('dt/USA_CA_Napa.County.AP.724955_TMYx.2007-2021.wea')
```

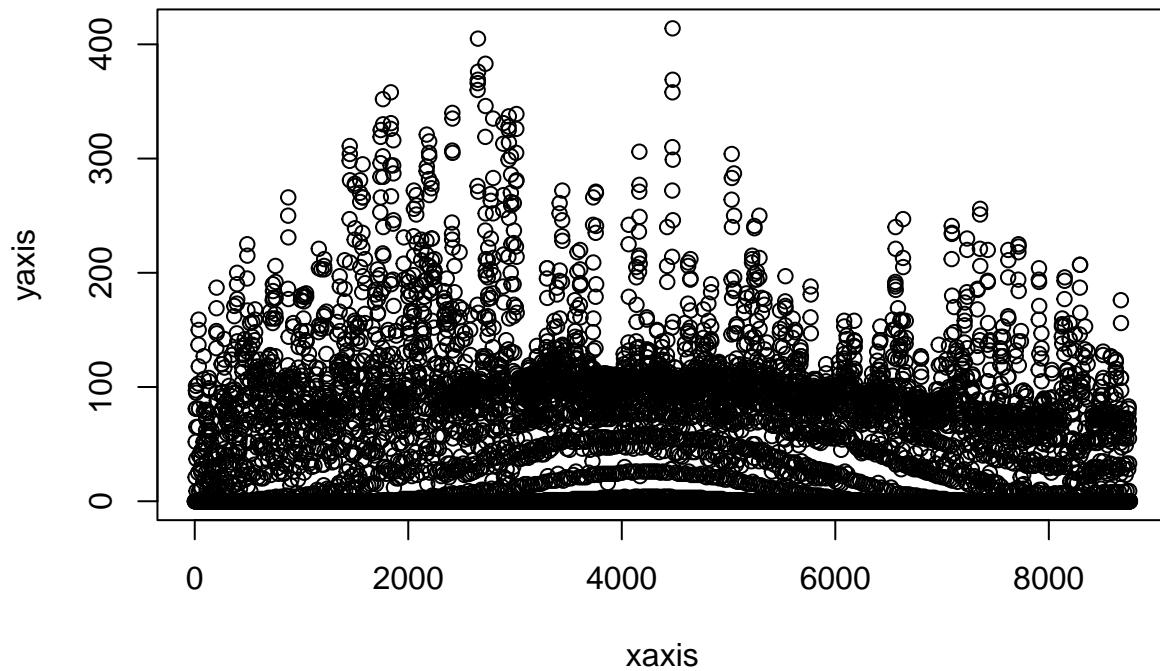
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 927
## [1] 0 994
## integer(0)
## [1] 312
## [1] 0 414
```





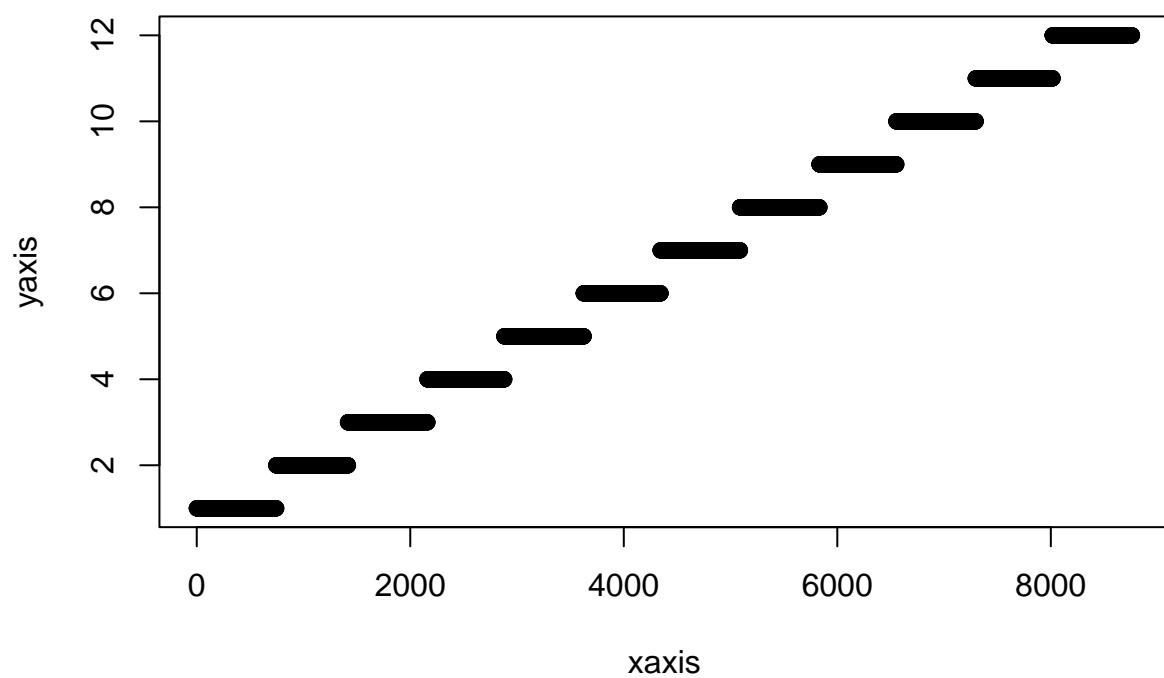


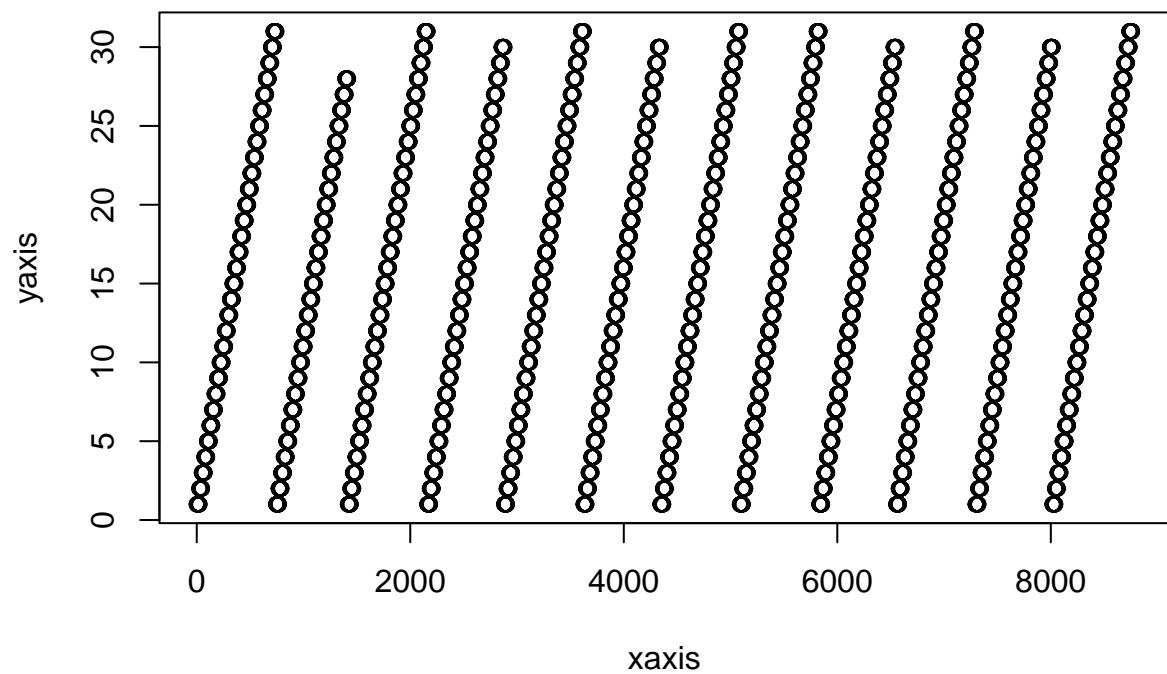


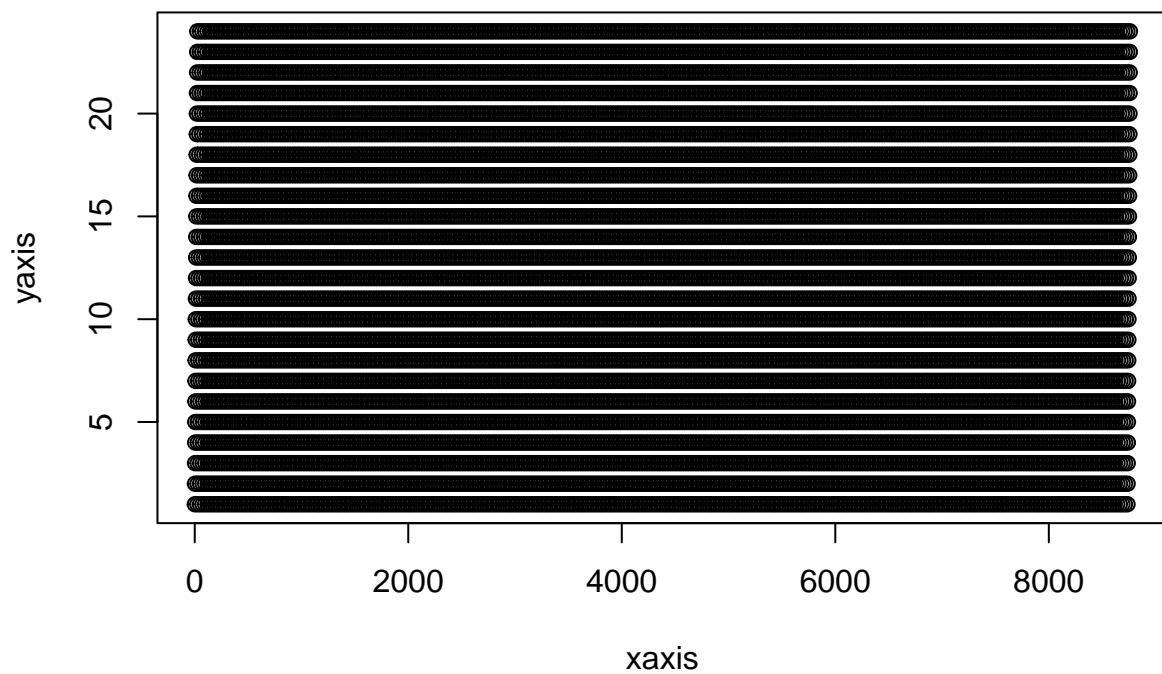


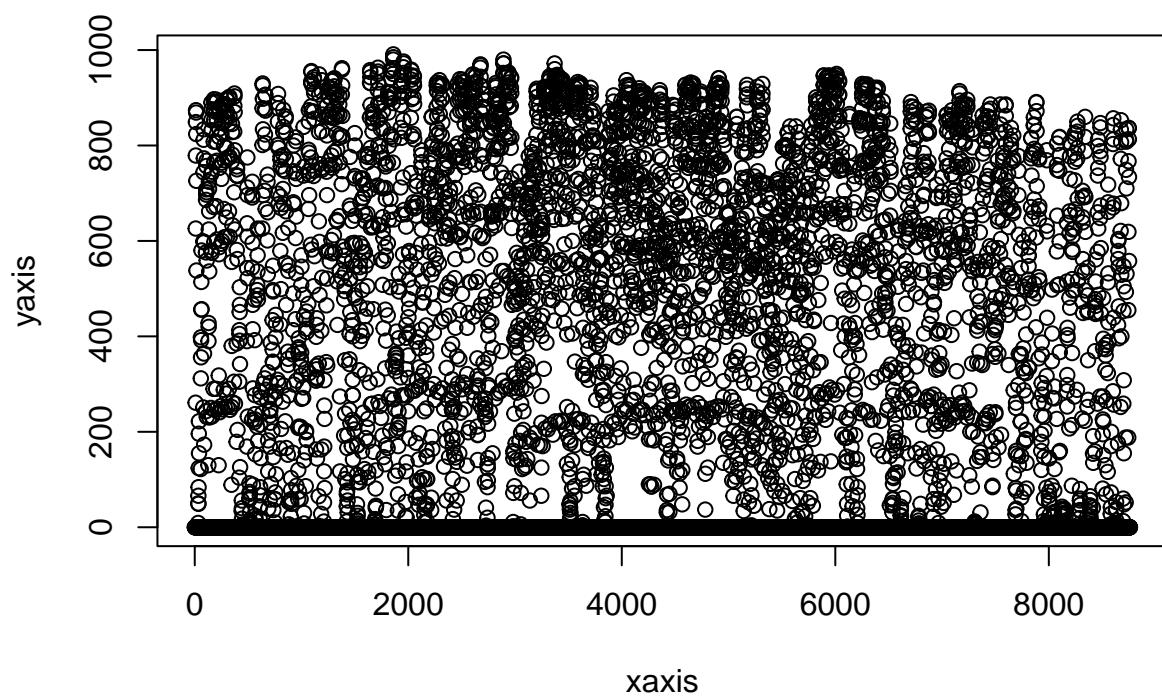
```
wea_reyes = wea_parser('dt/USA_CA_Point.Reyes.Lighthouse.724959_TMYx.2007-2021.wea')
```

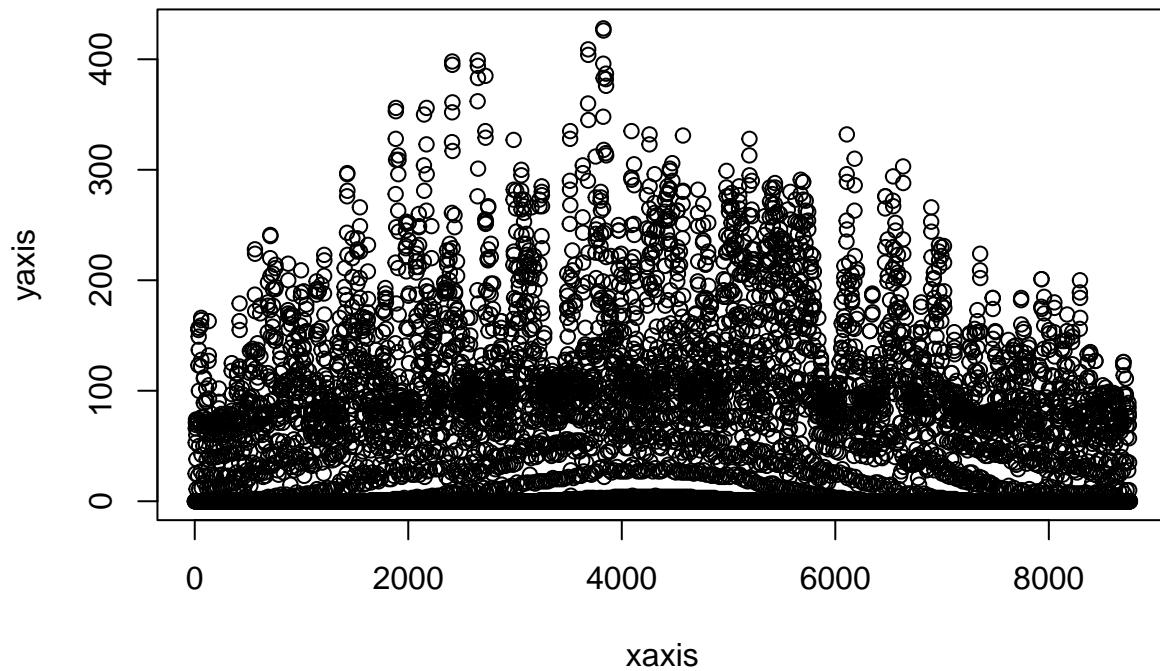
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 949
## [1] 0 991
## integer(0)
## [1] 340
## [1] 0 428
```





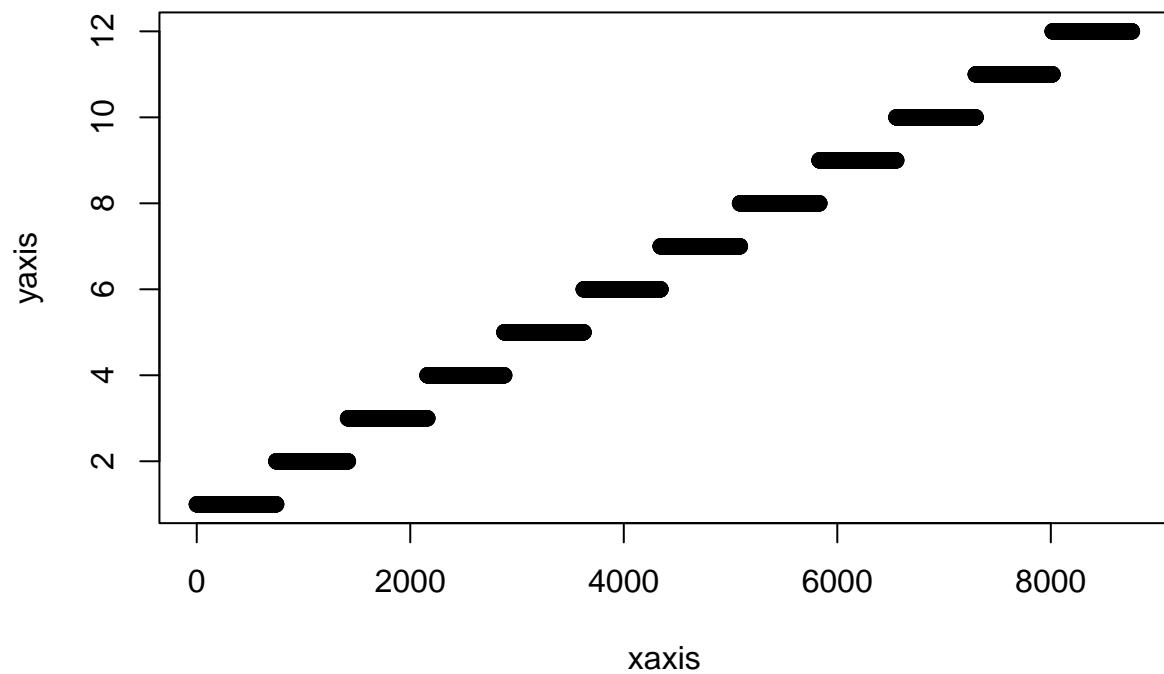


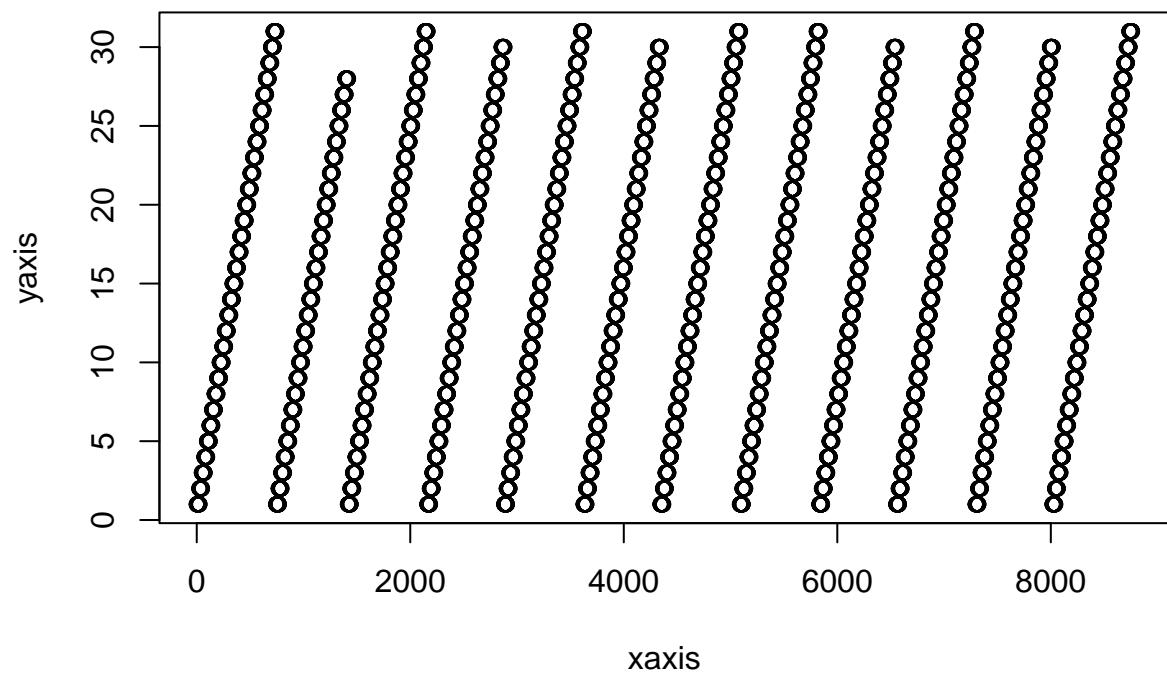


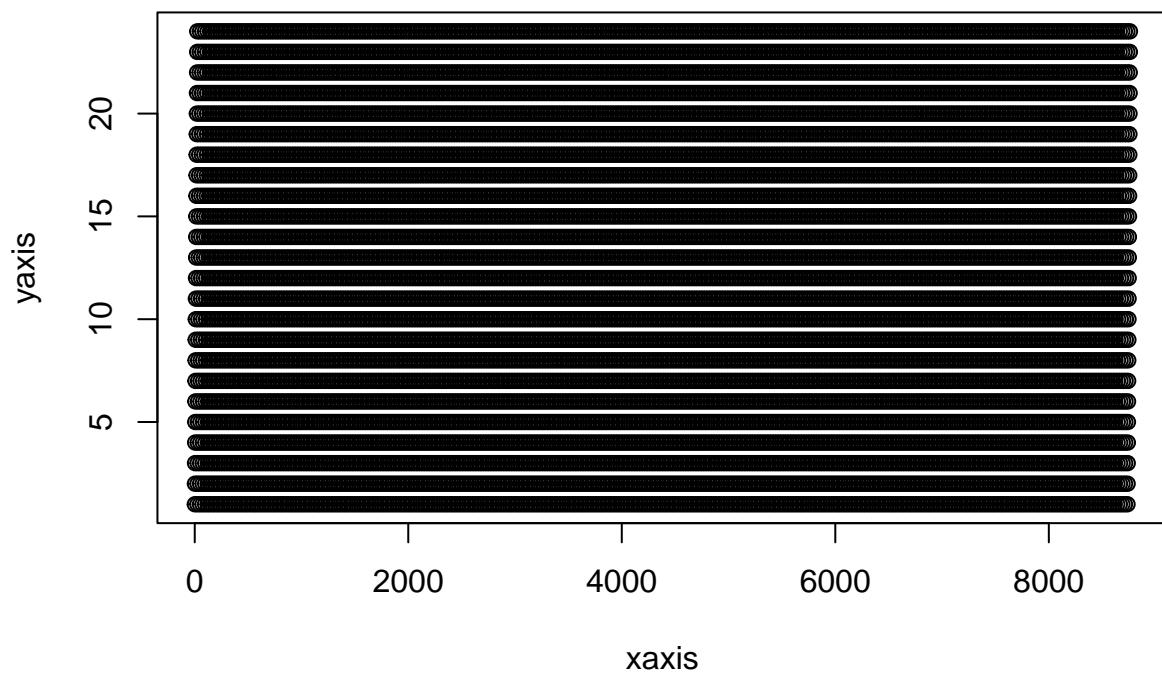


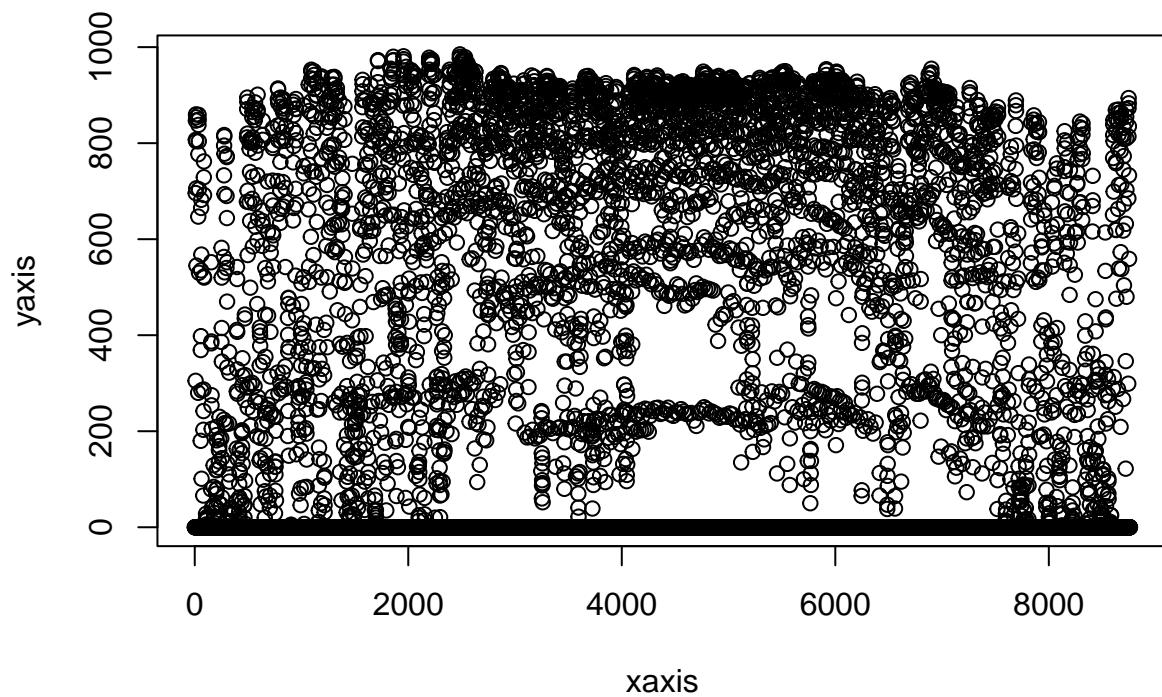
```
wea_UCD = wea_parser('dt/USA_CA_UC-Davis-University.AP.720576_TMYx.2007-2021.wea')
```

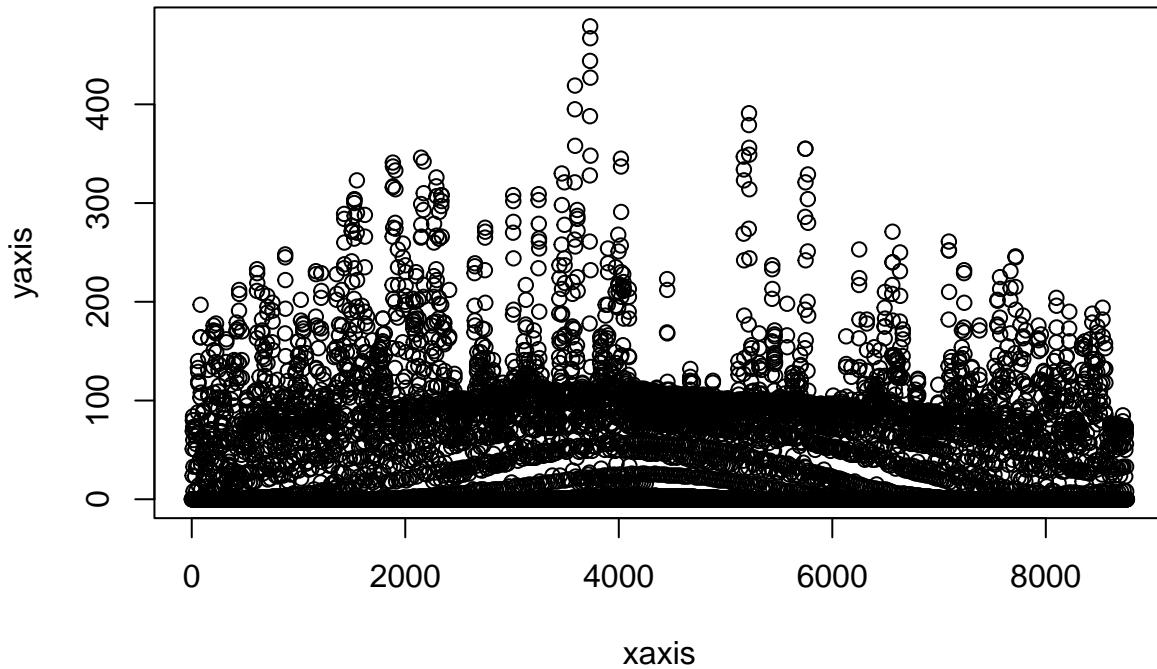
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 928
## [1] 0 985
## integer(0)
## [1] 322
## [1] 0 479
```











## PVSYST Parser

```

pvsyst_parser <- function(pvsyst_path){
  pvsyst_ll = readLines(pvsyst_path)
  start = grep("W/m2,W/m2,W/m2,deg.C,m/sec", pvsyst_ll) + 1
  pvsyst_dt = pvsyst_ll[start:length(pvsyst_ll)]

  #separate data by commas, like a CSV
  pv_df = read.table(textConnection(pvsyst_dt), sep = ',')

  #add the column names in
  col_names = pvsyst_ll[grep('Year,Month,Day,Hour,Minute,GHI,DHI,DNI,Tamb,WindVel,WindDir', pvsyst_ll)]
  #[3] How to use unlist to turn a list into a character value so we can use it for colnames
  colnames(pv_df) = unlist(strsplit(col_names, ','))

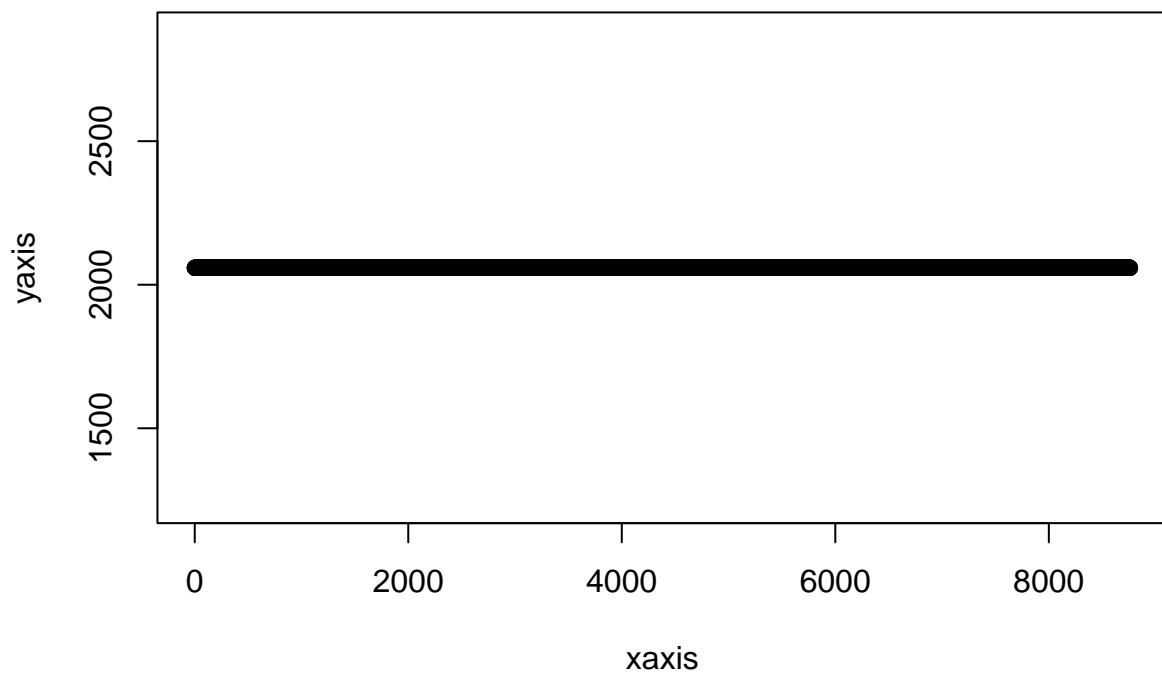
  if (DEBUG == TRUE){
    debug_stats(pv_df, 11)
    vis_inspc(pv_df, c(1:ncol(pv_df)))
  }
  return(pv_df)
}

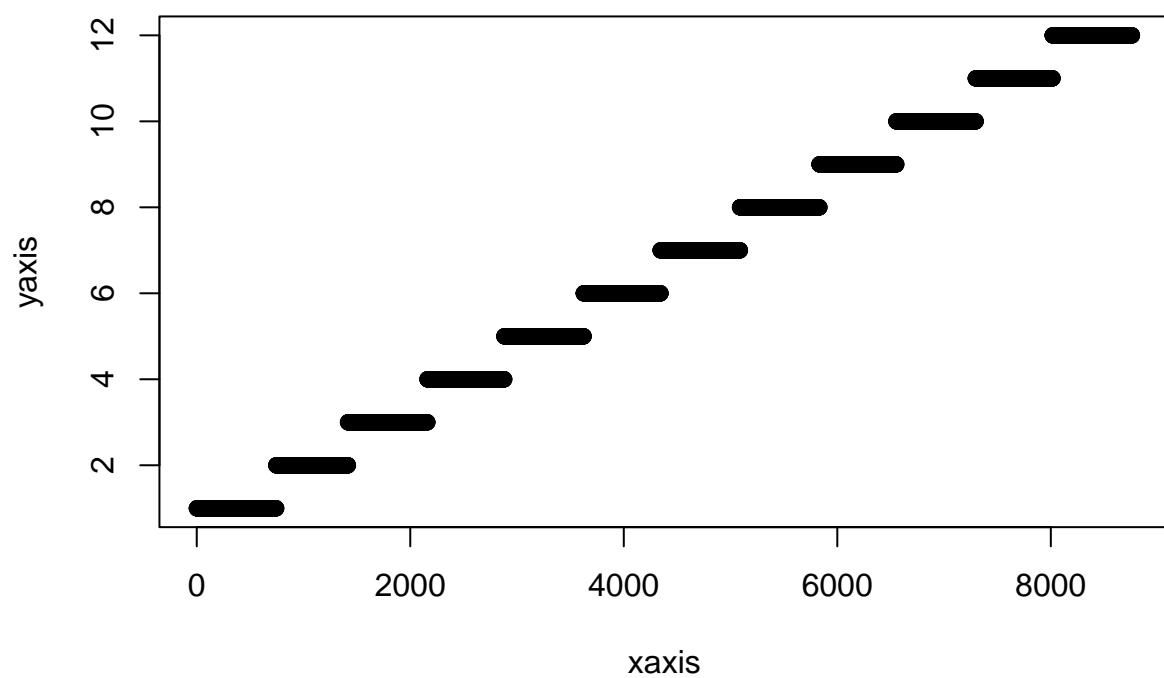
pv_SF = pvsyst_parser('dt/USA_CA_Fairfield-San.Francisco.Bay.Reserve.998011_TMYx.2007-2021.pvsyst')

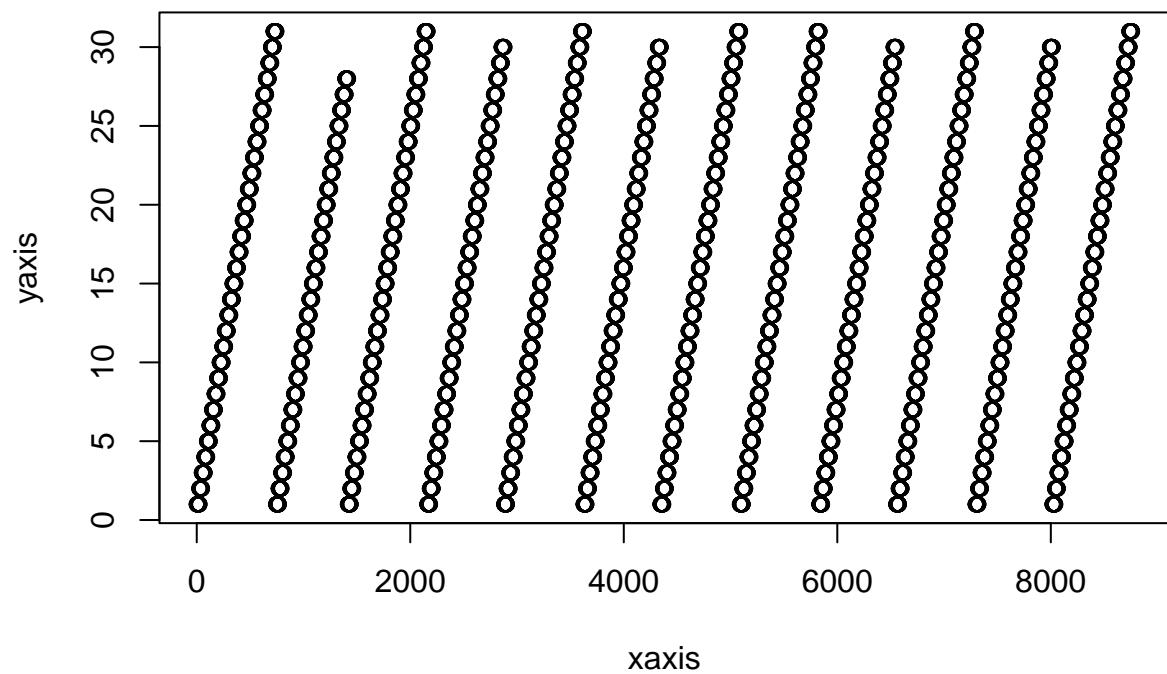
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)

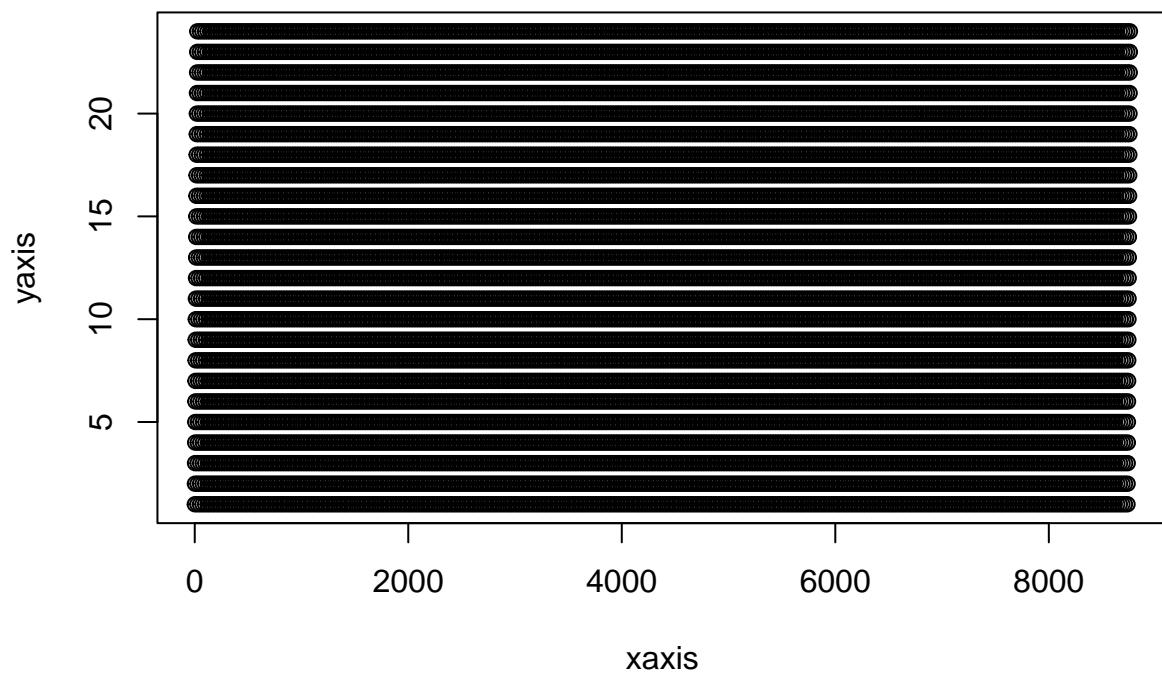
```

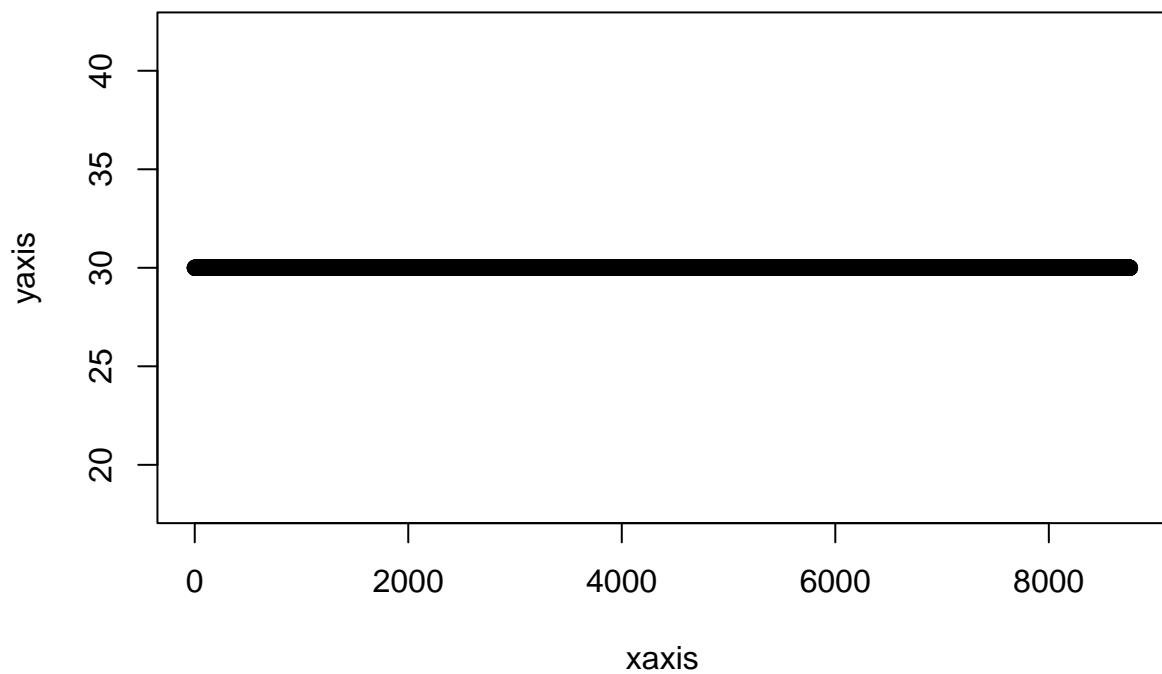
```
## [1] 1
## [1] 2059 2059
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 1
## [1] 30 30
## integer(0)
## [1] 980
## [1] 0 1016
## integer(0)
## [1] 326
## [1] 0 433
## integer(0)
## [1] 934
## [1] 0 998
## integer(0)
## [1] 40
## [1] -1 38
## integer(0)
## [1] 13
## [1] 0 12
## integer(0)
## [1] 263
## [1] 0 360
```

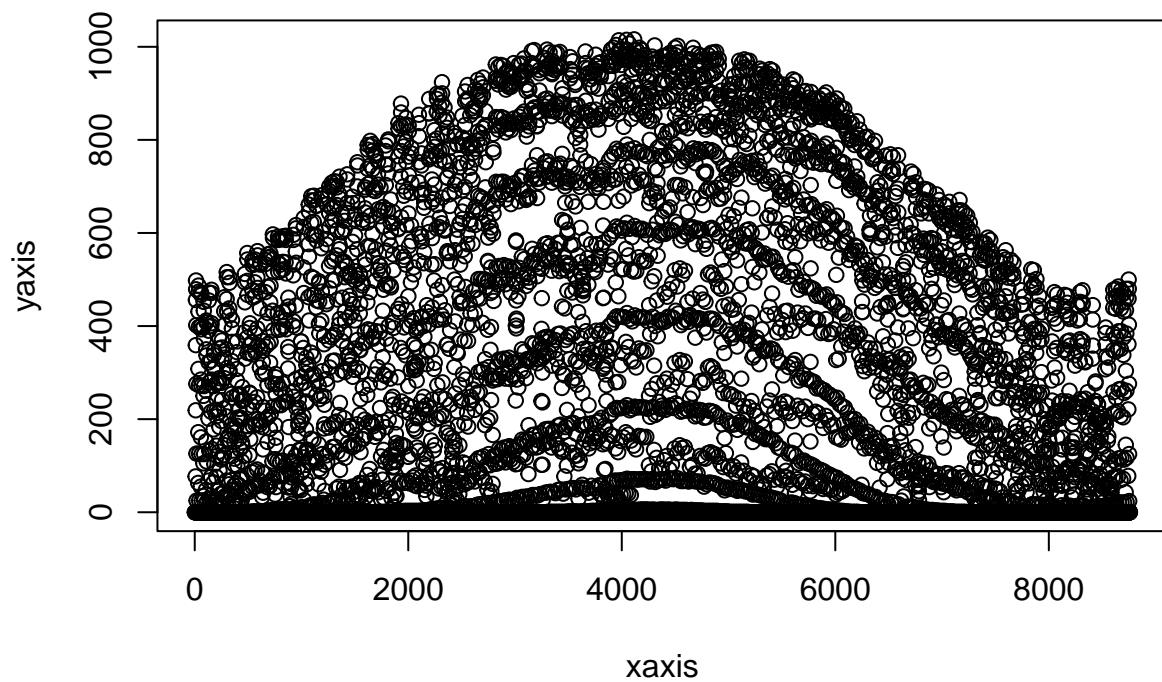


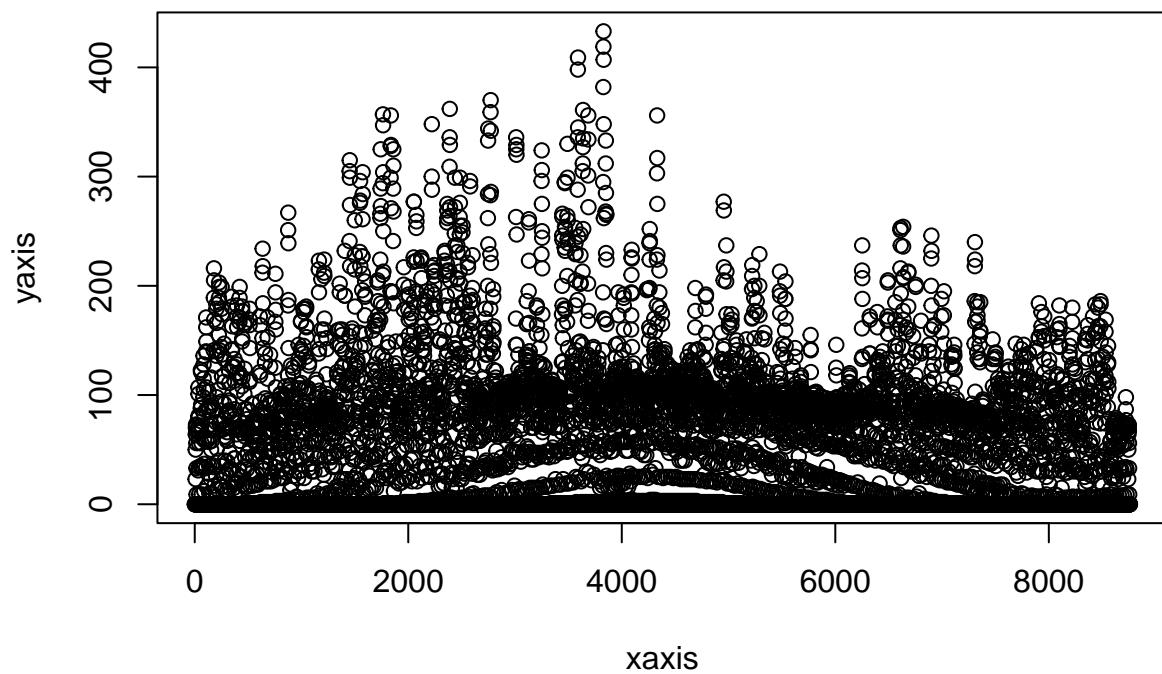


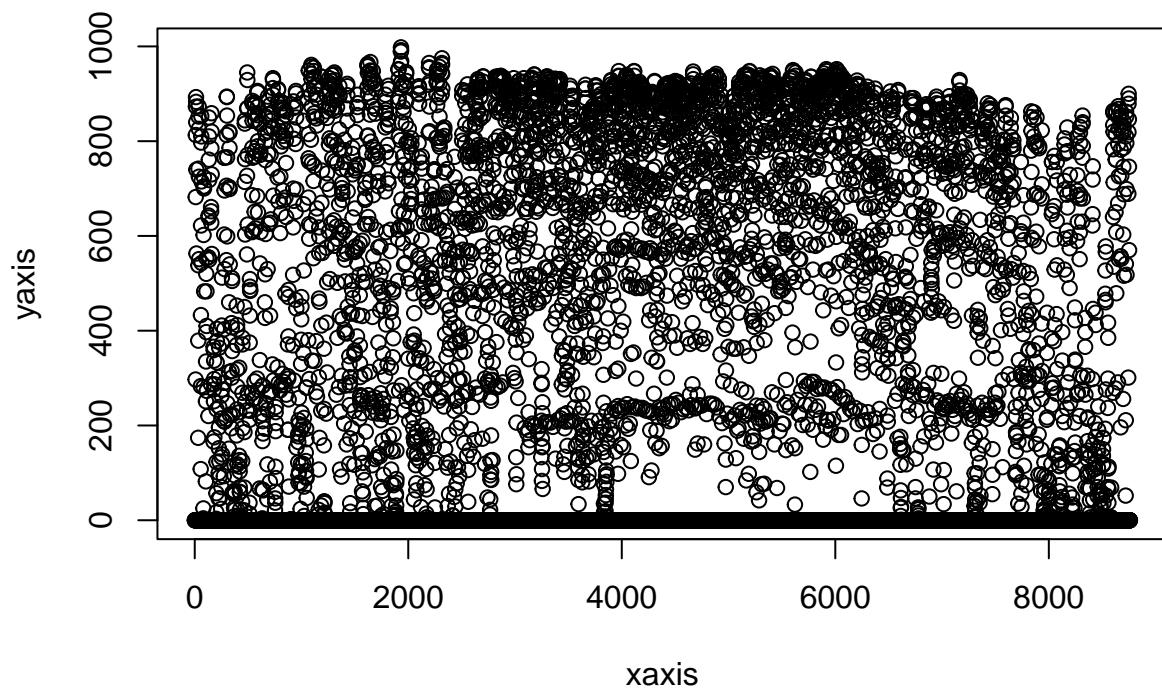


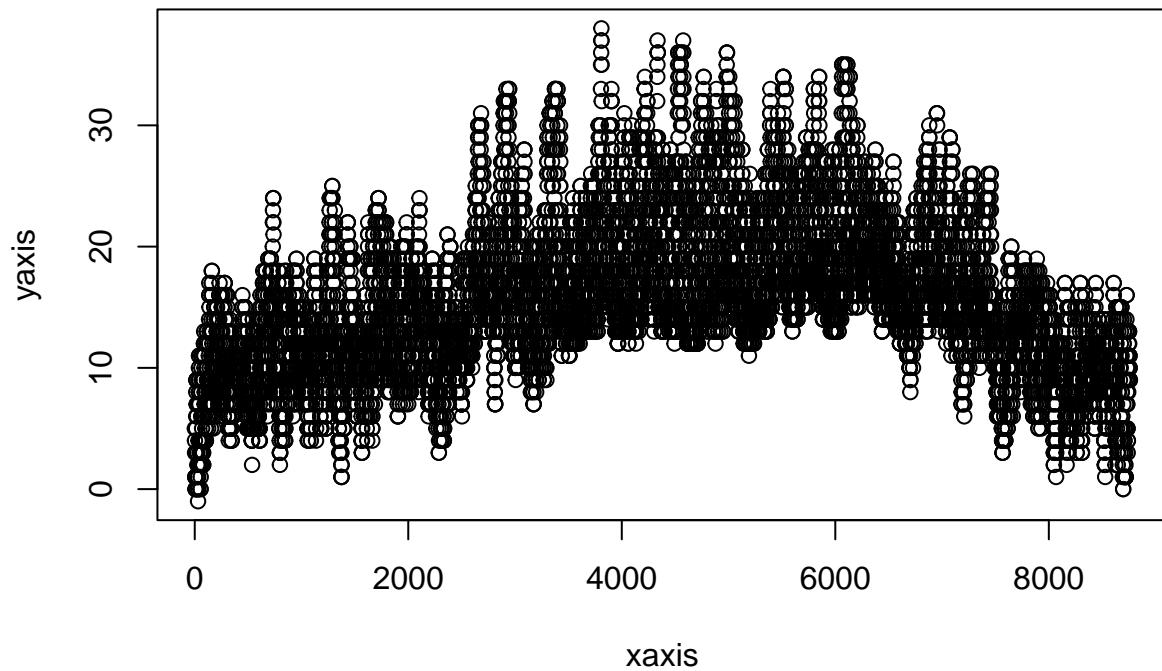


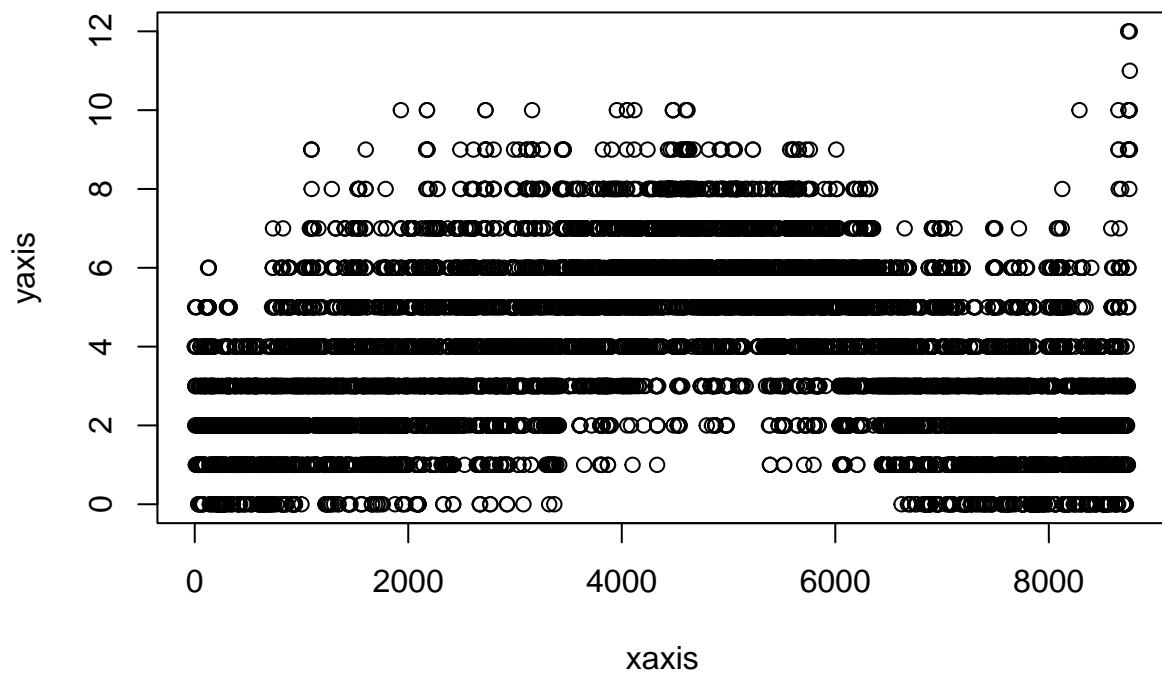


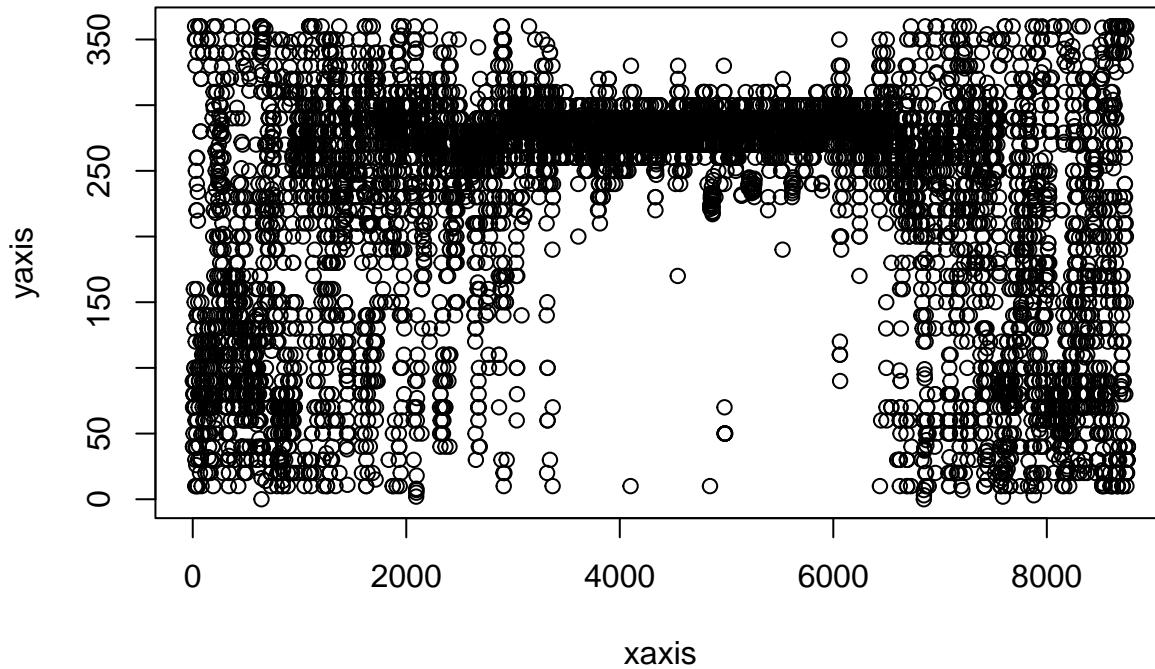












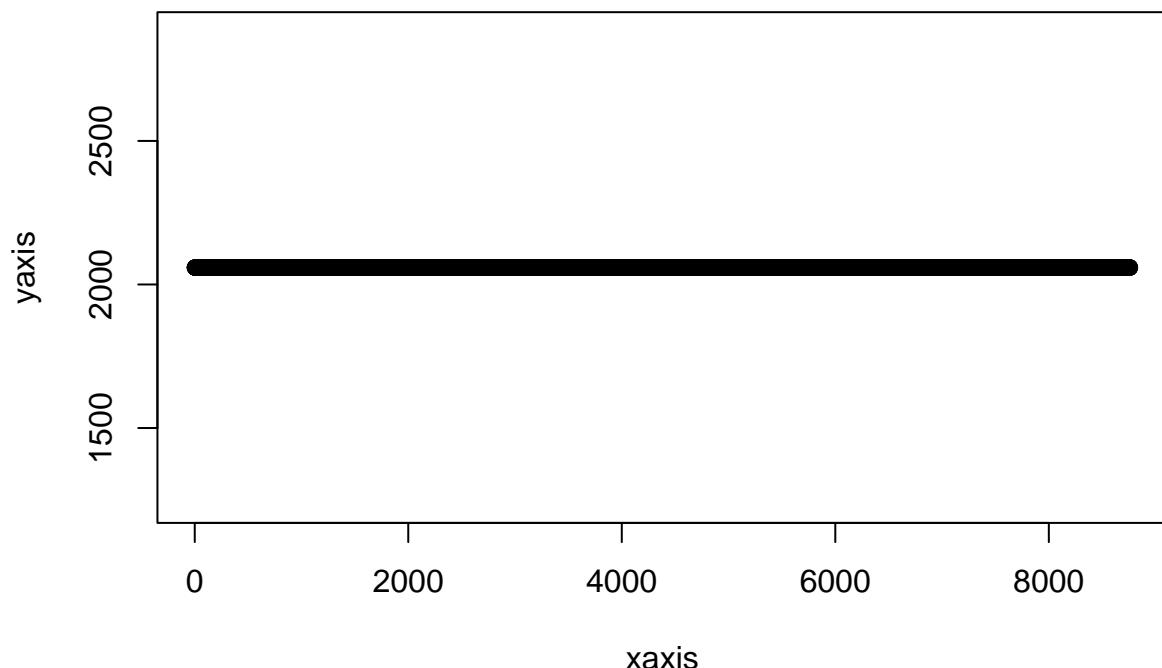
```

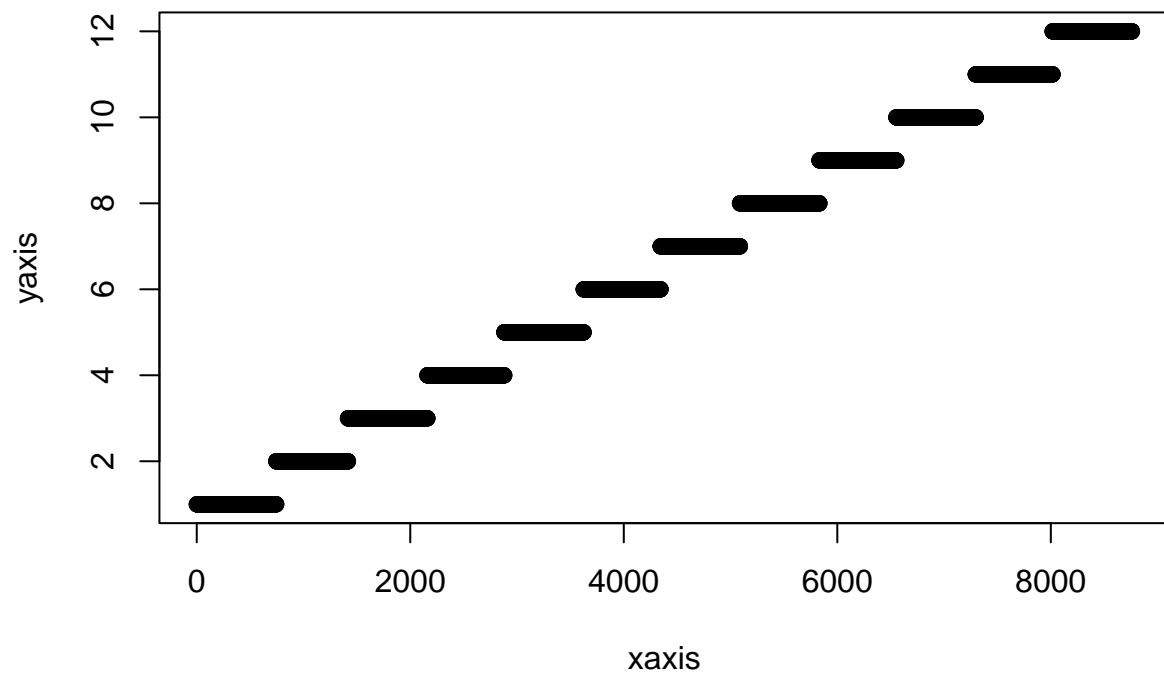
pv_marin = pvsyst_parser('dt/USA_CA_Marin.County.AP-Gnoss.Field.720406_TMYx.2007-2021.pvsyst')

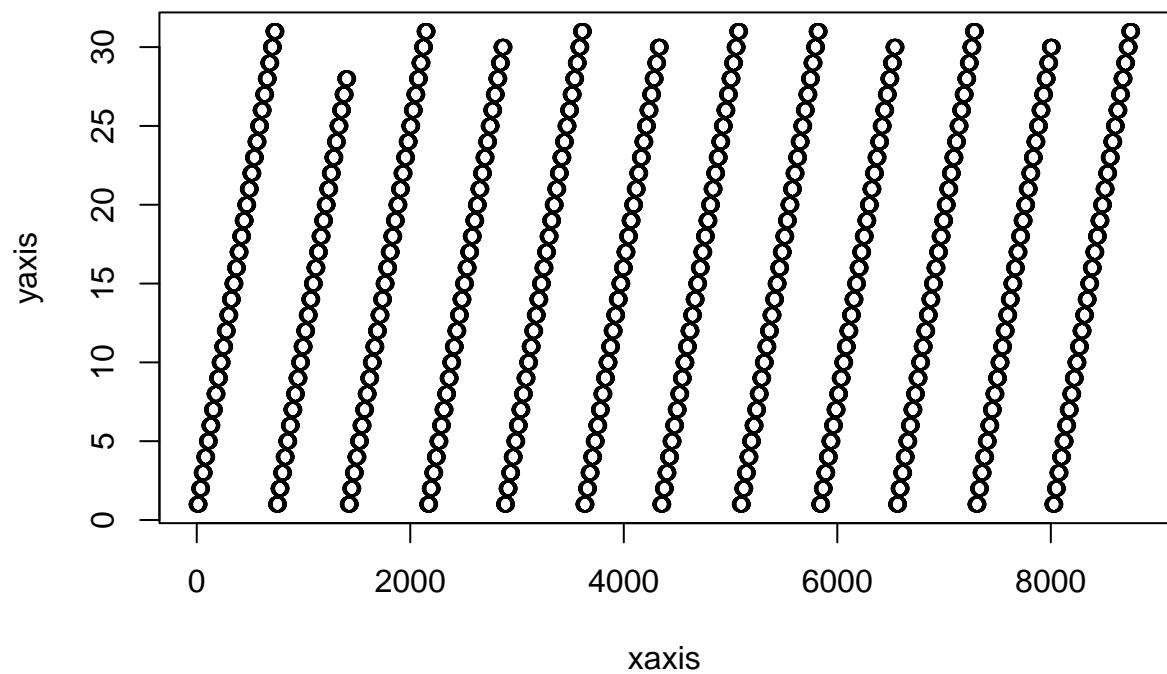
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] 2059 2059
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 1
## [1] 30 30
## integer(0)
## [1] 979
## [1] 0 1013
## integer(0)
## [1] 328
## [1] 0 452
## integer(0)

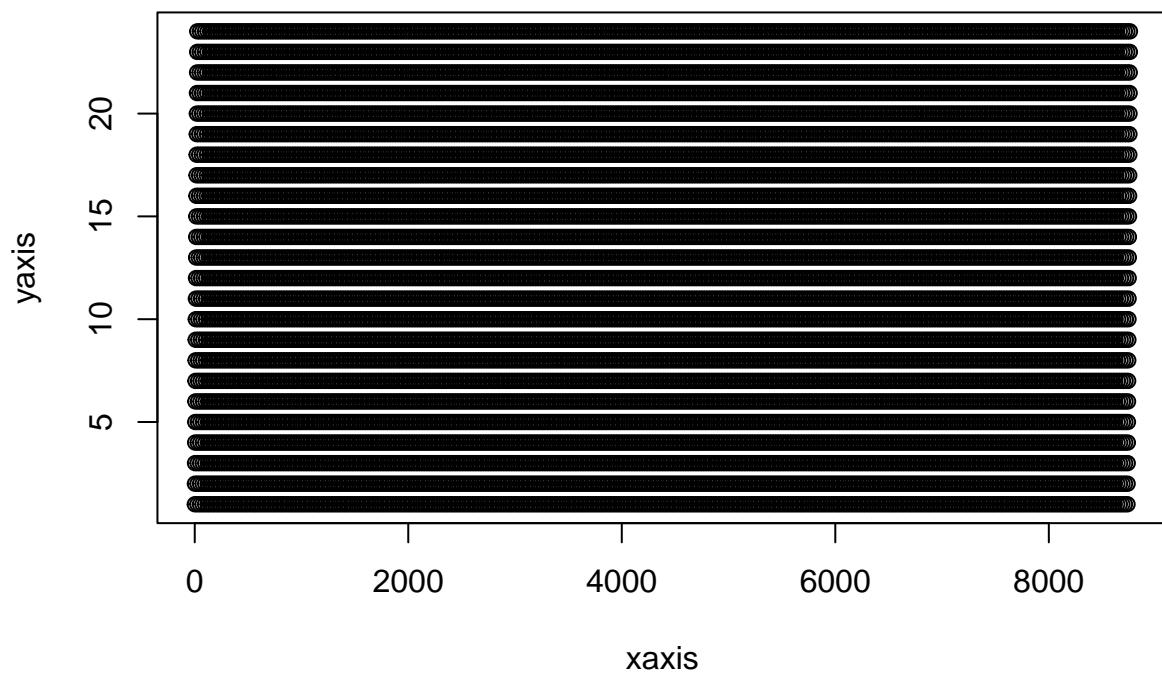
```

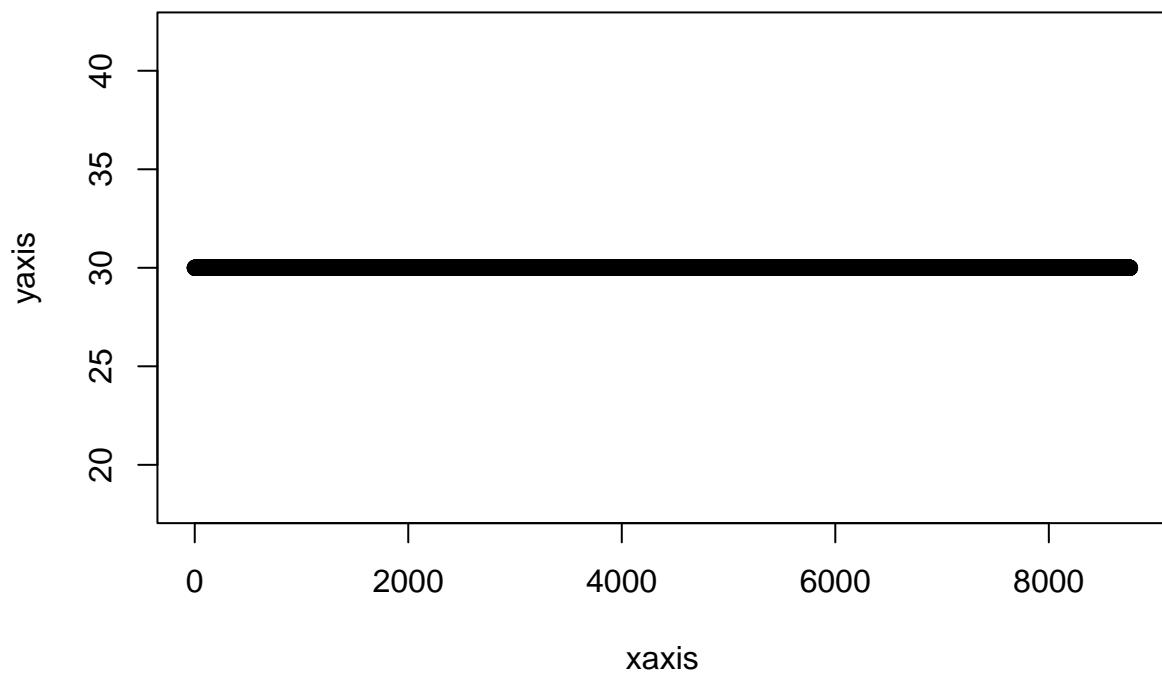
```
## [1] 930
## [1] 0 988
## integer(0)
## [1] 41
## [1] -3 37
## integer(0)
## [1] 13
## [1] 0 12
## integer(0)
## [1] 357
## [1] 0 360
```

