**Computer Lab: Time Series Data**

**Exercise 1: Regression-based Model & Smoothing Methods**

# install.packages('forecast')

# install.packages("zoo")

# Read data

data **<-** read.csv**(**"Amtrak.csv"**)**

str**(**data**)**

head**(**data**)**

# Convert data into time series object in R

library**(**forecast**)**

# start: the time of the first observation

# frequency: number of times per year

x **<-** ts**(**data**$**Ridership, start**=**c**(**1991,1**)**,frequency **=** 12**)**

x

plot**(**x**)**

############################################################################

# Model 1: Linear Trend Model

Amtrack.lm **<-** tslm**(**x**~**trend**)**

summary**(**Amtrack.lm**)**

# Data partition for time series data

# Use the last 36 months data as the training dataset

nValid **<-** 36

nTrain **<-** length**(**x**)-**nValid

train.ts **<-** window**(**x,start**=**c**(**1991,1**)**,end**=**c**(**1991,nTrain**))**

valid.ts **<-** window**(**x,start**=**c**(**1991,nTrain**+**1**)**,end**=**c**(**1991,nTrain**+**nValid**))**

train.lm **<-** tslm**(**train.ts**~**trend**)**

summary**(**train.lm**)**

train.lm.pred **<-** forecast**(**train.lm,h**=**nValid,level**=**0**)**

# Visualize the linear trend model

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

plot**(**train.lm.pred, ylim **=** c**(**1300, 2600**)**, ylab **=** "Ridership", xlab **=** "Time",

bty **=** "l", xaxt **=** "n", xlim **=** c**(**1991,2006**)**,main **=** "", flty **=** 2**)**

axis**(**1, at **=** seq**(**1991, 2006, 1**)**, labels **=** format**(**seq**(**1991, 2006, 1**)))**

lines**(**train.lm.pred**$**fitted, lwd **=** 2, col **=** "blue"**)**

lines**(**valid.ts**)**

# Evaluate model performance

accuracy**(**train.lm.pred,valid.ts**)**

# Polynomial Trend

train.lm.poly.trend **<-** tslm**(**train.ts **~** trend **+** I**(**trend**^**2**))**

summary**(**train.lm.poly.trend**)**

train.lm.poly.trend.pred **<-** forecast**(**train.lm.poly.trend, h **=** nValid, level **=** 0**)**

accuracy**(**train.lm.poly.trend.pred,valid.ts**)**

# A model with seasonality

# In R, function tslm() uses ts() which automatically creates the categorical Season column (called season) and converts it into dummy variables.

train.lm.season **<-** tslm**(**train.ts **~** season**)**

summary**(**train.lm.season**)**

train.lm.season.pred **<-** forecast**(**train.lm.season, h **=** nValid, level **=** 0**)**

accuracy**(**train.lm.season.pred,valid.ts**)**

# A model with trend and seasonality

train.lm.trend.season **<-** tslm**(**train.ts **~** trend **+** I**(**trend**^**2**)** **+** season**)**

summary**(**train.lm.trend.season**)**

train.lm.trend.season.pred **<-** forecast**(**train.lm.trend.season, h **=** nValid, level **=** 0**)**

accuracy**(**train.lm.trend.season.pred,valid.ts**)**

###########################################################################

# Model 2: Simple Moving Average

library**(**zoo**)**

ma **<-** rollmean**(**x,k**=**12,align**=**"right"**)**

summary**(**ma**)**

# Observe the difference between forecasted ma vs original data x

ma

x

# Calculate MAPE

MAPE **=** mean**(**abs**((**ma**-**x**)/**x**)**,na.rm**=**T**)**

##########################################################################

# run simple exponential smoothing

# and alpha = 0.2 to fit simple exponential smoothing.

ses **<-** ses**(**train.ts, alpha **=** 0.2, h**=**36**)**

autoplot**(**ses**)**

accuracy**(**ses,valid.ts**)**

# Use ses function to estimate alpha

ses1 **<-** ses**(**train.ts, alpha **=** **NULL**, h**=**36**)**

summary**(**ses1**)**

accuracy**(**ses1,valid.ts**)**

**Exercise 2: ARIMA Models**

# install.packages('ts')

# install.packages('forecast')

# Read data

data **<-** read.csv**(**'Tractor-Sales.csv'**)**

head**(**data**)**

str**(**data**)**

# Convert data into time series object in R

library**(**forecast**)**

# start: the time of the first observation

# frequency: number of times per year

x **<-** ts**(**data**[**,2**]**,start**=**c**(**2003,1**)**,frequency**=**12**)**

x

# Observe the data: homscedasticity?

# Increasing variances over time

plot**(**x**)**

# log transformation to achieve homoscedasticity

z**<-** log10**(**x**)**

plot**(**z**)**

# Observe the data: stationary?

# Increasing mean over time

# The data has a trend, let's take the difference

y **<-** diff**(**z**)**

plot**(**y**)**

# Is the data randome walk?

# Use Phillips-Perron Unit Root Test to check if the data is random walk

# If p-value is sifnificant, reject the null hypothesis (i.e., data is not random walk)

PP.test**(**y**)**

# ACF test for White Noise

# ACF shows correlation between y\_t and lagged terms y\_(t-h)

# The figure suggests seasonal lagged autocorrelation

acf**(**y,main**=**"ACF Tractor Sales"**)**

# Use auto.arima function in the package "forecast"

# Apply auto.arima to data without differencing

library**(**forecast**)**

ARIMAfit **<-** auto.arima**(**z, approximation**=FALSE**,trace**=TRUE)**

summary**(**ARIMAfit**)**

# Use the best ARIMA model to forecast future scales

pred **<-** predict**(**ARIMAfit,n.ahead**=**36**)**

pred

# Plot the data

# Remember initial log-transformation?

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

plot**(**x,type**=**'l',xlim**=**c**(**2004,2018**)**,ylim**=**c**(**1,1600**)**,xlab **=** 'Year',ylab **=** 'Tractor Sales'**)**

lines**(**10**^(**pred**$**pred**)**,col**=**'blue'**)**

lines**(**10**^(**pred**$**pred**+**2**\***pred**$**se**)**,col**=**'orange'**)**

lines**(**10**^(**pred**$**pred**-**2**\***pred**$**se**)**,col**=**'orange'**)**