01_Clean_Data_Plot

Callum Weinberg

November 26, 2021

Libraries

```
library(dplyr, warn.conflicts = FALSE) #Using
library(tidyr) #Using
library(knitr) #Using
library(lubridate, warn.conflicts = FALSE) #Using
library(ggplot2) #Using
library(MASS) #Uncertain
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(qpcR) #Using
```

```
## Loading required package: minpack.lm
```

```
## Loading required package: rgl
```

```
## This build of rgl does not include OpenGL functions.  Use
##  rglwidget() to display results, e.g. via options(rgl.printRglwidget = TRUE).
```

```
## Loading required package: robustbase
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(forecast) #Using
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(cowplot) #Using
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
```

```
##
##     stamp
```

```
library(TSA) #Using
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

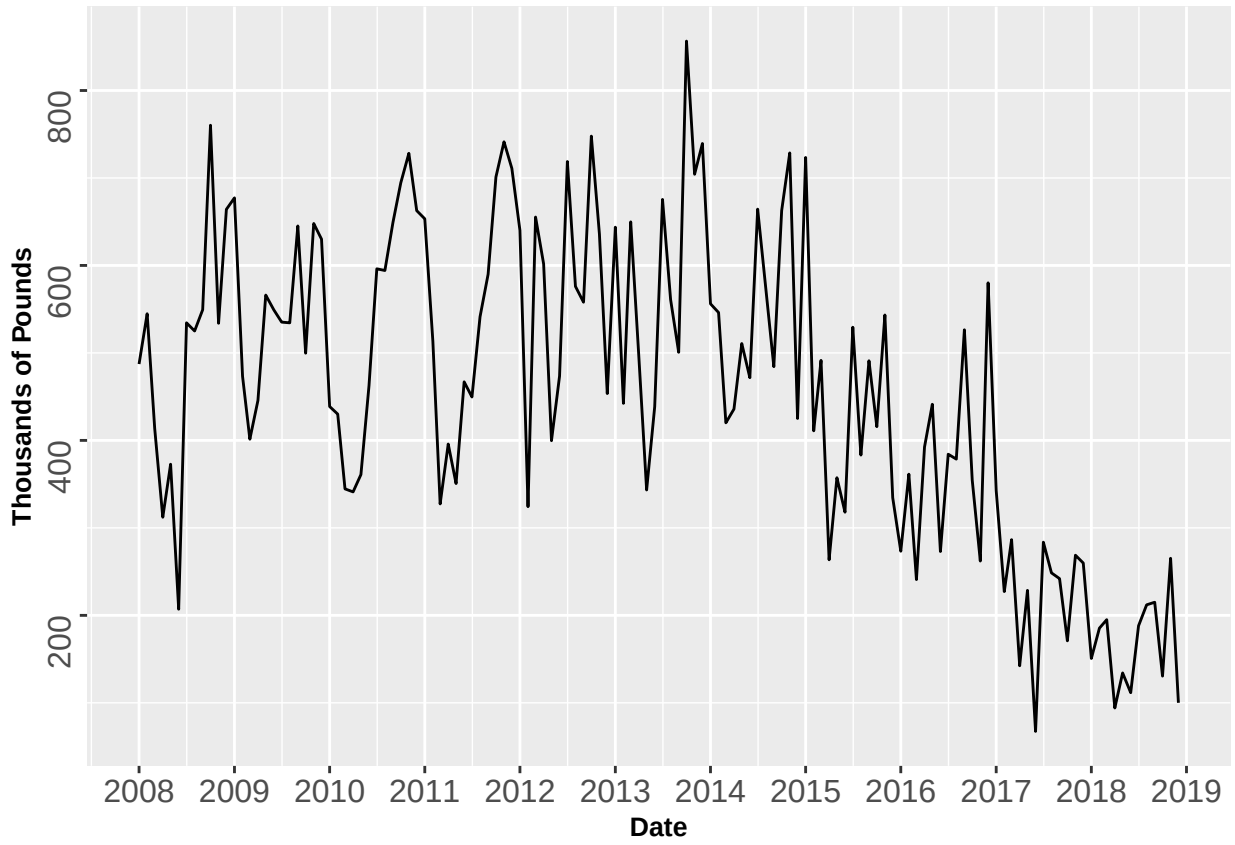
```
## The following object is masked from 'package:utils':
##
##     tar
```

Import and Clean the Data

```
## Load the Data
landings_full = read.csv("Data/SB_Red_Sea_Urchin_Landings_2008_2019.csv")

## Clean the Data
# Rename the Sea-Urchin Landings variable to pounds
# for simplicity in analysis
names(landings_full)[3] = "pounds"

# Create a Monthly Date Variable
landings_full$date = as.Date(with(landings_full,
                             paste0(as.character(landings_full$Year),"-",
                                    as.character(landings_full$Month),"-01"), "%Y-%m-%d"))

