A/B Testing

Jameson Watts, Ph.D.

Agenda

1. Regression Review
2. A/B Testing
3. 1 on 1 consultations (hopefully)

Environment setup

```
library(tidyverse)
library(moderndive)
data(evals)
```

Regression Review

Practice

1. Load the “evals” dataset from the moderndive library with the command $data(evals)$
2. Pair up and create a linear model that uses **bty_avg** to predict **score**
3. Transform the dependent and/or independent variables as necessary
4. Create a sentence that explains the magnitude of the relationship
5. Graph the residuals for adherence to assumptions of linear regression

- Linearity
- Independence
- Normality
- Equality

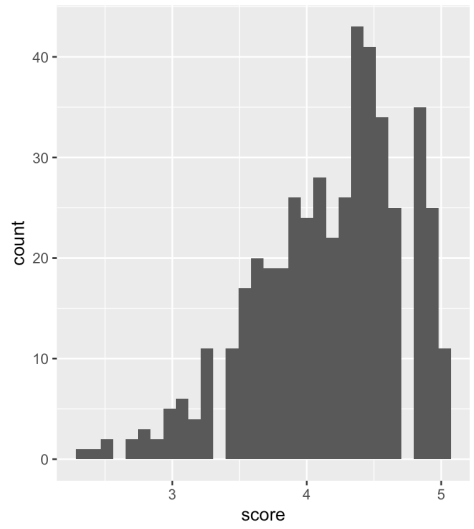
Data overview

```
glimpse(evals)
```

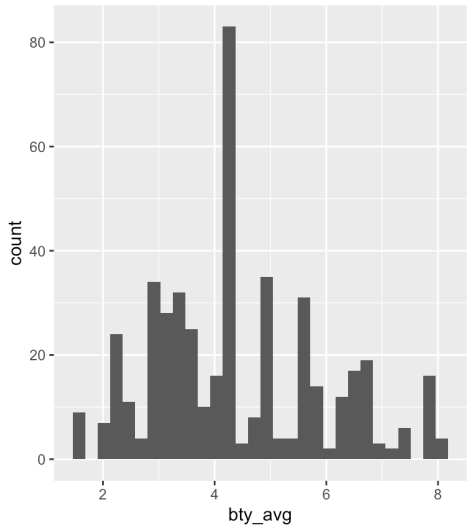
```
## Observations: 463
## Variables: 14
## $ ID           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, …
## $ prof_ID      <int> 1, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 4, …
## $ score        <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3, 2.8, 4.1, 3.4, 4.5, 3…
## $ age          <int> 36, 36, 36, 36, 59, 59, 59, 51, 51, 40, 40, 40, 40,…
## $ bty_avg      <dbl> 5.000, 5.000, 5.000, 5.000, 3.000, 3.000, 3.000, 3.…
## $ gender       <fct> female, female, female, female, male, male, male, m…
## $ ethnicity    <fct> minority, minority, minority, minority, not minorit…
## $ language     <fct> english, english, english, english, english, englis…
## $ rank         <fct> tenure track, tenure track, tenure track, tenure tr…
## $ pic_outfit   <fct> not formal, not formal, not formal, not formal, not…
## $ pic_color    <fct> color, color, color, color, color, color, color, co…
## $ cls_did_eval <int> 24, 86, 76, 77, 17, 35, 39, 55, 111, 40, 24, 24, 17…
## $ cls_students <int> 43, 125, 125, 123, 20, 40, 44, 55, 195, 46, 27, 25,…
## $ cls_level    <fct> upper, upper, upper, upper, upper, upper, upper, up…
```

Histograms

```
ggplot(evals, aes(score))+geom_histogram()
```




```
ggplot(evals, aes(bty_avg))+geom_histogram()
```



Correlations

```
summarise(evals,correlation=cor(score,bty_avg))$correlation
```

```
## [1] 0.1871424
```

```
summarise(evals,correlation=cor(log(score),bty_avg))$correlation
```

```
## [1] 0.1823194
```

```
summarise(evals,correlation=cor(score,log(bty_avg)))$correlation
```

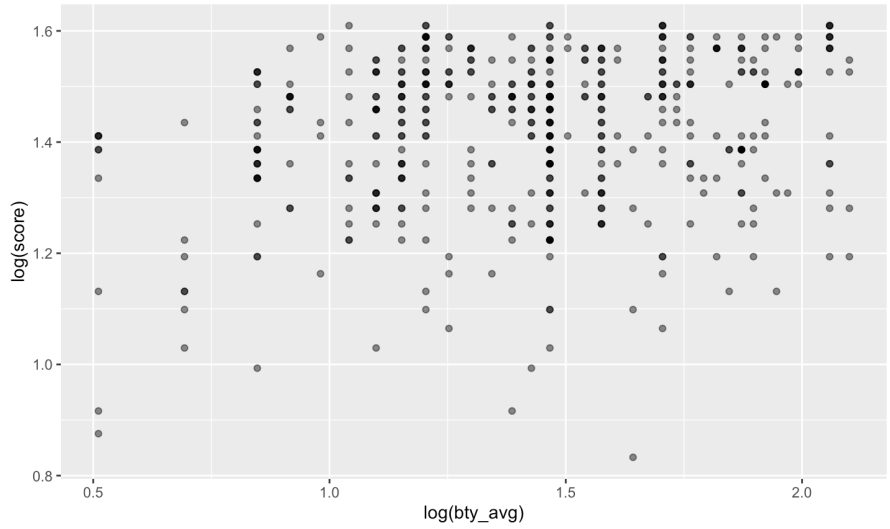
```
## [1] 0.2109599
```

```
summarise(evals,correlation=cor(log(score),log(bty_avg)))$correlation
```

```
## [1] 0.2083852
```

