# B
# Solutions to Selected Exercises

## *Chapter 1*

**Problem 1.2** Table 1.7 is not so tidy.

1. The values of the `president` variable are not all in the same form. The entry for Lincoln is in the form last-name, first-name, while the other values are not. It might be appropriate to divide the name into two variables: given name and surname. For instance, "Van" is not the middle name of Martin Van Buren; it's part of his surname "Van Buren."
2. The `in office` variable contains two numbers, with different punctuation between them. It would be appropriate to split `in office` into two variables, one for the start year and the other for the end year.
3. The `number of states` is not a number in Abraham Lincoln's case. (The situation changed rapidly during the US Civil War.) All the values in a variable should be the same kind of thing.

A bit more detail — one rule for tidy data is that all of the values for a variable should be the same kind of thing. You could argue that "1861-1865" and "1837 to 1841" are the same kind of thing: a set of characters that can be interpreted by a person. Don't buy it. It's much better, in general, if numerical quantities (like the year) are represented as numbers.

**Problem 1.4**

1. The variables are:
   - Table **A**: Year, Algeria, Brazil, Columbia
   - Table **B**: Country, Y2000, Y2001
   - Table **C**: Country, Year, Value
2. The cases are:
   - Table **A**: a year
   - Table **B**: a country
   - Table **C**: a country in a year

**Problem 1.5**
Each case is an airport. There are seven variables: `faa` (categorical), `name` (categorical), `lat` (numerical), `lon` (numerical), `alt` (numerical), `tz` (more or less numerical: data about times can be complicated), `dst` (categorical)

---

## *Chapter 2*

**Problem 2.1**
The R expression is:

```
Engines <-
  data.table::fread("http://tiny.cc/mosaic/engines.csv")
```

Here's an example of an answer that uses the words listed (shown in *italics*) to describe the expression.

> `data.table` is a *package* that defines the *function* `fread()`. The function takes as an *argument* a *quoted character string* identifying the file to be read. The *object* created by `fread()` is a *data table* that is being *assigned* the *object name* `Engines`.

**Problem 2.3**
Current Population Survey

**Problem 2.6**

1. `essay14`: no problems
2. `first-essay`: a dash (`-`) is not one of the allowed punctuation marks in an object name.
3. `"MyData"`: Being in quotes, `"MyData"` is a constant, not an object name.
4. `third_essay`: no problems. An underscore is legitimate in an object name
5. `small sample`: a space is not allowed in an object name
6. `functionList`: no problems

7. `FuNcTiOnLiSt`: no problems. Admittedly, it's a perverse and hard to type name, but it's legal
8. `.MyData.`: no problems. Periods are allowed in a function name. It doesn't matter where they occur. It would even be legal to use and name like this: `.....<- 7`. But this is bad style!
9. `sqrt()`: parentheses are not allowed in function names. In the text of this book, the author uses parentheses when referring to a function. That's just to help remind you that the object name, in this case `sqrt` is referring to a function as opposed to a data table or variable.

---

## *Chapter 3*

**Problem 3.1**

1. a data frame: Put it at the start of a chain, e.g. `fireplace %>% nrow()`
2. a function: Follow it by an open parenthesis, `fileplace(`
3. the name of a named argument. Follow it by = inside the parentheses of a function, e.g., `fun(fileplace = 7)`
4. a variable: place it inside the parentheses of a function, but not in the position of the name of a named argument, e.g., `fun(fileplace)` or `fun(x = fireplace)`

Although we encourage you to start names of data table with a capital letter, this is not a requirement of R syntax.

**Problem 3.3**
The named argument `package` should be used with the equal sign (=) as in `package = "NHANES"`

**Problem 3.5**

a) mistake 3; b) mistake 1; c) mistake 4; d) mistake 2; e) no mistake

**Problem 3.6**
Statement (c) is different. Rather than calculating the total number of births (using `sum()`), it finds the mean number of births across all the cases in each year/sex group.

---

## *Chapter 4*

**Problem 4.1**

1. Put the word "one" in italics.
2. Put the word "two" in bold face.
3. An item in a bullet point list
4. A first-level header
5. The word "five" in a computer font, that is, `five`
6. A second-level header
7. A web link showing the word "seven" and linking to the `tiny.cc` URL for this book.

**Problem 4.2**

a. This uses the apostrophe rather than the back-quote.
b. Two mistakes. 1) Uses the quotation mark rather than the back-quote. 2) Curly braces rather than parentheses are needed.
c. There are only two back-quotes in the closing fence.
d. The command and the fences must be on separate lines.
e. Four back-quotes in the closing fence.