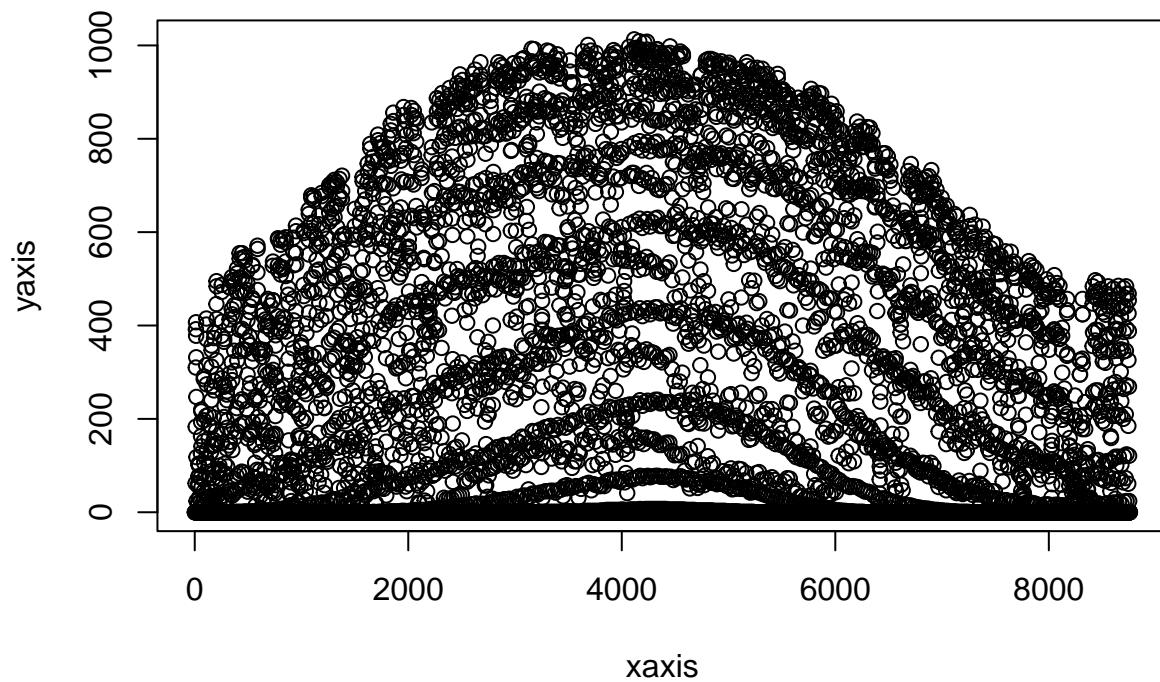


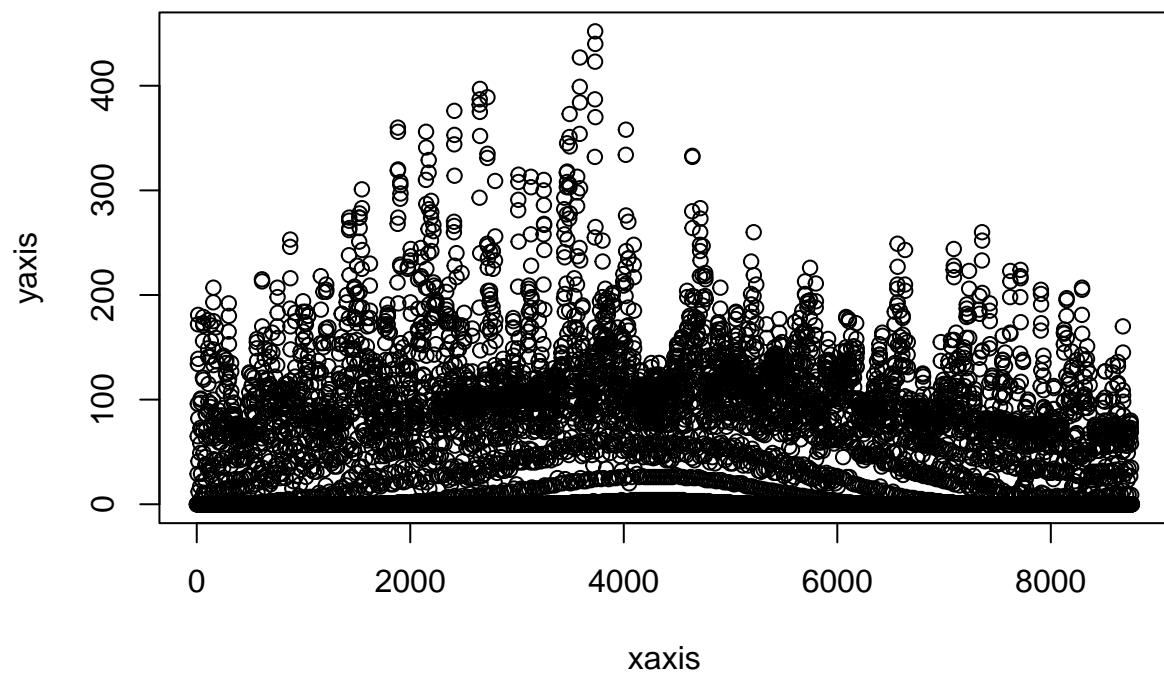


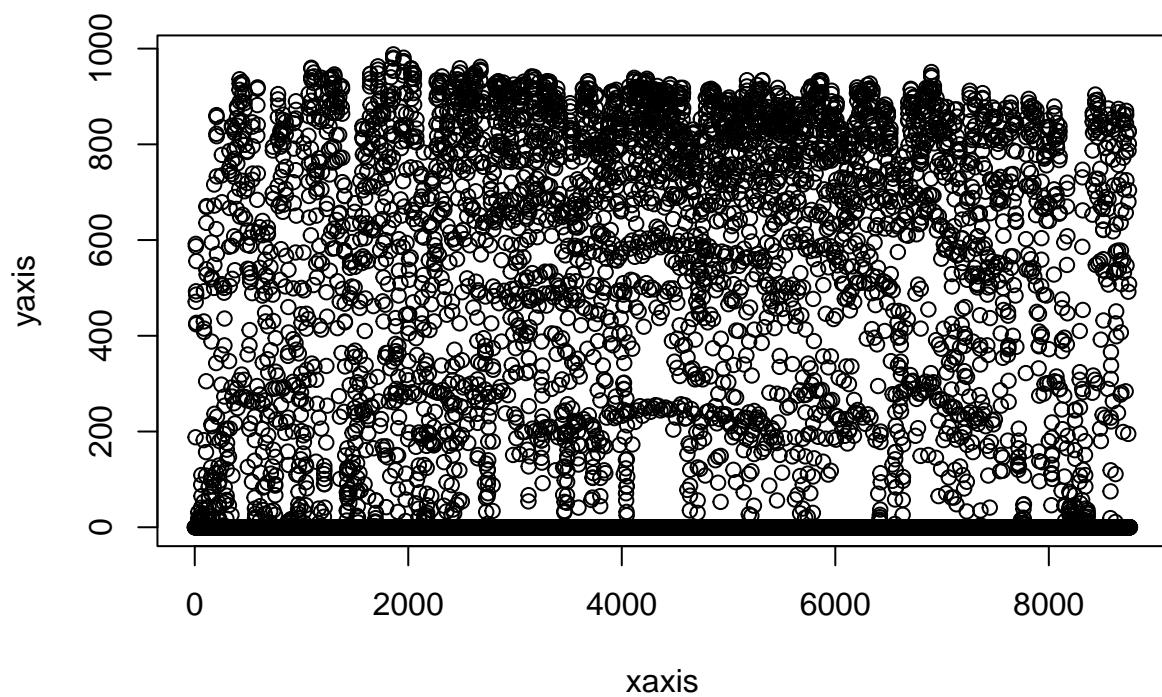


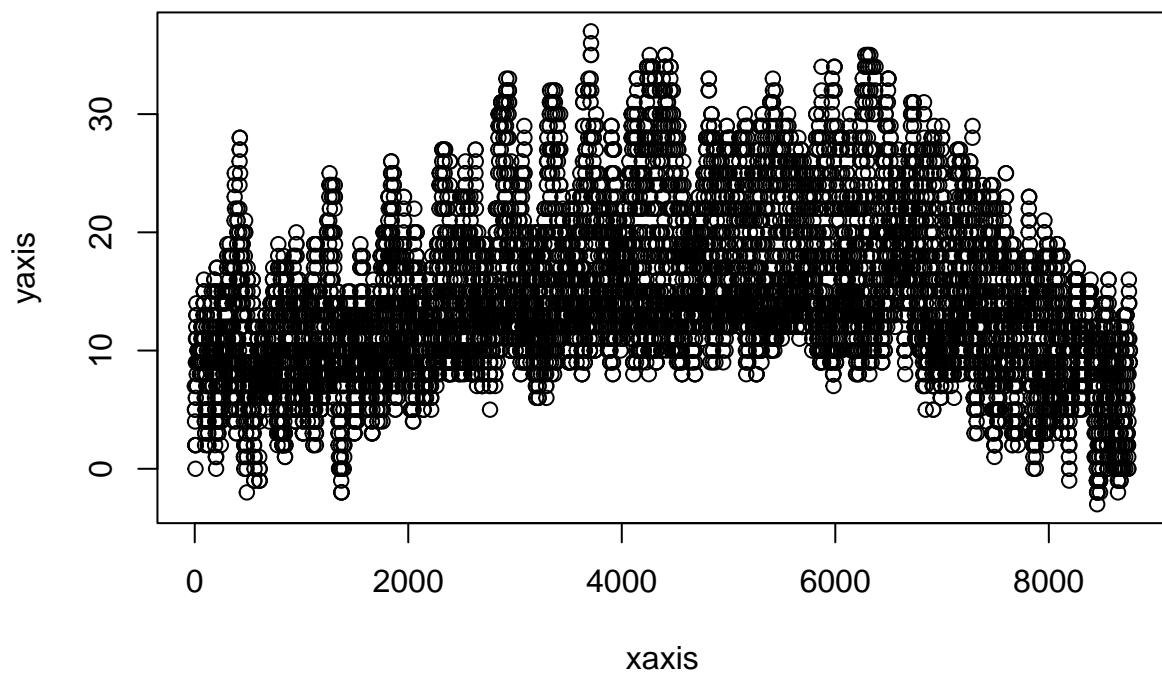


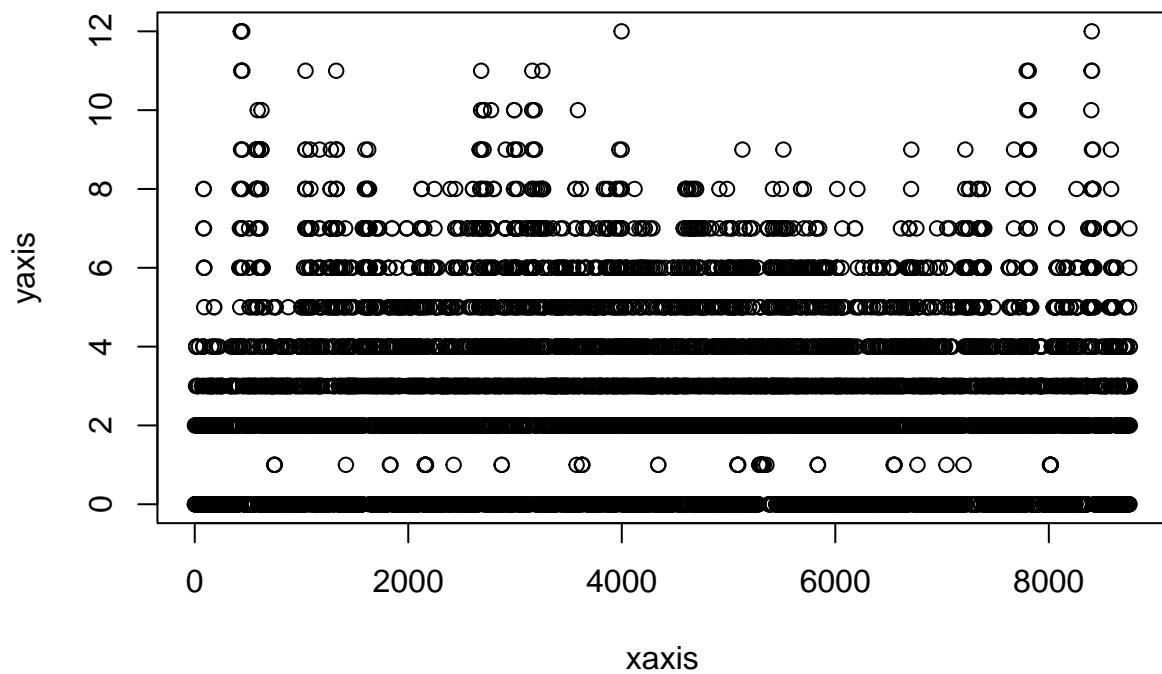


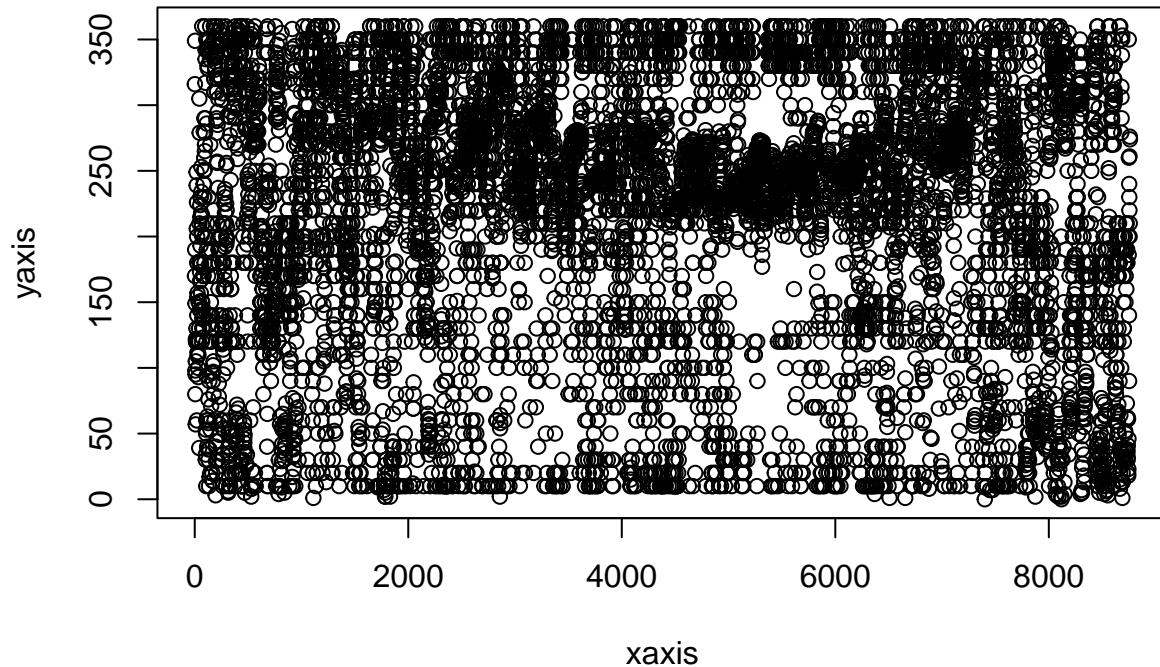








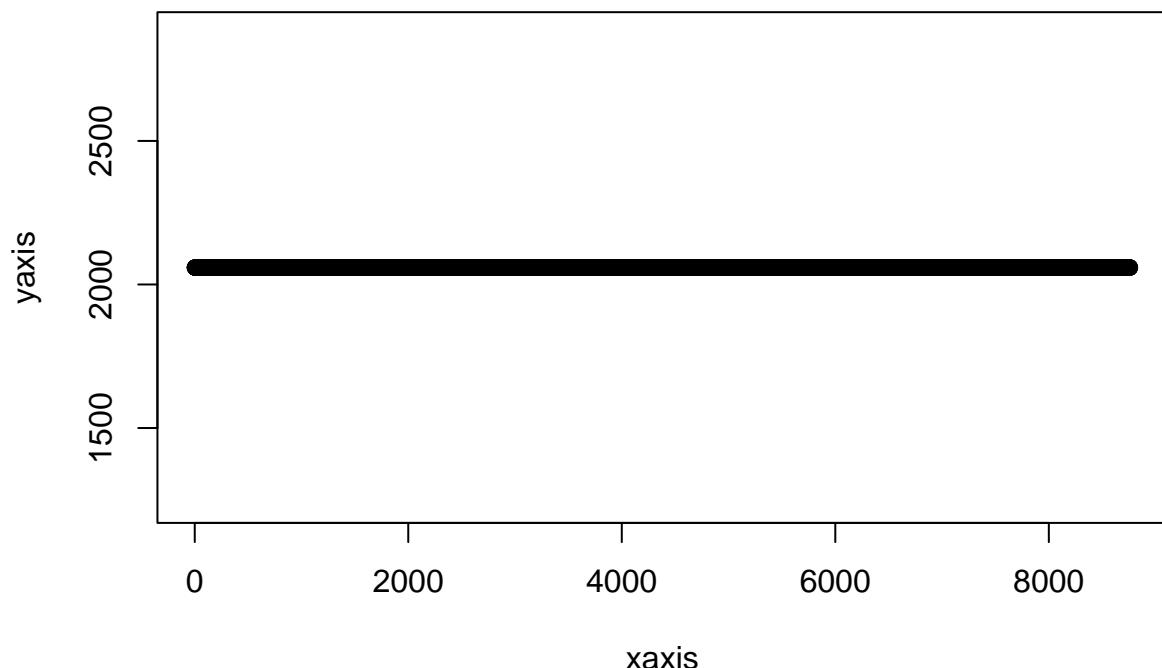


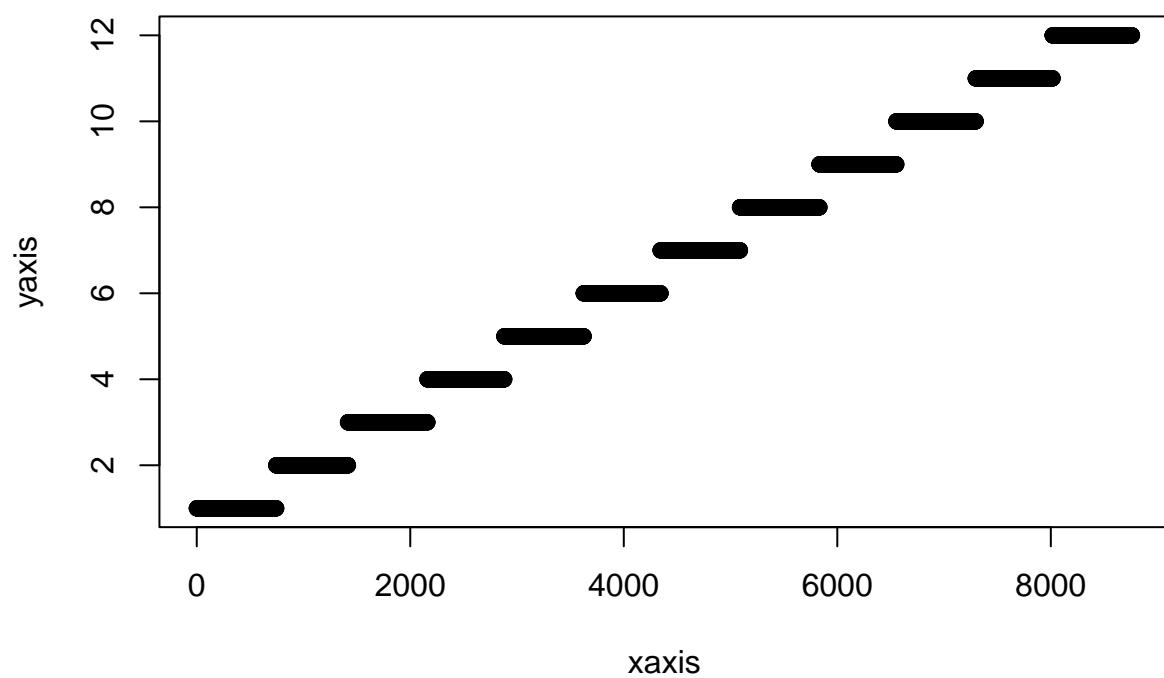


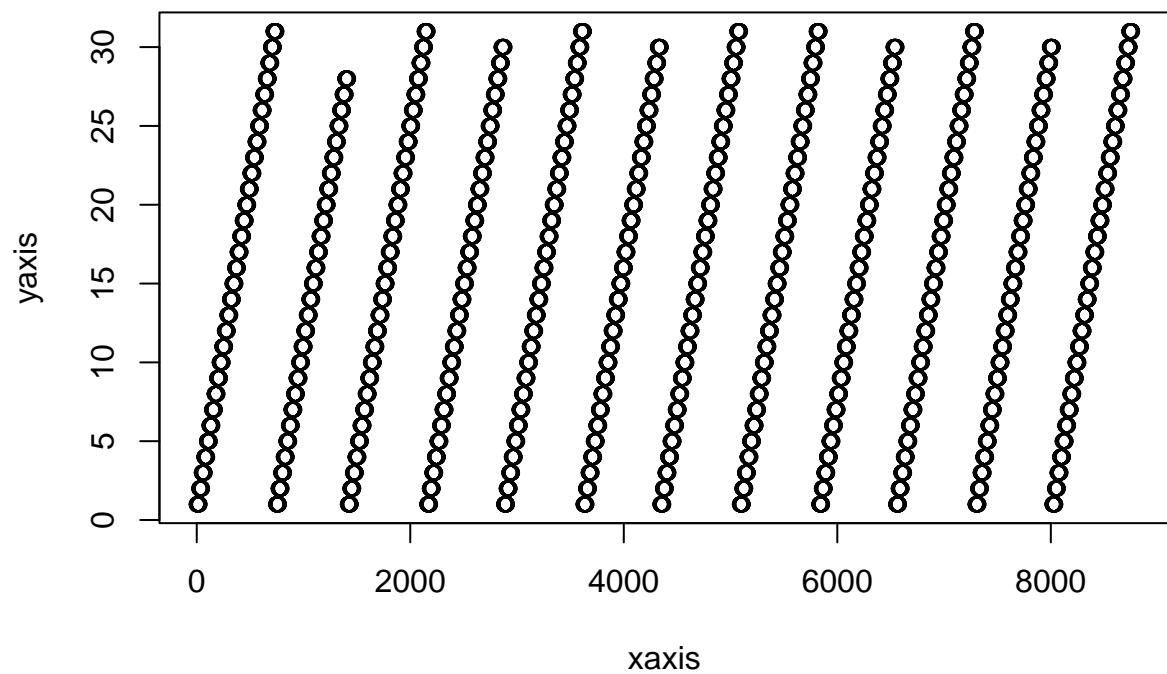
```
pv_napa = pvsyst_parser('dt/USA_CA_Napa.County.AP.724955_TMYx.2007-2021.pvsyst')
```

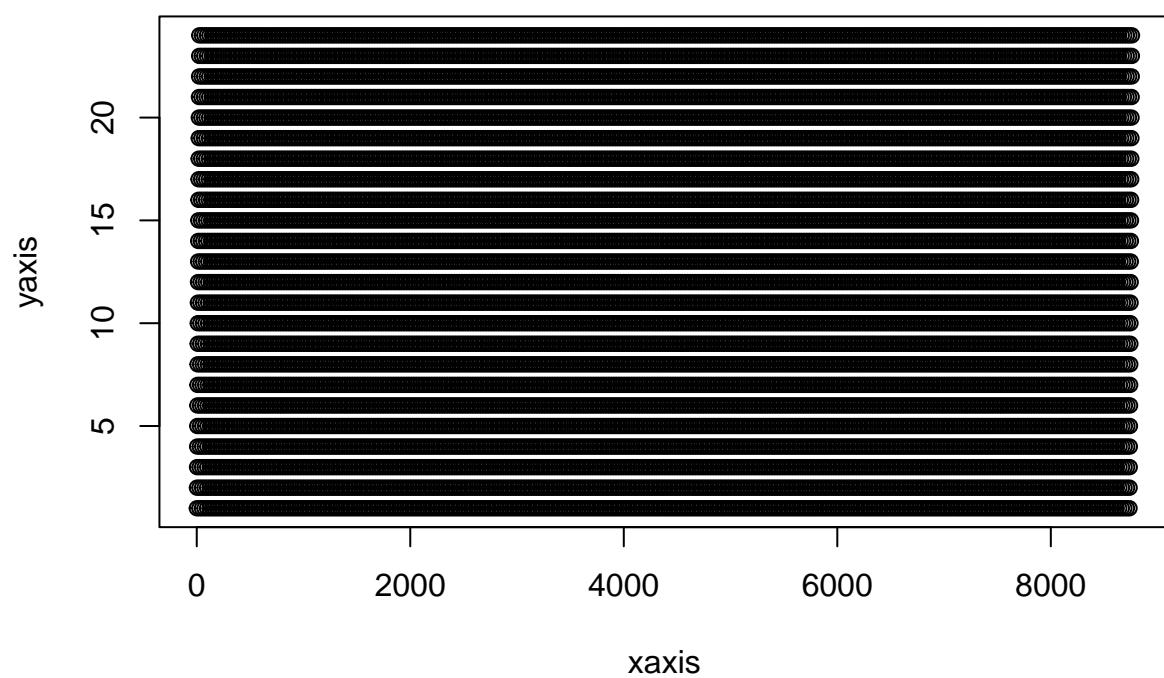
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] 2059 2059
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 1
## [1] 30 30
## integer(0)
## [1] 991
## [1] 0 1024
## integer(0)
## [1] 312
## [1] 0 414
## integer(0)
```

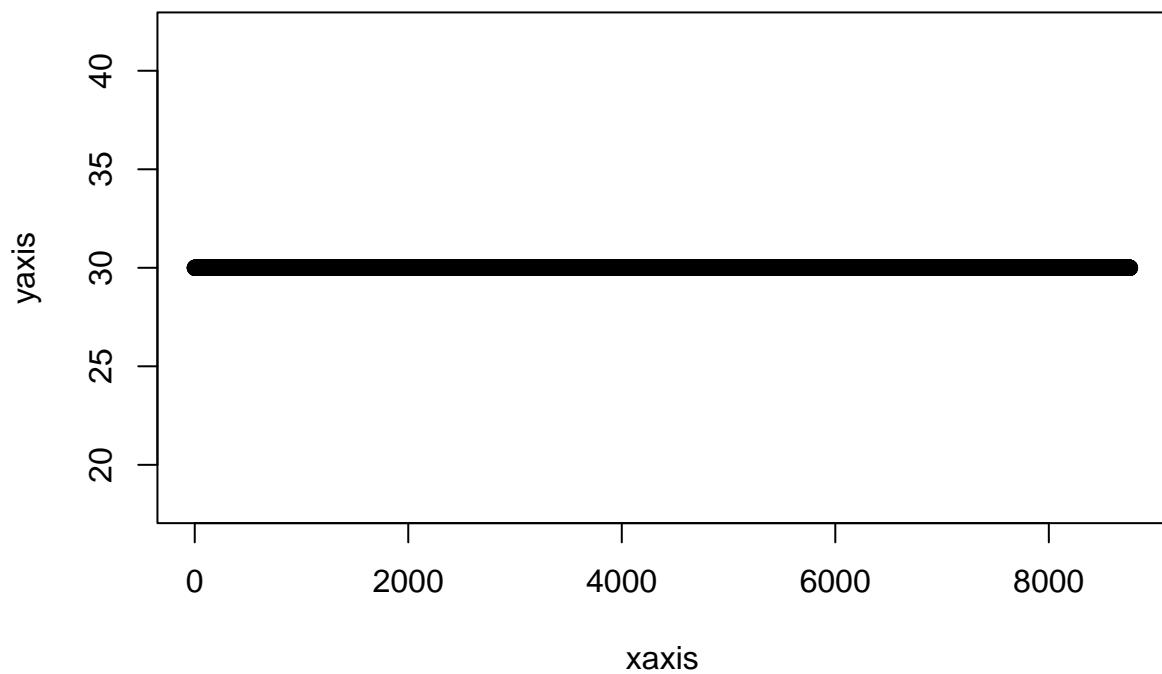
```
## [1] 927
## [1] 0 994
## integer(0)
## [1] 40
## [1] -3 36
## integer(0)
## [1] 17
## [1] 0 17
## integer(0)
## [1] 330
## [1] 0 360
```

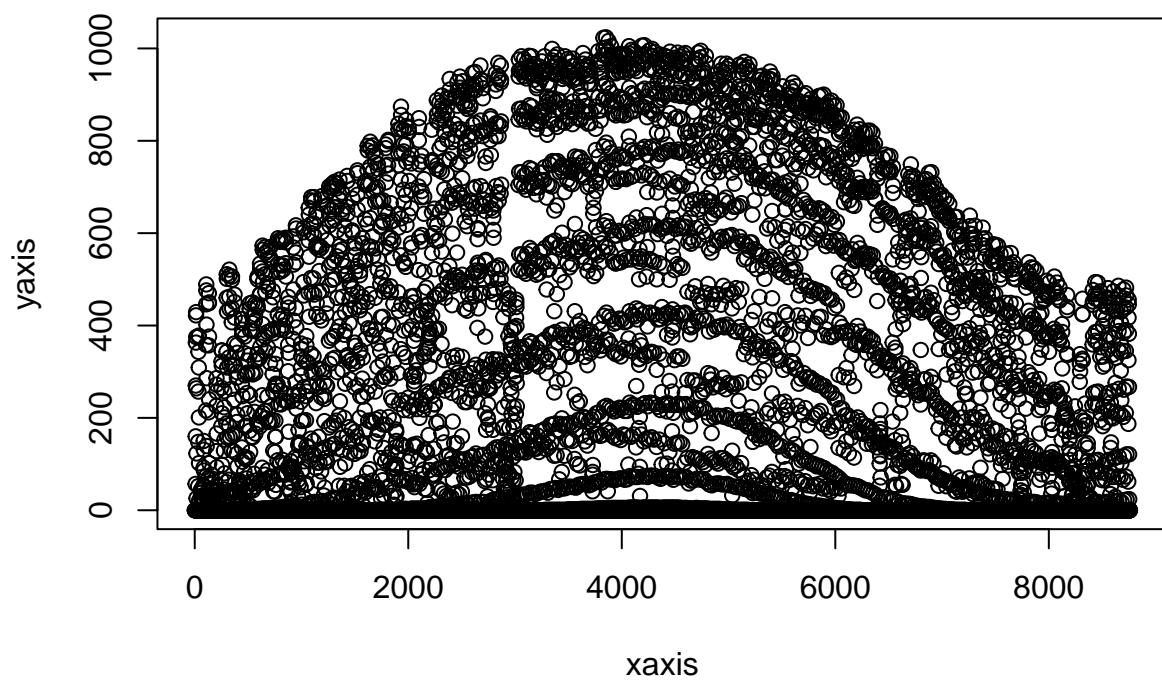


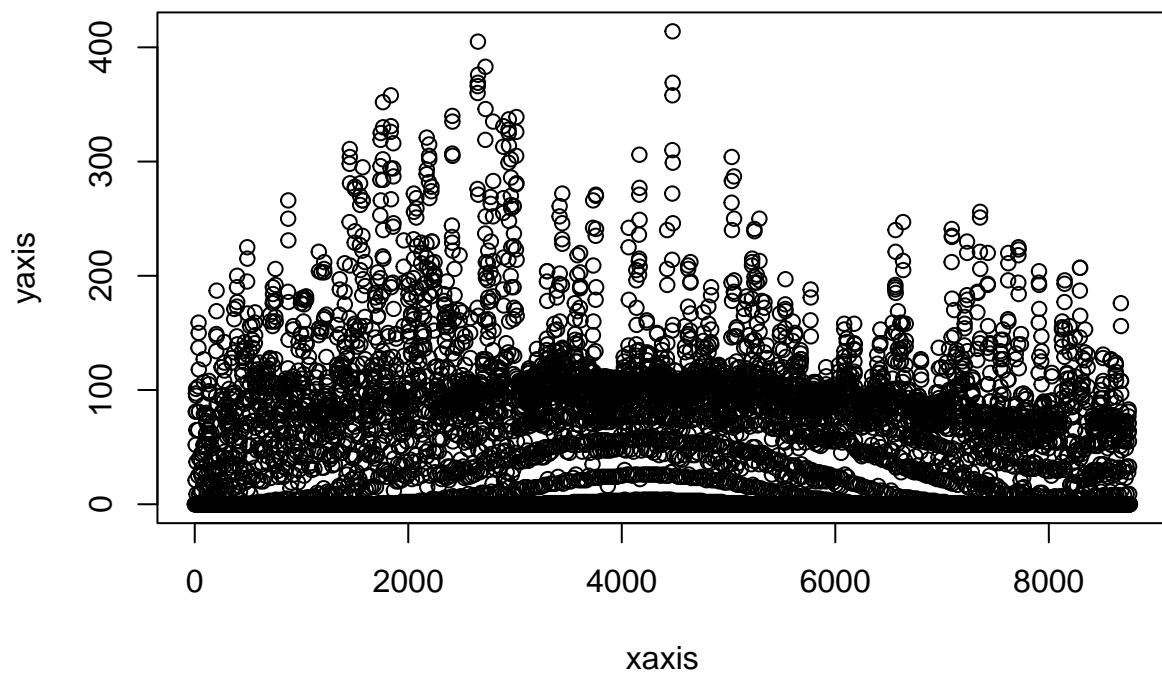


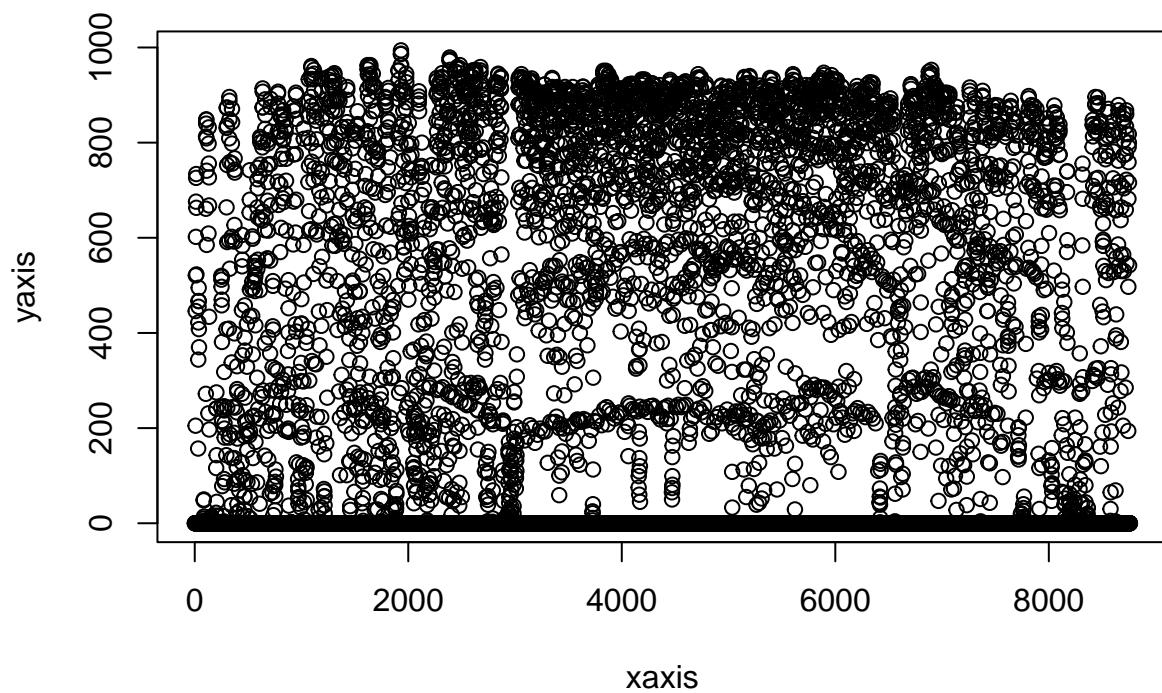


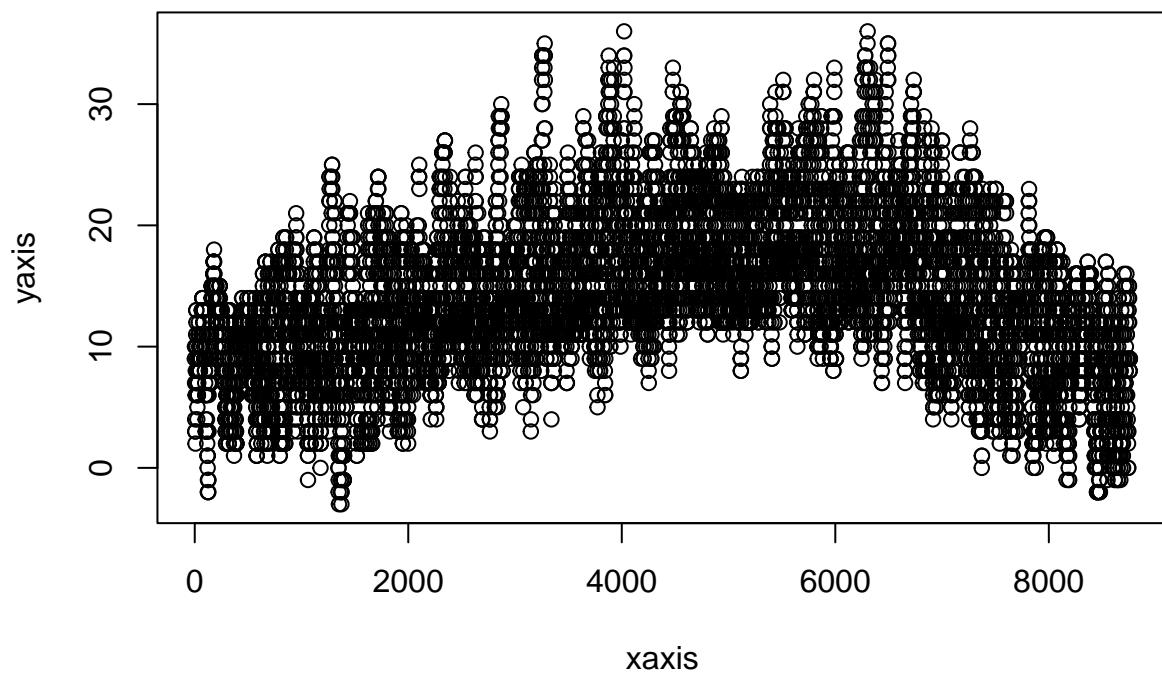


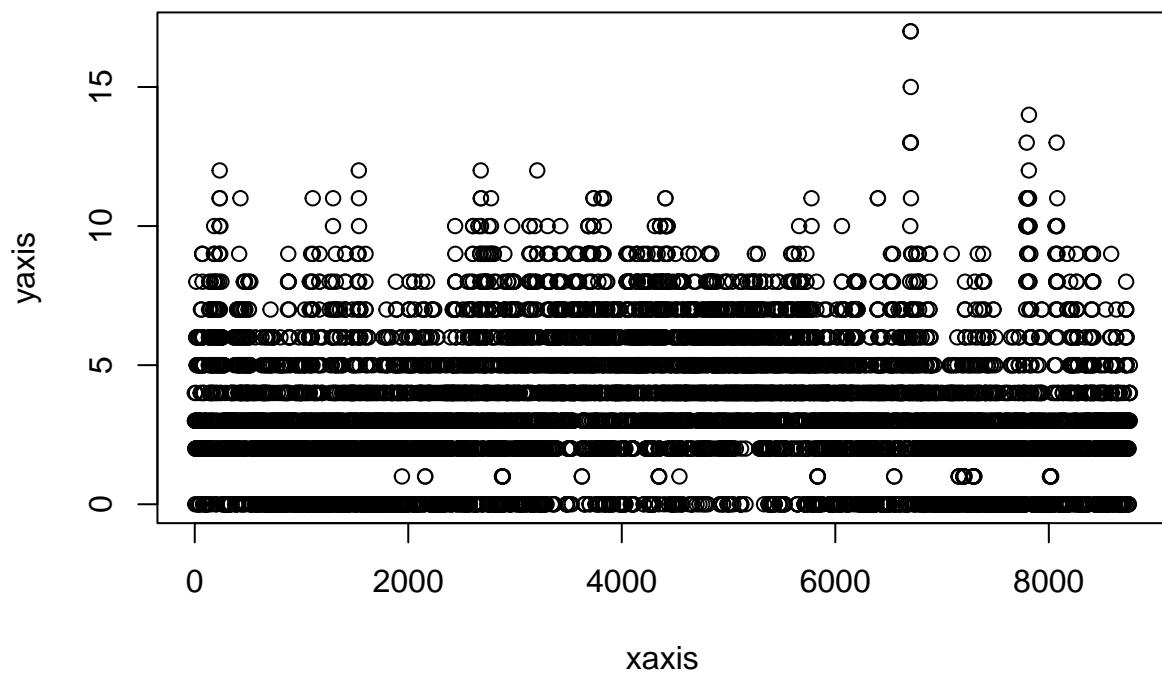


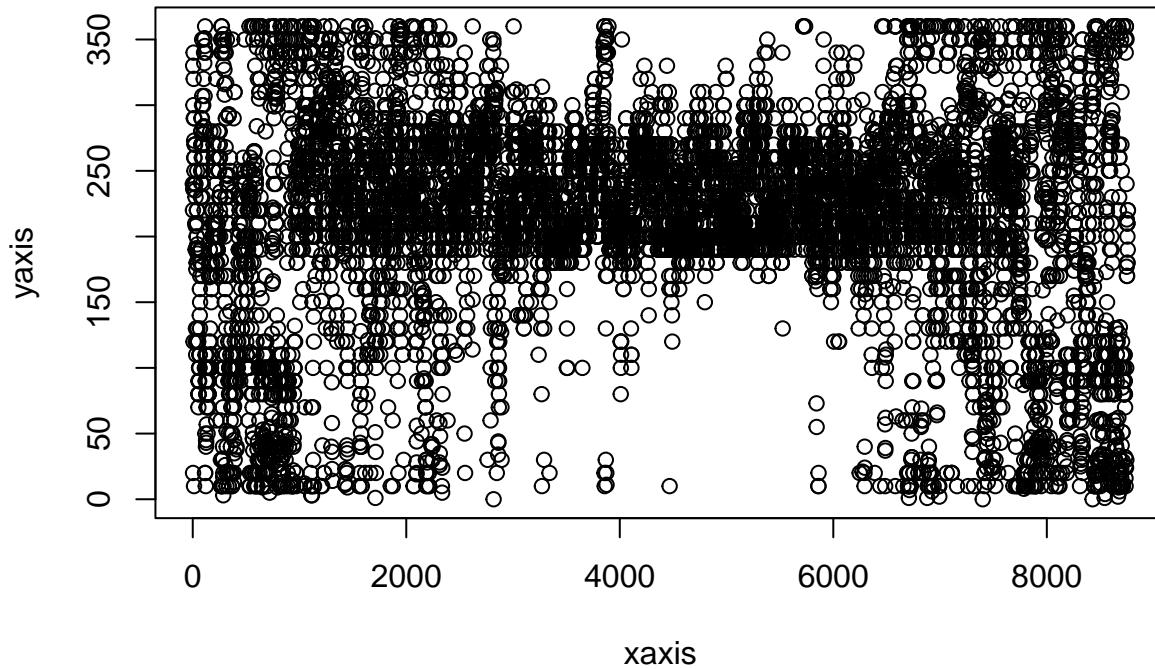








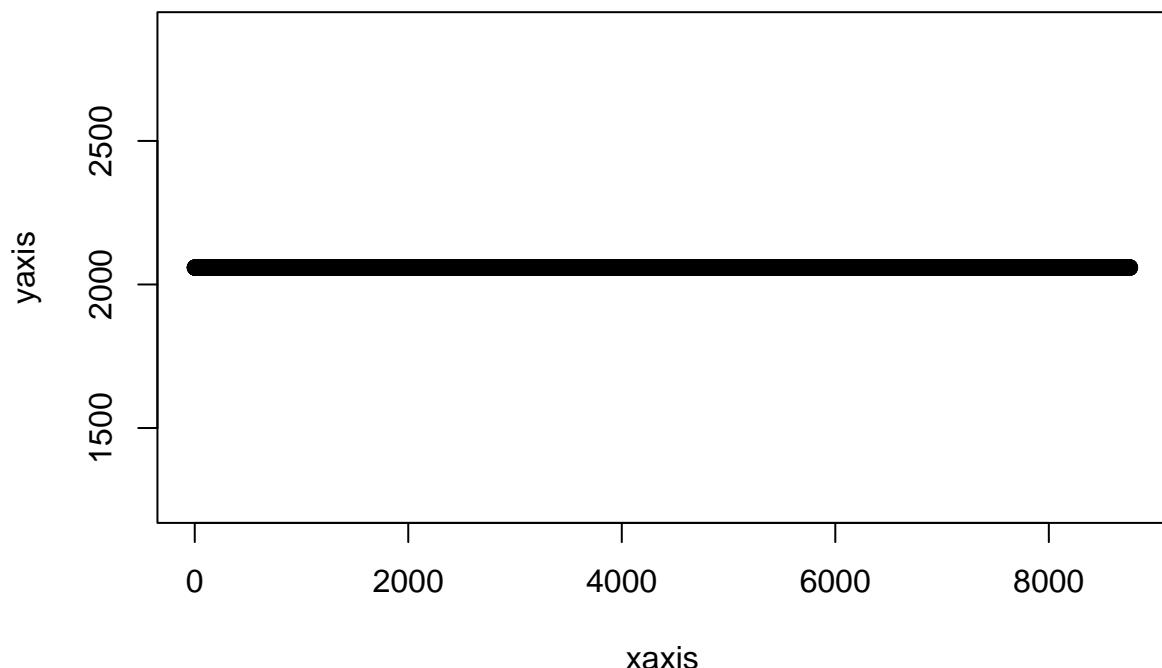


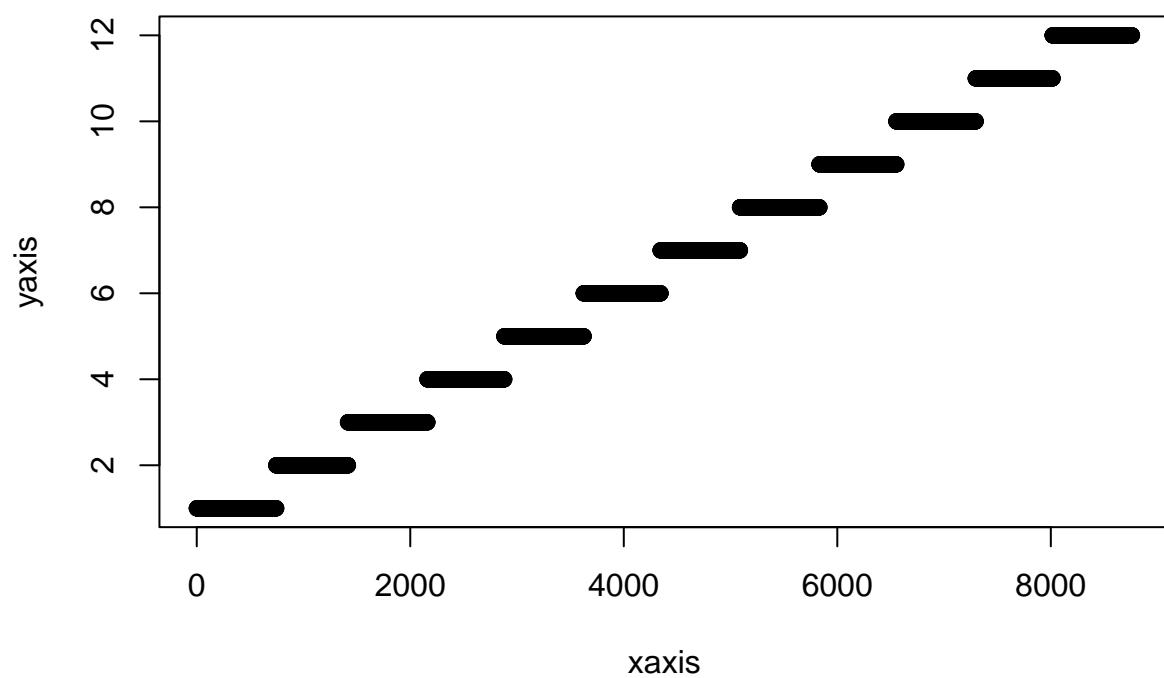


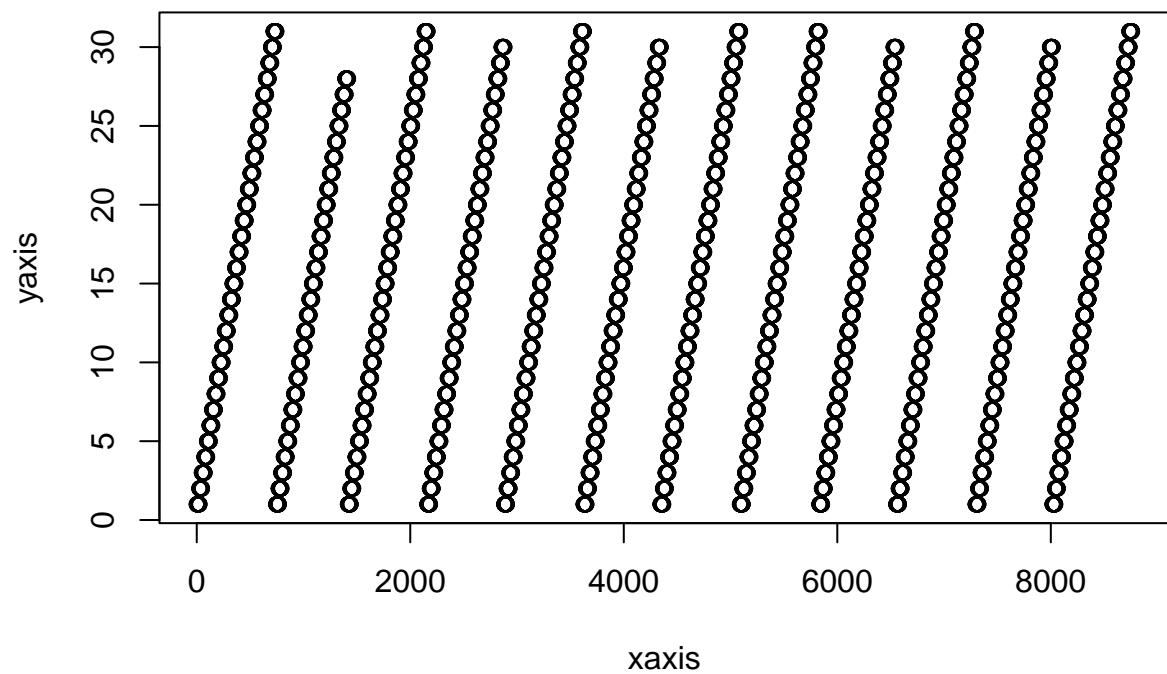
```
pv_reyes = pvsyst_parser('dt/USA_CA_Point.Reyes.Lighthouse.724959_TMYx.2007-2021.pvsyst')
```

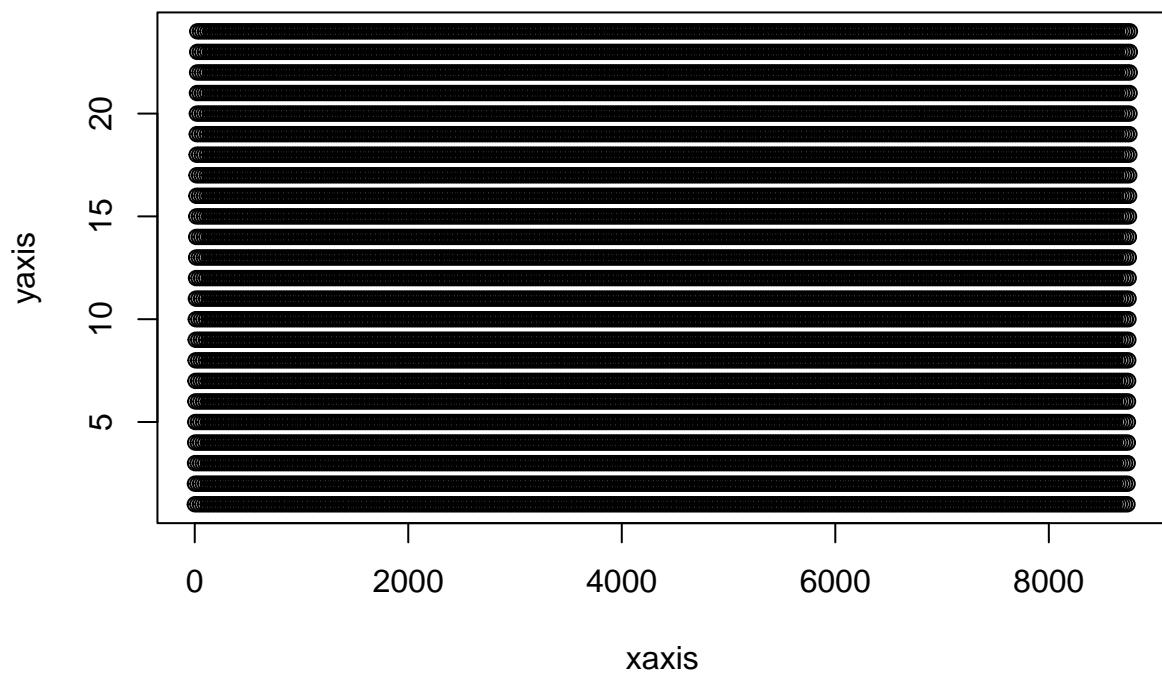
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] 2059 2059
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 1
## [1] 30 30
## integer(0)
## [1] 977
## [1] 0 1016
## integer(0)
## [1] 340
## [1] 0 428
## integer(0)
```

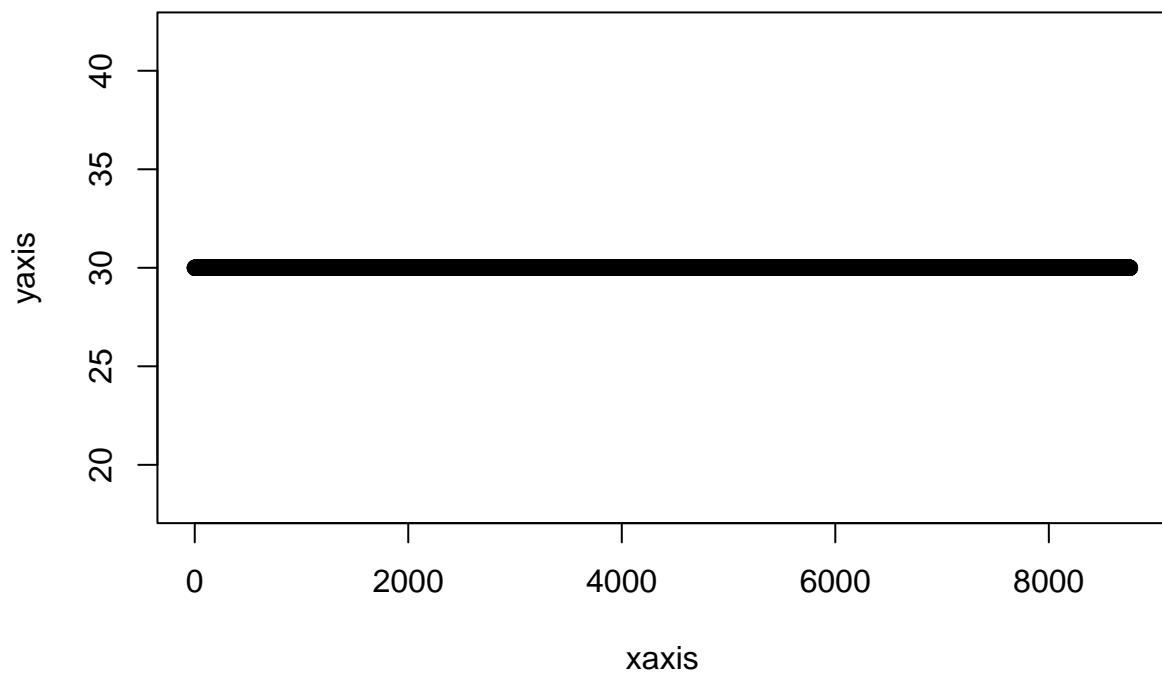
```
## [1] 949
## [1] 0 991
## integer(0)
## [1] 23
## [1] 5 27
## integer(0)
## [1] 20
## [1] 0 19
## integer(0)
## [1] 303
## [1] 1 360
```

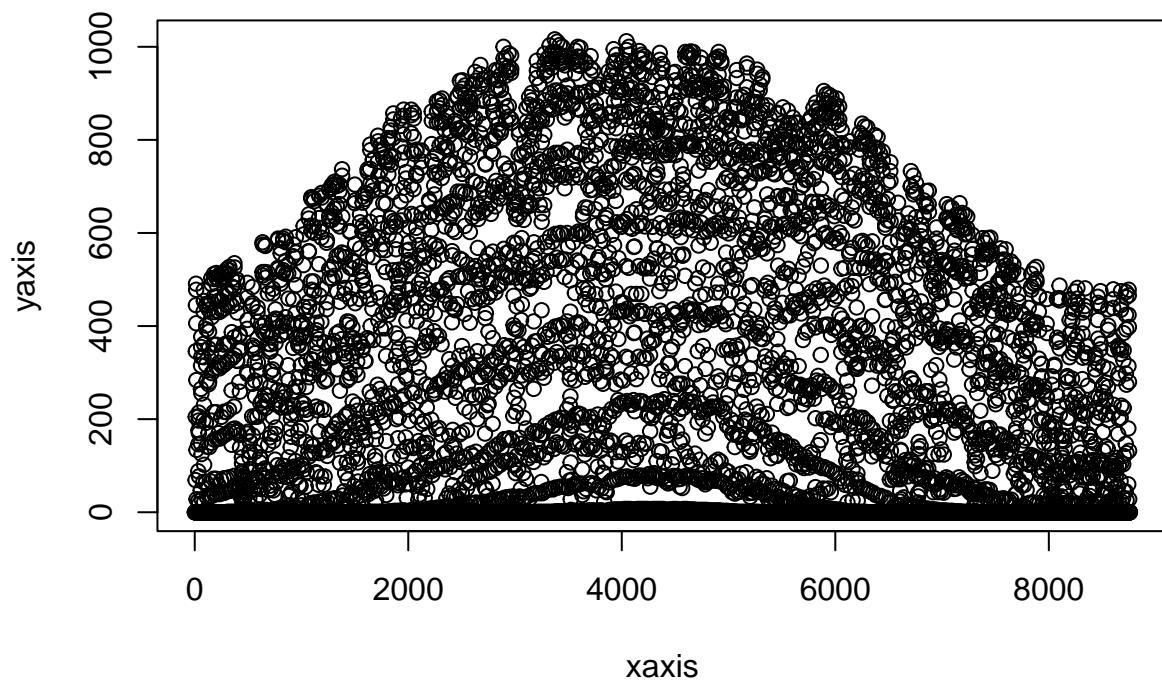


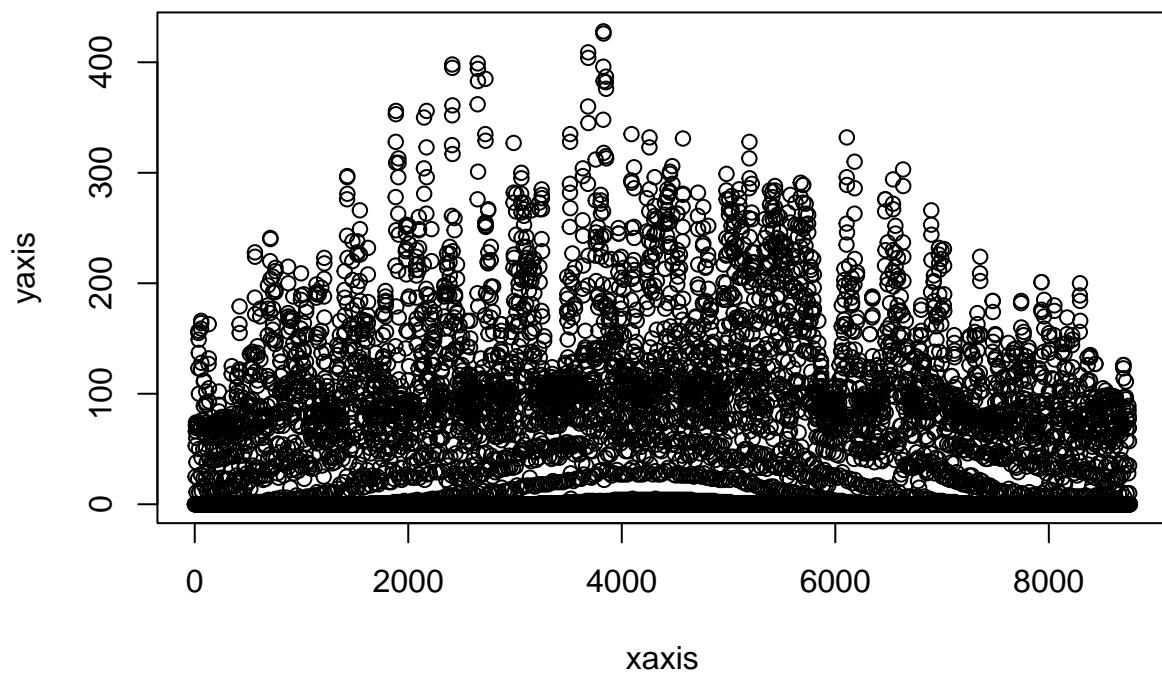


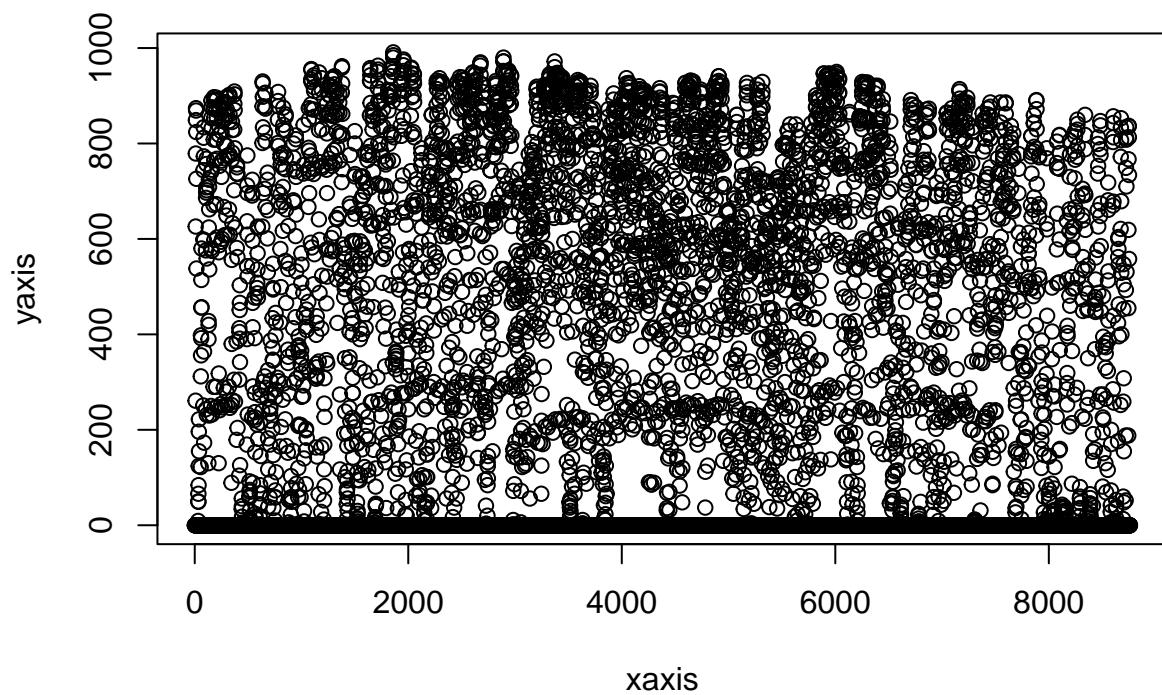


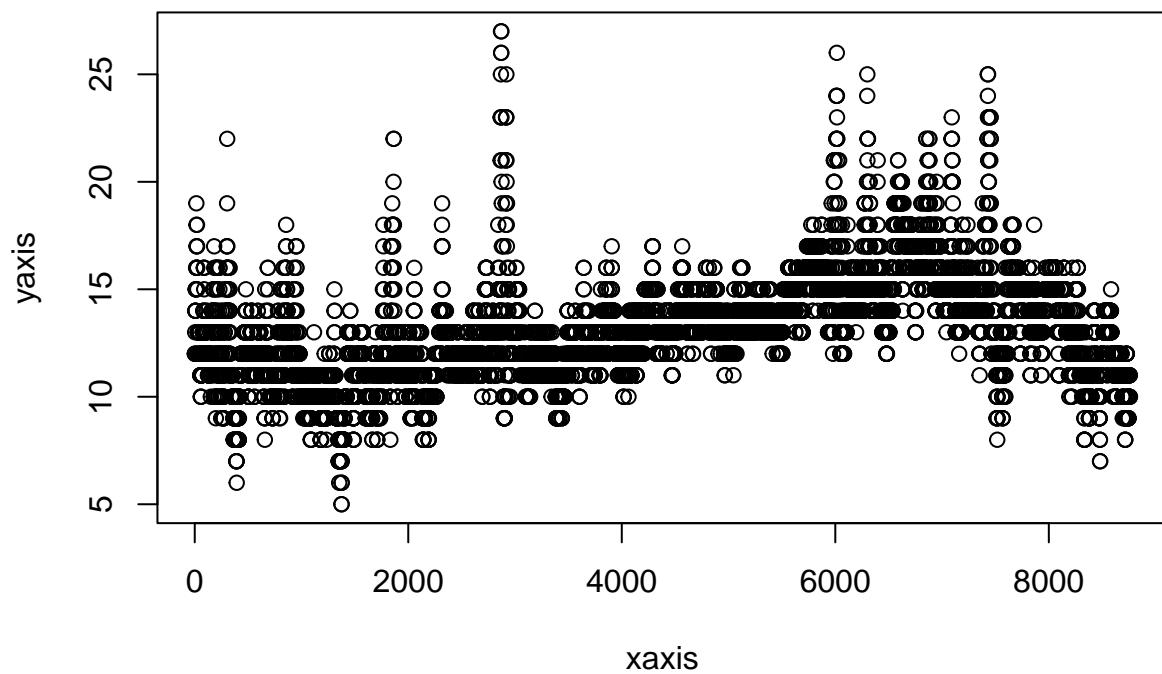


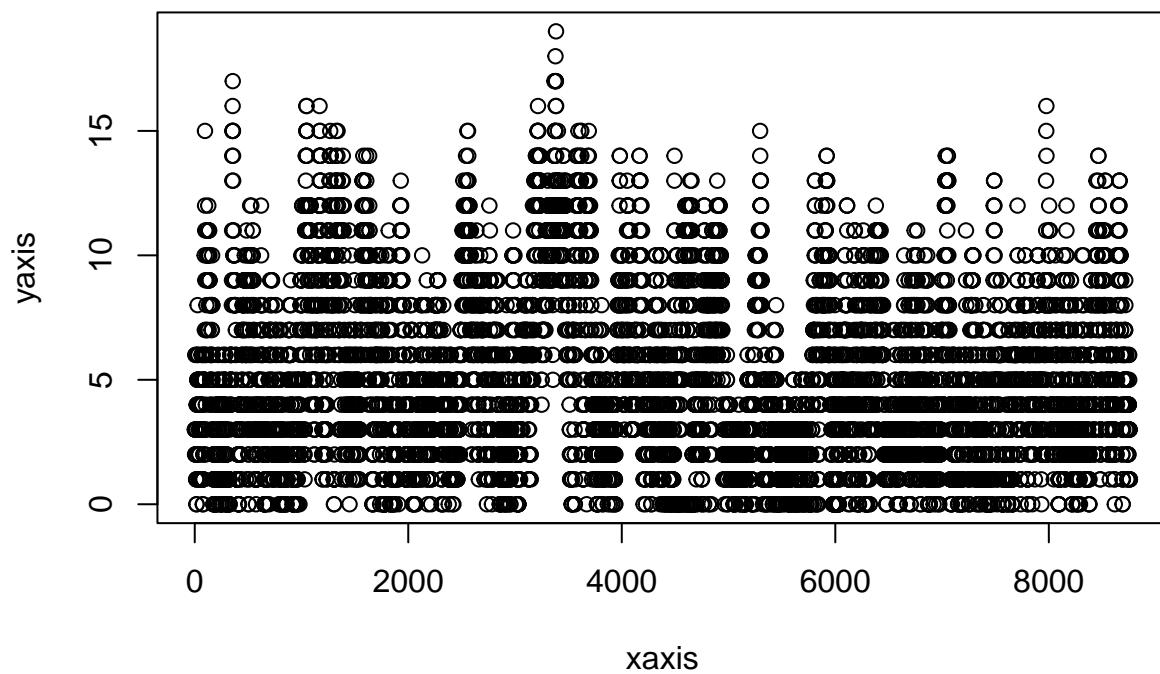


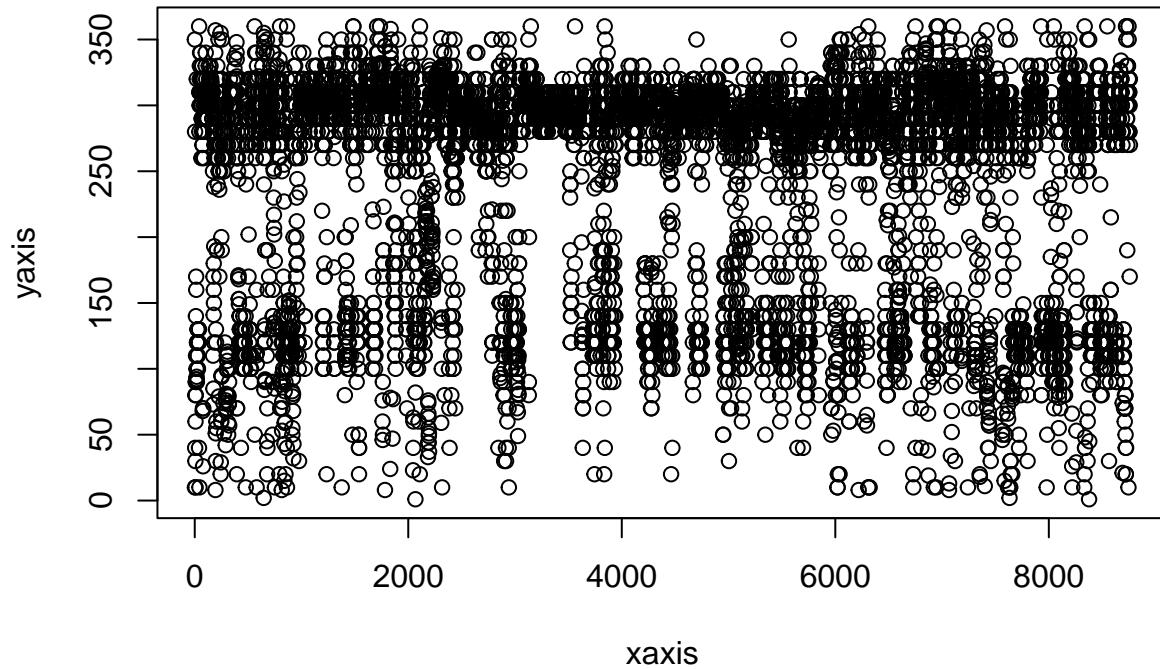








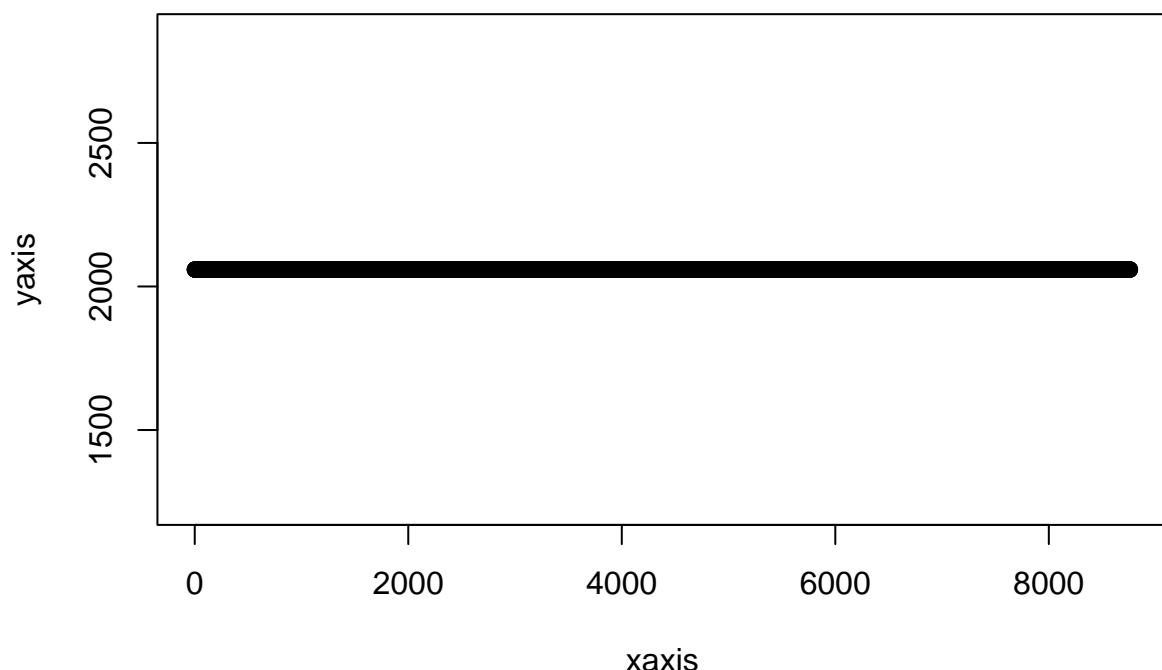


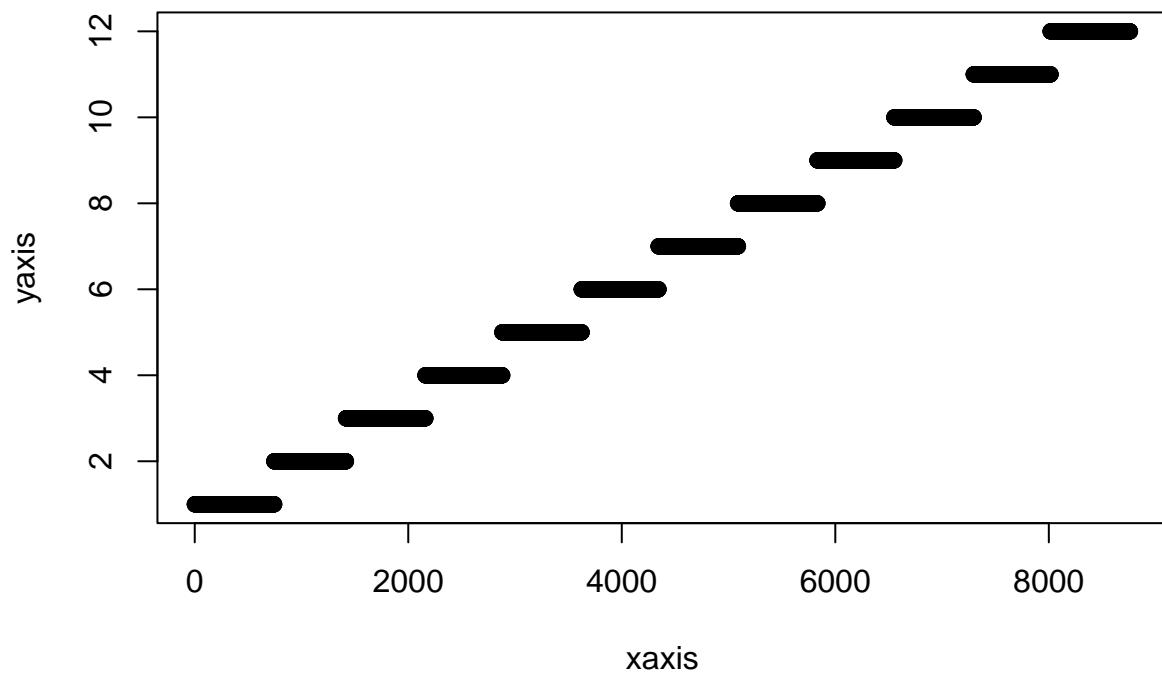


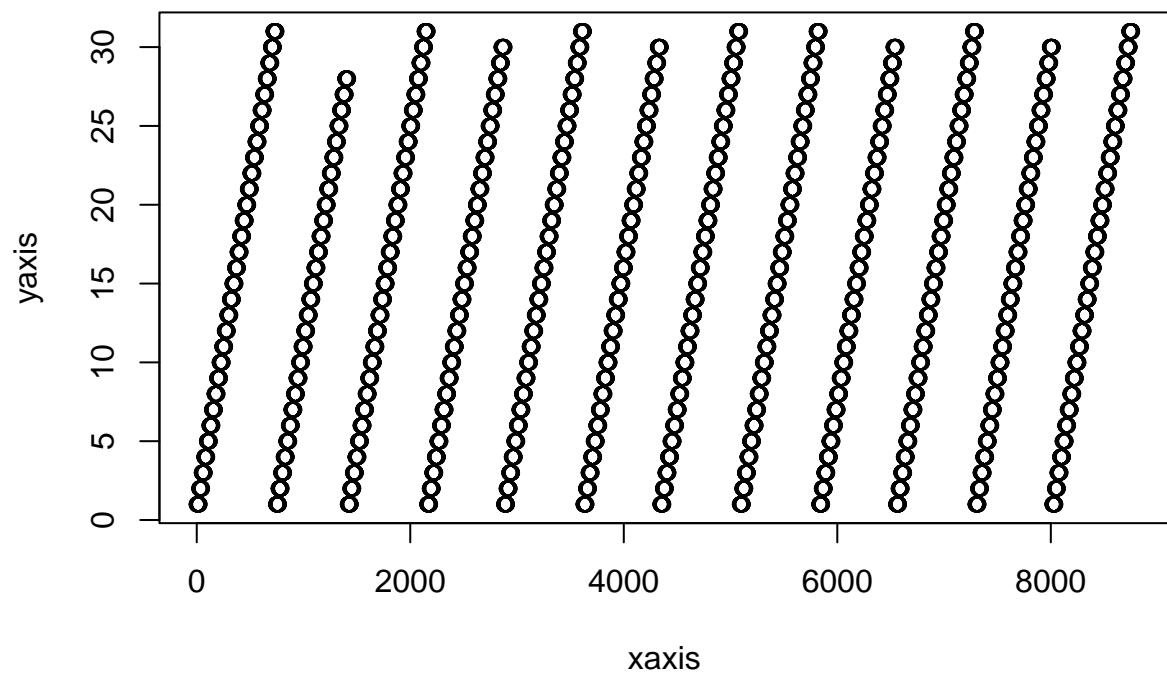
```
pv_UCD = pvsyst_parser('dt/USA_CA_UC-Davis-University.AP.720576_TMYx.2007-2021.pvsyst')
```