Linearity

```
evals %>%
  ggplot(aes(log(bty_avg), log(score)))+
  geom_point(alpha=.5)
```



Models

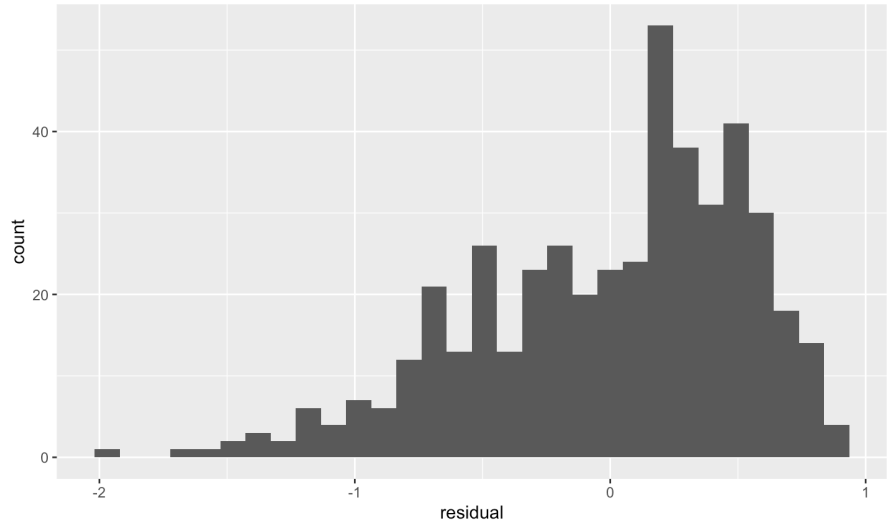
```
estimates = list()
estimates[[1]] <- get_regression_table(lm(score~bty_avg, data = evals))$estimate[2]
estimates[[2]] <- get_regression_table(lm(log(score+1)~bty_avg, data = evals))$estimate[2]
estimates[[3]] <- get_regression_table(lm(score~log(bty_avg+1), data = evals))$estimate[2]
estimates[[4]] <- get_regression_table(lm(log(score+1)~log(bty_avg+1), data = evals))$estimate[2]
for(e in 1:4){
  print(estimates[[e]])
}
```

```
## [1] 0.067
## [1] 0.013
## [1] 0.394
## [1] 0.079
```

- plain ~ plain: 1 point increase in EV -> x change in DV
- log ~ plain: 1 point increase in EV -> $e^x - 1 * 100$ percent change in DV
- plain ~ log: 1 percent increase in EV -> $x/100$ change in DV
- log ~ log: 1 percent increase in EV -> x percent change in DV

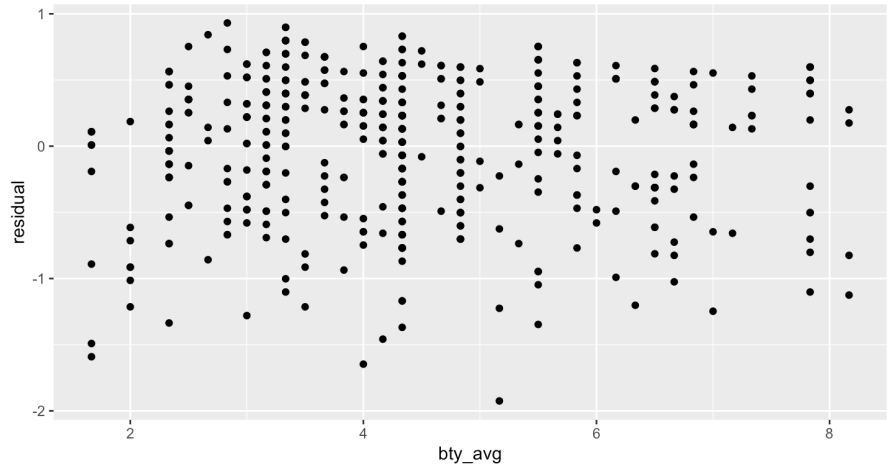
Normality

```
get_regression_points(lm(score~bty_avg, data = evals)) %>%
  ggplot(aes(residual))+
    geom_histogram()
```



