Econ 144 Project 1

Seungyeon yoo

2023-10-21

Contents

I. Introduction **2**

II. Results **2**

1. Modeling and Forecasting Trend . . . 2
    - Loading the dataset . . . 2
    - (a) Plotting the data . . . 2
    - (b) Assessing Stationary . . . 3
    - (c) ACF and PACF . . . 3
    - (d) Fitting a Model . . . 4
    - (e) Inspecting the Residuals of Each Model . . . 5
    - (f) Histograms of the Residuals . . . 6
    - (g) Diagnostic Statistics . . . 8
    - (h) Model Selection using AIC and BIC . . . 9
    - (i) h-steps Forcasting . . . 9
2. Trend and Seasonal Adjustments . . . 10
    - (a) Additive Decomposition . . . 10
    - (b) Multiplicative Decomposition . . . 12
    - (c) Additive vs. Multiplicative . . . 14
    - (d) Cycles in randomness . . . 14
    - (e) Seasonal factor . . . 14
    - (f) Trend-Seasonal Model . . . 15
    - Evaluating the Model . . . 16

III. Conclusion and Future Work **18**

IV. References **19**

I. Introduction

This project attempts to model and forecast the **Advance Retail Sales** in the United States. The advance retail sales refers to the early estimated amount of retail sales, specifically in a monthly frequency. Retail sales are an important economic factor that provides a statistical measure of consumers' behavior in the market and offers an indication of how the overall economy is or will be performing. Since economic factors tend to stimulate one another, and the retail trade is one of the major components that consists of the total GDP, modeling the overall trend of the historical data and attempting to forecast the future values based on them may provide valuable insights on the economic analysis, business performance assessment, and market trend prediction.

The dataset `RetailTrade` has monthly observations from January 1992 to August 2023 and contains a column, `DATE` for the date of each observation and `RSXFSN` for the numerical values (in millions of dollar units) of **the monthly retail trade amount**. The dataset is sourced from the ***Federal Reserve Economic Data***, a database maintained by the Federal Reserve Bank of St. Louis.

II. Results

```r
# Loading packages needed for the project
library(fpp3)
library(forecast)
library(ggplot2)
```

1. Modeling and Forecasting Trend

Loading the dataset

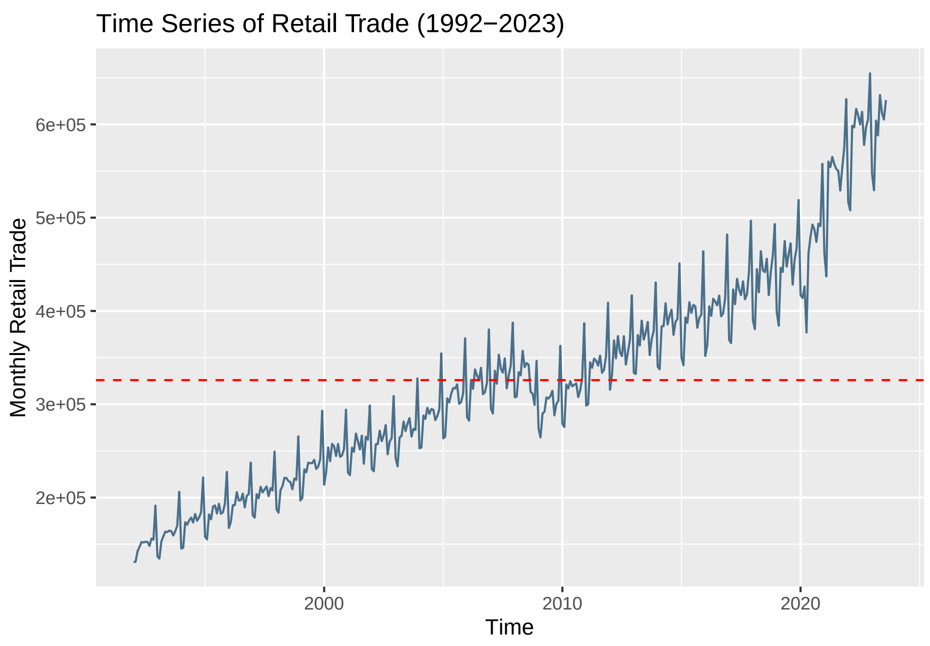
```r
retail_trade <- read.csv("RetailTrade.csv") #Loading the data set
# checking for any missing values
paste("There is", sum(is.na(retail_trade)), "missing value(s)")
```

```
## [1] "There is 0 missing value(s)"
```

```r
# Converting the dataset to a time series object
ts_retail <- ts(retail_trade[,2], start = 1992, freq = 12)
# Converting the dataset to a tsibble object
tsibble_retail <- retail_trade %>%
  mutate(DATE = yearmonth(DATE)) %>%
  as_tsibble()
```

(a) Plotting the data

```r
autoplot(ts_retail, color = "skyblue4") +
  labs(y = "Monthly Retail Trade", title = "Time Series of Retail Trade (1992-2023)") +
  geom_hline(aes(yintercept = mean(ts_retail)), color = "red", linetype = "dashed")
```

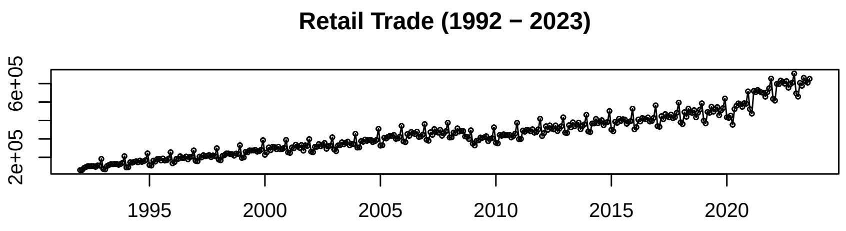


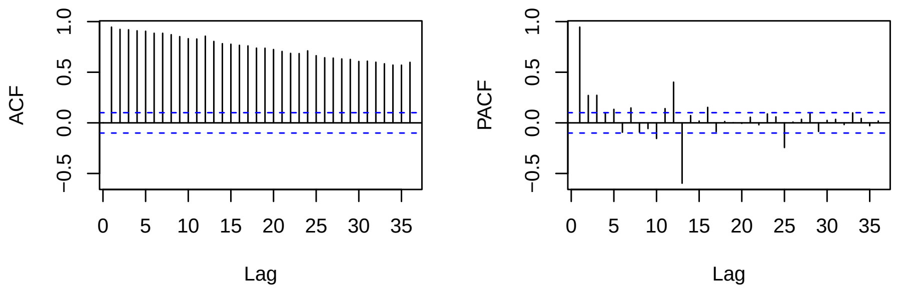
(b) Assessing Stationary

The plot in **(a)** exhibits an upward trend over time, and it can be visually observed that each point in the time period has different means. In addition, while the plot seems to have a consistent seasonality, the amplitude tends to progressively increase over time, indicating non-constant variance. **In short, the data are not covariance stationary.**