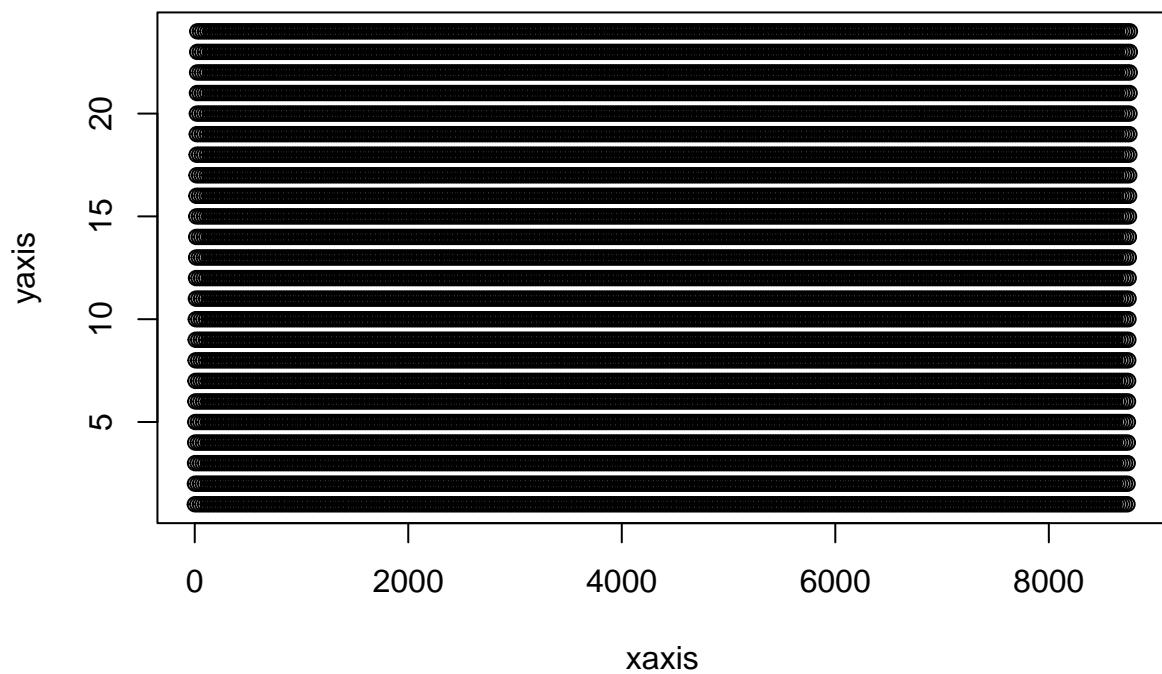
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] 2059 2059
## integer(0)
## [1] 12
## [1] 1 12
## integer(0)
## [1] 31
## [1] 1 31
## integer(0)
## [1] 24
## [1] 1 24
## integer(0)
## [1] 1
## [1] 30 30
## integer(0)
## [1] 986
## [1] 0 1017
## integer(0)
## [1] 322
## [1] 0 479
## integer(0)
```

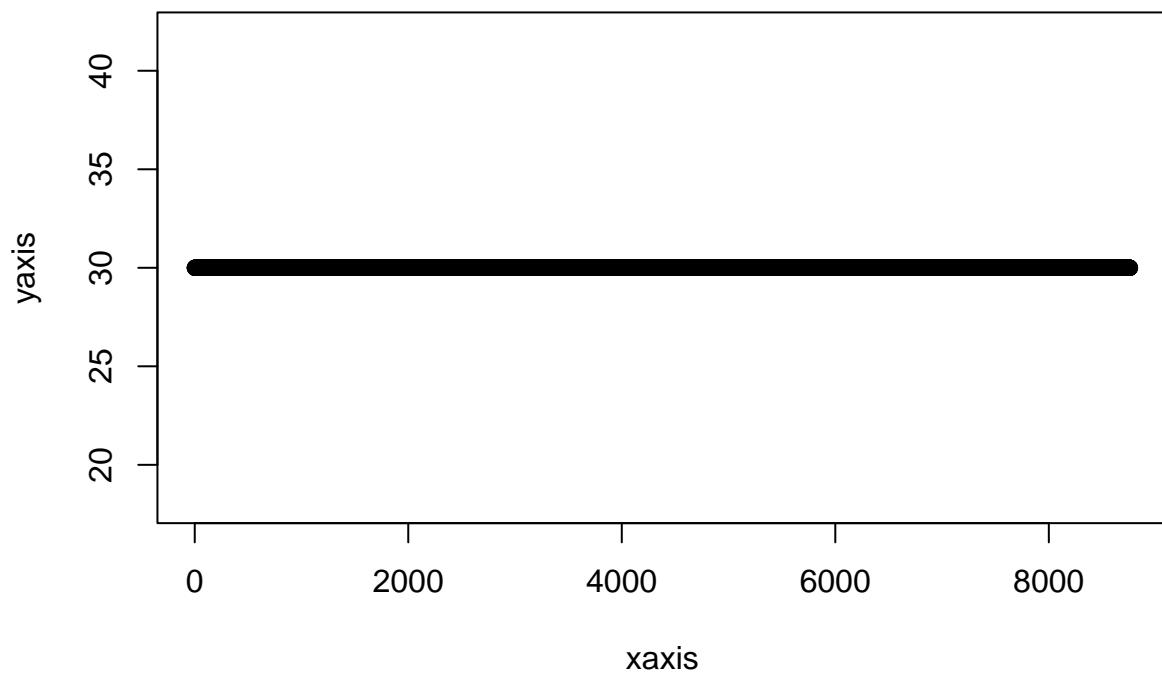
```
## [1] 928
## [1] 0 985
## integer(0)
## [1] 44
## [1] -1 42
## integer(0)
## [1] 15
## [1] 0 14
## integer(0)
## [1] 354
## [1] 0 360
```

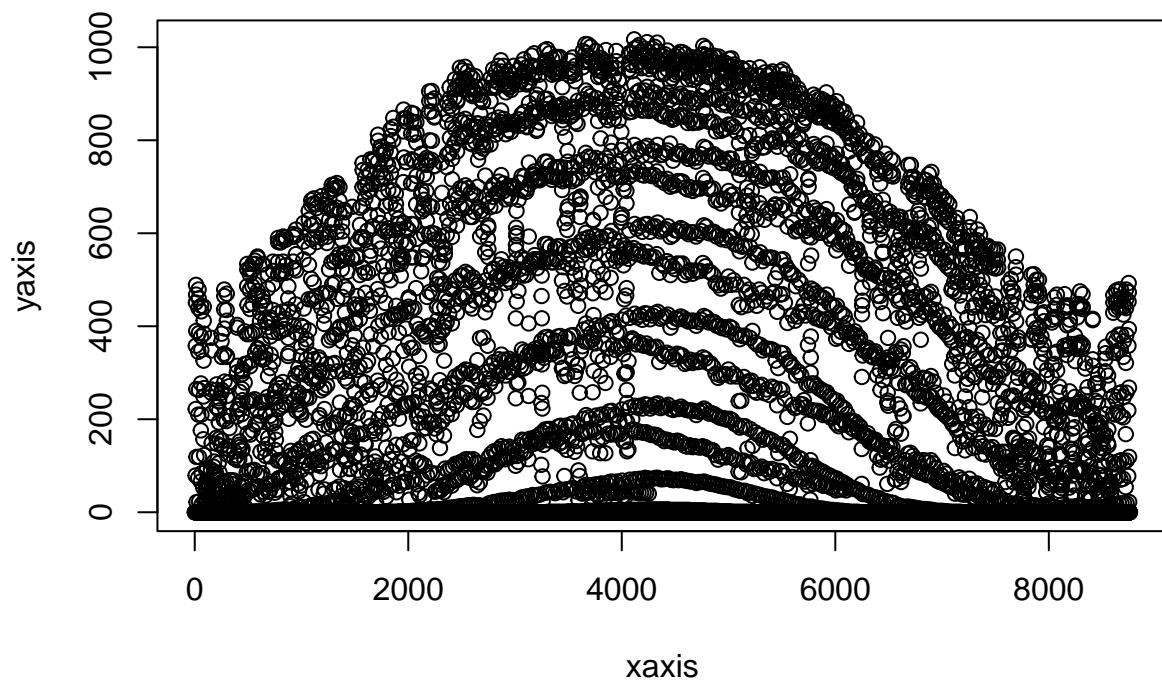


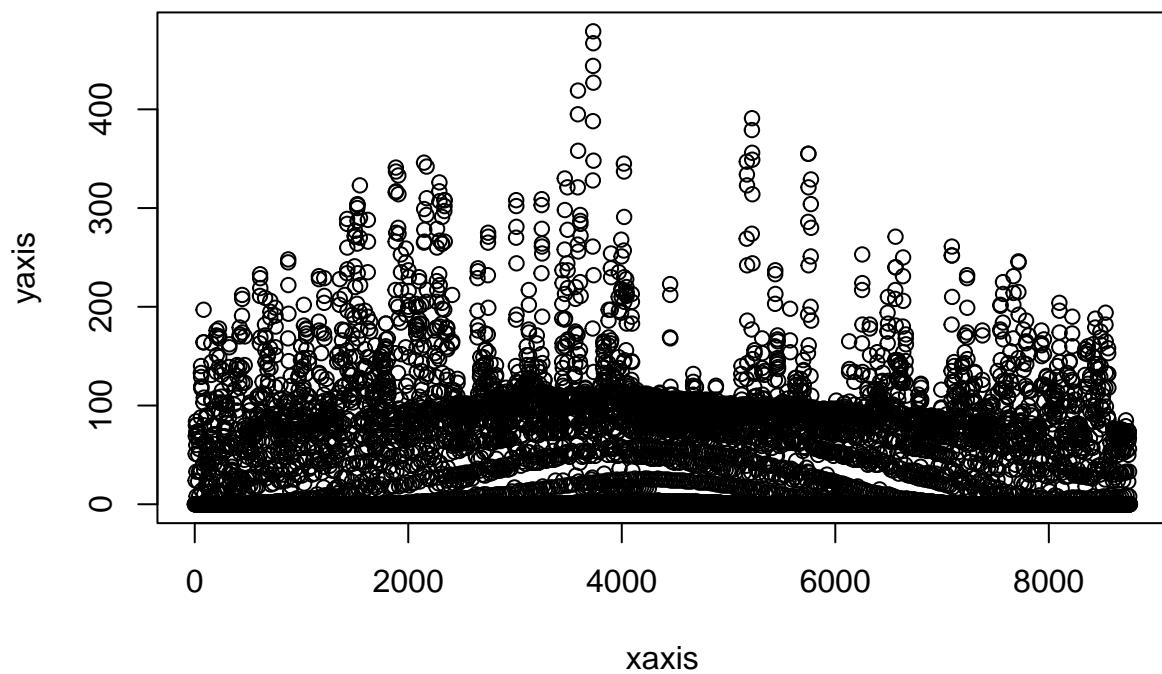


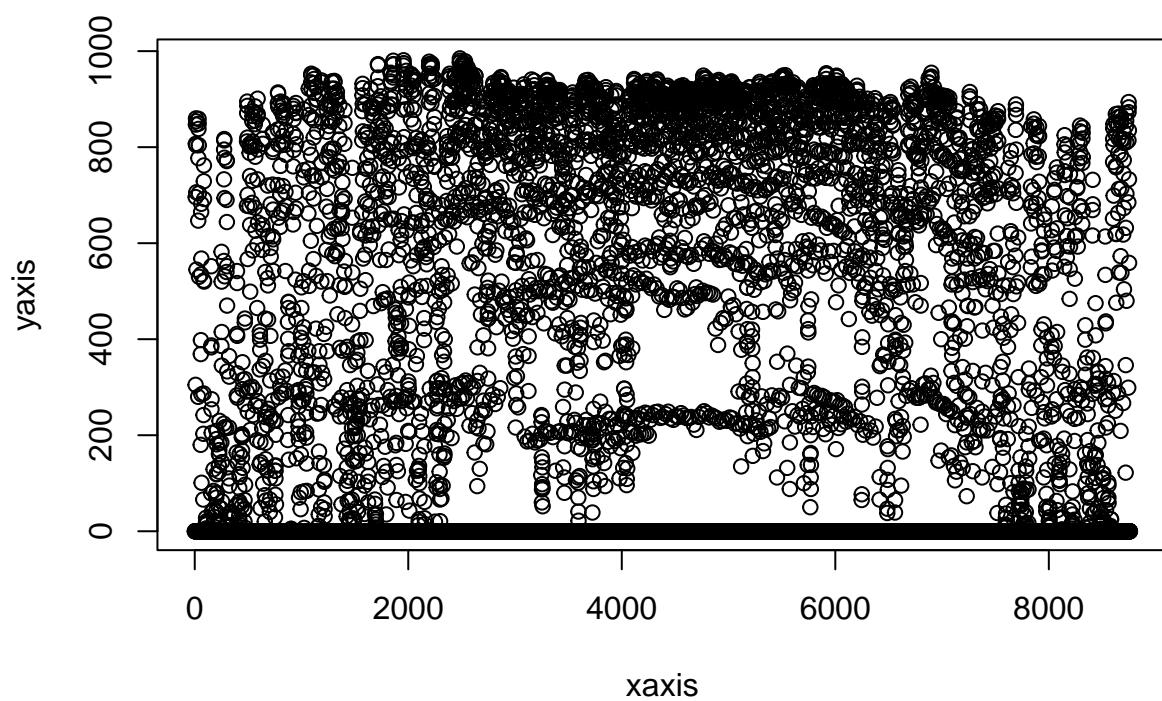


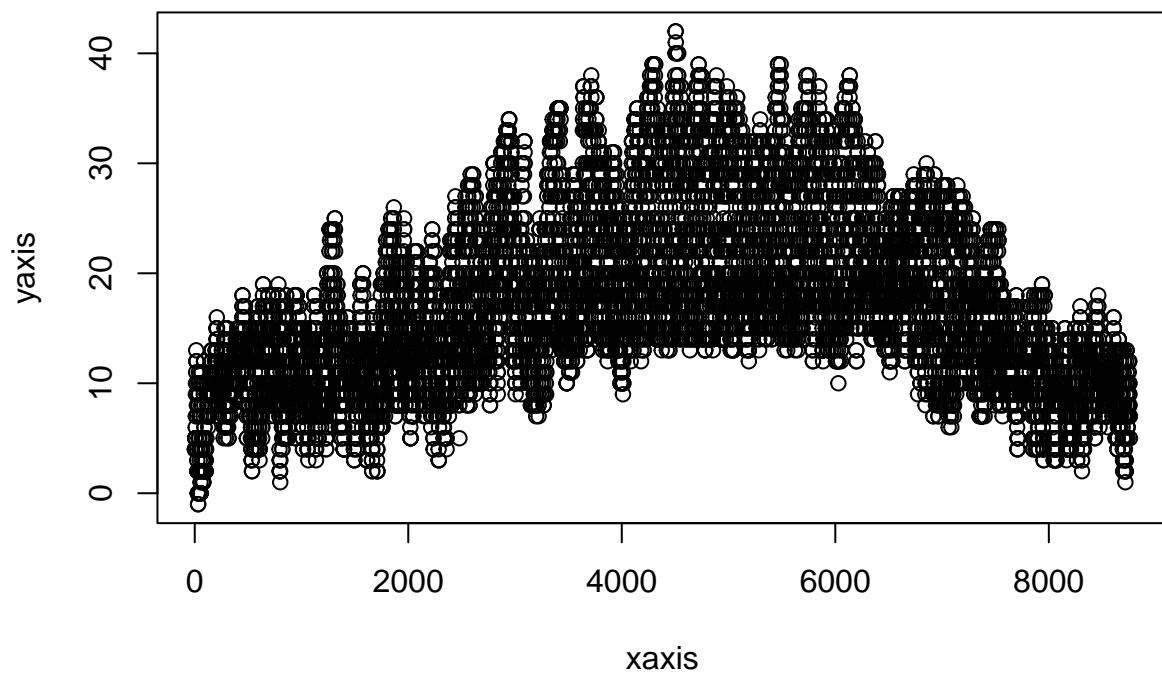


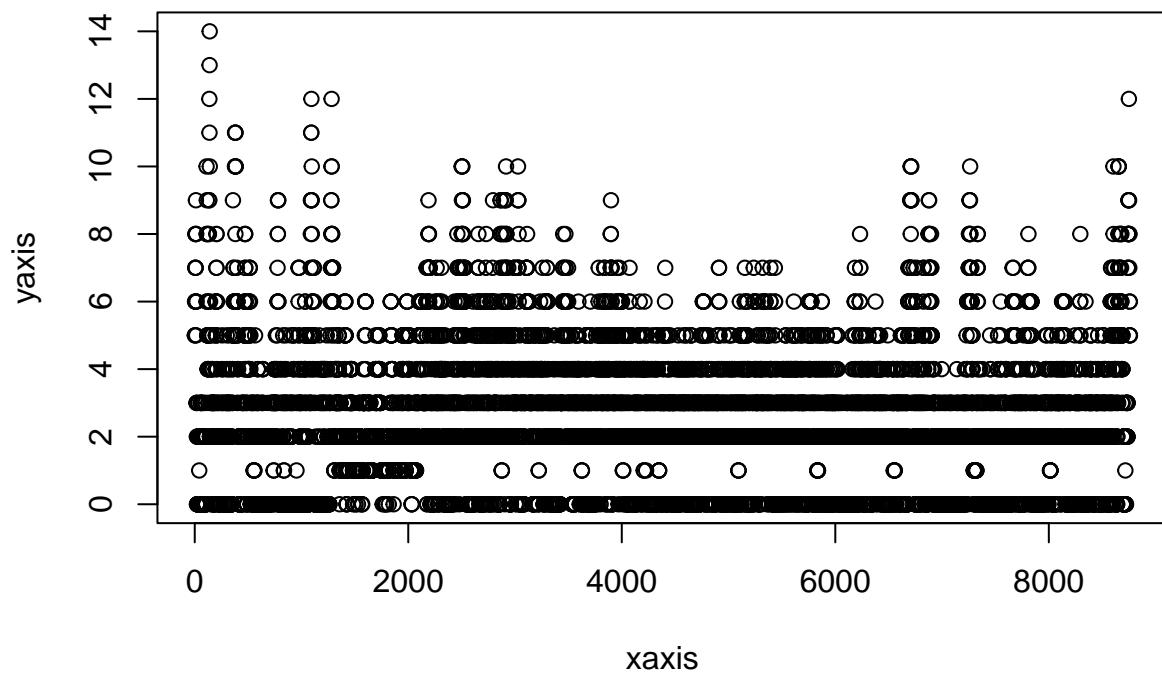


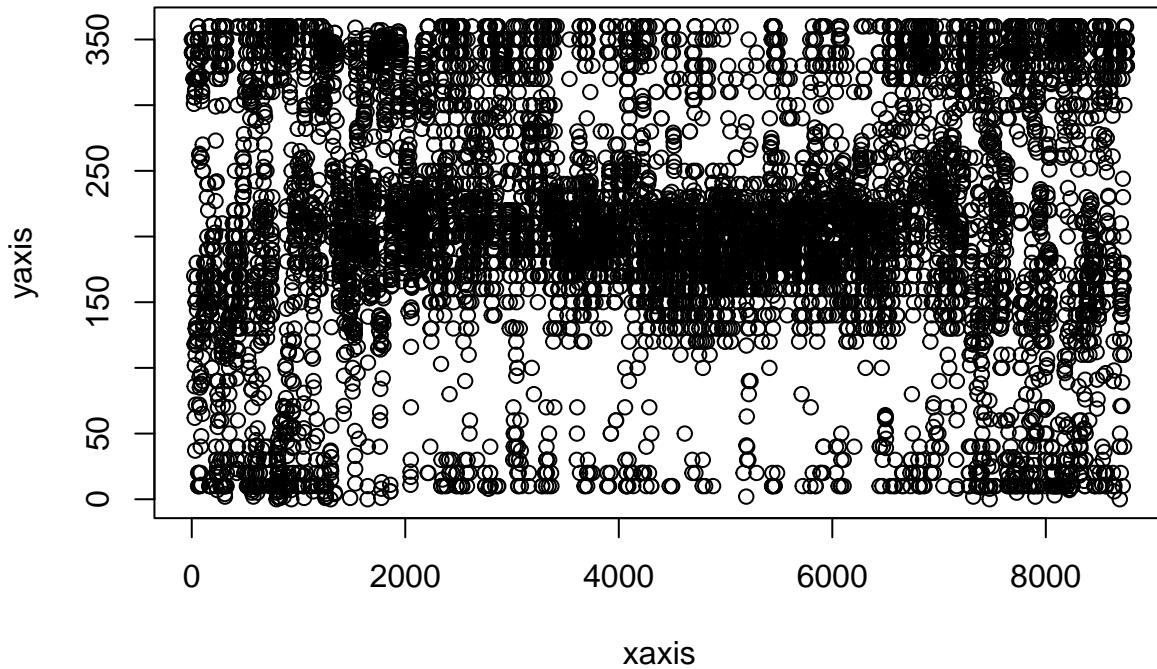












## STAT FILE

### Monthly Statistics for Dew Point & Dry Bulb

```

month_parser <- function(stat_ll, start, end, wind=FALSE){
  db_month_dt = stat_ll[(start+1):(end-1)]

  #remove white space lines
  db_month_dt = db_month_dt[c(-which(db_month_dt == ""))]
  #Omits the month row and the Max/Min row from the actual table
  db_table = read.table(textConnection(db_month_dt[2:(length(db_month_dt)-2)]), sep = '\t')

  #add column names
  colnames(db_table) = read.table(textConnection(db_month_dt[1]), sep = '\t')
  db_table = db_table[,c(-1, -ncol(db_table))]

  #Reformat table so months are rows and min/max/etc is columns
  db_table = data.frame(t(db_table))
  colnames(db_table) = db_table[1,]
  db_table = db_table[-1,]

  #convert all non-times to integers
  db_table = asnum_col(db_table, c(1,3,5:ncol(db_table)))
  #convert all times to POSIXct
  db_table = posix_convert(db_table, c(2,4))
}

```

```

#verify max/min times:
maxmin_dt = db_month_dt[c(length(db_month_dt)-1,length(db_month_dt))]
#True Max, Min
max_db = max(db_table$Maximum)
min_db = min(db_table$Minimum)
#extract the values from str
max_str = strsplit(maxmin_dt[[1]], " ")
min_str = strsplit(maxmin_dt[[2]], " ")
max_str = max_str[[1]][-c(which(max_str[[1]] == ""))]
min_str = min_str[[1]][-c(which(min_str[[1]] == ""))]

#Special case to accommodate format for Wind Spd Monthly table
val_idx = 7
month_sub = 1
if (wind == TRUE){
  val_idx = 6
  month_sub = 0}

#Max Handling
max_data = c(as.numeric(substr(max_str[val_idx], 1,nchar(max_str[val_idx])-month_sub)), max_str[9], as.numeric(max_str[2]))
#Min Handling
min_data = c(as.numeric(substr(min_str[val_idx], 1,nchar(min_str[val_idx])-month_sub)), min_str[9], as.numeric(min_str[2]))

#print Warnings in case they do not match
if (max_db != db_table[as.numeric(max_data[2]),1]){
  print('Maximum values do not match!')
  print(paste(max_db, '!=', db_table[as.numeric(max_data[2]),1])))
#max_data[1] =1
if (max_db != as.numeric(max_data[1])){
  print("True Maximum and Stated Maximum do not match!")
  print(paste(max_db, '!=', as.numeric(max_data[1])))}
#Min Warnings
if (min_db != db_table[as.numeric(min_data[2]),3]){
  print('Minimum values do not match!')
  print(paste(min_db, '!=', db_table[as.numeric(min_data[2]),3]))}
if (min_db != as.numeric(min_data[1])){
  print("True Minimum and Stated Minimum do not match!")
  print(paste(min_db, '!=', as.numeric(min_data[1])))}

if (DEBUG == TRUE){
  debug_stats(db_table, ncol(db_table))
  vis_inspc(db_table, c(1:ncol(db_table)))
}

return(db_table)
}

```

Monthly Wind Direction (Interval 11.25 deg from displayed deg)

```

wind_dir_parse <- function(stat_ll){
  start = grep('Monthly Wind Direction', stat_ll)
  end = grep('Monthly Statistics for Wind Speed', stat_ll)
  db_month_dt = stat_ll[(start+1):(end-1)]

  #remove white space lines
  db_month_dt = db_month_dt[c(-which(db_month_dt == ""))]
  #Omits the month row and the Max/Min row from the actual table
  db_table = read.table(textConnection(db_month_dt[2:length(db_month_dt)]), sep = '\t')

  #add column names
  colnames(db_table) = read.table(textConnection(db_month_dt[1]), sep = '\t')
  db_table = db_table[,c(-1, -ncol(db_table))]

  #Reformat table so months are rows and min/max/etc is columns
  db_table = data.frame(t(db_table))
  colnames(db_table) = db_table[1,]
  db_table = db_table[-1, ]

  if (DEBUG == TRUE){
    debug_stats(db_table, ncol(db_table))
    vis_inspc(db_table, c(1:ncol(db_table)))
  }

  return(db_table)
}

```

## Hourly Parser

```

hourly_parser <- function(stat_ll, start, end){
  db_month_dt = stat_ll[(start+1):(end-1)]

  #remove white space lines
  db_month_dt = db_month_dt[c(-which(db_month_dt == ""))]
  #Omits the month row and the Max/Min row from the actual table
  db_table = read.table(textConnection(db_month_dt[2:(length(db_month_dt)-2)]), sep = '\t')

  #add column names
  colnames(db_table) = read.table(textConnection(db_month_dt[1]), sep = '\t')
  db_table = db_table[,c(-1, -ncol(db_table))]

  #Reformat table so months are rows and min/max/etc is columns
  db_table = data.frame(t(db_table))
  colnames(db_table) = db_table[1,]
  db_table = db_table[-1, ]

  #We now create our (24 x 12=) 288 * 3 table
  months = row.names(db_table)
  hours = colnames(db_table)
  data_col = vector()
  #Orders all Jan hours, then all Feb hours, ..., then all Dec hours
  for (mon in 1:length(months)){
    for (hr in 1:length(hours)){

```

```

    data_col = c(data_col, db_table[mon, hr])}

#We know that we will have 288 values at the end
if (DEBUG == TRUE){
  if (length(data_col) != 288){
    print('Missing hourly data!')
    print(paste('Length of hourly data:', length(data_col)))}

#Using that fact, we can generate appropriate length hour & month columns
hour_col = rep(hours, length(data_col)/length(hours))
month_col = vector()
#generate months in batches of 24 to line up with the data temporarily
for (i in 1:length(months)){
  month_col = c(month_col, rep(months[[i]], 24))
}

hourly_table = data.frame(data_col, month_col, hour_col)

hourly_table = trim_time(hourly_table, 3)
hourly_table = asnum_col(hourly_table, c(1,3))

if (DEBUG == TRUE){debug_stats(db_table, ncol(db_table))}

return(hourly_table)
}

trim_time <- function(df, col){
  for (i in 1:nrow(df)){
    df[i,col] = as.integer(as.integer(trimws(substring(df[i,col], 0,2))))
  }
  return(df)}

```

## Combining the hourly data

```

combine_hours <- function(df1, df2, df3, df4, df5){
  col_names = c('Month', 'Hour', 'Dry Bulb Temp', 'Dew Point Temp', 'Relative Humidity', 'Dir Normal So
  df_c = c(df2, df3, df4, df5)

  return_df = data.frame(df1$month_col, df1$hour_col, df1$data_col, df2$data_col, df3$data_col, df4$data
  colnames(return_df) = col_names

  if (DEBUG == TRUE){debug_stats(return_df, ncol(return_df))}

  return(return_df)
}

```

## Graphing the hourly data

```

graphAttrs <- function(df, locationstr){
  lct_str = substring(locationstr, 4, nchar(locationstr)-27)

  month_list = unique(df[,1])

```

```

for (i in 1:length(month_list)){
  #[5] Taking subsets of data in ggplot
  #[6] How to make ggplot appear in for loop
  print(ggplot(subset(df, Month %in% month_list[i])) +
    geom_point(aes(Hour, `Dry Bulb Temp`, color = 'orange')) +
    geom_point(aes(Hour, `Dew Point Temp`, color = 'lightblue')) +
    geom_point(aes(Hour, `Relative Humidity`, color = 'blue')) +
    geom_point(aes(Hour, `Dir Normal Solar Radiation`, color = 'yellow')) +
    geom_point(aes(Hour, `Wind Spd`)) +
    ylab('Variables') + ggtitle(paste(month_list[i], 'Data:', lct_str)))
}
}

```

### stat file master parser

```

stat_parser <- function(stat_path){
  stat_ll <- readLines(stat_path)

  start = grep('Monthly Statistics for Dry Bulb temperatures', stat_ll)
  end = grep('Monthly Statistics for Extreme Dry Bulb temperatures', stat_ll)
  drybulb_month_df = month_parser(stat_ll, start, end)

  start = grep('Monthly Statistics for Dew Point temperatures', stat_ll)
  end = grep('Average Hourly Statistics for Dry Bulb temperatures', stat_ll)
  dew_point_month_df = month_parser(stat_ll, start, end)

  start = grep('Monthly Statistics for Wind Speed', stat_ll)
  end = grep('Average Hourly Statistics for Wind Speed', stat_ll)
  wind_spd_month_df = month_parser(stat_ll, start, end, TRUE)

  wind_dir_month_df = wind_dir_parse(stat_ll)

  start = grep('Average Hourly Statistics for Dry Bulb temperatures', stat_ll)
  end = grep('Average Hourly Statistics for Dew Point temperatures', stat_ll)
  avg_hr_dry_bulb_temp_df = hourly_parser(stat_ll, start, end)

  start = grep('Average Hourly Statistics for Dew Point temperatures', stat_ll)
  end = grep('Monthly Statistics for Relative Humidity', stat_ll)
  avg_hr_dew_pnt_temp_df = hourly_parser(stat_ll, start, end)

  start = grep('Average Hourly Relative Humidity', stat_ll)
  end = grep('Monthly Statistics for Enthalpy', stat_ll)
  avg_hr_relative_hum_df = hourly_parser(stat_ll, start, end)

  start = grep('Average Hourly Statistics for Direct Normal Solar Radiation ', stat_ll)
  end = grep('Average Hourly Statistics for Diffuse Horizontal Solar Radiation', stat_ll)
  avg_hr_dir_norm_solar_rad_df = hourly_parser(stat_ll, start, end)

  start = grep('Average Hourly Statistics for Wind Speed', stat_ll)
  end = grep('Average Hourly Statistics for Wind Direction', stat_ll)
  avg_hr_wind_spd_df = hourly_parser(stat_ll, start, end)

  final_hr_df = combine_hours(avg_hr_dry_bulb_temp_df,

```

```

    avg_hr_dew_pnt_temp_df,
    avg_hr_relative_hum_df,
    avg_hr_dir_norm_solar_rad_df,
    avg_hr_wind_spd_df
  )

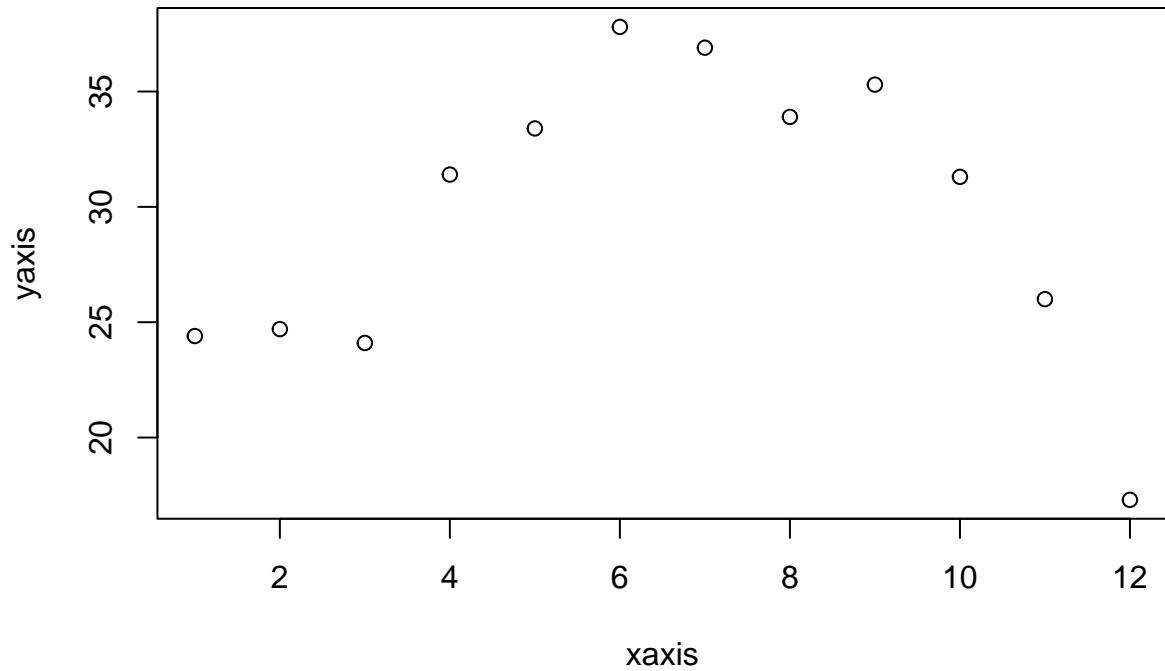
graphAttrs(final_hr_df, stat_path)

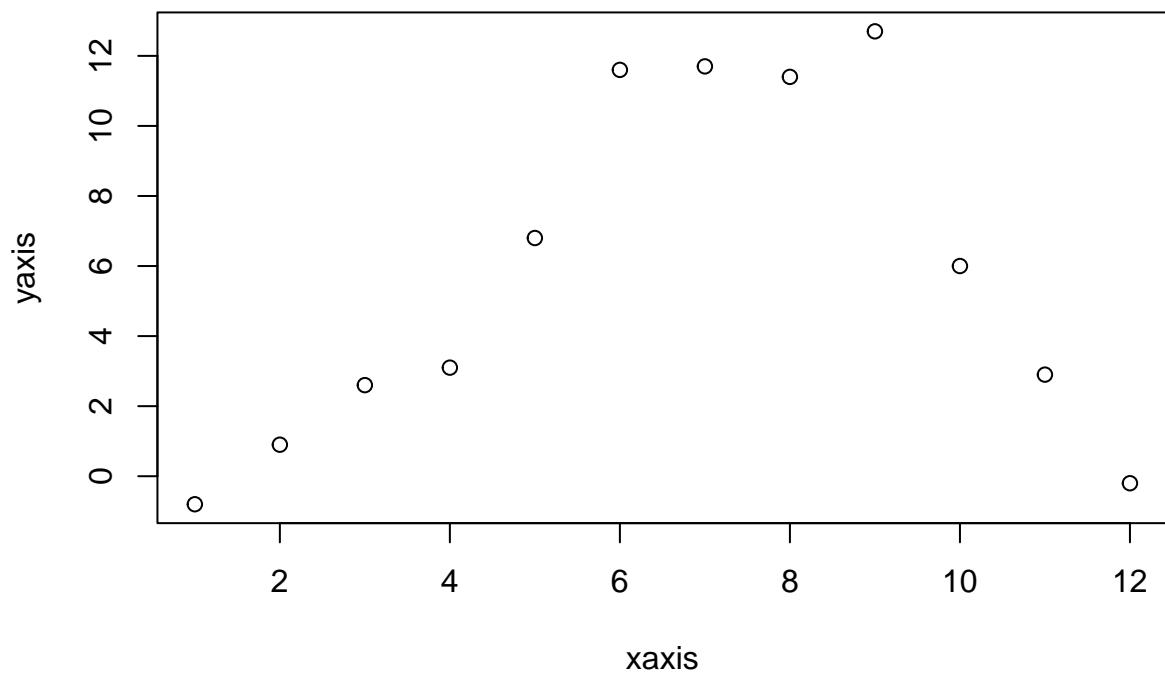
if (ReturnDataframes == FALSE){
  print(drybulb_month_df)
  print(dew_point_month_df)
  print(wind_spd_month_df)
  print(wind_dir_month_df)
  print(final_hr_df)
} else {
  return(c(drybulb_month_df, dew_point_month_df, wind_spd_month_df, wind_dir_month_df, final_hr_df))
}
stat_SF = stat_parser('dt/USA_CA_Fairfield-San.Francisco.Bay.Reserve.998011_TMYx.2007-2021.stat')

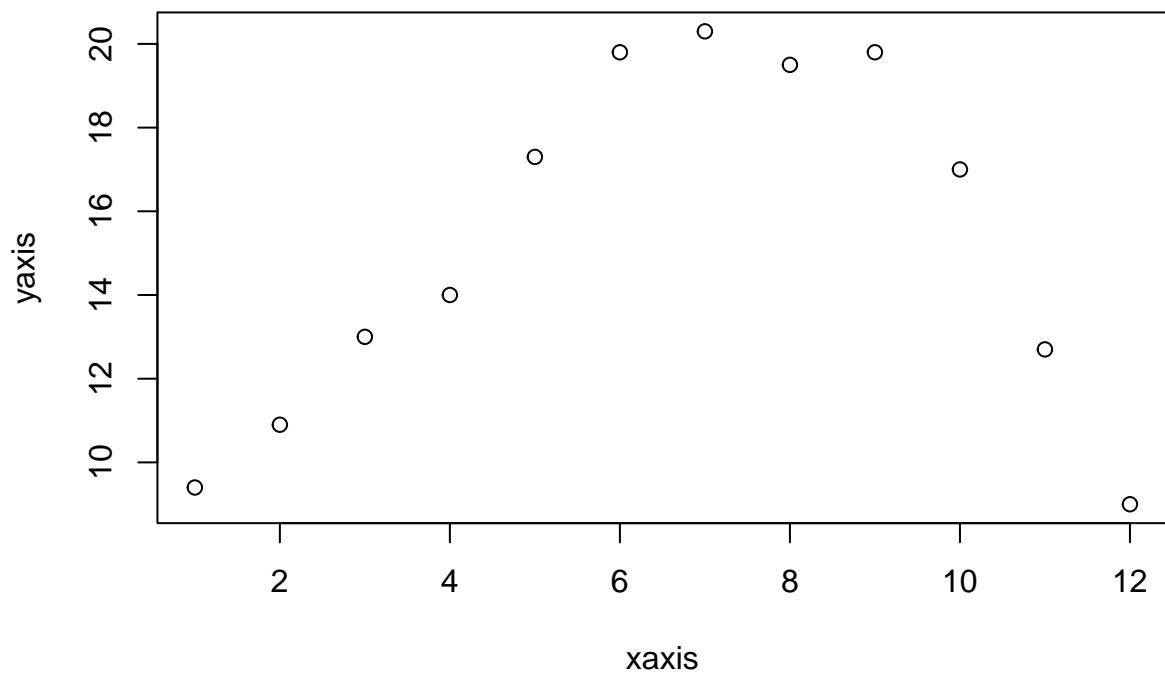
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 17.3 37.8
## integer(0)
## [1] 12
## [1] "2023-01-31 16:00:00" "2023-12-25 16:00:00"
## integer(0)
## [1] 12
## [1] -0.8 12.7
## integer(0)
## [1] 12
## [1] "2023-01-02 07:00:00" "2023-12-29 08:00:00"
## integer(0)
## [1] 11
## [1] 9.0 20.3
## integer(0)
## [1] 10
## [1] 9.5 15.8
## integer(0)
## [1] 12
## [1] 17.3 37.8
## integer(0)
## [1] 10
## [1] -0.8 12.7
## integer(0)
## [1] 11
## [1] 10.6 23.7
## integer(0)
## [1] 12
## [1] 13.1 34.6
## integer(0)
## [1] 11

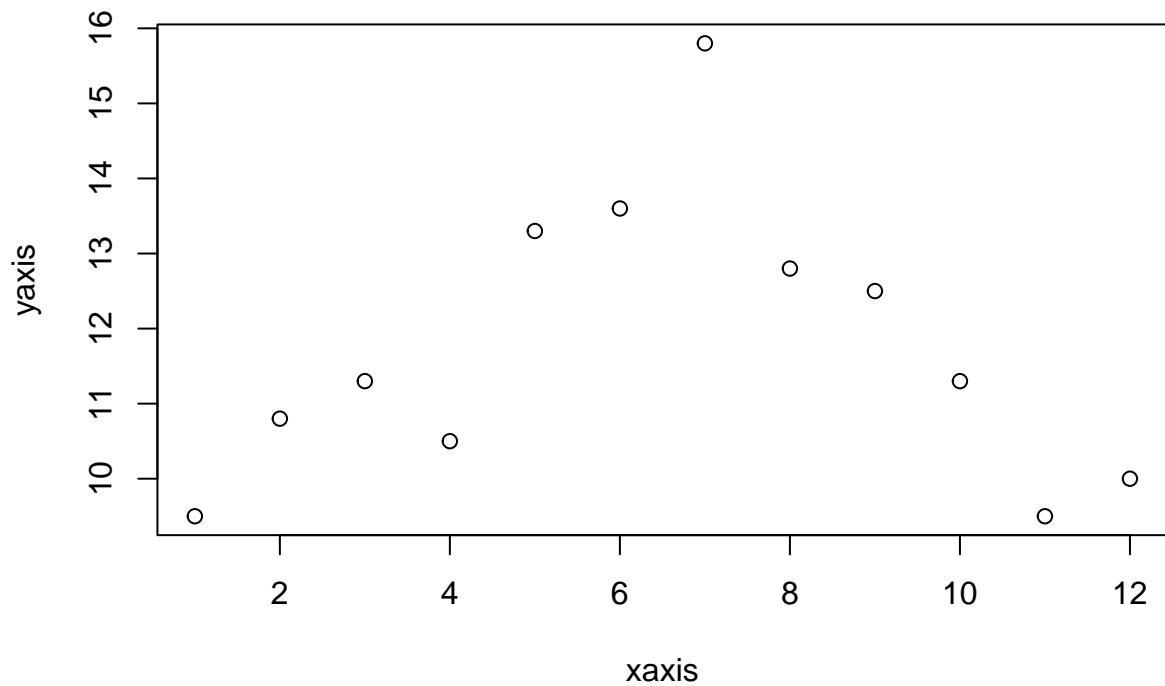
```

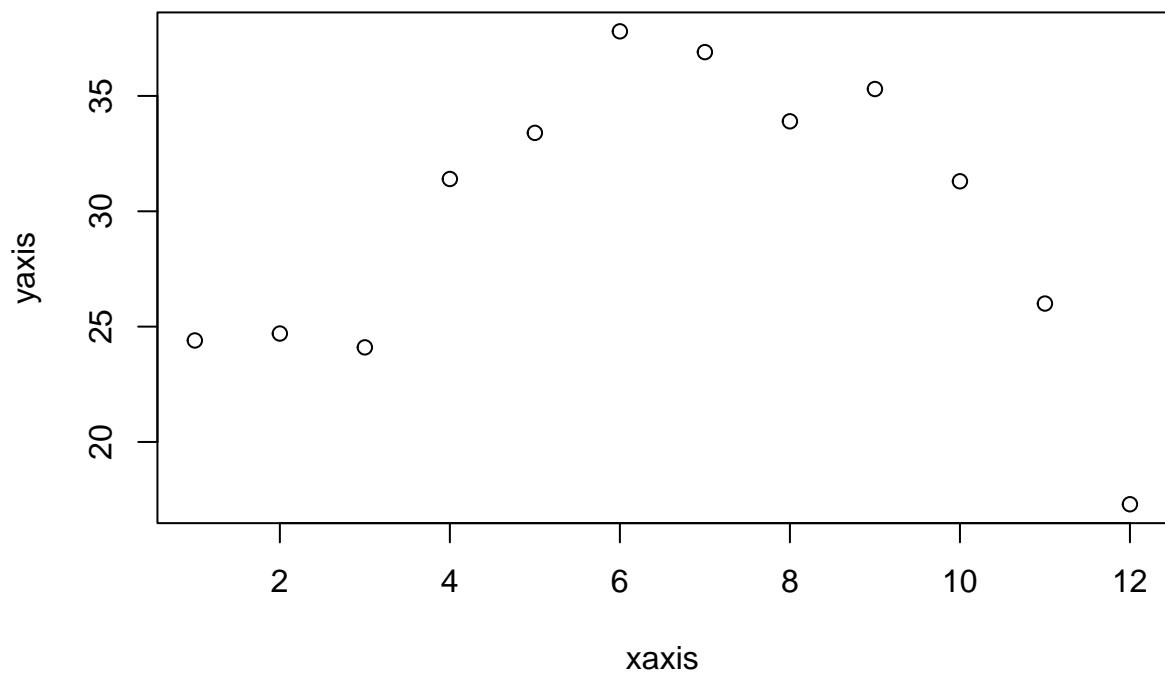
```
## [1] -0.3 12.8  
## integer(0)  
## [1] 11  
## [1] 7.4 17.5
```

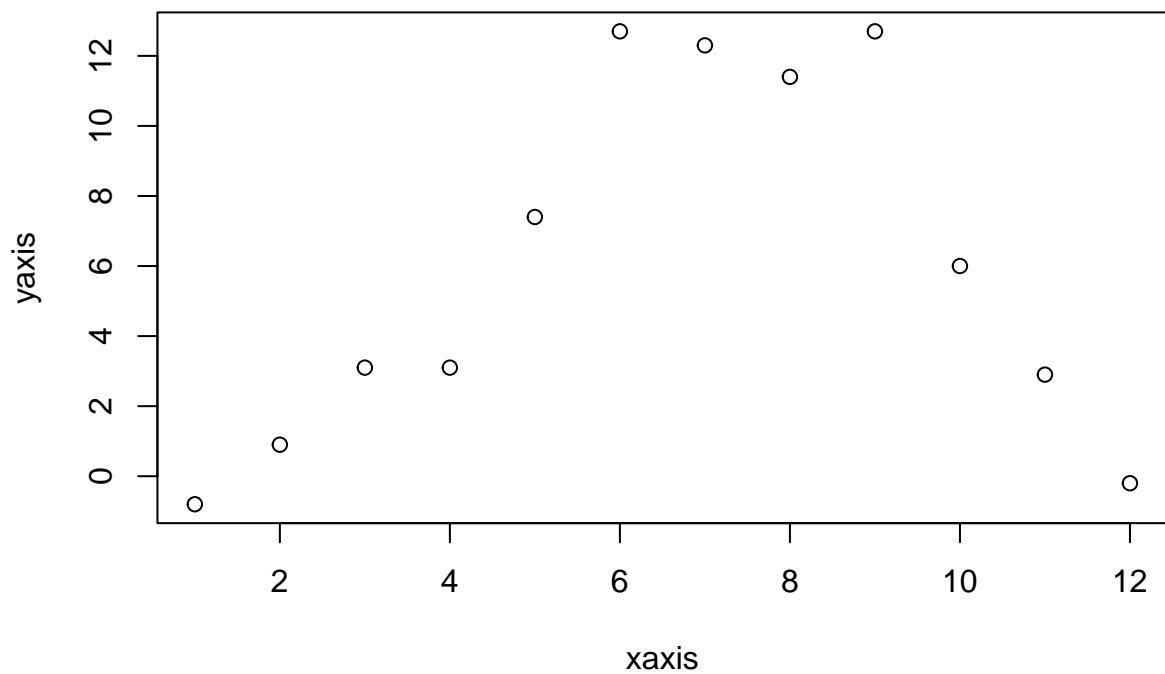


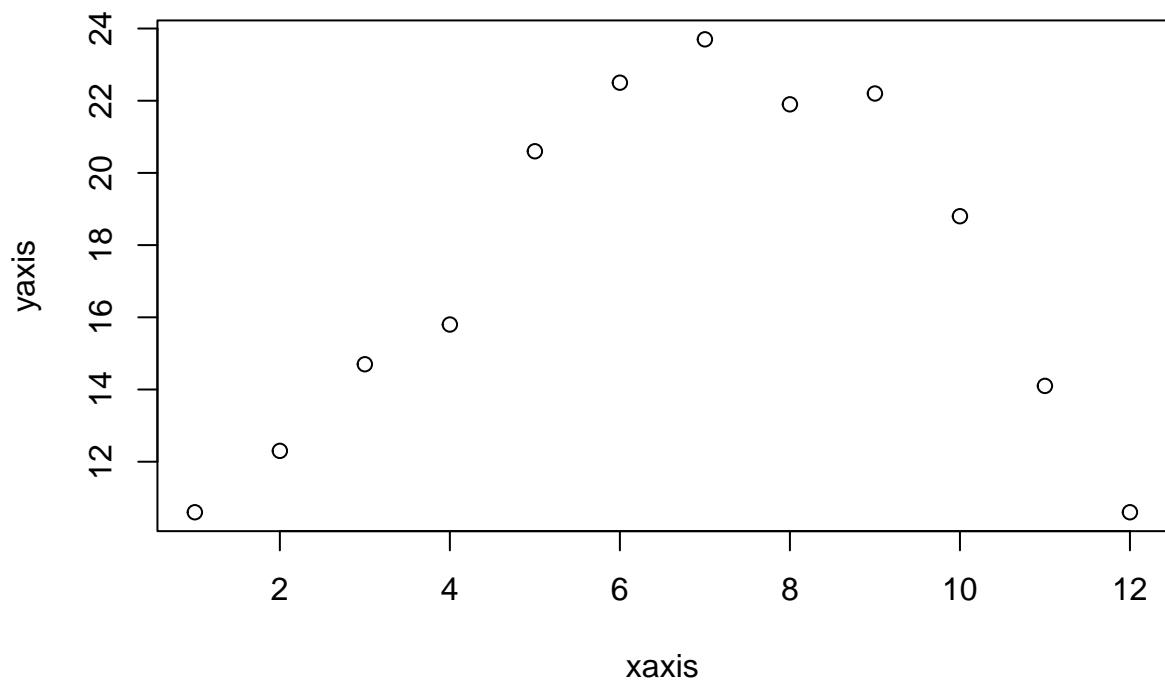


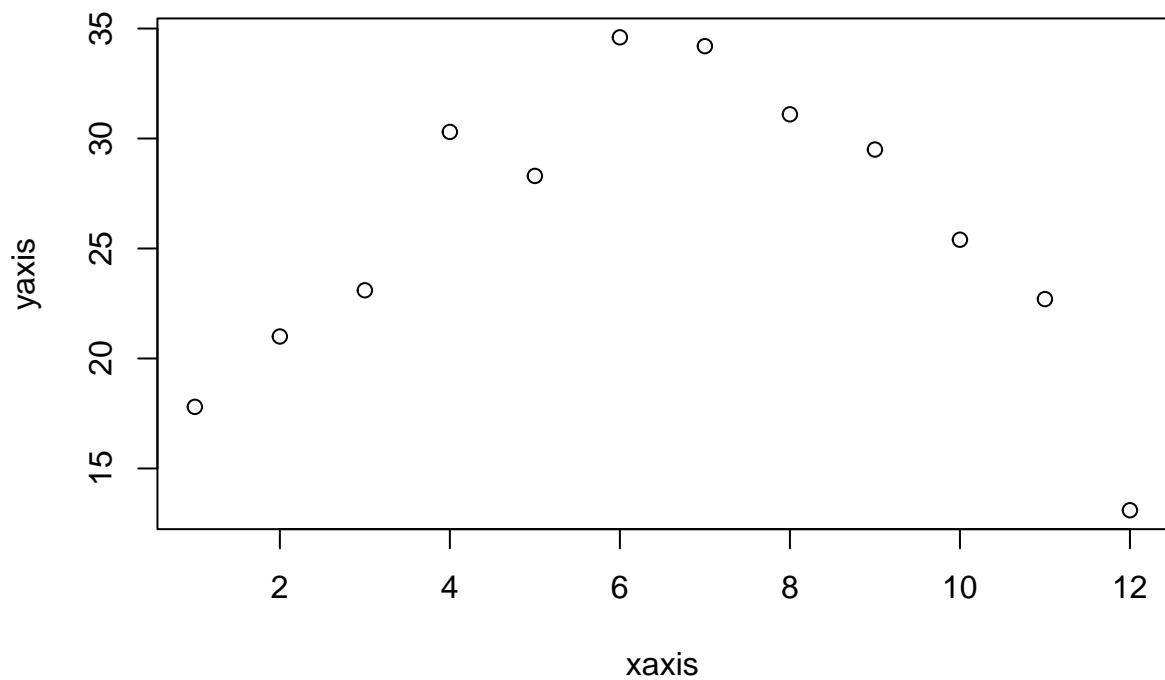


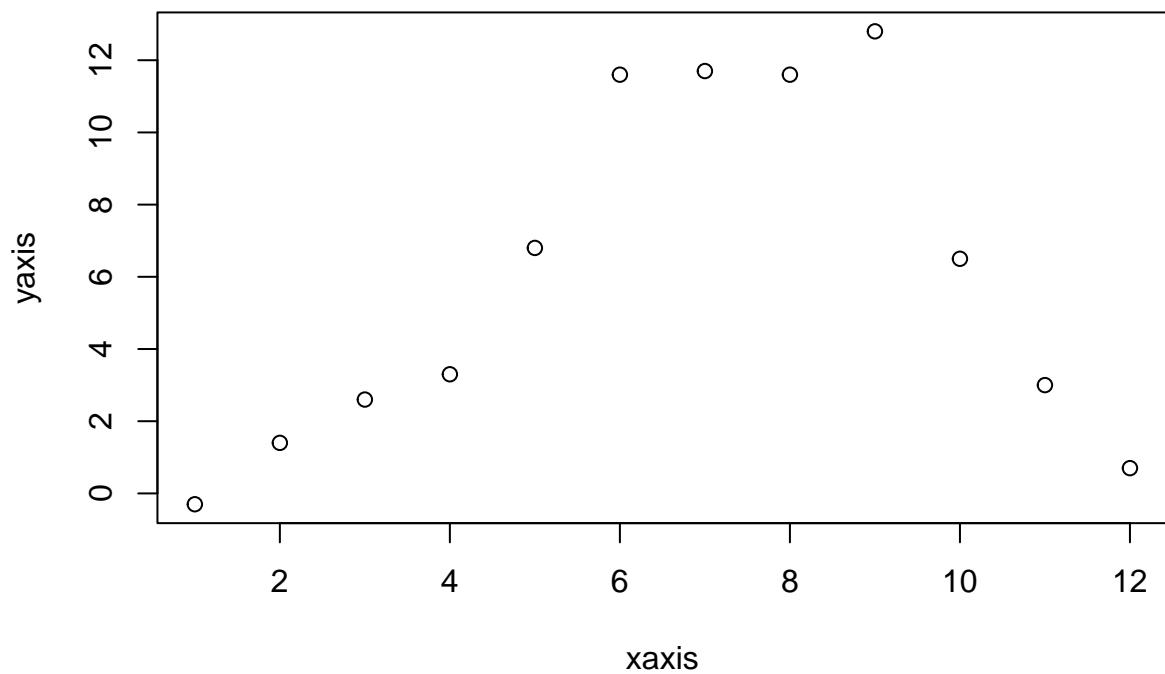


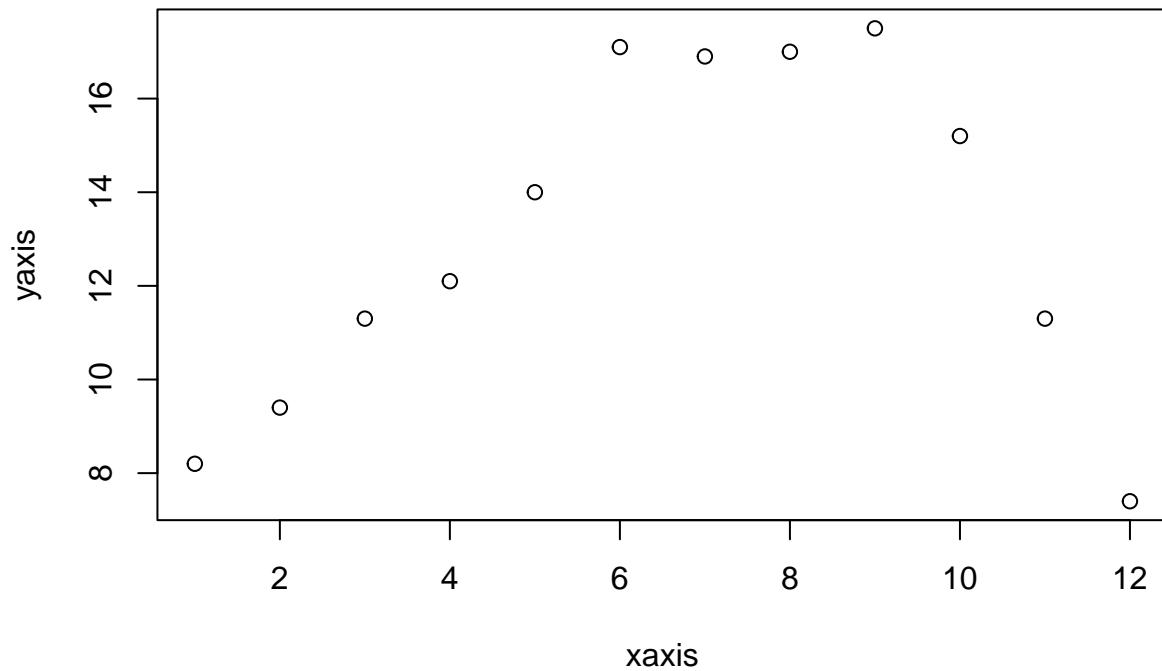




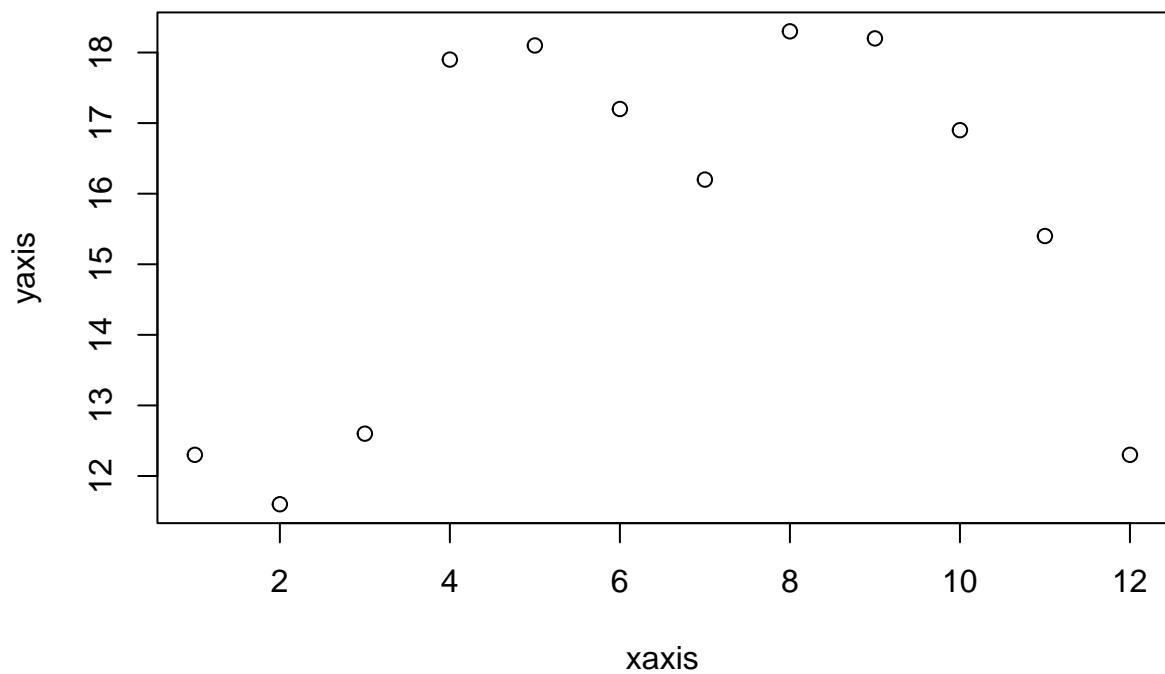


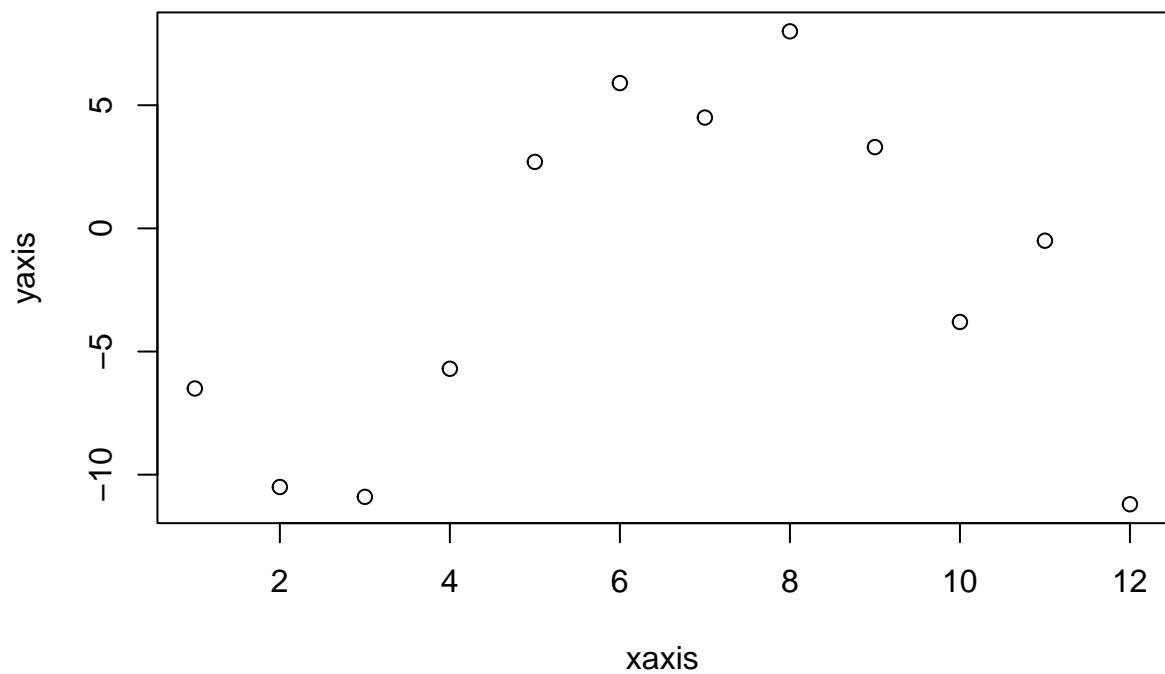


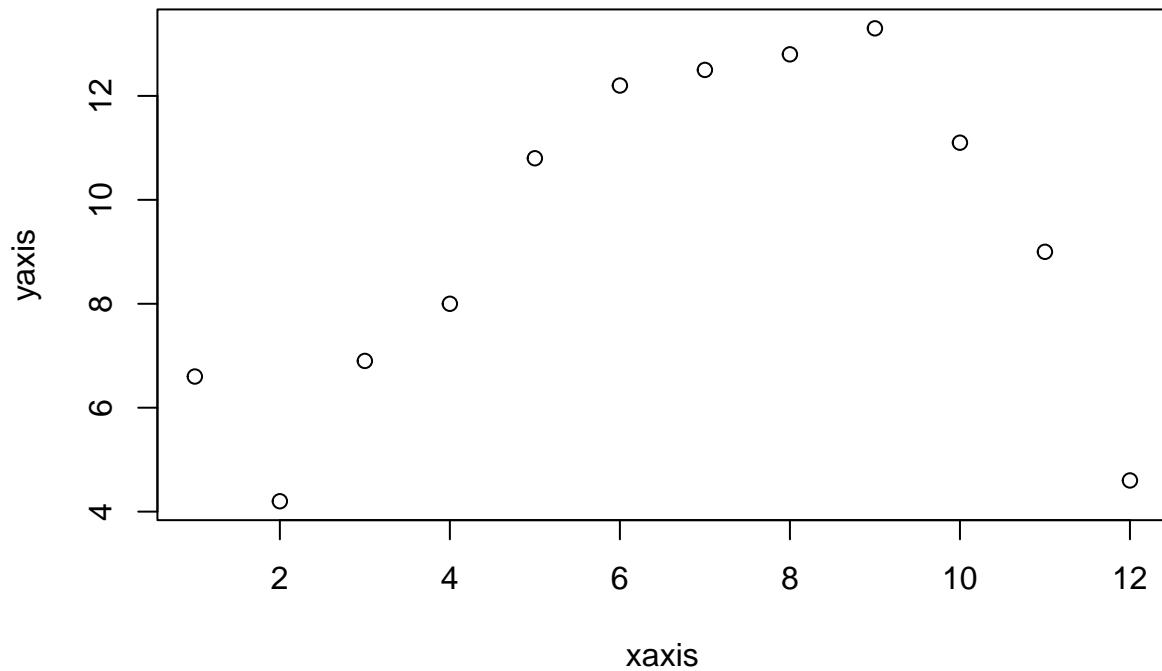




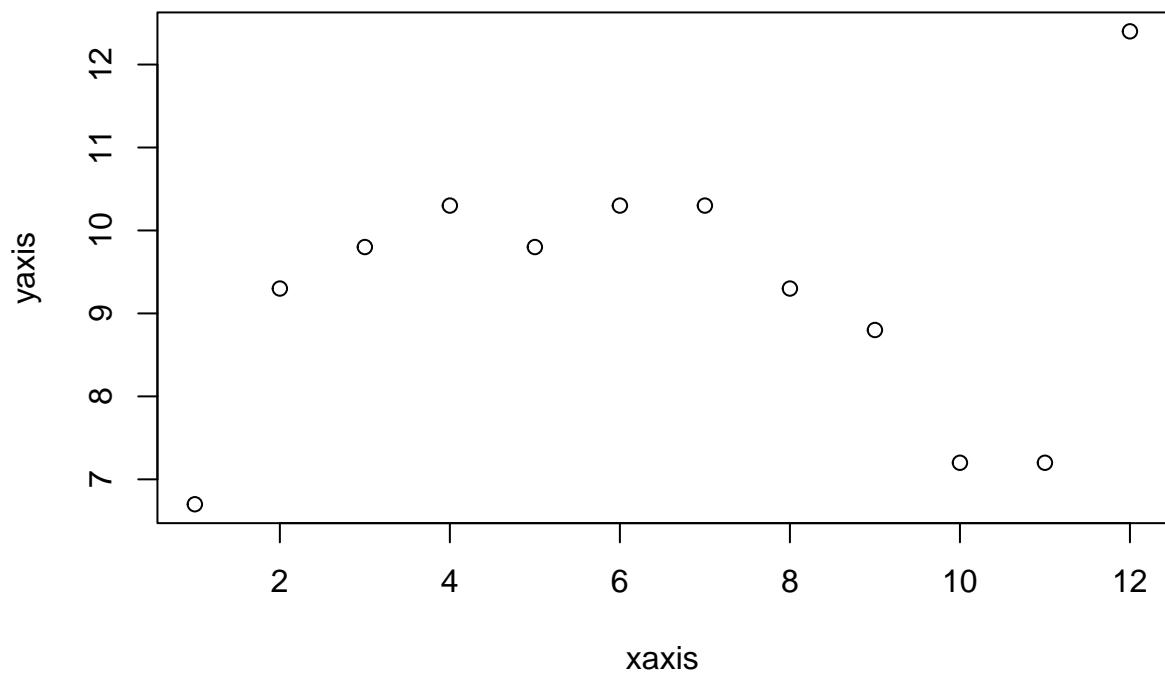
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] 11.6 18.3
## integer(0)
## [1] 12
## [1] "2023-01-18 18:00:00" "2023-12-24 16:00:00"
## integer(0)
## [1] 12
## [1] -11.2   8.0
## integer(0)
## [1] 12
## [1] "2023-01-31 12:00:00" "2024-01-01"
## integer(0)
## [1] 12
## [1]  4.2 13.3
```

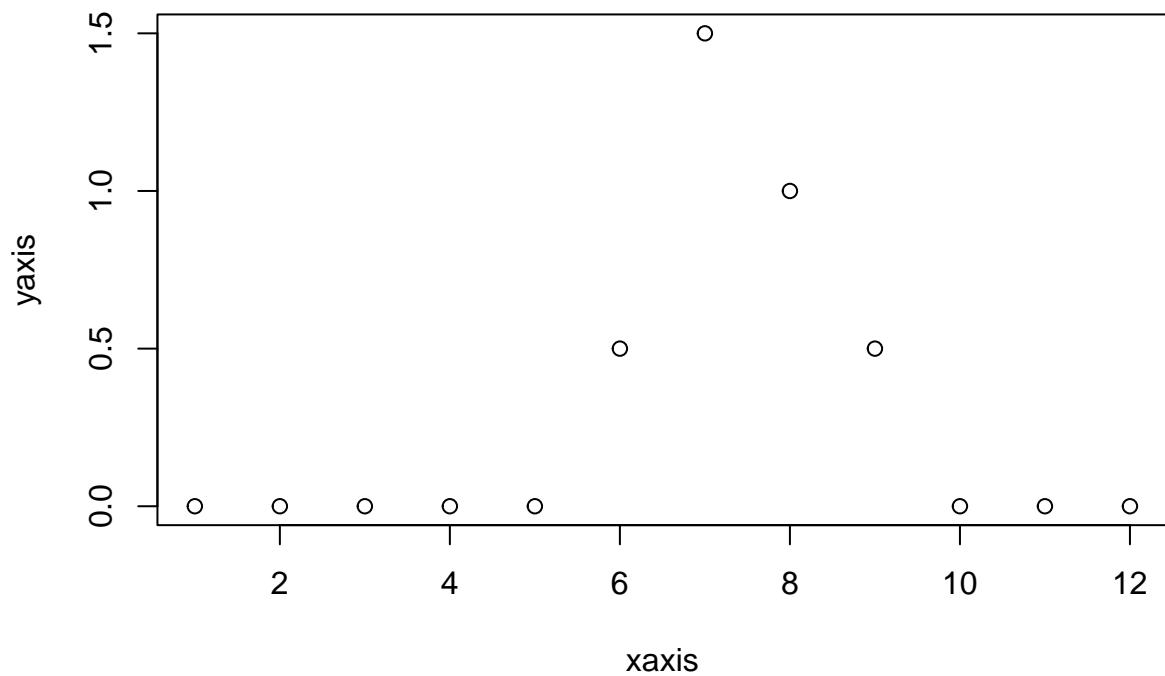


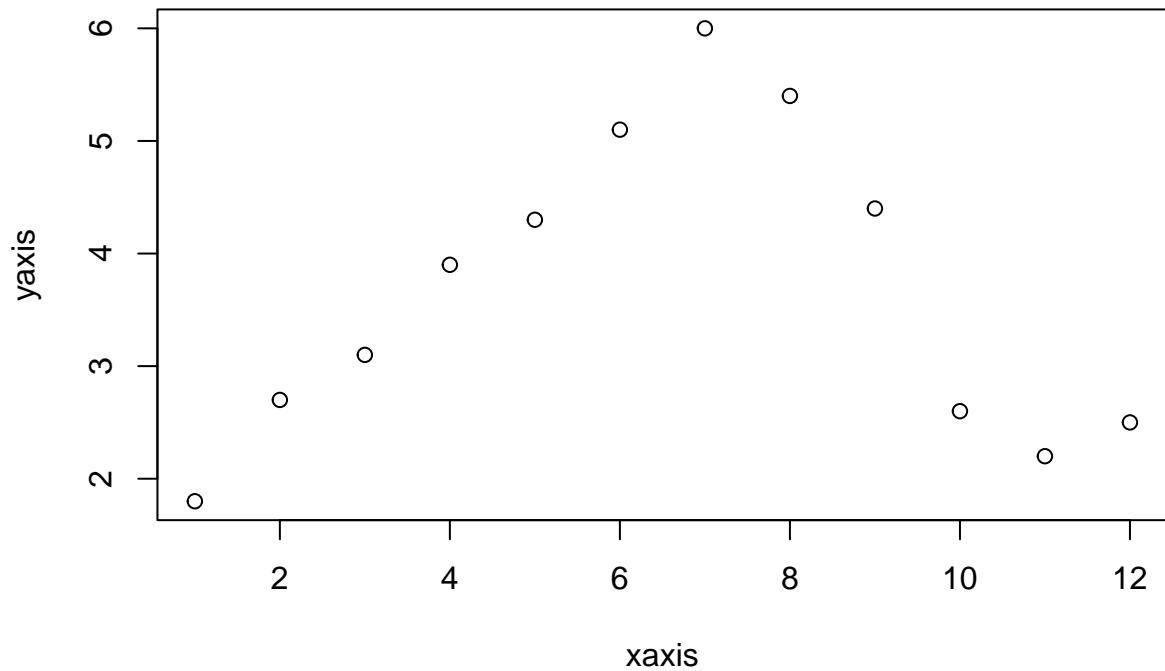




```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 7
## [1] 6.7 12.4
## integer(0)
## [1] 12
## [1] "2023-01-31 13:00:00" "2023-12-31 07:00:00"
## integer(0)
## [1] 4
## [1] 0.0 1.5
## integer(0)
## [1] 12
## [1] "2023-01-02 09:00:00" "2023-12-02 07:00:00"
## integer(0)
## [1] 12
## [1] 1.8 6.0
```







```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 5
## [1] " 0" " 7"
## integer(0)
## [1] 7
## [1] " 0" " 8"
## integer(0)
## [1] 6
## [1] " 0" " 7"
## integer(0)
## [1] 8
## [1] " 0" "12"
## integer(0)
## [1] 8
## [1] " 0" "21"
## integer(0)
## [1] 6
## [1] " 0" "11"
## integer(0)
## [1] 5
## [1] " 0" " 9"
## integer(0)
## [1] 7
## [1] " 0" " 6"
## integer(0)
```

```

## [1] 6
## [1] " 0" " 9"
## integer(0)
## [1] 5
## [1] " 0" " 5"
## integer(0)
## [1] 6
## [1] " 1" "10"
## integer(0)
## [1] 7
## [1] " 2" "11"
## integer(0)
## [1] 12
## [1] " 4" "62"
## integer(0)
## [1] 12
## [1] " 3" "42"
## integer(0)
## [1] 6
## [1] " 1" " 7"
## integer(0)
## [1] 5
## [1] " 0" " 4"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] " 7.1" "16.7"
## integer(0)
## [1] 12
## [1] " 6.9" "16.5"
## integer(0)
## [1] 12
## [1] " 6.8" "16.1"
## integer(0)
## [1] 12
## [1] " 6.4" "15.7"
## integer(0)
## [1] 12
## [1] " 6.5" "15.5"
## integer(0)
## [1] 12
## [1] " 6.3" "15.2"
## integer(0)
## [1] 12
## [1] " 6.1" "15.0"
## integer(0)
## [1] 12
## [1] " 6.0" "16.2"
## integer(0)
## [1] 11
## [1] " 6.3" "18.0"
## integer(0)
## [1] 12
## [1] " 7.4" "20.1"

```

```

## integer(0)
## [1] 12
## [1] "8.9" "22.2"
## integer(0)
## [1] 12
## [1] "10.6" "24.5"
## integer(0)
## [1] 12
## [1] "12.0" "26.5"
## integer(0)
## [1] 12
## [1] "13.1" "28.2"
## integer(0)
## [1] 11
## [1] "14.0" "29.3"
## integer(0)
## [1] 12
## [1] "14.0" "29.3"
## integer(0)
## [1] 12
## [1] "13.4" "28.4"
## integer(0)
## [1] 11
## [1] "10.9" "27.0"
## integer(0)
## [1] 12
## [1] "9.5" "24.8"
## integer(0)
## [1] 12
## [1] "8.9" "21.8"
## integer(0)
## [1] 11
## [1] "8.3" "19.4"
## integer(0)
## [1] 11
## [1] "7.8" "18.1"
## integer(0)
## [1] 12
## [1] "7.6" "17.5"
## integer(0)
## [1] 12
## [1] "7.1" "17.1"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] "10.3" "8.9"
## integer(0)
## [1] 12
## [1] "10.3" "8.8"
## integer(0)
## [1] 12
## [1] "10.1" "8.6"
## integer(0)
## [1] 12

```

```
## [1] "10.1" "8.5"
## integer(0)
## [1] 12
## [1] "10.1" "8.3"
## integer(0)
## [1] 12
## [1] "10.0" "8.4"
## integer(0)
## [1] 12
## [1] "10.3" "8.3"
## integer(0)
## [1] 12
## [1] "10.5" "8.1"
## integer(0)
## [1] 12
## [1] "11.1" "8.4"
## integer(0)
## [1] 12
## [1] "10.9" "9.4"
## integer(0)
## [1] 12
## [1] "10.8" "9.6"
## integer(0)
## [1] 11
## [1] "11.0" "9.7"
## integer(0)
## [1] 12
## [1] "11.1" "9.5"
## integer(0)
## [1] 11
## [1] "11.1" "9.4"
## integer(0)
## [1] 12
## [1] "10.6" "9.3"
## integer(0)
## [1] 12
## [1] "10.9" "9.1"
## integer(0)
## [1] 11
## [1] "10.8" "9.1"
## integer(0)
## [1] 12
## [1] "11.2" "9.6"
## integer(0)
## [1] 12
## [1] "11.4" "9.5"
## integer(0)
## [1] 12
## [1] "11.2" "9.5"
## integer(0)
## [1] 12
## [1] "10.7" "9.5"
## integer(0)
## [1] 11
```

```

## [1] "10.5" "9.4"
## integer(0)
## [1] 11
## [1] "10.5" "9.2"
## integer(0)
## [1] 12
## [1] "10.4" "8.9"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] "79" "90"
## integer(0)
## [1] 8
## [1] "81" "92"
## integer(0)
## [1] 7
## [1] "82" "93"
## integer(0)
## [1] 7
## [1] "84" "93"
## integer(0)
## [1] 7
## [1] "84" "93"
## integer(0)
## [1] 7
## [1] "85" "93"
## integer(0)
## [1] 8
## [1] "83" "94"
## integer(0)
## [1] 8
## [1] "78" "94"
## integer(0)
## [1] 12
## [1] "69" "93"
## integer(0)
## [1] 11
## [1] "61" "92"
## integer(0)
## [1] 9
## [1] "56" "89"
## integer(0)
## [1] 10
## [1] "50" "84"
## integer(0)
## [1] 9
## [1] "45" "78"
## integer(0)
## [1] 9
## [1] "42" "73"
## integer(0)
## [1] 10
## [1] "40" "69"
## integer(0)

```

```

## [1] 10
## [1] "40" "67"
## integer(0)
## [1] 10
## [1] "41" "66"
## integer(0)
## [1] 10
## [1] "44" "71"
## integer(0)
## [1] 10
## [1] "48" "76"
## integer(0)
## [1] 10
## [1] "54" "82"
## integer(0)
## [1] 11
## [1] "63" "86"
## integer(0)
## [1] 9
## [1] "68" "89"
## integer(0)
## [1] 10
## [1] "73" "91"
## integer(0)
## [1] 8
## [1] "77" "91"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 2
## [1] " 0" " 65"
## integer(0)
## [1] 7
## [1] " 0" "365"
## integer(0)
## [1] 11
## [1] " 0" "526"
## integer(0)
## [1] 12
## [1] "137" "657"
## integer(0)
## [1] 12
## [1] "327" "758"

```

```

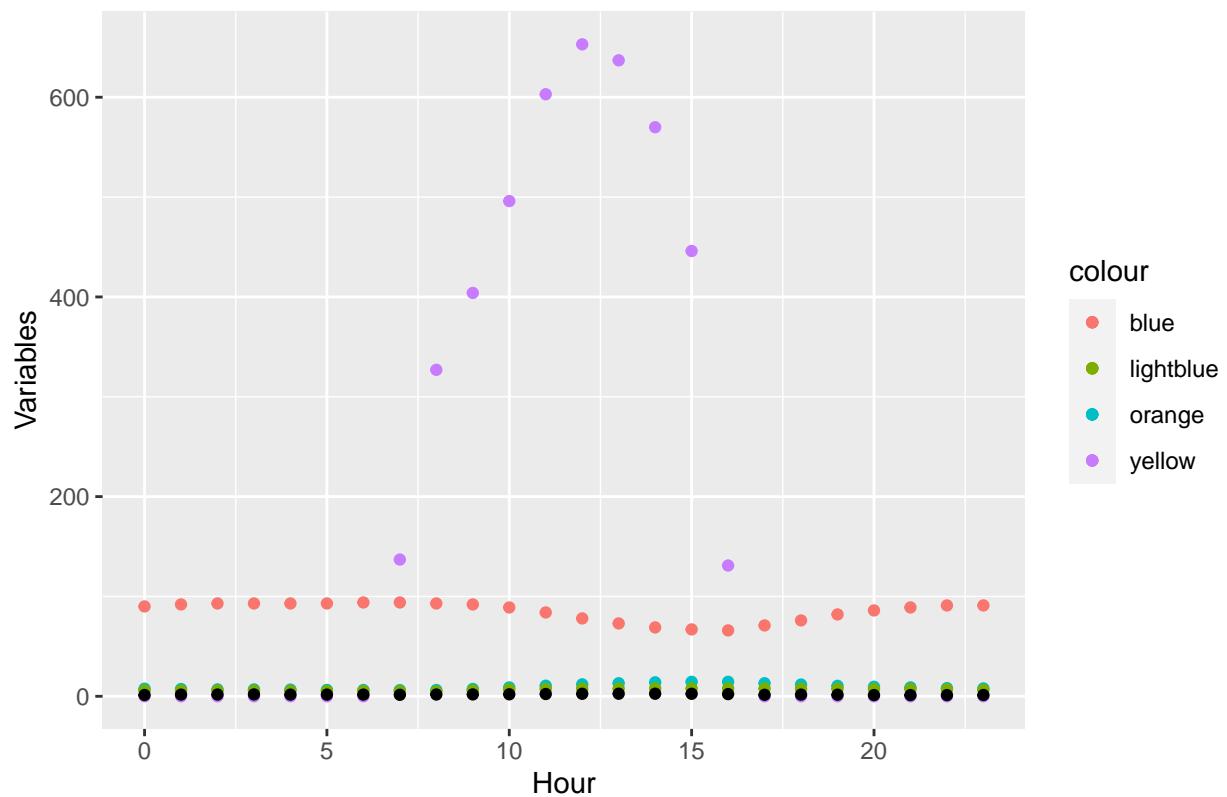
## integer(0)
## [1] 12
## [1] "404" "832"
## integer(0)
## [1] 12
## [1] "474" "885"
## integer(0)
## [1] 12
## [1] "517" "899"
## integer(0)
## [1] 12
## [1] "511" "894"
## integer(0)
## [1] 12
## [1] "479" "878"
## integer(0)
## [1] 12
## [1] "402" "851"
## integer(0)
## [1] 12
## [1] "226" "801"
## integer(0)
## [1] 12
## [1] " 0" "715"
## integer(0)
## [1] 8
## [1] " 0" "557"
## integer(0)
## [1] 5
## [1] " 0" "228"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] "1.3" "5.8"
## integer(0)
## [1] 11
## [1] "1.7" "5.9"
## integer(0)
## [1] 10

