Homework 5 (FISH 553)

Question 1

A)

I'll read in each dataset using base R functions, as they are the most straightforward, and use the col.names call within the read.table() function to rename each column. Using the tidyverse rename() function within a single line of code would result in a longer line of code than the result of base R functions.

```
colNames <- c("Year","spawners","recruits", "catch", "fishMortality")
mack.ices <- read.table("MACKICES.txt", col.names = colNames)
mack.black <- read.table("MACKBLACK.txt", col.names = colNames)
mack.nafo <- read.table("MACKNAFO.txt", col.names = colNames)
```

B)

I will first download the tidyverse package. Then I will use the inner_join() function, which will only keep "Year" observations found in both the mack.nafo and mack.black datasets.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
## Warning: package 'tibble' was built under R version 4.0.2
```

```
## Warning: package 'tidyr' was built under R version 4.0.2
```

```
## Warning: package 'readr' was built under R version 4.0.2
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
mack.partial <- inner_join(mack.nafo, mack.black, by="Year", suffix = c(".nafo", ".black"))
```

C)

I repeat the steps in part b.

```
mack <- inner_join(mack.ices, mack.partial, by="Year")
```

D)

I didn't include the suffix argument in the inner_join() function because I didn't want to rewrite the column names of mack.partial. Instead, I use tidyverse's rename() function to put the ".ices" suffix on all the mack.ices columns.

```
mack <- mack %>% rename(spawners.ices=spawners) %>% rename(recruits.ices=recruits) %>% rename(catch.ices
```

E)

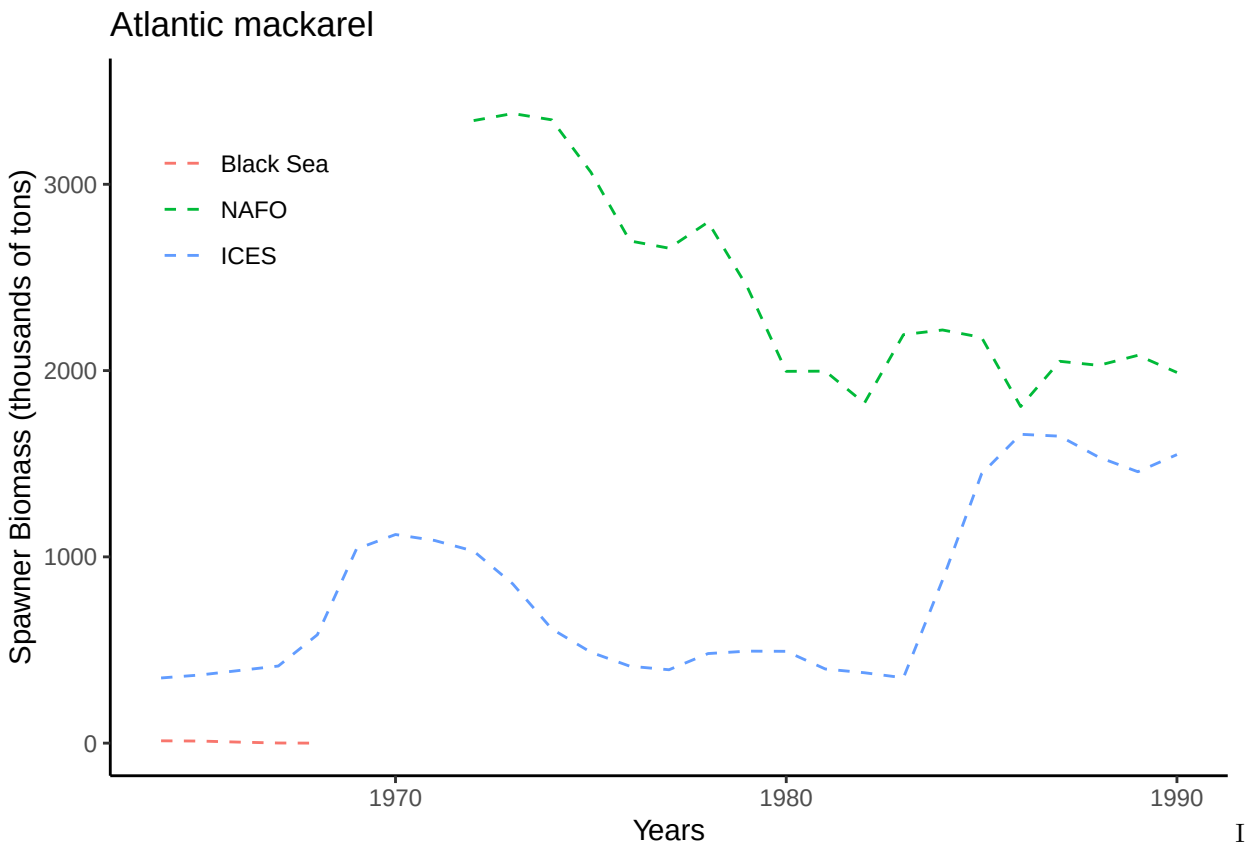
I used ggplot to recreate this plot because it has a cleaner output. I had to do some extra subsetting to plot multiple groups of data (like matplot would achieve). While this is good ggplot practice, I would prefer matplot for layered data.

