irish_whiskey_ww

Devin DeLeon-Dowd

2023-07-26

After uploading the data into R Studio.

```
#melt data into long format
melt_irish_whisk <- irish_whiskey %>% melt()

#turn data into df
data_irish_whiskey <- tibble(melt_irish_whisk)

#check headers of tibble
head(data_irish_whiskey)
```

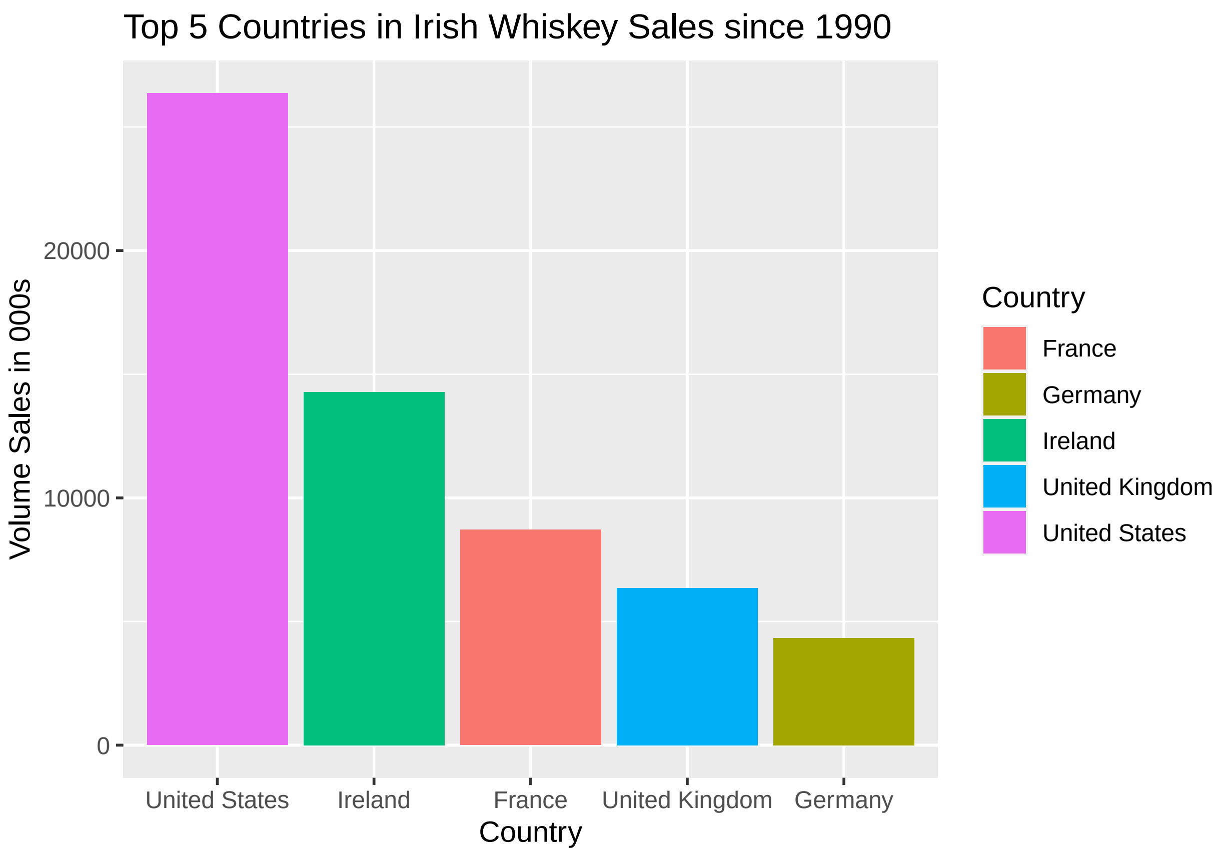
```
## # A tibble: 6 x 5
##   ...1          Quality  Country       variable  value
##   <chr>         <chr>    <chr>         <fct>     <dbl>
## 1 Irish Whiskey Standard United States 1990     243
## 2 Irish Whiskey Standard Ireland       1990     538.
## 3 Irish Whiskey Standard France        1990      92
## 4 Irish Whiskey Standard South Africa  1990       7.00
## 5 Irish Whiskey Standard Russia        1990      NA
## 6 Irish Whiskey Standard Germany       1990      80
```

```
#rename rearrange the data columns
order_irish_whisk <- data_irish_whiskey %>%
  select(Quality, Country, variable, value) %>%
  rename(Year = variable, Volume = value) %>%
  relocate(Year, Country, Quality, Volume)

#check headers of ordered irish whiskey
head(order_irish_whisk)
```

```
## # A tibble: 6 x 4
##   Year  Country       Quality  Volume
##   <fct> <chr>         <chr>     <dbl>
## 1 1990  United States Standard 243
## 2 1990  Ireland       Standard 538.
## 3 1990  France        Standard  92
## 4 1990  South Africa  Standard   7.00
## 5 1990  Russia        Standard  NA
## 6 1990  Germany       Standard  80
```

