Sampling and Data Analysis in R

1 Purpose

The purpose of this activity is to provide you with an understanding of statistical inference and to both develop and apply that knowledge to the use of the R statistical programming environment. The activity will be focused on random sampling methods, with some discussion of model-based notions of "representativeness".

2 Overview

This lab can be completed during class time and at-home. The final problem set for the course (Problem Set 8) will revisit some of this material, in tandem with the new material we will cover in LT.

3 Your Task

Using R as instructed, complete the following activities.

3.1 Populations

1. As we talked about in lecture, simple random sampling is the easiest *design-based* strategy for ensuring that your sample data are *representative* of the population from which those observations are drawn. We are not always interested in obtaining such a representative sample (e.g., because we are interested in particular cases or sets of cases) but when we are, we can attempt to construct a representative sample either by:

   (a) Using "random sampling" methods to select cases from a population such that our sample will tend *on average* or *in expectation* to match the population's characteristics.

   (b) Identifying a set of features (variables) that distinguish cases from one another and selecting cases that vary according to the population distribution of those characteristics, or

   The first of today's activities reiterates sampling- or design-based approaches using R. To do this, we are going to examine a population dataset containing the names of all babies born in the United States. This dataset contains the name and sex of every baby in the entire population of US babies born since approximately 1936. The dataset is available as an R package, so we can install and load it as usual:

   ```
   install.packages("babynames")
   library("babynames")
   ```

2. Looking at the first few rows of the data (what you see when you type `head(babynames)`), what is the *unit of analysis* of the dataset? How many rows are there in the dataset? How many variables? You should note that this is dataset is actually an *aggregation* of the population data: each row is a year–name–sex observation (so the unit of analysis is a name used in a given year for a baby of a given sex). How many unit name–sex combinations are there for 2014?

3. Are there any names that are used for both male and female babies in 2014? Here's the code for one example:

   ```
   babynames[(babynames[["year"]] == 2014) & (babynames[["name"]] == "Skylar"), ]
   ```

   Can you find others? You should be able to, there are 2465 names that appeared for both males and females that year. Hint: the `duplicated()` function can help you find the answer.

4. How common is the name Skylar as of 2014? How many males and female babies had this name? You will need to subset the data to find out.

5. Using ggplot2, plot the change in the number of Skylars over time:

   ```
   skylar <- babynames[babynames[["name"]] == "Skylar", ]
   library("ggplot2")
   ggplot(skylar, aes(x = year, y = n)) + geom_line(aes(colour = sex))
   ```

   Are there more male or female Skylars? For fun, examine how the change in other names has played out over time.

6. Now, because the dataset is an aggregation, it isn't exactly in the form we need. We need to *disaggregate* the data to create a dataset where the unit of analysis is an individual person (rather than a year–name–sex aggregation). To do that, we need to create a new data frame such that we create multiple rows based on the count those names in the `babynames` dataset. Do not try to do this for all years (your computer likely does not have enough memory to do it!), but you should be able to create a new dataset that contains one row per person for every baby born in a given year (e.g., 2014).

   To start on that, figure out how many babies were born in 2014. Hint: use `sum()`. The correct answer is 3,670,151.

7. Our new data frame should therefore have 3,670,151 rows. To create it, we will start by subsetting `babynames` to only 2014.

   ```
   babynames2014 <- babynames[babynames[["year"]] == 2014, ]
   ```

   Then, we need to know how many of each name–sex combination there are. This is contained in `"n"` variable of the dataset. Our new data frame will have one row for each person, or said another way n rows for each name–sex combination in the babynames2014 subset. We can do that by repeating each row index from `babynames2014` by the number of babies with that name–sex combination:

```
expand <- rep(seq_len(nrow(babynames2014)), babynames2014[["n"]])
length(expand)
dat <- babynames2014[expand, -c(4:5)]
dim(dat)
head(dat)
```

The resulting `dat` data frame should have the intended number of rows, with one row for each of the 3,670,151 babies.

8. How many males and how many females are there in this new dataset? What proportion are male and what proportion are female? You should find that 0.518 (or 51.8%) are male.

3.2 Sampling

9. In practice, we very rarely have the ability to study every case in an entire population. Because of time limitations or other resources, we only investigate a smaller number of individuals. Imagine, for example, if we wanted to study the U.S. population of babies born in a given year, it would be time consuming to contact the families of all 3.7 million of them. If we want to make claims about that entire population, we need to be careful about how we choose which subset to study. We cannot simply take an *arbitrary* subsample and expect it to be representative of the population. Instead, social scientists typically use *random sampling* to select cases from a population. Such random sampling has nice properties, among them (1) the sample will tend to be representative of the population as a whole and (2) it requires relatively small numbers of cases to make very precise claims about unknown or unobserved features of the population based on the information provided by the sample.

