Chapter 5
Benchmark Forecasting Methods

1 Introduction

In this chapter, we turn our focus to producing forecasts. Throughout the remainder of the course, I will refer to "statistical models," "forecasting models," or "forecasting methods." These all refer to the same thing - a statistical model used to produce forecasts. We will start with some of the simplest forecasting methods, most of which consist of either zero or one parameter. The models we will consider in this chapter will serve as "benchmarks" — decent forecasting models that we hope to improve on with more sophisticated techniques. These benchmark methods are:

1. The intercept model.
2. The random walk, or "naive" model.
3. The seasonal naive model.

2 Benchmark Methods

We will analyze some properties of each of these forecasting models, and discuss when each can or should be used. For each, we will assume that we have made transformations to remove the trend of the series, so that our series is at least trend-stationary (i.e. it may still have seasonality or a unit root, but it does not have a trend). Furthermore, we will assume that our loss function is squared error loss. Recall that under squared error loss, the optimal forecast is given by some type of mean.

2.1 The Intercept Model

We first encountered the intercept model in the last chapter. The intercept model is a one parameter model, given by:

$$
\begin{aligned}
y_t &= b + \varepsilon_t \\
\varepsilon_t &\sim \mathcal{N}(0, \sigma)
\end{aligned}
$$

In order to use this model for forecasting, we need to either know or estimate the value of b. If we are given data, Y, from time $t = 1$ through time $t = T$, then we can estimate b many different ways: it could be the mean, the median, the average of only the largest and smallest observations, etc. However, our goal should be to form the optimal point-forecast for any future time period, $T + h$. Since we are using squared error loss, the optimal point-forecast is given by the mean, or the *expected value*.

Suppose we want to form the forecast for time period $T + h$. Since we have an equation for y at any point in time, t, let's plug in $T+h$ in for t in the intercept model equation given above:

$$
\begin{aligned}
y_t &= b + \varepsilon_t \\
y_{T+h} &= b + \varepsilon_{T+h}
\end{aligned}
$$

Next, since we know the optimal forecast is given by the expected value, take the expected value of each side, conditional on observing the data set Y:

$$
\begin{aligned}
E(y_{T+h}|Y) &= E(b + \varepsilon_{T+h}|Y) \\
E(y_{T+h}|Y) &= E(b|Y) + E(\varepsilon_{T+h}|Y) \\
E(y_{T+h}|Y) &= E(b|Y) \\
\hat{y}_{T+h|T} &= E(b|Y)
\end{aligned}
$$

We were able to go from the second to the third line above due to the fact that ε_{T+h} is unknown, and ε_t has mean zero. In other words, on average, we expect the error term to be zero. Under squared error loss, the optimal estimate of b is the mean of the series, Y: $E(b|Y) = \frac{1}{T}\sum_{t=1}^{T} y_t$

Finally, note that under this model, the optimal forecast does NOT depend on the forecast horizon, h:

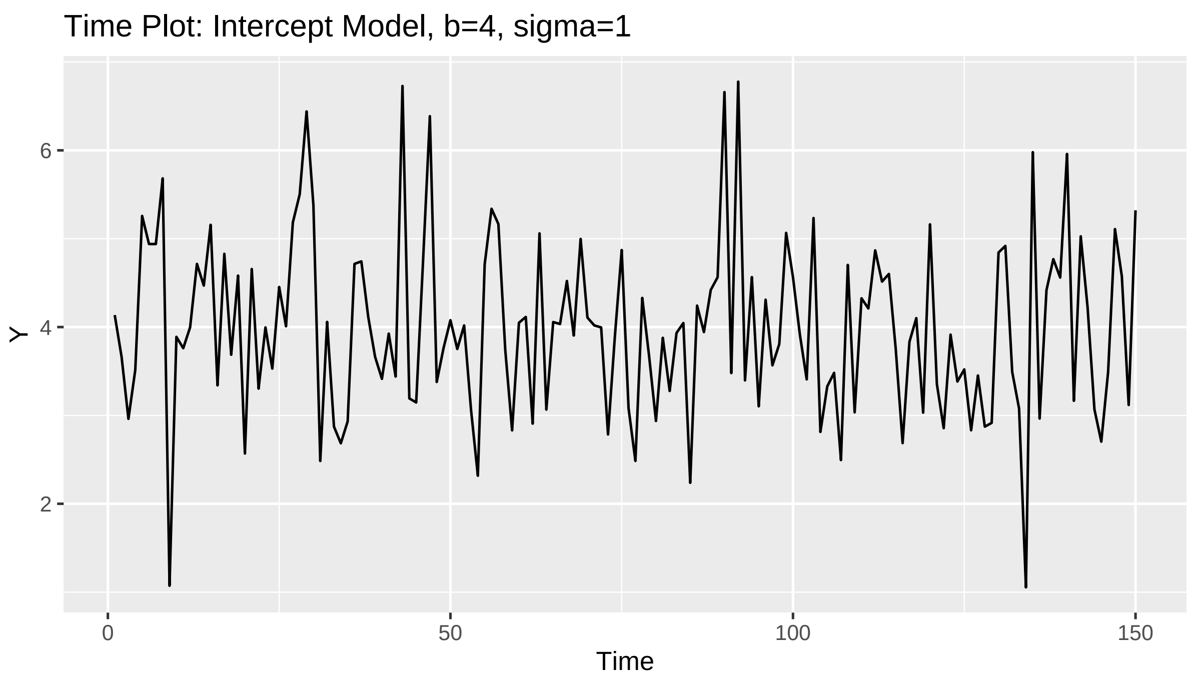
$$\hat{y}_{T+h|T} = E(b|Y)$$
$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

since there is no term involving h on the right hand side of the model. In summary, regardless of how far in the future we are forecasting, if we believe that our data can be well approximated by the intercept model and we are using squared error loss, then our optimal forecast is simply the mean of the series.

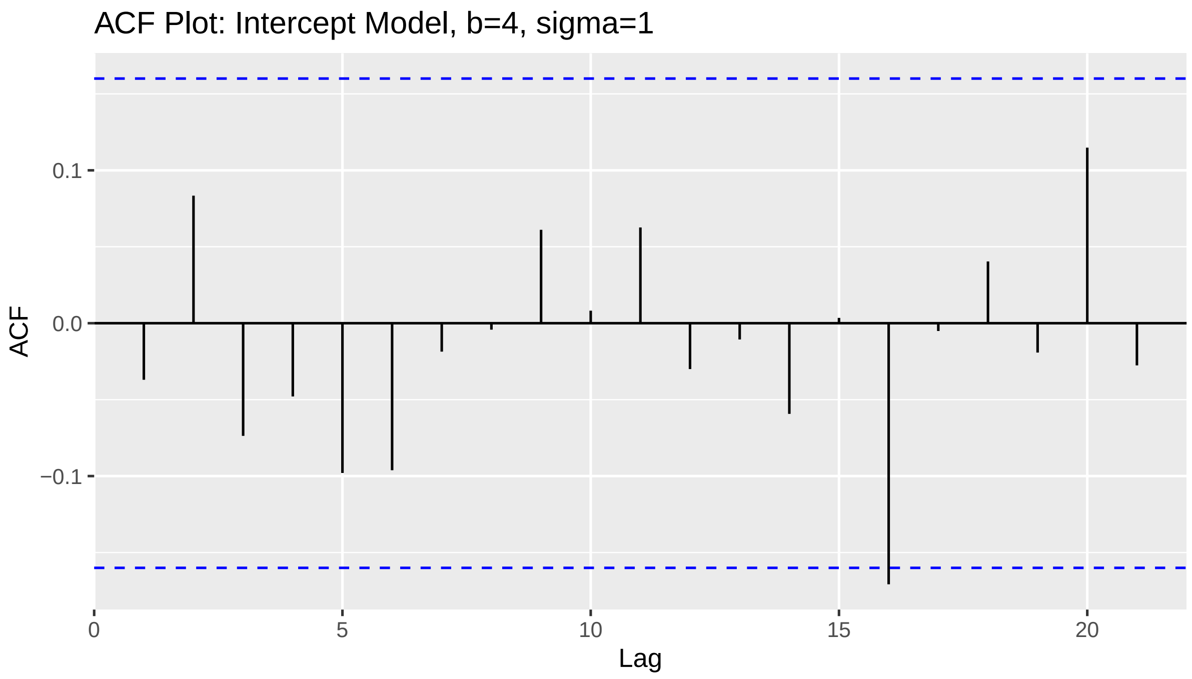
2.1.1 When to Use

So when should you use the intercept model? Before fitting the model or using it to forecast, there is no way to know for sure. However, the data should certainly be stationary — there should be no seasonality, no trend, and no unit root. The data should also have no (or only very slight) autocorrelation. You can use a time plot, an ACF plot, a seasonal plot, and a seasonal subseries plot to try and determine if the data is stationary. Below are a few examples of data that was randomly generated according to an intercept model. If the time plot and ACF plot of your data look like this, the intercept model is likely appropriate.

```r
  set.seed(1988)
  T <- 150 # Number of time periods
  b <- 4.0 # Value of B0
  Y <- b + rnorm(T,0,1) # Create data with error distributed N(0,1)
  Y <- ts(Y) # Declare as time-series
autoplot(Y) + ggtitle("Time Plot: Intercept Model, b=4, sigma=1")
```