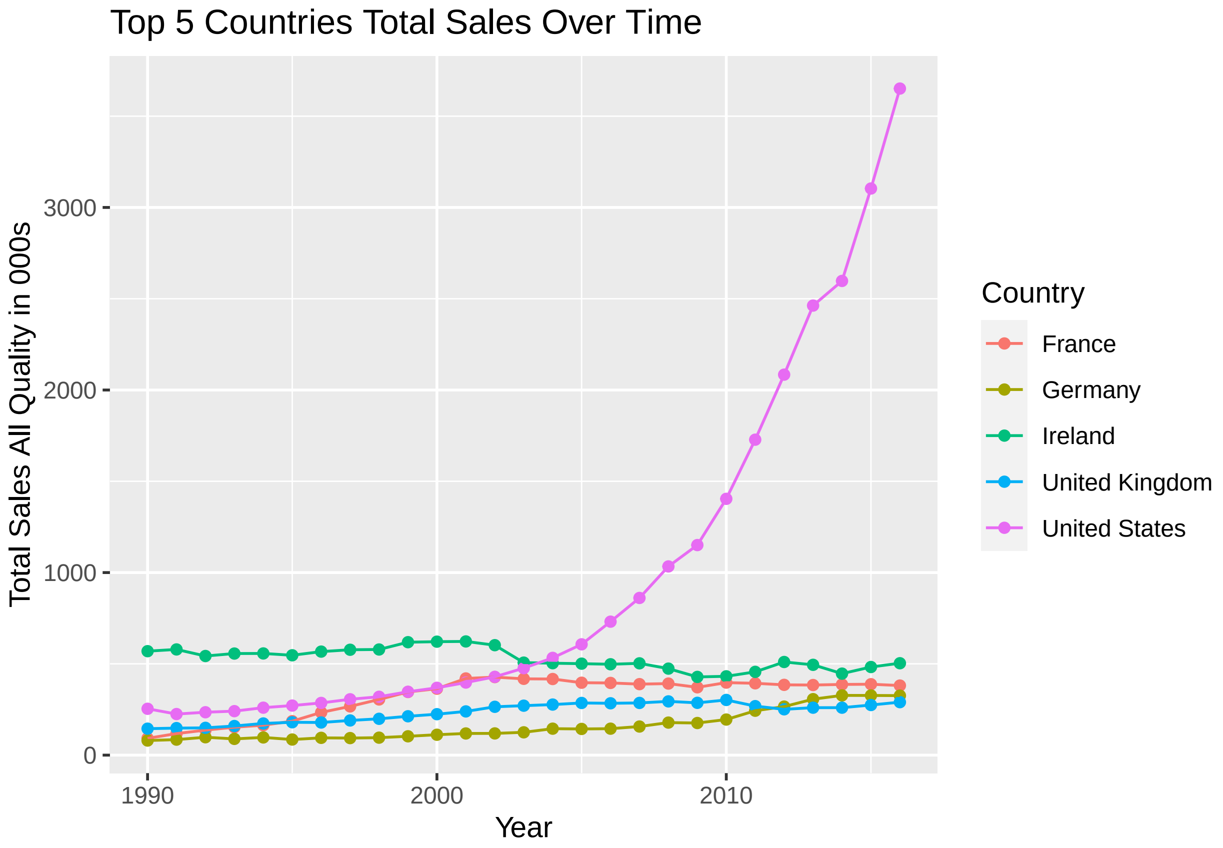
```
#change the date into the correct format for plot and time series analysis
order_irish_whisk$Year <- year(as.POSIXct(order_irish_whisk$Year,
                                          format = "%Y"))

#create total of sales per year, quality, not counting NA values
```

```
order_irish_whisk_total_sales <- order_irish_whisk %>%
  group_by(Year,Country,Quality) %>%
  summarize(total_sales = sum(Volume, na.rm = TRUE))

#order data by country and mutate to create total sales by country
total_sales_country <- order_irish_whisk %>% group_by(Country) %>%
  summarize(country_total_sales = sum(Volume, na.rm = TRUE))

#create top 5 countries in descending order based on sales
top_5 <- total_sales_country %>% arrange(desc(country_total_sales)) %>%
  top_n(5)

#plot the top 5 countries by sales in descending order
ggplot(top_5, aes(x = reorder(Country, -country_total_sales),
                  y = country_total_sales, fill = Country)) +
  geom_bar(stat = "identity") +
  ggtitle("Top 5 Countries in Irish Whiskey Sales since 1990") +
  xlab("Country") +
  ylab ("Volume Sales in 000s")
```

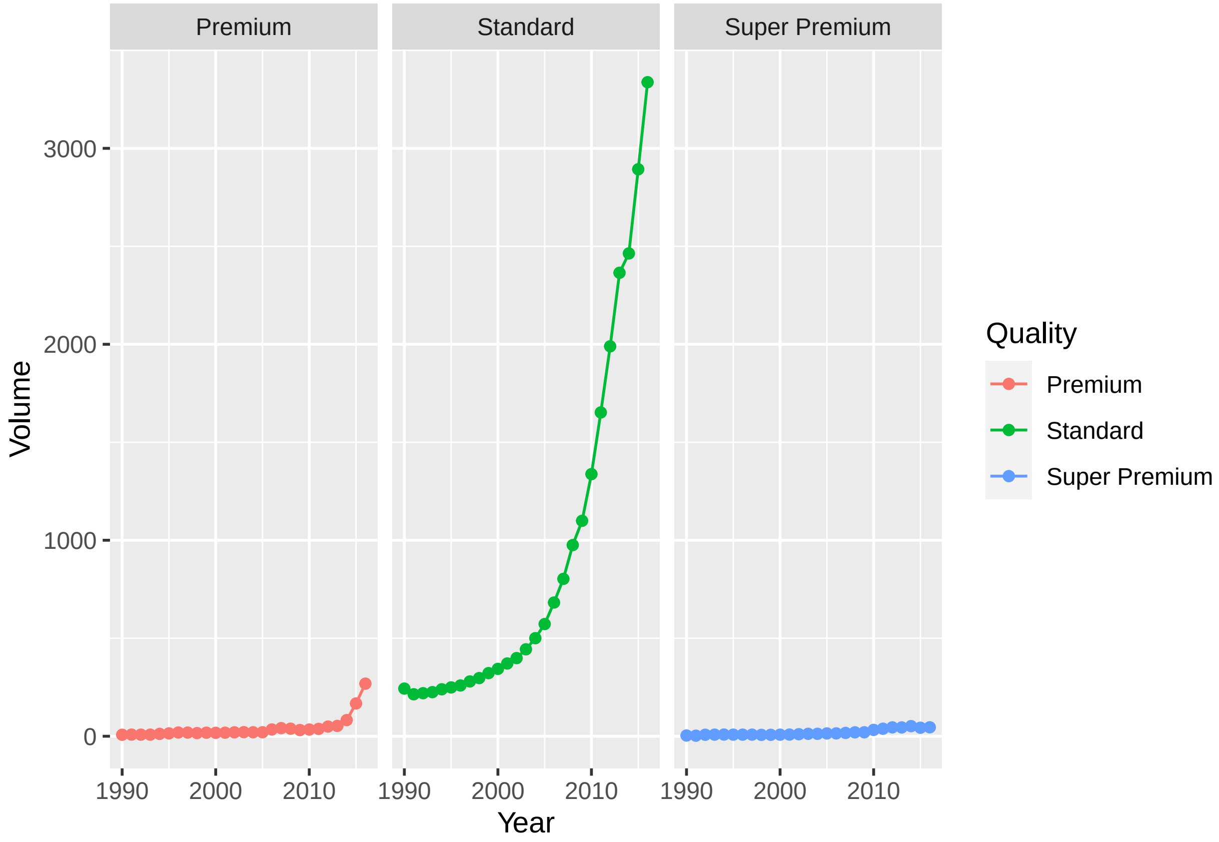


```
#create top 5 country sales over time for all quality of whiskey
top_5_over_time <- order_irish_whisk_total_sales %>%
  filter(Country %in%
      c('United States', 'Ireland', 'France', 'United Kingdom', 'Germany')) %>%
  group_by(Country, Year) %>%
  summarize(total_sales_all_quality = sum(total_sales))
```

