1 - Getting Started

Rob Hyndman (with Deppa modifications and examples)

January 13, 2021

* [Example 1.1 - Quarterly Beer Production in Australia (1956 - Q2 2010)](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#example-1.1---quarterly-beer-production-in-australia-1956---q2-2010)
* [Example 1.2 - Number of Annual International Visitors to Australia (1980-2015)](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#example-1.2---number-of-annual-international-visitors-to-australia-1980-2015)
* [Examples from Introductory Powerpoint](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#examples-from-introductory-powerpoint)
  + [1.3 - CO2 Levels at Mauna Loa Observatory (1959-1997)](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#co2-levels-at-mauna-loa-observatory-1959-1997)
  + [1.4 - U.S. Monthly Housing Starts (1959 - Present)](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#u.s.-monthly-housing-starts-1959---present)
  + [1.5 - U.S. Monthly Liquor Sales (1980-2007)](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#u.s.-monthly-liquor-sales-1980-2007)
  + [1.6 - Dow Jones Industrial Average](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#dow-jones-industrial-average)
* [Discussion](http://course1.winona.edu/bdeppa/FIN%20335/Handouts/Getting_Started.html#discussion)

This R Markdown file will show how to produce the time series displays shown in Chapter 1 of the text. It will cover some R basics such as reading a .csv datafile into R, data storage in R, and interacting with a dataset in R.

Example 1.1 - Quarterly Beer Production in Australia (1956 - Q2 2010)

The file (<http://course1.winona.edu/bdeppa/FIN%20335/Datasets/AusBeer.csv>) on my course website contains 4 columns of information giving quarterly beer production figures in Australia from 1956 through the 2nd quarter of 2010. The beer production figure is in megalitres. If you download the AusBeer.csv file and open it in Excel you find that it has four columns: Time (1 - 218), Year (1956 - 2010), Quarter (Q1-Q4), and finally Beer.Production, which is the beer production in that quarter in megalitres. We will read these data into R using the command read.csv which will be the primary way we will read all subsequent course datasets into R. The read.csv command will read the comma delimited data into an R object called a *data frame*.

**require**(fpp2)

## Loading required package: fpp2

## Warning: package 'fpp2' was built under R version 3.6.1

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.1

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.6.1

## Registered S3 method overwritten by 'xts':

## method from

## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':

## method from

## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':

## method from

## fitted.fracdiff fracdiff

## residuals.fracdiff fracdiff

## Loading required package: fma

## Warning: package 'fma' was built under R version 3.6.1

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.6.1

AusBeer = read.csv(file="http://course1.winona.edu/bdeppa/FIN%20335/Datasets/AusBeer.csv")

View(AusBeer)

names(AusBeer)

## [1] "Time" "Year" "Quarter" "Beer.Production"

str(AusBeer)

## 'data.frame': 218 obs. of 4 variables:

## $ Time : int 1 2 3 4 5 6 7 8 9 10 ...

## $ Year : int 1956 1956 1956 1956 1957 1957 1957 1957 1958 1958 ...

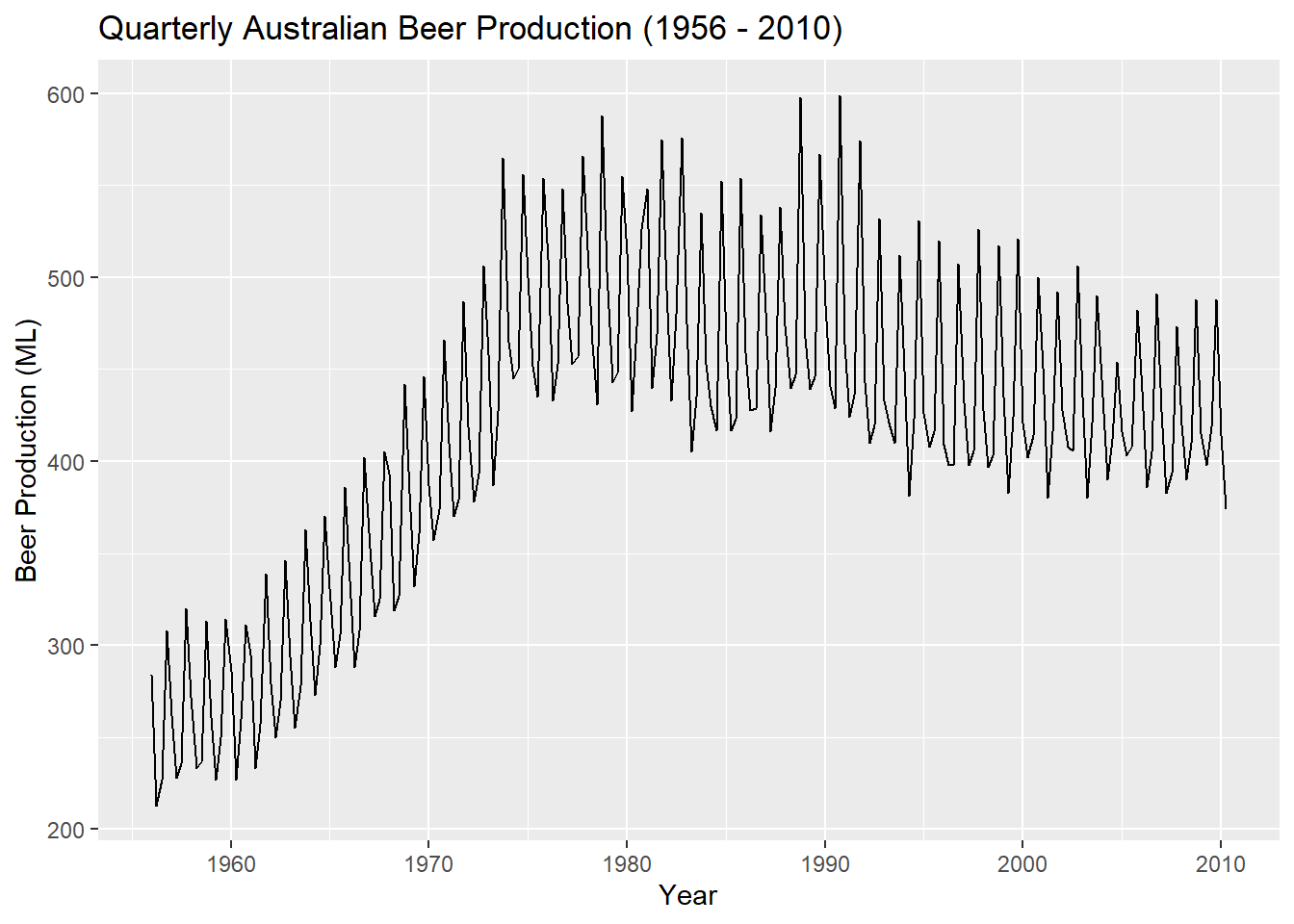
## $ Quarter : Factor w/ 4 levels "Q1","Q2","Q3",..: 1 2 3 4 1 2 3 4 1 2 ...

