STAT1005 Foundations of Data Science

Lecture (7): Hypothesis testing & statistical decision

Yuanhua Huang (黃淵華)

Office: Rm 1-05E, 1/F, JCBIR, 5 Sassoon Road (Medical Campus)
Q&A contact hours: Wed 3-5pm

Email: yuanhua@hku.hk | Web: https://web.hku.hk/~yuanhua

25/10/2021

Department of Statistics & Actuarial Science 統計及精算學系 THE UNIVERSITY OF HONG KONG

HKU Med LKS Faculty of Medicine School of Biomedical Sciences 香港大學生物醫學學院

Objectives today

1. Hypothesis testing & random chance
2. Significance level and *p* value
3. Permutation test

4. Common testing methods: *t* test, ANOVA
5. Multiple testing & Types of errors
6. Power and sample size

7. Regression-based test

- scipy.stats: https://docs.scipy.org/doc/scipy/reference/stats.html
- statsmodels: https://www.statsmodels.org/stable/stats.html
- Notebooks: https://github.com/huangyh09/foundation-data-science/

Example | The lady tasting tea

- **Ronald Fisher**, *The Design of Experiments*, 1935.
- Wiki: https://en.wikipedia.org/wiki/Lady_tasting_tea
- Youtube: https://youtu.be/lgs7d5saFFc?t=13

Example | The lady tasting tea; random chance?

- Can the lady genuinely detect milk or tea first in the cup?
  - Experiment: 4 out of 8 cups with milk first.
  - Observation: the lady picked all cups correctly.

- Question: how likely this is purely by random chance?

- Formula of combination: pick 4 out of 8 cups, there are 70 combinations:
  - scipy.special.comb
  - Only one out 70 is full successes

```
In [1]: from scipy.special import comb

In [2]: comb(8, 4)
Out[2]: 70.0
```

- The chance we are fooled by randomness is $1/70 = 0.014$

Hypothesis testing | random chance to blame

- Purpose of hypothesis testing: help us learn whether random chance might be **responsible** for observations.
- *N.B.*, random chance is *random* but not always in *uniform* or *normal* distribution. The distribution sometimes can be complicated.

- Examples (decision to make & random chance to blame for the observation):
  - Can the lady genuinely **detect milk or tea first in the cup**? How much should we blame the observed data on random chance?
  - Can drug A **reduce the recovery time from Covid-19**? Can the observed difference between using and not using drug A is explained by random chance?
  - Is there a genuine **climate change**? Can the observed climate difference be explained by random chance?

Hypothesis testing | a statistical way for decision

- Null hypothesis (H_0):
  The hypothesis that random chance is to blame
- Alternative hypothesis (H_1 or H_a):
  Counterpart to the null; namely the hypothesis you want to prove

Example (A/B test: covid-19 recovery time by using or not using drug A):

- H_0: Drug A has no effect on Covid-19 recovery time, $\mu_A = \mu_B$
- H_1: $\mu_A \neq \mu_B$

Main idea of hypothesis testing

- It is difficult to prove that a fact (H_1) is "right".
- But it is easy to prove that an opposite fact (H_0) is "wrong".

Hypothesis testing | a statistical way for decision

- With null and alternative hypotheses set up, we then try to show that, in light of our collected data, the null hypothesis is false.

- In order to do so, we first need to define a suitable test statistic, e.g., mean, difference of A/B mean, difference of A/B median, variance of group mean

- Under the *null hypothesis*, we have a distribution of the defined statistic, e.g., by resampling or analytical form, named ***null distribution***.

- Then from the the null distribution we can calculate the probability of seeing the test statistic at least as extreme as the observed value, termed as *p* value

Hypothesis testing | *p* value

- *P* value: the probability of obtaining test results (i.e., predefined statistic) at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct.
  - If this probability is very small, it suggests that the null hypothesis is false.
  - If this probability is large, it suggests that there is not enough evidence to reject the null hypothesis.

- Intuition of *p* value: assume the null hypothesis is true, how surprising to see the observed data (in terms of the predefined statistic).

Hypothesis testing | *p* value; example (1)

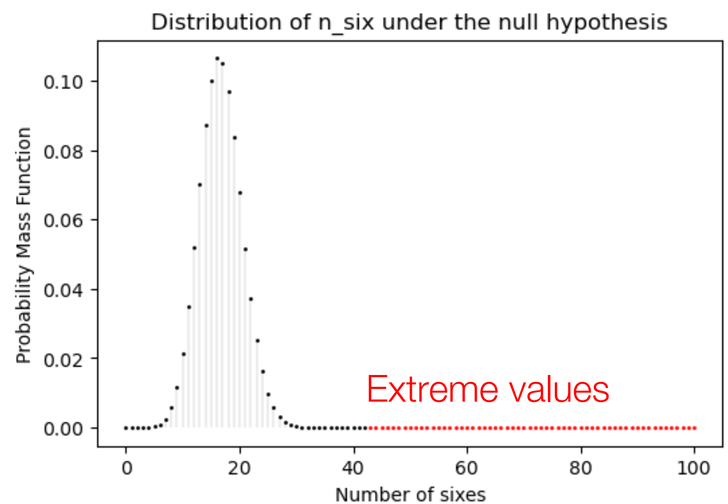
Example: whether a dice is equal

- H_0: probability of obtaining six $p = 1/6$;
- H_1: $p > 1/6$
- Data: n=100 observations, k=43 times of six

```
data = np.array([6, 1, 5, 6, 2,
6, 4, 3, 4, 6, 1, 2, 5, 6, 6, 3,
6, 2, 6, 4, 6, 2, 5, 4, 2, 3, 3,
6, 6, 1, 2, 5, 6, 4, 6, 2, 1, 3,
6, 5, 4, 5, 6, 3, 6, 6, 1, 4, 6,
6, 6, 6, 6, 2, 3, 1, 6, 4, 3, 6,
2, 4, 6, 6, 6, 5, 6, 2, 1, 6, 6,
4, 3, 6, 5, 6, 6, 2, 6, 3, 6, 6,
1, 4, 6, 4, 2, 6, 6, 5, 2, 6, 6,
4, 3, 1, 6, 6, 5, 5])
```

How to calculate p value? Binomial test:

- Test statistic: number of six;
- Null distribution: `Binomial(k; n=100, p=1/6)`
- Observed value: k=43
- *P* value: `1 - stats.binomial.cdf(k=42, n=100, p=1/6)` = 5.4e-10



Try it yourself!
Notebook: https://bit.ly/3GchiDv
CoLab: https://bit.ly/3vFL1Qc

Hypothesis testing | resampling for null distribution

- Recall: bootstrap for mimicking the true distribution of sample mean
- Resampling can be used to approximate the null distribution too.

