ESP106-Lab4

Fran M

January 29, 2023

Lab 4

In this lab we will look at daily tidy 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. 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

Before you start: add this file to your github repository, and commit your changes throughout the time you work on it.

Part 1 - Monday

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

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.

```
#Hint: you can use list.files() to get a vector of all the file names in a directory. Setting full.name

#Hint 2: you might want to create a data frame using the first csv file. Then loop through from files 2

#Hint 3: It will be easiest to add the names of the city to the data frames immediately after reading t
```

I set my working directory, unzipped the file with the data, and created a list called tables in which each item is the data for one city. I added a column to each data frame with the appropriate city and combined the data frames into one large one called combined.

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

The data seems to be in a tidy format because each row is an entry, the columns represent different variables, and each cell is an entry.

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 as.Date to formally use Date objects

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.

```
#Hint: Use paste0() to combine data and characters (i.e. the required separators "-")
date <- c()

for(i in 1:nrow(combined)){
  date <- c(date, paste0(c(combined[i, 1], combined[i, 2], 01), collapse = "-"))
}
```

I created en empty vector called data. I then created a for loop that used the Year and Month columns to create a date for each entry.

3b. Use as.Date to convert your new date column to a date object in R

```
date <- as.Date(date)
combined$Date <- date
```

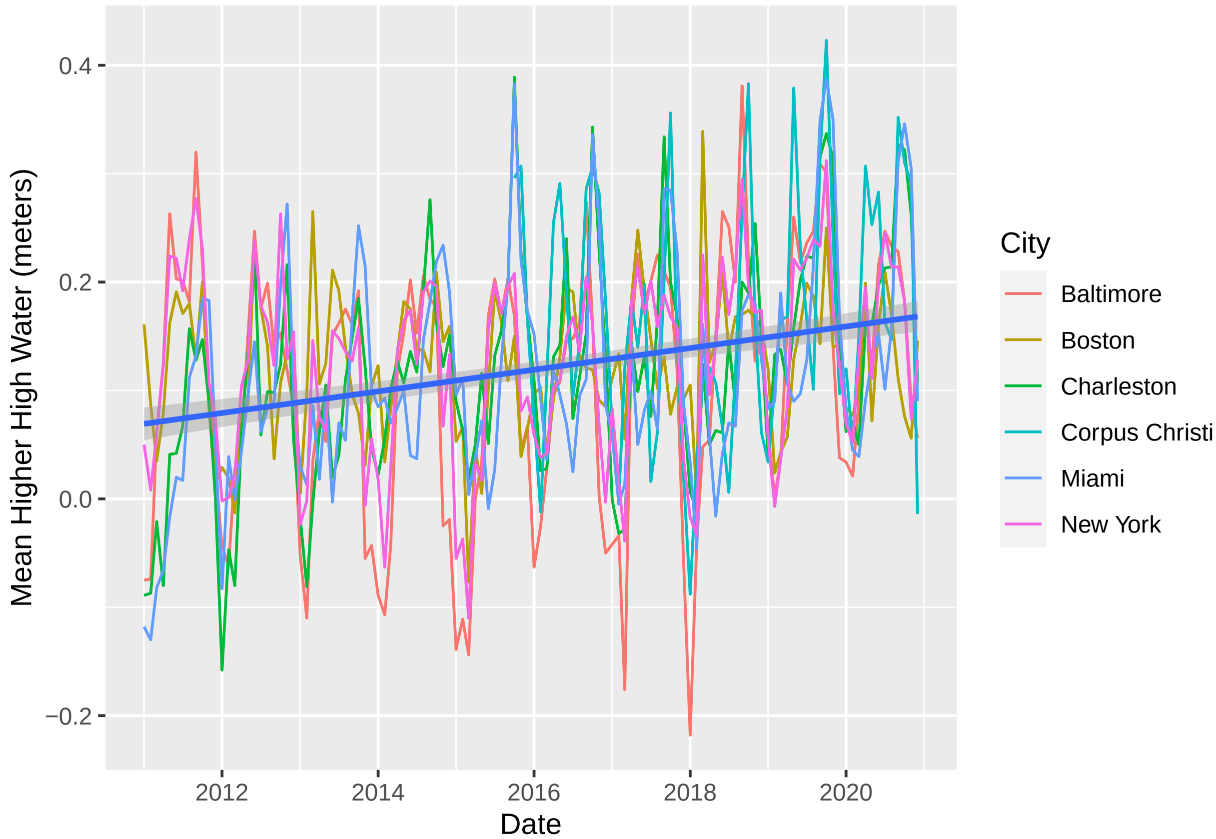
**I used the as.Date function to format the date and then added a column to the data frame called Date.

