Buoy 41002

Isaac Allen

October 23, 2016

The Data

Buoy 41002 South of Cape Hatteras

The data analyzed here is standard metereological data for the Buoy 225 Nautical Miles South of Cape Hatteras. This data, especially considering the location, could be useful in weather and hurricane prediction.

The variables consist of two to four letter variable names. They are documented here:

YY, MM, DD Year, month and day respectively. hh, mm Hour and minute respectively. WD, WDIR Wind direction in degrees clockwise from North. WSPD Wind speed averaged over an eight-minute period. (m/s) GST Peak gust speed in the eight minute period. (m/s) WVHT Significant wave height. (m) DPD Dominant wave period, the period with maximum wave energy (s) APD Average wave period during a 20-minute period (s) MWD Wave direction during DPD. Degrees clockwise from North. PRES, BAR Sea level pressure. (hPa) ATMP Air temperature. (C) WTMP Sea surface temperature. (C) DEWP Dewpoint tempurate at the height of ATMP. (C) VIS Station visibility. (Nautical Miles) TIDE Water level in feet above or below mean lower low water.

More information on the variables can be found on NOAA's website. http://www.ndbc.noaa.gov/measdes.shtml

Combining The Files

With each year being save as its own file on their website, the first step was to save all the files into the "Raw Data" folder, and save them as "YYYY.txt", where YYYY represents the four digit year. Afterwards I created a list of all the file names in R, which allowed me to create a script that is capable of working with any of the metereological data from NOAA. After saving the new files into the "Raw Data" folder with the way, and then changing the list appropriately, there is very little else that must be considered before running the script for cleaning.

```
years <- c(1973:2015)
years <- years[years != 2009 & years != 2011]
files <- paste("Raw Data/", years, ".txt", sep = "")
head(files)
```

```
## [1] "Raw Data/1973.txt" "Raw Data/1974.txt" "Raw Data/1975.txt"
## [4] "Raw Data/1976.txt" "Raw Data/1977.txt" "Raw Data/1978.txt"
```

Evaluating Variables

Not all of the data files use the same variable names. Leaving these differences unidentified and unchanged could lead to row overlap, surplus columns, and general confusion. Creating a matrix of variables woked well in identifying the different combinations of variables. The function below will create such a matrix. With this information and a little bit of research, it is easy to find an ideal combination of variables for the final dataframe.

From this function I found that BAR = PRES, YYYY = YY, X.YY = YY, #YY = YY, WD = WDIR. All other variable names are unique.

```
variable_list <- function(file_list) {
    variables = matrix(0, nrow = length(file_list), ncol = 18)
    for (i in c(1:length(file_list))) {
        names = colnames(read.table(file_list[i], header = TRUE, fill = TRUE,
            comment.char = ""))
        variables[i, ] = c(names, rep("NA", times = 18 - length(names)))
    }
    variables
}
all_variables <- variable_list(files)
tail(all_variables)
```

```
##       [,1]   [,2] [,3] [,4] [,5] [,6]   [,7]   [,8]  [,9]   [,10] [,11]
## [36,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
## [37,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
## [38,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
## [39,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
## [40,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
## [41,] "X.YY" "MM" "DD" "hh" "mm" "WDIR" "WSPD" "GST" "WVHT" "DPD" "APD"
##       [,12] [,13]  [,14]  [,15]  [,16]  [,17] [,18]
## [36,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
## [37,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
## [38,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
## [39,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
## [40,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
## [41,] "MWD" "PRES" "ATMP" "WTMP" "DEWP" "VIS" "TIDE"
```

The function can be made to work for any series of data files, the only thing that would need to be changed is the maximum number of unique variable names, which would involve changing any occurances of 18.