(c) ACF and PACF

```r
tsdisplay(ts_retail, main = "Retail Trade (1992 - 2023)") # Plotting ACF and PACF
```

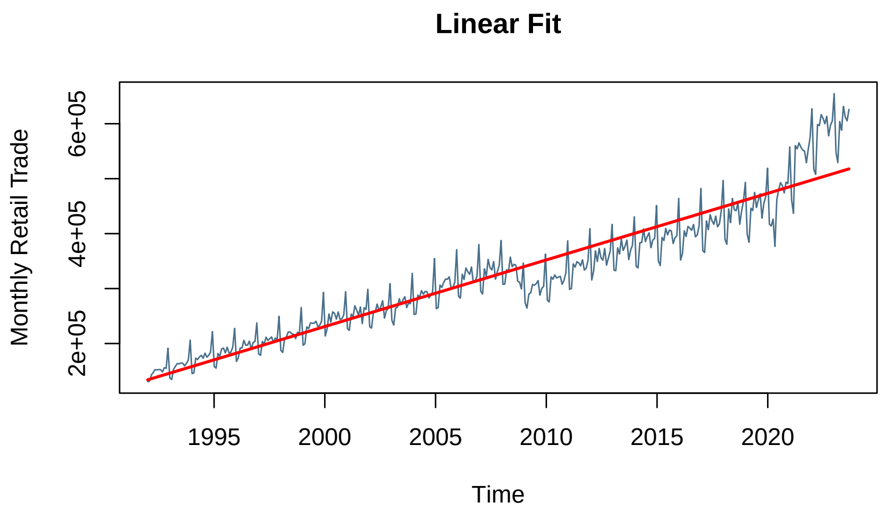


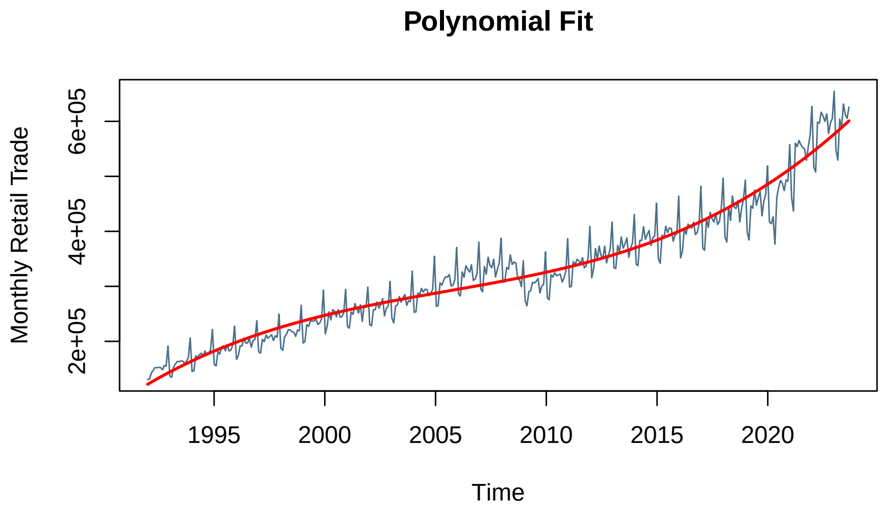


The plots above show ACF (Autocorrelation function) and PACF (Partial Autocorrelation Function). The ACF shows that the Retail Trade time series is strongly correlated to its own lagged data. Especially, it is noteworthy that there is a slight spike in every 12th lag (ex. lag 12, lag 24, lag 36), which indicates consistent seasonality of the time series. From the PACF, it becomes clear that the monthly retail trade also has some significant direct correlation with the past months, particularly with the most recent three months and the months at lag 12 and 13.

(d) Fitting a Model

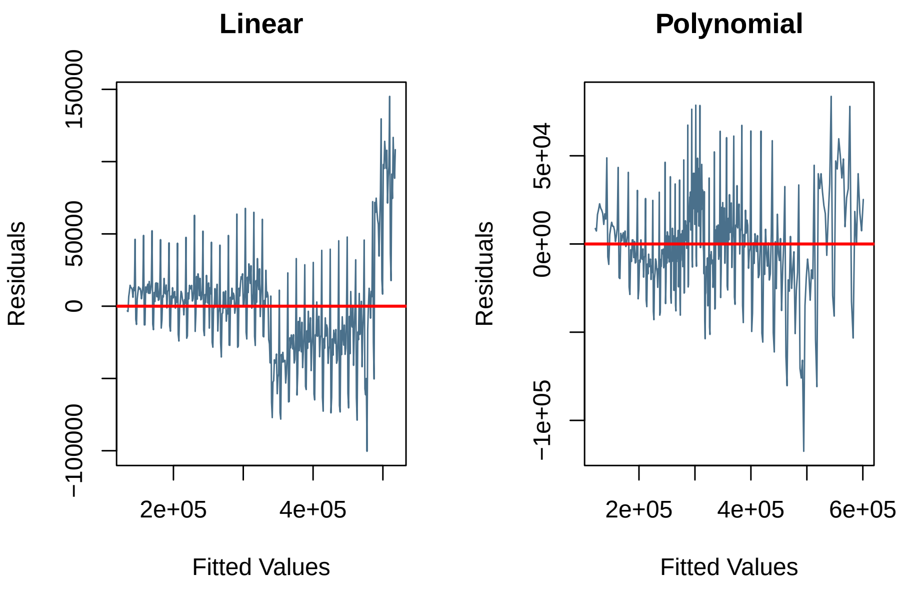
```r
t <- seq(1992, 2023.666, length = length(ts_retail)) # Creating a time dummy variable


# 1. Linear Fit
par(mfrow = c(2,1))
lmodel <- lm(ts_retail ~ t)
plot(t, ts_retail, type = "l", col = "skyblue4",
     main = "Linear Fit", xlab = "Time" ,ylab = "Monthly Retail Trade")
lines(t, lmodel$fit, col = "red", lwd = 2)


# 2. Polynomial Fit
pmodel <- lm(ts_retail ~ t + I(t^2) + I(t^4))
plot(t, ts_retail, type = "l", col = "skyblue4",
     main = "Polynomial Fit", xlab = "Time" ,ylab = "Monthly Retail Trade")
lines(t, pmodel$fit, col = "red", lwd = 2)
```





