Examples of interval estimation: one- and two-population location problems

Topic 2.2: Basics of statistical inference - confidence intervals

Sonja Petrović

Created for ITMD/ITMS/STAT 514

Spring 2021.

Reminder

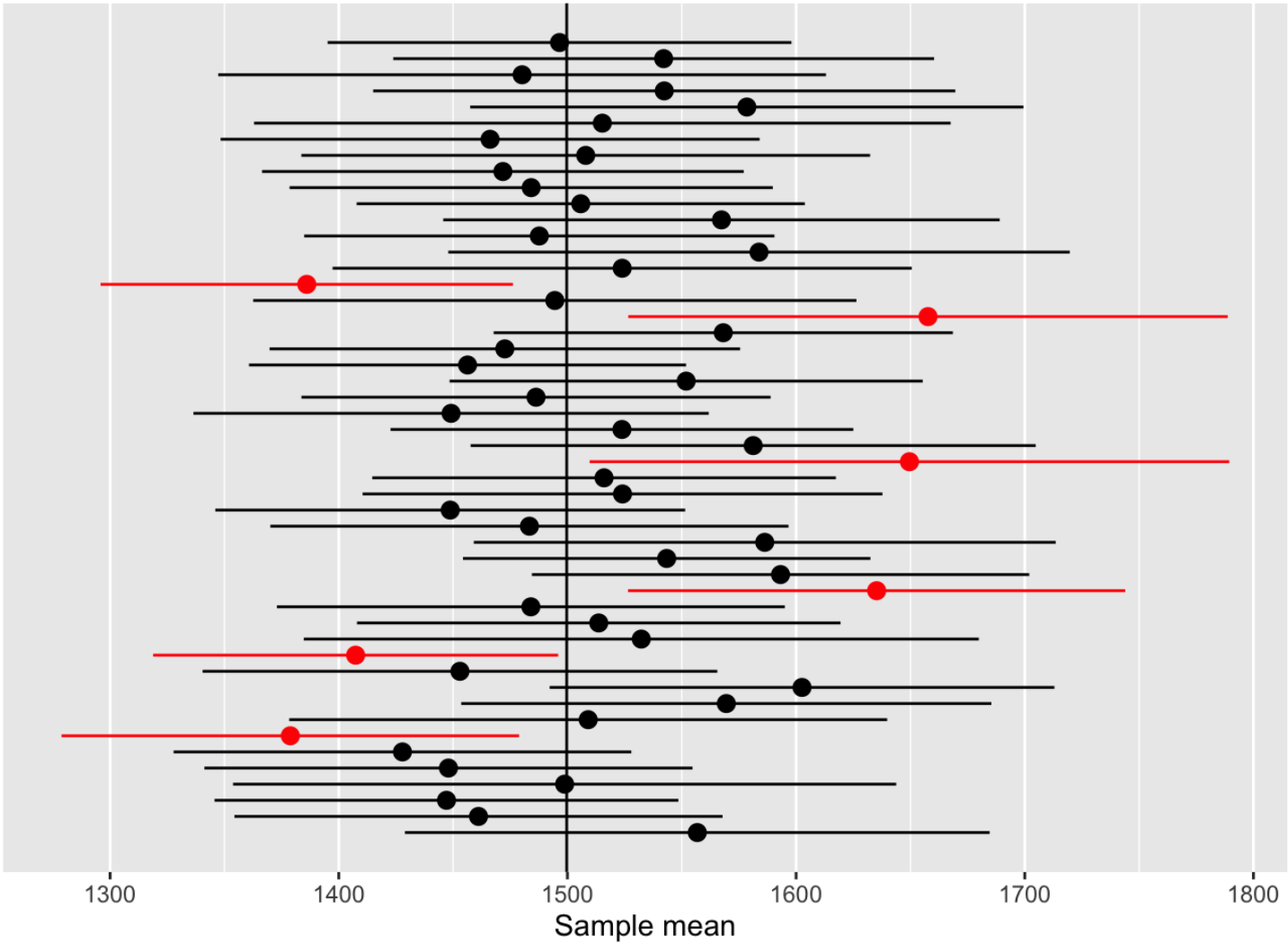


Figure 1: 100 confidence intervals for Ames housing

Some well-known confidence intervals

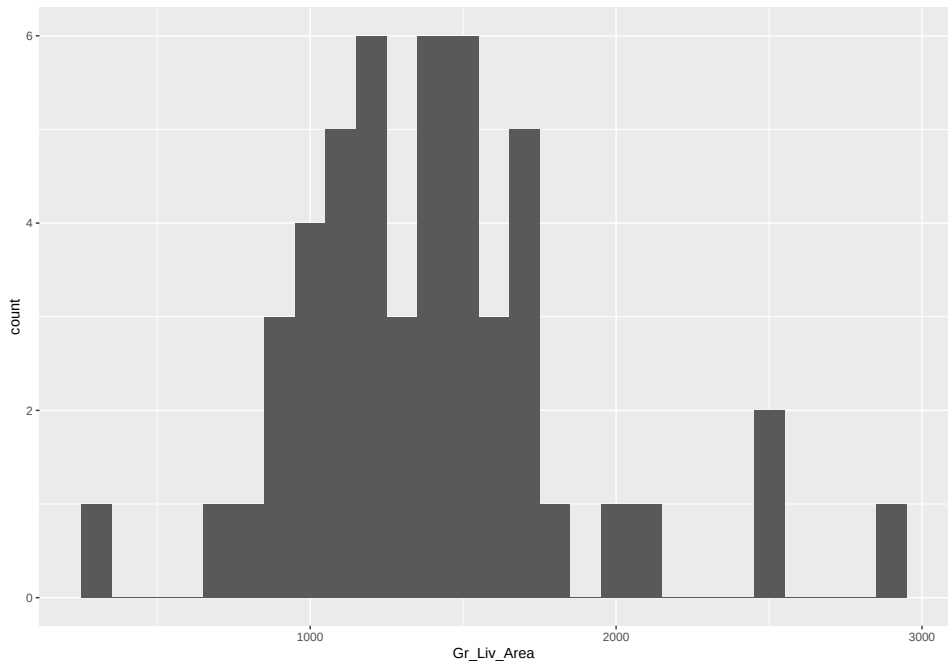
Theoretical constructs

During the lecture, we will go over confidence intervals for means, differences in means, by hand. Includes small examples.

Estimating population mean

Application: Ames housing: R

```
sampl <- ames %>%
  sample_n(size = 50)
library(ggplot2)
ggplot(data = sampl, aes(x = Gr_Liv_Area)) +
  geom_histogram(binwidth = 100)
```



What is the mean area of a home sold in Ames, IA?

Question:

What statistics are we supposed to use?

What is the mean area of a home sold in Ames, IA?

Question:

What statistics are we supposed to use? What is its sampling distribution?

What is the mean area of a home sold in Ames, IA?

Question:

What statistics are we supposed to use? What is its sampling distribution?

```
housing.area.t <- t.test(sampl$Gr_Liv_Area)
#ignore all output for now except the confidence interval
#(because it is NOT meaninguful!)
housing.area.t
```

```
    One Sample t-test

data:  sampl$Gr_Liv_Area
t = 21.519, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1261.807 1521.753
sample estimates:
```

Compare:

```
t.test(sampl$Gr_Liv_Area,conf.level = .98)
```

```
    One Sample t-test

data:  sampl$Gr_Liv_Area
t = 21.519, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
98 percent confidence interval:
 1236.24 1547.32
sample estimates:
mean of x
  1391.78
```

Estimating difference in means between two populations

Application: Birhtweight: R

- Recall the data set:

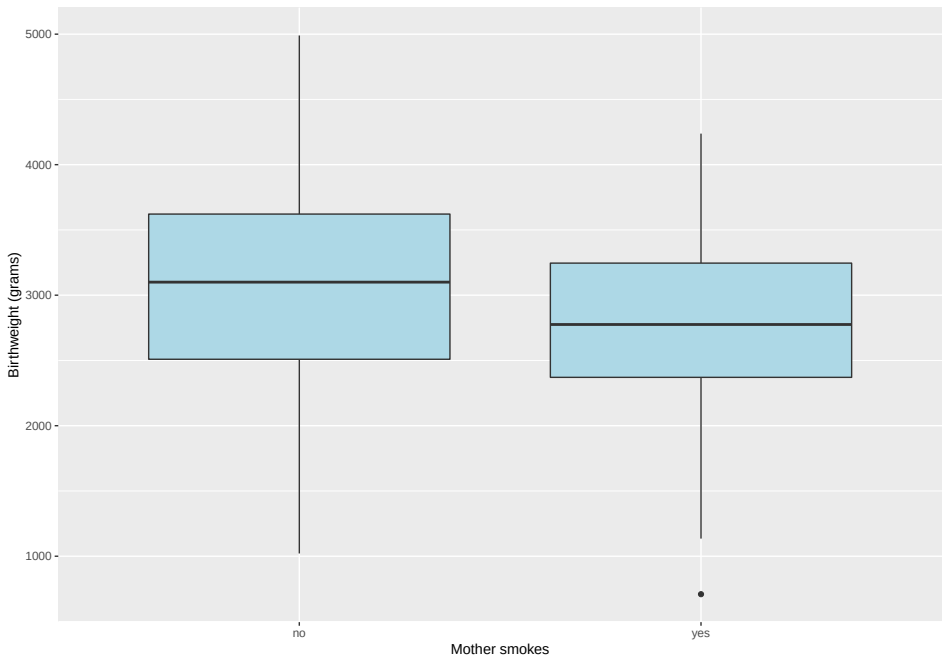
```
birthwt <- as_tibble(birthwt)
birthwt
```

```
# A tibble: 189 x 10
   birthwt.below.2~ mother.age mother.weight race  mother.smok
   <fct>                  <int>         <int> <fct> <fct>
 1 no                        19           182 black no
 2 no                        33           155 other no
 3 no                        20           105 white yes
 4 no                        21           108 white yes
 5 no                        18           107 white yes
 6 no                        21           124 other no
 7 no                        22           118 white no
 8 no                        17           103 other no
 9 no                        29           123 white yes
10 no                        26           113 white yes
```

Difference in means

Create boxplot showing how birthwt.grams varies between smoking status:

```
qplot(x = mother.smokes, y = birthwt.grams,
      geom = "boxplot", data = birthwt,
      xlab = "Mother smokes",
      ylab = "Birthweight (grams)",
      fill = I("lightblue"))
```



This plot suggests that smoking is associated with lower birth weight.

How can we assess whether this difference is statistically significant?

Let's compute a summary table

```
birthwt %>%
  group_by(mother.smokes) %>%
  summarize(mean.birthwt = round(mean(birthwt.grams), 0),
            sd.birthwt = round(sd(birthwt.grams), 0))
```

```
# A tibble: 2 x 3
  mother.smokes mean.birthwt sd.birthwt
* <fct>                <dbl>      <dbl>
1 no                    3056        753
2 yes                   2772        660
```

Question:

What statistics are we supposed to use?

Question:

What statistics are we supposed to use? What is its sampling distribution?

Question:

What statistics are we supposed to use? What is its sampling distribution?

Therefore, we should use commands that involve `t`,

Question:

What statistics are we supposed to use? What is its sampling distribution?

Therefore, we should use commands that involve `t`, such as `t.test`:

```
birthwt.t.test <- t.test(birthwt.grams ~ mother.smokes,
                         data = birthwt)
birthwt.t.test
```

```
    Welch Two Sample t-test

data:  birthwt.grams by mother.smokes
t = 2.7299, df = 170.1, p-value = 0.007003
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  78.57486 488.97860
sample estimates:
 mean in group no mean in group yes
         3055.696          2771.919
```

We see from this output that the difference is highly significant. The t.test() function also outputs a confidence interval for us.

Notice that the function returns a lot of information, and we can access this information element by element:

```
names(birthwt.t.test)
```

```
 [1] "statistic"   "parameter"   "p.value"     "conf.int"    "
 [6] "null.value"  "stderr"      "alternative" "method"      "
```

```
birthwt.t.test$estimate  # group means
```

```
 mean in group no mean in group yes
         3055.696           2771.919
```

```
birthwt.t.test$conf.int  # confidence interval for difference
```

```
[1]  78.57486 488.97860
attr(,"conf.level")
[1] 0.95
```