## $ Beer.Production: int 284 213 227 308 262 228 236 320 272 233 ...

Beer = AusBeer$Beer.Production

Beer = ts(Beer,frequency=4,start=1956)

autoplot(Beer)+ggtitle("Quarterly Australian Beer Production (1956 - 2010)") + xlab("Year") + ylab("Beer Production (ML)")

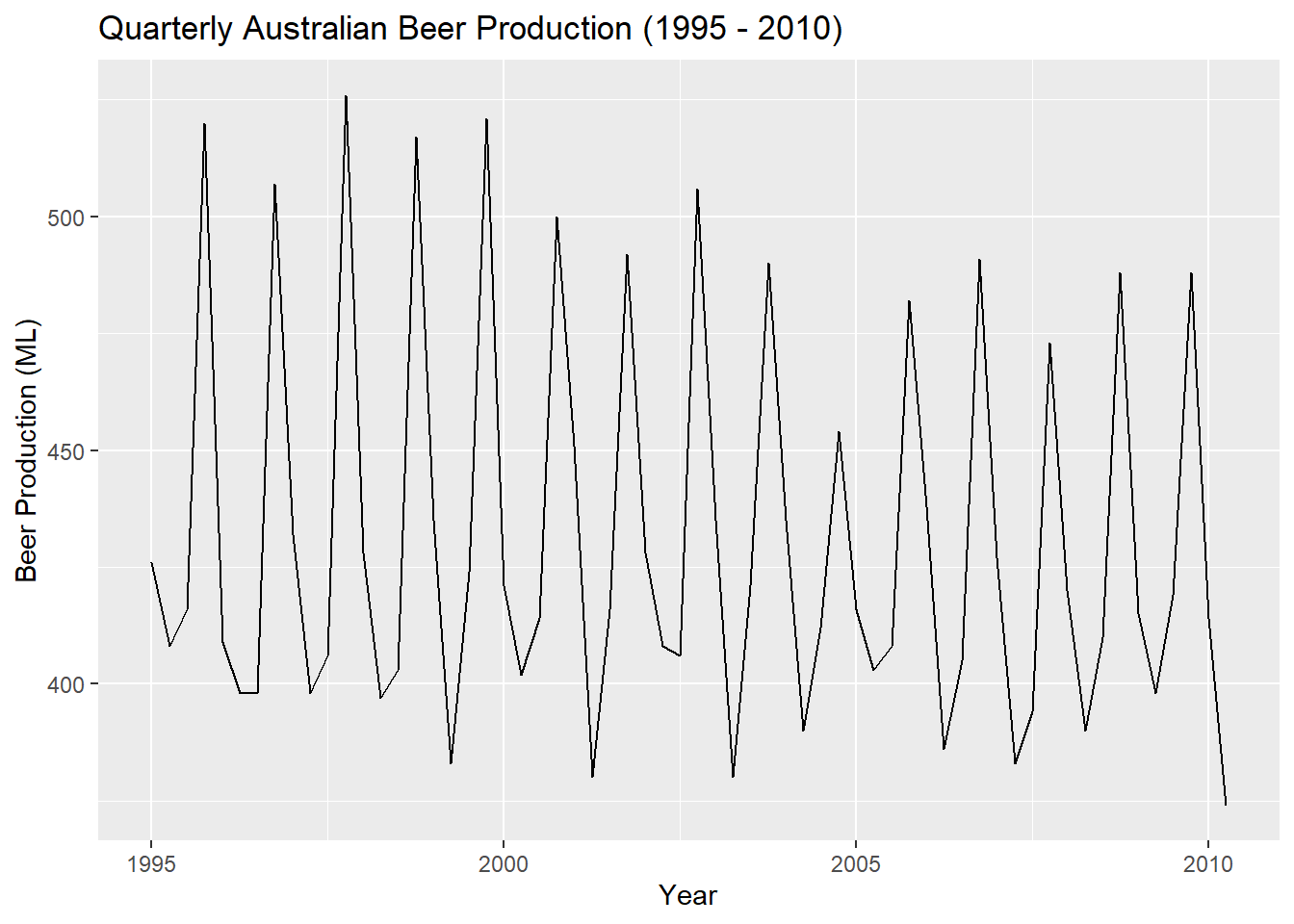


Here we have read the AusBeer.csv file into a data frame called AusBeer. The names(data frame name) command show the names of the variables/columns in the data frame. We can refer to individual variables by typing the name of the data frame followed by a $ and the name of variable as we did above with the Beer.Production column. The Beer.Production column is actually a time series containing the quarterly (frequency=4) beer production starting in 1956 (start = 1956). The command ts turns a variable into a time series object. Time series objects in R can then be supplied to other functions in R that will allow us to plot the time series, summarize its structure, fit models to the time series, and make forecasts. Here we used the command autoplot from the forecast library, developed by the textbook author, to create a basic plot of this time series. We will cover time series graphics in more detail in Chapter 2.

The window command to subset the time series to only consider 1995 to 2010 with the subset being stored in a new time series objects Beersub which can then be plotted using autoplot().

Beersub = window(Beer,start=1995)

autoplot(Beersub)+ggtitle("Quarterly Australian Beer Production (1995 - 2010)") + xlab("Year") + ylab("Beer Production (ML)")



Example 1.2 - Number of Annual International Visitors to Australia (1980-2015)

The second example of time series plotted in Chapter 1 is the annual number of international visitors to Australia (1980-2015). We will again read the .csv file into R and then create a time series object for the number of visitors (Visits). The sequence of commands in R is basically the same as in our first example.

**require**(fpp2)

AusVisit = read.csv(file="http://course1.winona.edu/bdeppa/FIN%20335/Datasets/AusVisitors.csv")

names(AusVisit)

## [1] "Time" "Year" "Visitors"

str(AusVisit)

## 'data.frame': 36 obs. of 3 variables:

## $ Time : int 1 2 3 4 5 6 7 8 9 10 ...

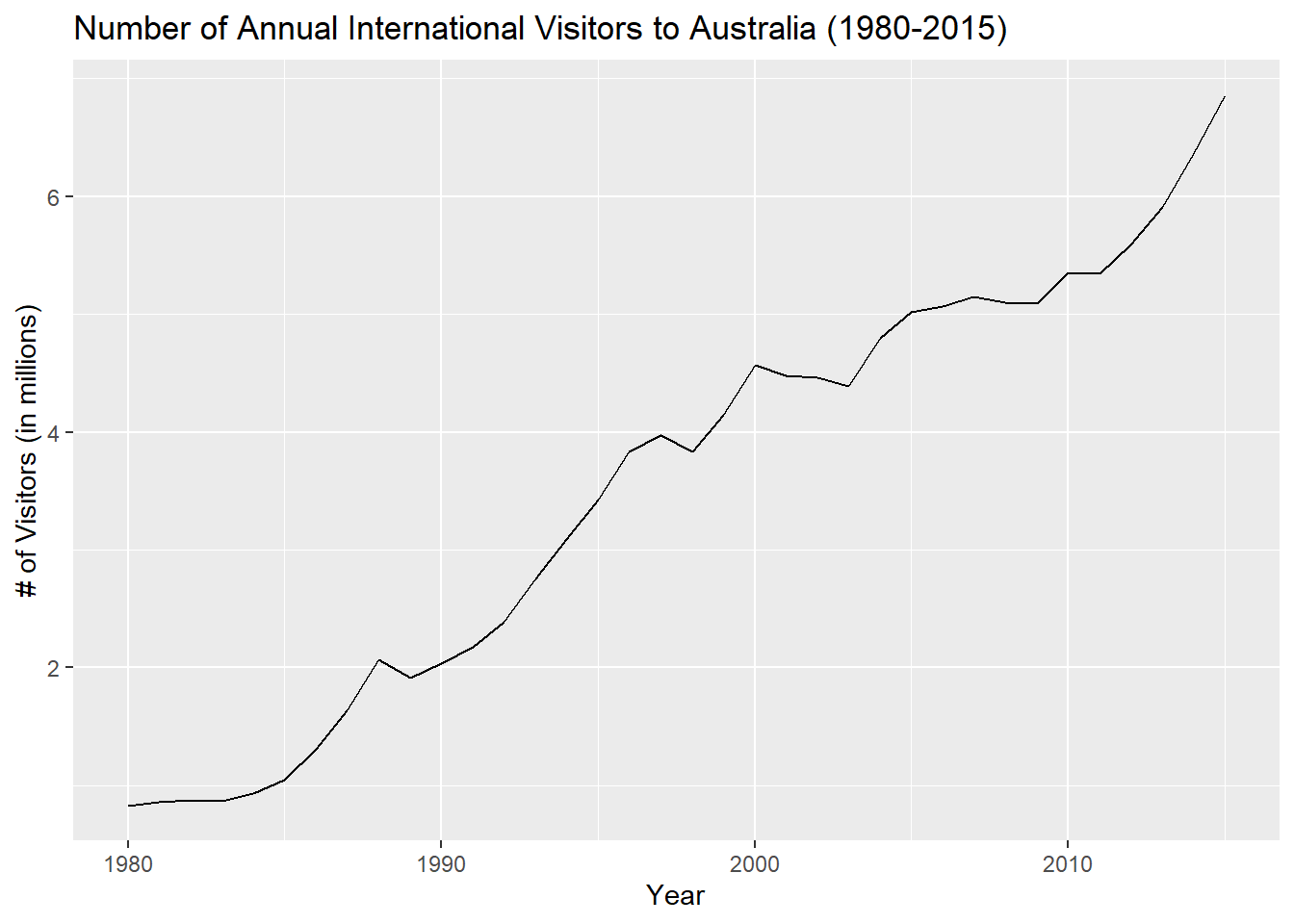
## $ Year : int 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 ...

