ADS 506 Module 4 Exercises

Zachariah Freitas

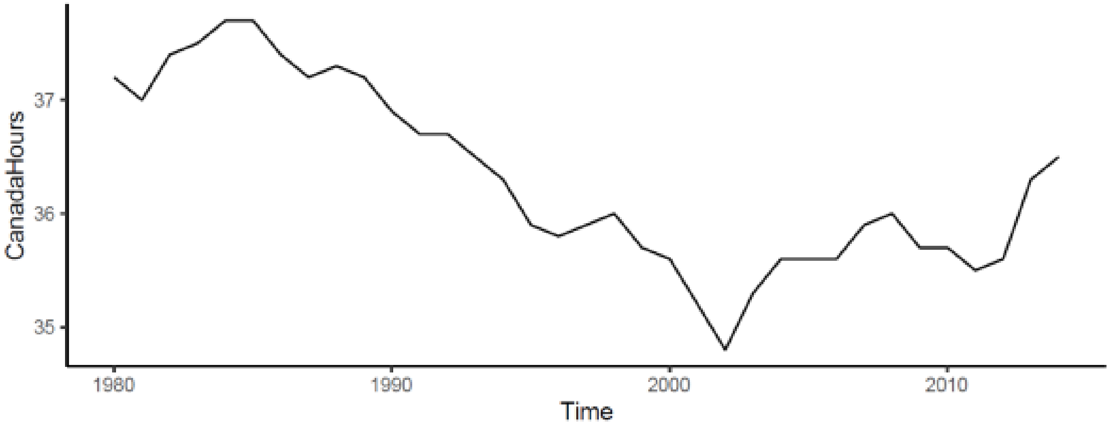
2022-11-18

ADS 506 Module 4 Exercises: Chapter 7

This assignment is due on Day 7 of the learning week. The assignment for this module is a mixture of programming and written work. Complete this entire assignment in R Markdown. You will need to include the question and number that you are answering within your submitted assignment. Once completed, you will knit your deliverable to a Word/PDF file.

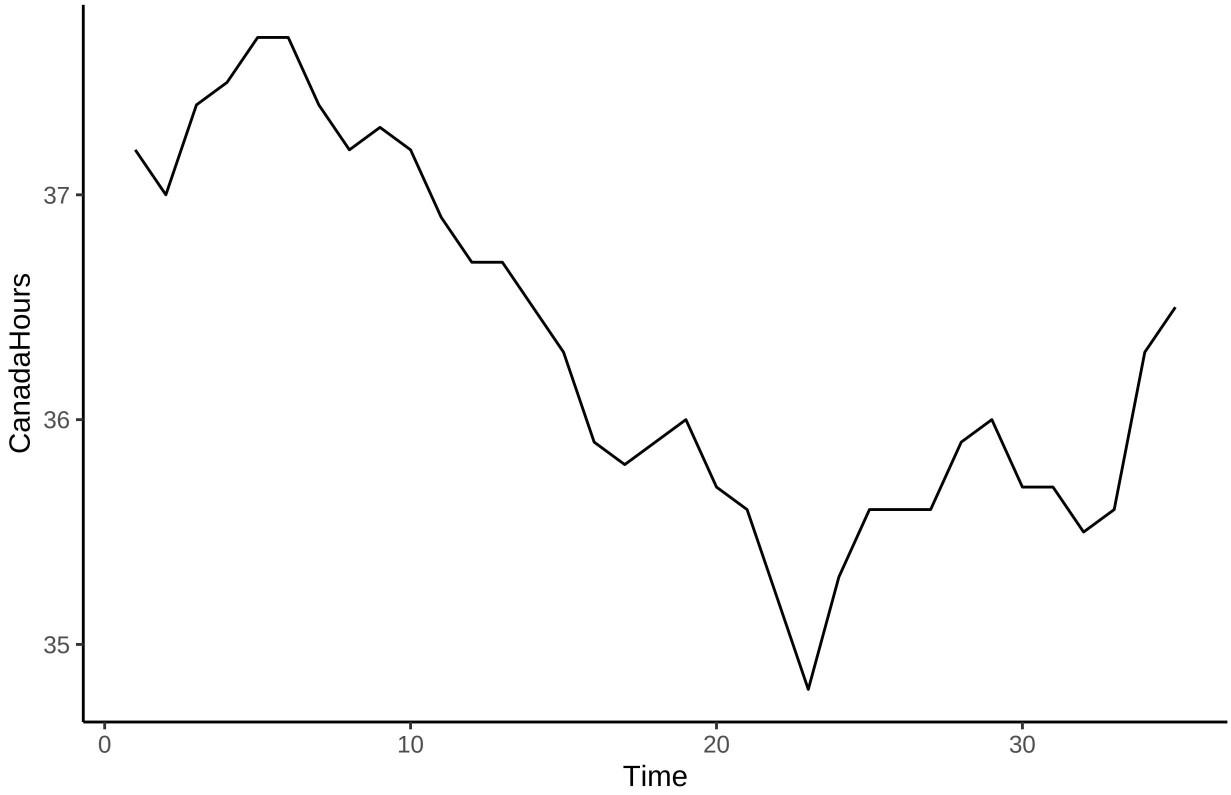
Chapter 7: Regression Models: Autocorrelation & External Info (Pages 170-178): #1, 2, & 6

1. Analysis of Canadian Manufacturing Workers Work-Hours: The time series plot in Figure 7.7 describes the average annual number of weekly hours spent by Canadian manufacturing workers. The data is available in CanadianWorkHours.csv.



```
library(fpp2)
library(zoo)
library(readr)
library(dplyr)

set.seed(506)

CanadianWorkHours <- read_csv("Data/CanadianWorkHours.csv", show_col_types = FALSE)
```

```
cwh.ts <- ts(CanadianWorkHours$Hrs_per_Wk, frequency = 1)

autoplot(cwh.ts) +
  theme_classic() +
  labs(x = "Time",
     y = "CanadaHours")
```

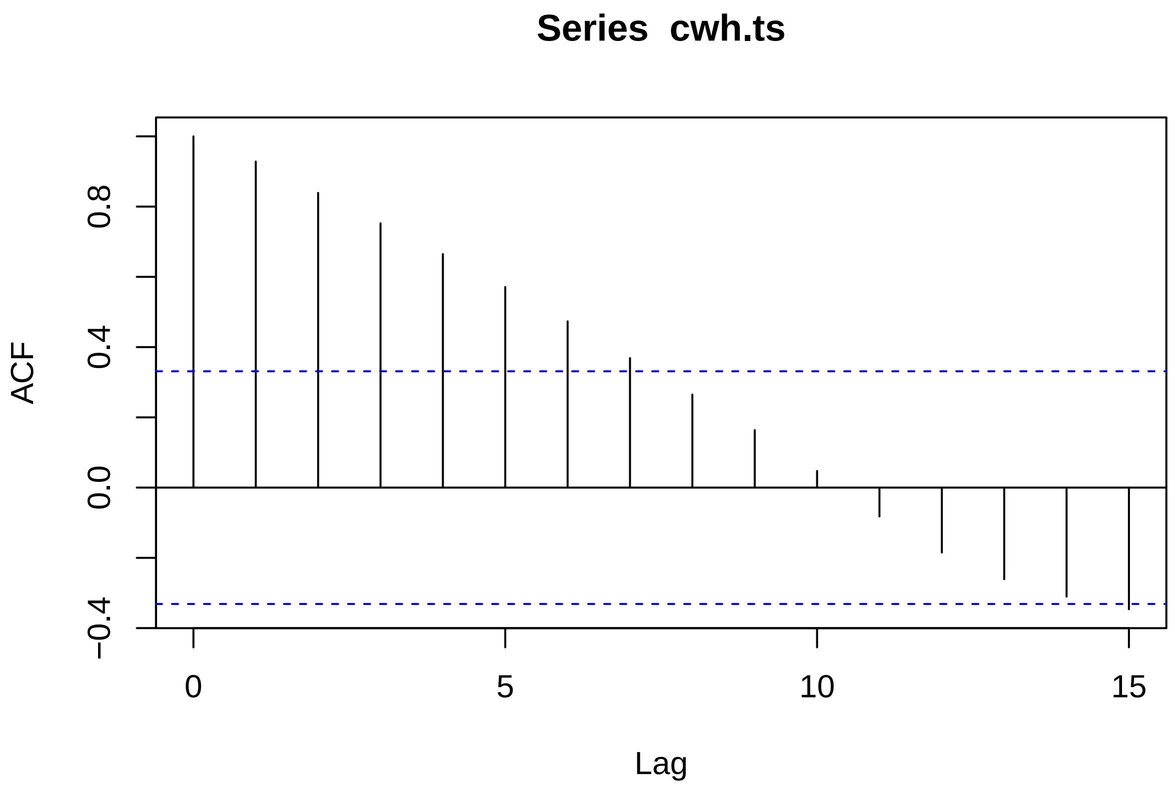


a. If we computed the autocorrelation of this series, would the lag-1 autocorrelation exhibit negative, positive, or no autocorrelation? How can you see this from the plot?