(e) Inspecting the Residuals of Each Model

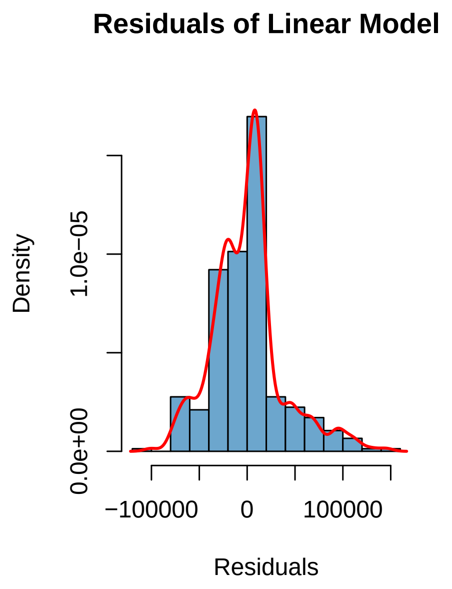
```
# Linear model residuals
par(mfrow = c(1,2))
plot(lmodel$fitted.values, lmodel$residuals,
     type = "l", xlab = "Fitted Values", ylab = "Residuals",
     main = "Linear", col = "skyblue4")
abline(h = 0, col = "red", lwd = 2)

# Polynomial model residuals
plot(pmodel$fitted.values, pmodel$residuals,
     type = "l", xlab = "Fitted Values", ylab = "Residuals",
     main = "Polynomial", col = "skyblue4")
abline(h = 0, col = "red", lwd = 2)
```

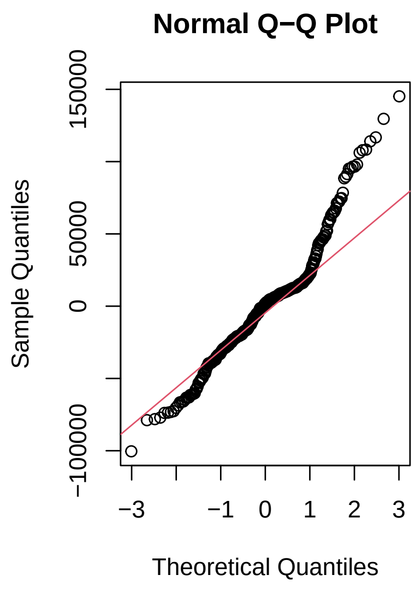


For a residual plot, what should be carefully looked at is its randomness (No certain patterns around the horizontal line) and relatively constant variances. Just by visual inspection, there may be at least some patterns in the plots, but the ***Polynomial*** model seems to perform relatively well based on the residual plots. As for the ***Linear*** model, it shows a good fit in the beginning, but it fails to capture "shocks", which is a sudden decrease in retail trade around 2008, the great recession period, and a slight increase around 2021. In short, although further investigations and more sophisticated modeling methods would likely create a better fit, up to this point, the polynomial model better describes the time series and would possibly perform better in forecasting.

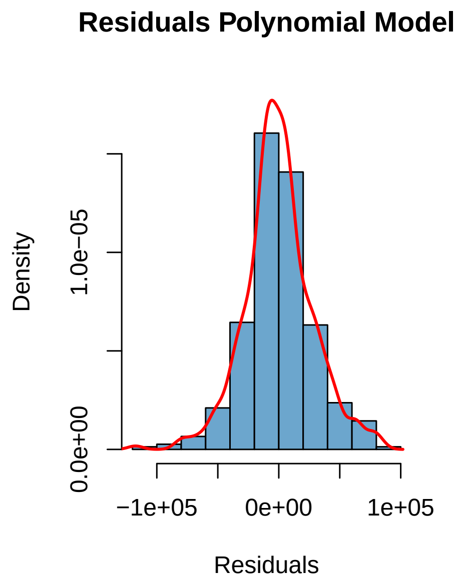
(f) Histograms of the Residuals

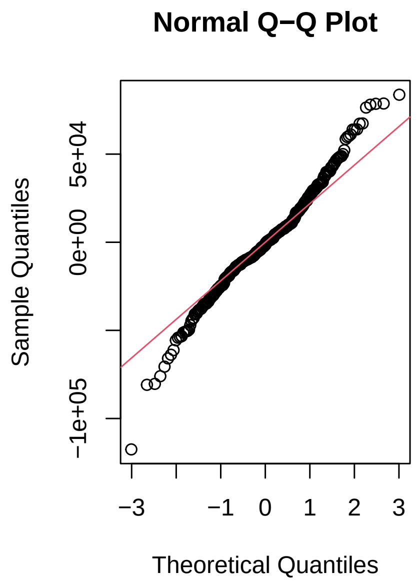
```r
# 1. Histogram of Linear Model Residuals Distribution
par(mfrow = c(1,2))
hist(lmodel$residuals, prob = TRUE,
     ylim = c(0, 1.8e-05), col = "skyblue3", xlab = "Residuals",
     main = "Residuals of Linear Model") # Residuals distribution of linear model
lines(density(lmodel$residuals), col = "red", lwd = 2)


# qqplot for linear model residuals
qqnorm(lmodel$residuals)
qqline(lmodel$residuals, col = 2)
```





```
# 2. Histogram of Polynomial Model Residuals Distribution
hist(pmodel$residuals, prob = TRUE,
     ylim = c(0, 1.8e-05), col = "skyblue3", xlab = "Residuals",
     main = "Residuals Polynomial Model") # Residuals distribution of linear model
lines(density(pmodel$residuals), col = "red", lwd = 2)

# qqplot for polynomial model residuals
qqnorm(pmodel$residuals)
qqline(pmodel$residuals, col = 2)
```





The plots above visualize the residual distributions and Q-Q plots of each model. First, the distribution of the linear model is slightly right-skewed, which is also supported by the Q-Q plot. As also observed earlier in the residuals vs. fitted values plot, the linear model overestimated the sudden decrease of the time series, thus the error terms are positively skewed. In contrast, the residual distribution of polynomials seems to be more normally distributed than the linear model, and the Q-Q plot shows a better alignment with the theoretical line. This also agrees with the previous observation in (e).

