More on Inference

Sumanta Basu

Logistics

Reading list updated

HW8 posted, due on Monday

If you have not submitted exploration components 1 or 2, you can do so before Thanksgiving break

Thanks to those who shared exploration reports on blackboard discussion forum

Reading

Textbook Sections: 5.3.6, 6.2.3, 6.5, 6.6

What is statistical inference?

:)

> An economist, a financier & a statistician went hunting.

> They saw a bird & the economist took aim. He shot slightly to the left and missed.

> The financier shot to the right and missed.

> The statistician celebrated: “We got him!”

- [*Courtesy: Rachel N., BTRY 6010 Fall’19*]

Confidence Intervals

- Give a point estimate, add/subtract margin of error, give measure of uncertainty

Hypothesis Tests

Choose between two alternatives (null: status quo, alternative: your *exciting* research findings)

Keep in mind:

- be conservative, i.e. measure chances of making false discovery (reduce type-I error, control **level**)
- but try to make sure you don't walk away from a true discovery (reduce type-II error, increase **power**)

Types of tests we have seen so far

- one sample mean (z-test, t-test) and proportion (z-test)
- two-sample mean (z-test, t-test) and proportions (z-test)
- Tests of independence, goodness-of-fit (χ^2-test)

Assumptions: can’t live with ’em, can’t live without ’em

Assumptions for one-sample mean

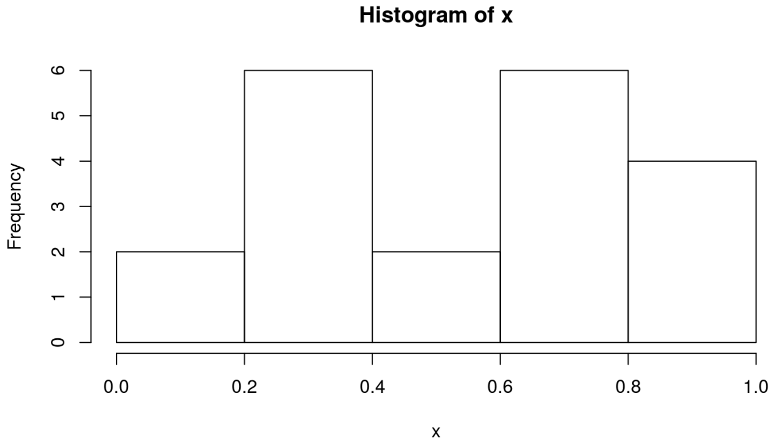
- large n, independence, data distribution not too skewed (use normal approximation and z-test)

OR …

- small n, independence, data distribution exactly normal (use t-test)

What if data provides strong evidence of departure from null mean, but fails normality assumption?

- Example: $H_0 : \mu = 5, H_A : \mu \neq 5$, data is between 0 and 1, but does not look normal



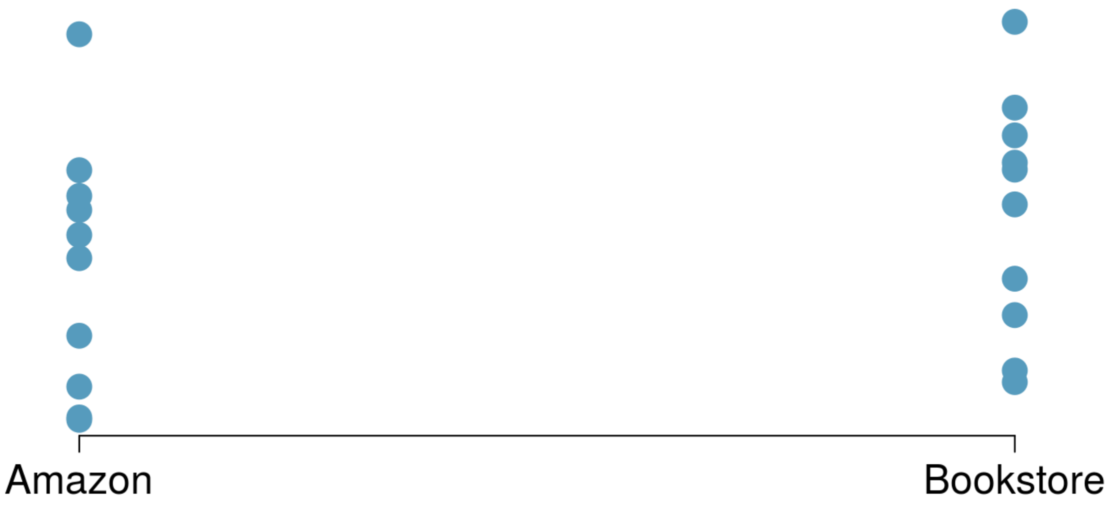
Role of Assumptions

- If you make assumptions (and they are reasonably satisfied in your data), you can construct a more powerful test
- If you use tests with less assumptions, they are typically less powerful but more conservative

