```r
library("readxl")
library("lubridate")
library("proto")
library("gsubfn")
library("RSQLite")
library("sqldf")
library("tidyverse")
library("ggplot2")
library("mice")
library("caret")
library("dplyr")
library("reshape2")
library("janitor")
library("tidyr")
library("MASS")
library("car")
library("Metrics")
library("glmnet")
library("xgboost")
library("gbm")
library("mboost")
library("elasticnet")
library("tseries")
library("zoo")
library("forecast")
library("chron")
library("RColorBrewer")
library("lattice")
library("ncdf4")
library("fpp")
library("Rmisc")
```

```
#NASA
nasa  <- read.csv("./Assignment//Team/Assignment 1//Dataset//nasa_clean.csv")

summary(nasa)
head(nasa)

#changing all to numeric
nasa$Date <- as.Date(nasa$Date,  format = "%Y-%m-%d")
class(nasa$Date )
nasa$Temp_Diff <- as.numeric(as.character(nasa$Temp_Diff))
#nasa$Avg_temp <- as.numeric(as.character(nasa$Avg_temp))
nasa <- drop_na(nasa)

#adding Temp_Diff to baseline (14)
nasa$Avg_Temp <- 14 + nasa$Temp_Diff
#nasa$Avg_Temp <- 14 + nasa$Avg_temp
nasa = nasa[,!(names(nasa) %in% c("Temp_Diff"))]
#nasa = nasa[,!(names(nasa) %in% c("Avg_temp"))]
#nasa = nasa[,!(names(nasa) %in% c("X"))]
nasa$Avg_Temp<-as.numeric(nasa$Avg_Temp)
#frequency: 1=annual, 4=quartly, 12=monthly
#creating timeseries
#save a numeric vector of 2040 observations (from 1850 jan - 2020 feb)
nasa <- nasa[order(as.Date(nasa$Date, format="%Y-%m-%d")),]
nasats <- ts(nasa$Avg_Temp, start=c(1880,1), end=c(2020,03), frequency=12)

tail(nasats)
plot(nasats)

#plot various decompositions into error/noise, trend and seasonality
```

```
fit <- decompose(nasats, type="multiplicative") #decompose using "classical" method, multiplicative form

plot(fit)


fit <- decompose(nasats, type="additive") #decompose using "classical" method, additive form

plot(fit)


#gives you the confident interval: there is a trend, high confidence interval for seasonality

fit <- stl(nasats, t.window=12, s.window="periodic") #decompose using STL (Season and trend using Loess)

plot(fit)



#creating a test and train

#Split the data into train and test sets at "2007-01-01"

train_tbl <- nasa %>% filter(nasa$Date < ymd("2007-01-01"))

test_tbl  <- nasa %>% filter(nasa$Date >= ymd("2007-01-01"))

train_tbl <- train_tbl %>% filter(train_tbl$Date >= ymd("1940-01-01"))

#to order by date, not necessary, but good to know

#train_tbl <- train_tbl[order(as.Date(train_tbl$Date, format="%Y-%m-%d")),]

#test_tbl <- test_tbl[order(as.Date(test_tbl$Date, format="%Y-%m-%d")),]


tail(train_tbl)


#creating timeseries for the data

traints <- ts(train_tbl$Avg_Temp, start=c(1940,1), end=c(2006,12), frequency=12)

testts <- ts(test_tbl$Avg_Temp,  start=c(2007,1), end=c(2020,3), frequency=12)


tail(traints)


#Qn 1 & 2: Forecast the global average temperatures & 90% confidence intervals
```

Create exponential smoothing models: additive vs multiplicative noise (first A vs M), additive vs multiplicative trend (second A vs M) and no seasonality vs automatic detection (third N vs Z) trend and no seasonlity (AAN), multiplicative (MMN)

nasa_AAN <- ets(traints, model="AAN", damped=TRUE)

nasa_MMN <- ets(traints, model="MMN", damped=TRUE)

nasa_AAA <- ets(traints, model="AAA", damped=TRUE)

nasa_MMM <- ets(traints, model="MMM", damped=TRUE)

nasa_MAZ <- ets(traints, model="MAZ", damped=TRUE)

nasa_AAN

nasa_MMN

nasa_AAA

nasa_MMM

nasa_MAZ

Create their prediction "cones" for 159 months (2007-01-01 to 2020-03-01) into the future with quintile confidence intervals

nasa_AAN_pred <- forecast(nasa_AAN, h=159, level=c(0.8, 0.95))

nasa_MMN_pred <- forecast(nasa_MMN, h=159, level=c(0.8, 0.95))

nasa_AAA_pred <- forecast(nasa_AAA, h=159, level=c(0.8, 0.95))

nasa_MMM_pred <- forecast(nasa_MMM, h=159, level=c(0.8, 0.95))

nasa_MAZ_pred <- forecast(nasa_MAZ, h=159, level=c(0.8, 0.95))

