Inferential Stats with R

Welcome to the course!

Image by [Allison Horst]

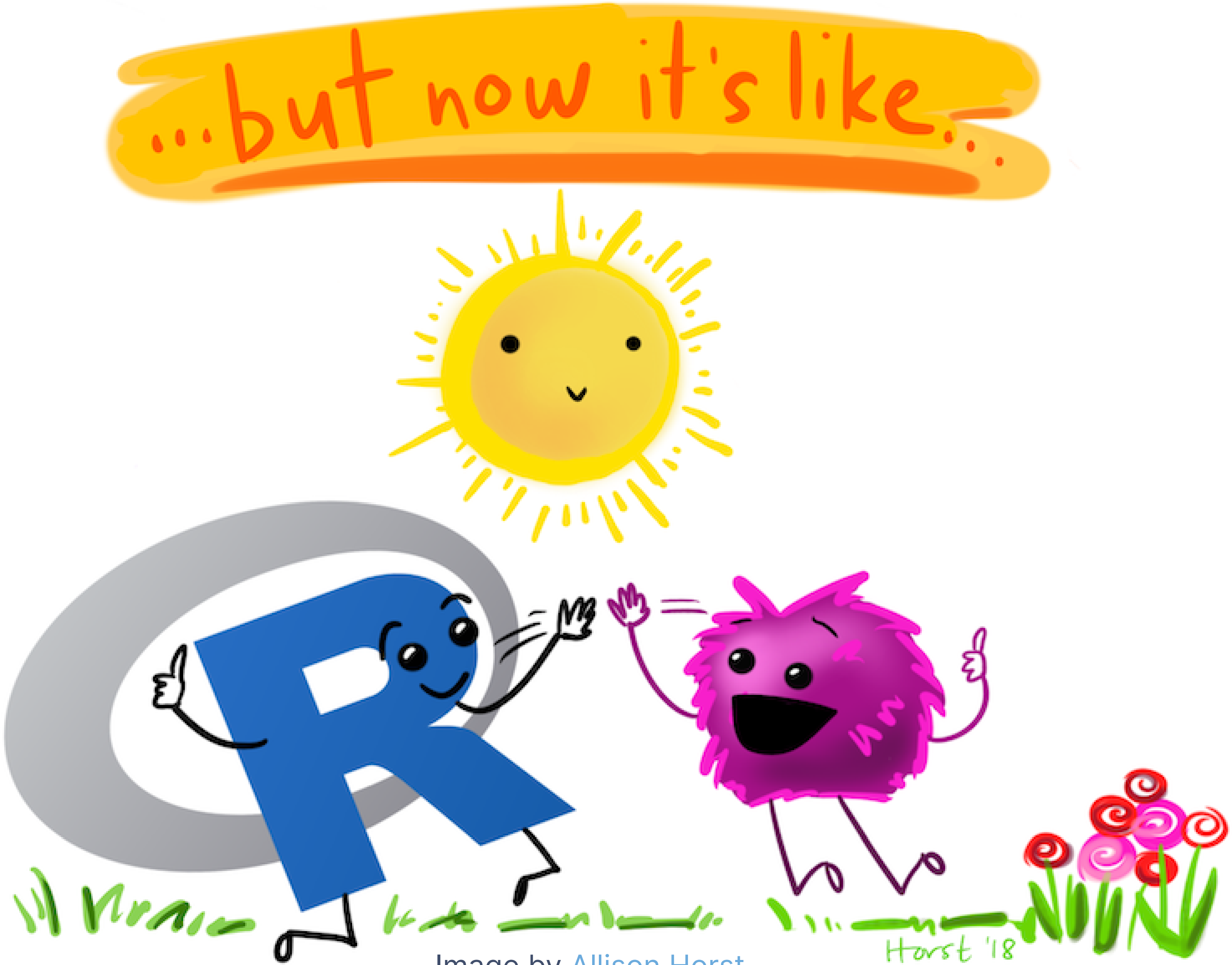


Image by Allison Horst

What you will learn in this course

1. t-tests: independent and dependent
2. ANOVA: mainly the one-way ANOVA
3. chi-square
4. correlation
5. regression
6. reliability (internal consistency)
7. reporting statistics in R
8. testing parametric assumptions

What you will need for this course

1. R and RStudio!

2. Dataset: college.csv

3. Packages:

- tidyverse
- broom
- rstatix
- janitor
- correlation
- lm.beta
- psych
- and more!

