Chapter 1 Solutions

1.1.1 Exercises

Exercise 1.1

YouTube Video illustrating how to do this Exercise is at `https://youtu.be/giNlPUsseYU`

(a) For example, I looked at the term "baking" in the United States, Canada, United Kingdom, Panama and France since 2004 until the time this solution was written. Then I looked at the trend for the same term for the last 12 months, which included January to December of 2020. I looked at baking because a reaction of many people to the pandemic has been to cook and bake more and I wanted to see it there was some detectable change in the trend due to the pandemic. As it is, the pandemic resulted in a noticeable surge in the interest in baking in all countries.

(b) Looking at data since 2004, popularity of baking has been highest, although very similar to the US, in Canada, followed by the UK. The interest has increased linearly over time since 2004 in those three countries, with strong seasonality in the US and Canada, and not so much in the UK. France and Panama have very little interest in baking, compared to the other countries.

The short term trend for the last 12 months, show a sharp noticeable increase in the interest for baking in all countries except France. The rise is in the beginning period of the pandemic, March or May.

(c) Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

(d) The interest in baking has increased steadily over time in the US, Canada and the UK, with regular seasonality in all three countries. Panama and France show very little interest in baking, standing much lower than US, Canada and the UK. In Panama, the interest went up at the beginning of the pandemic.

(e) The effect of COVID-19 is clearly seen. More people turned to baking for consolation. But not to the same extent everywhere. And COVID-19 does not explain the seasonal behavior in Canada, UK and US throughout many years. That is probably the winter and holiday months, where it is traditional to bake pies, breads and other things in those three countries, but not so much in France and Panama, where bakeries are everywhere.

□

Exercise 1.2

Notice that the data and the plots may have changed slightly at the time that the reader looks at them.

(a) There is more data for Germany than for any of the other three countries (Maldives, Bangladesh and Congo, Dem. Rep). To interpret the following comments, it may help to just look at plots of pairs of time series that are similar

in

$$CO_2$$

level (for example, Congo and Maldive to start with), otherwise, the much higher levels of Germany obscure the comparisons. Now for the comments.

The shortest series is for Maldives. Germany started with steady growth in the 1800, but over time the trend has flattened, and even a noticeable downward trend is noticeable since the start of the 21st century. On the other hand, Congo also increased at first since data started in the 1920, but it is also on the downward trend . On the other hand, Maldives and Bangladesh, show a very sharp increase since the 1960.

(b) Bangladesh and Maldives, the fastest growing polluters, perhaps have less government regulation to prevent pollution, whereas Congo and Germany have more regulations in place, alternative technologies, and perhaps more pollution-aware residents, and that is why these countries are reducing emissions. For more accurate interpretations, the reader could conduct more research on pollution in these countries. The above are just possible things to look at.

□

1.4.1 Exercises

Exercise 1.3

The reader already knows the names of the seasonally adjusted unemployment rate (UNRATE) and the seasonally adjusted unemployment rate for Bachelor's degree (LNS14027662) used in Figures 1.3 and 1.4. So it is recommended to first search in FRED for the key words "seasonally adjusted" and "unemployment" and get the names of the series that interest the reader.

For the plot, first do the plot of UNRATE and then using the edit graph button and the tab at the top of the edit window that says "ADD LINE" enter one by one the names of the series that interest the reader in the "add data lines" window. Notice that we can also enter a key word in the "add data lines" window, and get names and series code for what we want in there. The reader can explore.

The reader should notice that when selecting unemployment series for which the name just mentions the age, or location, the series follow very much the pattern of UNRATE. However, when the reader selects a specific group, by ethnicity or education or other category, the series differ considerably from the UNRATE. Age alone, or location alone, are very aggregate. It is necessary to disaggregate by other variables to see differences among groups.