Answer I would say that we would see positive autocorrelation. We can see this because there is trending in the data. If something is going down, it continues to follow that trend.

b. Compute the autocorrelation and produce an ACF plot. Verify your answer to the previous question.

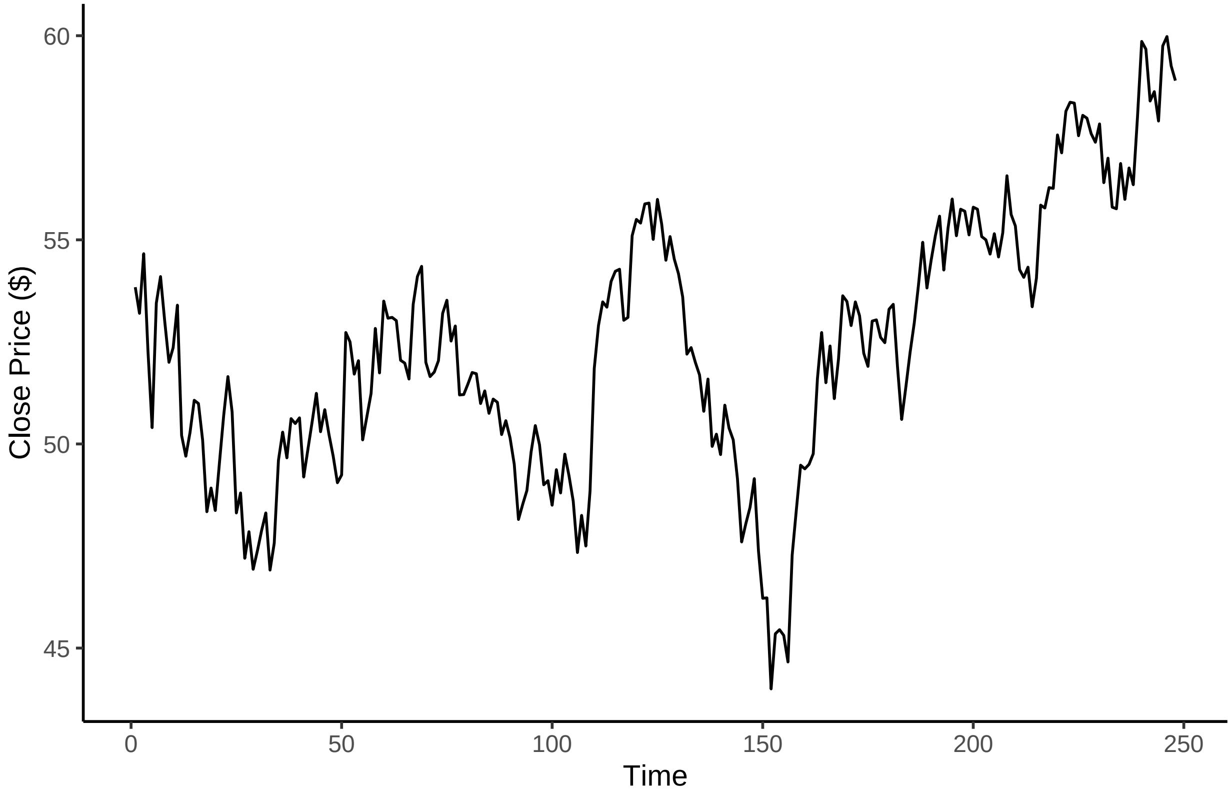
```
acf(cwh.ts)
```



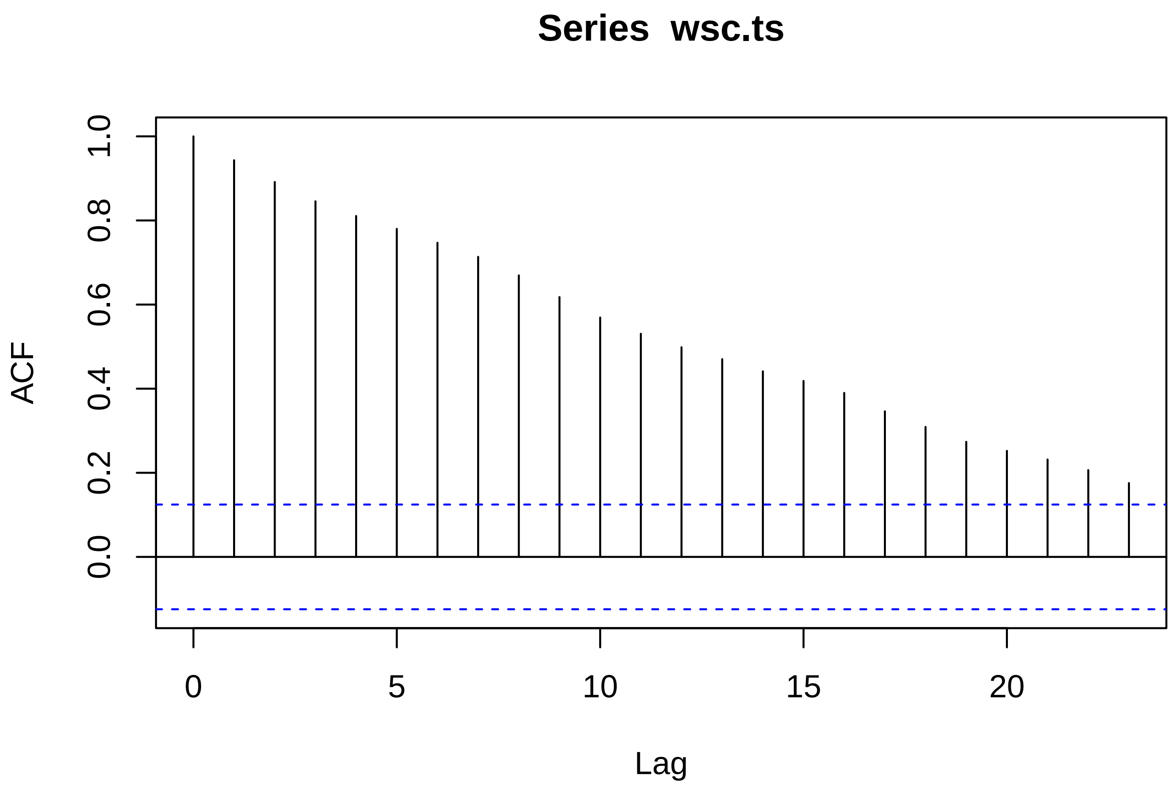
Answer The chart above shows us that we do have positive autocorrelation.

2. Forecasting Walmart Stock: Figure 7.10 shows a time plot of Wal-Mart daily closing prices between February 2001 and February 2002. The data is available at finance.yahoo.com and in WalmartStock.csv. The ACF plots of these daily closing prices and its lag-1 differenced series are in Figure 7.11. Table 7.4 shows the output from fitting an AR(1) model to the series of closing prices and to the series of differences. Use all the information to answer the following questions.