A quick overview of tidyverse

There are a few tidyverse functions we'll use:

- `dplyr::select()`
- `dplyr::filter()`
- `dplyr::mutate()`
- `tidyr::pivot_longer()`

How this course will work

- Each video will walk through the topic, showing you relevant R code and output.
- At the end of each video will be an opportunity for you to practice what you've learned.
- You can check your answers to the Your Turn activities on the R for the Rest of Us website.

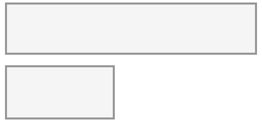
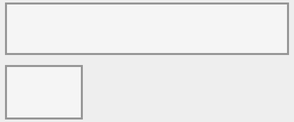
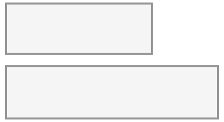
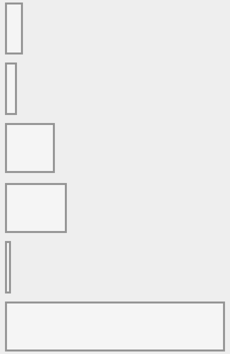
Introduction to the dataset

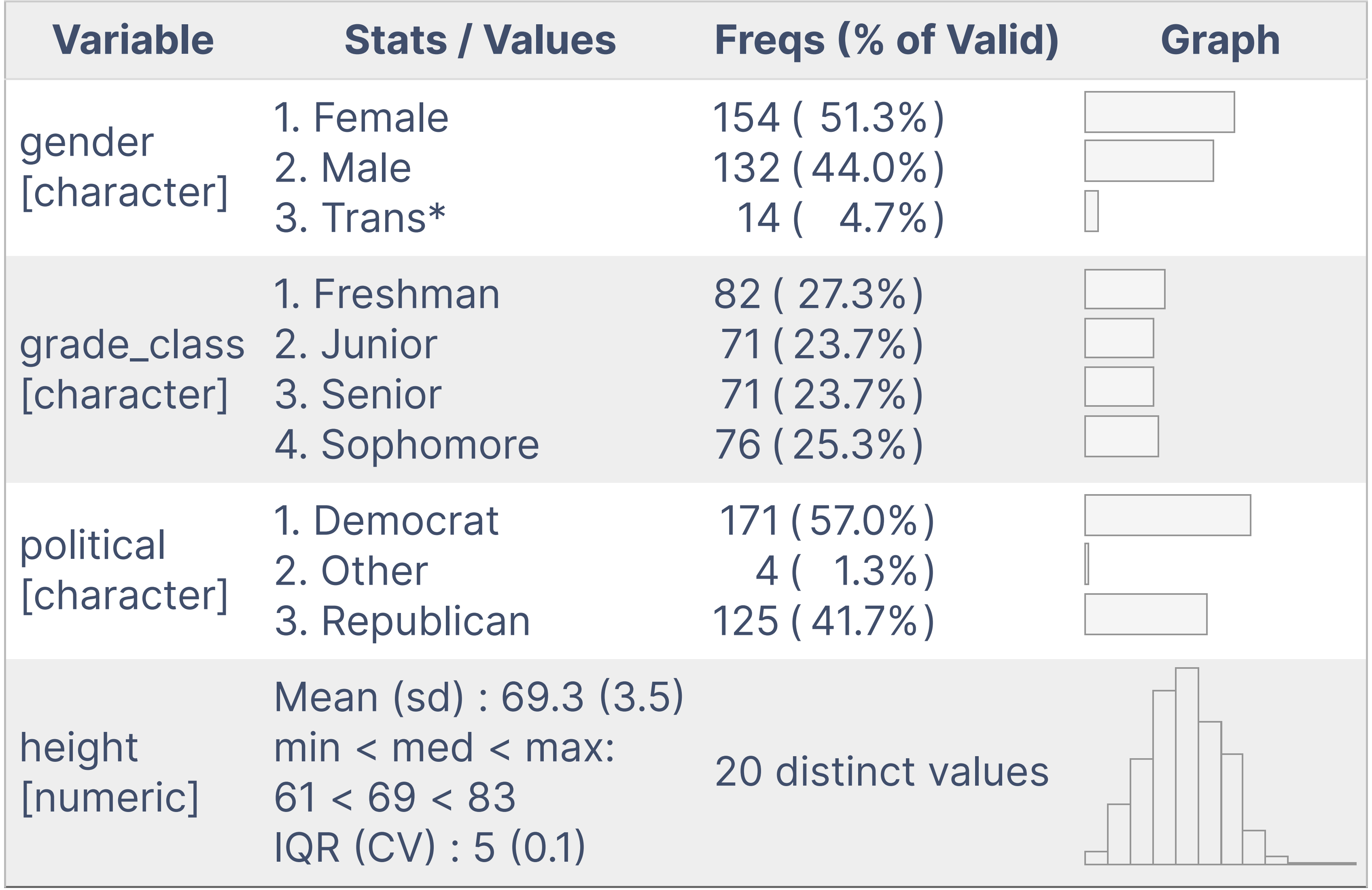
college.csv

```r
college <- read_csv("../data/college.csv")

college
```