□

Exercise 1.4

The plot can be started by first plotting Unemployment Rate - College Graduates - Doctoral Degree, 25 to 64 years, Men, Percent, Not Seasonally Adjusted (CGDD2564M). Then use the edit graph button and ADD LINE tab and enter Unemployment Rate - College Graduates - Doctoral Degree, 25 to 64 years, Women, Percent, Not Seasonally Adjusted (CGDD2564W).

Compare. It is clear that throughout the years unemployment rate for women with doctoral degrees is higher than for men.

For the following steps it is possible that all the title (not just the variable code) needs to be entered. Then add to the plot Unemployment Rate - College Graduates - Master's Degree, 25 years and over, Women, Percent, Not Seasonally Adjusted (CGMD25OW) and Unemployment Rate - College Graduates - Master's Degree, 25 years and

over, Men, Percent, Not Seasonally Adjusted (CGMD25OM).It will be noticeable that Masters degrees have higher unemployment rates than doctoral degrees more often than not, but perhaps to appreciate the differences moving the slider at the bottom will help.

The plot with four curves on it can be hard to interpret, despite the colors being different for each time series. A suggestion is to look at the time series first by pairs and then on

These time series are not very long, extending from Jan 2013 to the present.

To determine whether being not seasonally adjusted makes a difference, the reader could highlight some vertical segments of the graph to see a smaller window of the data. It will be noticeable that while there is some annual seasonality in the Master's series, no seasonality is discerned in the Doctoral series.

When looking at the whole time period plotted, it is clear that there are cycles like those for the general unemployment rate observed in these new time series. The cycles take about four to five years to occur, sometimes more years. □

1.5.1 Exercises

Exercise 1.5

The time series is monthly and seasonally adjusted, according to the metadata. Thus we do not expect to see seasonality. Because there are observations from January 1992 (1992:1) there are enough years to be able to observe cycles if there are any. In FRED, click on *download* and select .csv. A file will download to your hard drive.

In RStudio, enter the following code line by line to read the time series and make it a `ts()` object and explore using the commands learned in Section 1.5.

```
 #set the working directory to the folder where the data file was downloaded
  data.monthly=read.csv("LNS14027662.csv", header=T)
 head(data.monthly)  # see first date and value of the ts
 tail(data.monthly)  # see last date

  # create an object of class ts
  data.ts=ts(data.monthly[,2], start=c(1992,1), frequency=12)
      #notice that we select only the second column of data
      #notice that we enter only start date, as R will read until the end by default.

### We do a time plot

plot.ts(data.ts, ylab="Unemployment rate",
main=" Unemployment rate- Bachelor's degree and higher, \n 25 years and over.")

### We check that the commands of Example 1.11 work

  start(data.ts) # double check start date
  end(data.ts)  # double check end date

  length(data.ts)  # check total number of observations

  cycle(data.ts)  # notice index for month 1 to 12.

    #explore the window() function to subset
```

```
    window(data.ts, start=c(1993,1), end=c(2001,10))

#play with the aggregate function. For example, let's
# convert the time series to a quarterly time series.

   data.quarterly= aggregate(data.ts, nfrequency=4, FUN= mean)
```

□

Exercise 1.6

According to the metadata UNRATE is a monthly seasonally adjusted, and is recorded since 1948:1.

We just modify slightly and add a few lines to the R code used for Exercise 1.5 to obtain the answers.