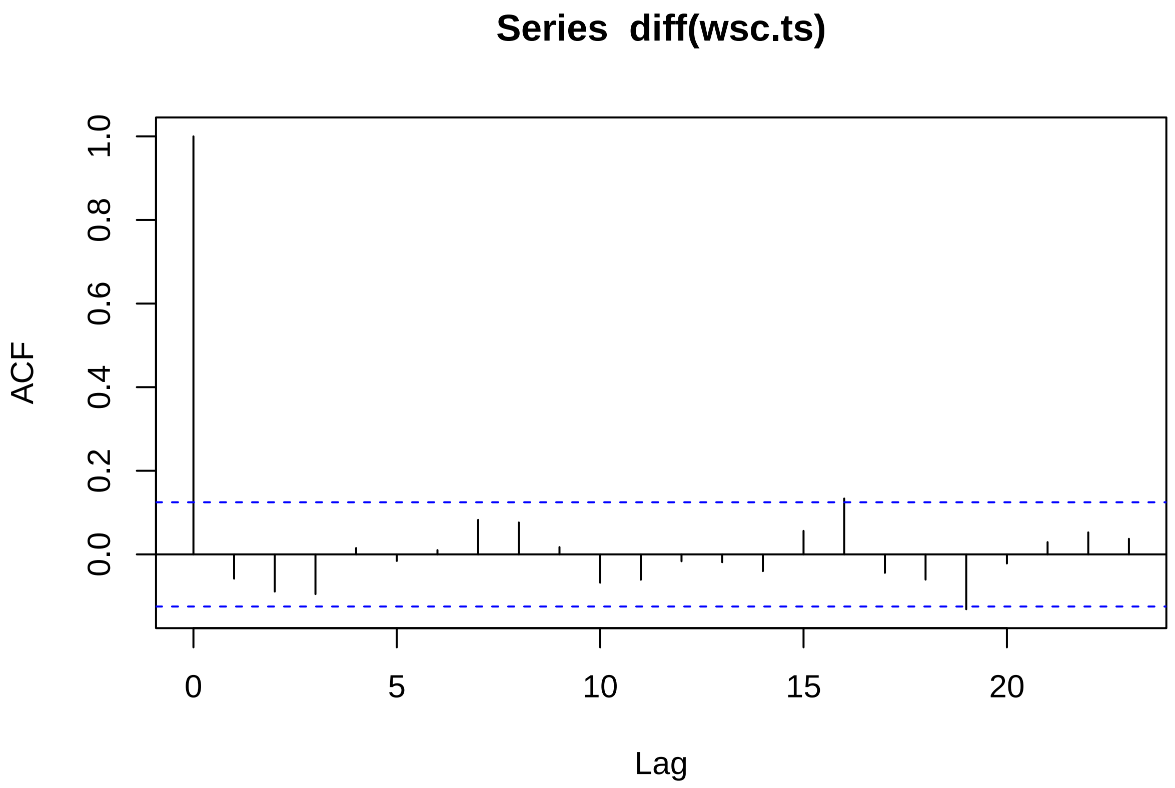
```
WalmartStock <- read_csv("Data/WalmartStock.csv",
    col_types = cols(Date = col_date(format = "%m/%d/%Y")),
    show_col_types = FALSE)

wsc.ts <- ts(WalmartStock$Close, frequency = 1)

autoplot(wsc.ts) +
  theme_classic() +
  labs(x = "Time",
     y = "Close Price ($)")
```



```
acf(wsc.ts)
```

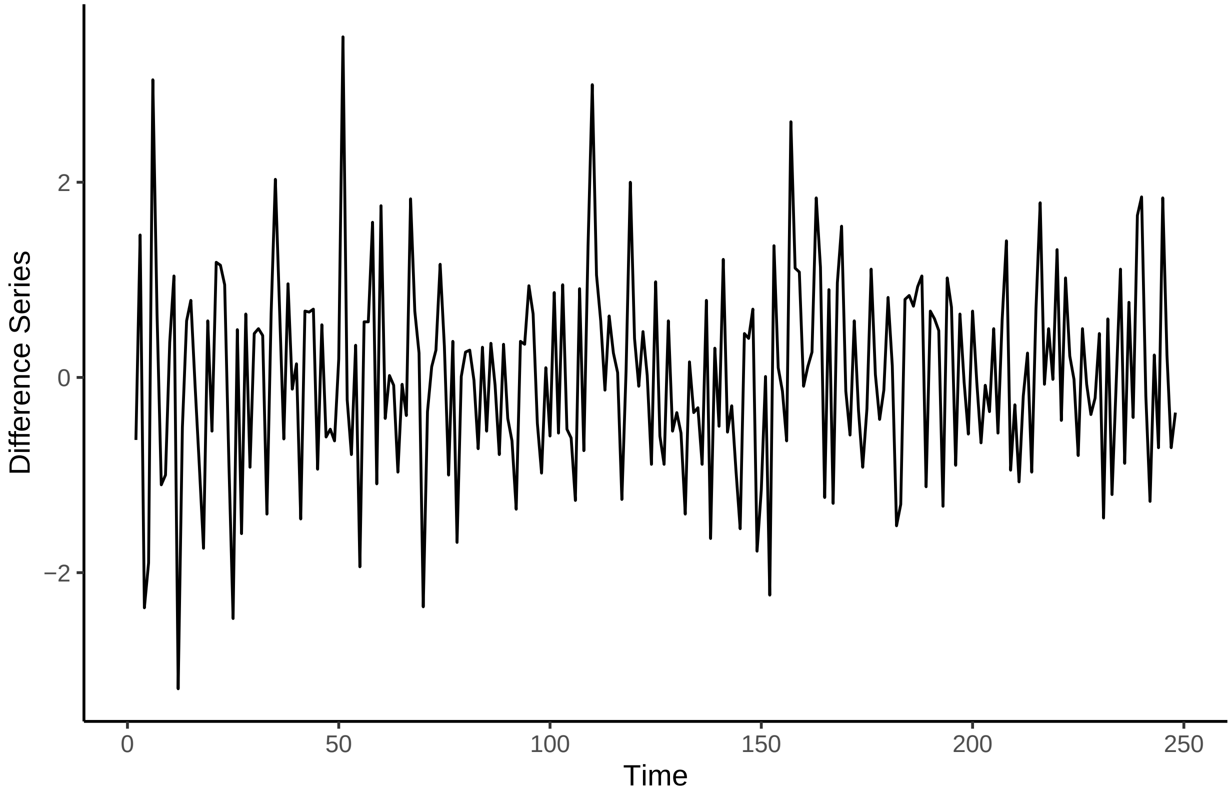


```
acf(diff(wsc.ts))
```



a. Create a time plot of the differenced series.

```
autoplot(diff(wsc.ts)) +
  theme_classic() +
  labs(x = "Time",
     y = "Difference Series")
```



b. Which of the following is/are relevant for testing whether this stock is a random walk?

☐ The autocorrelations of the closing price series.
☒ The AR(1) slope coefficient for the closing price series.
☐ The AR(1) constant coefficient for the closing price series.
☒ The autocorrelations of the differenced series.
☐ The AR(1) slope coefficient for the differenced series. - The AR(1) constant coefficient for the differenced series.

c. Recreate the AR(1) model output for the Close price series shown in the left panel of Table 7.4. Does the AR model indicate that this is a random walk? Explain how you reached your conclusion.

Answer The slope coefficient is not 1, it is statistically smaller at -0.0483 it is 16 standard deviations away from 1. Suggesting that this is not a random walk.

```
# our arima model
my_arima <- arima(wsc.ts, order = c(1, 0, 0))
summary(my_arima)
```

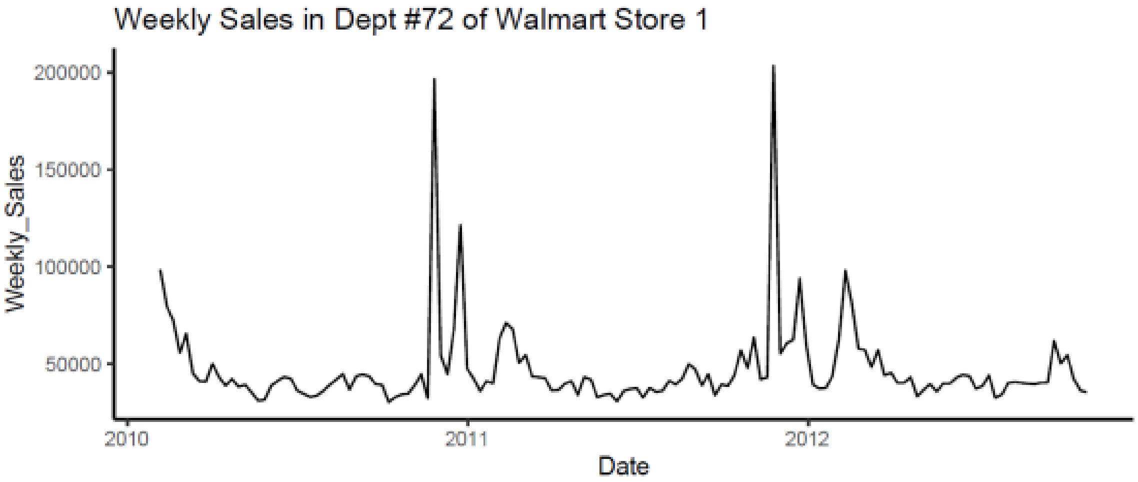
```
##
## Call:
## arima(x = wsc.ts, order = c(1, 0, 0))
##
## Coefficients:
```

```
##          ar1  intercept
##       0.9558    52.9497
## s.e.  0.0187     1.3280
##
## sigma^2 estimated as 0.9735:  log likelihood = -349.8,  aic = 705.59
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.005900455 0.9866824 0.7687247 -0.04870259 1.483133 0.9799494
##                      ACF1
## Training set -0.02979752
```

d. What are the implications of finding that a time series is a random walk? Choose the correct statement(s) below:

☒ It is impossible to obtain useful forecasts of the series.
☒ The series is random.
☒ The changes in the series from one period to the other are random.