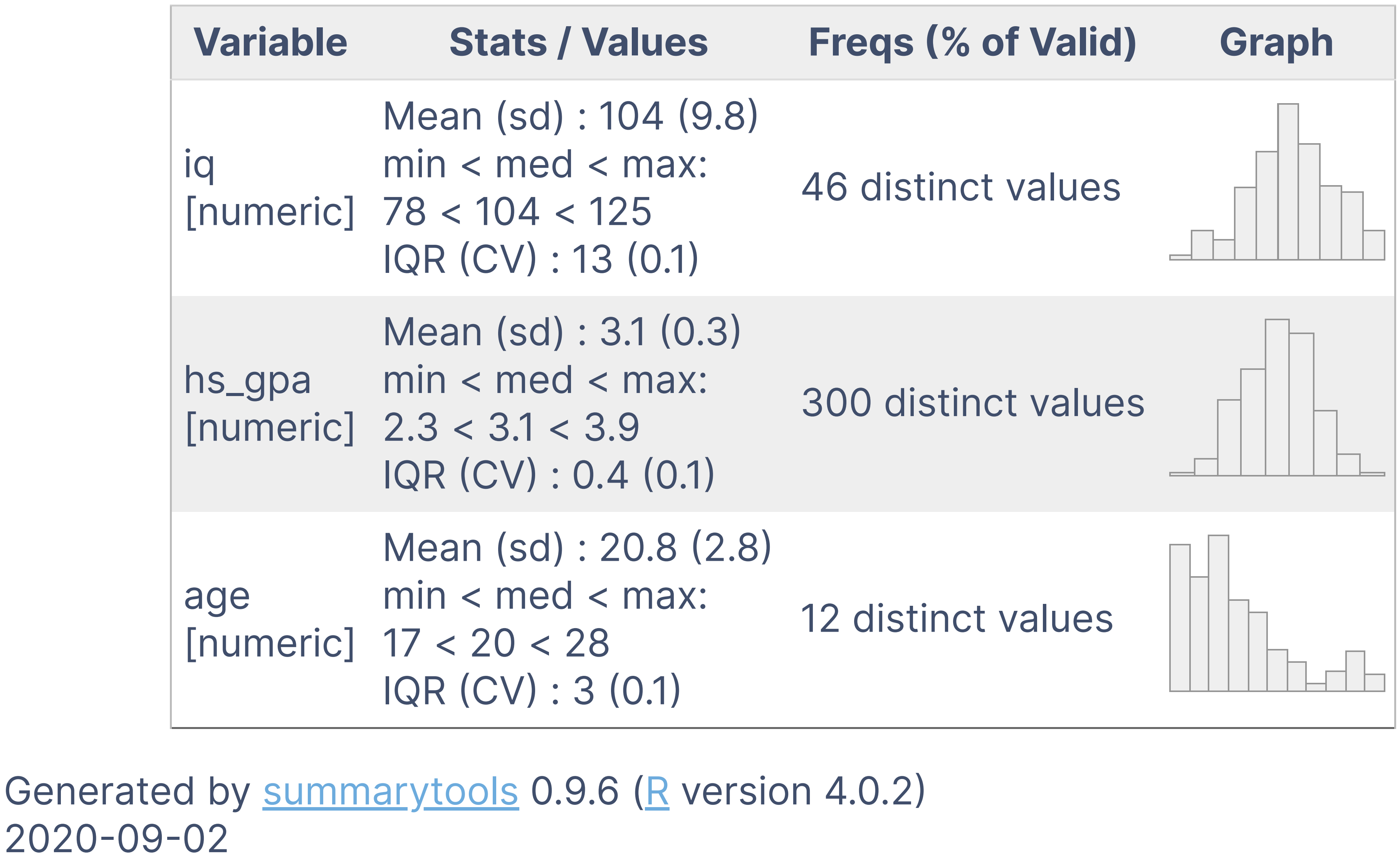
id	athlete	smokes	live_on_campus	race	gender	grade_class
<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>
1	0	0	1	Black	Male	Freshman
2	0	0	0	White	Female	Sophomore
3	1	0	1	White	Male	Senior
4	1	0	1	Hispanic	Male	Freshman
5	0	0	1	White	Male	Junior