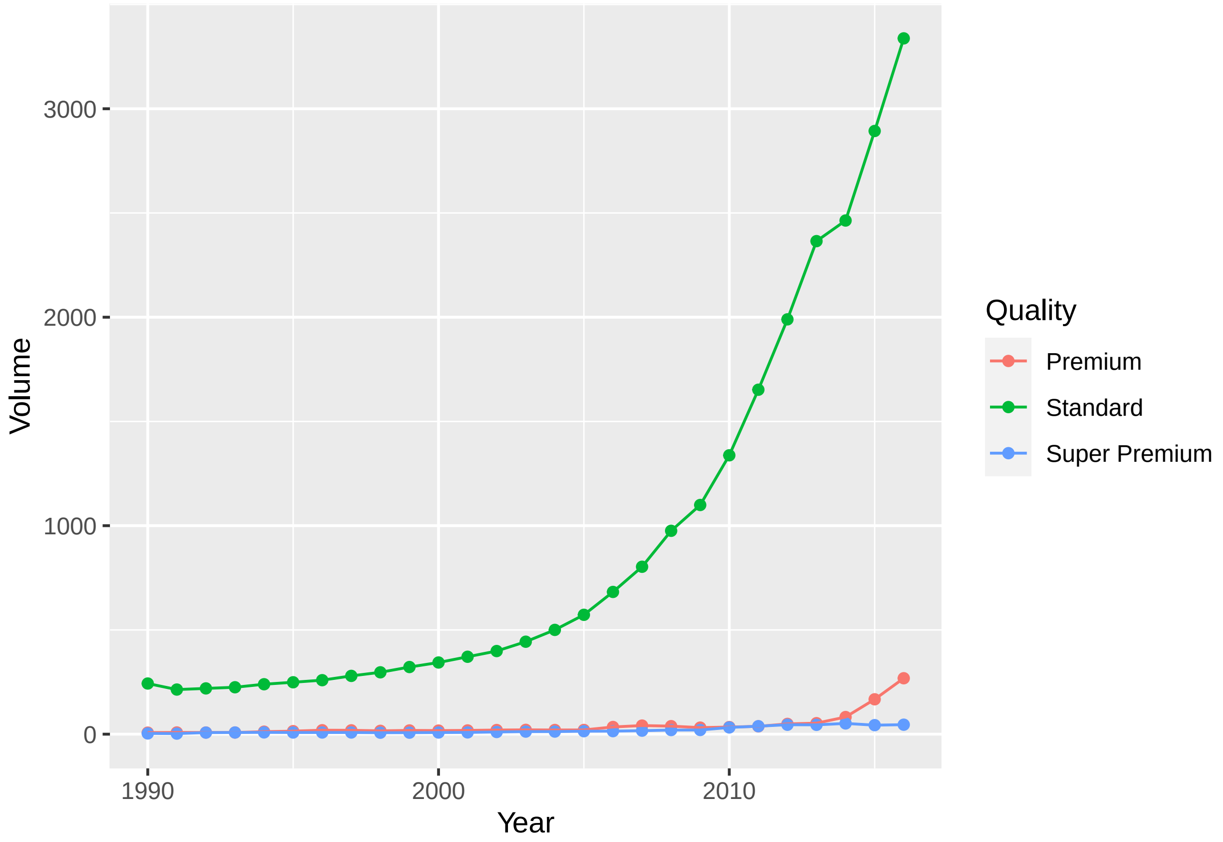
```r
#plot the geom line of the top five countries for all the years
ggplot(top_5_over_time, aes(x = Year, y = total_sales_all_quality,
                            color = Country)) +
  geom_point() + geom_line() +
  ggtitle("Top 5 Countries Total Sales Over Time") +
  xlab("Year") +
  ylab("Total Sales All Quality in 000s")
```



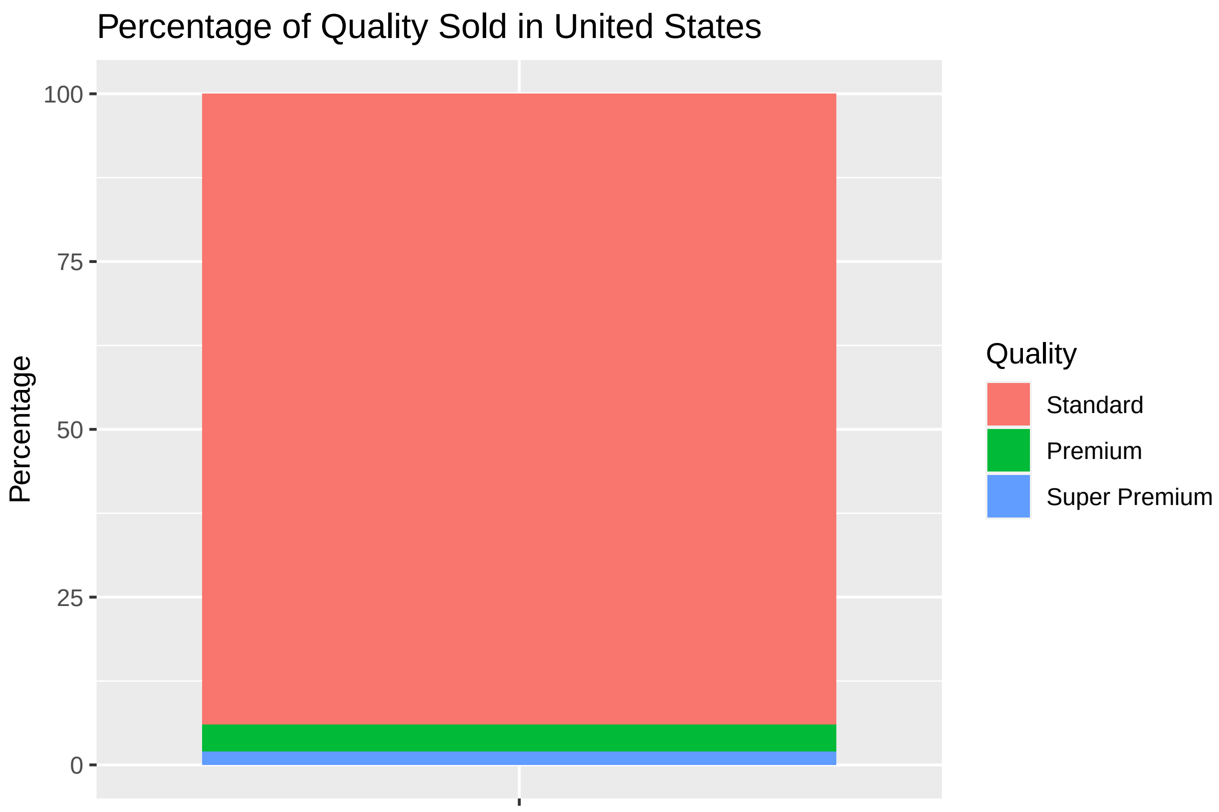
```r
#focus on United States sales because it sells the most and has highest growth over time
usa_sales <- order_irish_whisk %>% filter(Country == "United States")

#plot data over time for each quality, facet_warp and together
usa_sales %>%
  ggplot(aes(x = Year, y = Volume, color = Quality)) +
  geom_point() + geom_line() +
  facet_wrap(~ Quality, nrow = 1, ncol = 3)
```



