MH3511-project-submission

Zhiyi, Nicholas, Ancel (Group 57)

2023-04-20

## Set up

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(readxl)  
library(dplyr)  
library(scales)

##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

library(gghalves)  
library(ggplot2)  
library(lubridate)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

library(ncdf4)  
library(tibbletime)

##   
## Attaching package: 'tibbletime'  
##   
## The following object is masked from 'package:stats':  
##   
## filter

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

#check which ones are needed later  
library(parallel)  
library(neuralnet)

##   
## Attaching package: 'neuralnet'  
##   
## The following object is masked from 'package:dplyr':  
##   
## compute

library(randomForest)

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:gridExtra':  
##   
## combine  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(xgboost)

##   
## Attaching package: 'xgboost'  
##   
## The following object is masked from 'package:dplyr':  
##   
## slice

library(keras)  
library(tensorflow)  
library(rnn)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:tensorflow':  
##   
## train  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(tseries)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(forecast)  
library(pracma)

##   
## Attaching package: 'pracma'  
##   
## The following object is masked from 'package:purrr':  
##   
## cross

library(car)

## Loading required package: carData  
##   
## Attaching package: 'car'  
##   
## The following object is masked from 'package:pracma':  
##   
## logit  
##   
## The following object is masked from 'package:dplyr':  
##   
## recode  
##   
## The following object is masked from 'package:purrr':  
##   
## some

# 1. Introduction

Singapore’s climate is affected by many different driving factors, like the El Niño-Southern Oscillation (ENSO), the Indian Ocean Dipole, the Madden-Julian Oscillation, etc., with varying timescales and impacts on Singapore. In this report, we will attempt to understand how ENSO affects Singapore.

We will look at 4 spatial datasets, specifically focusing on monthly data for a region in the Pacific Ocean from 1982-2020, that are used an indicators for the strength of ENSO: Sea Surface Temperature, Subsurface Sea Temperature, Low Level Wind Speed, and Outgoing Longwave Radiation (as a measure of cloudiness). We will also look at Singapore’s rainfall and surface air temperature over the corresponding period, taken from the Changi weather station, to understand how they relate to the ENSO indicators.

# 2. Data Description

Datasets for the ENSO indicators were obtained from the following sources.

* Sea Surface Temperature - OISST.v2, <https://www.cpc.ncep.noaa.gov/data/indices/sstoi.indices>
* Subsurface Sea Temperature - EN4, (<https://climatedataguide.ucar.edu/climate-data/en4-subsurface-temperature-and-salinity-global-oceans>)
* Low Level Wind Speed - ICOADS, (<https://psl.noaa.gov/data/gridded/data.coads.1deg.html>)
* Outgoing Longwave Radiation - CDR (<https://www.ncei.noaa.gov/products/climate-data-records/outgoing-longwave-radiation-monthly>)

These spatial datasets generally contained global data, but of interest to our project is data that lies within the Nino3.4 region (Latitude: 5N-5S, Longitude:170W-120W).

Datasets for Singapore were obtained from data.gov.sg

* Rainfall - <https://data.gov.sg/dataset/rainfall-monthly-total>
* Surface Air Temperature (SAT) - <https://data.gov.sg/dataset/surface-air-temperature-monthly-mean>

Before proceeding to data analysis, we had to clean and process the data. We will outline the general procedure for doing so, though any exceptions will be mentioned in Section 3. The main goal of the processing is to calculate a baseline value for each month based on a reference period (i.e., what is the average value for Jan, Feb, Mar… Dec from Jan 1980 - Dec 2010), which we can use to calculate the anomaly of the data from Jan 2011 - Dec 2022 (i.e. how much the data for a month differs from the average value for that month)

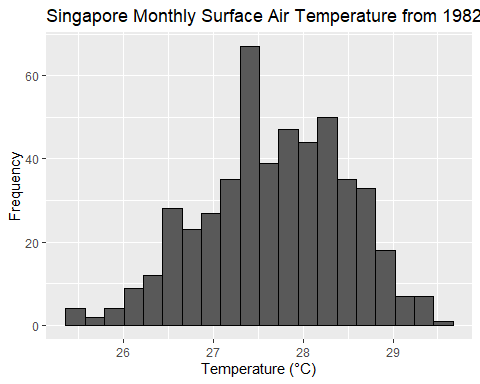
* Data was stored in a .nc file, read using nc\_open from library ncdf4
* 3 dimensions, corresponding to latitude (1 degree), longitude (1 degree), time (monthly)
* For each time value:
* Find all the values with latitude and longitude that fall within the Nino3.4 region
* Check if there are any NA values, if there are, interpolate to fill them
* Calculate the mean of those values
* Extract the means corresponding to the reference period (Jan 1982 - Dec 2010), and calculate the baseline mean for each month (Jan, Feb, Mar… Dec)
* Calculate the difference between the means for the study period (Jan 2011 - Dec 2022) and the baseline means

# 3. Description and Cleaning of Dataset

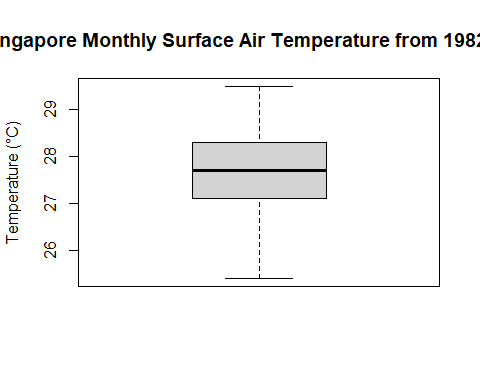
We obtained data from various sources, spanning a timeframe of 1982-2022. We segmented our data into 2 portions, 1980-2010 as a baseline to evaluate anomalies, and 2011-2023 as a test to be evaluated based on our baseline to check if they are anomalous. We plotted the data through a histogram, and boxplot to determine the suitability of using the data.

### 3.1.1: Singapore Monthly Surface Air Temperature (sat)

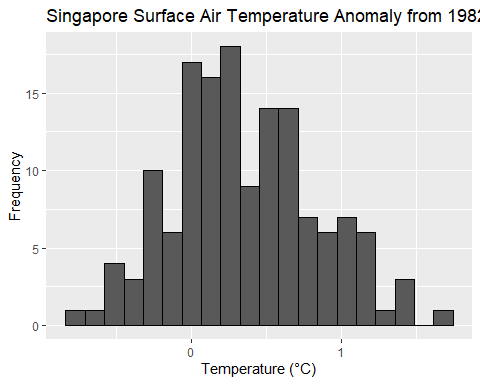
#creating a time vector for the period of data we will use. 1982-2022 (41 years)  
time.all <- seq(as.Date('1982-01-01'),as.Date('2022-12-01'),by='month')  
  
#creating a time vector for the study period we will use. 2011-2022 (12 years)  
time.study <- seq(as.Date('2011-01-01'),as.Date('2022-12-01'),by='month')  
  
#creating a time vector for the reference period we will use. 1982-2010 (29 years)  
time.ref <- seq(as.Date('1982-01-01'),as.Date('2010-12-01'),by='month')  
  
sat.all <- read.csv('data/surface-air-temperature-monthly-mean.csv')  
  
sat.all[,1]<-as.Date(paste(sat.all[,1], "-01", sep=""))  
  
sat.all <- sat.all[sat.all$month >= as.Date('1982-01-01') & sat.all$month <= as.Date('2022-12-31'),]  
  
#getting reference range..  
sat.refVal <- sat.all$mean\_temp[sat.all$month >= as.Date('1982-01-01') & sat.all$month <= as.Date('2010-12-31')]  
  