- Define the test statistic: difference of group mean (can be other statistic)
- Generate null distribution, approximated by resampling
  - Step1: pooling samples in both groups A and B
  - Step2: permute (i.e., randomly shuffle) the pooled sample and split the pooled data into two groups with equal size to the original groups
  - Step3: calculate the test statistic (e.g., difference of group mean) and keep it
  - Step4: repeating steps 1 to 3 for R times (iterations)

- This method is call permutation test (default statistic: difference of group mean)

Hypothesis testing | *p* value; example (2)

Example: The birth weights of babies (in kg) is the same between two groups: heavy smoking (A) and non-smoking (B) mothers

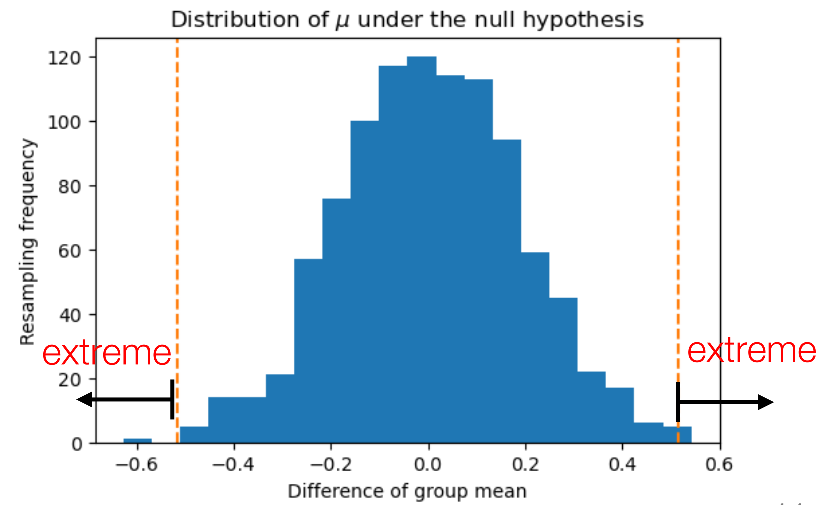
- H_0: no difference: $\mu = \mu_A - \mu_B = 0$;
- H_1: have difference: $\mu = \mu_A - \mu_B \neq 0$.

```
data_heavysmoking = np.array([
3.18, 2.84, 2.90, 3.27, 3.85,
3.52, 3.23, 2.76, 3.60, 3.75,
3.59, 3.63, 2.38, 2.34, 2.44])
data_nonsmoking = np.array([
3.99, 3.79, 3.60, 3.73, 3.21,
3.60, 4.08, 3.61, 3.83, 3.31,
4.13, 3.26, 3.54])
```

- Data: 15 instances for heavy smoking & 13 instances for non-smoking

How to calculate p value?

- Test statistic: difference of group mean;
- Null distribution: approximate by *resampling*
- Observed value: $\bar{x} = \bar{x}_A - \bar{x}_B = -0.52$
- *P* value: P(|X|>= |obs_val|) = 0.004



Hypothesis testing | two-tailed vs one-tailed

Two-tailed test: H0: $\mu = 0$, H1: $\mu \neq 0$

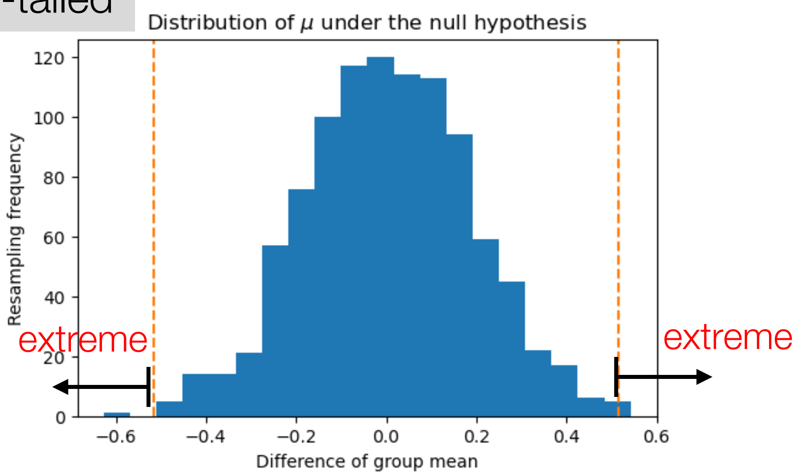
- The extreme value refers to both side
- P value: P(|X| >= |x_obs|)

One-tailed test: H0: $\mu = 0$, H1: $\mu < 0$ (or $\mu > 0$)

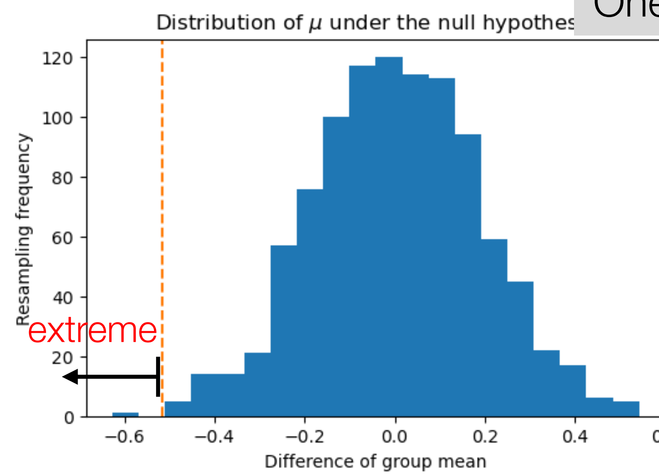
- The extreme value only refers to one specific side
- P value: P(X <= x_obs) for left side or P(X >= x_obs) for right side

How do you define extreme: one predefined side or either side?

