Sea levels

Matthew Hurworth

2022-04-03

###To run the code save the excel files emailed to you into a folder without changing the names. Then set the working directory to the file location they are saved in. Code should run entirely from that.

```
knitr::opts_chunk$set(echo = TRUE, results="hide", fig.show="hide", error=FALSE, warning=FALSE, message

swd<-("D:/Data analytics year/Sea levels")

ND<-read.csv("Drake_Passage_North_Deep.csv")

SD<-read.csv("Drake_Passage_South_Deep.csv")

S<-read.csv("Drake_Passage_South.csv")

N<-read.csv("Drake_Passage_North.csv")

CO2<-read.csv("Imputed_world_emissions.csv")

Globtemp<-read.csv("NASA_Global_Temp.csv")

ArcticIce<-read.csv("NASA_Arctic_Ice.csv")

Sealevel<-read.csv("Sea_level_data.csv")
```

##Call libraries

```
library(dplyr)
library(ggplot2)
library(corrplot)
library(forecast)
```

renaming variables for CO2

```
#Renaming variables in the CO2 data - from the emissions excel file
CO2i<-rename(CO2, CO2C=CO2.concentrations, SST=Sea.surface.temp, SSTLB=Sea.surface.temp..lower.bound., S
```

```
NDQ<-filter(ND, Quality == 0 )

summary(NDQ)

ggplot(data = NDQ, mapping = aes(x=Year, y = Res.Dr)) + geom_boxplot()
ggplot(data = NDQ, mapping = aes(x=Year, y = Res.Dr.1)) + geom_boxplot()+ ggtitle("Res DR 1 for North d
```

```
ggplot(data = NDQ, mapping = aes(x=Day, y = Res.Dr, colour=Year)) + geom_point()+ geom_smooth(method="l
```

```
ggplot(data = NDQ, mapping = aes(x=Day, y = Res.Dr.1, colour=Year)) + geom_point()+ geom_smooth(method=
```

```
SDQ<-filter(SD, Quality == 0 )

ggplot(data = SDQ, mapping = aes(x=Year, y = Res.Dr)) + geom_boxplot()
```

```
ggplot(data = SDQ, mapping = aes(x=Year, y = Res.Dr.1)) + geom_boxplot()+ ggtitle("Res DR 1 for South d
```

```
ggplot(data = SDQ, mapping = aes(x=Day, y = Res.Dr, colour=Year)) + geom_point()+ geom_smooth(method="l
```

```
ggplot(data = SDQ, mapping = aes(x=Day, y = Res.Dr.1, colour=Year)) + geom_point()+ geom_smooth(method=
```

```
SQ<-filter(S, Quality == 0 )

ggplot(data = SQ, mapping = aes(x=Year, y = Res.Dr)) + geom_boxplot()
```

```
ggplot(data = SQ, mapping = aes(x=Year, y = Res.Dr.1)) + geom_boxplot()+ ggtitle("Res DR 1 for South dr
```

```
ggplot(data = SQ, mapping = aes(x=Day, y = Res.Dr, colour=Year)) + geom_point() + geom_smooth(method="l
```

```
ggplot(data = SQ, mapping = aes(x=Day, y = Res.Dr.1, colour=Year)) + geom_point() + geom_smooth(method=
```

```
library(ggplot2)

ggplot(data = SD, mapping = aes(x=Year, y = Res.Dr.1, colour=Year))+ ylim(-10,10) + geom_point() + ggti
```

```
ggplot(data = ND, mapping = aes(x=Quality, y = Res.Dr.1, colour=Year))+ ylim(-10,10) + geom_point()+ ge
```

```
NQ<-filter(N, Quality == 0 )

summary(NQ)

NQ1992<-filter(NQ, Year==1992)

NQ2000<-filter(NQ, Year==2000)

NQ2008<-filter(NQ, Year==2008)

ggplot(data = NQ, mapping = aes(x=Year, y = Res.Dr)) + geom_boxplot()
```

```
ggplot(data = NQ, mapping = aes(x=Year, y = Res.Dr.1)) + geom_boxplot()+ ggtitle("Res DR 1 for North dr
ggplot(data = NQ1992, mapping = aes(x=Day, y = Res.Dr.1, colour=Year))+ ylim(-10,10) + geom_point()+ ge
```

```
ggplot(data = NQ2000, mapping = aes(x=Day, y = Res.Dr.1, colour=Year))+ ylim(-10,10) + geom_point()+ ge
```

```
ggplot(data = NQ2008, mapping = aes(x=Day, y = Res.Dr.1, colour=Year)) + geom_point()+ geom_smooth(meth
```

```
summary(CO2i)

library(corrplot)

CO2icorr<- CO2i[,-1]#Remove column

CO2icorr<-cor(CO2i)

corrplot(CO2icorr, method="square", main="World Emissions")
```

```
ggplot(data=CO2i, mapping=aes(x=CO2C, y=SST, colour=Year)) + geom_line(size=1) + ggtitle("CO2 concentra
```

```
geom_smooth(method="loess", formula=y~x, colour="black")

ggplot(data=CO2i, mapping=aes(x=CO2C, y=SST, colour=Year)) + geom_line(size=1) + ggtitle("CO2 concentra
```

```
ggplot(data=CO2i, mapping=aes(y=CO2C, x=Year)) + geom_line(size=1) + ggtitle("CO2 concentration over ti
```