#calculate avg temp/month in ref period to act as a baseline for anomalies..  
sat.baseline <- sapply(1:12, function(i){  
 #avg.temp <- round(mean(sat.ref[which(month(sat.ref$month)==m),]$mean\_temp),4)  
 sat.refVal[seq(i,length(sat.refVal),12)] %>% mean()  
})  
  
sat.studyVal <- sat.all$mean\_temp[sat.all$month >= as.Date('2011-01-01') & sat.all$month <= as.Date('2022-12-31')]  
  
#calculating sat anomalies from 2011 onwards..  
sat.studyAnom = sat.studyVal - rep(sat.baseline,(2023-2011))  
  
ggplot(sat.all)+  
 geom\_histogram(aes(x=mean\_temp),bins = 20,color='black')+  
 labs(x = 'Temperature (\u00B0C)', y='Frequency') +  
 ggtitle("Singapore Monthly Surface Air Temperature from 1982-2022")



boxplot(sat.all$mean\_temp,  
 main='Singapore Monthly Surface Air Temperature from 1982-2022',  
 ylab = 'Temperature (\u00B0C)')



ggplot(data.frame(sat.studyAnom))+  
 geom\_histogram(aes(x=sat.studyAnom),bins = 20,color='black')+  
 labs(x = 'Temperature (\u00B0C)', y='Frequency') +  
 ggtitle("Singapore Surface Air Temperature Anomaly from 1982-2022")

 The data was obtained from data.gov.sg, and saved in the form of a csv. We used the function read.csv to extract the data from the file where we obtained surface air temperature (°C) from 1982-2023.

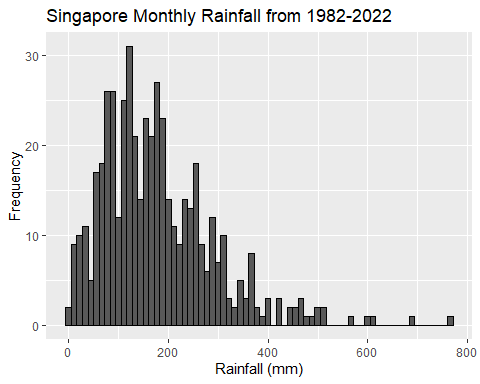
For the monthly histogram, we see a unimodal distribution with no apparent outliers. The temperature values seem to be logical as well, ranging roughly from 25 to 30 degrees.

From the monthly boxplot, we also see no apparent outliers. Thus, we conclude that the data is suitable for further processing as it is.

From here, we filtered out the data to create our reference and anomaly data sets. The anomaly data is plotted with both a histogram to show the distribution. The anomaly histogram is generally unimodal.

### 3.1.2: Singapore Monthly Rainfall (rainf)

#reading rainfall data..  
rainf.all <- read.csv('data/rainfall-monthly-total.csv')  
  
rainf.all[,1]<-as.Date(paste(rainf.all[,1], "-01", sep=""))  
  
rainf.all <- rainf.all[rainf.all$month >= as.Date('1982-01-01') & rainf.all$month <= as.Date('2022-12-31'),]  
  
#getting reference range for rainfall..  
rainf.refVal <- rainf.all$total\_rainfall[rainf.all$month >= as.Date('1982-01-01') & rainf.all$month <= as.Date('2010-12-31')]  
  
#calculate avg rainf/month in ref period to act as a baseline for anomalies..  
rainf.baseline<- sapply(1:12, function(i){  
 #avg.temp <- round(mean(sat.ref[which(month(sat.ref$month)==m),]$mean\_temp),4)  
 rainf.refVal[seq(i,length(rainf.refVal),12)] %>% mean()  
})  
  
#find rainfall data period of interest, 2011 onwards..  
rainf.studyVal <- rainf.all$total\_rainfall[rainf.all$month >= as.Date('2011-01-01') & rainf.all$month <= as.Date('2022-12-31')]  
  
#caluclating rainfall anomalies from 2011..  
rainf.studyAnom<- rainf.studyVal - rep(rainf.baseline,(2023-2011))  
  
ggplot(rainf.all)+  
 geom\_histogram(aes(x=total\_rainfall),bins = 70,color='black')+  
 labs(x = 'Rainfall (mm)', y='Frequency') +  
 ggtitle("Singapore Monthly Rainfall from 1982-2022")



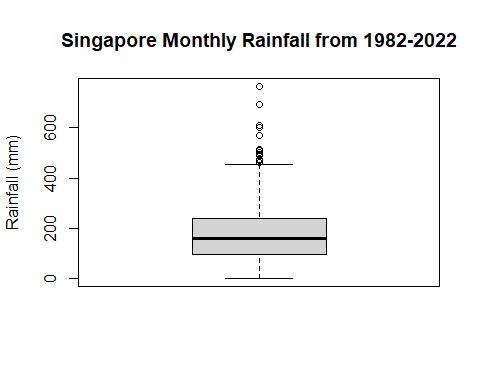
#find outlier month and print  
outlier.month <- rainf.all$month[which.max(rainf.all$total\_rainfall)]  
print('Potential outlier at the date:')

## [1] "Potential outlier at the date:"

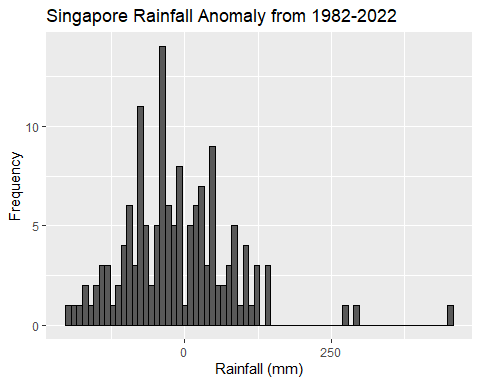
print(outlier.month)

## [1] "2006-12-01"

boxplot(rainf.all$total\_rainfall,  
 main='Singapore Monthly Rainfall from 1982-2022',  
 ylab = 'Rainfall (mm)')



ggplot(data.frame(rainf.studyAnom))+  
 geom\_histogram(aes(x=rainf.studyAnom),bins = 70,color='black')+  
 labs(x = 'Rainfall (mm)', y='Frequency') +  
 ggtitle("Singapore Rainfall Anomaly from 1982-2022")

 The data was obtained from data.gov.sg, and saved in the form of a csv. We used the function read.csv to extract the data from the file where we obtained total monthly rainfall (mm) from 1982-2023.

From the histogram, the distribution is generally unimodal, but we see clear right skew, with extreme values on the right greater than 450 mm which could potentially be outliers.

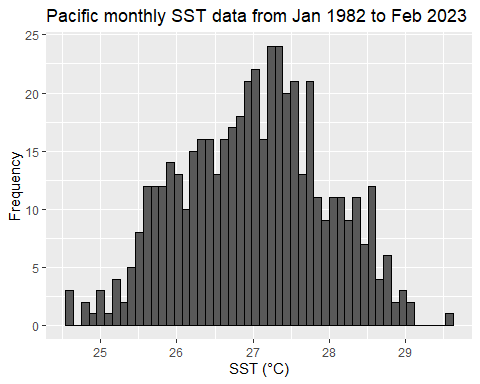
From the boxplot, we again see a similar right skew and potential outliers with extreme values greater than about 450 mm.

However,according to Meteorological Service Singapore (MSS), December 2006 had a record rainfall level. This verified that this data point was not an outlier, thus we can safely regard the other values lower than this data point as logical values. <http://www.weather.gov.sg/climate-historical-extremes/>

Therefore, we concluded that this data set is safe to use in future processings. Again, we filtered out the data to create the reference and anomaly data sets. The anomaly data was plotted with a histogram to show its distribution. The distribution is generally unimodal with a large right skew.