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 ggp lot 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

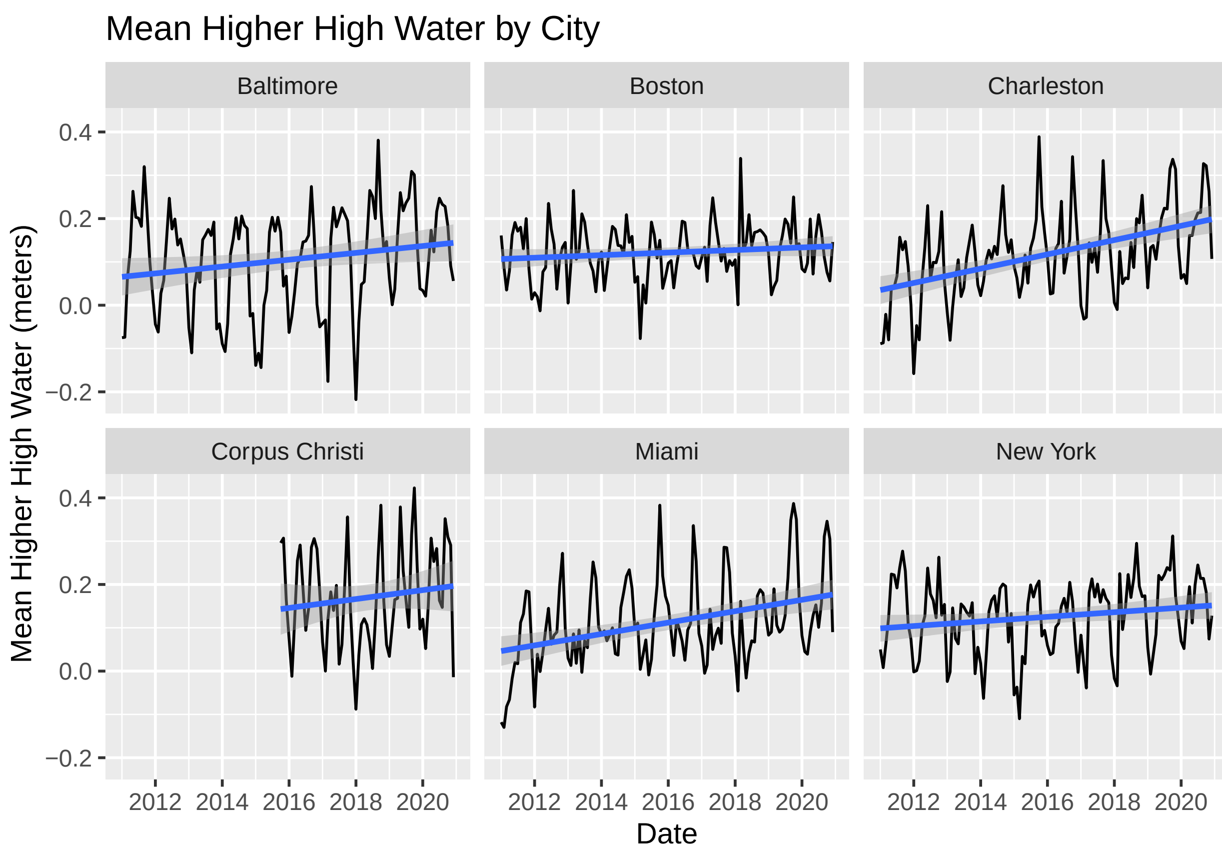
```
#if you don't already have ggplot2 then install it with install.packages("ggplot2")
library(ggplot2)
ggplot(combined, aes(x = date, y = MHHW, group = City, color = City))+
  geom_line()+
  labs(x = "Date", y = "Mean Higher High Water (meters)")+
  geom_smooth(aes(x = date, y = MHHW), method="lm", inherit.aes = FALSE)
```



This graph plots Mean Higher High Water versus Date grouped by city.