Case I: pairing increases power

Assumptions can help: pairing increases power

Use example from Amazon and local bookstore prices



Paired vs. Unpaired t-tests

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

```
##
##  Paired t-test
##
## data:  x and y
## t = -9.606, df = 9, p-value = 4.995e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.401393 -0.248376
## sample estimates:
## mean of the differences
##              -0.3248845
```

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

```
##
##  Welch Two Sample t-test
##
## data:  x and y
## t = -0.95096, df = 17.965, p-value = 0.3542
## alternative hypothesis: true difference in means is not e
## 95 percent confidence interval:
##  -1.0427406  0.3929716
## sample estimates:
## mean of x mean of y
##  8.132203  8.457087
```

Detour: two-sided vs. one sided tests

- $H_0 : \mu_1 = \mu_2$ vs. $H_A : \mu_1 - \mu_2 \neq 0$
- p-value for the two-sided unpaired test is 0.35, sample mean difference $\bar{x}_1 - \bar{x}_2 = -0.32$

Question: What is the p-value if $H_A : \mu_1 - \mu_2 < 0$?

- Answer: (A) 0.35, (B) 1-0.35, (C) 0.35/2, (D) 1 - 0.35/2

Question: What is the p-value if $H_A : \mu_1 - \mu_2 > 0$?

- Answer: (A) 0.35, (B) 1-0.35, (C) 0.35/2, (D) 1 - 0.35/2

Try on your own:

```
x = c(7.37, 8.18, 7.16, 9.60, 8.33, 7.18,  8.49, 8.74, 8.58, 7.69)  # Amazon Prices
y = c(7.82, 8.52, 7.40, 9.67, 8.74, 7.48, 8.79, 9.13, 8.96, 8.05) # Local Prices
t.test(x, y, alternative = 'two.sided')
t.test(x, y, alternative = 'less')
t.test(x, y, alternative = 'greater')
```

Case II: pooling increases power

Two sample test: variance unknown

Comparison of two unrelated means $H_0 : \mu_X = \mu_Y$ vs. $H_A : \mu_X \neq \mu_Y$

Sample statistic = $\bar{X}_{n_X} - \bar{Y}_{n_Y}$

If $n_X > 30$ and $n_Y > 30$, then using $S^X_{n_X}$ and $S^Y_{n_Y}$ is fine:

$$\frac{(\bar{X}_{n_X} - \bar{Y}_{n_Y}) - (\mu_X - \mu_Y)}{\sqrt{\frac{[S^X_{n_X}]^2}{n_X} + \frac{[S^Y_{n_Y}]^2}{n_Y}}} \text{ approximately } N(0, 1).$$

but what if n_X or n_Y is small?

Variance unknown, small n

Unfortunately, not as simple as one-sample case

$$\frac{(\bar{X}_{n_X} - \bar{Y}_{n_Y}) - (\mu_X - \mu_Y)}{\sqrt{\frac{[S^X_{n_X}]^2}{n_X} + \frac{[S^Y_{n_Y}]^2}{n_Y}}} \text{ approximately } t_{df}$$

- this is really an approximation, not actually a t distribution even when $\bar{X}_{n_X} - \bar{Y}_{n_Y}$ is exactly normal (unless $\sigma_X = \sigma_Y$).
- complicated formula for df (R uses this).
- simpler choice (but overly conservative) is $df = \min\{n_X - 1, n_Y - 1\}$.

Assumptions can help: pooled variance estimate

If we assume $\sigma_X = \sigma_Y$, we can revise our test statistic with a more precise estimate of standard error

$$\frac{(\bar{X}_{n_X} - \bar{Y}_{n_Y}) - (\mu_X - \mu_Y)}{S_p\sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}} \text{ approximately } t_{n_X+n_Y-2}$$

where the pooled estimator of standard deviation comes from the formula

$$S_p^2 = \frac{(n_X - 1)[S_{n_X}^X]^2 + (n_Y - 1)[S_{n_Y}^Y]^2}{n_X + n_Y - 2}$$

Intuition: we are using more data to estimate the noise level (σ^2), reasonable to expect our procedure is more precise now

Example: Fuel efficiency (1973-1974)

Question: Is manual or automatic transmission more fuel efficient?