#check mape of prediction on test dataset

#https://stackoverflow.com/questions/43048018/accuracy-function

print(accuracy(nasa_AAN_pred,testts)) #RMSE: 0.1934915, MAPE: 1.0244427

print(accuracy(nasa_MMN_pred,testts)) #RMSE: 0.1941361, MAPE: 1.0259993

print(accuracy(nasa_AAA_pred,testts)) #RMSE: 0.1906236, MAPE: 1.0182961

print(accuracy(nasa_MMM_pred,testts)) #RMSE: 0.191294, MAPE: 1.0168532

print(accuracy(nasa_MAZ_pred,testts)) #RMSE: 0.1946198, MAPE: 1.0282960

#Create a trigonometric box-cox autoregressive trend seasonality (TBATS) model

```
nasa_tbats <- tbats(traints)
nasa_tbats

#forecast for 159 months (2007-01-01 to 2020-03-01)
nasa_tbats_pred <-forecast(nasa_tbats, h=159, level=c(0.8, 0.95))
par(mfrow=c(1,1))
plot(nasa_tbats_pred, xlab="Time", ylab="Predicted Global Avg Temp")

#check accuracy with test dataset
print(accuracy(nasa_tbats_pred,testts)) #RMSE: 0.2325858, MAPE: 1.2098155

#Creating ARIMA model
fit <- auto.arima(traints,seasonal=TRUE)
fit

#forecast for 159 months (2007-01-01 to 2020-03-01)
nasa_arima_pred <- forecast(fit, h=159, level=c(0.8, 0.95))
plot(nasa_arima_pred, xlab="Time", ylab="Predicted Global Avg Temp")

#check accuracy with test dataset
print(accuracy(nasa_arima_pred,testts)) #RMSE: 0.2305829, MAPE: 1.1754087

#Creating ARIMA model
fit <- auto.arima(traints,seasonal=FALSE)
fit

#forecast for 159 months (2007-01-01 to 2020-03-01)
nasa_arima_pred <- forecast(fit, h=159, level=c(0.8, 0.95))
plot(nasa_arima_pred, xlab="Time", ylab="Predicted Global Avg Temp")

#check accuracy with test dataset
```

```
print(accuracy(nasa_arima_pred,testts)) #RMSE: 0.2305829, MAPE: 1.1754087 #worst model also omg


#####

##### ARIMA with regressors

#####

month <- as.numeric(format(train_tbl$Date, "%m"))


monthMatrix <- cbind(Months=model.matrix(~as.factor(month))) # Create dummies for each day-of-week

monthMatrix <- monthMatrix[,-1]# Remove "intercept" (12th month) dummy

colnames(monthMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_regressors <- monthMatrix

nasatrain_arima <- auto.arima(traints, xreg=matrix_of_regressors,seasonal=TRUE, approximation = FALSE) # Train a model

nasatrain_arima # See what it is


#forecast for 159 months (2007-01-01 to 2020-03-01)

monthTestPred <- 1

monthTestPred <- rbind(monthTestPred, 2)

monthTestPred <- rbind(monthTestPred, 3)

for (i in 1:12){

 for(j in 1:13)

 {

   monthTestPred <- rbind(monthTestPred, i)

   }}

monthTestPred

#remove first row

#monthPred <- monthPred[-1,]

#making the matrix
```

```
monthTestPredMatrix <- cbind(Months=model.matrix(~as.factor(monthTestPred))) # Create dummies for each month of year

monthTestPredMatrix

monthTestPredMatrix <- monthTestPredMatrix[,-1]# Remove "intercept" dummy

colnames(monthTestPredMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_test_pred_regressors <- monthTestPredMatrix


nasatest_arima_pred <- forecast(nasatrain_arima, h=159, xreg = matrix_of_test_pred_regressors, level=c(0.8, 0.95))

plot(nasatest_arima_pred, xlab="Time", ylab="Predicted Avg Temp")

print(accuracy(nasatest_arima_pred,testts)) #RMSE: 0.1897208, MAPE: 0.9724306 #best so far



#####

##### Forecast predictions using ARIMA on regressors for 969 months (2020 March to 2100 December)

#####

#month Matrix on full NASA dataset

month <- as.numeric(format(nasa$Date, "%m"))


monthMatrix <- cbind(Months=model.matrix(~as.factor(month))) # Create dummies for each month of year

monthMatrix <- monthMatrix[,-1]# Remove "intercept" dummy

colnames(monthMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_regressors <- monthMatrix

nasatemp_arima <- auto.arima(nasats, xreg=matrix_of_regressors,seasonal=TRUE, approximation = FALSE) # Train a model

nasatemp_arima # See what it is
```

```
#forecast for 969 months (2020-04-01 to 2100-12-01)

monthPred <- 4

monthPred <- rbind(monthPred, 5)

monthPred <- rbind(monthPred, 6)

monthPred <- rbind(monthPred, 7)

monthPred <- rbind(monthPred, 8)

monthPred <- rbind(monthPred, 9)

monthPred <- rbind(monthPred, 10)

monthPred <- rbind(monthPred, 11)

monthPred <- rbind(monthPred, 12)

for (i in 1:12){

 for(j in 1:80)

 {

  monthPred <- rbind(monthPred, i)

 }}

monthPred

#remove first row

#monthPred <- monthPred[-1,]

#making the matrix