```

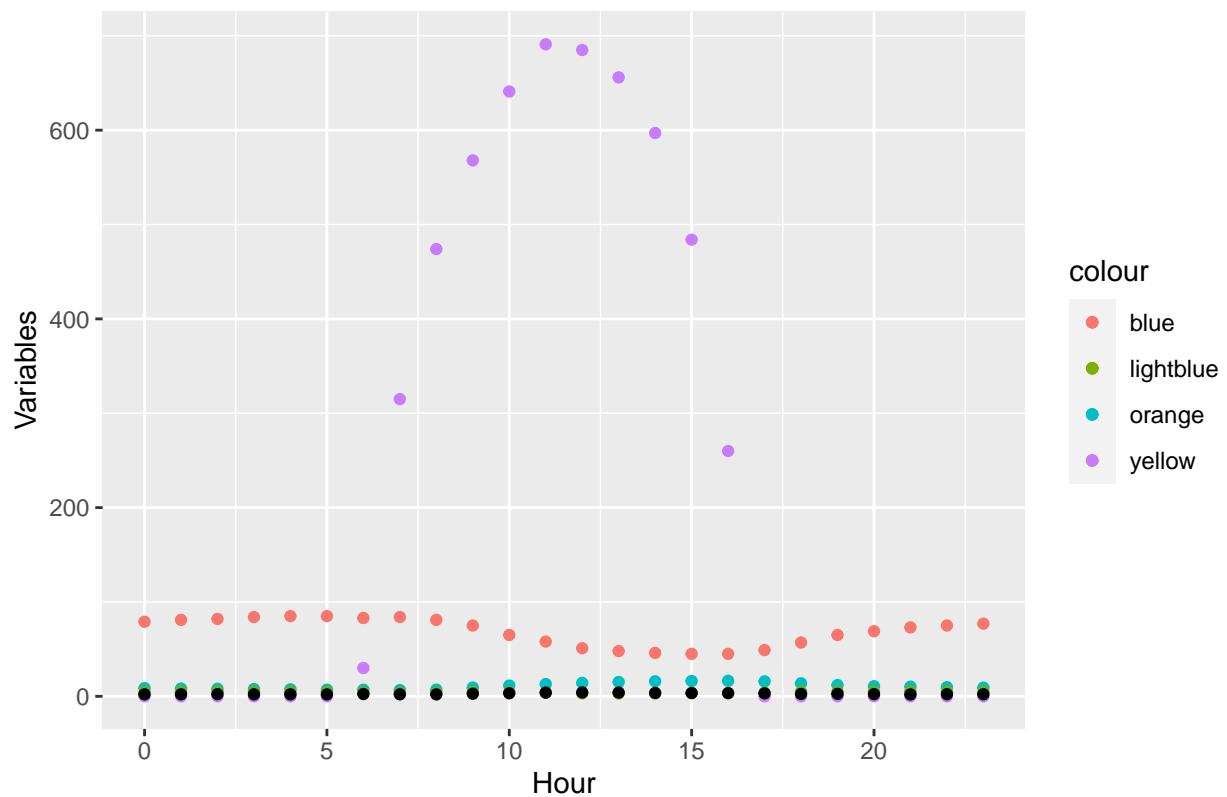
```
## [1] "1.6" "5.6"
## integer(0)
## [1] 12
## [1] "1.6" "5.6"
## integer(0)
## [1] 10
## [1] "1.6" "5.9"
## integer(0)
## [1] 11
## [1] "1.4" "5.1"
## integer(0)
## [1] 12
## [1] "1.4" "5.6"
## integer(0)
## [1] 12
## [1] "1.4" "5.7"
## integer(0)
## [1] 11
## [1] "1.7" "5.9"
## integer(0)
## [1] 11
## [1] "1.9" "5.8"
## integer(0)
## [1] 11
## [1] "2.0" "5.8"
## integer(0)
## [1] 11
## [1] "2.3" "5.8"
## integer(0)
## [1] 10
## [1] "2.5" "6.0"
## integer(0)
## [1] 11
## [1] "2.6" "6.4"
## integer(0)
## [1] 12
## [1] "2.6" "6.7"
## integer(0)
## [1] 11
## [1] "2.5" "7.2"
## integer(0)
## [1] 12
## [1] "2.2" "7.4"
## integer(0)
## [1] 11
## [1] "1.5" "7.1"
## integer(0)
## [1] 10
## [1] "1.7" "6.8"
## integer(0)
## [1] 12
## [1] "1.6" "6.0"
## integer(0)
## [1] 10
```

```
## [1] "1.2" "5.5"
## integer(0)
## [1] 12
## [1] "1.2" "5.7"
## integer(0)
## [1] 11
## [1] "1.2" "5.7"
## integer(0)
## [1] 10
## [1] "1.3" "5.6"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "Apr" "Sep"
## integer(0)
## [1] 24
## [1] 0 23
## integer(0)
## [1] 157
## [1] 6.0 29.3
## integer(0)
## [1] 94
## [1] 3.0 14.1
## integer(0)
## [1] 54
## [1] 40 94
## integer(0)
## [1] 140
## [1] 0 899
## integer(0)
## [1] 59
## [1] 1.2 7.4
```

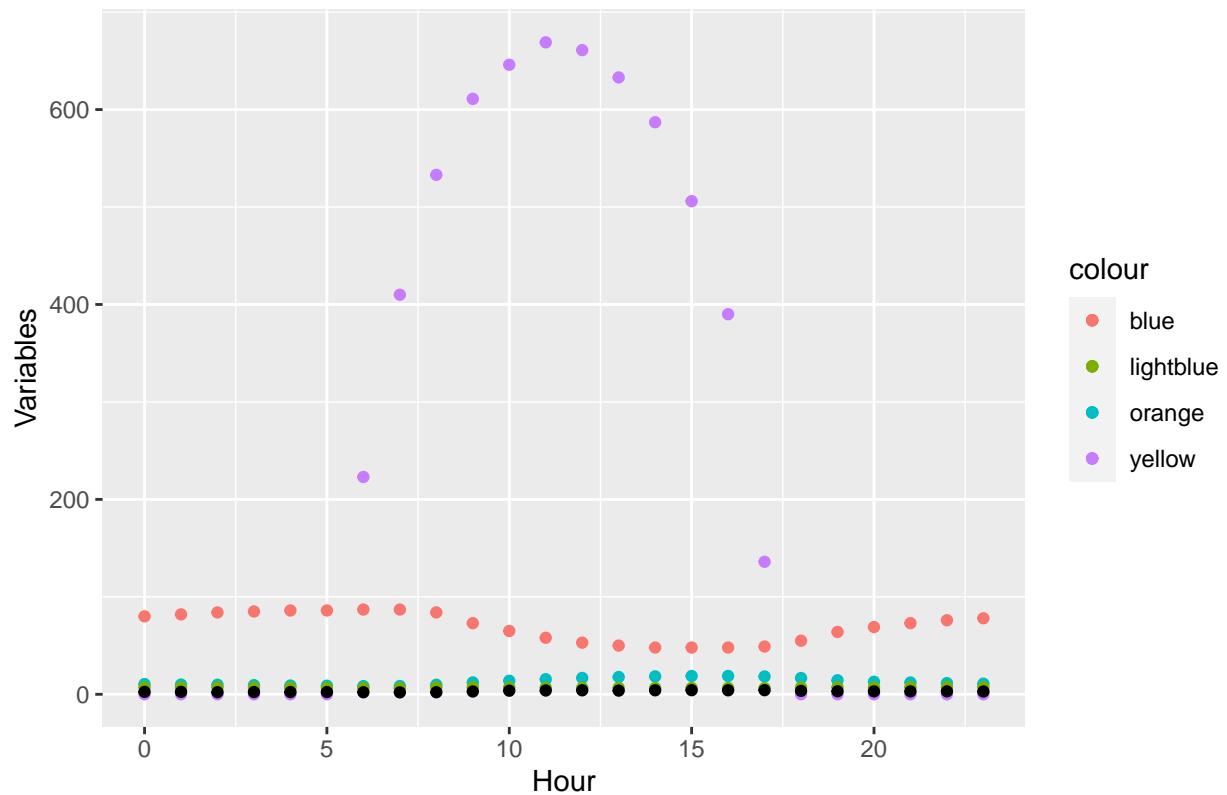
Jan Data: USA\_CA\_Fairfield-San.Francisco.Bay.Reserve



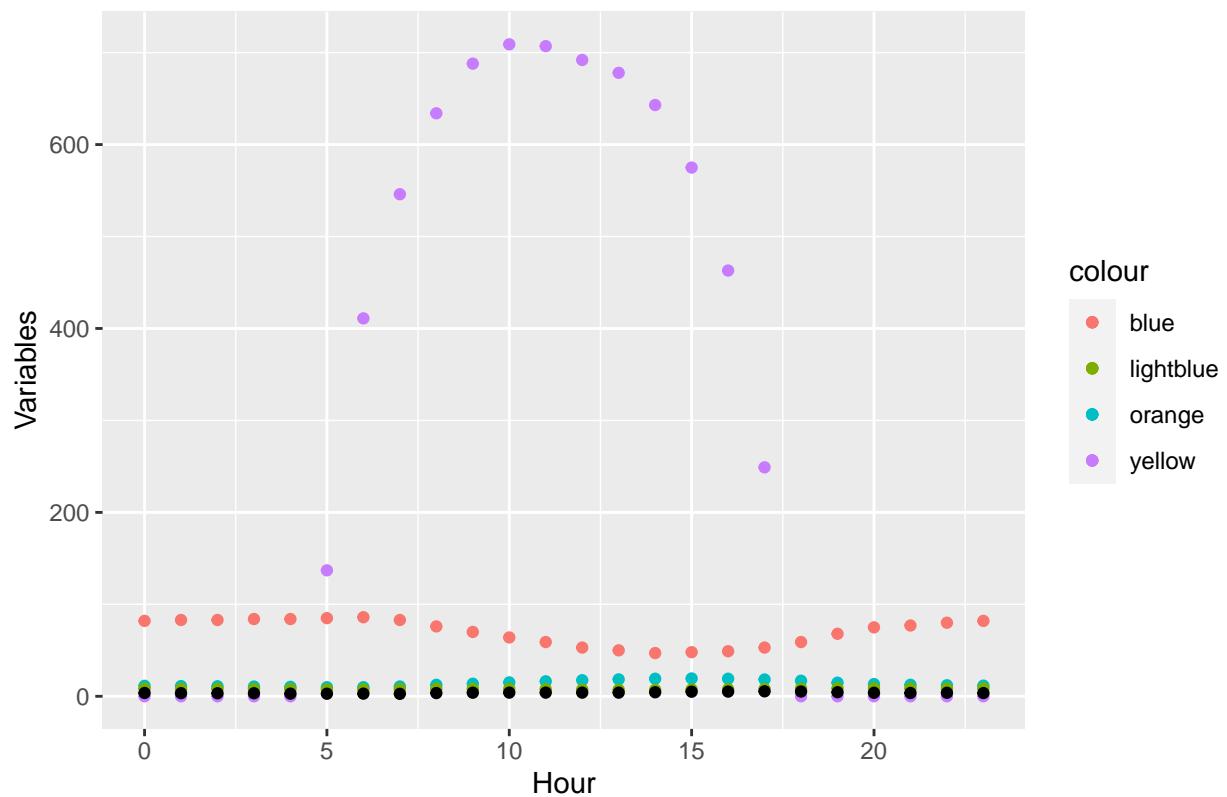
Feb Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



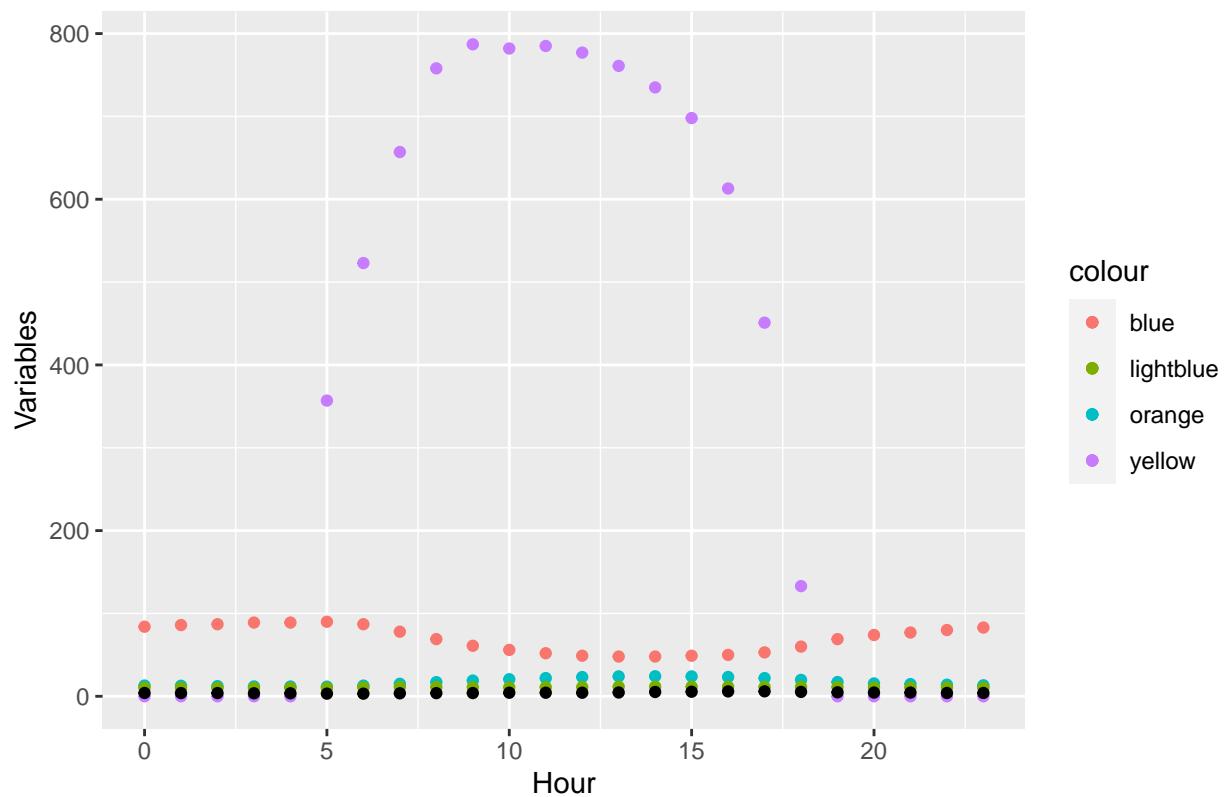
Mar Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



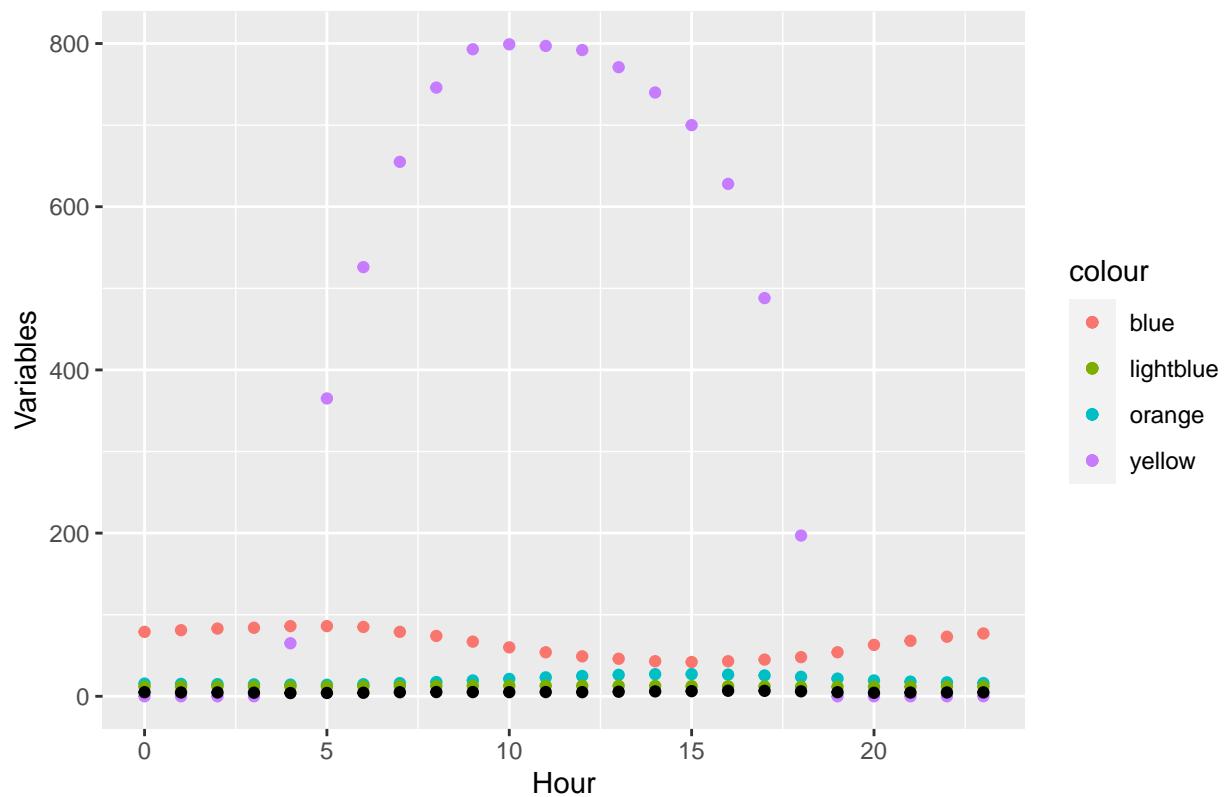
### Apr Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



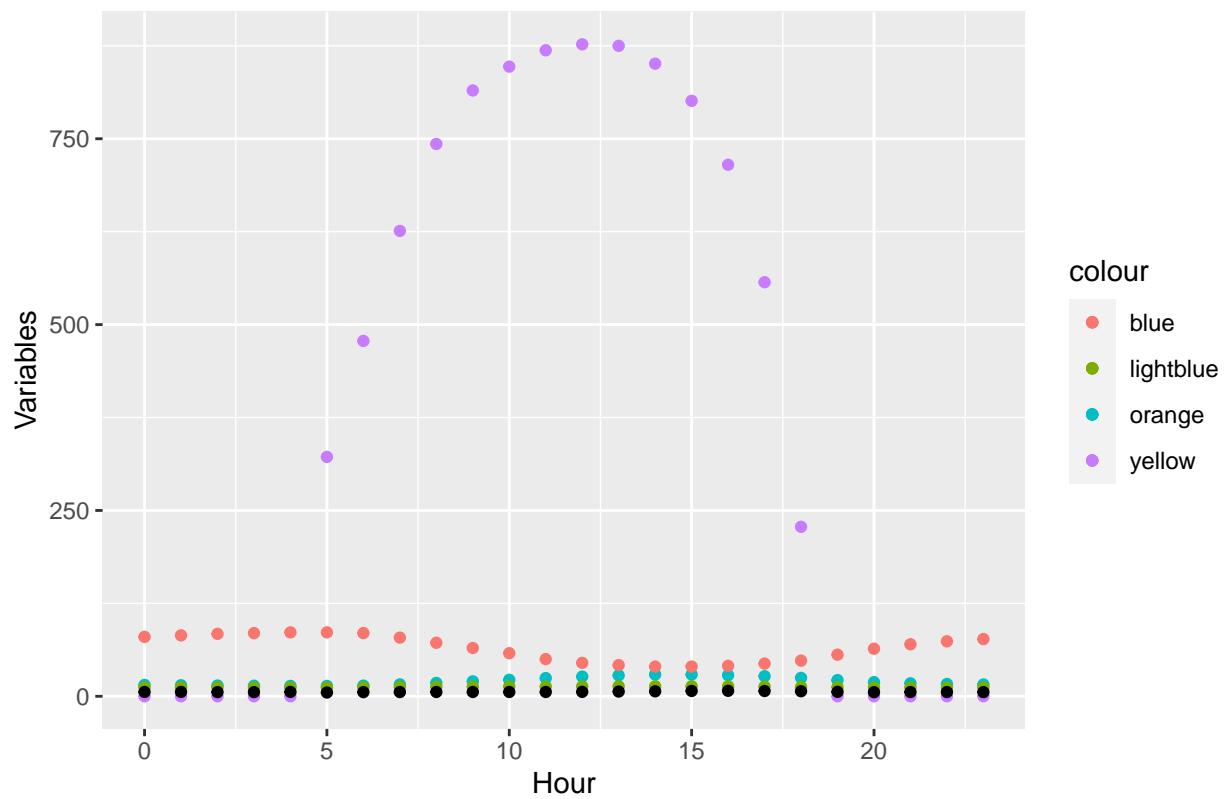
### May Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



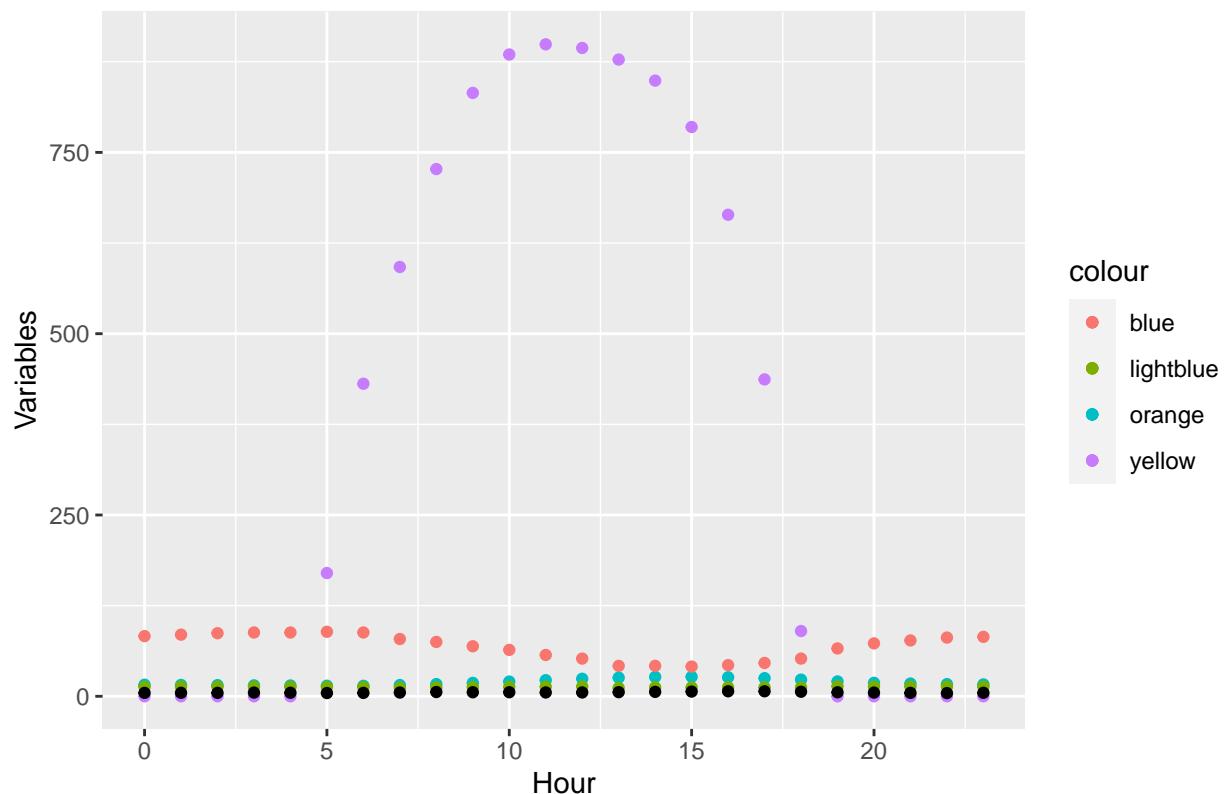
### Jun Data: USA\_CA\_Fairfield-San.Francisco.Bay.Reserve



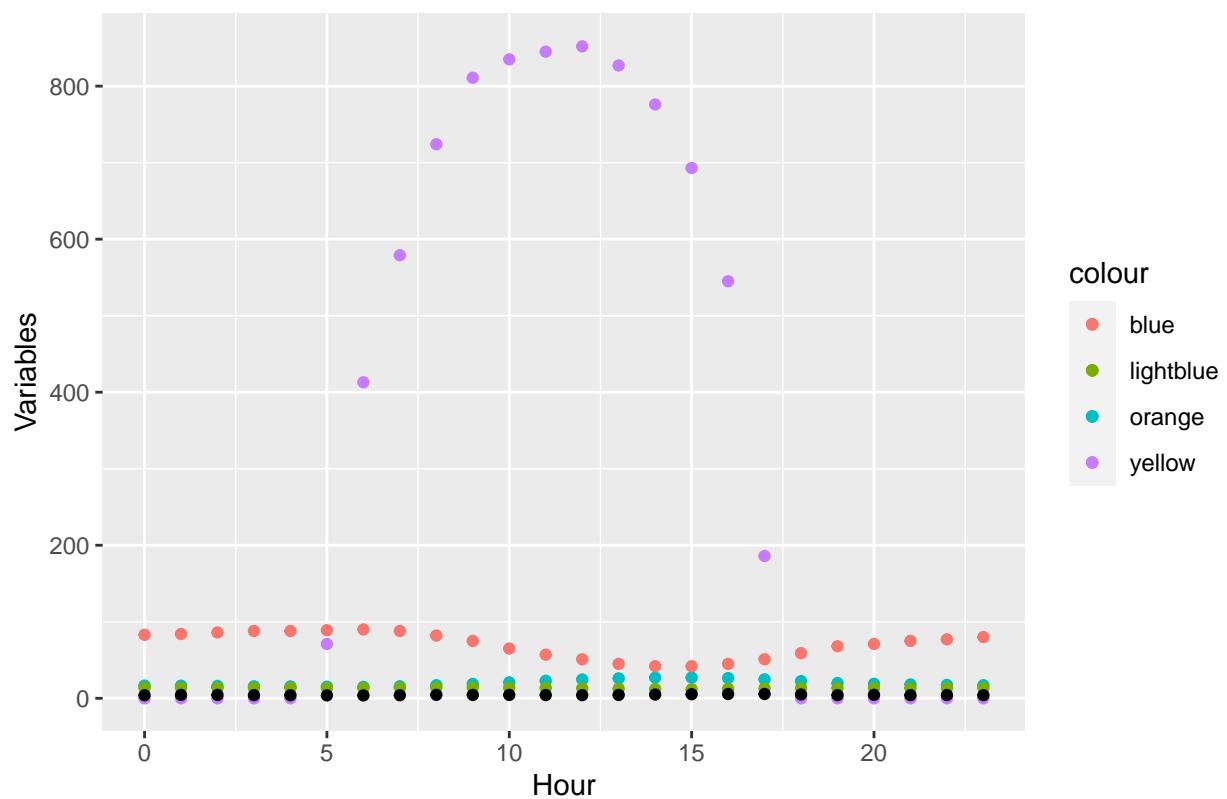
Jul Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



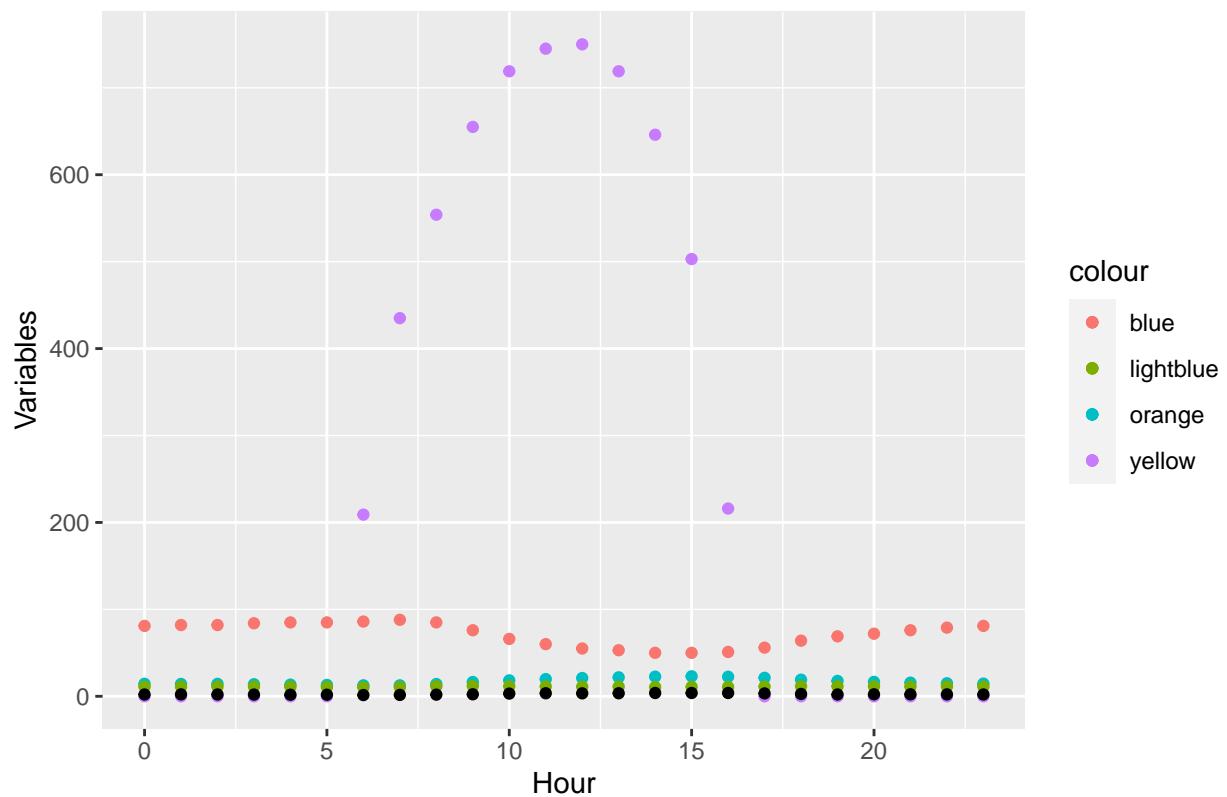
### Aug Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



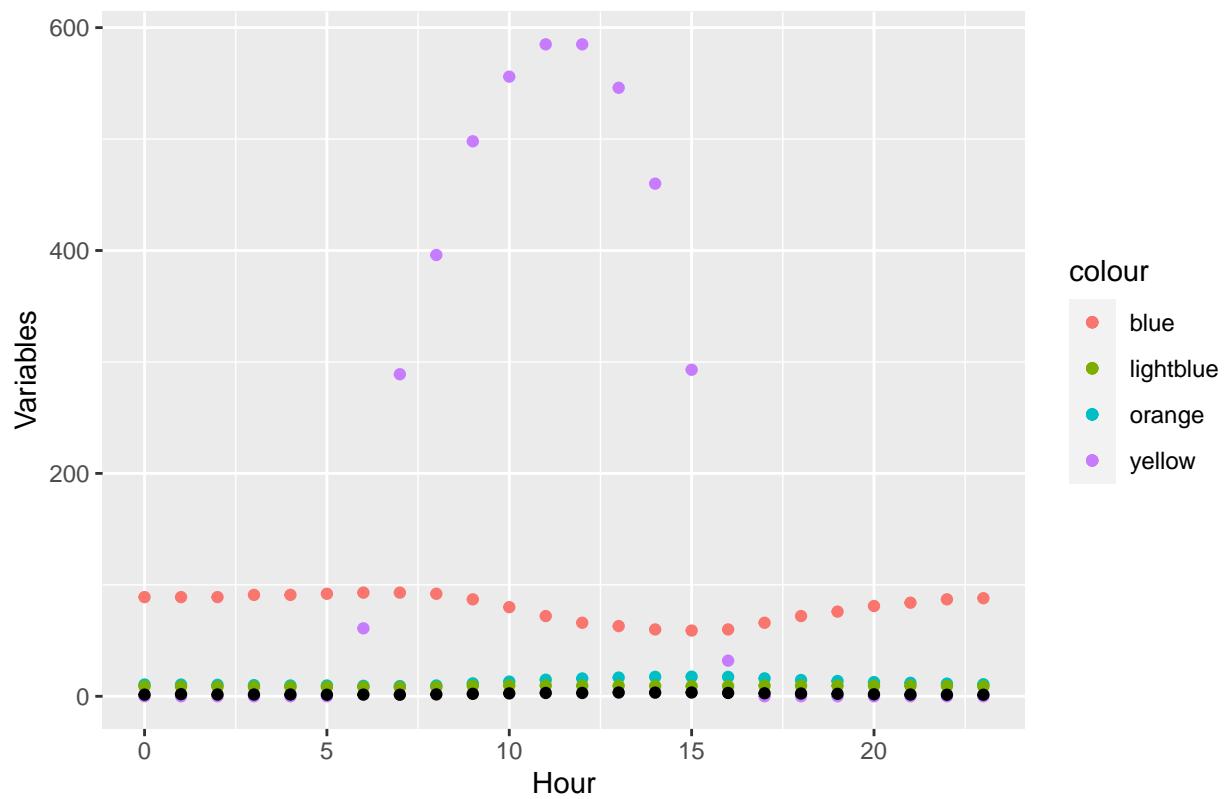
### Sep Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



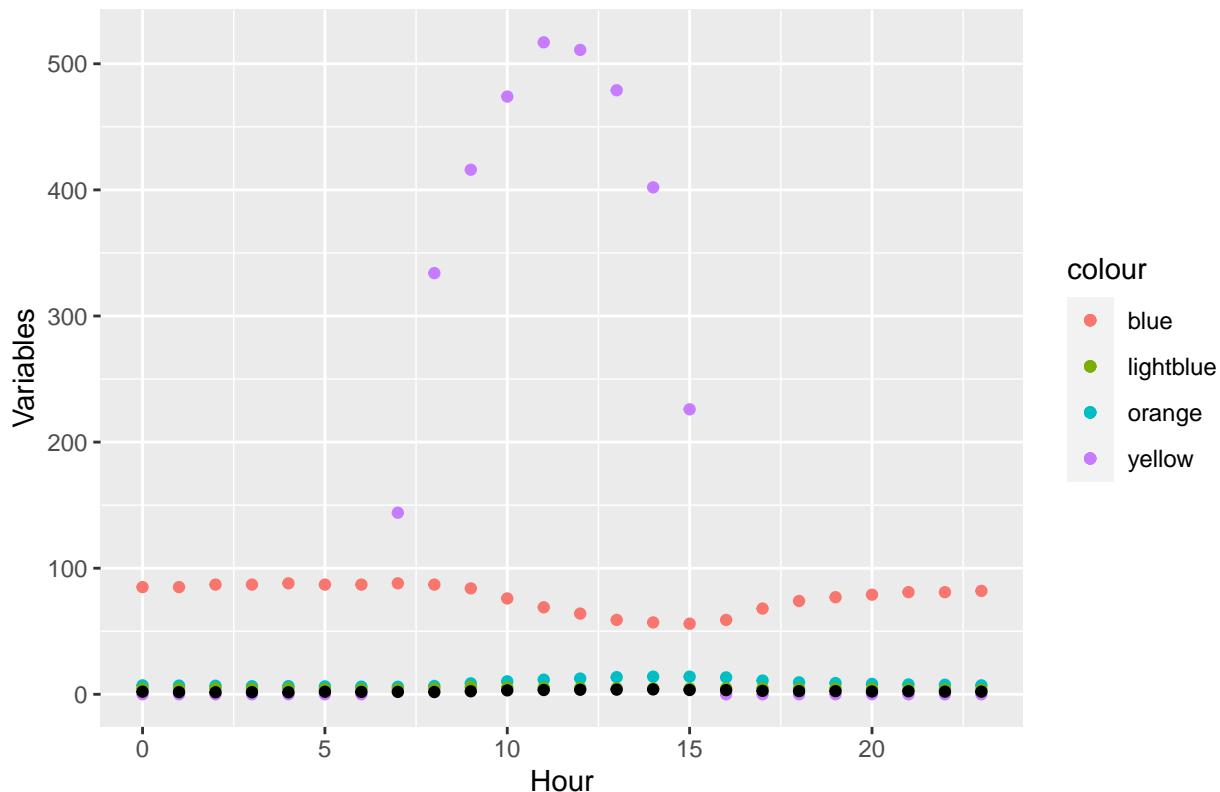
### Oct Data: USA\_CA\_Fairfield-San.Francisco.Bay.Reserve



### Nov Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve



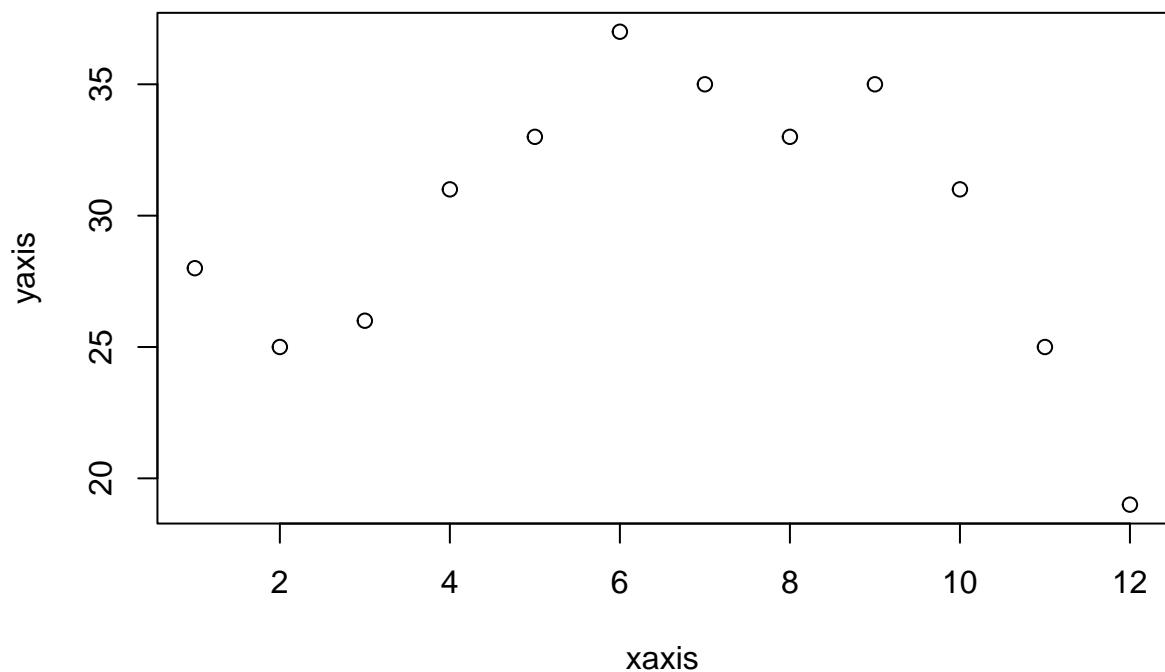
## Dec Data: USA\_CA\_Fairfield–San.Francisco.Bay.Reserve

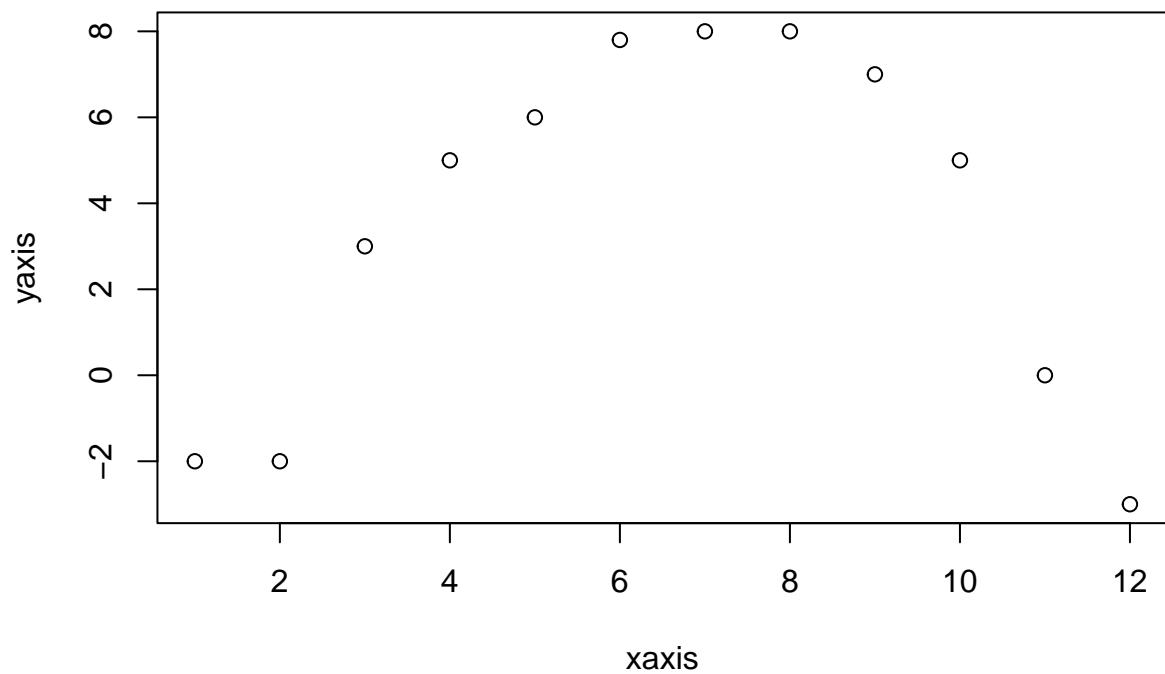


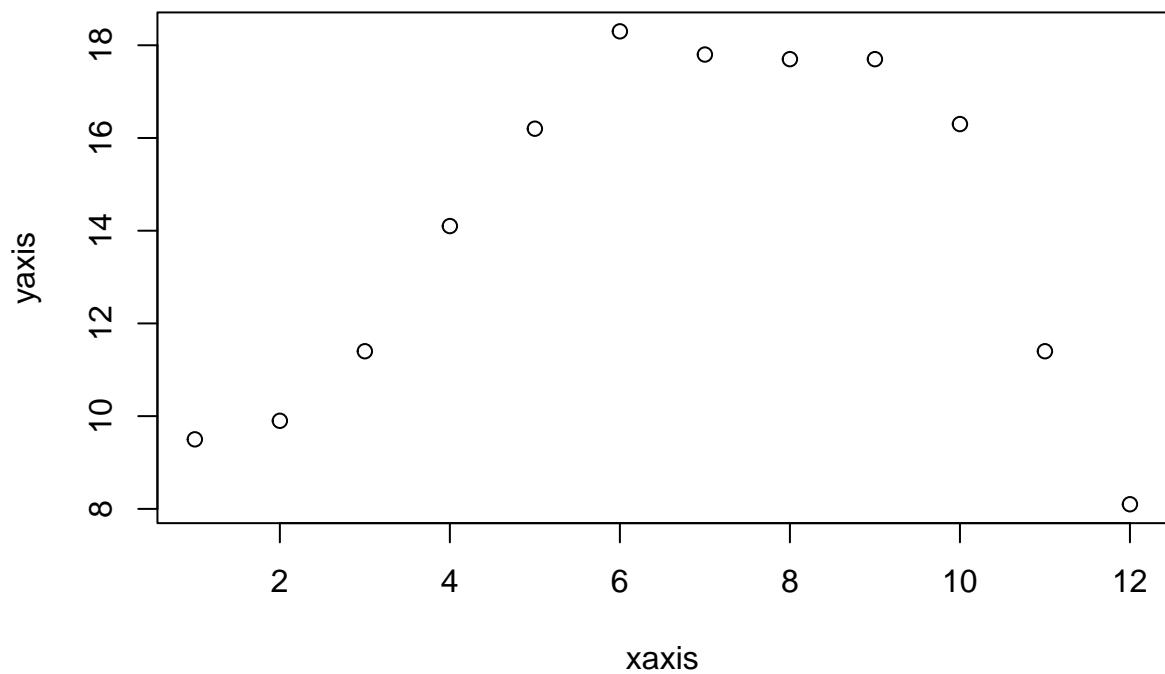
```
stat_marin = stat_parser('dt/USA_CA_Marin.County.AP-Gnoss.Field.720406_TMYx.2007-2021.stat')
```

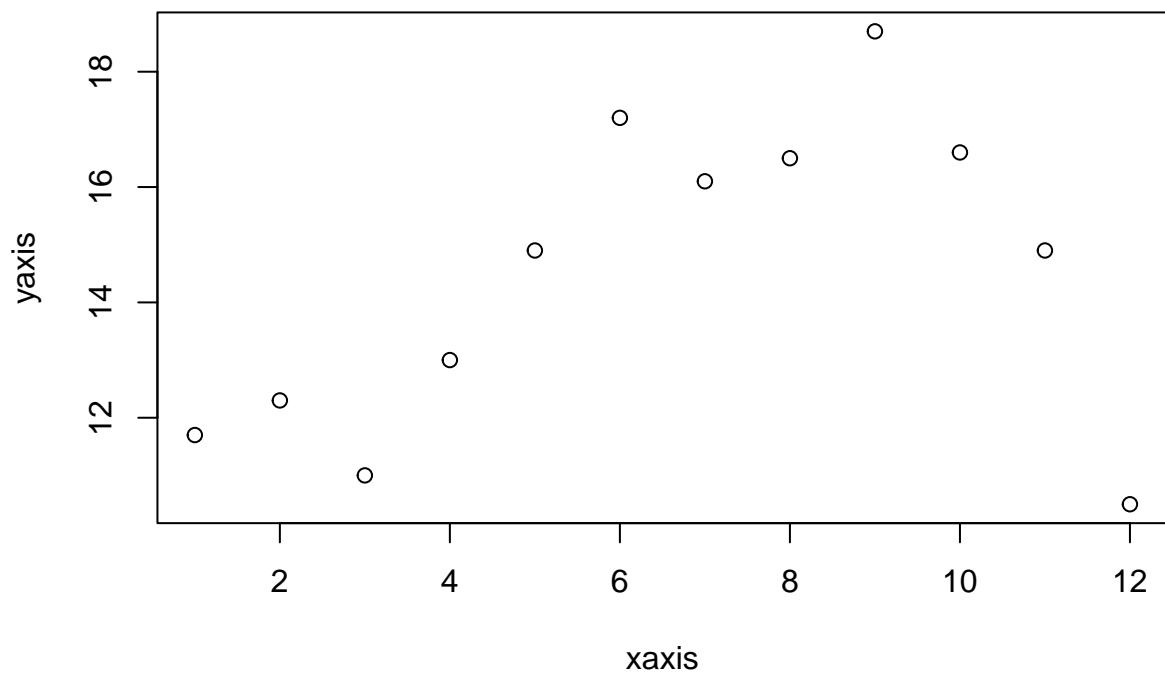
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] 19 37
## integer(0)
## [1] 12
## [1] "2023-01-18 14:00:00" "2023-12-03 14:00:00"
## integer(0)
## [1] 9
## [1] -3 8
## integer(0)
## [1] 12
## [1] "2023-01-21 08:00:00" "2023-12-19 05:00:00"
## integer(0)
## [1] 10
## [1] 8.1 18.3
## integer(0)
## [1] 11
## [1] 10.5 18.7
## integer(0)
## [1] 8
## [1] 19 37
## integer(0)
```

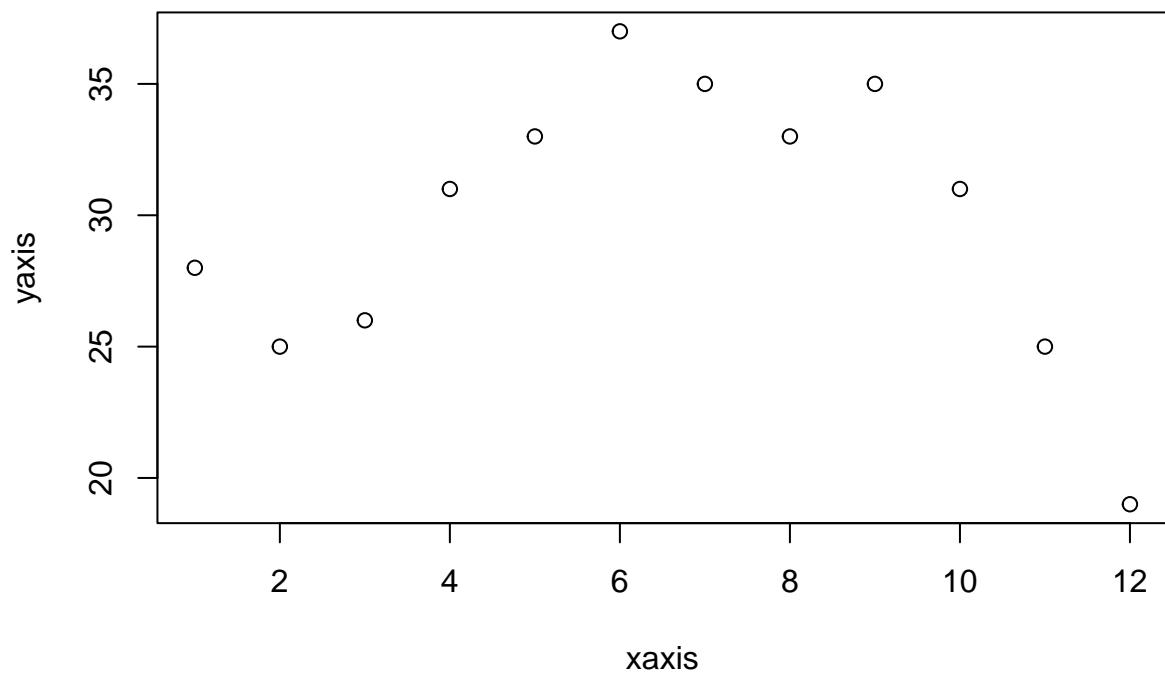
```
## [1] 7
## [1] -2 10
## integer(0)
## [1] 11
## [1] 9.8 22.4
## integer(0)
## [1] 9
## [1] 16 30
## integer(0)
## [1] 9
## [1] -3 8
## integer(0)
## [1] 11
## [1] 6.5 14.4
```

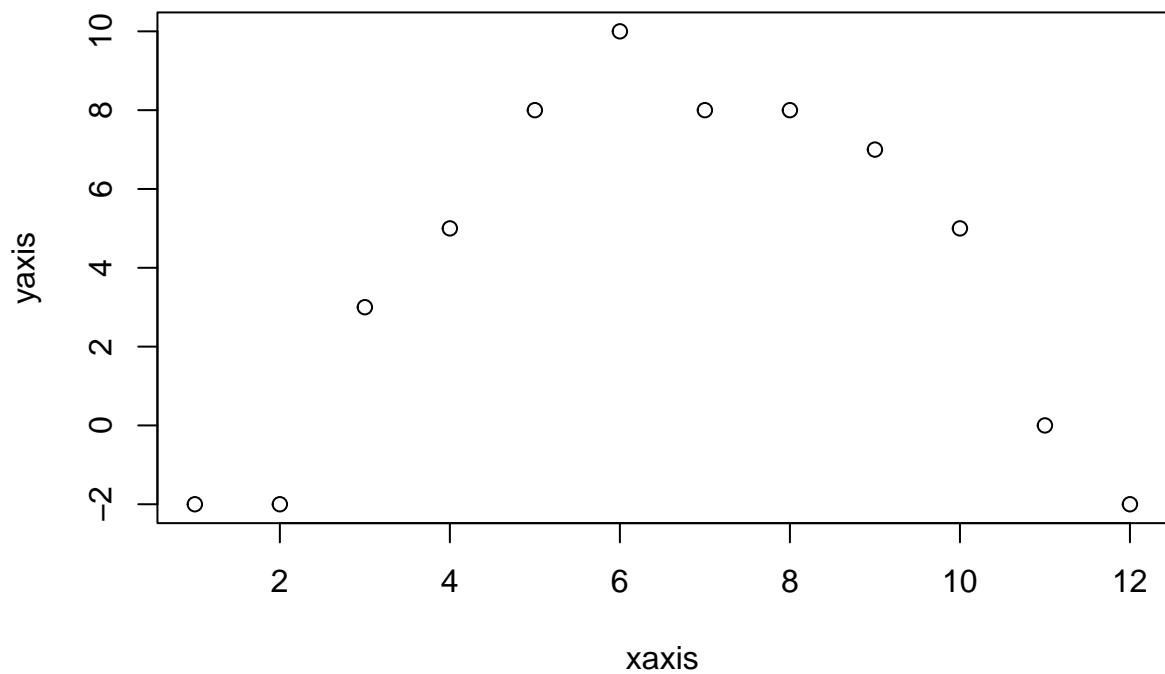


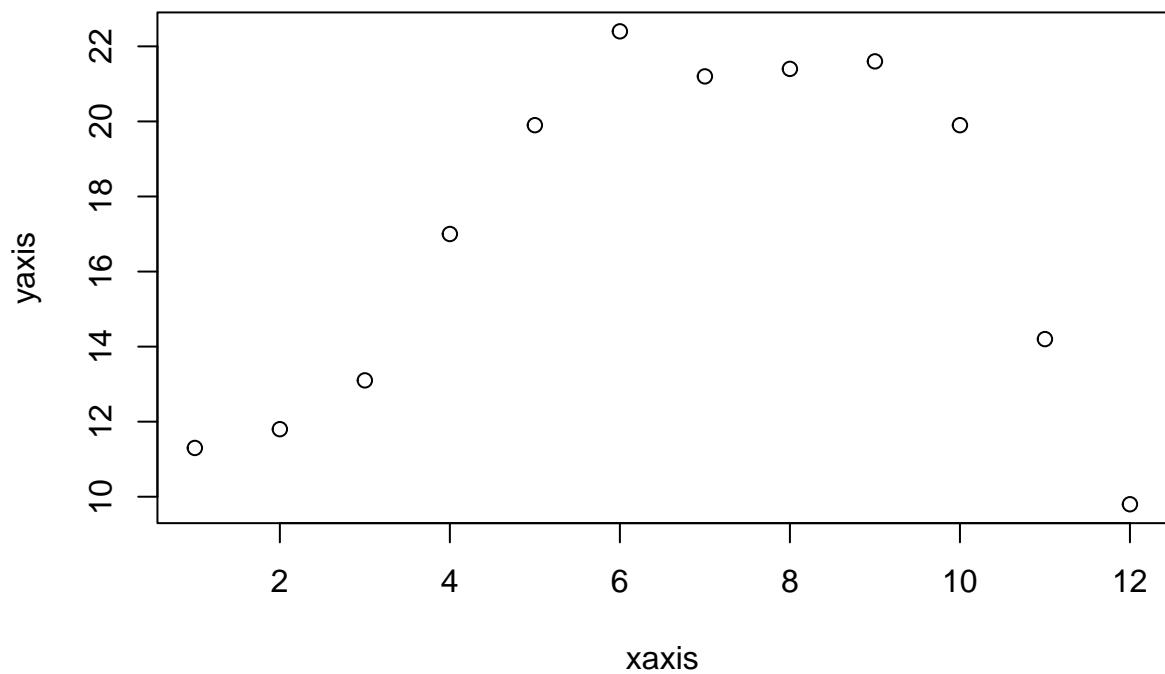


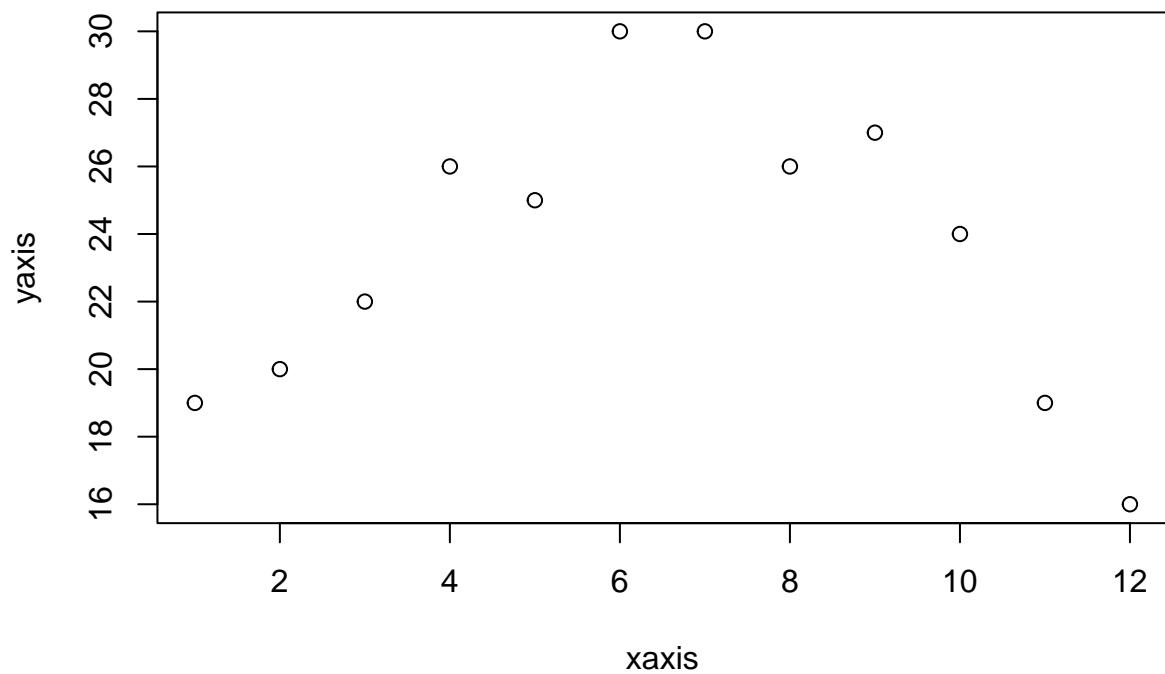


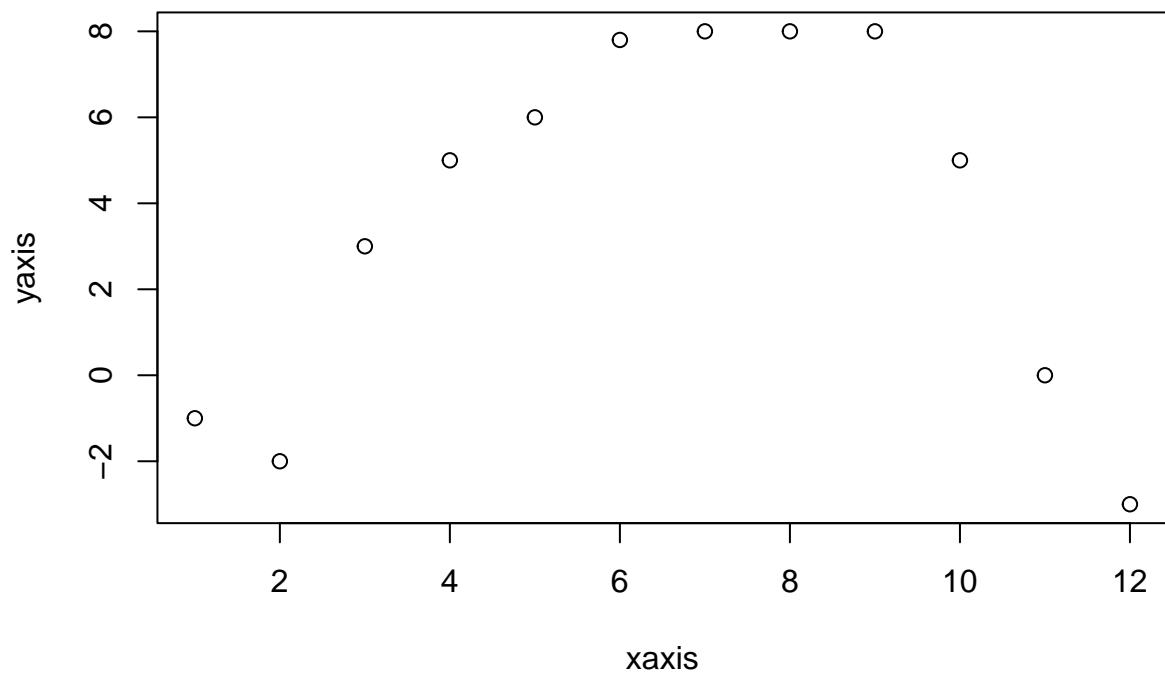


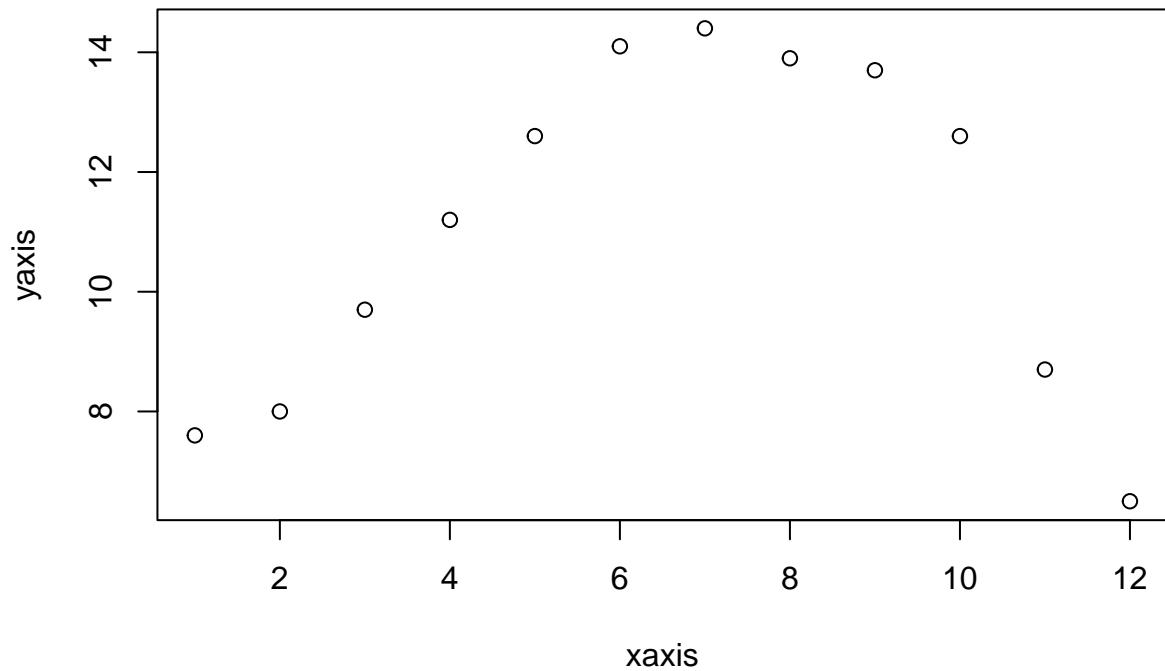




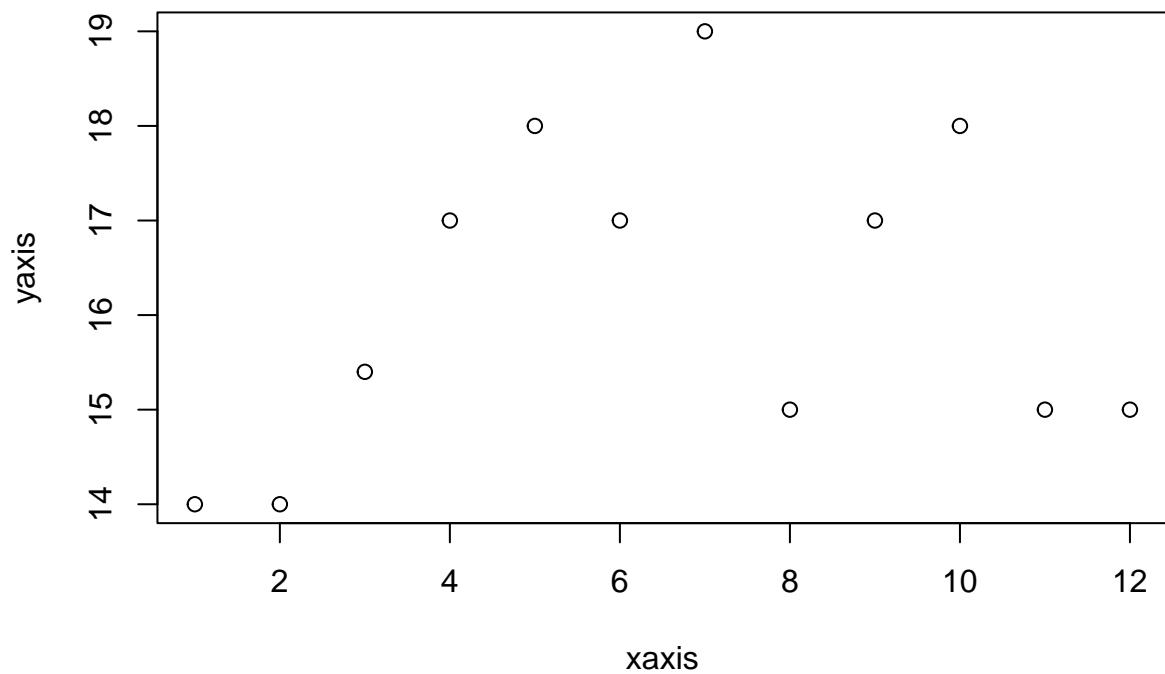


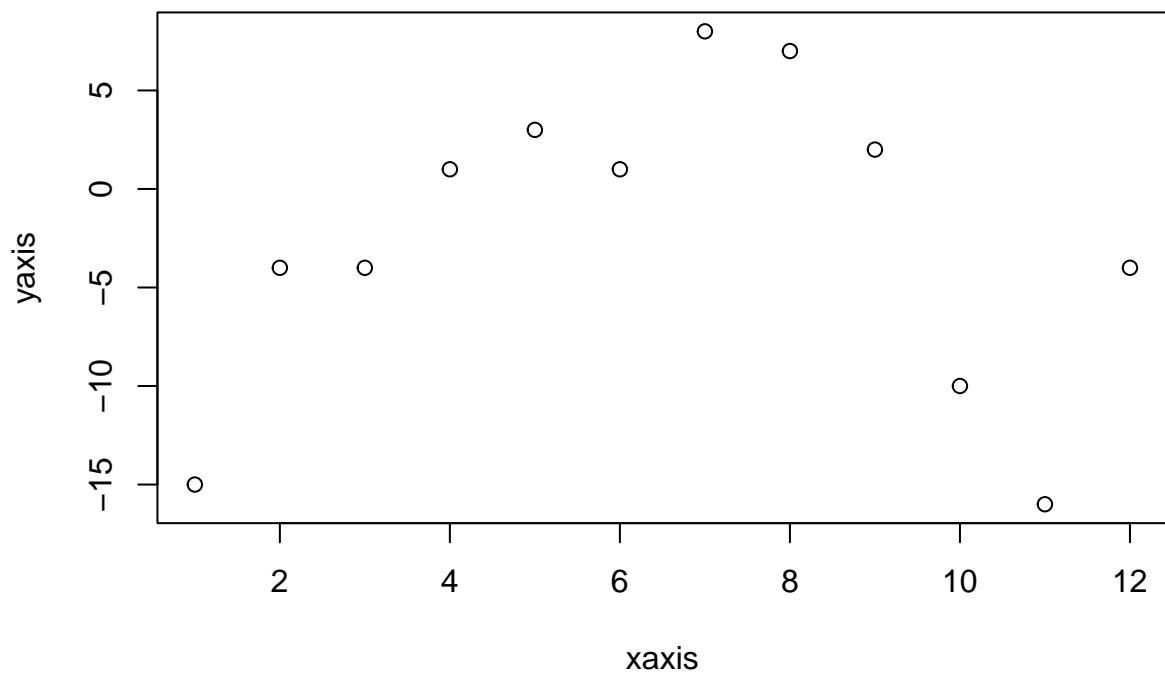


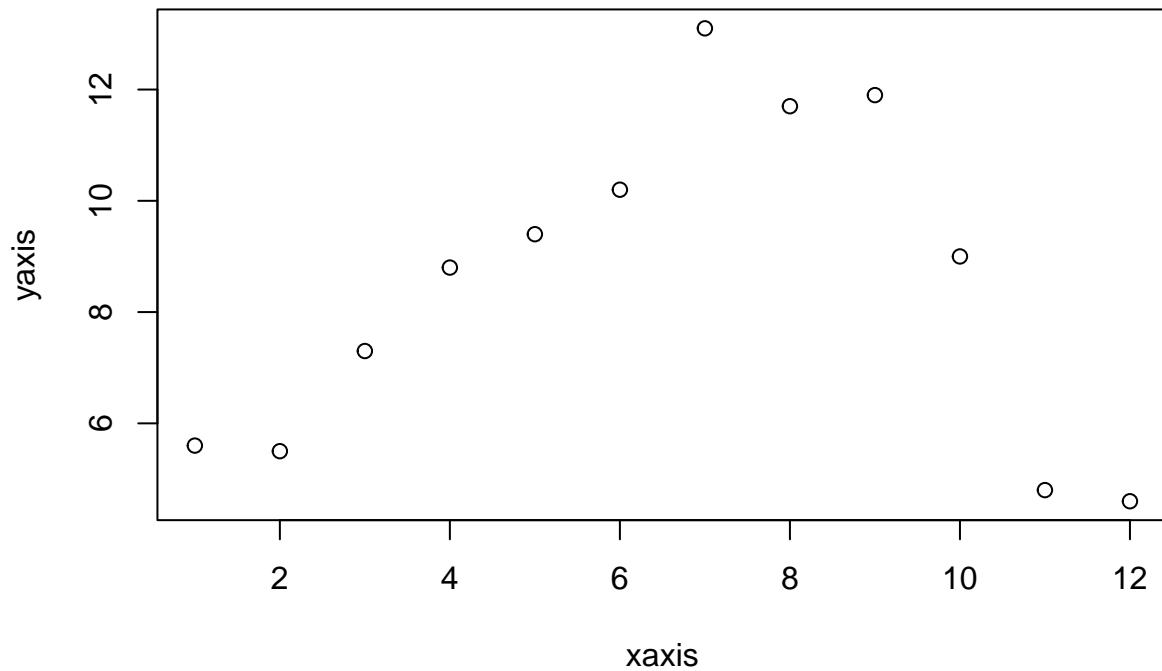




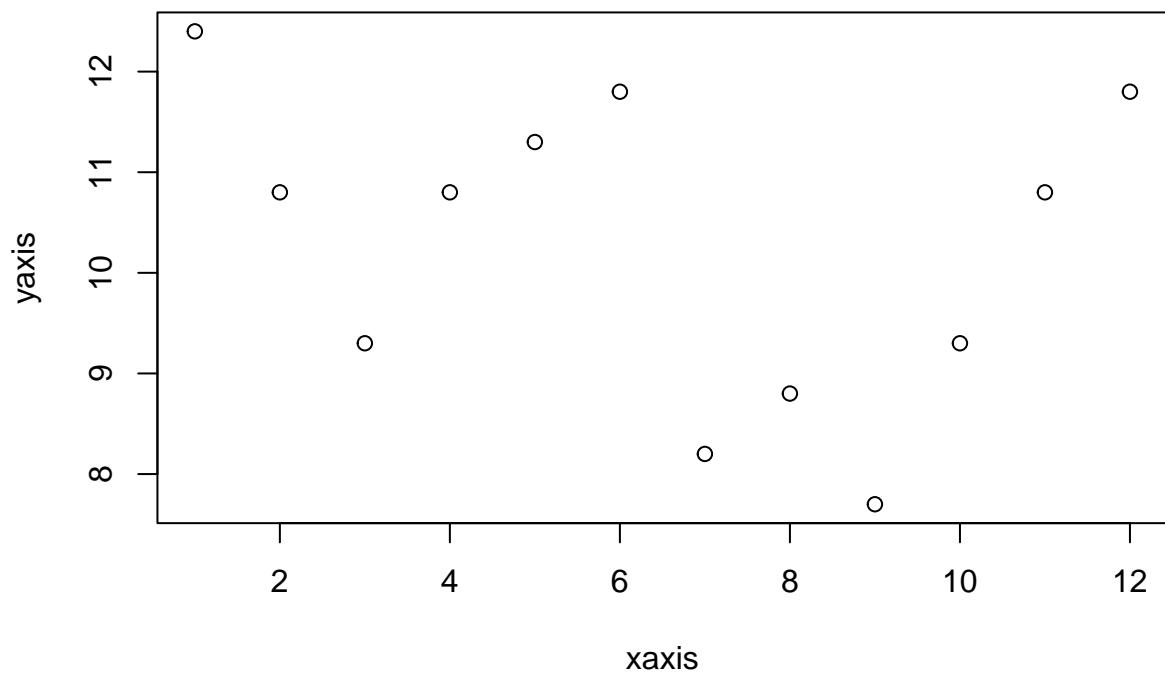
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 6
## [1] 14 19
## integer(0)
## [1] 12
## [1] "2023-01-04 11:00:00" "2023-12-11 02:00:00"
## integer(0)
## [1] 9
## [1] -16   8
## integer(0)
## [1] 12
## [1] "2023-01-18 15:00:00" "2023-12-03 06:00:00"
## integer(0)
## [1] 12
## [1]  4.6 13.1
```

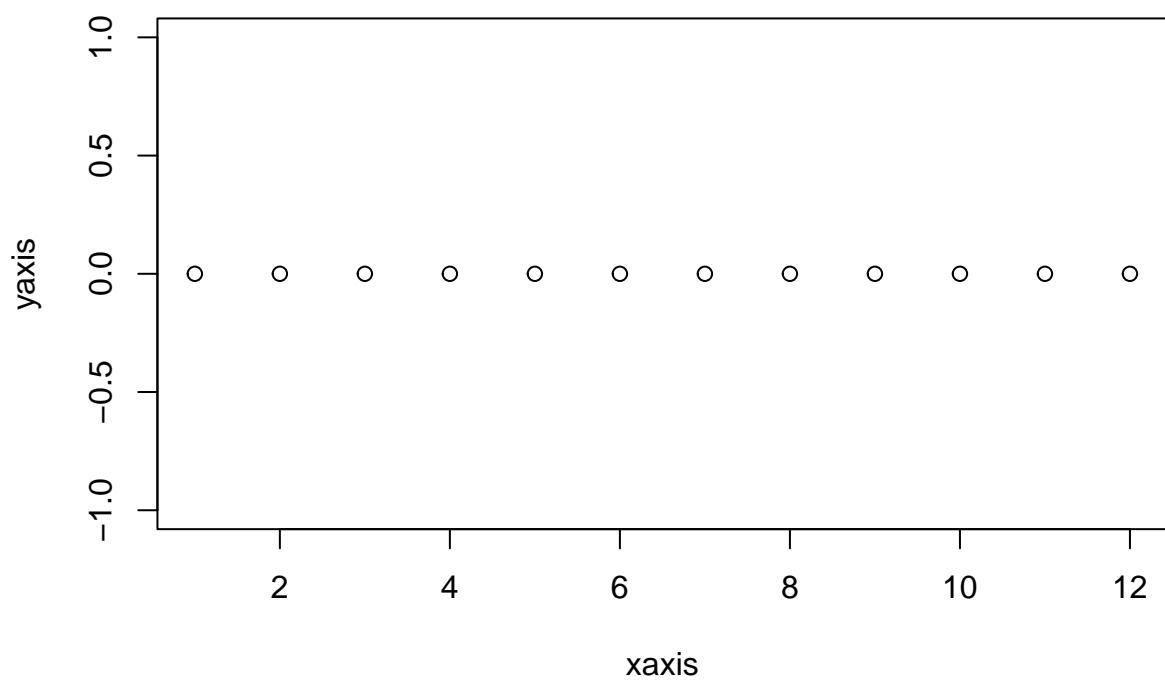


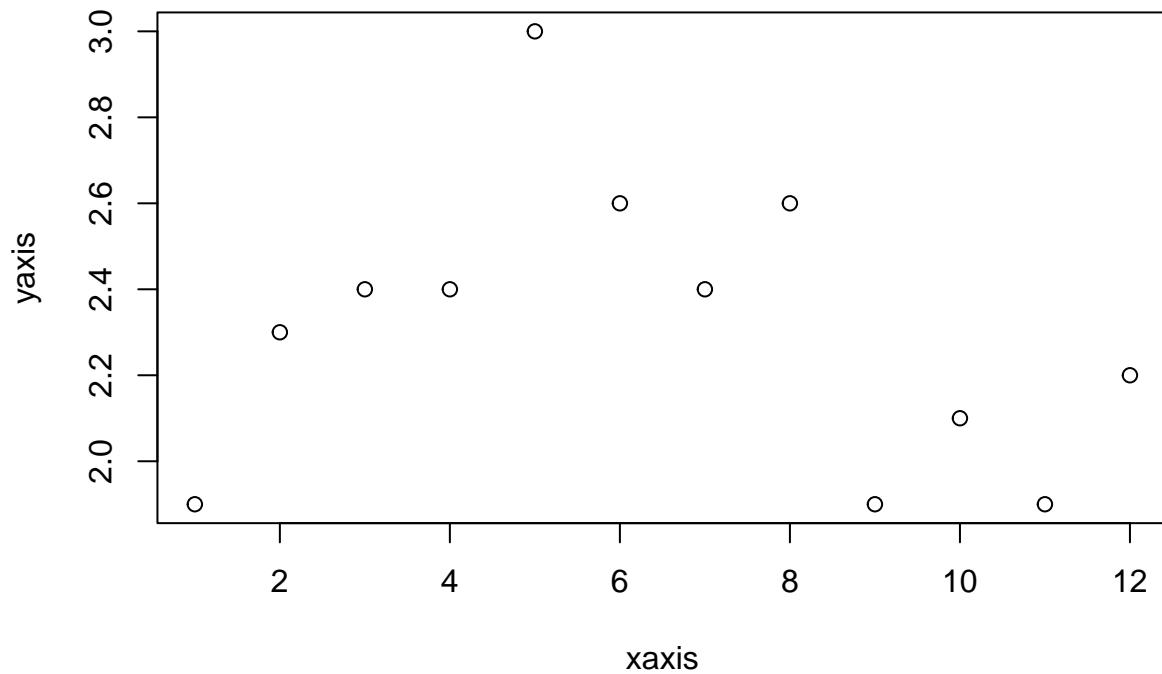




```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] 7.7 12.4
## integer(0)
## [1] 12
## [1] "2023-01-18 23:00:00" "2023-12-17 02:00:00"
## integer(0)
## [1] 1
## [1] 0 0
## integer(0)
## [1] 12
## [1] "2023-01-01 01:00:00" "2023-12-01 04:00:00"
## integer(0)
## [1] 7
## [1] 1.9 3.0
```







```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] " 7" "9"
## integer(0)
## [1] 7
## [1] " 2" "4"
## integer(0)
## [1] 8
## [1] " 0" "7"
## integer(0)
## [1] 5
## [1] " 1" "6"
## integer(0)
## [1] 6
## [1] " 1" "3"
## integer(0)
## [1] 6
## [1] " 2" "3"
## integer(0)
## [1] 8
## [1] " 1" "8"
## integer(0)
## [1] 8
## [1] " 0" "5"
## integer(0)
```

```

## [1] 9
## [1] " 1" "9"
## integer(0)
## [1] 6
## [1] " 3" "7"
## integer(0)
## [1] 10
## [1] " 2" "5"
## integer(0)
## [1] 12
## [1] " 1" "5"
## integer(0)
## [1] 10
## [1] " 5" "8"
## integer(0)
## [1] 10
## [1] " 1" "6"
## integer(0)
## [1] 9
## [1] " 2" "7"
## integer(0)
## [1] 6
## [1] " 8" "8"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] " 5.9" "12.9"
## integer(0)
## [1] 12
## [1] " 5.6" "12.8"
## integer(0)
## [1] 12
## [1] " 5.4" "12.9"
## integer(0)
## [1] 11
## [1] " 5.2" "12.6"
## integer(0)
## [1] 12
## [1] " 4.9" "12.5"
## integer(0)
## [1] 12
## [1] " 4.9" "12.6"
## integer(0)
## [1] 12
## [1] " 4.5" "12.9"
## integer(0)
## [1] 10
## [1] " 4.4" "14.8"
## integer(0)
## [1] 12
## [1] " 5.1" "17.3"
## integer(0)
## [1] 12
## [1] " 7.5" "20.0"

