Project 3 - Analyzing Seasonal Temperature Changes in the Pacific Ocean

Sunny

2023-02-28

Introduction

In this R Project, I will be working with surface level sea temperature data of the Pacific Ocean collected from 12 various sites in British Columbia.

Objectives The *main objective* of this project is to be able to identify the seasonal trends in the Pacific Ocean's surface temperature, and determine if those seasonal trends have changed in any way throughout the last century.

The *secondary objective* of this project is to be able to learn more about R by applying my existing knowledge of R towards this project.

Setting up the R environment

Before we begin with this project, it's important for us to first start by loading in all the tools (or packages) that we'll be needing in order to complete this project within the R environment. The main package we'll use is the *tidyverse* package.

Since I've already installed this package before, let's just load *tidyverse* into our RStudio environment:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Setting up our Working Directory Next let's setup our *working directory*, which is the file location on our computer which R will utilize to import and export our data to and from our device. This step is arbitrary as the location of where you saved your data for your device may be different than where it can be found on mine.

```
setwd("~/Documents/Learning Data Analytics/Data Projects/Project 3 - Analyzing Seasonal Pacific Ocean T
```

Importing our dataset into the R environment Now let's get our Pacific Ocean temperature data set loaded in! Now keep in mind, there's a lot of data we're going to be working with here.

We'll be working with 12 different CSV files, each containing monthly temperature data collected for numerous years (sometimes up to a century worth of data). Each CSV file represents a different coastline location for which those temperature measurements were collected. Let's import these 12 CSV files:

```
Amphitrite_temps <- read.csv("Amphitrite.csv")
Bonilla_temps <- read.csv("Bonilla Island.csv")
Chrome_temps <- read.csv("Chrome Island.csv")
Departure_bay_temps <- read.csv("Departure Bay.csv")
Egg_Island_temps <- read.csv("Egg Island.csv")
Entrance_temps <- read.csv("Entrance Island.csv")
Kains_temps <- read.csv("Kains Island.csv")
Langara_temps <- read.csv("Langara Island.csv")
McInnes_temps <- read.csv("McInnes Island.csv")
Nootka_temps <- read.csv("Nootka.csv")
Pine_temps <- read.csv("Pine Island.csv")
Race_temps <- read.csv("Race Rocks.csv")
```

Now that we've imported these 12 different CSV files, let's take a quick look at one of these tables:

```
head(Amphitrite_temps, 5)
```

```
##   AMPHITRITE.POINT.LIGHTSTATION..AVERAGE.MONTHLY.SEA.SURFACE.TEMPERATURES..C.
## 1                                                                          YEAR
## 2                                                                          1934
## 3                                                                          1935
## 4                                                                          1936
## 5                                                                          1937
##        X    X.1    X.2    X.3    X.4    X.5    X.6   X.7   X.8   X.9  X.10 X.11
## 1    JAN    FEB    MAR    APR    MAY    JUN    JUL   AUG   SEP   OCT   NOV  DEC
## 2 999.99 999.99 999.99 999.99 999.99 999.99 999.99 13.98 12.34 10.45  10.1 8.97
## 3   7.33   7.63    7.4   8.86   9.95  11.62  12.74 12.53 12.77 11.26  8.14 8.38
## 4   8.02   6.64    7.6   8.56  10.45  13.72  14.29 14.56 12.93 11.76  9.09 8.41
## 5   6.91   6.36   8.03   9.15  10.29  11.72  12.15 12.92 11.73 11.02 10.41 9.04
```

Cleaning our Data

Now that we've successfully imported our temperature data for the Pacific Ocean, let's get cleaning! But what exactly do we need to do for our data cleaning step? Here's a few flaws about our data we notice immediately:

- The column headers are mislabelled
- The data type is wrongly categorized as *character* rather than *numeric*
- There are values of 999.99 which seems to be used to represent missing data
- The format of our data is temperature per month rather than per season