monthPredMatrix <- cbind(Months=model.matrix(~as.factor(monthPred))) # Create dummies for each month of year

monthPredMatrix

monthPredMatrix <- monthPredMatrix[,-1]# Remove "intercept" dummy

colnames(monthPredMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_pred_regressors <- monthPredMatrix


nasatemp_arima_pred <- forecast(nasatemp_arima, h=969, xreg = matrix_of_pred_regressors, level=c(0.8, 0.90))

plot(nasatemp_arima_pred, xlab="Time", ylab="Predicted Avg Temp")
```

```
write.csv(nasatemp_arima_pred, "NASA ARIMA Predictions.csv", row.names = FALSE, na = "")

#Qn 6: constant temperature prediction per Mr. Armstrong
#getting 2006 dataset
data_2006 <- nasa %>% filter(nasa$Date < ymd("2007-01-01"))
data_2006  <- data_2006 %>% filter(data_2006$Date >= ymd("2006-01-01"))
data_2006

#getting the average temp of 2016
avg_temp_2006<- mean(data_2006$Avg_Temp)
avg_temp_2006

#Create dataset for 2007-01-01 to 2017-12-01 with constant average temp of 2016
constant_temp <- test_tbl %>% filter(test_tbl$Date <= ymd("2017-12-01"))
constant_temp$Avg_Temp <- avg_temp_2006
constant_temp

#Create dataset for 2007-01-01 to 2017-12-01 for NASA dataset
test_tbl_actual <- test_tbl %>% filter(test_tbl$Date <= ymd("2017-12-01"))

#creating timeseries for the data
head(train_tbl)
traints <- ts(train_tbl$Avg_Temp, start=c(1940,1), end=c(2006,12), frequency=12) #train
constantts <- ts(constant_temp$Avg_Temp,  start=c(2007,1), end=c(2017,12), frequency=12)
#"actual"
testts_act <- ts(test_tbl_actual$Avg_Temp,  start=c(2007,1), end=c(2017,12), frequency=12) #actual
NASA

plot(traints)
plot(constantts)
plot(testts_act)
```

```
#####

##### ARIMA with regressors [train]

#####

month <- as.numeric(format(train_tbl$Date, "%m"))


monthMatrix <- cbind(Months=model.matrix(~as.factor(month))) # Create dummies for each month of year

monthMatrix <- monthMatrix[,-1]# Remove "intercept" (12th month) dummy

colnames(monthMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_regressors <- monthMatrix

nasatrain_arima <- auto.arima(traints, xreg=matrix_of_regressors,seasonal=TRUE, approximation = FALSE) # Train a model

nasatrain_arima # See what it is


#forecast for 132 months (2007-01-01 to 2017-12-01)

month_constant_temp <- as.numeric(format(test_tbl_actual$Date, "%m"))


monthMatrix_constant_temp <- cbind(Months=model.matrix(~as.factor(month_constant_temp))) # Create dummies for each day-of-week

monthMatrix_constant_temp <- monthMatrix_constant_temp[,-1]# Remove "intercept" (12th month) dummy

colnames(monthMatrix_constant_temp) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns


matrix_of_regressors_constant_temp <- monthMatrix_constant_temp


nasa_arima_pred <- forecast(nasatrain_arima, h=132, xreg = matrix_of_regressors_constant_temp, level=c(0.8, 0.95))

print(accuracy(nasa_arima_pred,testts_act)) #predicted vs actual

print(accuracy(constantts,testts_act)) #constant vs actual
```

```
par(mfrow=c(1,1))

plot(nasa_arima_pred,xlim=c(2007,2017), ylim=c(14.3, 15.4),  main='NASA - Actual temperature against forecasted and constant')

par(new=TRUE)

plot(testts_act, ylab='Avg Global Monthly Temp', xlim=c(2007,2017), ylim=c(14.3, 15.4), main='NASA - Actual temperature against forecasted and constant')

abline(constant_temp$Avg_Temp,0, col="red")


#Qn7: same analyses with another starting time and long-enough time-interval

#Split the data into train and test sets at "1996-01-01"

train_tbl <- nasa %>% filter(nasa$Date < ymd("1996-01-01"))

test_tbl  <- nasa %>% filter(nasa$Date >= ymd("1996-01-01"))

train_tbl <- train_tbl %>% filter(train_tbl$Date >= ymd("1970-01-01"))

test_tbl  <- test_tbl %>% filter(test_tbl$Date <= ymd("2005-12-01"))

#getting 1995 dataset

data_1995  <- train_tbl %>% filter(train_tbl$Date >= ymd("1995-01-01"))

data_1995


#getting the average temp of 1996

data_1995<- mean(data_1995$Avg_Temp)

data_1995

#Create dataset for 1996-01-01 to 2005-12-01 with constant average temp of 2016

constant_temp <- test_tbl

constant_temp$Avg_Temp <- data_1995

constant_temp


#creating timeseries for the data

head(train_tbl)

traints <- ts(train_tbl$Avg_Temp, start=c(1970,1), end=c(1995,12), frequency=12) #train

constantts <- ts(constant_temp$Avg_Temp,  start=c(1996,1), end=c(2005,12), frequency=12) #constant
```