**Problem 4.3**
The browser window will render the HTML to look like this:

---

An Introduction

Arithmetic is *easy*! For instance

```
3 + 2
```

```
[1] 5
```

---

**Problem 4.4**

1. `DataComputing.org`: both. It's a proper URL but it can also be the name of a file.
2. `ahab/whale.Rmd`: a file only. The relative path is `ahab` with the file `whale.Rmd` in the `ahab` folder
3. `ptth://world-bank.org`: neither a valid file name nor a URL. `ptth:` is not a recognized prefix.
4. `http://world-bank.org`: a valid URL. It cannot be a file name since `//` is not an allowed fragment of a path.
5. `//world-bank.org/index.html`: most browsers will not recognize the leading `//` as valid in a URL. It's not valid in a file name either.
6. `world-bank.org/index.html`: a valid file name (`index.html` in the `world-bank.org` folder) or a valid URL.

---

## *Chapter 5*

**Problem 5.1**

- `sex` is mapped to the x-axis.
- `wage` is mapped to the y-axis.
- `married` is mapped to the fill color.
- The different fill colors are positioned to be "dodged" rather than "stacked."
- The graph is facetted based on `race`.

**Problem 5.3**
This is a density plot made with

```
distributionGraphHelper(NCHS)
```

- `bmi` is mapped to the x axis
- The y-axis is set by the plot type, "density"
- `sex` determines the color (shown as level of gray in the printed version)
- facetting is based on `pregnant`

**Problem 5.4**
This is a scatterplot made with the help of `scatterGraphHelper()`.

- `exper` is mapped to the x axis
- `wage` is mapped to the y axis
- `married` sets color (shown as grayscale in the printed version)
- faceting is based on `sector`
- both the x- and y- axes have been set to logarithmic scales.

---

## *Chapter 6*

**Problem 6.1**

1. Facet labels: 1433B,1433E, 1433G, 1433Z
2. No. The scale of the vertical axis differs among the frames.
3. The glyph types are:
   - A vertical "error bar"
   - A path connecting the mid-points of the error bars.
   - Single and double stars.

   Comment: The usual scientific vernacular is that the error bars show the spread of several measurements and the stars indicate whether there is a statistically "significant" difference between the measurements marked and some reference condition. Critique: It might be better to indicate the individual measurements rather than summarizing them using an error bar. There's no particularly good reason for the path. The reference condition isn't identified in the graph.

**Problem 6.3**

1. The star glyph has these attributes: y-position, x-position, number of stars — 0, 1, or 2, to judge from the graph. The "error bar" glyph has the y-position of the center dot, the x-position of protein, the length of the bars reaching up and down, the label naming the protein, and color indicating the polarity. (This last is hard to see.)
2. The y-axis is labelled as cell density. In the data, it corresponds to the `center` variable. The x-axis is `protein`, a categorical variable.
3. Color is *not* an attribute of the ** glyph.
4. Guides: the tick marks on the y-axis. There is no guide right on the x-axis. The labels on the error bar glyphs are effectively a guide to the x-axis.

**Problem 6.5**
State (e.g. New Hampshire) on the vertical axis, Polling organization (e.g. NYT) on the horizontal axis.

---

## *Chapter 7*

**Problem 7.1**

a) `summarise()` b) `mutate()` c) `arrange()` d) `filter()` e) `select()` f) `group_by()` and `summarise()`

**Problem 7.3**

a. summary function
b. neither a summary nor a transformation. It is a pair of values rather than a single value as required for a summary function.
c. summary function
d. transformation function
e. transformation function
f. summary function
g. transformation function
h. summary function: an entire set of dates is being turned into a single number.

**Problem 7.5**

a. `arrange(sex, count)`
b. `filter(sex == "F")`
c. `filter(sex == "M", count > 10)`
d. `summarise(total = sum(count))`
e. `select(name, count)`

**Problem 7.7**

a) `BabyNames` is pushed twice as an input to `group_by()` and `summarise()`, once by the chaining syntax, once as the first object listed after the parentheses.

```
BabyNames %>%
  group_by(year, sex) %>%
  summarise(total = sum(count))
```

b) Rather than `ZipGeography` being the first input to `group_by()`, it's being used to store the result. Alas, there won't be any result, because there is no data table input to the `group_by()` data table.

```
ZipGeography %>%
  group_by(State) %>%
  summarise(pop = sum(Population))
```

c) The -> after `group_by()` should be the chaining `%>%`

d) It's more or less upside down.

```
Minneapolis2013 %>%
  group_by(First) %>%
  summarise(votesReceived = n())
```

| First | votesReceived |
|---|---|
| ABDUL M RAHAMAN "THE ROCK" | 338 |
| ALICIA K. BENNETT | 351 |
| BETSY HODGES | 28935 |
| BILL KAHN | 97 |
| BOB "AGAIN" CARNEY JR | 56 |
| *... and so on for 38 rows* | |

**Problem 7.9**

a) `summarise()` b) join, whether it be `inner_join()`, `left_join()`, etc.