Residuals

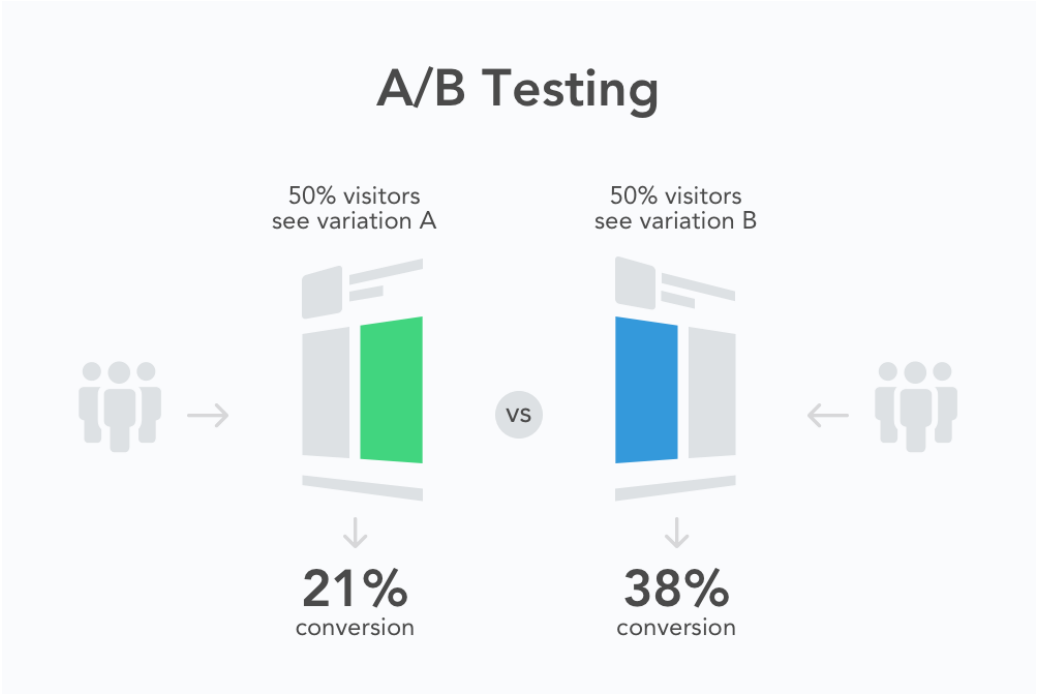
```
get_regression_points(lm(score~bty_avg, data = evals)) %>%
  ggplot(aes(bty_avg, residual))+
    geom_point()
```



What about independence?

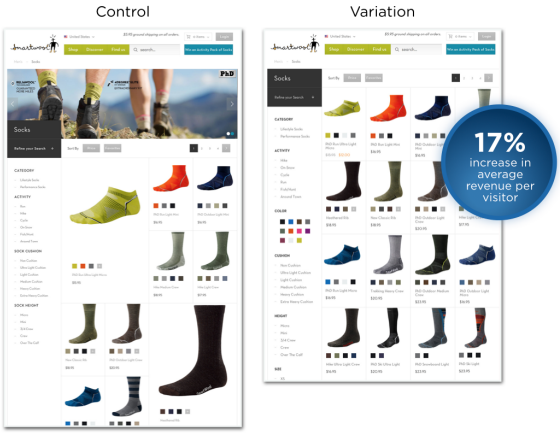
Field experiments (A/B testing)

Credit: Warm thanks to Elea Feit for content on A/B testing.



Source: splitmetrics.com

Example A/B test



Source: Optimizely Blog

1. Randomly assign customers to treatments
2. Measure response(s)
3. Compare groups to determine how the treatment changes response

Why A/B tests work

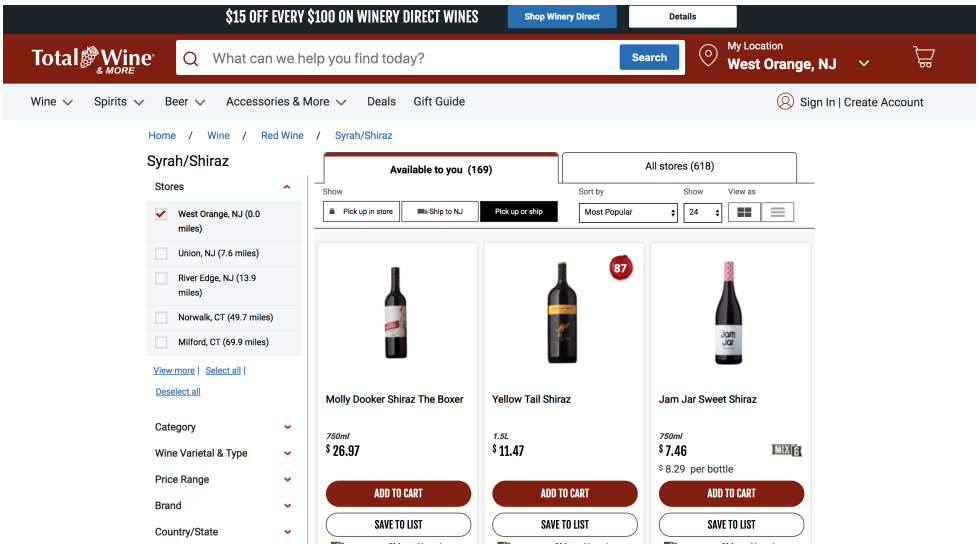
By **randomizing** over a large number of customers, we create groups that are the same, on average.

Any behavioral differences between these groups is **caused by the treatments** we randomly assigned.

1, 2, 3. Repeat with me. Randomization will set you free.

Example email A/B test

The email A/B test we will analyze was conducted by an online-only wine store.



