

UCI Statistics Workshop for RNA Club

January 28, 2022

Zhaoxia Yu

Department of Statistics, UCI

zhaoxia@ics.uci.edu

Outline

- A brief intro to R
- Statistical inference
- Which test?
- Two useful nonparametric methods: bootstrap and permutation
- Power calculation and sample size
- Multiple comparisons
- Beyond basic methods
- Visualization
- Future topics

A Brief Intro to R

- S: open-source 1976:
created Bell Labs

S-Plus: commercial

- 1988: founded and owned
by a faculty member of UW
- ...
- 2008: acquired by TIBCO



R: open-source

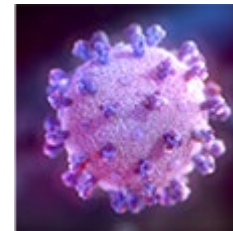
- 1991: Ross Ihaka and Robert Gentleman at the University of Auckland
- 1997: The Comprehensive R Archive Network (CRAN) was officially announced
- Over 10000+ packages. Examples
 - 2001: bioconductor
 - 2005: ggplot2 package released

A Brief Intro to R

- Install R. You can choose either R or R Studio
 - <https://cran.r-project.org/>
- Install R packages.
 - <https://www.r-bloggers.com/2010/11/installing-r-packages/>
- reading and importing data into R
 - <https://www.r-bloggers.com/2015/04/r-tutorial-on-reading-and-importing-excel-files-into-r/>

Statistical Inference: Estimation

- Example: Novovax vaccine
 - Vaccine efficacy: 90.4%;
 - 95% confidence interval: [82.9, 94.6], $P < 0.001$
- https://www.nejm.org/doi/full/10.1056/NEJMoa2116185?query=featured_home



- <https://apps.who.int/iris/bitstream/handle/10665/264550/PMC2491112.pdf>
- https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_confidence_intervals/bs704_confidence_intervals8.html

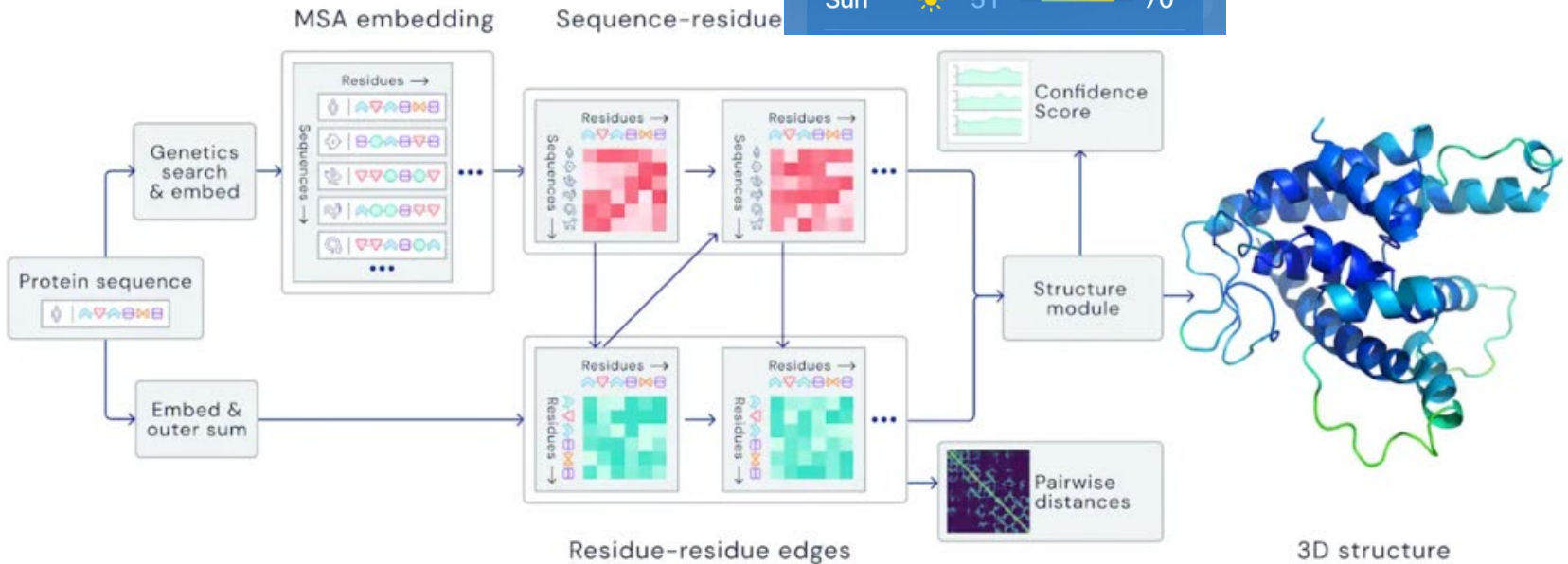
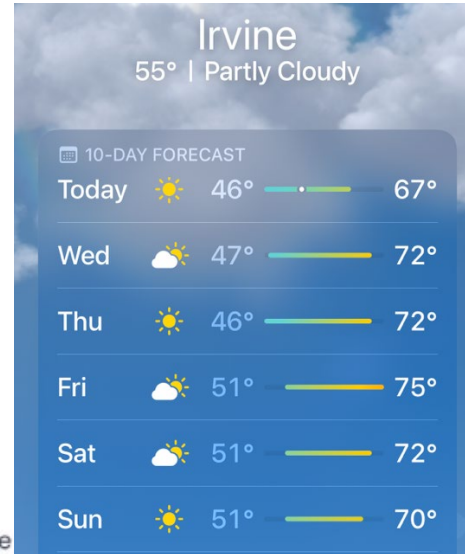
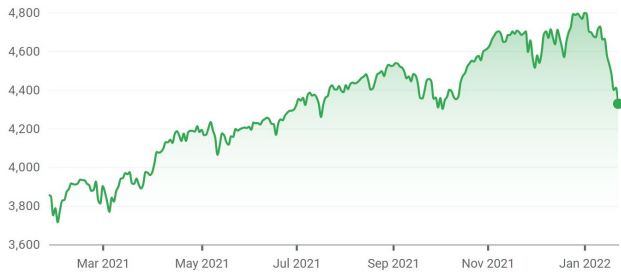
Statistical Inference: Prediction

S&P 500

4,329.32 ↑ 12.29% +473.96 1Y

Jan 25, 12:38:21 PM UTC-5 · INDEXSP · Disclaimer

1D 5D 1M 6M YTD 1Y 5Y MAX



Statistical Inference: Hypothesis Testing

- Which test?

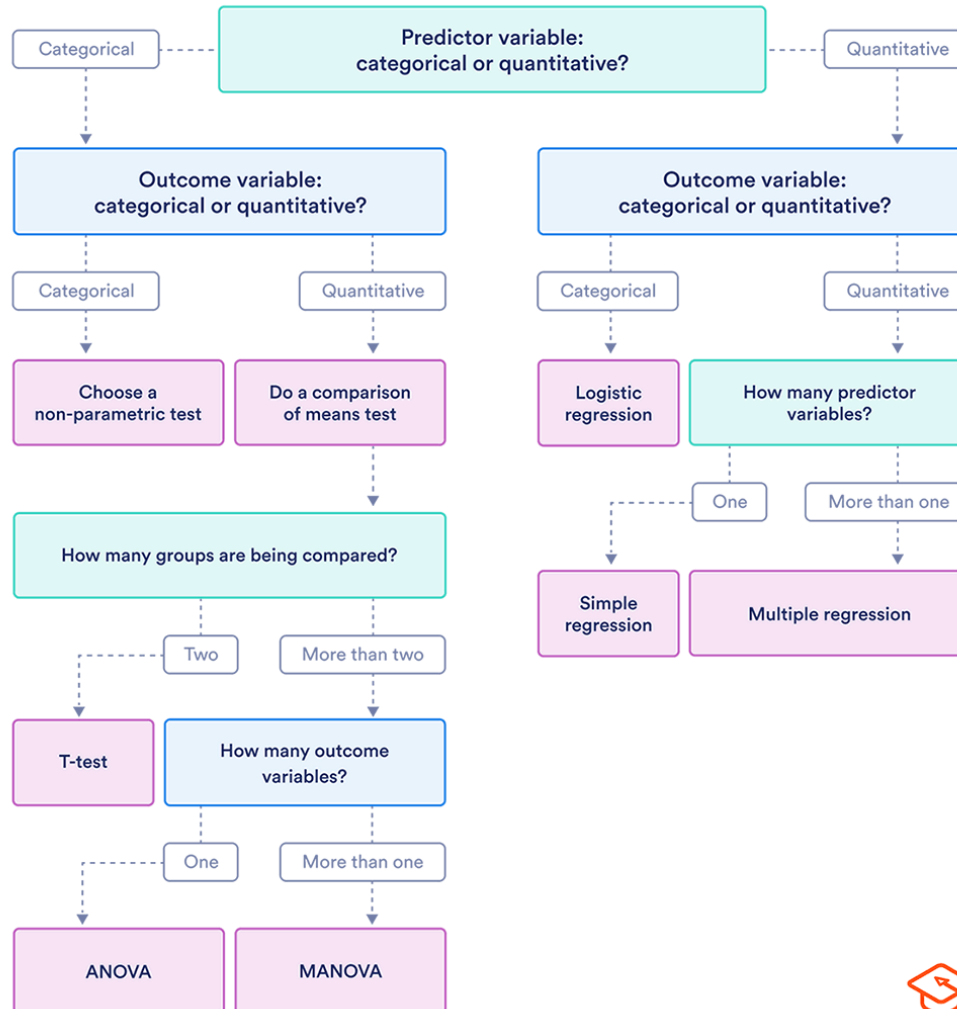
Types of Cheese



- Which model?

Choosing a statistical test

This flowchart helps you choose among parametric tests



Which test?

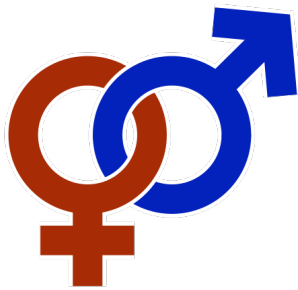
- So many tests. For example,
 - Parametric: one-sample t-test, two-sample t-test, ANOVA
 - Nonparametric: Wilcoxon signed-rank test, Mann-Whitney U-test, Kruska-Wallis test, permutation tests,
 - Chi-squared vs Fisher's exact test
 - Other considerations: one-sided vs two-sided, multiple comparisons
- How to decide?
 - Scientific question (associated? Greater? Less?)
 - Experimental design (independent?)
 - Nature of data (continuous? sample size? normal?)



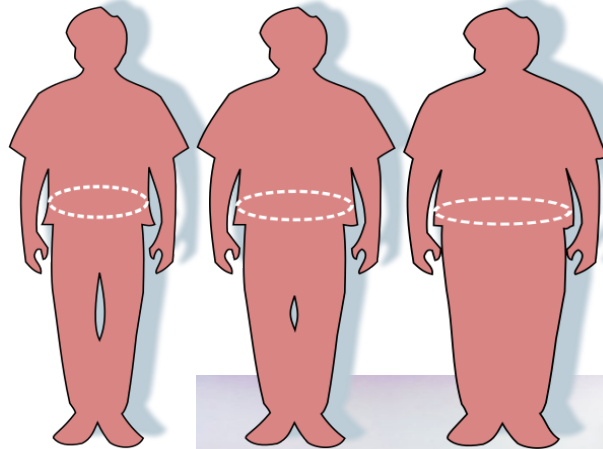
A case study: gender and obesity

Case study: Obesity and Gender

- Q1. Are men and women similar in obesity?
- Q2. Do women tend to be more obese than men
- Q3. Is obesity associated with gender?



wikipedia



Case study: Obesity and Gender

- **Q1. Are men and women similar in obesity?**
- For simplicity, we focus on a relatively homogenous population, such as all adults in US
- What is your choice of response variable?
 - Continuous: BMI
 - Categorical:
 - Binary: Obesity (BMI>30) vs non-obesity
 - Weight status: underweight (<18.5), normal (18.5-24.9), overweight (25-29.9), obesity (>30)
- Study design:
 - Are observations independent?
 - Is the sample size large enough?

Case study: Obesity and Gender

- Q1. Are men and women similar in obesity?