» Markdown tricks! «

The ability to pull specific information from the output of the hypothesis test allows you to report your results using inline code chunks. That is, you don't have to hardcode estimates, p-values, confidence intervals, etc.

```
# Calculate difference in means between smoking and nonsmoking
birthwt.t.test$estimate
```

```
 mean in group no mean in group yes
        3055.696          2771.919
```

```
birthwt.smoke.diff <-
  round(birthwt.t.test$estimate[1]
        - birthwt.t.test$estimate[2], 1)
# Confidence level as a %
conf.level <-
  attr(birthwt.t.test$conf.int, "conf.level") * 100
conf.level
```

```
[1] 95
```

Example: Here's what happens when we knit the following paragraph.

```
Our study finds that birth weights are on average
`r birthwt.smoke.diff`g higher in the non-smoking group
compared to the smoking group
(t-statistic `r round(birthwt.t.test$statistic,2)`,
  p=`r round(birthwt.t.test$p.value, 3)`,
  `r conf.level`% CI [`r round(birthwt.t.test$conf.int,1)`]g)
```

Output:

Our study finds that birth weights are on average 283.8g higher in the non-smoking group compared to the smoking group (t-statistic 2.73, p=0.007, 95% CI [78.6, 489]g)

Extras

There are nicer ways (that are not the basic thing we're using so far) to plot and visulaize t-test and its outputs. For a nice reference, see this page.

Application: Python

Scipy.Stats t! The `t` class have similar behavior like `t.test` in R, for constructing a mean sample t-test you can modify the parameters: `df`(degree of freedom), `mean`(sample mean), `sd`(sample standard error) and `Confidence_level`, and then use the following command:

```
confidence_interval =
scipy.stats.t.interval(Confidence_level, df, mean, sd).
```

For 2 sample t-test, you can use the following function (remember that you can change the confidence level as your desired value in the function):

```
# py_install("pandas")
# py_install("numpy")
```

```
import numpy as np
from scipy.stats import ttest_ind
from scipy.stats import t
import pandas as pd

def welch_ttest(x1, x2):

    n1 = x1.size
    n2 = x2.size
    m1 = np.mean(x1)
    m2 = np.mean(x2)

    v1 = np.var(x1, ddof=1)
    v2 = np.var(x2, ddof=1)
    pooled_se = np.sqrt(v1 / n1 + v2 / n2)  #computing the sd
    delta = m1-m2

    tstat = delta /  pooled_se
    df = (v1 / n1 + v2 / n2)**2 / (v1**2 / (n1**2 * (n1 - 1)) + v2**2 / (n2**2 * (n2 - 1)))
              #computing the df

    # two side t-test
    p = 2 * t.cdf(-abs(tstat), df)   #p-value

    # upper and lower bounds
    lb = delta - t.ppf(0.975,17.9)*pooled_se
    ub = delta + t.ppf(0.975,17.9)*pooled_se

    return pd.DataFrame(np.array([tstat,df,p,delta,lb,ub]).reshape(1,-1),
                        columns=['T statistic','df','pvalue 2 sided','Difference in mean','lb','ub'])
                        #the interval is (lb,ub)
```

There are several scenarios that can happen when there are two populations!

All kinds of t-tests...[1]

The `t.test( )` function in `R` produces a variety of t-tests. Unlike most statistical packages, the default assumes unequal variances... here is the scoop:

```
# independent 2-group t-test
t.test(y~x) # where y is numeric and x is a binary factor
# independent 2-group t-test
t.test(y1,y2) # where y1 and y2 are numeric
# paired t-test
t.test(y1,y2,paired=TRUE) # where y1 & y2 are numeric
# one sample t-test
t.test(y,mu=3) # Ho: mu=3
```

You can use the var.equal = TRUE option to specify equal variances and a pooled variance estimate. You can use the alternative="less" or alternative="greater" option to specify a one tailed test.

[1] Neat summary from https://www.statmethods.net/stats/ttest.html

Wait. What are these ‘tests’ to which we are referring?!

Next, let’s introduce a formal statistical procedure called a **hypothesis test**!

License

This document is created for ITMD/ITMS/STAT 514, Spring 2021, at Illinois Tech.

Acknowledgement

Parts of this lecture were sourced from Prof. Alexandra Chouldechova, released under a Attribution-NonCommercial-ShareAlike 3.0 United States license.