# R Code Appendix

```
library(bsts)
library(lubridate)
library(bsts)
library(dplyr)
library(ggplot2)
library(randtests)
library(qqtest)
library(knitr)
library(changepoint)

data <-read.csv("C:/Users/julia18/Desktop/tourism.csv")
attach(data)
ts_tourism <-ts(data$value, start=1986, frequency =4) #time series object of all
observations
plot(ts_tourism, ylab="Spending per million", main="Tourism Spending Per Million in
Canada")

#split into training and testing
train.data <-ts(data$value[1:104], start=c(1986,1), end=c(2011,4),frequency =
4)#training data set
test.data <-ts(data$value[105:127], start=c(2012,1), frequency = 4)#testing data set

#plot of all data, training, and testing data
par(mfcol=c(1,2))
plot(ts_tourism, xlab="Year", ylab="Millions of Canadian Dollars ")
plot(ts_tourism, xlab="Year", ylab="Millions of Canadian Dollars ")
lines(train.data, col="blue")
lines(test.data, col="red")

#change points plots
par(mfcol=c(1,2))
plot(ts_tourism, xlab="Year", ylab="Millions of Canadian Dollars")
points(decompose(ts_tourism, type="mult")$trend, type="l", col="red", lwd=2)
plot(ts_tourism, xlab="Year", ylab="Millions of Canadian Dollars")
abline(v=2003.25, lty=2, col="blue")
abline(v=2008.75, lty=2, col="blue")

#verify change points
mean_cp=cpt.mean(data$value)
plot(mean_cp,cpt.col='blue')
print(mean_cp)

holt.data <- HoltWinters(train.data, seasonal="mult")
par(mfcol=c(1,1))
```

```r
plot(holt.data)

#residuals for holts winter method
holt.data.fitted <- holt.data$fitted[,1]
holt.data.res <- train.data[4:100] - holt.data.fitted

residualdiagnostics_holt <- function(residual){
 par(mfcol=c(2,2))
 plot(residual, main="Residuals vs. time", type="l")
 abline(h=0, lty=2)
 qqnorm(residual)
 qqline(residual)
 acf(residual, main="SACF")
 acf(residual, type="partial", main="SPACF")
}
residualdiagnostics_holt(holt.data.res)

predict.holt <- predict(holt.data, n.ahead=23, prediction.interval=TRUE)
par(mfcol=c(1,1))
plot(test.data, ylim=c(10000, 35000))
tn <- time(test.data)

points(tn, test.data)
points(tn, predict.holt[,1], type="l", col="red")
points(tn, predict.holt[,2], type="l", col="orange")
points(tn, predict.holt[,3], type="l", col="orange")

#testing normality assumption
shapiro.test(holt.data.res)
#MAPE
MAPE_holt <-mean(abs(test.data-predict.holt[,1])/test.data) #0.041127*100=4.11

# Regression method
train.data.log <- log(train.data)# Stabilize the variance
par(mfcol=c(1,2))
plot(train.data)
plot(train.data.log)

# classical decomposition model
t <- time(train.data)
season <- as.factor(cycle(train.data))
data.reg <- lm(train.data.log ~ t + season)
par(mfcol=c(1,1))
plot(train.data.log)
points(t, data.reg$fitted, col="red", type="l")
```

```
#add change points
changept1 <- ts(as.numeric(t >= 2003.25), start=c(1986,1), frequency=4)
changept2 <- ts(as.numeric(t >= 2008.75), start=c(1986,1), frequency=4)
reg.changept <- lm(train.data.log ~ t + season + changept1 + changept2)
par(mfcol=c(1,1))
plot(train.data.log)

par(mfcol=c(1,2))
plot(reg.changept$res, type="l", ylim=c(-0.1,0.1)) # not stationary
abline(h=0, lty=2)
diff.res.reg = diff(reg.changept$res) # take one ordinary difference
plot(diff.res.reg, type="l", ylim=c(-0.1,0.1)) # looks stationary