## $ Visitors: num 0.83 0.86 0.877 0.867 0.932 ...

Visits = AusVisit$Visitors

Visits = ts(Visits,start=1980,frequency=1)

autoplot(Visits) + ggtitle("Number of Annual International Visitors to Australia (1980-2015)") + xlab("Year") + ylab("# of Visitors (in millions)")



In this course we examine a number of methods for making forecasts based on our observed data. One such method is the ARIMA model (AutoRegressive Integrated Moving Average model). Below without anymore sense than a rabbit as to what is happening we are able to fit an ARIMA model and make forecasts (with confidence) for the number of annual visitors in millions for the years 2011-2020.

fit = auto.arima(Visits)

forecast(fit,h=10)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

## 2016 7.108647 6.873183 7.344111 6.748536 7.468758

## 2017 7.282129 6.895831 7.668428 6.691337 7.872922

## 2018 7.455612 6.962652 7.948571 6.701695 8.209529

## 2019 7.629094 7.048756 8.209432 6.741543 8.516644

## 2020 7.802576 7.146393 8.458758 6.799031 8.806120

## 2021 7.976058 7.251932 8.700184 6.868603 9.083513

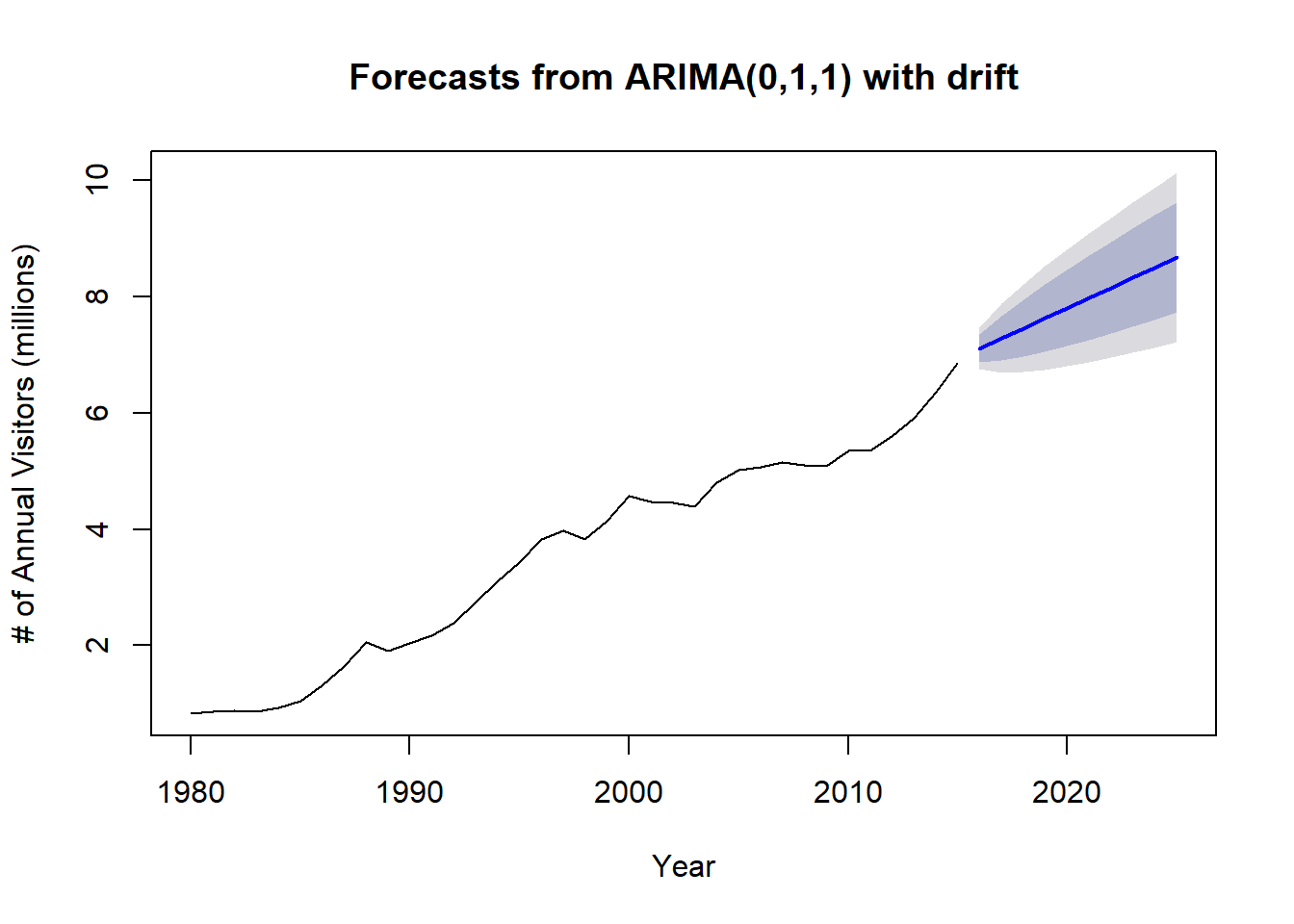
## 2022 8.149540 7.363321 8.935760 6.947121 9.351960