6. Forecasting Weekly Sales at Walmart: The data in WalmartStore1Dept72.csv is a subset from a larger datasets on weekly department-wise sales at 45 Walmart stores, which were released by Walmart as part of a hiring contest hosted on kaggle.com. The file includes data on a single department at one specific store.



The fields include:

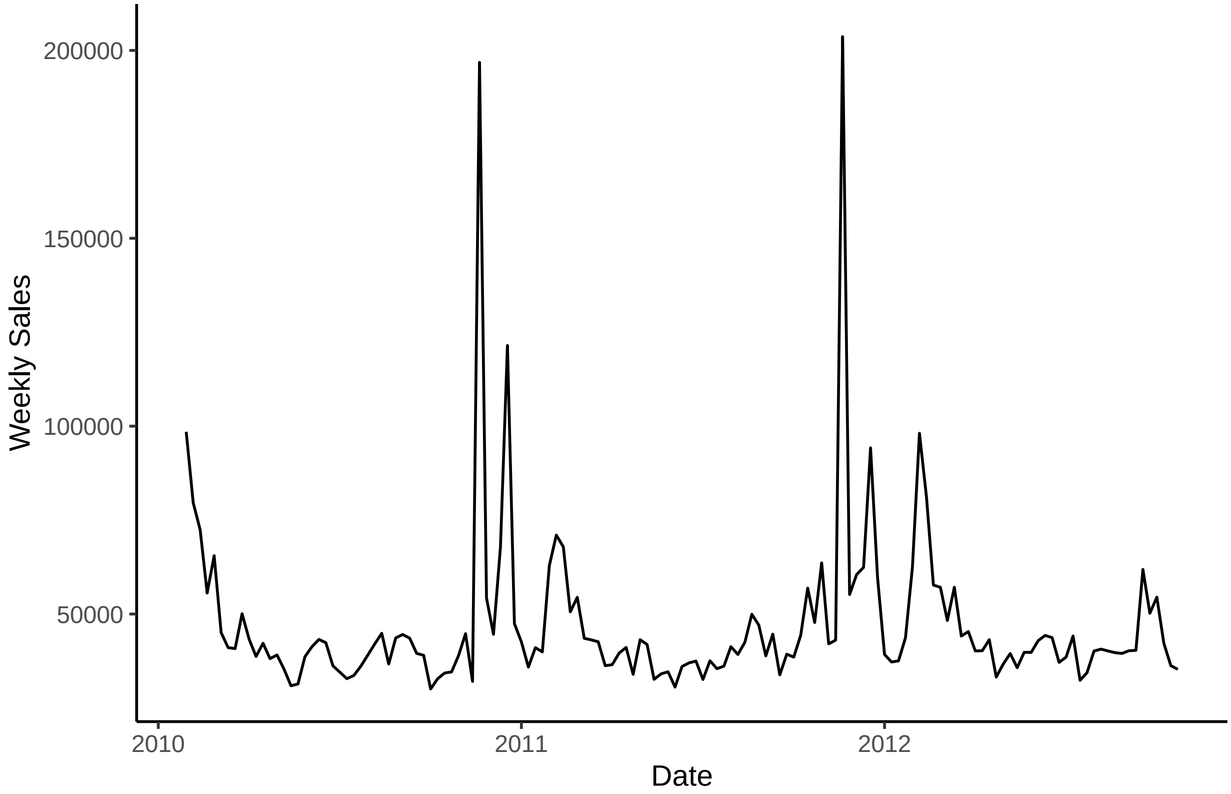
- Date - the week -
- Weekly_Sales - sales for the given department in the given store
- IsHoliday - whether the week is a special holiday week
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region
- MarkDown1-5 - anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time.
- CPI - the consumer price index
- Unemployment - the unemployment rate

Figure 7.15 shows a time plot of weekly sales in this department. We are interested in creating a forecasting model for weekly sales for the next 26 weeks.

```
WalmartStore1Dept72 <- read_csv("Data/WalmartStore1Dept72.csv",
                                col_types = cols(Date = col_date(format = "%m/%d/%Y"),
                                                 IsHoliday = col_logical()),
                                show_col_types = FALSE)


wsd72.ts <- ts(WalmartStore1Dept72$Weekly_Sales, start = c(2010, 5), frequency = 52)
```

a. Recreate the time plot of the weekly sales data. Which systematic patterns appear in this series? **Answer** I see annual patterns where sales appear greatest around November, December, and February.

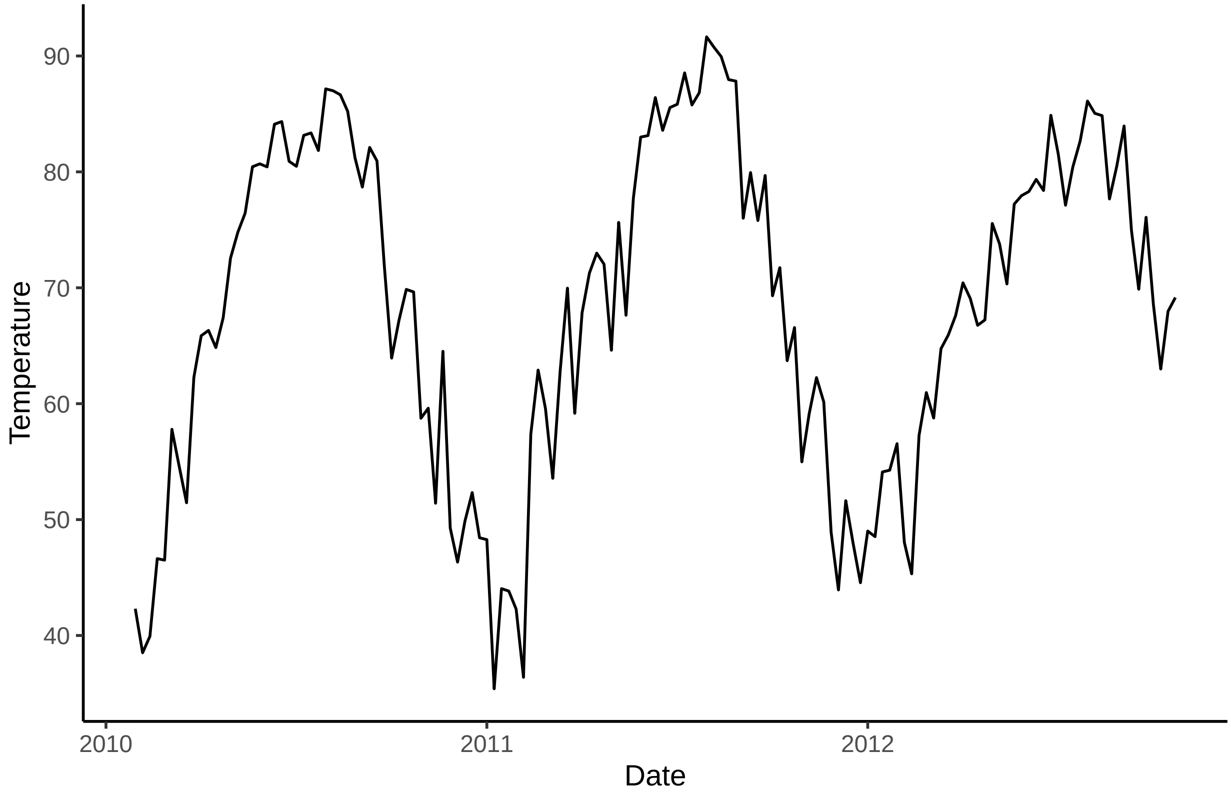
```
# visualize the data to see if there is seasonality, trending or cycles.
autoplot(wsd72.ts) +
  theme_classic() +
  labs(x = "Date",
     y = "Weekly Sales")
```