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)

```
#Hint: you should only need minor modification of the code from question 4 to make this plot
ggplot(combined, aes(x = date, y = MHHW, group = City))+
  geom_line()+
  labs(x = "Date", y = "Mean Higher High Water (meters)", title = "Mean Higher High Water by City")+
  geom_smooth(method="lm")+
  facet_wrap(~City)
```

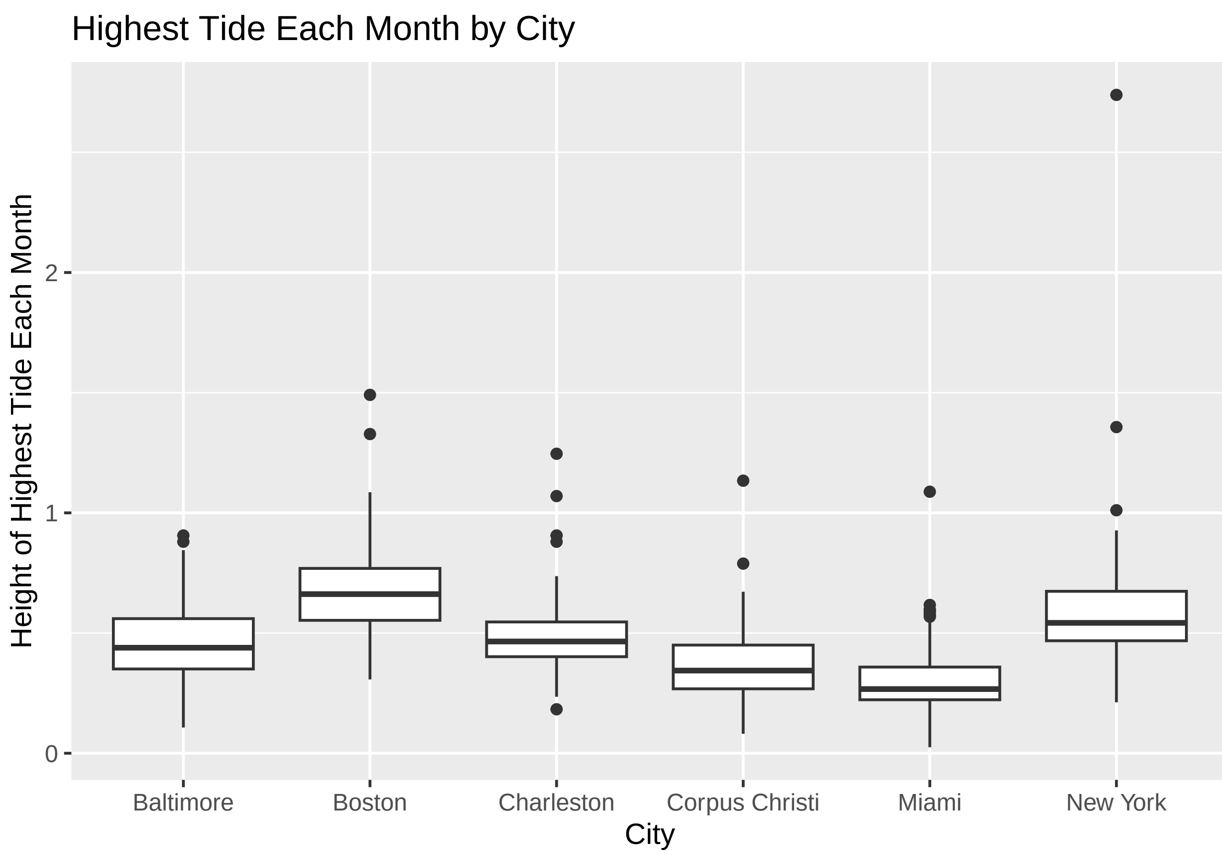


Part 2 - Wednesday

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(combined, aes(x = City, y = Highest))+
  geom_boxplot()+
  labs(x = "City", y = "Height of Highest Tide Each Month", title = "Highest Tide Each Month by City")
```



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

```
#Hint: The which.max() function might be useful here
which.max(combined$Highest)
```

```
## [1] 141
```

```
combined[141,]
```

```
##     Year Month Highest  MHHW   MHW    MSL    MTL    MLW   MLLW    DTL    GT
## 141 2012    10   2.739 0.263 0.145 -0.555 -0.587 -1.318 -1.384 -0.561 1.647
##         MN   DHQ   DLQ  HWI  LWI Lowest Inferred     City       Date
## 141 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)

Hurricane Sandy hit New York in October of 2012.