Wine retailer email test

Test setting: email to retailer customers

Unit: customer (email address)

Treatments: email version A, email version B, holdout

Reponse: open, click and 1-month purchase ($)

Selection: all active customers

Assignment: randomly assigned (1/3 each)

Wine retailer email test data

```
setwd("~/GDrive/teaching/GSMDS-5001/")
d <- read.csv("resources/ab-data.csv")
head(d)
```

```
##   user_id   cpgn_id   group open click purch  chard sav_blanc syrah   cab
## 1 1000001 1901Email    ctrl    0     0  0.00   0.00       0.00 33.94  0.00
## 2 1000002 1901Email email_B    0     0  0.00   0.00       0.00 16.23 76.31
## 3 1000003 1901Email email_A    1     1 12.95 516.39       0.00 16.63  0.00
## 4 1000004 1901Email email_A    0     0  0.00   0.00       0.00  0.00 41.21
## 5 1000005 1901Email email_B    1     0 18.85 426.53    1222.48  0.00  0.00
## 6 1000006 1901Email email_A    0     0  0.00   0.00       0.00  0.00  0.00
##   past_purch last_purch visits
## 1      33.94        119     11
## 2      92.54         60      3
## 3     533.02          9      9
## 4      41.21        195      6
## 5    1649.01         48      9
## 6       0.00        149      6
```

Types of variables associated with a test

- **Treatment indicator** (x's)
  - Which (randomized) treatment was received
- **Response** (y's)
  - Outcome(s) measured for each customer. Aka the DV or dependant variable.
- **Baseline variables** ("z's")
  - Other stuff we know about customers **prior** to the randomization

Everything measured after the randomization is an outcome

Treatment indicator

```
d %>%
  group_by(group) %>%
  count()
```

```
## # A tibble: 3 x 2
## # Groups:   group [3]
##   group       n
##   <fct>   <int>
## 1 ctrl    41330
## 2 email_A 41329
## 3 email_B 41329
```

Responses

- open test email (load images)
- click test email to visit website
- purchases ($) in 30 days after email sent

```
summary(d[,c("open", "click", "purch")])
```

```
##       open             click              purch        
##  Min.   :0.0000   Min.   :0.00000   Min.   :   0.00  
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:   0.00  
##  Median :0.0000   Median :0.00000   Median :   0.00  
##  Mean   :0.4569   Mean   :0.07374   Mean   :  13.27  
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:   0.00  
##  Max.   :1.0000   Max.   :1.00000   Max.   :1820.70
```

Baseline variables

- days since last purchase
- website visits
- total past purchases ($)

```
summary(d[,c("last_purch", "visits", "past_purch")])
```

```
##    last_purch         visits         past_purch
##  Min.   :  0.00   Min.   : 0.000   Min.   :   0.00
##  1st Qu.: 26.00   1st Qu.: 4.000   1st Qu.:   0.00
##  Median : 63.00   Median : 6.000   Median :  91.22
##  Mean   : 89.98   Mean   : 5.946   Mean   : 188.79
##  3rd Qu.:125.00   3rd Qu.: 7.000   3rd Qu.: 246.87
##  Max.   :992.00   Max.   :51.000   Max.   :9636.92
```

More baseline variables

- total past purchases by category ($)

```
summary(d[, c("chard", "sav_blanc", "syrah", "cab")])
```

```
##      chard           sav_blanc           syrah              cab        
##  Min.   :   0.00   Min.   :   0.00   Min.   :   0.00   Min.   :   0.00  
##  1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00  
##  Median :   0.00   Median :   0.00   Median :   0.00   Median :   0.00  
##  Mean   :  73.31   Mean   :  72.45   Mean   :  26.68   Mean   :  16.35  
##  3rd Qu.:  54.06   3rd Qu.:  57.42   3rd Qu.:  20.91   3rd Qu.:  12.96  
##  Max.   :9636.92   Max.   :6609.92   Max.   :2880.15   Max.   :2365.90
```

Whoa! That's a lot of chardonnay for one customer!

Analysis of A/B tests

What is the first question you should ask about an A/B test?

~~Did the treatment affect the response?~~

Was the randomization done correctly?

How could we check the randomization with the data?

Randomization checks 1

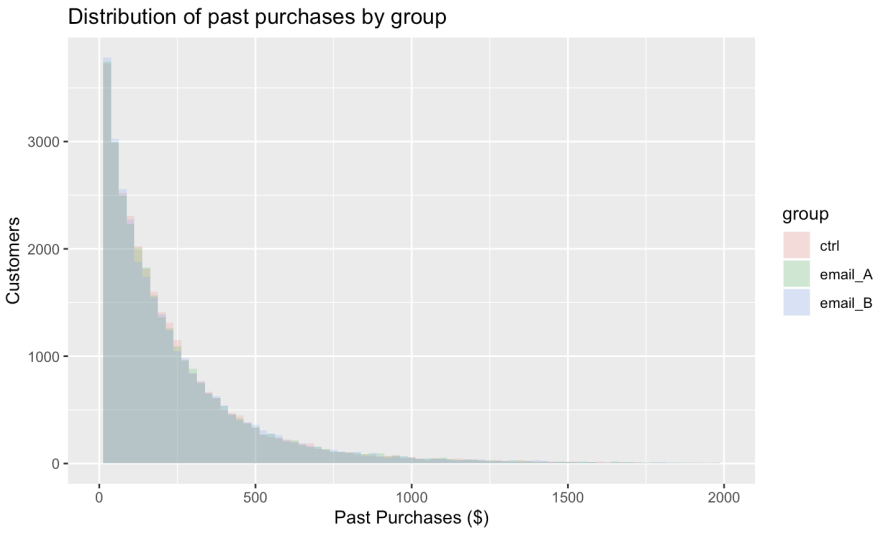
Randomization checks confirm that the baseline variables are distributed similarly for the treatment and control groups.