Combining the Files

The first step in combining the files was to create a helper function that made all of the data files have the same number of variables and the same variables. After making appropriate adjustments here, based on your findings from the previous matrix, the script is ready to be run, and will freely clean the metereological data from NOAA.

```
same_var <- function(df) {
    df = rename(df, c(YYYY = "YY", X.YY = "YY", WD = "WDIR", BAR = "PRES"),
        warn_missing = FALSE)
    if (length(df) < 18) {
        mm = rep(99, times = length(df$YY))
        df = cbind(df, mm)
        if (length(df) < 18) {
            TIDE = rep(99, times = length(df$YY))
            df = cbind(df, TIDE)
        }
    }
    df
}
```

It will add the numeric equivalent of NA values to the added variables and change the previous variable names to a standardized variable name. keeping "YY", "WDIR", and "PRES" seemed to be the cleaner and more understandable option, however they could be changed to an individuals preference.

Next was creating a dataframe that consists of all of the smaller dataframes.

```
combine_files <- function(file_list) {
    full_data <- data.frame()
    for (i in c(1:length(file_list))) {
        temp_data = read.table(file_list[i], header = TRUE, fill = TRUE, comment.char = "")
        temp_data = same_var(temp_data)
        full_data = rbind(full_data, temp_data)
    }
    full_data
}

combined_data <- combine_files(files)
(head(combined_data))
```

```
##   YY MM DD hh WDIR WSPD GST WVHT DPD APD MWD PRES ATMP WTMP DEWP VIS mm
## 1 73 12  1  1  149  1.5  99   99  99  99 999 1031 15.5  999  5.4  99 99
## 2 73 12  1  2  145  0.3  99   99  99  99 999 1031 13.9  999  7.3  99 99
## 3 73 12  1  3  315  1.4  99   99  99  99 999 1031 11.4  999  6.5  99 99
## 4 73 12  1  4  312  1.6  99   99  99  99 999 1031 10.4  999    6  99 99
## 5 73 12  1  5  315  2.2  99   99  99  99 999 1031  9.5  999  6.2  99 99
## 6 73 12  1  6  314  2.4  99   99  99  99 999 1031  9.2  999  5.8  99 99
##   TIDE
## 1   99
## 2   99
## 3   99
## 4   99
## 5   99
## 6   99
```

Each new data file is made into a dataframe and given the same variables with the helper function, and then added to the larger dataframe. This recursive process will . . . ..

Removing Large Groups of NA

There exists entire columns and rows with only NA values. In order to remove them, it is imperitive to understand the structure.

First, the few occurances of text within the data used to explain need to be removed. Then, the NA numbers to be changed to actual NA values, which were 99, 999, and 9999 depending on the number of digits the maximum value for a column had. This meant I had to find the maximum value and then choose which number to convert to NA; a column could include 99 or 999 in its range, so I wanted to make sure not to remove any legitimate data.

Finally I removed the columns that had only NA values. This was all done with the function below, which will work for all dataframes set up completely numerically with the same NA coding.

```
no_na_var <- function(dataset) {
    j = c(1:length(dataset))
    dataset[, j] = lapply(dataset[, j], function(x) as.numeric(as.character(x)))
    dataset = dataset[rowSums(is.na(dataset)) != ncol(dataset), ]
    dataset[is.na(dataset)] = -1
```

```
    for (i in c(1:length(dataset))) {
        if (max(dataset[, i]) %in% c(99, 999, 9999)) {
            dataset[, i][dataset[, i] == max(dataset[, i])] = -1
        }
    }
    dataset[dataset == -1] <- NA
    dataset = dataset[colSums(!is.na(dataset)) > 0]
    dataset
}

reduced_data <- no_na_var(combined_data)
```

The Tide and the Visability variables were removed, as they only contained NA values.

Unecessary Data

I thought there might be some data that gives no information - data with only the date and time marked without any actual information collected. So I removed rows which only contained date, time, and NA values

```
nand_time <- function(dataset) {
    dataset[, 1][dataset[, 1] < 100] = paste("19", dataset[, 1][dataset[, 1] <
        100], sep = "")
    times = match(c("YY", "MM", "DD", "hh", "mm"), colnames(dataset))
    i = c(1:length(dataset))
    times = subset(i, !(i %in% times))
    new_data = filter(dataset, (rowSums(dataset[, times], na.rm = TRUE)) !=
        0)
    new_data
}


reduced_rows <- nand_time(reduced_data)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

Only about 4,000 rows were removed, as shown below by the lengths, but they did not add anything to our dataset.