### 3.1.3: SST Summary Statistics for SST

#read sst data  
sst <- read.table('data/sstoi.indices.txt',header=T)  
  
sst.all <- data.frame(month = as.Date(paste(sst$YR,sst$MON,'01',sep='-')),  
 sst = sst$NINO3.4)  
  
sst.all <- sst.all[!duplicated(sst.all$month),]  
  
sst.hist <- ggplot(data=sst.all, aes(x=sst))+  
 geom\_histogram(color='black',bins=50)+  
 xlab('SST (\u00B0C)')+  
 ylab('Frequency')+  
 ggtitle('Pacific monthly SST data from Jan 1982 to Feb 2023')  
   
sst.hist #generally unimodal, normal dist.



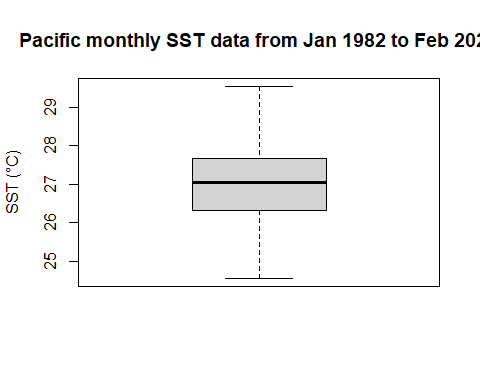
#find outlier month and print  
outlier.month <- sst.all$month[which.max(sst.all$sst)]  
print('Potential outlier at the date:')

## [1] "Potential outlier at the date:"

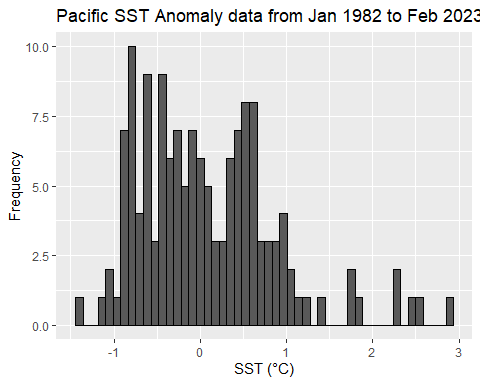
print(outlier.month)

## [1] "2015-11-01"

boxplot(sst.all$sst,   
 main = 'Pacific monthly SST data from Jan 1982 to Feb 2023',  
 ylab = 'SST (\u00B0C)')



#finding ref range 1980-2010  
sst.ref <- sst.all[sst.all$month >= '1982-01-01' & sst.all$month < '2011-01-01',]  
  
sst.studyVal <- sst.all[sst.all$month >= '2011-01-01' & sst.all$month < '2023-01-01',2]  
  
sst.refVal <- sst.ref[,2]  
  
#sst baseline per month   
sst.baselines <- sapply(1:12, function(m){  
 baseline <- mean(sst.ref[which(month(sst.ref$month)==m),2])  
})  
  
# finding anomaly with fixed dates. assume data was with respect to first day of the month  
# note the data has anomaly values but wrt to 1990-2020. So we calc our own  
sst.studyAnom <- sst.studyVal - rep(sst.baselines,(144/12))  
  
sst.histAnom <- ggplot(data.frame(sst.studyAnom), aes(x=sst.studyAnom))+  
 geom\_histogram(color='black',bins=50)+  
 xlab('SST (\u00B0C)')+  
 ylab('Frequency')+  
 ggtitle('Pacific SST Anomaly data from Jan 1982 to Feb 2023')  
   
sst.histAnom #generally unimodal, normal dist.

 The data was obtained from National Oceanic and Atmospheric Administration and saved in the form of a text file where we used read.table to extract the data from the file where we obtained sea surface temperature (°C) from 1982-2023.

From the histogram, the distribution is generally unimodal and quite normal. We observe a potential outlier on the right of the plot.

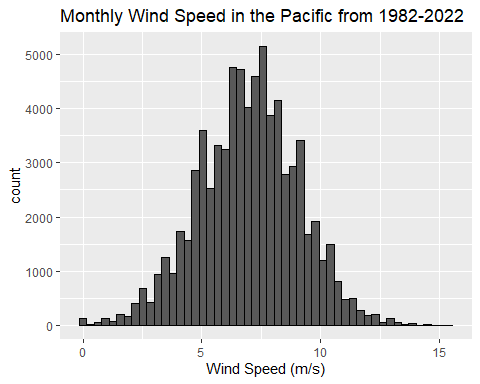
From the boxplot, we see no skewness of the data, and no potential outliers. However, we will check the outlying point from the histogram.

Upon further research, we discovered that the European Centre for Medium-Range Weather Forecasts (ECMWF) had reported an El Nino peak during November 2015, supporting this high temperature recorded during that same month. Thus, this data has no outliers and we will proceed with the data. <https://www.ecmwf.int/en/about/media-centre/news/2015/records-tumble-el-nino-peaks>

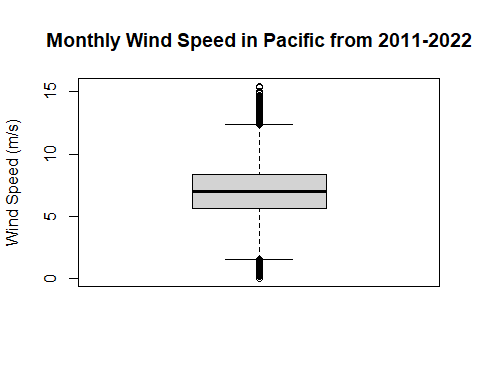
Finally, we segmented our data to form the reference and anomaly data sets. We the plotted the anomaly data on a histogram. The distribution of the anomaly histogram is generally bimodal. This seems to make sense since ENSO will have cycles of low and high temperatures, resulting in negative and positive anomalies.

### 3.1.4: Reading Low level wind data

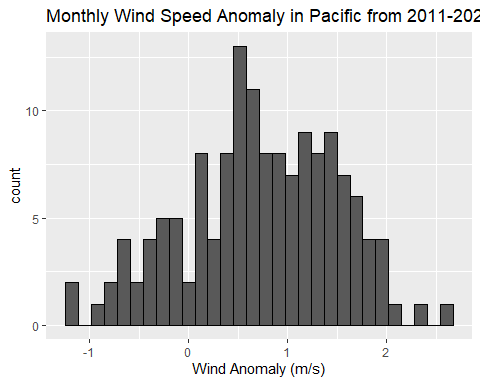
wind.nc = nc\_open('data/wspd.mean.nc')  
  
#3D array, lat, lon, time dimensions  
wind.data = ncvar\_get(wind.nc, 'wspd')  
  
#calculating baseline values from 1980-2010  
wind.time = ncvar\_get(wind.nc,"time") #data starts from 1800/01, end at 2023/01  
# startMonth = (1980-1800)\*12+1 #1980/01  
# endMonth = (2011-1800)\*12 #2010/12  
  
wind.lat = ncvar\_get(wind.nc,'lat')  
wind.lon = ncvar\_get(wind.nc,'lon')  
wind.latIndices = which(wind.lat<=5 & wind.lat>=-5)  
wind.lonIndices = which(wind.lon<=240 & wind.lon>=190)  
  
#get the mean of the wind values within the Nino3.4 region  
wind.ninoArea = function(index) {  
 wind.slice = wind.data[,,index]  
 wind.slice = zoo::na.approx(wind.slice)  
 return(wind.slice[wind.lonIndices,wind.latIndices])  
}  
  
#all wind values  
wind.all = lapply(((1982-1800)\*12+1):((2023-1800)\*12),wind.ninoArea)  
wind.allDF = data.frame(values=as.vector(do.call(rbind,wind.all)))  
  
ggplot(data=wind.allDF) +  
 geom\_histogram(aes(x=values),bins=50,color='black')+  
 ggtitle('Monthly Wind Speed in the Pacific from 1982-2022')+  
 xlab('Wind Speed (m/s)')



boxplot(wind.allDF$values,  
 main='Monthly Wind Speed in Pacific from 2011-2022',  
 ylab = 'Wind Speed (m/s)')



wind.allMeans = sapply(wind.all, mean)  
  
wind.endRefIndex = ((2011-1982)\*12)  
#get the means corresponding to reference period  
wind.refVal = wind.allMeans[1:wind.endRefIndex]#sapply(startMonth:endMonth, ninoAreaMean)  
  