(g) Diagnostic Statistics

```r
summary(lmodel) # Summary Statistics of the linear model
```

```
##
## Call:
## lm(formula = ts_retail ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -100392  -22064    1767   12880  145156
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.401e+07  4.175e+05  -57.52   <2e-16 ***
## t            1.212e+04  2.079e+02   58.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37150 on 378 degrees of freedom
## Multiple R-squared:  0.8999, Adjusted R-squared:  0.8996
## F-statistic:  3399 on 1 and 378 DF,  p-value: < 2.2e-16
```

```r
summary(pmodel) # Summary Statistics of the polynomial model
```

```
##
## Call:
## lm(formula = ts_retail ~ t + I(t^2) + I(t^4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -117526  -14741   -1468   14771   83686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.823e+11  1.449e+10  -12.58   <2e-16 ***
## t            2.424e+08  1.925e+07   12.59   <2e-16 ***
## I(t^2)      -9.065e+04  7.191e+03  -12.61   <2e-16 ***
## I(t^4)       3.756e-03  2.973e-04   12.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28150 on 376 degrees of freedom
## Multiple R-squared:  0.9428, Adjusted R-squared:  0.9424
## F-statistic:  2067 on 3 and 376 DF,  p-value: < 2.2e-16
```

The summary statistics provide a diagnostic statistical measure that allows us to evaluate the previous assumptions that we made based on the visualization. For the overall fit, the linear model has $R^2 = 0.8999$ and the polynomial model has $R^2 = 0.9428$, which supports the ***initial assumption that the polynomial model is a better fit***. Additionally, in terms of the significance of the models, both linear and polynomial have a large F-statistic of 3399 and 2067 respectively, and their p-values are very close to 0 as $<$ 2.2e-16, which demonstrates that they are both statistically significant.

(h) Model Selection using AIC and BIC

```r
# Creating a data frame object to store the results of AIC and BIC Test
aicbic <- data.frame(AIC = AIC(lmodel, pmodel)[,2], BIC = BIC(lmodel, pmodel)[,2])
rownames(aicbic) <- c("Linear", "Polynomial") # Assigning row names
aicbic
```

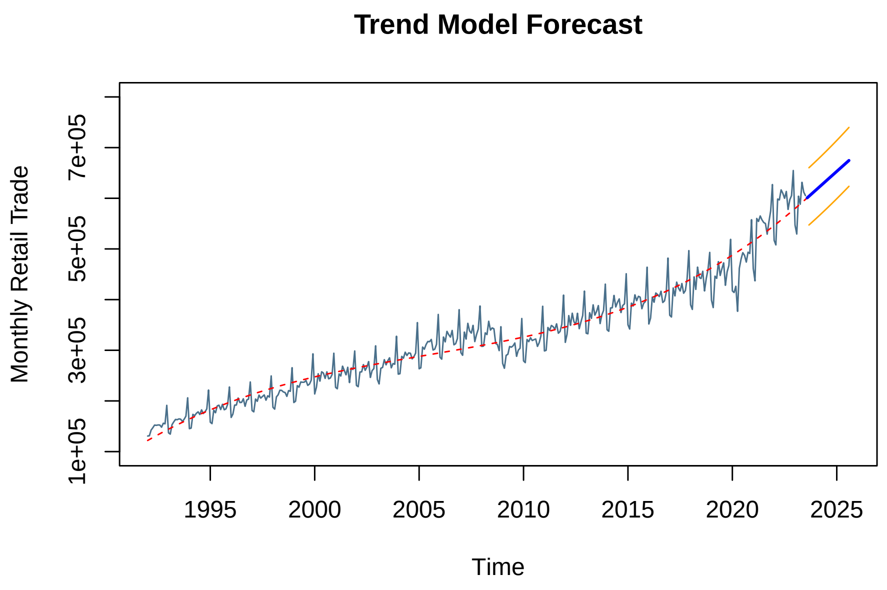
```
##                 AIC      BIC
## Linear     9079.662 9091.482
## Polynomial 8870.814 8890.514
```

The linear model has AIC and BIC of 9079.662 and 9091.482 respectively. The polynomial model has AIC and BIC of 8870.814 and 8890.514, both of which are smaller than the test values of the linear model. In other words, the test results from both AIC and BIC tests agree not only with each other but also with all the previous steps up to this point. Thus, from this point forward, the polynomial model will be used.

(i) h-steps Forcasting

```r
# Creating a new ts_retail object with 404 length (making a room for forecasted values)
fc_sample1 <- numeric(404)
fc_sample1[1:380] <- ts_retail
fc_sample1[380:404] <- NA
fc_sample1 <- ts(fc_sample1, start = 1992, freq = 12)

# Creating an ts object that includes the fitted values of pmodel
fc_sample2 <- numeric(404)
fc_sample2[381:404] <- NA
fc_sample2[1:380] <- pmodel$fitted.values
fc_sample2 <- ts(fc_sample2, start = 1992, freq = 12)


fc_fitted <- as.numeric(predict(pmodel$fitted.values, h = 24)$mean) # forcasted fitted values

# Creating a ts object for forecasted fitted values of pmodel
fc_sample3 <- numeric(404)
fc_sample3[381:404] <- fc_fitted
fc_sample3[1:380] <- NA
fc_sample3[380] <- fc_sample2[380]
fc_sample3 <- ts(fc_sample3, start = 1992, freq = 12)

# Combining the fitted values and forecasted fitted values
fc_sample4 <- numeric(404)
fc_sample4 <- fc_sample2
```

```r
fc_sample4[381:404] <- fc_sample3[381:404]


library(zoo) # loading zoo library for as.yearmon() function

# Creating a new time dummy variable with 404 length
new_t <- seq(1992, 2025.666, length = length(fc_sample4))

# plotting
plot(fc_sample1, ylim = c(1e+05, 8e+05), col = "skyblue4",
     ylab = "Monthly Retail Trade", main = "Trend Model Forecast")
lines(fc_sample2, col = "red", lty = "dashed")
lines(fc_sample3, col = "blue", lwd = 2) # forcasted fitted line
lines(as.yearmon(new_t)[381:404],
      forecast(pmodel, new_t)$lower[381:404 ,2], col = "orange") # 95 prediction interval
lines(as.yearmon(new_t)[381:404],
      forecast(pmodel, new_t)$upper[381:404 ,2], col = "orange") # 95 prediction interval
```



