More on t, part 2

Chriss Jordan Oboa

4/20/2022

In this file we continue investigating confidence intervals for means based on small samples from a population with a normally distributed numerical variable. We are following the path of William Sealey Gosset's original analysis.

Packages and data

We load our usual packages: `tidyverse` as a general tool, and `infer` to help with random sampling.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(infer)
```

We also load the same population from our earlier activities. This population is close to what Gosset used in his work.

```
population <- read_csv("data/population.csv")
```

```
## Rows: 3000 Columns: 1
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): height
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

We again store the population mean and population standard deviation with the names `mu` and `sigma`.

```
mu <- mean(population$height)
sigma <- sd(population$height)
```

Working on a larger scale

The 50 or so dots from our class samples are not enough to see everything about how the t-statistics work. Gosset made a graph using 750 samples, calculating everything by hand. Since we have R to help us, we will use 10,000 samples.

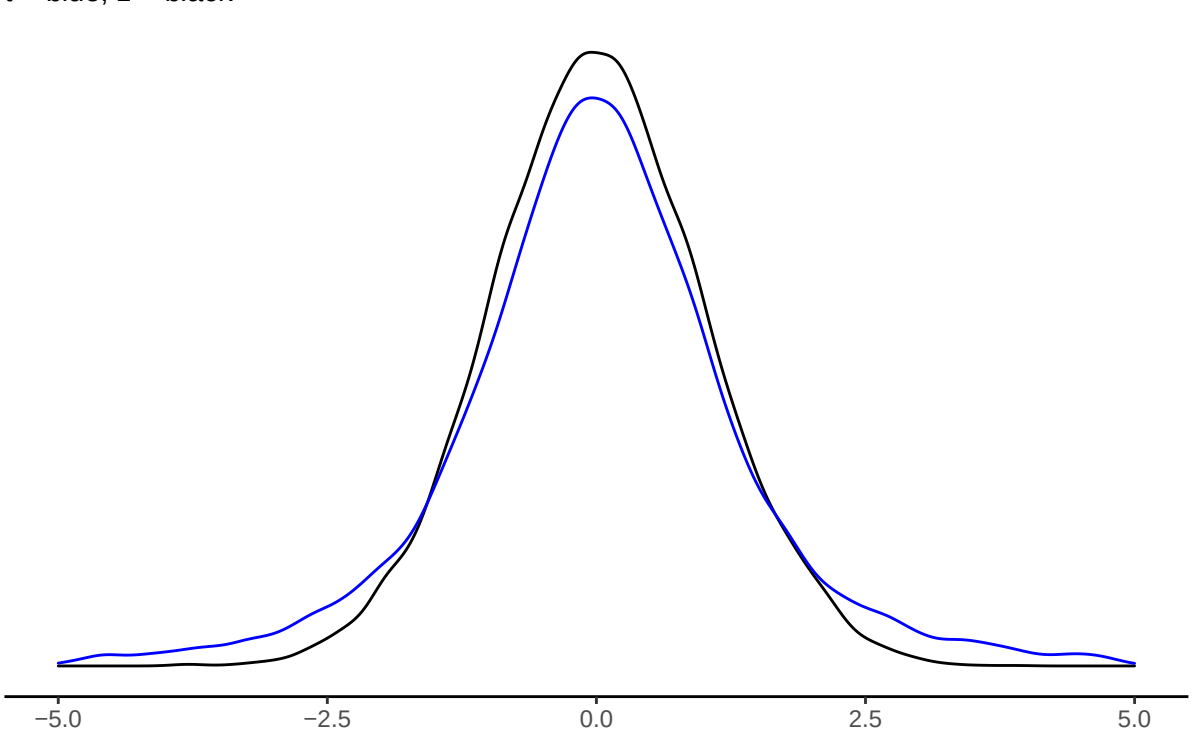
1. The following chunk takes 10,000 simple random samples of size 4 from the population and computes their z-statistics and t-statistics.

```
samples_4 <- population %>%
  rep_slice_sample(n = 4, reps = 10000) %>%
  group_by(replicate) %>%
  summarize( z = (mean(height) - mu) / (sigma / sqrt(4)),
             t = (mean(height) - mu) / (sd(height) / sqrt(4)) )
```