Two-tailed



One-tailed



Hypothesis testing | permutation test, hands-on

Try it yourself (same link as before)!
Notebook: https://bit.ly/3GchiDv
CoLab: https://bit.ly/3vFL1Qc

Permutation test: null distribution approximated by resampling

```
[10] def get_permutation_null(x1, x2, n_permute=1000):
         """Simple function to generate permutation distribution
         """
         _n1, _n2 = len(x1), len(x2)
         x_pool = np.append(x1, x2)

         RV = np.zeros(n_permute)
         for i in range(n_permute):
             _x_perm = np.random.permutation(x_pool)
             RV[i] = _x_perm[:_n1].mean() - _x_perm[_n1:].mean()
         return RV
```

Hypothesis testing | significance level

- Statistical significance is how statisticians measure whether an experiment yields a result ***more extreme*** than what chance might produce.
- The significance level *α (Alpha)* is a **predefined** probability threshold of "unusualness" that chance results must surpass for actual outcome to be deemed statistically significant.
- We will reject H_0 when $p < \alpha$. From the definition of the p-value, α is the probability of incorrectly rejecting H_0 if it is true. By choosing a smaller α, we can specify a more conservative test.
- Example: if $\alpha = 0.05$, $p = 0.004 < \alpha$, reject H_0

Hypothesis testing | procedure

1. Propose a research question
2. Formulate the null hypothesis H_0 and alternative hypothesis H_1
3. Choose an **appropriate statistical test** (incl. test statistic, and its null distribution)
4. Choose an appropriate significance level, α
5. Calculate the test statistic
6. Calculate the p-value
7. Reject H_0 if $p < \alpha$, otherwise don't enough evidence to reject H_0

Commonly used analytical methods

Test Methods | resampling & analytical methods

- Resampling methods, like permutation test, are one-size-fits-all methods and becomes increasingly population partly thanks to better computing power

- Analytical methods (or formula approach), based on certain assumptions, are generally fast and accurate especially when the model assumption is not heavily violated.

Test methods | *t* test

Recall: when data follows normal distribution with unknown variance and sample size is small (8~29), distribution of sample mean can be approximated by *t* distribution; degree of freedom = n_instance – 1.

t test (independent samples):

- Test statistic: difference of group mean, $t = \bar{x}_A - \bar{x}_B = -0.52$
- Null distribution: approximated by t distribution
  - Mean = 0
  - Pooled standard deviation: $s_{\mathrm{p}} = \sqrt{\frac{(n_A-1)s_A^2+(n_B-1)s_B^2}{n_A+n_B-2}}$
  - Standard error of group mean difference: $\sigma = s_{\mathrm{p}}\sqrt{\frac{1}{n_A}+\frac{1}{n_B}}$
  - Degree of freedom: $n_A + n_B - 2$

s_A and s_B are unbiased estimate; divided by n_A-1 or n_B-1

https://en.wikipedia.org/wiki/Student%27s_t-test#Independent_two-sample_t-test

Test methods | *t* test; example

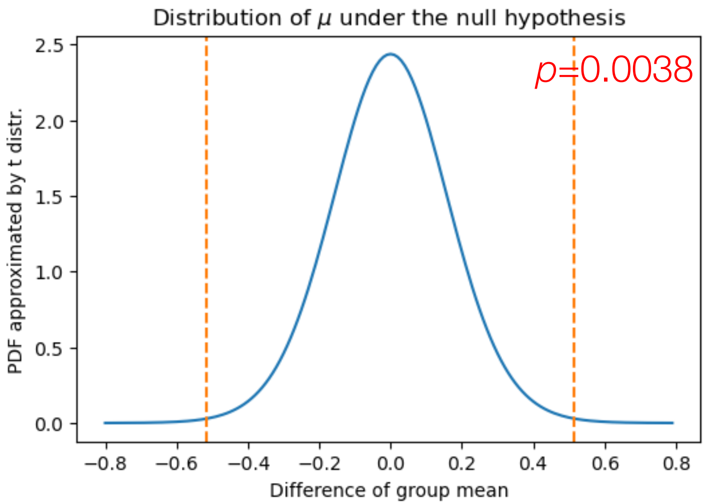
Example: The birth weights of babies (in kg) heavy smoking (A) and non-smoking (B):

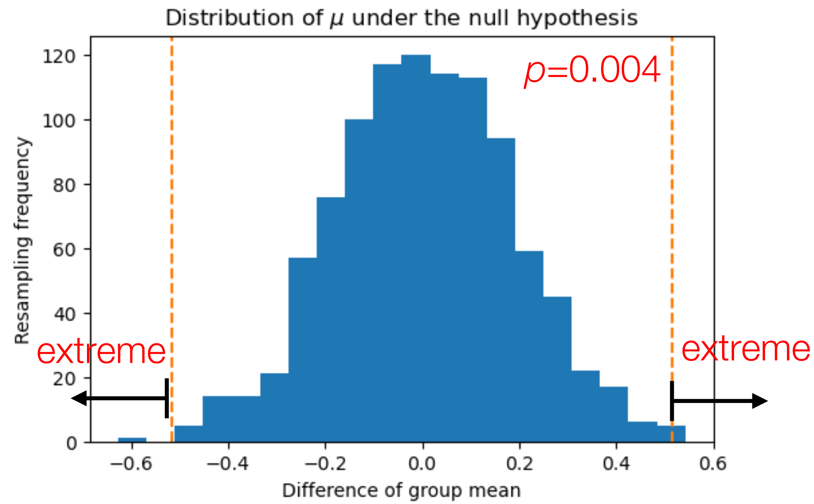
H_0: $\mu = \mu_A - \mu_B = 0$; H_1: $\mu = \mu_A - \mu_B \neq 0$.