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

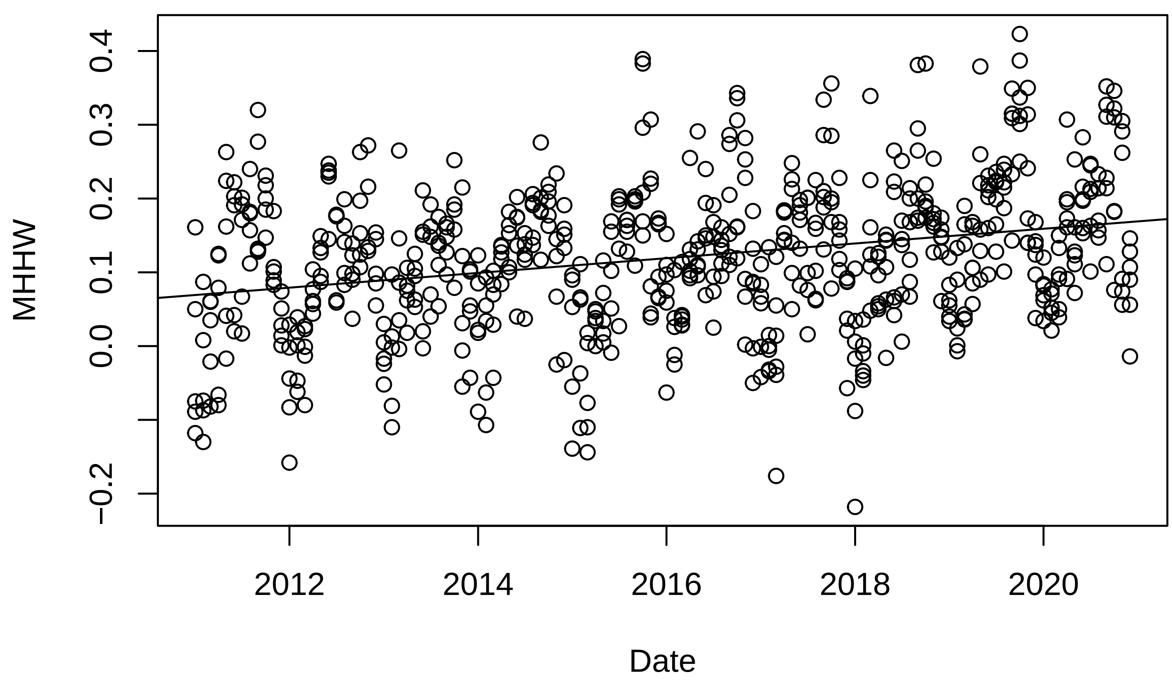
8a. Fit a linear regression with the mean higher high water (MHHW) as the dependent variable and date (i.e. time) as the independent variable.

```r
#Hint: the forumla in your lm() function is of the form y~x where y here is MHHW and x is your date col

m <- lm(MHHW ~ Date, data = combined)
m
```

```
##
## Call:
## lm(formula = MHHW ~ Date, data = combined)
##
## Coefficients:
## (Intercept)         Date
##  -3.399e-01    2.732e-05
```

```r
plot(MHHW ~ Date, data = combined)
abline(m)
```




```r
coefficients(m)
```

```
##   (Intercept)          Date
## -3.398793e-01  2.732001e-05
```

```r
summary(m)
```

```
##
## Call:
## lm(formula = MHHW ~ Date, data = combined)
##
## 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)?

The p-value is 5.314e-14, which is less than 0.05, so the estimated coefficient 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.