1-5 of 300 rows | 1-7 of 31 columns Previous **1** 2 3 4 5 6 ... 60 Next

Variable	Stats / Values	Freqs (% of Valid)	Graph
athlete [numeric]	Min : 0 Mean : 0.3 Max : 1	0: 209 (69.7%) 1: 91 (30.3%)	
smokes [numeric]	Min : 0 Mean : 0.2 Max : 1	0: 236 (78.7%) 1: 64 (21.3%)	
live_on_campus [numeric]	Min : 0 Mean : 0.6 Max : 1	0: 122 (40.7%) 1: 178 (59.3%)	
race [character]	1. Asian 2. Bi-Racial 3. Black 4. Hispanic 5. Native 6. White	13 (4.3%) 9 (3.0%) 40 (13.3%) 51 (17.0%) 4 (1.3%) 183 (61.0%)	

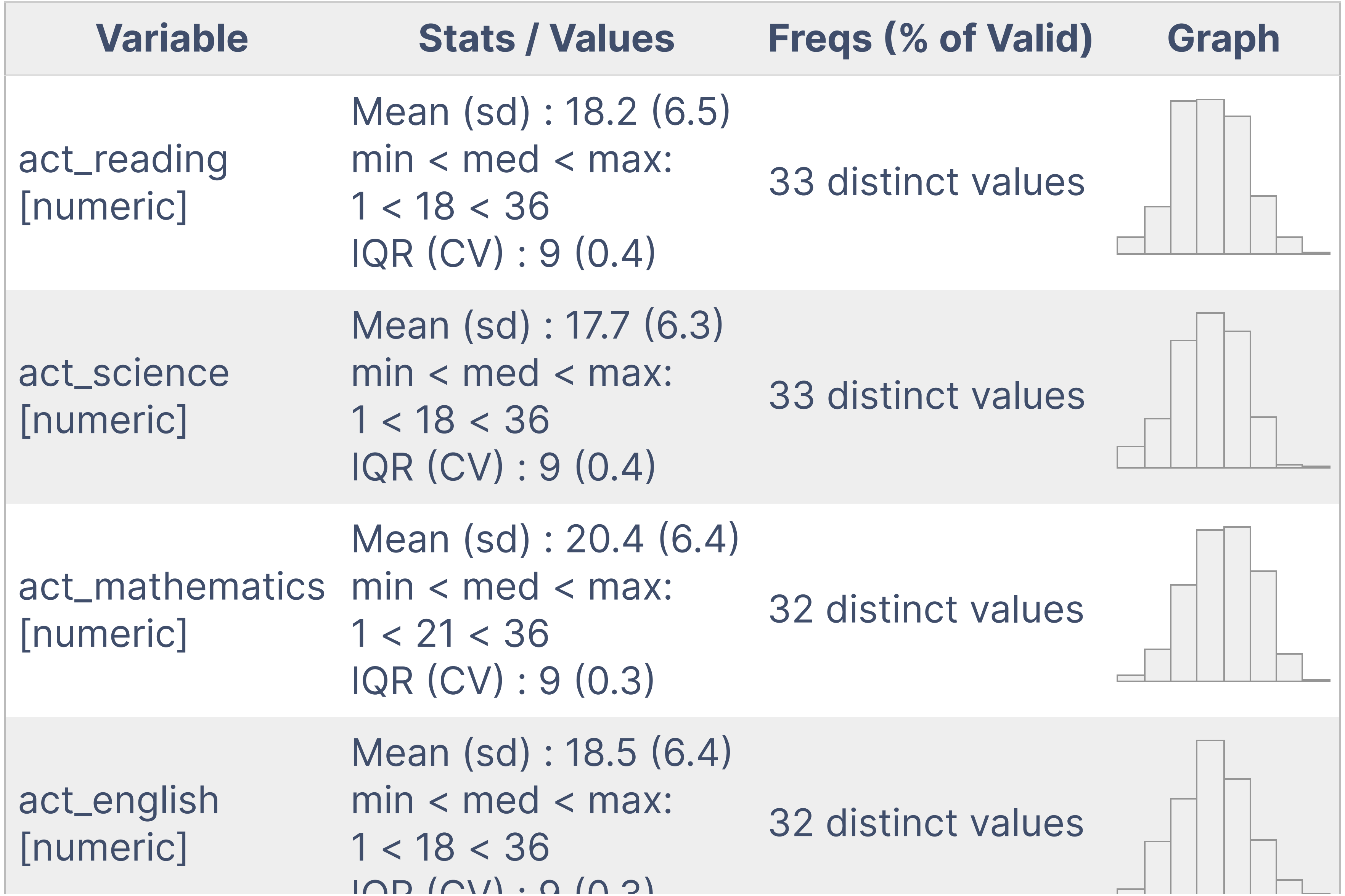


Variable	Stats / Values	Freqs (% of Valid)	Graph
gender [character]	1. Female 2. Male 3. Trans*	154 (51.3%) 132 (44.0%) 14 (4.7%)	
grade_class [character]	1. Freshman 2. Junior 3. Senior 4. Sophomore	82 (27.3%) 71 (23.7%) 71 (23.7%) 76 (25.3%)	
political [character]	1. Democrat 2. Other 3. Republican	171 (57.0%) 4 (1.3%) 125 (41.7%)	
height [numeric]	Mean (sd) : 69.3 (3.5) min < med < max: 61 < 69 < 83 IQR (CV) : 5 (0.1)	20 distinct values	



Variable	Stats / Values	Freqs (% of Valid)	Graph
iq [numeric]	Mean (sd) : 104 (9.8) min < med < max: 78 < 104 < 125 IQR (CV) : 13 (0.1)	46 distinct values	
hs_gpa [numeric]	Mean (sd) : 3.1 (0.3) min < med < max: 2.3 < 3.1 < 3.9 IQR (CV) : 0.4 (0.1)	300 distinct values	
age [numeric]	Mean (sd) : 20.8 (2.8) min < med < max: 17 < 20 < 28 IQR (CV) : 3 (0.1)	12 distinct values	