**Problem 7.11**

The variables given as arguments to `group_by()` will always appear in the output, along with whatever variables are created by `summarise()`.

a. `sex, count, meanAge`
b. `diagnosis, count, meanAge`
c. `sex, diagnosis, count, meanAge`
d. `age, diagnosis,count, meanAge`
e. `age, count, meanAge`

**Problem 7.12**

1) `nrow()` 2) `names()` 3) `help()` 4) `library()`
2) `group_by()` 6) `summarise()`

---

## *Chapter 8*

**Problem 8.3**

All of the graphics have the same frame:

```
Frame <- CPS85 %>% ggplot(aes(x=age, y=wage))
```

Using that frame, saved in an object called `Frame` ...

- Graph (A)

  ```
  Frame + geom_point()
  ```

- Graph (B)

  ```
  Frame + geom_point(aes(shape=sex))
  ```

- Graph (C)

  ```
  Frame + geom_point(aes(shape=sex)) +
    facet_grid( married ~ .)
  ```

- Graph (D)

  ```
  Frame + geom_point(aes(shape=married)) +
    ylim(0,30)
  ```

## *Chapter 9*

**Problem 9.1**

- `str()` Quick presentation
- `group_by()` Data verb
- `rank()` Transformation
- `mean()` Summary Function
- `filter()` Data Verb
- `summary()` Quick presentation
- `summarise()` Data verb
- `merge()` Data verb
- `glimpse()` Quick presentation

**Problem 9.3**

1. Which color diamonds are largest on average?

```
diamonds %>%
  group_by(color) %>%
  summarise(size=mean(carat, na.rm = TRUE)) %>%
  arrange(desc(size)) %>%
  head(1)
```

| color | size |
|---|---|
| J | 1.16 |

2. Clarity with the largest average "table."

```
diamonds %>%
  group_by(clarity) %>%
  summarise(
    ave_table = mean(table, na.rm = TRUE)) %>%
  arrange(desc(ave_table)) %>%
  head(1)
```

| clarity | ave_table |
|---|---|
| I1 | 58.30 |

## *Chapter 10*

**Problem 10.1**

Most of the data verbs operate on a single data table as input. The join verbs, however, combine *two* data tables. Since only one can be passed into the verb by the chaining syntax, the other must be included as an argument inside the parentheses.

**Problem 10.2**

Not all the cases in the demographics table are contained in the geographic table, and there are cases in the geographic table that are not in the demographic table. For instance, Åland is in the geographic table but not in demographics. Akrotiri is in the demographics but not in the geographic table. You wouldn't want to combine the location of Åland with the demographics of Akrotiri!

One of the purposes of the join family of data verbs is to handle such missing or extra cases in the tables being combined.

**Problem 10.3**

1) Table **B** is in a format that makes it easy to find the change between years for each country.

```
tableB %>%
  mutate(diff = Y2001-Y2000)
```

2). Table **C** would make it easy to `join()` the country/continent data. Then finding the sum for each continent in each year could be accomplished with `group_by()` and `summarise()`

```
tableC %>%
  left_join( ContinentData ) %>%
  group_by( Continent, Year ) %>%
  summarise( total = sum( Value ) )
```

## *Chapter 11*

**Problem 11.3**

- Table **A** versus **C**: A is wide, C is narrow.
- Table **B** versus **C**: B is wide, C is narrow.
- Table **A** versus **B**: They are both the same relative to each other. In a narrow format of data, the cases are more detailed than in the wide format. For instance, table **C** has cases that are "a country in a year," which is more detailed than either "year" or "country."