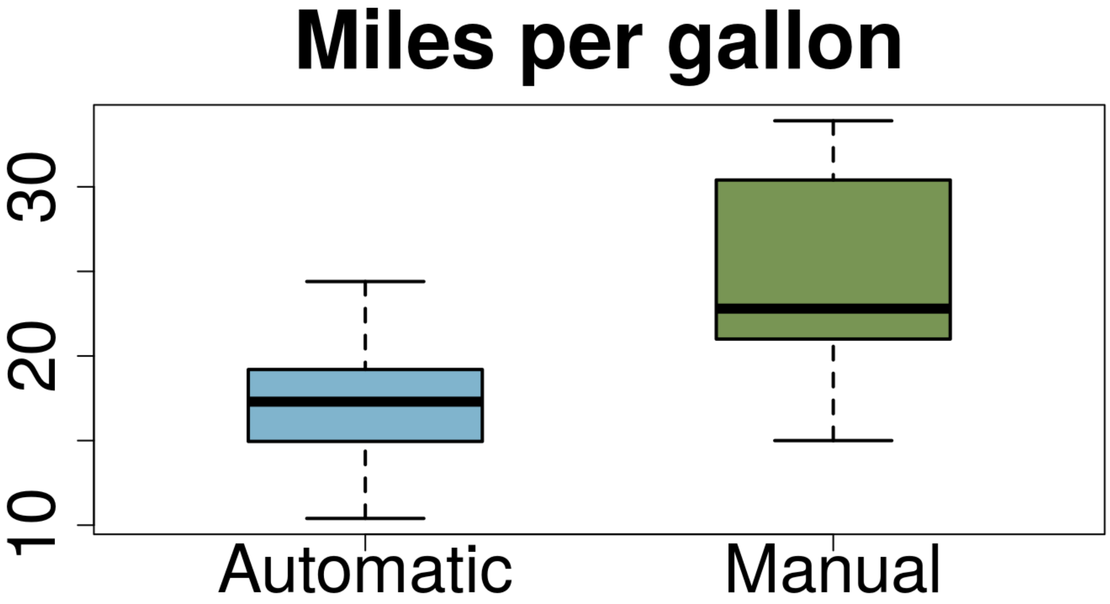
- data:
  - miles per gallon (mpg)
  - 13 manual, 19 automatic
- Hypotheses:
  - $H_0 : \mu_A = \mu_M$
  - $H_A : \mu_A \neq \mu_M$
- Paired or unpaired?

mtcars data in R (first 10 rows)

```
##                    mpg am
## Mazda RX4         21.0  1
## Mazda RX4 Wag     21.0  1
## Datsun 710        22.8  1
## Hornet 4 Drive    21.4  0
## Hornet Sportabout 18.7  0
## Valiant           18.1  0
## Duster 360        14.3  0
## Merc 240D         24.4  0
## Merc 230          22.8  0
## Merc 280          19.2  0
```

Always plot first

```
boxplot(mpg ~ am, data = mtcars, names = c("Automatic", "Manual"),
        main="Miles per gallon")
```



Check assumptions

- Need $\bar{X}_{n_X}$ approximately normal…
    - $n_X = 19$… would need data distribution normal
    - independent?
- Same for $\bar{Y}_{n_Y}$.
- And need the two samples independent (since not paired)

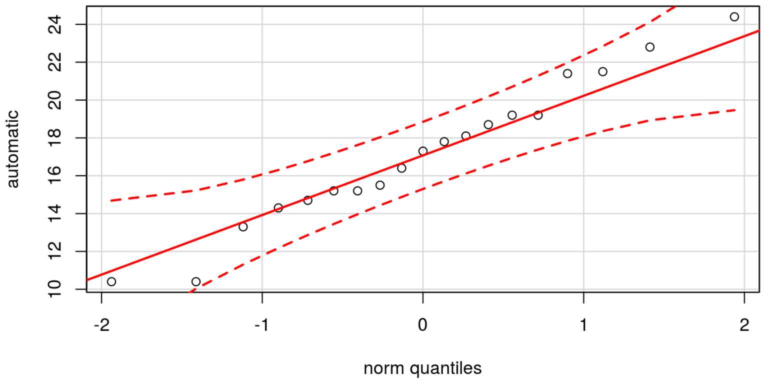
In R

```
automatic <- subset(mtcars, am == 0)$mpg
manual <- subset(mtcars, am == 1)$mpg
library(car) # for qqPlot function
```