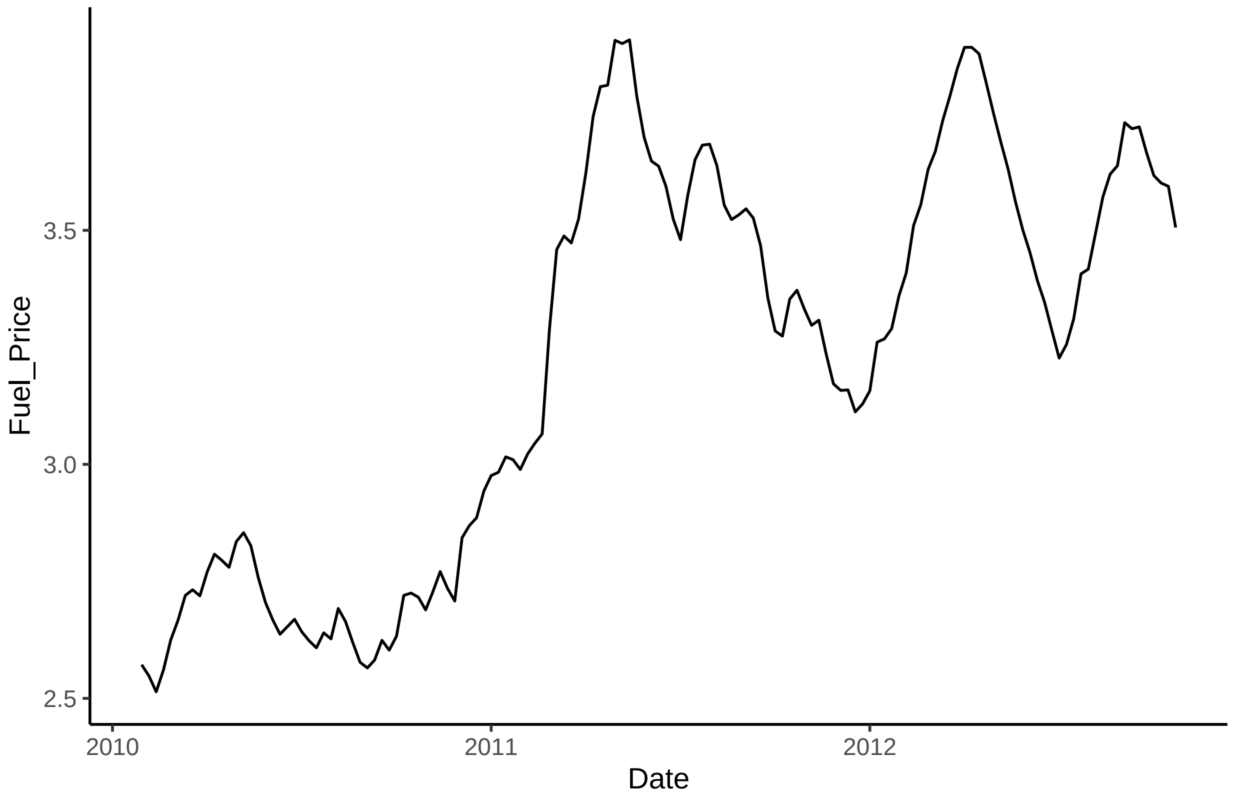
b. Create time plots of the other numerical series (Temperature, Fuel_Price, CPI, and Unemployment). Also create scatter plots of the sales series against each of these four series (each point in the scatter plot will be a week). From the charts, which of the four series would potentially be useful as external predictors in a regression model for forecasting sales?

Answer I think Temperature and maybe Fuel_Price will useful external predictor in a regression model.

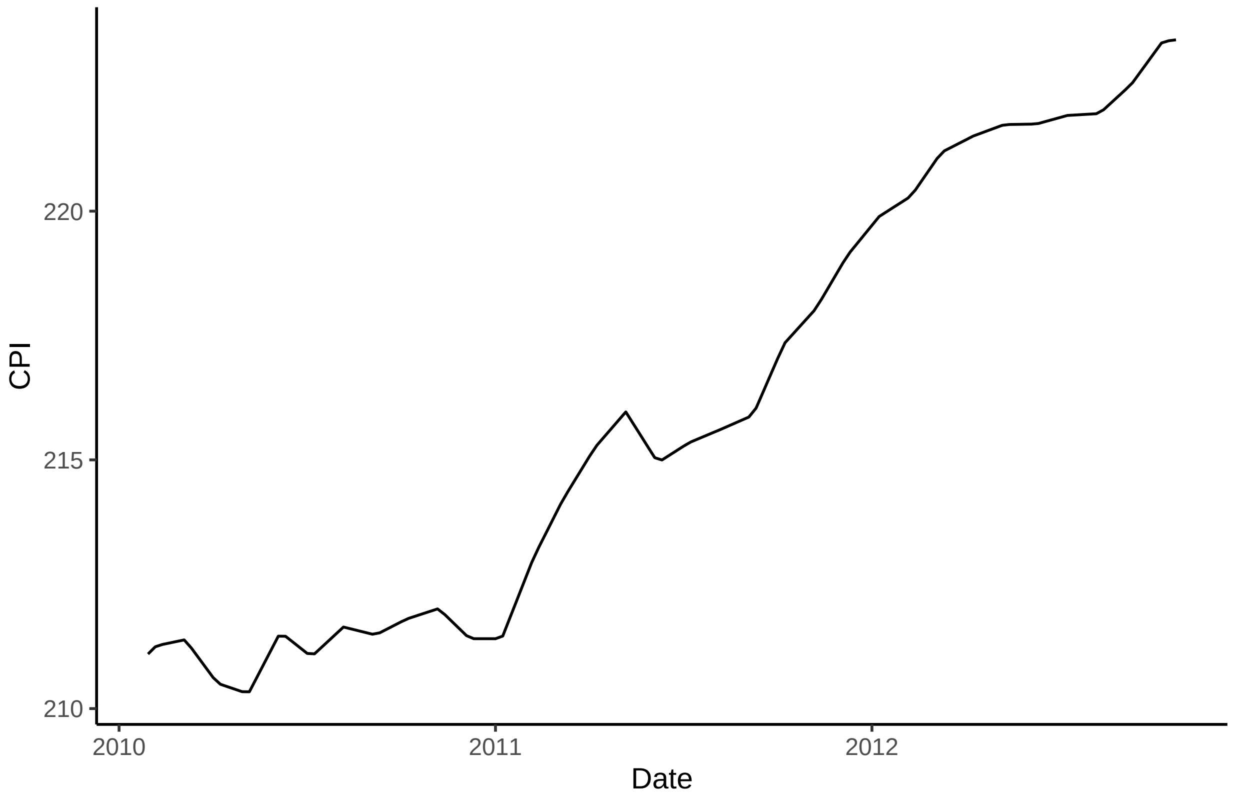
```
temp_wsd72.ts <- ts(WalmartStore1Dept72$Temperature, start = c(2010, 5), frequency = 52)
fp_wsd72.ts <- ts(WalmartStore1Dept72$Fuel_Price, start = c(2010, 5), frequency = 52)
cpi_wsd72.ts <- ts(WalmartStore1Dept72$CPI, start = c(2010, 5), frequency = 52)
unemp_wsd72.ts <- ts(WalmartStore1Dept72$Unemployment, start = c(2010, 5), frequency = 52)

autoplot(temp_wsd72.ts) +
  theme_classic() +
  labs(x = "Date",
     y = "Temperature")
```