1. Cleaning up column names Just from previewing our tables, we notice that the *column headers* are mislabeled due to our CSV files having a *TITLE* row. In other words, the column headers we want is actually being represented by Row #1, where it displays **YEAR**, **JAN**, **FEB**, **MAR**, and etc. To correct this, we'll need to rename each of our column headers and remove our first row in each table. We do this in the code chunk below:

```
Amphitrite_temps_clean <- Amphitrite_temps %>%
  rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, OC

Bonilla_temps_clean <- Bonilla_temps %>%
  rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, OC

Chrome_temps_clean <- Chrome_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Departure_bay_temps_clean <- Departure_bay_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Egg_Island_temps_clean <- Egg_Island_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Entrance_temps_clean <- Entrance_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Kains_temps_clean <- Kains_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Langara_temps_clean <- Langara_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

McInnes_temps_clean <- McInnes_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Nootka_temps_clean <- Nootka_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Pine_temps_clean <- Pine_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O

Race_temps_clean <- Race_temps %>%
   rename(YEAR = 1, JAN = 2, FEB = 3, MAR = 4, APR = 5, MAY = 6, JUN = 7, JUL = 8, AUG = 9, SEP = 10, O
```

We've also assigned a new variable for each of the 12 CSV files where the column headings were corrected. Again, what we did was we renamed each of the column headers and then we removed the first row.

The new tables with corrected columns have been assigned to new variables which contain the word *clean* in the name. For example *Amphitrite_temps_clean* instead of *Amphitrite_temps*. We can now remove all the older tables with the mislabeled headers as we will not be using them any further. To do that, we run the following code-chunk:

```
rm(Amphitrite_temps, Bonilla_temps, Chrome_temps, Departure_bay_temps, Egg_Island_temps, Entrance_temps
```

Now that we have corrected our column headers, let's address our next data set deficiency: datatypes.

2. From character to numeric datatypes We notice that currently, for each of our CSV files containing temperature data, our datatype is recorded as *character* rather than *numeric*. Having this datatype will mean that we won't be able to perform calculations on the temperature data later. To correct this, we use the code-chunk below:

```
Amphitrite1 <- mutate(Amphitrite_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_i
Bonilla1 <- mutate(Bonilla_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 9
Chrome1 <- mutate(Chrome_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 999
DepartureBay1 <- mutate(Departure_bay_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric,
EggIsland1 <- mutate(Egg_Island_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if
Entrance1 <- mutate(Entrance_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(.,
Kains1 <- mutate(Kains_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 999.9
Langara1 <- mutate(Langara_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 9
McInnes1 <- mutate(McInnes_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 9
Nootka1 <- mutate(Nootka_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 999
Pine1 <- mutate(Pine_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 999.99))
Race1 <- mutate(Race_temps_clean, across(2:13, as.numeric)) %>% mutate_if(is.numeric, ~na_if(., 999.99))
```

We actually tackled two brids with one code above! We not only changed the datatype from *character* to *numeric*, but we've also replaced all of our temperature values of *999.99* to *NA*. We will discuss the rationale of this latter portion shortly in data cleaning **step 3**.

Again, to keep our work space clean from clutter, we'll now remove the old variables by using the following code chunk:

```
rm(list = ls(pattern = "temps_clean"))
```

This code chunk is a lot cleaner than our last *rm()* function as we've identified a way to tell R to remove all variables that share similar words; in this case "temps_clean".

Moving on, now that we've addressed our column headers and datatype problems, we're ready to address our third problem which is missing data encoded with values of 999.99.

3. Addressing missing data entries Now let's take a quick moment to address why we changed our temperature data; where values equal to *999.99* became *NA*. The reason we did this is because the 999.99 values were likely *arbitrarily* filled-in for the months where data was not recorded. Keeping these numbers in our data set would mean that we'd inflate our *average seasonal temperature* calculations later. Replacing 999.99 values with *NA* enables us to calculate our averages while omitting these cells for a more accurate *average seasonal temperature* calculation.

Now that we've replaced our *999.99* values with *NA*, our next step is to look at how we can reformat the data in a way that is more meaningful to us. What we want is to see if seasonal changes have impacted the Pacific Ocean's surface level temperatures, and if those average seasonal temperatures have changed over the course of the last century.

4. Reformatting our data to align with our objectives Currently the format of our data tables show monthly temperature measurements collected across many years. What we want is seasonal measurements rather than monthly over the course of these years. To accomplish this, we will first need to group our various *month columns* into the 4 different seasons.