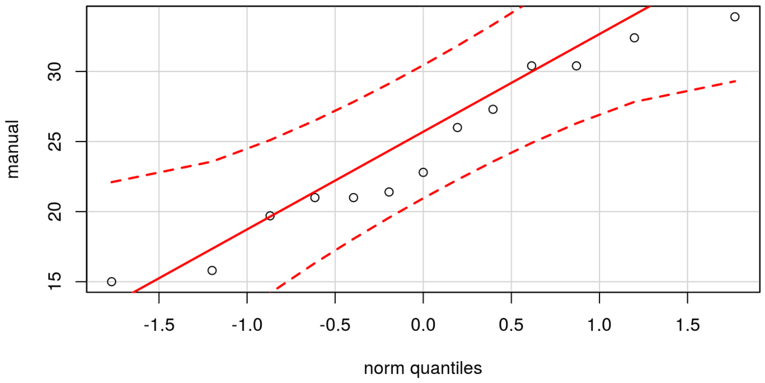
- package `car` has nothing to do with cars (stands for “Companion to Applied Regression”)

Approximately normal?

`qqPlot(automatic)`



`qqPlot(manual)`



Doing test

```
out = t.test(x = manual, y = automatic)
out
```

```
##
##  Welch Two Sample t-test
##
## data:  manual and automatic
## t = 3.7671, df = 18.332, p-value = 0.001374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.209684 11.280194
## sample estimates:
## mean of x mean of y
##  24.39231  17.14737
```

```
out$p.value
```

```
## [1] 0.001373638
```

```
out2 = t.test(x = manual, y = automatic,
  var.equal = TRUE) # var.equal = FALSE by default
out2
```

```
##
##  Two Sample t-test
##
## data:  manual and automatic
## t = 4.1061, df = 30, p-value = 0.000285
## alternative hypothesis: true difference in means is not e
## 95 percent confidence interval:
##   3.64151 10.84837
## sample estimates:
## mean of x mean of y
##  24.39231  17.14737
```

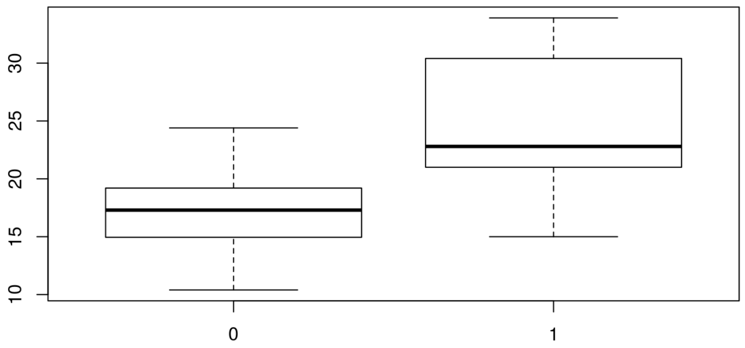
```
out2$p.value
```

```
## [1] 0.0002850207
```

Checking equal variance assumption

Compare box plots or sample standard deviations of the two groups

```
boxplot(mpg ~ am, data = mtcars)
```



```
c(sd(mtcars$mpg[mtcars$am == 0]), sd(mtcars$mpg[mtcars$am == 1]))
```

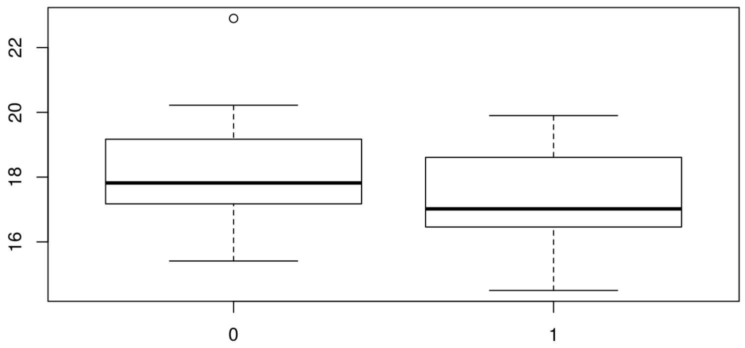
```
## [1] 3.833966 6.166504
```

Equal variance assumption does not seem to be met for mpg

Comparing acceleration across groups

Compare `qsec` (time taken to cover 1/4 mile) between manual and automatic

```
boxplot(qsec ~ am, data = mtcars)
```



```
c(sd(mtcars$qsec[mtcars$am == 0]), sd(mtcars$qsec[mtcars$am == 1]))
```

```
## [1] 1.751308 1.792359
```

Try on your own: Check normality, compare `qsec` using `t.test()`. Use `var.equal = TRUE` and `var.equal = FALSE`. See how the p-values differ.