Null distribution approximated by t distribution:

t(loc = 0, std=0.162, df=26)

```
data_heavysmoking = np.array([
3.18, 2.84, 2.90, 3.27, 3.85,
3.52, 3.23, 2.76, 3.60, 3.75,
3.59, 3.63, 2.38, 2.34, 2.44])
data_nonsmoking = np.array([
3.99, 3.79, 3.60, 3.73, 3.21,
3.60, 4.08, 3.61, 3.83, 3.31,
4.13, 3.26, 3.54])
```

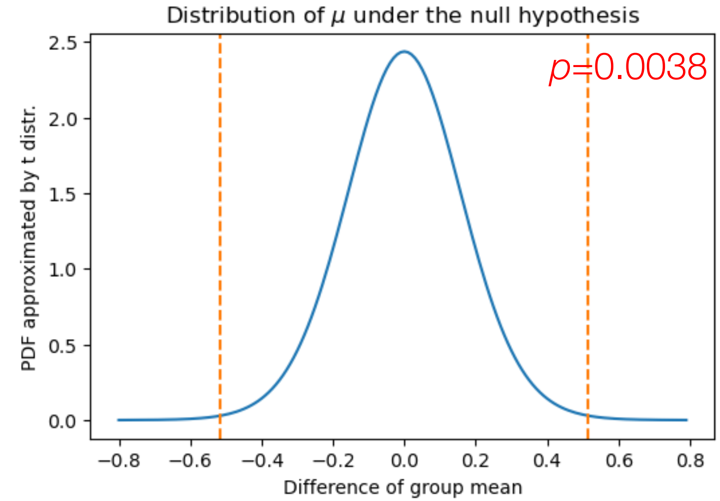


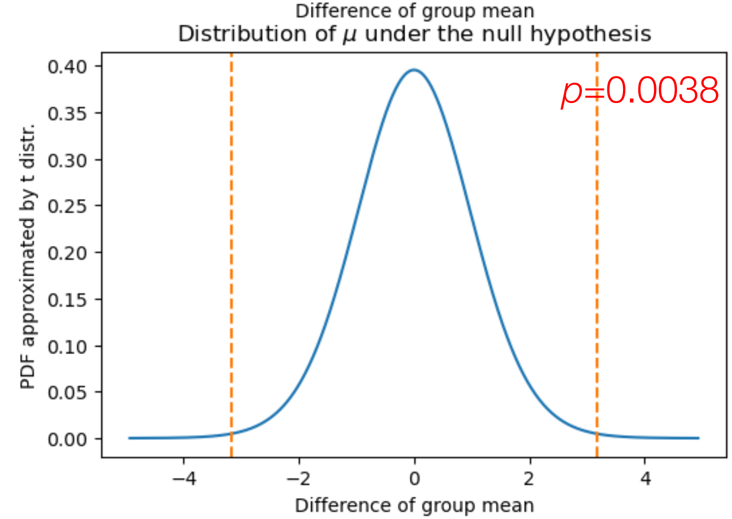


Test methods | t test, standardized form

t test (independent samples):

- Test statistic: $t = \frac{\bar{x}_A - \bar{x}_B}{\sigma}$
    - Pooled standard deviation: $s_\mathrm{p} = \sqrt{\frac{(n_A-1)s_A^2+(n_B-1)s_B^2}{n_A+n_B-2}}$
    - Standard error of group mean difference: $\sigma = s_\mathrm{p}\sqrt{\frac{1}{n_A}+\frac{1}{n_B}}$
- Null distribution: approximated by t distribution
    - Mean = 0
    - Standard error = 1
    - Degree of freedom: $n_A + n_B - 2$





Test methods | *t* test, hands-on

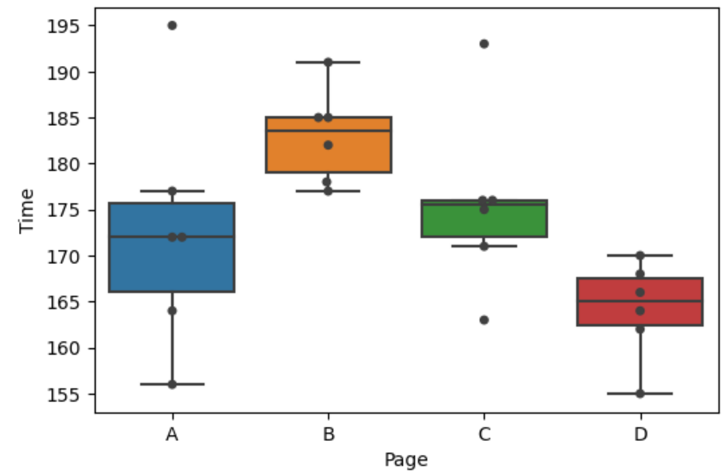
Try it yourself (same link as before)!
Notebook: https://bit.ly/3GchiDv
CoLab: https://bit.ly/3vFL1Qc

Permutation test: null distribution approximated by resampling

```
[10] def get_permutation_null(x1, x2, n_permute=1000):
         """Simple function to generate permutation distribution
         """
         _n1, _n2 = len(x1), len(x2)
         x_pool = np.append(x1, x2)

         RV = np.zeros(n_permute)
         for i in range(n_permute):
             _x_perm = np.random.permutation(x_pool)
             RV[i] = _x_perm[:_n1].mean() - _x_perm[_n1:].mean()
         return RV
```

Test Methods | ANOVA (concept)

- A/B test: each time two categories.
- What if *K*>2 groups, e.g., A/B/C/D?



Option 1: each time only use two groups

- There will be K*(K-1) / 2=6 comparisons
- Detect difference between any pair

Option 2: a joint comparison

- The cross-group variation is from random chance
- ANOVA: analysis of variance
- Use the statistic of variance between group means

Test Methods | ANOVA (concept)

ANOVA: analysis of variance

- Whether the variance of group means is explained by random chance.

Method 1: Resampling methods

- Test statistic: the variance of group means
- Null distribution:
  - Step1: Pool all samples
  - Step2: Permute the pooled samples and divide them into groups with the same size to the original groups
  - Step3: calculate the test statistic and record it
  - Step4: repeat steps 1 to 3 for many times (iterations)

Method 2: Analytical method (*F* statistic; null distribution *F* distribution):

$$F = \frac{\text{between-group variability}}{\text{within-group variability}}$$

https://en.wikipedia.org/wiki/F-test#Multiple-comparison_ANOVA_problems

Evaluation of testing methods & power analysis

Multiple testing | null distribution of p value

- Testing if a gene expression changes between with and without treatment
  - 30 Covid-19 patients, half with drug A and half without drug
  - There are 10,000 genes to test, namely 10,000 hypothesis to perform

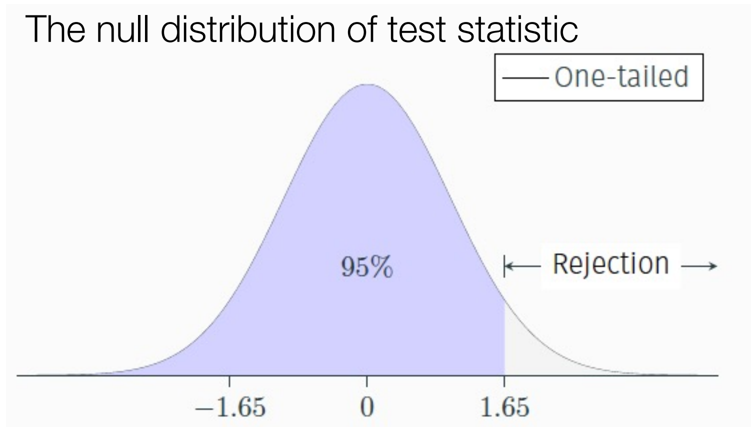
Question 1: if treatment does not make any difference to any of these genes, what would be the lowest *p* value? 1, 0.5, 0.1, or 0.0001

Question 2: What is the distribution of *p* value if the null model is true?