Before we begin, we must first define which months belong to which seasons. According to timeanddate.com, the 4 seasons for Earth's Northern Hemisphere can be defined by the following months:

- **Spring** - March to May

- **Summer** - June to August
- **Fall** - September to November
- **Winter** - December to February

Now that we know which months correspond to which seasons, we'll want to create new data tables where we have *average seasonal temperatures per year* rather than *monthly temperatures per year*. We'll use the following code to accomplish this:

```
Amphitrite2 <- rowwise(Amphitrite1) %>%
  mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRUE)

Bonilla2 <- rowwise(Bonilla1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Chrome2 <- rowwise(Chrome1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

DepartureBay2 <- rowwise(DepartureBay1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

EggIsland2 <- rowwise(EggIsland1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Entrance2 <- rowwise(Entrance1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Kains2 <- rowwise(Kains1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Langara2 <- rowwise(Langara1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

McInnes2 <- rowwise(McInnes1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Nootka2 <- rowwise(Nootka1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Pine2 <- rowwise(Pine1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU

Race2 <- rowwise(Race1) %>%
    mutate(Spring = mean(c_across(MAR:MAY), na.rm = TRUE), Summer = mean(c_across(JUN:AUG), na.rm = TRU
```

Again, like previous steps, we're clear to remove our previous tables again using the following code:

```
rm(list = ls(pattern = 1))
```

Our next objective is to create a new standalone table where we have average temperature per season with all of our locations combined. To do this, we'll first need to aggregate all of our 2-series tables (ie. Chrome2 or Entrance2) into a single new table. We will run the following code to accomplish this:

```
Pacific_Ocean_Temperatures <- left_join(Amphitrite2, Bonilla2, by = "YEAR", keep = NULL) %>%
  left_join(., Chrome2, by = "YEAR", keep = NULL) %>%
  left_join(., DepartureBay2, by = "YEAR", keep = NULL) %>%
  left_join(., EggIsland2, by = "YEAR", keep = NULL) %>%
  left_join(., Entrance2, by = "YEAR", keep = NULL) %>%
  left_join(., Kains2, by = "YEAR", keep = NULL) %>%
  left_join(., Langara2, by = "YEAR", keep = NULL) %>%
  left_join(., McInnes2, by = "YEAR", keep = NULL) %>%
  left_join(., Nootka2, by = "YEAR", keep = NULL) %>%
  left_join(., Pine2, by = "YEAR", keep = NULL) %>%
  left_join(., Race2, by = "YEAR", keep = NULL) %>%
  rename(Amphitrite_Spring = 2, Amphitrite_Summer = 3, Amphitrite_Fall = 4, Amphitrite_Winter = 5, Boni
```

With the help of the *LEFT_JOIN* function, we've now aggregated all of our data into a single table. Next, we're going to perform another average calculation. This will help us reach our end-goal, which is to identify the average seasonal temperature of the Pacific Ocean (harboring British Columbia) where all our locations are accounted for. We'll be using the following code-chunk to accomplish this:

```
Average_Seasonal_Temps <- rowwise(Pacific_Ocean_Temperatures) %>%
  mutate(Spring = mean(c_across(c(2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46)), na.rm = TRUE), Summer
```

Now before we finish off, we just want to reformat our table to make it more *"legible"* as we move into our data visualization step. We want to convert our seasons column headers (ie. Spring, Summer, Fall, and Winter) into a single column with multiple rows; in otherwords, we're going from *wide-format* data to *long-format* data. Formatting it this way will be handy when we get into creating visualizations with *ggplot*; this will become apparent shortly. We'll use the pivot function in *tidyverse* to accomplish this:

```
Avg_Temps <- pivot_longer(Average_Seasonal_Temps, 2:5, names_to = "Seasons") %>% rename(Temperature = 3
```

This marks the end of our data cleaning! We now have our data cleaned and formatted in a way that will enable us to answer our main question in *Objective 1*: How have seasonal temperatures changed in the Pacific Ocean (near British Columbia) over the years?

Before we move onto the analysis and visualization phase, let's quickly remove some of the unnecessary data tables in our R-environment. Again, we'll use the following code:

```
rm(list = ls(pattern = "2"))
rm(Average_Seasonal_Temps)
```