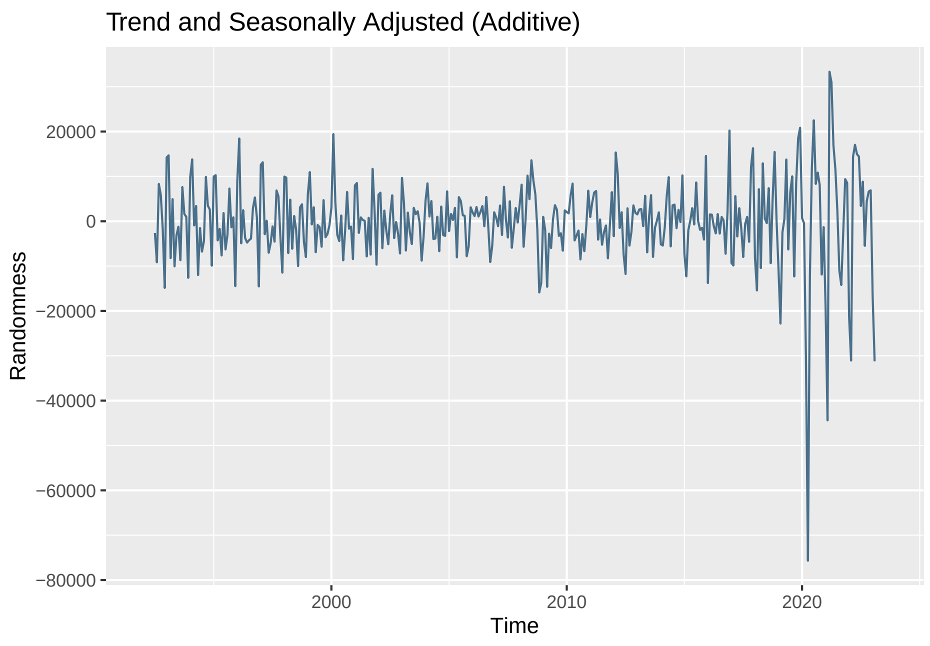
The plot above is **a 24-steps ahead forecast**. The red dashed line indicates the polynomial model fit, and the blue is the trend forecasting line, and the orange lines below and above the blue line form the 95% prediction intervals.

2. Trend and Seasonal Adjustments

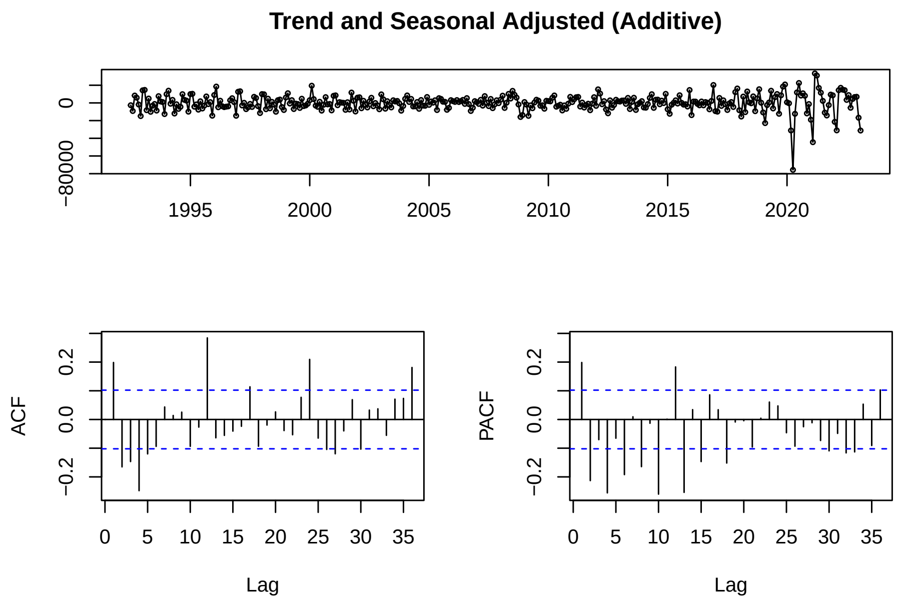
(a) Additive Decomposition

```r
dcp_retail_add <- decompose(ts_retail, type = "additive")

# Manual Additive Decomposition
```

```r
Y <- ts_retail
add_T <- dcp_retail_add$trend
add_S <- dcp_retail_add$seasonal
add_R <- dcp_retail_add$random

add_detrend <- Y - add_T # trend adjusted
add_seasonally_adj <- Y - add_S # seasonally adjusted

autoplot(Y - add_T - add_S, color = "skyblue4") + # Only left with R (randomness)
  labs(y = "Randomness", title = "Trend and Seasonally Adjusted (Additive)")
```