```
## [1] 258496
```

```
## [1] 254436
```

Combining Time Variables

The hour and minute variables can be combined and so can the year, month, and day. I also fixed the date column to look better.

```
time_comb <- function(dataset) {
    times = match(c("hh", "mm"), colnames(dataset))
    dataset$mm[is.na(dataset$mm)] = "00"
    unite(dataset, "Time", times, sep = ":")
}
```

```
date_comb <- function(dataset) {
    dates = match(c("YY", "MM", "DD"), colnames(dataset))
    unite(dataset, "Date", dates, sep = "-")
}

reduced_vars <- time_comb(reduced_rows)
reduced_vars <- date_comb(reduced_vars)

time_fix <- function(dataset) {
    dataset[, 1] = as.Date(dataset[, 1], format = "%Y-%m-%d")
    dataset
}

reduced_vars <- time_fix(reduced_vars)

(head(reduced_vars))
```

```
##         Date Time WDIR WSPD GST WVHT DPD APD MWD PRES ATMP WTMP DEWP
## 1 1973-12-01 1:00  149  1.5  NA   NA  NA  NA  NA 1031 15.5   NA  5.4
## 2 1973-12-01 2:00  145  0.3  NA   NA  NA  NA  NA 1031 13.9   NA  7.3
## 3 1973-12-01 3:00  315  1.4  NA   NA  NA  NA  NA 1031 11.4   NA  6.5
## 4 1973-12-01 4:00  312  1.6  NA   NA  NA  NA  NA 1031 10.4   NA  6.0
## 5 1973-12-01 5:00  315  2.2  NA   NA  NA  NA  NA 1031  9.5   NA  6.2
## 6 1973-12-01 6:00  314  2.4  NA   NA  NA  NA  NA 1031  9.2   NA  5.8
```

Seperate Files

At this point it seemed like a good time to save and figure out how to best seperate the files. I created another matrix that contains all of the combinations of variables that exist to look for some kind of pattern.

```
var_combinations <- function(dataset) {
    i = c(1:length(dataset[, 1]))
    mat = dataset
    mat[!is.na(mat)] = col(mat)[!is.na(mat)]
    mat = mat = distinct(mat)
    mat
}

variable_combos <- var_combinations(reduced_rows)