When running the next R code, we find in response to part (b) that the overlapping times start on 1992:1, and end with the most recent month and year in which you access the time series, assuming the series are not discontinued by FRED. For part (c) we obtained a window ranging from 2017:1 to 2021:1.

```
#set the working directory to the folder where the data file was downloaded

## (a)
# since we have now two time series, it is good to use names that distinguish them

unemp.bachelor=read.csv("LNS14027662.csv", header=T)  # read the .csv file downloaded from FRED

########## Reading and making ts LNS14027662  ############

head(unemp.bachelor)  # see first date and value of the ts
tail(unemp.bachelor)  # see last date

# create an object of class ts  for unemp.bachelor
unemp.bachelor.ts=ts(unemp.bachelor[,2], start=c(1992,1), frequency=12)
#notice that we select only the second column of data
#notice that we enter only start date, as R will read until the end by default.

start(unemp.bachelor.ts) # double check start date
end(unemp.bachelor.ts)  # double check end date

length(unemp.bachelor.ts)  # check total number of observations

cycle(unemp.bachelor.ts)  # notice index for month 1 to 12.

    ########## Reading and making ts UNRATE  ############

    unrate=read.csv("UNRATE.csv", header=T)  # read the .csv file downloaded from FRED
    head(unrate); tail(unrate)

    # create an object of class ts  for unrate
    unrate.ts=ts(unrate[,2], start=c(1948,1), frequency=12)
    #notice that we select only the second column of data
```

```
#notice that we enter only start date, as R will read until the end by default.

start(unrate.ts) # double check start date
end(unrate.ts)  # double check end date

### (b) Determine the intersection of the two series ##########
 both.ts=ts.intersect(unrate.ts, unemp.bachelor.ts)
 class(both.ts)  # this is a mts object (multiple ts class)
 start(both.ts)  # the date columns are just time indexes.
 end(both.ts)



 #### With the new time series, make a window of our choice ###

 both.ts.window=window(both.ts, start=c(2017,1), end=c(2021,1))
```

□

1.6.9 Exercises

Exercise 1.7

In searching for the article, the reader should notice that the title should be *2017 Traffic Data for U.S. Airlines and Foreign Airlines U.S. Flights*. Because the article was written in 2018, it will not show anymore on the page provided. But from within that page, in the "Search" window, enter the title of the article (corrected as above) and the article will appear. To see the plots, you will need to click on the titles of the tables provided in that article. Once you do that, you will notice the following?

The BTS chose to display the trend only, which is increasing. They do not explain what they did to obtain just the trend. In fact, the graph is not even mentioned in the narrative of the article. The detail of the seasonality is not showing because they just estimate the trend, and instead of superimposing it on the original time series like we did in Figure 1.10, they do no show the original series. They must have done seasonal adjustment by using some decomposition of the series, smoothing.

□

Exercise 1.8

The pattern of increasing variability and seasonality in the time series analyzed in that video resembles the one in Figure 1.11. The pattern of seasonality is like that in Figure 1.7, but we do not see increasing variability in Figure 7 probably because the time series is shorter. The author in the video identifies a point in time when there was a world rugby cup and another point when there was a financial crisis. We can not see any points that stand out in the domestic passengers time series that we analyze.

□

Exercise 1.9

This solution goes with Program *ch1Roomstimeplot.R*. Add to the program what is requested from *ch1passengersplots.R* to obtain the following conclusions. Alternatively, copy and paste the code into the *ch1passengersplots.R* file to replace the reading of the data given there with the reading of the rooms data. Then use the latter code file to answer the questions.

(a) The room time series ends in December 1990. It has 168 observations. We do not see missing values (no NA recorded) but lacking more metadata, that is not to say that we are certain that there was an observation for every single month. It could be that some month was not recorded and there is no indication of it by using an NA or some other missind data code. To reassure the reader, there are no missing values, all months had an observation. But if we did not tell you, certainly, you would need to agree that there is not enough metadata to say that indeed the last observation is December 1990.

(b) See page 23 of the book. If the ts object created is called `rooms`, then the window can be obtained with code

```
rooms.window=window(rooms, start=c(1980,1), end=c(1988,12))
```

(c) The time plot of the period 1980-1988 indicates that the number of rooms is increasing over time, with pronounced seasonal swings every year. The spaghetti plot indicates that, every year, the month with the largest number of rooms is August. The months February and November have the lowest value of rooms. From February until August the number of rooms increases overall and after that it declines overall. Also noticeable in the spaghetti plot is that the spaghetti is higher in the plot the later the year, indicating growth in the number of rooms with each year, noticeable each month. The seasonal box plot confirms the information of the spaghetti plot and adds the variability information. We can see that the month with the most variability across the years is July, as indicated by the length from end of lower whisker to end of upper whisker, followed by August. The month with the least variability is March. The lines in the middle of the boxplot are medians. There is some skewness in the distribution of the number of rooms each month (for that reason, perhaps it is better to look at variability by looking only at the box (which indicates the interquartile range), in which case, August would be the most variable).