Generated by summarytools 0.9.6 (R version 4.0.2)
2020-09-02

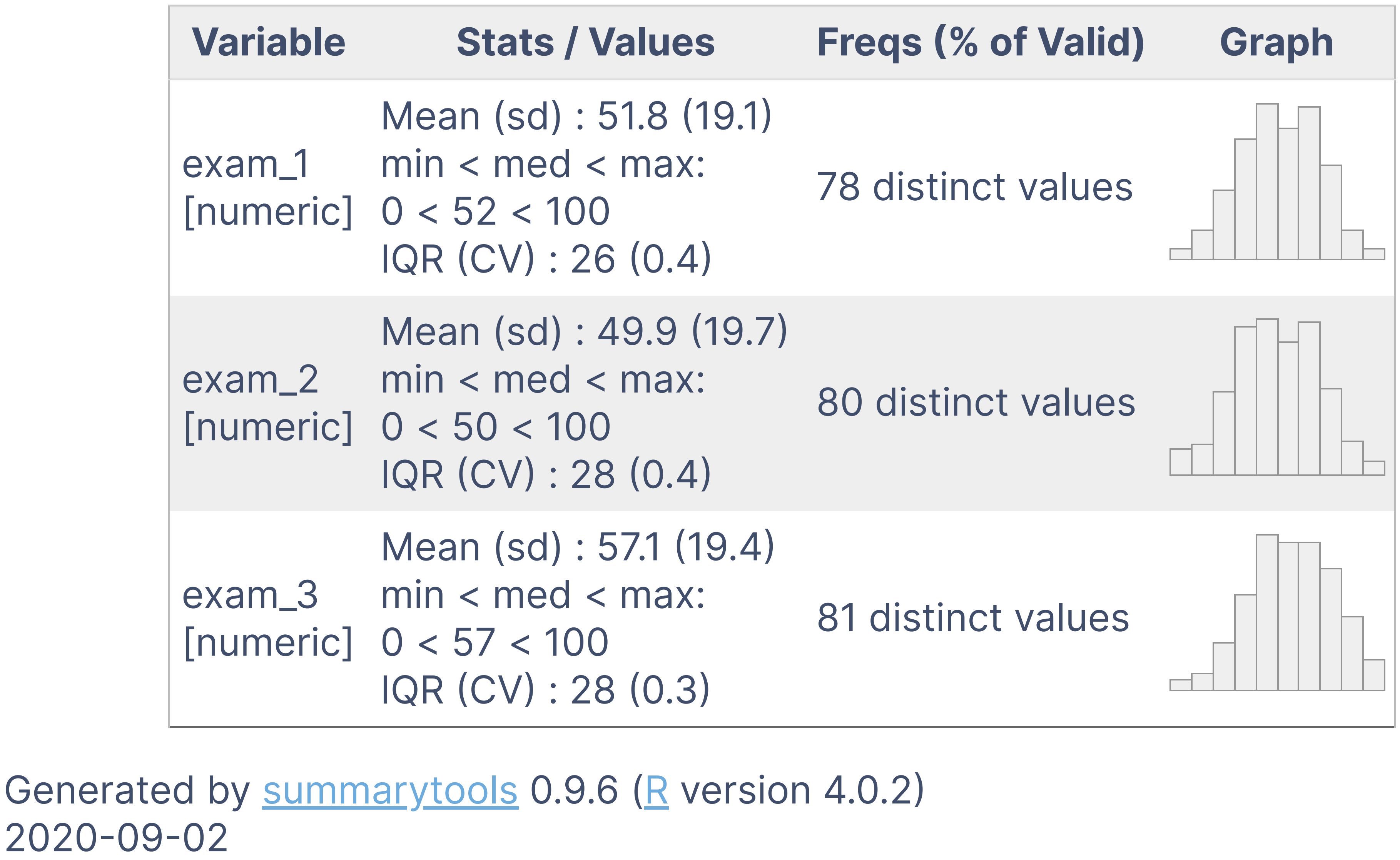


Variable	Stats / Values	Freqs (% of Valid)	Graph
act_reading [numeric]	Mean (sd) : 18.2 (6.5) min < med < max: 1 < 18 < 36 IQR (CV) : 9 (0.4)	33 distinct values	
act_science [numeric]	Mean (sd) : 17.7 (6.3) min < med < max: 1 < 18 < 36 IQR (CV) : 9 (0.4)	33 distinct values	
act_mathematics [numeric]	Mean (sd) : 20.4 (6.4) min < med < max: 1 < 21 < 36 IQR (CV) : 9 (0.3)	32 distinct values	
act_english [numeric]	Mean (sd) : 18.5 (6.4) min < med < max: 1 < 18 < 36 IQR (CV) : 9 (0.3)	32 distinct values	

Satisfaction with Life Scale

variable <chr>	mean <dbl>	min <dbl>	max <dbl>
swls_time1_1	2.920000	1	5
swls_time1_2	2.906667	1	5
swls_time1_3	2.913333	1	5
swls_time1_4	2.950000	1	5
swls_time1_5	2.896667	1	5
swls_time2_1	2.943333	1	5
swls_time2_2	2.970000	1	5
swls_time2_3	2.973333	1	5
swls_time2_4	3.026667	1	5
swls_time2_5	2.950000	1	5