	Continuous measurements (BMI)	Binary measurement (obesity)
Independent observations	Two-sample t-test or <u>Wilcoxon signed-rank test</u>	Chi-squared test or <u>Fisher's exact test</u>
Couples (data are paired)	Paired t-test (eqt one-sample t-test), <u>Wilcoxon signed-rank test</u>	Mc-Nemar's test or <u>binomial sign test</u>

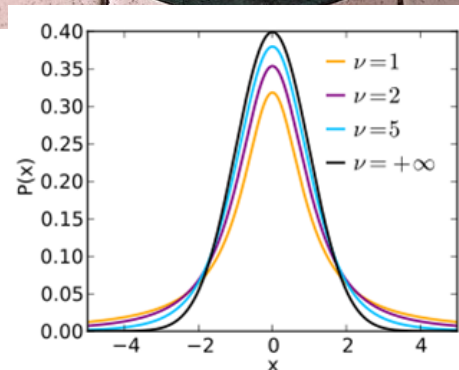
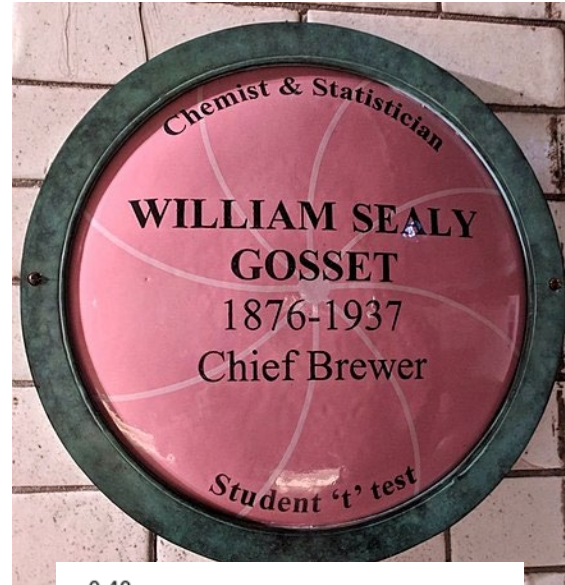
Note: nonparametric or exact tests are underlined. They are recommended for small sample sizes.

Case study: Obesity and Gender

- Q2. Do women tend to be more obese than men
 - One-sided/tailed vs two-sided/tailed
- Q3. Is obesity associated with gender?
 - The answer to question 1 provides partial information
 - For observational studies, we prefer to prevent spurious association as much as we can by accounting for confounding factors. As a result, regression is preferred
 - Continuous responses: linear regression. Transformation will be conducted if necessary
 - Other type of responses: generalized linear regression such as logistic regression

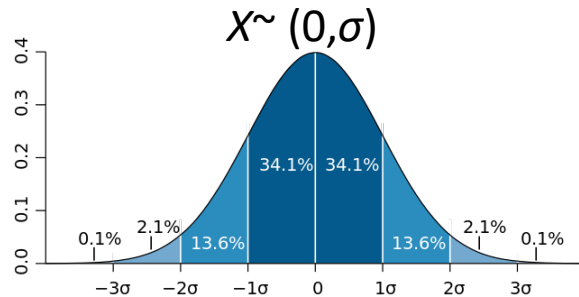
One-sample t-test

- **Student's** t-distribution
- It was derived in late 19th century
- It gets its name from a British brewer who used "**Student**" as his pen-name (1908)
 - Gosset developed the t-test to test the quality of stout



One-sample t-test

- Standard deviation vs standard error



$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

iid

iid:
independent and
identically **d**istributed

$$s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

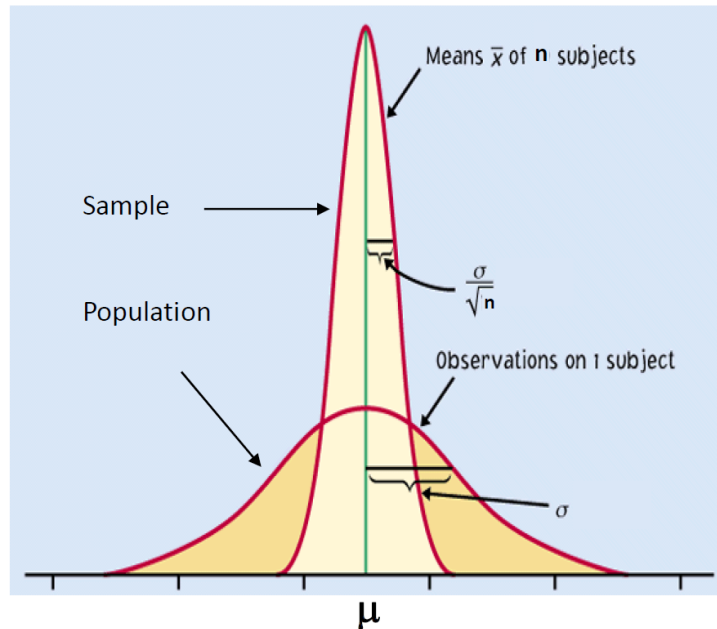
- Population characteristics (often unknown):
 - Population mean: μ , which is 0 in this example.
 - Variance: σ^2 . Its square root, i.e. σ , is called the standard deviation (SD).
- Sample characteristics
 - Sample mean: $\bar{x} = (x_1 + \dots + x_n)/n$
 - Sample variance: $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
- $\hat{\mu} = \bar{x}, \hat{\sigma}^2 = s^2$
- Variance of the sample mean: $var(\bar{x}) = \frac{\sigma^2}{n}$. $\widehat{var}(\bar{x}) = \frac{\hat{\sigma}^2}{n} = \frac{s^2}{n}$
- Standard error (SE) of the sample mean: $se(\bar{x}) = \sqrt{\widehat{var}(\bar{x})} = \frac{s}{\sqrt{n}}$

One-sample t-test

- Consider a population with mean μ and standard deviation σ .

- **Fact1:** The mean and standard deviation of the sampling distribution of \bar{X} is μ and σ/\sqrt{n} , respectively.
 - The sample mean is an unbiased estimator of the population mean
 - σ/\sqrt{n} measures how the sample mean varies from sample to sample

- **Fact2:** If the distribution of the population is Normal,
$$\bar{X} \sim N(\mu, \sigma^2/n)$$



One-sample t-test

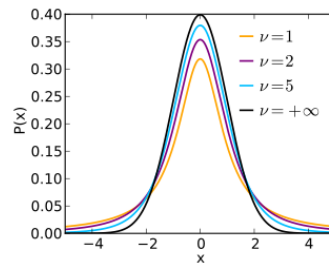
- What is t-test?
 - Suppose $H_0: \mu = \mu_0$ (very often $\mu_0=0$)
 - What does $(\bar{x} - \mu_0)$ tell you? If we change the unit of the measurement, the value will change!
 - A better quantity/statistic is the “standardized” version:

$$t = \frac{\bar{x} - \mu_0}{se(\bar{x})} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

- How to use the t-statistic? We compare an observed t value to a reference distribution, i.e., the distribution under H_0 .

One-sample t-test

- The **NULL** distribution (when H_0 is true)
 - Under the assumption of iid and normality, we can derive the distribution of the t-statistic under the null hypothesis. The distribution is known as t-distribution with $(n-1)$ degrees of freedom



- If normality does not hold, which is likely true, for large sample sizes, the t-distribution is still a good approximation

One-sample t-test

- One-sided/tailed vs two-sided/tailed.

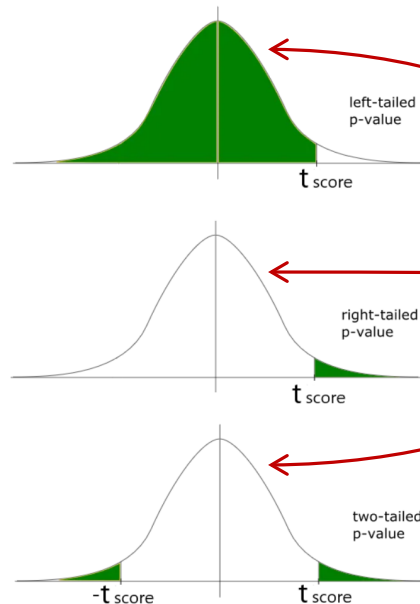
THE ONE-SAMPLE t TEST

Draw an SRS of size n from a large population having unknown mean μ . To test the hypothesis $H_0: \mu = \mu_0$, compute the one-sample t statistic

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

In terms of a variable T having the $t(n-1)$ distribution, the P -value for a test of H_0 against

μ_0 : parameter value under H_0
 \bar{x} : sample mean
 s : sample standard deviation
 n : sample size.



pdf of t_{n-1}

One-sample t-test

- One-sided/tailed vs two-sided/tailed.
- E.g., $n=10$, $t=2.4$, $H_0: \mu = \mu_0$

omni CALCULATOR

Test setup

Choose test type: [one-sample](#)

t-test for the population mean, μ , based on one independent sample.

Null hypothesis $H_0: \mu = \mu_0$

Alternative hypothesis $H_1: \mu \neq \mu_0$

Test details

Approach: [p-value](#)

Do you know the t-score? [Yes](#)

t-score: 2.4

Degrees of freedom: 9

Test results

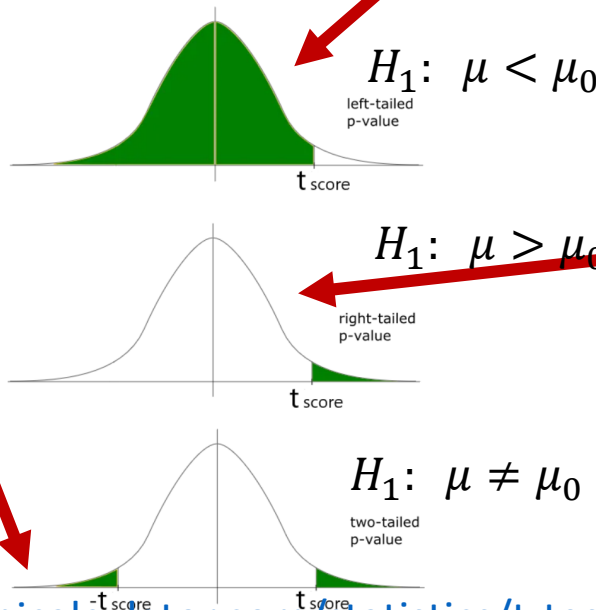
p-value: 0.03989788

Decision:

You can reject H_0 at the significance level 0.05, because your p-value does not exceed 0.05.

p-value from t-test

Recall that the p-value is the probability (calculated under the assumption that the null hypothesis is true) that the **test statistic will produce values at least as extreme as the t-score produced for your sample**. As probabilities correspond to areas under the density function, p-value from t-test can be nicely illustrated with the help of the following pictures:



omni CALCULATOR

Test setup

Choose test type: [one-sample](#)

t-test for the population mean, μ , based on one independent sample.

Null hypothesis $H_0: \mu = \mu_0$

Alternative hypothesis $H_1: \mu < \mu_0$

Test details

Approach: [p-value](#)

Do you know the t-score? [Yes](#)

t-score: 2.4

Degrees of freedom: 9

omni CA

Test setup

Choose test type: [one-sample](#)

t-test for the population mean, μ , based on one independent sample.

Null hypothesis $H_0: \mu = \mu_0$

Alternative hypothesis $H_1: \mu \neq \mu_0$

Test results

p-value: 0.980051

Decision:

There is not enough evidence to reject H_0 at the significance level 0.05, because your p-value is greater than 0.05.

Test setup

Choose test type: [one-sample](#)

t-test for the population mean, μ , based on one independent sample.

Null hypothesis $H_0: \mu = \mu_0$

Alternative hypothesis $H_1: \mu \neq \mu_0$

Test details

Approach: [p-value](#)

Do you know the t-score? [Yes](#)

t-score: 2.4

Degrees of freedom: 9

Test results

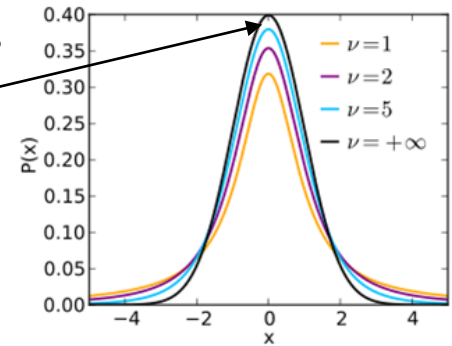
p-value: 0.01994894

Decision:

You can reject H_0 at the significance level 0.05, because your p-value does not exceed 0.05.

One-sample t-test vs Z-test

$N(0,1)$



- The **difference** between t-test and z-test choice of the reference/null distribution
 - A Z-test uses the standard normal ($N(0,1)$) as the reference distribution
 - A t-test uses t as the reference distribution, which is more accurate when the sample size is not large (rule of thumb: $n > 25$, 30?)
- What if the histogram is far from a bell shape or the sample size is small? Non-parametric (Wilcoxon signed-rank test). We will demonstrate a similar idea for two-sample t-test

Paired t-test

- Compare men and women's bmi using couples

Couple ID	Husband's bmi	Wife's bmi	Difference
1	28	35	$d_1 = -7$
2	25	27	$d_2 = -2$
...			
49	24	21	$d_{49} = 3$
50	22	26	$d_{50} = -4$

- Apply the one-sample t-test to the d's to test $H_0: \mu_d = 0$
- Note that two-sample t-test should not be used here. Why?