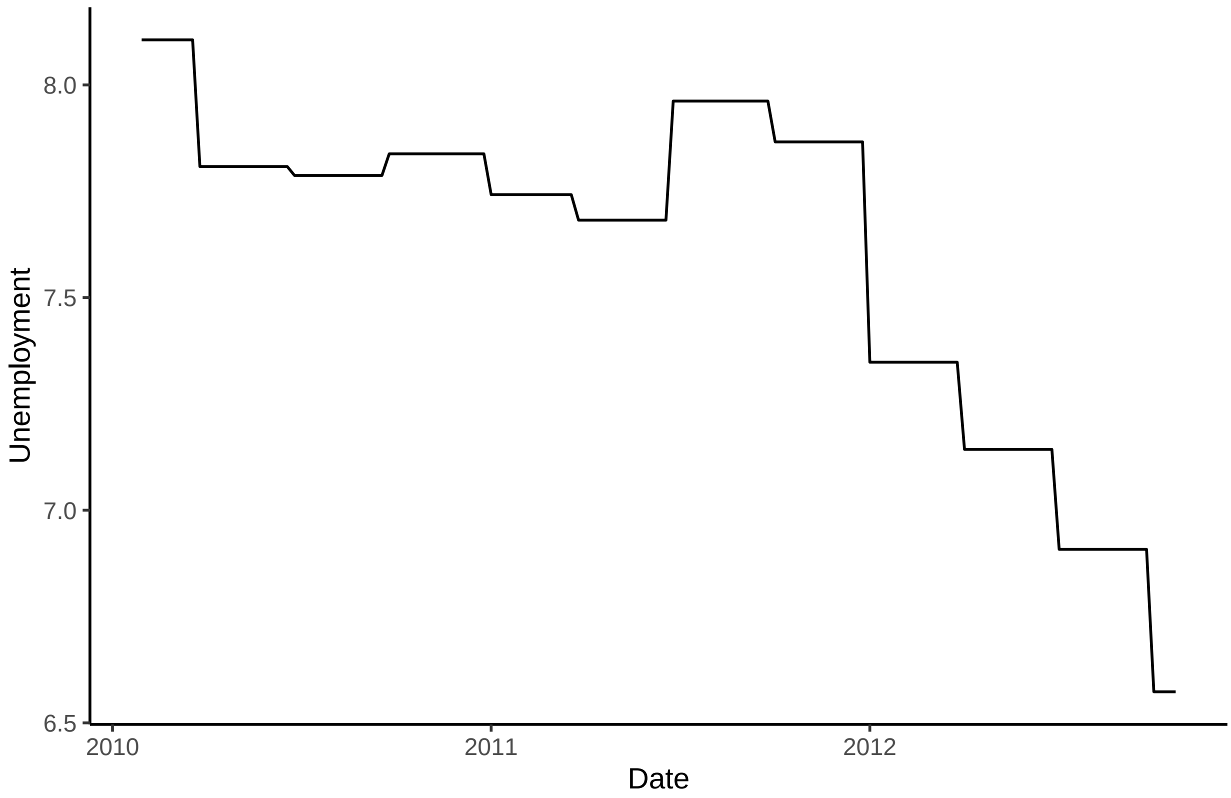
```
autoplot(fp_wsd72.ts) +
  theme_classic() +
  labs(x = "Date",
     y = "Fuel_Price")
```



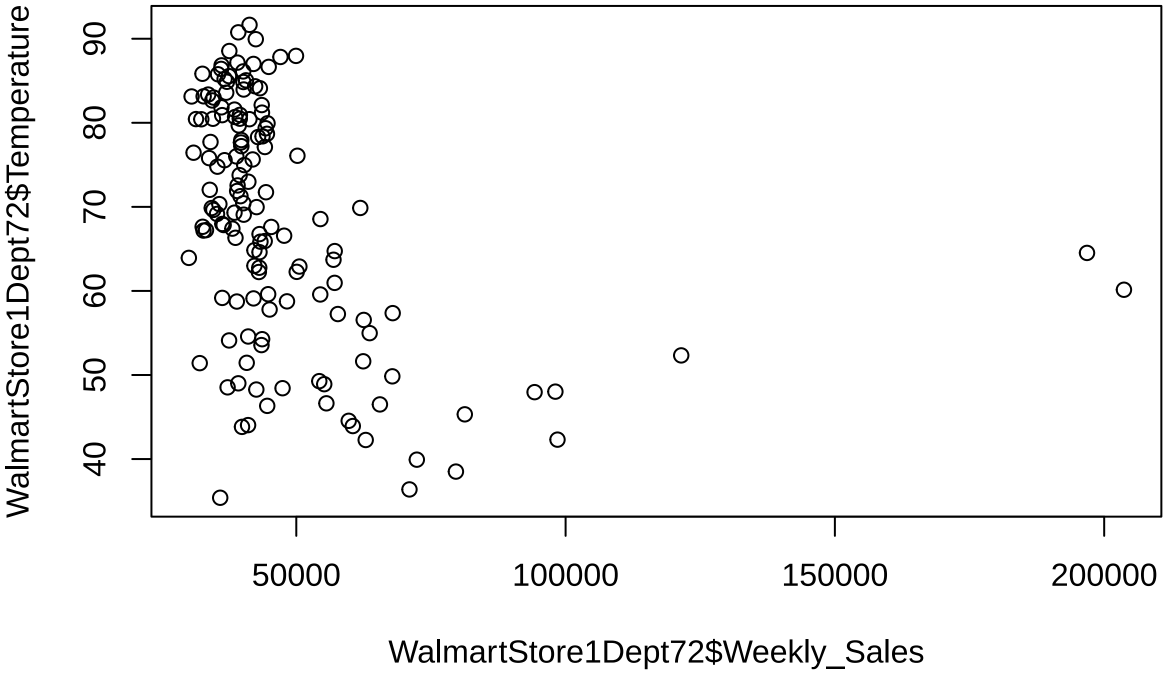
```
autoplot(cpi_wsd72.ts) +
  theme_classic() +
  labs(x = "Date",
     y = "CPI")
```



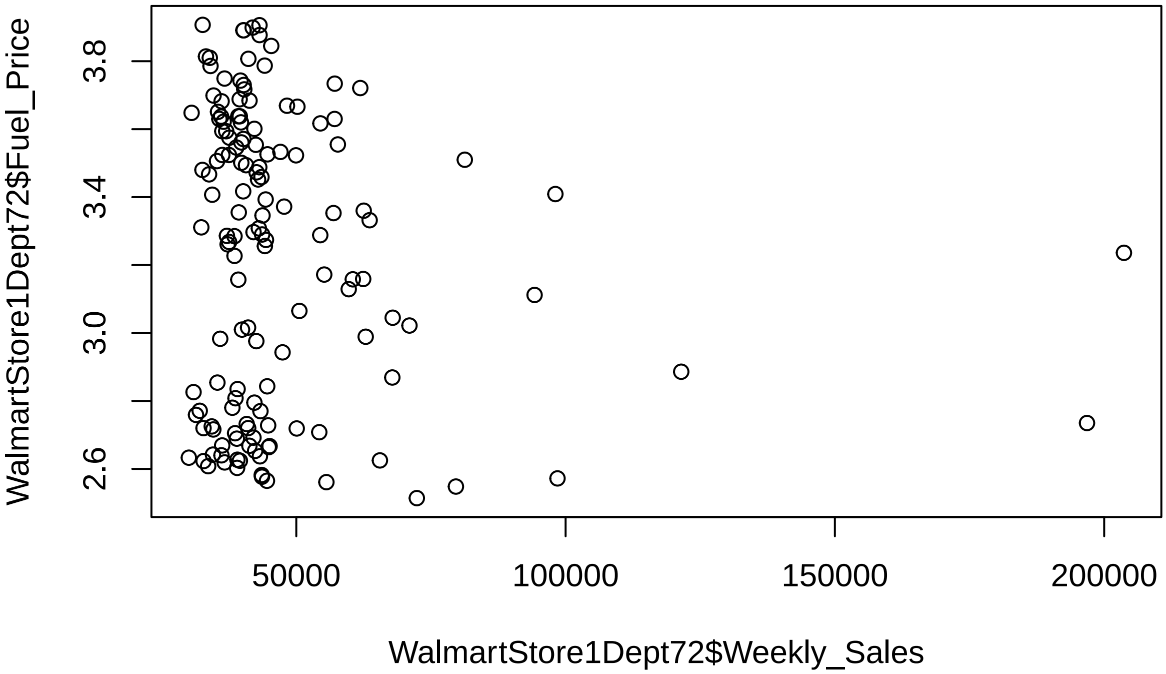
```
autoplot(unemp_wsd72.ts) +
  theme_classic() +
  labs(x = "Date",
     y = "Unemployment")
```