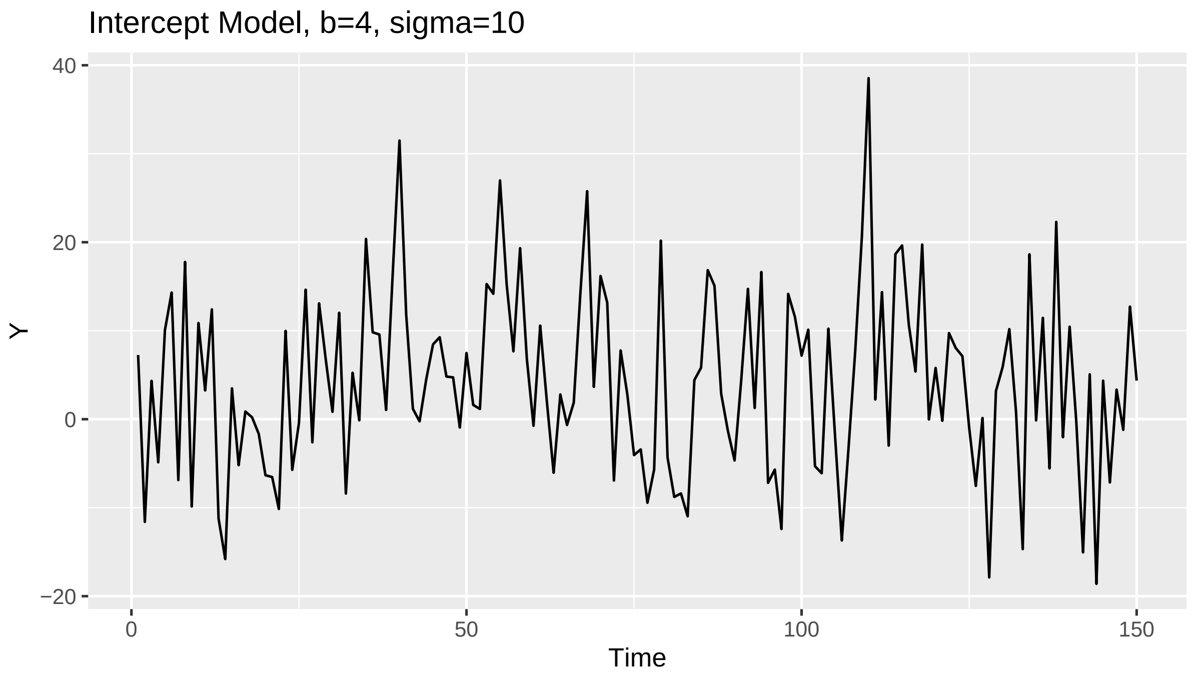


```
ggAcf(Y) + ggtitle("ACF Plot: Intercept Model, b=4, sigma=1")
```

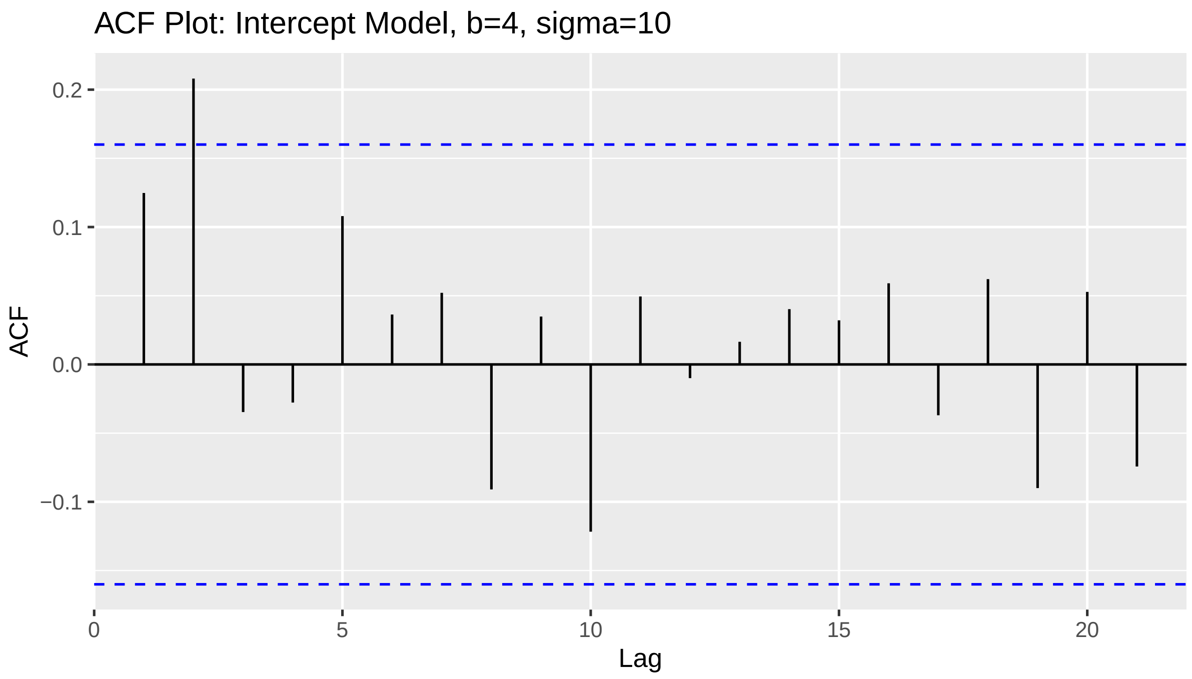


```
T <- 150

b <- 4.0

Y <- b + rnorm(150,0,10)

Y <- ts(Y)
```

```r
autoplot(Y) + ggtitle("Intercept Model, b=4, sigma=10")
```

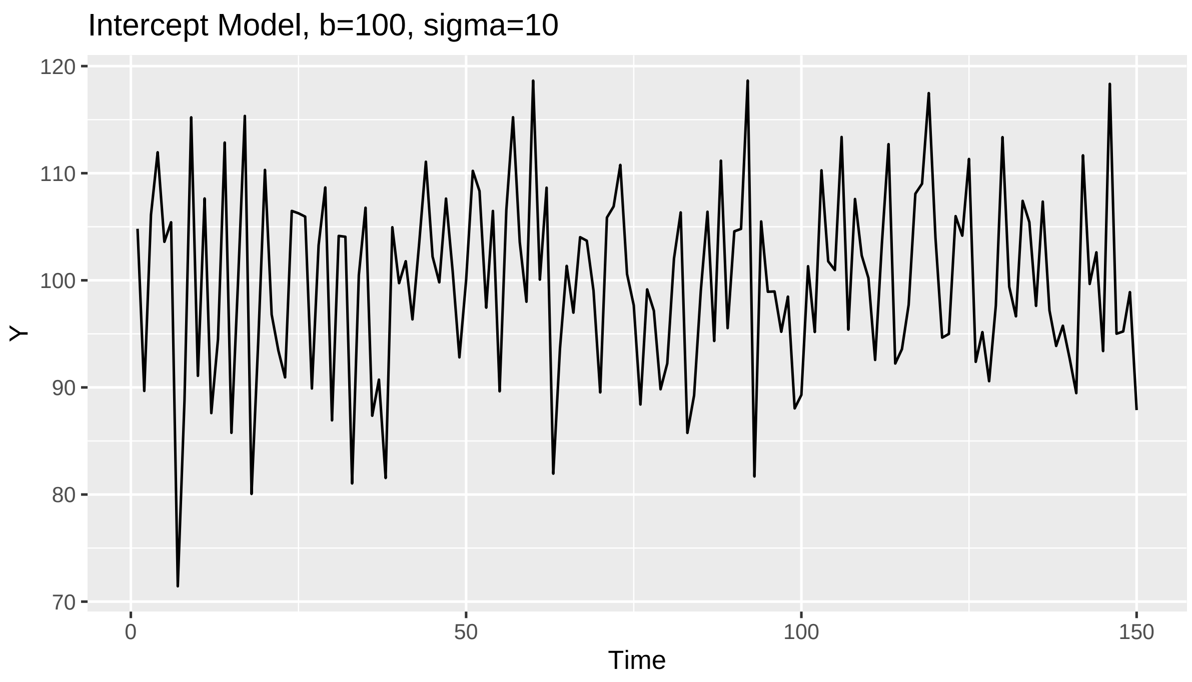


```r
ggAcf(Y) + ggtitle("ACF Plot: Intercept Model, b=4, sigma=10")
```

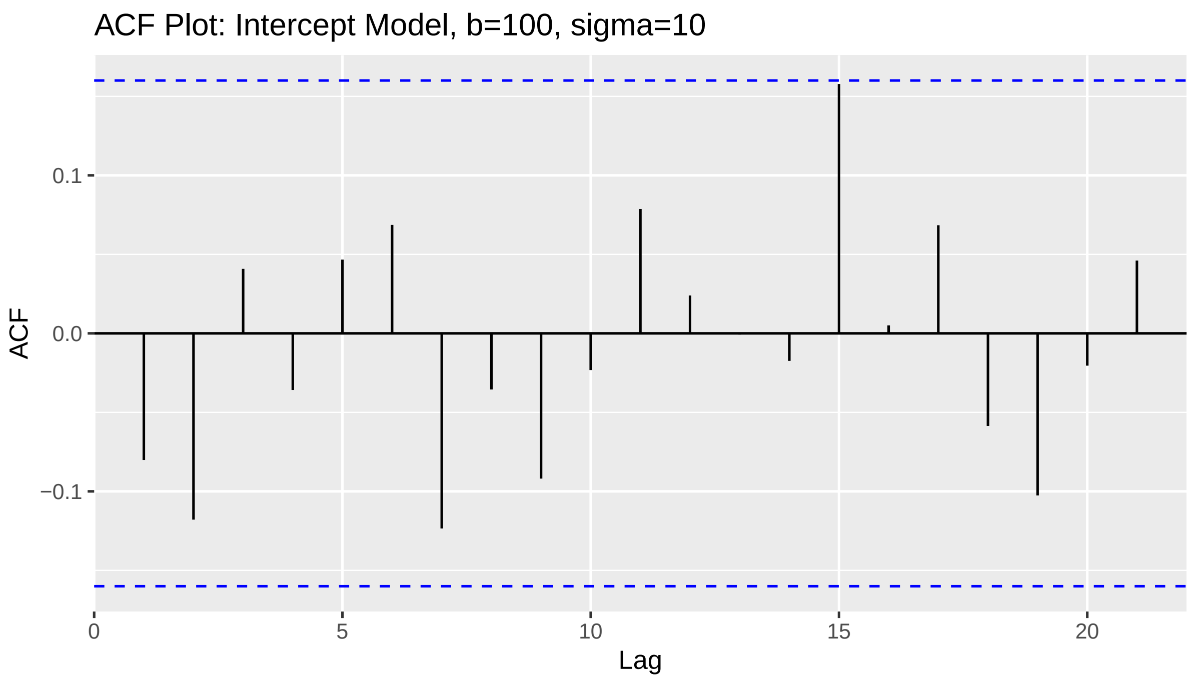


```r
T <- 150

b <- 100.0

Y <- b + rnorm(T,0,10)
```

```r
Y <- ts(Y)

autoplot(Y) + ggtitle("Intercept Model, b=100, sigma=10")
```



```r
ggAcf(Y) + ggtitle("ACF Plot: Intercept Model, b=100, sigma=10")
```



2.2 Random Walk (Naive)

The random walk (or "naive") method assumes that the data moves randomly through time, but does not have a long-run trend. In contrast with the intercept method, the random walk method works best when the data has high autocorrelation or a unit root.

