R Assignment 3

The central limit theorem

2024-12-30

Intro/ Goals

In this Assignment you will do two things.

We will look at the penelope data. The data is complete in that it has every guess made about Penelope's weight and so we can figure out the true average guess (its 1287 pounds). Another way of saying this is that we have the guesses from the entire population. Normally we do not have population data.

With the penelope data I will demonstrate that the averages from random samples end up looking bell shaped (or normally distributed). (All the averages from many samples make up the sampling distribution.) This is the key take away from the central limit theorem. Its important because we use this fact to calculate probabilities that help us understand the population by taking a sample.

Revist the penelope data

If you have forgotten about the penelope from R Assignment 2 data you should revisit it. Also I've made a histogram of the guesses. This is called the data distribution.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

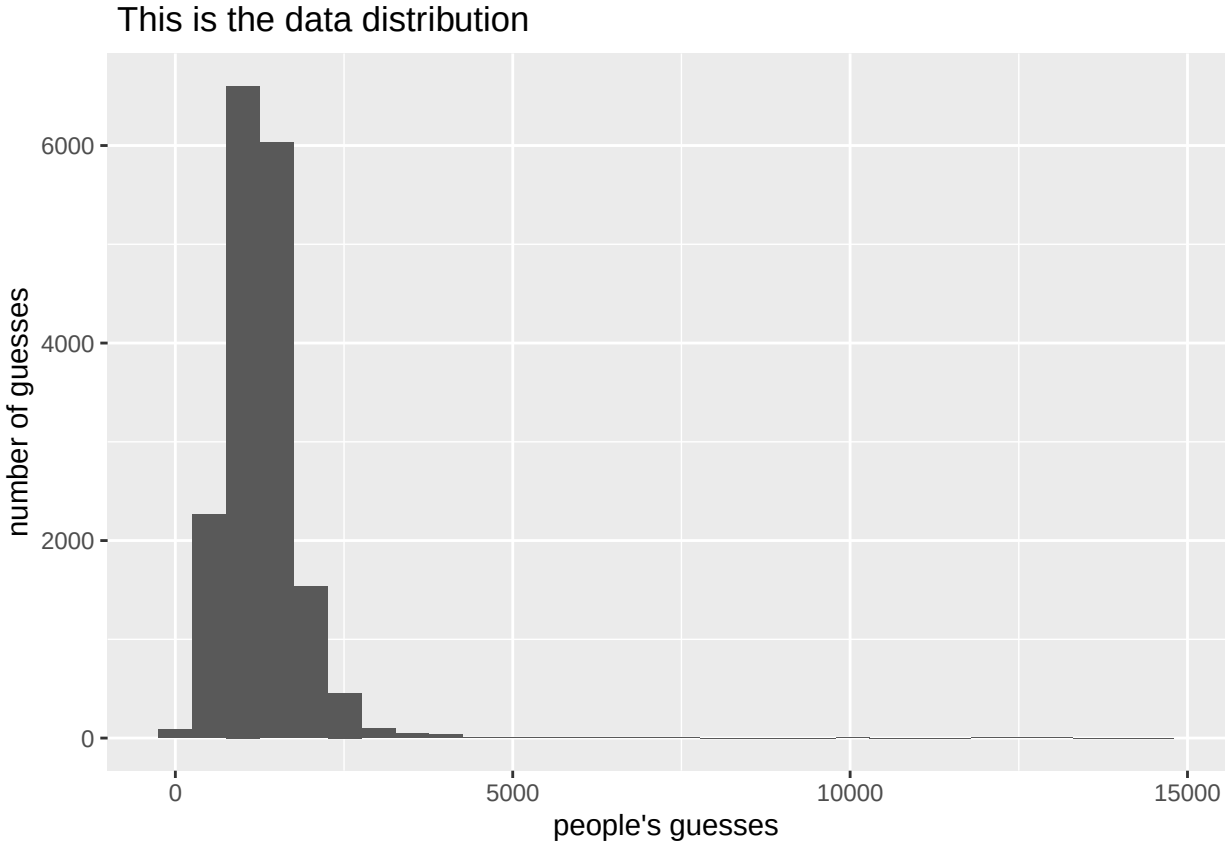
```
library(openintro)
```

```
## Loading required package: airports
## Loading required package: cherryblossom
## Loading required package: usdata
```

```
?penelope
```

```
ggplot(data = penelope, aes(x=weight))+
  geom_histogram() +
  labs(
    title = " This is the data distribution",
    x = "people's guesses",
    y = "number of guesses"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Take a sample and find its mean.

We are going to have R randomly sample 30 guesses and find the average of those guesses. Since there is a bit of randomness we have to set a seed for each chunk, this ensures we all get the same answer.