```
subset<- mack %>% select(Year, starts_with("spawners"))

subset2 <- subset %>% gather(key= Spawners, value=Number, 2:4) %>% slice(9:59)

subset2$Number <- as.numeric(subset2$Number)
subset2 <- subset2 %>% arrange(Number)

ggplot(data=subset2) +
  geom_line(aes(Year, Number, group=Spawners, color=Spawners), lty=2) +
  labs(x="Years", y= "Spawner Biomass (thousands of tons)", title = "Atlantic mackarel") +
  coord_cartesian(xlim =c(1964, 1990), ylim=c(0,3500)) + theme_classic() +
  theme(legend.position = c(0.12, 0.8), legend.title = element_blank()) +
  scale_color_discrete(labels=c("Black Sea","NAFO", "ICES"))
```



I was not sure how to the line change colors and axis limits.

Question 2

A)

I used the following code chunk (from homework 3) to create a data frame named temperature which has 2 columns: the dates Jan 1 2010 through Jun 30 2010 and a randomly generated temperature for each day. I thought using the tidyverse and Hmisc packages would make this easier because it is easier to add a column in tidyverse, and easier to identify the number of days in a month using Hmisc.

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.0.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 4.0.2
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
temperature<- data.frame(date=seq(from=as.Date("2010/1/1"), to=as.Date("2010/6/30"), by=1))

month_lengths<- c(monthDays("2010-01-01"), monthDays("2010-02-01"),monthDays("2010-03-01"),monthDays("2
means<- c(40 ,42 ,51 ,55 ,58 ,62)

temp <- rep(NA, length.out=181)
for(i in 1:length(means)){
   a <- rnorm(month_lengths[i], mean = means[i], sd = 5)
   if(i==1){temp[1:31]<- a}
   if(i==2){temp[32:59]<- a}
   if(i==3){temp[60:90]<- a}
   if(i==4){temp[91:120]<- a}
   if(i==5){temp[121:151]<- a}
   if(i==6){temp[152:181]<- a}
}

temperature <- temperature %>% mutate(Temp = temp)
temperature$Temp <- round(temperature$Temp)
```

B)

I used the group_by and summarise_ functions of tidyverse. I think it's as easy as using tapply, I just feel more comfortable with the tidyverse functions.

```
temperature %>% mutate(month = format(date, "%m")) %>% group_by(month) %>% summarise(meanTemp = mean(Te
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 2
##   month meanTemp
##   <chr>    <dbl>
## 1 01        40.6
## 2 02        41.9
## 3 03        51.0
## 4 04        55.0
## 5 05        58.5
## 6 06        62.0
```

C)