- a) p value follows the same as the distribution as the test statistic
- b) p value is always 1.
- c) p value follows a uniform distribution between 0 and 1.

Multiple testing | null distribution of p value

Question: What is the distribution of *p* value if the null model is true?



Cumulative distribution function of p value:

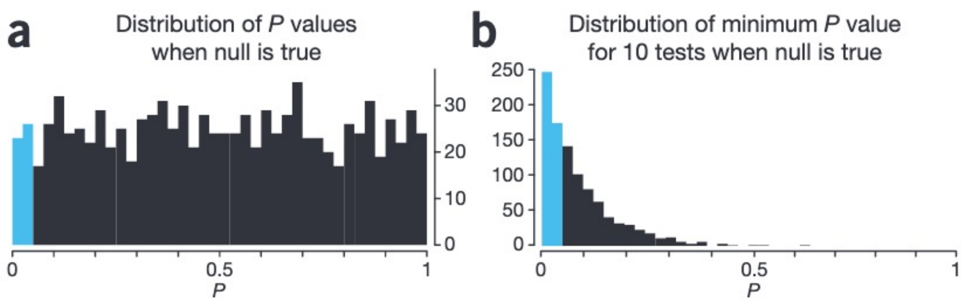
$$P(X<p) = p$$

Exactly as uniform distribution, no matter what the null distribution is.

Multiple testing | correction of p value

- What is the distribution of *p* value if the null model is true?
  - Under the null, the chance we see p value < 0.05 is 5%
  - By performing 10 times, the chance to have the lowest p value < 0.05 is 40%

- Multiple testing correction
  - None perfect methods, but some are practically useful
  - Benjamini-Hochberg correction, namely, False Discovery Rate (FDR) is commonly used



FDR: For a given FDR α, find the largest k that the kth $P_k < \frac{k}{n\_test}\alpha$

Require large sample size or do smaller number of tests

Altman & Krzywinski. P values and the search for significance. Nature methods 2017

Multiple testing | hands-on

Try it yourself!
Notebook: https://bit.ly/3pvc53L
CoLab: https://bit.ly/3EscEQb

Multiple test

Hypothetic null distribution. Feel feel to try any null distribution, examples below

```
## Example null distributions

# any_null_dist = stats.t(df=26, loc=0, scale=1)
# any_null_dist = stats.norm(loc=0.5, scale=3)

any_null_dist = stats.chi2(df=3, loc=0, scale=1)
```

Performance of testing | Types of errors

- Testing if a gene expression changes between with and without treatment
  - 30 Covid-19 patients, half with drug A and half without drug
  - There are 10,000 genes to test, namely 10,000 hypothesis to perform

- What errors in each of these 10,000 decisions?
  - False positive (type I error): Genes are **genuine not different**, but we thought they are (reject the null hypothesis)
  - False negative (type II error): Genes are **genuine different**, but we missed it (we didn't reject the null hypothesis)

- Type I error is generally more concerning, as we worried more on being fooled by random chance.

Performance of testing | Evaluation metrics

- True positive rate (Power, Sensitivity, Hit rate, Recall): $TPR = \frac{TP}{TP+FN}$
- True negative rate (Specificity): $TNR = \frac{TN}{TN+FP}$
- Precision (Positive Predictive Value; 1- false discovery rate):

$$Precision = \frac{TP}{TP + FP} = 1 - FDR$$

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection

Power analysis | sample size and power

- Power (sensitivity, recall, hit rate, true positive rate):

  Power = TPR = TP / (TP + FN) = TP / P

  Power = 1 – Type II error

- Power analysis answers questions like "how much statistical power does my study have?" and "how big a sample size do I need?".

- Relationship between power and other factors:
  - Significance level (p value threshold) increase → power increase (detect more)
  - Effect size increases → observed p value decreases → power increase
  - Sample size increases → standard error decrease → observed p value decreases → power increase

Power analysis | sample size and power

Relationship between four factors:

- Sample size
- Effect size (normalized to standard deviation) we want to detect
- Significance level (p value threshold)
- Power

- When knowing three of them, the remaining one can be estimated.

- “Power analyses are normally run before a study is conducted. A prospective or a priori power analysis can be used to estimate any one of the four power parameters but is most often used to estimate required sample sizes.”

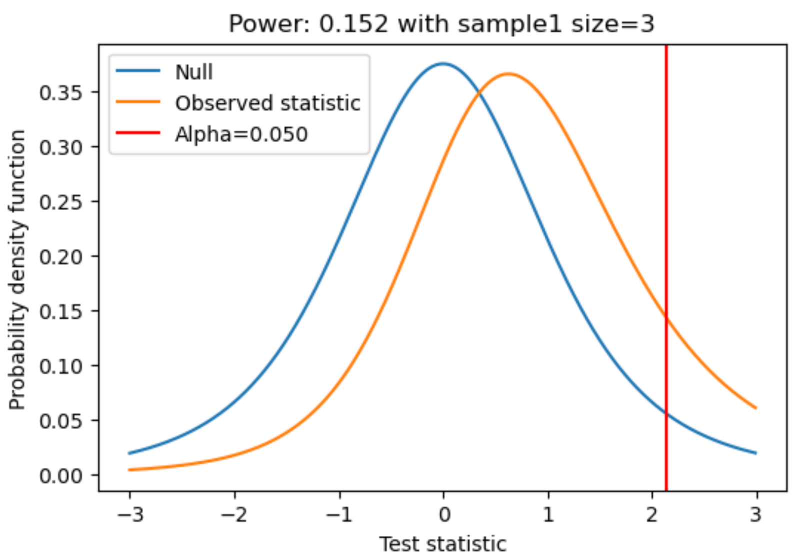
https://machinelearningmastery.com/statistical-power-and-power-analysis-in-python/