10. To showcase this, let's draw a very small sample from this population using the `sample()` function. Start by drawing a small sample of just ten babies from the 2014 dataset:

```
set.seed(234) # this just keeps our answers consistent
s1 <- sample(seq_len(nrow(dat)), 10, FALSE)
dat[s1,]
```

11. What proportion of this sample of 10 babies are male? The correct answer is $\hat{p} = 0.7$. (it's easy to see this graphically: `ggplot(dat[s1,], aes(x = sex)) + geom_bar()`). Is that value, $\hat{p}$, a good *estimate* of the proportion of males in the population as a whole, p? Why or why not?

12. Try taking another sample. Indeed, try taking a large number of different samples. How much does the proportion of males, $\hat{p}$ in each sample vary? Here's some code to do it:

```
x <- replicate(15, {
    prop.table(table(dat[sample(seq_len(nrow(dat)), 10, FALSE), "sex"]))
})
x
```

In the above code, `replicate()` just repeats an R expression the specified number of times.[1] Hopefully you can see that there is considerable *between*-sample variation in the estimated proportion of males, $\hat{p}$. You can see this visually by plotting the estimates from each sample:

[1]For example: `replicate(5, rnorm (1) + rnorm (1))` returns a vector of five numeric values.

```
ggplot(, aes(x = t(x)[,2])) + geom_histogram(binwidth = 0.05)
```

13. Clearly the "point estimate" from each sample, $\hat{p}$ is not a very good estimate of the true population proportion, p. But we know this and consequently, scientists who use random sampling are more interested in what are called "interval estimates" that express a range of plausible values of the population parameter, p given the estimate, $\hat{p}$ that we obtained from our sample. Recall our uncertainty about the true value of the population parameter is primarily influenced by three factors[2]:

    (a) The sample size that have drawn from the population (n)
    (b) The sampling *procedure* (in this case, simple random sampling)
    (c) The variance ($Var(Y)$) of the population variable (Y) we are interested in

    We typically summarize our uncertainty about the value of the population parameter, p by a quantity called the "standard error" (SE). The SE expresses how much our sample estimates, $\hat{p}$ vary across samples (were we to repeatedly sample from the population). The standard error is meant to capture the idea that if we repeated our sampling process and calculated our statistic of interest (in this case, the proportion of males, $\hat{p}$) on each sample, the *standard deviation* of those estimates around the true proportion, p would be the SE.

    The SE can then be used to construct a "margin of error" that conveys the interval in which we think the population parameter (the true proportion of males) is likely to be given the sample estimate and how much uncertainty we have due to "sampling error" (i.e., the fact that we are not observing the whole population).

14. To get a better grasp on this idea, we are going to draw numerous random samples from our population and then calculate the standard deviation of those estimated proportions, $\hat{p}$ from each sample. The standard deviation of our estimates from repeated sampling is the standard error, $SE(\hat{p})$. Let's repeat our random sampling of 10 babies and calculate the proportion for each sample. We will repeat this sampling procedure 1000 times to produce a vector of 1000 estimated values of p (one estimate per sample of $n = 10$):

    ```
    set.seed(123)
    est <- replicate(1000, {
        prop.table(table(dat[sample.int(nrow(dat), 10, FALSE), "sex"]))[2]
    })
    ```

    The `est` vector contains each of those 1000 estimated values of p. In statistics, were we to repeat this exercise an infinite number of times, the `est` vector of estimates is what would be called the "sampling distribution" of the estimate. This term refers to the distribution of a given statistic across repeated samples of the same size from a population.

15. When you have the `est` vector (note it may take some time to compute), examine the results:

    - What does the histogram look like:
      `ggplot(, aes(x = est)) + geom_histogram(binwidth = 0.05)`
    - Are the sample proportions "unbiased" (meaning the mean of the sample estimates, $\hat{p}$ is close to the population proportion, p): `mean(est)`?

[2] In *small populations* (those with less than perhaps 1 million cases), we also care about the size of the population (N), but that is not relevant here.

- How much do they vary? What is the standard deviation of the estimates (i.e., the SE)?

You should find that the answer to the last question is 0.154. This value tells us how much *sampling variation* there is in the estimated $Pr(Male)$ across the 1000 samples. Try plotting the $\hat{p}$ estimates as a histogram to see how the estimates vary.