length(variable_combos[, 1])
```

```
## [1] 130
```

There are 130 rows, meaning 130 different combinations of variables with no distinct patterns for seperation. Analysis will have to be done by subsetting the information to what is needed, as it would be impractical to create 130 different files for this data.

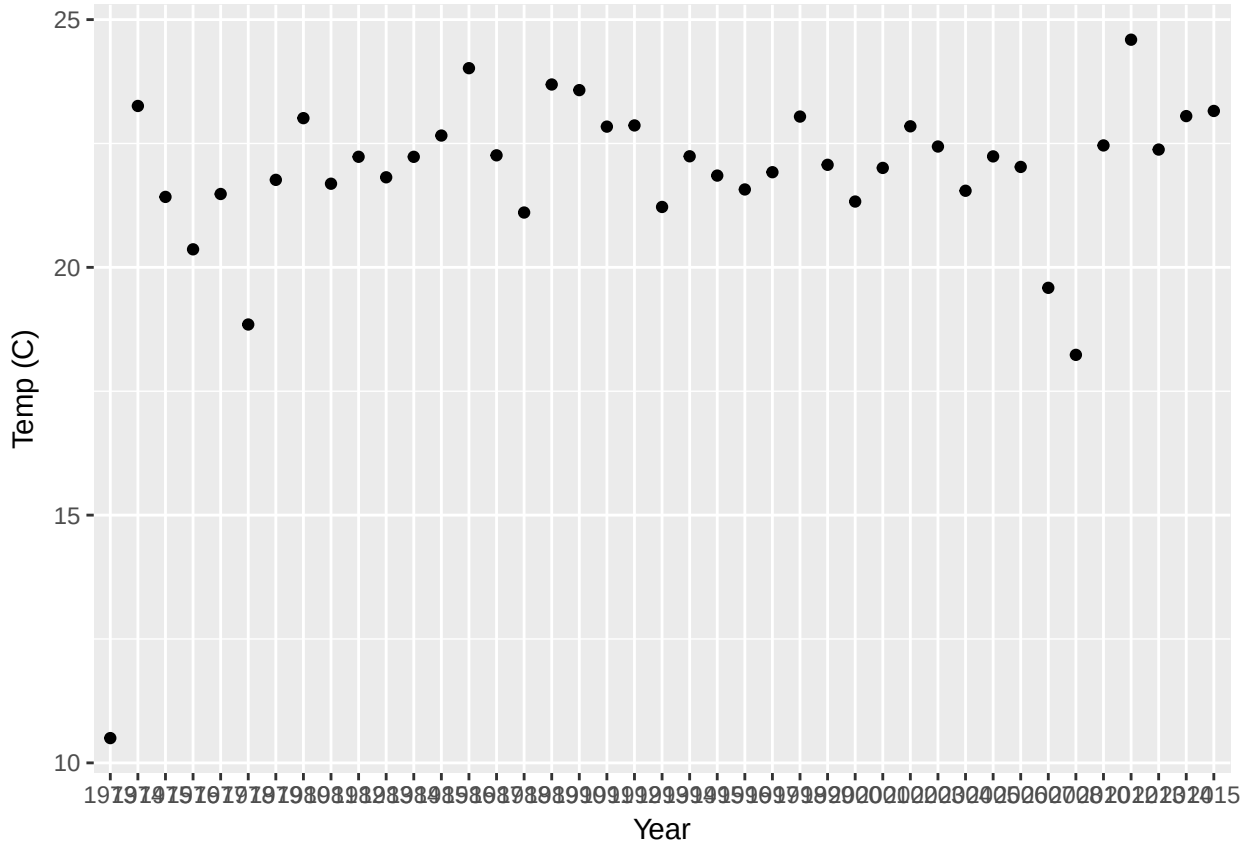
Saving Files

I saved both csv and RDS files, one file with the dates and times together and one with them seperate.

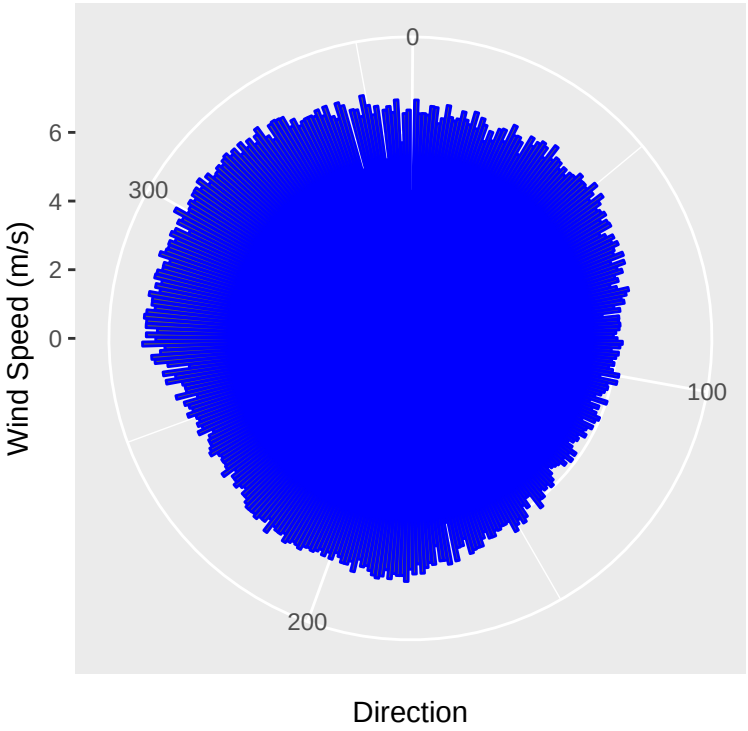
```
write.csv(reduced_vars, "Reduced Data - combined time.txt")
write.csv(reduced_rows, "Reduced Data.txt")