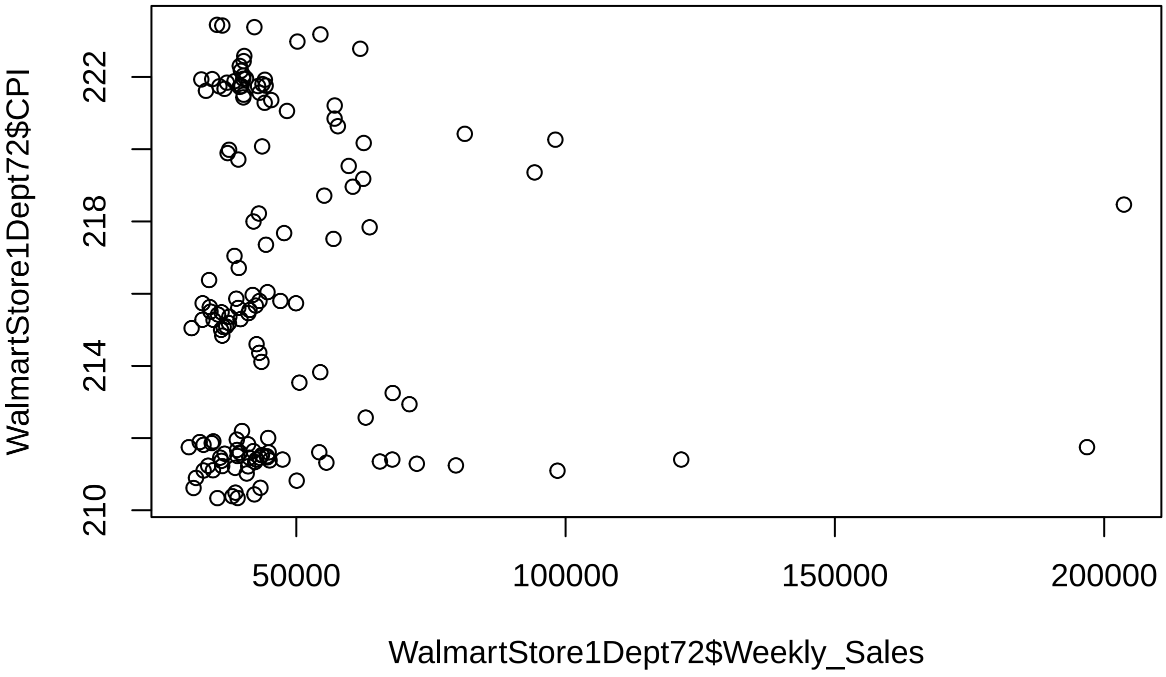
```
plot(WalmartStore1Dept72$Weekly_Sales,WalmartStore1Dept72$Temperature)
```



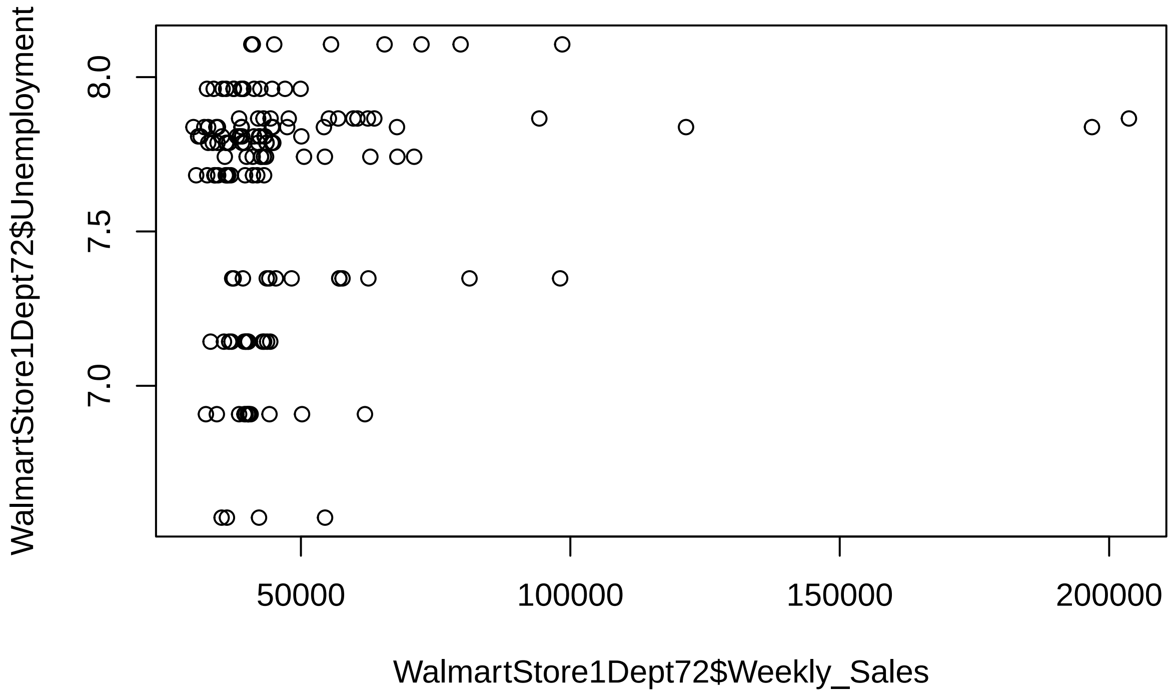
```
plot(WalmartStore1Dept72$Weekly_Sales,WalmartStore1Dept72$Fuel_Price)
```



```
plot(WalmartStore1Dept72$Weekly_Sales,WalmartStore1Dept72$CPI)
```



```
plot(WalmartStore1Dept72$Weekly_Sales,WalmartStore1Dept72$Unemployment)
```



The following questions are not in your textbook. You will need to also complete these programming questions in R Markdown.

c. Fit an ARIMA model with 1 lag and external predictors for Weekly_Sales that treats Nov 4, 2011 to Oct 26, 2012 as the training period, and the next 26 weeks as the validation period. Compute the RMSE for the training period.

```
training_set <- WalmartStore1Dept72 %>% filter(Date >= as.Date('2011-11-04') &
                                               Date <= as.Date('2012-10-26') )

outcome_v <- training_set$Weekly_Sales
predictors <- as.matrix(training_set %>% select(Temperature, Fuel_Price, CPI, Unemployment))


my_arima_reg_model <- Arima(outcome_v, order = c(1,0,0), xreg = predictors)
summary(my_arima_reg_model)
```

```
## Series: outcome_v
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##           ar1   intercept  Temperature  Fuel_Price       CPI  Unemployment
##       -0.0532  -656817.5    -339.2327   -7929.292  2422.295      30755.24
## s.e.   0.1376  1880410.7     340.9932   17011.569  7804.623      27754.72
##
```

```
## sigma^2 = 568917600:  log likelihood = -594.74
## AIC=1203.48   AICc=1206.02   BIC=1217.14
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -4.164588 22433.75 11907.24 -7.565639 20.68684 0.878544
##                      ACF1
## Training set -0.008048997
```

```
# RMSE
sqrt(mean(my_arima_reg_model$residuals^2))
```

```
## [1] 22433.75
```

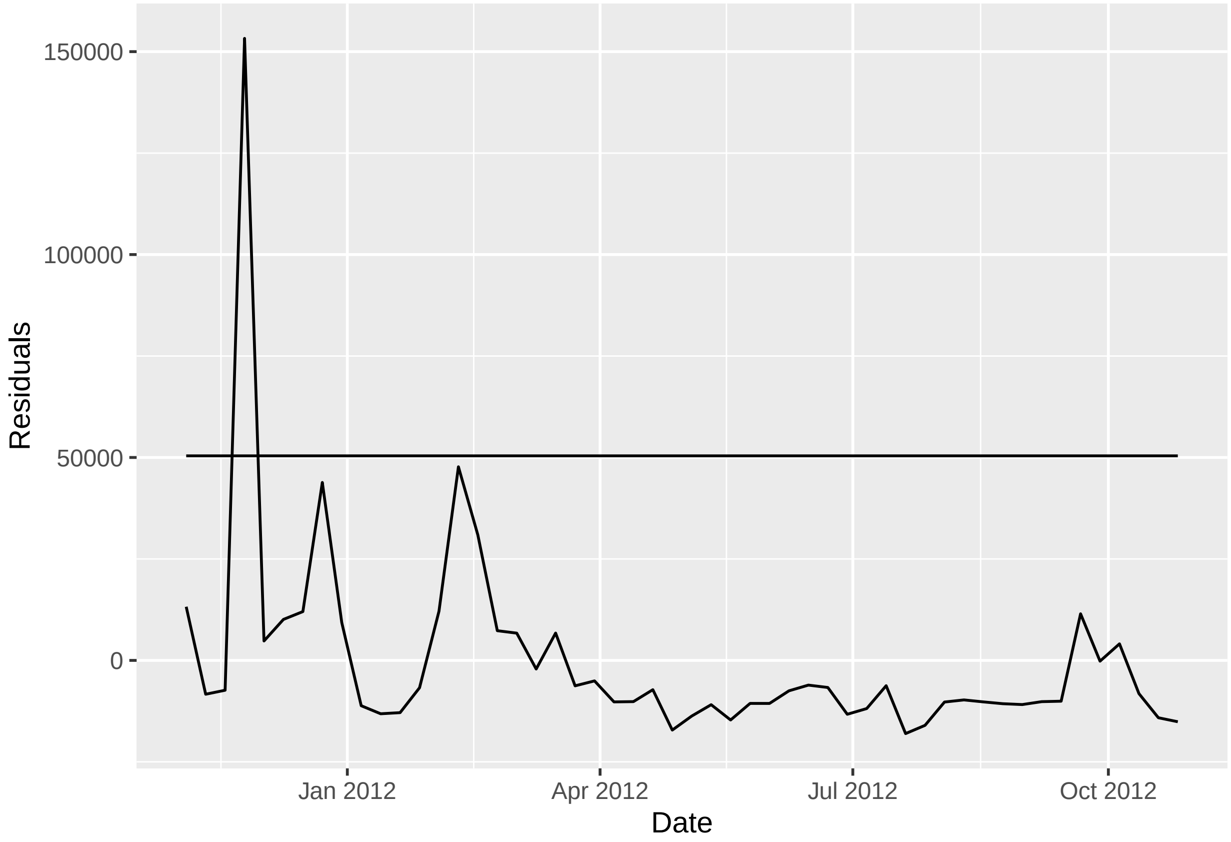
d. Create a mean forecasts for the validation period. Create a time plot of these forecasts and a plot of the forecast errors series. Compute the RMSE for the training period.

