Appendix: Source Code

General Functions Code

```
data <- read_csv("sales_revenue_clean.csv")

totalSales <- ts(data$Tot_Sales,start=c(1990,1),frequency=12)
totalRev <- ts(data$Tot_Revenue,start=c(1990,1),frequency=12)
totalPrice <- ts(data$Tot_Price,start=c(1990,1),frequency=12)


rSales <- ts(data$R_Sales,start=c(1990,1),frequency=12)
rRev <- ts(data$R_Revenue,start=c(1990,1),frequency=12)
rPrice <- ts(data$R_Price,start=c(1990,1),frequency=12)

cSales <- ts(data$C_Sales,start=c(1990,1),frequency=12)
cRev <- ts(data$C_Revenue,start=c(1990,1),frequency=12)
cPrice <- ts(data$C_Price,start=c(1990,1),frequency=12)

iSales <- ts(data$I_Sales,start=c(1990,1),frequency=12)
iRev <- ts(data$I_Revenue,start=c(1990,1),frequency=12)
iPrice <- ts(data$I_Price,start=c(1990,1),frequency=12)

tSales <- ts(data$T_Sales,start=c(1990,1),frequency=12)
tRev <- ts(data$T_Revenue,start=c(1990,1),frequency=12)
tPrice <- ts(data$T_Price,start=c(1990,1),frequency=12)

oSales <- ts(data$O_Sales,start=c(1990,1),frequency=12)
oRev <- ts(data$O_Revenue,start=c(1990,1),frequency=12)
oPrice <- ts(data$O_Price,start=c(1990,1),frequency=12)

plot_months <- function(series,  main = "title", ylab = "value", xlab = "Time") {
  Month=c('J','F','M','A','M','J','J','A','S','O','N','D')

  plot(series, main = main, ylab = ylab, xlab = xlab)
  points(series,pch=Month)
}

acf_pacf <- function(series, name = "Series"){
  acf. = acf(series, plot = FALSE)
  plot(acf., main = paste(name, "ACF"))
  pacf. = pacf(series, plot = FALSE)
  plot(pacf., main = paste(name, "PACF"))
}

stationary_test <- function(series){
  acf(series)
  pacf(series)
```

```
  print(runs(series))
  print(shapiro.test(series))
}

Month=c('J','F','M','A','M','J','J','A','S','O','N','D')

plot(totalSales, main = "Total Sales")
points(totalSales,pch=Month)
```

```
plot(totalPrice, main = "Total Price")
points(totalPrice,pch=Month)
```

```
plot(totalRev, main = "Total Revenue")
points(totalRev,pch=Month)
```

```
totalMA = filter(totalRev,rep(1,13)/13,sides=2) # 1 year MA
rMA = filter(rRev,rep(1,13)/13,sides=2) # 1 year MA
cMA = filter(cRev,rep(1,13)/13,sides=2) # 1 year MA
iMA = filter(iRev,rep(1,13)/13,sides=2) # 1 year MA

plot(cbind(totalMA/1e6,rMA/1e6,cMA/1e6, iMA/1e6),plot.type='single',
     col=4:1,lwd=2, ylab="sales (Millions of $)",
     main = "Electricity Sales (12 month MA) by Sector")

legend(x = 'topleft',
       legend = c('Total', 'Residential', 'Commercial', 'Industrial'),
       col = 4:1, lwd = 2)
```

Total Revenue

Data Processing

```
totalRev2 = window(totalRev/1000000, start = c(2008, 1))


# hold actual data for later predictions
actual = window(totalRev2, start = c(2019,1))
totalRev2 = window(totalRev2, start = c(2008,1), end = c(2018,12))

plot_months(totalRev2, main = "Total Revenue (2008-2019)",
            ylab = "Revenue (Millions)", xlab = "Year")
```

```
#acf and pacf
acf_pacf(totalRev2, "Total Revenue")
```