```
# You don't need to edit this code, just run the chunk and make sure you understand what's happening.
set.seed(1)

# This code randomly samples 30 guesses from the penelope weight variable. We could do more, but we don
sample_of_guesses = sample(penelope$weight, size = 30, replace = FALSE)

# This finds the mean of those guesses.
mean(sample_of_guesses)
```

```
## [1] 1267.5
```

The central limit theorem says if we find a lot of means of samples sized 30 (or larger), the distribution of those means will be normal. That is, if we make a histogram of a thousand means from a thousand samples it'll look bell shaped.

Fortunately there is a package called infer that has a function that can take many samples and calculate the mean of each sample.

```
#load infer and tidyverse
library(infer)
```

Let's start by taking 1000 samples sized 30 and finding each one of their means.

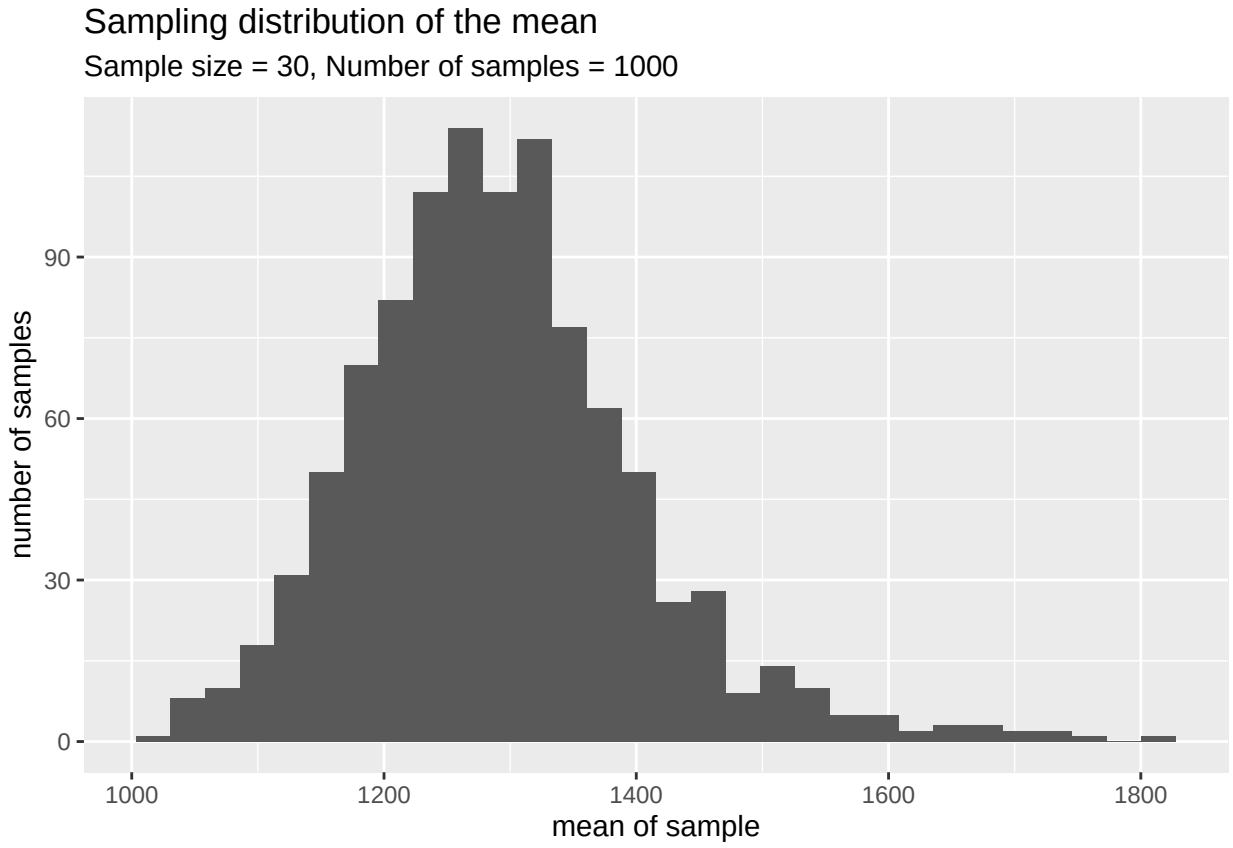
```r
# Here we use a special operator called the pipe |>, which takes the output of the previous line and pu