Two-sample t-test: equal variance

- Suppose that we have two independent samples
 - X_1, \dots, X_m from $N(\mu_X, \sigma^2)$
 - Y_1, \dots, Y_n from $N(\mu_Y, \sigma^2)$
- We are interested in $H_0: \mu_X = \mu_Y$
- It can be shown that

$$\bar{X} \sim N(\mu_X, \frac{1}{m}\sigma^2)$$

$$\bar{Y} \sim N(\mu_Y, \frac{1}{n}\sigma^2)$$

$$\bar{X} - \bar{Y} \sim N(\mu_X - \mu_Y, \sigma^2(\frac{1}{m} + \frac{1}{n}))$$

Two-sample t-test: equal variance

- Similar to the one-sample t-test, we standardize $(\bar{X} - \bar{Y})$ by its standard error (se), which is $s_{\bar{X}-\bar{Y}} = s_p \sqrt{\frac{1}{m} + \frac{1}{n}}$, where

$$s_p^2 = \frac{\sum_{i=1}^m (X_i - \bar{X})^2 + \sum_{i=1}^n (Y_i - \bar{Y})^2}{m + n - 2}$$

- The t-statistic is

$$t = \frac{\bar{X} - \bar{Y}}{s_p \sqrt{\frac{1}{m} + \frac{1}{n}}} \stackrel{H_0: \mu_X = \mu_Y}{\sim} t_{m+n-2}$$

Two-sample t-test: unequal variance

- The assumption of equal variance can be relaxed
- Suppose that we have two independent samples
 - X_1, \dots, X_m from $N(\mu_X, \sigma_X^2)$
 - Y_1, \dots, Y_n from $N(\mu_Y, \sigma_Y^2)$
- We are interested in $H_0: \mu_X = \mu_Y$

• We have
$$\text{se}(\bar{X} - \bar{Y}) = \sqrt{\frac{s_X^2}{n} + \frac{s_Y^2}{m}}$$

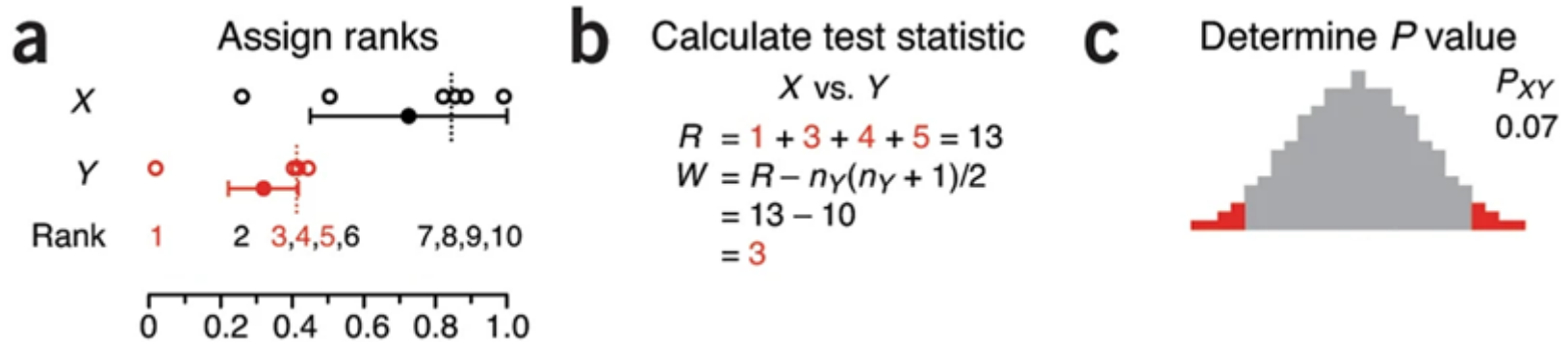
$$\frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_X^2}{n} + \frac{s_Y^2}{m}}} \stackrel{H_0: \mu_X = \mu_Y}{\sim} t_{df}$$

where
$$df = \frac{(s_X^2/m + s_Y^2/n)^2}{\frac{(s_X^2/m)^2}{m-1} + \frac{(s_Y^2/n)^2}{n-1}}$$

Nonparametric tests

- They are distribution-free
- Some nonparametric tests are ranked-based. For example
 - Wilcoxon signed-rank test (for one-sample)
 - Mann-Whitney U-test (for two-sample)
 - Kruska-Wallis test (for \geq two samples)
 - Spearman's r (more robust to outliers than Pearson's r)
- Permutation-based test
- Resampling methods

Mann-Whitney U-test tests



- We used to rely on tables
- Nowadays software reports p-values obtained from either exact or approximate distributions

Critical Values of the Mann-Whitney U
(Two-Tailed Testing)

n_2	α	n_1														
		3	4	5	6	7	8	9	10	11	12	13	14	15	16	
3	.05	--	0	0	1	1	2	2	3	3	4	4	5	5		
	.01	--	0	0	0	0	0	0	0	0	1	1	1	2		
4	.05	--	0	1	2	3	4	4	5	6	7	8	9	10	11	
	.01	--	--	0	0	0	1	1	2	2	3	3	4	5		
5	.05	0	1	2	3	5	6	7	8	9	11	12	13	14	15	
	.01	--	--	0	1	1	2	3	4	5	6	7	7	8		
6	.05	1	2	3	5	6	8	10	11	13	14	16	17	19	20	
	.01	--	0	1	2	3	4	5	6	7	9	10	11	12	13	
7	.05	1	3	5	6	8	10	12	14	16	18	20	22	24	25	
	.01	--	0	1	3	4	6	7	9	10	12	13	15	16	17	

Back to “which test?”

- Q1. Are men and women similar in obesity?

	Continuous measurements (BMI)	Binary measurement (obesity)
Independent observations	Two-sample t-test or <u>Wilcoxon signed-rank test</u>	Chi-squared test or <u>Fisher’s exact test</u>
Couples (data are paired)	Paired t-test (eqt one-sample t-test) or <u>Wilcoxon signed-rank test</u>	Mc-Nemar’s test or <u>binomial sign test</u>

Note: nonparametric or exact tests are underlined. They are recommended for small sample sizes.

Binary X, binary Y

obesity gender	Non-obese (0)	Obese (1)	Total	Prop of obesity
Male	43 (n_{m0}, p_{m0})	9 (n_{m1}, p_{m1})	52 (p_m)	$\frac{p_{m1}}{p_{m0} + p_{m1}} = \frac{p_{m1}}{p_m}$
Female	44 (n_{f0}, p_{f0})	4 (n_{f1}, p_{f1})	48 (p_f)	$\frac{p_{f1}}{p_{f0} + p_{f1}} = \frac{p_{f1}}{p_f}$
Total	87 (n_0, p_0)	13 (n_1, p_1)	100 ($n, 1$)	$p_{f1} + p_{m1}$

• Useful measurements

- Difference in proportions: $\frac{p_{f1}}{p_f} - \frac{p_{m1}}{p_m} = \frac{p_{f1}p_{m0} - p_{f0}p_{m1}}{p_f p_m}$

- Relative risk (RR): $\frac{p_{f1}/p_f}{p_{m1}/p_m} = \frac{p_{f1}}{p_f} \frac{p_m}{p_{m1}} = \frac{(p_f - p_{f0})(p_{m0} + p_{m1})}{p_f p_{m1}} = \frac{p_f p_{m1} + p_{f1} p_{m0} - p_{f0} p_{m1}}{p_f p_{m1}} = 1 + \frac{p_{f1} p_{m0} - p_{f0} p_{m1}}{p_f p_{m1}}$

risks

- Odds ratio (OR): $\frac{p_{f1}/p_{f0}}{p_{m1}/p_{m0}} = \frac{p_{f1} p_{m0}}{p_{f0} p_{m1}} = 1 + \frac{p_{f1} p_{m0} - p_{f0} p_{m1}}{p_{f0} p_{m1}}$

odds

Binary X, Binary Y: Test Statistics

- To estimate the quantities, we can simply replace p with n

- Test statistics

- Z-test for difference in proportions: $Z = \frac{\frac{n_{f1}}{n_f} - \frac{n_{m1}}{n_m}}{\sqrt{(\frac{n_1}{n})(1-\frac{n_1}{n})(1/n_f+1/n_m)}}$

<http://www.sthda.com/english/wiki/two-proportions-z-test-in-r>

- RR: $\frac{\log(\frac{n_{f1}}{n_f} \frac{n_m}{n_{m1}})}{\sqrt{\frac{1}{n_f} + \frac{1}{n_{f1}} + \frac{1}{n_m} + \frac{1}{n_{m1}}}}$

- OR: $\frac{\log(\frac{n_{f1}n_{m0}}{n_{m1}n_{f0}})}{\sqrt{\frac{1}{n_{f0}} + \frac{1}{n_{f1}} + \frac{1}{n_{m0}} + \frac{1}{n_{m1}}}}$

Binary X, binary Y

- Different terms might be preferred in different scenarios. For example,
 - RR is preferred in randomized trials
 - OR is preferred in case-control studies because ...
- The null hypotheses of them are all equivalent to

$$p_{f1}p_{m0} - p_{f0}p_{m1} = 0$$

- Thus, it is not surprising that they share test statistics. For example,
 - When sample sizes are small, Fisher's exact test should be used



Lady testing tea

https://en.wikipedia.org/wiki/Fisher%27s_exact_test

Chi-squared Tests for Categorical Variables

- The idea is to evaluate significance by comparing the **observed** data to the **expected** under the null hypothesis.
- It turns out that the expected tables for “no difference in proportions”, “RR=1”, and “OR=1” are the same

Observed and Expected Counts

gender	obesity	Non-obese (0)	Obese (1)
Male		$n_{m0}, e_{m0} = \frac{n_m n_0}{n}$	$n_{m1}, e_{m1} = \frac{n_m n_1}{n}$
Female		$n_{f0}, e_{f0} = \frac{n_f n_0}{n}$	$n_{f1}, e_{f1} = \frac{n_f n_1}{n}$

$$X = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - e_{ij})^2}{e_{ij}} \stackrel{H_0}{\sim} \chi_{(I-1)(J-1)}^2, \text{ for large sample}$$

$I=J=2$ in this example

Back to “which test?”

- Q1. Are men and women similar in obesity?

	Continuous measurements (BMI)	Binary measurement (obesity)
Independent observations	Two-sample t-test or <u>Wilcoxon signed-rank test</u>	Chi-squared test or <u>Fisher's exact test</u>
Couples (data are paired)	Paired t-test (eqt one-sample t-test) or <u>Wilcoxon signed-rank test</u>	Mc-Nemar's test or <u>binomial sign test</u>

Note: nonparametric or exact tests are underlined. They are recommended for small sample sizes.

Paired Binary Data



- 50 couples and their obesity status
- The null hypothesis is
 - $H_0: \Pr(\text{obese} | \text{wife}) = \Pr(\text{obese} | \text{husband})$, i.e.,
 - $p_{01} + p_{11} = p_{10} + p_{11}$, i.e., $p_{01} = p_{10}$, i.e.,
 - $H_0: \frac{p_{01}}{p_{01} + p_{10}} = \frac{p_{10}}{p_{01} + p_{10}} = \frac{1}{2}$

	wife	Non-obese (0)	Obese (1)
Husband			
Non-obese (0)	20 (n_{00}, p_{00})	15 (n_{01}, p_{01})	
Obese (1)	5 (n_{10}, p_{10})	10 (n_{11}, p_{11})	

McNemar's test

$$X = (n_{10} - \frac{1}{2}(n_{10} + n_{01}))^2 + (n_{01} - \frac{1}{2}(n_{10} + n_{01}))^2 = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}} \overset{H_0}{\sim} \chi_1^2$$

for large sample

- For small sample size, use binomial test

R example:

<https://rpubs.com/mbh038/614538>

Two Useful Nonparametric Methods: Bootstrap and Permutation (re-randomization)

- Motivating example: Inference of a ratio parameter

- **Average of Ratios**

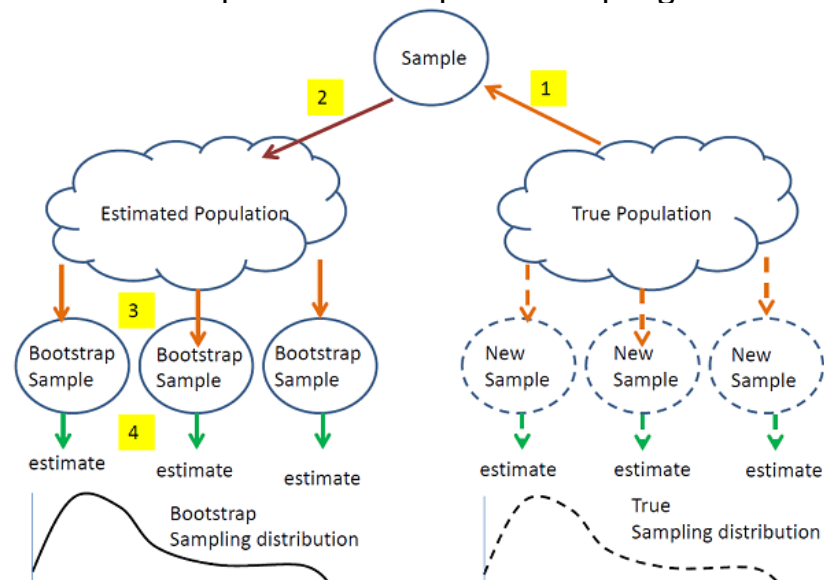
• Experiment condition A	Sugar (3 tech replicates)	Alcohol	ratio
• Biological replicate #1	20, 25, 30 (mean=25)	100, 110, 120 (mean=110)	25/110
• Biological replicate #2	30, 31, 32 (mean=31)	120, 130, 140 (mean=130)	31/130
•	
• Biological replicate #10	35, 38, 38 (mean=37)	160, 140, 120 (mean=140)	<u>37/140</u>
		average of ratios	→ \bar{R}_A
• Experiment condition B	Sugar (3 tech replicates)	Alcohol	
• Biological replicate #1	24, 30, 30 (mean=28)	95, 105, 115 (mean=105)	28/105
• Biological replicate #2	36, 33, 39 (mean=36)	120, 120, 105 (mean=115)	36/115
•	
• Biological replicate #20	42, 45, 45 (mean=44)	120, 129, 120 (mean=123)	<u>44/123</u>
		average of ratios	→ \bar{R}_B