To figure out the days in which duplicate temperatures exist, I use base R functions. To my knowledge, tidyverse does not have an equivalent function besides distinct() which is to remove duplicate rows.

```
temperature$date[duplicated(temperature$Temp)==TRUE]
```

```
##   [1] "2010-01-04" "2010-01-07" "2010-01-08" "2010-01-13" "2010-01-14"
##   [6] "2010-01-16" "2010-01-17" "2010-01-20" "2010-01-21" "2010-01-23"
##  [11] "2010-01-25" "2010-01-26" "2010-01-27" "2010-01-28" "2010-01-29"
##  [16] "2010-01-30" "2010-02-01" "2010-02-02" "2010-02-03" "2010-02-05"
##  [21] "2010-02-06" "2010-02-07" "2010-02-08" "2010-02-09" "2010-02-10"
##  [26] "2010-02-12" "2010-02-13" "2010-02-14" "2010-02-15" "2010-02-16"
##  [31] "2010-02-17" "2010-02-18" "2010-02-19" "2010-02-21" "2010-02-22"
##  [36] "2010-02-23" "2010-02-24" "2010-02-25" "2010-02-26" "2010-02-27"
##  [41] "2010-02-28" "2010-03-03" "2010-03-06" "2010-03-07" "2010-03-08"
##  [46] "2010-03-10" "2010-03-11" "2010-03-12" "2010-03-14" "2010-03-15"
##  [51] "2010-03-16" "2010-03-17" "2010-03-19" "2010-03-20" "2010-03-21"
##  [56] "2010-03-22" "2010-03-23" "2010-03-24" "2010-03-25" "2010-03-26"
##  [61] "2010-03-27" "2010-03-28" "2010-03-29" "2010-03-31" "2010-04-02"
##  [66] "2010-04-03" "2010-04-04" "2010-04-05" "2010-04-06" "2010-04-07"
##  [71] "2010-04-08" "2010-04-11" "2010-04-13" "2010-04-15" "2010-04-16"
##  [76] "2010-04-17" "2010-04-18" "2010-04-19" "2010-04-21" "2010-04-22"
##  [81] "2010-04-23" "2010-04-24" "2010-04-25" "2010-04-27" "2010-04-29"
##  [86] "2010-04-30" "2010-05-01" "2010-05-02" "2010-05-03" "2010-05-04"
##  [91] "2010-05-05" "2010-05-06" "2010-05-07" "2010-05-08" "2010-05-09"
##  [96] "2010-05-10" "2010-05-11" "2010-05-12" "2010-05-13" "2010-05-14"
## [101] "2010-05-16" "2010-05-17" "2010-05-18" "2010-05-19" "2010-05-20"
## [106] "2010-05-21" "2010-05-22" "2010-05-23" "2010-05-24" "2010-05-25"
## [111] "2010-05-26" "2010-05-27" "2010-05-28" "2010-05-29" "2010-05-30"
## [116] "2010-06-01" "2010-06-02" "2010-06-03" "2010-06-04" "2010-06-05"
## [121] "2010-06-06" "2010-06-07" "2010-06-08" "2010-06-09" "2010-06-10"
## [126] "2010-06-11" "2010-06-12" "2010-06-13" "2010-06-16" "2010-06-18"
## [131] "2010-06-19" "2010-06-20" "2010-06-21" "2010-06-22" "2010-06-23"
## [136] "2010-06-24" "2010-06-25" "2010-06-26" "2010-06-27" "2010-06-28"
## [141] "2010-06-29" "2010-06-30"
```

D)

I also decided to use base R for this question. While one could create separate data frames for the conditions and wind speed, then use one of the join() functions to unite them, this way is more parsimonious. Furthermore, tidyverse could be used to isolate the values with negative signs and change them to zero, but base R is more parsimonious.

```
observations<- data.frame(date=seq(from=as.Date("2010/1/1"), to=as.Date("2010/7/31"),by=2),
                          conditions=sample(x=c("sunny", "cloudy", "partly cloudy"),
```

```
                                                size=length(seq(from=as.Date("2010/1/1"), to=as.Date("2010/1
                                "wind speed"= rnorm(n=length(seq(from=as.Date("2010/1/1"), to=as.Date("2010/7/

observations[observations$wind.speed < 0,3] <- 0
```

E)

I used the join() functions to unite the two data frames. These allow specification in which rows and columns to keep. In this case, inner_join allowed me to only keep the date observations that matched.

```
weather <- inner_join(temperature, observations, by="date")
```

F)

I used a pipeline with group_by and summarise to get summary statistics of the weather data, as opposed to using the apply family. This code is more parsimonious than that of Homework 3.

```
weather %>% group_by(conditions) %>% summarise(minim = min(Temp), maxim = max(Temp))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 3
##   conditions    minim maxim
##   <chr>         <dbl> <dbl>
## 1 cloudy           36    70
## 2 partly cloudy    33    69
## 3 sunny            31    67
```