In mathematics, the random walk model is given by:

$$\begin{gathered} y_t = y_{t-1} + \varepsilon_t \\ \varepsilon_t \sim \text{iid, mean } 0 \end{gathered}$$

Under squared error loss, the point-forecast from a random walk model can be calculated using expected value. Note that in this model, the point-forecast will not depend on the forecast horizon — just like in the intercept model, the optimal point-forecast will remain the same for all forecast horizons. Therefore, all we need to do is find the optimal one-period ahead point-forecast:

$$\begin{aligned} y_{T+1} &= y_T + \varepsilon_{T+1} \\ E(y_{T+1}|Y) &= E(y_T|Y) + E(\varepsilon_{T+1}|Y) \\ E(y_{T+1}|Y) &= E(y_T|Y) \\ \hat{y}_{T+1|T} &= y_T \end{aligned}$$

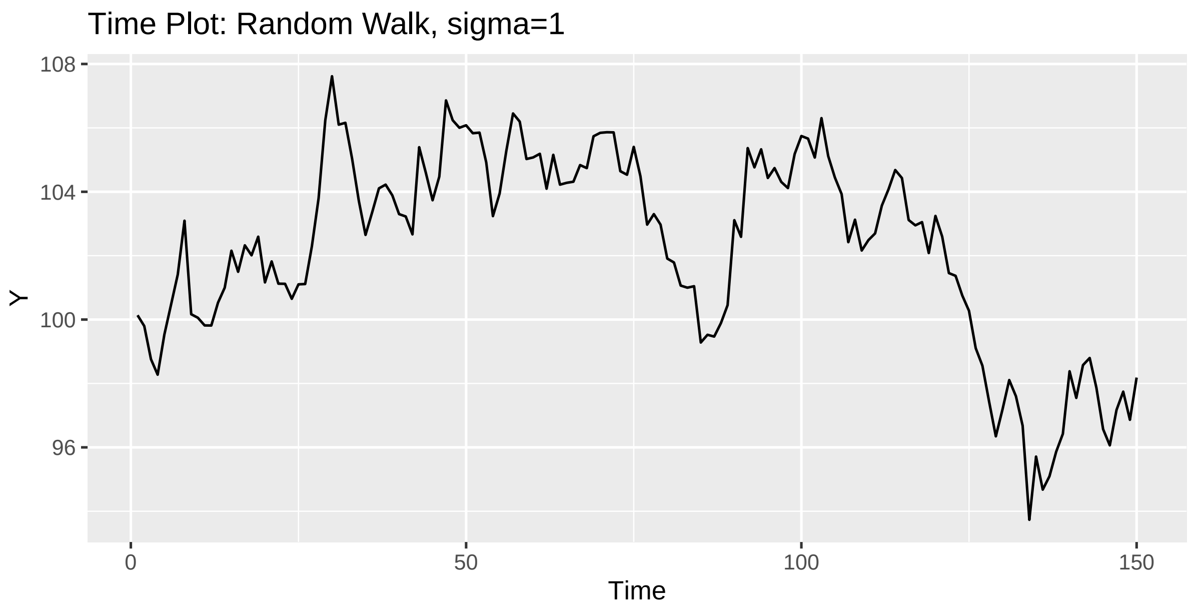
In words, the one-period ahead point-forecast is simply the value of the last observation - our best guess is that next period will be exactly the same as this period. Furthermore, as I asserted above, this will be the optimal forecast for *any* future time period:

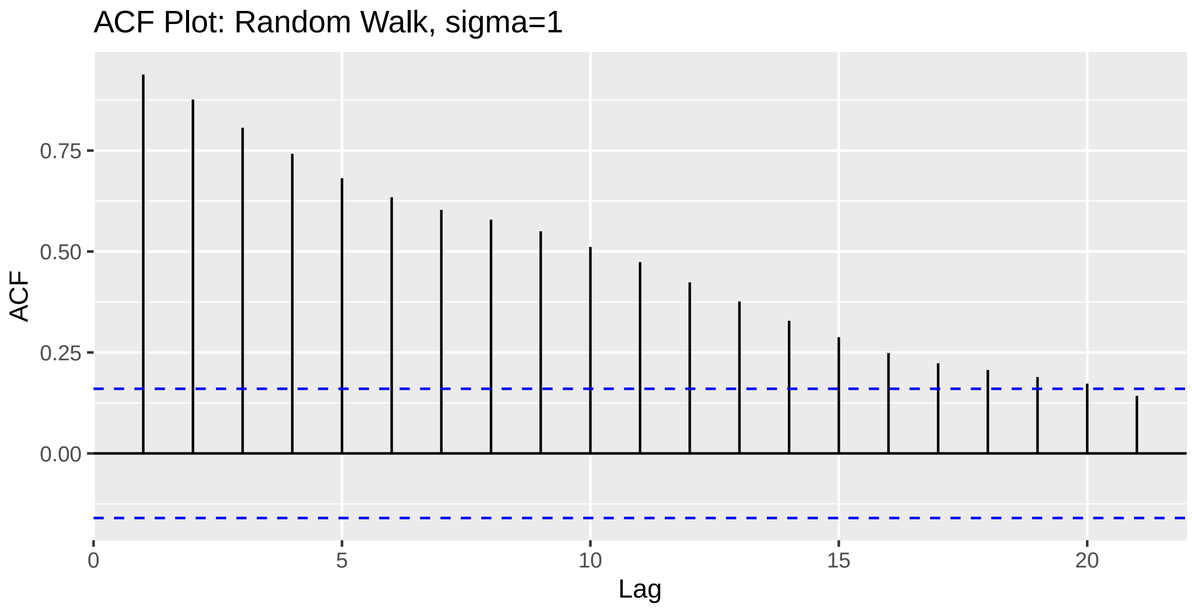
$$\hat{y}_{T+h|T} = y_T$$

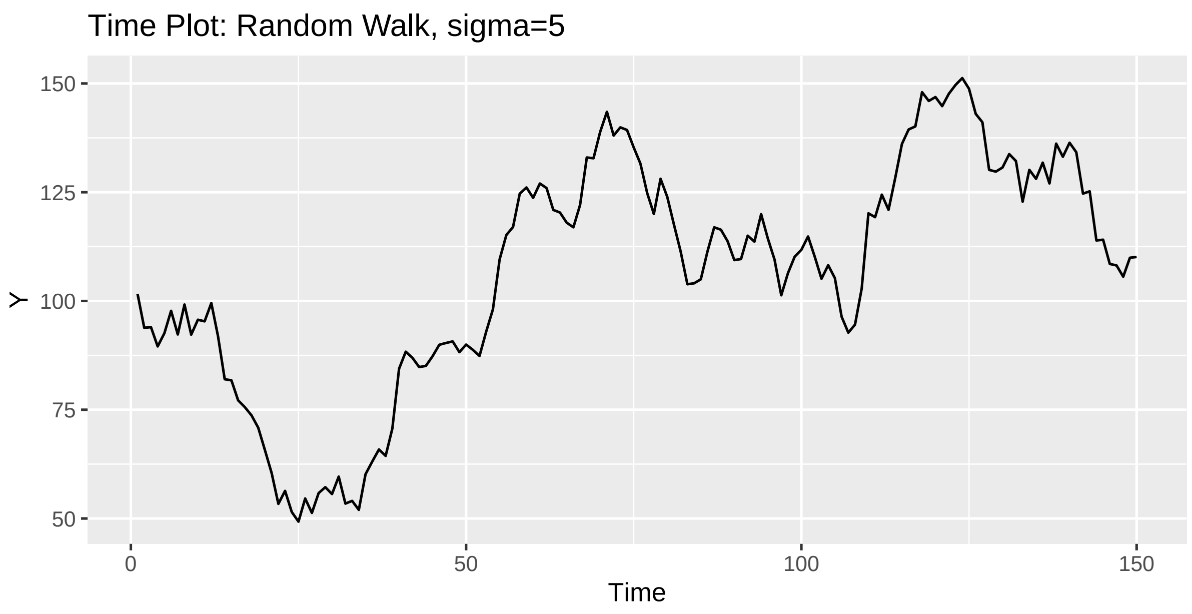
This forecasting method may seem too simple to perform well in practice. While it is often possible to improve upon this forecast, in some contexts it has proven remarkably difficult. For example, when forecasting daily stock prices or daily exchange rates, it has proven nearly impossible to beat this simple random walk forecast.