#calculate a baseline temperature for each month based on the reference period  
wind.baselines = sapply(1:12,function(i) {  
 monthValues = wind.refVal[seq(i,length(wind.refVal),12)]  
 return(mean(monthValues))  
})  
  
#calculating anomaly (temperature for a given month - corresponding baseline)  
  
wind.studyVal = wind.allMeans[(wind.endRefIndex+1):length(wind.allMeans)]  
wind.studyAnom = wind.studyVal - rep(wind.baselines,(2023-2011))  
  
wind.studyDF <- data.frame(wind.studyAnom, month=time.study)  
  
ggplot(wind.studyDF)+  
 geom\_histogram(aes(x=wind.studyAnom),bins = 30,color='black')+  
 ggtitle('Monthly Wind Speed Anomaly in Pacific from 2011-2022')+  
 xlab('Wind Anomaly (m/s)')

 The data was obtained from International Comprehensive Ocean-Atmosphere Data Set (ICOADS) and was saved in the form of an nc file. We used the function nc\_open to access the file. The data saved in the file is 3 dimensional as it contains the data recorded by latitude, longitude, and time. Therefore, we had to extract values first based on the latitude and longitude of our area of interest, and averaging the values of the data points within this area to convert the data set from 3D to 2D. We plotted the monthly data using a histogram and a box plot.

From the monthly histogram, we see a unimodal normal distribution of windspeeds with no outliers.

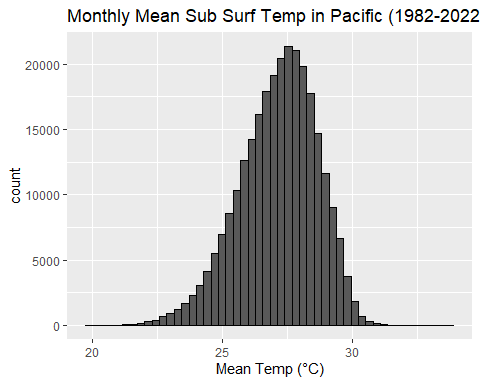
From the boxplot, there is some suggestion of extreme values that could be outliers.

However, upon further research, these extreme values seem to be very reasonable as open ocean winds can be very strong, thus we decided to take these values as valid points. This is supported by the data in the link below: <https://iridl.ldeo.columbia.edu/maproom/Global/Atm_Circulation/Tropical_Wind_PacOcean.html?T=Oct%202022>

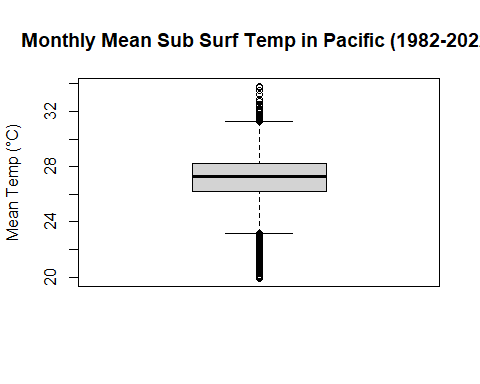
Finally, we obtained our reference and anomaly data and plotted the anomaly using a histogram. The histogram is generally unimodal.

### 3.1.5: Reading Sub Surface Temp data

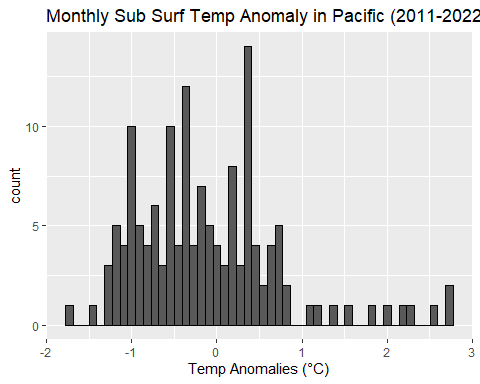
months <- 01:12  
  
years <- 1982:2022  
  
substemp.folder <- 'data/subsurfacetempfiles/'  
  
substemp.file <- paste0(substemp.folder, 'EN.4.2.2.f.analysis.g10.')  
  
#extracting subsurface data from netcdf files  
substemp <- lapply(1:length(years),function(i){  
   
 monthly <- lapply(1:length(months),function(k){  
   
 if(months[k]<10){  
 data <- nc\_open(paste(substemp.file,years[i],'0',months[k],'.nc',sep=''))  
 }  
 else{  
 data <- nc\_open(paste(substemp.file,years[i],months[k],'.nc',sep=''))  
 }  
  
   
 })  
   
})  
  
#finding the temp at each month and each year  
substemp.all <- lapply(1:length(substemp),function(j){  
   
 monthly <- lapply(1:length(months),function(n){  
   
 #finding all the temperature data  
 data.temp <- (ncvar\_get(substemp[[j]][[n]],'temperature'))  
   
 data.temp <- zoo::na.approx(data.temp)  
   
 data.temp <- data.temp - 273.15  
   
 #monthly temp in the area of interest, lon 190-240 (-170,-120), lat 78-88 (-5,5)  
 all.temp <- data.temp[190:240,78:88]  
 #temp <- data.temp[190:240,78:88,]  
   
 })  
   
})  
  
substemp.all <- unlist(substemp.all,recursive=F)  
  
substemp.all.plot <- unlist(substemp.all) #561924 NA values  
  
#plotting timeseries for substemp  
substemp.ts.plot <- data.frame(month = seq(as.Date('1982-01-01'),  
 as.Date('2022-12-01'),  
 by='month'),  
 substemp.all = substemp.all.plot)  
  
substemp.hist <- ggplot(data=data.frame(substemp.all.plot),aes(x=substemp.all.plot))+  
 geom\_histogram(color='black', bins=50)+  
 ggtitle('Monthly Mean Sub Surf Temp in Pacific (1982-2022)')+  
 xlab('Mean Temp (\u00B0C)')  
  
substemp.hist



substemp.box <- boxplot(substemp.all.plot,  
 main='Monthly Mean Sub Surf Temp in Pacific (1982-2022)',  
 ylab = 'Mean Temp (\u00B0C)')



#finding the mean temp at each month and each year  
substemp.mean <- sapply(1:length(substemp),function(j){  
   
 monthly <- sapply(1:length(months),function(n){  
   
 #finding all the temperature data  
 data.temp <- (ncvar\_get(substemp[[j]][[n]],'temperature'))  
   
 data.temp <- zoo::na.approx(data.temp)  
   
 data.temp <- data.temp - 273.15  
   
 #monthly temp in the area of interest, lon 190-240 (-170,-120), lat 78-88 (-5,5)  
 all.temp <- mean(data.temp[190:240,78:88])  
 #temp <- data.temp[190:240,78:88,]  
   
 })  
   
})  
  
rm(substemp) #remove substemp to save space  
  
substemp.refVal <- substemp.mean[,1:(2010-1982+1)] #29 columns for 29 years (1982-2010)  
  
colnames(substemp.refVal) <- seq(1982,2010)  
  
substemp.studyVal <- substemp.mean[,(2011-1982+1):ncol(substemp.mean)] #29 columns for 29 years (1982-2010)  
  
colnames(substemp.studyVal) <- seq(2011,2022)  
  
#calc baseline from refVal  
substemp.baselines <- sapply(1:nrow(substemp.refVal),function(i){  
 monthly.mean <- mean(substemp.refVal[i,])  
})  
  
#calc studyAnom  
substemp.studyAnom <- c(substemp.studyVal-rep(substemp.baselines,(2022-2011+1)))  
  
substemp.histAnom <- ggplot(data=data.frame(x=substemp.studyAnom),  
 aes(x=x))+  
 geom\_histogram(color='black', bins=50)+  
 ggtitle('Monthly Sub Surf Temp Anomaly in Pacific (2011-2022)')+  
 xlab('Temp Anomalies (\u00B0C)')  
  
substemp.histAnom



The data was obtained from National Center for Atmospheric Research and saved in the form of multiple nc files, where each file corresponded to 1 year of data. We had to create a loop to extract each files consecutively and save the data into a variable, which was a 3D array with longitiude, latitude, and month. From there, we had to extract the data that we needed from the specific latitudes and longitudes that we needed (nino 3.4), as well as the time frame that we wanted (1982-2022). However, we did find that this data had NA values in it, and therefore thought to interpolate the data for those missing data points. We plotted the monthly data as a histogram and boxplot.