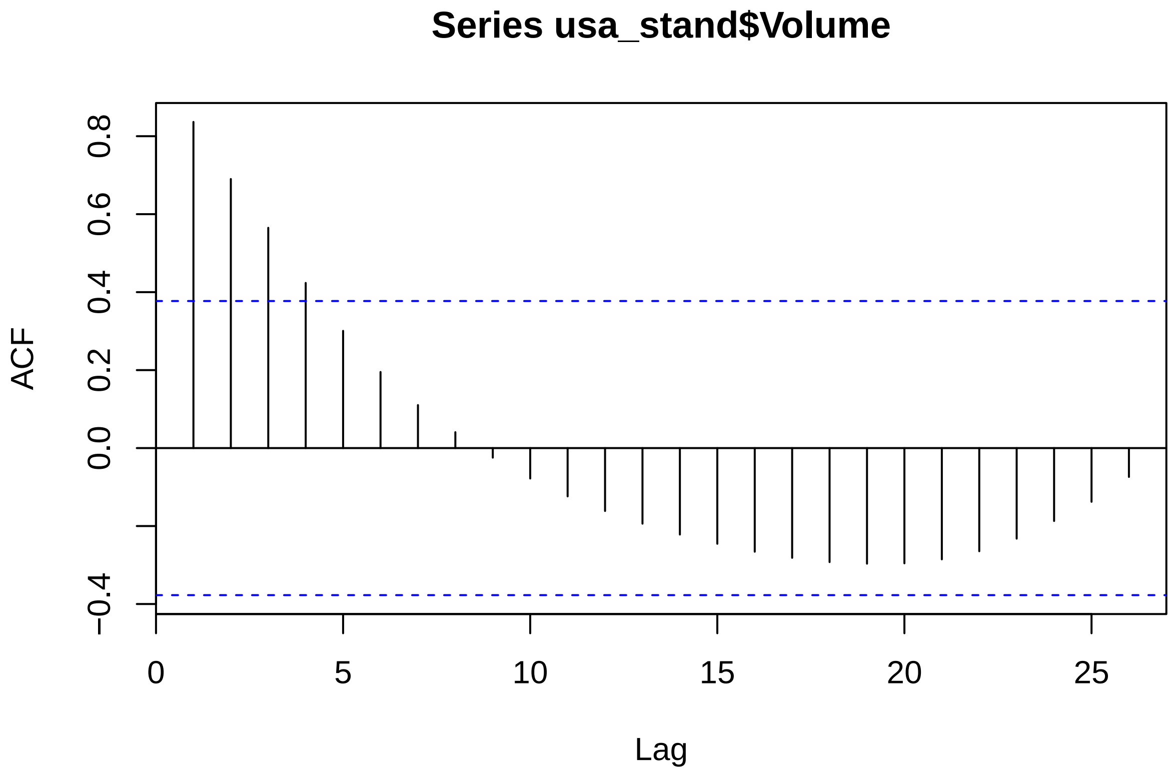
```
#focus on standard irish whiskey sales bc it is the most purchased over time
usa_sales %>% ggplot(aes(x = Year, y = Volume, color = Quality)) +
  geom_point() + geom_line()
```



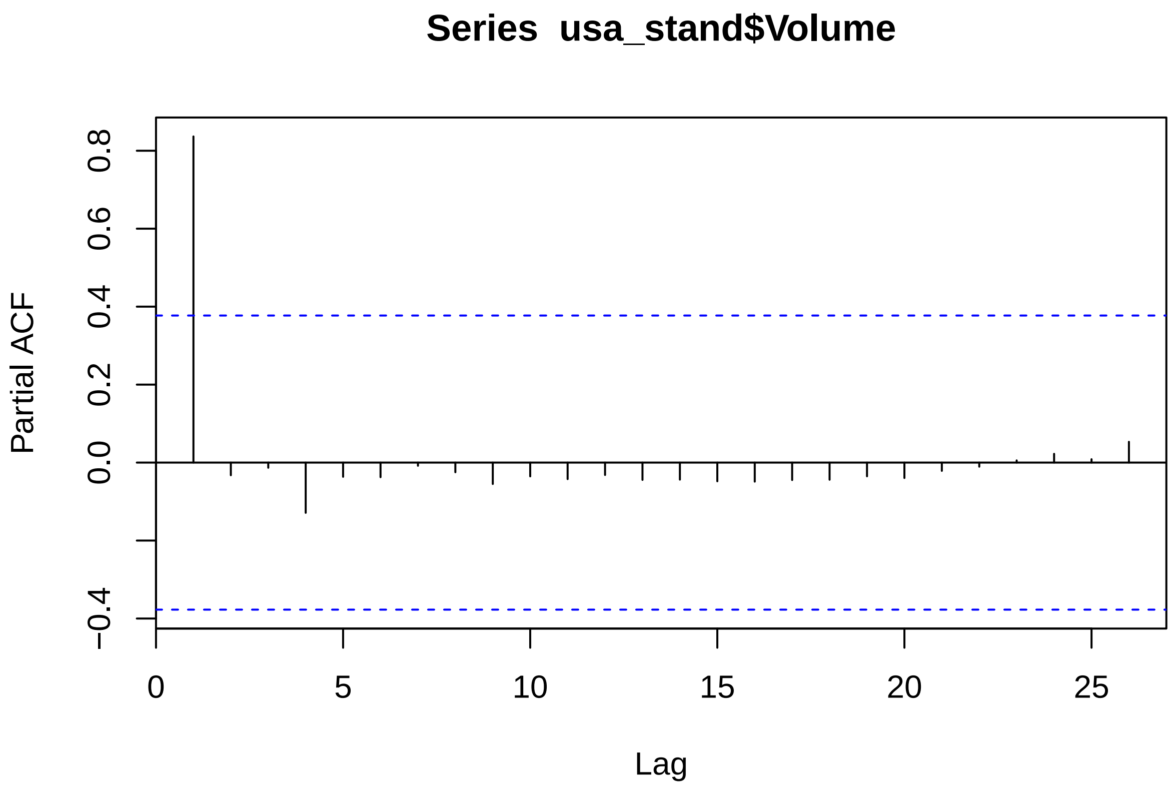
```
#plot proportion of usa quality of whiskey sales
usa_sales %>%
  group_by(Quality) %>%
  summarize(Total = sum(Volume)) %>%
  mutate(Percentage = round(Total / sum(Total) * 100),2) %>%
  arrange(desc(Percentage)) %>%
  ggplot(aes(x = "",y =  Percentage, fill = fct_inorder(Quality))) +
  geom_bar(stat = "identity") +
  ggtitle("Percentage of Quality Sold in United States") +
  xlab(element_blank()) +
  ylab("Percentage") +
  labs(fill = "Quality")
```



```
#filter usa data based on qualities
usa_stand <- usa_sales %>% filter(Quality == "Standard")

#acf and pacf plots for stationarity analysis
acf(usa_stand$Volume, lag.max = 50)
```



```
pacf(usa_stand$Volume, lag.max = 50)
```



There is a sinusoidal patter that appers to converge to zero with

```
#adf test and kpss test for stationarity
#adf test for stationarity
#if p value > 0.05 then non stationary, if p value < 0.05 then stationary
adf.test(usa_stand$Volume) # series is non stationary
```