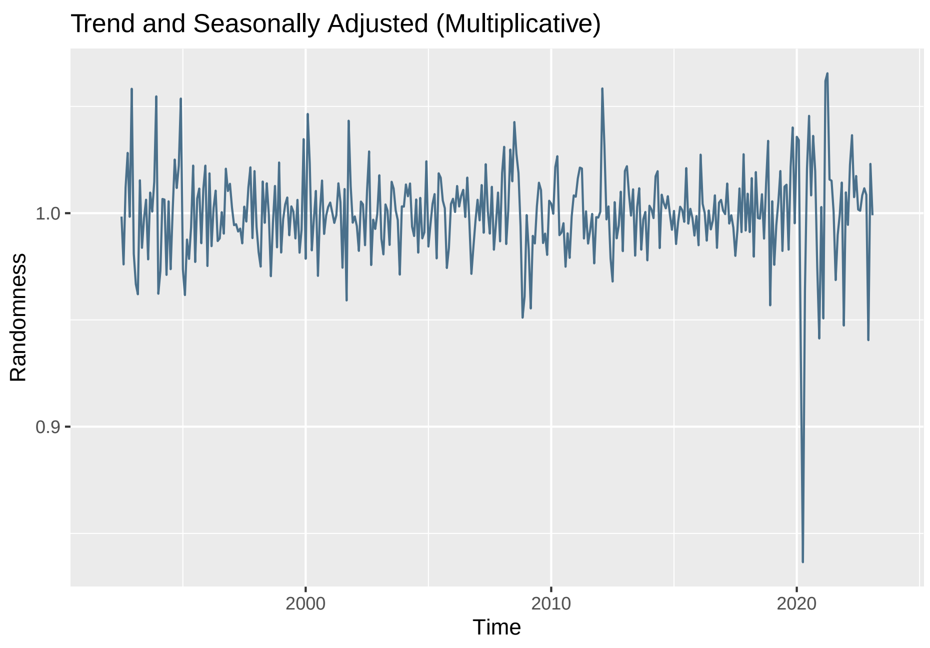
```r
tsdisplay(na.omit(Y - add_T - add_S),
          main = "Trend and Seasonal Adjusted (Additive)") # plotting ACF plot
```



We can see that there is much less correlation between lagged values, compared to the original data. However, even after detrending and removing the seasonality, ACF and PACF still indicate that there may be some more room for improvement. For example, although the amplitude of the plots is around 0.3, which is not as large as before, some seasonality is still observed in the ACF (every 12th lag), and both plots show that a considerable amount of lags are shown to be still significant.

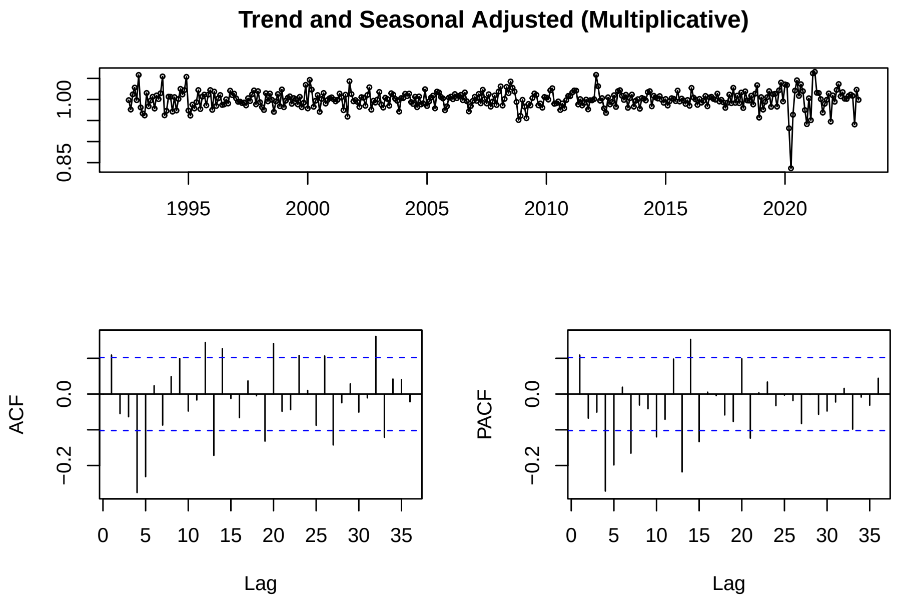
(b) Multiplicative Decomposition

```r
dcp_retail_mltp <- decompose(ts_retail, type = "multiplicative")

# Manual Additive Decomposition
Y <- ts_retail
mltp_T <- dcp_retail_mltp$trend
mltp_S <- dcp_retail_mltp$seasonal
mltp_R <- dcp_retail_mltp$random

mltp_detrend <- Y/mltp_T
mltp_seasonally_adj <- Y/mltp_S

autoplot(Y / mltp_T / mltp_S, color = "skyblue4") + # Only left with Randomness
  labs(y = "Randomness", title = "Trend and Seasonally Adjusted (Multiplicative)")
```



```
# The randomness that is left is much smaller in Multiplicative decomposition
```

```
tsdisplay(na.omit(Y / mltp_T / mltp_S),
          main = "Trend and Seasonal Adjusted (Multiplicative)") # ploting ACF and PACF
```



For the multiplicative decomposed series, it is noteworthy that the amplitude of the plot has changed, compared to the additive decomposition plot. The amplitude of the plot is smaller than the additive decomposition plot, which means the correlation decreased as well. Moreover, although there are still a good amount of lines that could be considered "significant", the seasonal pattern that has been consistently observed seems to be removed after multiplicative decomposition.

(c) Additive vs. Multiplicative

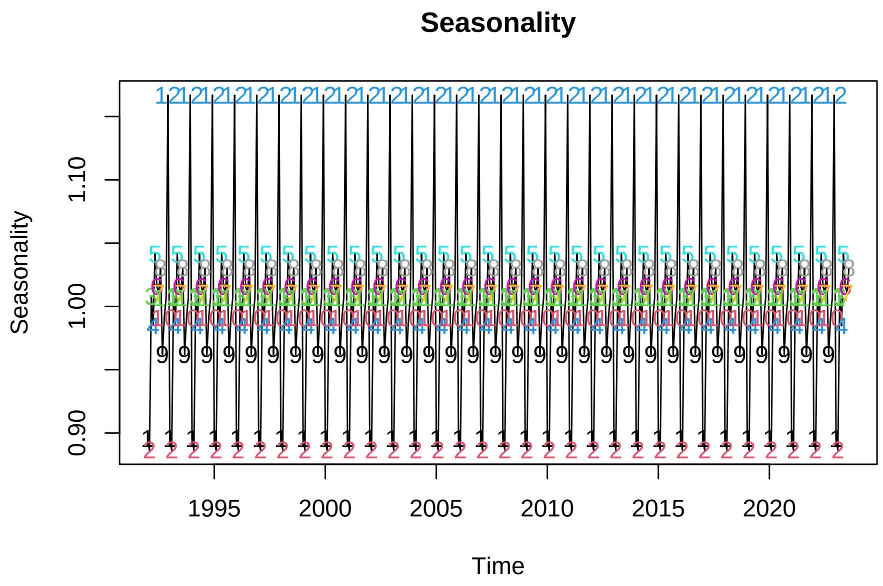
I would have to say that the multiplicative decomposition is better in this case because when comparing the two plots of randomness, the amplitude of the multiplicative decomposition plot was much smaller than the additive decomposition, which means that the magnitude of the error/randomness is smaller. In other words, we can minimize the unexpected errors using the multiplicative decomposition when attempting to forecast.