Answer I'm assuming you really meant training period rather than validation period because we have no actual values to compare to in the validation set.

```
# Mean Forecast
mf <- mean(training_set$Weekly_Sales)

# RMSE of Mean Forecast
sqrt(mean((training_set %>% mutate(resids = Weekly_Sales - mf) %>% pull(resids))^2))
```

```
## [1] 25544.33
```

```
training_set %>% mutate(Residuals = Weekly_Sales - mf,
                        Mean_Forecast = mf) %>%
  ggplot(aes(x=Date)) +
  geom_line(aes(y=Residuals)) +
  geom_line(aes(y=Mean_Forecast))
```



e. Compare the performance of the ARIMA model to the mean forecasts. Which one performs better?

Answer The ARIMA model performs better than the mean model.

```
# RMSE of ARIMA Model Forecast
sqrt(mean(my_arima_reg_model$residuals^2))
```

```
## [1] 22433.75
```

```
# RMSE of Mean Forecast
sqrt(mean((training_set %>% mutate(resids = Weekly_Sales - mf) %>% pull(resids))^2))
```

```
## [1] 25544.33
```

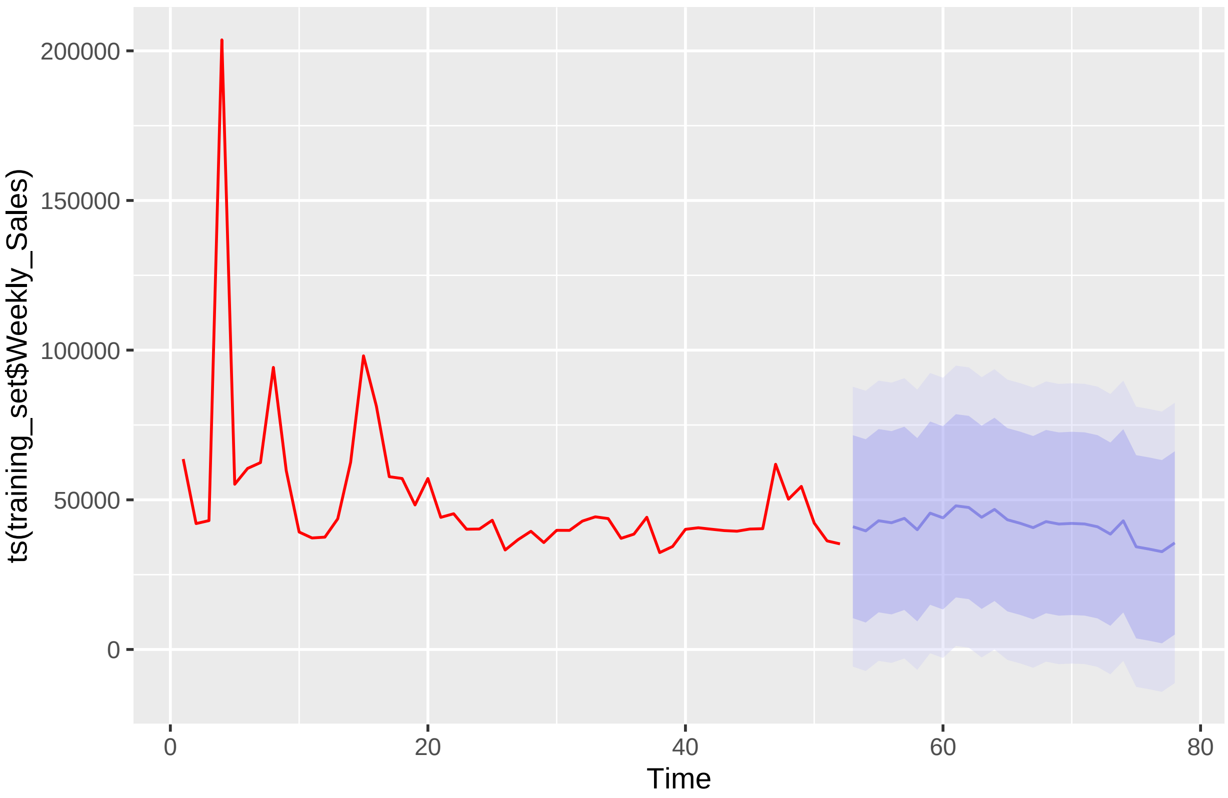
f. Plot the ARIMA model forecasted values. Use WalmartStore1Dept72_validation.csv for your regression model data.

Answer See below:

```
# forecasting

WalmartStore1Dept72_valid <- read_csv("Data/WalmartStore1Dept72_validation.csv",
                                      col_types = cols(Date = col_date(format = "%m/%d/%Y"),
                                                       IsHoliday = col_logical()),
                                      show_col_types = FALSE)


my_predictors <- as.matrix(WalmartStore1Dept72_valid %>%
                             select(Temperature, Fuel_Price, CPI, Unemployment))


my_forecast <- forecast(my_arima_reg_model, xreg = my_predictors)

my_forecast
```

```
##    Point Forecast     Lo 80    Hi 80        Lo 95    Hi 95
## 53       41016.76 10449.203 71584.32  -5732.28598 87765.81
## 54       39622.02  9011.175 70232.87  -7193.22966 86437.28
## 55       43012.73 12401.757 73623.70  -3802.71181 89828.17
## 56       42333.74 11722.773 72944.72  -4481.69623 89149.19
## 57       43803.86 13192.888 74414.83  -3011.58190 90619.30
## 58       39997.84  9386.872 70608.81  -6817.59770 86813.28
## 59       45545.77 14934.799 76156.74  -1269.67032 92361.21
## 60       43974.15 13363.174 74585.12  -2841.29586 90789.59
## 61       47996.49 17385.517 78607.46   1181.04770 94811.93
## 62       47430.09 16819.118 78041.06    614.64860 94245.53
## 63       44157.93 13546.960 74768.90  -2657.50903 90973.37
## 64       46798.82 16187.849 77409.79    -16.62005 93614.26
## 65       43332.50 12721.527 73943.47  -3482.94208 90147.94
## 66       42148.84 11537.867 72759.81  -4666.60252 88964.28
## 67       40705.20 10094.227 71316.17  -6110.24204 87520.64
## 68       42722.69 12111.721 73333.66  -4092.74820 89538.13
## 69       41904.56 11293.585 72515.53  -4910.88475 88720.00
## 70       42109.88 11498.911 72720.85  -4705.55865 88925.32
## 71       41929.67 11318.696 72540.64  -4885.77343 88745.11
## 72       40989.22 10378.251 71600.19  -5826.21886 87804.66
## 73       38534.60  7923.632 69145.57  -8280.83764 85350.04
## 74       42990.07 12379.100 73601.04  -3825.36941 89805.51
## 75       34310.86  3699.893 64921.84 -12504.57669 81126.31
## 76       33540.57  2929.595 64151.54 -13274.87441 80356.01
## 77       32673.21  2062.241 63284.18 -14142.22890 79488.65
## 78       35612.57  5001.598 66223.54 -11202.87160 82428.01
```

```
autoplot(ts(training_set$Weekly_Sales), color = 'red') +
  autolayer(my_forecast, alpha = .3)
```


200000
150000
100000
50000
0
ts(training_set$Weekly_Sales)
0
20
40
60
80
Time