Compare Two Ratios (Average of Ratios)

- Use $\bar{R}_A - \bar{R}_B$ to estimate the true difference
- How to quantify uncertainty?
 - Method 1: this is a two-sample problem. Use “t.test” in R
 - Method 2: Use **bootstrap** to find standard errors and confidence intervals, use **permutations/re-randomizations** to compute p-values

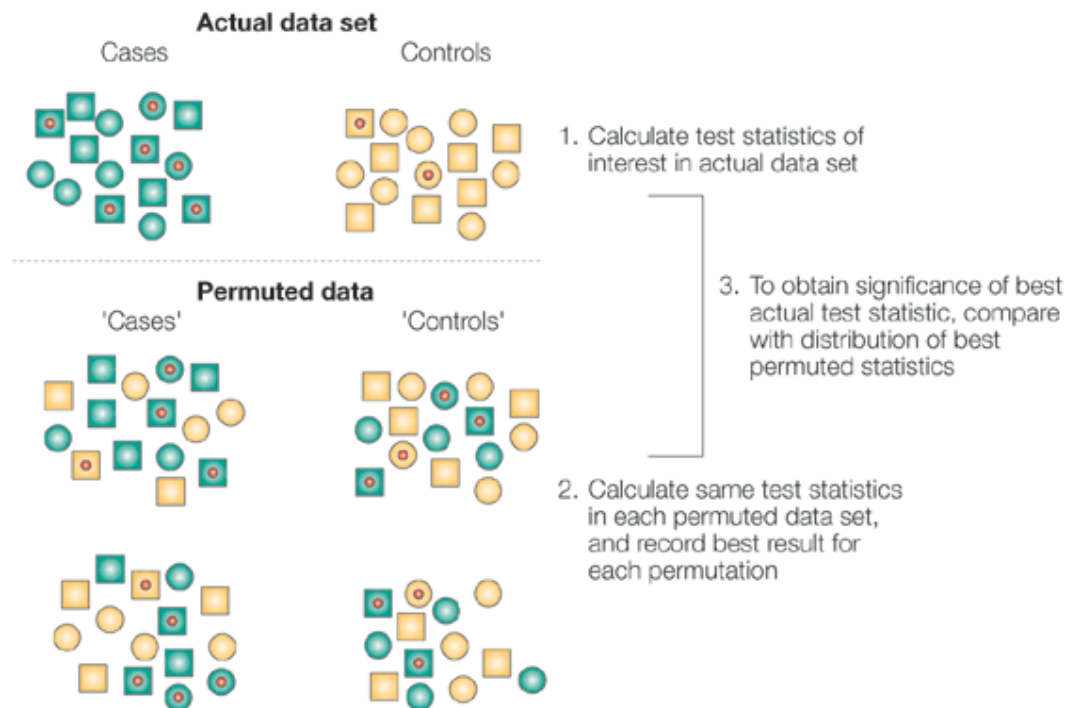
Bootstrap: example

- The idea of bootstrap is to find the sampling distribution of an estimator/statistic by resampling with replacement
- Example:
 - Consider $\bar{R}_A - \bar{R}_B$, which is an estimator of the underlying the true difference in ratio
 - Resample with replacement (stratified based on experimental conditions)
 - For each resampled data set, compute $\bar{R}_A - \bar{R}_B$
 - Do in many times. The results provide an empirical sampling distribution of estimator



Permutation/re-randomization test

- The idea is to find the null distribution of a statistic by randomly **shuffling** labels (such as the cases and control labels)



Bootstrap vs Permutation

- Bootstrap
 - Idea: sampling with replacement. Stratification might be needed
 - Confidence intervals can be obtained easily. For example, by using empirical quantiles
 - Can also produce p-values. Remark: to produce p-values, resampling must be modified in a way that reflects the null hypothesis.
- Permutation
 - Idea: shuffling group memberships to produce permuted data
 - Produce p-values
 - Easy to implement
- Advanced consideration
 - Standardized statistics tend to be more accurate:
 - Hall, P. and Wilson, S.R., 1991. Two guidelines for bootstrap hypothesis testing. *Biometrics*, pp.757-762.

Compare Two Ratios (Average of Ratios)

Use **Bootstrap** to produced 95% **confidence interval**

- `set.seed(20220128)` #to ensure reproducibility
- `#observations 1-10 are from population A; 11-30 are from population B`
- `y=c(25/110, runif(7, 0.2, 0.3), 31/130, 37/140,`
- `28/105, 36/115, runif(17, 0.25, 0.3), 44/123)`
- `ex.cond=c(rep(1,10), rep(2,20))`

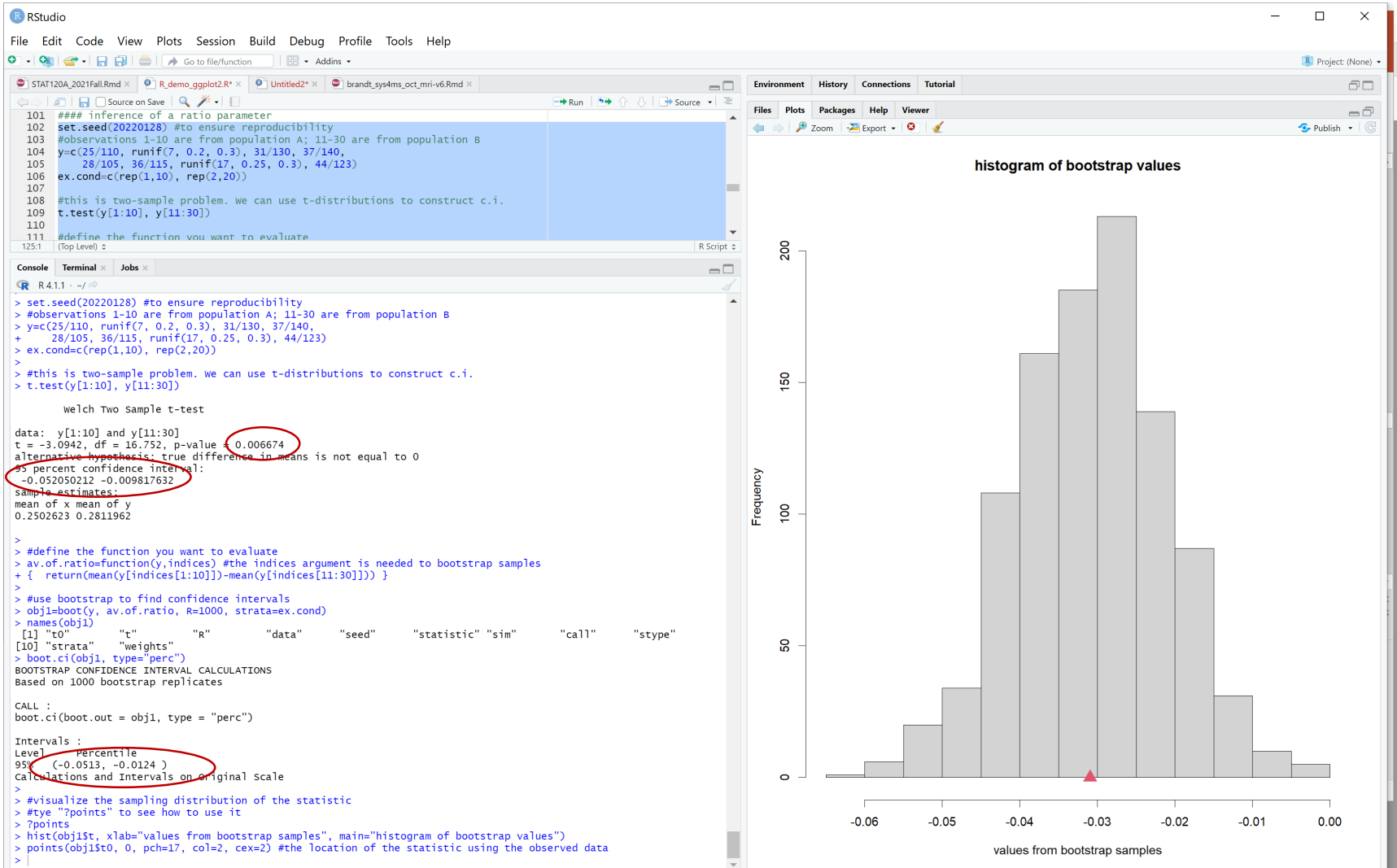
- `#this is two-sample problem. We can use t-distributions to construct c.i.`
- `t.test(y[1:10], y[11:30])`

- `#define the function you want to evaluate`
- `av.of.ratio=function(y,indices) #the indices argument is needed to bootstrap samples`
- `{ return(mean(y[indices[1:10]])-mean(y[indices[11:30]])) }`

- `#use bootstrap to find confidence intervals`
- `obj1=boot(y, av.of.ratio, R=1000, strata=ex.cond)`
- `names(obj1)`
- `boot.ci(obj1, type="perc")`

- `#visualize the sampling distribution of the statistic`
- `#tye "?points" to see how to use it`
- `?points`
- `hist(obj1$t, xlab="values from bootstrap samples", main="histogram of bootstrap values")`
- `points(obj1$t0, 0, pch=17, col=2, cex=2) #the location of the statistic using the observed data`

Compare Two Ratios (Average of Ratios)



Compare Two Ratios (Average of Ratios)

Use **permutations** to calculate **p-value**

- `#compute p-values using permutations`
- `set.seed(20220128) #to ensure reproducibility`
- `org.diff=av.of.ratio(y, 1:30)`
- `perm.diff=rep(0,10000)`
- `perm.diff[1]=org.diff`
- `for(i in 2:10000)`
- `{`
- `perm.diff[i]=av.of.ratio(y,sample(1:30))`
- `}`
- `mean(abs(perm.diff)>=abs(org.diff)) #two-sided p-value`
- `hist(perm.diff, xlab="values from 1000 permutations", main="histogram of permuted values")`
- `points(org.diff, 0, pch=17, col=2, cex=2)`
- `text(org.diff, 5, "observed", col=2)`

Two Useful Nonparametric Methods: Bootstrap and Permutation (re-randomization)

- Motivating example: Inference of a ratio parameter

- **Ratio of averages**

• Experiment condition A	Sugar (3 tech replicates)	Alcohol	
• Biological replicate #1	20, 25, 30 (mean=25)	100, 110, 120 (mean=110)	
• Biological replicate #2	30, 31, 32 (mean=31)	120, 130, 140 (mean=130)	
•	
• Biological replicate #10	<u>35, 38, 38 (mean=37)</u> average of sugar	<u>160, 140, 120 (mean=140)</u> average of alcohol	→ \bar{R}_A
• Experiment condition B	Sugar (3 tech replicates)	Alcohol	
• Biological replicate #1	24, 30, 30 (mean=28)	95, 105, 115 (mean=105)	
• Biological replicate #2	36, 33, 39 (mean=36)	120, 120, 105 (mean=115)	
•	
• Biological replicate #20	<u>42, 45, 45 (mean=44)</u> average of sugar	<u>120, 129, 120 (mean=123)</u> average of alcohol	→ \bar{R}_B

Compare Two Ratios (Ratio of averages)

- Use $\bar{R}_A - \bar{R}_B$ to estimate the true difference
- How to quantify uncertainty?
 - Method 1: Use approximation methods to find their standard errors (e.g., the “survey” package in R). Then

$$\frac{\bar{R}_A - \bar{R}_B - (\text{true diff})}{\sqrt{[se(\bar{R}_A)]^2 + [se(\bar{R}_B)]^2}} \sim N(0,1), \text{ for large sample}$$

- Method 2: Use **bootstrap** to find standard errors and confidence intervals , use **permutations/re-randomizations** to compute p-values

Compare Two Ratios (Ratio of averages)

Use **Bootstrap** to produced 95% **confidence interval**

- `### Suppose that ratio of averages makes more sense`
- `set.seed(123)`
- `sugar=c(runif(10, 20, 30), runif(20, 25, 35))`
- `alcohol=c(runif(10, 100, 120), runif(20, 110, 120))`
- `y=data.frame(sugar, alcohol)`
- `ratio.of.av=function(y, indices)`
- `{`
- `booty.sugar=y$sugar[indices]`
- `booty.alcohol=y$alcohol[indices]`
- `return(mean(booty.sugar[1:10])/mean(booty.alcohol[1:10]) -`
- `mean(booty.sugar[11:30])/mean(booty.alcohol[11:30]))`
- `}`
- `obj2=boot(y, ratio.of.av, R=1000, strata=ex.cond)`
- `boot.ci(obj2, type="perc")`
- `hist(obj2$t, xlab="values from bootstrap samples", main="histogram of bootstrap values")`
- `points(obj2$t0, 0, pch=17, col=2, cex=2) #the location of the statistic using the observed data`

Compare Two Ratios (Ratio of averages)

