STAB57 Week-12

Shahriar Shams

April 04, 2023

Learning goals

- Some useful codes
- Bootstrap confidence interval
- Permutation test
- Disitribution of p-value

Some useful codes

Equality of the variances

- We are interested in testing $H_0 : \sigma_x^2 = \sigma_y^2$
- Alternative hypothesis can be any of the followings
  - $H_a : \sigma_x^2 \neq \sigma_y^2$
  - $H_a : \sigma_x^2 < \sigma_y^2$
  - $H_a : \sigma_x^2 > \sigma_y^2$
- Suppose these following two groups of sample observations were collected from two different populations.

```
G1=c(79.98,80.04,80.02,80.04,80.03,80.03,80.04,79.97,
     80.05,80.03,80.02,80.00,80.02)
G2=c(80.02,79.94,79.98,79.97,79.97,80.03,79.95,79.97)
```

```
var.test(G1,G2,
         alternative = "two.sided",
         conf.level = 0.95)
```

```
##
##  F test to compare two variances
##
## data:  G1 and G2
## F = 0.58374, num df = 12, denom df = 7, p-value = 0.3938
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1251097 2.1052687
## sample estimates:
## ratio of variances
##          0.5837405
```

Equality of the means

- We are interested in testing $H_0 : \mu_x = \mu_y$
- Alternative hypothesis can be any of the followings
  - $H_a : \mu_x \neq \mu_y$
  - $H_a : \mu_x < \mu_y$
  - $H_a : \mu_x > \mu_y$
- When we are testing for the equality of the two means, the test depends on the variance parameters.
- If the variances are equal (i.e. $\sigma_x^2 = \sigma_y^2$), we use one type of test, if the variances are not equal we use a different test.

Variances are assumed to be equal

```
t.test(G1,G2,
         alternative = "two.sided",
         conf.level = 0.95,
         var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  G1 and G2
## t = 3.4722, df = 19, p-value = 0.002551
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01669058 0.06734788
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

Variances are assumed to be unequal

```
t.test(G1,G2,
         alternative = "two.sided",
         conf.level = 0.95,
         var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  G1 and G2
## t = 3.2499, df = 12.027, p-value = 0.006939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01385526 0.07018320
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

testing $H_0 : \mu_x - \mu_y = 1$

```
t.test(G1,G2,
         mu = 1,
         alternative = "two.sided",
```

```
        conf.level = 0.95,
        var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  G1 and G2
## t = -79.162, df = 19, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 1
## 95 percent confidence interval:
##  0.01669058 0.06734788
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

Equality of proportions

- Suppose we tested 250 students at UTSC and 20 of them tested positive. And we tested 300 students at UofT St George and 35 of them tested positive.
- We want to test if the proportion of test positives are the same in the two campuses.
- We want to test $H_0 : \pi_x = \pi_y$

```
x= c(20,35)
n = c(250,300)
prop.test(x,n)
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  x out of n
## X-squared = 1.65, df = 1, p-value = 0.199
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.08983624  0.01650291
## sample estimates:
##    prop 1    prop 2
## 0.0800000 0.1166667
```

Paired t-test

- Example taken from week-8 lecture slides.

```
x=c(10.19, 7.92, 6.67, 12.22, 8.21, 8.26, 13.06, 8.20, 9.83, 5.94)
y=c(7.00, 7.53, 6.45, 1.31, 5.42, 2.81, 6.60, 0.55, 3.13, 5.00)

t.test(x, y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  x and y
## t = 3.9873, df = 9, p-value = 0.003171
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  1.933984 7.006016
## sample estimates:
## mean difference
##            4.47
```


two sample t-test using regression approach

Suppose we have measurements from two groups: Group1 and Group2.