Power analysis | sample size and power

Relationship between four factors:

- Sample size (each group): 3
- Effect size to detect: 0.6
- Significance level (p value threshold): 5%
- Power calculation

t statistic = 0.6 / sqrt(2 / 3) = 0.734



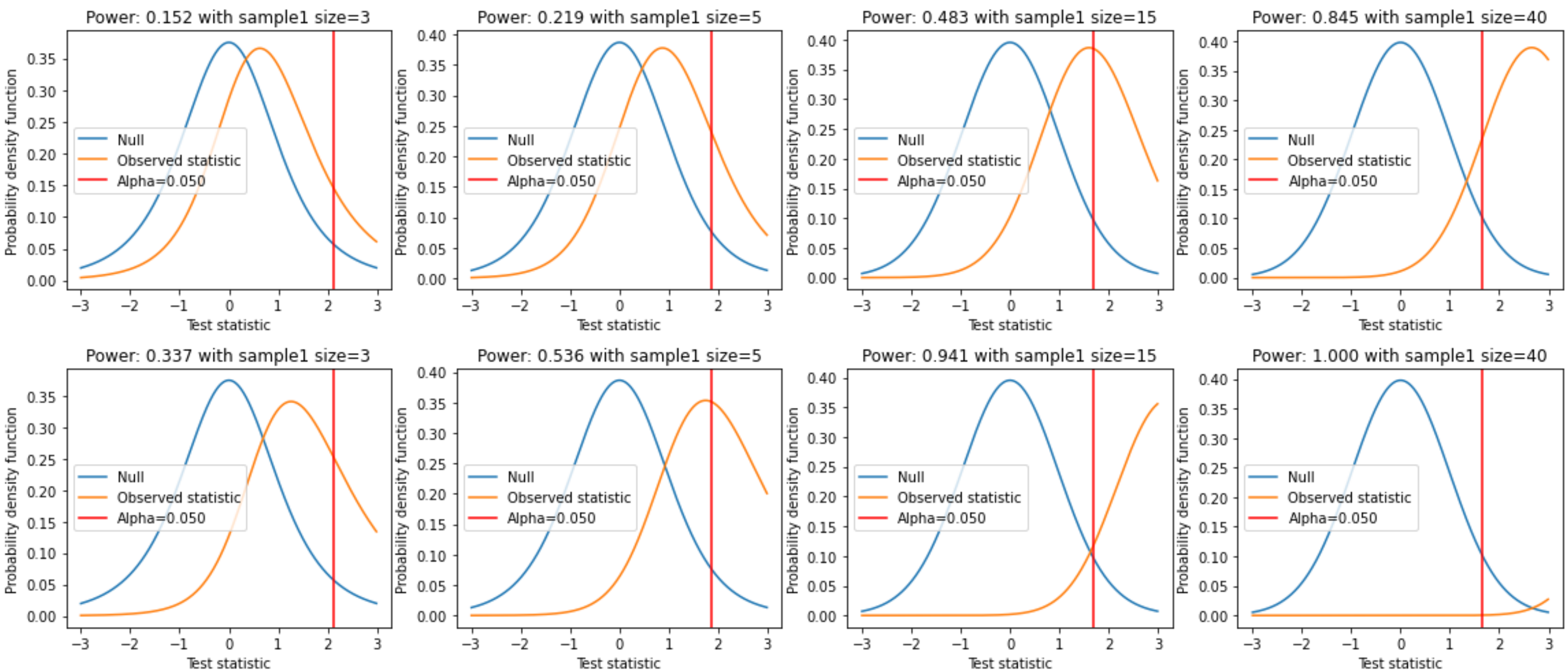

```
# perform power analysis
from statsmodels.stats.power import TTestIndPower

analysis = TTestIndPower()
analysis.power(effect_size = 0.6, nobs1=3,
               alpha=0.05, alternative='larger’)

[out] 0.15213899208943416
```

Power analysis | sample size and power

Relationship between four factors (alpha=0.05):

- Varying: sample size & effect size to detect

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$



Power analysis | required samples size, *t*-test

How many samples do we need to detect smaller effect size?

- Effect size to detect: 0.1 kg / 0.162 = 0.617 (baby birth weight difference, normalized)
- Significance level: 0.05
- Power: 80%

```
# perform power analysis
from statsmodels.stats.power import TTestIndPower

analysis = TTestIndPower()
result = analysis.solve_power(effect_size=0.617,
power=0.8, nobs1=None, alpha=0.05, alternative='larger')
```

Results: 46 samples are needed for each group

https://www.statsmodels.org/dev/generated/statsmodels.stats.power.TTestIndPower.html

Power analysis | hands-on

Try it yourself (same link as before)!
Notebook: https://bit.ly/3pvc53L
CoLab: https://bit.ly/3EscEQb

Power analysis

```
from statsmodels.stats.power import TTestIndPower
```

```
# parameters for power analysis

# population standard deviation
# pop_std = 0.162

standard_effect = 0.1 / 0.162
# standard_effect = 0.52 / 0.162

alpha = 0.05
power = 0.9

# perform power analysis
analysis = TTestIndPower()
result = analysis.solve_power(effect_size = standard_effect,
                              power=power, nobs1=None,
                              alpha=alpha, alternative='larger')
```

Regression-based testing

Regression-based testing | formula

Example: whether advertising on news papers increase sales of houses.