RStudio

```

143 ### Suppose that ratio of averages makes more sense
144 set.seed(123)
145 sugar=c(runif(10, 20, 30), runif(20, 25, 35))
146 alcohol=c(runif(10, 100, 120), runif(20, 110, 120))
147 y=data.frame(sugar, alcohol)
148 ratio.of.av=function(y, indices)
149 {
150   booty.sugar=y$sugar[indices]
151   booty.alcohol=y$alcohol[indices]
152   return( mean(booty.sugar[1:10])/mean(booty.alcohol[1:10]) -
153           mean(booty.sugar[11:30])/mean(booty.alcohol[11:30]) )
154 }
155 obj2=boot(y, ratio.of.av, R=1000, strata=ex.cond)
156 boot.ci(obj2, type="perc")
157 hist(obj2$t, xlab="values from bootstrap samples", main="histogram of bootstrap values")
158 points(obj2$t0, 0, pch=17, col=2, cex=2) #the location of the statistic using the observed data
  
```

```

>
>
> ### Suppose that ratio of averages makes more sense
> set.seed(123)
> sugar=c(runif(10, 20, 30), runif(20, 25, 35))
> alcohol=c(runif(10, 100, 120), runif(20, 110, 120))
> y=data.frame(sugar, alcohol)
> ratio.of.av=function(y, indices)
+ {
+   booty.sugar=y$sugar[indices]
+   booty.alcohol=y$alcohol[indices]
+   return( mean(booty.sugar[1:10])/mean(booty.alcohol[1:10]) -
+           mean(booty.sugar[11:30])/mean(booty.alcohol[11:30]) )
+ }
> obj2=boot(y, ratio.of.av, R=1000, strata=ex.cond)
> boot.ci(obj2, type="perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = obj2, type = "perc")

Intervals :
Level Percentile
95% (-0.0580, -0.0129)
calculations and intervals on original scale
> hist(obj2$t, xlab="values from bootstrap samples", main="histogram of bootstrap values")
> points(obj2$t0, 0, pch=17, col=2, cex=2)
  
```

Environment History Connections Tutorial

Files Plots Packages Help Viewer

Project: (None)

Zoom Export

Publish

histogram of bootstrap values

Frequency

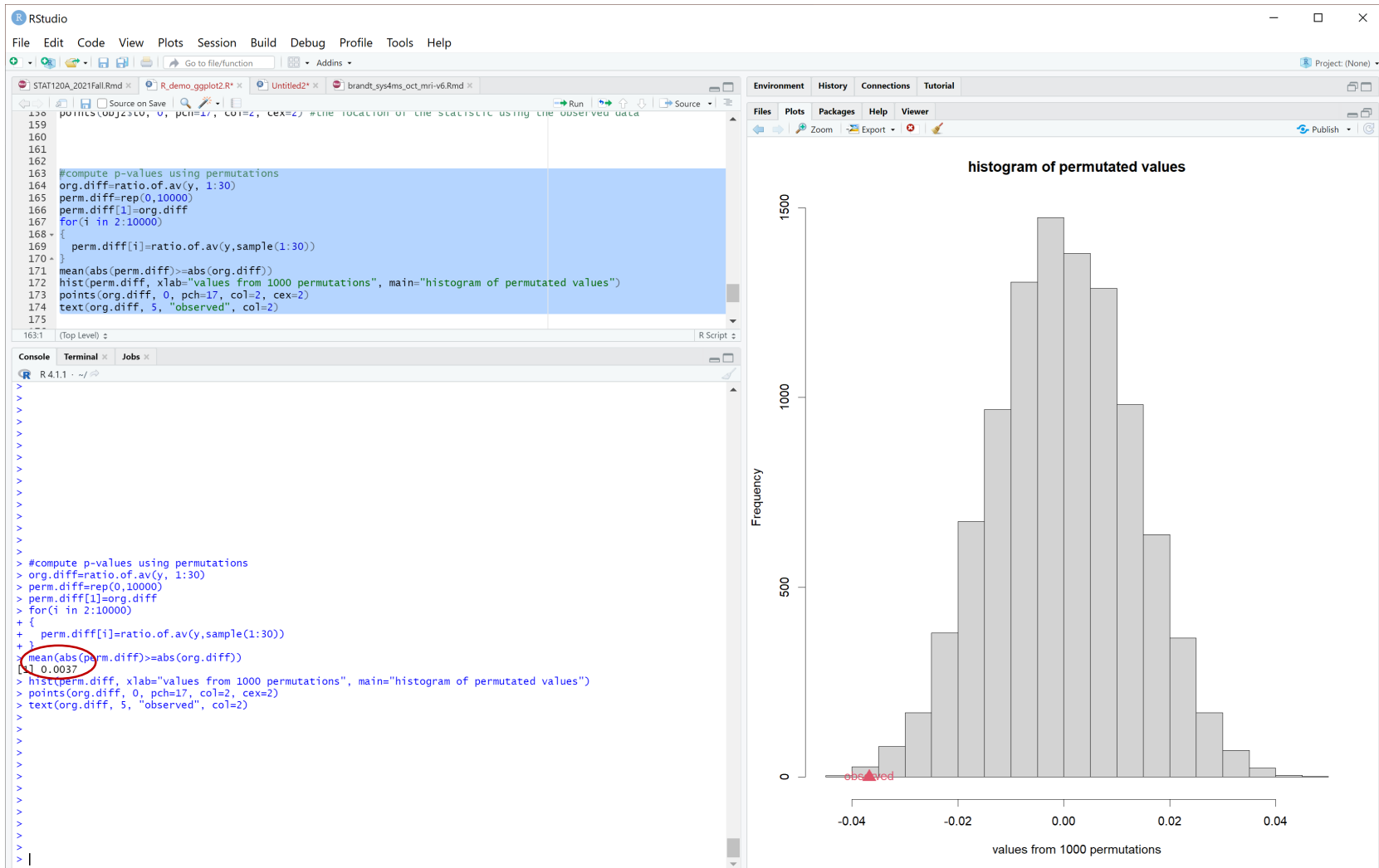
values from bootstrap samples

Compare Two Ratios (Ratio of averages)

Use **permutations** to calculate **p-value**

- #compute p-values using permutations
- org.diff=ratio.of.av(y, 1:30)
- perm.diff=rep(0,10000)
- perm.diff[1]=org.diff
- for(i in 2:10000)
- {
- perm.diff[i]=ratio.of.av(y,sample(1:30))
- }
- mean(abs(perm.diff)>=abs(org.diff))
- hist(perm.diff, xlab="values from 1000 permutations", main="histogram of permuted values")
- points(org.diff, 0, pch=17, col=2, cex=2)
- text(org.diff, 5, "observed", col=2)

Compare Two Ratios (Ratio of averages)



Type I and Type II errors

- Mistakes in hypothesis testing: the null hypothesis might be rejected wrongly or the alternative hypothesis might be accepted wrongly
- **Type I error (false positive)** occurs when the null is true but we reject the null. For a test at significance level α ,

$$\Pr(\text{Type I error}) = \Pr(\text{reject } H_0 \mid H_0 \text{ is true}) = \alpha.$$

- **Type II error (false negative)** occurs when the alternative is true but we fail to reject the null
 - $\Pr(\text{Type II error}) = \Pr(\text{fail to reject } H_0 \mid H_0 \text{ is not true})$
 - $1 - \Pr(\text{Type II error}) = \Pr(\text{reject } H_0 \mid H_0 \text{ is not true})$ is called the power of a test, denoted as β

		H_0 true	H_a true
Do not reject H_0	Reject H_0	Type I error	Correct decision
	Do not reject H_0	Correct decision	Type II error

Test power

- Ways to increase test power

- Increase sample size
 - Increase the significance level α
 - Increase the difference between the sample estimate and the null value
 - Decrease the population standard deviation
- } not practical

In practice, we increase test power by increasing sample size.



Power and Sample Size

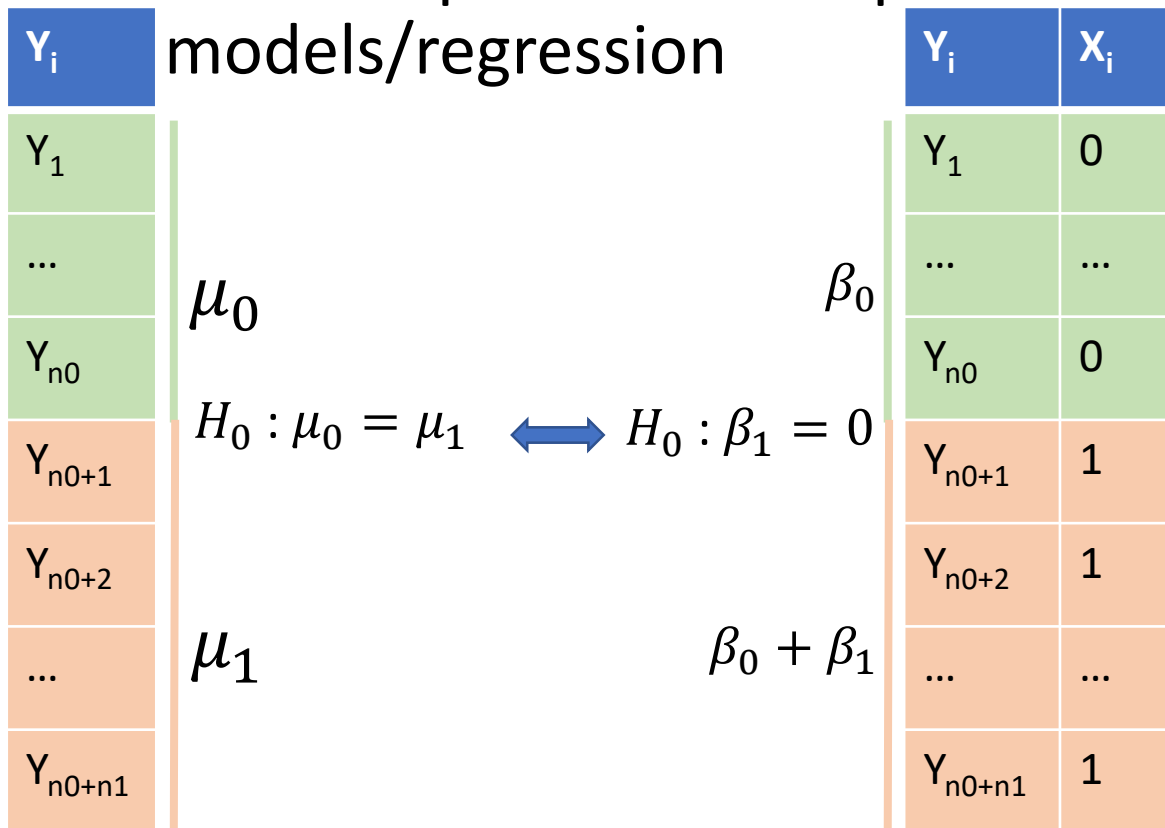
- Power analysis in R
 - pwr: basic functions for power analysis. <https://cran.r-project.org/web/packages/pwr/index.html>
- <https://statpages.info/#power> provides links to online calculators
- Beyond basic tests
 - Packages have been releases for specific topics. For example, for single-cell RNA-seq
 - <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3167-9>
 - http://www.bioconductor.org/packages/release/bioc/html/POWS_C.html
 - For complicated models, the calculation is often simulation-based

Multiple Comparisons in R

- Post-hoc pairwise comparisons (ANOVA)
 - <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/pairwise.t.test>
 - <https://stats.oarc.ucla.edu/r/faq/how-can-i-do-post-hoc-pairwise-comparisons-in-r/>
- For a list of p-values, multiple comparisons can be corrected by a simple function “p.adjust”
 - <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html>
 - Web-based tools such as <https://www.multipletesting.com/analysis>
 - Bonferroni’s correction for family wise error (the probability of at least one false positive)
 - Very simple. Instead of using 0.05 as the cutoff, use $0.05/K$ where K is the number of tests performed
 - False discovery rate (FDR): the proportion of false positives among the discovered ones
 - Preferred when there is a large number of tests such as gene expression
 - <https://www.sdmproject.com/utilities/?show=FDR>

Beyond Basic Methods: Linear Models

- Two-sample t-test is a special case of linear models/regression



$$Y_i = \beta_0 + X_i \beta_1 + \varepsilon_i, \quad i = 1, \dots, n_0, n_0 + 1, \dots, n_0 + n_1$$

Beyond Basic Methods: linear models

- ANOVA is also a special case of linear models/regression

	Y_i	$X_{i,1}$	$X_{i,2}$	
μ_0	Y_1	0	0	β_0
	...	0	0	
	Y_9	0	0	
μ_1	Y_{10}	1	0	$\beta_0 + \beta_1$
	...	1	0	
	Y_{21}	1	0	
μ_2	Y_{22}	0	1	$\beta_0 + \beta_2$
	...	0	1	
	Y_{40}	0	1	

$$H_0: \mu_0 = \mu_1 = \mu_2$$

$$H_0: \mu_1 - \mu_0 = \mu_1 - \mu_0 = 0$$

$$H_0: \beta_1 = \beta_2 = 0$$

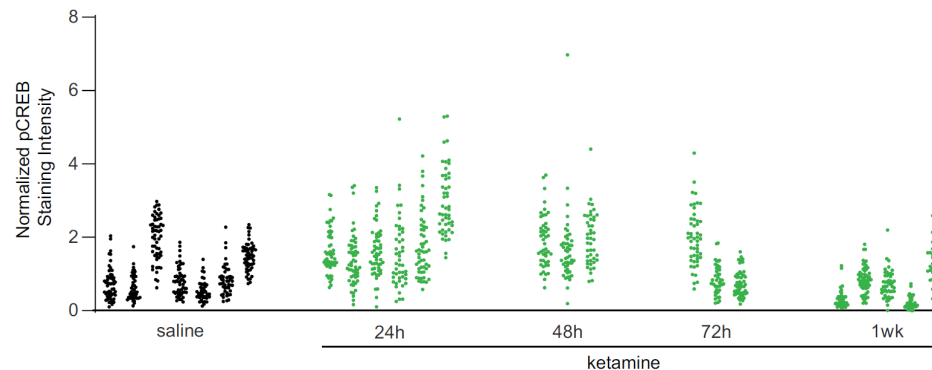
$$Y_i = \beta_0 + X_{i,1}\beta_1 + \dots + X_{i,p}\beta_p + \varepsilon_i$$

Why are linear models useful?