```

```

## integer(0)
## [1] 10
## [1] "9.7" "22.5"
## integer(0)
## [1] 12
## [1] "11.2" "24.4"
## integer(0)
## [1] 12
## [1] "12.4" "26.0"
## integer(0)
## [1] 12
## [1] "13.1" "27.0"
## integer(0)
## [1] 12
## [1] "13.4" "28.1"
## integer(0)
## [1] 11
## [1] "13.1" "28.3"
## integer(0)
## [1] 12
## [1] "12.1" "27.5"
## integer(0)
## [1] 11
## [1] "10.5" "25.2"
## integer(0)
## [1] 12
## [1] "9.1" "22.4"
## integer(0)
## [1] 12
## [1] "8.2" "19.6"
## integer(0)
## [1] 11
## [1] "7.9" "16.6"
## integer(0)
## [1] 12
## [1] "7.3" "14.8"
## integer(0)
## [1] 11
## [1] "6.8" "13.8"
## integer(0)
## [1] 12
## [1] "6.3" "13.2"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "8.3" "7.2"
## integer(0)
## [1] 11
## [1] "8.1" "6.6"
## integer(0)
## [1] 12
## [1] "7.9" "7.1"
## integer(0)
## [1] 12

```

```
## [1] " 7.9" "6.6"
## integer(0)
## [1] 12
## [1] " 7.8" "6.7"
## integer(0)
## [1] 12
## [1] " 7.6" "6.2"
## integer(0)
## [1] 12
## [1] " 7.8" "6.4"
## integer(0)
## [1] 11
## [1] " 8.6" "6.6"
## integer(0)
## [1] 12
## [1] " 8.8" "7.6"
## integer(0)
## [1] 11
## [1] " 8.8" "8.2"
## integer(0)
## [1] 11
## [1] " 9.1" "8.1"
## integer(0)
## [1] 11
## [1] " 9.3" "7.6"
## integer(0)
## [1] 12
## [1] " 9.3" "7.3"
## integer(0)
## [1] 12
## [1] " 8.9" "7.2"
## integer(0)
## [1] 12
## [1] " 8.9" "7.1"
## integer(0)
## [1] 11
## [1] " 8.4" "7.3"
## integer(0)
## [1] 9
## [1] " 9.0" "7.2"
## integer(0)
## [1] 12
## [1] " 8.9" "7.7"
## integer(0)
## [1] 11
## [1] " 8.9" "8.4"
## integer(0)
## [1] 10
## [1] " 8.9" "8.1"
## integer(0)
## [1] 12
## [1] " 9.0" "8.1"
## integer(0)
## [1] 11
```

```

## [1] " 8.9" "7.5"
## integer(0)
## [1] 12
## [1] " 8.5" "7.3"
## integer(0)
## [1] 12
## [1] " 8.3" "7.3"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] "82" "98"
## integer(0)
## [1] 9
## [1] "83" "98"
## integer(0)
## [1] 9
## [1] "83" "98"
## integer(0)
## [1] 10
## [1] "83" "98"
## integer(0)
## [1] 7
## [1] "84" "99"
## integer(0)
## [1] 10
## [1] "84" "99"
## integer(0)
## [1] 9
## [1] "82" "98"
## integer(0)
## [1] 11
## [1] "73" "97"
## integer(0)
## [1] 9
## [1] "66" "92"
## integer(0)
## [1] 11
## [1] "57" "89"
## integer(0)
## [1] 9
## [1] "50" "84"
## integer(0)
## [1] 8
## [1] "46" "76"
## integer(0)
## [1] 11
## [1] "42" "69"
## integer(0)
## [1] 11
## [1] "39" "64"
## integer(0)
## [1] 11
## [1] "38" "61"
## integer(0)

```

```

## [1] 10
## [1] "38" "61"
## integer(0)
## [1] 10
## [1] "39" "66"
## integer(0)
## [1] 10
## [1] "41" "74"
## integer(0)
## [1] 11
## [1] "46" "83"
## integer(0)
## [1] 9
## [1] "56" "86"
## integer(0)
## [1] 11
## [1] "68" "88"
## integer(0)
## [1] 10
## [1] "74" "91"
## integer(0)
## [1] 9
## [1] "79" "94"
## integer(0)
## [1] 9
## [1] "81" "96"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 7
## [1] " 0" "376"
## integer(0)
## [1] 11
## [1] " 0" "528"
## integer(0)
## [1] 12
## [1] "126" "635"
## integer(0)
## [1] 12
## [1] "301" "714"

```

```

## integer(0)
## [1] 12
## [1] "394" "766"
## integer(0)
## [1] 12
## [1] "470" "827"
## integer(0)
## [1] 12
## [1] "515" "860"
## integer(0)
## [1] 12
## [1] "539" "878"
## integer(0)
## [1] 12
## [1] "501" "866"
## integer(0)
## [1] 12
## [1] "421" "829"
## integer(0)
## [1] 12
## [1] "268" "765"
## integer(0)
## [1] 12
## [1] " 0" "688"
## integer(0)
## [1] 8
## [1] " 0" "545"
## integer(0)
## [1] 5
## [1] " 0" "226"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 7
## [1] "1.1" "2.2"
## integer(0)
## [1] 8
## [1] "0.9" "2.1"
## integer(0)
## [1] 7

```

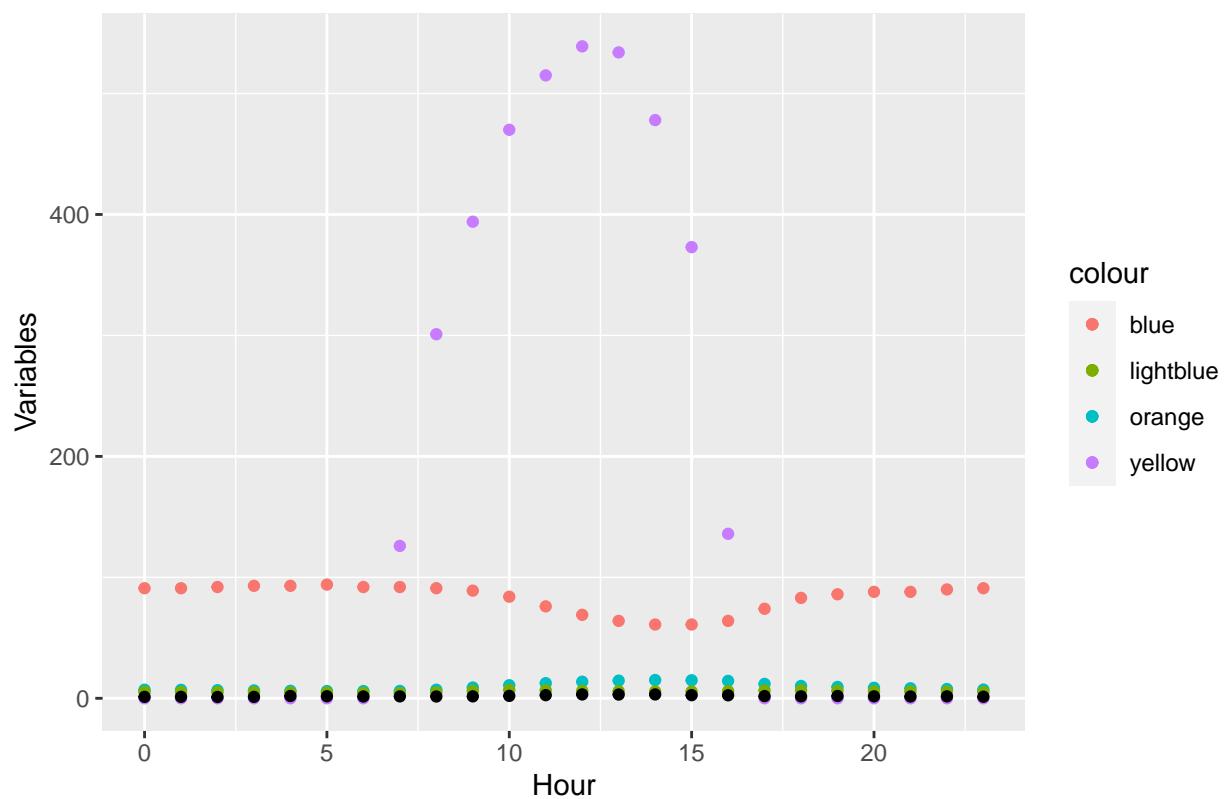
```

## [1] "0.9" "2.2"
## integer(0)
## [1] 7
## [1] "0.8" "2.0"
## integer(0)
## [1] 9
## [1] "0.6" "1.9"
## integer(0)
## [1] 8
## [1] "0.4" "2.2"
## integer(0)
## [1] 9
## [1] "0.5" "2.0"
## integer(0)
## [1] 8
## [1] "0.8" "2.0"
## integer(0)
## [1] 9
## [1] "0.9" "2.1"
## integer(0)
## [1] 9
## [1] "1.1" "2.3"
## integer(0)
## [1] 8
## [1] "1.8" "3.0"
## integer(0)
## [1] 8
## [1] "2.0" "3.4"
## integer(0)
## [1] 9
## [1] "2.5" "3.8"
## integer(0)
## [1] 10
## [1] "2.7" "4.7"
## integer(0)
## [1] 10
## [1] "3.2" "5.2"
## integer(0)
## [1] 12
## [1] "2.6" "6.0"
## integer(0)
## [1] 11
## [1] "2.5" "5.4"
## integer(0)
## [1] 10
## [1] "1.8" "5.5"
## integer(0)
## [1] 9
## [1] "1.6" "4.5"
## integer(0)
## [1] 11
## [1] "1.9" "3.7"
## integer(0)
## [1] 11

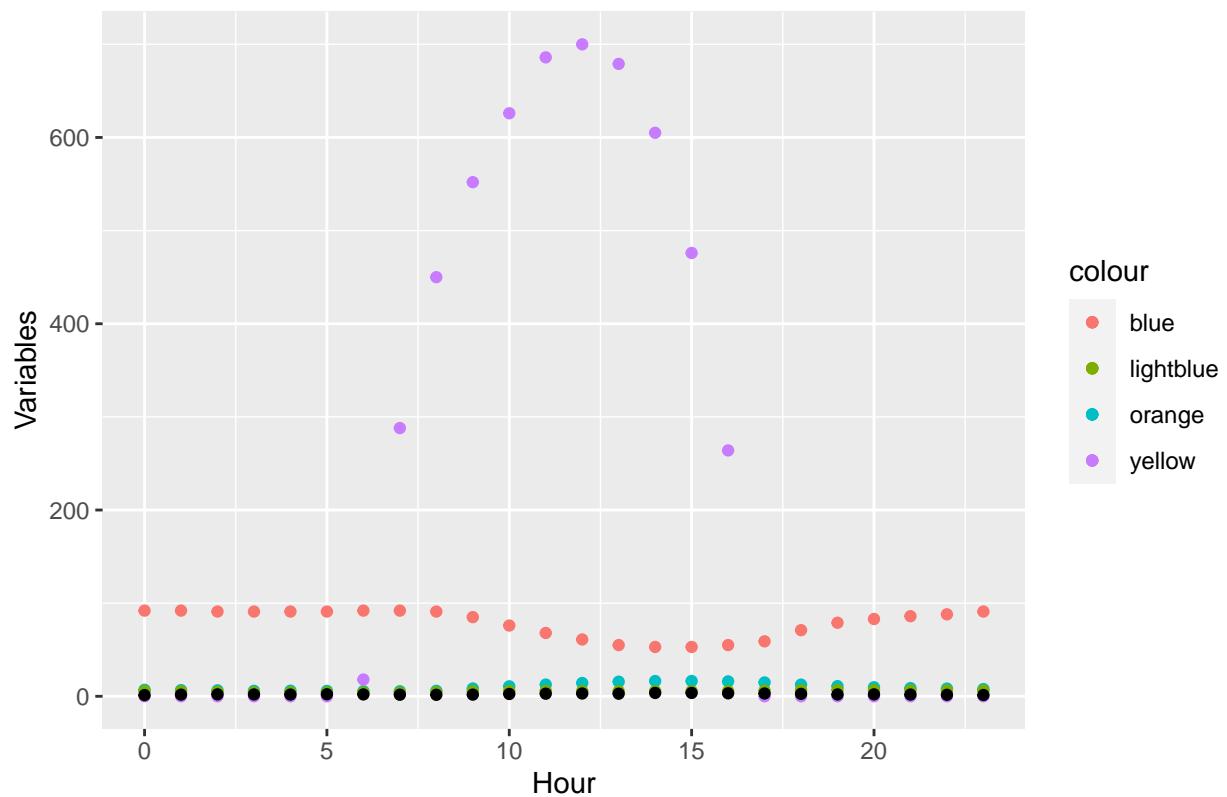
```

```
## [1] "1.6" "3.2"
## integer(0)
## [1] 8
## [1] "1.5" "2.8"
## integer(0)
## [1] 7
## [1] "1.6" "2.5"
## integer(0)
## [1] 9
## [1] "1.1" "2.1"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "Apr" "Sep"
## integer(0)
## [1] 24
## [1] 0 23
## integer(0)
## [1] 165
## [1] 4.4 28.3
## integer(0)
## [1] 105
## [1] 3.1 14.9
## integer(0)
## [1] 62
## [1] 38 99
## integer(0)
## [1] 131
## [1] 0 878
## integer(0)
## [1] 49
## [1] 0.4 6.0
```

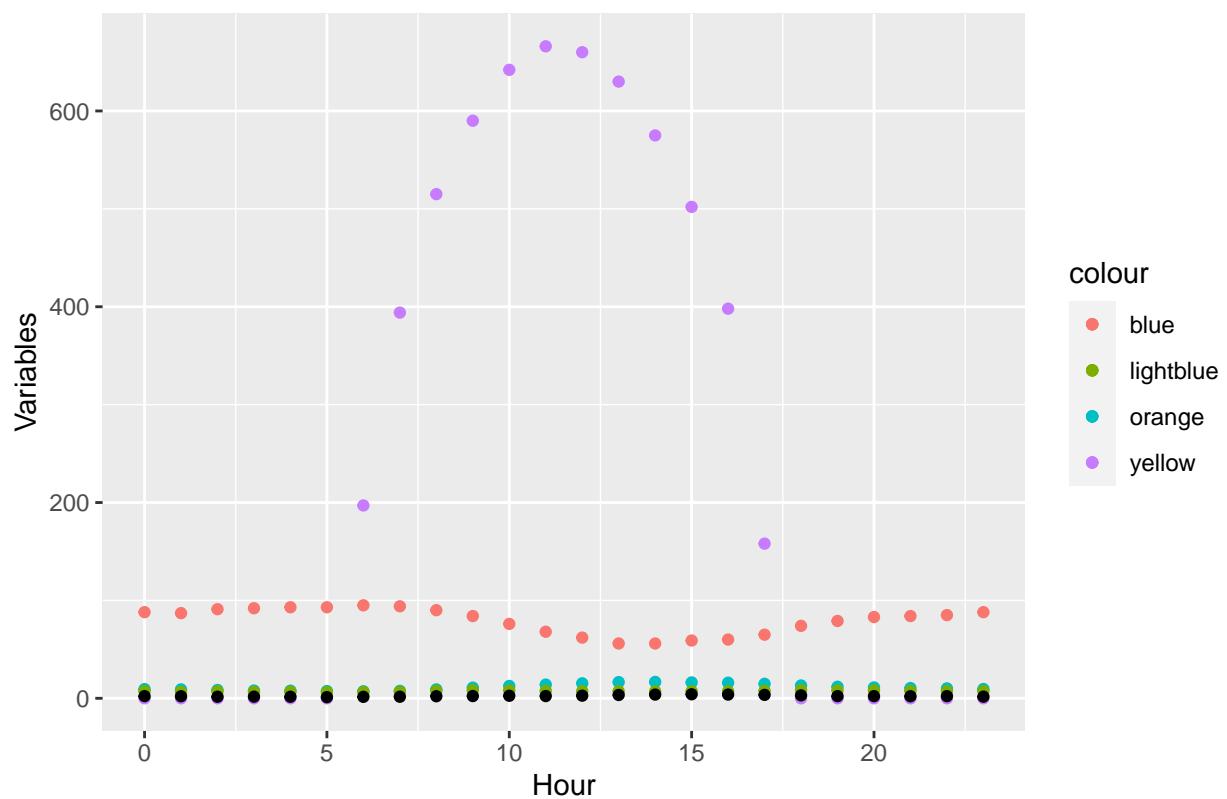
Jan Data: USA\_CA\_Marin.County.AP–Gnoss.Field



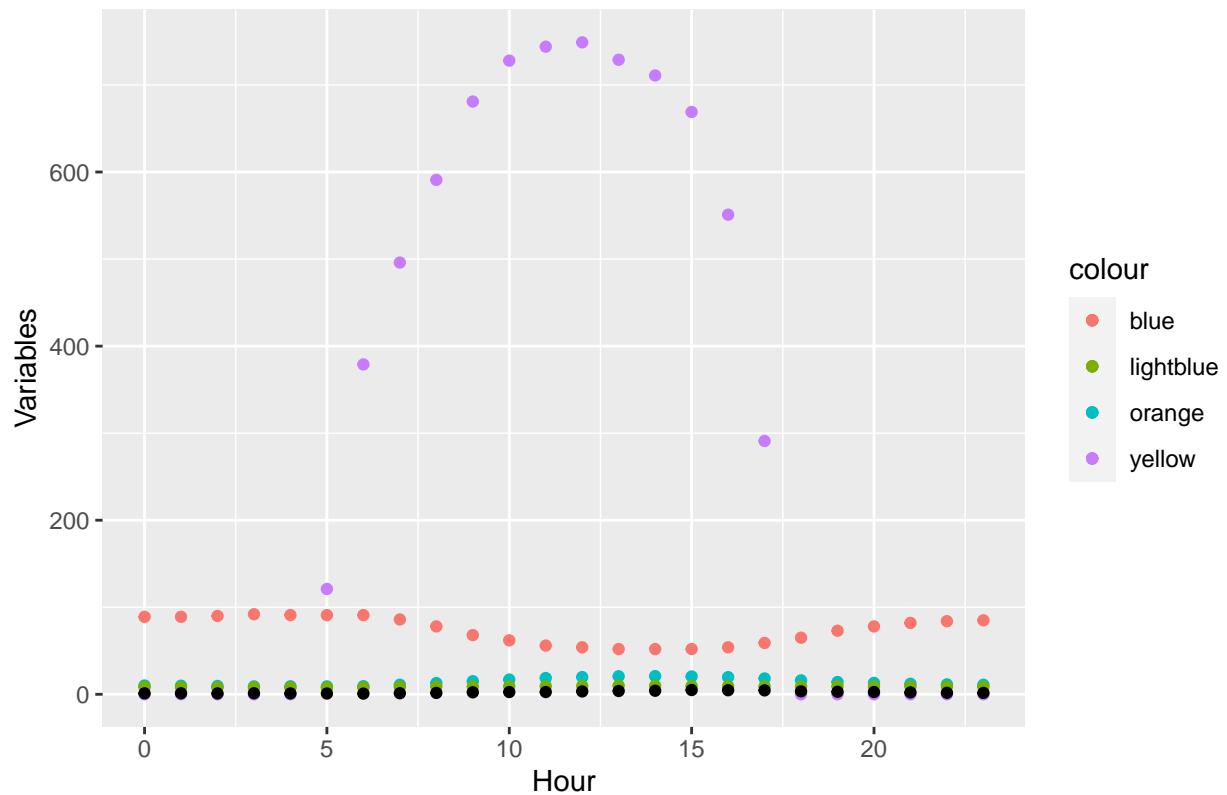
Feb Data: USA\_CA\_Marin.County.AP–Gnoss.Field



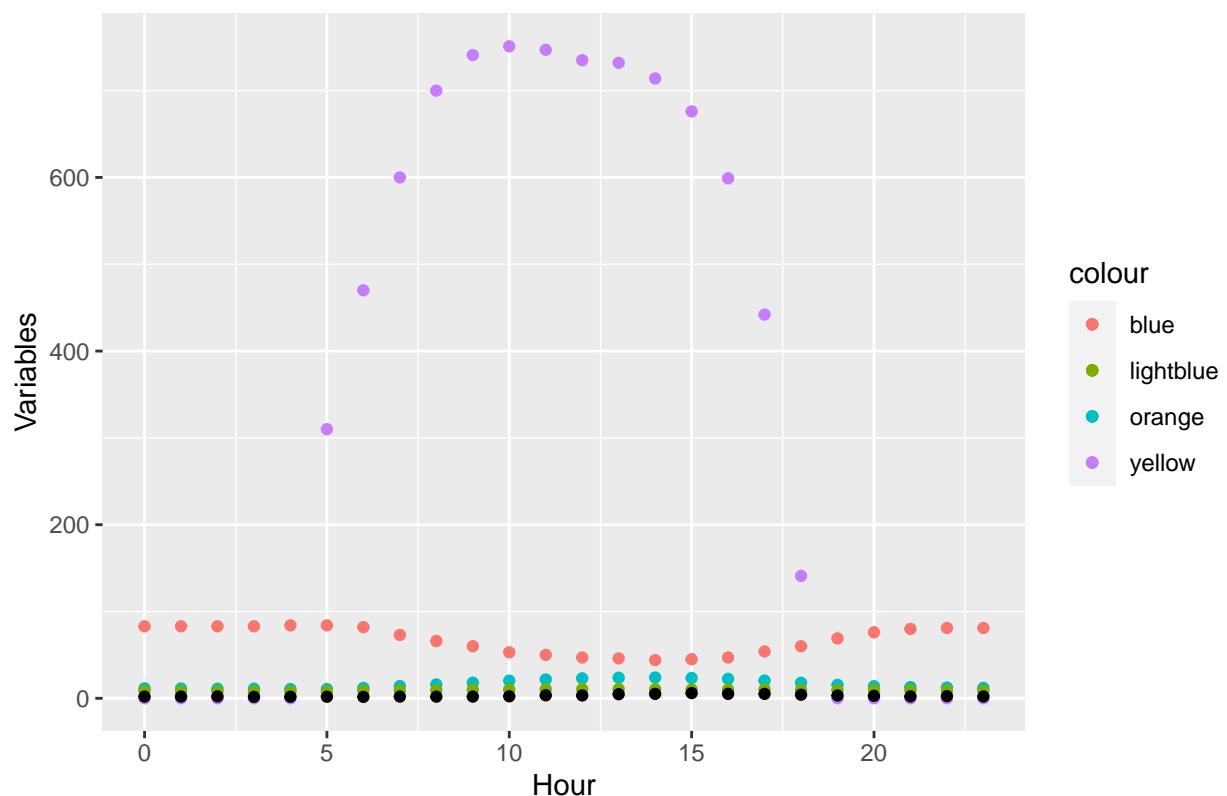
Mar Data: USA\_CA\_Marin.County.AP–Gnoss.Field



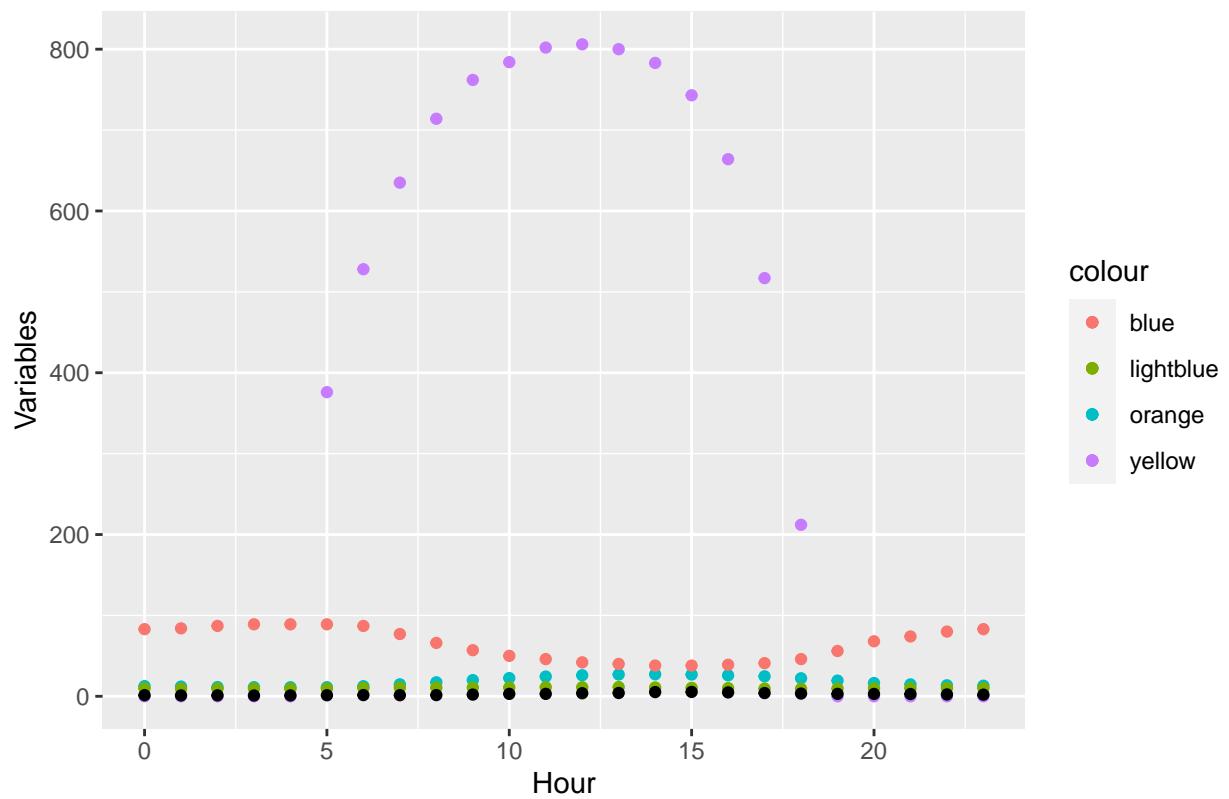
Apr Data: USA\_CA\_Marin.County.AP–Gnoss.Field



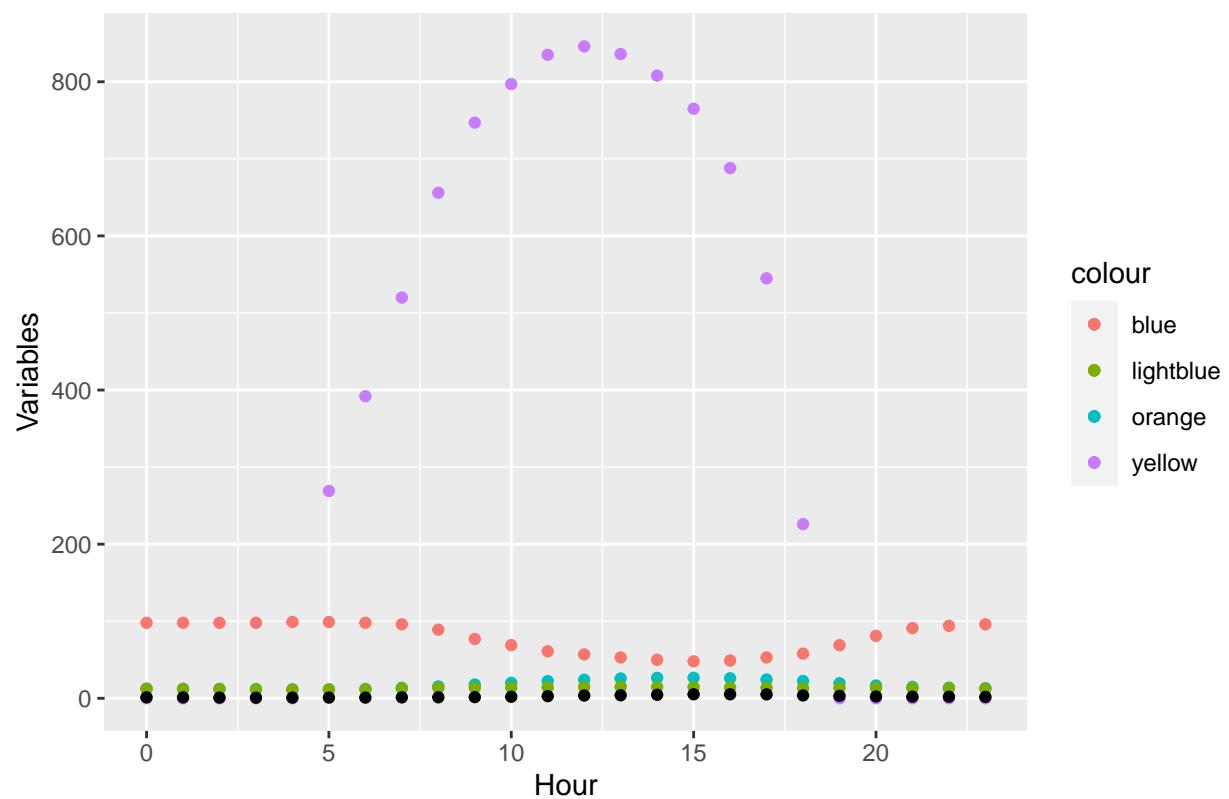
### May Data: USA\_CA\_Marin.County.AP–Gnoss.Field



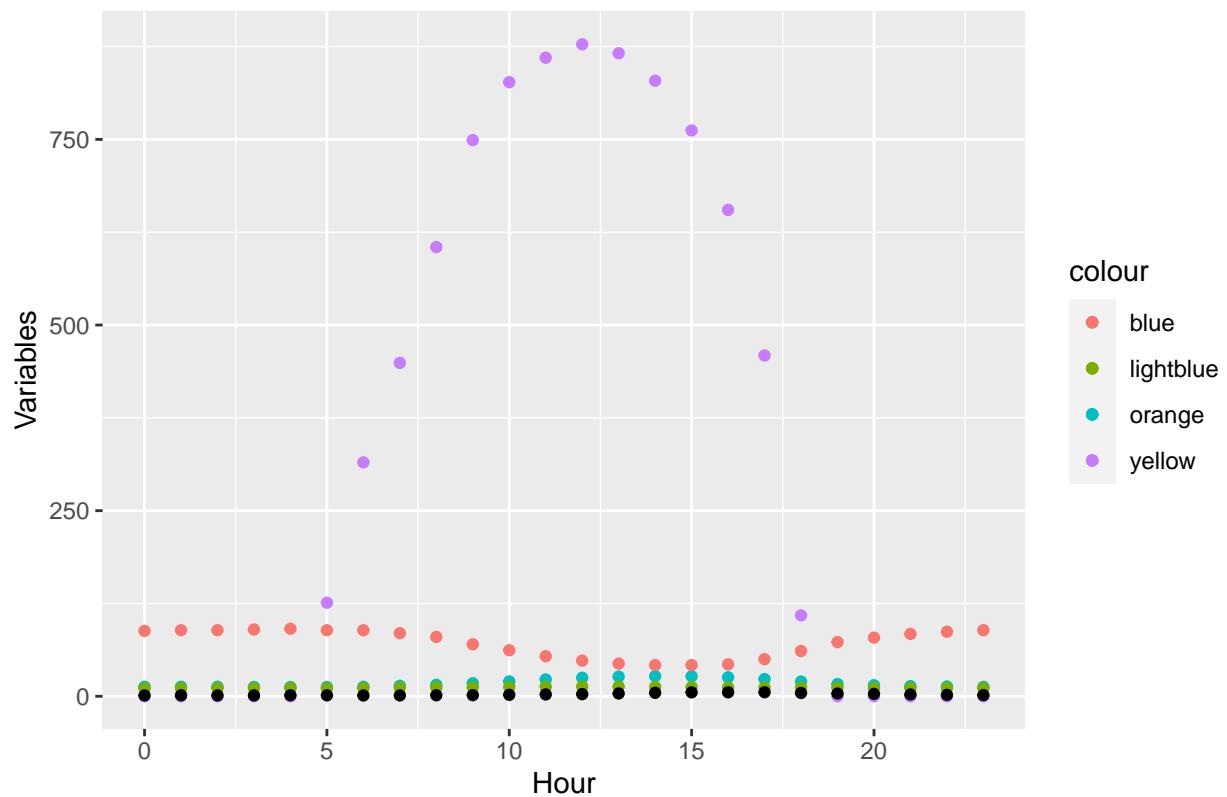
Jun Data: USA\_CA\_Marin.County.AP–Gnoss.Field



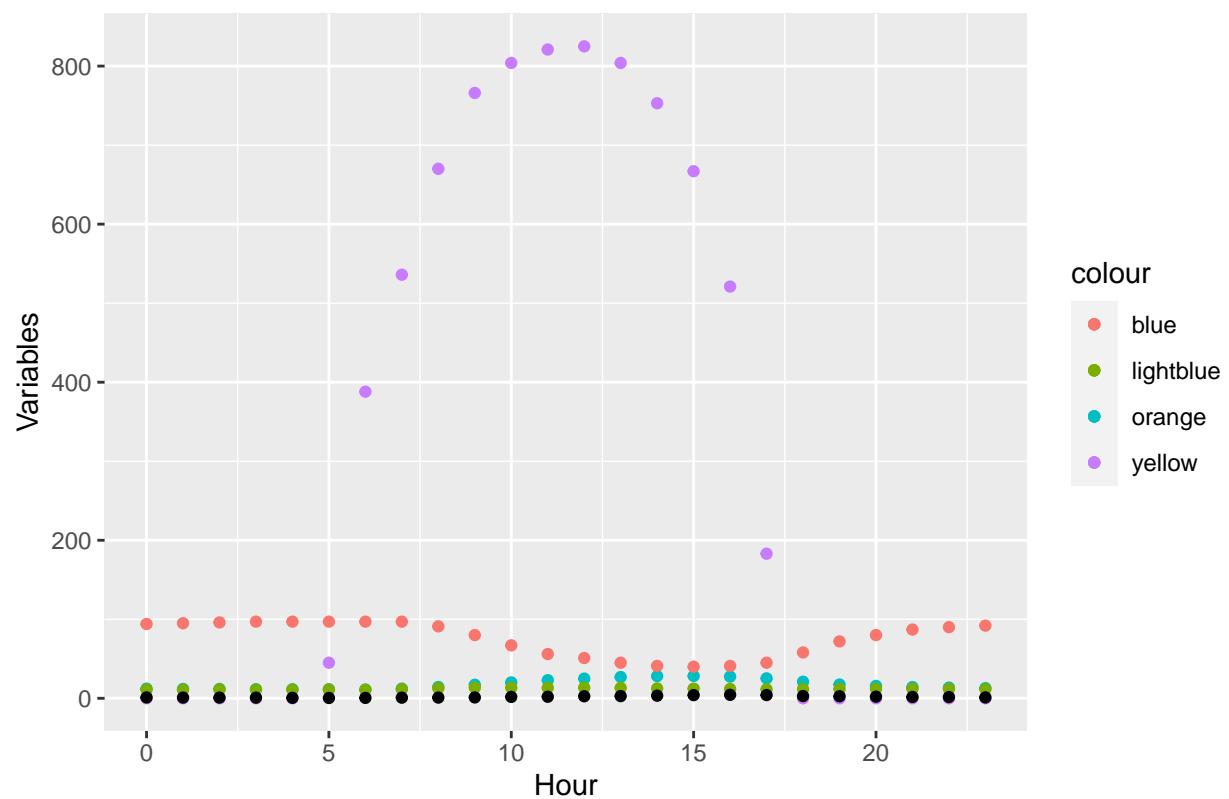
Jul Data: USA\_CA\_Marin.County.AP–Gnoss.Field



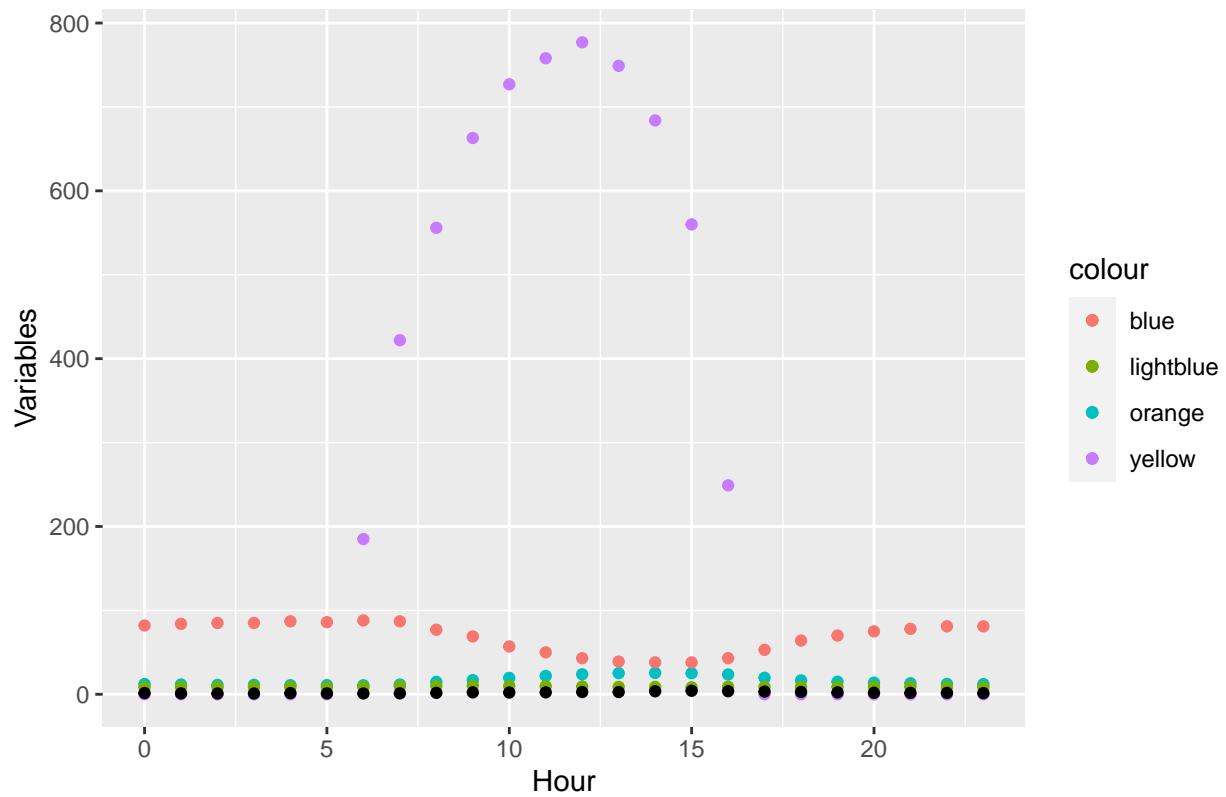
Aug Data: USA\_CA\_Marin.County.AP–Gnoss.Field



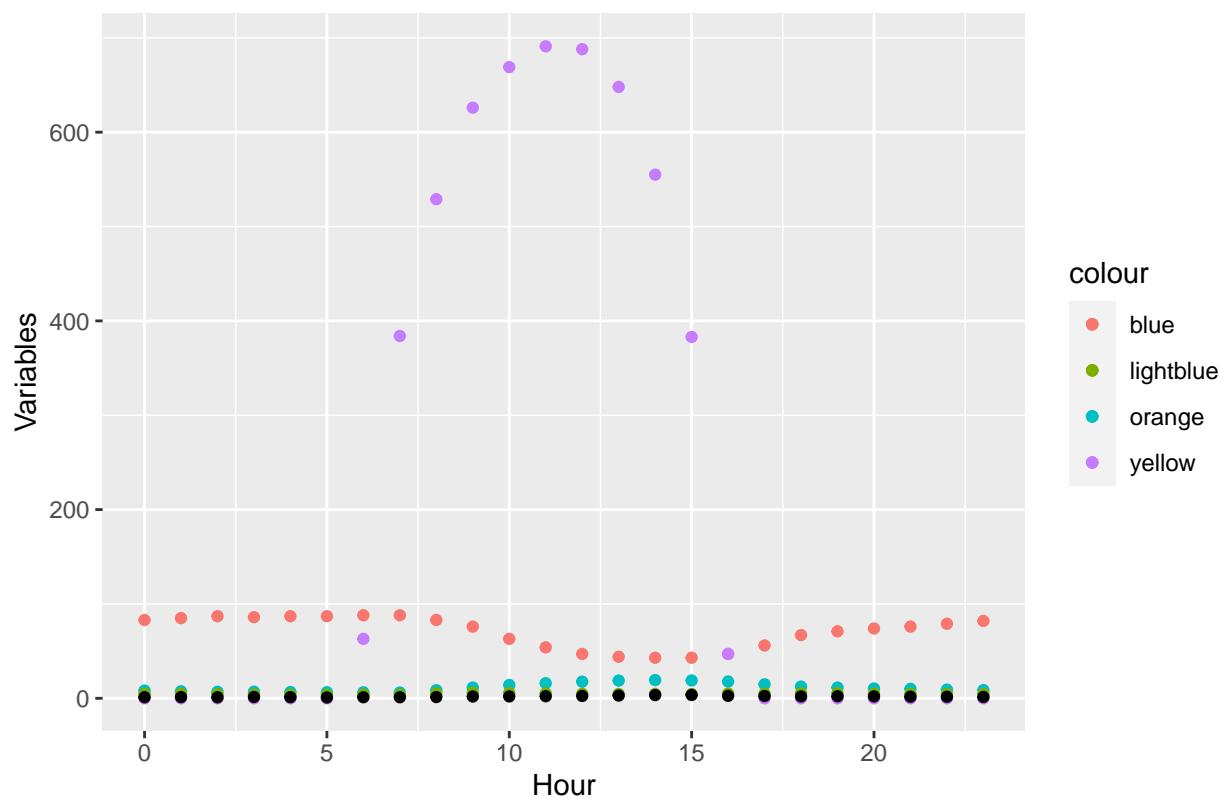
Sep Data: USA\_CA\_Marin.County.AP–Gnoss.Field



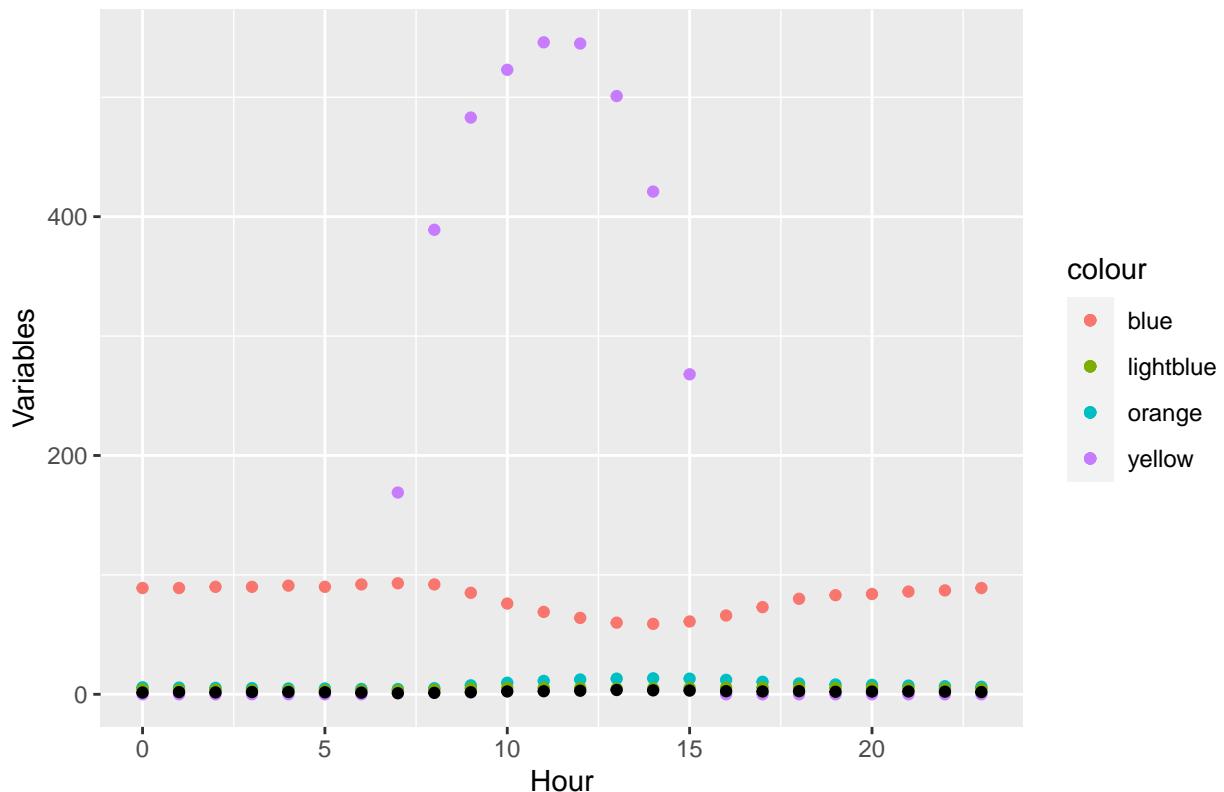
Oct Data: USA\_CA\_Marin.County.AP–Gnoss.Field



### Nov Data: USA\_CA\_Marin.County.AP–Gnoss.Field



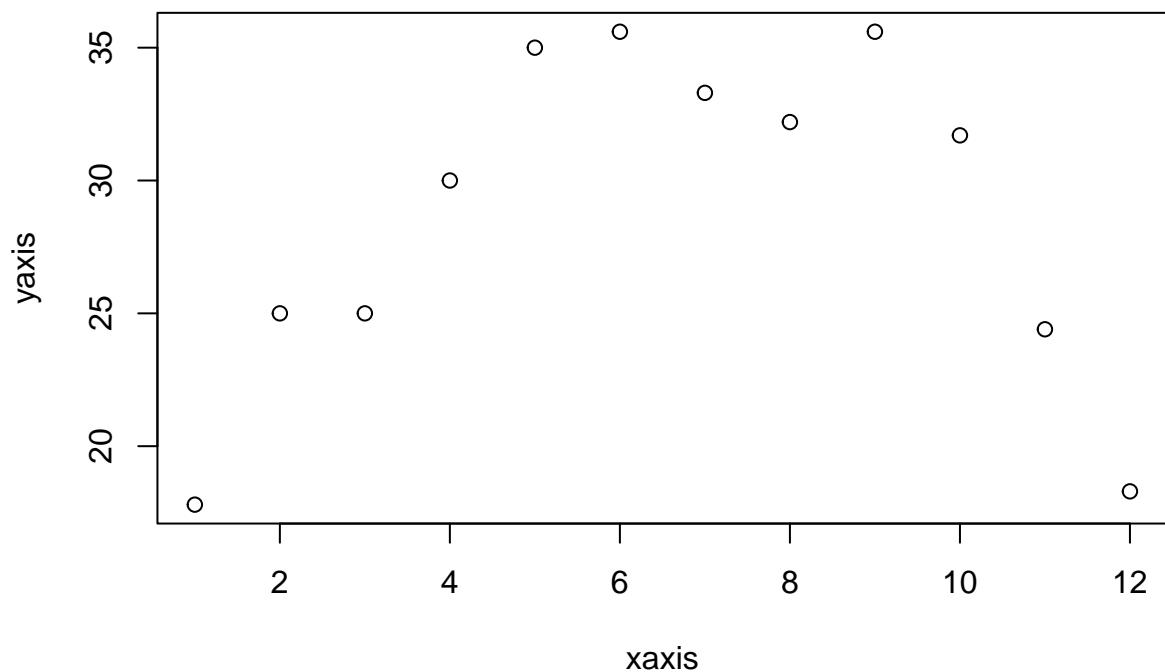
## Dec Data: USA\_CA\_Marin.County.AP–Gnoss.Field

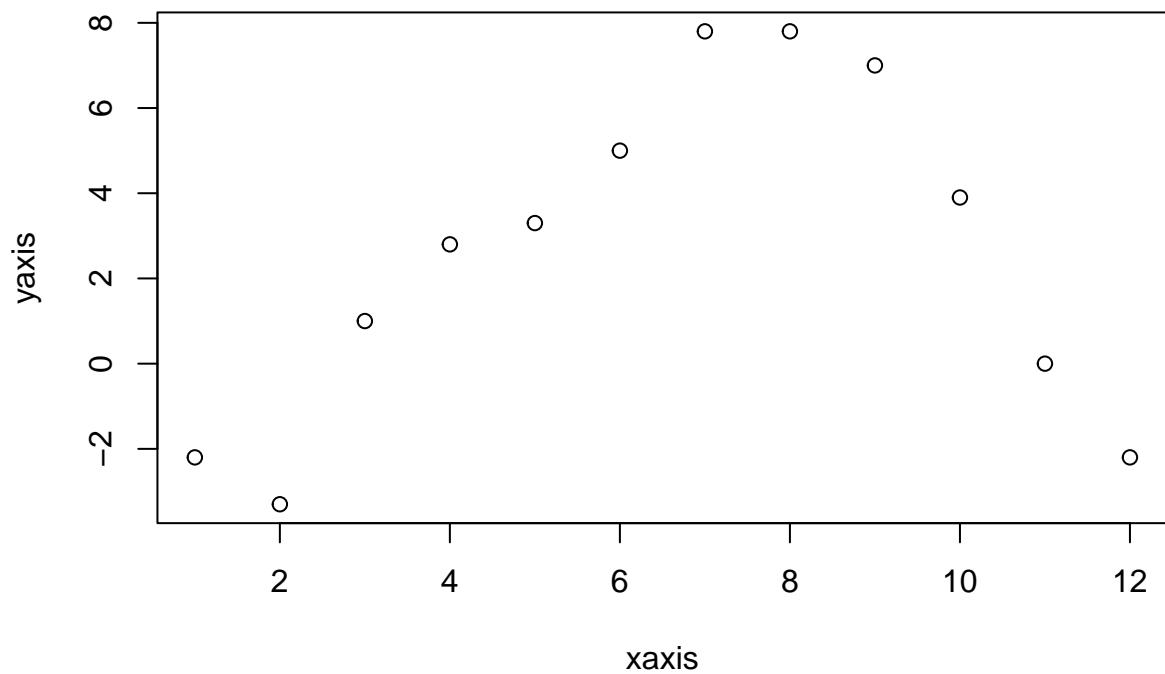


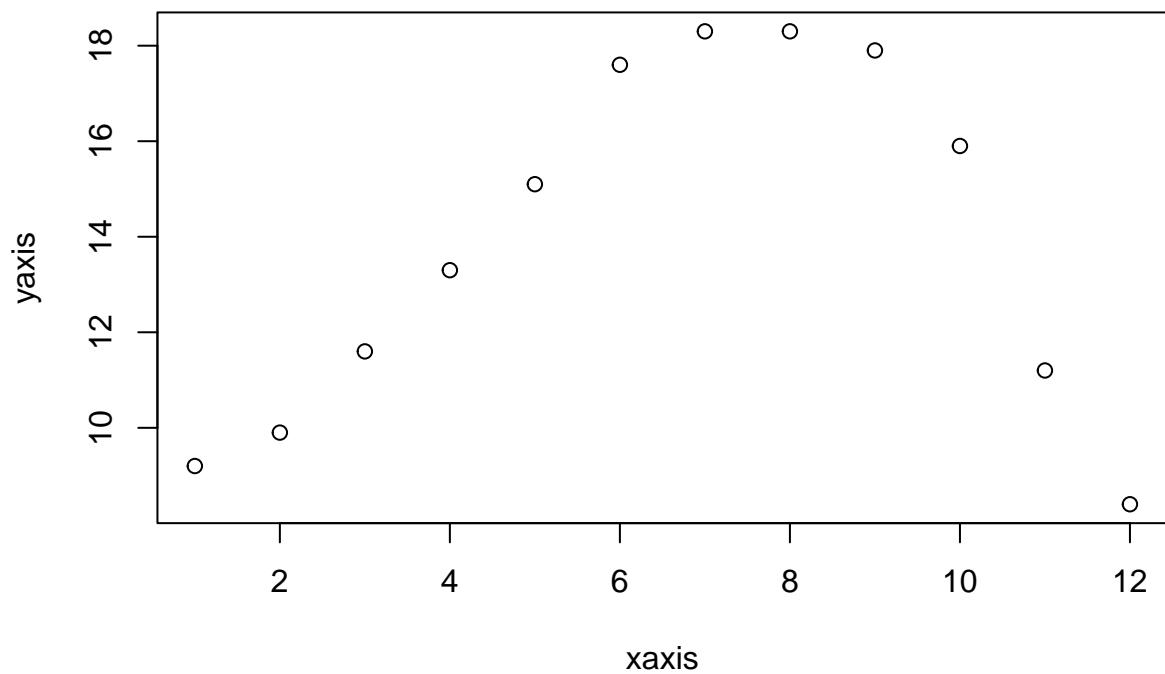
```
stat_napa = stat_parser('dt/USA_CA_Napa.County.AP.724955_TMYx.2007-2021.stat')
```

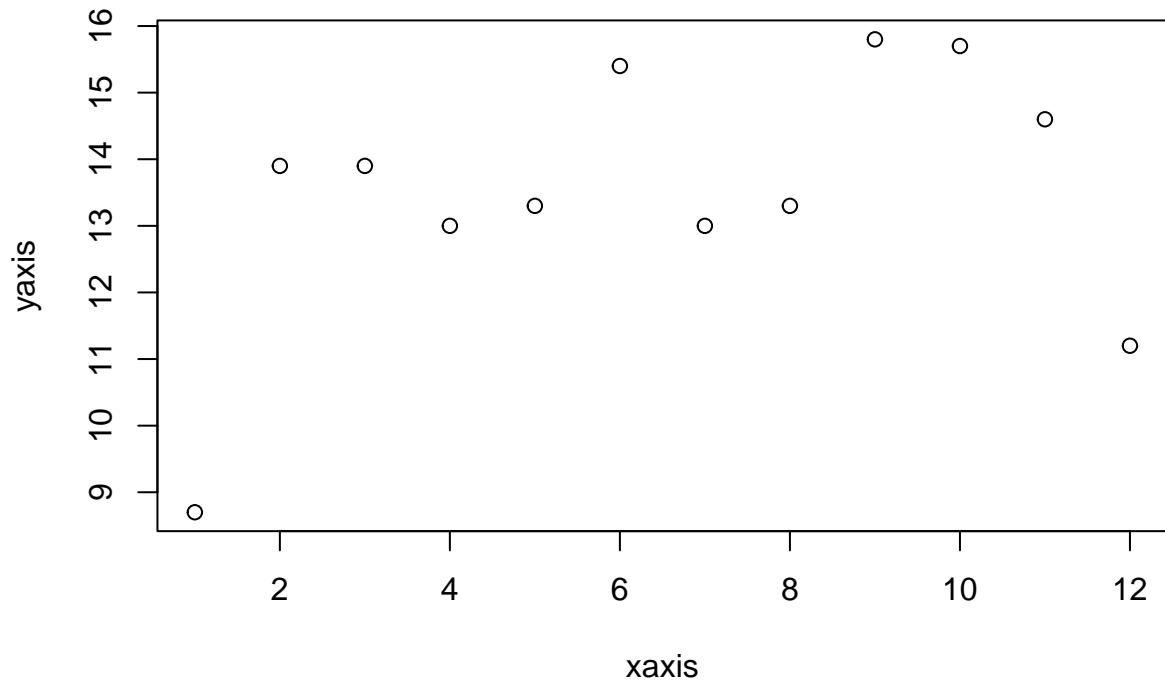
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 10
## [1] 17.8 35.6
## integer(0)
## [1] 12
## [1] "2023-01-08 13:00:00" "2023-12-03 14:00:00"
## integer(0)
## [1] 10
## [1] -3.3 7.8
## integer(0)
## [1] 12
## [1] "2023-01-06 05:00:00" "2023-12-19 04:00:00"
## integer(0)
## [1] 11
## [1] 8.4 18.3
## integer(0)
## [1] 9
## [1] 8.7 15.8
## integer(0)
## [1] 10
## [1] 17.8 35.6
## integer(0)
```

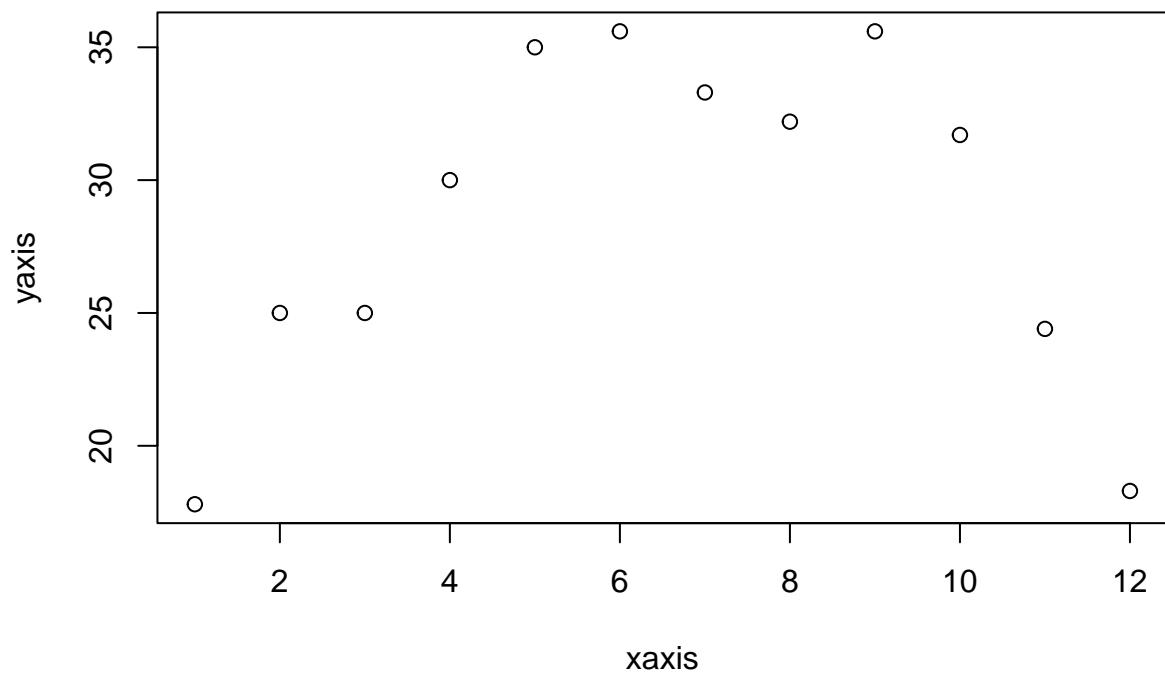
```
## [1] 11
## [1] -2.8 12.0
## integer(0)
## [1] 10
## [1] 10.5 21.8
## integer(0)
## [1] 11
## [1] 16.0 27.8
## integer(0)
## [1] 10
## [1] -3.3 7.8
## integer(0)
## [1] 12
## [1] 6.3 15.2
```

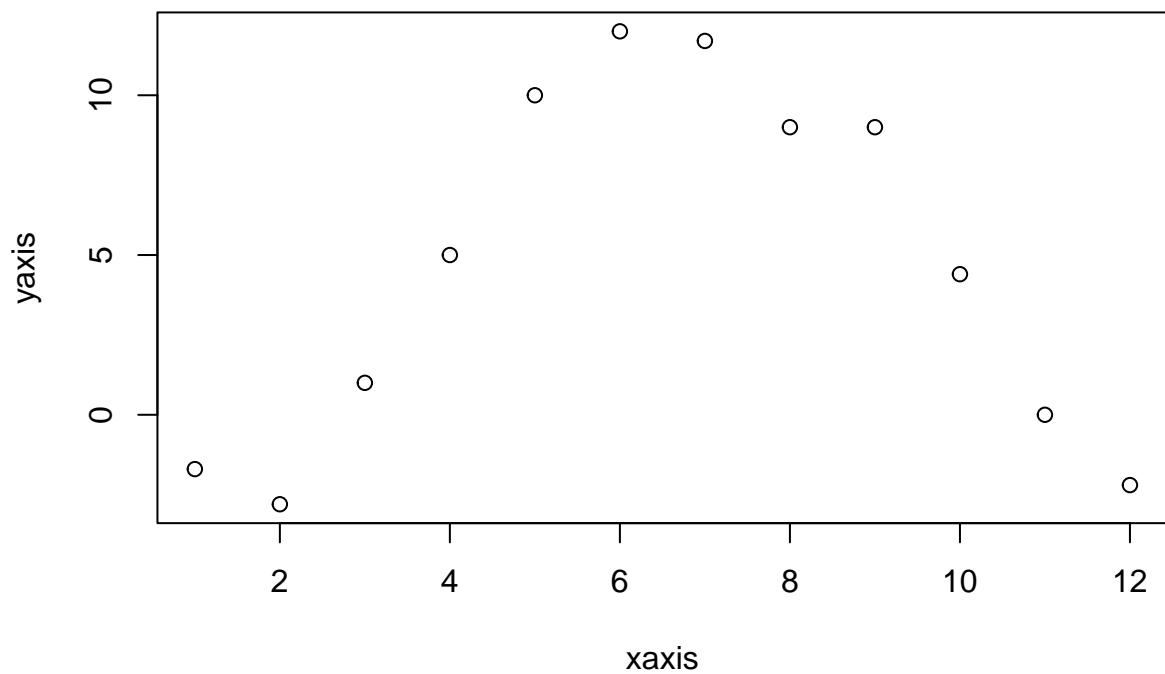


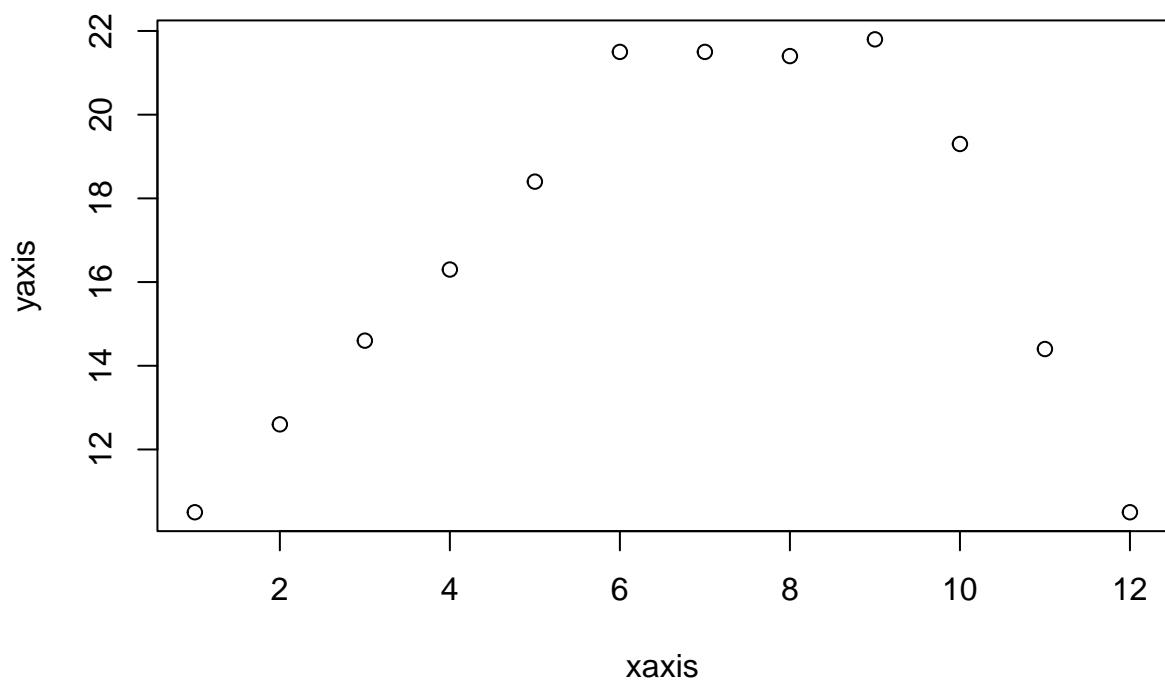


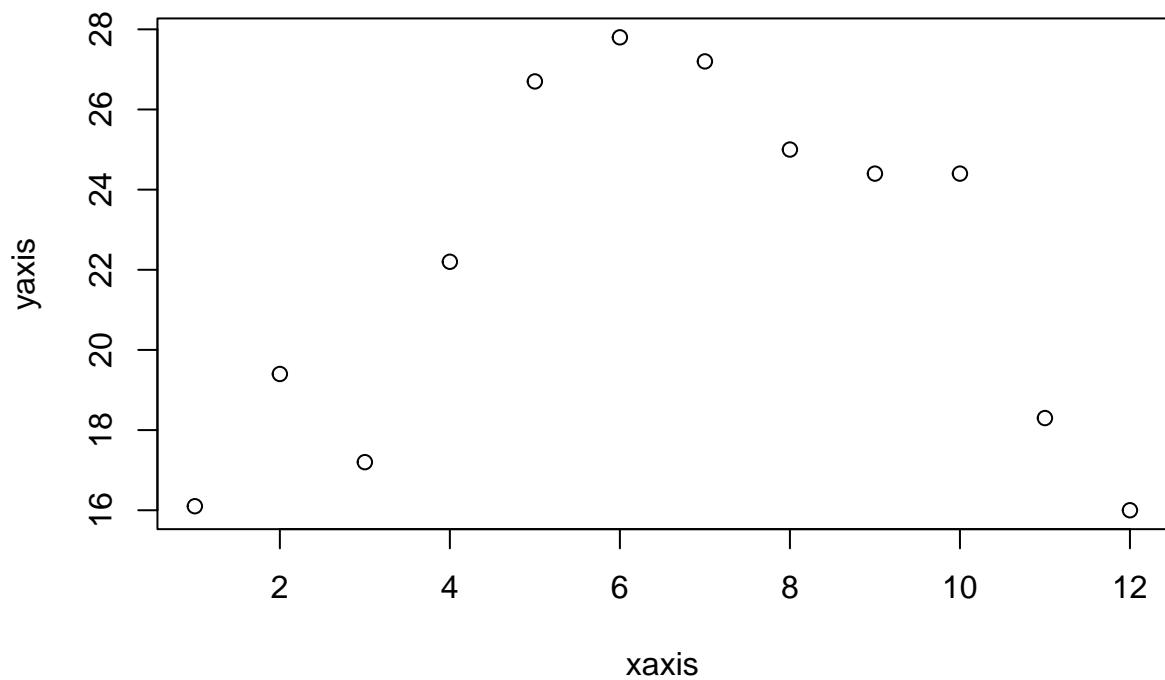


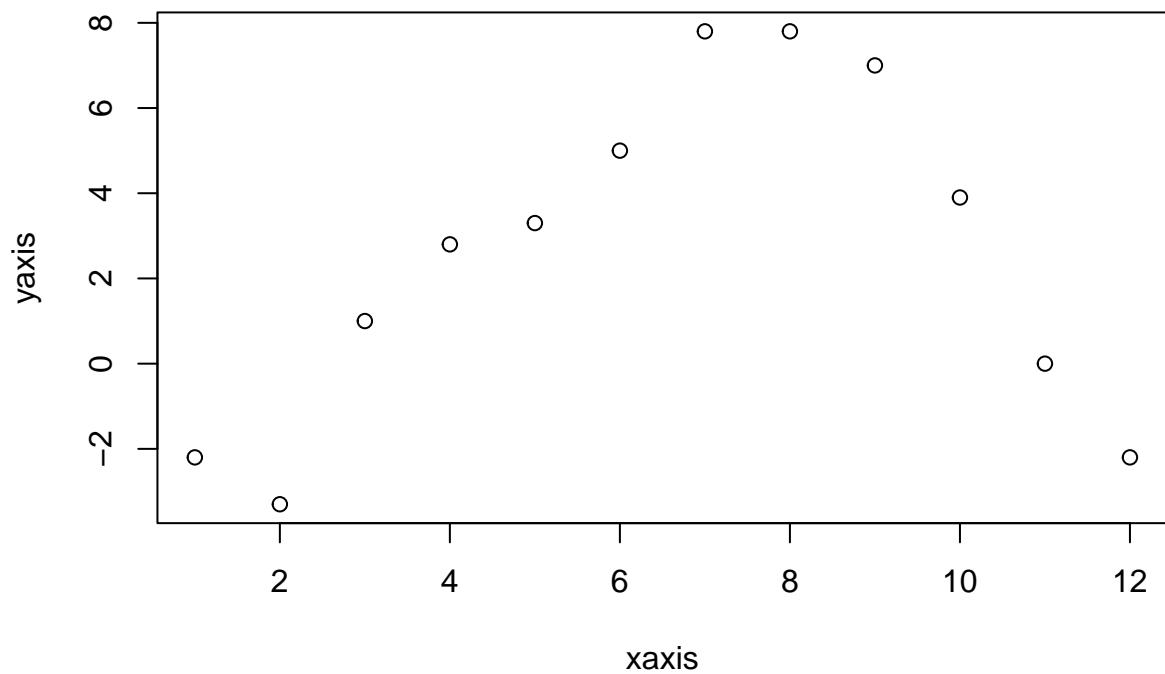


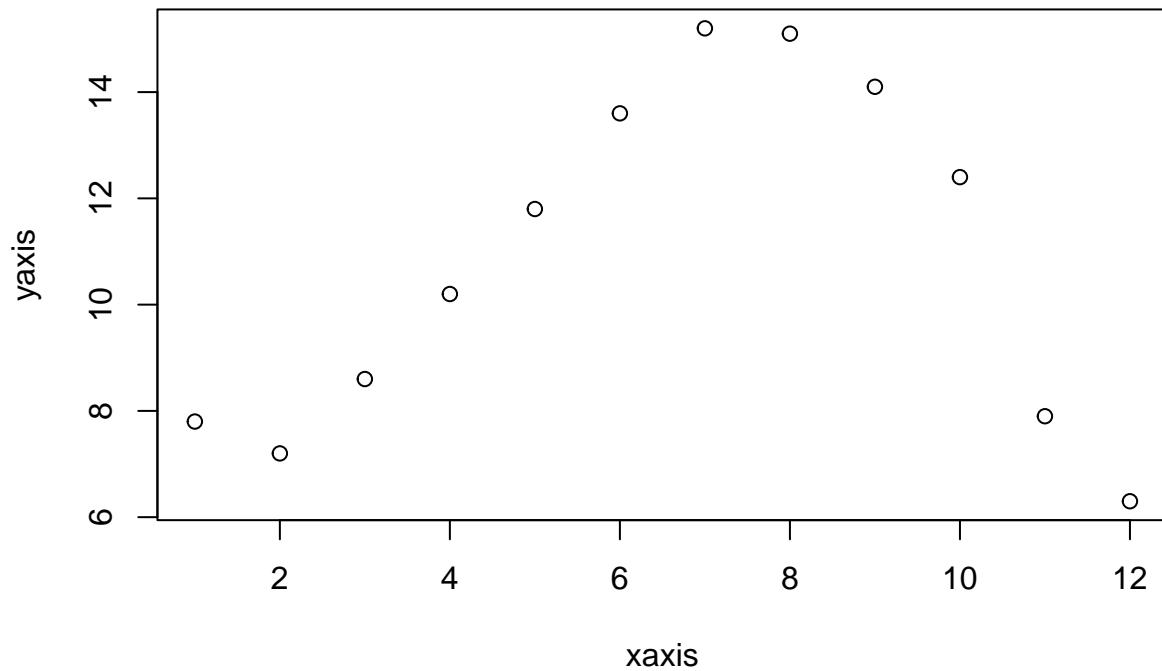




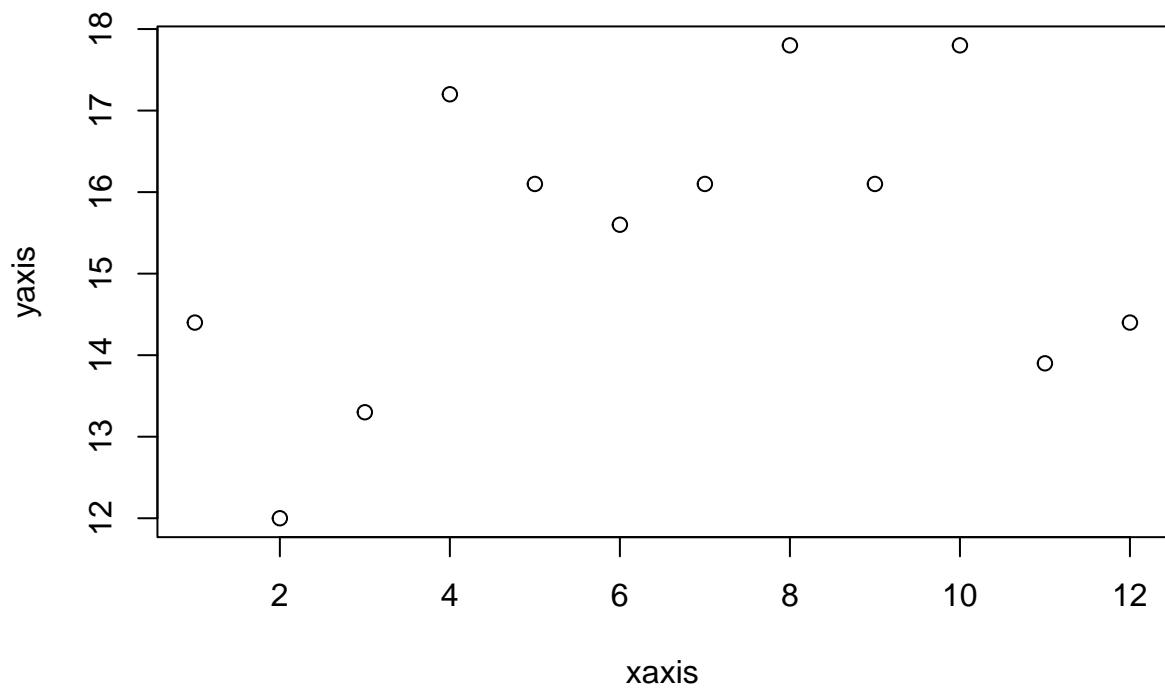


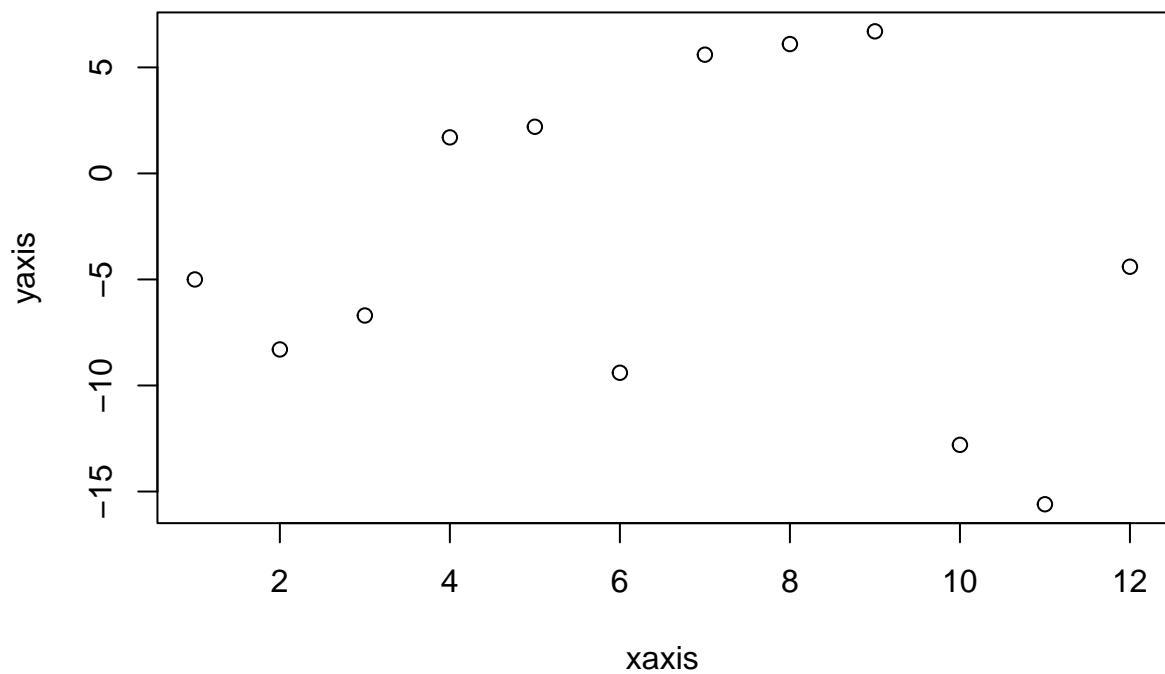


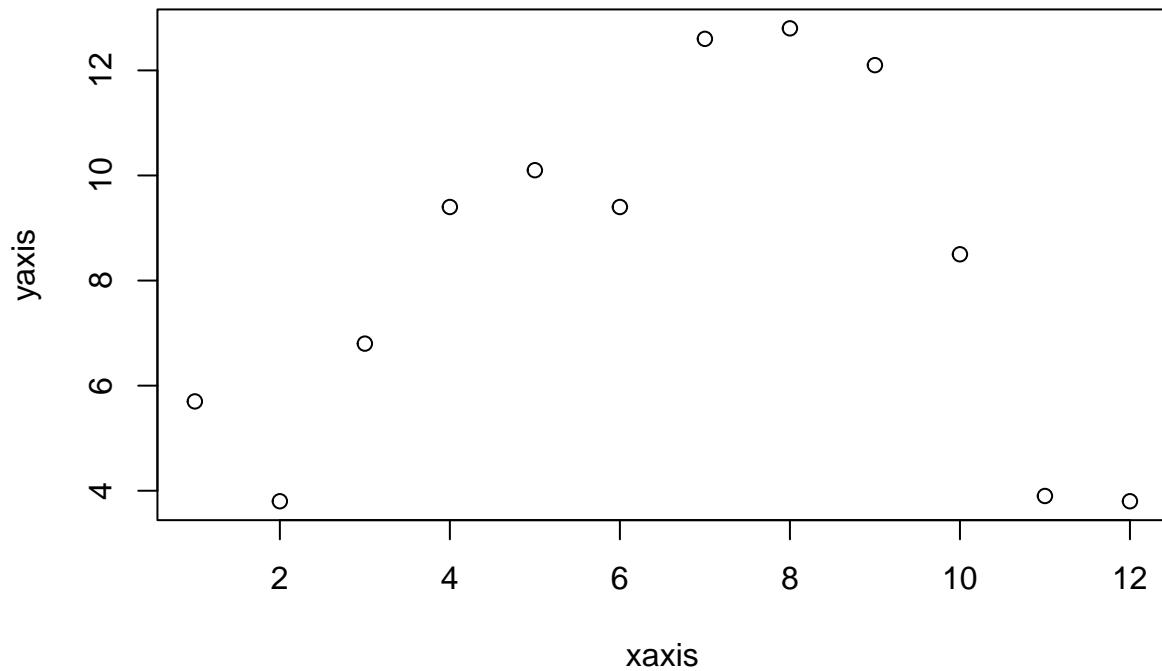




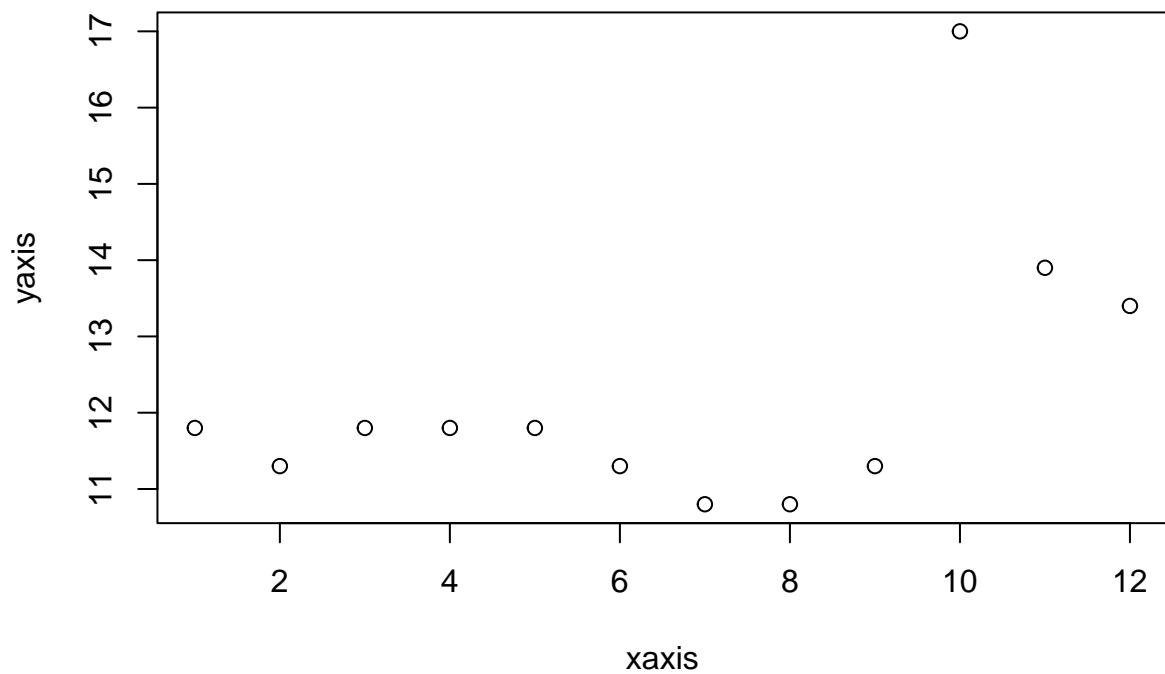
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] 12.0 17.8
## integer(0)
## [1] 12
## [1] "2023-01-08 09:00:00" "2023-12-10 13:00:00"
## integer(0)
## [1] 12
## [1] -15.6   6.7
## integer(0)
## [1] 12
## [1] "2023-01-06 05:00:00" "2023-12-19 05:00:00"
## integer(0)
## [1] 10
## [1]  3.8 12.8
```