**Problem 11.4**

| when | sbp |
|---|---|
| subject | BHO |
| subject | GWB |
| subject | WJC |
| before | 120 |
| before | 115 |
| before | 135 |
| after | 160 |
| after | 135 |
| after | 145 |

## *Chapter 12*

**Problem 12.1**

```
BabyNames %>%
  group_by(sex, name) %>%
  summarise(count = sum(count)) %>%
  mutate(popularity = rank(desc(count))) %>%
  filter(popularity <= 5)
```

| sex | name | count | popularity |
|---|---|---|---|
| F | Elizabeth | 1591439 | 2 |
| F | Jennifer | 1461186 | 4 |
| F | Linda | 1450328 | 5 |
| F | Mary | 4112464 | 1 |
| F | Patricia | 1570135 | 3 |
| | *... and so on for 10 rows* | | |

**Problem 12.2**

```
PopularCounts <-
  BabyNames %>%
  group_by(year, name) %>%
  summarise(total = sum(count)) %>%
  mutate(ranking =
           ifelse(rank(desc(total)) <= 100,
                  "Top_100", "Below" )) %>%
  group_by(year, ranking) %>%
  summarise(total = sum(total))
GlyphReady <-
  PopularCounts %>%
  spread(ranking, total) %>%
  mutate(frac_in_top_100 =
           Top_100 / (Top_100 + Below))
GlyphReady %>%
  ggplot(aes(x=year, y=frac_in_top_100)) +
  geom_line() +
  ylim(0, NA)
```

**Problem 12.3**

a. summary function
b. neither a summary nor a transformation. It is a pair of values rather than a single value as required for a summary function.
c. summary function
d. transformation function
e. transformation function
f. summary function
g. transformation function
h. summary function: an entire set of dates is being turned into a single number.

## *Chapter 13*

**Problem 13.1**

1. There are 7 distinct vertices: China, France, Germany, Italy, UK, USA, USSR
2. There is one row for each edge, so 9 edges.

**Problem 13.3**

NA

## *Chapter 14*

**Problem 14.1**

```
mosaicData::CPS85 %>%
  ggplot(aes(x = sex, y = wage)) +
  geom_boxplot(aes(fill=sex))
```

**Problem 14.2**
About 1.63 meters.

**Problem 14.3**
For women, a litle less than 1.60 meters. For men, about 1.75 meters.

**Problem 14.5**

- Graph (A)

```
CPS85 %>%
  ggplot(aes(x=wage)) +
  geom_density(aes(fill=sex,color=sex),
        alpha=0.5) +
  facet_grid( . ~ married) +
  ggtitle("(A)") + xlim(0,30)
```

- Graph (B)

```
CPS85 %>%
  ggplot(aes(x=age, y=wage)) +
  geom_smooth(aes(color=sex)) +
  ggtitle("(B)") +
  facet_grid( married ~ .) + ylim(0,NA)
```

- Graph (C)

```
CPS85 %>%
  ggplot(aes(x=age, y=wage)) +
  ggtitle("(C)") +
  geom_smooth(aes(color=sex)) +
  facet_wrap( ~ sector) + ylim(0,25)
```

**Problem 14.7**

```
mosaicData::CPS85 %>%
  ggplot(aes(x = educ, y = wage,
       color = sex)) +
  geom_point() +
  stat_smooth(method="lm") +
  ylim(0,15)
```

**Problem 14.8**

```
mosaicData::Galton %>%
  ggplot(aes(x = height, y = mother)) +
  geom_density2d() +
  facet_grid( ~ sex)
```

**Problem 14.11**

a. The fraction of people aged 65 and older is larger at the lowest income groups. But since the confidence intervals overlap among all the income groups, this relationship might be illusory: the medians might be roughly the same for all groups.
b. With more data, the notches are smaller. According to the $\sqrt{n}$ rule, with 16 times as much data, the notches should be about 1/4 the size of those in the smaller sample. Measuring the notches with a ruler confirms this in the graph.

---

## *Chapter 15*

**Problem 15.1**

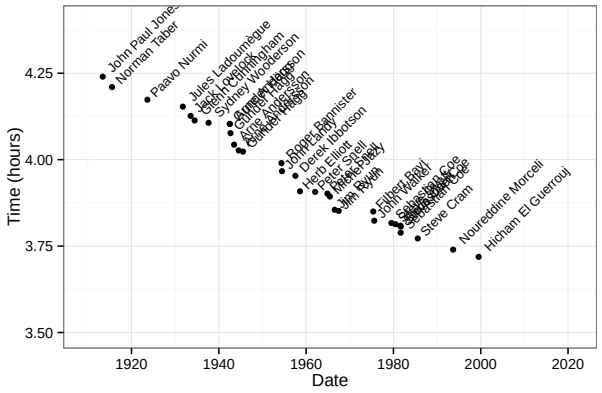
- `"April 30, 1777"` — `mdy()`
- `"06-23-1912"` — `mdy()`
- `"3 March 1847"` — `dmy()`
- `"11:00 am on Nov. 11th, 1918 at 11:00 am"` — `mdy_hm()`
- `"July 20, 1969"` — `mdy()`