Supplementary Reading

- We have seen how pooling variance estimates help increase power when comparing two means.
- Does pooling also work when comparing two proportions?
- **Answer**: yes, when comparing $H_0 : p_1 = p_2$ vs. $H_A : p_1 \neq p_2$ we can change standard error

$$SE(\hat{p}_1 - \hat{p}_2) = \sqrt{\frac{\hat{p}_1(1 - \hat{p}_1)}{n_1} + \frac{\hat{p}_2(1 - \hat{p}_2)}{n_2}}$$

to a more precise estimate (assuming $H_0 : p_1 = p_2$ holds)

$$SE(\hat{p}_1 - \hat{p}_2) = \sqrt{\frac{\hat{p}(1 - \hat{p})}{n_1} + \frac{\hat{p}(1 - \hat{p})}{n_2}}, \quad \hat{p} := \frac{n_1 p_1 + n_2 p_2}{n_1 + n_2}$$

- Reading: Textbook section 6.2.3

Detour: what else increases power?

- recall: high power means you have better chance of discovery
- large n, small σ, larger effect size

How do we check assumptions?

- independence: from the context, sampling procedure
- normality: histogram, Q-Q plot
- equal **population** variance: side-by-side boxplots, *sample* standard deviations, Levene's test (more on this in ANOVA)

What happens when assumptions fail (or are hard to check)?

Assumptions for one-sample mean

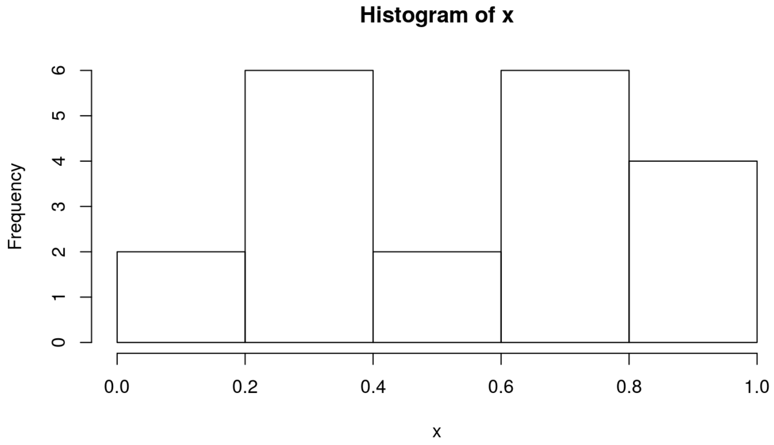
- large n, independence, data distribution not too skewed (use normal approximation and z-test)

OR …

- small n, independence, data distribution exactly normal (use t-test)

What if data provides strong evidence of departure from null mean, but fails normality assumption?

- Example: $H_0 : \mu = 5$, $H_A : \mu \neq 5$, data is between 0 and 1, but does not look normal



Assumptions for one-sample mean

Can we still devise a test that rejects $H_0 : \mu = 5$?

- rich literature of nonparametric statistics – aims to relax normality assumptions
- paves the way for thinking outside parametric family
- with modern computing power, can do more (permutation, bootstrap, Monte Carlo simulation) to measure uncertainty

A little bit of Nonparametric Inference

Outline

- Sign test, Signed rank test (paired mean problem)
- Mann-Whitney-Wilcoxon U-test or rank sum test (unpaired mean problem)
- Kruskal-Wallis test (~~test of independence~~)

Non parametric

- Spearman's rank correlation, Kendall's tau (nonparametric measures of correlation)
- many more …

Instead of working with raw data we work with signs and rank

Sign Test

Non parametric analog for paired test

- work on board

In nonparametric we don't use "mean" as much. More, do they come from the same distribution, or different?