Averages of baseline variables by treatment group

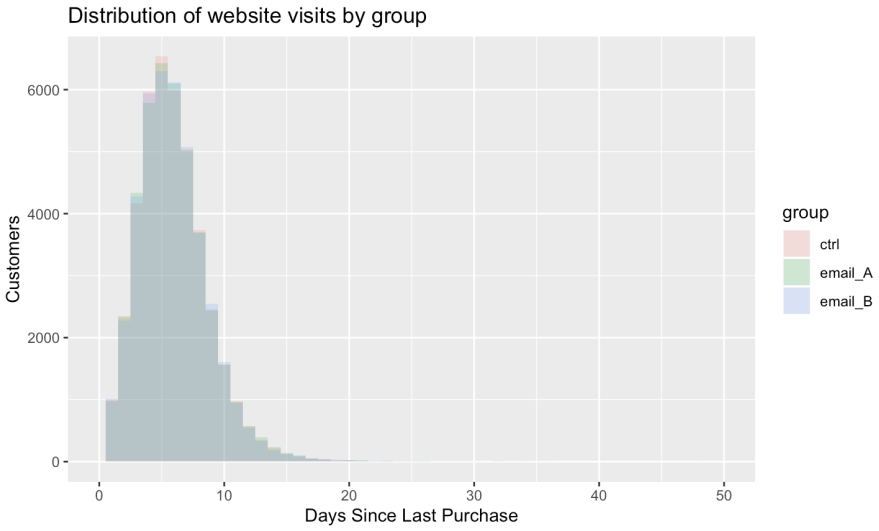
```
## # A tibble: 3 x 5
##   group  `mean(last_purc… `mean(visits)` `mean(past_purc… `mean(past_purch…
##   <fct>             <dbl>          <dbl>             <dbl>             <dbl>
## 1 ctrl               90.1           5.94              188.             0.745
## 2 email…             90.1           5.95              190.             0.742
## 3 email…             89.8           5.94              188.             0.739
```

Randomization checks 2

The distributions of baseline variables should also be the same between groups.



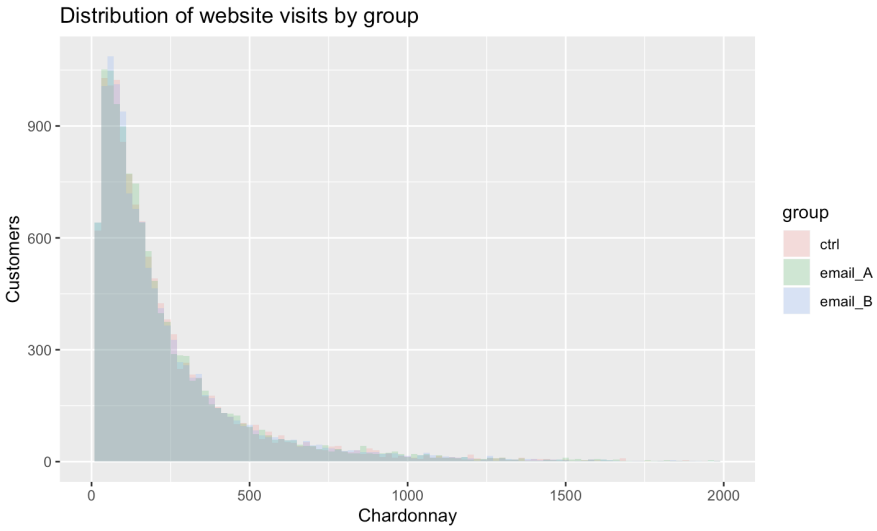
Randomization checks 3



Exercise

1. Choose a wine category
2. Compare the past purchases in this category to confirm that the randomization produced groups with similar distributions.

Solution


Distribution of website visits by group
Customers
900
600
300
0
0
500
1000
1500
2000
Chardonnay
group
ctrl
email_A
email_B

Did the treatments affect the response?

```
## # A tibble: 3 x 4
##   group   `mean(open)` `mean(click)` `mean(purch)`
##   <fct>          <dbl>         <dbl>         <dbl>
## 1 ctrl           0             0              12.3
## 2 email_A        0.717         0.132          13.8
## 3 email_B        0.654         0.0897         13.7
```

Email A looks better for opens and clicks, but maybe not purchases. Both emails seem to generate higher average purchases than the control.