(d) Cycles in randomness

Based on the two decompositions conducted above and the investigation of random components, I believe that the models for the cycles can be very different from each other. The reason is that although the shapes of their residuals are very similar, the amplitudes of the residuals are significantly different. A higher amplitude of residuals means more significant variations that are not explained, which creates unpredictability in data. On the other hand, a smaller amplitude of residuals implies less fluctuation or more stability in data. Therefore, considering this reason, I believe that models for their cycles may be very different.

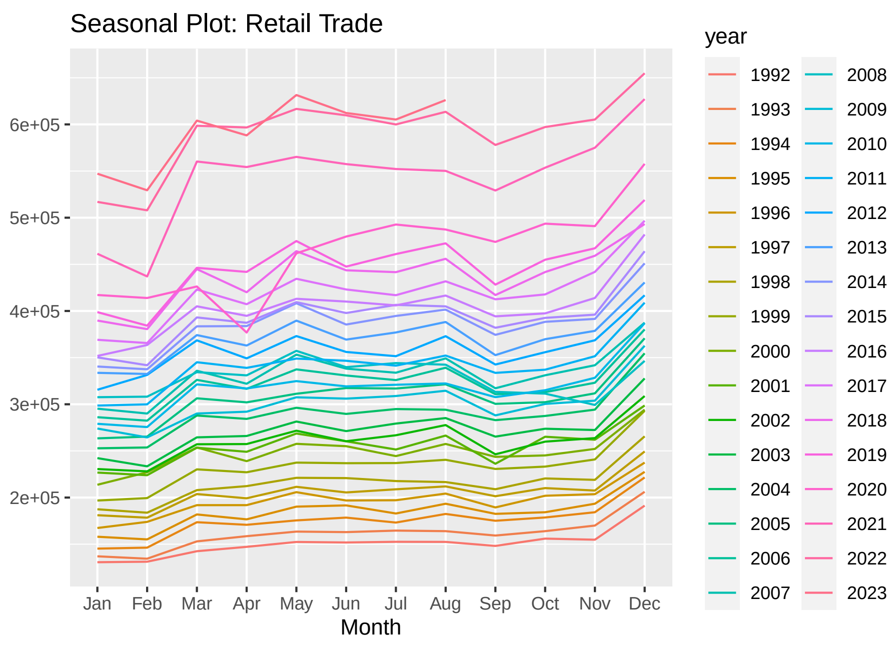
(e) Seasonal factor

```r
# Plotting the seasonal factors
plot(mltp_S, ylab = "Seasonality", main = "Seasonality")
points(mltp_S, type = "n")
text(mltp_S, c("1","2","3", "4", "5", "6", "7", "8", "9", "10", "11", "12"), col = 1:12)
```



As shown above, I labeled each spike with number, and it shows that every 12 (December), there is a huge and constant spike in retail trade.

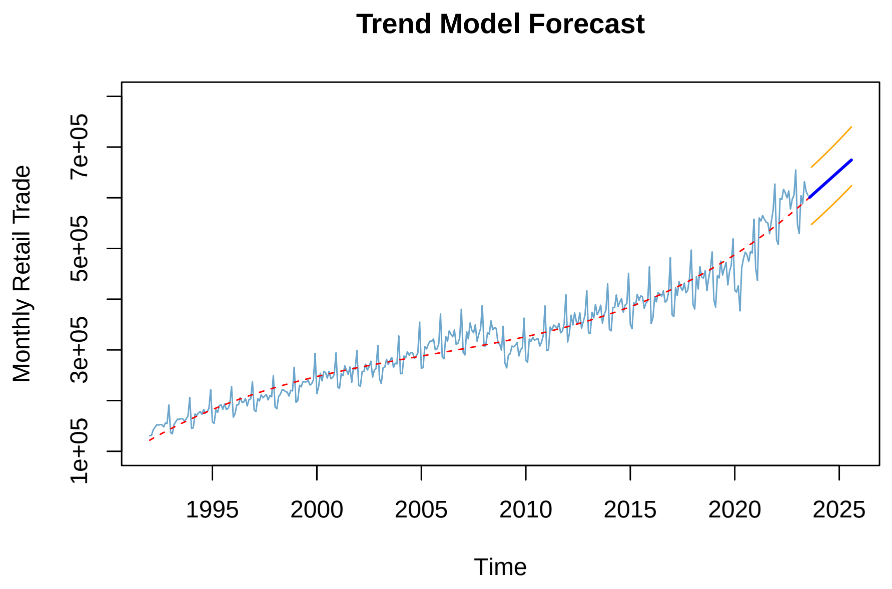
```r
ggseasonplot(ts_retail) + # there is a seasonal spike in December (ex. Christmas)
  labs(title = "Seasonal Plot: Retail Trade")
```

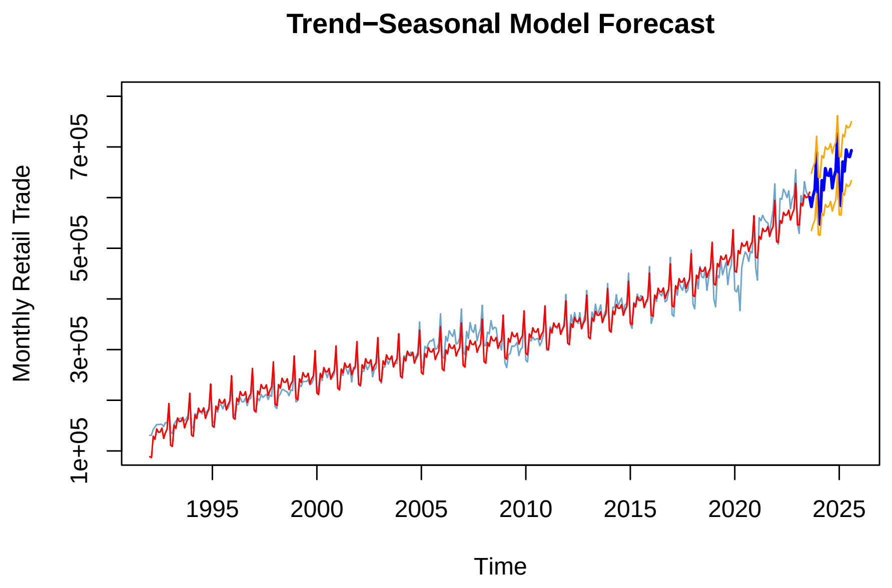


This plot also demonstrates the seasonality explored in the previous plot. Generally, there is a noticeable increase in December and a decrease in February.