##Yearly temperature average

```
library(ggplot2)

ggplot(data=Globtemp, mapping=aes(x=Year, y=Yearly_average)) + geom_line(size=1) + ggtitle("Average yea
```

```
library(forecast)

GlobtempTS=ts(Globtemp)

arima_fittemp = auto.arima(GlobtempTS[,2])

arima_forecasttemp=forecast(arima_fittemp, h=10)

plot(arima_forecasttemp)
```

1= means depends on previous value 3= depend son 3 average values before middle 1 =

#Need to understand the equation and understand this in enough depth to enterpret it and explain whether it fits the data set - research and fully understand the equation. ARIMA you must look at each variable independantly among their own observations over time.

###Ideas: Use other model to compare two variables rather than just comparing temp over time (bivriate model). Can predict sea levels using other variables.

#Model - multivariate regression. Use models to compare other variates to sea levels and perhaps hypothesise using these models future sea levels based on other covariates. Also allows more discussion/connection between the variables.

#Talk about the fact that time series must be collected at the same time to properly be compared - we cannot be sure of this but as long as the data has been collected appropriately (i.e not in Jan and then Aug) it should suit for the task. ~worth discussing though.

```
ggplot(data=ArcticIce, mapping=aes(x=year, y=area)) + geom_line(size=1) + ggtitle("Average area of Arcti
```

```
iceTS=ts(ArcticIce)

arima_fitice = auto.arima(iceTS[,3])

arima_forecastice=forecast(arima_fitice, h=5)

plot(arima_forecastice)
```

```
sealevelsav<- data.frame(Sealevel$X, Sealevel$GMSL)

library(tidyverse)

yearlyavSL<-sealevelsav%>%group_by(Sealevel.X)%>%summarise(average=mean(Sealevel.GMSL))

library(ggplot2)

ggplot(data=yearlyavSL, mapping=aes(x=Sealevel.X, y=average)) + geom_line(size=1) + ggtitle("Average sea
```

```
ggplot(data = yearlyavSL, mapping = aes(x=Sealevel.X, y = average)) + geom_point()+ geom_smooth(method=
```

```
avsealevelTS=ts(yearlyavSL)

arima_fitSL = auto.arima(avsealevelTS[,2])

arima_forecastSL=forecast(arima_fitSL, h=10)

plot(arima_forecastSL)
```

```
OBPav<- data.frame(NQ$Year, NQ$Res.Dr.1)

library(tidyverse)
library(dplyr)


OBPyearav<-OBPav%>%group_by(NQ.Year)%>%summarise(average=mean(NQ.Res.Dr.1))


write.csv(OBPyearav, "D:/data analytics year/Communicating and presenting results\\OBPyearlyav.csv", ro
```

```
OBPavS<- data.frame(SQ$Year, SQ$Res.Dr.1)



OBPyearavS<-OBPavS%>%group_by(SQ.Year)%>%summarise(average=mean(SQ.Res.Dr.1))


write.csv(OBPyearavS, "D:/data analytics year/Communicating and presenting results\\OBPyearlyavsouth.cs
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.