```
# Let's enter the data in R
G1=c(79.98,80.04,80.02,80.04,80.03,80.03,80.04,79.97,
     80.05,80.03,80.02,80.00,80.02)
G2=c(80.02,79.94,79.98,79.97,79.97,80.03,79.95,79.97)

mean(G1)
```

```
## [1] 80.02077
```

```
mean(G2)
```

```
## [1] 79.97875
```

```
# Converting into what is known as "long" format data
Y=c(G1,G2)
# Creating the dummy variable
X_G1=c(rep(1,length(G1)),rep(0,length(G2)))

# Let's see how the data looks
cbind(Y,X_G1)
```

```
##           Y X_G1
##  [1,] 79.98    1
##  [2,] 80.04    1
##  [3,] 80.02    1
##  [4,] 80.04    1
##  [5,] 80.03    1
##  [6,] 80.03    1
##  [7,] 80.04    1
##  [8,] 79.97    1
##  [9,] 80.05    1
## [10,] 80.03    1
## [11,] 80.02    1
## [12,] 80.00    1
## [13,] 80.02    1
## [14,] 80.02    0
## [15,] 79.94    0
## [16,] 79.98    0
## [17,] 79.97    0
## [18,] 79.97    0
## [19,] 80.03    0
## [20,] 79.95    0
## [21,] 79.97    0
```

```r
# Fitting a linear model
m=lm(Y~X_G1)
summary(m)
```

```
##
## Call:
## lm(formula = Y ~ X_G1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.050769 -0.008750 -0.000769  0.019231  0.051250
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 79.978750   0.009521 8399.914  < 2e-16 ***
## X_G1         0.042019   0.012101    3.472  0.00255 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02693 on 19 degrees of freedom
## Multiple R-squared:  0.3882, Adjusted R-squared:  0.356
## F-statistic: 12.06 on 1 and 19 DF,  p-value: 0.002551
```

```r
# Confidence intervals of regression parameters
confint(m,level = 0.95)
```

```
##                   2.5 %      97.5 %
## (Intercept) 79.95882153 79.99867847
## X_G1         0.01669058  0.06734788
```

Let's do a t-test assuming variances are equal

```r
t.test(G1,G2, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  G1 and G2
## t = 3.4722, df = 19, p-value = 0.002551
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01669058 0.06734788
## sample estimates:
## mean of x mean of y
##  80.02077  79.97875
```

Bootstrap confidence interval

- Bootstrapping is the idea of mimicking the concept of re-sampling.

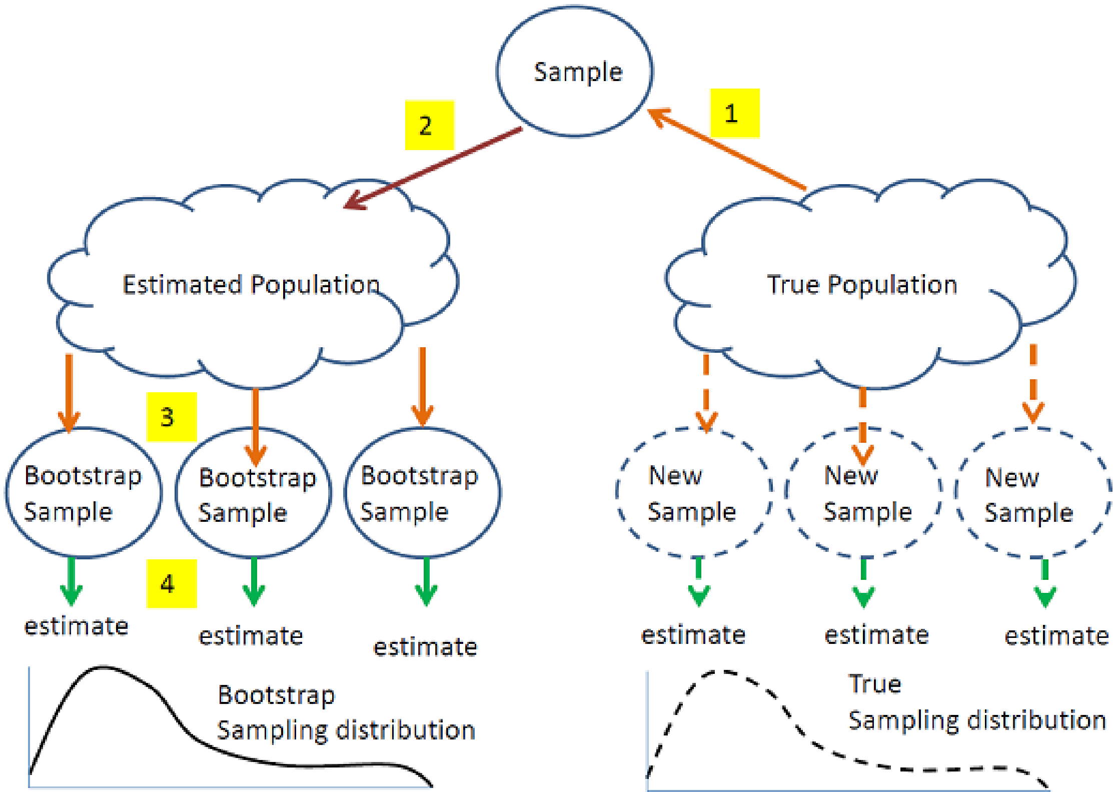


fig source: url{https://online.stat.psu.edu/stat555/node/119/}

- Central limit theorem only applies to the distribution of sample mean.
- If we are interested in any other parameter other than the population mean and our estimator is not sample mean, we can't apply central limit theorem anymore.
- Then we rely on figuring out the distribution of our estimator which could be mathematically challenging.
- Bootstrap is one quick way of solving this problem.

- R demonstration:

```r
# A hypothetical population (height of 60K UofT students)
pop=rnorm(60000, mean=160, sd=10)

set.seed(999)
z=sample(pop,size=20)


boot_function=function(){
  boot_s = sample(z,size=20,replace=TRUE)
  return(mean(boot_s))
}


boot_X_bar = replicate(100000,boot_function())

#95% bootstrap CI
quantile(boot_X_bar, c(0.025,0.975))
```

```
##     2.5%    97.5%
## 158.2884 167.3873
```

```r
#95% CI using t-dist
t.test(z)
```

```
##
##  One Sample t-test
##
## data:  z
## t = 68.25, df = 19, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  158.0442 168.0444
## sample estimates:
## mean of x
##  163.0443
```

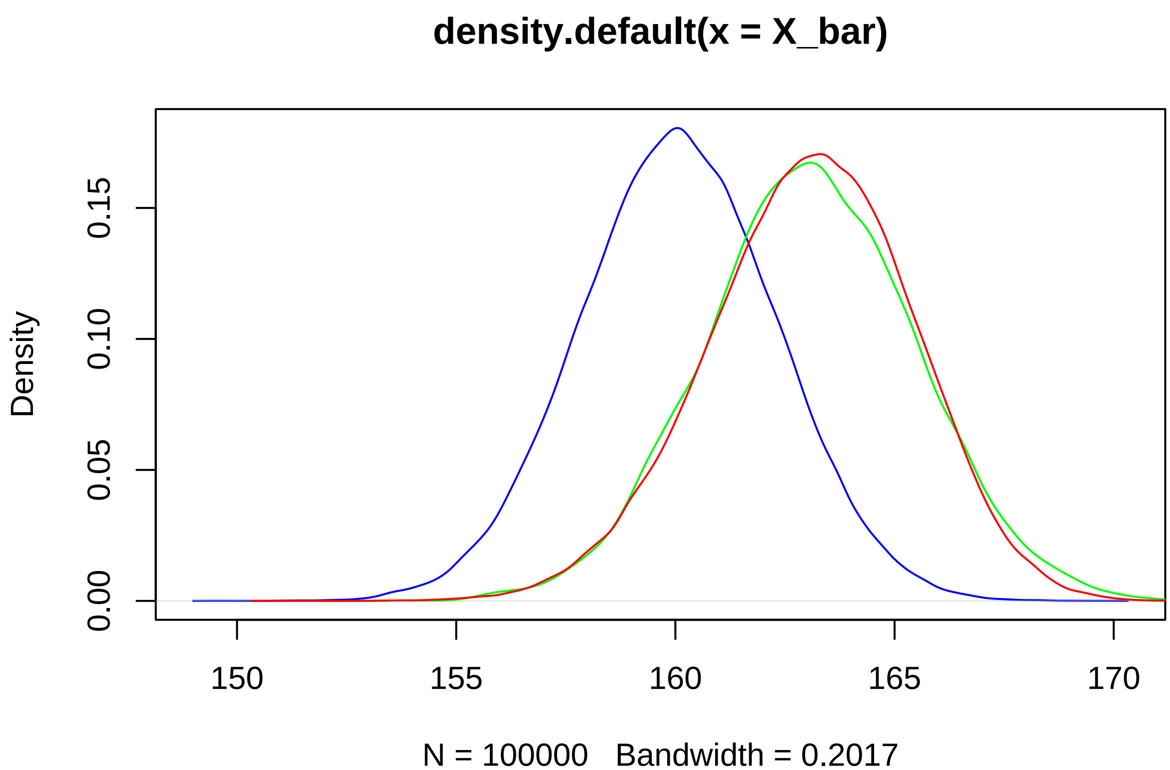
```r
#95% CI using t-dist (another way)
confint(lm(z~1))
```

