Prep8

Sebastian Montesinos

Due by midnight, Monday April 18

Reminder: Prep assignments are to be completed individually. Upload a final copy of the .Rmd and renamed .pdf to your private repo, and submit the renamed pdf to Gradescope.

Reading

The associated reading for the week is Chapters 7 and 13 on Iteration and Simulation, respectively.

Note: There is no practice set this week, so that you can focus on completing the final project proposal due Friday by midnight (submitted in your group repos).

Practice 9 next week will contain question(s) pertinent to this material and next week's material on SQL.

1 - Iteration Basics

One of the key concepts in Chapter 7 is that R is designed to work with vectors. A number of functions benefit from this. This means there may be ways to write code more effectively than without using vectors. There are other ways to iterate as well, and iteration is a key concept in simulations. Let's look at a few ideas from the chapter (and your previous Computer science knowledge).

> part a - What is the difference between a *for* and a *while* loop, conceptually?

Solution: A while loop will continue to run until a certain condition is met, while a for loop will iterate through a specified number of times.

> part b - What does the following code from the textbook demonstrate?

```
# you may need to install the bench package
x <- 1:1e5
bench::mark(
  exp(x),
  map_dbl(x, exp)
)
```

```
## # A tibble: 2 x 6
##   expression           min   median `itr/sec` mem_alloc `gc/sec`
##   <bch:expr>      <bch:tm> <bch:tm>     <dbl> <bch:byt>    <dbl>
## 1 exp(x)            1.04ms   1.23ms     780.     1.53MB     24.3
## 2 map_dbl(x, exp)  51.94ms  51.94ms      19.3  788.59KB    173.
```

```
rm(x)
```

Solution: Since exp() is vectorized, it is much faster to use that to compute the exponents of the first 10,000 integers than iterating over the same vector over and over again.

> part c - At times, we've had to use the *across* function. Suppose you have a data set (mydataset) where the last 5 variables (named myvar1, myvar2, ..., myvar5) are being interpreted by R as characters but they are actually numeric. Write a command to have across help fix this.

Hint: You may need other wrangling commands too!

Solution:

```
# example code not evaluated as there is no data set, leave eval = FALSE

mydataset %>%
  mutate(across(myvar1:myvar5, ~ as.numeric(.)))
```

> part d - Describe the uses of *map()* and its variants from the *purrr* package.

Solution: Map() allows you to apply a function to each item in a list or vector, or to the columns in a data frame. For instance, if you wanted to compute the mean for all the columns in a dataframe you could use map_dbl and specify the data frame and computation (mean) and it would calculate all the means you want at once. the _dbl ending means that the output of the vector needs to be of the double type. This ending could be changed to adjust the output type.

part e - The following code to generate some uniform random numbers can be re-written to be more effective. What is the more effective code?

Hint: Remember you can get help about functions by typing ?functionname in the console.

Solution:

```r
# Keep this value
num <- 100

# Original code
x <- rep(0, num)
set.seed(231)
for(i in 1:num){
  x[i] <- runif(1)
}

# More effective code (provided by you)
set.seed(231)
x <- runif(num)
```

Generating random values is very important for many simulation ideas. So, let's head to our simulation question!

2 - Simulation

Chapter 13 looks at several different types of simulations. Some are based on actual data sets - to help understand the variability in data, while others involve theoretical situations which are then used to generate data.

If you have taken or are planning to take Probability (Stat 360), the simulation in 13.4.1 about Joan and Sally can be solved with probability concepts, OR you can use a simulation.

For this problem, we'll look at the simulation in 13.4.3, since we've seen that data set before (a long time back). The following code chunks all come from the text, and we'll explore what each one does as you run them.

The data set is the Restaurant Health Violations data set from NYC.

> part a - Explain what the provided code chunk below does. This has nothing to do with simulation, yet. You can answer this by writing sentences, or adding comments to the code.

```
minval <- 7
maxval <- 19
violation_scores <- Violations %>%
  filter(lubridate::year(inspection_date) == 2015) %>%
  filter(score >= minval & score <= maxval) %>%
  select(dba, score)
```

Solution: This creates a dataframe that sorts the values into a subset of possible grade ranges in only the year 2015. The first filter sets the inspection year until 2015, and the second filter subsets the range of grades to 7-19, finally the select chooses only these variables to keep in the new data frame.

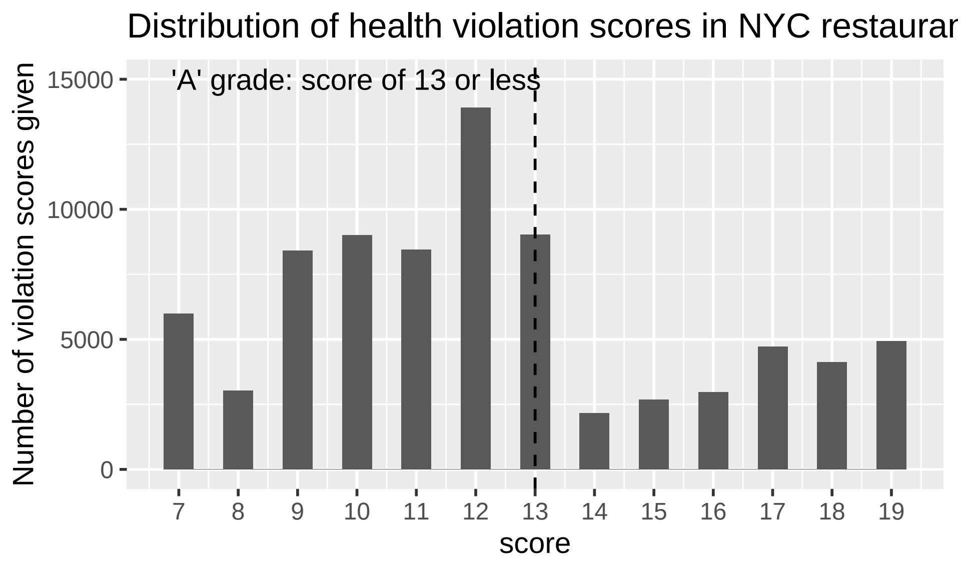
Hint: Being able to describe what the chunk above does in a sentence or two can help you as you work to describe the wrangling process for your final projects.