Research hypothesis (alternative hypothesis)

- H_1: the newspaper adverting has impact on sales

$$H_1: y = \beta_0 + \beta_1 \times \text{Newspaper}; \ \beta_1 \neq 0$$

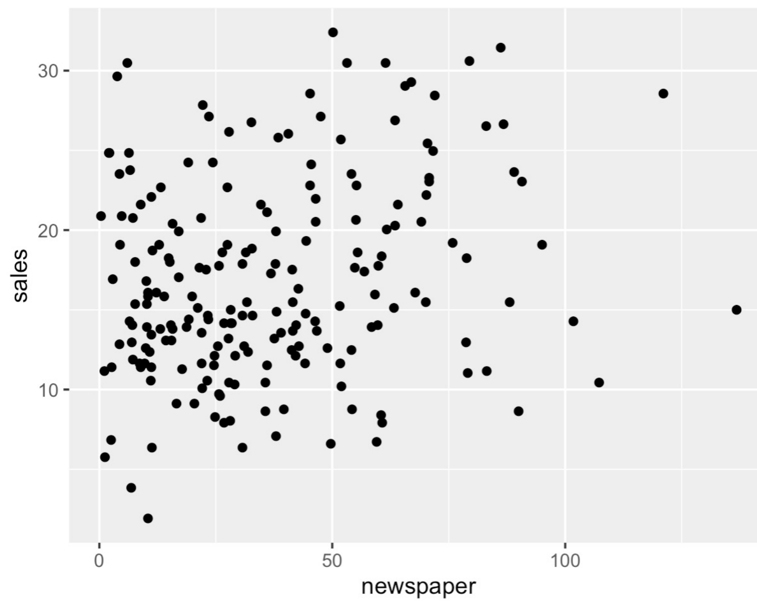
Null hypothesis (default hypothesis, you don’t need to prove it, just assume it)

- H_0: the newspaper adverting has no impact on sales

$$H_0: y = \beta_0 + \beta_1 \times \text{Newspaper}; \ \beta_1 = 0$$

Regression-based testing | example

- Data collection
  - 200 samples with both newspaper advertising costs and sales of cars



Dataset: https://search.r-project.org/CRAN/refmans/datarium/html/marketing.html
https://github.com/huangyh09/foundation-data-science/blob/main/w8-hypothesis-testing/marketing.csv

Regression-based testing | model fitting

- Likelihood: describes the joint probability of the observed data as a function of the parameters of the chosen statistical model.

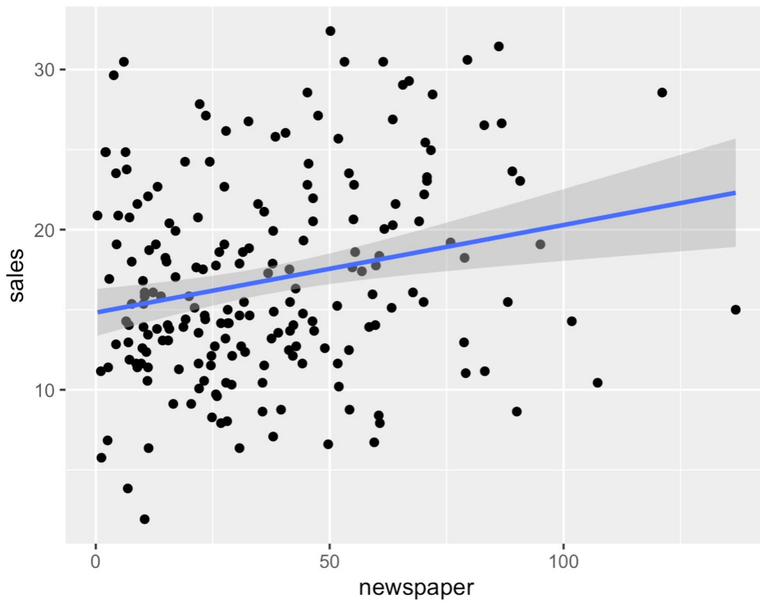
- Here, we assume y follows a normal distribution condition on features
$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

- Likelihood:
$$L(\beta_0, \beta_1, \sigma) = \prod_{i=1}^{n} P(y_i | \beta_0 + \beta_1 x_i, \sigma^2)$$

- Optimization: we can find a set of value for $(\beta_0, \beta_1, \sigma)$, to maximize the likelihood, namely obtain a maximum-likelihood estimate. Their standard error can also be approximated by through the likelihood function.

Regression-based testing | model fitting

- Fitting a regression model with maximum likelihood
  - $y = \beta_0 + \beta_1 \times \text{Newspaper};$