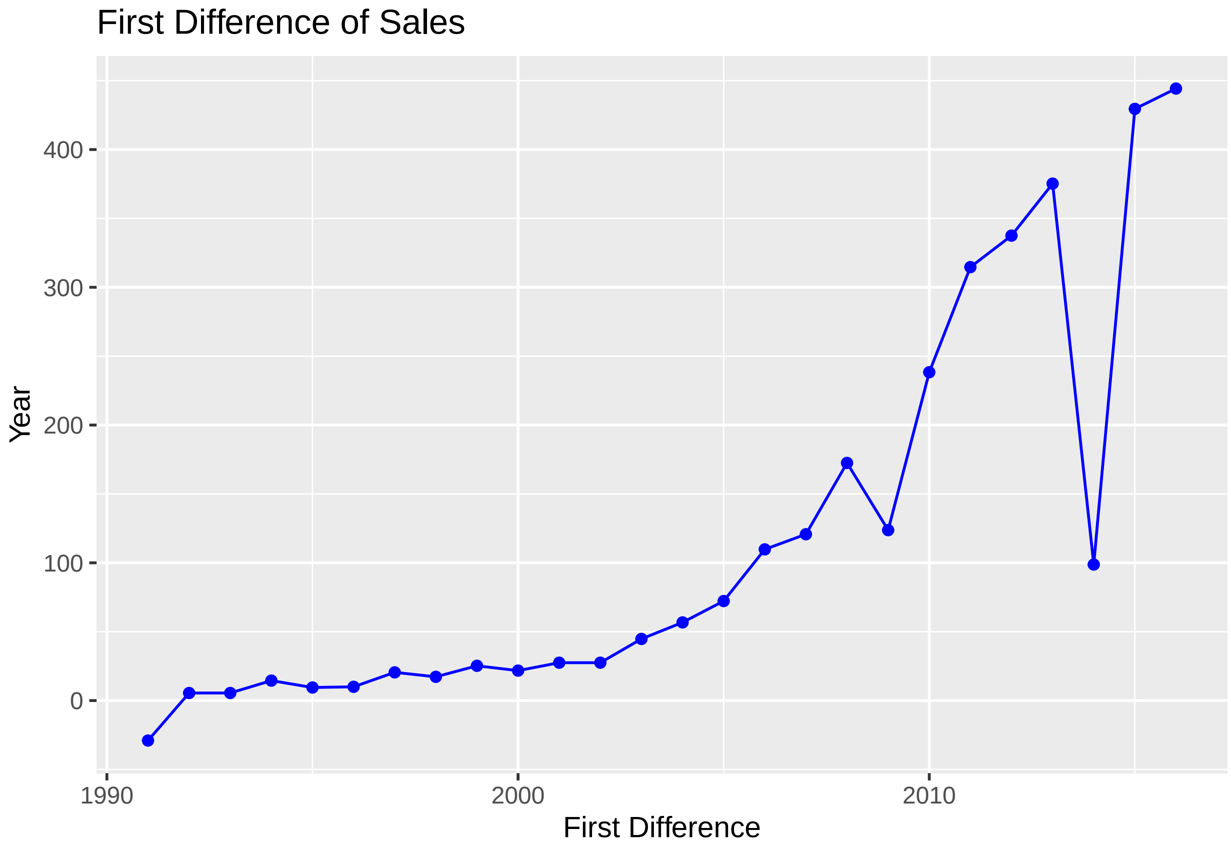
```
##
##  Augmented Dickey-Fuller Test
##
## data:  usa_stand$Volume
## Dickey-Fuller = 2.4102, Lag order = 2, p-value = 0.99
## alternative hypothesis: stationary
```

```
#kpss test for trend stationarity
#if p value > 0.05 then stationary, if p vaulue < 0.05 then non stationary
kpss.test(usa_stand$Volume) #series is non stationary
```

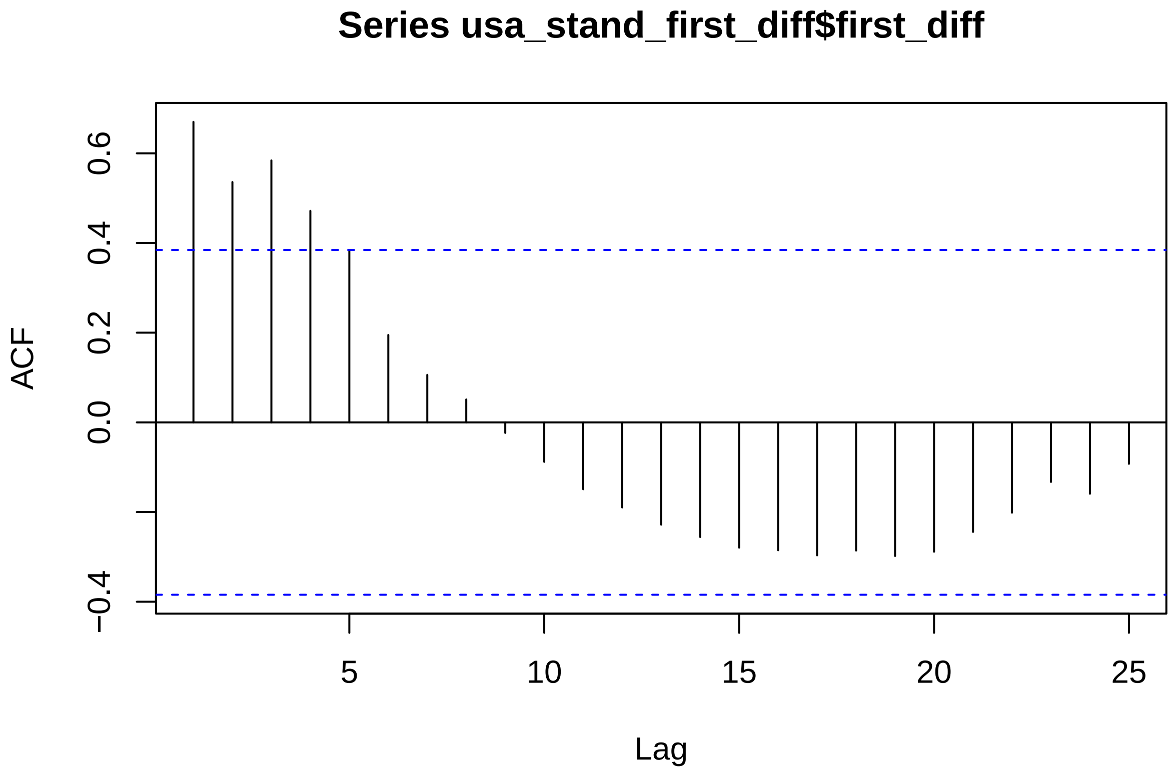
```
##
##  KPSS Test for Level Stationarity
##
## data:  usa_stand$Volume
## KPSS Level = 0.8211, Truncation lag parameter = 2, p-value = 0.01
```

```
#find difference of series to achieve stationarity
usa_stand_first_diff <- usa_stand %>%
  mutate(first_diff = difference(Volume, lag = 1, difference = 1)) %>%
  drop_na(first_diff)
```