## 2023 8.323023 7.479266 9.166779 7.032609 9.613436

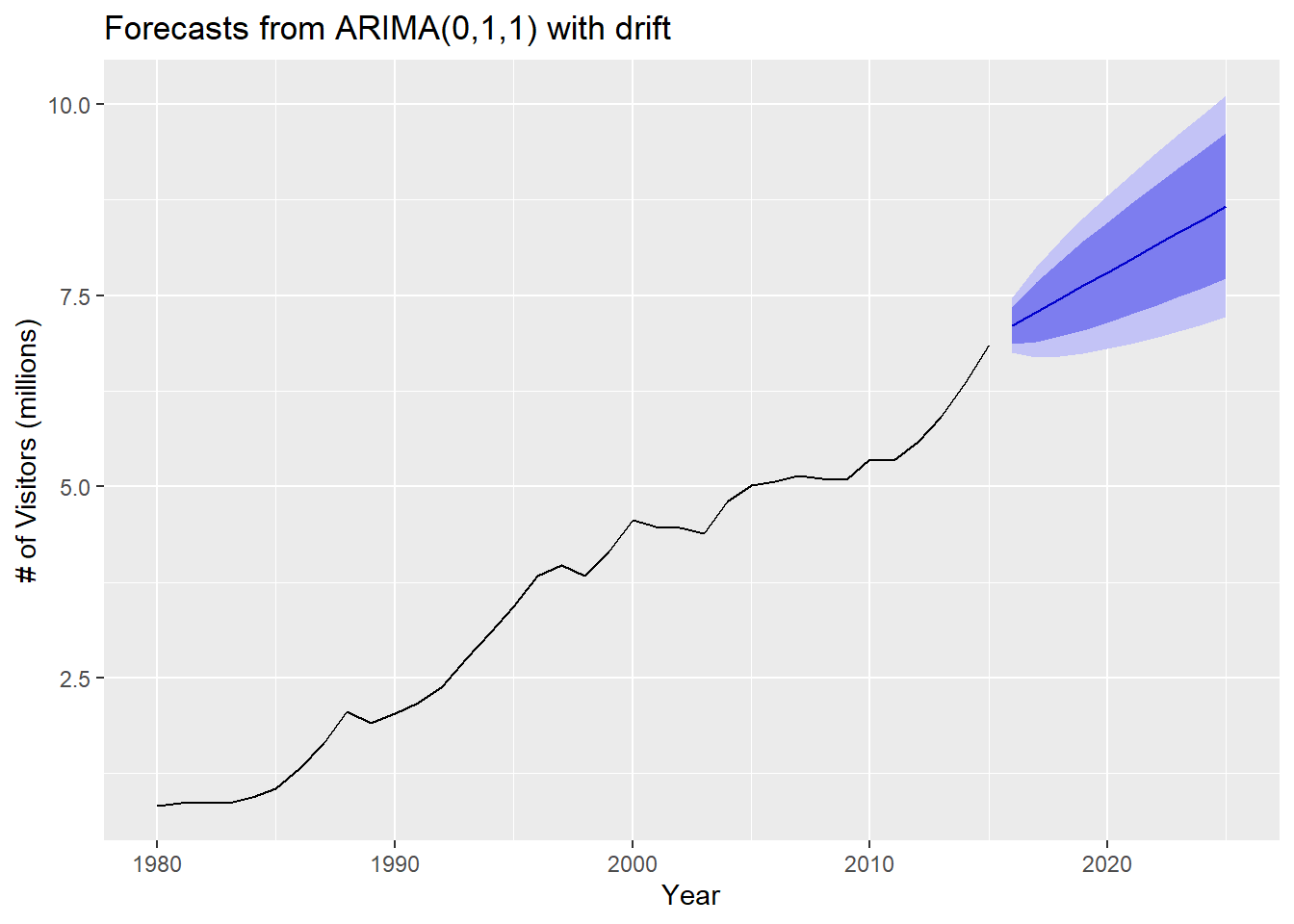
## 2024 8.496505 7.598893 9.394117 7.123725 9.869284

## 2025 8.669987 7.721572 9.618402 7.219512 10.120462

plot(forecast(fit,h=10),xlab="Year",ylab="# of Annual Visitors (millions)")



fit %>% forecast(h=10) %>% autoplot() + xlab("Year") + ylab("# of Visitors (millions)")



Notice that we created the same forecast plot two different ways, which is typical in R - “there is always more than one way to skin a cat”. You will find the author tends to use the latter approach, especially in later chapters of the text.

Examples from Introductory Powerpoint

These last few examples come from the introductory Powerpoint.

1.3 - CO2 Levels at Mauna Loa Observatory (1959-1997)

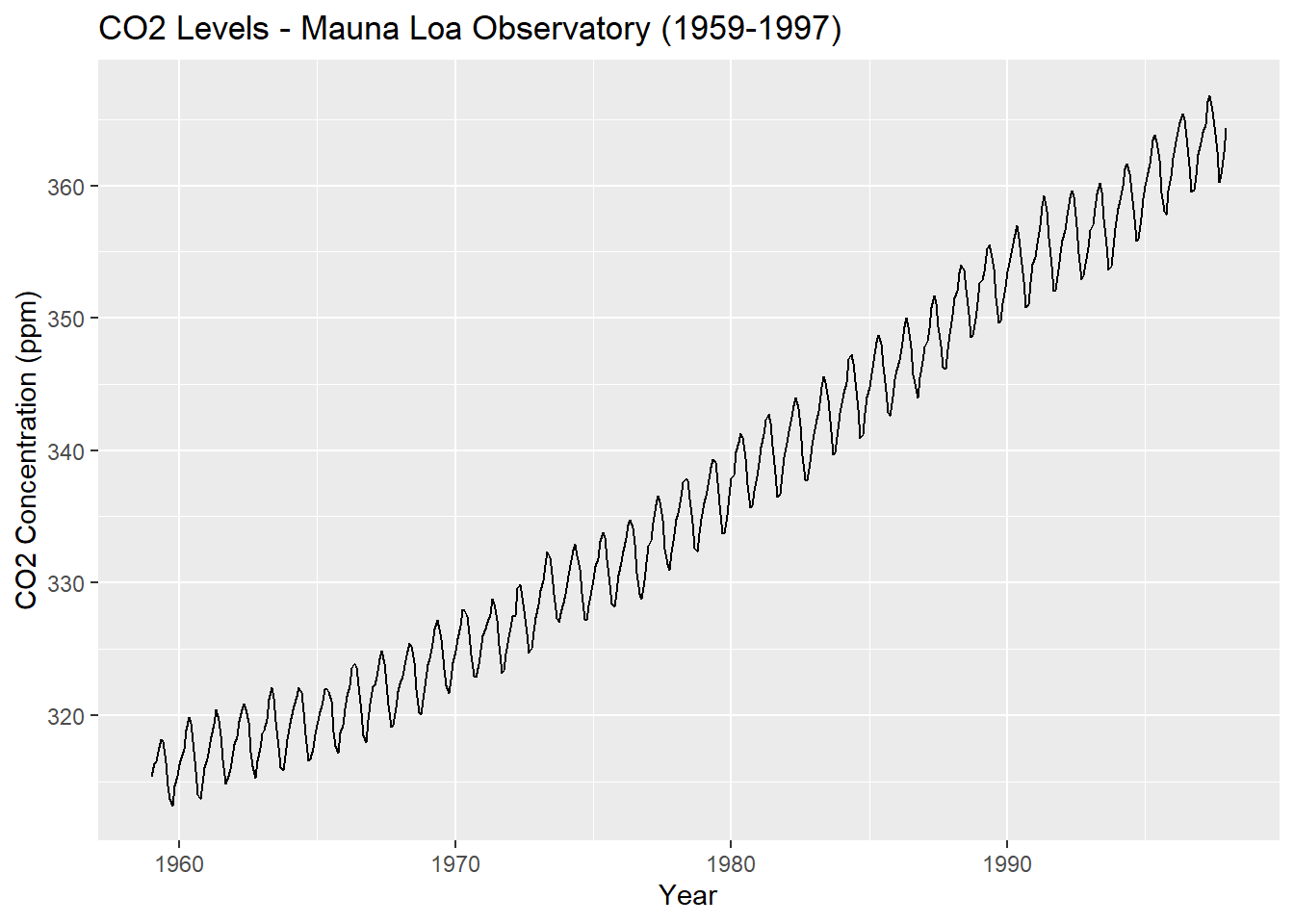
MaunaLoa = read.csv("http://course1.winona.edu/bdeppa/FIN%20335/Datasets/CO2%20Mauna%20Loa.csv")

names(MaunaLoa)