- Adjust for covariates, which is particularly important in observational studies
 - E.g., association between obesity and gender might be dependent on other factors, such as ethnicity, countries, income, etc
- Study multiple factors/conditions easily
 - E.g., cell type and experimental conditions. Missing data can be handled naturally
- Account for design effects such as clustering

Beyond Basic Methods

- A generalized form, known as linear mixed-effects models (LME), can take data dependency into consideration



- A generalized form, known as generalized linear models (GLM), can model non-continuous data. E.g., logistic regression
- Generalized linear mixed-effects models (GLMM) include both the above extensions

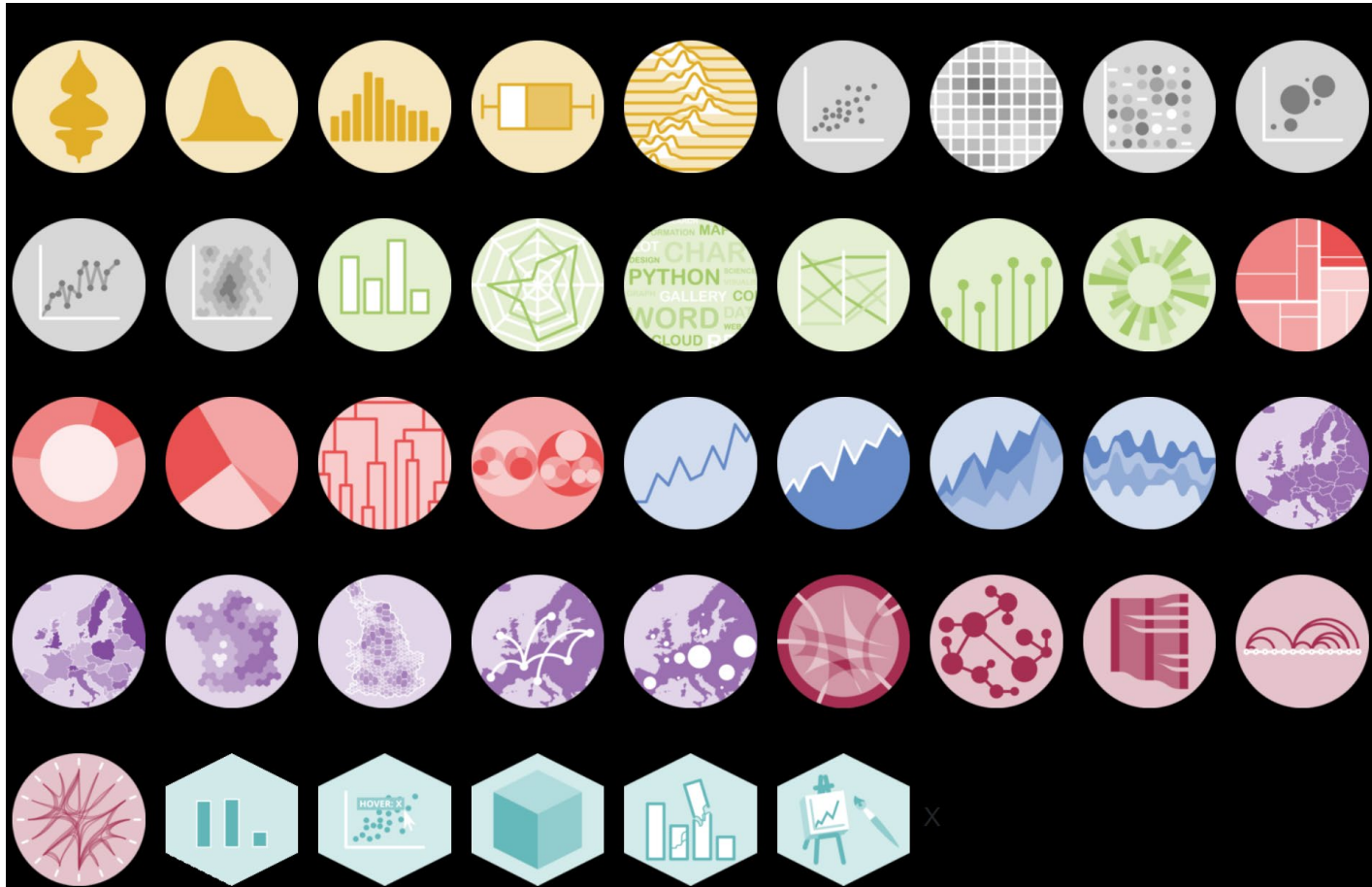
<https://www.sciencedirect.com/science/article/pii/S089662732100845X>

<https://cnmcm.som.uci.edu/lmem-intro/>

Data Visualization

- Data visualization should be the first, rather than the last, step
- <https://www.r-graph-gallery.com>
- <https://www.r-graph-gallery.com/base-R.html>
- <https://www.r-graph-gallery.com/ggplot2-package.html>
- <https://r-charts.com/>
- <http://www.sthda.com/english/wiki/ggplot2-violin-plot-quick-start-guide-r-software-and-data-visualization>
- https://ggplot2.tidyverse.org/reference/geom_violin.html

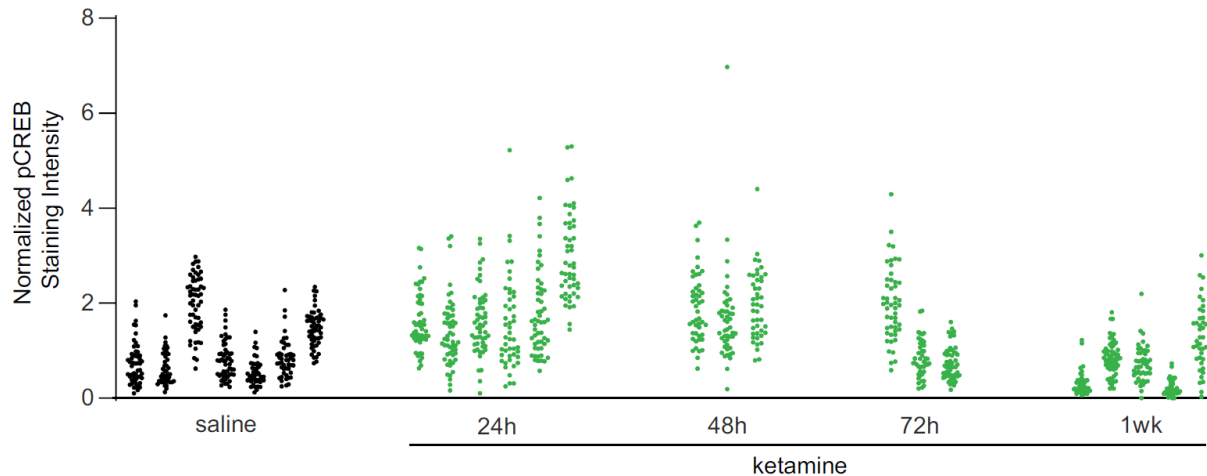
R



<https://www.r-graph-gallery.com/>

Example: create boxplots using R

- Data: 1200 neurons from 24 mice; 5 conditions/groups



- We will show how to use R to visualize data

Example: create boxplots using R

- Read and check data

```
ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
#Remark 1: https won't work for this version
#Remark 2: read.csv is used because the separator used in the data is ","
names(ex1)
dim(ex1)
table(ex1$treatment_idx)
table(ex1$midx)
table(ex1$treatment_idx, ex1$midx)
ex1$midx=as.factor(ex1$midx)
ex1$treatment_idx=as.factor(ex1$treatment_idx)
```

Example: create boxplots using R

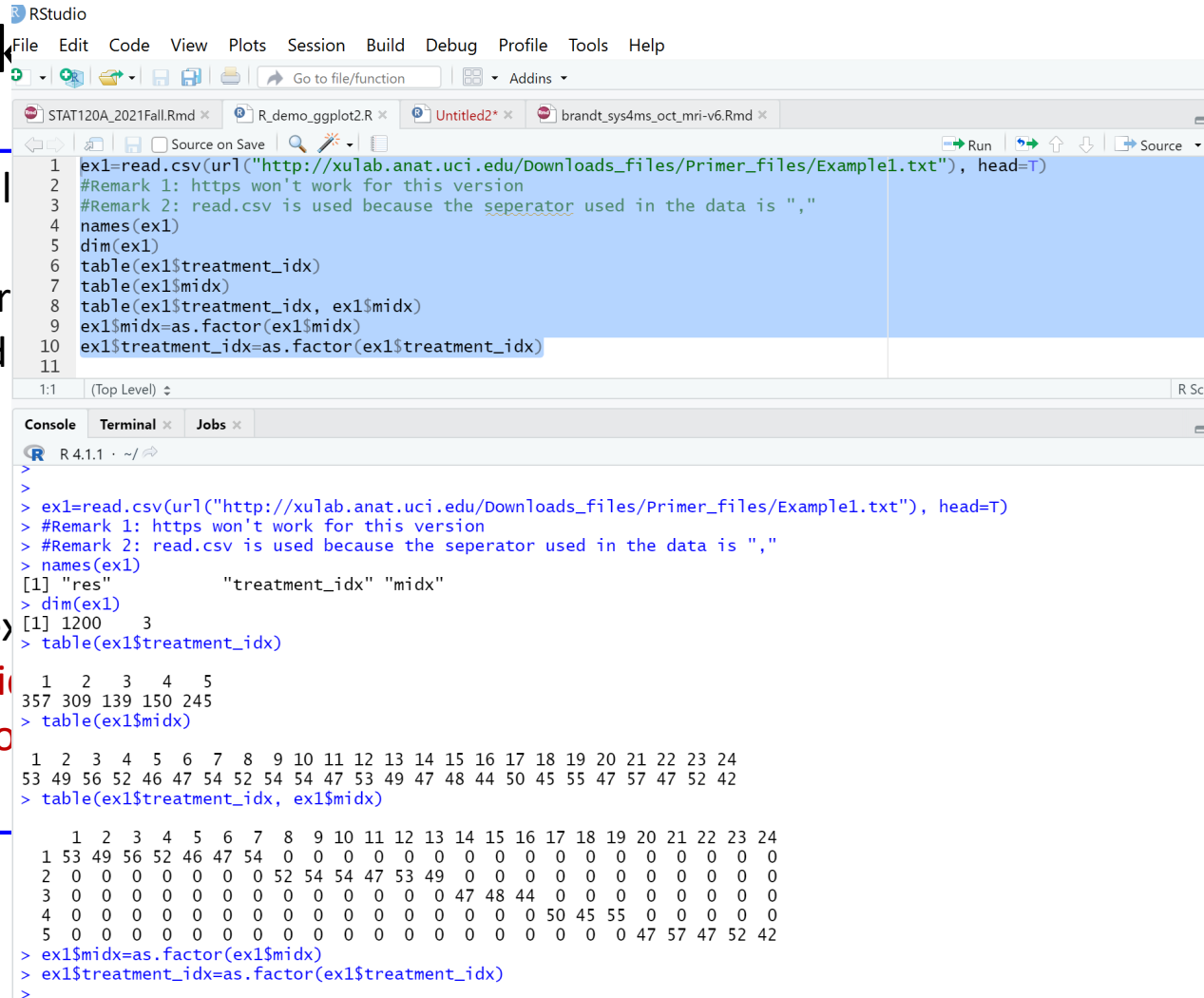
- Read and check data

```
ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
#Remark 1: https won't work for this version
#Remark 2: read.csv is used because the separator used in the data is ","
names(ex1)
dim(ex1)
table(ex1$treatment_idx)
table(ex1$midx)
table(ex1$treatment_idx, ex1$midx)
ex1$midx=as.factor(ex1$midx)
ex1$treatment_idx=as.factor(ex1$treatment_idx)
```

Example: create boxplots using R

- Read and check

```
ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
#Remark 1: https won't work for this version
#Remark 2: read.csv is used because the separator used in the data is ","
names(ex1)
dim(ex1)
table(ex1$treatment_idx)
table(ex1$midx)
table(ex1$treatment_idx, ex1$midx)
ex1$midx=as.factor(ex1$midx)
ex1$treatment_idx=as.factor(ex1$treatment_idx)
```



The screenshot shows the RStudio interface with the following code in the editor and output in the console:

```
1 ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
2 #Remark 1: https won't work for this version
3 #Remark 2: read.csv is used because the separator used in the data is ","
4 names(ex1)
5 dim(ex1)
6 table(ex1$treatment_idx)
7 table(ex1$midx)
8 table(ex1$treatment_idx, ex1$midx)
9 ex1$midx=as.factor(ex1$midx)
10 ex1$treatment_idx=as.factor(ex1$treatment_idx)
11
```