set.seed(1)

a_thousand_averages =  penelope |>
  specify(response = weight) |>     # <- This line tells which variable we want to find the averages.
  rep_sample_n(size =30, reps = 1000) |>   # <- This gets 1000 samples of size 30.
  group_by(replicate)|>                # <- A replicate is one sample size 30, we need to group them.
  summarise(mean = mean(weight))     # <- This gets the mean of each sample size 30.
```

If you click on "a_thousand_averages" in the environment you'll see its got the means of 1000 samples.

Now let's visualize the distribution of these means with a histogram. Notice the bell shape.

```r
ggplot(data = a_thousand_averages, aes(x = mean)) +
  geom_histogram() +
  labs(
    x = "mean of sample",
    y = "number of samples",
    title = "Sampling distribution of the mean",
    subtitle = "Sample size = 30, Number of samples = 1000"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



If we do the process again with a different 1000 samples we would get a slightly different shape. You can verify this by changing the seed above and rerunning the sampling code.

Note that the graph above is a distribution of means, generally it is called the sampling distribution. The

data distribution, is what the original guesses look like, scroll back to the top to remind yourself the shape of that graph.

Note that even though the data is not normally distributed the sampling distribution is.

Excersises

I want to know what happens to the distribution when the sample size is increased or decreased. You will answer this by editing the code that we used to make the distribution