## [1] "Time" "Month" "Year" "CO2"

co2.ts = ts(MaunaLoa$CO2,start=1959,frequency=12)

autoplot(co2.ts) + ggtitle("CO2 Levels - Mauna Loa Observatory (1959-1997)") + xlab("Year") + ylab("CO2 Concentration (ppm)")



1.4 - U.S. Monthly Housing Starts (1959 - Present)

HousingStarts = read.csv("http://course1.winona.edu/bdeppa/FIN%20335/Datasets/US%20Housing%20Starts.csv")

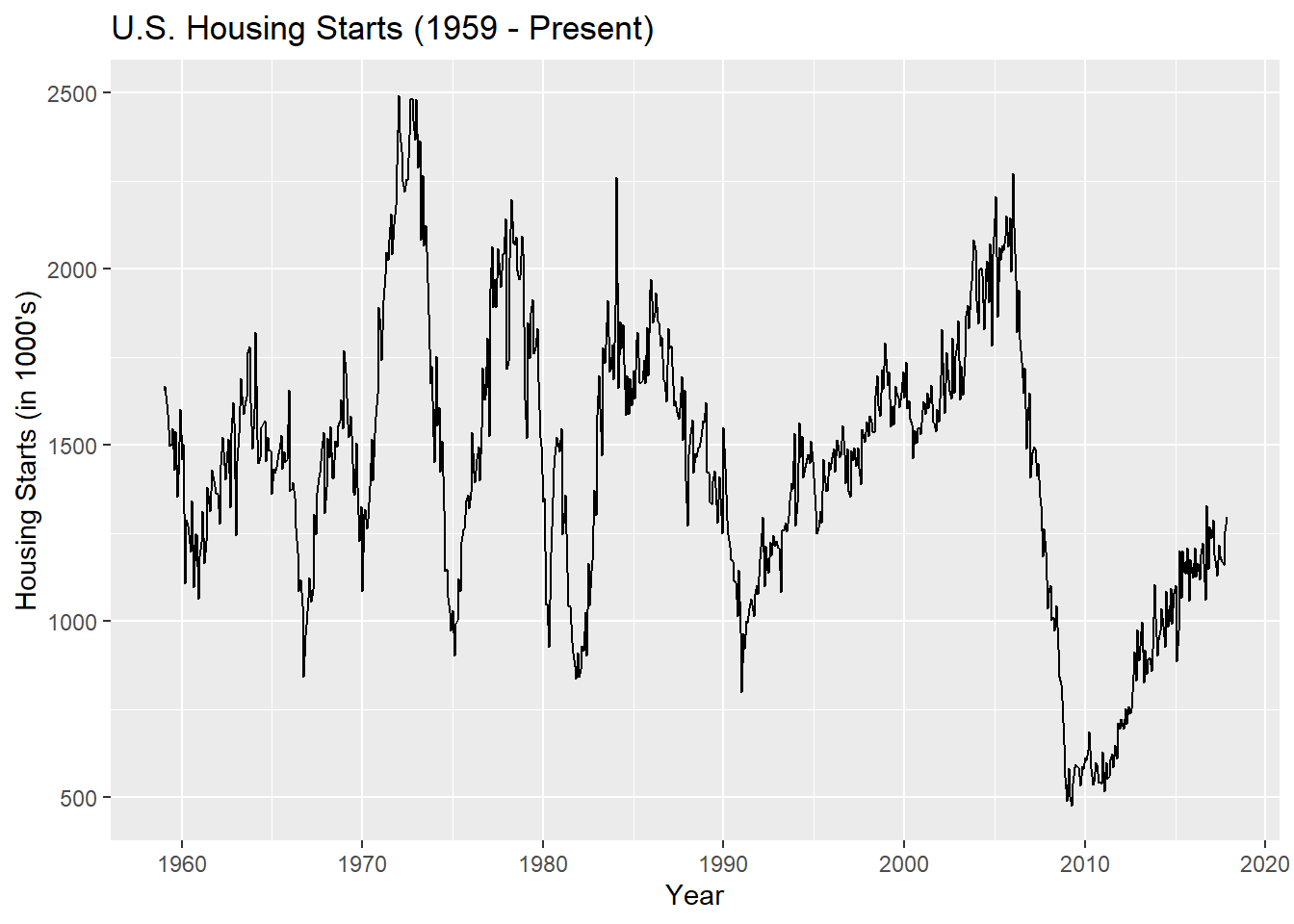
names(HousingStarts)

## [1] "Time" "Date" "Month" "Year"

## [5] "Housing.Starts"

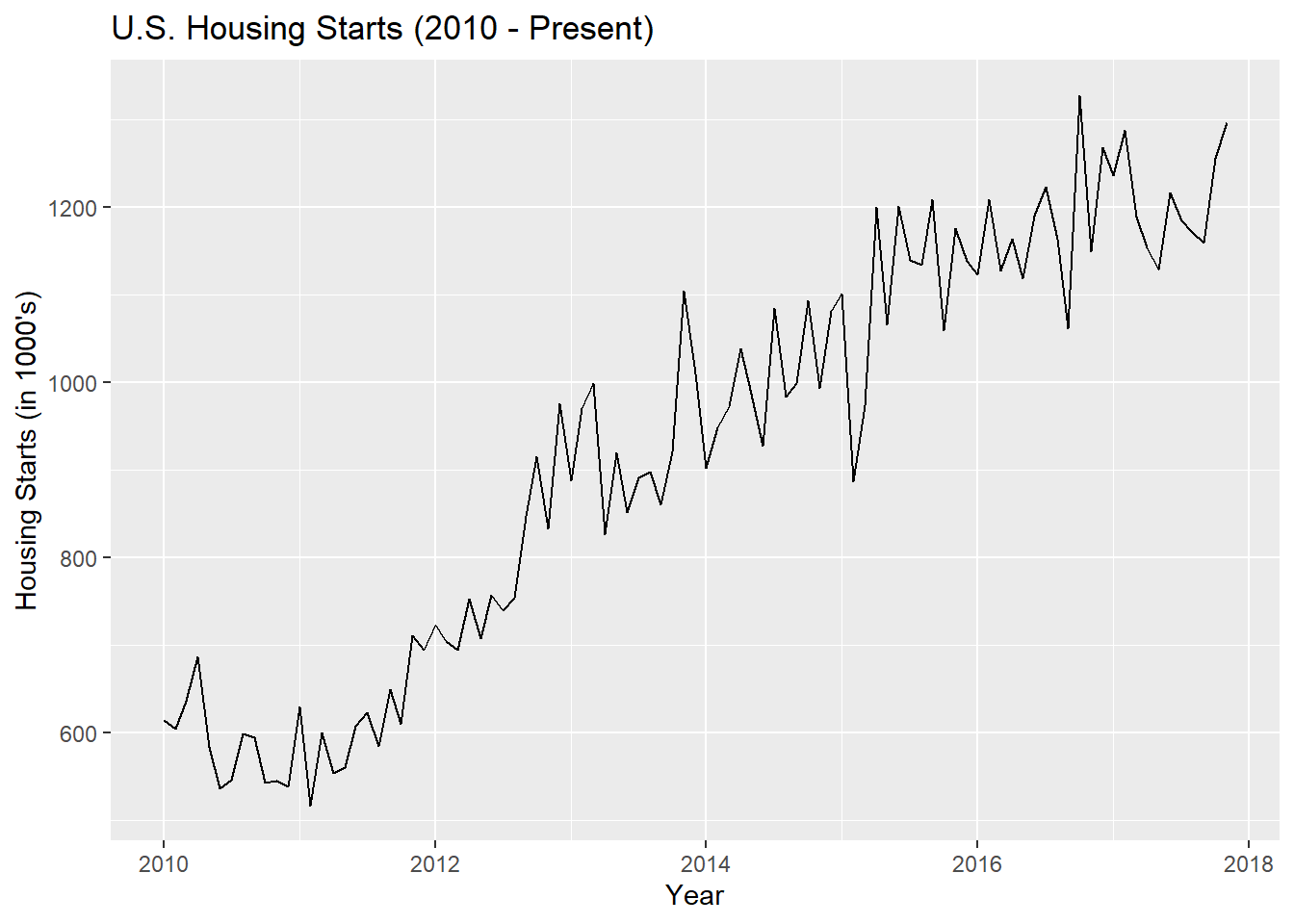
housing = ts(HousingStarts$Housing.Starts,start=1959,frequency=12)