□

Exercise 1.10

This solution goes with Program ch1-JJ.R

Run the program function by function to observe the following. The time plot reveals that earnings per share increased over time, with increasing seasonal variability that seems proportional to the trend. According to the seasonal box plot the most prominent seasonal upswings are in the second and third quarters (the medians are slightly higher). The spaghetti plot confirms that, with the peaks for each year being mostly in those two quarters (some years in the third and other years in the second). The increasing variability shown in the time plot reflects itself in the spaghetti plot by the higher curves of the last years (the curve for 1980 is much higher in the plot than the one for the ones in the 1960).

□

Exercise 1.11

To make a comparison, you need to realize that you should look at the same time period. During approximately the same time period, the data sets of all countries show some seasonality, but the trends are different in each country. The seasonalities are also different. New Zealand is in the Southern Hemisphere, hence their Summers occur in the months when it is Winter in the Northern Hemisphere.

□

Exercise 1.12

To obtain the answer for this exercise, run Program *ch1amazon.R*

There isn't a well defined trend, but a trend would be hard to discern clearly with so few observations, as we saw in Section 1.1 of the book. With few observations, what look like trends could just be the up or down swing of a cycle. There is no seasonality. And there is no clear cut relation between the two time series. In some years, the high and the low move in opposite directions, in other years they move together, meaning that if for example the high increases so does the low. □

Exercise 1.13

This problem goes with Program *ch1precipitation.R*

This is a univariate data set. The data set is monthly, and it is presented with the year as column 1, and the value of precipitation in each of the months October to September in 12 columns. In order to convert it to a `ts()` object and work with this data set, it has to be reshaped. Then it has to be sorted by year. After doing all that, the precipitation value in the data set can be converted to `ats()` object and plotted as a univariate time series.

This time series looks very different from the data sets seen so far. Precipitation has no upward, lower or any other pattern of trend and varies a lot. There is also clear seasonality, with the lowest median values occurring in July and August. Values in those months are very similar across the years, but that is not the case for other months. We can see that in the height of the boxes of the seasonal box plot.

□

1.8.2 Exercises

Exercise 1.14

This solution goes with program *ch1uber.R*

It would be nice if the reader can read

`https://medium.com/uber-movement/working-with-uber-movement-speeds-data-cc01d35937b3`

It says there that: This dataset provides the average speed on a given road segment for each hour of each day in the specified month. Only includes road segments with at least 5 unique trips in that hour. The variables are given in that web site.

(a) I selected segments 239464357 to 4318478540 , 4179595252 to 65293753, 65328185 to 65332198. and 3967052495 to 57801307.

(b) Means, standard deviation, minimum, maximum, interquartile range, medians are basic summary features to start with. As indicated in the chapter, Section 1.8, automated summary feature generation with some of the specialized packages are possible, but this exercise is just introductory, and to practice the method. So we use only those basic features that we hand-programmed.

(c) Use the R program to create the features. After completing other exercises in the book that use the `tsfeatures` package, the reader may want to come back to this problem later and use, for example, that package to obtain many more features. The R program identifies a cluster for each travel segment. We assumed two clusters. The result is that three segments are put in one cluster and one segment in another cluster.

(d) We ended up with segments that start with 65328185, 239464357 and 4179595252 in one cluster and segment that starts with 3967052495 in another cluster.