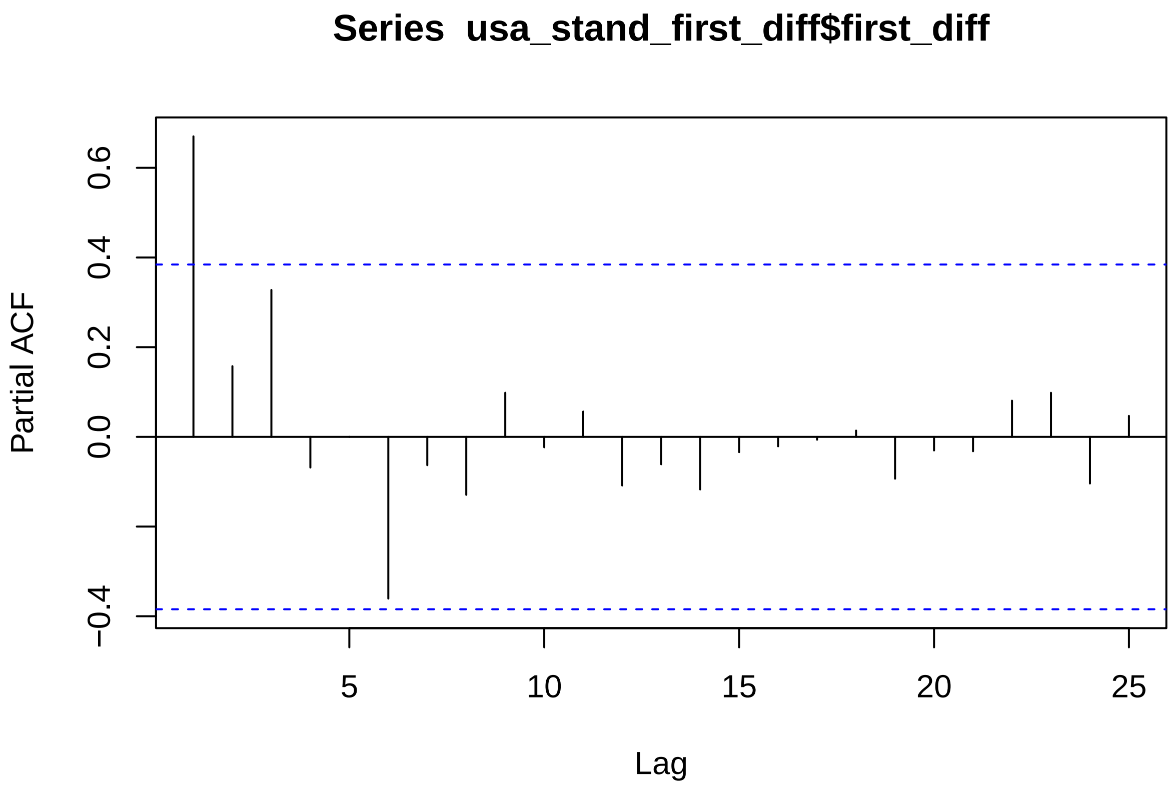
```
#plot first difference of sales
usa_stand_first_diff %>%
  ggplot(aes(x = Year, y = first_diff)) + geom_point(color = "blue") +
  geom_line(color = "blue") +
  ggtitle("First Difference of Sales") +
  xlab("First Difference") +
  ylab("Year")
```



```
#acf and pacf plots for first difference for stationarity analysis
acf(usa_stand_first_diff$first_diff, lag.max = 50)
```



```
pacf(usa_stand_first_diff$first_diff, lag.max = 50)
```



```
#adf test and kpss test for stationarity
#adf test for stationarity
#if p value > 0.05 then non stationary, if p value < 0.05 then stationary
adf.test(usa_stand_first_diff$first_diff) # series is non stationary
```

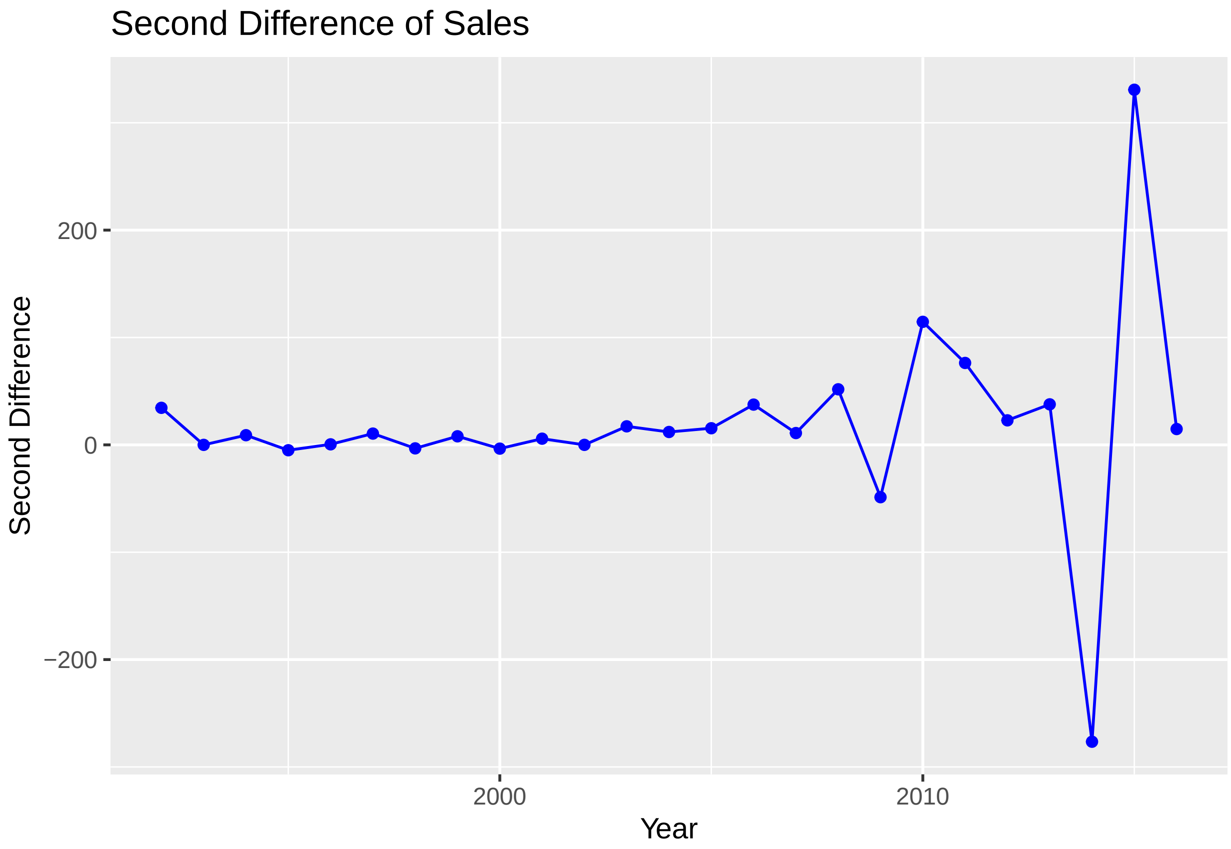
```
##
##  Augmented Dickey-Fuller Test
##
## data:  usa_stand_first_diff$first_diff
## Dickey-Fuller = -1.5136, Lag order = 2, p-value = 0.7577
## alternative hypothesis: stationary
```

```
#kpss test for trend stationarity
#if p value > 0.05 then stationary, if p vaulue < 0.05 then non stationary
kpss.test(usa_stand_first_diff$first_diff) #series is non stationary
```