Doing our Analysis through Visualization

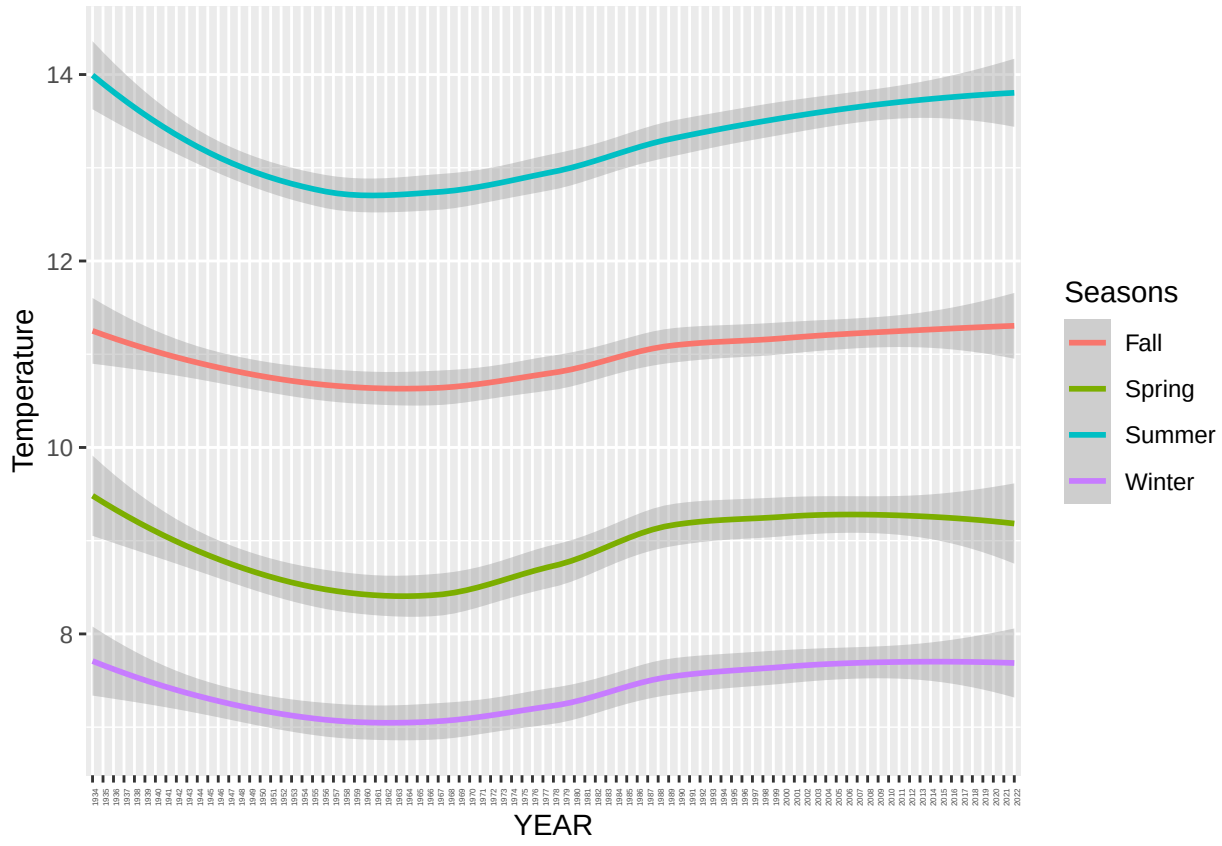
We're now going focus on our *Avg_Temps* dataset as we want to identify whether seasonal ocean temperatures have changed drastically over the past century near British Columbia. At a glance, purely looking at the data set alone, it's difficult to determine whether there were any changes across the years; there are simply too many numbers! Using visualizations may help make this distinction easier.