abline(h=0, lty=2)

PP.test(diff.res.reg)# Test for stationarity
# Dickey-Fuller = -12.18, Truncation lag parameter = 4, p-value = 0.01

# SACF and SPACF
par(mfcol=c(1,2))
acf(diff.res.reg, main="SACF", lag.max=20)
acf(diff.res.reg, type="partial", main="SPACF", lag.max=20)

# Fit (0,1,2)x(1,0,1)_4 to the residuals
reg.sarima <- model.matrix(reg.changept)[,2:7]
reg.changept2 <- arima(train.data.log, order=c(0,1,2), seasonal=list(order=c(1,0,1),
period=4), xreg=reg.sarima)
reg.cp2fit <- train.data.log - reg.changept2$residuals
par(mfcol=c(1,1))
plot(train.data.log)
points(t, reg.cp2fit, col="red", type="l")

# residual diagnostics
residualdiagnostics_holt(reg.changept2$residuals)

# Predictions with SARIMA residuals
sarima.reg <- reg.sarima[77:97,]
sarima.reg[,1] <- sarima.reg[,1]+6
sarima.reg[,6] <- rep(1,21)
predictionreg.changept2 <- predict(reg.changept2, n.ahead=23, newxreg=sarima.reg,
interval="pred")
par(mfcol=c(1,1))
plot(test.data, ylim=c(12000, 40000))
points(tn, test.data)
points(tn, exp(predictionreg.changept2$pred), type="l", col="red")
points(tn, exp(predictionreg.changept2$pred + 1.96*predictionreg.changept2$se),
type="l", col="orange")
```

```
points(tn, exp(predictionreg.changept2$pred - 1.96*predictionreg.changept2$se),
type="l", col="orange")
MAPE_reg<-mean(abs(test.data-
exp(predictionreg.changept2$pred))/test.data)#0.06835469*100=6.83

#box-jenkins approach:sarima models
par(mfrow=c(1,1))
data.diff <- diff(diff(train.data.log,lag=4))
par(mfcol=c(1,2))
plot(train.data.log)
plot(data.diff)
PP.test(data.diff)

#SACF and SPACF
par(mfcol=c(1,2))
acf(data.diff, lag.max=24, main="SACF")
acf(data.diff,type="partial", lag.max=24, main="SPACF")


sarima1<-arima(train.data.log,order=c(0,1,0),seasonal=list(order=c(0,1,1),period=4))
sarima2<-arima(train.data.log,order=c(0,1,0),seasonal=list(order=c(1,1,1),period=4))
sarima3<-arima(train.data.log,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
sarima4<-arima(train.data.log,order=c(0,1,1),seasonal=list(order=c(1,1,1),period=4))
sarima5<-arima(train.data.log,order=c(1,1,0),seasonal=list(order=c(0,1,1),period=4))
sarima6<-arima(train.data.log,order=c(1,1,0),seasonal=list(order=c(1,1,0),period=4))
sarima7<-arima(train.data.log,order=c(0,1,0),seasonal=list(order=c(1,1,0),period=4))
sarima8<-arima(train.data.log,order=c(0,1,1),seasonal=list(order=c(1,1,0),period=4))
sarima9<-arima(train.data.log,order=c(1,1,0),seasonal=list(order=c(1,1,1),period=4))
sarima10<-arima(train.data.log,order=c(0,1,1),seasonal=list(order=c(1,1,1),period=4))

#sarima1$aic

# Fitted plot
par(mfcol=c(1,1))
plot(train.data)
points(t, exp(train.data.log - sarima1$res), col="red", type="l")
length(exp(train.data.log - sarima1$res))

# Residual diagnostics
residualdiagnostics_holt(sarima1$res)

#predictions
pred.sarima <- predict(sarima1, n.ahead=23, interval="pred")
par(mfcol=c(1,1))
plot(test.data, ylim=c(10000, 45000))
points(tn, test.data)
```

```
points(tn, exp(pred.sarima$pred), type="l", col="red")
points(tn, exp(pred.sarima$pred+1.96*pred.sarima$se), type="l", col="orange")
points(tn, exp(pred.sarima$pred-1.96*pred.sarima$se), type="l", col="orange")