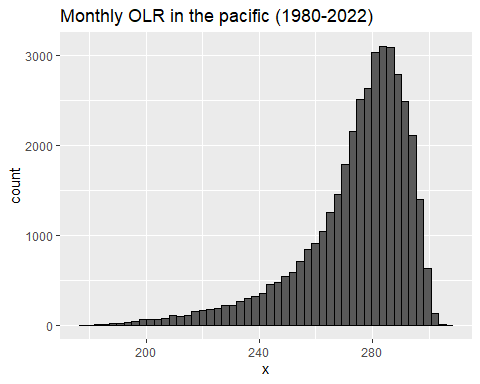
From the monthly histogram, we see a unimodal normal distribution with a very slight left skew, with no potential outliers.

From the boxplot, we see that the extreme values could be potential outliers. However, these values are still within the accepted range for this type of data, thus we decided to keep it.

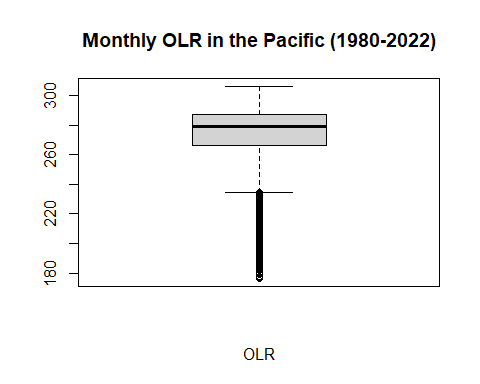
We then obtained our reference and anomaly data sets, plotting the anomaly data on a histogram. The anomaly histogram is multimodal with a right skew. However, we note that the original data had no outliers, thus this distribution is not erroneous.

### 3.1.6: Monthly OLR (Okla) in the Pacific

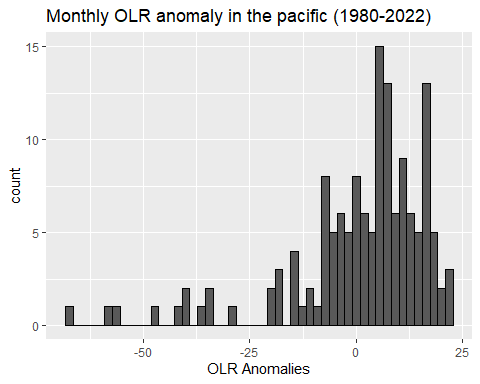
nc\_data <- nc\_open("data/olr-monthly\_v02r07\_197901\_202303.nc")  
  
lat <- ncvar\_get(nc\_data, "lat")  
  
nlat <- dim(lat)  
  
lon <- ncvar\_get(nc\_data, "lon")  
  
nlon <- dim(lon)  
  
time <- ncvar\_get(nc\_data, "time")  
  
tunits <- ncatt\_get(nc\_data, "time", "units")   
nt <- dim(time)   
  
cloudiness <- ncvar\_get(nc\_data, "olr")   
  
fillvalue <- ncatt\_get(nc\_data, "olr", "\_FillValue")  
  
cloudiness[cloudiness==fillvalue$value] <- NA  
  
# Convert the number of hours to POSIXct format, specifying the origin as "1800-01-01"  
datetime <- as.Date(time, origin="1979-01-01")  
  
lat\_range <- c(-5, 5)  
lon\_range <- c(190, 240)  
  
lat\_idx <- which(lat >= lat\_range[1] & lat <= lat\_range[2])  
lon\_idx <- which(lon >= lon\_range[1] & lon <= lon\_range[2])  
  
id.all <- which(datetime >= '1982-01-01' & datetime < '2023-01-1')  
okla.all <- cloudiness[lon\_idx,lat\_idx,id.all]  
  
okla.all.mean <- sapply(1:dim(okla.all)[3],function(i){  
 val <- mean(okla.all[,,i])  
})  
  
okla.all.mean <- data.frame(okla = okla.all.mean,  
 month = seq(as.Date('1982-01-01'),  
 as.Date('2022-12-01'),  
 by='month'))  
colnames(okla.all.mean) <- c('okla','month')  
  
# plotting hist and box for okla values  
okla.hist <- ggplot(data=data.frame(x=c(okla.all)),aes(x=x))+  
 geom\_histogram(color='black',bins=50)+  
 ggtitle('Monthly OLR in the pacific (1980-2022)')  
okla.hist



okla.box <- boxplot(c(okla.all),  
 main='Monthly OLR in the Pacific (1980-2022)',  
 xlab = 'OLR')



id.ref <- which(datetime >= '1982-01-01' & datetime < '2011-01-1')  
okla.refVal <- sapply(1:(length(id.ref)/12),function(i){  
 monthly <- sapply(1:12,function(j){  
 val <- mean(cloudiness[lon\_idx,lat\_idx,i\*j])  
 })  
})  
  
id.study <- which(datetime >= '2011-01-01' & datetime < '2023-01-1')  
okla.studyVal <- sapply(1:length(id.study),function(i){  
 val <- mean(cloudiness[lon\_idx,lat\_idx,i])  
})  
  
#get baseline values  
okla.baselines <- sapply(1:12,function(i){  
 val <- mean(okla.refVal[i,])  
})  
  
#get anom values  
okla.studyAnom <- okla.studyVal - rep(okla.baselines,144/12)  
  
#plotting hist and box of okla anomalies  
okla.histAnom <- ggplot(data=data.frame(okla.studyAnom),aes(x=okla.studyAnom))+  
 geom\_histogram(color='black',bins=50)+  
 ggtitle('Monthly OLR anomaly in the pacific (1980-2022)')+  
 xlab('OLR Anomalies')  
okla.histAnom



The data was obtained from National Oceanic and Atmospheric Administration and was saved in the form of an nc file. We used the function nc\_open to access the file. The data saved in the file is 3 dimensional as it contains the data recorded by latitude, longitude, and time. Therefore, we had to extract values first based on the latitude and longitude of our area of interest, and averaging the values of the data points within this area to convert the dataset from 3D to 2D. We plotted our monthly OLR data using a histogram and a boxplot.

From our monthly histogram, we observe a unimodal distribution with a heavy left skew.

From our boxplot, we also see a heavy left skew with potential outliers on the left. Based on literature, there was a strong El Nino event during that period, which would result in higher cloud cover, thus low OLR values. These values of low OLR are thus justified and likely accurate data for strong El Nino events. We therefore decide to include them in our analysis. <https://www.ecmwf.int/en/about/media-centre/news/2015/records-tumble-el-nino-peaks>

We obtained the reference and anomaly data for OLR and plotted the anomaly with a histogram. The distribution is generally unimodal with a similar left skew.

### 3.1.7: Compilation of all variables

enso.df <- data.frame(month = time.study,   
 sat.sg=sat.studyAnom, # Normally distributed  
 rainf.sg=rainf.studyAnom, #  
 sst.pcf=sst.studyAnom,  
 substemp.pcf=substemp.studyAnom,  
 okla.pcf=okla.studyAnom,  
 wind.pcf=wind.studyAnom)  
  
enso.df.nd <- enso.df[,2:ncol(enso.df)]

We now compiled all our anomaly data into one cohesive data set for ease of processing. We made 2 data sets, 1 with the date column included and one without.