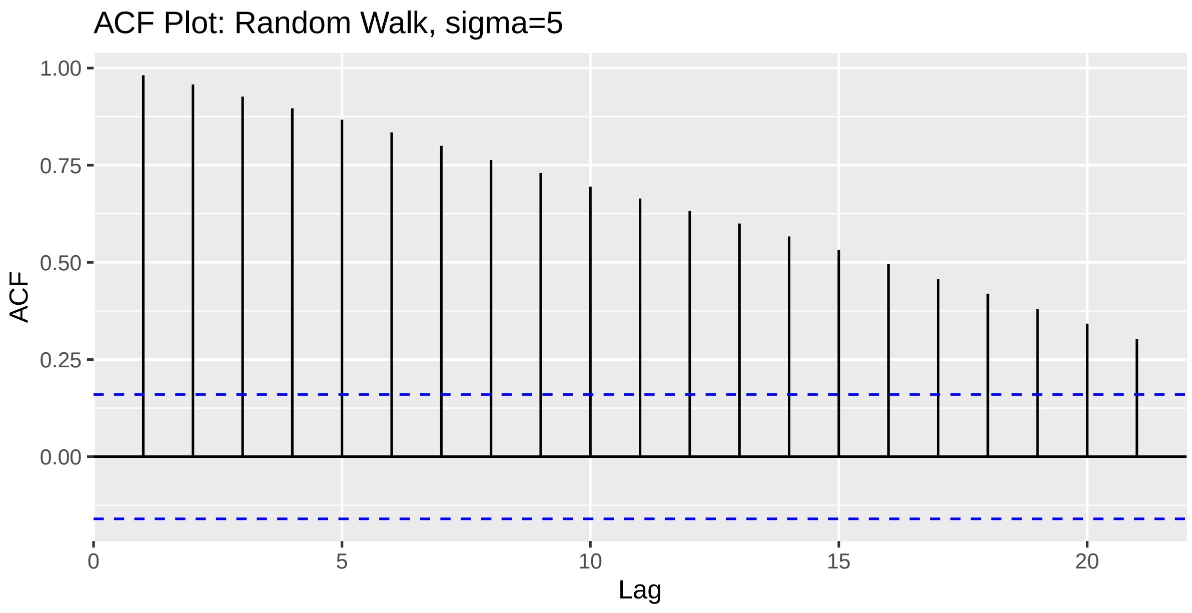
2.2.1 When to Use Random Walk

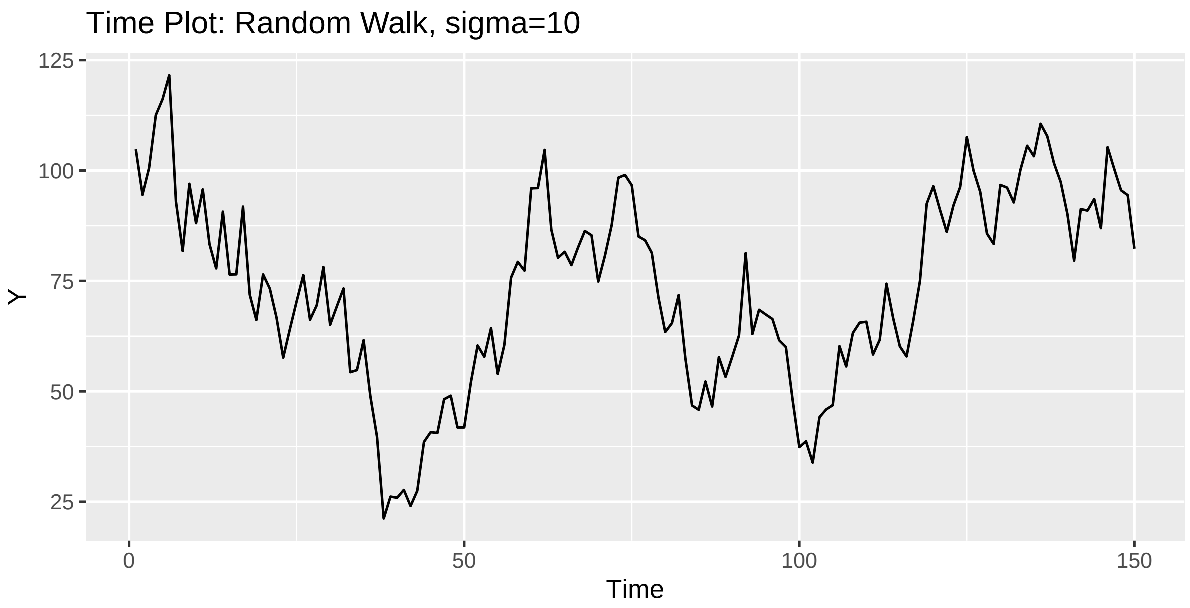
You should only use the random walk method when forecasting a variable that has very high autocorrelation, or a unit root, but does not have a trend or seasonality. Later in the course, we will discuss statistical tests that can help you determine if the data has a unit root. For now, you should acquaint yourself with what random walk data can look like, and only use the random walk method if your data looks something like these examples. Below are a few time plots and ACF plots of data that has been generated according to a random walk.

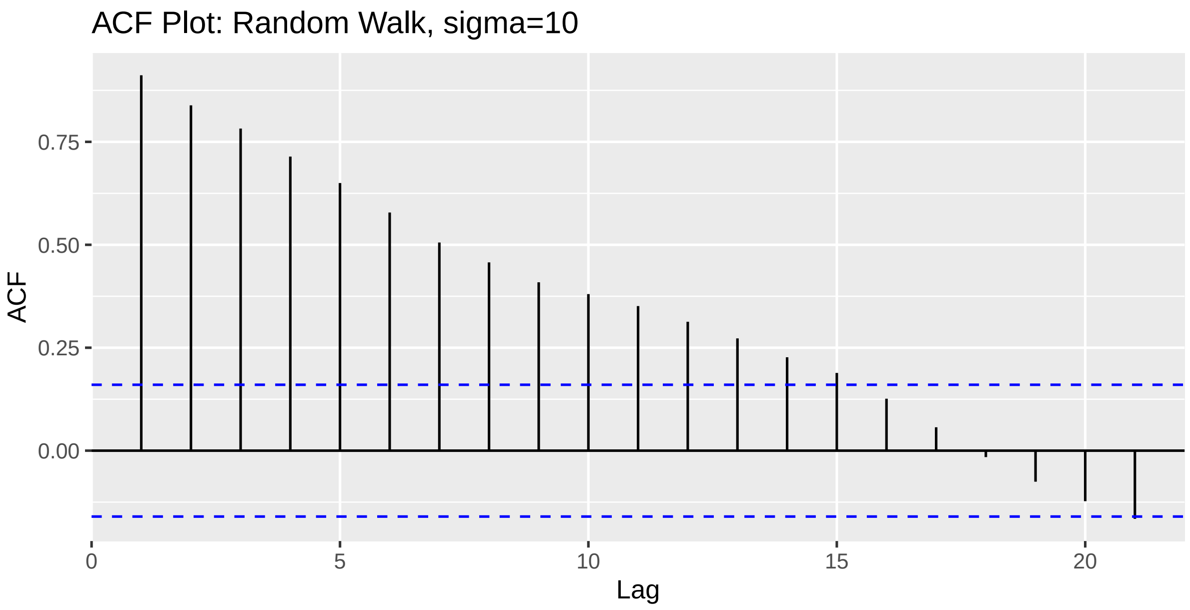





Time Plot: Random Walk, sigma=5
150
125
100
75
50
Y
0
50
100
150
Time


ACF Plot: Random Walk, sigma=5
1.00
0.75
0.50
0.25
0.00
ACF
0
5
10
15
20
Lag





2.3 Seasonal Random Walk (Seasonal Naive)

The last benchmark method we will learn about is the seasonal naive model. Recall that the random walk model posits that the data this period is very similar to the data last period. In contrast, the seasonal random walk model posits that the that the data this period will be very similar to the data in the same period last year. For example, if we were forecasting unemployment using the seasonal random walk model, it would say that the unemployment rate in October 2017 will be similar to the unemployment rate in October 2016. This method works well when the data is trend-stationary (i.e. no trend), but exhibits a large degree of seasonality.

Mathematically, let m be the possible number of seasons in a year. For example, if your data is monthly, $m = 12$. If your data is weekly, $m = 52$. Then, the seasonal random walk model can be written as:

$$y_t = y_{t-m} + \varepsilon_t$$
$$\varepsilon_t \sim \text{iid, mean } 0$$

Where y_{t-m} is the value of the data m periods ago (i.e. in the same period as today, one year prior).

The optimal forecast under squared error loss will be similar to the optimal forecast in the random walk model. However, you will have a unique prediction for each period. For example, if you are forecasting December 2017 sales using the seasonal naive method, then your forecast would be equal to sales recorded in December 2016. If you are forecasting January 2018 sales, the forecast would be equal to sales recorded in January 2017. Note that however long in the future you are forecasting, the forecast for any particular period will be identical. For example, the forecast for December 2017, December 2018, and December 2019 would all be sales recorded in December 2016, since that was the last observed sales number in a December.

Mathematically, the optimal forecast up to one year out (i.e. $h \leq s$) is:

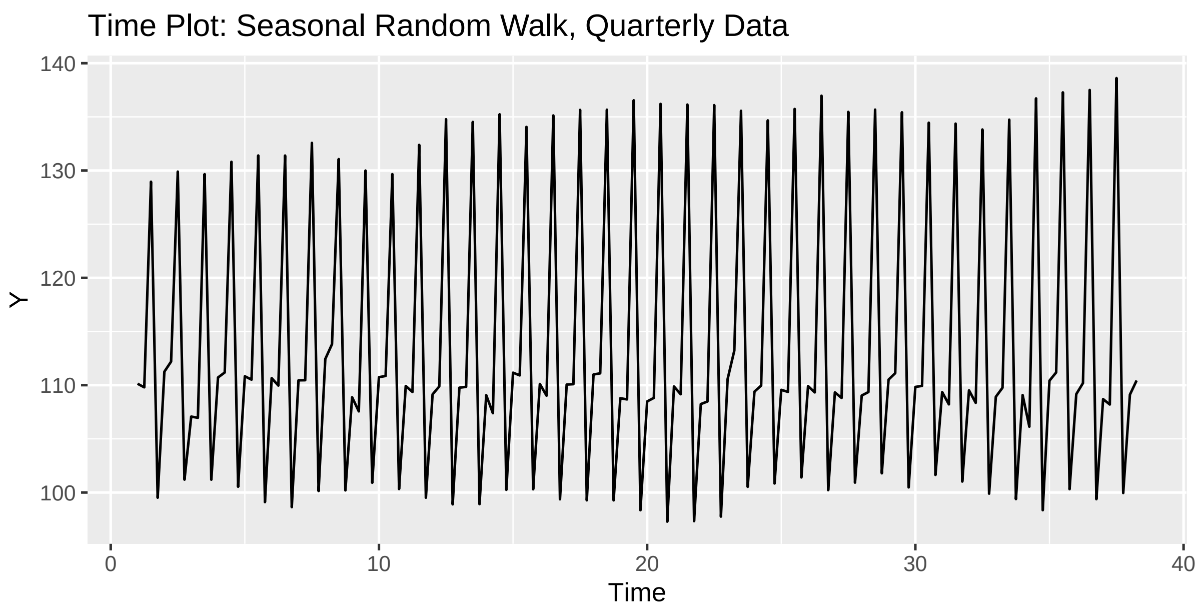
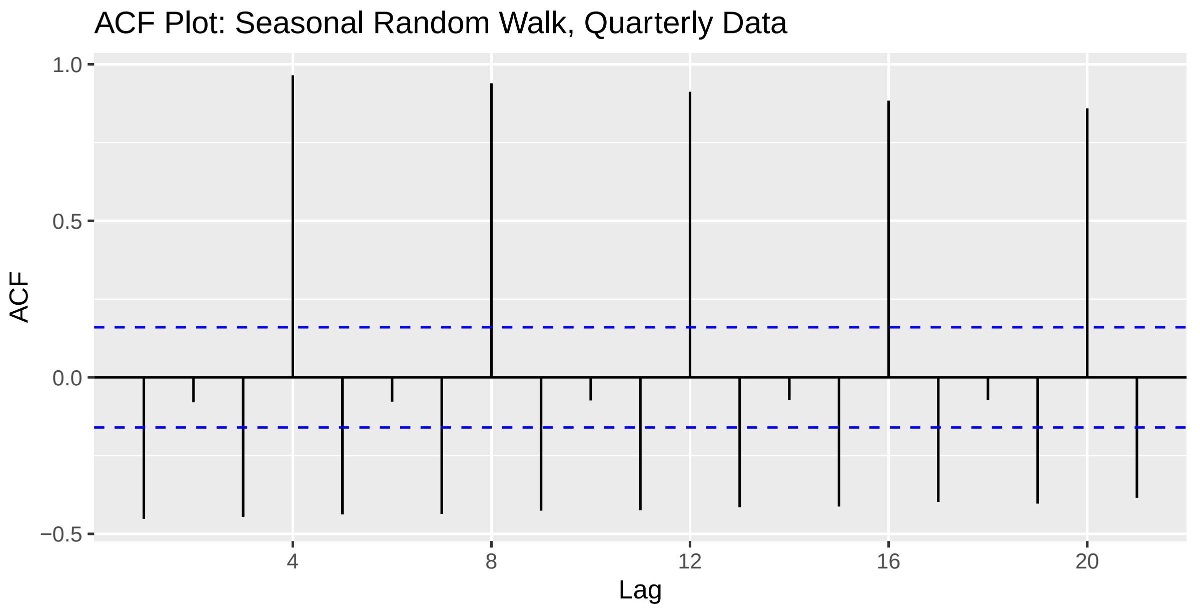
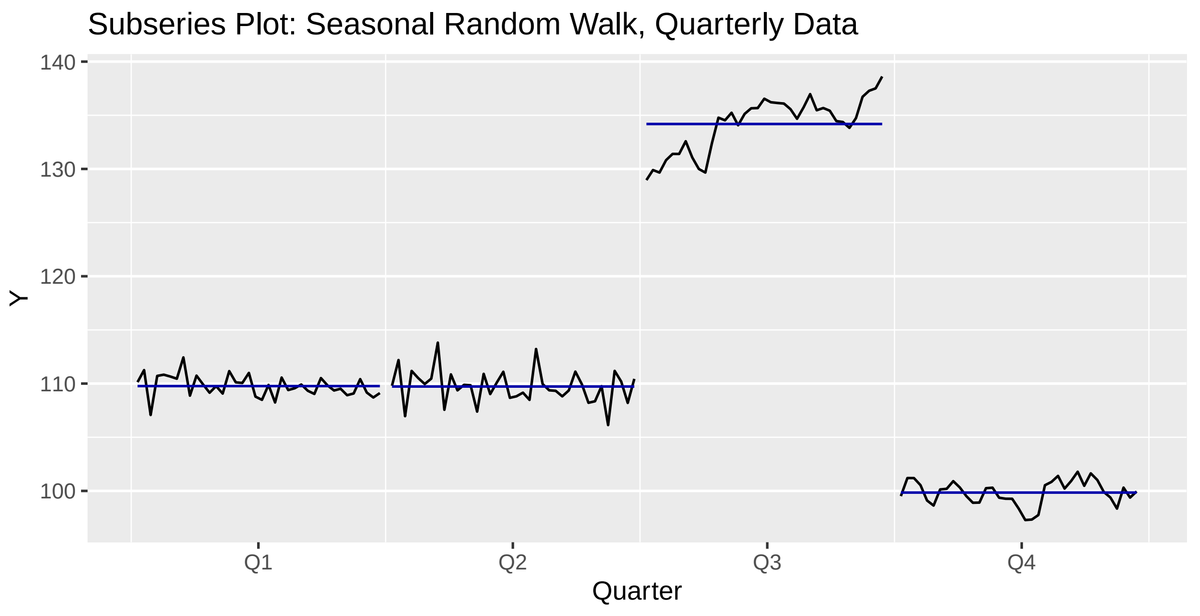
$$\hat{y}_{T+h|T} = y_{T-m+h}$$

While it is easy to describe the forecasts for more than one year in the future (see the paragraph above), it is difficult to write them succinctly mathematically.

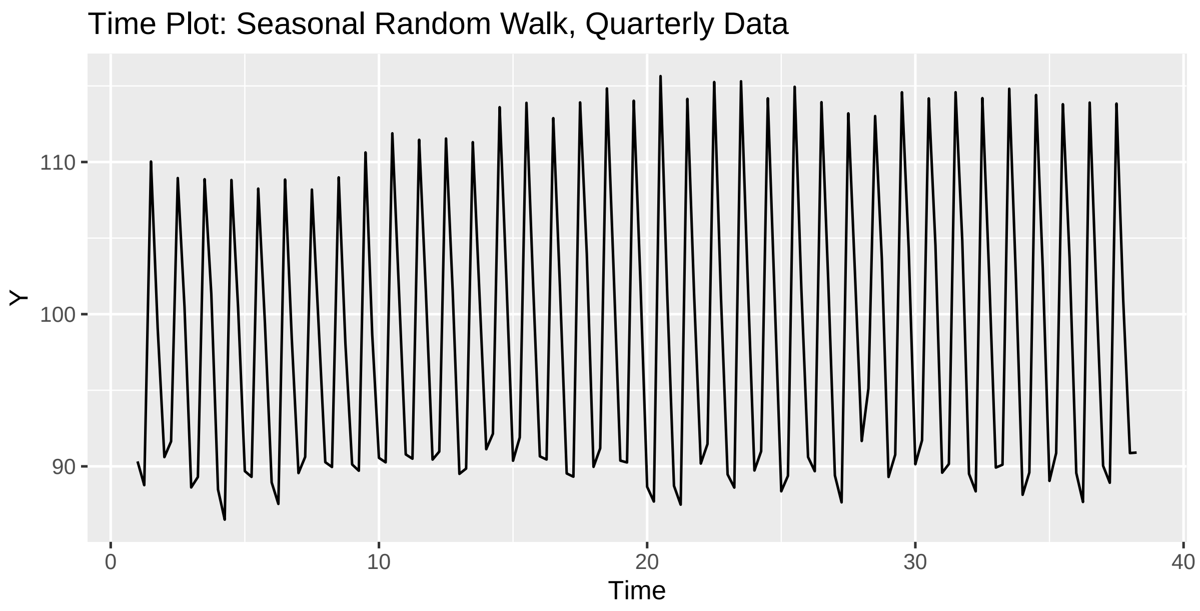
2.3.1 When to Use Seasonal Random Walk

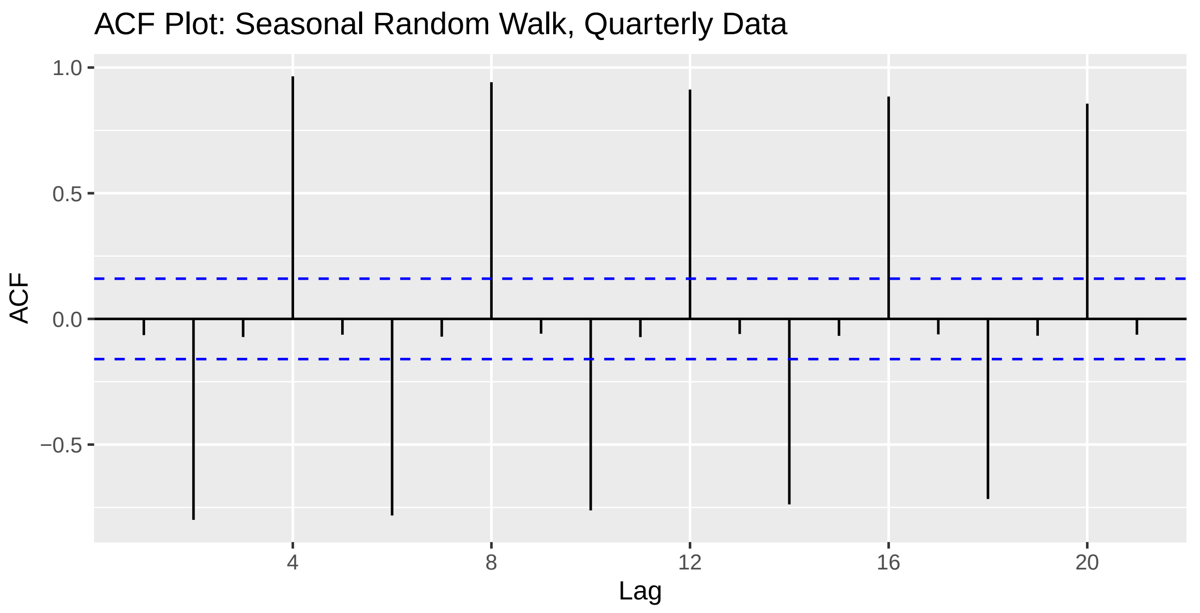
You should only use the seasonal random walk model with your data displays strong, persistent seasonality. Like other features, we can test for this statistically. For now, you should determine when to use this method based on plots of your data. I have plotted two examples below.