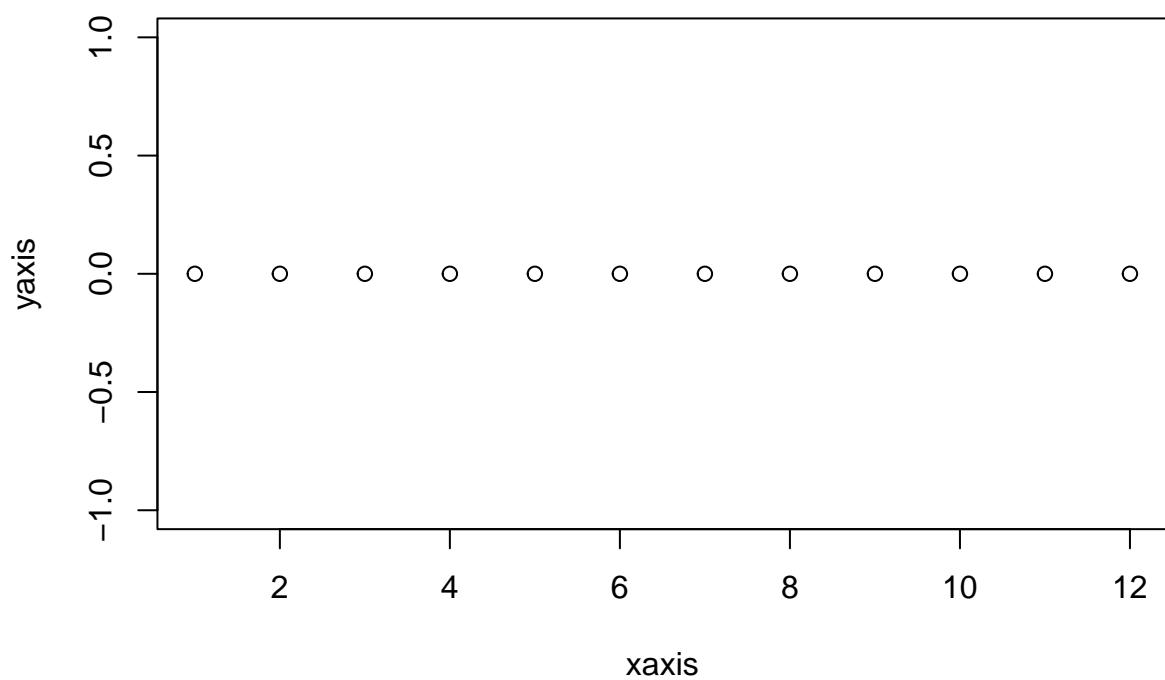


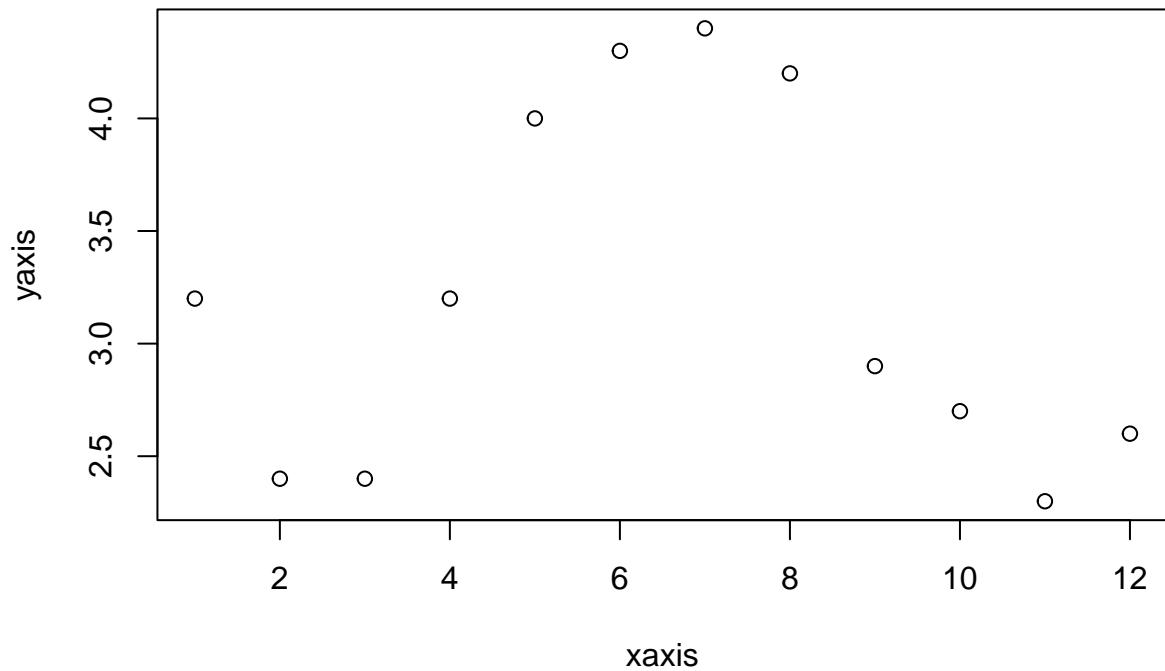




```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 6
## [1] 10.8 17.0
## integer(0)
## [1] 12
## [1] "2023-01-10 18:00:00" "2023-12-03 09:00:00"
## integer(0)
## [1] 1
## [1] 0 0
## integer(0)
## [1] 12
## [1] "2023-01-01 05:00:00" "2023-12-01 08:00:00"
## integer(0)
## [1] 10
## [1] 2.3 4.4
```







```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] "0" "13"
## integer(0)
## [1] 6
## [1] "0" "9"
## integer(0)
## [1] 6
## [1] "0" "6"
## integer(0)
## [1] 5
## [1] "0" "4"
## integer(0)
## [1] 7
## [1] "0" "17"
## integer(0)
## [1] 7
## [1] "0" "8"
## integer(0)
## [1] 4
## [1] "0" "3"
## integer(0)
## [1] 4
## [1] "0" "3"
## integer(0)
```

```

## [1] 7
## [1] " 5" "16"
## integer(0)
## [1] 11
## [1] " 6" "32"
## integer(0)
## [1] 8
## [1] " 4" "18"
## integer(0)
## [1] 9
## [1] " 4" "17"
## integer(0)
## [1] 10
## [1] " 6" "26"
## integer(0)
## [1] 5
## [1] " 2" " 7"
## integer(0)
## [1] 5
## [1] " 1" " 7"
## integer(0)
## [1] 6
## [1] " 0" " 7"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] " 5.6" "14.6"
## integer(0)
## [1] 11
## [1] " 5.5" "14.4"
## integer(0)
## [1] 11
## [1] " 5.2" "14.1"
## integer(0)
## [1] 12
## [1] " 5.1" "13.7"
## integer(0)
## [1] 11
## [1] " 4.9" "13.5"
## integer(0)
## [1] 10
## [1] " 4.7" "13.6"
## integer(0)
## [1] 12
## [1] " 4.8" "14.7"
## integer(0)
## [1] 10
## [1] " 5.1" "16.3"
## integer(0)
## [1] 11
## [1] " 7.4" "18.2"
## integer(0)
## [1] 10
## [1] " 9.8" "20.2"

```

```

## integer(0)
## [1] 12
## [1] "10.9" "22.0"
## integer(0)
## [1] 12
## [1] "12.0" "23.9"
## integer(0)
## [1] 12
## [1] "12.8" "25.4"
## integer(0)
## [1] 12
## [1] "13.1" "26.2"
## integer(0)
## [1] 12
## [1] "13.0" "26.5"
## integer(0)
## [1] 12
## [1] "12.5" "25.5"
## integer(0)
## [1] 12
## [1] "11.2" "24.1"
## integer(0)
## [1] 11
## [1] " 9.8" "22.3"
## integer(0)
## [1] 12
## [1] " 8.9" "20.0"
## integer(0)
## [1] 12
## [1] " 8.3" "17.9"
## integer(0)
## [1] 12
## [1] " 7.8" "16.7"
## integer(0)
## [1] 12
## [1] " 7.1" "15.9"
## integer(0)
## [1] 11
## [1] " 6.5" "15.2"
## integer(0)
## [1] 11
## [1] " 6.4" "14.7"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] " 8.2" "9.1"
## integer(0)
## [1] 11
## [1] " 7.9" "8.7"
## integer(0)
## [1] 12
## [1] " 7.7" "8.5"
## integer(0)
## [1] 11

```

```

## [1] " 7.7" "8.3"
## integer(0)
## [1] 11
## [1] " 7.6" "8.3"
## integer(0)
## [1] 12
## [1] " 7.6" "9.4"
## integer(0)
## [1] 11
## [1] " 8.8" "9.6"
## integer(0)
## [1] 12
## [1] " 9.5" "9.5"
## integer(0)
## [1] 10
## [1] " 9.7" "9.6"
## integer(0)
## [1] 10
## [1] "10.0" "9.7"
## integer(0)
## [1] 11
## [1] "10.4" "9.8"
## integer(0)
## [1] 12
## [1] "10.8" "9.8"
## integer(0)
## [1] 12
## [1] "11.2" "9.5"
## integer(0)
## [1] 11
## [1] "11.4" "9.7"
## integer(0)
## [1] 12
## [1] "11.1" "9.8"
## integer(0)
## [1] 12
## [1] "10.8" "9.6"
## integer(0)
## [1] 12
## [1] "10.5" "9.5"
## integer(0)
## [1] 10
## [1] "10.2" "9.4"
## integer(0)
## [1] 12
## [1] " 9.8" "9.5"
## integer(0)
## [1] 11
## [1] " 9.4" "9.8"
## integer(0)
## [1] 12
## [1] " 9.6" "9.6"
## integer(0)
## [1] 12

```

```

## [1] " 9.2" "9.5"
## integer(0)
## [1] 12
## [1] " 8.8" "9.4"
## integer(0)
## [1] 12
## [1] " 8.6" "9.2"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 7
## [1] "80" "93"
## integer(0)
## [1] 8
## [1] "81" "94"
## integer(0)
## [1] 8
## [1] "81" "95"
## integer(0)
## [1] 9
## [1] "82" "96"
## integer(0)
## [1] 8
## [1] "82" "96"
## integer(0)
## [1] 9
## [1] "82" "96"
## integer(0)
## [1] 7
## [1] "74" "94"
## integer(0)
## [1] 10
## [1] "66" "87"
## integer(0)
## [1] 10
## [1] "59" "85"
## integer(0)
## [1] 10
## [1] "53" "79"
## integer(0)
## [1] 10
## [1] "48" "75"
## integer(0)
## [1] 12
## [1] "45" "71"
## integer(0)
## [1] 9
## [1] "40" "68"
## integer(0)
## [1] 11
## [1] "40" "65"
## integer(0)
## [1] 12
## [1] "40" "65"
## integer(0)

```

```

## [1] 9
## [1] "42" "68"
## integer(0)
## [1] 9
## [1] "44" "72"
## integer(0)
## [1] 11
## [1] "49" "78"
## integer(0)
## [1] 9
## [1] "57" "79"
## integer(0)
## [1] 7
## [1] "66" "80"
## integer(0)
## [1] 10
## [1] "71" "86"
## integer(0)
## [1] 9
## [1] "74" "87"
## integer(0)
## [1] 8
## [1] "77" "89"
## integer(0)
## [1] 9
## [1] "78" "92"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 2
## [1] " 0" " 7"
## integer(0)
## [1] 7
## [1] " 0" "426"
## integer(0)
## [1] 10
## [1] " 0" "589"
## integer(0)
## [1] 12
## [1] "109" "701"
## integer(0)
## [1] 12
## [1] "272" "781"

```

```

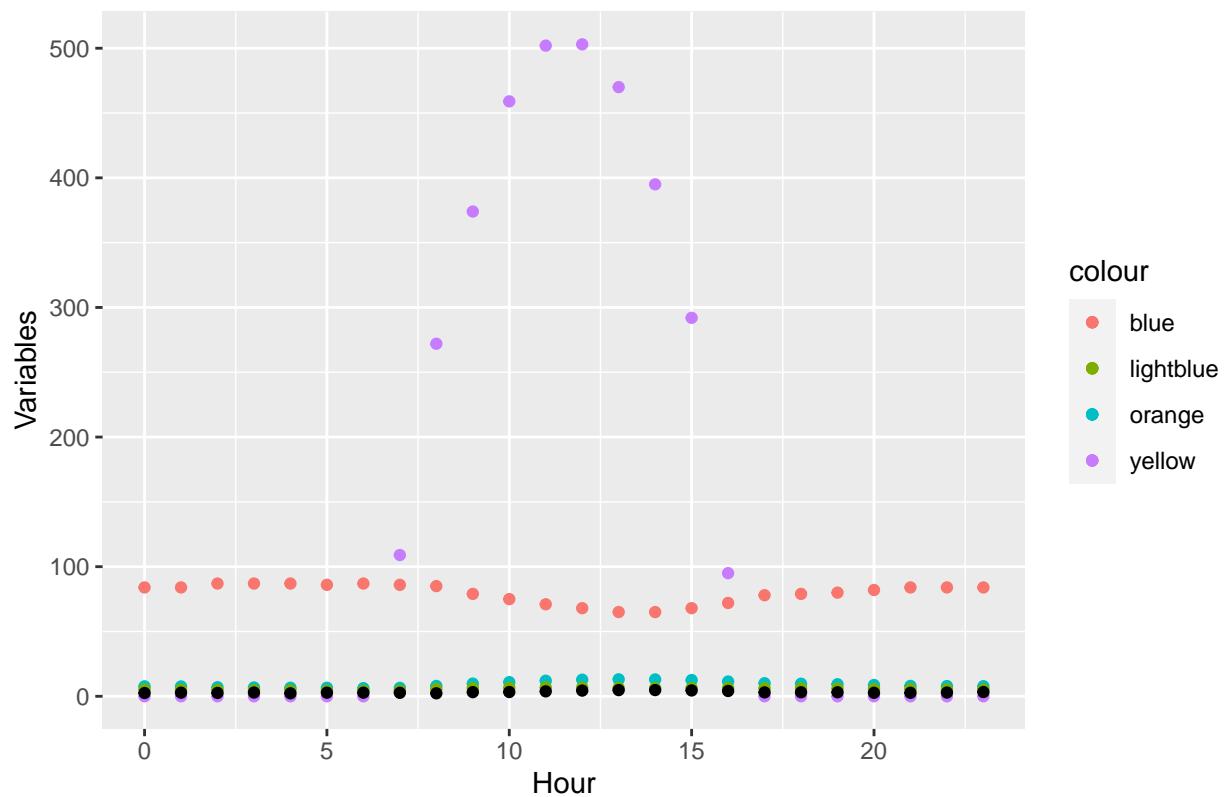
## integer(0)
## [1] 12
## [1] "374" "824"
## integer(0)
## [1] 12
## [1] "459" "863"
## integer(0)
## [1] 12
## [1] "502" "884"
## integer(0)
## [1] 12
## [1] "503" "883"
## integer(0)
## [1] 12
## [1] "470" "866"
## integer(0)
## [1] 12
## [1] "395" "835"
## integer(0)
## [1] 12
## [1] "253" "776"
## integer(0)
## [1] 12
## [1] " 0" "693"
## integer(0)
## [1] 8
## [1] " 0" "540"
## integer(0)
## [1] 4
## [1] " 0" "220"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] "1.0" "3.5"
## integer(0)
## [1] 10
## [1] "1.2" "3.2"
## integer(0)
## [1] 8

```

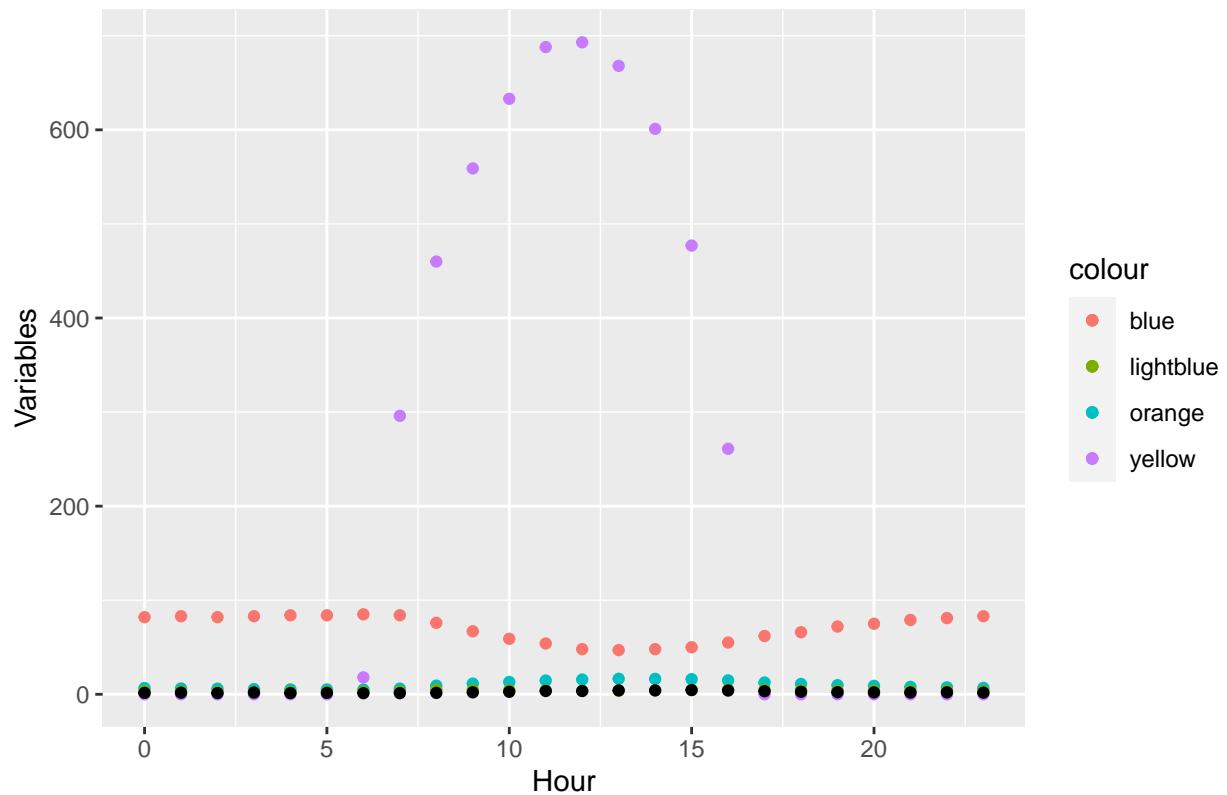
```
## [1] "1.3" "3.1"
## integer(0)
## [1] 9
## [1] "1.4" "2.9"
## integer(0)
## [1] 8
## [1] "1.0" "2.4"
## integer(0)
## [1] 10
## [1] "1.1" "2.8"
## integer(0)
## [1] 8
## [1] "1.2" "2.8"
## integer(0)
## [1] 11
## [1] "1.1" "3.1"
## integer(0)
## [1] 10
## [1] "1.2" "3.5"
## integer(0)
## [1] 10
## [1] "1.9" "4.1"
## integer(0)
## [1] 10
## [1] "2.6" "4.7"
## integer(0)
## [1] 9
## [1] "3.2" "5.2"
## integer(0)
## [1] 10
## [1] "3.4" "5.9"
## integer(0)
## [1] 10
## [1] "3.4" "6.5"
## integer(0)
## [1] 10
## [1] "3.6" "7.1"
## integer(0)
## [1] 10
## [1] "3.8" "7.2"
## integer(0)
## [1] 11
## [1] "3.1" "7.0"
## integer(0)
## [1] 9
## [1] "2.8" "6.7"
## integer(0)
## [1] 10
## [1] "2.3" "5.7"
## integer(0)
## [1] 11
## [1] "1.8" "5.1"
## integer(0)
## [1] 7
```

```
## [1] "1.8" "4.4"
## integer(0)
## [1] 10
## [1] "1.4" "4.2"
## integer(0)
## [1] 11
## [1] "1.4" "4.1"
## integer(0)
## [1] 10
## [1] "1.4" "3.8"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "Apr" "Sep"
## integer(0)
## [1] 24
## [1] 0 23
## integer(0)
## [1] 156
## [1] 4.7 26.5
## integer(0)
## [1] 108
## [1] 1.9 14.0
## integer(0)
## [1] 56
## [1] 40 96
## integer(0)
## [1] 130
## [1] 0 884
## integer(0)
## [1] 60
## [1] 1.0 7.2
```

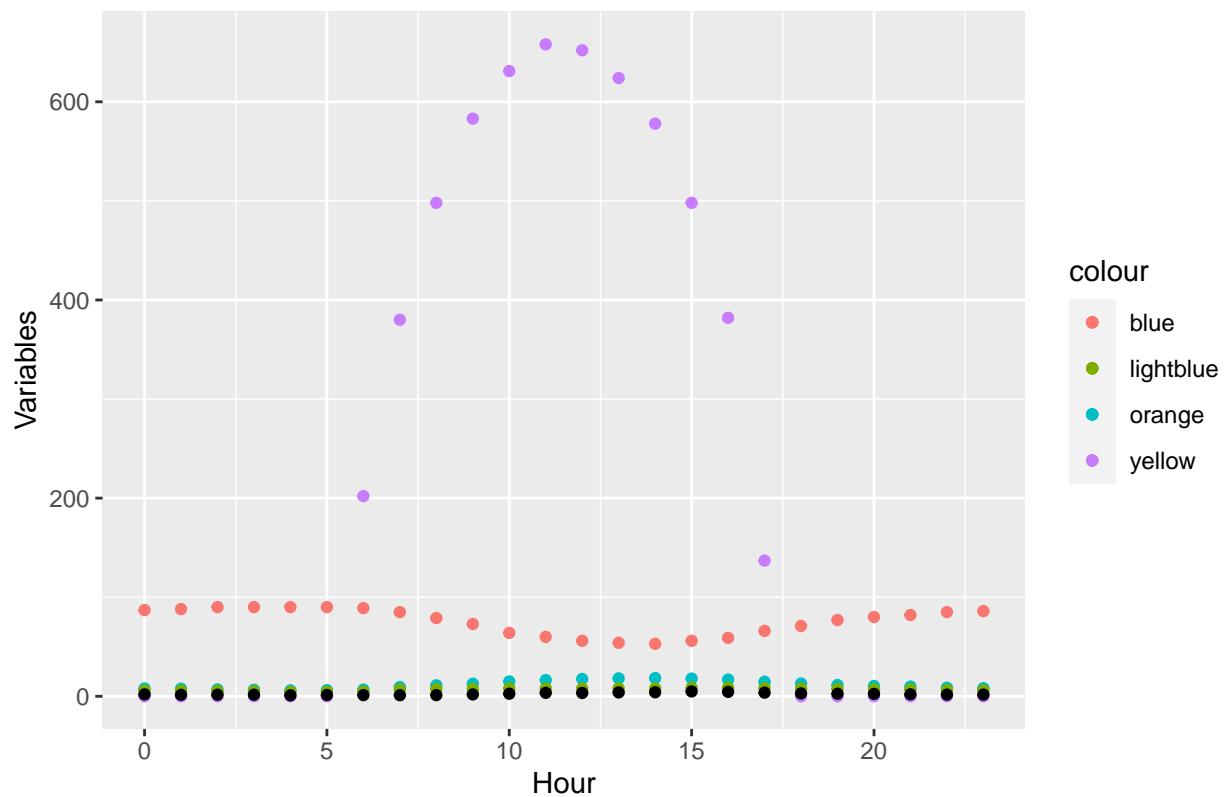
Jan Data: USA\_CA\_Napa.County.AP



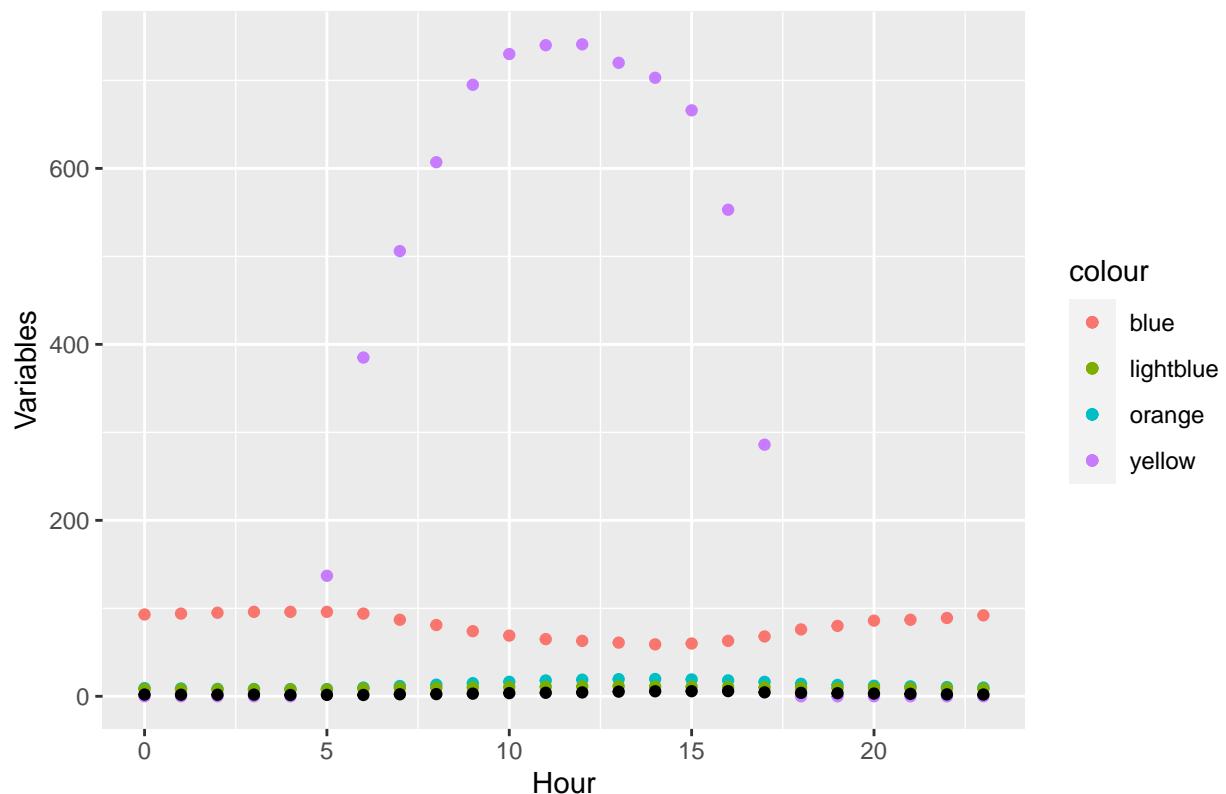
Feb Data: USA\_CA\_Napa.County.AP



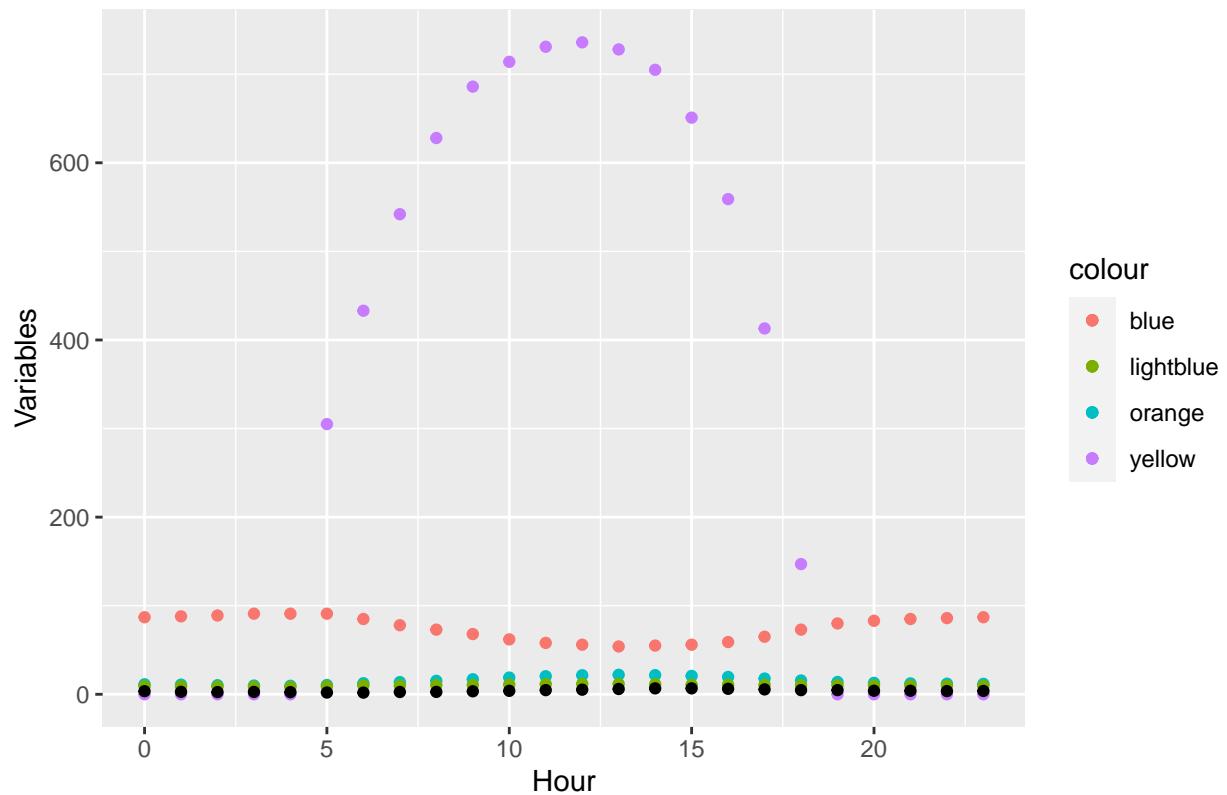
Mar Data: USA\_CA\_Napa.County.AP



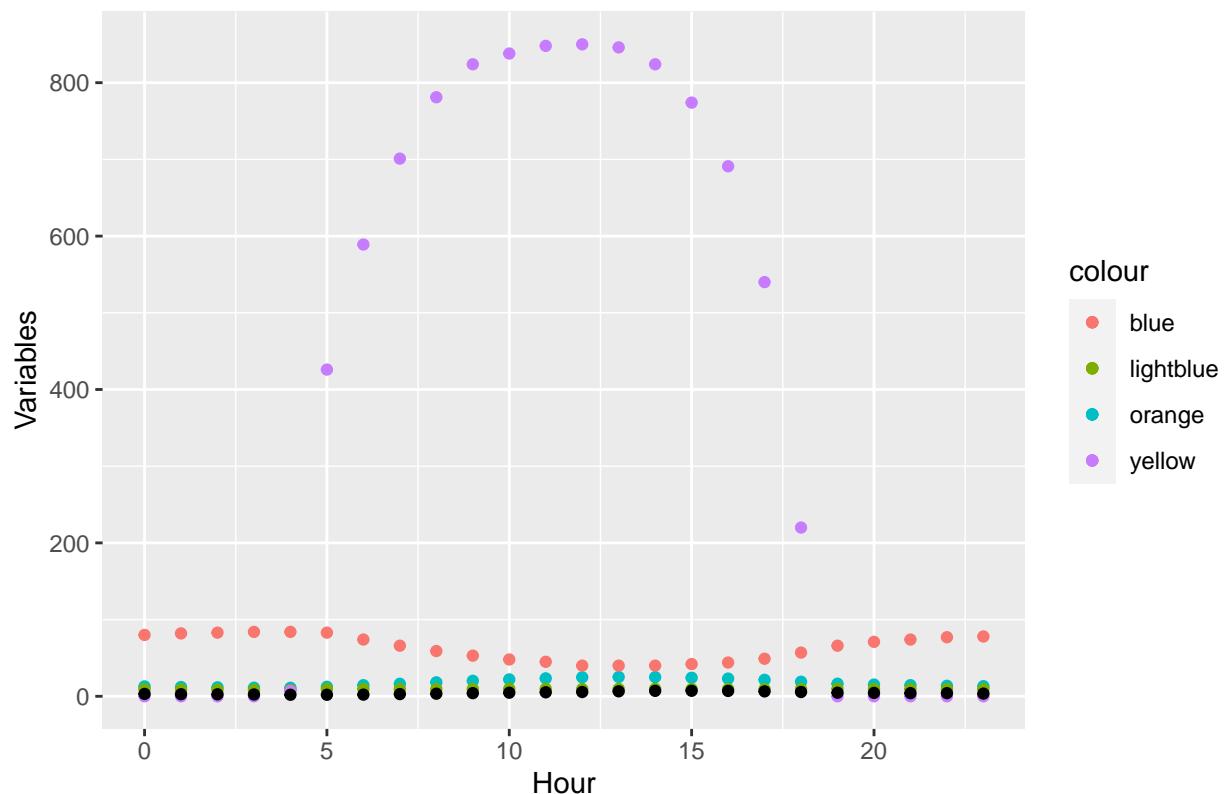
Apr Data: USA\_CA\_Napa.County.AP



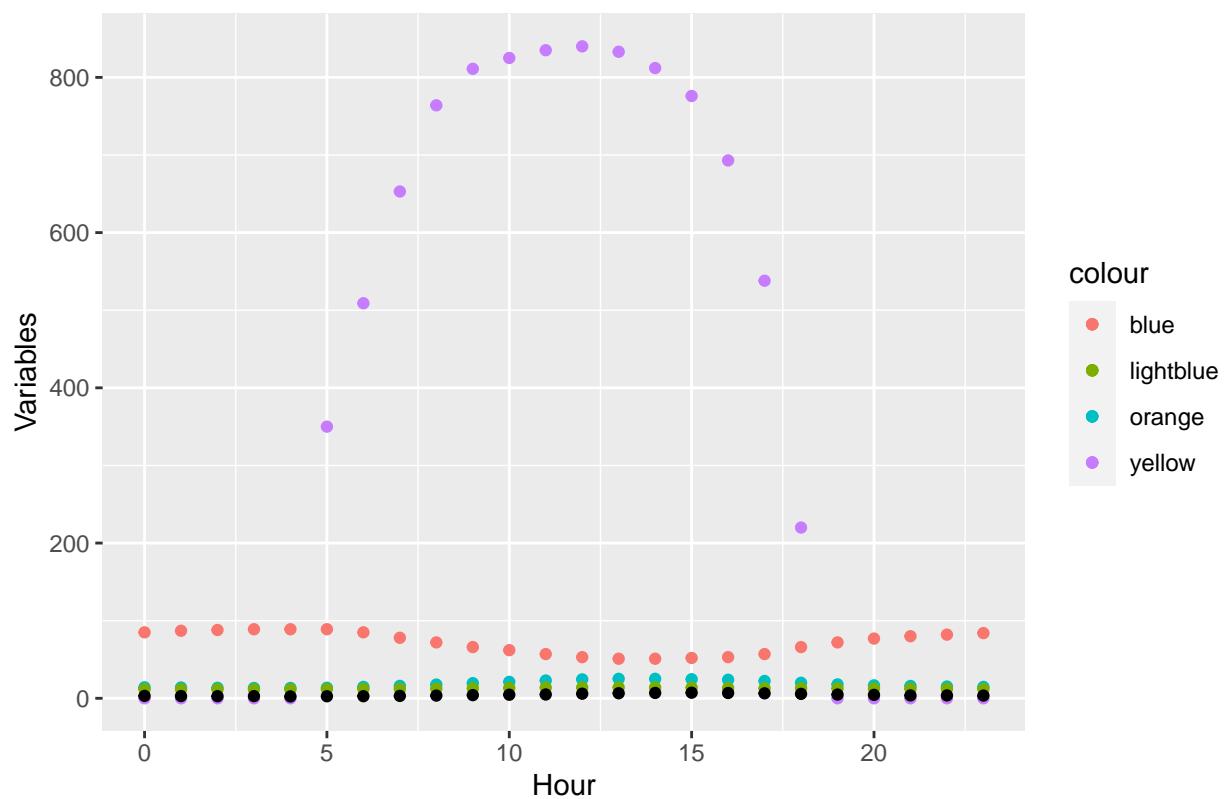
May Data: USA\_CA\_Napa.County.AP



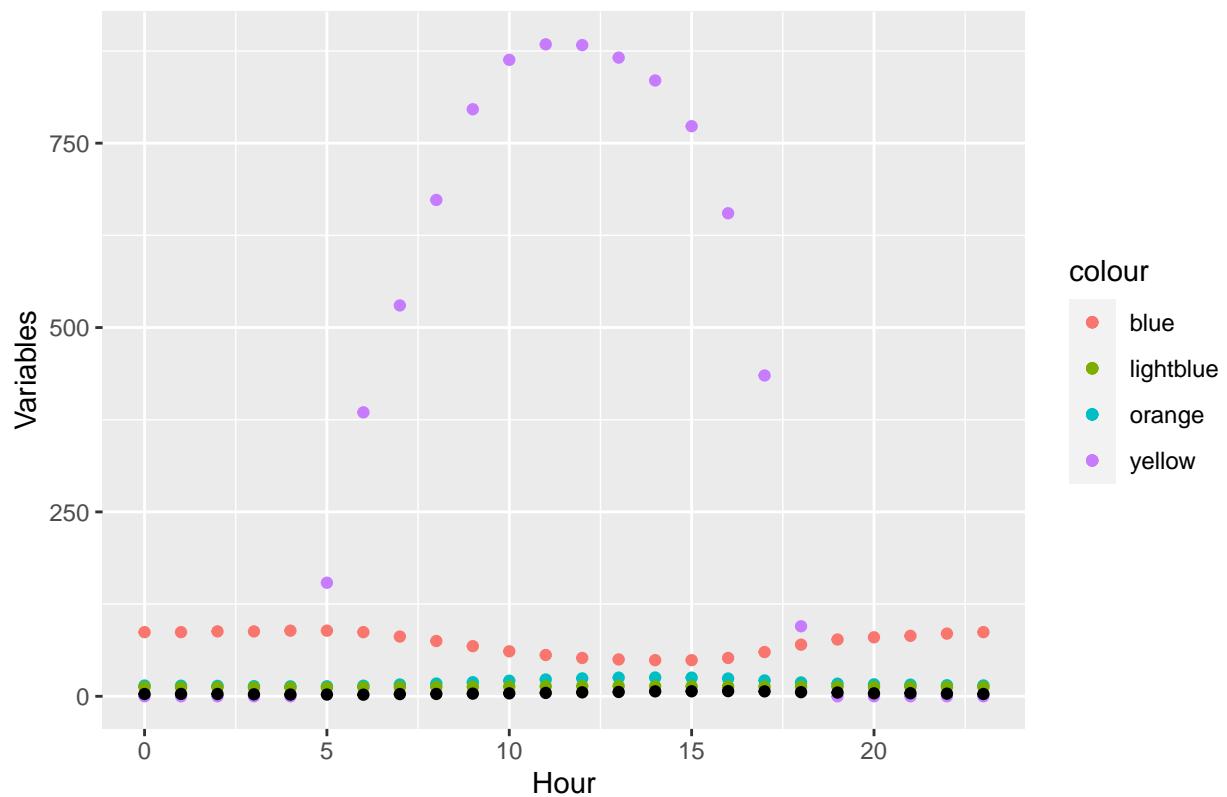
Jun Data: USA\_CA\_Napa.County.AP



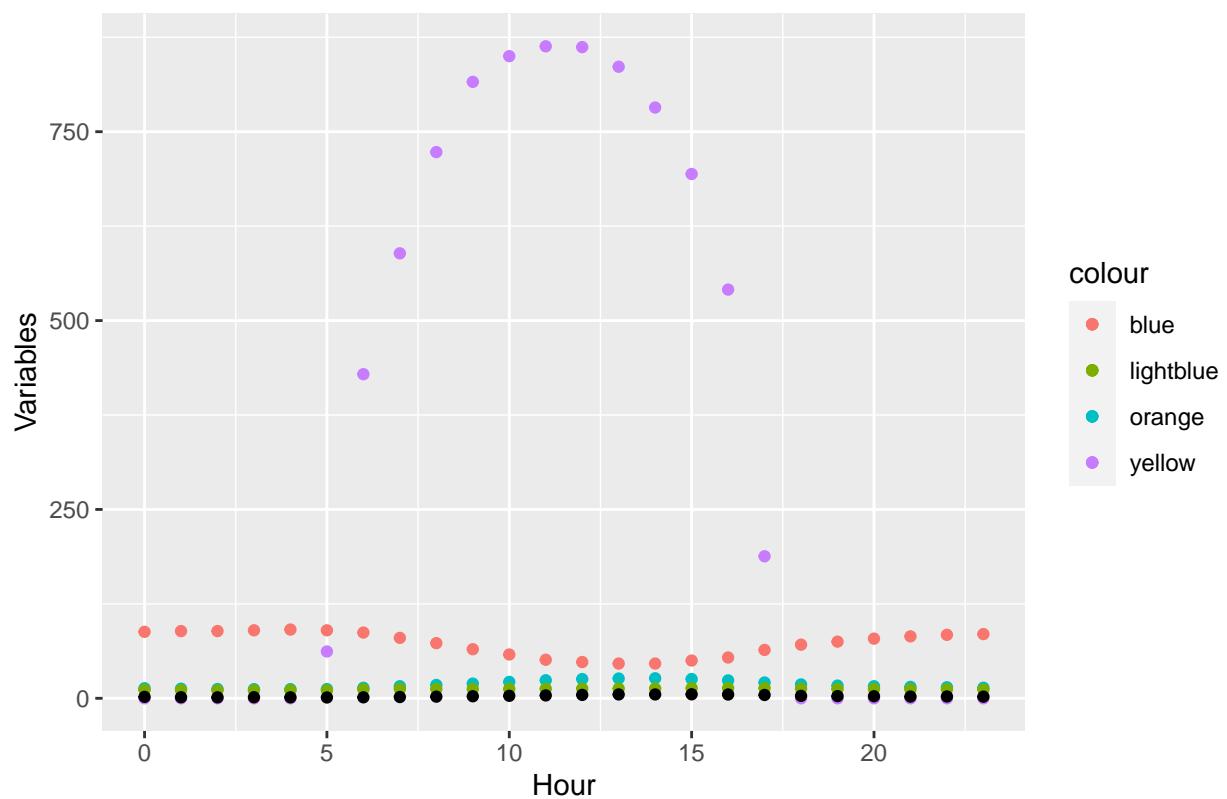
Jul Data: USA\_CA\_Napa.County.AP



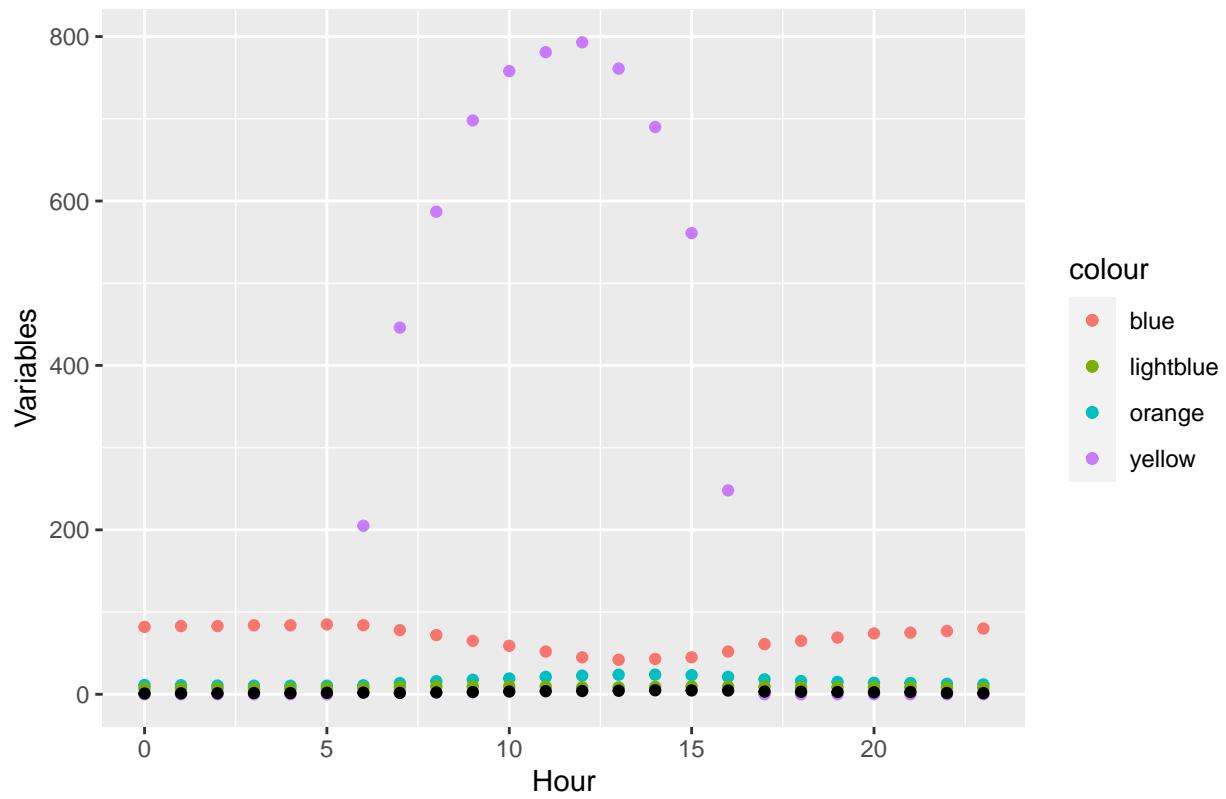
Aug Data: USA\_CA\_Napa.County.AP



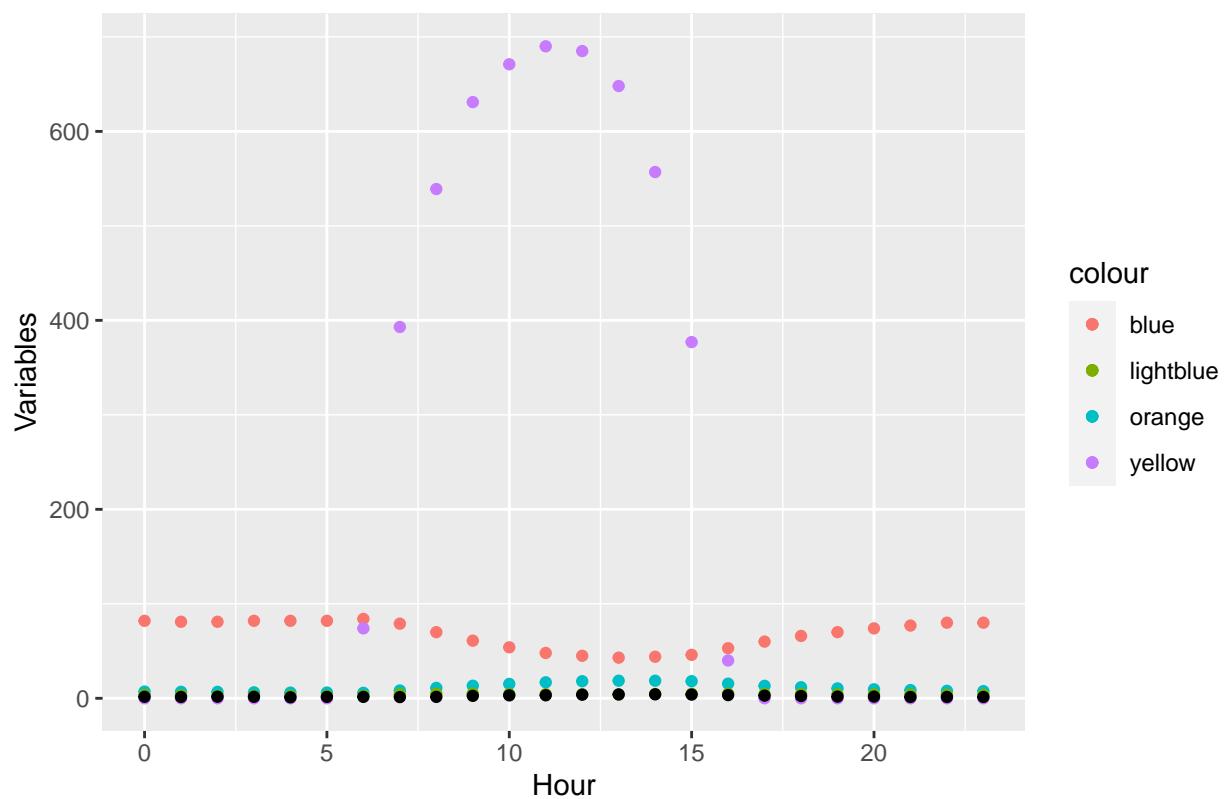
Sep Data: USA\_CA\_Napa.County.AP



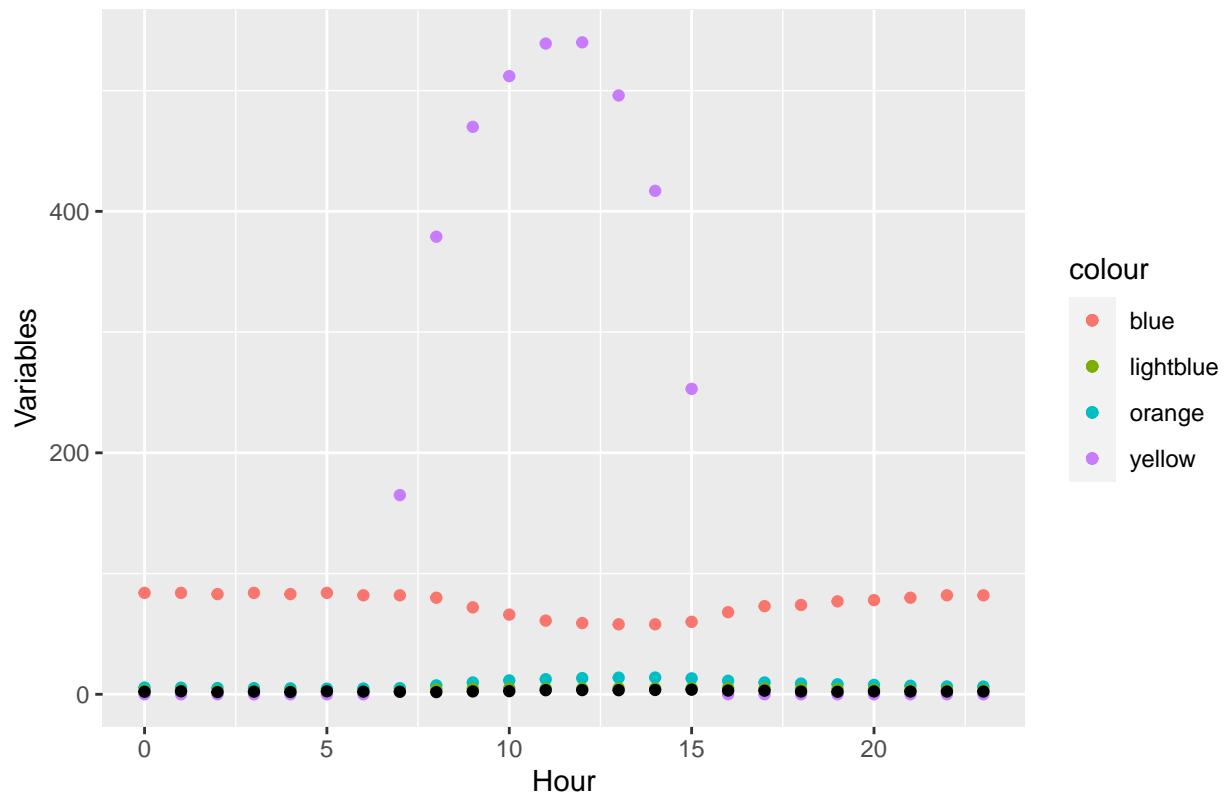
Oct Data: USA\_CA\_Napa.County.AP



Nov Data: USA\_CA\_Napa.County.AP



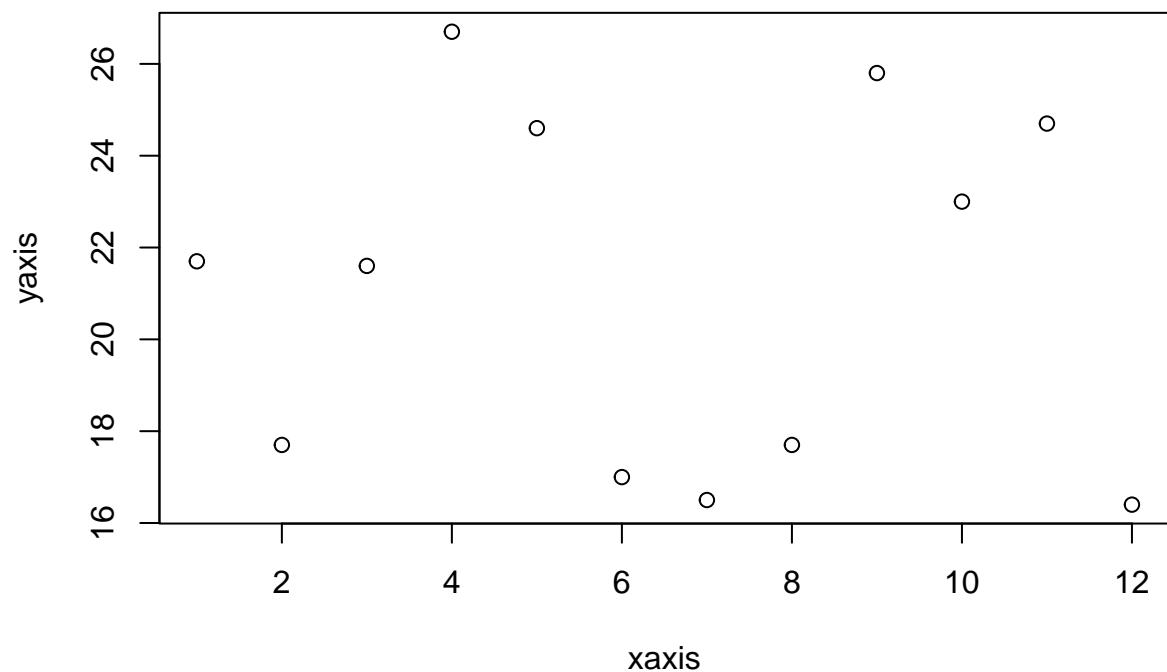
## Dec Data: USA\_CA\_Napa.County.AP

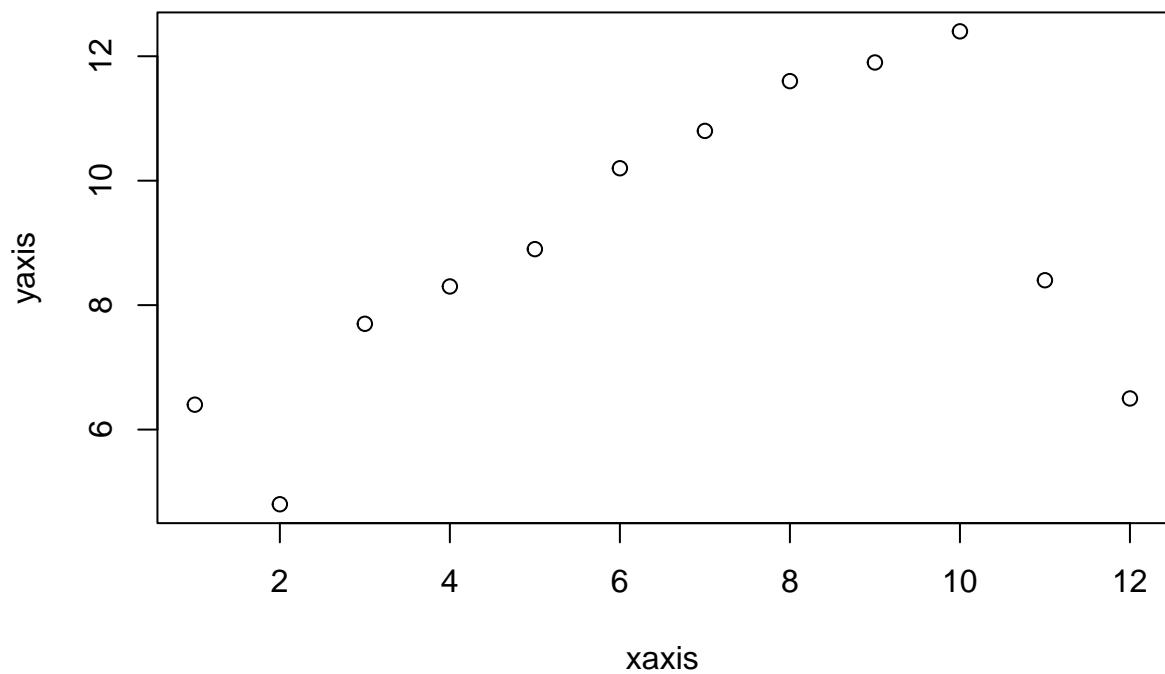


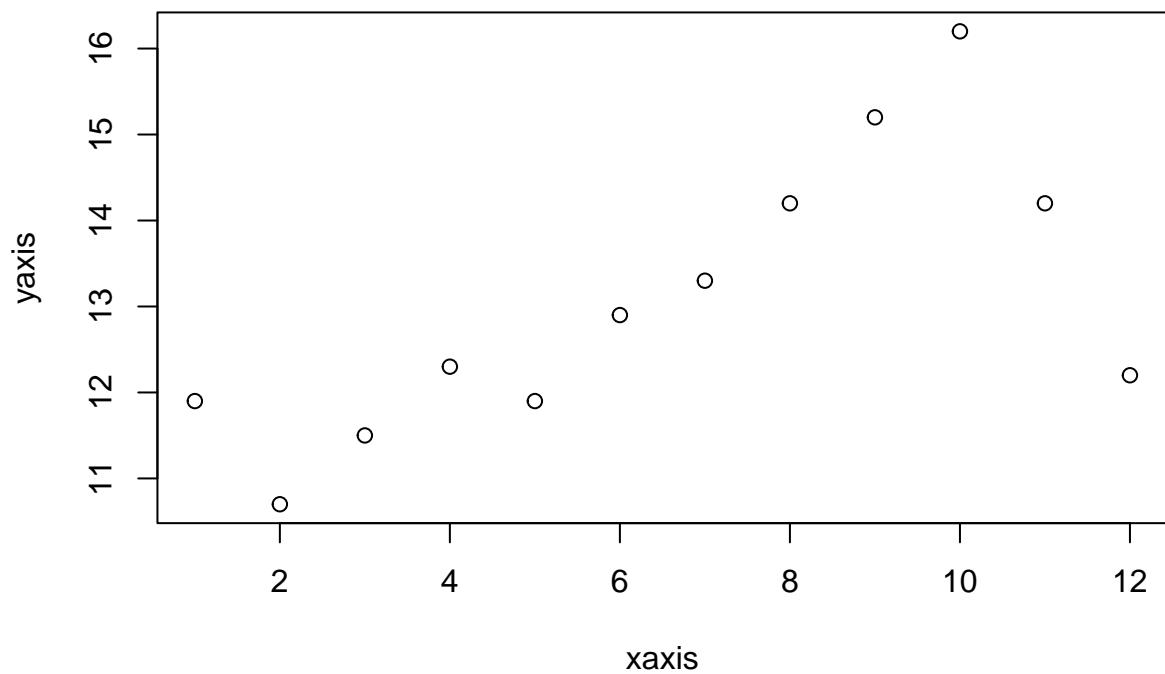
```
stat_reyes = stat_parser('dt/USA_CA_Point.Reyes.Lighthouse.724959_TMYx.2007-2021.stat')
```