2. To compare the distributions of t-statistics and z-statistics, we use density plots. **Put a copy of the following graph on our shared slides.**

```
ggplot(data = samples_4) +
  geom_density(mapping = aes(x = z), color = "black") +
  geom_density(mapping = aes(x = t), color = "blue") +
  scale_x_continuous(limits = c(-5,5), name = "") +
  theme_classic() +
  theme(axis.line.y = element_blank()) +
  scale_y_continuous(breaks = NULL, name = "") +
  labs(title = "t-statistics and z-statistics for sample size 4",
       subtitle = "t = blue, z = black")
```

```
## Warning: Removed 175 rows containing non-finite values (stat_density).
```



3. Use `summarize` to calculate Q1 and Q3 for the `z` and `t` statistics. For each distribution, the middle

50% of samples ("most") fall between Q1 and Q3. Calculate also the 0.025 and 0.975 quantiles. For each distribution, the middle 95% of samples ("almost all") fall between these quantiles. **Report your results on our shared slides.**

```
samples_4 %>% summarize(q025 = quantile(z, 0.025),
                        q25 = quantile (z, 0.25),
                        q75 = quantile (z, 0.75),
                        q975 = quantile (z, 0.975) )
```

```
## # A tibble: 1 x 4
##    q025    q25   q75  q975
##   <dbl>  <dbl> <dbl> <dbl>
## 1 -1.94 -0.649 0.674  1.99
```

```
samples_4 %>% summarize(q025 = quantile(t, 0.025),
                        q25 = quantile (t, 0.25),
                        q75 = quantile (t, 0.75),
                        q975 = quantile (t, 0.975) )
```

```
## # A tibble: 1 x 4
##    q025    q25   q75  q975
##   <dbl>  <dbl> <dbl> <dbl>
## 1 -3.27 -0.746 0.758  3.29
```

4. In our first work with confidence intervals, we found that using 2 standard errors would give us confidence intervals that captured the population parameter for about 95% of samples. Looking at your work on t-statistics, you can see why 2 approximate standard errors does not work: the interval from -2 to +2 does not include the entire middle 95% of t-statistics. Judging from your 0.025 and 0.975 quantiles, how many approximate standard errors would you recommend that we use for 95% confidence intervals based on the standard error estimated from the sample standard deviation? **Record your response on the shared slides.**

Gosset found that for *any* normally distributed population, t-statistics of samples of size 4 follow the same distribution that you see here. For good reasons, this distribution is called the "t-distribution with 3 degrees of freedom."

Larger sample sizes

For comparison, we can repeat the same analysis but with larger sample sizes to see what happens. We will try sample size 30.