MAPE.sarima <- mean(abs(test.data-
exp(pred.sarima$pred))/test.data)#0.026242*100=2.62


#bayesian structure model
Y <- window(ts_tourism, end=c(2010,4)) #using 2011-2017 as testing
y <- log10(Y)

#bsts model
ss <- AddLocalLinearTrend(list(), y)
ss <- AddSeasonal(ss, y, nseasons = 4)
model <- bsts(y, state.specification = ss, niter = 500, ping=0, seed=2016)

#burn-ins
burn <- SuggestBurn(0.1, model)

#Predict
p <- predict.bsts(model, horizon = 27, burn = burn, quantiles = c(.025,.975))#predicting
27 periods into the future

df1 <- data.frame(c(10^as.numeric(-colMeans(model$one.step.prediction.errors[-
(1:burn),])+y),
    10^as.numeric(p$mean)),as.numeric(ts_tourism),as.Date(time(ts_tourism)))
names(df1) <- c("Fitted", "Actual", "Date")

MAPE_bsts <- filter(df1, year(Date)>2010) %>%
summarise(MAPE_bsts=mean(abs(Actual-Fitted)/Actual))

# 95% forecast credible interval
posterior.interval <- cbind.data.frame(
  10^as.numeric(p$interval[1,]),
  10^as.numeric(p$interval[2,]),
  subset(df1, year(Date)>2010)$Date)
names(posterior.interval) <- c("LL", "UL", "Date")

df2 <- left_join(df1, posterior.interval, by="Date")

ggplot(data=df2, aes(x=Date)) +
  geom_line(aes(y=Actual, colour = "Actual"), size=1.2) +
  geom_line(aes(y=Fitted, colour = "Fitted"), size=1.2, linetype=2) +
  theme_bw() + theme(legend.title = element_blank()) + ylab("") + xlab("") +
  geom_vline(xintercept=as.numeric(as.Date("2010-12-01")), linetype=2) +
  geom_ribbon(aes(ymin=LL, ymax=UL), fill="grey", alpha=0.5) +
```

```
  theme(axis.text.x=element_text(angle = -90, hjust = 0))

pred1 <- predict(model, horizon = 48)
Date <- as.Date(as.yearmon(2011 + seq(0, 47)/4))
p2 <- data.frame(c( 10^as.numeric(pred1$mean)),Date)
names(p2) <- c("Predict", "Date")
posterior.interval <- cbind.data.frame(
  10^as.numeric(pred1$interval[1,]),
  10^as.numeric(pred1$interval[2,]),
  Date)
names(posterior.interval) <- c("LL", "UL", "Date")
p3 <- left_join(p2, posterior.interval, by="Date")

ggplot(data=p3, aes(x=Date)) +
  geom_line(aes(y=Predict, colour = "predict"), size=1.2)+ theme_bw() +
theme(legend.title = element_blank()) + ylab("Millions of Canadian Dollars") +
xlab("Years") +
  geom_vline(xintercept=as.numeric(as.Date("2010-12-01")), linetype=2) +
  geom_ribbon(aes(ymin=LL, ymax=UL), fill="grey", alpha=0.5) +
  ggtitle(paste0("                              Forecast for next 5 years ")) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0))

plot(model, y = c("state", "components", "residuals",
"coefficients", "prediction.errors","forecast.distribution",
"predictors", "size", "dynamic", "seasonal", "help"))
PlotBstsResiduals(model)
PlotBstsComponents(model)
PlotBstsForecastDistribution(model)
PlotSeasonalEffect(model)
plot(model)
PlotBstsForecastDistribution(model)
```

# References Appendix


[1] https://tiac-aitc.ca/advocacy_global.htm
[2]http://www.statcan.gc.ca/eng/subjects/travel_and_tourism/tourism_indicators
[3] Scott, Steven, and Hal Varian. 2014. “Predicting the Present with Bayesian Structural Time Series”
[4] http://www.unofficialgoogledatascience.com/2017/07/fitting-bayesian-structural-time-series.html
[5] http://sisifospage.tech/2017-10-30-forecasting-bsts.html