Calculate the mean, look at the sign and then the rank

reject null if # of +1 in diff is >> Expected value (n/2) `binom.test` in r

Signed Rank test

- work on board

If the diff is >> than 0, we may want to include the rank of the absolute value
Then, sign * rank. e.g. if the sign is - and the rank is 5, then signrank = -5.

reject null if sum of sign rank >> or << 0

Wilcoxon Rank Sum test

Also known as Mann-Whitney-Wilcoxon U-test

- work on board

Compare book prices: signed rank test

```r
x = c(7.37, 8.18, 7.16, 9.60, 8.33, 7.18,  8.49, 8.74, 8.58, 7.69)  # Amazon Prices
y = c(7.82, 8.52, 7.40, 9.67, 8.74, 7.48, 8.79, 9.13, 8.96, 8.05) # Local Prices

wilcox.test(x, y, mu = 0, paired = TRUE, conf.int = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  x and y
## V = 0, p-value = 0.001953
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.395 -0.240
## sample estimates:
## (pseudo)median
##          -0.34
```

sign rank test statistic

This is used when distribution is NOT normal. low power test

Compare MPG: rank sum test

```
automatic <- subset(mtcars, am == 0)$mpg
manual <- subset(mtcars, am == 1)$mpg
wilcox.test(x = manual, y = automatic, paired = FALSE, conf.int = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  manual and automatic
## W = 205, p-value = 0.001871
## alternative hypothesis: true location shift is not equal to 0
```

Non parametric analog for unpaired test. Put all obs. in the same bucket and rank them.

Reject null if the distribution of the two categories are not evenly distributed

Permutation (Randomization) test

Main Idea

- Similar to bootstrap principle of simulating pseudo data from observed data
- Assuming null hypothesis is true, permute data many times and caculate sample (or test) statistic on each of these new data sets
- Distribution of these sample statistics (over many permuted samples) will approximate sampling distribution under null
- Check how extreme is your observed statistic in this historgram
- If your observed statistics looks more like an outlier in this distribution, data provides stronger evidence against H_0

Permutation (randomization) Test

Consider the book price comparison example

Under null (*distributions* of book prices are same in both stores), we could switch labels of some books (Amazon to local and vice versa), and the distributions of book prices will still be the same.

Key Idea: randomly shuffle book prices for each pair, and look at the distribution of sample mean differences (draw a histogram)

```
x = c(7.37, 8.18, 7.16, 9.60, 8.33, 7.18,  8.49, 8.74, 8.58, 7.69)  # Amazon Prices
y = c(7.82, 8.52, 7.40, 9.67, 8.74, 7.48, 8.79, 9.13, 8.96, 8.05) # Local Prices
mean(x - y)
```

```
## [1] -0.324
```

Now, flip some of the labels in random (e.g. switch prices of book 2, 3, 9)!

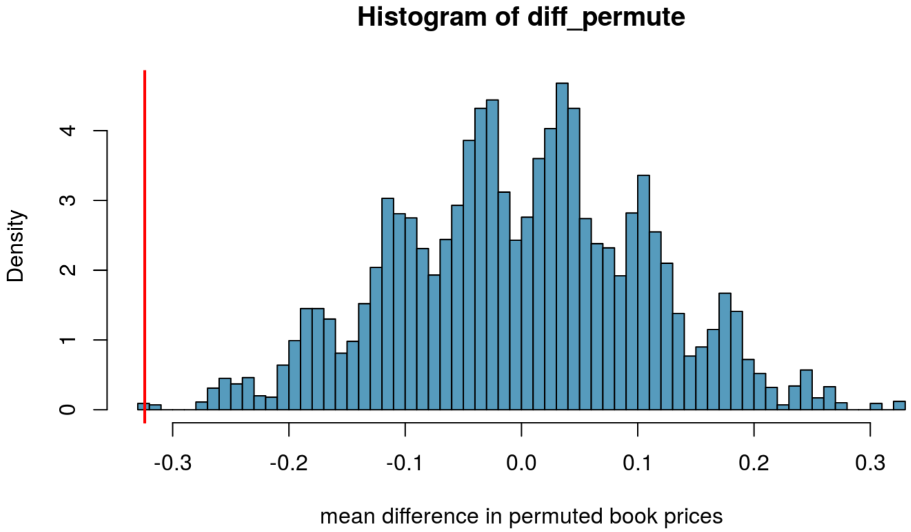
```
x_permute = c(7.37, 8.52, 7.40, 9.60, 8.33, 7.18,  8.49, 8.74, 8.96, 7.69)  # Amazon Prices
y_permute = c(7.82, 8.18, 7.16, 9.67, 8.74, 7.48, 8.79, 9.13, 8.58, 8.05) # Local Prices
mean(x_permute - y_permute)
```

```
## [1] -0.132
```

Permutation (Re-randomization) Test

Repeat this permutation procedure many, many times and draw a histogram of `mean(x_permute - y_permute)` values. This is the approximation of sampling distribution of $\bar{x}_d = \bar{x}_{amazon} - \bar{x}_{local}$.