```
p = predict(m, combined[c(1, nrow(combined)), ])
diff(p)
```

```
##        658
## 0.09895307
```

```
predict(m)
```

```
##          1          2          3          4          5          6          7
## 0.06923779 0.07008471 0.07084967 0.07169659 0.07251619 0.07336311 0.07418271
##          8          9         10         11         12         13         14
## 0.07502963 0.07587655 0.07669615 0.07754307 0.07836267 0.07920959 0.08005651
##         15         16         17         18         19         20         21
## 0.08084879 0.08169571 0.08251531 0.08336223 0.08418183 0.08502875 0.08587567
##         22         23         24         25         26         27         28
## 0.08669527 0.08754219 0.08836179 0.08920872 0.09005564 0.09082060 0.09166752
##         29         30         31         32         33         34         35
## 0.09248712 0.09333404 0.09415364 0.09500056 0.09584748 0.09666708 0.09751400
##         36         37         38         39         40         41         42
## 0.09833360 0.09918052 0.10002744 0.10079240 0.10163932 0.10245892 0.10330584
##         43         44         45         46         47         48         49
## 0.10412544 0.10497236 0.10581928 0.10663888 0.10748580 0.10830540 0.10915232
##         50         51         52         53         54         55         56
```

```
## 0.10999924 0.11076420 0.11161112 0.11243072 0.11327764 0.11409724 0.11494416
##         57         58         59         60         61         62         63
## 0.11579108 0.11661068 0.11745760 0.11827720 0.11912412 0.11997104 0.12076332
##         64         65         66         67         68         69         70
## 0.12161024 0.12242984 0.12327676 0.12409636 0.12494328 0.12579020 0.12660980
##         71         72         73         74         75         76         77
## 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205 0.13240165
##         78         79         80         81         82         83         84
## 0.13324857 0.13491509 0.13576201 0.13658161 0.13742853 0.13824813 0.13909505
##         85         86         87         88         89         90         91
## 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037 0.14403997 0.14488689
##         92         93         94         95         96         97         98
## 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685 0.14991377 0.15067873
##         99        100        101        102        103        104        105
## 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869 0.15570561 0.15652521
##        106        107        108        109        110        111        112
## 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785 0.16152477 0.16234437
##        113        114        115        116        117        118        119
## 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434 0.16737126 0.16819086
##        120        121        122        123        124        125        126
## 0.06923779 0.07008471 0.07084967 0.07169659 0.07251619 0.07336311 0.07418271
##        127        128        129        130        131        132        133
## 0.07502963 0.07587655 0.07669615 0.07754307 0.07836267 0.07920959 0.08005651
##        134        135        136        137        138        139        140
## 0.08084879 0.08169571 0.08251531 0.08336223 0.08418183 0.08502875 0.08587567
##        141        142        143        144        145        146        147
## 0.08669527 0.08754219 0.08836179 0.08920872 0.09005564 0.09082060 0.09166752
##        148        149        150        151        152        153        154
## 0.09248712 0.09333404 0.09415364 0.09500056 0.09584748 0.09666708 0.09751400
##        155        156        157        158        159        160        161
## 0.09833360 0.09918052 0.10002744 0.10079240 0.10163932 0.10245892 0.10330584
##        162        163        164        165        166        167        168
## 0.10412544 0.10497236 0.10581928 0.10663888 0.10748580 0.10830540 0.10915232
##        169        170        171        172        173        174        175
## 0.10999924 0.11076420 0.11161112 0.11243072 0.11327764 0.11409724 0.11494416
##        176        177        178        179        180        181        182
## 0.11579108 0.11661068 0.11745760 0.11827720 0.11912412 0.11997104 0.12076332
##        183        184        185        186        187        188        189
## 0.12161024 0.12242984 0.12327676 0.12409636 0.12494328 0.12579020 0.12660980
##        190        191        192        193        194        195        196
## 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205 0.13240165
##        197        198        199        200        201        202        203
## 0.13324857 0.13406817 0.13491509 0.13576201 0.13658161 0.13742853 0.13824813
##        204        205        206        207        208        209        210
## 0.13909505 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037 0.14403997
##        211        212        213        214        215        216        217
## 0.14488689 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685 0.14991377
##        218        219        220        221        222        223        224
## 0.15067873 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869 0.15570561
##        225        226        227        228        229        230        231
## 0.15652521 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785 0.16152477
##        232        233        234        235        236        237        238
## 0.16234437 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434 0.16737126
##        239        240        241        242        243        244        245
```