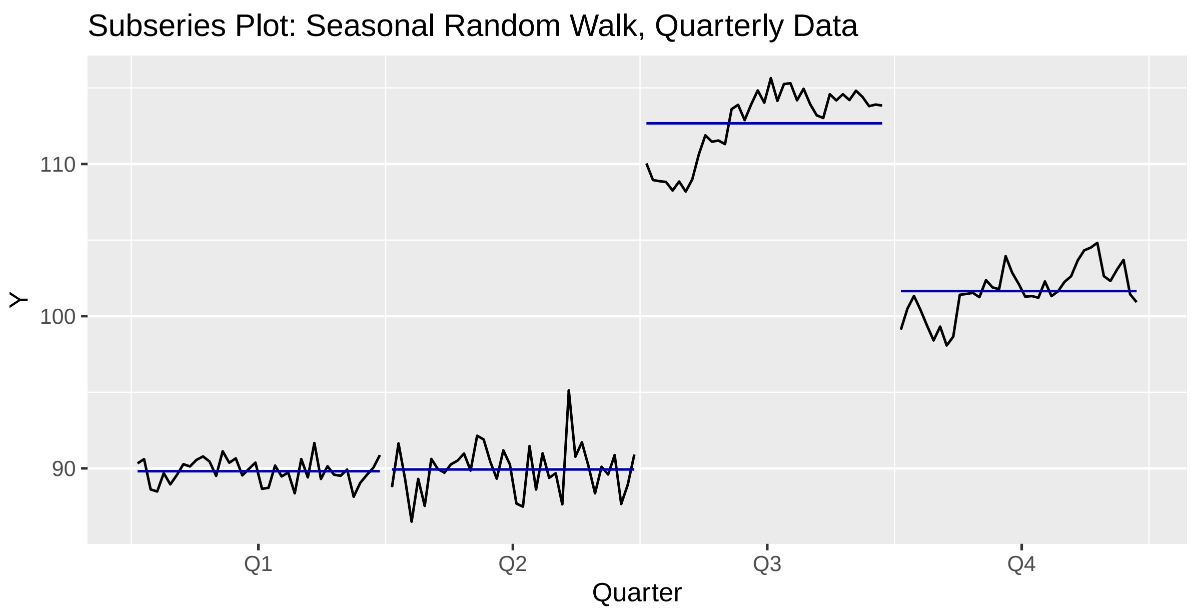
Example 1:


Time Plot: Seasonal Random Walk, Quarterly Data
Y
140
130
120
110
100
0
10
20
30
40
Time

ACF Plot: Seasonal Random Walk, Quarterly Data
ACF
1.0
0.5
0.0
−0.5
4
8
12
16
20
Lag

Subseries Plot: Seasonal Random Walk, Quarterly Data
Y
140
130
120
110
100
Q1
Q2
Q3
Q4
Quarter

Example 2:


Time Plot: Seasonal Random Walk, Quarterly Data
Y
110
100
90
0
10
20
30
40
Time


ACF Plot: Seasonal Random Walk, Quarterly Data
ACF
1.0
0.5
0.0
−0.5
4
8
12
16
20
Lag

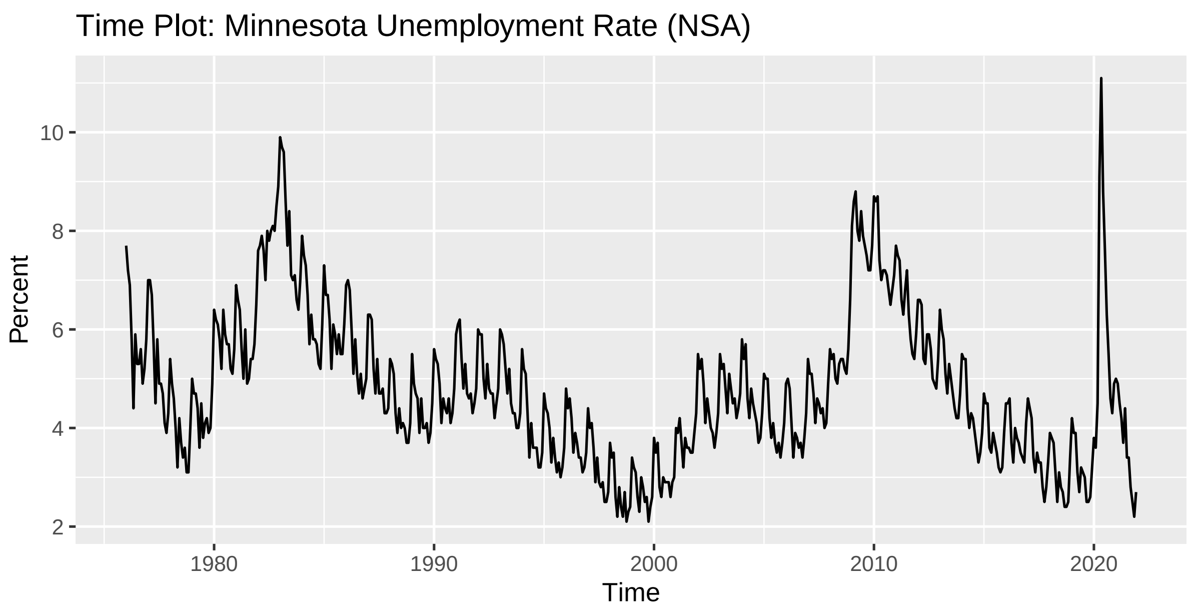


3 Application: Minnesota Unemployment Rate

In this section, we will apply all three benchmark forecasting methods to the unemployment rate in Minnesota. First, let's look at the summary plots of Minnesota's unemployment rate since data collection began in 1976. [1]

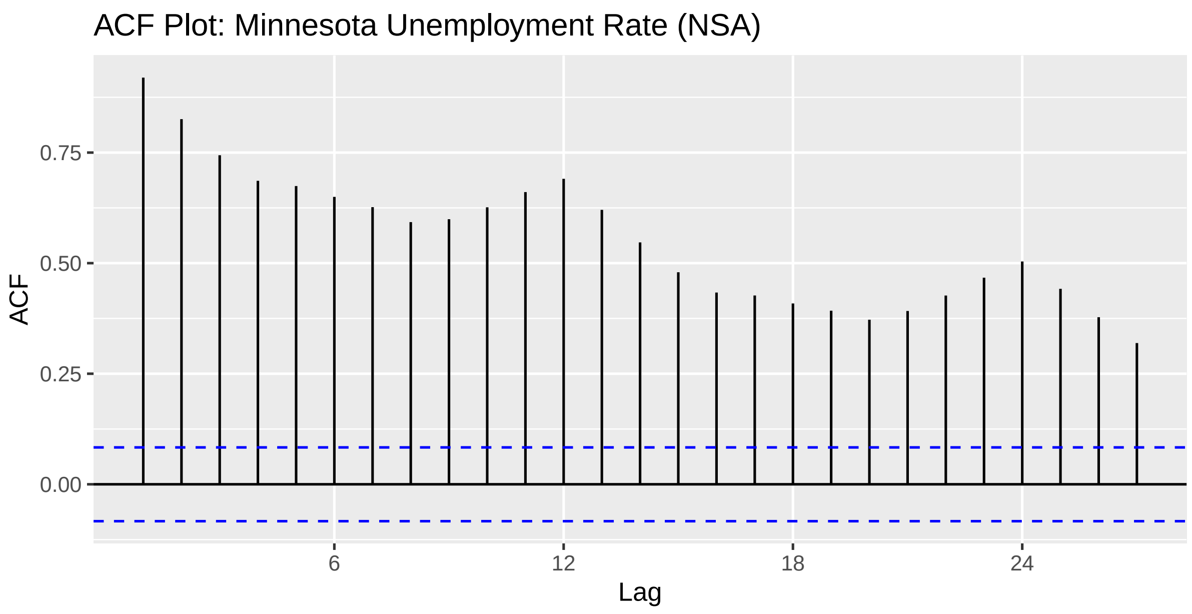
```
data_set <- fredr(series_id = "MNURN")
Y <- data_set$value
Y <- ts(Y,start=c(1976,1),frequency=12)
# Time Plot
autoplot(Y) +
  ggtitle("Time Plot: Minnesota Unemployment Rate (NSA)") +
  ylab("Percent")
```

[1]The code snippet loads the data using the "fredr" package. This way of loading the data will not work unless you install the "fredr" package, apply for an API key from FRED, and set up the API key correctly. Alternatively, you could download this data from FRED as a CSV file, and load it in the way we have in class.



There does not appear to be any long-run trend. Let's check for a unit root:

```
# ACF Plot
ggAcf(Y) +
  ggtitle("ACF Plot: Minnesota Unemployment Rate (NSA)")
```

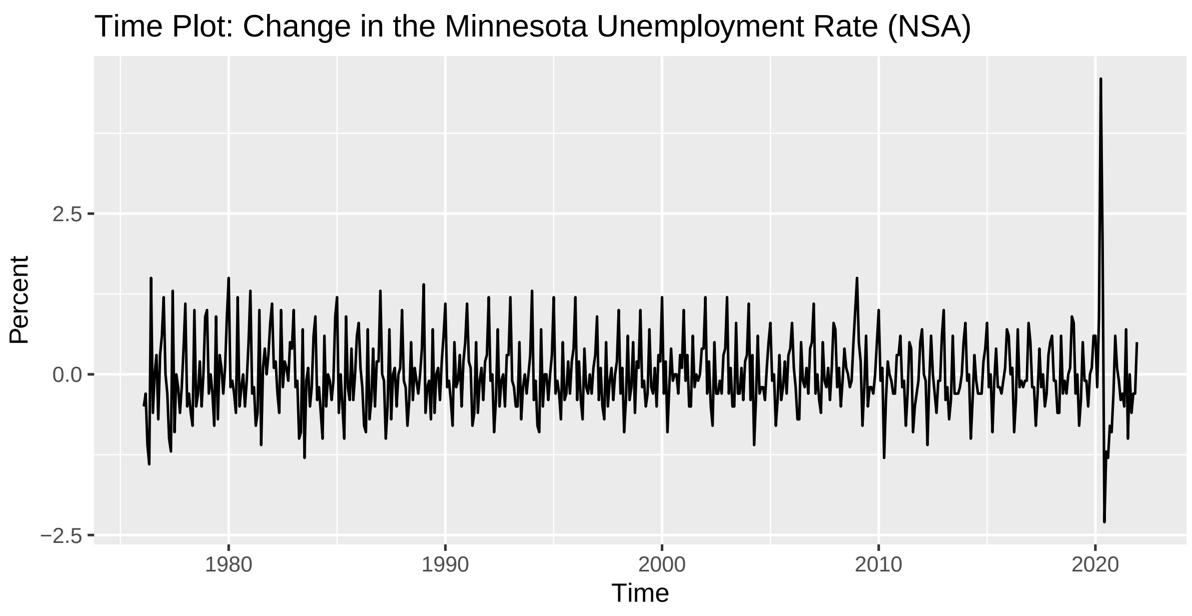


This is series appears that it may have a unit root, since the autocorrelation is relatively close to one at the first lag, and remains high at all past lags.[2] Let's proceed by taking the first