# Create a Separate Dataset for 2008-2018
landings = landings_full[1:132,]
```

Plot the Sea-Urching Landings Data

```
## Plot the Original Data
full_plot = ggplot(data = landings, mapping = aes(x = date, y = pounds/1000)) +
  geom_line() +
  labs(x = "Date", y = "Thousands of Pounds") +
  #labs(title = "Red Sea Urchin Landings in the Santa Barbara Area\nMeasured in Thounsands of Pounds\nM
  scale_x_date(breaks = scales::breaks_pretty(10)) +
  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"))
full_plot
```




```
## Plots for report
png(filename = "Images/2008_2018_plot.png", width = 960, height = 480)
full_plot
dev.off()
```

```
## pdf
##   2
```

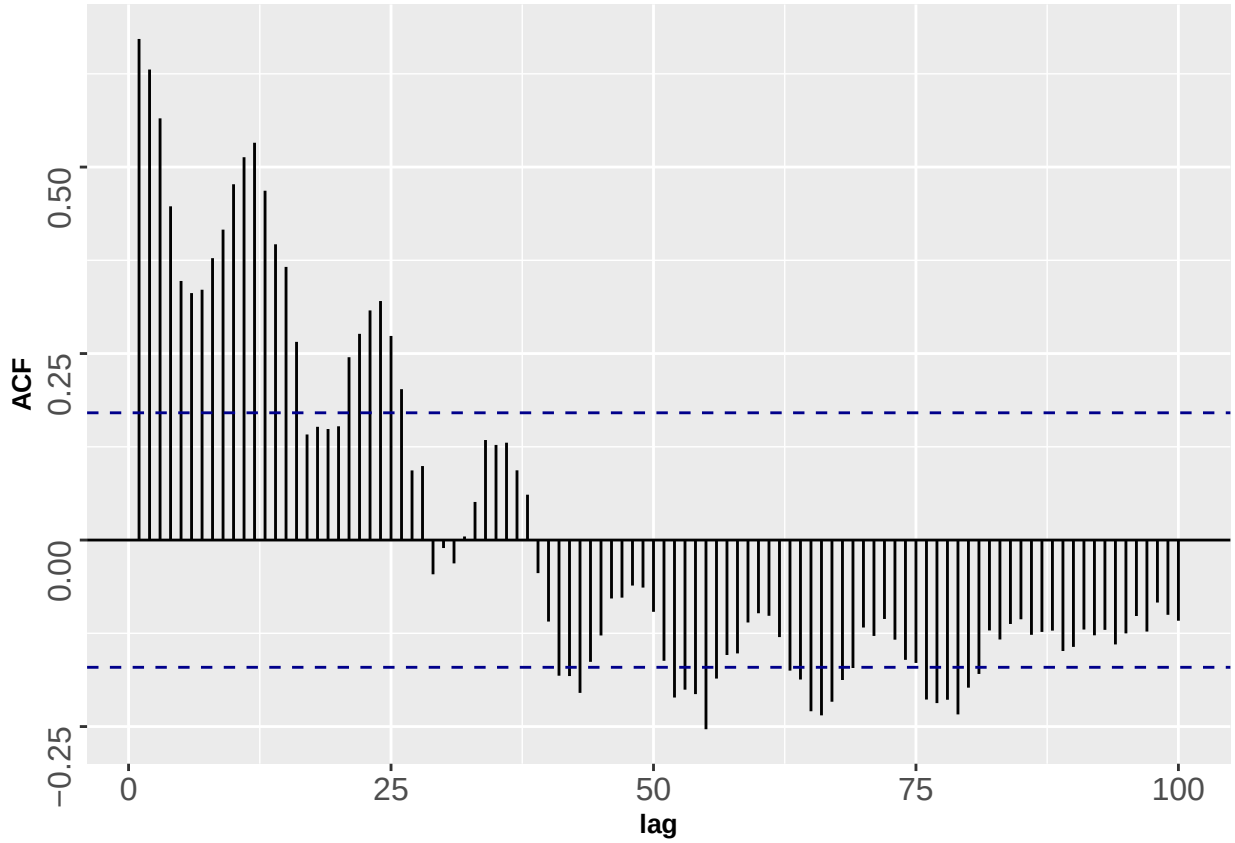
Save the Cleaned Data Out

```
save(landings,file="Data/landings.Rdata")
save(landings_full,file="Data/landings_full.Rdata")
```

ACF of Original Data

```
## Sample ACF
sample_acf_list = acf(landings$pounds, 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$pounds))

# Plot
ACF_original= 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 = "") +
  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"))

ACF_original
```




```
## Plots for report
png(filename = "Images/ACF_original.png", width = 960, height = 480)
ACF_original
dev.off()

## pdf
##   2
```