**Problem 15.3**

```
library(rvest)
library(RCurl)
web_page <-
 "http://en.wikipedia.org/wiki/no break here
 Mile_run_world_record_progression"
SetOfTables <-
  web_page %>%
  rvest::html() %>%
  readHTMLTable(stringsAsFactors=FALSE)
T4 <- SetOfTables[[4]]

T4 %>%
  mutate(Date = gsub("\\[.\\]$", "", Date),
         Time = lubridate::as.duration(ms(Time))/60,
         Date = dmy(Date)) %>%
  ggplot(aes(x=Date, y=Time)) +
  geom_point() +
  geom_text(aes(label=Athlete),
     size=3, angle=45,hjust=-0.1) +
```

```
  ylim(3.5,4.4) + ylab("Time (hours)") +
  xlim(ymd("1910-01-01"), ymd("2021-01-01"))
```




## *Chapter 16*

**Problem 16.1**

1) Boys' names ending in a vowel.

```
BabyNames %>%
  #    Filter to keep only the boys.
  filter(sex == "M") %>%
  #    Add up across all years for each name
  group_by(name) %>%
  summarise(total = sum(count)) %>%
  #    Filter to find names ending in a vowel
  #    The $ is the end-of-string marker
  filter(grepl("[aeiou]$", name)) %>%
  #    Grab the top 10.
  arrange(desc(total)) %>%
  head(10)
```

| name | total |
|---|---|
| George | 1451408 |
| Joshua | 1163815 |
| Jose | 539924 |
| Kyle | 469045 |
| Lawrence | 454323 |
| *... and so on for 10 rows* | |

2) Names ending in "jo" or "joe". This is much the same as (1), without needing to filter by sex. The regex uses the "or" (vertical bracket).

```
BabyNames %>%
  #    Add up across all years for each name
  group_by(name) %>%
  summarise(total = sum(count)) %>%
  #    Filter to find names ending in [jo|joe]
  #    The $ is the end-of-string marker
  filter(grepl(".*(jo$|joe$)", name)) %>%
  #    Grab the top 10.
  arrange(desc(total)) #%>%
```

| name | total |
|---|---|
| Maryjo | 6973 |
| Billiejo | 1455 |
| Marijo | 1275 |
| Bobbijo | 1230 |
| Bobbiejo | 1009 |
| *... and so on for 72 rows* | |

```
  head(10)
```

```
[1] 10
```

**Problem 16.3**

1. Each matched string will be two letters, e.g. `AL`, `AK`, `AS`.
2. For a pattern like `A[LKSZRAP]`, there are seven strings that match. For an pattern like `D[EC]`, there are two strings that match: `DE` and `DC`. For a pattern like `RI`, there is only one string that matches. Altogether, 61 patterns.
3. These are appreviations for the US states and territories.

**Problem 16.5**

```
my_regex <-
  "\\(\\s*([+-]*[0-9]*[0-9\\.][0-9]*)no break here
  \\s*,\\s*([+-]*[0-9]*[0-9.][0-9]*)\\s*\\)"
Result <-
  CrimeSample %>%
  extract(Location,
          into=c("lat", "long"),
          regex = my_regex,
          convert = TRUE)
```

| lat | long |
|---|---|
| 42.34 | -71.11 |
| 42.26 | -71.16 |
| 42.33 | -71.08 |
| *... and so on for 50 rows* | |

## *Chapter 17*

**Problem 17.2**

1. Price is the variable to be explained, called the response variable. The explanatory variables included in the model are `living_area`, `bathrooms`, `bedrooms` and `fireplaces`.
2. No. Those are the houses to the left in node 3. For those houses, the `bathrooms` variable does not enter into the model.
3. Of those houses (in node 5), those with 1 1/2 or fewer bathrooms have a typical price of $151,000 while those with 2 or more bathrooms have a typical price of $180,000.
4. The `fireplaces` variable enters into the model only for houses larger than 2816 square feet. For those houses, having more than 1 fireplace is associated with a much larger price: $502,000 compared to $385,000.

Keep in mind that the model shows associations between variables. This is not at all the same thing as causal relationships. For example, in question (4), it might be that very fancy houses have much higher prices and they tend to have more than one fireplace. Perhaps it's the overall fanciness rather than just the existence of a second fireplace that influences the price.