```
##                2.5 %   97.5 %
## (Intercept) 158.0442 168.0444
```

```
# Our known sampling distribution
sample_function= function(){
  s=sample(pop,size=20)
  return(mean(s))
}

X_bar=replicate(100000, sample_function())


# Comparing bootstrap distribution and estimated sampling distribution
# to the original sampling distribution
plot(density(X_bar),col="blue")
lines(density(rnorm(10000,mean=mean(z),sd=sqrt(var(z)/length(z)))),col="green")
lines(density(boot_X_bar),col="red")
```



Permutation test

- t.test() relies on the assumption that the population distribution is Normal.
- If the true distribution is not Normal, the test and the decision made based on the test become questionable.
- Permutation test is an alternative to two sample t-test (or proportion test).
- Suppose we are comparing two treatment arms. Let's call them treatment X and treatment Y.
- Our null hypothesis is $H_0 : \mu_x = \mu_y$.
- Suppose we have 50 observation from treatment X and another 50 from treatment Y.
- Suppose the null hypothesis is true. In that case, we don't expect the samples to be any different.
- If the null is true, then changing the label of treatment X to treatment Y (and the other way around) will not effect the outcome.
- Permutation test comes from idea of permuting the labels.
- By keeping the observations as they are, and by changing the label from X to Y and Y to X repeatedly, we can construct a series of probable scenarios (with the current set of samples).
- We calculate the difference in sample mean in all these scenarios and use that to construct a "sampling distribution" of the difference.
- We then use this to calculate our p-value in the direction of the alternative hypothesis.
- Here is a really nice graphical representation of permutation test (https://www.jwilber.me/permutationtest/)

```
group_ind = c(rep("X",50),rep("Y",50))

null.obs = rnorm(100, mean = 10, sd=2)

alt.obs = rnorm(100, mean = 10 + 1*(group_ind =="Y"), sd=2)

t.test(null.obs~group_ind, var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  null.obs by group_ind
## t = -1.7156, df = 98, p-value = 0.08939
## alternative hypothesis: true difference in means between group X and group Y is not equal
## 95 percent confidence interval:
##  -1.622310  0.117863
## sample estimates:
## mean in group X mean in group Y
##        9.726166       10.478389
```

```
t.test(alt.obs~group_ind, var.equal = T)
```

```
##
```

```
##  Two Sample t-test
##
## data:  alt.obs by group_ind
## t = -1.8777, df = 98, p-value = 0.0634
## alternative hypothesis: true difference in means between group X and group Y is not equal
## 95 percent confidence interval:
##  -1.61692285  0.04470816
## sample estimates:
## mean in group X mean in group Y
##        10.11771        10.90382
```

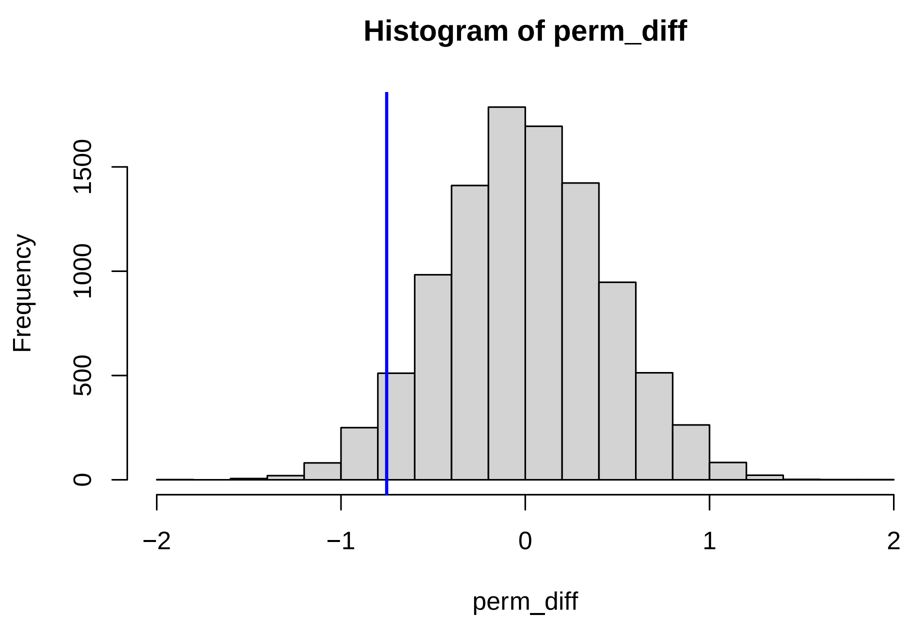
Let's code our permutation test

