---
title: "03_ACF_PACF_Plots"
author: "Callum Weinberg"
date: "December 1, 2021"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Libraries

```{r}
library(dplyr, warn.conflicts = FALSE) #Using
library(tidyr) #Using
library(knitr) #Using
library(lubridate, warn.conflicts = FALSE) #Using
library(ggplot2) #Using
library(MASS) #Uncertain
library(qpcR) #Using
library(forecast) #Using
library(cowplot) #Using
library(TSA) #Using
```

Seasonal (Lag 12) Only Differenced Data

```{r}
load(file="Data/landings_transformed_season_only.Rdata")
```

```{r}
## Sample ACF
sample_acf_list = acf(landings_transformed_season_only$pounds_transformed, plot = FALSE, lag.max = 60)

# Put into Dataframe
sample_acf = as.data.frame(do.call(cbind, sample_acf_list))

# Confidence Interval Line
conf.level = 0.95
ciline = qnorm((1 - conf.level)/2)/sqrt(length(landings_transformed_season_only$pounds_transformed))

# Plot
ACF_Sample_Graph = ggplot(data = sample_acf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "ACF") +
  #labs(title = "Sample Autocorrelation Function\nfor De-Trended De-Seasoned Red Sea Urchin Landings\nMonthly for 2008-
2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 12),
    axis.text.x = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))


## Sample PACF
sample_pacf_list = pacf(landings_transformed_season_only$pounds_transformed, plot = FALSE, lag.max = 60)

# Put into Dataframe
sample_pacf = as.data.frame(do.call(cbind, sample_pacf_list))

# Plot
PACF_Sample_Graph = ggplot(data = sample_pacf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "Partial ACF") +
  #labs(title = "Sample Partial Autocorrelation Function\nfor De-Trended, De-Seasoned Red Sea Urchin Landings\nMonthly
for 2008-2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
```

```
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))

## Plots for Report
png(filename = "Images/acf_pacf.png", width = 960, height = 480)
plot_grid(ACF_Sample_Graph,PACF_Sample_Graph, labels = NULL, label_size = 12, ncol = 2, nrow = 1)
dev.off()
```


Seasonal AND Trend-Differenced Data

NOT USED IN REPORT
```{r}
load(file="Data/landings_transformed.Rdata")
```

```{r}
## Sample ACF
sample_acf_list = acf(landings_transformed$pounds_transformed, plot = FALSE, lag.max = 100)

# Put into Dataframe
sample_acf = as.data.frame(do.call(cbind, sample_acf_list))

# Confidence Interval Line
conf.level = 0.95
ciline = qnorm((1 - conf.level)/2)/sqrt(length(landings_transformed$pounds_transformed))

# Plot
ACF_Sample_Graph = ggplot(data = sample_acf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "ACF",
       title = "Sample Autocorrelation Function\nfor De-Trended De-Seasoned Red Sea Urchin Landings\nMonthly for 2008-
2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))


## Sample PACF
sample_pacf_list = pacf(landings_transformed$pounds_transformed, plot = FALSE, lag.max = 100)

# Put into Dataframe
sample_pacf = as.data.frame(do.call(cbind, sample_pacf_list))

# Plot
PACF_Sample_Graph = ggplot(data = sample_pacf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "Partial ACF",
       title = "Sample Partial Autocorrelation Function\nfor De-Trended, De-Seasoned Red Sea Urchin Landings\nMonthly
for 2008-2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))

## Plot Both
ACF_Sample_Graph
PACF_Sample_Graph
```

Option 3: Trend-Only Differenced Data

NOT USED IN THE REPORT
```{r}
load(file="Data/landings_transformed_no_season.Rdata")
```

NOT USED IN REPORT
```{r}
## Sample ACF
sample_acf_list = acf(landings_transformed_no_season$pounds_transformed, plot = FALSE, lag.max = 100)

# Put into Dataframe
sample_acf = as.data.frame(do.call(cbind, sample_acf_list))

# Confidence Interval Line
conf.level = 0.95
ciline = qnorm((1 - conf.level)/2)/sqrt(length(landings_transformed_no_season$pounds_transformed))

# Plot
ACF_Sample_Graph = ggplot(data = sample_acf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "ACF",
       title = "Sample Autocorrelation Function\nfor De-Trended De-Seasoned Red Sea Urchin Landings\nMonthly for 2008-
2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))


## Sample PACF
sample_pacf_list = pacf(landings_transformed_no_season$pounds_transformed, plot = FALSE, lag.max = 100)

# Put into Dataframe
sample_pacf = as.data.frame(do.call(cbind, sample_pacf_list))

# Plot
PACF_Sample_Graph = ggplot(data = sample_pacf, mapping = aes(x = as.numeric(lag), y = as.numeric(acf))) +
  geom_hline(aes(yintercept = 0)) +
  geom_segment(mapping = aes(xend = as.numeric(lag), yend = 0)) +
  geom_hline(aes(yintercept = ciline), linetype = 2, color = 'darkblue') +
  geom_hline(aes(yintercept = -ciline), linetype = 2, color = 'darkblue') +
  labs(x = "lag", y = "Partial ACF",
       title = "Sample Partial Autocorrelation Function\nfor De-Trended, De-Seasoned Red Sea Urchin Landings\nMonthly
for 2008-2018") +
  theme(text = element_text(size = 20),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.key.width=unit(1,"cm"),
    axis.text.y = element_text(angle=90, hjust=1, size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title=element_text(size=10,face="bold"))

## Plot Both
ACF_Sample_Graph
PACF_Sample_Graph
```