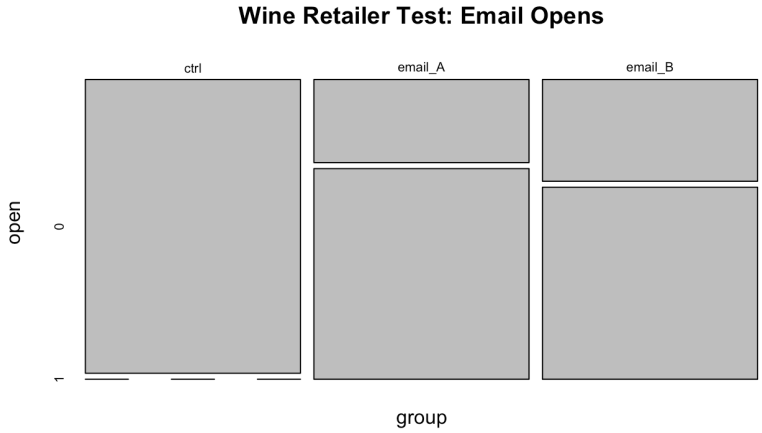
Email A have higher open rate than email B?

```
success <- d %>%
  filter(group!="ctrl") %>%
  group_by(group,open) %>%
  count() %>%
  filter(open==1)
trials <- d %>%
  filter(group!="ctrl") %>%
  count(group)

prop.test(success$n,trials$n)
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  success$n out of trials$n
## X-squared = 385.13, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.05707725 0.06975862
## sample estimates:
##    prop 1    prop 2
## 0.7170994 0.6536814
```

Email A have a higher open rate than email B?



Email A have a higher click rate than email B?

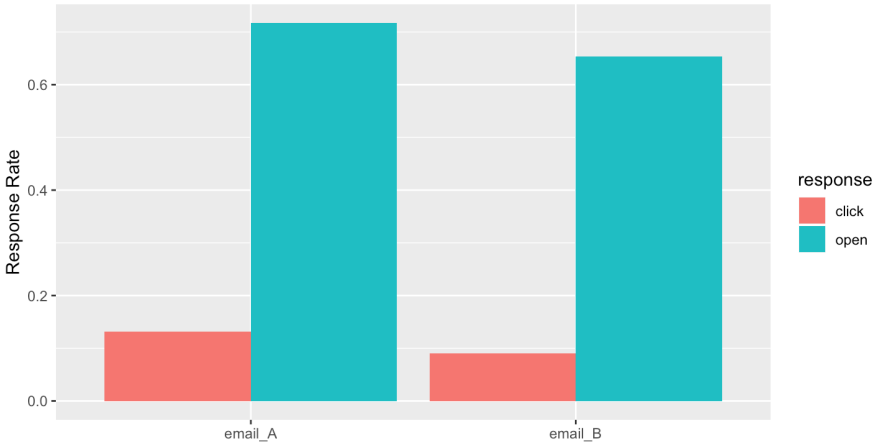
```
success <- d %>%
  filter(group!="ctrl") %>%
  group_by(group,click) %>%
  count() %>%
  filter(click==1)
trials <- d %>%
  filter(group!="ctrl") %>%
  count(group)

prop.test(success$n,trials$n)
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  success$n out of trials$n
## X-squared = 367.2, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.03754391 0.04612615
## sample estimates:
##     prop 1     prop 2
## 0.13152992 0.08969489
```

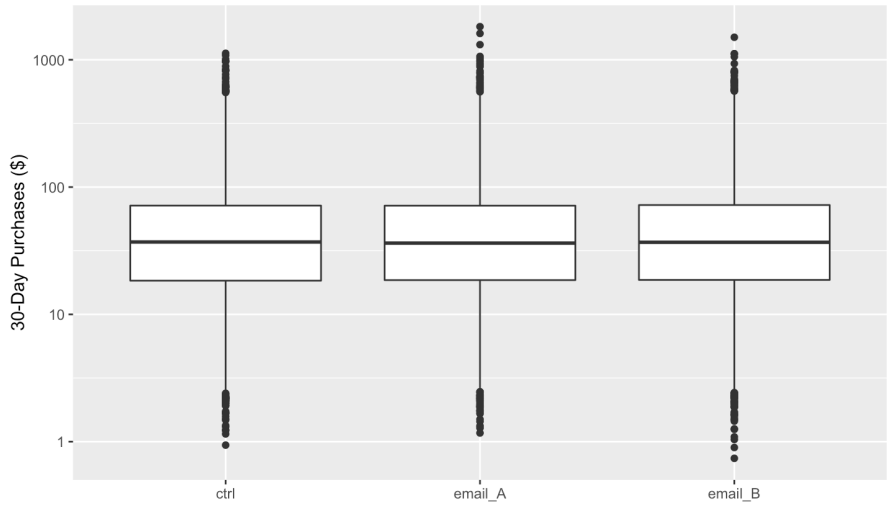
Note that we analyze click rate among all who *received* the email. There may be systematic differences in the types of customers who opened email A versus email B.

Email A have higher opens and clicks than email B?