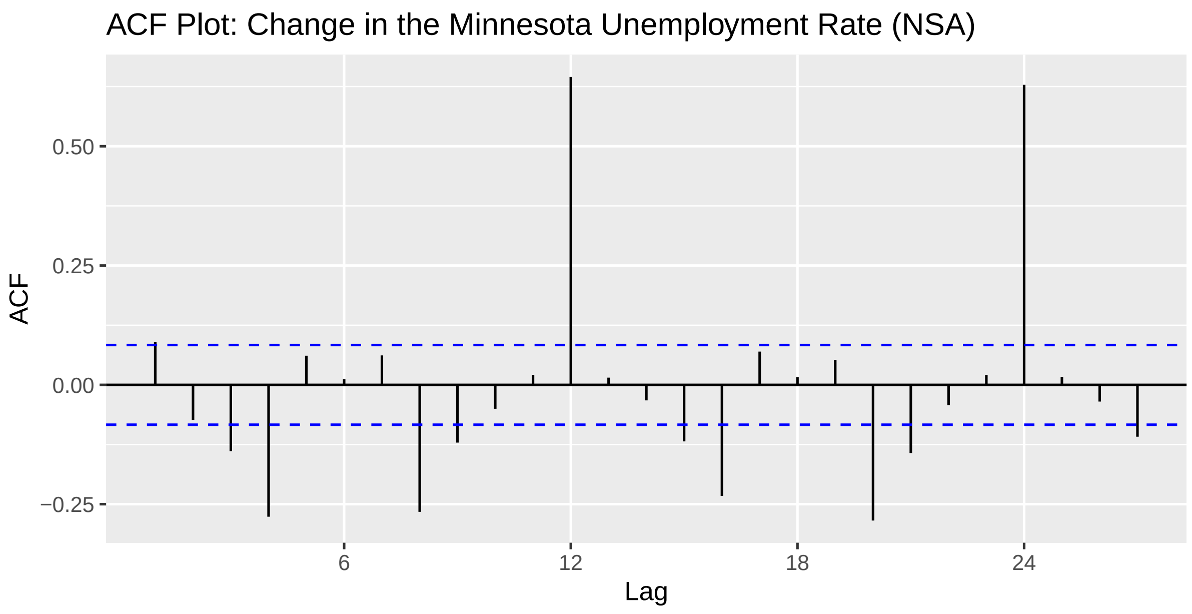
[2]This is a borderline case, as you could also argue that the spike at the first lag is not close enough to one to signal the presence of a unit root.

difference of the series:

```r
# ACF Plot
DY <- diff(Y)
autoplot(DY) +
  ggtitle("Time Plot: Change in the Minnesota Unemployment Rate (NSA)") +
  ylab("Percent")
```

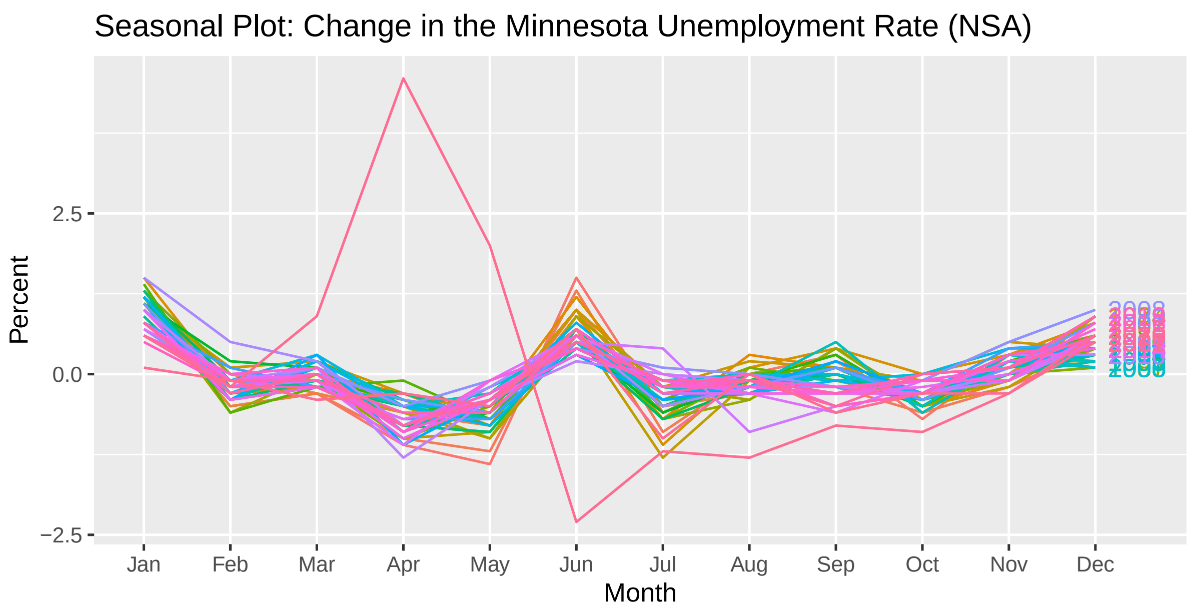


```r
ggAcf(DY) +
  ggtitle("ACF Plot: Change in the Minnesota Unemployment Rate (NSA)")
```



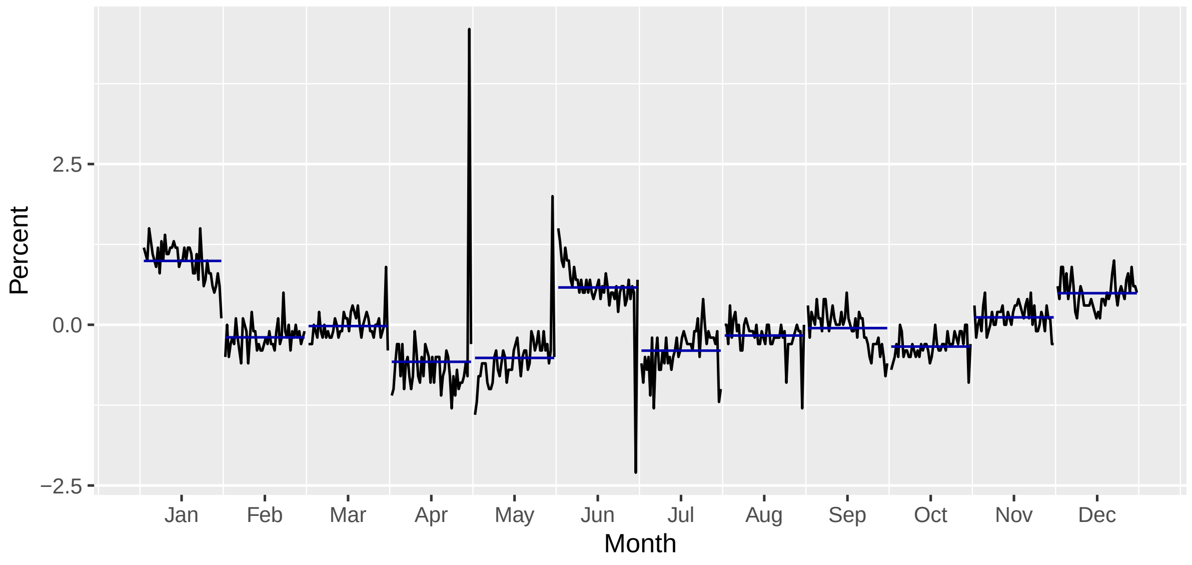
The change data looks much better, and no longer appears to have a unit root. Finally, let's investigate for seasonality:

```r
# Seasonal Plot
ggseasonplot(DY, year.labels=TRUE) +
  ggtitle("Seasonal Plot: Change in the Minnesota Unemployment Rate (NSA)") +
  ylab("Percent")
```



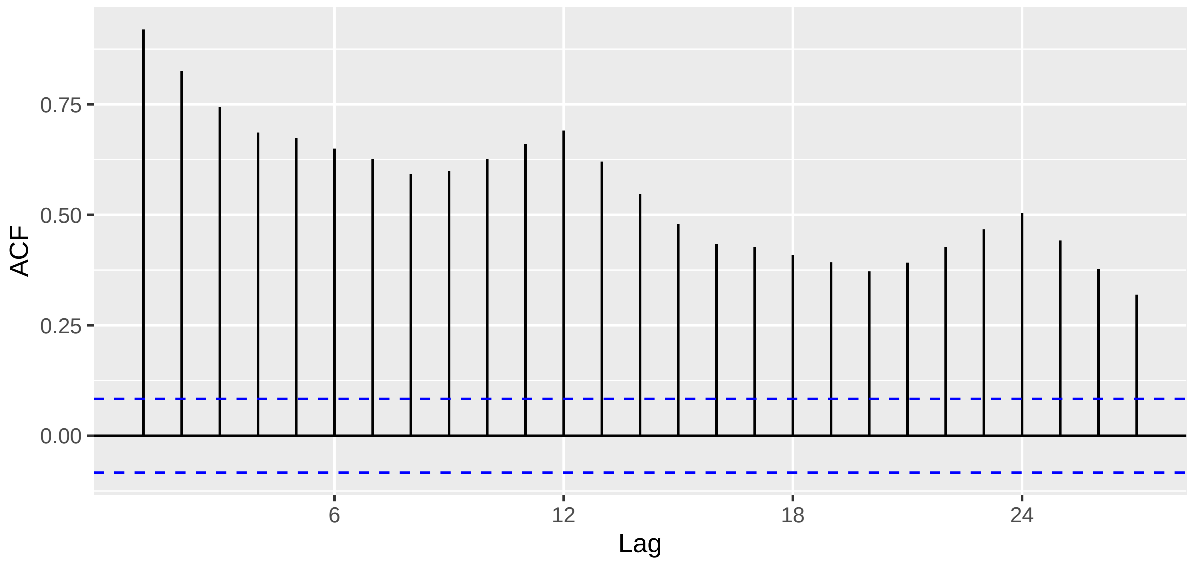
```r
# Seasonal Subseries Plot
ggsubseriesplot(DY) +
  ggtitle("Seasonal Subseries Plot: Change in the Minnesota Unemployment Rate (NSA)") +
  ylab("Percent")
```

Seasonal Subseries Plot: Change in the Minnesota Unemployment Rate (NSA)



```r
# ACF Plot
ggAcf(Y) +
  ggtitle("ACF Plot: Minnesota Unemployment Rate (NSA)")
```

ACF Plot: Minnesota Unemployment Rate (NSA)



It appears that there is seasonality due to the distinct averages between months. For example, the unemployment rate seems to rise sharply every January, as the average change in January is roughly 1 percentage point. In April, the unemployment rate tends to decrease, falling by an average of roughly 0.75 percentage points.