autoplot(housing) + ggtitle("U.S. Housing Starts (1959 - Present)") + xlab("Year") + ylab("Housing Starts (in 1000's)")



hs.sub = window(housing,start=2010)

autoplot(hs.sub) + ggtitle("U.S. Housing Starts (2010 - Present)") + xlab("Year") + ylab("Housing Starts (in 1000's)")



1.5 - U.S. Monthly Liquor Sales (1980-2007)

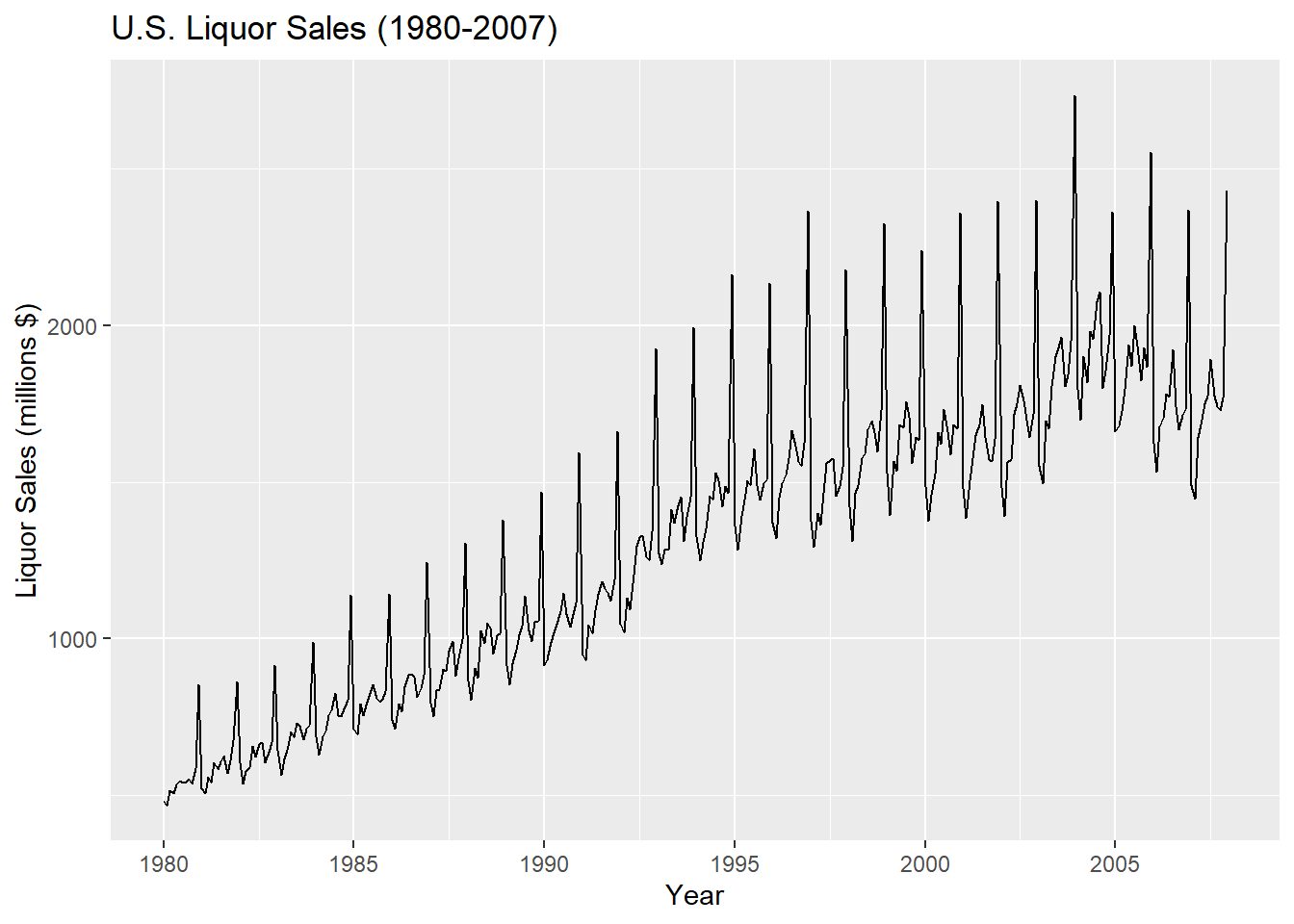
LiquorSales = read.csv("http://course1.winona.edu/bdeppa/FIN%20335/Datasets/US%20Liquor%20Sales.csv")

names(LiquorSales)

## [1] "Time" "Month" "Year" "Liquor.Sales"

liquor = ts(LiquorSales$Liquor.Sales,start=1980,frequency=12)

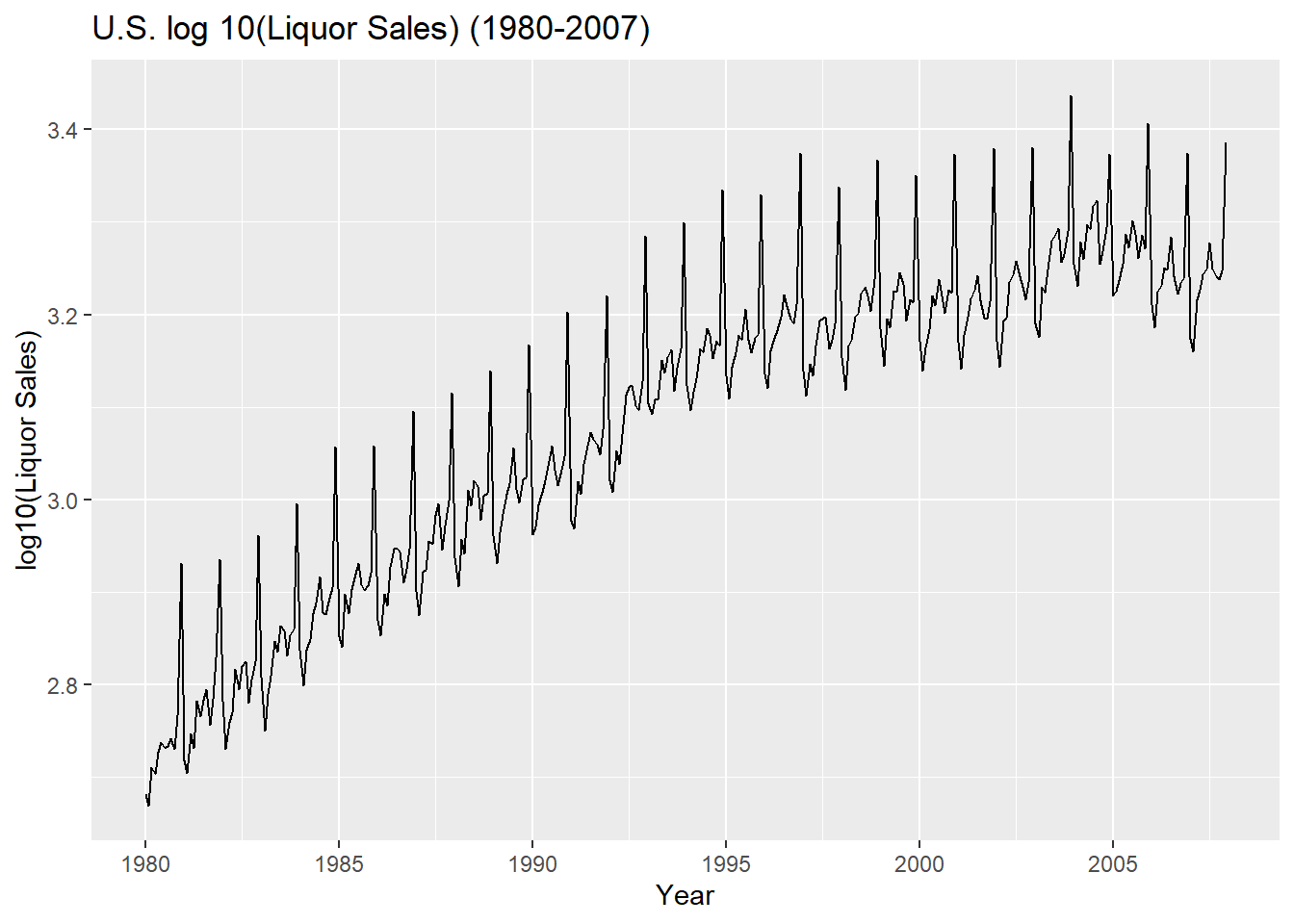
autoplot(liquor) + ggtitle("U.S. Liquor Sales (1980-2007)") + xlab("Year") + ylab("Liquor Sales (millions $)")