1-10 of 10 rows



Variable	Stats / Values	Freqs (% of Valid)	Graph
exam_1 [numeric]	Mean (sd) : 51.8 (19.1) min < med < max: 0 < 52 < 100 IQR (CV) : 26 (0.4)	78 distinct values	
exam_2 [numeric]	Mean (sd) : 49.9 (19.7) min < med < max: 0 < 50 < 100 IQR (CV) : 28 (0.4)	80 distinct values	
exam_3 [numeric]	Mean (sd) : 57.1 (19.4) min < med < max: 0 < 57 < 100 IQR (CV) : 28 (0.3)	81 distinct values	

Generated by summarytools 0.9.6 (R version 4.0.2)
2020-09-02

Your Turn

You'll be working with the college dataset to run all your analyses.

1. Create a new project. Make sure you put it somewhere you'll be able to find it again later!
2. Download the dataset "college.csv" from https://bit.ly/college-dataset
3. Create a new R script file or RMarkdown document where you'll do all of your inferential statistics
4. Import the spreadsheet into a dataframe `college` using `readr::read_csv()`

Independent t-test

Independent vs dependent t-test

Independent t-test

- Between-subjects
- Independent samples
- Examine the mean difference between two unrelated groups (e.g., males and females, a group of mothers and a group of daughters)

Dependent t-test

- Within-subjects
- Dependent or paired samples
- Examine the mean difference between two related groups (e.g., same group at two time points, pairs of mothers and daughters)

Both t-tests are performed using the `t_test()` function in the **`rstatix`** package.

Independent t-test in R

```
dat %>%
  t_test(DV ~ IV)
```

An example

Let's try running an independent samples t-test with our `college` dataset.

Let's test whether weight at time 2 differs by smoking status.

```
dat %>%
  t_test(DV ~ IV)
```

```
college %>%
  t_test(weight_2 ~ smokes)
```

	.y.	group1	group2	n1	n2	statistic	df	p
	<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>
1	weight_2	0	1	236	64	0.1216377	102.6969	0.903

1 row

Notice the degrees of freedom has decimals. This is because the `t_test()` function defaults to var.equal = FALSE.

Variances equal

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE)
```

	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	df <dbl>	p <dbl>
1	weight_2	0	1	236	64	0.1191742	298	0.905

1 row

Detailed t-test results

You can also asked for detailed results.

```
college %>%
  t_test(weight_2 ~ smokes,
         detailed = TRUE)
```

	estimate	estimate1	estimate2	.y.	group1	group2	n1	n2
	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<int>	<int>
1	0.2473517	175.9661	175.7188	weight_2	0	1	236	64

1 row | 1-9 of 16 columns

Alternative hypothesis

The default alternative hypothesis is a **two-sided hypothesis** that the mean difference is not equal to 0.

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE, alternative = "two.sided")
```

What if you have a **one-sided hypothesis**?

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE, alternative = "greater")
```

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE, alternative = "less")
```

Alternative hypothesis

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE, detailed = TRUE, alternative = "two.sided")
```

```
     estimate  estimate1  estimate2 .y.        group1  group2   n1   n2
        <dbl>      <dbl>      <dbl> <chr>      <chr>   <chr>  <int><int>
1   0.2473517   175.9661   175.7188 weight_2   0       1       236   64
1 row | 1-9 of 16 columns
```

```
college %>%
  t_test(weight_2 ~ smokes, var.equal = TRUE, detailed = TRUE, alternative = "greater")
```

```
     estimate  estimate1  estimate2 .y.        group1  group2   n1   n2
        <dbl>      <dbl>      <dbl> <chr>      <chr>   <chr>  <int><int>
1   0.2473517   175.9661   175.7188 weight_2   0       1       236   64
1 row | 1-9 of 16 columns
```

Other arguments in the t-test

```
college %>%
  t_test(weight_2 ~ smokes, detailed = TRUE, mu = 1)
```

	estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>
1	0.2473517	175.9661	175.7188	weight_2	0	1	236	64

1 row | 1-9 of 16 columns