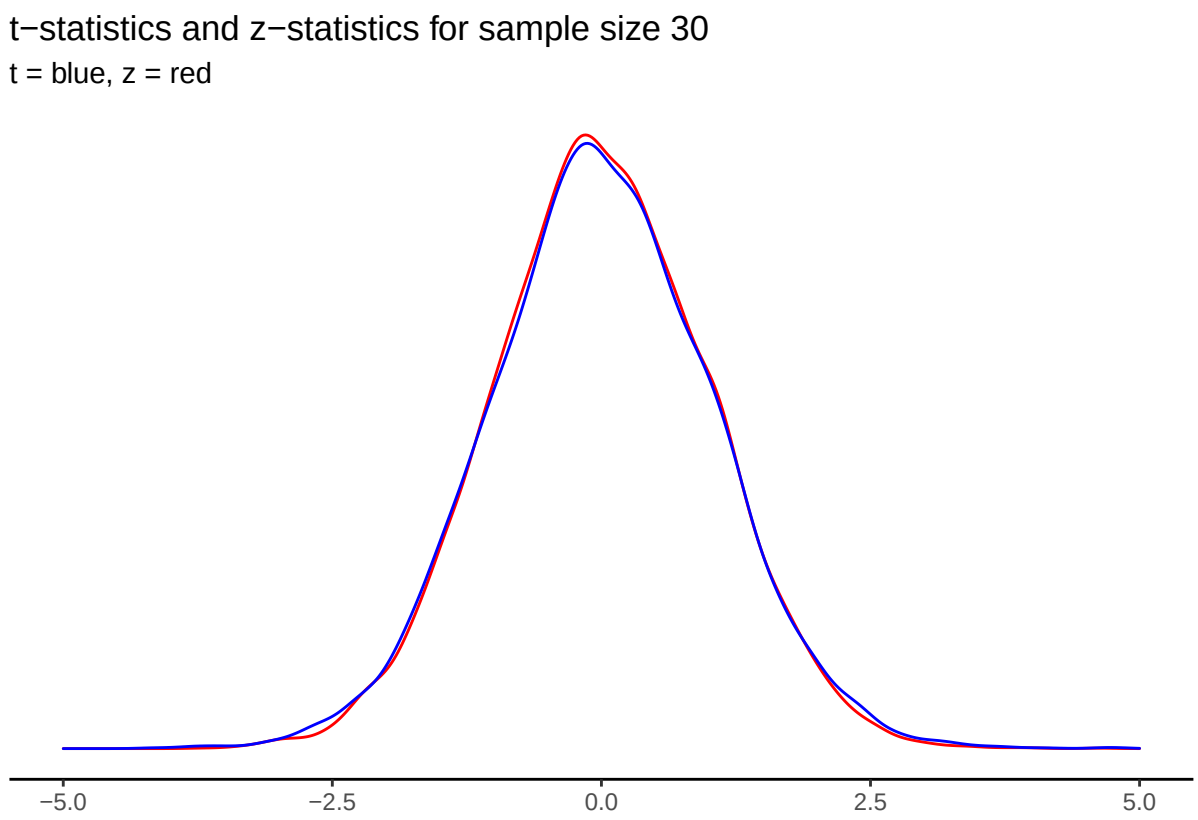
1. The following chunk takes 10,000 simple random samples of size 30 from the population and computes their z-statistics and t-statistics.

```
samples_30 <- population %>%
  rep_slice_sample(n = 30, reps = 10000) %>%
  group_by(replicate) %>%
  summarize( z = (mean(height) - mu) / (sigma / sqrt(30)),
             t = (mean(height) - mu) / (sd(height) / sqrt(30)) )
```

2. To compare the distributions of t-statistics and z-statistics, we use density plots. **Put a copy of this graph on the shared slides and answer the questions there about comparing the z- and t-distributions for samples of size 30.**

```
ggplot(data = samples_30) +
  geom_density(mapping = aes(x = z), color = "red") +
  geom_density(mapping = aes(x = t), color = "blue") +
```

```
  scale_x_continuous(limits = c(-5,5), name = "") +
  theme_classic() +
  theme(axis.line.y = element_blank()) +
  scale_y_continuous(breaks = NULL, name = "") +
  labs(title = "t-statistics and z-statistics for sample size 30",
       subtitle = "t = blue, z = red")
```



Trying out t-distribution-based confidence intervals

The following chunk takes a fresh 100 random samples of size 4 and calculates the corresponding +/- 2*SE confidence intervals using the sample standard deviation to estimate the standard error SE.

