```
#Purpose: Build predictive time series model to evaluate the topic of global warming and climate change

library(ggplot2)
library(dplyr)
library(tidyr)
library(stringr)
library(lubridate)
library(sqldf)
library(readxl)
library(car)
library(estimatr)
library(caret)
library(janitor)
library(glmnet)
library(geosphere)
library(esquisse)
library(MLmetrics)
library(gridExtra)
library(forecast)
library(fpp)
library(vars)
library(MLmetrics)

#import datasets
MET_data <- read.csv("Queen's MMA\\MMA 867\\Assignment 2\\MET_HadCRUT4_Data.csv",
header=TRUE, sep = ",")
NASA_data <- read.csv("Queen's MMA\\MMA 867\\Assignment 2\\NASA.csv", header=TRUE, sep = ",")

#convert NASA_data into tidy dataset
NASA_data <- gather(NASA_data, Month, Median_Temp_Difference, `Jan`:`Dec`)

NASA_data <- NASA_data %>%
  arrange(Year)

NASA_data <- tibble::rowid_to_column(NASA_data, "X")

#realign MET_data row id
MET_data <- MET_data %>%
  dplyr::select(-X)

MET_data <- tibble::rowid_to_column(MET_data, "X")

#explore summary statistics
summary(NASA_data)
```

```
str(NASA_data)

summary(MET_data)
str(MET_data)

#change necessary data types
NASA_data$Median_Temp_Difference <- as.numeric(NASA_data$Median_Temp_Difference)
NASA_data$J.D <- as.numeric(NASA_data$J.D)
NASA_data$D.N <- as.numeric(NASA_data$D.N)
NASA_data$DJF <- as.numeric(NASA_data$DJF)
NASA_data$MAM <- as.numeric(NASA_data$MAM)
NASA_data$JJA <- as.numeric(NASA_data$JJA)
NASA_data$SON <- as.numeric(NASA_data$SON)
NASA_data$Month <- match(NASA_data$Month, month.abb)

#**NOTE**: Median temperature in both datasets is 14 degrees celsius; explore and standardize
baselines
#change temperature_difference field to temperature by adding 14 deg celsius baseline
MET_data <- MET_data %>%
  mutate(Temperature = Median_Temp_Difference + 14) %>%
  dplyr::select(-Median_Temp_Difference)

#Start time series analysis at 1940 (this is when the upwards trend occurs; post industrial revolution)
MET_data <- MET_data %>%
  filter(Year > 1939)

###################     QUESTION 1,2 MET DATA     ###############

## MET_data ##
#fit time series model for MET_data
MET_ts <- ts(MET_data$Temperature, start=1940, frequency=12) # ts function defines the dataset as
timeseries starting Jan 2004 and having seasonality of frequency 12 (monthly)

#plot the ts model
plot(MET_ts)

#decompose the ts model
MET_fit <- stl(MET_ts, t.window=12, s.window="periodic") #decompose using STL (Season and trend
using Loess)
plot(MET_fit)

#Split the data into train and test sets at about 2007
MET_train <- MET_data %>%
  filter(MET_data$Year < 2007)
```

```
MET_test  <- MET_data %>%
  filter(MET_data$Year >= 2007)

MET_train_ts <- ts(MET_train$Temperature, start=1940, frequency=12)
MET_test_ts <- ts(MET_test$Temperature,  start=2007, frequency=12)

#test ETS models
MET_AAN <- ets(MET_train_ts, model="AAN", damped=TRUE)
MET_MMN <- ets(MET_train_ts, model="MMN", damped=TRUE)
MET_AAA <- ets(MET_train_ts, model="AAA", damped=TRUE)
MET_MMM <- ets(MET_train_ts, model="MMM", damped=TRUE)

#examine model stats
MET_AAN #AIC = 1689
MET_MMN #AIC = 1695
MET_AAA #AIC = 1691
MET_MMM #AIC = 1694

#create their prediction "cones" for 158 months (covering test set) into the future with quintile
confidence intervals
MET_AAN_pred <- forecast(MET_AAN, h=158, level=c(0.8, 0.9))
MET_MMN_pred <- forecast(MET_MMN, h=158, level=c(0.8, 0.9))
MET_AAA_pred <- forecast(MET_AAA, h=158, level=c(0.8, 0.9))
MET_MMM_pred <- forecast(MET_MMM, h=158, level=c(0.8, 0.9))

#compare the prediction "cones" visually
par(mfrow=c(1,4))
plot(MET_AAN_pred, xlab="Year", ylab="Predicted Global Avg Temp")
plot(MET_MMN_pred, xlab="Year", ylab="Predicted Global Avg Temp")
plot(MET_AAA_pred, xlab="Year", ylab="Predicted Global Avg Temp")
plot(MET_MMM_pred, xlab="Year", ylab="Predicted Global Avg Temp")

#check accuracy
print(RMSLE(MET_AAN_pred$mean,MET_test_ts)) #RMSLE = 0.01081 #most accurate of ETS models
print(RMSLE(MET_MMN_pred$mean,MET_test_ts)) #RMSLE = 0.01094
print(RMSLE(MET_AAA_pred$mean,MET_test_ts)) #RMSLE = 0.01112
print(RMSLE(MET_MMM_pred$mean,MET_test_ts)) #RMSLE = 0.01103

f_MET_AAN  <- function(y, h) forecast(ets(y, model="AAN"), h = h)
errors_MET_AAN <- tsCV(MET_test_ts, f_MET_AAN, h=1)

f_MET_MMN  <- function(y, h) forecast(ets(y, model="MMN"), h = h)
errors_MET_MMN <- tsCV(MET_test_ts, f_MET_MMN, h=1)
```

```
f_MET_AAA  <- function(y, h) forecast(ets(y, model="AAA"), h = h)
errors_MET_AAA <- tsCV(MET_test_ts, f_MET_AAA, h=1)

f_MET_MMM  <- function(y, h) forecast(ets(y, model="MMM"), h = h)
errors_MET_MMM <- tsCV(MET_test_ts, f_MET_MMM, h=1)