```
## 0.16819086 0.06923779 0.07008471 0.07084967 0.07169659 0.07251619 0.07336311
##        246        247        248        249        250        251        252
## 0.07418271 0.07502963 0.07587655 0.07669615 0.07754307 0.07836267 0.07920959
##        253        254        255        256        257        258        259
## 0.08005651 0.08084879 0.08169571 0.08251531 0.08336223 0.08418183 0.08502875
##        260        261        262        263        264        265        266
## 0.08587567 0.08669527 0.08836179 0.08920872 0.09005564 0.09082060 0.09166752
##        267        268        269        270        271        272        273
## 0.09248712 0.09333404 0.09415364 0.09500056 0.09584748 0.09666708 0.09751400
##        274        275        276        277        278        279        280
## 0.09833360 0.09918052 0.10002744 0.10079240 0.10163932 0.10245892 0.10330584
##        281        282        283        284        285        286        287
## 0.10412544 0.10497236 0.10581928 0.10663888 0.10748580 0.10830540 0.10915232
##        288        289        290        291        292        293        294
## 0.10999924 0.11076420 0.11161112 0.11243072 0.11327764 0.11409724 0.11494416
##        295        296        297        298        299        300        301
## 0.11579108 0.11661068 0.11745760 0.11827720 0.11912412 0.11997104 0.12076332
##        302        303        304        305        306        307        308
## 0.12161024 0.12242984 0.12327676 0.12409636 0.12494328 0.12579020 0.12660980
##        309        310        311        312        313        314        315
## 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205 0.13240165
##        316        317        318        319        320        321        322
## 0.13324857 0.13406817 0.13491509 0.13576201 0.13658161 0.13742853 0.13824813
##        323        324        325        326        327        328        329
## 0.13909505 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037 0.14403997
##        330        331        332        333        334        335        336
## 0.14488689 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685 0.14991377
##        337        338        339        340        341        342        343
## 0.15067873 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869 0.15570561
##        344        345        346        347        348        349        350
## 0.15652521 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785 0.16152477
##        351        352        353        354        355        356        357
## 0.16234437 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434 0.16737126
##        358        359        360        361        362        363        364
## 0.16819086 0.06923779 0.07008471 0.07084967 0.07169659 0.07251619 0.07336311
##        365        366        367        368        369        370        371
## 0.07418271 0.07502963 0.07587655 0.07669615 0.07754307 0.07836267 0.07920959
##        372        373        374        375        376        377        378
## 0.08005651 0.08084879 0.08169571 0.08251531 0.08336223 0.08418183 0.08502875
##        379        380        381        382        383        384        385
## 0.08587567 0.08669527 0.08754219 0.08836179 0.08920872 0.09005564 0.09082060
##        386        387        388        389        390        391        392
## 0.09166752 0.09248712 0.09333404 0.09415364 0.09500056 0.09584748 0.09666708
##        393        394        395        396        397        398        399
## 0.09751400 0.09833360 0.09918052 0.10002744 0.10079240 0.10163932 0.10245892
##        400        401        402        403        404        405        406
## 0.10330584 0.10412544 0.10497236 0.10581928 0.10663888 0.10748580 0.10830540
##        407        408        409        410        411        412        413
## 0.10915232 0.10999924 0.11076420 0.11161112 0.11243072 0.11327764 0.11409724
##        414        415        416        417        418        419        420
## 0.11494416 0.11579108 0.11661068 0.11745760 0.11827720 0.11912412 0.11997104
##        421        422        423        424        425        426        427
## 0.12076332 0.12161024 0.12242984 0.12327676 0.12409636 0.12494328 0.12579020
##        428        429        430        431        432        433        434
```