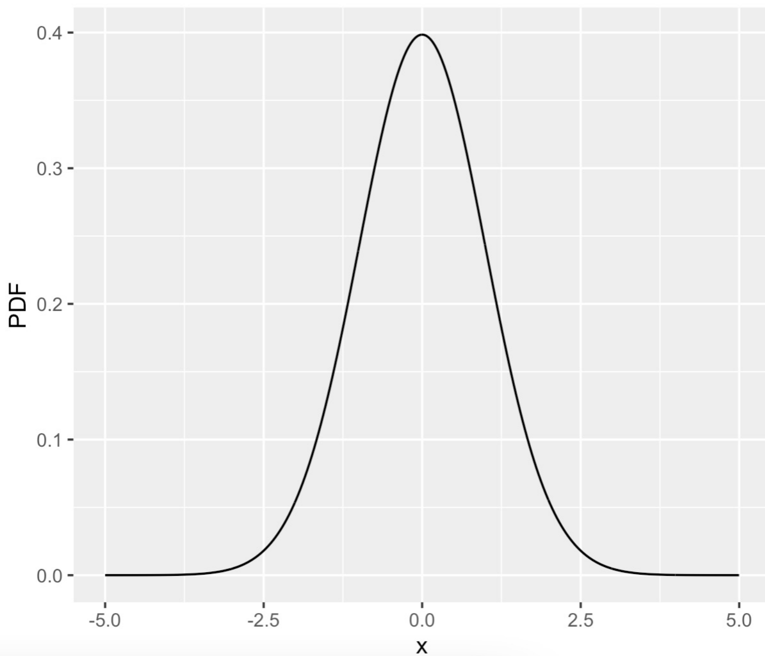


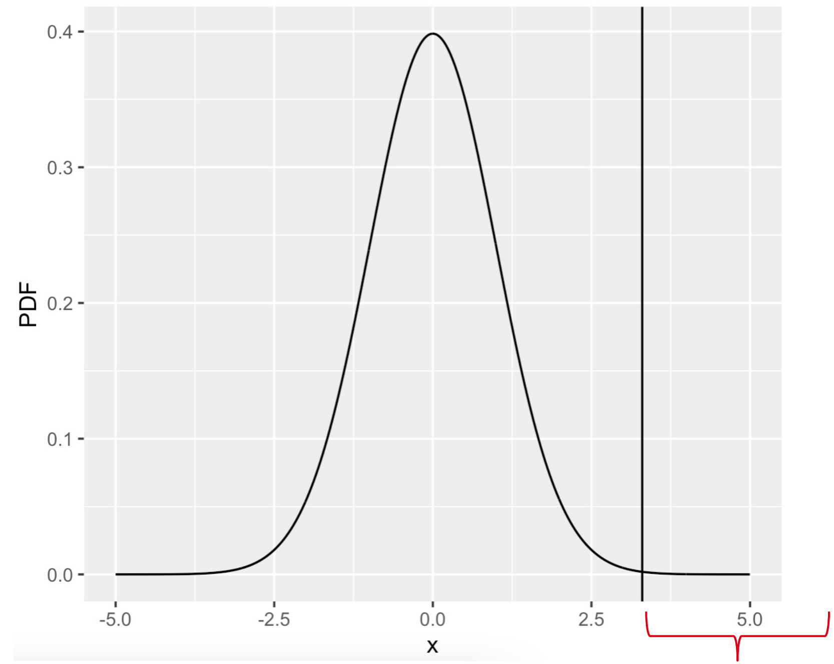
Maximum likelihood estimate:
mean and standard error

Intercept: $\beta_0 = 14.82 \pm 0.746$
Newspaper: $\beta_1 = 0.0547 \pm 0.0166$

T-statistic for β_1:
t value = 0.0547 / 0.0166 = 3.3

Regression-based testing | *t* statistic (Wald test)





Under the null, the distribution of ***t***-statistic;
Degree of freedom = n_sample – n_coefficient = 198

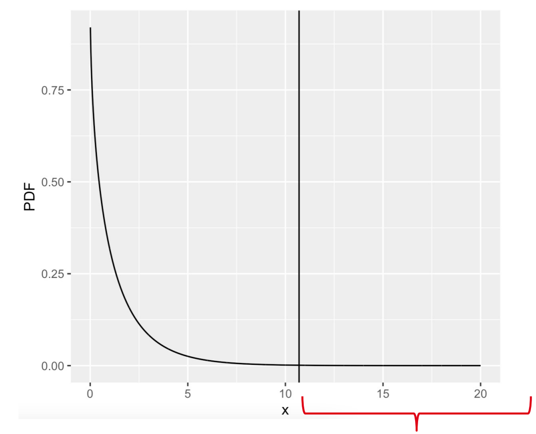
P value = prob(x > *t* value) * 2 = 0.00115
Reject null hypothesis at significance level of 0.01

Regression-based testing | likelihood ratio test

- Likelihood ratio test
  - Null model log likelihood L_0: $y = \beta_0$
  - Alternative model log likelihood L_1: $y = \beta_0 + \beta_1 \times \text{Newspaper}$

- Likelihood with maximum likelihood estimate
  - Null hypothesis: $L_0 = -650.15$
  - Alternative hypothesis: $L_1 = -644.8$

- Likelihood ratio statistic
  - Observed results: $\lambda = -2(L_0 - L_1) = 10.7$
  - Distribution under the Null: $\lambda \sim \chi^2\ (df = 1)$
  - P value: $P(x > \lambda) = 0.00107$



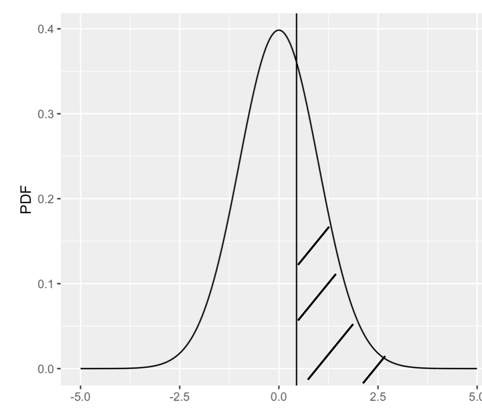
Regression-based testing | additional covariates

- Condition on other covariate, e.g., advertising on Facebook
  - $H_1: y = \beta_0 + \beta_1 \times \text{Newspaper} + \beta_2 \times \text{Facebook}\ ;\ \beta_1 \neq 0$
  - $H_0: y = \beta_0 + \beta_1 \times \text{Newspaper} + \beta_2 \times \text{Facebook}\ ;\ \beta_1 = 0$

	youtube <dbl>	facebook <dbl>	newspaper <dbl>	sales <dbl>
1	276.12	45.36	83.04	26.52
2	53.40	47.16	54.12	12.48
3	20.64	55.08	83.16	11.16
4	181.80	49.56	70.20	22.20
5	216.96	12.96	70.08	15.48
6	10.44	58.68	90.00	8.64

- Fitting the model with collected data
  - $\beta_0 = 11.02 \pm 0.753$
  - $\beta_1 = 0.0066 \pm 0.0149$; t value = 0.0066/0.0149 = 0.446
  - $\beta_2 = 0.199 \pm 0.022$

- P value = 0.656; fail to reject the null hypothesis at significance level of 0.05.



Regression-based testing | hands-on

Try it yourself!
Notebook: https://bit.ly/3jyrIDs
CoLab: https://bit.ly/3pE9Fjr

Wald test (t test on coefficient)

```
# Fit and summarize OLS model
Y = df['sales']
X0 = df[['constant']]
X1 = df[['constant', 'newspaper']]

mod1 = sm.OLS(Y, X1)
res1 = mod1.fit()
```

```
print(res1.summary())
```

Summary

- Hypothesis testing (Null vs alternative hypothesis):
  - Is the observed statistic (data) likely generated just by random chance?
  - Null distribution (approximated by resampling or analytical methods)
  - P value: the probability to see at least as extreme statistic under the null
- Evaluation:
  - Multiple testing: distribution of p values under the null
  - Type I and type II errors
  - Power (sensitivity, recall, True positive rate), its relation to sample size, effect size to detect, and significance level.
- Regression-based test:
  - Estimate parameters (alternative hypothesis, mean and standard error)
  - T-test (Wald test) & Likelihood ratio test
  - Condition on additional covariates

Resources & Acknowledgement

- IPython Notebook for this lecture note:
  - On Moodle
  - Also: https://github.com/huangyh09/foundation-data-science/

Other reference resources with acknowledgement:

- Chapter 3, Bruces & Gedeck, Practical Statistics for Data Science
- Imperial College course: Introduction to Sampling & Hypothesis Testing (by Dr John Pinney) https://github.com/johnpinney/sampling_and_hypothesis_testing
- Chapters 9 & 10, Introductory Statistics: https://opentextbc.ca/introbusinessstatopenstax/