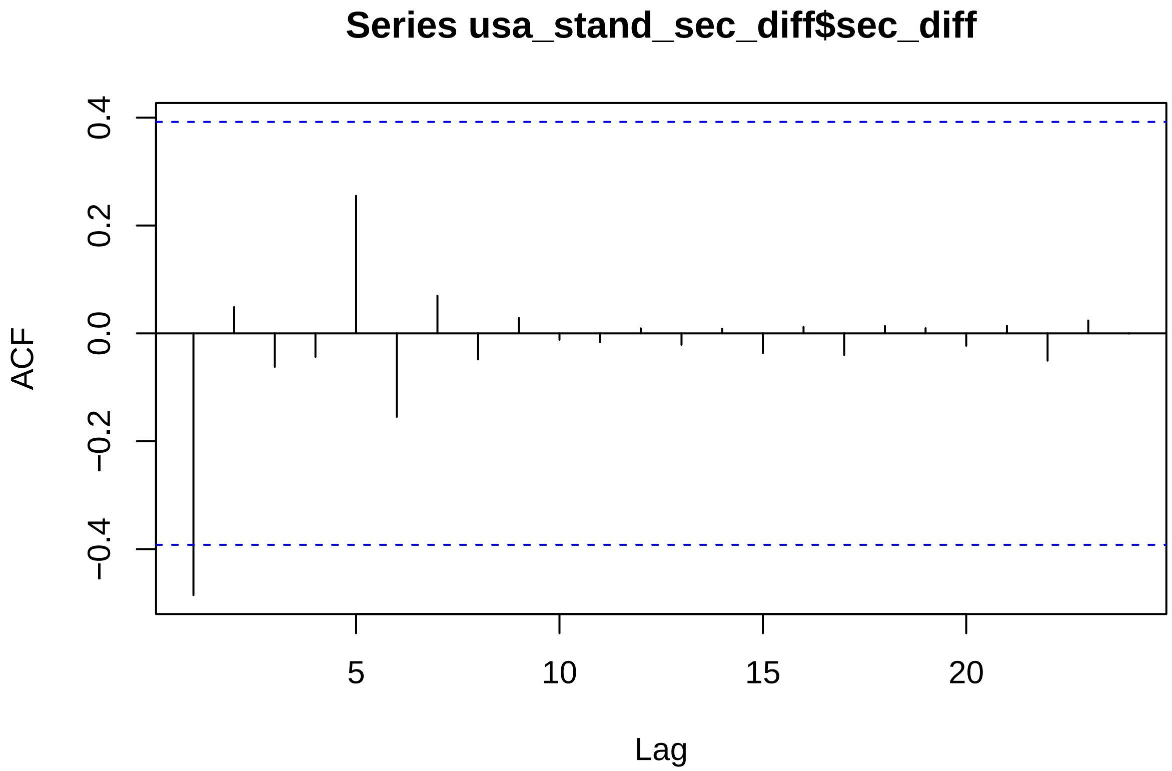
```
##
##  KPSS Test for Level Stationarity
##
## data:  usa_stand_first_diff$first_diff
## KPSS Level = 0.87247, Truncation lag parameter = 2, p-value = 0.01
```

```
#take the second difference of series to achieve stationarity
usa_stand_sec_diff <- usa_stand_first_diff %>%
  mutate(sec_diff = difference(first_diff, lag = 1, difference = 1)) %>%
  drop_na(sec_diff)

#plot the sec difference of sales
```

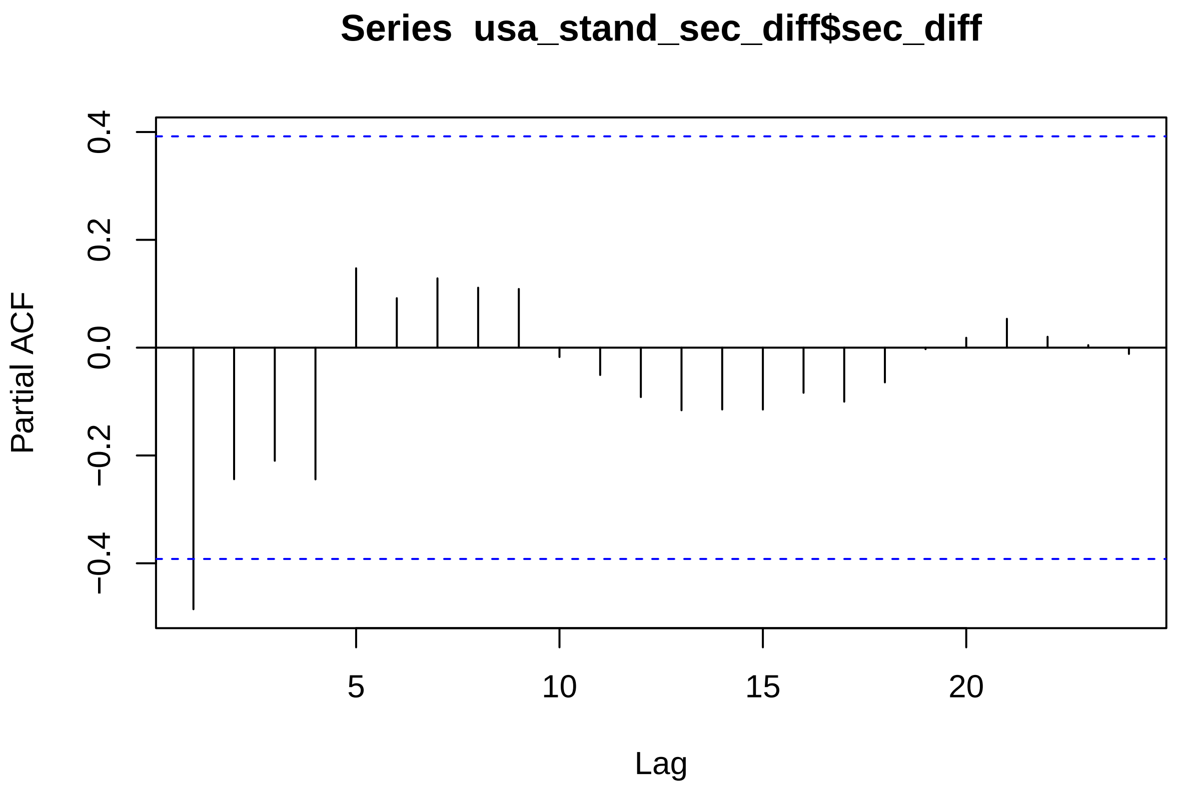
```
usa_stand_sec_diff %>%
  ggplot(aes(x = Year, y = sec_diff)) + geom_point(color = "blue") +
  geom_line(color = "blue") +
  ggtitle("Second Difference of Sales") +
  xlab("Year") +
  ylab("Second Difference")
```



```
#acf and pacf plots for first difference for stationarity analysis
acf(usa_stand_sec_diff$sec_diff, lag.max = 50)
```



```
pacf(usa_stand_sec_diff$sec_diff, lag.max = 50)
```



```
#adf test and kpss test for stationarity
#adf test for stationarity
#if p value > 0.05 then non stationary, if p value < 0.05 then stationary
adf.test(usa_stand_sec_diff$sec_diff) # series is stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  usa_stand_sec_diff$sec_diff
## Dickey-Fuller = -4.3888, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