To identify the features with most distinct means across clusters, we produce in the R program the means of each of the two clusters. Actually, `kmeans` produces them automatically. Here they are, with cluster 1 denoting the cluster with segment that starts with 3967052495.

```
Cluster means:
    avsmean  avssd   minsmean maxsmean   minssd  maxssd  iqrsmean iqrssd    mediansmean medianssd
1 43.61220 4.630400   41.781 45.50000 3.861000  5.60800 2.590000 0.616000    42.96000     4.529
2 17.20589 5.478913   11.772 25.81633 1.134333 15.09967 2.652167 1.493833    17.04867     5.098
```

We can see that features that are quite distinct are avsmean, minsmean, and a few others. We will select avsmean (average speed mean) and maxssd (maximum speed standard deviation) to plot the clusters. See the R program to see how we do it. The reader may want to experiment plotting other features.

□

1.9.2 Exercises

Exercise 1.15

There is no January 2002 in the data set needed. The time series starts in October 2002. Use Program *ch1passengersprogram* to produce the plots needed for period October 2002 to December 2005. Then the output should be compared with the images produced by Program *ch1passengersplots*. Do not compare the plots obtained with *ch1passengersprogram* to those in Section 1.6 (See comments in the Errata Sheet for the book).

□

1.12 Problems

Problem 1.1

(a) See section 1.1. in Chapter 1.

(b) See page 5 of Chapter 1.

(c) Monthly data is data on a variable that has been collected once per month. See Example 1.7 and Example 1.13.

(d) To appreciate the sequential nature of the data, and to be able to observe the seasonal patterns and trends.

(e) A seasonal pattern. The repeated pattern observed each cycle (a year, a week, a day) due to recurring phenomena, e.g., holiday shopping.

(f) Sort the data in temporal order, clean it.

□

Problem 1.2

This problem requires program *ch1problem1.R*. Review Section 1.5 before answering.

(a) You find out the metadata for these series by using the R statements `?AirPassengers` and `?JohnsonJohnson` in the R program. Notice that R datasets do not come with a lot of metadata associated with the series. For example, R does not indicate whether the series are seasonally adjusted or not.

(b) One is monthly (which?). The other is quarterly (which?). By executing the functions `start()`, `end()` and `frequency` you will find the answers to these questions.

(c) Hint: The `AirPassengers` data resembles lightly the plot of Figure 1.11. So you may use the description used for that image and compare the differences observed. The `JohnsonJohnson` time series is a more extreme pattern, in that although resembling the `AirPassengers` data in the second part of the data, the first part differs. Can you see the differences?

(d) Execute `cycle(AP)` in *ch1problem1.R*. For each year, you obtain 12 numbers, one for each month. The number 1 representing January, the last one representing December. The function `time(AP)` gives a time index that contains for each month, something1.something2. The something1 before the dot is the year. The something2 after the dot is a fraction of twelve, For example, for January 1949, we have 1949.000, for February 1949 we have 1949.083 (where the 083 is one twelfth), for March we have 1949.167 (where 167 is rounded 2/12) and so on. The information can be obtained for `JohnsonJohnson` using similar commands. Notice the difference when using those commands.

(e) The seasonal box plot of AP is conveying the message that median monthly totals of airline passengers is highest in July, decreases in the nearby months before July and after July and again increases as it approaches December to decrease again after.

You can see this pattern in a different way using the spaghetti or seasonal time plot. In the *ch1problem1.R* you will find the code to do that. Similar code was used to obtain Figure 1.12in the book.

□

Problem 1.3

Go over the commands in Program *ch1problem2.R* to do all that is asked in this problem. Do not just source (run) all the program at once but rather try to go line by line to see what the program is doing.

(a) Aggregate the `AP` and `JJ` time series of Problem 1.2 to make them total annual first. Then aggregate to get average annual values. You will do the latter in two ways.

(b) Obtain the time plots of the aggregated time series. Add to the code given in Program *ch1problem2.R* meaningful arguments `ylab` and `main`.

(c) We learned about the trends. That is what observe when we aggregate or average. The plots reveal that whereas the AirPassengers time series has a rather linear increasing trend, with a couple of inflexion points, the Johnson-Johnson data set has an exponential increasing trend. Both the plots of the totals and the averages indicate the same.

