ESP 106 Lab 4

Leo Hecht

January 31, 2022

In this lab we will look at daily tide data downloaded from NOAA’s Tides and Currents API (Application Programming Interface) for six cities around the US. I used the API to obtain six csv files containing data for tide gauges in each city. These are in the “Data” folder. The tide gauges have numerical codes that correspond to the city as follows:

1. Boston: 8443970
2. New York: 8518750
3. Baltimore: 8574680
4. Charleston: 8665530
5. Miami: 8723214
6. Corpus Christi: 8775296

### Part 1 - Monday Jan 29th

# 1. Create a data frame containing data on the city name and tide gauge ID given above

city <- c("Boston", "New York", "Baltimore", "Charleston", "Miami", "Corpus Christi")  
  
code <- as.factor(c(8443970, 8518750, 8574680, 8665530, 8723214, 8775296))  
  
d <- data.frame(city, code)  
  
d

## city code  
## 1 Boston 8443970  
## 2 New York 8518750  
## 3 Baltimore 8574680  
## 4 Charleston 8665530  
## 5 Miami 8723214  
## 6 Corpus Christi 8775296

# 2a. Use a for-loop to read in the csv files and bind them together into a single data frame. Add a column to the data frame giving the name of the city the data is from.

csvfiles <- list.files("C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data", full.names = TRUE)  
  
csvfiles

## [1] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8443970.csv"  
## [2] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8518750.csv"  
## [3] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8574680.csv"  
## [4] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8665530.csv"  
## [5] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8723214.csv"  
## [6] "C:\\Users\\Leo Hecht\\Documents\\ESP 106\\Data/8775296.csv"

nchar(csvfiles[1])

## [1] 53

data <- read.csv(csvfiles[1])  
data$code <- substr(csvfiles[1], start = 43, stop = 49)  
  
for(i in 2:length(csvfiles)) {  
   
 temp <- read.csv(csvfiles[i])   
   
 temp$code <- substr(csvfiles[i], start = 43, stop = 49)  
   
 data <- rbind(data, temp)  
}  
  
data <- merge(data, d, by = "code")

# 2b. Take a look at your data frame - is this in a tidy format

# It is mostly tidy except for the lack of a unique date column.

We are going to examine the question of whether these gauges show evidence of rising sea levels. One of the first things we have to deal with is the issue of dates.

Your data frame right now has one column with a year and one with the month. We are going to combine these into a single column, and use the lubridate pacakage to tell R to interpret that column as a date

# 3a. Create a new column named “Date” that has the first day of the month for that row in the format YYYY-MM-01 where YYYY is the data in the Year column and MM is the data in the Month column.

?paste0

## starting httpd help server ... done

data$Date <-paste0(data$Year, sep = "-", data$Month, sep = "-", 01)

3b. Use the ymd() (i.e. year-month-day) function from the lubridate package to convert your new date column to a date object in R

library(lubridate) #this is a great package for handling date and time objects in R. If you don't have it already install it using install.packages("lubridate")

## Warning: package 'lubridate' was built under R version 4.2.3

##   
## Attaching package: 'lubridate'

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

data$Date <- ymd(data$Date)

Now lets use ggplot to make some cool graphs of this data using ggplot.

# 4. Make a plot showing data from all 6 gauges on the same plot. Use different colors to distinguish lines for the different cities. See the example plot uploaded to Canvas (Plot 1)

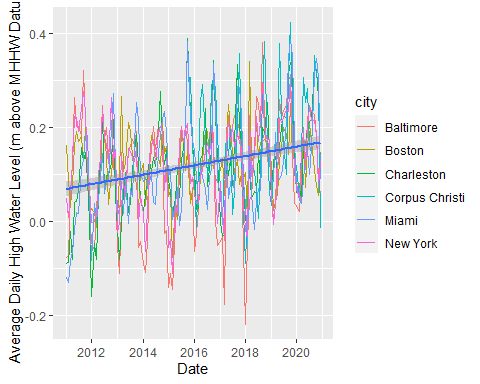
* Plot the date on the x axis and MHHW (mean higher high water - i.e. the average daily high water level) on the y axis Make sure to add proper axis labes and units (using +labs(x=““,y=”“))
* Add a single best-fit line through the full data set using geom\_smooth(method=“lm”) - note that by default ggplot will fit one best fit line for each city. To override this specify the aestetic mapping (aes()) again within the geom\_smooth function and add the argument inherit.aes=FALSE

library(ggplot2) #if you don't already have ggplot2 then install it using install.packages("ggplot2")