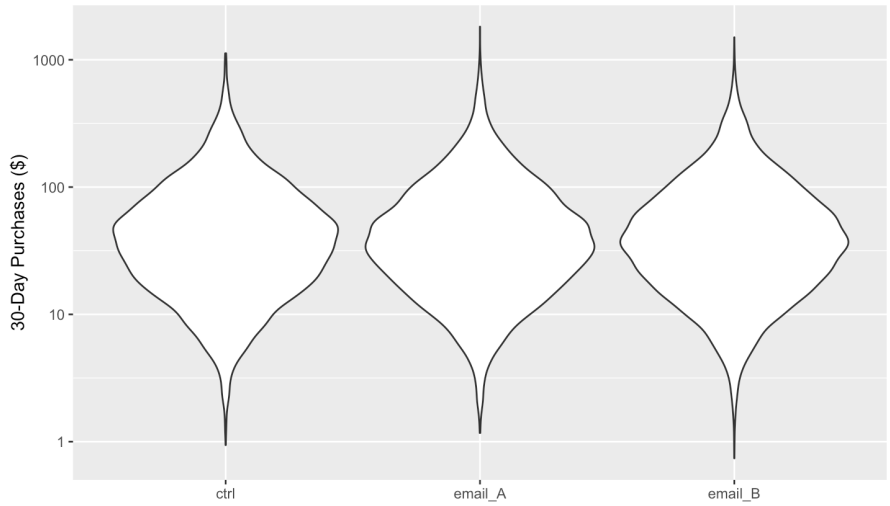


- Email A has a significantly higher open rate than email B. The diference in open rate is between 5.7% and 7.0% (95% CI).
- Email A has a significantly higher click rate than email B. The difference is between 3.8% and 4.6% (95% CI).

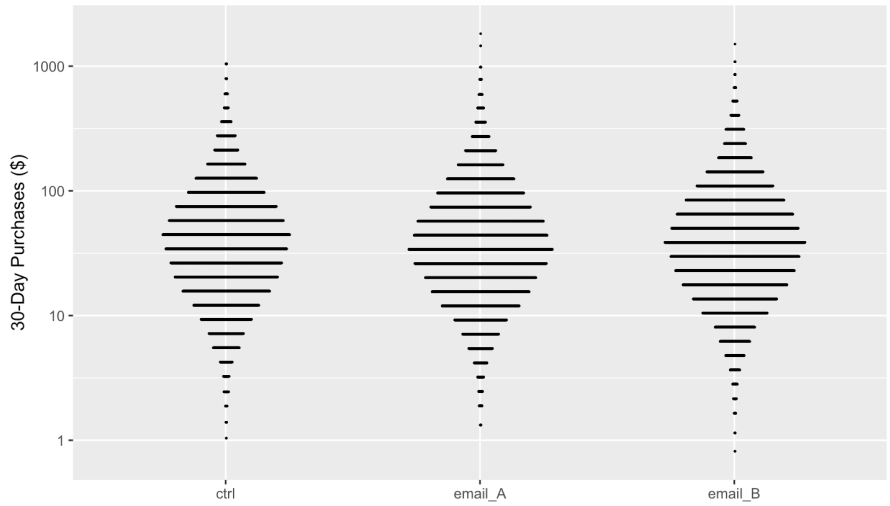
Do any groups have higher average purchases?



Does groups have higher average purchases?



Do groups have higher average purchases?



Email A generate more purchases than B?

```
t.test(purch ~ group, data=d[d$group != "ctrl",])
```

```
##
##  Welch Two Sample t-test
##
## data:  purch by group
## t = 0.41463, df = 82516, p-value = 0.6784
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4936308  0.7585155
## sample estimates:
## mean in group email_A mean in group email_B
##              13.81592              13.68348
```

There is not a significant difference in average purchases between email A and email B.

Do emails generate higher purchases?

```
d$email <- d$group == "email_A" | d$group == "email_B"
t.test(purch ~ email, data=d)
```

```
##
##  Welch Two Sample t-test
##
## data:  purch by email
## t = -5.4485, df = 89037, p-value = 5.093e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.9417432 -0.9143263
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            12.32166            13.74970
```

Those who received an email have $0.91 to $1.94 higher average purchases (95% CI).

Summary of findings (suitable for texting)

- Email A has significantly higher opens and clicks than email B, but purchase are similar for both emails -> Send email A!
- Both emails generate higher average purchases than the control -> Send emails!

Design of A/B tests

Seven key questions

1. Business question
2. Test setting (lab v. field)
3. Unit of analysis (visit, customer, store)
4. Treatments
5. Response variable(s)
6. Selection of units
7. Assignment to treatments
8. Sample size

If you can answer these questions, you have a test plan.

Email test

Business question: Does email work? If so which email is better?

Test setting: email to retailer customers

Unit: email address

Treatments: email version A, email version B, holdout

Reponse: open, click and 30-day purchase ($)

Selection: all active customers

Assignment: randomly assigned (1/3 each)

Sample size: 123,988 emails

Typical website test

Business question: Which version of a page?

Test setting: website (field)

Unit of analysis: visitor (cookie-tracked)

Treatments: versions A and B

Response variable: clicks, conversions

Selection of units: all who visit

Assignment to treatments: random (by testing sw)

Sample size: ???