16. If we change the sample size, the SE will change. Larger samples produce smaller SEs (to the point that if we sample all units in the population, the SE is 0). Repeat the above exercise but use a larger sample size ($n = 20$):

    ```
    est <- replicate(1000, {
        prop.table(table(dat[sample.int(nrow(dat), 20, FALSE), "sex"]))[2]
    })
    sd(est, na.rm = TRUE)
    ggplot(, aes(x = est)) + geom_histogram() + xlim(c(0,1))
    ```

    How large is the SE for this sample size of $n = 20$?

17. Repeat the above exercise but for several larger sample size ($n = 50$, $n = 100$, $n = 1000$). How large is the SE for each sample size? For $n = 1000$, your answer should be in the neighborhood of 0.015.

18. Indeed, when we have access to population data we can calculate this SE without ever drawing any samples because the SE is simply a function of the variance of the population variable and the size of the samples.[3] The formula is simply: $\sqrt{\frac{Var(Y)}{n}}$. Because Y is a binary variable in this case, $Var(Y)$ is very easy to calculate, it is simply: $Var(Y) = p(1 - p)$. In R we can calculate that as:

    ```
    prod(prop.table(table(dat[["sex"]])))
    ```

    This is about 0.25. The SE of our estimated proportion of males for a given sample size is thus simply the square root of 0.25 over sample size. In R, we can calculate that for a range of different sample sizes that we've considered ($n = 10, 20, 50, 100, 1000$):

    ```
    sqrt(0.25/c(10,20,50,100,1000))
    ```

    The resulting numbers should closely correspond to the "true" SEs that we calculated earlier using the standard deviation of the sample estimates.

19. In practice, we do not typically draw multiple samples so we do not actually know how much our estimates would vary, nor do we observe the population data so we do not know what $Var(Y)$ is. Statisticians get around this by using the sample data to estimate $\widehat{Var}(Y)$ using the "sample element variance"[4]:

    ```
    prod(prop.table(table(dat[sample.int(nrow(dat), 20, FALSE), "sex"])))
    ```

[3] Again, in small populations, it is more complicated, but most populations are large.

[4] For a numeric variable, the element variance of the data would be $Var(Y) = s^2_Y = \sum_{i=1}^{n} \frac{(Y_i - \bar{Y})^2}{n-1}$, which can be calculated by `var()` in R.

This is imperfect, of course, but allows us to calculate the SE from only one sample and therefore be able to make statements of uncertainty about the value of p knowing only $\hat{p}$ and the estimated SE.

20. In particular, we often report what is known as the "margin of error" or "confidence interval" for the population parameter. This conveys a range of possible values where we would expect the population parameter to be, centered around our estimated value $\hat{p}$. For a typical margin of error, we simply double the standard error and add and subtract that value to $\hat{p}$ to create an interval within which we estimate p to be (i.e., $\hat{p} + / - SE(p)$). What is the margin of error for the proportion of males for a sample of $n = 1000$? Is the population proportion (0.518) within the estimated margin of error (i.e., within the confidence interval)? Repeat this exercise for different sample sizes and check your understanding.

21. The interpretation of confidence intervals can be quite complicated. The key idea is that a confidence interval communicates, first, the amount of variation that one would expect to see in the estimates drawn from samples of this size from this population. But because the population parameter (in this case, the proportion of males) is estimated from the sample data, the interval does not *for sure* include the true population parameter of interest. Indeed, there is a non-zero chance that the confidence interval does not include the true population parameter value (i.e., the true proportion of males) and this is invariant to the size of the sample. It is not essential that you understand this yet — we will spend a whole week (LT 7) on it — but it's good if you start to think a bit about how sampling generates estimates that carry potentially considerable uncertainty about the population to which we claim interest in generalizing to.

22. The key intuition to derive from that interpretation is that a given sample may be highly "biased" (i.e., it may not resemble the population particularly well with respect to any variable). Yet, "design-based" claims to representativeness are premised upon the fact that a statistic estimated on multiple, repeated samples of the same size, drawn in the same way will *on average* be correct. Taking the mean of the proportions (or any statistic) estimated from many repeated samples will, as the number of samples increases, be the true population proportion. Any given sample is likely bias, so any estimated values must be communicated with measures of uncertainty that convey the variation we would expect across those many sample estimates were we to actually do that repeated sampling (which is something we rarely ever do).

    This can be a bit confusing, but it is an important matter to understand because design-based, random sampling generates samples that — on average — are representative of the population with respect to *every* variable, even those that we do not measure or that are unobservable. A random sampling process will always, if repeated, generate on average representative subsets of the population as a whole! And that conclusion is the core insight of statistics.