The annual total and the annual average smooth the time series. Because we are just showing the smooth series, instead of the original raw series and the smooth ones superimposed, We do not see any longer the seasonal swings observed in the raw data in Problem 1.2. We could also see the seasonal swings just by plotting the AP directly in one plot and the JJ directly in another, as we did in Problem 1.2, by adding R commands to obtain them. We added those to Program *ch1problem2.R*

□

Problem 1.4

The following R code is to get you started with this problem. As you can see, it is the same code used to start Program *ch1passengersprogram.R*. Instead of selecting column 3, though, we select now column 4.

```
###
data=read.csv("ch1passengers.csv", header=T)
```

```
attach(data)
head(data,50)  #view content
## We can see the commas
tail(data)
str(data)   # types of variables
## notice that the commas make the numbers characters.


y=data[,4]   # extract international passengers from raw data
ywocomma=as.numeric(gsub("[,]", "", y)) # remove commas
head(ywocomma)
y11=ts(ywocomma,start=c(2002,10), end=c(2019,6), freq=12) # the whole ts
class(y11)

passenger.int=window(y11,start=c(2012,1),end=c(2017,12),freq=12) # the window requested.
class(passenger.int)


# Do a correct time plot (a line plot) for international passengers

par(
mfrow=c(1,1),
font.axis=2,
mar=c(5,5,5,5),
font.main=2,
font.lab=2
)

### Notice that the plot is slightly different from Figure 1.7,
## on the left and the right of the plot,
## due to us incorrectly entering the start date for x11 at 2002:2
## when publishing.

plot.ts(passenger.int,
main="Time series with long term\n upward trend and seasonality",
ylab="Number of International Passengers", xlab="Time (months)", lwd=1.5,cex=0.5)
dev.off()


### Continue applying the code of Program {\it ch1passengersprogram.R} but making
## sure that the right variable name is inserted in the R code when needed.
## Just replace the domestic passenger variable name with the international one.
```

□

Problem 1.5

The time series is updated monthly so your number of observations could be different. But since we are choosing a window within the available time series data, we all should get the same conclusion. That window requested is before the year book was published, i.e., before 2023.

Using the following program, which reads the file you downloaded from FRED, we obtain the required plots.

- The plot of the whole time series is much more revealing, indicating that California had several cycles of unemployment growth and decline, each cycle lasting on average, from trough to trough, approximately 10 years (except the cycle between 2001 and 2006 approximately, which is shorter). We observe the anomalous unemployment starting in 2020 until 2022, due to the COVID-19 pandemic.
- The plot of the window of the data, which covers 6 years, is showing part of the downward phase of the 1990 to 2000 cycle, and the upward phase of the smaller 2001-2006 cycle. We do not learn much from this window. If someone gave us this data set without telling us where it comes from, and without giving us the longer series, we would just might be tempted to conclude that there is a downward and an upward trend in the data, but this would be tentative given that we would qualify that by saying that given that this is just a short time series, the image could just be a part of a cycle, which it is.

```
# set working directory to where the time series data file is.
caur=read.csv("CAUR.csv", header=T)
head(caur) # From January 1976
head(caur[,2])


caur.ts=ts(caur[,2], start=c(1976,1), frequency=12)
caur.ts   # the end of the series will differ depending on when you download the data

caur.window=window(caur.ts, star=c(1997, 1), end=c(2003,3),frequency=12)

par(mfrow=c(2,1))
plot.ts(caur.ts, main="California's monthly unemployment,\n
1976:1-2023:10,seasonally adjusted",
ylab="Unemployement rate",
xlab="Time (year, month)")

plot.ts(caur.window, main="California's monthly unemployment,\n
1997:1-2003:3,seasonally adjusted",
ylab="Unemployement rate",
xlab="Time (year, month)")
```

□

Problem 1.6

The article can be found also at `https://academic.oup.com/jrssig/article/12/6/34/7029090`

The data set for this article is not available, so we can not replicate the results. However, the methodology used to study can be replicated with other dataset . The author uses multiple regression (see Chapter 9). The variables in this regression capture past values of the dependent variable (number of calls in previous days), a trend, seasonalities of different granularity (more on this in later chapters), holidays and special days. Models like that are functional models widely used nowadays to handle data with multiple seasonalities. Sometimes they are good forecasting models, but other times they are not. Depends on the data analyzed.