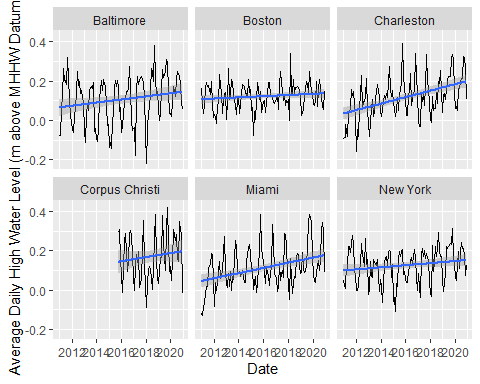
## Warning: package 'ggplot2' was built under R version 4.2.3

x11()  
  
plot1 <- ggplot(data, aes(Date, MHHW, col = city)) +  
 geom\_line() +  
 labs(x = "Date", y = "Average Daily High Water Level (m above MHHW Datum)") +  
 geom\_smooth(method = "lm", inherit.aes = FALSE, aes(Date, MHHW))  
  
plot1

## `geom\_smooth()` using formula = 'y ~ x'



# 5. Now make a slightly different plot with the same x and y variables, but use facet\_wrap() to make a subplot separately for each city. Add a best-fit line for each subplot. See the example plot uploaded to Canvas (Plot 2)

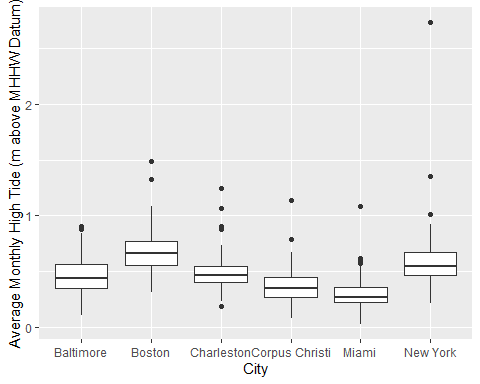


### Part 2 - Wednesday Jan 31st

In this part of the lab we will identify some outliers, and practice running regressions

# 6. Make a box plot showing the distribution of the highest tides each month (“Highest” column in the NOAA data) . (Ideally practice using ggplot by using geom\_boxplot() - put the city on the x axis and Highest on the y. But this can also be done in base R). See the example plot on Canvas (Plot 3)

ggplot(data, aes(city, Highest)) +  
 geom\_boxplot() +  
 labs(x = "City", y = "Average Monthly High Tide (m above MHHW Datum)")



Notice the very extreme value in New York City - a major outlier both within New York and compared to all the other cities

# 7a. Find the row in the data corresponding to this outlier observation

which.max(data$Highest)

## [1] 141

data[141,]

## code Year Month Highest MHHW MHW MSL MTL MLW MLLW DTL  
## 141 8518750 2012 10 2.739 0.263 0.145 -0.555 -0.587 -1.318 -1.384 -0.561  
## GT MN DHQ DLQ HWI LWI Lowest Inferred city Date  
## 141 1.647 1.463 0.118 0.066 0.81 7.23 -1.842 0 New York 2012-10-01

# 7b. What month and year did this outlier event occur in? What meteorological event happened in New York in that month that probably caused this outlier event? (Feel free to use Google - I don’t expect you to know this off hand)

# The event occurred inn October of 2012. Hurricane Sandy was likely the cause of this outlier event.

Finally, we will fit a linear model to estimate the rate of sea-level rise across these 6 cities.

# 8a. Fit a linear regression with the mean higher high water (MHHW) as the dependent variable and date (i.e. time) as the independent variable.

regression <- lm(MHHW~Date, data = data)  
  
sum\_regression <- summary(regression)  
  
sum\_regression

##   
## Call:  
## lm(formula = MHHW ~ Date, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.35710 -0.06419 0.00207 0.06161 0.27239   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.399e-01 6.002e-02 -5.663 2.23e-08 \*\*\*  
## Date 2.732e-05 3.552e-06 7.692 5.31e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09572 on 656 degrees of freedom  
## Multiple R-squared: 0.08274, Adjusted R-squared: 0.08134   
## F-statistic: 59.17 on 1 and 656 DF, p-value: 5.314e-14

# 8b. Give the estimated coefficient of the date column. Is it statistically significant (i.e. has a p-value less than 0.05)?

sum\_regression$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -3.398793e-01 6.002228e-02 -5.662553 2.231680e-08  
## Date 2.732001e-05 3.551610e-06 7.692288 5.313671e-14

datecoef <- sum\_regression$coefficients[2,1]  
  
datecoef

## [1] 2.732001e-05

pval <- sum\_regression$coefficients[2,4]  
  
pval

## [1] 5.313671e-14

# The estimated coefficient of the date column is 2.732e-05. It is statistically significant.

This coefficient gives us the average increase in high tide levels each day, across all six cities, for this ten year time frame (i.e. the units of the coefficient are in m per day).

# 8c. Using your estimated coefficient, estimate the mean increase in sea-level over the 10 year time frame from 2011-2020.

mean\_increase <- datecoef \* 365 \* 10  
  
mean\_increase

## [1] 0.09971803