par(mfrow=c(1,1))
plot(errors_MET_AAN, ylab='tsCV errors')
abline(0,0)
lines(errors_MET_MMN, col="red")
lines(errors_MET_AAA, col="green")
lines(errors_MET_MMM, col="blue")
legend("left", legend=c("CV_error_AAN", "CV_error_MMN","CV_error_AAA","CV_error_MMM"),
col=c("black", "red", "green", "blue"), lty=1:4)

mean(abs(errors_MET_AAN/MET_test_ts), na.rm=TRUE)*100 #0.5164
mean(abs(errors_MET_MMN/MET_test_ts), na.rm=TRUE)*100 #1.1150
mean(abs(errors_MET_AAA/MET_test_ts), na.rm=TRUE)*100 #0.5640
mean(abs(errors_MET_MMM/MET_test_ts), na.rm=TRUE)*100 #1.1903

#-----

#test TBATS model (want to predict through year 2100...h = 970 months)
#formulate model
MET_tbats <- tbats(MET_train_ts)
MET_tbats #AIC = 1660

#predict using model
MET_tbats_pred <- forecast(MET_tbats, h=158, level=c(0.8, 0.9))
plot(MET_tbats_pred, xlab="Year", ylab="Predicted Median_Temp_Difference")

#evaluate tbats model predictions against test set
print(RMSLE(MET_tbats_pred$mean,MET_test_ts)) #RMSLE = 0.01192 #ehh

#cross-validate model
fMET_tbats  <- function(y, h) forecast(tbats(y), h = h)
errors_MET_tbats <- tsCV(MET_test_ts, fMET_tbats, h=1)

par(mfrow=c(1,1))
plot(errors_MET_tbats, ylab='tsCV errors')
abline(0,0)
lines(errors_MET_MMN, col="red")
lines(errors_MET_AAA, col="green")
```

```
lines(errors_MET_MMM, col="blue")
legend("left", legend=c("CV_error_TBATS", "CV_error_MMN","CV_error_AAA","CV_error_MMM"),
col=c("black", "red", "green", "blue"), lty=1:4)

mean(abs(errors_MET_tbats/MET_ts), na.rm=TRUE)*100 #0.5322

#-----

#test ARIMA models
#auto-correlation function
Acf(MET_ts,main="") # data "as is"
Acf(log(MET_ts),main="") # log-transformed data
Acf(diff(log(MET_ts),12),main="") # difference-12 log data
#Observations: The autocorrelations for the differences

#partial auto-correlation function
par(mfrow=c(1,2))
Acf(diff(log(MET_ts),12),main="")
Pacf(diff(log(MET_ts),12),main="")
#Observations: Significance???

#define arima models
MET_arima1 <- auto.arima(MET_train_ts,seasonal=FALSE) #try first assuming no seasonality...this is
certainly doubtful
MET_arima1 #AIC = -1435 significantly lower/better than ETS and TBATS attempts

MET_arima2 <- auto.arima(MET_train_ts,seasonal=TRUE) #now try assuming seasonality (we expect this
is the case on an annual basis)
MET_arima2 #AIC = -1442...stronger than the arima with no seasonality (for AIC we care about absolute
lowest value)

#predict using arima model
MET_arima1_pred <- forecast(MET_arima1, h=158, level=c(0.8, 0.9))
plot(MET_arima1_pred, xlab="Time", ylab="Predicted Global Avg Temp")

MET_arima2_pred <- forecast(MET_arima2, h=158, level=c(0.8, 0.9))
plot(MET_arima2_pred, xlab="Time", ylab="Predicted Global Avg Temp")

#check accuracy
print(RMSLE(MET_arima1_pred$mean,MET_test_ts)) #NO SEASONALITY --> RMSLE = 0.00952...new
best...use this!!

print(RMSLE(MET_arima2_pred$mean,MET_test_ts)) #WITH SEASONALITY --> RMSLE = 0.00961...
```

```
#preview what the forecast to 2021 would look like
par(mfrow=c(1,1))
Acf(residuals(MET_arima2))
plot(forecast(MET_arima2,970)) # 970 months to get to year 2100

#test ARIMA with regressors (dynamic regression)

#create dummies for each month
#monthMatrix <- cbind(Month=model.matrix(~as.factor(MET_data$Month)))
#remove "intercept" (7th day) dummy
#monthMatrix <- monthMatrix[,-1]
#colnames(monthMatrix) <- c("Feb","Mar","Apr","May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec") #
Rename columns

#matrix_of_regressors <- monthMatrix

#build the model
#MET_regarima <- auto.arima(MET_data$Temperature, xreg=matrix_of_regressors)
#MET_regarima #AIC = -2846...a bit stronger than vanilla arima

#xreg.pred<-matrix_of_regressors[-c(1345:5664),] # Build a 2-weeks-out prediction matrix

#MET_regarima_pred <- forecast(MET_regarima, h=970, xreg = xreg.pred, level=c(0.8, 0.90))
#plot(MET_regarima_pred, xlab="Time", ylab="Predicted Global Avg Temp")


#test ARIMA on residuals
METlm_msts <- tslm(MET_train_ts ~ trend + season) # Build a linear model for trend and seasonality
summary(METlm_msts)
METlm_msts

residarima1 <- auto.arima(METlm_msts$residuals) # Build ARIMA on it's residuals
residarima1 #AIC = -1465
residualsArimaForecast <- forecast(residarima1, h=158, level=c(0.8, 0.9)) #forecast from ARIMA
residualsF <- as.numeric(residualsArimaForecast$mean)

regressionForecast <- forecast(METlm_msts,h=158, level=c(0.8, 0.9)) #forecast from lm
regressionF <- as.numeric(regressionForecast$mean)

forecastR <- regressionF+residualsF # Total prediction

print(RMSLE(forecastR,MET_test_ts)) #RMSLE = 0.00967

#plot
```