```
## 0.12660980 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205
##        435        436        437        438        439        440        441
## 0.13240165 0.13324857 0.13406817 0.13491509 0.13576201 0.13658161 0.13742853
##        442        443        444        445        446        447        448
## 0.13824813 0.13909505 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037
##        449        450        451        452        453        454        455
## 0.14403997 0.14488689 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685
##        456        457        458        459        460        461        462
## 0.14991377 0.15067873 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869
##        463        464        465        466        467        468        469
## 0.15570561 0.15652521 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785
##        470        471        472        473        474        475        476
## 0.16152477 0.16234437 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434
##        477        478        479        480        481        482        483
## 0.16737126 0.16819086 0.06923779 0.07008471 0.07084967 0.07169659 0.07251619
##        484        485        486        487        488        489        490
## 0.07336311 0.07418271 0.07502963 0.07587655 0.07669615 0.07754307 0.07836267
##        491        492        493        494        495        496        497
## 0.07920959 0.08005651 0.08084879 0.08169571 0.08251531 0.08336223 0.08418183
##        498        499        500        501        502        503        504
## 0.08502875 0.08587567 0.08669527 0.08754219 0.08836179 0.08920872 0.09005564
##        505        506        507        508        509        510        511
## 0.09082060 0.09166752 0.09248712 0.09333404 0.09415364 0.09500056 0.09584748
##        512        513        514        515        516        517        518
## 0.09666708 0.09751400 0.09833360 0.09918052 0.10002744 0.10079240 0.10163932
##        519        520        521        522        523        524        525
## 0.10245892 0.10330584 0.10412544 0.10497236 0.10581928 0.10663888 0.10748580
##        526        527        528        529        530        531        532
## 0.10830540 0.10915232 0.10999924 0.11076420 0.11161112 0.11243072 0.11327764
##        533        534        535        536        537        538        539
## 0.11409724 0.11494416 0.11579108 0.11661068 0.11745760 0.11827720 0.11912412
##        540        541        542        543        544        545        546
## 0.12076332 0.12161024 0.12242984 0.12327676 0.12409636 0.12494328 0.12579020
##        547        548        549        550        551        552        553
## 0.12660980 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205
##        554        555        556        557        558        559        560
## 0.13240165 0.13324857 0.13406817 0.13491509 0.13576201 0.13658161 0.13742853
##        561        562        563        564        565        566        567
## 0.13824813 0.13909505 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037
##        568        569        570        571        572        573        574
## 0.14403997 0.14488689 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685
##        575        576        577        578        579        580        581
## 0.14991377 0.15067873 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869
##        582        583        584        585        586        587        588
## 0.15570561 0.15652521 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785
##        589        590        591        592        593        594        595
## 0.16152477 0.16234437 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434
##        596        597        598        599        600        601        602
## 0.16737126 0.16819086 0.11661068 0.11745760 0.11827720 0.11912412 0.11997104
##        603        604        605        606        607        608        609
## 0.12076332 0.12161024 0.12242984 0.12409636 0.12494328 0.12579020 0.12660980
##        610        611        612        613        614        615        616
## 0.12745673 0.12827633 0.12912325 0.12997017 0.13073513 0.13158205 0.13240165
##        617        618        619        620        621        622        623
```

```
## 0.13324857 0.13406817 0.13491509 0.13576201 0.13658161 0.13742853 0.13824813
##        624        625        626        627        628        629        630
## 0.13909505 0.13994197 0.14070693 0.14155385 0.14237345 0.14322037 0.14403997
##        631        632        633        634        635        636        637
## 0.14488689 0.14573381 0.14655341 0.14740033 0.14821993 0.14906685 0.15067873
##        638        639        640        641        642        643        644
## 0.15152565 0.15234525 0.15319217 0.15401177 0.15485869 0.15570561 0.15652521
##        645        646        647        648        649        650        651
## 0.15737213 0.15819173 0.15903865 0.15988557 0.16067785 0.16152477 0.16234437
##        652        653        654        655        656        657        658
## 0.16319129 0.16401089 0.16485781 0.16570474 0.16652434 0.16737126 0.16819086
```

```r
print(2.732e-05 * (3650))
```

```
## [1] 0.099718
```

0.09895307 meters over the 10-year period. The first method I did used the predict function to predict the values based on the model. I then subtracted the first entry from the last entry. The second method I used relied in the equation y = mx + b. The coefficient times the 3650 (the number of days in 10 years) should give the average increase over 10 years. These two methods produced very similar answers (0.09895307 for method one and 0.099718 for method 2).

Upload your .Rmd file and you knitted file with the answers and plots to Canvas

##STRETCH GOAL

If you are looking for a challenge, have a go downloading the original csv files directly from the NOAA API. Details on the API are here: https://api.tidesandcurrents.noaa.gov/api/prod/

You will want to paste together a URL describing the data you want from the API, then use download.file() to download the data from that URL into a directory on your computer.

The URL you want will have the following form, except you will loop through to replace *GAUGEID* with each of the six tide gauge ID numbers:

paste0("https://api.tidesandcurrents.noaa.gov/api/prod/datagetter?begin_date=20110101&end_date=20201231&station=",*GAUGEID*,"&product=monthly_mean&datum=MHHW&units=metric&time_zone=lst&format=csv"

See if you can make sense of this URL given the options listed at the website describing access to the API

begin_date=20110101&end_date=20200101