Console output:

```
R 4.1.1 · ~/
>
>
> ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
> #Remark 1: https won't work for this version
> #Remark 2: read.csv is used because the separator used in the data is ","
> names(ex1)
[1] "res"          "treatment_idx" "midx"
> dim(ex1)
[1] 1200  3
> table(ex1$treatment_idx)
 1  2  3  4  5
357 309 139 150 245
> table(ex1$midx)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
53 49 56 52 46 47 54 52 54 54 47 53 49 47 48 44 50 45 55 47 57 47 52 42
> table(ex1$treatment_idx, ex1$midx)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
1 53 49 56 52 46 47 54 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 52 54 54 47 53 49 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 47 48 44 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 50 45 55 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 47 57 47 52 42
> ex1$midx=as.factor(ex1$midx)
> ex1$treatment_idx=as.factor(ex1$treatment_idx)
>
```

Example: create boxplots using R

- Use base graphics

```
#Use base graphics
mycolors=rep(1:5, c(7,6,3,3,5)) #different colors for different groups
mycolors

boxplot(res~midx, data=ex1, col=mycolors, xaxt="n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))
#boxplot with gitters
boxplot(res~midx, data=ex1, col=0, xaxt="n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)
```

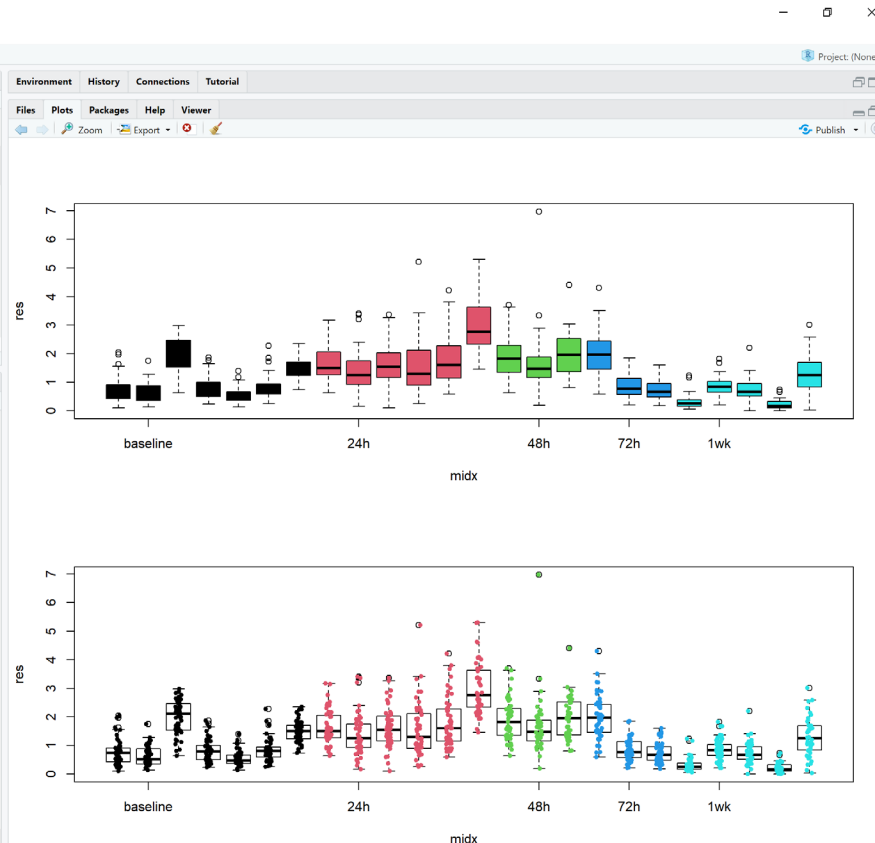

Example: create boxplots using R

- Use

```
#Use base  
mycolors=1  
mycolors
```

```
boxplot(re  
axis(1, at =  
labels =  
#boxplot w  
boxplot(re  
axis(1, at =  
labels =  
stripchart(  
method = "jitter", add = TRUE, pch = 20, col = mycolors)
```

```
RStudio  
File Edit Code View Plots Session Build Debug Profile Tools Help  
Go to file/function Addins  
STAT120A_2021FallRmd R_demo.ggplot2.R Untitled2* brandt_sys4ms_oct_mri-v6.Rmd  
11  
12  
13 #Use base graphics  
14 mycolors=rep(1:5, c(7,6,3,3,5)) #different colors for different groups  
15 mycolors  
16 parm(mfrow=c(2,1))  
17 boxplot(res=midx, data=ex1, col=mycolors, xaxt="n")  
18 axis(1, at = 1+c(1, 8, 14, 17, 20),  
19 labels = c("baseline", "24h", "48h", "72h", "1wk"))  
20 #boxplot with gitters  
21 boxplot(res=midx, data=ex1, col=0, xaxt="n")  
22 axis(1, at = 1+c(1, 8, 14, 17, 20),  
23 labels = c("baseline", "24h", "48h", "72h", "1wk"))  
24 stripchart(res = midx, vertical = TRUE, data = ex1,  
25 method = "jitter", add = TRUE, pch = 20, col = mycolors)  
26  
27  
28  
26:1 (Top Level)  
Console Terminal Jobs  
R 4.1.1 ~ /  
> #Use base graphics  
> mycolors=rep(1:5, c(7,6,3,3,5)) #different colors for different groups  
> mycolors  
[1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 4 4 4 5 5 5 5  
> parm(mfrow=c(2,1))  
> boxplot(res=midx, data=ex1, col=mycolors, xaxt="n")  
Error in parm(mfrow = c(2, 1)) : could not find function "parm"  
> boxplot(res=midx, data=ex1, col=mycolors, xaxt="n")  
> axis(1, at = 1+c(1, 8, 14, 17, 20),  
+ labels = c("baseline", "24h", "48h", "72h", "1wk"))  
> #boxplot with gitters  
> boxplot(res=midx, data=ex1, col=0, xaxt="n")  
> axis(1, at = 1+c(1, 8, 14, 17, 20),  
+ labels = c("baseline", "24h", "48h", "72h", "1wk"))  
> stripchart(res = midx, vertical = TRUE, data = ex1,  
+ method = "jitter", add = TRUE, pch = 20, col = mycolors)
```



```
method = "jitter", add = TRUE, pch = 20, col = mycolors)
```

Example: create boxplots using R

- Use “vioplot” package

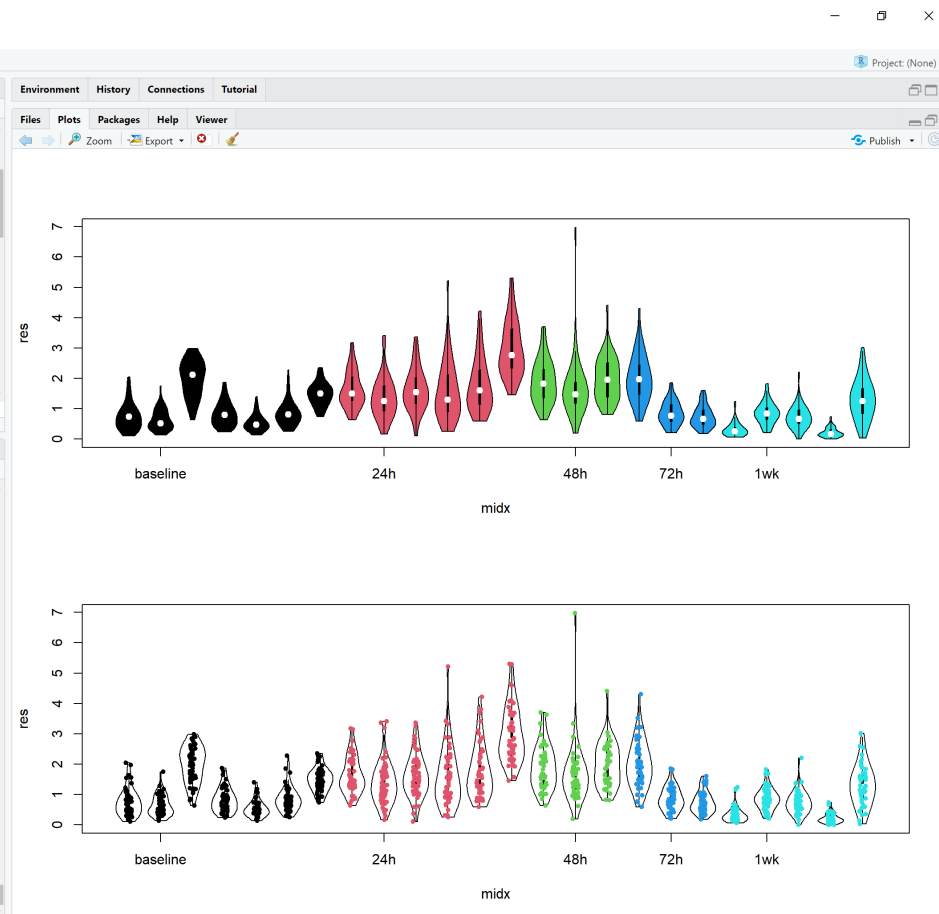
```
# install.packages("vioplot")
library("vioplot")
par(mfrow=c(2,1))
vioplot(res~midx, data=ex1, col=mycolors, xaxt = "n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))

#violin plot with jitters
vioplot(res~midx, data=ex1, col=0, xaxt="n")
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))
```

Example: create boxplots using R

- Use “vioplot” package

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
STAT120A_2021Fall.Rmd x R_demo_ggplot2.R x Untitled2 x brandt_sys4ms_oct_mri-v6.Rmd x
# install.packages("vioplot")
library("vioplot")
par(mfrow=c(2,1))
vioplot(res~midx, data=ex1, col=mycolors, xaxt = "n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
      labels = c("baseline", "24h", "48h", "72h", "1wk"))
#violin plot with jitters
vioplot(res~midx, data=ex1, col=0, xaxt="n")
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)
axis(1, at = 1+c(1, 8, 14, 17, 20),
      labels = c("baseline", "24h", "48h", "72h", "1wk"))
#violin plot with jitters
vioplot(res~midx, data=ex1, col=0, xaxt="n")
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)
axis(1, at = 1+c(1, 8, 14, 17, 20),
      labels = c("baseline", "24h", "48h", "72h", "1wk"))
```



Example: create boxplots using R

- Make fancier plots by ggplot2

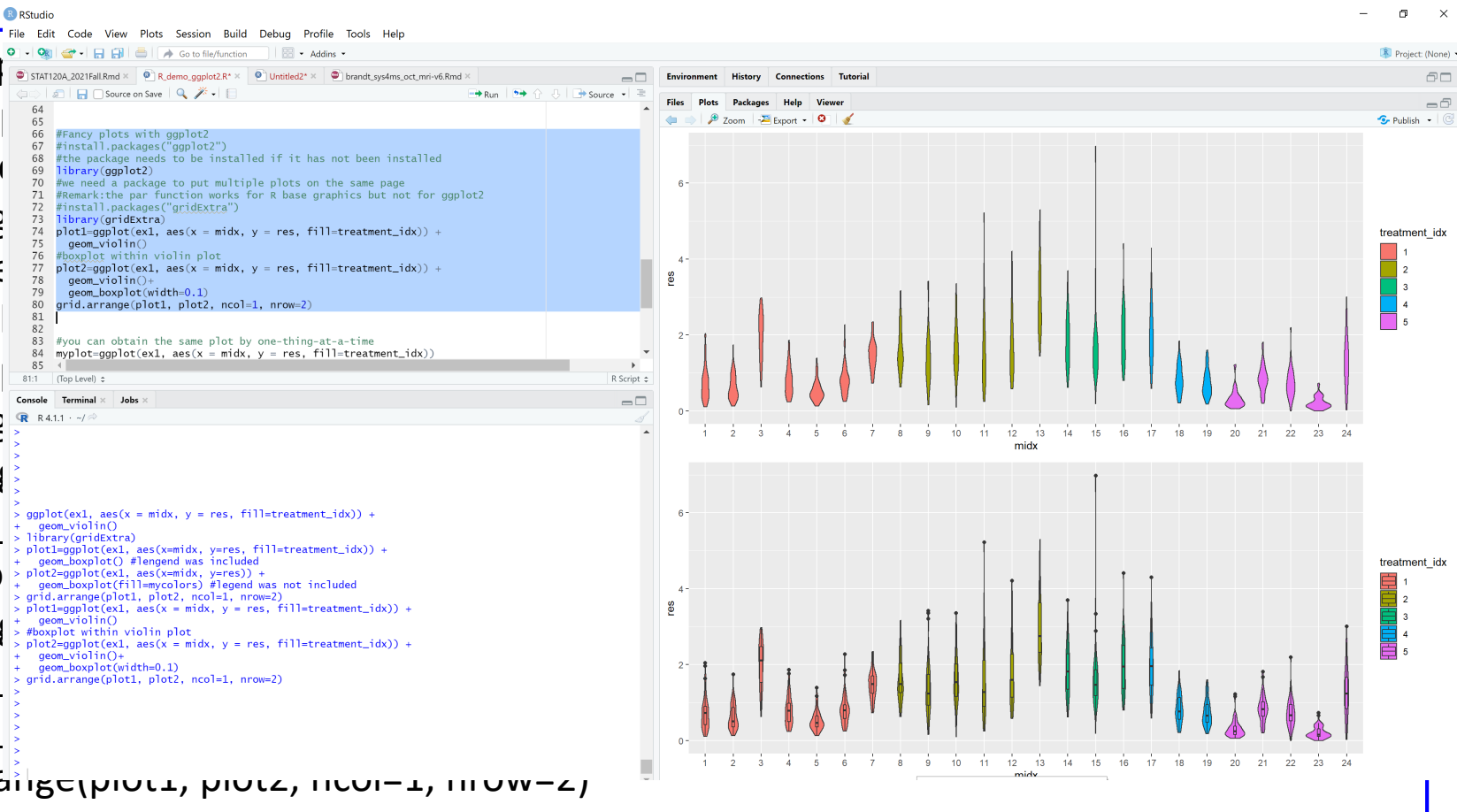
```
#Fancy plots with ggplot2
#install.packages("ggplot2")
#the package needs to be installed if it has not been installed
library(ggplot2)
#we need a package to put multiple plots on the same page
#Remark:the par function works for R base graphics but not for ggplot2
#install.packages("gridExtra")
library(gridExtra)
plot1=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()
#boxplot within violin plot
plot2=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()+
  geom_boxplot(width=0.1)
grid.arrange(plot1, plot2, ncol=1, nrow=2)
```

Example: create boxplots using R

- Make fancier plots by ggplot2