(a) The article is about a time series: daily volume of calls to a private insurance company call center.

(b) The data for air passengers and the data for calls are both seasonal, but whereas the air passengers data has one seasonality, monthly seasonality, the data in the article allegedly presents multiple seasonalities, due to its granularity being daily. We do not see a time plot of the data in the article, so we deduce that from the narrative of

the article. The author talks about a day of the week being heavier in calls than other, a time of the month being heavier, a particular quarter being heavier. The different seasonalities are very important in the volume of calls to the call center.

(c) Zelin uses dummy regression independent variables to represent the different seasonalities: dummies for quarter, dummies for time of the month, dummies for the month, and dummies for the day of the week. These dummy variables make the regression model rather large.

(d) The trend is modeled by the week number, and according to the author is increasing, the author says.

(e) Answering this requires familiarity with multiple regression with dummy variables. Models like the one in this article are widely used today to approach the modeling and forecasting of data with multiple seasonalities. They fit data well though, if the multiple seasonalities are very regular each cycle, and if the trend is well captured. But they do not perform very well if that is not the case. Chapter 9 of the book is about models of this kind. Supervised machine learning models that are regression based, use features like those used in this article to do regression-like modeling and forecasting.

□

Problem 1.7

Review Section 1.5 before answering this question.

Program *ch1airquality.R* will give the answers to those question as you execute it, function by function. Do not run all at once. The program has labels corresponding to the questions asked in this problem, so it will be easy to know which is the answer for each of (a), (b), etc.. if you execute those sections in that order, stopping to see what each part of them produces.

□

Problem 1.8

A good data to practice, given the skills gained so far in Chapter 1, is the `NN3` data stored in the `tscompdata`. It contains 111 monthly time series of class ts (the class introduced in this chapter). The data is an R list. Each element of the list is a time series. You may use the following R code to practice the code learned in this chapter and the exercises.

```
devtools::install_github("robjhyndman/tscompdata")
library(tscompdata)
?nn3  # to see some information about the data. There isn't metadata
nn3.1=nn3[[1]]  # extract the first time series from the list
nn3.100=nn3[[100]] # extract the 100th time series from the list

## inspect the length of all the time series
for(i in 1:111){print(length(nn3[[i]]))}  # we notice that the series have different lenghts
# See section 1.8. One thing we could do with these is to cluster based on summary features

## inspect the start time and end time of all the time series
for(i in 1:111){print(c(start(nn3[[i]]), end(nn3[[i]])))} ## they vary

### Some time series will intersect, but others will not.
common.times.ts=Reduce(ts.intersect, nn3)
start(common.times.ts)
end(common.times.ts)
```

```
# 46 series intersect in 1983:1 to 1984:12, two years. Others do not intersect
```

Given the results of the exploratory analysis, we can conclude that due to the different times periods covered by the series, clustering them using their summary features, like we did in Section 1.8 could certainly give us separate clusters, but this result is confounded by the fact that the clusters could be due to just observing different periods in each time series. We could however just try the clustering as a mere exercise in clustering. Revisit Programs *ch1uber.R* and *ch1riverscode.R* to use code there to create a dataframe of summary features of all the time series and do cluster analysis.

□

Problem 1.9

We used very simple summary features for the rivers in Section 1.8, Example 1.14 and in Exercise 1.14 (revisit the R programs for those), because we just wanted to illustrate the process of creating summary features and a data frame with the features of each of the multiple time series (one series for each river).

But there are many more summary features we could look at, and we should look at them, particularly if the time series have trends and cycles, and other interesting patterns. A good exercise at this point is to study the summary features that the automated `tsfeatures` package.