Model Specification

```
#boxcox and resulting transformation

bc = BoxCox.ar(totalRev2)
```

```
logRev = log(totalRev2)
logActual = log(actual)
```

```
plot_months(logRev, main = "Transformed Revenue (2008-2019)",
            ylab = "Log Revenue", xlab = "Year")
```

```
acf(logRev)
```

```
pacf(logRev)
```

```
diffRev = diff(logRev)
plot_months(diffRev, main = "First Difference of Log Revenue",
            ylab = "First Difference of Log Revenue", xlab = "Year")
```

```
acf(diffRev, main = "First Difference ACF")
```

```
plot_months(diffRev, main = "Second Diff of Log Revenue",
            ylab = "Second Difference of Log Revenue", xlab = "Year")
```

```
acf(diff(diffRev), main = "Second Diff ACF")
```

```
seasDiffRev = diff(diffRev, lag = 12)
plot_months(seasDiffRev, main = "First Non-Seasonal and Seasonal Difference of Log Revenue",
            ylab = "First Non-Seasonal and Seasonal Difference", xlab = "Year")
```

```
acf(seasDiffRev, main = "ACF")
```

```
pacf(seasDiffRev)
```

```
eacf(seasDiffRev, ar.max = 12)
plot(armasubsets(diffRev, nar = 12, nma = 12))
```

Diagnostics

```
model1 = arima(logRev, order = c(0,1,1), seasonal = list(order=c(0,1,2),period= 12))
model2 = arima(logRev, order = c(0,1,0), seasonal = list(order=c(0,1,1),period= 12))
model3 = arima(logRev, order = c(0,1,1), seasonal = list(order=c(0,1,2),period= 12))
model4 = arima(logRev, order = c(0,1,2), seasonal = list(order=c(1,1,1),period= 12))

res1 = rstandard(model1)
res2 = rstandard(model2)
res3 = rstandard(model3)
res4 = rstandard(model4)

# repeat for all residuals
runs(res1)
hist(res1)
```

```
qqnorm(res1)
qqline(res1)
```

```
shapiro.test(res1)
LB.test(model1)


plot(fitted(model4), col = 'red', main = "Plotting Log Revenue With Model 4",
     ylab = "Log Revenue", xlab = "Years")
lines(logRev, pch = 3)
```

```
model <- model4
res = rstandard(model)
plot_months(res, main = "Standardized Residuals of Model 4", ylab = "Residuals")
```

```
hist(res, main = "Histogram of Standardized Residuals")
```

Forecasting

```
num = length(actual)

res = plot(model,n.ahead=num, n1 = c(2016,2), transform = exp, col = "red",
           main = "ARIMA(0,1,2)x(1,1,1)_12 Forecast", ylab ="Total Revenue", xlab = "Year")
lines(actual,pch=3)
```

```
res = plot(model,n.ahead=36, transform = exp, col = "red", main = "Beyond 2020 Forecast",
           ylab ="Total Revenue", xlab = "Year")
```

Commercial Data

Data Processing

```
cRev <- ts(data$C_Revenue/1000000,start=c(1990,1),frequency=12)
plot_months(cRev, main = "Commercial Revenue (1990-2020)",
            ylab = "Revenue (Millions)", xlab = "Year")
```

```
cRev2 = window(cRev, start = c(2008, 1))

# hold actual data for later predictions
cRev_held = window(cRev2, start = c(2019,1))
cRev2 = window(cRev2, start = c(2008,1), end = c(2018,12))

plot_months(cRev2, main = "Commercial Revenue (2008-2018)",
            ylab = "Revenue (Millions)", xlab = "Year")
```

```
acf_pacf(cRev2, "Commercial Revenue")
```

```
bc = BoxCox.ar(cRev2, lambda = seq(-0.5, 1.5, 0.1))
```

```
cRev_diff01 = diff(cRev2, lag = 12)
cRev_diff10 = diff(cRev2)
cRev_diff11 = diff(cRev_diff01)
plot(cRev_diff10, main = "First Difference", ylab = "")
```

```
plot(cRev_diff01, main = "Seasonal First Difference", ylab = "")
```

```
plot(cRev_diff11, main = "First Difference and Seasonal First Difference",
     ylab = "")
```