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

##SELECT TOP MODELS AND TO USE FOR PREDICTION OF THE UNKNOWN (to year 2100)

#include top 2 ETS? TBATS, ARIMA
MET_MMN <- ets(MET_ts, model="MMN", damped=TRUE)
MET_AAA <- ets(MET_ts, model="AAA", damped=TRUE)
MET_tbats <- tbats(MET_ts)
MET_arima2 <- auto.arima(MET_ts,seasonal=FALSE) #notice that the seasonal component has been
removed

#create their prediction "cones" for 970 months (up to year 2100) into the future with quintile
confidence intervals
MET_MMN_pred <- forecast(MET_MMN, h=970, level=c(0.8, 0.9))
MET_AAA_pred <- forecast(MET_AAA, h=970, level=c(0.8, 0.9))
MET_TBATS_pred <- forecast(MET_tbats, h=970, level=c(0.8, 0.9))
MET_arima2_pred <- forecast(MET_arima2, h=970, level=c(0.8, 0.9))

#compare the prediction "cones" visually
par(mfrow=c(1,4)) # This command sets the plot window to show 1 row of 4 plots
plot(MET_MMN_pred, xlab="Year", ylab="Predicted Temperature")
plot(MET_AAA_pred, xlab="Year", ylab="Predicted Temperature")
plot(MET_TBATS_pred, xlab="Year", ylab="Predicted Temperature")
plot(MET_arima2_pred, xlab="Year", ylab="Predicted Temperature")

#lets look at what our models actually are
MET_MMN #AIC = 2177
MET_AAA #AIC = 2174
MET_tbats #AIC = 2137
MET_arima2 #AIC = -1736...noticeably better than the ETS and TBATS models head to head

