Lab 3: Sampling and Descriptive Inference

Due: Monday, January 29, 11:59 PM

Name: Your name here

Mac ID: The first half of your Mac email address

1 French speakers outside of QC

In our discussion, I highlighted how the online portion of the Canadian Election Study uses quotas to make sure they have enough coverage of English and French speakers. In Quebec, they sample 80% French and 20% English. Elsewhere, it's 10% French and 90% English.

> **Task 1**
>
> Why do you think it is important to do this?
>
> **Hint:** *There are no wrong answers, but spend some time discussing it with someone to see if yours checks out.*

Whatever the reason, stratified sampling with quotas is very expensive, so we better come up with a good reason to do it! We want to do this to ensure surveys are representative of our sampling frame. Presumably the frame is adults from the whole country, and the 80/20 and 10/90 breakdowns are good approximations of the census data.

In `DeclareDesign` language, we would say that our **MODEL** of the world suggests the language breakdowns described above. The model does not need to be right, it only needs

to be useful.

Canada has an **adult** population of about 30 million. That's a bit overkill to simulate in R, so let's assume our sampling frame has 30 *thousand* people.[1] Let's say that 20% of our sampling frame is from QC, that's about 6 thousand.

How would our sample look like if we were to draw a sample of 1,000 individuals completely at random?

We can use some `DeclareDesign` code to do so. One trick is that when a survey uses stratified sampling, we can think about different strata as completely independent surveys.

Let's start with places outside Quebec, where our model of the world suggests about 90% English and 10% French speakers.

```
model = declare_model(N = 30000 - 6000,
                      french = sample(c(0,1), N,
                                      replace = TRUE, prob = c(0.9, 0.1)))
```

Let's unpack what we just did. We encoded our beliefs about the world within `model`. We told R we have a sampling frame `N` of 30000 6000 = 24000. We then said that in our frame of `N` people, any given person has 90% probability of being `french = 0`. Here, zero means english speaker and one means french speaker.

We can isolate that from the function:

```
sample(c(0, 1), size = 100, replace = TRUE, prob = c(0.9, 0.1))
```

```
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 [38] 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
 [75] 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1
```

[1] You can definitely do it but you can learn what I want you to learn with a much smaller sampling frame.

Task 2

Explain what the different arguments of the `sample` function above are doing.

Write your answer here

We are introducing some variability by stating that any given person has a certain *probability* of being a french speaker. We can think of this as baking in uncertainty around our model. We know that such deterministic numbers are not accurate, even if they were the exact census figure, those are already outdated.

Because of this uncertainty, every time we sample from our sampling frame we get a different distribution of people. This is great for our current purposes. We will learn how to prevent this behavior soon.

Let's skip the **inquiry** and **answer strategy** steps for now. Our **DATA STRATEGY** suggests that we sample individuals completely at random.

```
sampling = declare_sampling(S =complete_rs(N, n = 1000))
```

We can chain steps of a research design together with the `+` operator.[2]

```
canada_survey = model + sampling
```

We can then interrogate our research design. For now, let draw one realization of our survey data.

```
# Notice that I am hiding the results to avoid printing
# the whole 1k observations in the pdf
# But you should explore this in RStudio
draw_data(canada_survey)
```

We can pipe operations to compute the proportion of french speakers:

[2]Note that the chaining syntax varies across packages.

```
canada_survey %>%
  draw_data() %>%
  summarize(prop_french = mean(french))
```

```
  prop_french
1       0.092
```

This works because the mean of a binary variable is the same as the proportion of ones over the total.

Again, you will notice that the result changes every time you run the code. Give it a try!

Task 3

What is the number of french speakers in *your* sample?

Hint: *To get a consistent answer that you can write about, include the* `set.seed(x)` *function in the line right before your calculation. Replace* `x` *with your student ID number. Doing this will fix the random number generator algorithm in R.*

```
# Write your code here
```

Write your answer in words here

1.1 How often do we get the it right?

How often would our complete random sampling procedure give an answer that is close to the original stratified quota sampling? In the future we will discuss about standards to compare. But for now we will eyeball it.

We can simulate our research design multiple times and record the result every time. But to do that we need to declare our **INQUIRY** and **ANSWER STRATEGY**:

```
inquiry = declare_inquiry(true_prop = mean(french))


# the wording for declare_estimator looks weird, but it will become clear
# later in the semester
sample_mean = declare_estimator(french ~ 1, inquiry = "true_prop")
```

And then create a new design putting them all together:

```
canada_survey_full = model + inquiry + sampling + sample_mean
```

The following code draws independent samples of 1,000 respondents 1,000 times. For each simulation, it will compute the sample mean and store the output as a data frame.