Decision Rule: Reject null if the observed mean price difference in our original sample is at the extreme of that histogram



```
## [1] "p-value = 0"
```

Code to do this …

```
set.seed(1)
diff_permute = array(0, 10000)
for (counter in 1:10000){
  x_permute = x
  y_permute = y
  for (i in 1:10){
    if (sample(c('H', 'T'), 1) == 'H'){
      x_permute[i] = y[i]
      y_permute[i] = x[i]
    }
  }
  diff_permute[counter] = mean(x_permute - y_permute)
}
hist(diff_permute, breaks = 50, col = COL[1,1],
     xlab = 'mean difference in permuted book prices', freq=FALSE)
abline(v = mean(x-y), lwd = 2, col = 'red')
pval = mean(abs(diff_permute) > abs(mean(x-y))) #p-value
print(paste0('p-value = ', pval))
```

Code to do this using R function

- Check out R package `coin`

- Use with caution, make sure you understand what each argument of a function is really doing

Permutation Test for unpaired data

We could play the same permutation trick to get an approximation of null distribution of sample mean difference

However, the permutation scheme is going to be different (since no pairing anymore!)

Let us start with the mpg data

```
manual
```

```
##  [1] 21.0 21.0 22.8 32.4 30.4 33.9 27.3 26.0 30.4 15.8 19.7 15.0 21.4
```

```
automatic
```

```
##  [1] 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4 10.4 14.7
## [15] 21.5 15.5 15.2 13.3 19.2
```

```
n_man = length(manual); n_auto = length(automatic)
mean(manual) - mean(automatic)
```

```
## [1] 7.244939
```

Permutation Test for unpaired data

Since under null, there is no difference in the mpg distribution of manual and automatic, pool the data from two groups

```
mpg_pooled = c(manual, automatic)
mpg_pooled
```

```
##  [1] 21.0 21.0 22.8 32.4 30.4 33.9 27.3 26.0 30.4 15.8 19.7 15.0 21.4 21.4
## [15] 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4 10.4 14.7 21.5
## [29] 15.5 15.2 13.3 19.2
```

```
mpg_pooled_permuted = sample(mpg_pooled, size = 13+19, replace = FALSE)
mpg_pooled_permuted
```

```
##  [1] 17.3 16.4 15.8 14.3 10.4 22.8 15.2 14.7 30.4 17.8 21.0 19.7 30.4 21.0
## [15] 33.9 15.0 21.4 21.5 32.4 24.4 18.7 27.3 10.4 19.2 22.8 19.2 21.4 13.3
## [29] 18.1 15.5 26.0 15.2
```

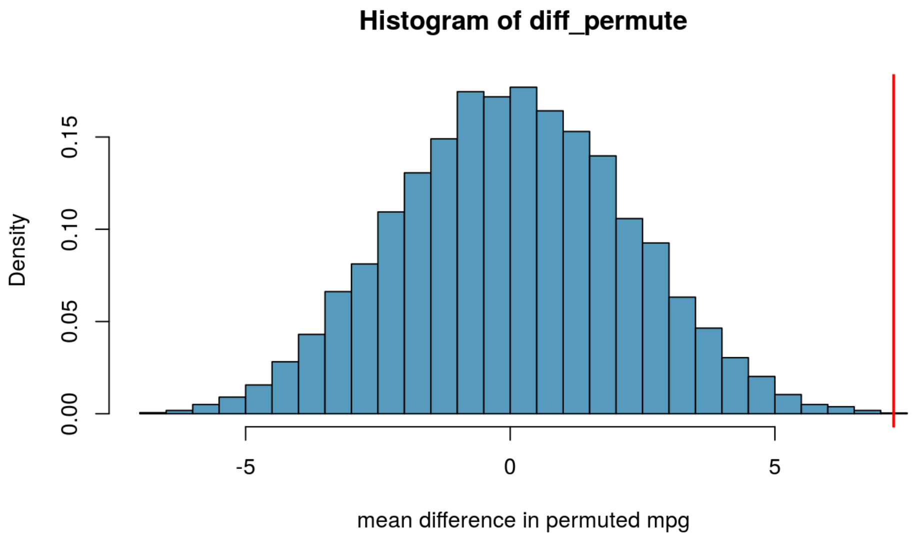
```
mpg_permuted_manual = mpg_pooled_permuted[1:13]
mpg_permuted_automatic = mpg_pooled_permuted[14:32]
mean(mpg_permuted_manual) - mean(mpg_permuted_automatic)
```

```
## [1] -1.940486
```

Permutation Test for unpaired data

Repeat such permutations many, many times and draw a histogram of mean mpg differences. This histogram approximates the sampling distribution of mean difference under null hypothesis.