3.3 Check Your Understanding

23. The previous activities were all focused on estimating the value of a single population parameter, namely the proportion of males born in a given year. As a check on your understanding of both substance and the R implementation, it is worth attempting to estimate other population parameters, possibly on other subsets of the data. For example, you might try to do the same for the following population parameters:

- The proportion of males named Benjamin
- The proportion of all names that start with "A"
- The proportion of all babies named Olivia (across all years) that were born in 2010.
- The proportion of all babies named Liam (across all years) that were born after 1980.
- etc.

For each parameter, calculate the true population quantity from the original data and the true population element variance. Then calculate the standard error and margin of error for a sample of a given size. You may want to try to calculate how large of a sample size you would need to be able to estimate that quantity of interest from to within a particular degree of precision (e.g., within 1 percentage point).

Note: One thing that may be particularly important to consider is how a given sample may yield a reasonable estimate of one population parameter but not a very good estimate of other population parameters. In small samples, something like the proportion of people in the population with a given name may easily be estimated to be 0 (because no one in the sample takes that name) even though the true population proportion of people with that name is non-trivial. In practicing these techniques it is important to remember this because we rarely collect data to only estimate one population parameter and instead typically estimate many parameters from one sample.

3.4 Model-Based Representativeness

All of the material thus far has discussed "design-based" sampling, wherein our claims to representativeness can also be premised on what we might call "model-based" ideas. A model in this case is simply a formal claim about the features of particular cases that matter for drawing inferences from a particular set of cases to some other set of cases (such as the population as a whole).

In design-based sampling, our ability to draw inferences to the population are not premised on any particular features of cases being important; instead, generalizability to the population is based upon the idea that if we repeatedly sampled, our samples would on average resemble the population with respect to every possible variable.

Model-based claims to representativeness or external validity, by contrast, require asserting what features (i.e., scores on variables) are important for making such claims. For example, a very naive model would be that sex is the only variable that matters; in essence, all men are the same and all women are the same, so information about any particular man is perfectly representative of features of all men. This is, obviously, problematic. Yet a more complex model (perhaps one incorporating many variables, such as age, education, national origin, personality traits, ideology, etc.) might much more credibly segment the population into groups that are relatively homogeneous.

If true, then claims about a particular "cell" or "stratum" in the population (e.g., white English men from North London who are single, educated at the LSE, 24 years old, work in finance, are somewhat narcissistic, and generally left-wing) might be more convincingly made based upon information obtained from just a few arbitrarily (as opposed to *randomly*) chosen individuals from that cell. Such arguments are premised entirely on the credibility of the model linking the particular cases being studied to the population (i.e., that all relevant variables have been identified that make cases different from one another in the population).

3.5 Further Practice with Descriptive Statistics

As some additional practice working with data in R, we will look at some real data from the Quality of Government project, which contains country-level data on a very large number of economic, social, health, and political indicators. Beyond the basic exercises here, you may wish to apply the sampling techniques from above to these data in order to apply and test your knowledge using a more politically relevant dataset.

24. We will import the data using the `import()` function from the **rio** package. Once the data are loaded, we can examine the data themselves by just confirming that they are loaded correctly:

```
# install.packages("rio")
library("rio")
d <- import("http://www.qogdata.pol.gu.se/data/qog_std_cs_jan17.dta")
dim(d)
nrow(d)
ncol(d)
names(d)
str(d)
```

25. To obtain some simple descriptive statistics about a few variables, we can use the `summary()` function:

```
summary(d$fh_polity2)  # Polity scores (a democracy measure)
summary(d$gle_cgdpc)   # GDP
summary(d$dpi_finter)  # executive term limits
summary(d$bti_cr)      # civil rights index
summary(d$bl_asy15f)   # female educational attainment
```

26. Use ggplot2 to create a histogram of the distributions of these variables (see code above). You may want to play with the `bins` argument to control the look of the histograms.

27. Use the R functions `mean()`, `median()`, and `table()` to inspect the central tendency and distribution of these variables.