## 3.2: Time Shift of predictor variables

There may be time lag between the Pacific and Singapore variables due to the distance between them. To investigate what this time lag may be, we calculated the correlation between each pacific variable and singapore variable for a series of offsets (i.e. Jan data from Pacific against Feb data from Singapore), up to a maximum of 12 months. We identified the offset that maximised correlation as the best timeshift value for that combination of Pacific-Singapore variables.

### 3.2.1: Time Shift of SST

# timeshifting to find best correlation for sst & sg sat, consider only forward shift since we using pacific data to predict sg data  
corr.ts.sat.sst <- sapply(1:12,function(i){  
   
 ts.sst <- sst.studyAnom[i:length(sst.studyAnom)]  
 ts.sat <- sat.studyAnom[1:(length(sat.studyAnom)-i+1)]  
 cor.ts <- cor(ts.sst,ts.sat)  
 month.cor <- c(i,cor.ts)  
 return (month.cor)  
})  
  
corr.ts.sat.sst.df <- data.frame(month = corr.ts.sat.sst[1,],  
 corr.sst.sat = corr.ts.sat.sst[2,])  
  
sst.timeshift.sat <- which.max(abs(corr.ts.sat.sst.df$corr.sst.sat)) -1  
  
# timeshifting to find best correlation for sst & sg rainf, consider only forward shift since we using pacific data to predict sg data  
corr.ts.rainf.sst <- sapply(1:12,function(i){  
   
 ts.sst <- sst.studyAnom[i:length(sst.studyAnom)]  
 ts.rainf <- rainf.studyAnom[1:(length(rainf.studyAnom)-i+1)]  
 cor.ts <- cor(ts.sst,ts.rainf)  
 month.cor <- c(i,cor.ts)  
 return (month.cor)  
})  
  
corr.ts.rainf.sst.df <- data.frame(month = corr.ts.rainf.sst[1,],  
 corr.sst.rainf = corr.ts.rainf.sst[2,])  
  
sst.timeshift.rainf <- which.max(abs(corr.ts.rainf.sst.df$corr.sst.rainf))-1  
  
paste('The best timeshift for SST with Rainfall is',sst.timeshift.rainf)

## [1] "The best timeshift for SST with Rainfall is 0"

paste('The best timeshift for SST with SAT is',sst.timeshift.sat)

## [1] "The best timeshift for SST with SAT is 0"

### 3.2.2: Time Shift of Wind

#timeshift for wind and sat  
cors = rep(0,13)  
for (i in 0:12) {  
 windanomaly = wind.studyAnom[1:(144-i)] #1:144 to 1:132  
 sat\_anom = sat.studyAnom[(1+i):(144)] #1:144 to 13:144  
 cors[i]=cor(windanomaly,sat\_anom)  
}  
  
#should use the wind data to predict SG temperature x months in the future  
#i.e. jan win data is used to predict x SG temp  
wind.timeshift.sat = which.max(abs(cors))  
  
#time shift for wind and rainf  
cors = rep(0,13)  
for (i in 0:12) {  
 windanomaly = wind.studyAnom[1:(144-i)] #1:144 to 1:132  
 rainf\_anom = rainf.studyAnom[(1+i):(144)] #1:144 to 13:144  
 cors[i]=cor(windanomaly,rainf\_anom)  
}  
  
#same process as for sat  
wind.timeshift.rainf = which.max(abs(cors))  
  
paste('The best timeshift for Wind with Rainfall is',wind.timeshift.rainf)

## [1] "The best timeshift for Wind with Rainfall is 7"

paste('The best timeshift for Wind with SAT is',wind.timeshift.sat)

## [1] "The best timeshift for Wind with SAT is 4"

### 3.2.3: Time Shift of Sub Surf Temp

# timeshifting to find best correlation for substemp & sg sat, consider only forward shift since we using pacific data to predict sg data  
corr.ts.sat.substemp <- sapply(1:12,function(i){  
   
 ts.substemp <- substemp.studyAnom[i:length(substemp.studyAnom)]  
 ts.sat <- sat.studyAnom[1:(length(sat.studyAnom)-i+1)]  
 cor.ts <- cor(ts.substemp,ts.sat)  
 month.cor <- c(i,cor.ts)  
 return (month.cor)  
})  
  
corr.ts.sat.substemp.df <- data.frame(month = corr.ts.sat.substemp[1,],  
 corr.substemp.sat = corr.ts.sat.substemp[2,])  
  
substemp.timeshift.sat <- which.max(abs(corr.ts.sat.substemp.df$corr.substemp.sat))-1  
  
# timeshifting to find best correlation for substemp & sg rainf, consider only forward shift since we using pacific data to predict sg data  
corr.ts.rainf.substemp <- sapply(1:12,function(i){  
   
 ts.substemp <- substemp.studyAnom[i:length(substemp.studyAnom)]  
 ts.rainf <- rainf.studyAnom[1:(length(rainf.studyAnom)-i+1)]  
 cor.ts <- cor(ts.substemp,ts.rainf)  
 month.cor <- c(i,cor.ts)  
 return (month.cor)  
})  
  
corr.ts.rainf.substemp.df <- data.frame(month = corr.ts.rainf.substemp[1,],  
 corr.substemp.rainf = corr.ts.rainf.substemp[2,])  
  
substemp.timeshift.rainf <- which.max(abs(corr.ts.rainf.substemp.df$corr.substemp.rainf))  
  
paste('The best timeshift for Sub Surface Temp with Rainfall is',substemp.timeshift.rainf)

## [1] "The best timeshift for Sub Surface Temp with Rainfall is 1"

paste('The best timeshift for Sub Surface Temp with SAT is',substemp.timeshift.sat)

## [1] "The best timeshift for Sub Surface Temp with SAT is 0"

### 3.2.4: Time Shift of OLR (Okla)

cor(rainf.studyAnom, okla.studyAnom)

## [1] 0.1982311

cor(sat.studyAnom, okla.studyAnom)

## [1] -0.1569565

cors = rep(0,11)  
for (i in -12:0) {  
 df\_plot <- data.frame(cloudanomaly = okla.studyAnom[1:(length(okla.studyAnom)+i)],  
 sat\_anom = sat.studyAnom[(-i+1):length(sat.studyAnom)])  
 cors[i+11]=cor(as.numeric(df\_plot$cloudanomaly),as.numeric(df\_plot$sat\_anom))  
}  
  
okla.timeshift.sat = which.max(abs(cors))  
  
cors2 = rep(0,11)  
for (i in -12:0) {  
 df\_plot <- data.frame(cloudanomaly = okla.studyAnom[1:(length(okla.studyAnom)+i)],  
 rain\_anom = rainf.studyAnom[(-i+1):length(rainf.studyAnom)])  
 cors2[i+11]=cor(as.numeric(df\_plot$cloudanomaly),as.numeric(df\_plot$rain\_anom))  
}  
  
okla.timeshift.rainf = which.max(abs(cors2))  
  
paste('The best timeshift for OLR with Rainfall is',okla.timeshift.rainf)

## [1] "The best timeshift for OLR with Rainfall is 11"

paste('The best timeshift for OLR with SAT is',okla.timeshift.sat)