Here is a suggested sequence for the work to do:

- Go to `https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html` and look at the list of summary features used by the automated feature generation R package `tsfeatures`.
- Watch the YouTube video `https://youtu.be/giNlPUsseYU` and read Section 1.1. In the video, there are instructions on how to access google trends time series. Create an image in google docs containing the long term trend for at least 6 countries. Do not use the short term trend automatically displayed by google trends.
- Go back to the list of features in the package and see which ones make sense to use. Do not blindly use features or features that you do not understand, as that would confound the clustering obtained.

□

1.13 Quiz

Question 1.1

Numerical data with an index representing the time of collection and a variable containing the values collected over time to observe changes over time. □

Question 1.2

To understand the past in order to predict the future. □

Question 1.3

Data recorded twelve times a year. □

Question 1.4

To make it possible to view seasonality and cycles in the data □

Question 1.5

All of the choices given are typical patterns. They do not necessarily appear all together in a given time series, but they are the patterns we should look for in any time series. □

Question 1.6

All of the choices

□

Question 1.7

A subset of consecutive values of the series. □

Question 1.8

Software specialized in storing, managing and analyzing time series data sets.

□

Question 1.9

Without all of the options given.

□

Question 1.10

We could summarize features of the time series and create a data set that is not a time series data set, and then use machine learning methods. This would be for unsupervised machine learning methods. If we wanted to retain the temporal nature of the series, we could create features that do so and end up with a multivariate time series. Examples of how to do that will be offered throughout the book. □

1.14.5 Exercises

Exercise 1.16

Watch YouTube video `https://youtu.be/280MerRODM0` and read Section 1.14 in Chapter 1 before doing this exercise.

Run Program *Quandl.R* which contains both programs. At the end of the program, we use the Base R function aggregate() with each of the time series to convert them to the annual average unemployments. The plot of the aggregated time series do not differ much from the ones not aggregated.

The raw time series that we read from Quandl are already seasonally adjusted, according to the metadata in FRED and UKONS. If the time series is seasonally adjusted, seasonal fluctuations are not present, the data has already been smoothed. Thus smoothing the raw data by taking the annual average is not transforming the data much. The only noticeable change is that the COVID-19 sudden jump in unemployment is not displayed in the aggregated plot, as that rate has been averaged with the ones of other months of the same year. □

Exercise 1.19

We can use the Program *Quandl.R* to do this exercise. At the bottom of this program, the reader will find code for Exercise 1.19, to access the UNRATE from FRED using Quandl and code to do the dygraph.

Notice that in the *Quandl.R* program, we call the UNRATE t5 instead of t2, since in that program, we used t2 for the Missouri monthly nonseasonally adjusted unemployment rate. Notice also that it is very important to let your audience know whether the data displayed is seasonally adjusted or not. The title should reflect that, or some label in the graph should do that.

At the bottom of the dygraph, there is the dyRangeSelector. Move the handle left and right to obtain windows of the data and see more details.

When comparing the graph obtained with the *Quandl.R* code and that in FRED, we can see that FRED's has much more color. More work would have to be done to the R plot to achieve the same qualities as that in FRED. We could add text to the plot to indicate the metadata as in FRED. We could set the background and so on with different colors. But overall, the two plots look very similar. The dyRangeSelector is very much like the one in FRED.

□

Exercise 1.20

Use Program `ch1.pageviews.R` to obtain the answer for this exercise. That program contains the program on Page 57 on Lovelace, and also the same program but for a different page of Wikipedia, the one for Albert Einstein.

It is not hard to see that Albert Einstein has been more popular during the same period in which Lovelace's page was viewed daily.

Notice that we set the ts object for the views using the granularity of the data. That is, we set frequency equal 7 (appropriate for daily data for which we would expect some weekly seasonality- see Section 1.9), and then we tell the program that we are starting with week 40 (October 1, 2018 was on week 40), day 1 (Monday). We need to use that convention if we want to use ts(). □