```
#Fancy plots with ggplot2
#install.packages("ggplot2")
#the package needs to be installed if it has not been installed
library(ggplot2)
#we need a package to put multiple plots on the same page
#Remark: the par function works for R base graphics but not for ggplot2
#install.packages("gridExtra")
library(gridExtra)
plot1=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()
#boxplot within violin plot
plot2=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()+
  geom_boxplot(width=0.1)
grid.arrange(plot1, plot2, ncol=1, nrow=2)

#you can obtain the same plot by one-thing-at-a-time
myplot=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx))
```



The image shows a screenshot of the RStudio interface. On the left, the R console displays R code for creating violin plots with overlaid boxplots. The code uses ggplot2 and gridExtra. The main plot area shows two vertically stacked violin plots. The x-axis is labeled 'midx' and ranges from 1 to 24. The y-axis is labeled 'res' and ranges from 0 to 6. Each violin plot represents a different 'midx' value. The violins are filled with colors corresponding to 'treatment_idx' values 1 through 5. The top plot shows only the violin shapes, while the bottom plot shows the violin shapes with a small boxplot overlaid on each. A legend on the right side of each plot indicates the color mapping for treatment_idx: 1 (red), 2 (orange), 3 (yellow), 4 (green), and 5 (purple).

Example: create boxplots using R

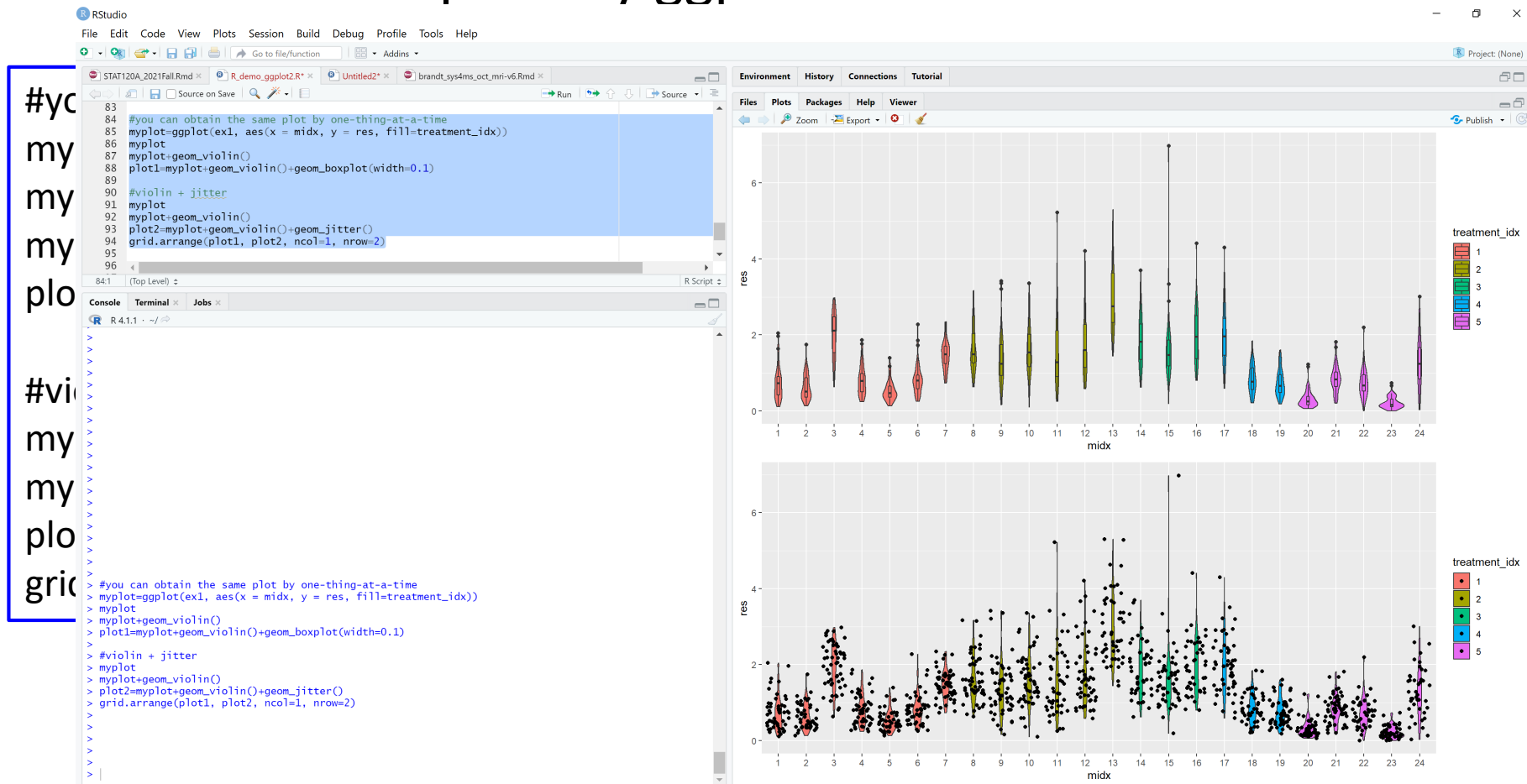
- Make fancier plots by ggplot2

```
#you can obtain the same plot by one-thing-at-a-time
myplot=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx))
myplot
myplot+geom_violin()
plot1=myplot+geom_violin()+geom_boxplot(width=0.1)

#violin + jitter
myplot
myplot+geom_violin()
plot2=myplot+geom_violin()+geom_jitter()
grid.arrange(plot1, plot2, ncol=1, nrow=2)
```

Example: create boxplots using R

- Make fancier plots by ggplot2



Future Topics

- Regression-based methods
- Multivariate analysis
- Single-cell RNA seq
- Spatial transcriptomics
- Integration of different methods

Too much for a Friday? Happy Friday!



```

ex1=read.csv(url("http://xulab.anat.uci.edu/Downloads_files/Primer_files/Example1.txt"), head=T)
#Remark 1: https won't work for this version
#Remark 2: read.csv is used because the separator used in the data is ","
names(ex1)
dim(ex1)
table(ex1$treatment_idx)
table(ex1$midx)
table(ex1$treatment_idx, ex1$midx)
ex1$midx=as.factor(ex1$midx)
ex1$treatment_idx=as.factor(ex1$treatment_idx)

#Use base graphics
mycolors=rep(1:5, c(7,6,3,3,5)) #different colors for different groups
mycolors
par(mfrow=c(2,1))
boxplot(res~midx, data=ex1, col=mycolors, xaxt="n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))
#boxplot with jitters
boxplot(res~midx, data=ex1, col=0, xaxt="n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)

# install.packages("vioplot")
library("vioplot")
par(mfrow=c(2,1))
vioplot(res~midx, data=ex1, col=mycolors, xaxt = "n")
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))

#violin plot with jitters
vioplot(res~midx, data=ex1, col=0, xaxt="n")
stripchart(res ~ midx, vertical = TRUE, data = ex1,
           method = "jitter", add = TRUE, pch = 20, col = mycolors)
axis(1, at = 1+c(1, 8, 14, 17, 20),
     labels = c("baseline", "24h", "48h", "72h", "1wk"))

```

```

#Fancy plots with ggplot2
#install.packages("ggplot2")
#the package needs to be installed if it has not been installed
library(ggplot2)

#we need a package to put multiple plots on the same page
#Remark:the par function works for R base graphics but not for ggplot2
#install.packages("gridExtra")
library(gridExtra)
plot1=ggplot(ex1, aes(x=midx, y=res, fill=treatment_idx)) +
  geom_boxplot() #legend was included
plot2=ggplot(ex1, aes(x=midx, y=res)) +
  geom_boxplot(fill=mycolors) #legend was not included
grid.arrange(plot1, plot2, ncol=1, nrow=2)
#the "fill" argument is tricky. The two methods produce the same plot, except for choice
of colors

plot1=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()
#boxplot within violin plot
plot2=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx)) +
  geom_violin()+
  geom_boxplot(width=0.1)
grid.arrange(plot1, plot2, ncol=1, nrow=2)

#you can obtain the same plot by one-thing-at-a-time
myplot=ggplot(ex1, aes(x = midx, y = res, fill=treatment_idx))
myplot
myplot+geom_violin()
plot1=myplot+geom_violin()+geom_boxplot(width=0.1)

#violin + jitter
myplot
myplot+geom_violin()
plot2=myplot+geom_violin()+geom_jitter()
grid.arrange(plot1, plot2, ncol=1, nrow=2)

#http://www.sthda.com/english/wiki/ggplot2-violin-plot-quick-start-guide-r-software-
and-data-visualization

```

Bootstrap and Permutation Test

Example 1: average of ratios

```
##### inference of a ratio parameter
## use the average of ratios
set.seed(20220128) #to ensure reproducibility
#observations 1-10 are from population A; 11-30 are from population B
y=c(25/110, runif(7, 0.2, 0.3), 31/130, 37/140,
    28/105, 36/115, runif(17, 0.25, 0.3), 44/123)
ex.cond=c(rep(1,10), rep(2,20))

#this is two-sample problem. We can use t-distributions to construct c.i.
t.test(y[1:10], y[11:30])

#define the function you want to evaluate
av.of.ratio=function(y,indices) #the indices argument is needed to bootstrap samples
{ return(mean(y[indices[1:10]])-mean(y[indices[11:30]])) }

#use bootstrap to find confidence intervals
obj1=boot(y, av.of.ratio, R=1000, strata=ex.cond)
names(obj1)
boot.ci(obj1, type="perc")

#visualize the sampling distribution of the statistic
#tye "?points" to see how to use it
?points
hist(obj1$t, xlab="values from bootstrap samples", main="histogram of bootstrap values")
points(obj1$t0, 0, pch=17, col=2, cex=2) #the location of the statistic using the observed data

#compute p-values using permutations
set.seed(20220128) #to ensure reproducibility
org.diff=av.of.ratio(y, 1:30)
perm.diff=rep(0,10000)
perm.diff[1]=org.diff
for(i in 2:10000)
{
  perm.diff[i]=av.of.ratio(y,sample(1:30))
}
mean(abs(perm.diff)>=abs(org.diff)) #two-sided p-value
hist(perm.diff, xlab="values from 1000 permutations", main="histogram of permuted values")
points(org.diff, 0, pch=17, col=2, cex=2)
text(org.diff, 5, "observed", col=2)
```

Example 2: ratio of averages

```
##### use the ratio of averages #####
### Suppose that ratio of averages makes more sense
set.seed(123)
sugar=c(runif(10, 20, 30), runif(20, 25, 35))
alcohol=c(runif(10, 100, 120), runif(20, 110, 120))
y=data.frame(sugar, alcohol)
ratio.of.av=function(y, indices)
{
  booty.sugar=y$sugar[indices]
  booty.alcohol=y$alcohol[indices]
  return( mean(booty.sugar[1:10])/mean(booty.alcohol[1:10]) -
    mean(booty.sugar[11:30])/mean(booty.alcohol[11:30]) )
}
obj2=boot(y, ratio.of.av, R=1000, strata=ex.cond)
boot.ci(obj2, type="perc")
hist(obj2$t, xlab="values from bootstrap samples",
  main="histogram of bootstrap values")
points(obj2$t0, 0, pch=17, col=2, cex=2) #the location of the
statistic using the observed data

#compute p-values using permutations
org.diff=ratio.of.av(y, 1:30)
perm.diff=rep(0,10000)
perm.diff[1]=org.diff
for(i in 2:10000)
{
  perm.diff[i]=ratio.of.av(y,sample(1:30))
}
mean(abs(perm.diff)>=abs(org.diff))
hist(perm.diff, xlab="values from 1000 permutations",
  main="histogram of permuted values")
points(org.diff, 0, pch=17, col=2, cex=2)
text(org.diff, 5, "observed", col=2)
```