testts_act <- ts(test_tbl_actual$Avg_Temp, start=c(1996,1), end=c(2005,12), frequency=12) #actual NASA

#####

ARIMA with regressors [train]

#####

month <- as.numeric(format(train_tbl$Date, "%m"))

monthMatrix <- cbind(Months=model.matrix(~as.factor(month))) # Create dummies for each month of year

monthMatrix <- monthMatrix[,-1]# Remove "intercept" (12th month) dummy

colnames(monthMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns

matrix_of_regressors <- monthMatrix

nasatrain_arima <- auto.arima(traints, xreg=matrix_of_regressors,seasonal=TRUE, approximation = FALSE) # Train a model

nasatrain_arima # See what it is

#forecast for 120 months (1996-01-01 to 2005-12-01)

month_constant_temp <- as.numeric(format(test_tbl$Date, "%m"))

monthMatrix_constant_temp <- cbind(Months=model.matrix(~as.factor(month_constant_temp))) # Create dummies for each day-of-week

monthMatrix_constant_temp <- monthMatrix_constant_temp[,-1]# Remove "intercept" (12th month) dummy

colnames(monthMatrix_constant_temp) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") # Rename columns

matrix_of_regressors_constant_temp <- monthMatrix_constant_temp

nasa_arima_pred <- forecast(nasatrain_arima, h=120, xreg = matrix_of_regressors_constant_temp, level=c(0.8, 0.95))

print(accuracy(nasa_arima_pred,testts_act)) #predicted vs actual

```
print(accuracy(constantts,testts_act)) #constant vs actual


par(mfrow=c(1,1))

plot(nasa_arima_pred,xlim=c(1996.32,2005), ylim=c(14.10, 15.4),  main='NASA - Actual temperature against forecasted and constant')

par(new=TRUE)

plot(testts_act, ylab='Avg Global Monthly Temp', xlim=c(1996.32,2005), ylim=c(14.10, 15.4), main='NASA - Actual temperature against forecasted and constant')

abline(constant_temp$Avg_Temp,0, col="red")
```