Sample size planning

Significance tests will erroneously detect effects that aren't there, if you repeatedly test for significance as the data comes in and stop when you get a significant difference.

```
sig <- rep(0, 1000)
for (r in 1:1000) {
  A <- rnorm(101); B <- rnorm(101)
  pval <- rep(NA, 100)
  for (n in 1:100) pval[n] <- t.test(A[1:(n+1)], B[1:(n+1)])$p.value  # repeated testing
  if (min(pval) < 0.05) sig[r] <- 1  # any significance along the way
}
mean(sig)   # bigger than the nominal significance of 5%
```

```
## [1] 0.382
```

Sample size planning

The traditional recommendation is to set the sample size **in advance** and not test for significance until the data comes in.

$$n_1 = n_2 \approx (z_{1-\alpha/2} + z_\beta)^2 \left(\frac{2s^2}{d^2} \right)$$

WTF? Seriously?

Sample size planning: key ideas

- My data is noisy (s), so the group with the higher average in the test is not always *actually* higher.
- There are two mistakes you can make:
  - Declare the treatments different, when they are the same (Type I)
  - Declare the treatment the same, when they are different (Type II)
- I want a low probability of both of those mistakes (α, β) given a specific known difference between treatments (d)

Sample size calculator

Sample size to detect at $1 difference in average 30-day purchases.

```
sd(d$purch)
```

```
## [1] 44.73296
```

```
power.t.test(sd=sd(d$purch), delta=1, sig.level=0.95, power=0.80)
```

```
##
##      Two-sample t test power calculation
##
##               n = 3272.935
##           delta = 1
##              sd = 44.73296
##       sig.level = 0.95
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

We need 3272 in each group. *Note:* Power is 1 minus probability we declare the treatment the same, when they are actually different.

Sample size planning

There is a slightly different formula for

Continous outcomes (eg money, time-on-site)

$$n_1 = n_2 \approx (z_{1-\alpha/2} + z_\beta)^2 \left(\frac{2s^2}{d^2} \right)$$

Binary outcomes (eg conversions)

$$n_1 = n_2 \approx (z_{1-\alpha/2} + z_\beta)^2 \left(\frac{2p(1-p)}{d^2} \right)$$

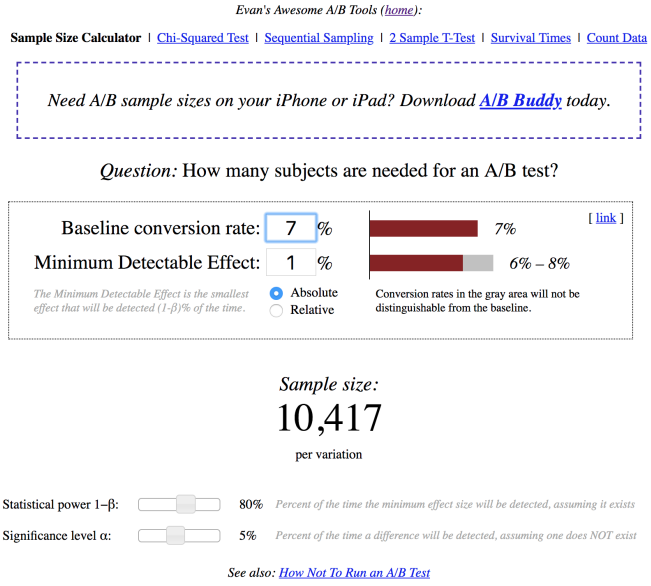
Sample size calculator

Binary outcomes

```
power.prop.test(p1=0.07, p2=0.07 + 0.01, sig.level=0.05, power=0.80)
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 10889.14
##              p1 = 0.07
##              p2 = 0.08
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Sample size calculator



Evan's Awesome A/B Tools (home):

Sample Size Calculator | Chi-Squared Test | Sequential Sampling | 2 Sample T-Test | Survival Times | Count Data

*Need A/B sample sizes on your iPhone or iPad? Download **A/B Buddy** today.*

Question: How many subjects are needed for an A/B test?

Baseline conversion rate: 7 %

Minimum Detectable Effect: 1 %

The Minimum Detectable Effect is the smallest effect that will be detected (1-β)% of the time.

Absolute
Relative

7%

6% – 8%

[link]

Conversion rates in the gray area will not be distinguishable from the baseline.

Sample size:

10,417

per variation

Statistical power 1−β: 80% *Percent of the time the minimum effect size will be detected, assuming it exists*

Significance level α: 5% *Percent of the time a difference will be detected, assuming one does NOT exist*

See also: How Not To Run an A/B Test

A word of caution about sample size calculators

There are a lot of different ideas about sample size calculations floating around. These formulas differ on what assumptions they may about what you are trying to do, but it is very hard to figure out what assumptions are being made (even for experts).

A decent sample size calculation will help you identify whether you are likely to end up with way too much or too little data.

Tips for getting started with A/B testing

- Keep it simple
- Be prepared to find no effect
- Choose “strong” treatments
- Run many tests in fast succession
- You are searching for a few “golden tickets”

Things you just learned (or reviewed)

- Three types of variables in test data
  - Treatment (x's)
  - Response (y's)
  - Baseline variables (z's)
- Analyzing tests with binary outcomes
  - Bar plot or mosaic plot
  - `prop.test()` for significance
- Analyzing tests with continuous outcomes
  - Dot plots or violin plots
  - `t.test()` for significance
- Eight key questions that define a test plan
- Sample size calculations