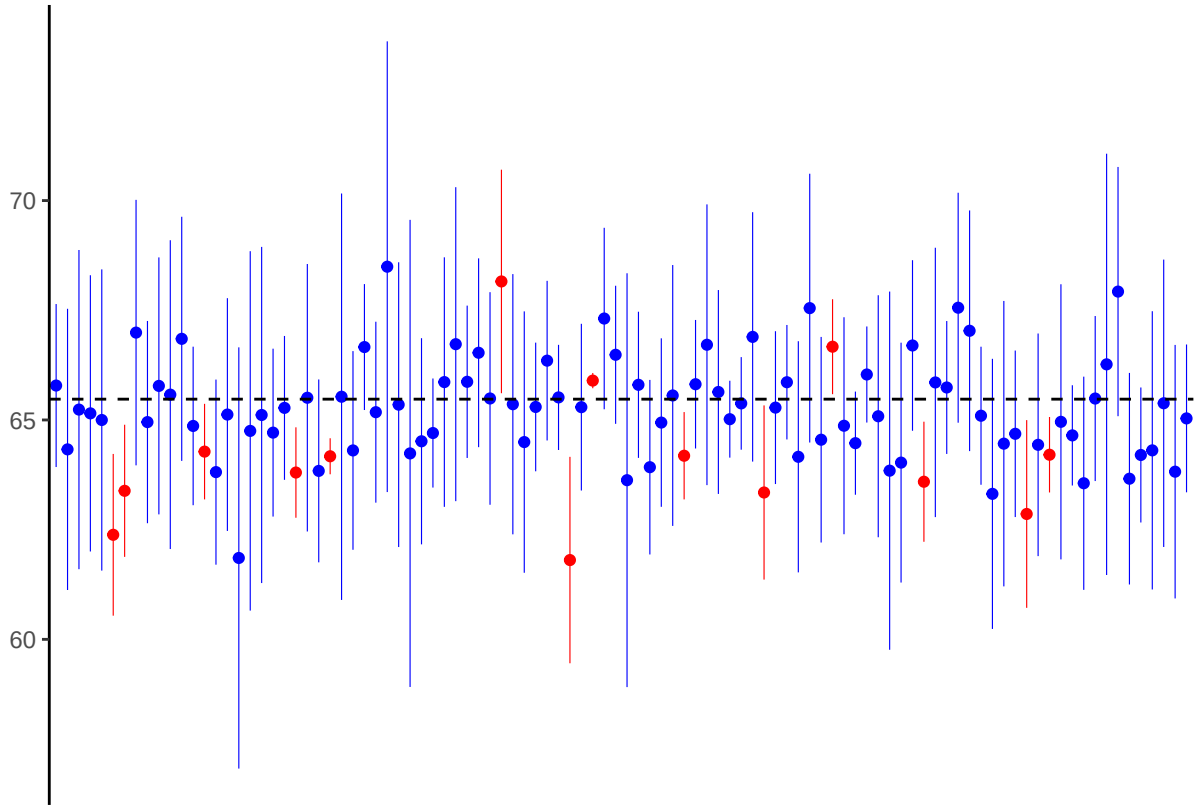
```
CIs_sample_sd <- population %>% rep_slice_sample(n = 4, reps = 100) %>%
  group_by(replicate) %>%
  summarize(x_bar = mean(height), s = sd(height)) %>%
  mutate(se = s / sqrt(4))

CIs_sample_sd <- CIs_sample_sd %>%
  mutate(conf.low = x_bar - 2*se, conf.high = x_bar + 2*se) %>%
  mutate(worked = case_when(
    (conf.low < mu) & (mu < conf.high) ~ "good",
    TRUE ~ "bad"
  ))
```

The next chunk graphs these confidence intervals. As we saw in an earlier activity, this procedure doesn't work at the 95% confidence level, since there are too many red (bad) intervals.

```
ggplot(data = CIs_sample_sd,
       mapping = aes(x = replicate, y = x_bar, ymin = conf.low, ymax = conf.high, color = worked)) +
  geom_pointrange(size = 0.2) +
  geom_hline(mapping = aes(yintercept = mu), linetype = "dashed") +
  scale_x_discrete(breaks = NULL, name = "") +
  scale_y_continuous(name = "") +
  scale_color_manual(values = c("good" = "blue", "bad" = "red"), guide = F) +
  theme_classic() +
  theme(axis.line.x = element_blank())
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



1. To make confidence intervals for small samples, we should follow Gosset's advice. Looking at his "t distribution", we see that we need something more than + / - 2 approximate standard errors to capture almost all (95% of) samples. You made a suggestion above for how many approximate standard errors we should use. To find the official number according to Gosset's mathematics, use the R command `qt(p = 0.975, df = 3)`. The letter "q" stands for "quantile", and "df" stands for "degrees of freedom".

```
qt(p = 0.975, df = 3)
```

```
## [1] 3.182446
```

2. Use this "official number" to make a new set of confidence intervals following the same procedure as above. Copy and paste the commands for making the intervals from above. The only thing that you need to change is number of estimated standard errors that you use for the confidence intervals. **Put a copy of your graph of the Gosset-style confidence intervals in the shared slides.**