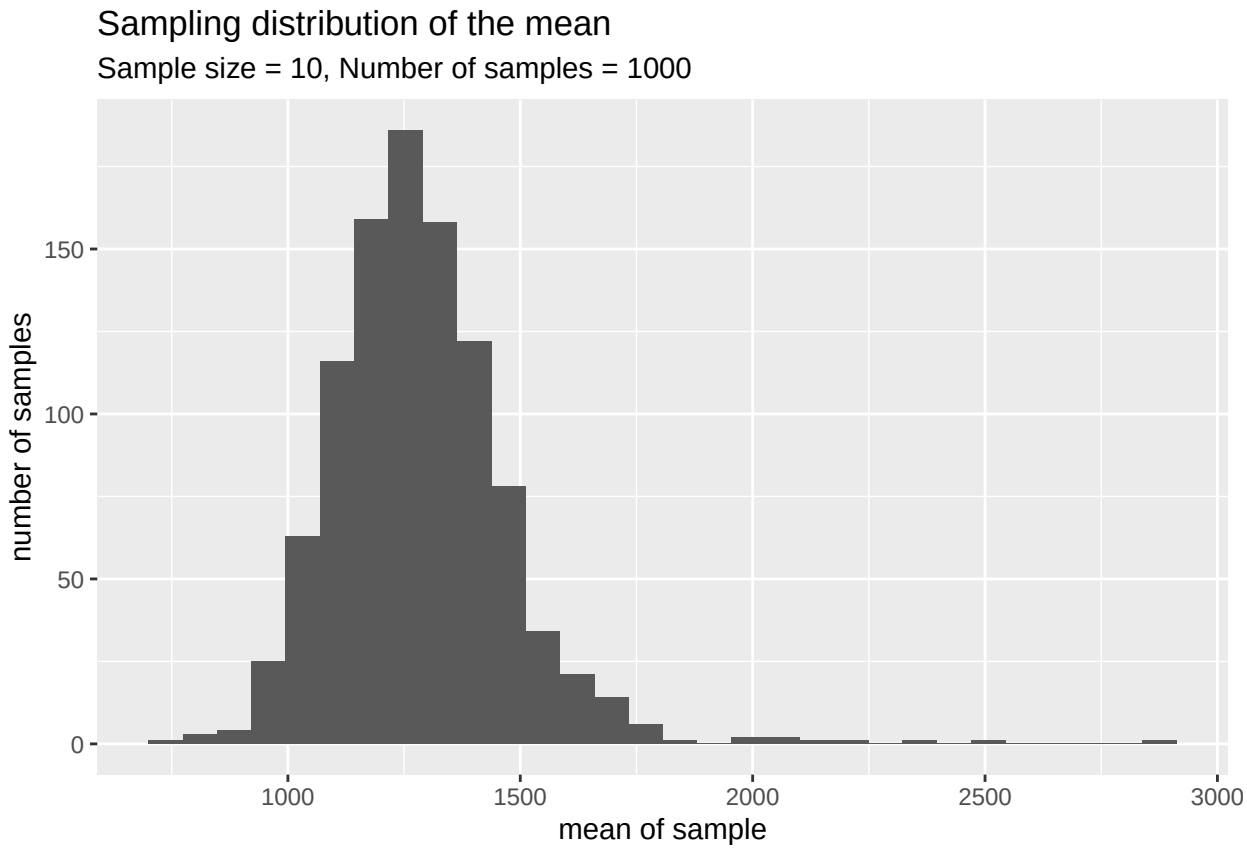
Decrease the sample size to 10. What happens to the spread of the sampling distribution, when compared to 30?