28. To assess the dispersion of each variable, use the functions we used above: `var()` and `sd()`.

29. If you're feeling ambitious, you can create some of your own ("user-defined") functions to calculate the skew and kurtosis statistics described in lecture:

```
skew <-  function(x) {
  m3 <- mean((x-mean(x))^3)
  skew <- m3/(sd(x)^3)
  skew
}
skew(d$gle_cgdpc)

kurtosis <- function(x) {
  m4 <- mean((x-mean(x))^4)
  kurt <- m4/(sd(x)^4)-3
  kurt
}
kurtosis(d$gle_cgdpc)
```

30. Skew and kurtosis are, in essence, meant to compare against a distribution known as the "normal distribution." It is "normal" for statistical reasons that have little to do with "normality" in the sense of common English. It is also called the "Gaussian" distribution, and that can sometimes be a less confusing (those less commonly used label). A normal distribution looks like the following graph:

    ```
    curve(dnorm, from = -4, to = 4, col = "red", lwd = 2)
    ```

    If a variable follows the normal distribution, its histogram will follow a very specific bell shape. We can "eyeball" this, but a more formal way to compare is by drawing what is called a "Q-Q plot". This is a special scatterplot drawn against a theoretical normal distribution based on the "quantiles" of the data (see `quantile()`). If the scatterplot has a straight line, then the data are normally distributed. If it deviates from that, then the data are skewed or "peaked" in a way that deviates from "normality". You can try it on two of the QoG variables:

    ```
    # on two of our observed variables from QoG:
    ggplot(d, aes(sample = gle_cgdpc)) + geom_qq()
    ggplot(d, aes(sample = bl_asy15f)) + geom_qq()

    # on a vector of random numbers drawn to follow the normal curve:
    ggplot(, aes(sample = rnorm(1e5))) + geom_qq()
    ```

31. Now repeat all of the above for the variables mentioned, and possibly explore other variables in the dataset. A codebook is available here: `http://qog.pol.gu.se/data/datadownloads/qogstandarddata`

32. Now, estimate the correlation between two variables. To do this, use `cor()`:

    ```
    cor(d$gle_cgdpc, d$bl_asy15f)
    ```

    We can also generate a "correlation matrix" showing the correlation between many variables, but this requires specifying the data in a slightly different way:

    ```
    cor(d[, c("gle_cgdpc", "bl_asy15f", "fh_polity2")])
    ```

33. Based on the correlations, imagine what the scatterplots might look like (keeping in mind what the correlation coefficient measures). If the data are categorical (rather than interval), you may want to use a cross-tabulation rather than correlation coefficient to summarize the results:

    ```
    table(d$dpi_finter, d$bti_cr)
    ```

    Note: You can also use `ftable()` to produce a slightly different looking table. You might also want to consider summarizing this relationship visually using a boxplot:

```
ggplot(d) + aes(x = factor(dpi_finter), y = bti_cr) + geom_boxplot()
```

34. Use ggplot to create a scatterplot of the relationship between two variables. Here's an example showing the relationship between GDP (x-axis) and average female educational attainment:

    ```
    ggplot(d) + aes(x = gle_cgdpc, y = bl_asy15f) + geom_point()
    ```

35. You will note that R prints a warning message when producing this plot. This relates to missing values in the dataset, where one or both of these variables are unobserved for a particular country. To see which countries we are missing data for, try the following:

    ```
    d$cname[is.na(d$gle_cgdpc)]  # for GDP
    d$cname[is.na(d$bl_asy15f)]  # for educational attainment
    ```

    You can also look at the data directly to see where these missing values are. You can use `table(is.na(`*var*`))` to count how many missing values there are in the variable. What does the presence of these missing values do for our ability to analyze the data? to estimate the values of population parameters? to represent the population? to draw a causal inference?

36. You may want to adjust the axis scales using, for example:

    ```
    ggplot(d, aes(x = gle_cgdpc, y = bl_asy15f)) +
      geom_point() + scale_x_log10()
    ```

37. You can modify the appearance of the plot in many, many ways. A common way to do this is by adding `aes()` features (see `? aes`) or by changing the plot theme (see `? theme`). Experiment with different plots until you feel comfortable with the various options.

38. One useful feature of ggplot2 is the ability to create multiple "panels" or "facets" (visual designer Edward Tufte calls these "small multiples"). To do this, you use the `facet_wrap()` function and specify a "formula" including the variable you would like to split the data by. This example create subpanels for different regions of the world:

    ```
    ggplot(d) + aes(x = gle_cgdpc, y = bl_asy15f) +
      geom_point() + facet_wrap(~ht_region)
    ```

39. Pause for a moment to consider how each facet represents a subset of the dataset. In this way, each facet is a summary of the dataset for only a subset of the dataset. If we want to summarize data in this way without plotting, we might consider using the `aggregate()` function. For example to calculate the mean level of GDP by region, you can do:

    ```
    aggregate(gle_cgdpc ~ ht_region, data = d, FUN = mean)
    ```

    Try this aggregate command using different variables and using a different value of the `FUN` argument (which takes the name of a function, such as `mean`, `sd`, etc. without the parentheses) until you feel comfortable with the process of generating data summaries.