Sometimes we log transform the time series for reasons we will discuss later.

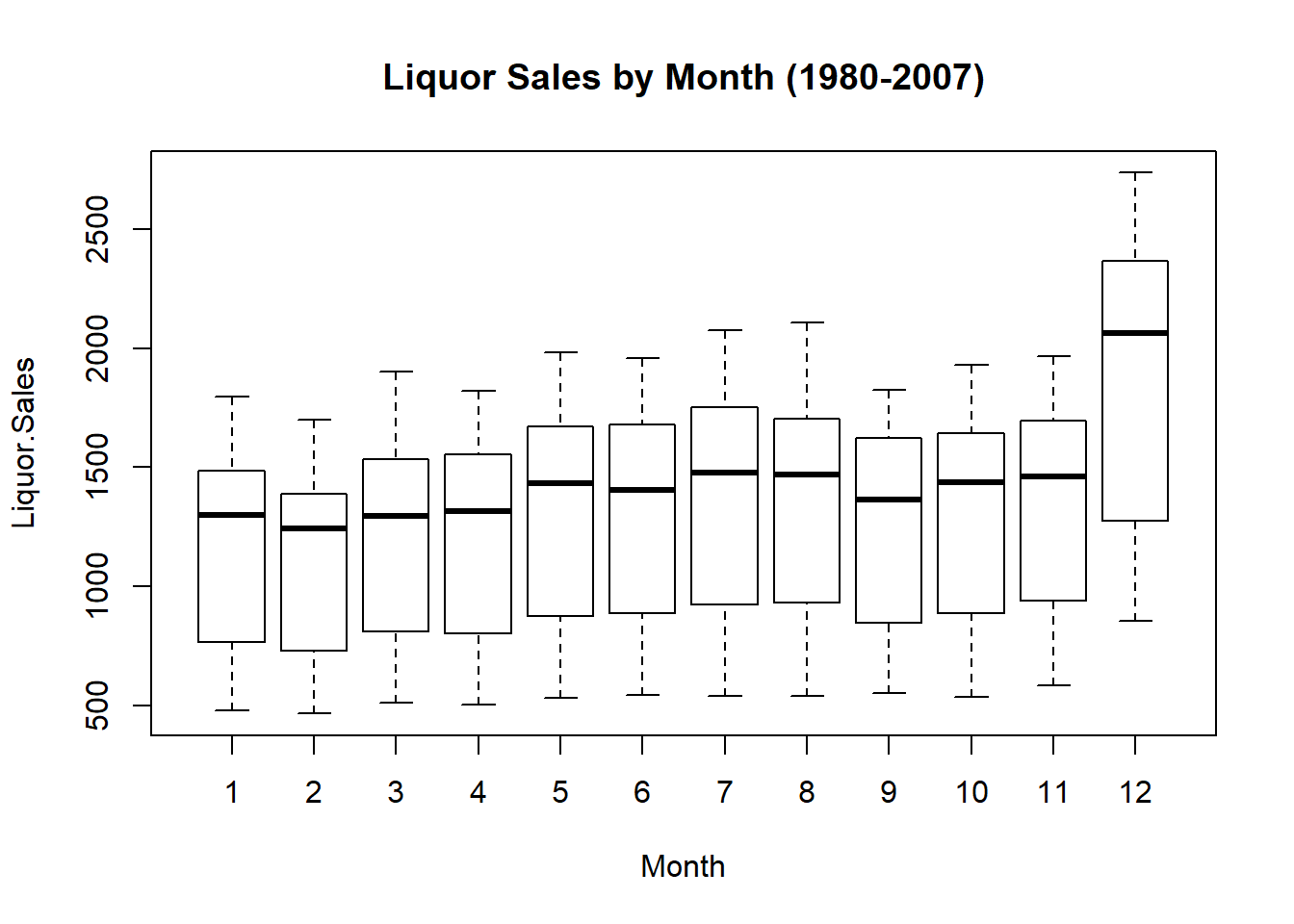
logLS = log10(liquor)

autoplot(logLS) + ggtitle("U.S. log 10(Liquor Sales) (1980-2007)") + xlab("Year") + ylab("log10(Liquor Sales)")