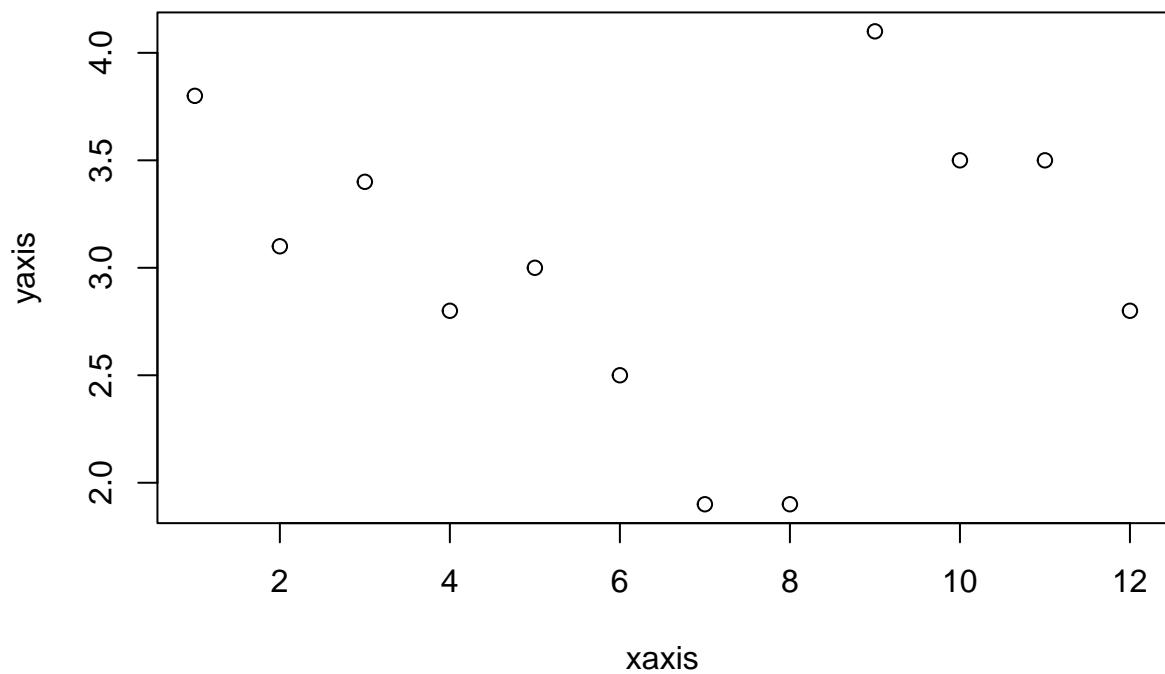
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] 16.4 26.7
## integer(0)
## [1] 12
## [1] "2023-01-13 16:00:00" "2023-12-03 05:00:00"
## integer(0)
## [1] 12
## [1] 4.8 12.4
## integer(0)
## [1] 12
## [1] "2023-01-17 09:00:00" "2023-12-20 08:00:00"
## integer(0)
## [1] 10
## [1] 10.7 16.2
## integer(0)
## [1] 9
## [1] 1.9 4.1
## integer(0)
## [1] 10
## [1] 16.3 26.7
## integer(0)
```

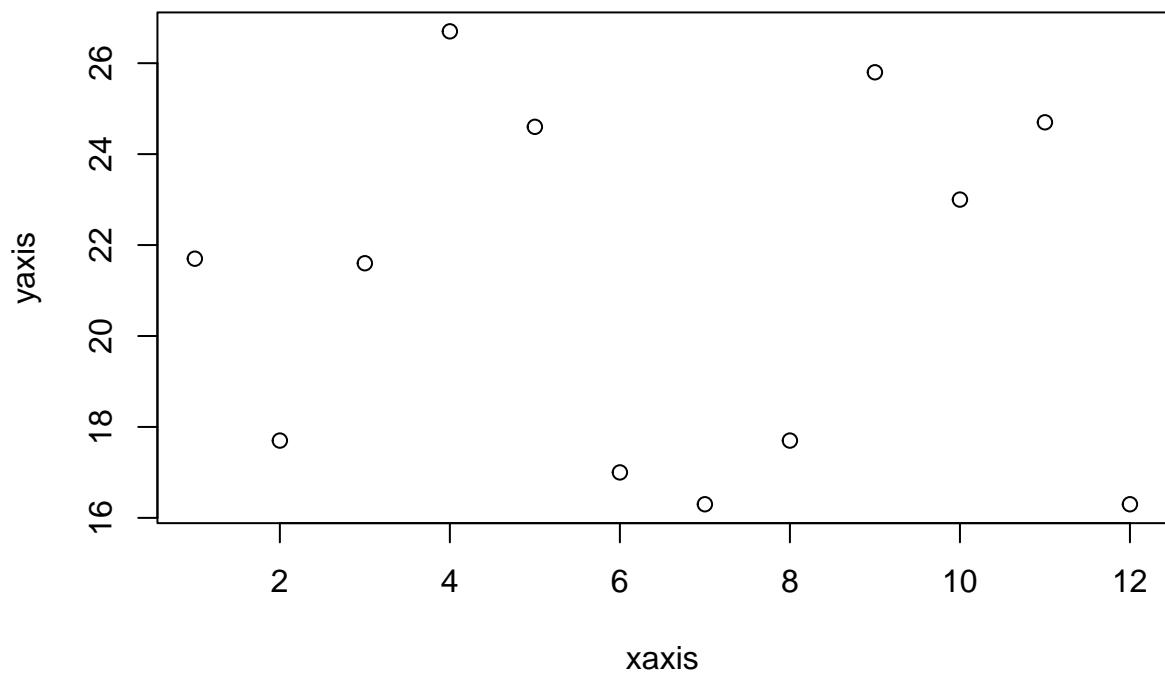
```
## [1] 12
## [1] 4.8 12.4
## integer(0)
## [1] 11
## [1] 11.0 16.5
## integer(0)
## [1] 11
## [1] 15.3 21.2
## integer(0)
## [1] 12
## [1] 5.2 12.5
## integer(0)
## [1] 12
## [1] 10.4 16.0
```

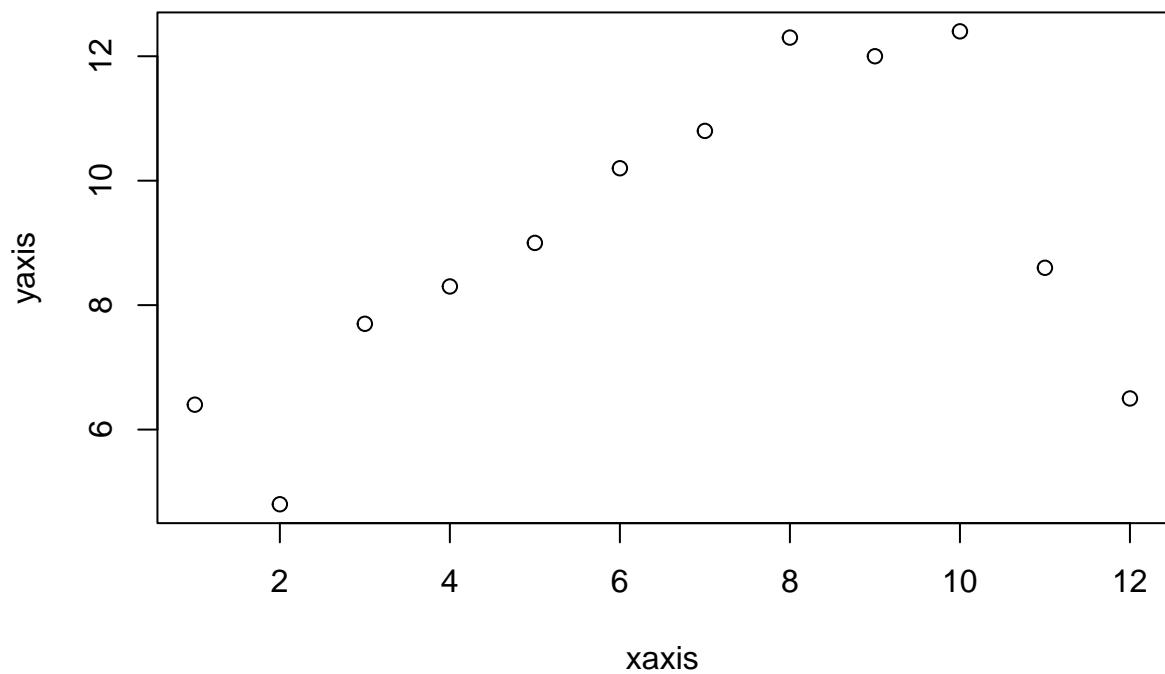


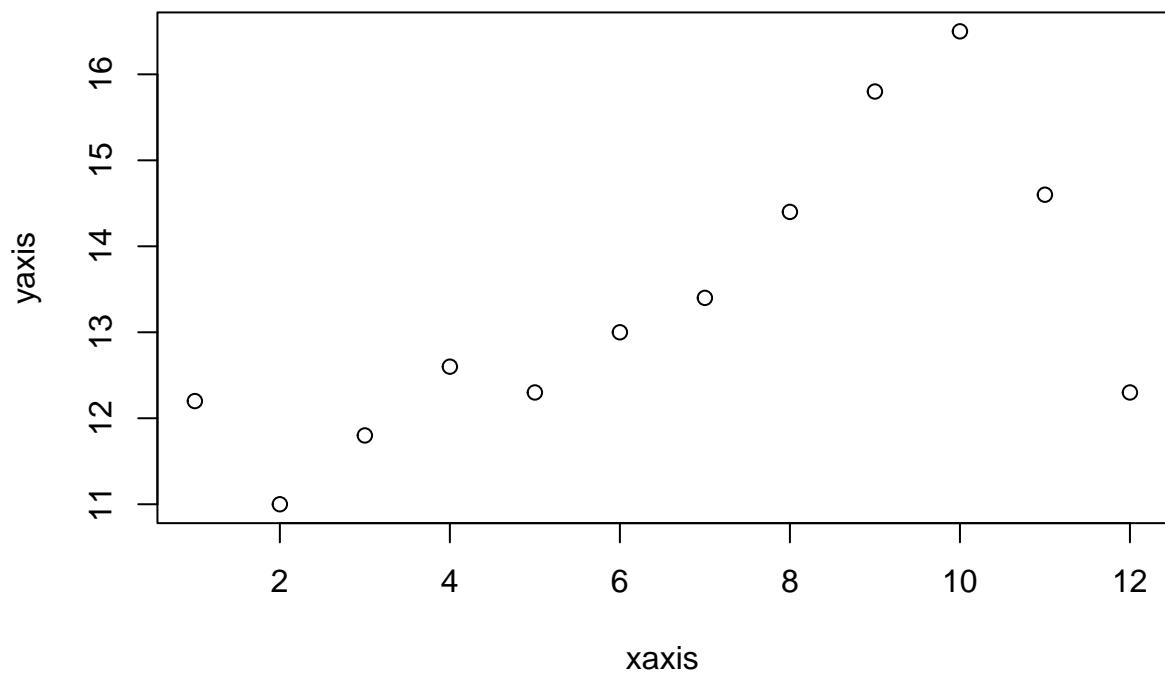


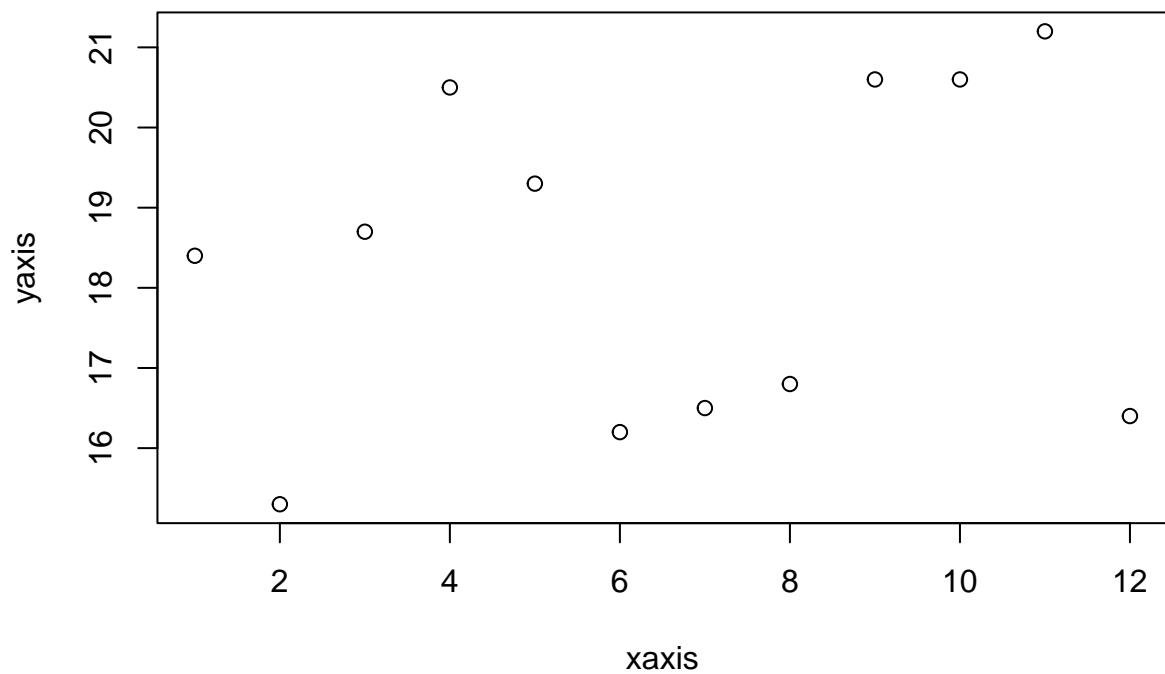


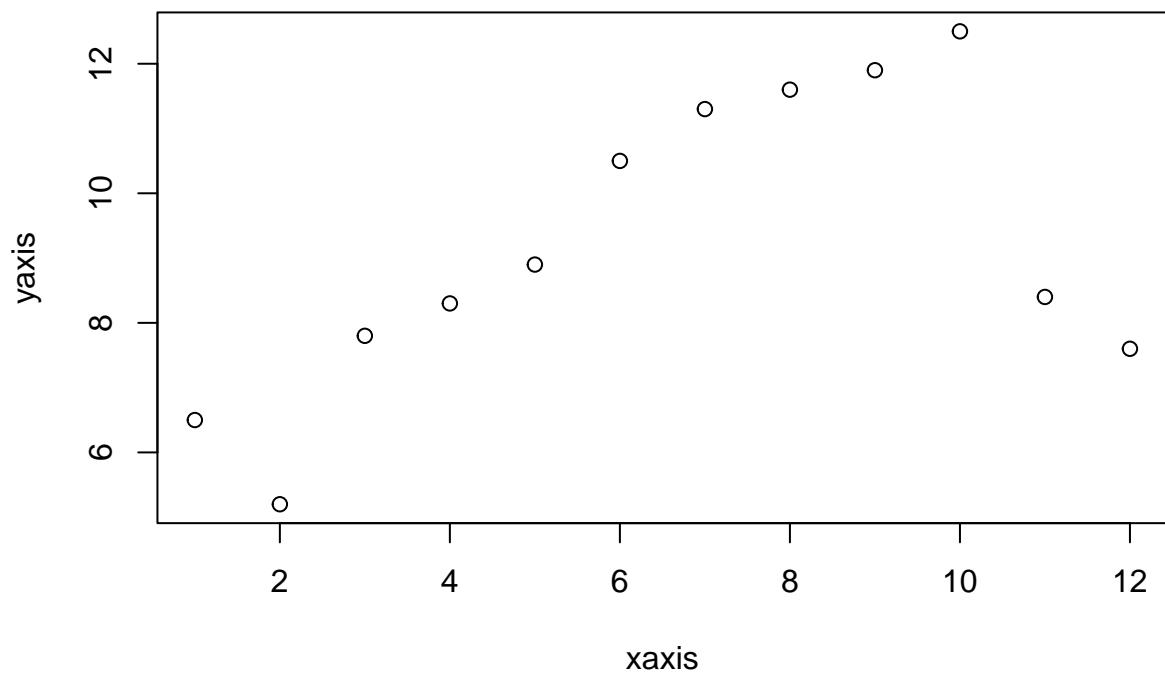


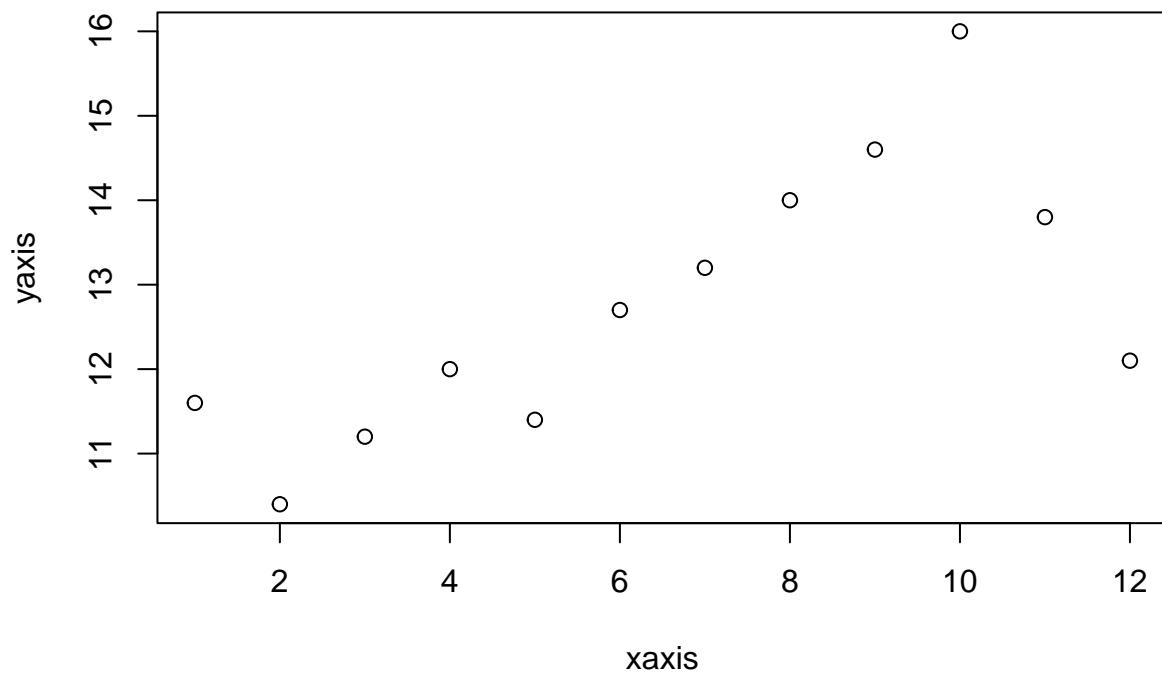




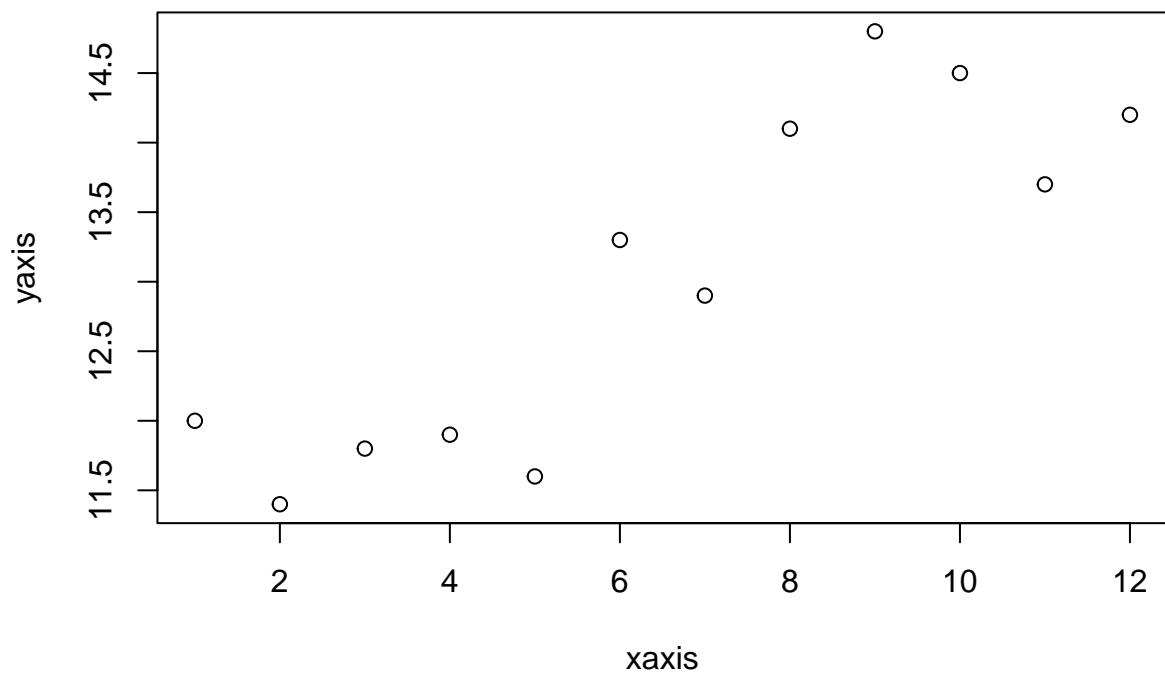


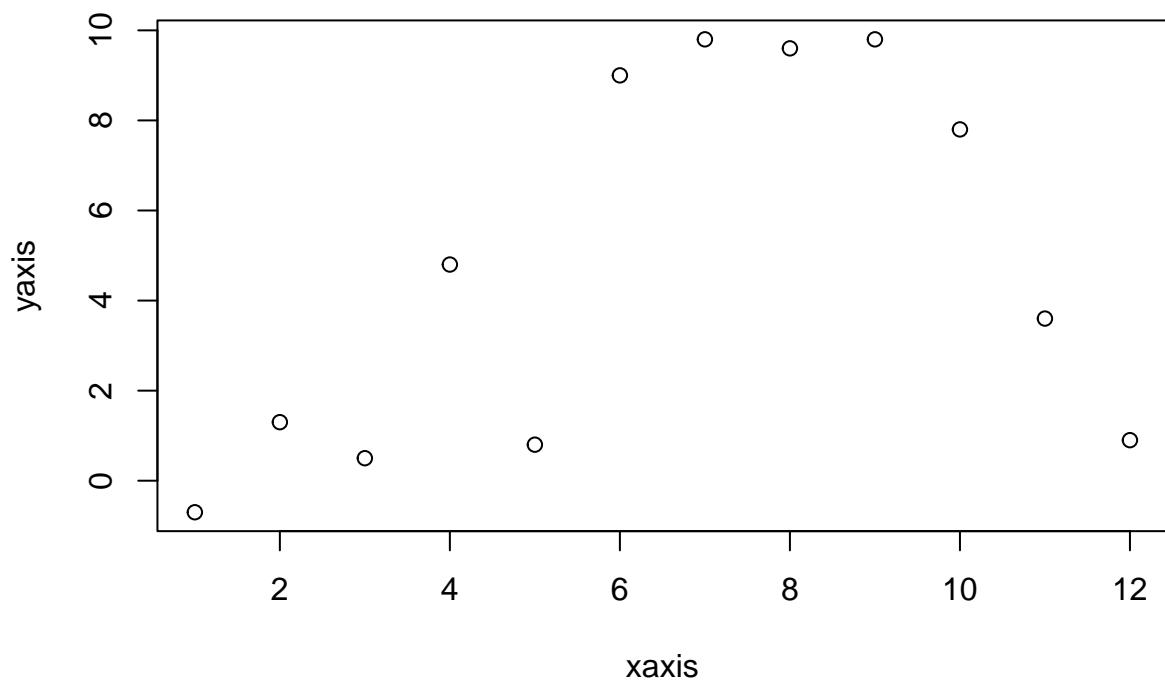


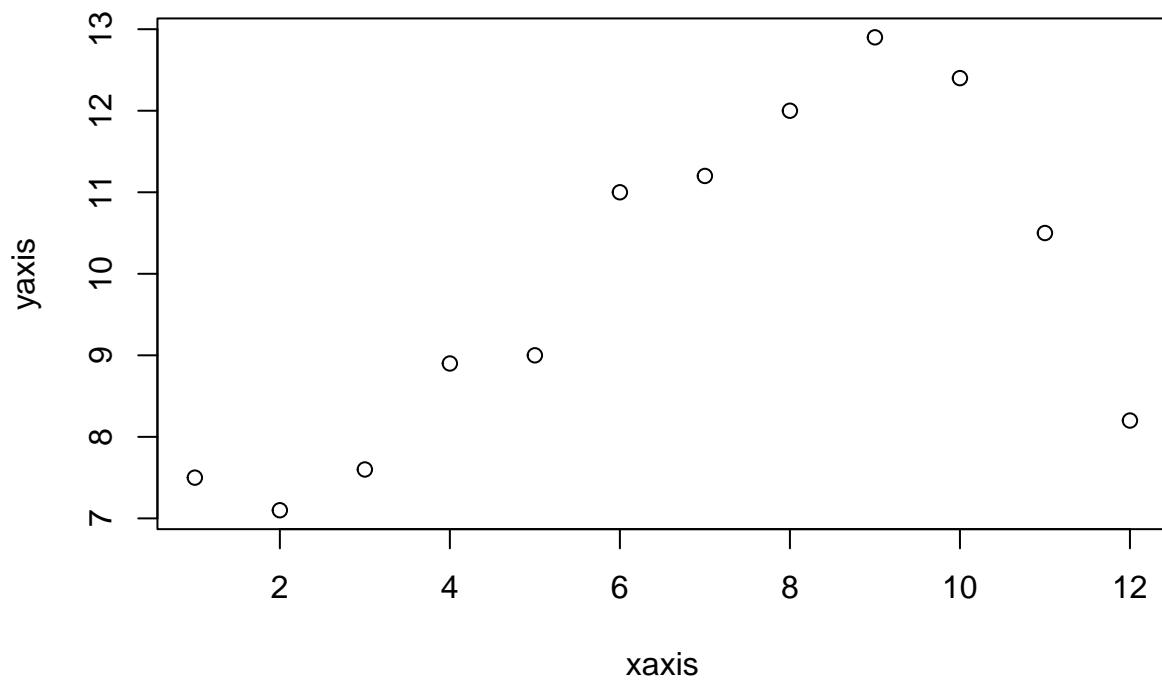




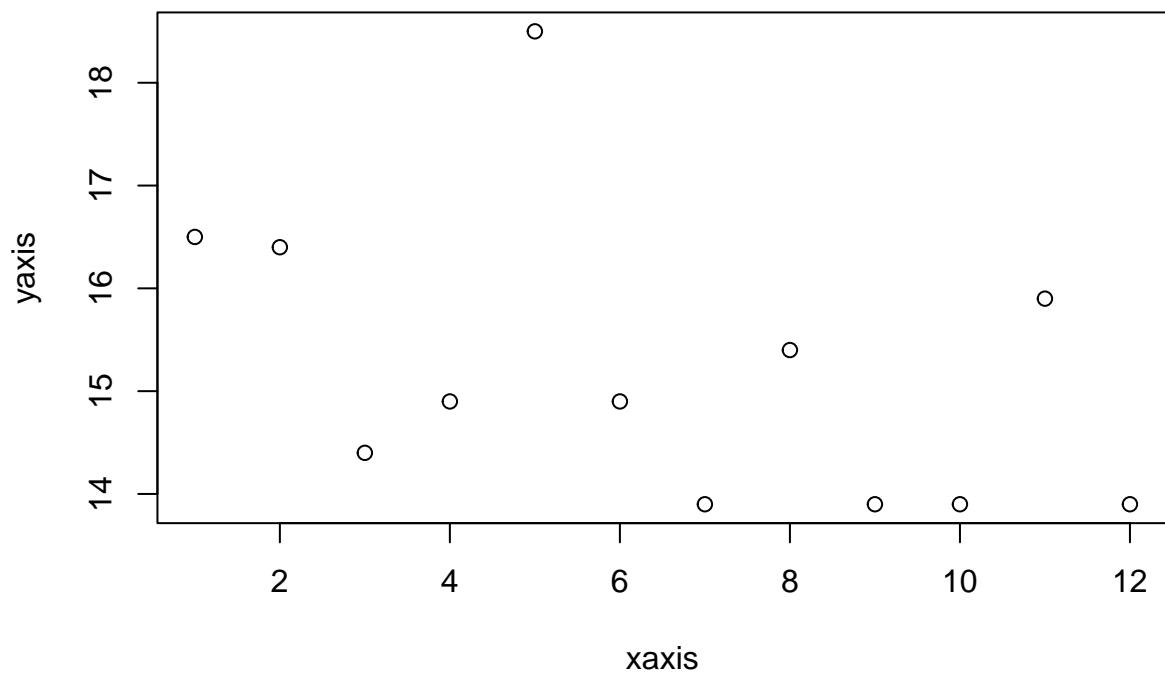
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] 11.4 14.8
## integer(0)
## [1] 12
## [1] "2023-01-20 22:00:00" "2023-12-03 01:00:00"
## integer(0)
## [1] 11
## [1] -0.7  9.8
## integer(0)
## [1] 12
## [1] "2023-01-16 13:00:00" "2023-12-20 15:00:00"
## integer(0)
## [1] 12
## [1]  7.1 12.9
```

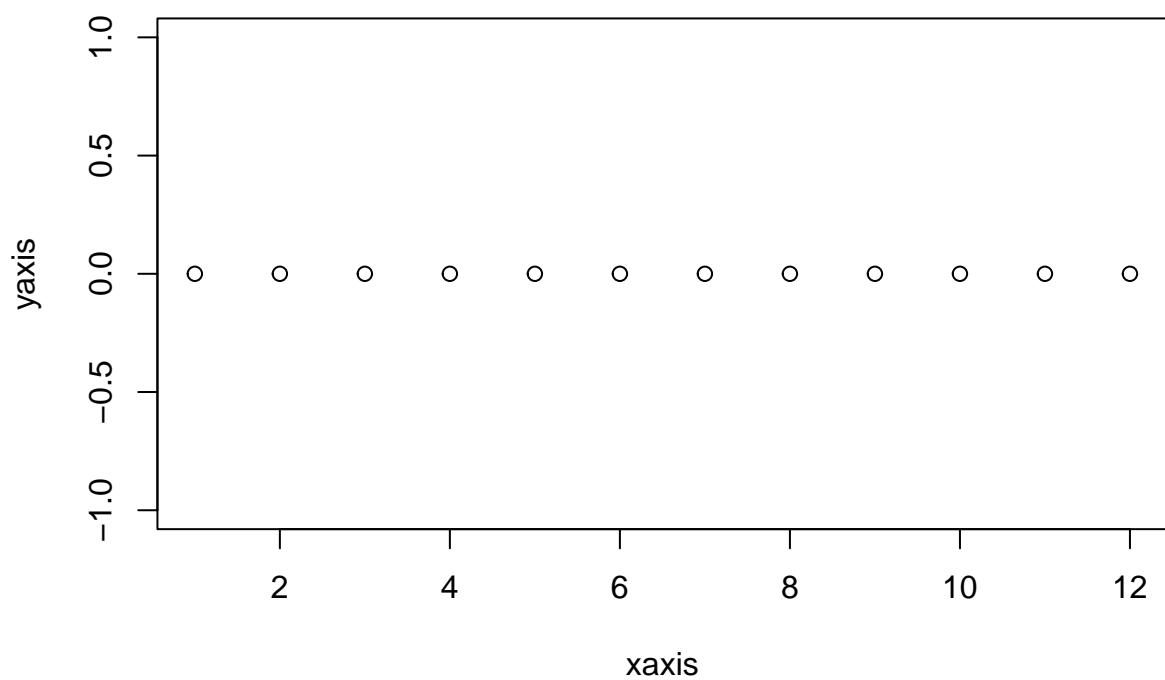


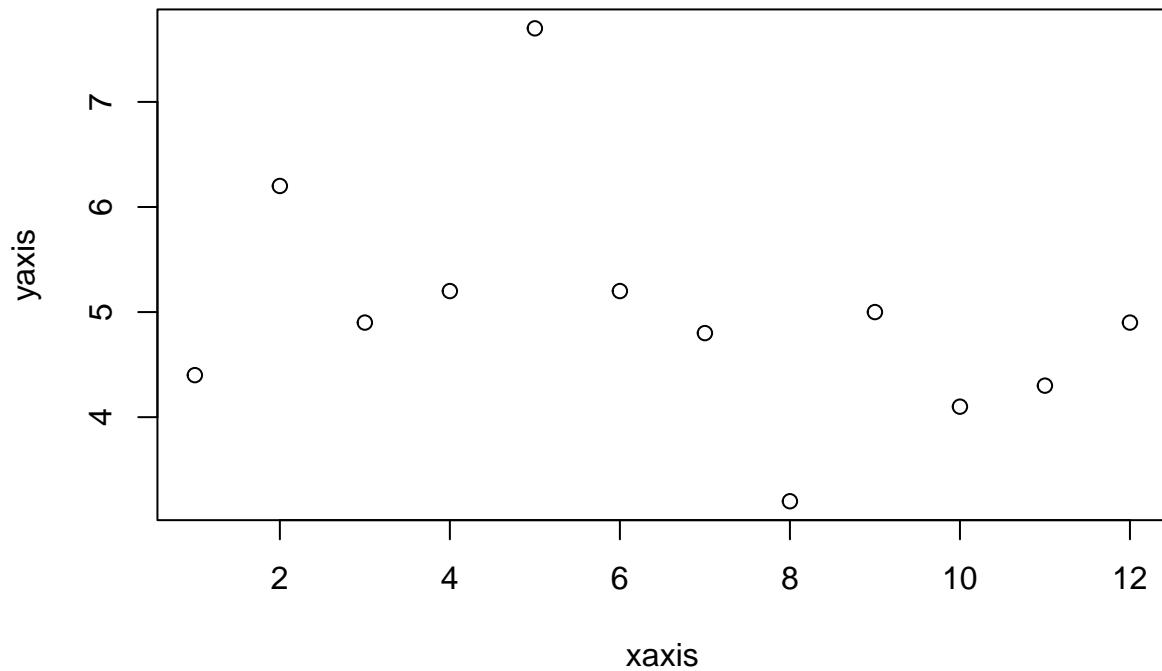




```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] 13.9 18.5
## integer(0)
## [1] 12
## [1] "2023-01-15 20:00:00" "2023-12-19 16:00:00"
## integer(0)
## [1] 1
## [1] 0 0
## integer(0)
## [1] 12
## [1] "2023-01-01 15:00:00" "2023-12-04 03:00:00"
## integer(0)
## [1] 10
## [1] 3.2 7.7
```







```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 3
## [1] "0" "2"
## integer(0)
## [1] 2
## [1] "0" "1"
## integer(0)
## [1] 3
## [1] "0" "2"
## integer(0)
## [1] 4
## [1] "0" "3"
## integer(0)
## [1] 7
## [1] "1" "8"
## integer(0)
## [1] 9
## [1] "1" "19"
## integer(0)
## [1] 8
## [1] "2" "10"
## integer(0)
## [1] 3
## [1] "1" "3"
## integer(0)
```

```

## [1] 4
## [1] " 1" " 5"
## integer(0)
## [1] 4
## [1] " 0" " 3"
## integer(0)
## [1] 3
## [1] " 0" " 2"
## integer(0)
## [1] 3
## [1] " 0" " 2"
## integer(0)
## [1] 7
## [1] "10" "19"
## integer(0)
## [1] 11
## [1] "19" "53"
## integer(0)
## [1] 8
## [1] "12" "25"
## integer(0)
## [1] 7
## [1] " 0" " 7"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] "10.4" "16.1"
## integer(0)
## [1] 11
## [1] "10.2" "15.9"
## integer(0)
## [1] 12
## [1] "10.2" "15.8"
## integer(0)
## [1] 12
## [1] "10.2" "15.9"
## integer(0)
## [1] 12
## [1] "10.3" "15.7"
## integer(0)
## [1] 11
## [1] "10.2" "15.6"
## integer(0)
## [1] 11
## [1] "10.0" "15.5"
## integer(0)
## [1] 12
## [1] " 9.8" "15.5"
## integer(0)
## [1] 12
## [1] "10.0" "15.8"
## integer(0)
## [1] 12
## [1] "10.6" "16.4"

```

```

## integer(0)
## [1] 11
## [1] "10.8" "16.6"
## integer(0)
## [1] 12
## [1] "11.2" "16.9"
## integer(0)
## [1] 11
## [1] "11.7" "17.1"
## integer(0)
## [1] 11
## [1] "11.7" "16.9"
## integer(0)
## [1] 12
## [1] "11.6" "17.1"
## integer(0)
## [1] 12
## [1] "11.5" "16.8"
## integer(0)
## [1] 12
## [1] "11.5" "16.8"
## integer(0)
## [1] 11
## [1] "11.1" "16.5"
## integer(0)
## [1] 12
## [1] "10.8" "16.2"
## integer(0)
## [1] 11
## [1] "10.7" "16.2"
## integer(0)
## [1] 11
## [1] "10.6" "16.2"
## integer(0)
## [1] 12
## [1] "10.6" "16.0"
## integer(0)
## [1] 12
## [1] "10.6" "16.0"
## integer(0)
## [1] 12
## [1] "10.4" "16.0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] "10.7" "9.1"
## integer(0)
## [1] 11
## [1] "10.6" "9.0"
## integer(0)
## [1] 12
## [1] "10.7" "9.0"
## integer(0)
## [1] 12

```

```

## [1] "10.6" "9.0"
## integer(0)
## [1] 11
## [1] "10.5" "8.9"
## integer(0)
## [1] 11
## [1] "10.5" "8.9"
## integer(0)
## [1] 12
## [1] "10.4" "8.9"
## integer(0)
## [1] 12
## [1] "10.4" "8.9"
## integer(0)
## [1] 12
## [1] "10.4" "8.9"
## integer(0)
## [1] 12
## [1] "10.4" "8.8"
## integer(0)
## [1] 12
## [1] "10.3" "8.8"
## integer(0)
## [1] 12
## [1] "10.2" "8.9"
## integer(0)
## [1] 12
## [1] "10.3" "9.1"
## integer(0)
## [1] 11
## [1] "10.5" "9.1"
## integer(0)
## [1] 12
## [1] "10.4" "9.2"
## integer(0)
## [1] 11
## [1] "10.5" "9.2"
## integer(0)
## [1] 10
## [1] "10.6" "9.2"
## integer(0)
## [1] 12
## [1] "10.7" "9.2"
## integer(0)
## [1] 12
## [1] "10.8" "9.2"
## integer(0)
## [1] 12
## [1] "10.8" "9.2"
## integer(0)
## [1] 12
## [1] "10.8" "9.2"
## integer(0)
## [1] 12

```

```

## [1] "10.8" "9.2"
## integer(0)
## [1] 12
## [1] "10.7" "9.3"
## integer(0)
## [1] 12
## [1] "10.7" "9.3"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] "78" "91"
## integer(0)
## [1] 10
## [1] "78" "91"
## integer(0)
## [1] 9
## [1] "78" "92"
## integer(0)
## [1] 9
## [1] "78" "92"
## integer(0)
## [1] 8
## [1] "78" "92"
## integer(0)
## [1] 9
## [1] "79" "93"
## integer(0)
## [1] 9
## [1] "78" "93"
## integer(0)
## [1] 8
## [1] "78" "92"
## integer(0)
## [1] 8
## [1] "77" "92"
## integer(0)
## [1] 8
## [1] "74" "91"
## integer(0)
## [1] 10
## [1] "71" "90"
## integer(0)
## [1] 10
## [1] "70" "89"
## integer(0)
## [1] 11
## [1] "70" "88"
## integer(0)
## [1] 10
## [1] "69" "86"
## integer(0)
## [1] 10
## [1] "69" "84"
## integer(0)

```

```

## [1] 11
## [1] "71" "84"
## integer(0)
## [1] 8
## [1] "74" "84"
## integer(0)
## [1] 8
## [1] "75" "85"
## integer(0)
## [1] 6
## [1] "76" "85"
## integer(0)
## [1] 8
## [1] "77" "87"
## integer(0)
## [1] 8
## [1] "77" "88"
## integer(0)
## [1] 8
## [1] "77" "88"
## integer(0)
## [1] 8
## [1] "78" "89"
## integer(0)
## [1] 7
## [1] "78" "89"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 7
## [1] " 0" "301"
## integer(0)
## [1] 11
## [1] " 0" "473"
## integer(0)
## [1] 12
## [1] "119" "582"
## integer(0)
## [1] 12
## [1] "284" "654"

```

```

## integer(0)
## [1] 12
## [1] "384" "708"
## integer(0)
## [1] 12
## [1] "461" "755"
## integer(0)
## [1] 12
## [1] "501" "783"
## integer(0)
## [1] 12
## [1] "512" "795"
## integer(0)
## [1] 12
## [1] "470" "791"
## integer(0)
## [1] 12
## [1] "392" "780"
## integer(0)
## [1] 12
## [1] "294" "737"
## integer(0)
## [1] 12
## [1] " 0" "650"
## integer(0)
## [1] 8
## [1] " 0" "515"
## integer(0)
## [1] 5
## [1] " 0" "215"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] "3.1" "8.1"
## integer(0)
## [1] 11
## [1] "3.0" "7.3"
## integer(0)
## [1] 10

```

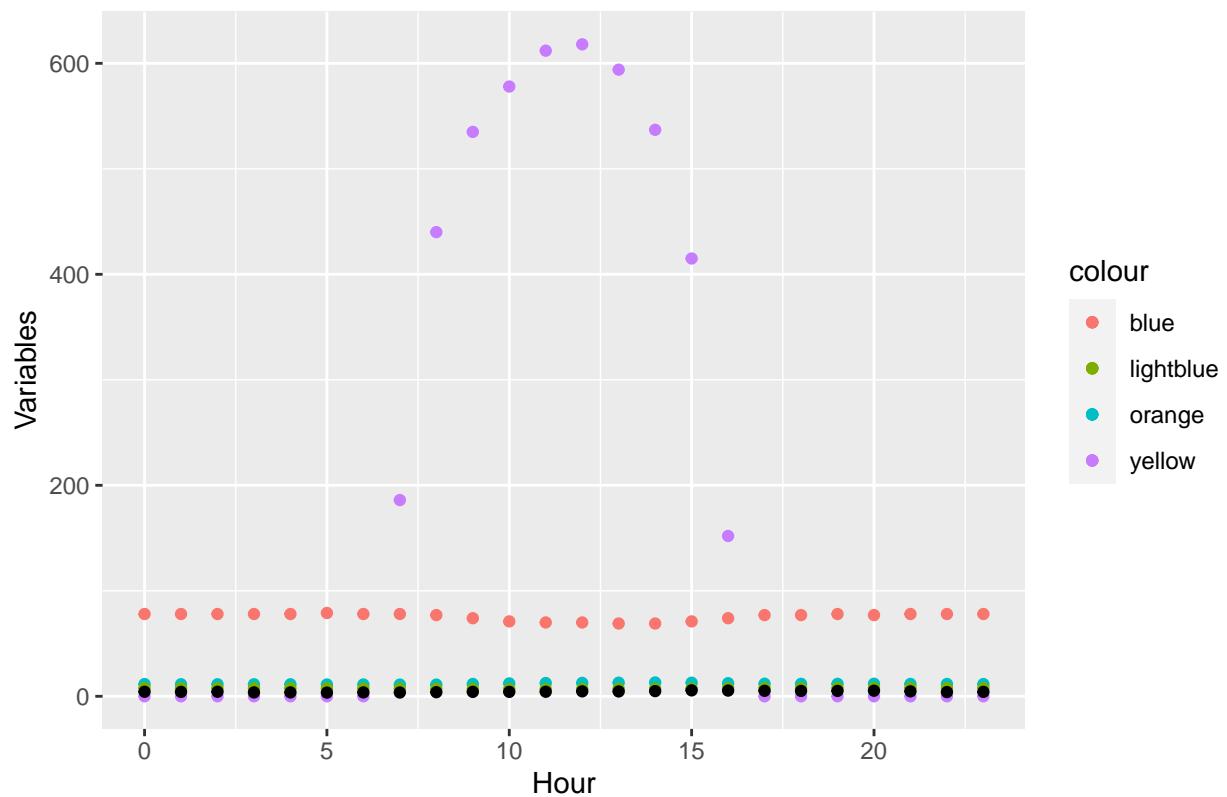
```

## [1] "3.1" "7.5"
## integer(0)
## [1] 12
## [1] "3.1" "7.4"
## integer(0)
## [1] 11
## [1] "3.0" "7.2"
## integer(0)
## [1] 10
## [1] "3.2" "6.7"
## integer(0)
## [1] 12
## [1] "3.0" "6.5"
## integer(0)
## [1] 9
## [1] "2.9" "6.6"
## integer(0)
## [1] 11
## [1] "2.9" "7.0"
## integer(0)
## [1] 9
## [1] "2.9" "6.5"
## integer(0)
## [1] 10
## [1] "3.2" "7.0"
## integer(0)
## [1] 11
## [1] "3.4" "7.0"
## integer(0)
## [1] 12
## [1] "3.4" "7.5"
## integer(0)
## [1] 11
## [1] "3.5" "7.7"
## integer(0)
## [1] 11
## [1] "2.5" "8.0"
## integer(0)
## [1] 10
## [1] "3.0" "8.2"
## integer(0)
## [1] 9
## [1] "2.5" "8.7"
## integer(0)
## [1] 10
## [1] "2.6" "8.6"
## integer(0)
## [1] 11
## [1] "3.0" "8.9"
## integer(0)
## [1] 10
## [1] "3.3" "8.6"
## integer(0)
## [1] 11

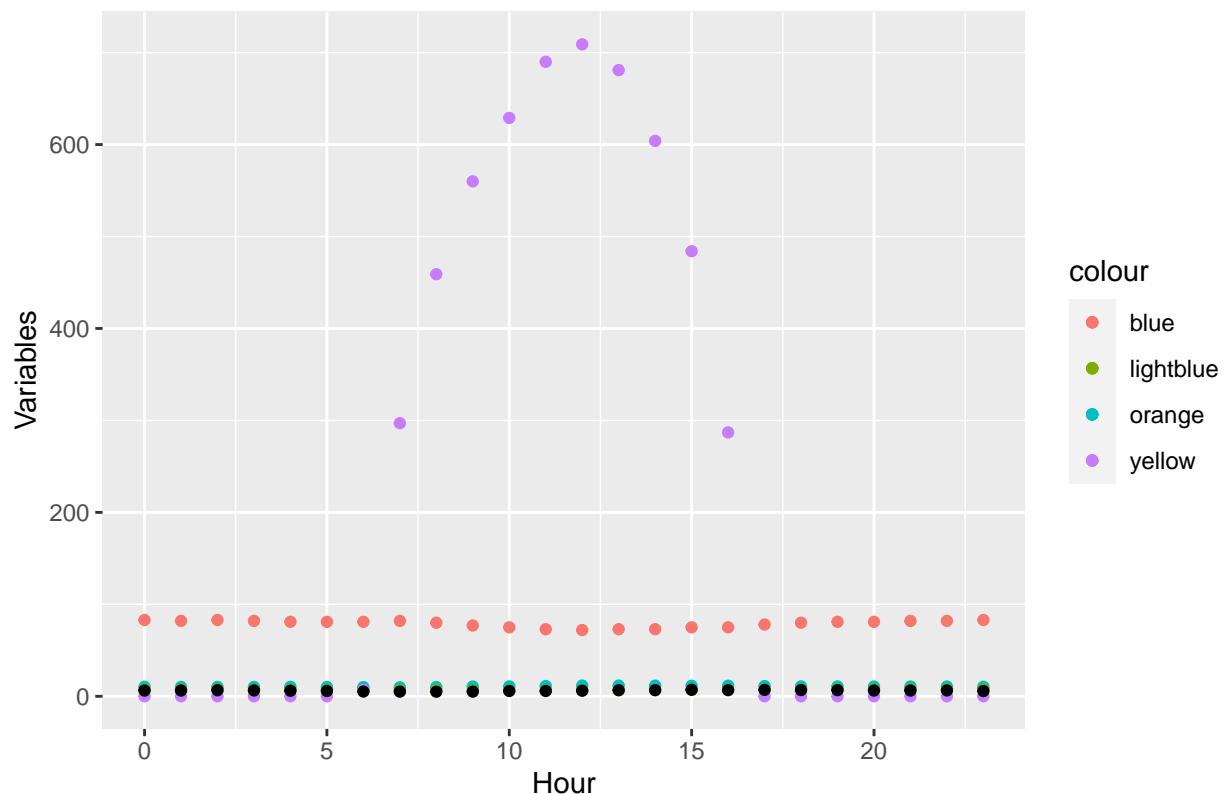
```

```
## [1] "3.9" "8.8"
## integer(0)
## [1] 11
## [1] "3.5" "8.3"
## integer(0)
## [1] 9
## [1] "3.4" "8.2"
## integer(0)
## [1] 11
## [1] "3.4" "8.2"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "Apr" "Sep"
## integer(0)
## [1] 24
## [1] 0 23
## integer(0)
## [1] 69
## [1] 9.8 17.1
## integer(0)
## [1] 59
## [1] 6.4 13.4
## integer(0)
## [1] 25
## [1] 69 93
## integer(0)
## [1] 134
## [1] 0 795
## integer(0)
## [1] 57
## [1] 2.5 8.9
```

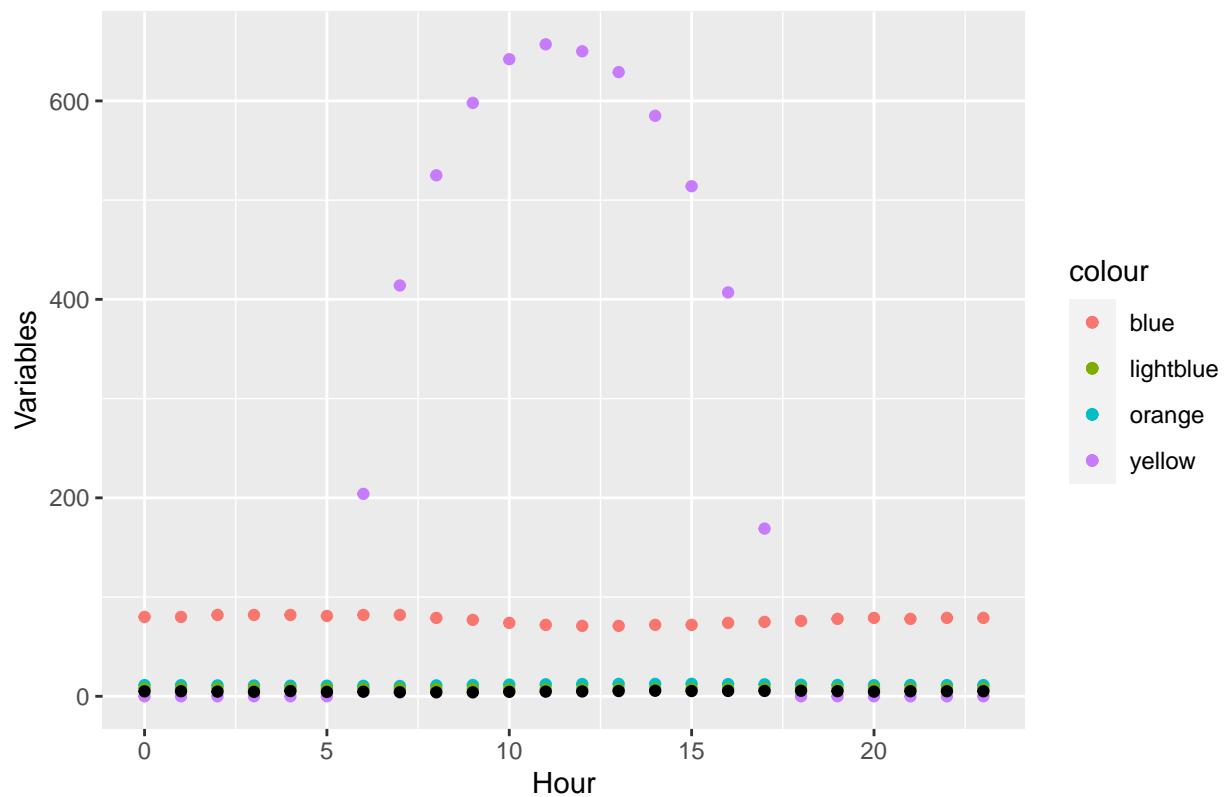
Jan Data: USA\_CA\_Point.Reyes.Lighthouse



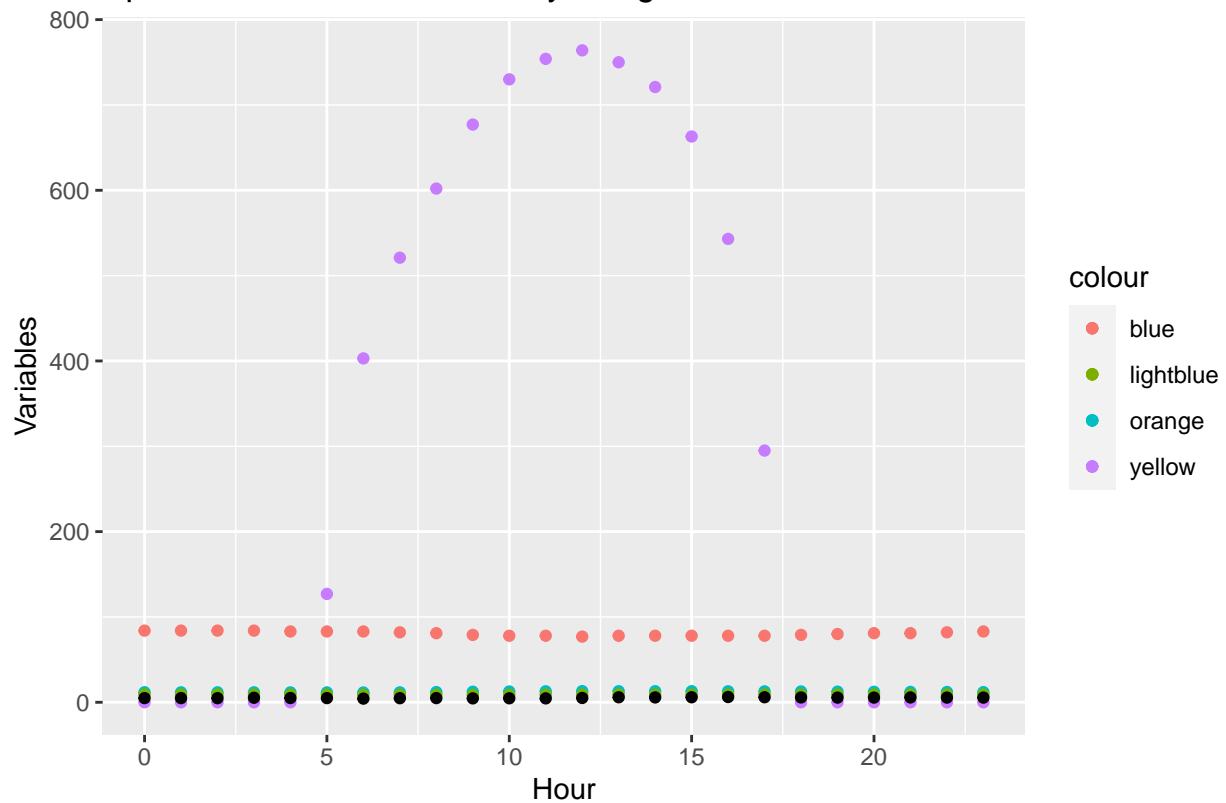
Feb Data: USA\_CA\_Point.Reyes.Lighthouse



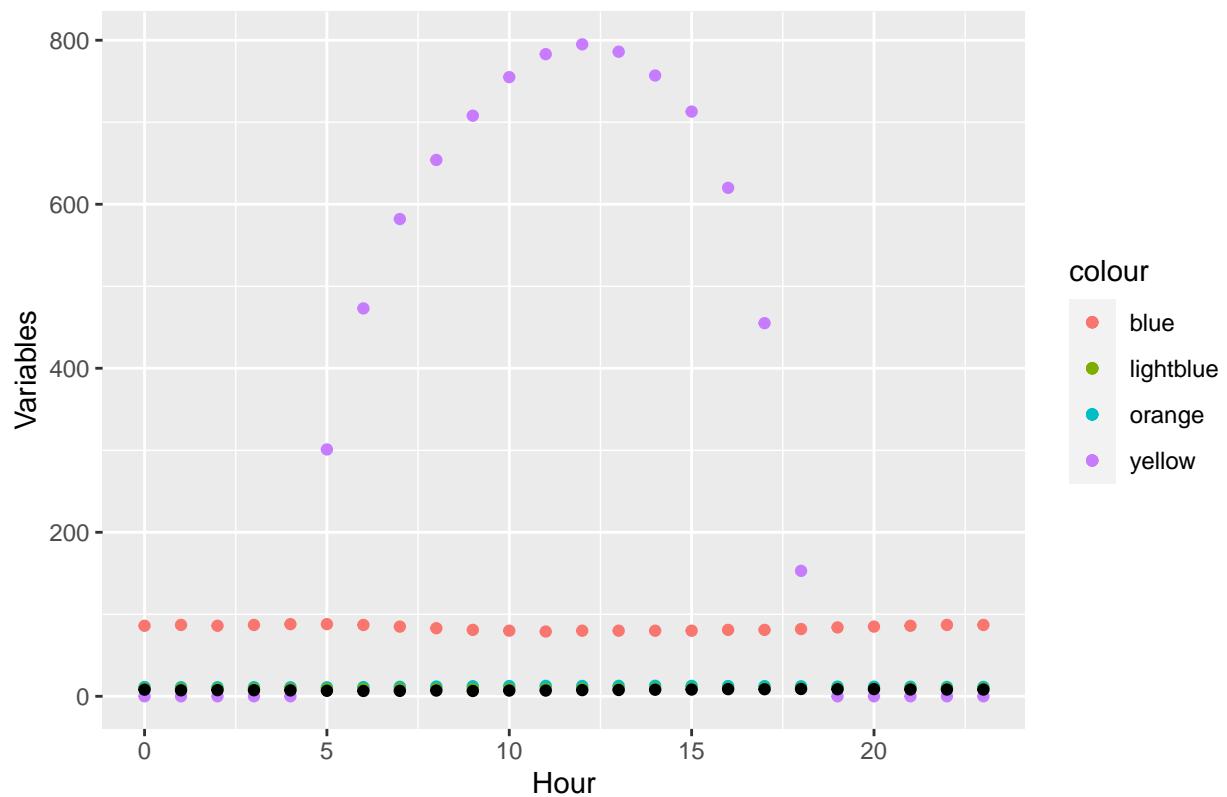
Mar Data: USA\_CA\_Point.Reyes.Lighthouse



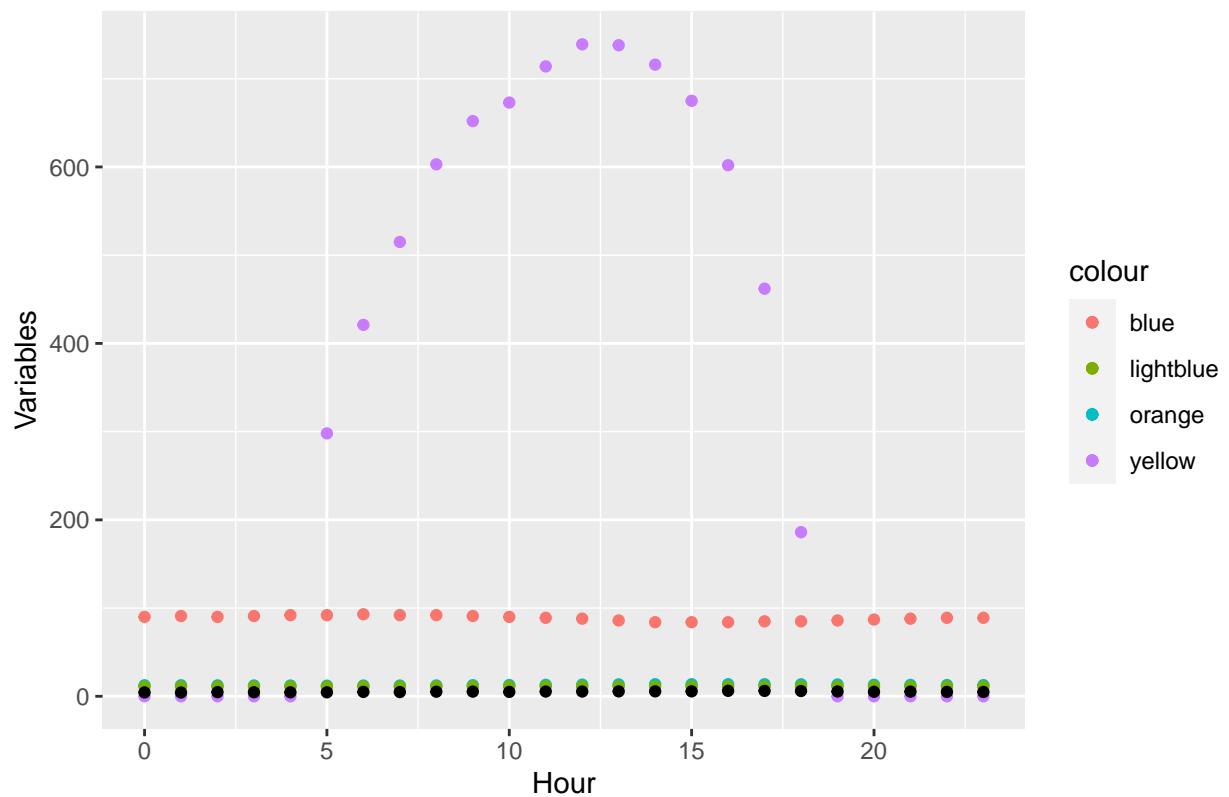
Apr Data: USA\_CA\_Point.Reyes.Lighthouse



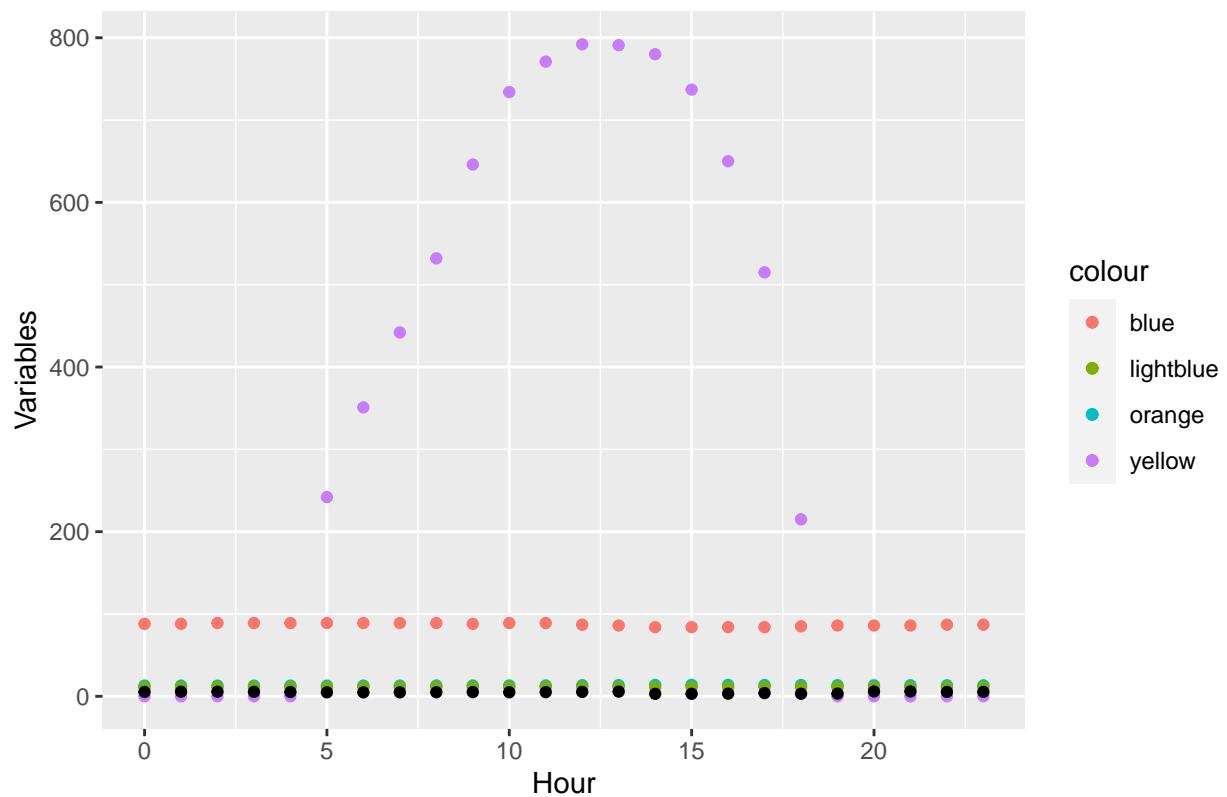
### May Data: USA\_CA\_Point.Reyes.Lighthouse



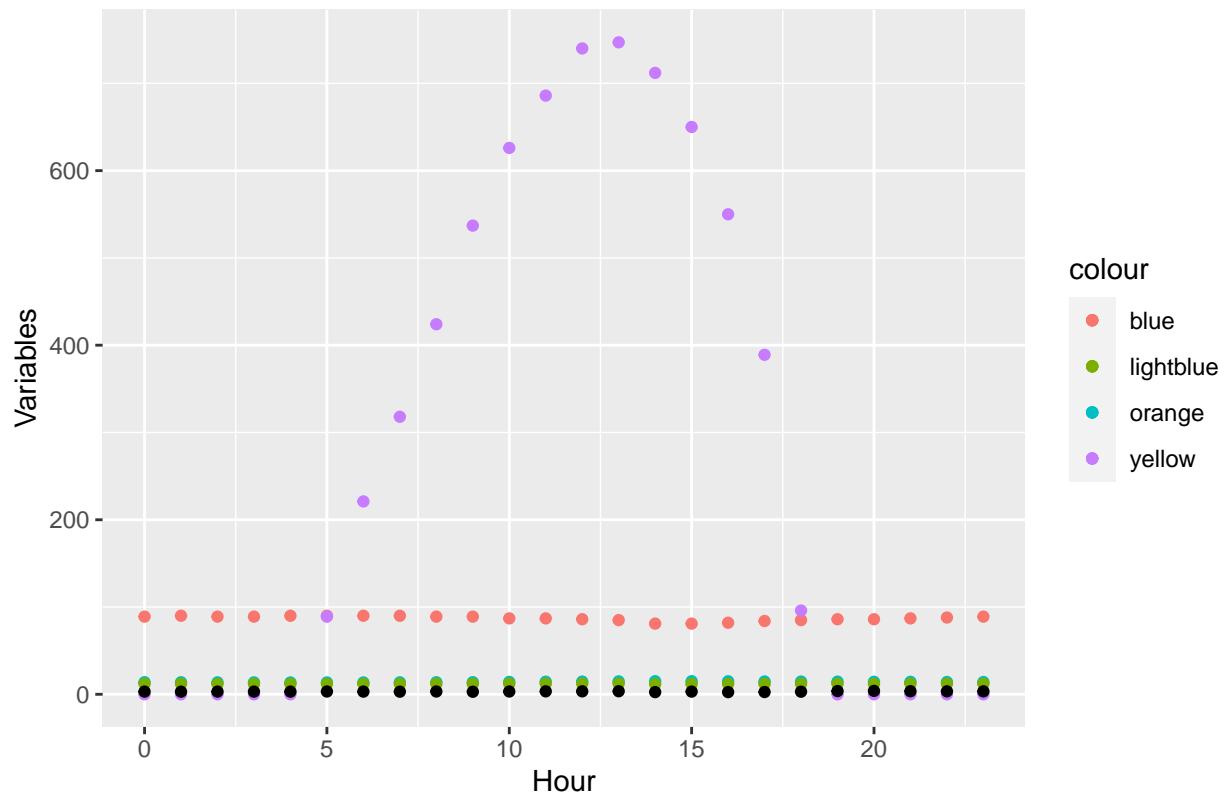
Jun Data: USA\_CA\_Point.Reyes.Lighthouse



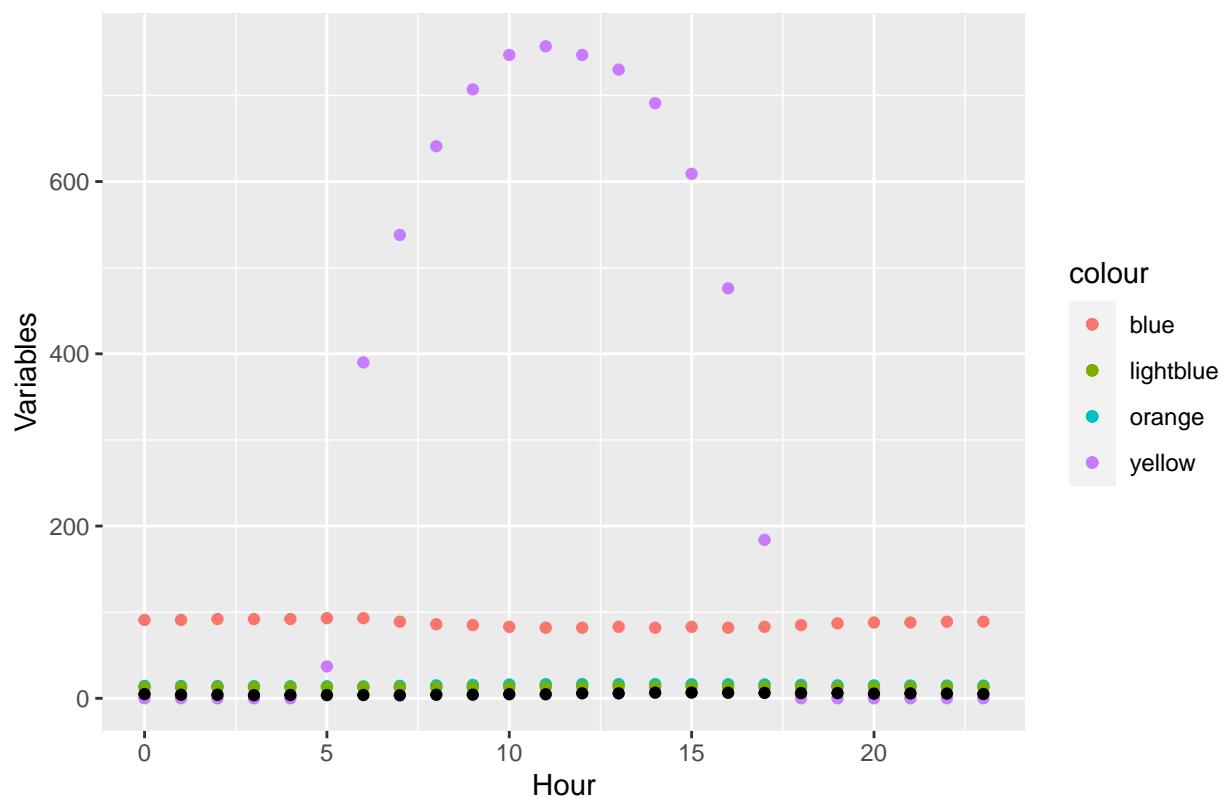
Jul Data: USA\_CA\_Point.Reyes.Lighthouse



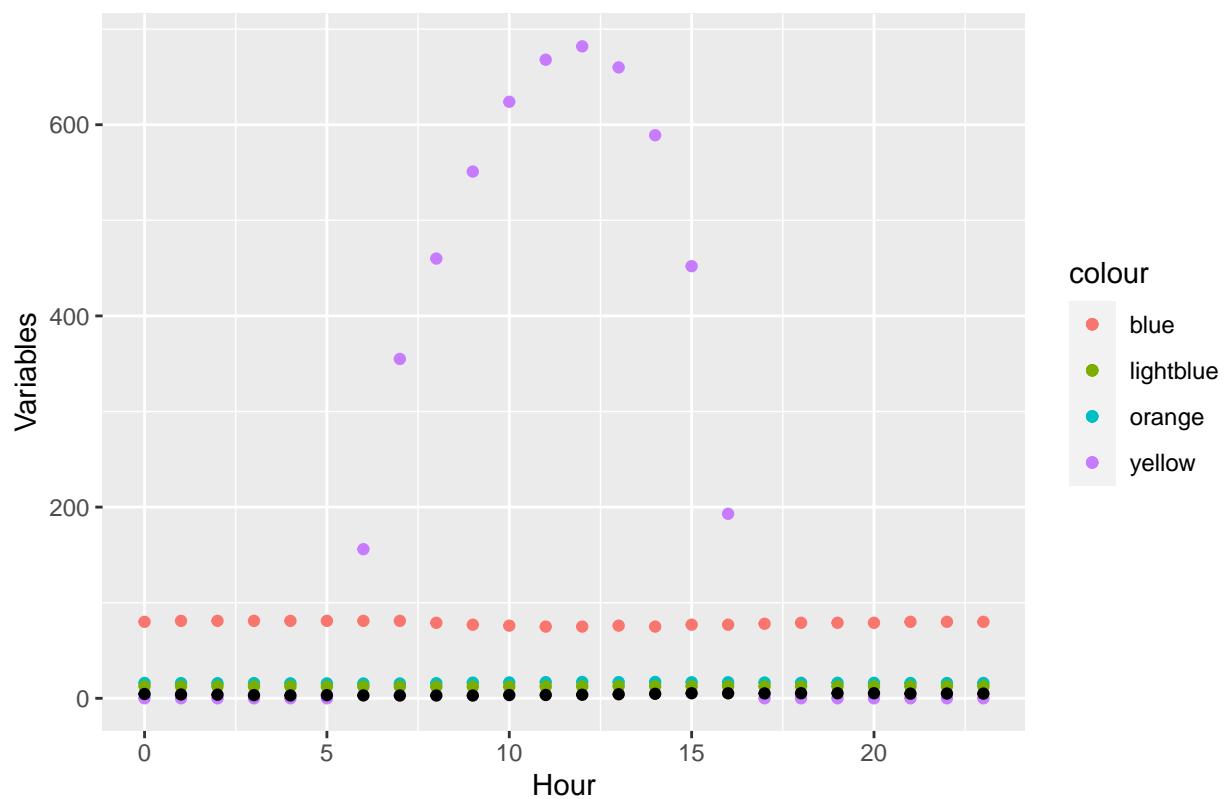
Aug Data: USA\_CA\_Point.Reyes.Lighthouse



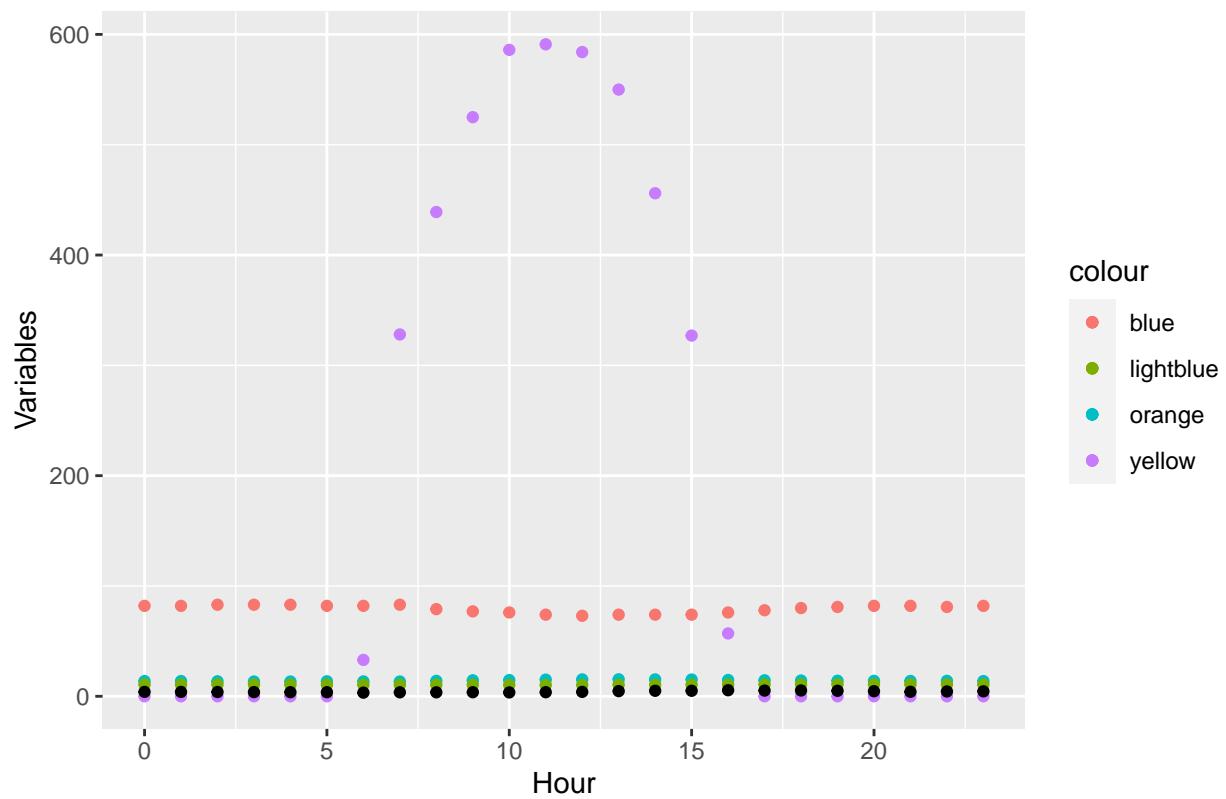
Sep Data: USA\_CA\_Point.Reyes.Lighthouse



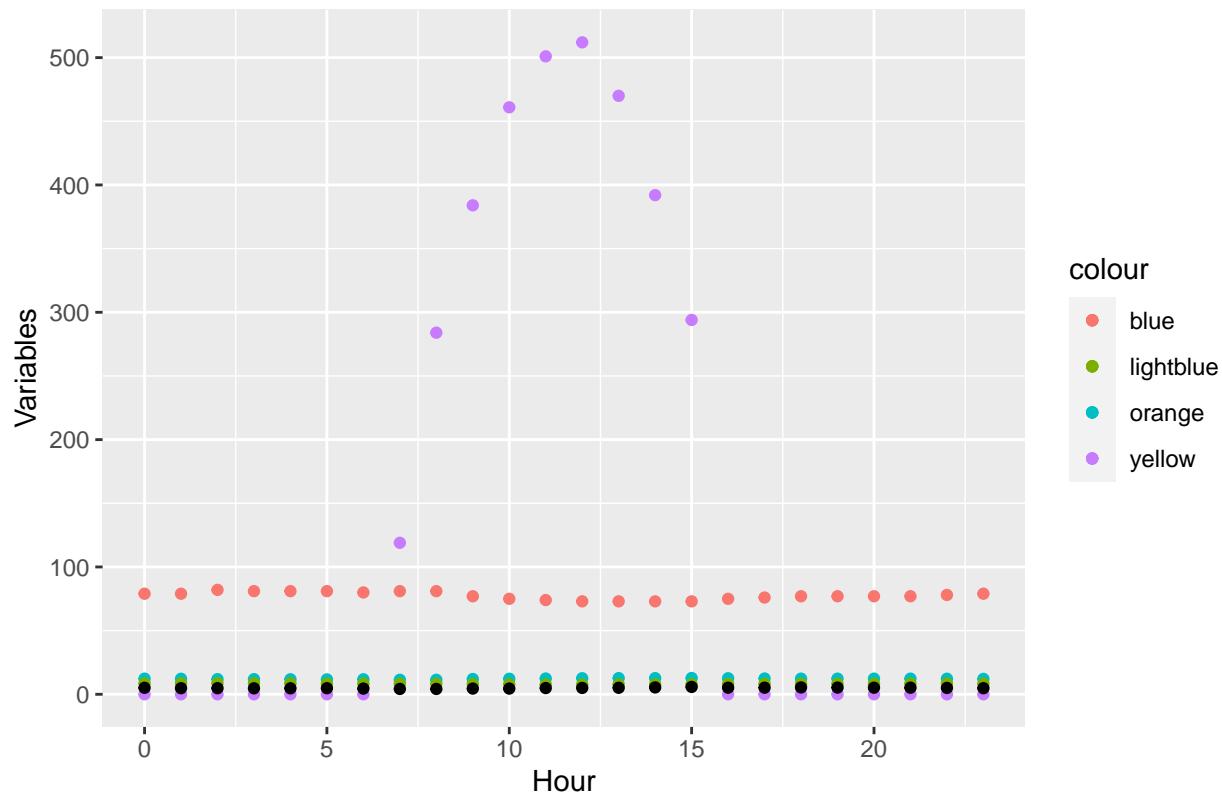
Oct Data: USA\_CA\_Point.Reyes.Lighthouse



Nov Data: USA\_CA\_Point.Reyes.Lighthouse



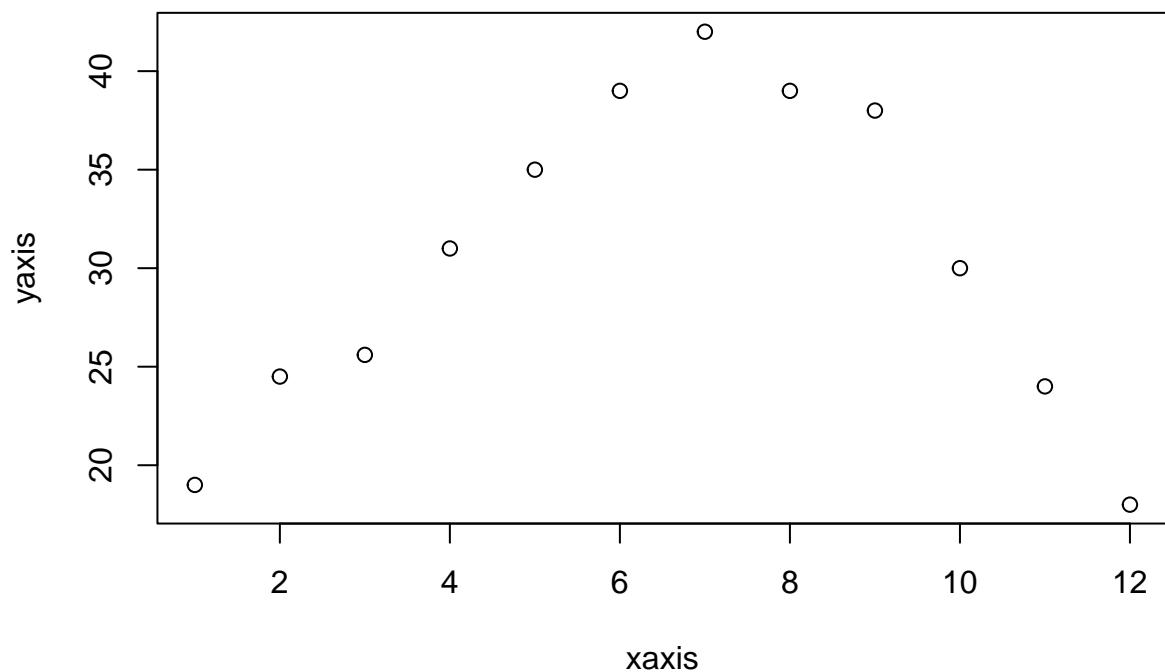
## Dec Data: USA\_CA\_Point.Reyes.Lighthouse

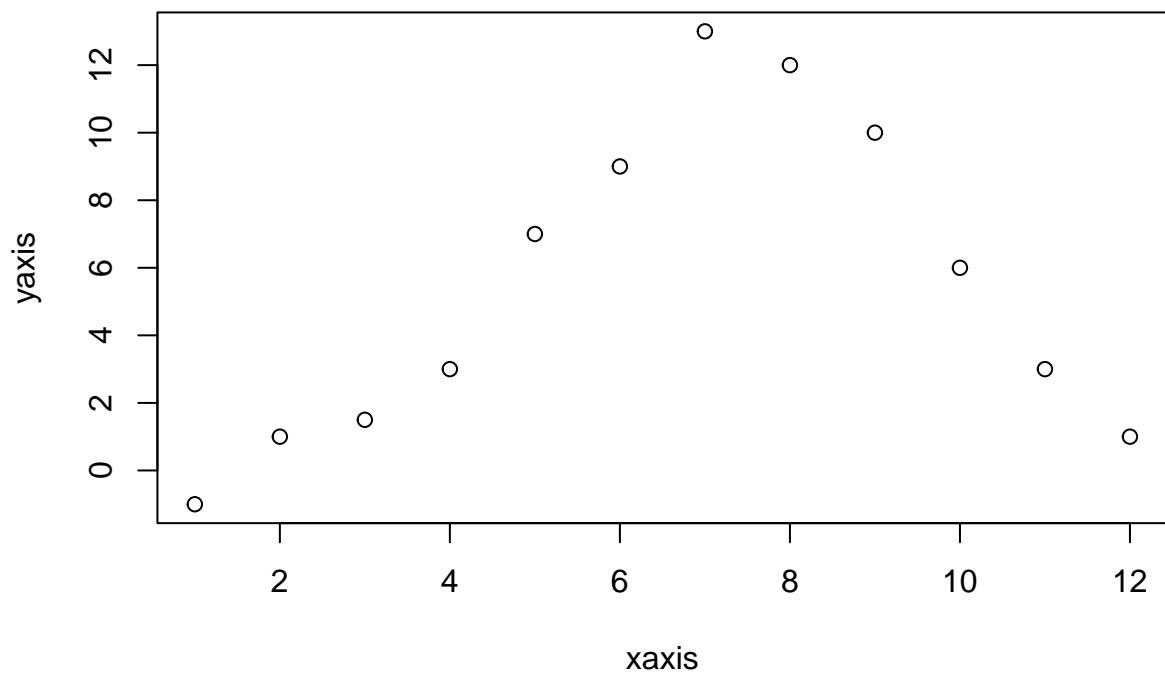


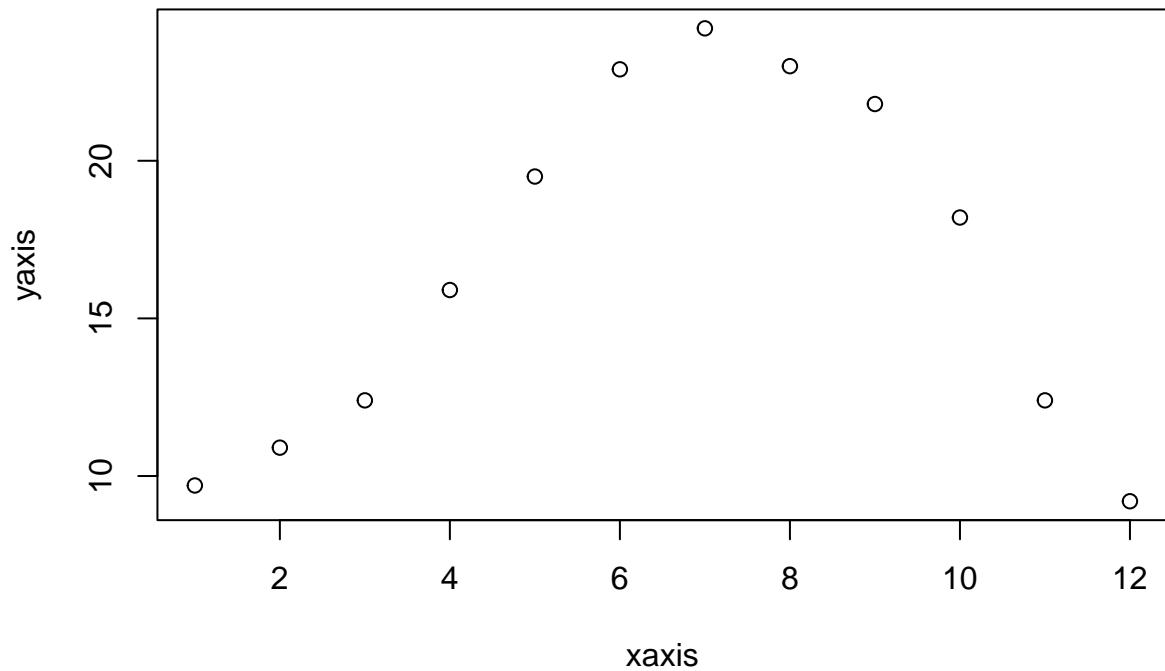
```
stat_UCD = stat_parser('dt/USA_CA_UC-Davis-University.AP.720576_TMYx.2007-2021.stat')
```