(f) Trend-Seasonal Model

```
# Combining the polynomial model and the seasonality
fc_seasonal <- decompose(forecast(ts_retail, h = 24)$mean)$seasonal
fc_seasonality <- predict(decompose(ts_retail)$seasonal, h = 24)$mean


# Creating an object for a predcited seasonality
fc_sample5 <- numeric(404)
fc_sample5[381:404] <- fc_seasonal


par(mfrow = c(2,1))
# The trend model
plot(fc_sample1, ylim = c(1e+05, 8e+05), col = "skyblue3", ylab = "Monthly Retail Trade", main = "Trend
lines(fc_sample2, col = "red", lty = "dashed")
lines(fc_sample3, col = "blue", lwd = 2) # forcasted fitted line
lines(as.yearmon(new_t)[381:404],
      forecast(pmodel, new_t)$lower[381:404 ,2], col = "orange") # 95 prediction interval
lines(as.yearmon(new_t)[381:404],
      forecast(pmodel, new_t)$upper[381:404 ,2], col = "orange") # 95 prediction interval

# Trend model + Seasonality component
plot(fc_sample1, ylim = c(1e+05, 8e+05), col = "skyblue3", ylab = "Monthly Retail Trade", main = "Trend-
lines(fc_sample2 + decompose(ts_retail)$seasonal, col = "red")
lines(fc_sample3 + fc_sample5, col = "blue", lwd = 2)
lines(as.yearmon(new_t)[381:404], forecast(pmodel, new_t)$lower[381:404 ,2] + fc_seasonality, col = "ora
lines(as.yearmon(new_t)[381:404], forecast(pmodel, new_t)$upper[381:404 ,2] + fc_seasonality, col = "ora
```





Compared to the first trend model, the second model, which captures the seasonality of the original data, shows a significant improvement.

Evaluating the Model

```r
Y <- ts_retail
y_hat <- fc_sample2 + decompose(ts_retail)$seasonal
RMSE <- sqrt(sum((Y - y_hat)^2) / length(Y)) # RMSE value
RMSE
```

```
## [1] 18131.53
```

```r
summary(Y)
```

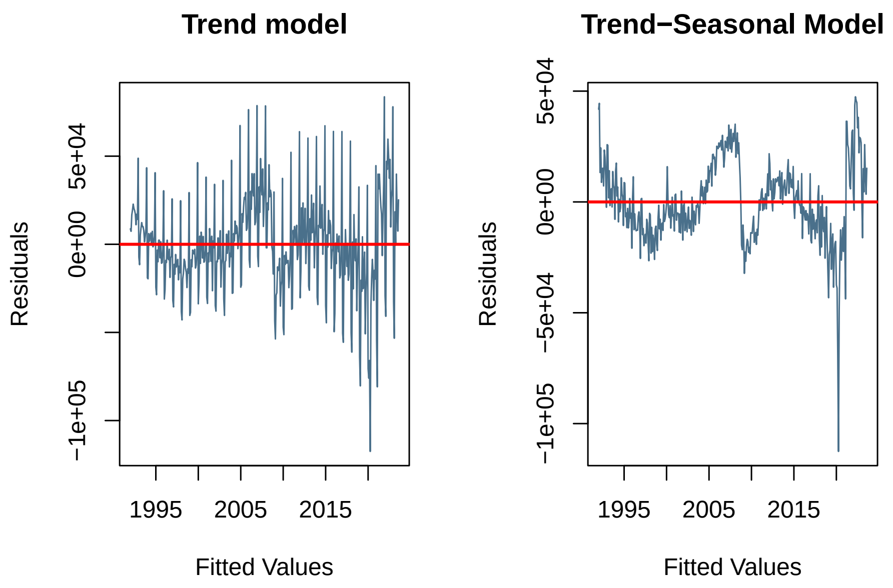
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  130683  236675  315300  325842  394709  654825
```

```r
(RMSE / median(Y))
```

```
## [1] 0.05750574
```

RMSE is informative here because it provides meaningful insights into how well the model predicts the actual values. Specifically, it shows the average residuals of the model in the same unit as the dependent variable. The output above shows that $RMSE = 18131.53$ and $RMSE/Median(RetailTrade) = 0.0575$. In other words, RMSE is 5.75% of the median value of retail trade data, which means the model is considerably performing well.

```r
# Plotting the error terms of the new forecasting model
par(mfrow = c(1,2))
plot(Y - pmodel$fitted.values,
     xlab = "Fitted Values", ylab = "Residuals", main = "Trend model", col = "skyblue4")
abline(h = 0, col = "red", lwd = 2)
plot(Y - y_hat, xlab = "Fitted Values",
     ylab = "Residuals", main = "Trend-Seasonal Model", col = "skyblue4")
abline(h = 0, col = "red", lwd = 2)
```



III. Conclusion and Future Work

As shown in the last figure, a good improvement is observed, adding the seasonality component from the original model to the polynomial trend model generated within the project. However, as can be seen from the residual plots, there is still some random pattern left over that the model fails to capture. Thus, for future reference, it is worth noting that the model and the forecast conducted based on it have a limitation at this level, and there may be more work to be done to obtain a more accurate and reliable model. For example, based on the fact that the model keeps failing to capture the decrease/increase before and after the recession period (ex. 2008 and 2020), the model could possibly improve by taking into account for other economic factors that influence the overall economic cycles.

IV. References

U.S. Census Bureau. “Advance Retail Sales: Retail Trade.” FRED, Federal Reserve Bank of St. Louis, 1 Jan. 1992, https://fred.stlouisfed.org/series/RSXFSN.

Bureau, US Census. Monthly Retail Trade - About the Advance Monthly Retail Trade Survey. https://www.census.gov/retail/marts/about_the_surveys.html. Accessed 20 Oct. 2023.