Let's do just that:

```
VisualTable <- Avg_Temps %>% group_by(Seasons) %>%
  arrange(YEAR)

ggplot(VisualTable) + geom_smooth(mapping = aes(x = YEAR, y = Temperature, group = Seasons, color = Seas
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Observations

Based on the visualization, we notice that as our seasons have changed, so have the average temperatures. Our visualization shows that in the Winter our average temperature hovers below 8 degrees Celsius, around 9 degrees for the Spring, 11 degrees around Fall, and between 13 to 14 degrees in the Summer. We also notice a drop in average temperatures across all seasons between *1942* and *1962*; although that drop in temperature does recover gradually after *1965* until around *1990*. Beyond that point, we see somewhat of a plateau in the average temperatures.

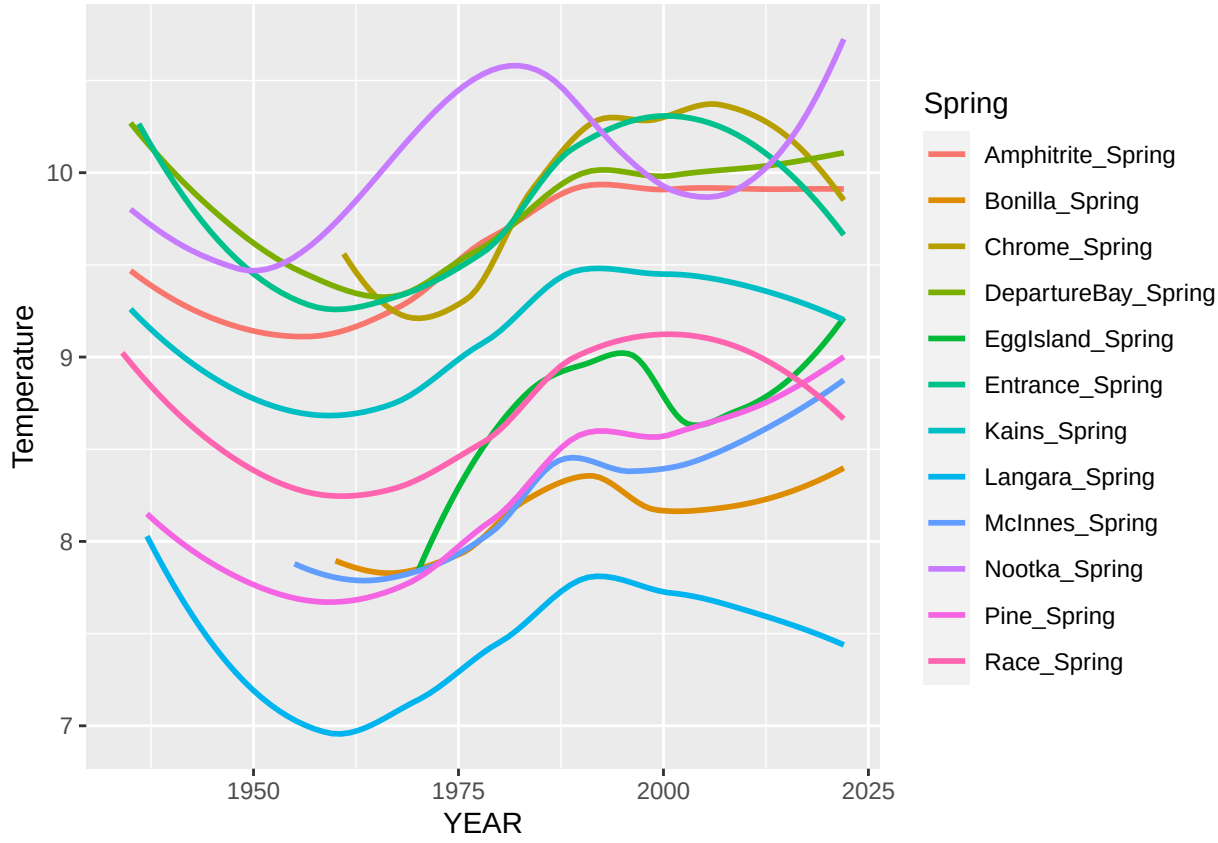
Overall, seeing a correlation between the measured and calculated average temperatures with seasonality is a good thing. It tells us that our data is somewhat relevant and that the ocean temperature does change along with the seasonal changes.

Exploring the Dip in Temperatures Naturally we may wonder what would have caused average temperatures across all seasons to drop between the years *1942* and *1962*? One potential reason for this could be due to the lack of data collected. If we look at our *Pacific_Ocean_Temperatures* table, we notice that there is missing data from several sites during the earlier decades. There are at least 4 different sites where data was not collected for the first couple decades; these include **Bonilla**, **Chrome**, **EggIsland**, and **McInnes**. *We may want to see if these sites, where measurements from earlier decades were missing, have higher temperatures compared to sites where data exists for those earlier decades.* If they do, it might explain why we saw a dip in average temperatures in the earlier decades. Let's take a look at one of the seasons with a more pronounced drop in average temperatures during those decades; Spring.

```
Spring_Temps <- Pacific_Ocean_Temperatures %>% select(YEAR, ends_with("Spring")) %>%
  pivot_longer(2:13, names_to = "Spring") %>% rename(Temperature = 3) %>%
  mutate(across(1, as.numeric)) %>%
  group_by(Spring)

ggplot(Spring_Temps) + geom_smooth(mapping = aes(x = YEAR, y = Temperature, color = Spring), na.rm = TRU
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