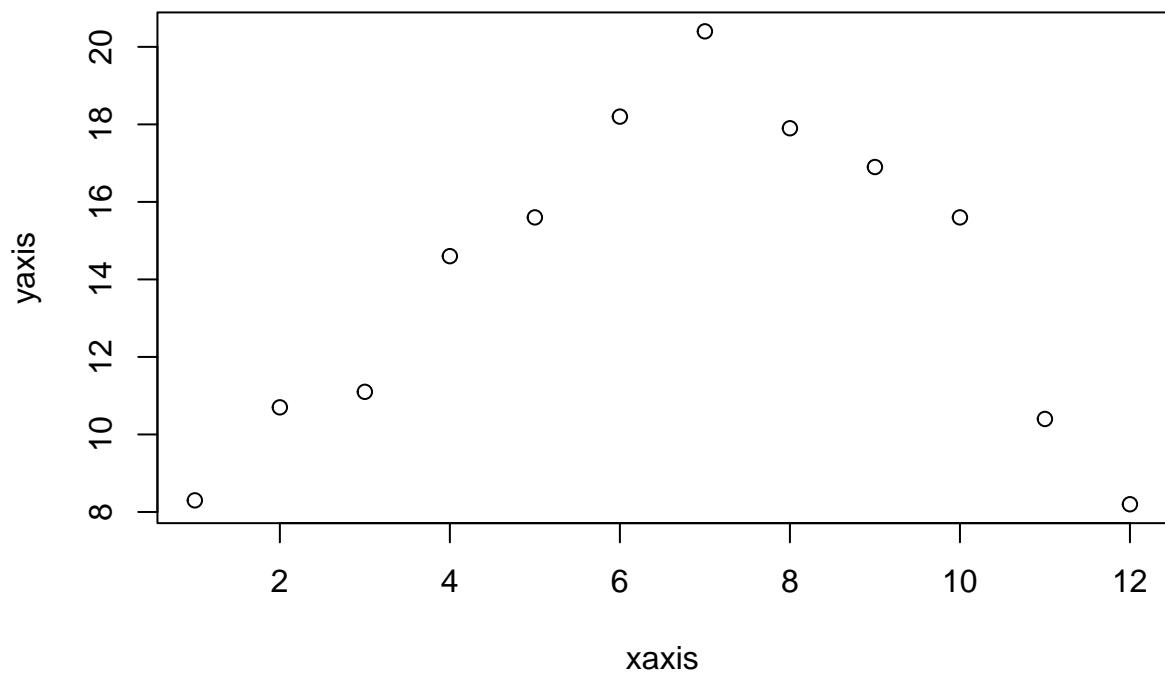
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] 18 42
## integer(0)
## [1] 12
## [1] "2023-01-27 16:00:00" "2023-12-19 15:00:00"
## integer(0)
## [1] 10
## [1] -1 13
## integer(0)
## [1] 12
## [1] "2023-01-02 07:00:00" "2023-12-30 05:00:00"
## integer(0)
## [1] 11
## [1] 9.2 24.2
## integer(0)
## [1] 11
## [1] 8.2 20.4
## integer(0)
## [1] 11
## [1] 18 42
## integer(0)
```

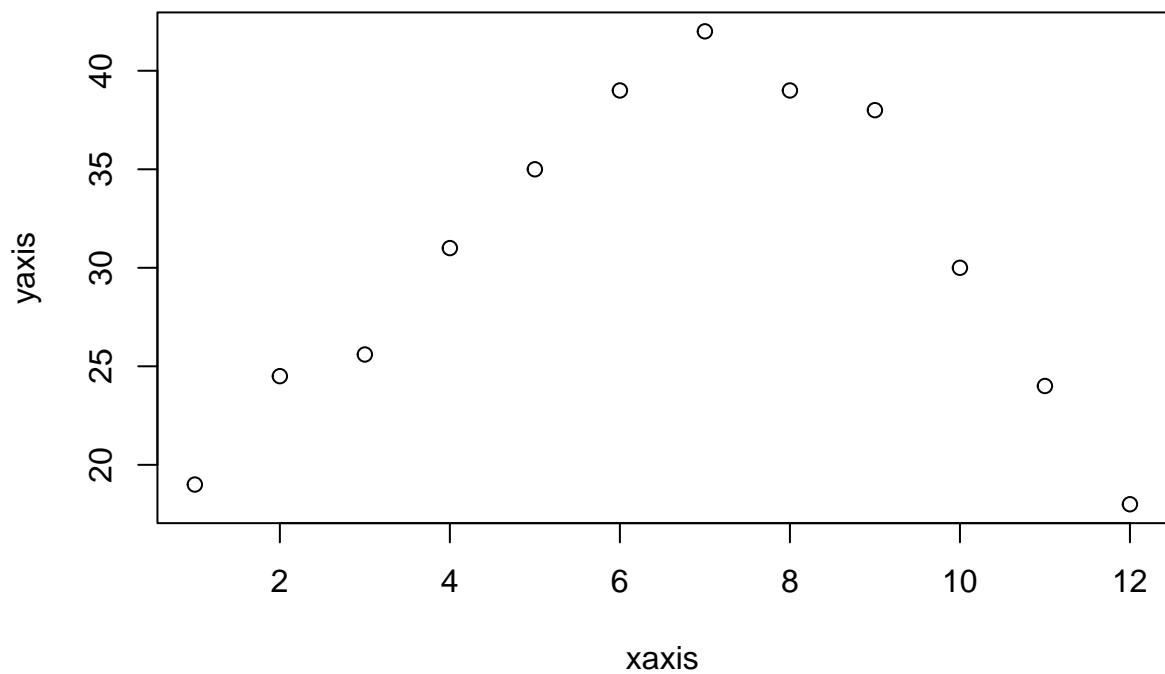
```
## [1] 9
## [1] -1 14
## integer(0)
## [1] 11
## [1] 10.4 28.5
## integer(0)
## [1] 12
## [1] 13 40
## integer(0)
## [1] 11
## [1] 0 13
## integer(0)
## [1] 11
## [1] 8.0 19.9
```

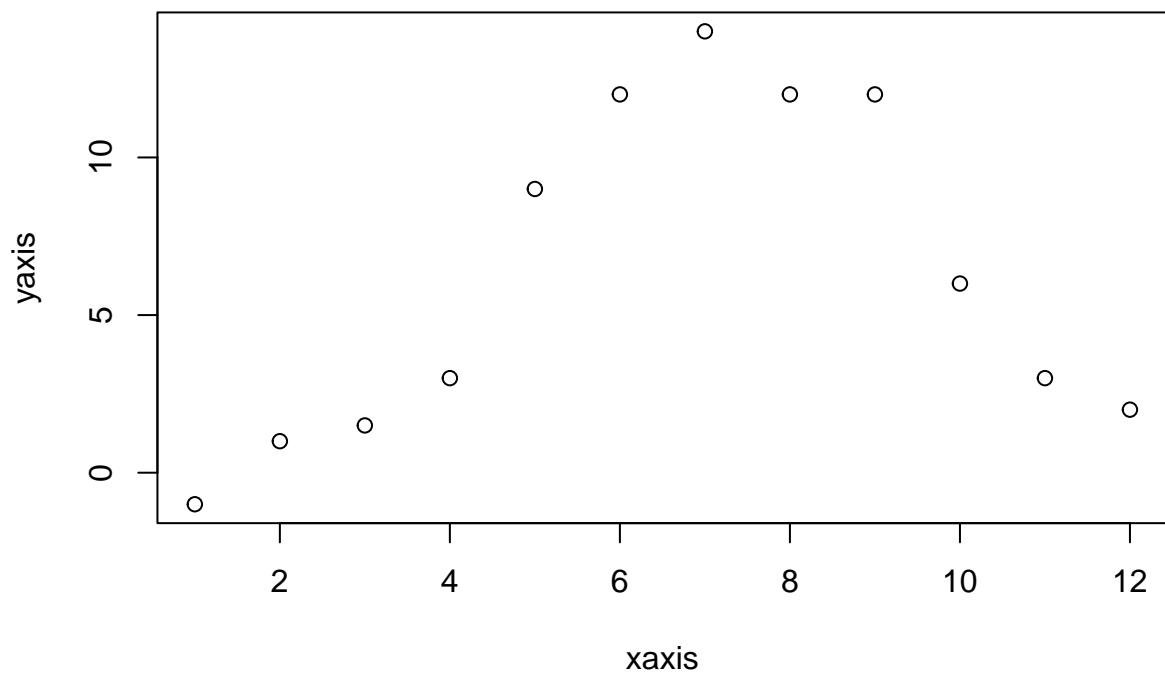


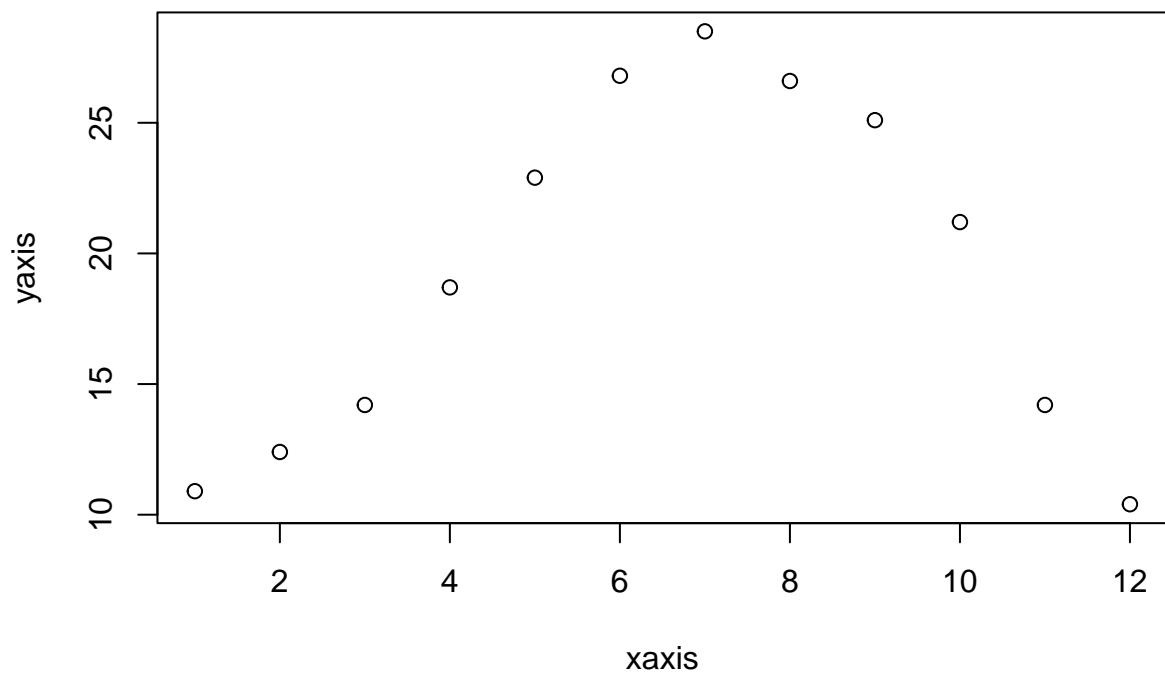


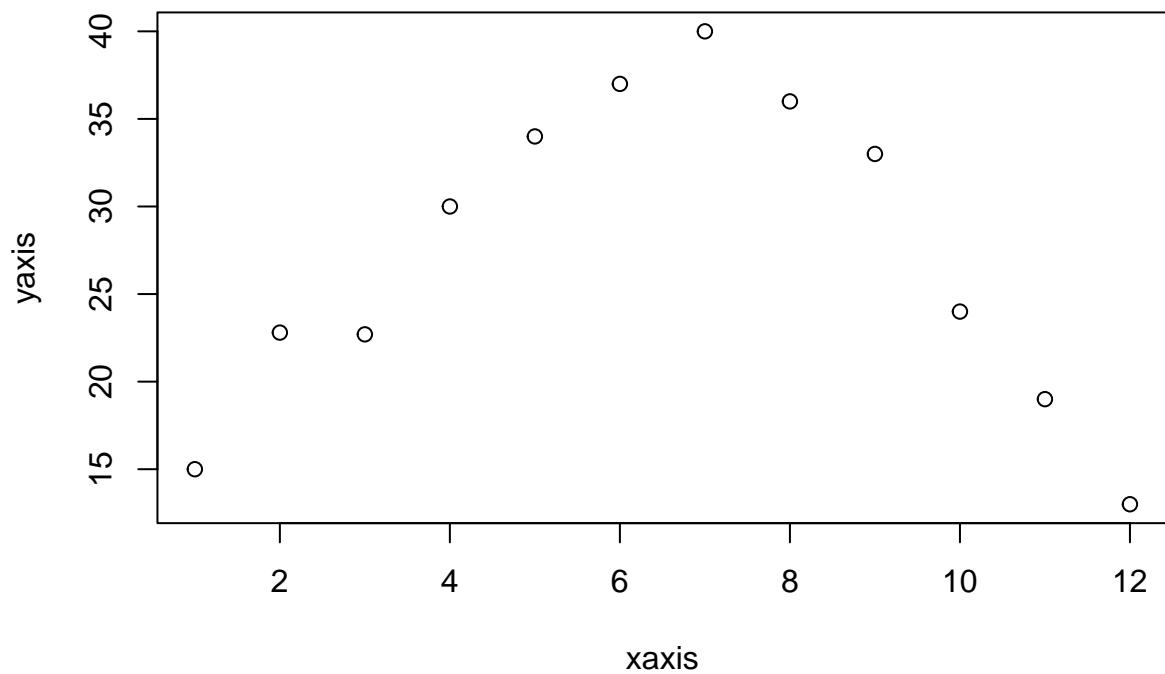


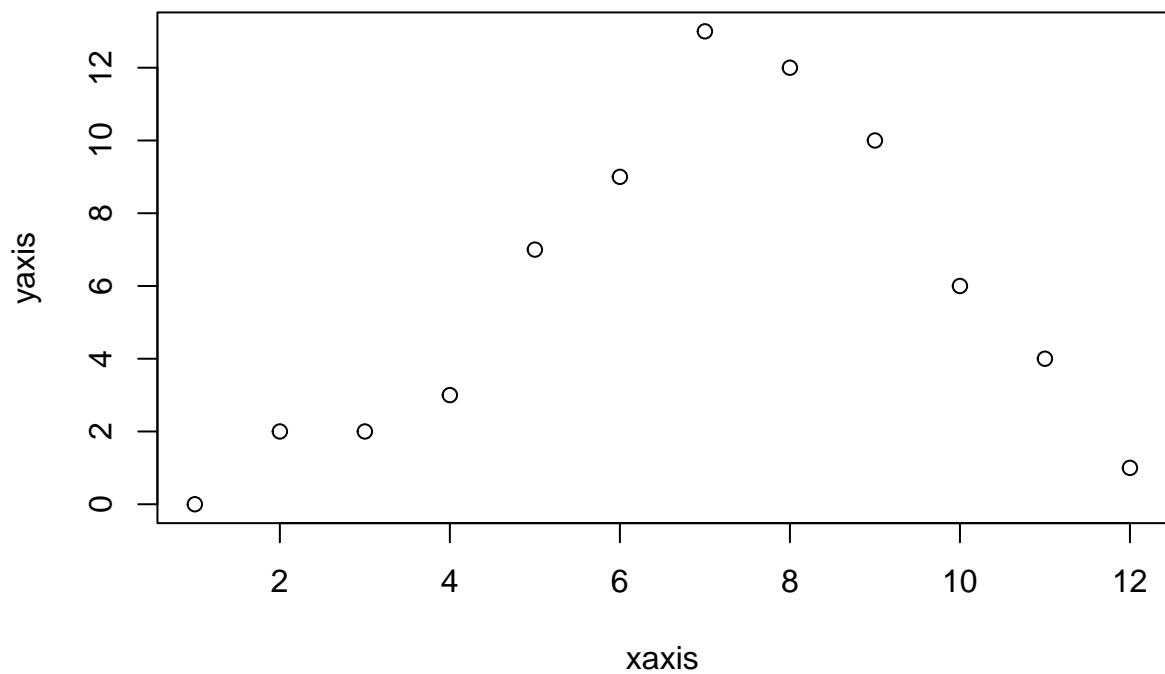


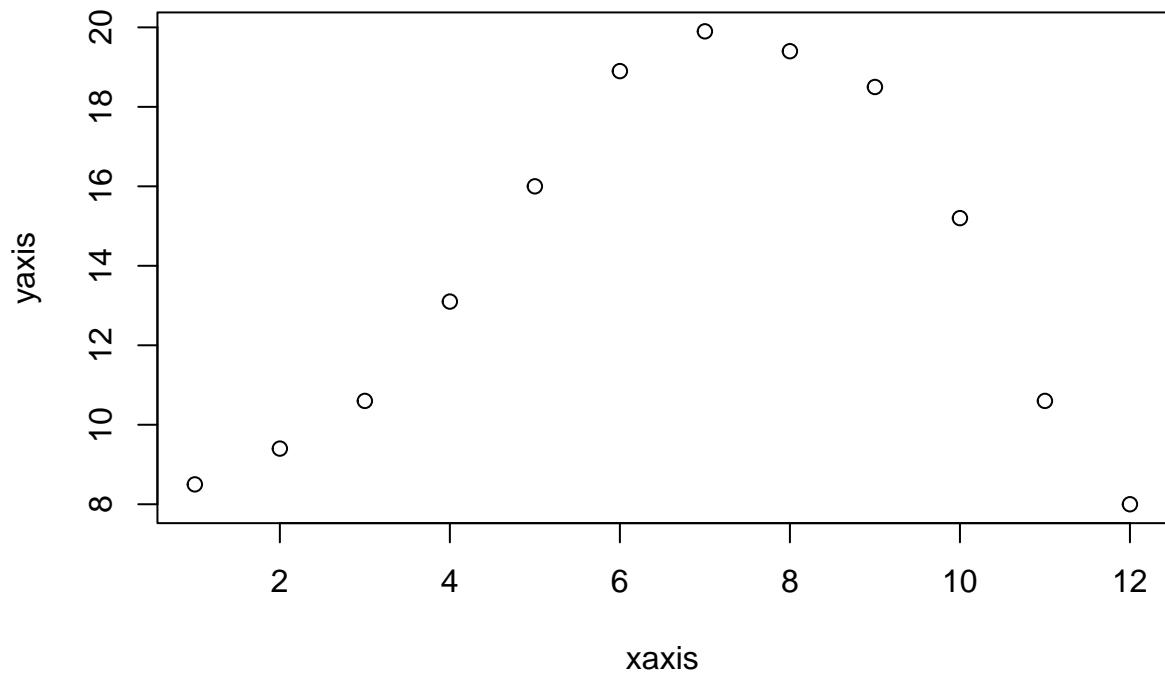




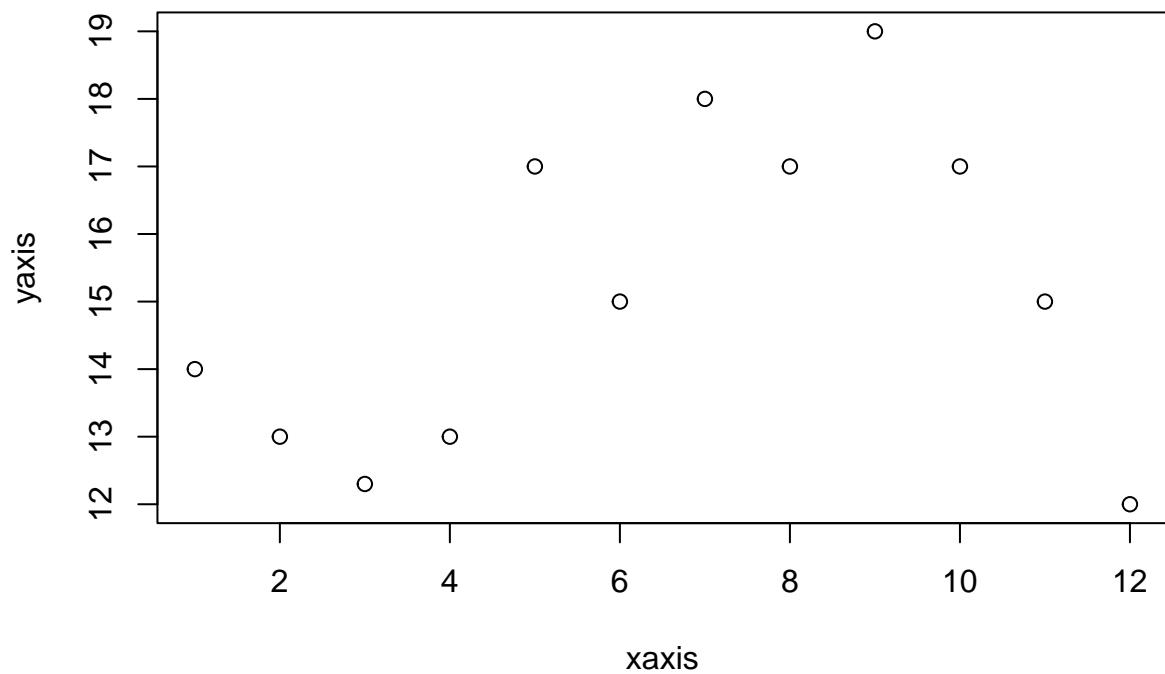


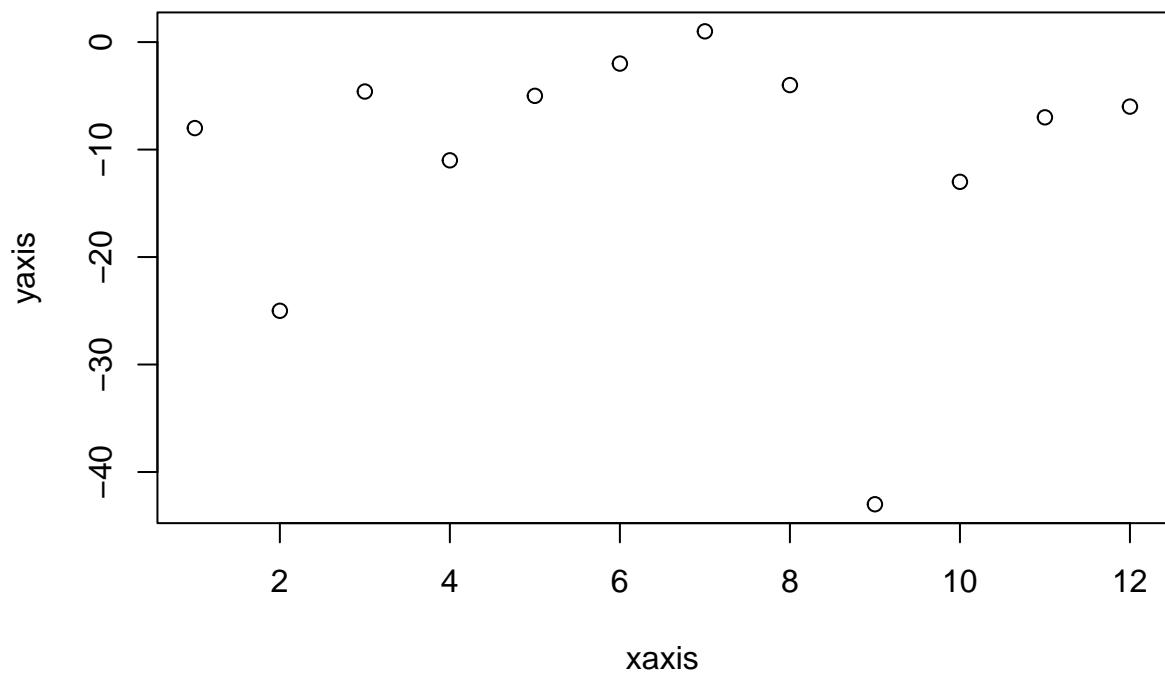


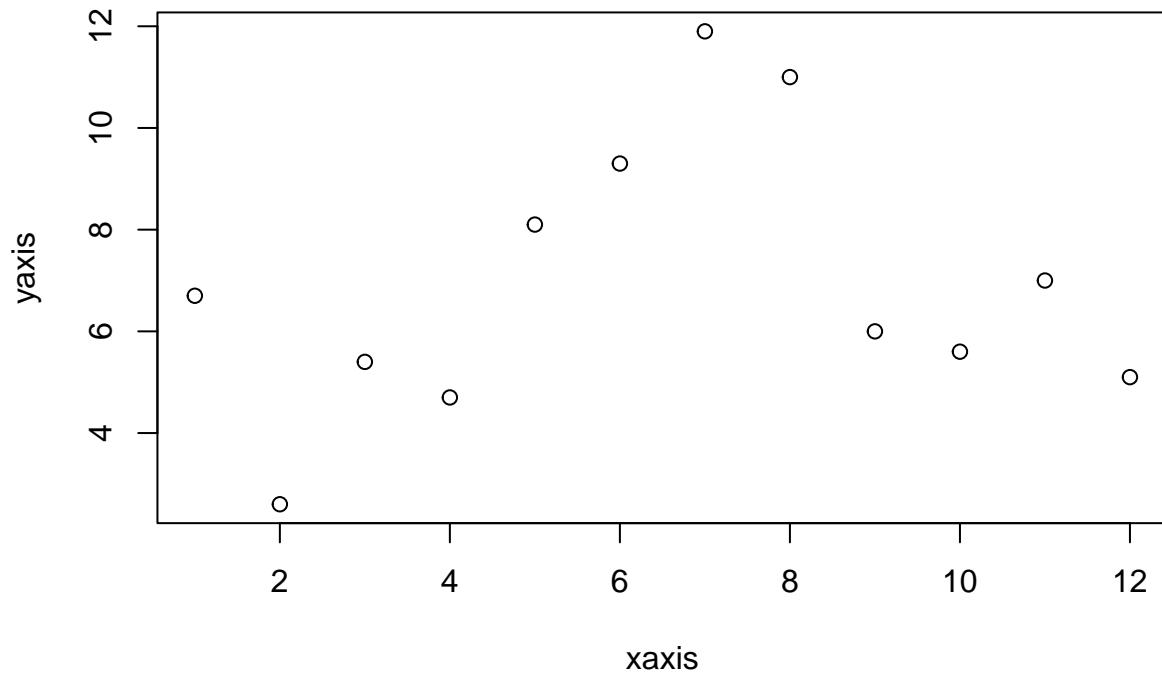




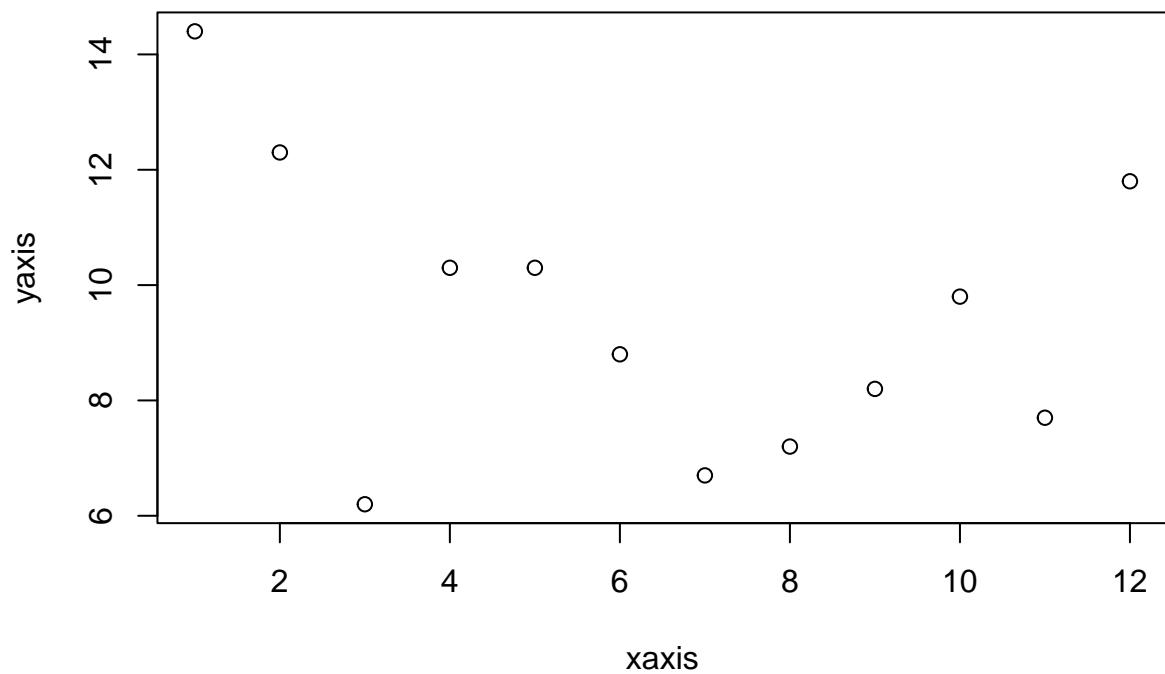
```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] 12 19
## integer(0)
## [1] 12
## [1] "2023-01-20 13:00:00" "2023-12-14 17:00:00"
## integer(0)
## [1] 12
## [1] -43    1
## integer(0)
## [1] 12
## [1] "2023-01-01 07:00:00" "2023-12-31 14:00:00"
## integer(0)
## [1] 12
## [1]  2.6 11.9
```

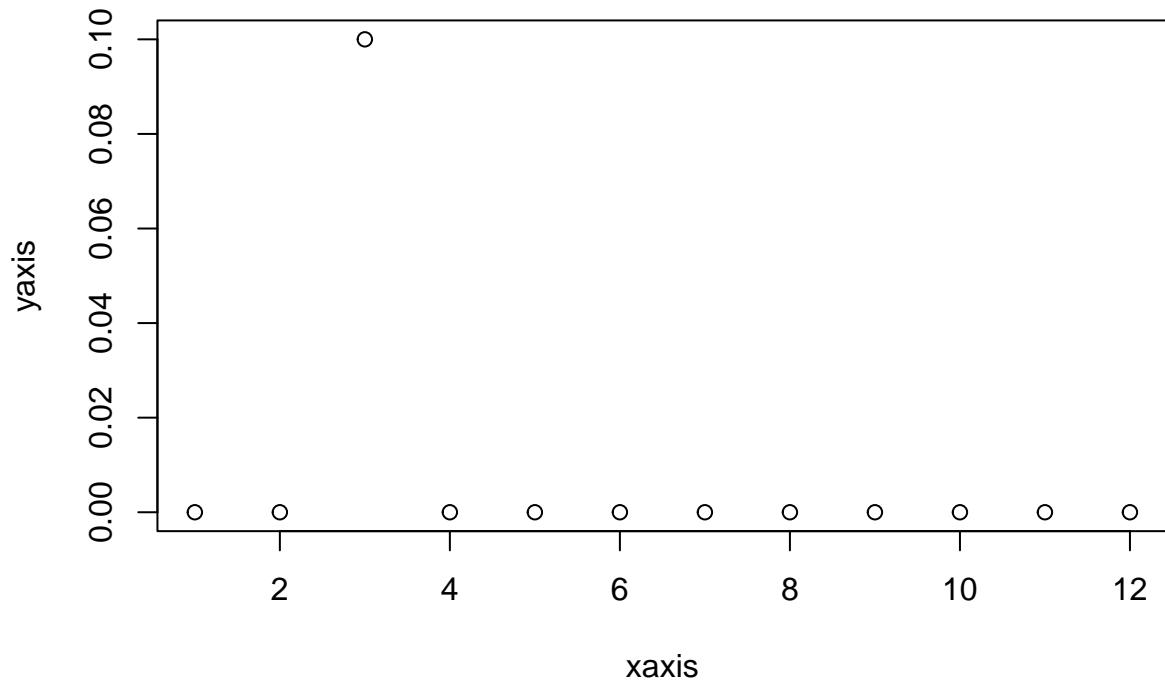


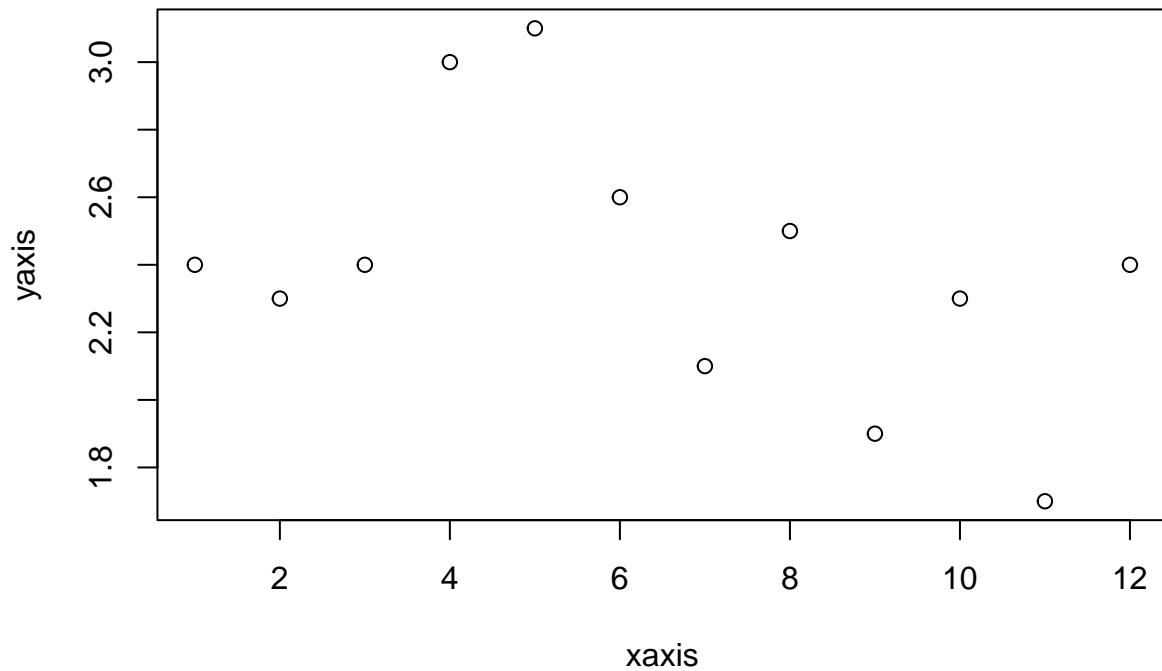




```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] 6.2 14.4
## integer(0)
## [1] 12
## [1] "2023-01-06 20:00:00" "2023-12-31 14:00:00"
## integer(0)
## [1] 2
## [1] 0.0 0.1
## integer(0)
## [1] 12
## [1] "2023-01-01 21:00:00" "2023-12-01 16:00:00"
## integer(0)
## [1] 9
## [1] 1.7 3.1
```







```
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 9
## [1] " 3" "25"
## integer(0)
## [1] 5
## [1] " 1" " 7"
## integer(0)
## [1] 5
## [1] " 0" " 4"
## integer(0)
## [1] 4
## [1] " 0" " 3"
## integer(0)
## [1] 4
## [1] " 0" " 3"
## integer(0)
## [1] 5
## [1] " 1" " 5"
## integer(0)
## [1] 7
## [1] " 1" " 9"
## integer(0)
## [1] 9
## [1] " 3" "13"
## integer(0)
```

```

## [1] 9
## [1] " 6" "32"
## integer(0)
## [1] 12
## [1] " 3" "28"
## integer(0)
## [1] 9
## [1] " 3" "18"
## integer(0)
## [1] 8
## [1] " 1" " 8"
## integer(0)
## [1] 6
## [1] " 1" " 6"
## integer(0)
## [1] 5
## [1] " 1" " 5"
## integer(0)
## [1] 6
## [1] " 1" " 6"
## integer(0)
## [1] 10
## [1] " 2" "20"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 11
## [1] " 7.4" "17.8"
## integer(0)
## [1] 12
## [1] " 7.2" "17.1"
## integer(0)
## [1] 12
## [1] " 7.2" "16.4"
## integer(0)
## [1] 11
## [1] " 7.1" "16.0"
## integer(0)
## [1] 11
## [1] " 6.9" "15.4"
## integer(0)
## [1] 11
## [1] " 6.7" "14.9"
## integer(0)
## [1] 11
## [1] " 6.8" "16.4"
## integer(0)
## [1] 11
## [1] " 6.5" "19.5"
## integer(0)
## [1] 12
## [1] " 6.8" "22.3"
## integer(0)
## [1] 11
## [1] " 8.2" "25.2"

```

```

## integer(0)
## [1] 12
## [1] " 9.7" "27.7"
## integer(0)
## [1] 12
## [1] "11.2" "29.7"
## integer(0)
## [1] 11
## [1] "12.2" "31.7"
## integer(0)
## [1] 12
## [1] "13.0" "33.2"
## integer(0)
## [1] 12
## [1] "13.3" "34.1"
## integer(0)
## [1] 11
## [1] "13.3" "34.7"
## integer(0)
## [1] 12
## [1] "12.9" "34.5"
## integer(0)
## [1] 12
## [1] "11.4" "33.3"
## integer(0)
## [1] 12
## [1] "10.4" "31.0"
## integer(0)
## [1] 11
## [1] " 9.7" "27.3"
## integer(0)
## [1] 11
## [1] " 9.0" "24.5"
## integer(0)
## [1] 12
## [1] " 8.5" "22.0"
## integer(0)
## [1] 12
## [1] " 8.1" "20.1"
## integer(0)
## [1] 11
## [1] " 7.6" "19.0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] " 9.4" "8.5"
## integer(0)
## [1] 10
## [1] " 9.3" "8.3"
## integer(0)
## [1] 10
## [1] " 9.5" "8.1"
## integer(0)
## [1] 12

```

```

## [1] " 9.4" "8.0"
## integer(0)
## [1] 12
## [1] " 9.7" "8.0"
## integer(0)
## [1] 12
## [1] " 9.5" "7.9"
## integer(0)
## [1] 12
## [1] "10.0" "8.5"
## integer(0)
## [1] 11
## [1] "10.3" "8.7"
## integer(0)
## [1] 9
## [1] "10.0" "8.5"
## integer(0)
## [1] 12
## [1] " 9.8" "8.3"
## integer(0)
## [1] 12
## [1] " 9.6" "8.1"
## integer(0)
## [1] 11
## [1] " 9.1" "7.5"
## integer(0)
## [1] 12
## [1] " 8.5" "7.5"
## integer(0)
## [1] 12
## [1] " 8.2" "7.3"
## integer(0)
## [1] 12
## [1] " 8.0" "7.3"
## integer(0)
## [1] 11
## [1] " 8.1" "7.3"
## integer(0)
## [1] 10
## [1] " 8.2" "7.2"
## integer(0)
## [1] 12
## [1] " 8.9" "8.0"
## integer(0)
## [1] 12
## [1] " 9.3" "8.4"
## integer(0)
## [1] 11
## [1] " 9.3" "8.8"
## integer(0)
## [1] 11
## [1] " 9.4" "8.7"
## integer(0)
## [1] 12

```

```

## [1] " 9.6" "8.8"
## integer(0)
## [1] 12
## [1] " 9.7" "8.7"
## integer(0)
## [1] 12
## [1] " 9.5" "8.7"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 10
## [1] "62" "89"
## integer(0)
## [1] 9
## [1] "64" "90"
## integer(0)
## [1] 12
## [1] "65" "89"
## integer(0)
## [1] 11
## [1] "66" "91"
## integer(0)
## [1] 10
## [1] "67" "90"
## integer(0)
## [1] 11
## [1] "68" "91"
## integer(0)
## [1] 11
## [1] "67" "91"
## integer(0)
## [1] 10
## [1] "56" "91"
## integer(0)
## [1] 12
## [1] "47" "91"
## integer(0)
## [1] 12
## [1] "41" "88"
## integer(0)
## [1] 11
## [1] "36" "84"
## integer(0)
## [1] 10
## [1] "32" "79"
## integer(0)
## [1] 10
## [1] "29" "73"
## integer(0)
## [1] 9
## [1] "27" "69"
## integer(0)
## [1] 9
## [1] "25" "67"
## integer(0)

```

```

## [1] 10
## [1] "23" "68"
## integer(0)
## [1] 11
## [1] "24" "68"
## integer(0)
## [1] 12
## [1] "27" "73"
## integer(0)
## [1] 11
## [1] "30" "79"
## integer(0)
## [1] 11
## [1] "35" "84"
## integer(0)
## [1] 11
## [1] "43" "86"
## integer(0)
## [1] 9
## [1] "50" "89"
## integer(0)
## [1] 10
## [1] "55" "90"
## integer(0)
## [1] 10
## [1] "59" "90"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 2
## [1] " 0" "151"
## integer(0)
## [1] 7
## [1] " 0" "475"
## integer(0)
## [1] 11
## [1] " 0" "656"
## integer(0)
## [1] 12
## [1] "128" "774"
## integer(0)
## [1] 12
## [1] "303" "850"

```

```

## integer(0)
## [1] 12
## [1] "379" "894"
## integer(0)
## [1] 12
## [1] "435" "915"
## integer(0)
## [1] 11
## [1] "476" "919"
## integer(0)
## [1] 12
## [1] "481" "911"
## integer(0)
## [1] 12
## [1] "447" "894"
## integer(0)
## [1] 12
## [1] "377" "865"
## integer(0)
## [1] 12
## [1] "215" "818"
## integer(0)
## [1] 12
## [1] " 0" "733"
## integer(0)
## [1] 8
## [1] " 0" "574"
## integer(0)
## [1] 5
## [1] " 0" "236"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## integer(0)
## [1] 1
## [1] " 0" " 0"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 8
## [1] "1.1" "2.3"
## integer(0)
## [1] 9
## [1] "1.2" "2.6"
## integer(0)
## [1] 11

```

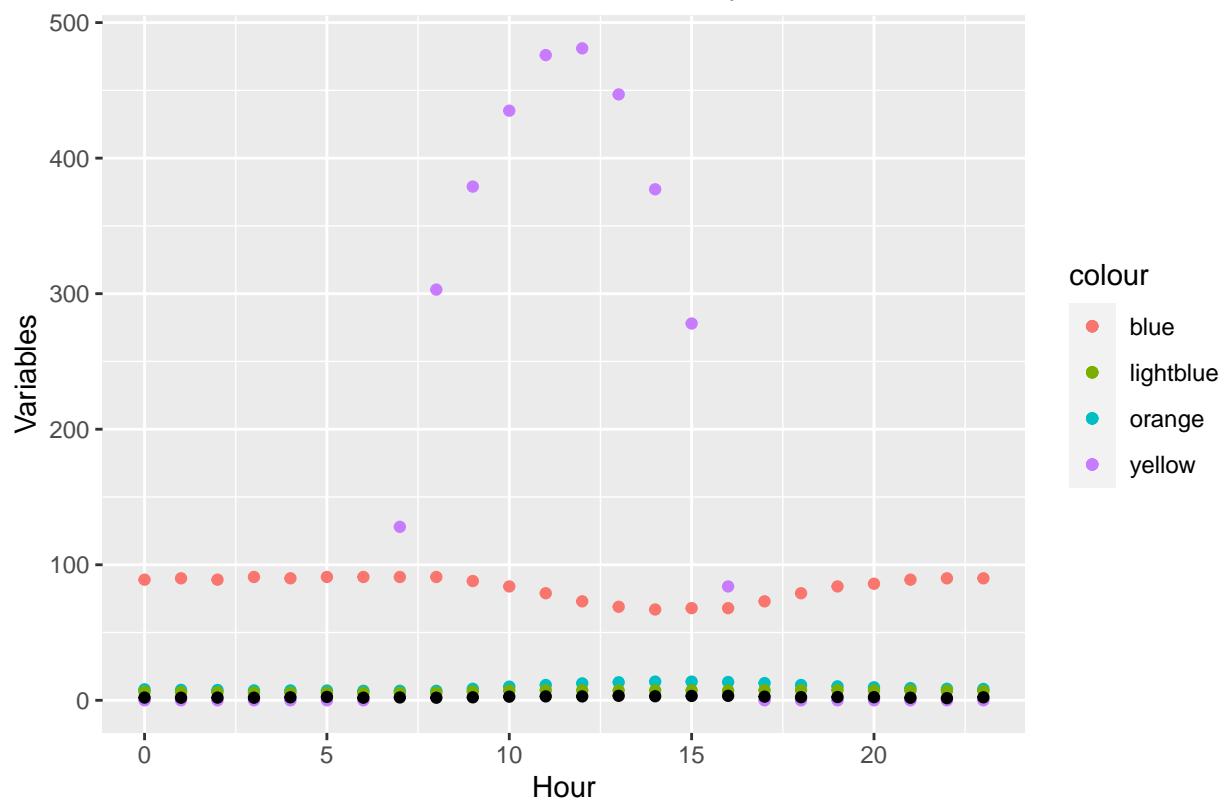
```

## [1] "1.2" "2.5"
## integer(0)
## [1] 11
## [1] "1.0" "2.5"
## integer(0)
## [1] 7
## [1] "0.5" "2.5"
## integer(0)
## [1] 9
## [1] "0.7" "2.5"
## integer(0)
## [1] 10
## [1] "0.9" "2.7"
## integer(0)
## [1] 9
## [1] "1.1" "3.1"
## integer(0)
## [1] 10
## [1] "1.4" "3.8"
## integer(0)
## [1] 10
## [1] "1.6" "4.2"
## integer(0)
## [1] 9
## [1] "2.1" "3.8"
## integer(0)
## [1] 8
## [1] "2.2" "3.8"
## integer(0)
## [1] 8
## [1] "2.6" "3.8"
## integer(0)
## [1] 8
## [1] "2.8" "3.6"
## integer(0)
## [1] 8
## [1] "2.5" "3.7"
## integer(0)
## [1] 10
## [1] "2.5" "4.2"
## integer(0)
## [1] 11
## [1] "1.8" "4.4"
## integer(0)
## [1] 12
## [1] "1.2" "4.2"
## integer(0)
## [1] 12
## [1] "0.9" "3.7"
## integer(0)
## [1] 9
## [1] "0.9" "3.1"
## integer(0)
## [1] 9

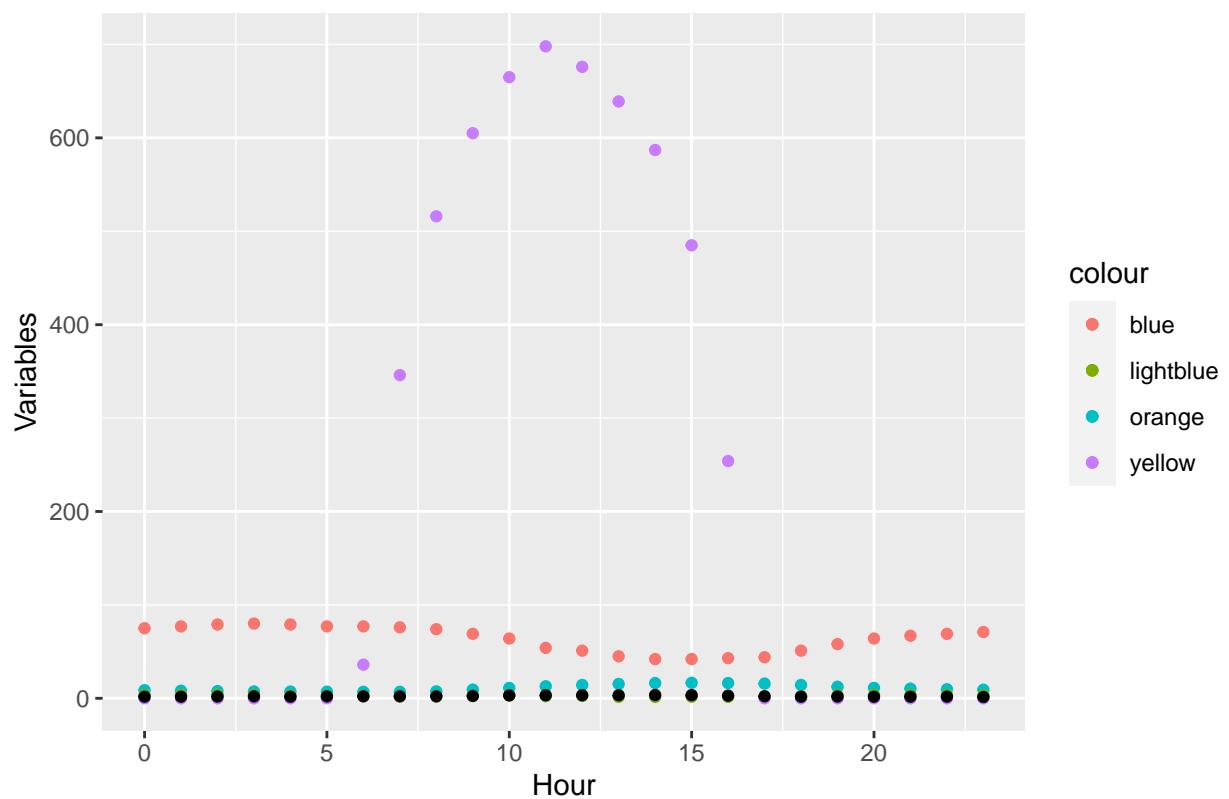
```

```
## [1] "1.0" "2.8"
## integer(0)
## [1] 8
## [1] "0.8" "2.6"
## integer(0)
## [1] 9
## [1] "1.1" "2.7"
## integer(0)
## [1] 9
## [1] "1.3" "2.4"
## [1] "DEBUG STATS: # of NAs, # of Unique Vals, Min, Max"
## integer(0)
## [1] 12
## [1] "Apr" "Sep"
## integer(0)
## [1] 24
## [1] 0 23
## integer(0)
## [1] 161
## [1] 6.5 34.7
## integer(0)
## [1] 98
## [1] 1.5 13.1
## integer(0)
## [1] 68
## [1] 23 91
## integer(0)
## [1] 134
## [1] 0 919
## integer(0)
## [1] 37
## [1] 0.5 4.4
```

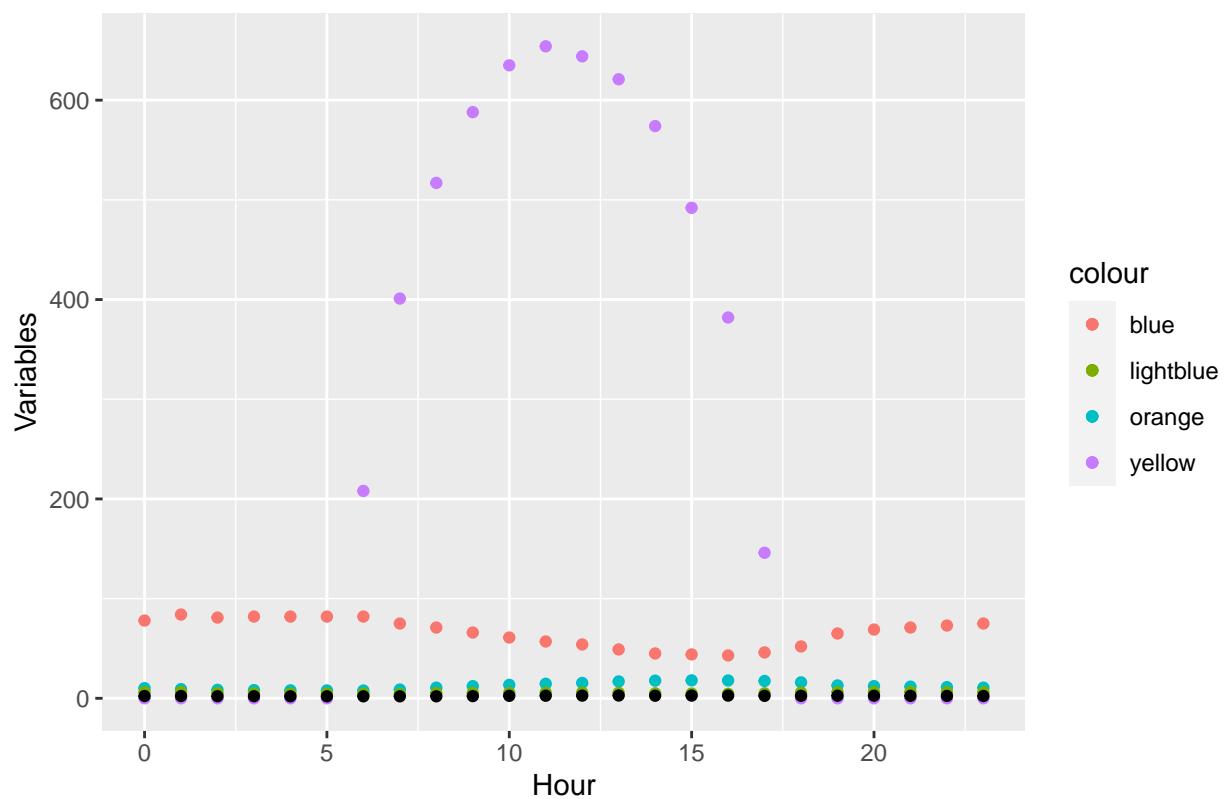
Jan Data: USA\_CA\_UC–Davis–University.AP



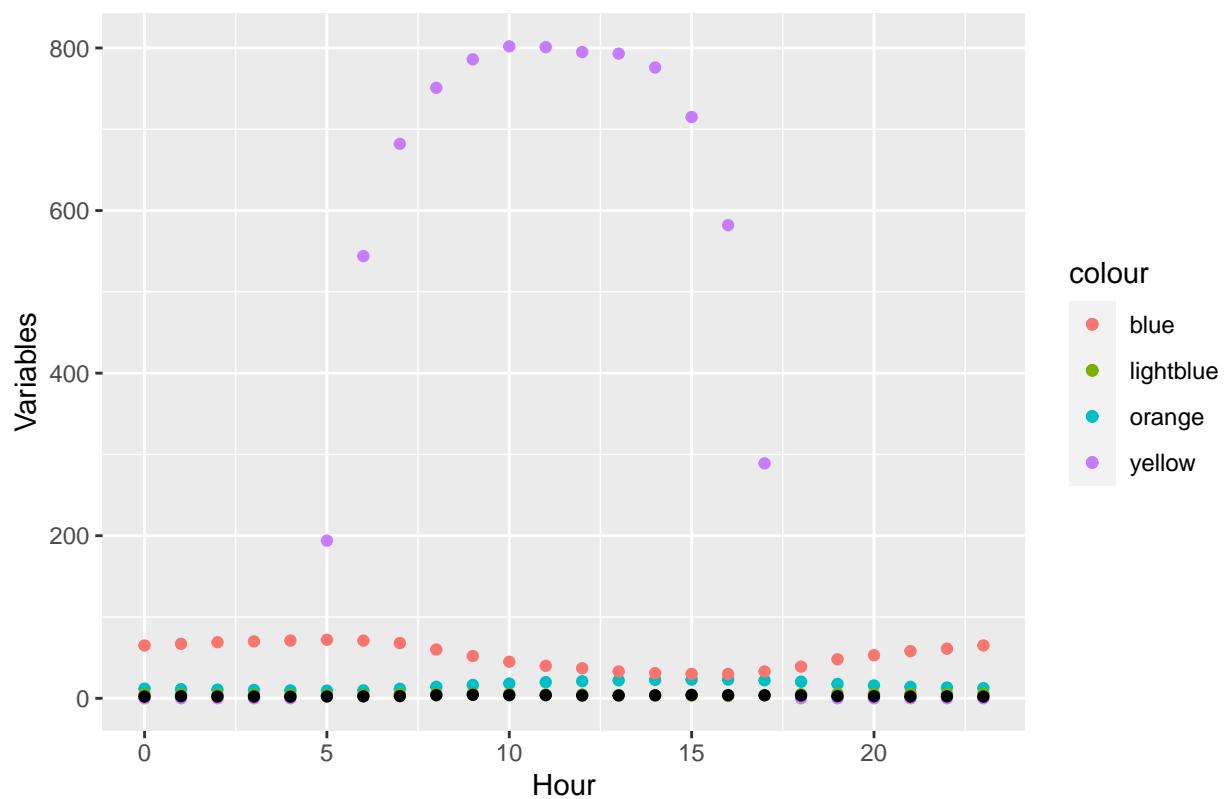
Feb Data: USA\_CA\_UC–Davis–University.AP



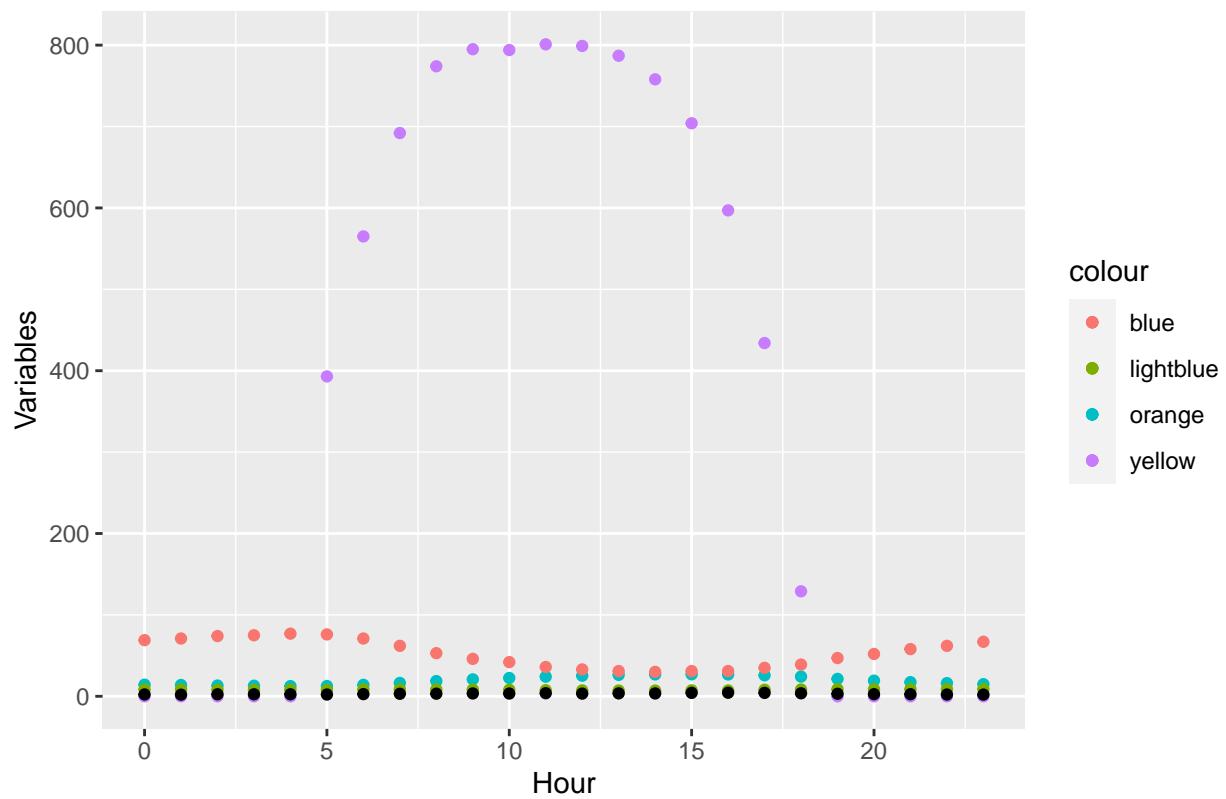
Mar Data: USA\_CA\_UC–Davis–University.AP



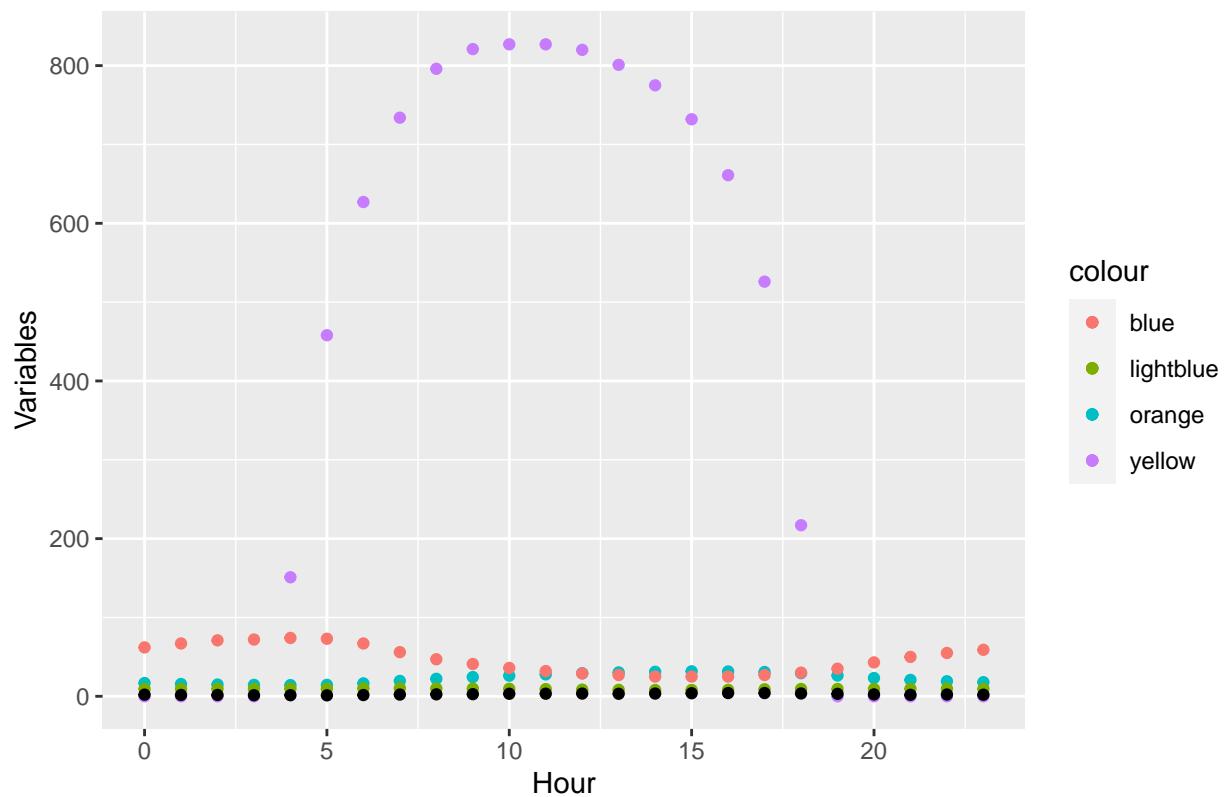
### Apr Data: USA\_CA\_UC-Davis-University.AP



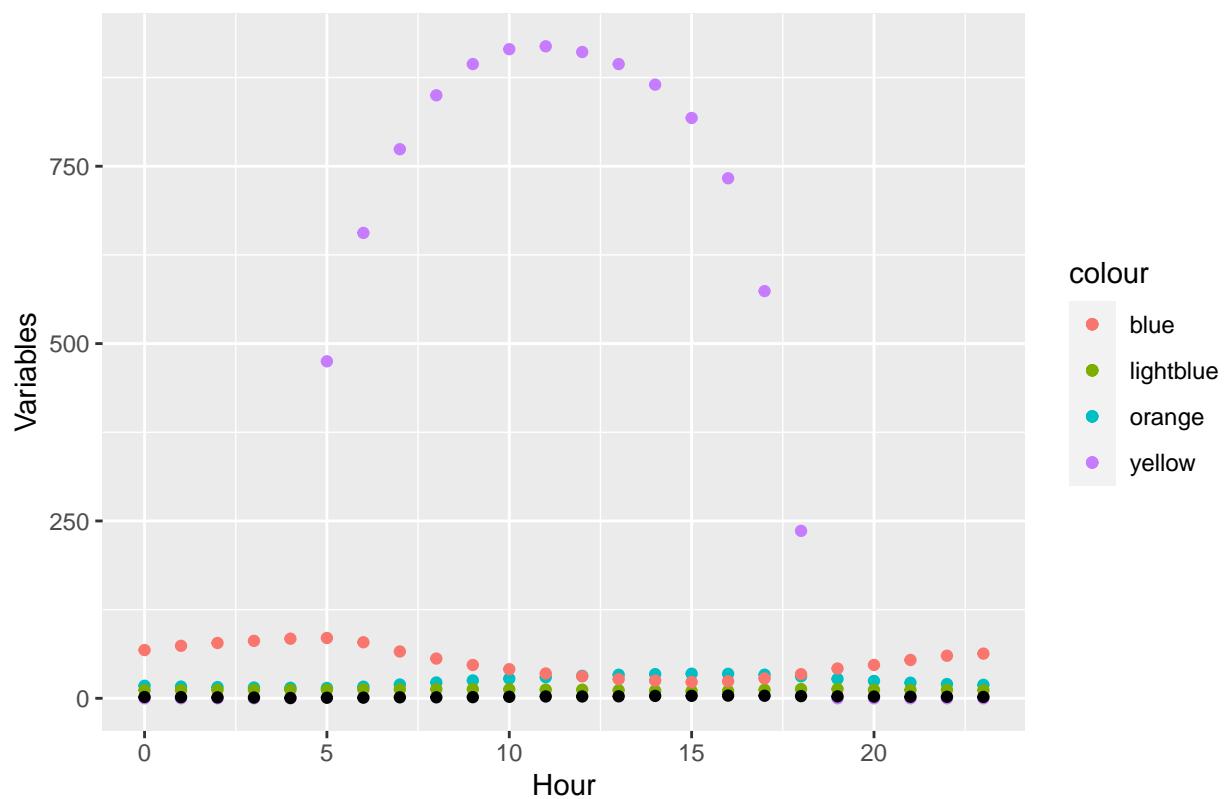
### May Data: USA\_CA\_UC-Davis–University.AP



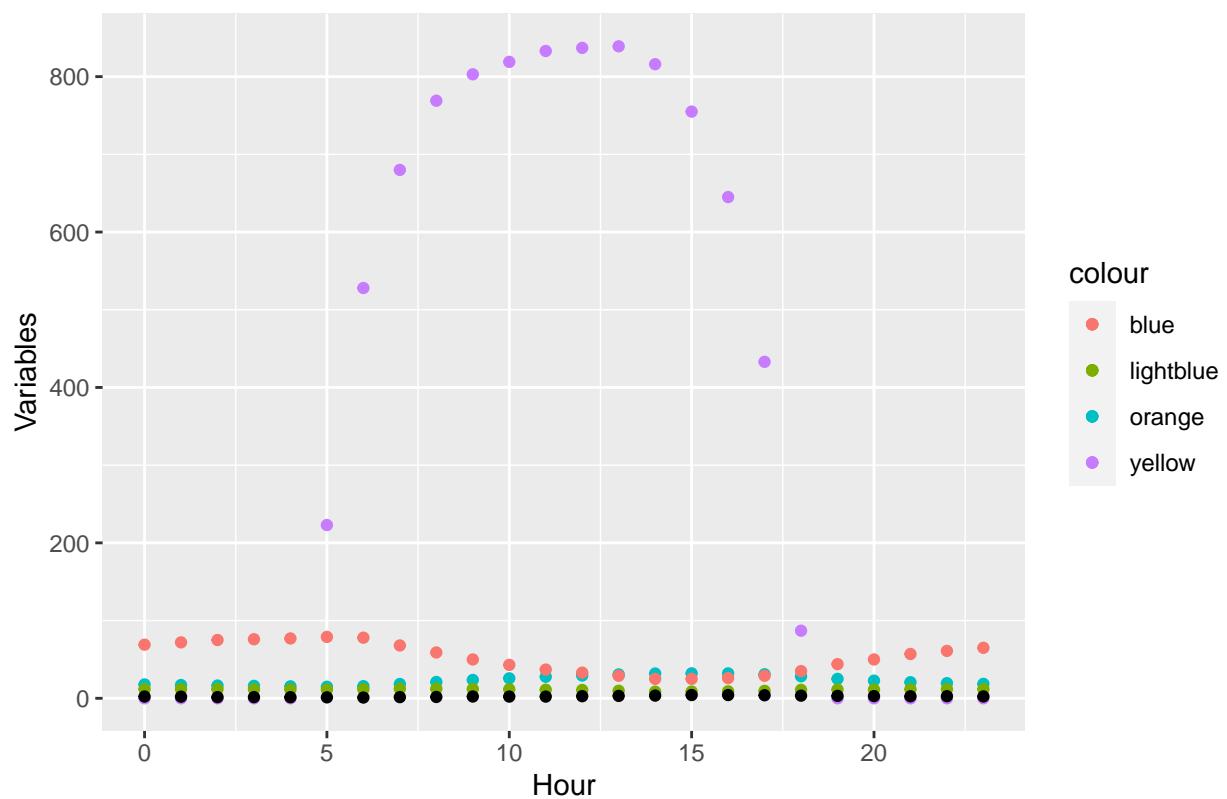
### Jun Data: USA\_CA\_UC–Davis–University.AP



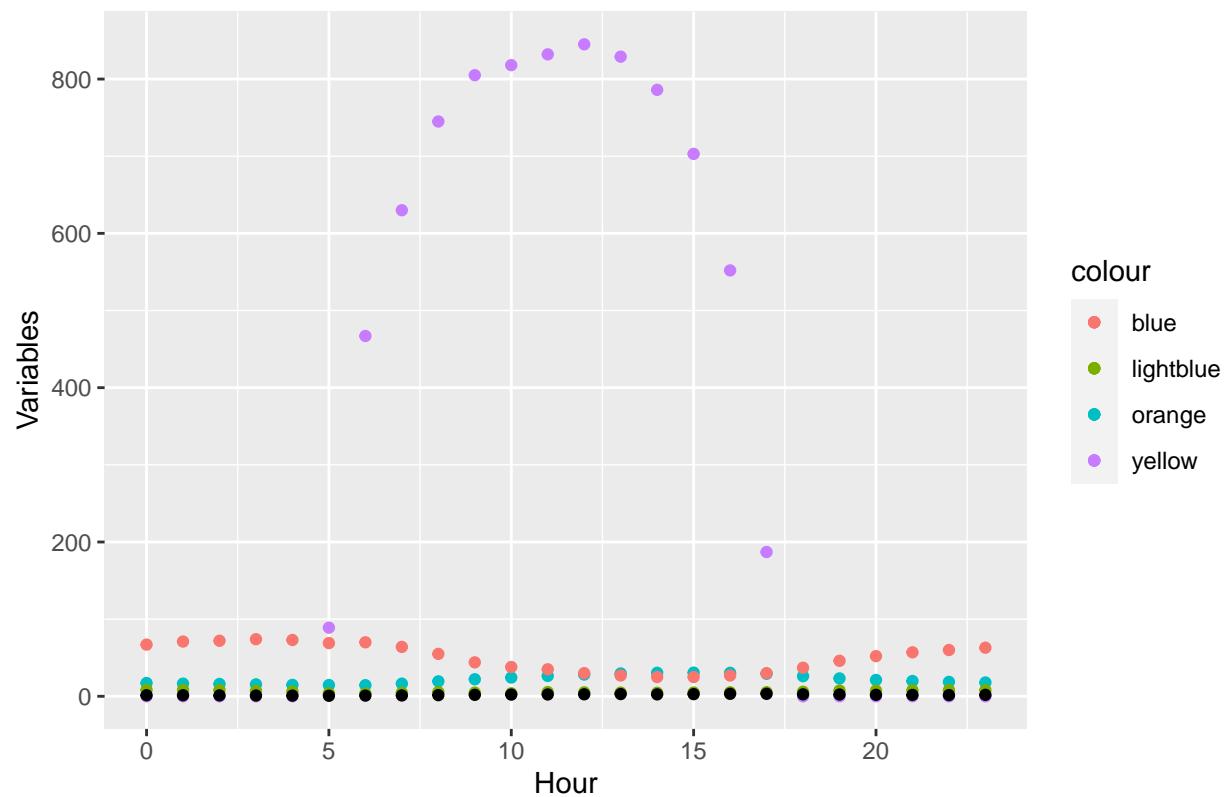
Jul Data: USA\_CA\_UC–Davis–University.AP



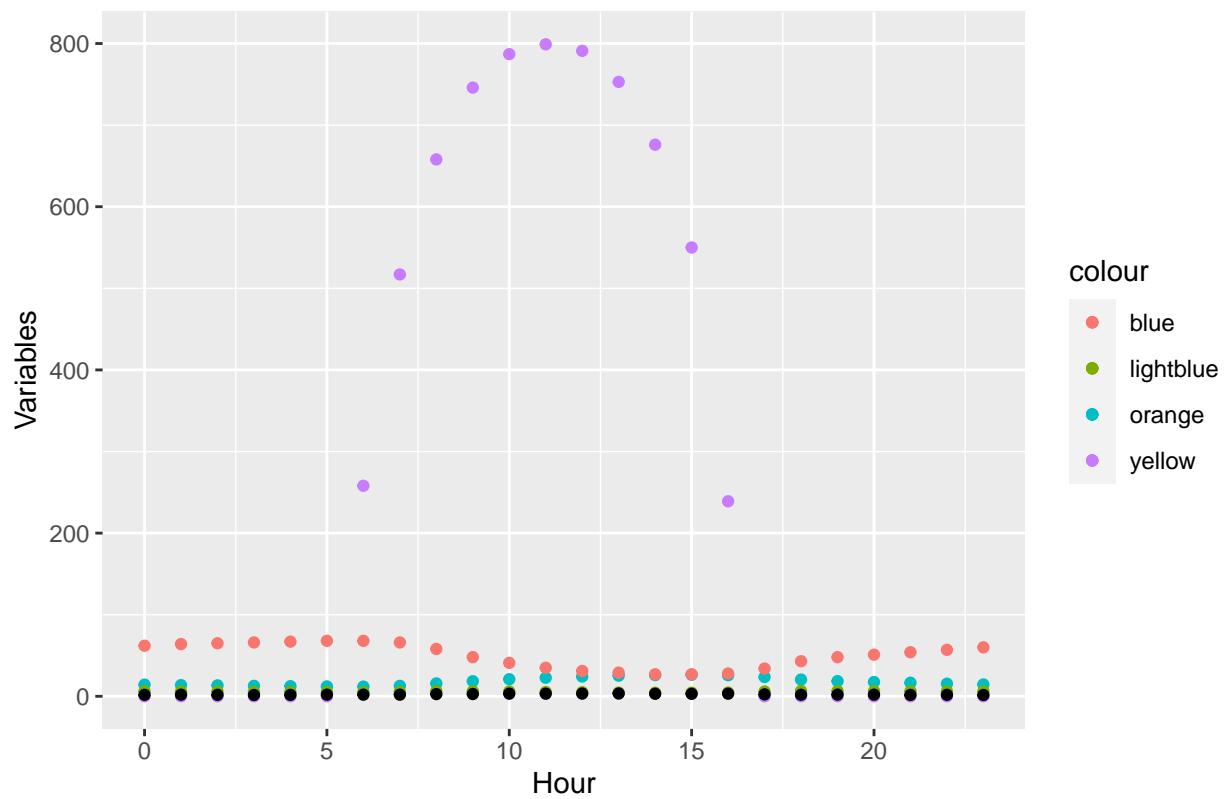
Aug Data: USA\_CA\_UC–Davis–University.AP



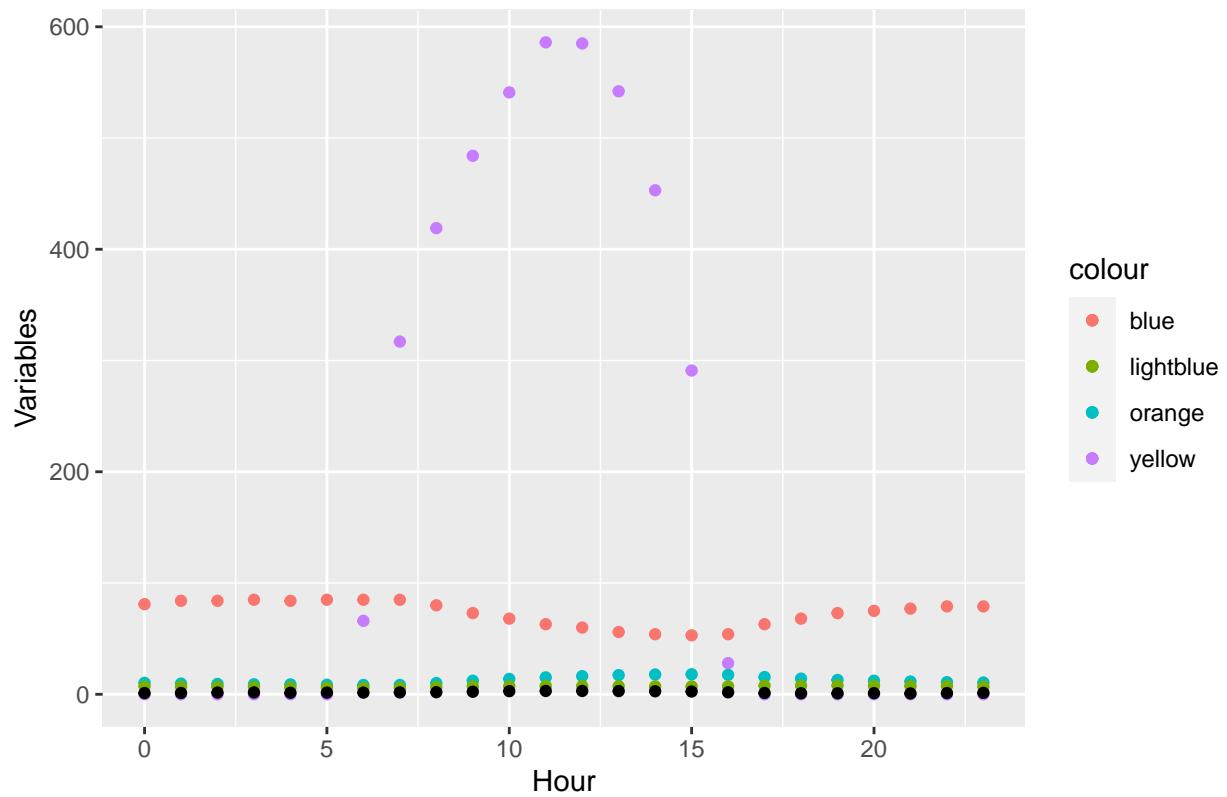
### Sep Data: USA\_CA\_UC–Davis–University.AP



### Oct Data: USA\_CA\_UC-Davis-University.AP



Nov Data: USA\_CA\_UC–Davis–University.AP



### Dec Data: USA\_CA\_UC–Davis–University.AP