saveRDS(reduced_vars, "Reduced Data - combined timeRDS.rds")
saveRDS(reduced_rows, "Reduced DataRDS.rds")
```

Graphing and Summary Statistics

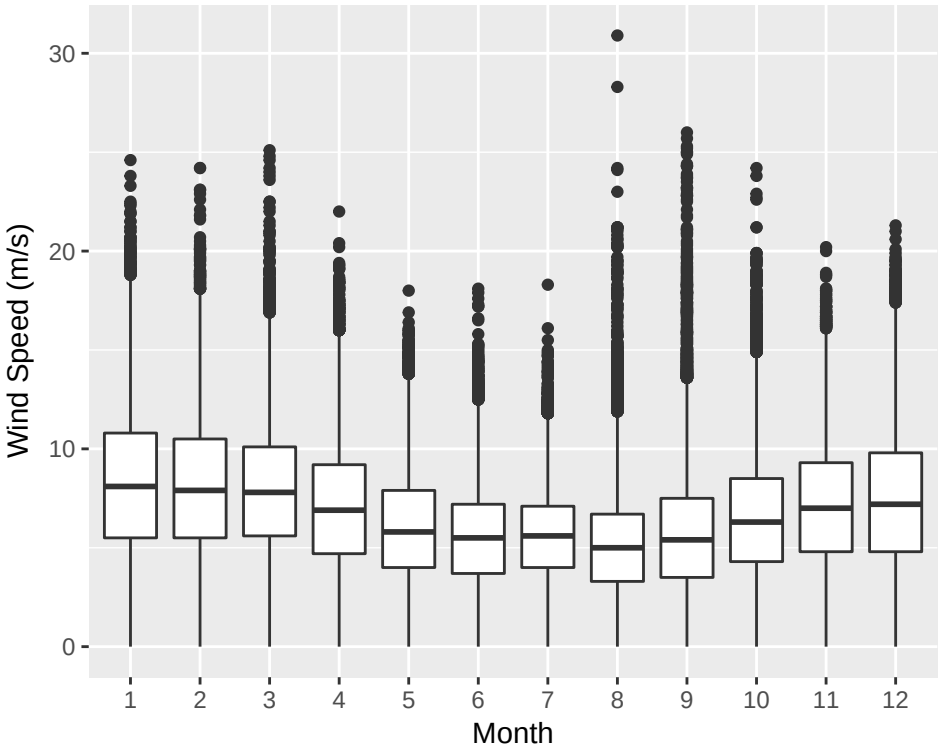
Graphs



A graph showing the average temperature versus year.



Average wind strength for each wind direction, pointing in the corresponding direction. North corresponds to 0.



Boxplot of wind speeds based on month.

Summary Statistics

I created a function to find the mean and variance of all variables not time related.

```
summary_stat <- function(dataset) {
    no_time = subset(colnames(dataset), !(colnames(dataset) %in% c("Date", "Time")))
    mat = matrix(0, nrow = 2, ncol = length(no_time))
    for (i in c(1:length(no_time))) {
        mat[1, i] = mean(dataset[, i + 2], na.rm = TRUE)
        mat[2, i] = var(dataset[, i + 2], na.rm = TRUE)
    }
    rownames(mat) <- c("Mean", "Variance")
    colnames(mat) <- no_time
    mat
}

statis <- summary_stat(time_tgth)
(statis)
```

```
##               WDIR      WSPD       GST      WVHT      DPD      APD
## Mean      182.1368  6.648561  8.356476 1.8181583 8.067198 5.773002
## Variance 8974.4841 10.863433 15.688261 0.9957879 4.566559 2.735569
##                MWD      PRES     ATMP      WTMP     DEWP
## Mean      142.1153 1017.7688 22.23320 24.387490 17.61682
## Variance 8614.1278   32.0931 18.31882  7.334719 36.61909
```