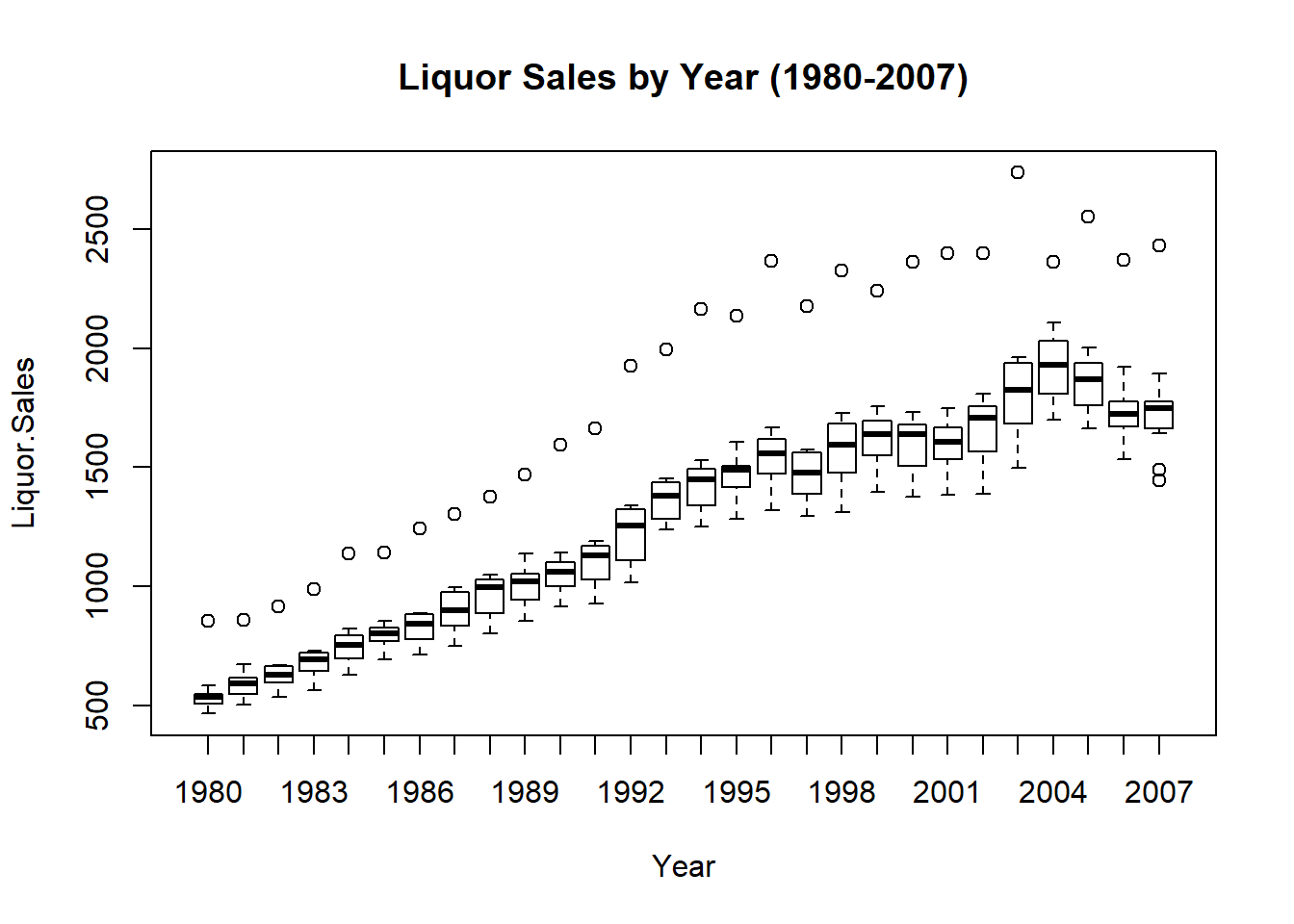


Boxplots can be effective tools when visualing trends/patterns in monthy or quarterly time series.

boxplot(Liquor.Sales~Month,data=LiquorSales,main="Liquor Sales by Month (1980-2007)")



boxplot(Liquor.Sales~Year,data=LiquorSales,main="Liquor Sales by Year (1980-2007)")



1.6 - Dow Jones Industrial Average

This time series is the weekly Dow Jones Industrial Average (DJIA) volume ending on Friday starting starting the week of January 10th, 2014 to present.

DJIAdf = read.csv("http://course1.winona.edu/bdeppa/FIN%20335/Datasets/DJIAtopresent.csv")

names(DJIAdf)

## [1] "DATE" "DJIA"

head(DJIAdf)

## DATE DJIA

## 1 1/10/2014 16460.12

## 2 1/17/2014 16397.86

## 3 1/24/2014 16216.06

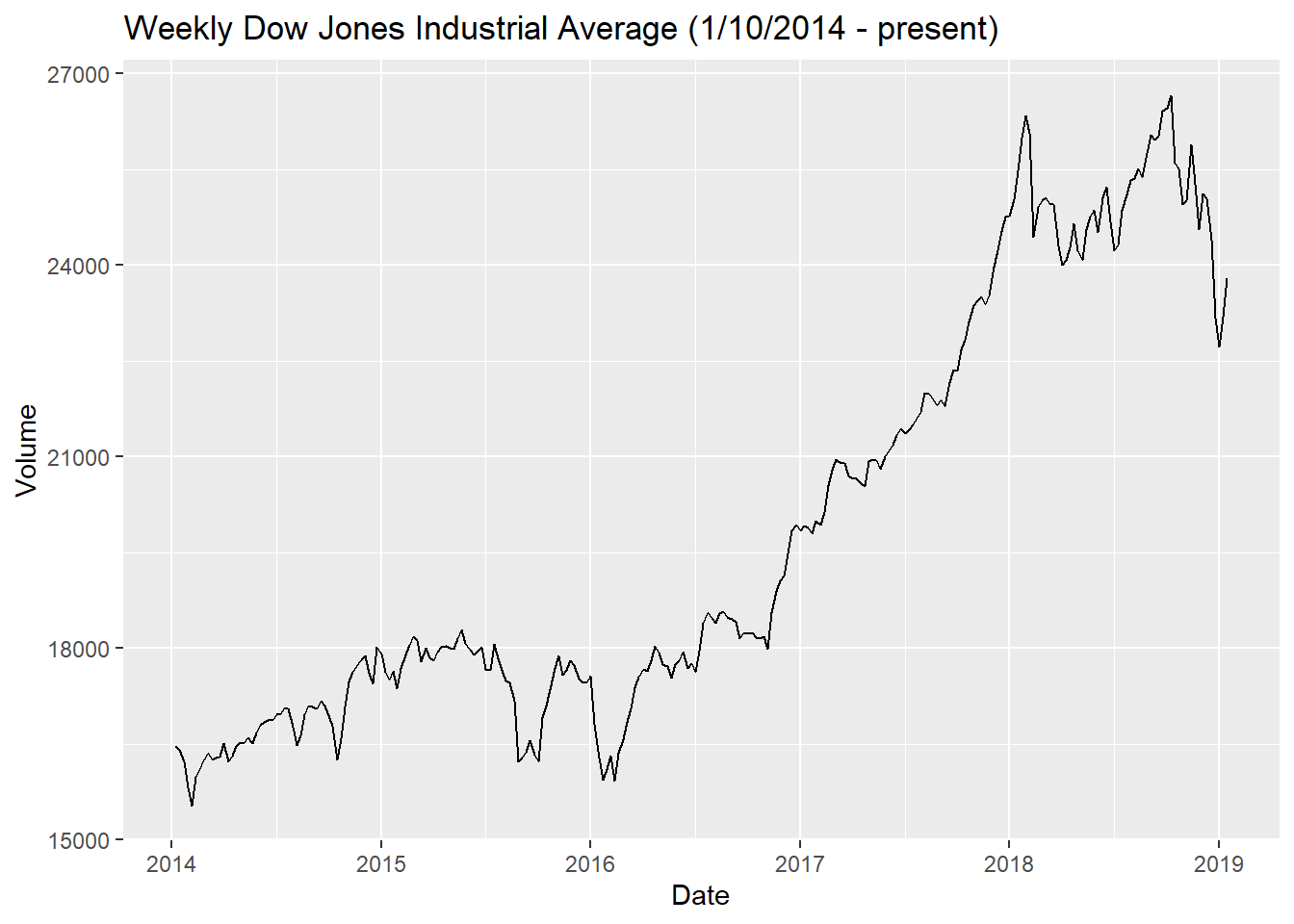
## 4 1/31/2014 15810.54

## 5 2/7/2014 15536.18

## 6 2/14/2014 15988.50

djia = ts(DJIAdf$DJIA,start=c(2014,2),frequency=52) *# series starts 2nd week of January 2014*

autoplot(djia) + ggtitle("Weekly Dow Jones Industrial Average (1/10/2014 - present)") + xlab("Date") + ylab("Volume")



Discussion

This was a lot of R to throw at you this early in the course, however you will find that loading time series data into R, plotting it, and building models from it follows a pretty consistent framework. I will be creating more R Markdown files like this as we progress through the course, so you will have plenty of example R code to follow. Furthermore, I will include starter code on some assignments to help you get going, particularly on the first few assignments in the course.