```
CIs_sample_sd <- population %>% rep_slice_sample(n = 4, reps = 100) %>%
  group_by(replicate) %>%
```

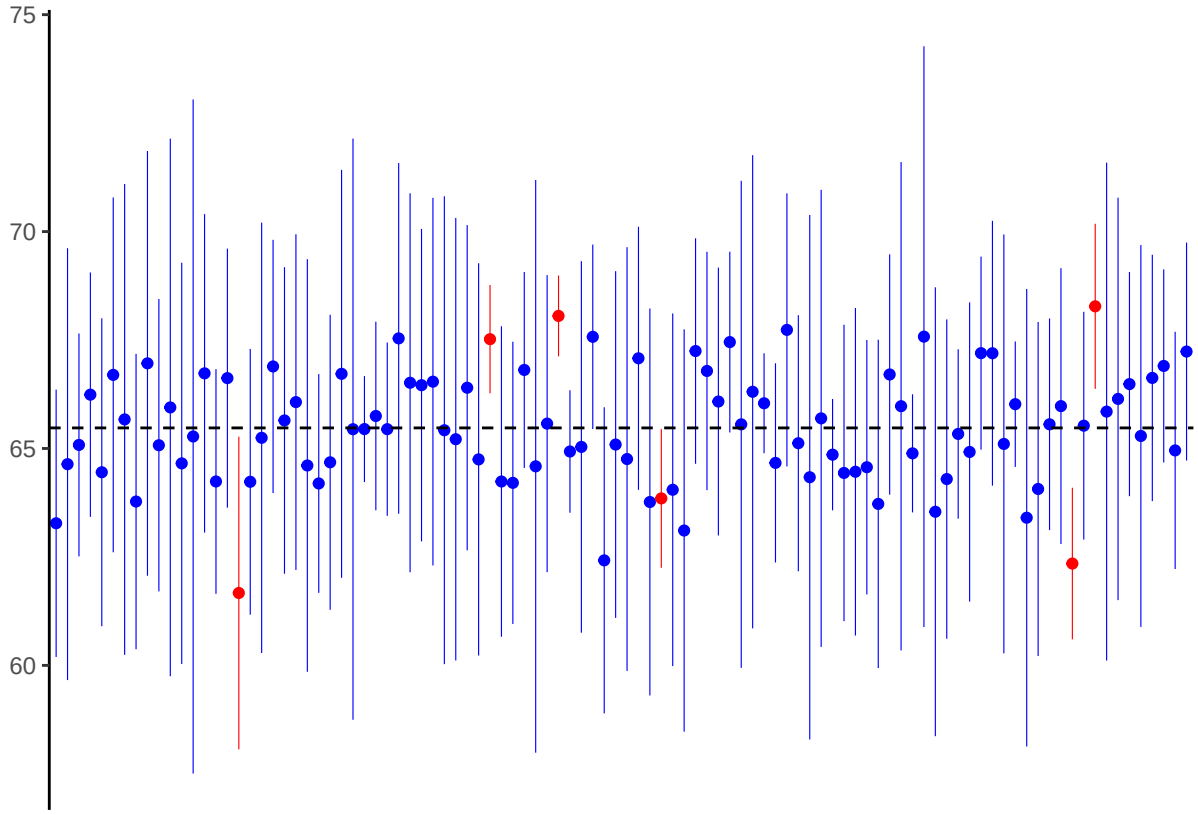
```
  summarize(x_bar = mean(height), s = sd(height)) %>%
  mutate(se = s / sqrt(4))

CIs_sample_sd <- CIs_sample_sd %>%
  mutate(conf.low = x_bar - 3.182446*se, conf.high = x_bar + 3.182446*se) %>%
  mutate(worked = case_when(
    (conf.low < mu) & (mu < conf.high) ~ "good",
    TRUE ~ "bad"
  ))
```

3. How many of the 100 95% intervals calculated using Gosset's recommendation include the population mean? **Answer on the shared slides.**

```
ggplot(data = CIs_sample_sd,
       mapping = aes(x = replicate, y = x_bar, ymin = conf.low, ymax = conf.high, color = worked)) +
  geom_pointrange(size = 0.2) +
  geom_hline(mapping = aes(yintercept = mu), linetype = "dashed") +
  scale_x_discrete(breaks = NULL, name = "") +
  scale_y_continuous(name = "") +
  scale_color_manual(values = c("good" = "blue", "bad" = "red"), guide = F) +
  theme_classic() +
  theme(axis.line.x = element_blank())
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



end