## [1] "The best timeshift for OLR with SAT is 3"

### 3.2.5: Compilation of Time Shifted Data

maxshift = max(c(wind.timeshift.rainf,  
 wind.timeshift.sat,  
 substemp.timeshift.rainf,  
 okla.timeshift.sat,  
 okla.timeshift.rainf))  
  
n = nrow(enso.df)  
  
rainf.timeshift.df = data.frame(rainf.sg = enso.df$rainf.sg[(1+maxshift):n],  
 sst.pcf = enso.df$sst.pcf[(1+maxshift):n],  
 substemp.pcf = enso.df$substemp.pcf[(1+(maxshift-substemp.timeshift.rainf)):(n-substemp.timeshift.rainf)],  
 okla.pcf = enso.df$okla.pcf[(1+maxshift-okla.timeshift.rainf):(n-okla.timeshift.rainf)],  
 wind.pcf = enso.df$wind.pcf[(1+maxshift-wind.timeshift.rainf):(n-wind.timeshift.rainf)])  
  
   
sat.timeshift.df = data.frame(sat.sg = enso.df$sat.sg[(1+maxshift):n],  
 sst.pcf = enso.df$sst.pcf[(1+maxshift):n],#not timeshifted,  
 substemp.pcf = enso.df$substemp.pcf[(1+maxshift):n],#not timeshifted  
 okla.pcf = enso.df$okla.pcf[(1+maxshift-okla.timeshift.sat):(n-okla.timeshift.sat)],  
 wind.pcf = enso.df$wind.pcf[(1+maxshift-wind.timeshift.sat):(n-wind.timeshift.sat)])

We then compiled our timeshifted data into 2 data.frames, one for timeshifting with Rainfall and the other for timeshifting with SAT.

# 4. Statistical Analysis

Before we began our project, we wanted to validate that ENSO in general has an effect on SG rainfall/SAT. First, we calculated the correlation between the general ENSO index and the SG data.

## 4.1: Relationship between ENSO & SG Rainfall/SAT

#monthly data from 1/1979 to 03/2023  
ensoIndex = read.csv('data/meiv2.csv',sep=",",header=F)  
  
startIndex = (2011-1979)\*12+1; endIndex = (2023-1979)\*12  
  
#i want only 1/1982 to 12/2022  
ensoIndex = ensoIndex[startIndex:endIndex]  
  
ensoRainCor = cor(as.vector(t(ensoIndex)),rainf.studyAnom)  
paste("The correlation between ENSO Index and Rainfall Anomaly: ", ensoRainCor)

## [1] "The correlation between ENSO Index and Rainfall Anomaly: -0.270008827128347"

ensoSatCor = cor(as.vector(t(ensoIndex)),sat.studyAnom)  
paste("The correlation between ENSO Index and SAT Anomaly: ", ensoSatCor)

## [1] "The correlation between ENSO Index and SAT Anomaly: 0.476739655225748"

Absolute correlation between ENSO and Rainfall is relatively low, so it might not be an effective predictor.

Absolute correlation between ENSO and SAT is relatively higher, so it might be an effective predictor.

To further confirm the relationship, we also split our rainfall/SAT anomaly data into 2 categories - ENSO index >=0 vs ENSO index < 0. We will compare the categories, to determine if there is a significant difference between data in the 2 categories.

ensoPos = ensoIndex>=0  
ensoNeg = ensoIndex<0  
  
  
#find enso variable: split sg rainfall and SAT into 2 groups based on enso>0 or enso<0  
  
ensoPos\_sgRainf = rainf.studyAnom[ensoPos]  
ensoNeg\_sgRainf = rainf.studyAnom[ensoNeg]  
  
ensoPos\_sgSAT = sat.studyAnom[ensoPos]  
ensoNeg\_sgSAT = sat.studyAnom[ensoNeg]  
  
#then welch 2-sample test to see if got effect  
var.test(ensoPos\_sgRainf, ensoNeg\_sgRainf)

##   
## F test to compare two variances  
##   
## data: ensoPos\_sgRainf and ensoNeg\_sgRainf  
## F = 0.64558, num df = 45, denom df = 97, p-value = 0.1035  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.3988797 1.0952968  
## sample estimates:  
## ratio of variances   
## 0.6455836

#p-value ~0.5, can be considered same variance?  
t.test(ensoPos\_sgRainf, ensoNeg\_sgRainf,var.equal=TRUE)

##   
## Two Sample t-test  
##   
## data: ensoPos\_sgRainf and ensoNeg\_sgRainf  
## t = -3.4487, df = 142, p-value = 0.0007418  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -87.22321 -23.66236  
## sample estimates:  
## mean of x mean of y   
## -51.062069 4.380718

#then welch 2-sample test to see if got effect  
var.test(ensoPos\_sgSAT, ensoNeg\_sgSAT)

##   
## F test to compare two variances  
##   
## data: ensoPos\_sgSAT and ensoNeg\_sgSAT  
## F = 1.3548, num df = 45, denom df = 97, p-value = 0.2159  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.8370743 2.2985495  
## sample estimates:  
## ratio of variances   
## 1.354798

#p-value ~0.5, can be considered same variance?  
t.test(ensoPos\_sgSAT, ensoNeg\_sgSAT,var.equal=FALSE)

##   
## Welch Two Sample t-test  
##   
## data: ensoPos\_sgSAT and ensoNeg\_sgSAT  
## t = 6.7155, df = 77.28, p-value = 2.821e-09  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.3686748 0.6794419  
## sample estimates:  
## mean of x mean of y   
## 0.7153673 0.1913089

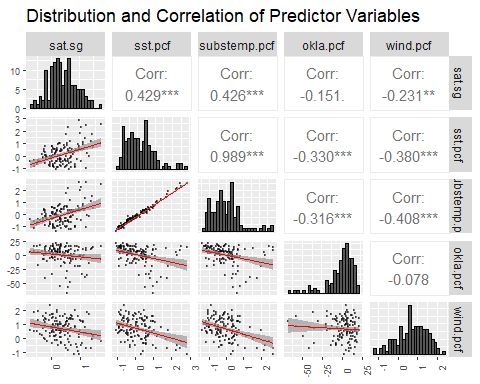
There is no significant difference between the 2 categories for rainfall, which confirms that using ENSO to understand how rainfall changes might not be effective.

There is a significant difference between the 2 categories for SAT, which also confirms that it might be an effective predictor.

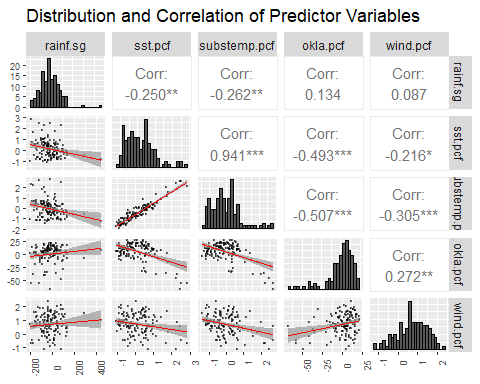
From here, we can go on further to explore exactly how each of the variables used to determine the strength of ENSO affect Singapore’s rainfall/SAT for a more detailed analysis, rather than rely on just the ENSO index. We would expect that the relationship between the Pacific variables and rainfall is weaker, while the relationship between the Pacific variables and SAT is stronger.

## 4.2: Pairs Plot between all variables (After Time Shift)

lowerFn <- function(data, mapping, method = "lm", ...) {  
 p <- ggplot(data = data, mapping = mapping) +  
 geom\_point(colour = "black",size=0.3,alpha=0.5) +  
 geom\_smooth(method = method, color = "red", alpha=0.7,linewidth=0.5,...)  
 p  
}  
  
  
ggpairs(sat.timeshift.df,columns = 1:ncol(sat.timeshift.df),  
 upper = list(continuous = 'cor', combo = "box\_no\_facet"),  
 lower = list(continuous = wrap(lowerFn, method='lm')),  
 diag = list(continuous = wrap('barDiag',color='black',bins=25)),  
 title='Distribution and Correlation of Predictor Variables')+  
 theme(axis.text.x = element\_text(angle=90),  
 axis.text = element\_text(size=7))



ggpairs(rainf.timeshift.df,columns = 1:ncol(rainf.timeshift.df),  
 upper = list(continuous = 'cor', combo = "box\_no\_facet"),  
 lower = list(continuous = wrap(lowerFn, method='lm')),  
 diag = list(continuous = wrap('barDiag',color='black',bins=25)),  
 title='Distribution and Correlation of Predictor Variables')+  
 theme(axis.text.x = element\_text(angle=90),  
 axis.text = element\_text(size=7))



From the pairs plots, none of the pacific variables correlate well with rainfall, while sst.pcf and substemp.pcf show moderate correlation with SAT, supporting our preliminary analysis. We could also consider transforming the variables here so they fit more linearly with rainfall/SAT, which could improve our correlation, but we decided not to for 2 reasons. Firstly, the scatters do not show any clear trends to suggest any transformations. Secondly, we timeshifted the data based on correlation values - if we also wanted to transform based on correlation values, to be rigorous, we would have to calculate correlation values for every type of transformation, to find the ideal for both, but we will leave that to further research.