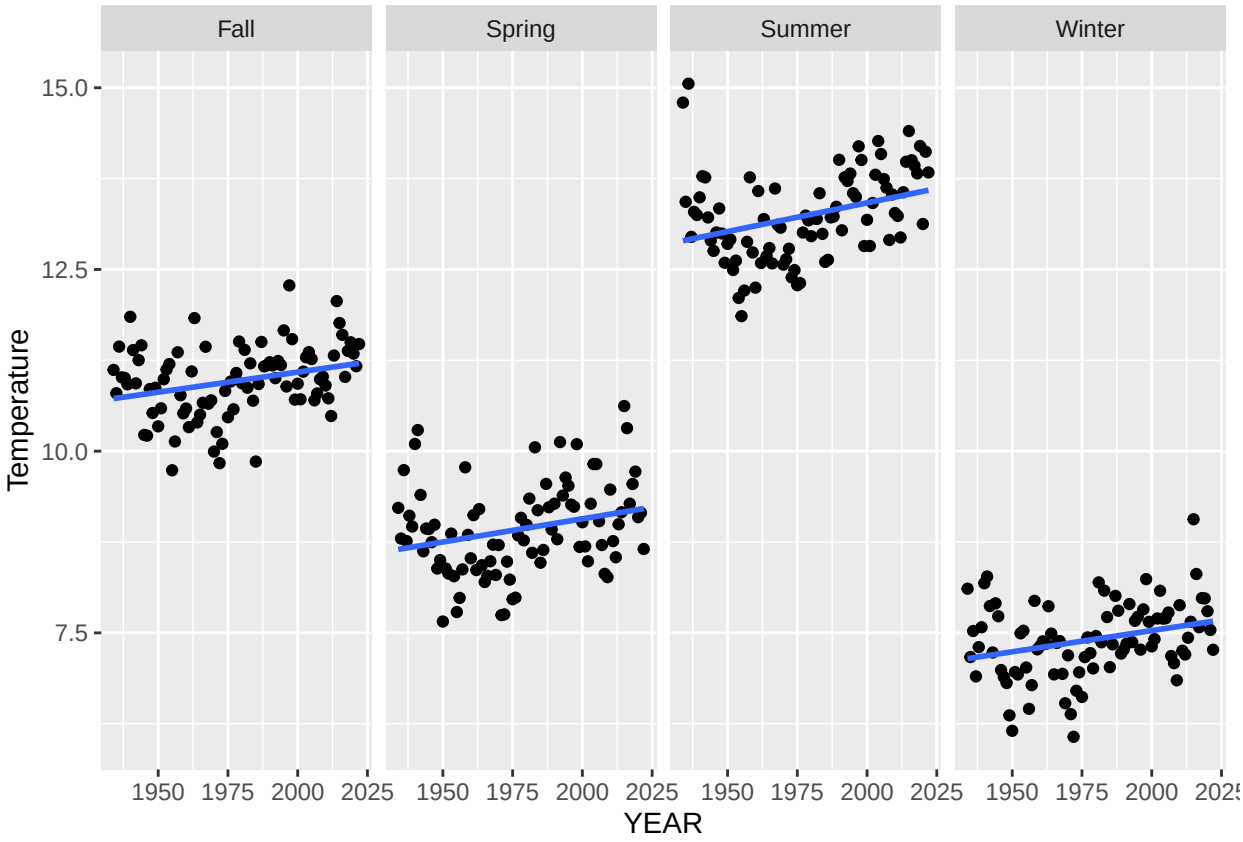


Based on the visualization produced above it seems unlikely that sites such as **Bonilla**, **Chrome**, **EggIsland**, and **McInnes**, where data from earlier decades were missing, would have had any significant impact on lowering the Pacific Ocean's average temperatures during those earlier decades. Furthermore, our visualization of the temperature change by location seems to show that each and everyone of these locations experienced a dip in temperatures during those early decades. This suggests that other factors may have been at play. We may require additional data to explore this phenomenon.

Did Average Seasonal Temperatures Rise of the Past Century? Alright, let's get back on track and answer our primary objective of this project. Did the average temperatures of the Pacific Ocean (harbouring British Columbia) change over the past century. One way to see if average temperatures have risen over the past century would be to create a linear regression using the data we have. We could recreate the first visualization we made, using the *VisualTable* data, but using a linear-regression function instead. Let's do that:

```
VisualTable2 <- VisualTable %>% mutate(across(1, as.numeric))

ggplot(VisualTable2, aes(x = YEAR, y = Temperature)) + geom_point() + geom_smooth(method = "lm", formula
```



Based on the visualization we've been able to produce, it seems that the Pacific Ocean's surface temperatures have increased over the past century! This is an observation consistent among all four seasons. Although greater exploration into this observation is necessary to determine potential causes, from a glance it seems that climate change (global warming) certainly is having its affect on the surface temperatures of the Pacific Ocean harboring British Columbia.