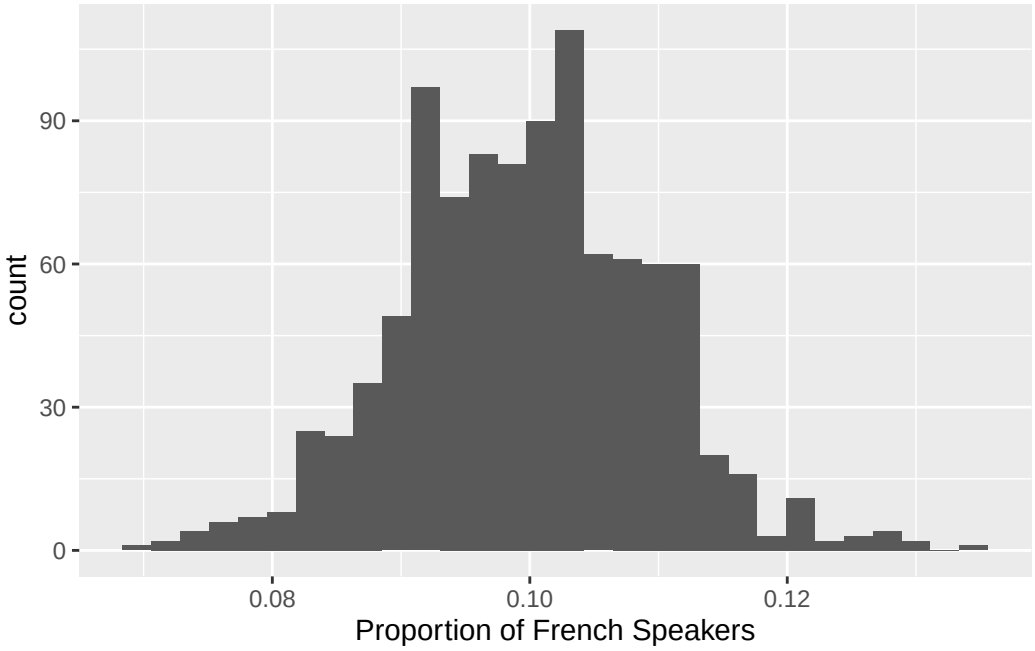
```
# This code will take a couple minutes to complete.
# Insert set.seed with your student number
# in the line below so your results are consistent.


sims = simulate_design(canada_survey_full, sims = 1000)


sims
```

There is a lot of information in `sims`, but right now we only care about the `estimate` column.

Let's visualize its distribution. The following figure shows the distribution of the proportion of french speakers in our simulated surveys. The taller the bar, the more frequently we see proportions around the corresponding value.

```
# ggplot is a tidyverse package for making figures.
# We will learn more about it soon.
ggplot(sims) +
  aes(x = estimate) +
```

```
  geom_histogram() +
  labs(x = "Proportion of French Speakers")
```



How does this compare with the original CES sampling procedure? It's hard to tell, because the CES approach guarantees a 10% proportion of French speakers by design. But, as we mentioned, this is expensive to implement.

What we can do is compute how off-the-mark our sampling procedure is in average:

```
sims %>% summarize(rmse = sqrt(mean((estimate-estimand)^2)))
```

```
         rmse
1 0.009179936
```

This number is the **Root Mean Squared Error (RMSE).** It tells us that, in average, our estimates of the proportion are off by a certain number of percentage points. The specific value will change based on your seed, but when I did it, I got a number that rounds up to 0.01. Which is about a ś1% margin of error.

2 English speakers in QC

Task 4

Repeat the simulation exercise for the smaller sample of English speakers in Quebec. Start with a sample size of 1,000 respondents. Is this a good number? Would you go higher or lower? Use either a figure or the RMSE as a criterion to choose. Report one new design with a different sample size and explain why it is better.**

Hint 1: *If you cannot come up with a reason to justify your design, consider that CES may be convinced either because the margin of error or the sample size is very small.*

Hint 2: *The chapters from the RD textbook we read this week show a way to try many sample sizes on a research design at once without having to rewrite the whole design.*

```
# Write your code here. Feel free to separate in many code chunks
```

Write your answer in words here. Feel free to intersperse writing and code.