## 4.3: ANOVA Analysis for decadal data

We wanted to understand if there were patterns on the scale of years that would affect our ability to model the effects of ENSO.

To do so, we decided to split the data into decadal periods, and test for any significant differences in the means and variances using ANOVA and Levene’s test.

#ANOVA  
#all data for 1982-2021 only (40 years).  
all.data <- data.frame(month = time.all,  
 sat.sg=c(sat.refVal,sat.studyVal),  
 rainf.sg=c(rainf.refVal,rainf.studyVal),  
 sst.pcf=c(sst.refVal,sst.studyVal),  
 substemp.pcf=c(substemp.refVal,substemp.studyVal),  
 okla.pcf=c(okla.refVal,okla.studyVal),  
 wind.pcf=c(wind.refVal,wind.studyVal))  
  
all.data <- all.data[1:480,]  
  
all.data$decade <- factor(c(rep(1,120),  
 rep(2,120),  
 rep(3,120),  
 rep(4,120)))  
  
var.sat<-leveneTest(sat.sg~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.sat <- summary(aov(sat.sg ~ decade, data=all.data))  
p.sat <- aov.sat[[1]]$`Pr(>F)`[1]  
  
var.rainf<-leveneTest(rainf.sg~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.rainf <- summary(aov(rainf.sg ~ decade, data=all.data))  
p.rainf <- aov.rainf[[1]]$`Pr(>F)`[1]  
  
var.sst<-leveneTest(sst.pcf~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.sst <- summary(aov(sst.pcf ~ decade, data=all.data))  
p.sst <- aov.sst[[1]]$`Pr(>F)`[1]  
  
var.substemp<-leveneTest(substemp.pcf~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.substemp <- summary(aov(substemp.pcf ~ decade, data=all.data))  
p.substemp <- aov.substemp[[1]]$`Pr(>F)`[1]  
  
var.okla<-leveneTest(okla.pcf~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.okla <- summary(aov(okla.pcf ~ decade, data=all.data))  
p.okla <- aov.okla[[1]]$`Pr(>F)`[1]  
  
var.wind<-leveneTest(wind.pcf~decade,data=all.data) #p > 0.05 so var are same (H0: var is same)  
aov.wind <- summary(aov(wind.pcf ~ decade, data=all.data))  
p.wind <-aov.wind[[1]]$`Pr(>F)`[1]  
  
p.values <- data.frame(p.sat=p.sat,  
 p.rainf=p.rainf,  
 p.sst=p.sst,  
 p.substemp=p.substemp,  
 p.okla=p.okla,  
 p.wind=p.wind)  
  
print(round(p.values,7))

## p.sat p.rainf p.sst p.substemp p.okla p.wind  
## 1 0 0.2730929 0.063421 0.5348572 0.9908606 0

# since p values of sat, sst, wind <0.05, we cannot rej H0. ie the means are different across decadal time scales. Therefore, in order to analyse enso trend, we will need more data to capture the variations better as a sample size of 5 decades are clearly insufficient to fully capture the variations.

From Levene’s test, we find that the variances are similar for all our datasets. Thus, we can conduct ANOVA.

From ANOVA, we find that there are significant differences in the means.

This suggests that there are trends in the predictor and output variables that vary over at least 5 years. (If the longest variations are on short time frames, each decade’s data should be similar to previous decades.) Given that our study period is only 12 years, we might not have enough data to capture these trends.

## 4.4: Modelling to predict SG SAT with ENSO variables

In this section, we attempted to model and predict SG SAT values with ENSO variables. We will use 3 types of models:

* General Linear Model
* Neural Network
* Random Forest

rsquarer = function(obs,pred) {  
 tss = sum((obs-mean(obs))^2)  
 rss = sum((obs-pred)^2)  
 return(1-rss/tss)  
}  
  
satFormula = (sat.sg ~ sst.pcf+substemp.pcf+okla.pcf+wind.pcf)  
  
k = 5  
n = nrow(sat.timeshift.df)  
sat.timeshift.df.random = sat.timeshift.df[sample(n),]  
groups = cut(1:n,k,label=F)  
  
numCores = detectCores()  
cl = makeCluster(numCores)  
clusterExport(cl,c('neuralnet','satFormula','sat.timeshift.df.random','groups','rsquarer','randomForest'))  
results = parSapply(cl, 1:k, function(i) {  
 train = sat.timeshift.df.random[groups!=i,]  
 test = sat.timeshift.df.random[groups==i,]  
   
 train.st = data.frame(scale(train))  
 means = colMeans(train); sds = apply(train,2,sd)  
 test.st = data.frame(scale(test,center=means,scale=sds))  
   
 model = glm(formula=satFormula,data=train.st)  
 pred = predict(model,test.st)  
 linearRMSE = sqrt(sum((pred-test.st$sat.sg)^2)/length(test.st))  
 linearRSQ = rsquarer(test.st$sat.sg,pred)  
   
 net = neuralnet(formula=satFormula,  
 data=train.st,  
 #act.fct='logistic',  
 hidden=c(4),  
 algorithm="sag",  
 stepmax=1e+08,  
 linear.output=FALSE)  
   
 pred = predict(object=net,newdata=test.st)[,1]  
 nnRMSE = sqrt(sum((pred-test.st$sat.sg)^2/length(test.st)))  
 nnRSQ = rsquarer(test.st$sat.sg,pred)  
   
 rf = randomForest(satFormula,data=train.st)  
 pred = predict(rf,newdata=test.st)  
 rfRMSE = sqrt(sum((pred-test.st$sat.sg)^2/length(test.st)))  
 rfRSQ = rsquarer(test.st$sat.sg,pred)  
  
 return(c(linearRMSE,linearRSQ,nnRMSE,nnRSQ,rfRMSE,rfRSQ))  
})  
  
results = data.frame(t(results))  
colnames(results) = c('linearrmse','linearrsq','nnrmse','nnrsq','rfRMSE','rfRSQ')  
colMeans(results)

## linearrmse linearrsq nnrmse nnrsq rfRMSE rfRSQ   
## 2.11595629 0.10542842 2.25333794 -0.01579568 2.12579363 0.10365733

Based on our RMSE and Rsquared values, we can safely conclude that the linear models, neural networks, and random forest models all perform extremely poorly. We acknolwedge that we might not have enough data to fit a proper model, or perhaps there are other climatic factors that we did not take into account.

# 5. Conclusion and Discussion

We use 4 ENSO variables, Sea Surface Temp (SST), Sub Surface Temp (substemp), Wind Speed, OLR, and tried to analysis and possibly predict the effects on Singapore climate, specifically, the local Rainfall and Surface Air Temp (SAT). We find that the ENSO variables were poor likely poor in predicting Rainfall data, but could perhaps be useful in modelling SAT values. However, the modelling attempts showed us that we lack additional information or other forcing climatic factors to make a meaningful model.

The data sourcing and cleaning was split equally among the 3 group members.

Nicholas and Zhi Yi contributed in the data analysis portion

Ancel contributed to the data description