Decision Rule: Reject null if the observed mean mpg difference in our original sample is at the extreme of that histogram



```
## [1] "p-value = 1e-04"
```

Code to do this using for loop

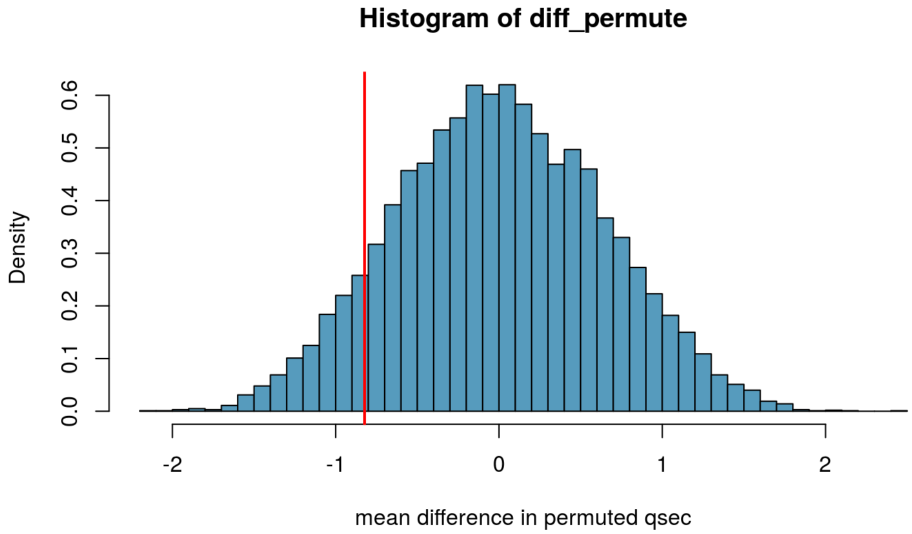
```
set.seed(1)
diff_permute = array(0, 10000)
for (counter in 1:10000){
  mpg_pooled = c(manual, automatic)
  mpg_pooled_permuted = sample(mpg_pooled, size = 13+19, replace = FALSE)

  mpg_permuted_manual = mpg_pooled_permuted[1:13]
  mpg_permuted_automatic = mpg_pooled_permuted[14:32]
  diff_permute[counter] = mean(mpg_permuted_manual) - mean(mpg_permuted_automatic)

}
hist(diff_permute, breaks = 50, col = COL[1,1],
     xlab = 'mean difference in permuted mpg', freq=FALSE)
abline(v = mean(manual) - mean(automatic), lwd = 2, col = 'red')
pval = mean(abs(diff_permute) > abs(mean(manual) - mean(automatic))) #p-value
print(paste0('p-value = ', pval))
```

Practice: compare qsec for manual and auto

```
automatic <- subset(mtcars, am == 0)$qsec
manual <- subset(mtcars, am == 1)$qsec
```



```
## [1] "p-value = 0.2083"
```

Code to do this using for loop

```r
set.seed(1)
diff_permute = array(0, 10000)
for (counter in 1:10000){
  qsec_pooled = c(manual, automatic)
  qsec_pooled_permuted = sample(qsec_pooled, size = 13+19, replace = FALSE)

  qsec_permuted_manual = qsec_pooled_permuted[1:13]
  qsec_permuted_automatic = qsec_pooled_permuted[14:32]
  diff_permute[counter] = mean(qsec_permuted_manual) - mean(qsec_permuted_automatic)

}
hist(diff_permute, breaks = 50, col = COL[1,1],
     xlab = 'mean difference in permuted qsec', freq=FALSE)
abline(v = mean(manual) - mean(automatic), lwd = 2, col = 'red')
pval = mean(abs(diff_permute) > abs(mean(manual) - mean(automatic))) #p-value
print(paste0('p-value = ', pval))
```