Model Specification

```
acf_pacf(cRev_diff01, name = "Seasonal Difference of Commercial Revenue")
```

```
acf_pacf(cRev_diff11, name = "Seasonal and Non-Seasonal Difference of Commercial Revenue")
```

```r
plot(armasubsets(cRev_diff01, nar = 12, nma = 12))
```

```r
plot(armasubsets(cRev_diff11, nar = 12, nma = 12))
```

```r
monthsTime_reg = matrix(nrow = length(cRev2), ncol = 12)
months. = season(cRev2)
for (i in 1:11){
  monthsTime_reg[,i] = as.numeric(months. == months.[i])
}
monthsTime_reg[,12] = time(cRev2) - 2008
monthsTime_residuals = lm(cRev2 ~ monthsTime_reg)$residuals
plot(armasubsets(monthsTime_residuals, nar = 7, nma = 10))
```

Diagnostics

```r
cRev_model1 = arima(cRev2, order = c(1,0,0), seasonal = list(order=c(0,1,0),period= 12))
cRev_model2 = arima(cRev2, order = c(2,0,0), seasonal = list(order=c(1,1,0),period= 12))
cRev_model3 = arima(cRev2, order = c(1,0,0), seasonal = list(order=c(0,1,1),period= 12))
cRev_model4 = arima(cRev2, order = c(0,1,1), seasonal = list(order=c(1,1,0),period= 12))
cRev_model5 = arima(cRev2, order = c(2, 0, 0), xreg = monthsTime_reg)

runs_tests = vector(mode = "numeric", length = 5)
runs_tests[1] = runs(cRev_model1$residuals)$pvalue
runs_tests[2] = runs(cRev_model2$residuals)$pvalue
runs_tests[3] = runs(cRev_model3$residuals)$pvalue
runs_tests[4] = runs(cRev_model4$residuals)$pvalue
runs_tests[5] = runs(cRev_model5$residuals)$pvalue
print(runs_tests)

shapiro_tests = vector(mode = "numeric", length = 5)
shapiro_tests[1] = shapiro.test(cRev_model1$residuals)$p.value
shapiro_tests[2] = shapiro.test(cRev_model2$residuals)$p.value
shapiro_tests[3] = shapiro.test(cRev_model3$residuals)$p.value
shapiro_tests[4] = shapiro.test(cRev_model4$residuals)$p.value
shapiro_tests[5] = shapiro.test(cRev_model5$residuals)$p.value
print(shapiro_tests)

LB_tests = vector(mode = "numeric", length = 5)
LB_tests[1] = LB.test(cRev_model1)$p.value
LB_tests[2] = LB.test(cRev_model2)$p.value
LB_tests[3] = LB.test(cRev_model3)$p.value
LB_tests[4] = LB.test(cRev_model4)$p.value
LB_tests[5] = LB.test(cRev_model5)$p.value
print(LB_tests)

AIC_df = AIC(cRev_model1, cRev_model2, cRev_model3, cRev_model4, cRev_model5)
BIC_df = BIC(cRev_model1, cRev_model2, cRev_model3, cRev_model4, cRev_model5)
print(AIC_df$AIC)
print(BIC_df$BIC)

hist(cRev_model5$residuals)
```

```r
qqnorm(rstandard(cRev_model5), main = "Normal Q-Q Plot for Model 5")
qqline(rstandard(cRev_model5))
```

Forecasting

```
num = length(cRev_held)

model5forecast_reg = matrix(nrow = num, ncol = 12)
months_forecast. = season(cRev_held)

for (i in 1:11){
  model5forecast_reg[,i] = as.numeric(months_forecast. == months_forecast.[i])
}
model5forecast_reg[,12] = time(cRev_held) - 2008

plot(cRev_model5,n.ahead=num, n1 = c(2017,1), newxreg = model5forecast_reg,
     col = "red", main = "Model 3 Forecast", ylab ="Revenue (Millions $)", xlab = "Year", add = TRUE)
lines(cRev_held,pch=3)
```