#export predictions form arima(2,1,1) model
write.csv(MET_arima2_pred, file = paste0("Queen's MMA\\MMA 867\\Assignment 2\\MET ARIMA
Predictions.csv"), row.names = FALSE, na = "")

##################    QUESTION 6 MET DATA    ###############

#update the test dataset to only go through 2017
MET_test2 <- MET_test %>%
  filter(Year < 2018)

MET_test2_ts <- ts(MET_test2$Temperature,  start=2007, frequency=12)

#repeat our winning ARIMA predictions on the new test set
```

```
MET_arima3 <- auto.arima(MET_train_ts,seasonal=FALSE)

MET_arima3_pred <- forecast(MET_arima3, h=132, level=c(0.8, 0.9)) #h is now 132 as we are only going
through 2017

plot(MET_arima3_pred, xlab="Year", ylab="Predicted Temperature")

print(RMSLE(MET_arima3_pred$mean, MET_test2_ts)) #RMSLE = 0.00962

print(accuracy(MET_arima3_pred, MET_test2_ts))

#evaluate Armstrong's constant prediction
subset2006 <- MET_train %>%
  filter(Year == 2006)

mean(subset2006$Temperature)

armstrong_constant <- MET_test2 %>%
  mutate(Temperature = 14.50575)

armstrong_ts <- ts(armstrong_constant$Temperature,  start=2007, frequency=12)

print(RMSLE(armstrong_ts, MET_test2_ts)) #RMSLE = 0.0111

print(accuracy(armstrong_ts, MET_test2_ts))

#plot the 3 to compare visibly
par(mfrow=c(1,1))
plot(MET_arima3_pred, main="MET - Actual temperature against forecasted and constant",
col="blue",xlim=c(2007,2017), ylim=c(14.1, 15.1))
par(new=TRUE)
plot(MET_test2_ts, ylab='Average Global Temperature', xlim=c(2007,2017), ylim=c(14.1, 15.1))
par(new=TRUE)
plot(armstrong_ts, ylab='', col = "red",xlim=c(2007,2017), ylim=c(14.1,15.1))
legend("topleft", legend=c("ARIMA Forecast", "Actual", "Constant Forecast"), col=c("blue", "black",
"red"), lty=1:3)

###################     QUESTION 7 MET DATA     ###############

#resplit the dataset
MET_train3 <- MET_data %>%
  filter(Year > 1969) %>%
  filter(Year < 1996)
```

```
MET_test3 <- MET_data %>%
  filter(Year > 1995) %>%
  filter(Year < 2006)

MET_train3_ts <- ts(MET_train3$Temperature, start=1970, frequency=12)
MET_test3_ts <- ts(MET_test3$Temperature,  start=1996, frequency=12)

#repeat our winning ARIMA predictions on the new test set
MET_arima4 <- auto.arima(MET_train3_ts,seasonal=TRUE)

MET_arima4_pred <- forecast(MET_arima4, h=120, level=c(0.8, 0.9)) #h is now 120

plot(MET_arima4_pred, xlab="Year", ylab="Predicted Temperature")

print(RMSLE(MET_arima4_pred$mean, MET_test3_ts)) #RMSLE = 0.01538

print(accuracy(MET_arima4_pred, MET_test3_ts))

#evaluate Armstrong's constant prediction
subset1995 <- MET_train %>%
  filter(Year == 1995)

mean(subset1995$Temperature)

armstrong_constant2 <- MET_test3 %>%
  mutate(Temperature = 14.32517)

armstrong2_ts <- ts(armstrong_constant2$Temperature,  start=1996, frequency=12)

print(RMSLE(armstrong2_ts, MET_test3_ts)) #RMSLE = 0.01119

print(accuracy(armstrong2_ts, MET_test3_ts))

#plot the 3 to compare visibly
par(mfrow=c(1,1))
plot(MET_arima4_pred, main="MET - Actual temperature against forecasted and constant",
col="blue",xlim=c(1996.32, 2005), ylim=c(14, 15))
par(new=TRUE)
plot(MET_test3_ts, ylab='Average Global Temperature', xlim=c(1996.32,2005), ylim=c(14, 15))
par(new=TRUE)
plot(armstrong2_ts, ylab='', col = "red",xlim=c(1996.32,2005), ylim=c(14,15))
legend("topleft", legend=c("ARIMA Forecast", "Actual", "Constant Forecast"), col=c("blue", "black",
"red"), lty=1:3)
```