```
obs.diff = mean(null.obs[group_ind== "X"]) - mean(null.obs[group_ind== "Y"])

perm.function = function(){
  temp_group_ind = sample(group_ind)
  d = mean(null.obs[temp_group_ind== "X"]) - mean(null.obs[temp_group_ind== "Y"])
  return(d)
}

perm_diff = replicate(10000, perm.function())

hist(perm_diff)
abline(v=obs.diff,lwd=2,col="blue")
```




```
# p-value from permutation test

mean(abs(perm_diff) > abs(obs.diff))
```

```
## [1] 0.0924
```

```r
# one sided p-values

# greater
mean(perm_diff > obs.diff)
```

```
## [1] 0.9533
```

```r
#smaller
mean(perm_diff < obs.diff)
```

```
## [1] 0.0467
```

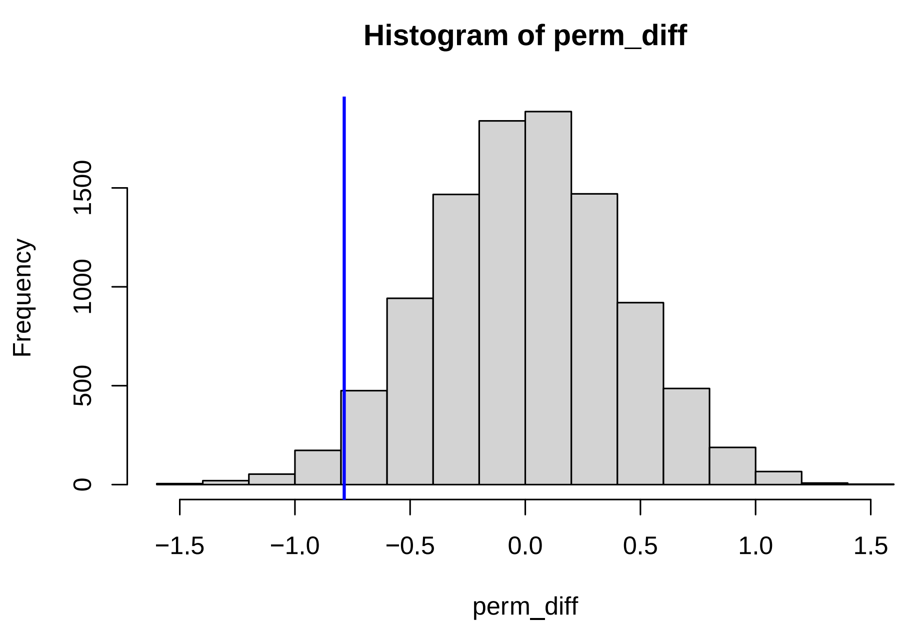
- Let's do the same test when the alternative is true (i.e. null is false)

```r
obs.diff = mean(alt.obs[group_ind== "X"]) - mean(alt.obs[group_ind== "Y"])

perm.function = function(){
  temp_group_ind = sample(group_ind)
  d = mean(alt.obs[temp_group_ind== "X"]) - mean(alt.obs[temp_group_ind== "Y"])
  return(d)
}

perm_diff = replicate(10000, perm.function())

hist(perm_diff)
abline(v=obs.diff,lwd=2,col="blue")
```



```r
# p-value from permutation test
mean(abs(perm_diff) > abs(obs.diff))
```

```
## [1] 0.0556
```

Disitribution of p-value

```r
#H0: mu=1
#Ha: mu=4
#sigma^2 = 9
#n=9

my_function=function(){
  x=rnorm(9,mean=1,sd=3)
  z=(mean(x)-1)/sqrt(9/9)
  p_val=1-pnorm(z)
  return(p_val)
}

output=replicate(10000,my_function())

hist(output)
```