Answer: The spread of the data increases when the sample size decreases.

```
# All you have to do here is edit the size of the sample run the sampling simulation and answer the que

set.seed(1)

a_thousand_averages =  penelope |>
  specify(response = weight) |>
  rep_sample_n(size = 10, reps = 1000) |>    # <- Edit the size
  group_by(replicate)|>
  summarise(mean = mean(weight))
```

```
# You should edit the subtitle below.

ggplot(data = a_thousand_averages, aes(x = mean)) +
  geom_histogram() +
  labs(
    x = "mean of sample",
    y = "number of samples",
    title = "Sampling distribution of the mean",
    subtitle = "Sample size = 10, Number of samples = 1000"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



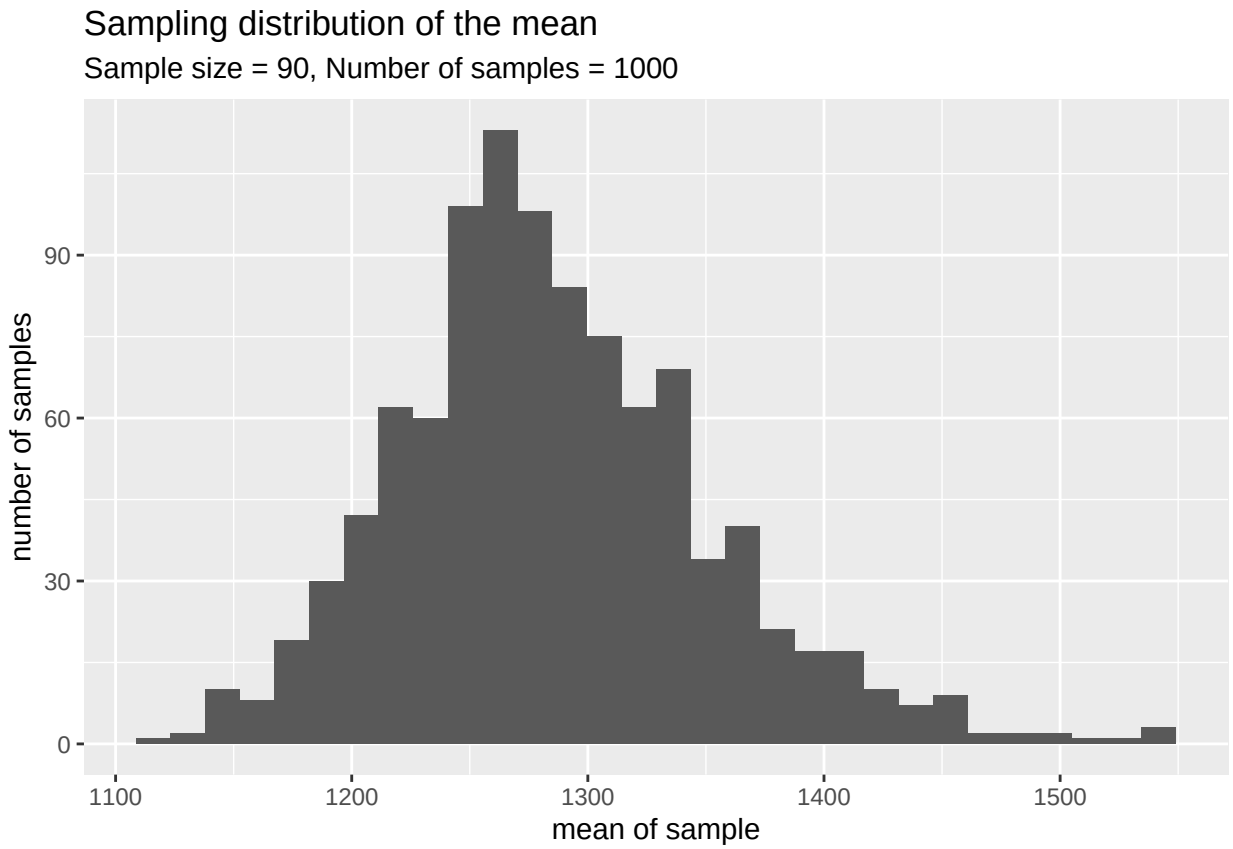
Increase the sample size to 90. What happens to the spread of the sampling distribution, when compared to 30? (Hint: Pay special attention to the values on the x-axis.)

Answer: The spread of the data decrease when the sample size increases

```
# Here we use a special operator called the pipe |>, which takes the output of the previous line and pu

set.seed(1)

a_thousand_averages =  penelope |>
  specify(response = weight) |>
  rep_sample_n(size = 90, reps = 1000) |>    # <- Edit the size
  group_by(replicate)|>
  summarise(mean = mean(weight))
```

```
ggplot(data = a_thousand_averages, aes(x = mean)) +
  geom_histogram() +
  labs(
    x = "mean of sample",
    y = "number of samples",
    title = "Sampling distribution of the mean",
    subtitle = "Sample size = 90, Number of samples = 1000"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Conclusion: The central limit theorem says that when we sample from a population, the sampling distribution (aka the distribution of averages) will be normally distributed. There are a few conditions we should adhere to (did you notice how a small sample of size 10 had a long right tail?). The conditions for each situation are different, in this R Assignment we were working with a single mean, you can read about these conditions in chapter 19.2.

Its worth remembering that normally we do not have the data from the full population, so generally we trust that if conditions are met the sampling distribution is normal.

```
# You can ignore this chunk. I just need to unload infer so it doesn't mess with future assignments. Po
detach("package:infer", unload = TRUE) # This unload infer, which we used last assignment
```