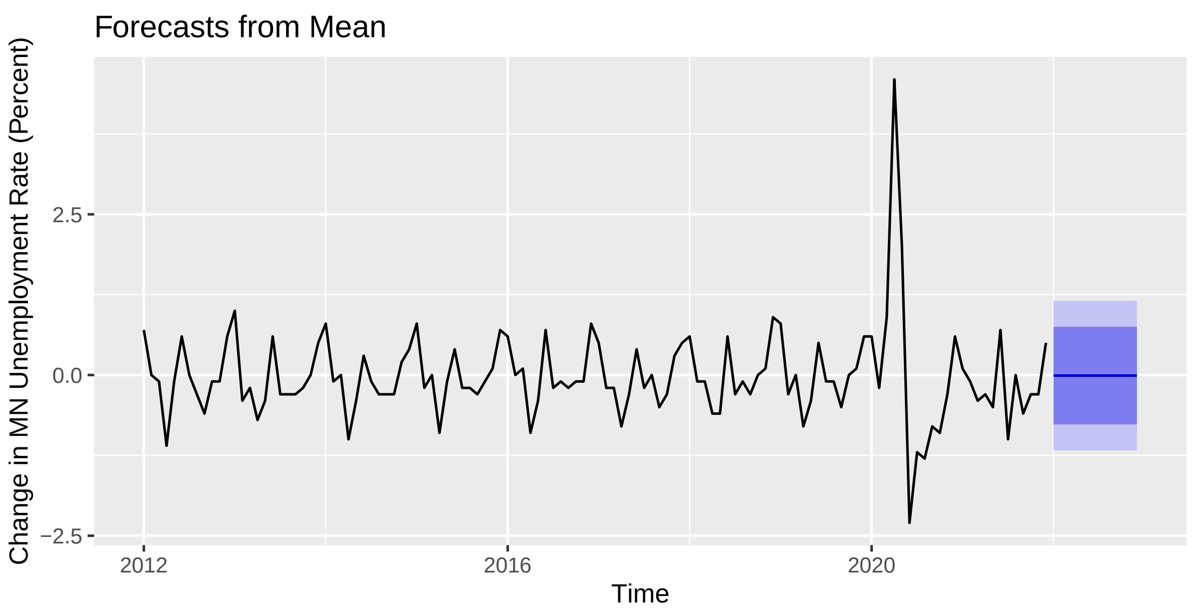
Summarizing:

1. There does NOT appear to be a long-run trend.
2. There may be a unit root — therefore we first-differenced the series before inspecting for seasonality.
3. There appears to be strong seasonality in the first-differenced data.

If there were no trend AND no seasonality, it might make sense to use the Random Walk model on the raw data, Y. Since there is seasonality present, it makes more sense to use the seasonal random walk model on the first-differenced data, ΔY. We will run the other methods as well, although they are not as appropriate in this case.

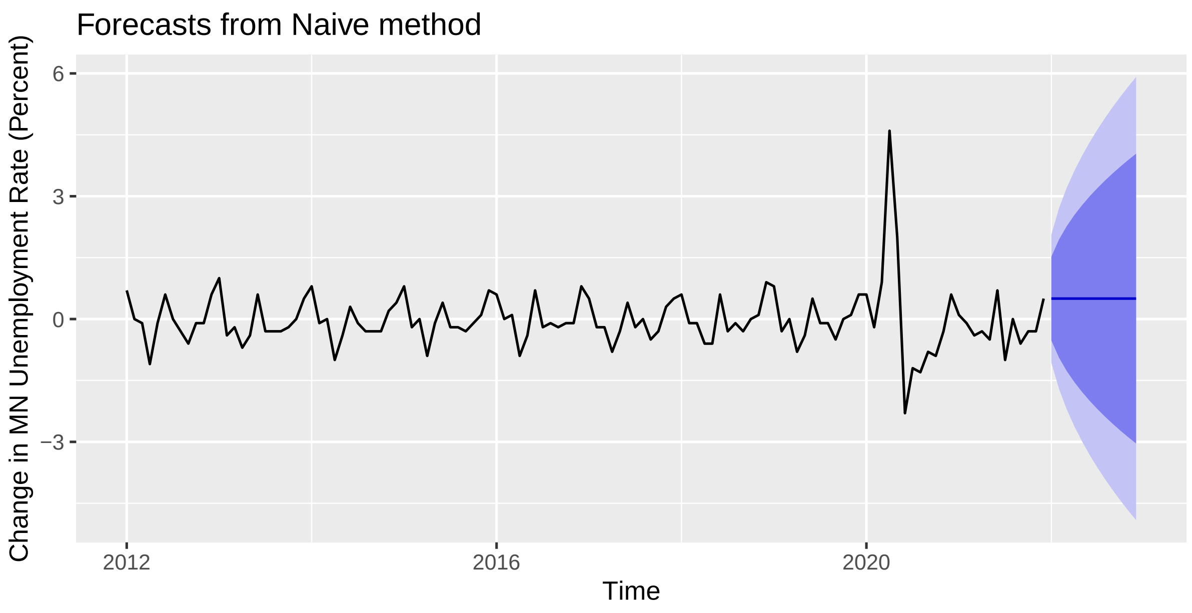
Although we do not expect the intercept model to do well in this context, we will still use it as an example. We will form forecasts up to one-year (12 months) in the future:

```r
# Fit the intercept model
fcst_mean <- meanf(DY,h=12)
# Plot the forecasts from the intercept model
autoplot(fcst_mean,include=120) + ylab("Change in MN Unemployment Rate (Percent)")
```



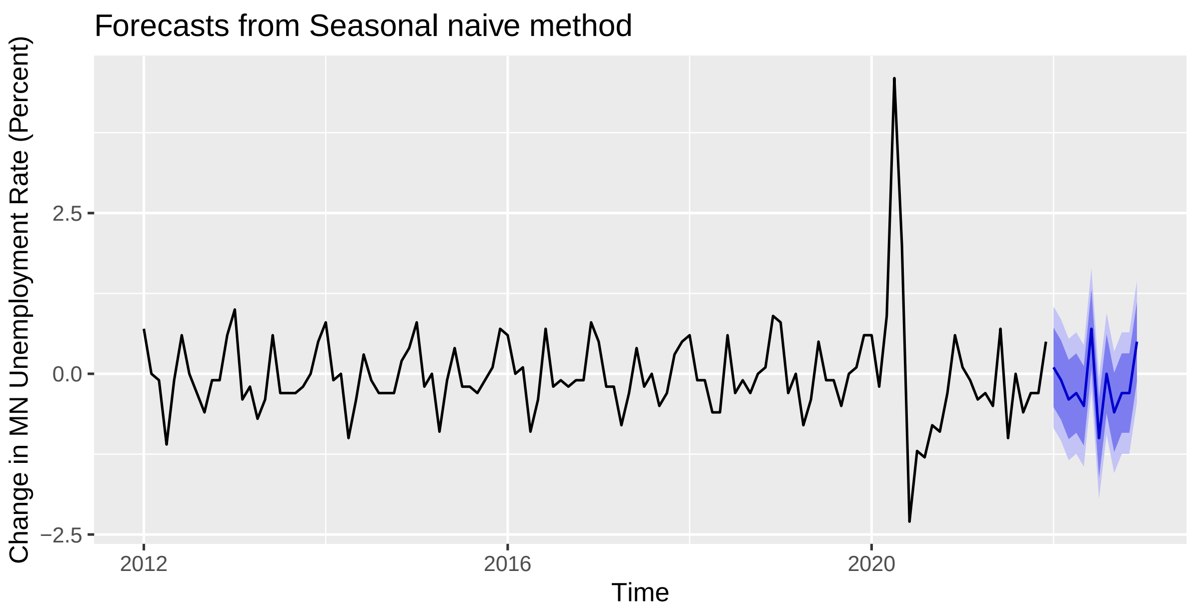
Next, the Random Walk (i.e. Naive):

```
# Fit the random walk model

fcst_rw <- naive(DY,h=12)

# Plot the forecasts from the random walk model

autoplot(fcst_rw,include=120) + ylab("Change in MN Unemployment Rate (Percent)")
```



Finally, the Seasonal Random Walk (i.e. Seasonal Naive):

```
# Fit the seasonal RW model

fcst_srw <- snaive(DY,h=12)

# Plot the forecasts from the seasonal RW model

autoplot(fcst_srw,include=120) + ylab("Change in MN Unemployment Rate (Percent)")
```



Due to the strong seasonality present in the data, before looking at any in-sample statistics or observing future data, it appears this would be the most appropriate model for this data set.

4 Exercises

1. Recall the intercept model:

$$y_t = b + \varepsilon_t$$

$$\varepsilon_t \sim N(0, \sigma)$$

Use R to simulate data from this model with the following properties:

- $T = 150$
- $b = 10$
- $\sigma = 3$

(a) What is the sample mean of the data you simulated? Under what loss function would this be the optimal forecast for time period 151?

(b) What is the sample median? Under what loss function would this be the optimal forecast for time period 151?

(c) What is the 80$^{\text{th}}$ percentile of the data? Under what loss function would this be the optimal forecast for time period 151?

2. Suppose you came across a time series data set, and after looking at a time plot and a seasonal plot, you thought that it had seasonality but no trend. Which of the three "benchmark" methods would probably be most appropriate? Why?

3. Go to FRED and find the data set titled "All Employees: Total Nonfarm Payrolls" (seasonally adjusted). Edit the graph so that the Units are "Change, Thousands of Persons," and then download the data in a .CSV file. This data is the net monthly change in the number of jobs in the United States. Read the data into R.

   (a) Use the intercept method to forecast payroll jobs growth next month. What is the point-forecast?

   (b) Use the random walk/naive method to forecast payroll jobs growth next month. What is the point-forecast?

   (c) Use the seasonal naive method to forecast payroll jobs growth next month. What is the point-forecast?

   (d) Which of these forecasts seems most reasonable to you? Why? (Hint: it might help to plot the data along with the forecast(s))