> part b - Improve the plot generated below to have a title and better name for the y-axis variable.

Be sure you understand what is being plotted!

Solution:

```
ggplot(data = violation_scores, aes(x = score)) +
  geom_histogram(binwidth = 0.5) +
  geom_vline(xintercept = 13, linetype = 2) +
  scale_x_continuous(breaks = minval:maxval) +
  annotate(
    "text", x = 10, y = 15000,
    label = "'A' grade: score of 13 or less"
  ) +
  labs(title = "Distribution of health violation scores in NYC restaurants in 2015",
      y = "Number of violation scores given")
```



part c - In this data set, a score of 13 or below was an "A" grade for the restaurant. Looking at the plot you generated, is there anything suspicious about the transition point?

Solution: Yes, the transition point is right at the cut-off between an A grade and a B grade, suggesting there could be a bias towards keeping people within the criteria for an A grade.

(This is the motivation for the simulation.)

part d - The next code chunk let's you look at the actual distribution of scores (to see the counts), and then performs a simulation to imagine what would happen if the score values of 13 and 14 were equally likely, and see how many values were returned as "14s" (heads). How many such simulations were performed?

```
# Look at distribution of actual scores
scores <- mosaic::tally(~score, data = violation_scores)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
scores
```

```
## score
##     7     8     9    10    11    12    13    14    15    16    17    18    19
##  5985  3026  8401  9007  8443 13907  9021  2155  2679  2973  4720  4119  4939
```

```
# Imagine if 13 and 14 were equally likely
mean(scores[c("13", "14")])
```

```
## [1] 5588
```

```r
set.seed(231)
random_flip <- 1:1000 %>%
  map_dbl(~mosaic::nflip(scores["13"] + scores["14"])) %>%
  enframe(name = "sim", value = "heads")
# Look at some results
head(random_flip, 3)
```

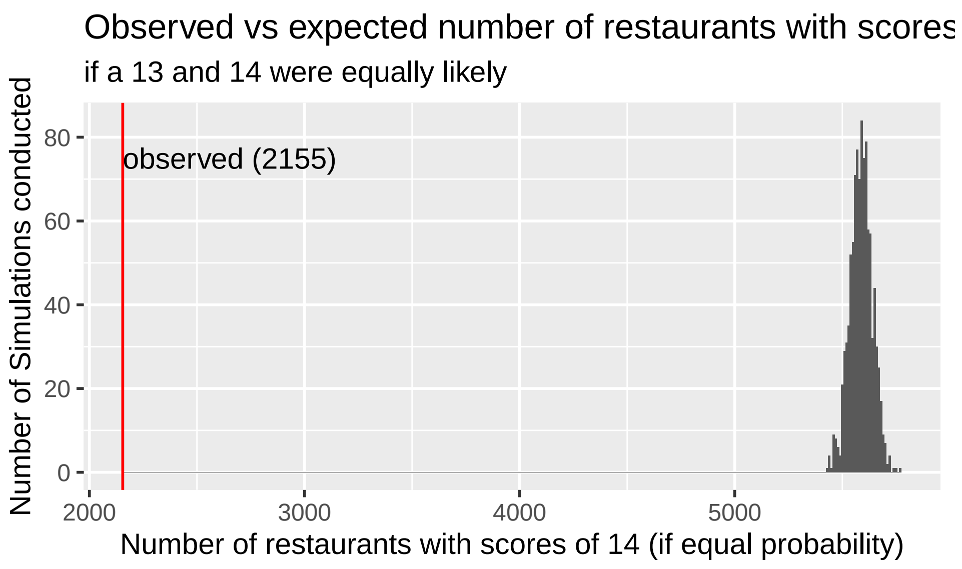
```
## # A tibble: 3 x 2
##     sim heads
##   <int> <dbl>
## 1     1  5632
## 2     2  5615
## 3     3  5565
```

Solution: 1000 simulations were performed.

> part e - To present the results of the simulation, the next plot was produced. Improve the plot by adding: a title, the observed value of restaurants with a score of 14 (add on to the existing label), and making a better y-axis label.

Solution:

```r
ggplot(data = random_flip, aes(x = heads)) +
  geom_histogram(binwidth = 10) +
  geom_vline(xintercept = scores["14"], col = "red") +
  annotate(
    "text", x = 2155, y = 75,
    label = "observed (2155)", hjust = "left"
  ) +
  labs(x = "Number of restaurants with scores of 14 (if equal probability)",
       y = "Number of Simulations conducted",
       title = "Observed vs expected number of restaurants with scores of 14",
       subtitle = "if a 13 and 14 were equally likely")
```



part f - What does the plot illustrate? In other words, what do we learn from the simulation? Be sure you can explain this based on the plot (don't just take the answer from the text).

Hint: This is an example of a permutation test. Be sure you are clear on what hypothesis was being tested.

Solution: The hypothesis being tested is whether the amount of restaurants given a 13 score vs a 14 score was equally likely to obtain. The plot illustrates the number of restaurants we would expect with a score of 14 on this assumption. Since we can see that the number of restaurants that actually received a 14 was far below this distribution, we can conclude that there was not an equal chance of receiving a 14 vs a 13: instead, inspectors were likely biased towards granting a 13.