```
#kpss test for trend stationarity
#if p value > 0.05 then stationary, if p vaulue < 0.05 then non stationary
kpss.test(usa_stand_sec_diff$sec_diff) #series is stationary
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  usa_stand_sec_diff$sec_diff
## KPSS Level = 0.17898, Truncation lag parameter = 2, p-value = 0.1
```

```
#create best model for forecasting the originally series, arima vs exp smoothing
bma_stand <- auto.arima(usa_stand$Volume, seasonal = FALSE) #aic = 295.88
summary(bma_stand)
```

```
## Series: usa_stand$Volume
## ARIMA(0,2,1)
```

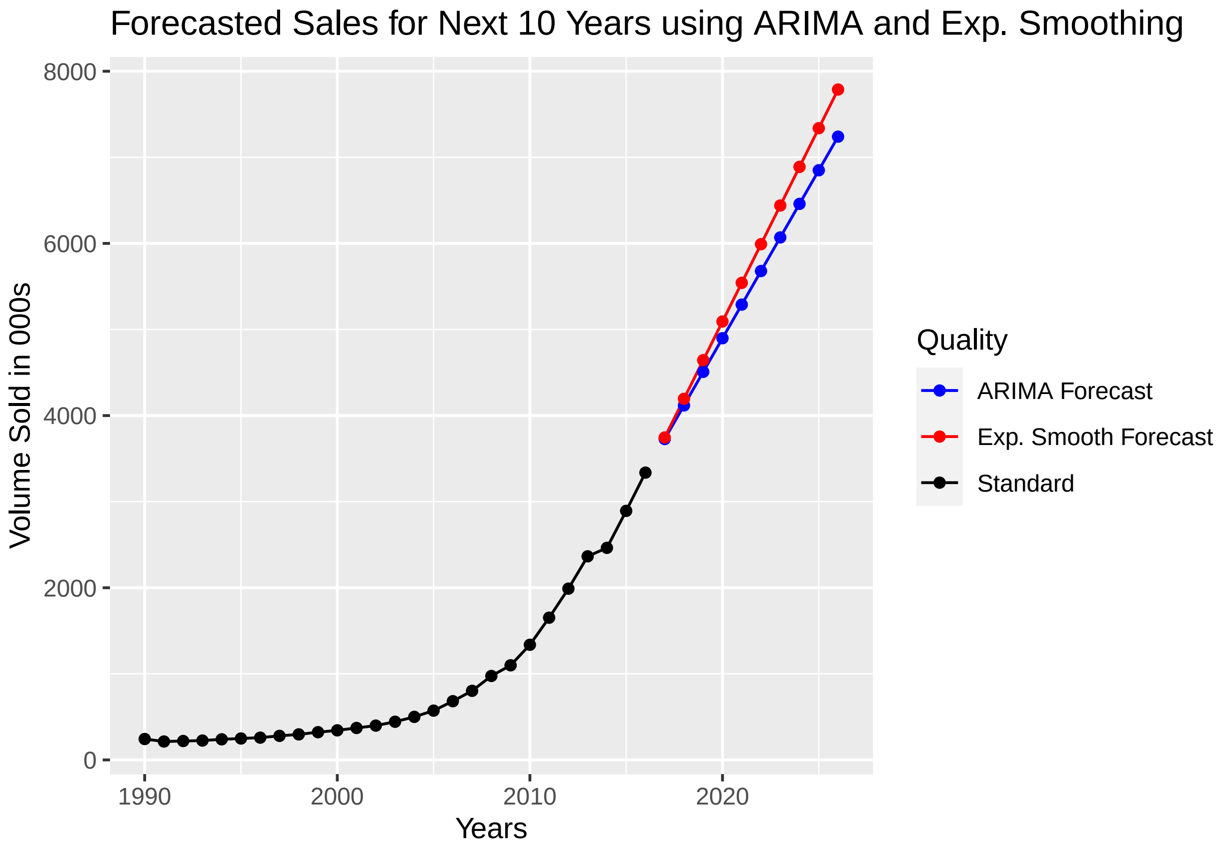
```
##
## Coefficients:
##           ma1
##       -0.4643
## s.e.   0.1500
##
## sigma^2 = 7105:  log likelihood = -145.94
## AIC=295.88   AICc=296.43   BIC=298.32
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 28.62882 79.46782 47.83591 3.360848 4.217867 0.3945229 -0.2135999
```

```
bma_forecast <- bma_stand %>% forecast(h = 10, level = 95)
#turn forecast into tibble
bma_forecast_t <- tibble(Year = c(2017:2026), Country = 'United States',
                         Quality = 'ARIMA Forecast', Volume = bma_forecast$mean)
#remember volume is predicted


#exp smoothing
besm_stand <- ets(usa_stand$Volume, model = "ZZZ")
#ZZZ means error,trend,season types are automatically selected, aic = 284.9730
summary(besm_stand)
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = usa_stand$Volume, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.7616
##     beta  = 0.7616
##
##   Initial states:
##     l = 248.8838
##     b = -17.2237
##
##   sigma:  0.0581
##
##      AIC     AICc      BIC
## 284.9730 287.8301 291.4522
##
## Training set error measures:
##                     ME     RMSE      MAE    MPE     MAPE      MASE       ACF1
## Training set 22.67647 81.24902 47.00336 2.4594 4.320791 0.3876566 -0.1160251
```

```
besm_forecast <- besm_stand %>% forecast(h = 10, level = 95)
#turn forecast into tibble
besm_forecast_t <- tibble(Year = c(2017:2026), Country = 'United States',
                  Quality = "Exp. Smooth Forecast", Volume = besm_forecast$mean)
```

```
#combine the forecasted and sales tibble in one data frame to plot all at
#once and compare
long_data <- bind_rows(usa_stand, bma_forecast_t) %>%
  bind_rows(besm_forecast_t)
#melt data into appropriate format
complete_data <- melt(long_data, id = c("Year", "Country","Quality","Volume"))

#both time series forecasting plots together
ggplot(complete_data, aes(x = Year, y = Volume, color = Quality)) +
  geom_line() +
  geom_point() +
  scale_color_manual(values = c("blue","red","black")) +
  ggtitle("Forecasted Sales for Next 10 Years using ARIMA and Exp. Smoothing") +
  xlab("Years") +
  ylab("Volume Sold in 000s")
```


Forecasted Sales for Next 10 Years using ARIMA and Exp. Smoothing
Volume Sold in 000s
Years
Quality
ARIMA Forecast
Exp. Smooth Forecast
Standard