```
college %>%
  t_test(weight_2 ~ smokes, detailed = TRUE, conf.level = .99)
```

	estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>
1	0.2473517	175.9661	175.7188	weight_2	0	1	236	64

1 row | 1-9 of 16 columns

Your Turn

1. Perform an independent samples t-test to test whether there is a difference in `exam_1` by `athlete`. Use `var.equal = TRUE`. Is there a difference? What is the p-value?

2. Perform an independent samples t-test to test whether there is a difference in `act_english` by `gender`. Use `var.equal = TRUE`. Is there a difference? What is the p-value?

Notice what happens since gender has 3 levels. Try out using the `filter()` function from dplyr to only check the difference between Female and Male students. Alternatively, check out the page on `t_test()` to see how to use the `comparisons` argument to specify the groups to compare.

Dependent t-test

Independent vs dependent t-test

Independent t-test

- Between-subjects
- Independent samples
- Examine the mean difference between two unrelated groups (e.g., males and females, a group of mothers and a group of daughters)

Dependent t-test

- Within-subjects
- Dependent or paired samples
- Examine the mean difference between two related groups (e.g., same group at two time points, pairs of mothers and daughters)

Dependent t-test in R

The dependent t-test adds a new argument to the `t_test()` function:

`paired = TRUE`

```
dat %>%
  t_test(DV ~ IV, paired = TRUE)
```

Converting wide to long format

Let's test the difference in means between `weight_1` and `weight_2`.

```
college %>%
  select(weight_1, weight_2) %>%
  pivot_longer(cols = everything(),
               names_prefix = "weight_",
               names_to = "time",
               values_to = "weight",)
```

time	weight
<chr>	<dbl>
1	156
2	158
1	176
2	179
1	169

Dependent t-test in R

```r
college %>%
  select(weight_1, weight_2) %>%
  pivot_longer(cols = everything(),
               names_to = "time",
               values_to = "weight",
               names_prefix = "weight_") %>%
  t_test(weight ~ time, paired = TRUE)
```

	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	df <dbl>	p <dbl>
1	weight	1	2	300	300	-18.73704	299	2.35e-52

1 row

Your Turn

1. Perform a dependent samples t-test to test whether there is a difference in exam scores: `exam_1` and `exam_2`. Is there a difference? What is the p-value?
2. Perform a dependent samples t-test to test whether there is a difference in `act_science` and `act_mathematics` scores. Is there a difference? What is the p-value?

Don't forget: You'll need to make your data long before running your t-test.

One-way ANOVA

One-way ANOVA

- Between-subjects
- Independent samples
- Examine the mean difference between **three or more** unrelated groups
- Tests the null hypothesis that the three or more means are the same
- To know *where* the mean differences are, we need planned contrasts or post-hoc procedures like Tukey's HSD (we'll cover that in the next video)

One-way ANOVA in R

The formatting for the one-way ANOVA is similar to that of `t_test()` except we replace it with `anova_test`.

```
dat %>%
  anova_test(DV ~ IV)
```

An example

Let's try running an one-way ANOVA with our `college` dataset.

Let's test whether exam 1 scores differ by grade rank (`grade_class`).

```
dat %>%
  anova_test(DV ~ IV)
```

```
college %>%
  anova_test(exam_1 ~ grade_class)
```

	Effect <chr>	DFn <dbl>	DFd <dbl>	F <dbl>	p <dbl>	p<.05 <chr>	ges <dbl>
1	grade_class	3	296	0.716	0.543		0.007

1 row

Detailed ANOVA results

```
college %>%
  anova_test(exam_1 ~ grade_class)
```

	Effect	DFn	DFd	F	p	p<.05	ges
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	grade_class	3	296	0.716	0.543		0.007

1 row

```
college %>%
  anova_test(exam_1 ~ grade_class, detailed = TRUE)
```

	Effect	SSn	SSd	DFn	DFd	F	p	p<.05	ges
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	grade_class	788.871	108684.9	3	296	0.716	0.543		0.007

1 row

hccm???

```
college %>%
  anova_test(exam_1 ~ grade_class)
```

```
## Coefficient covariances computed by hccm()
```

	Effect	DFn	DFd	F	p	p<.05	ges
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	grade_class	3	296	0.716	0.543		0.007

1 row

The warning `Coefficient covariances computed by hccm()` just tells us that it is using a heteroscedasticity-corrected coefficient covariance matrix.

Nothing for us to worry about!

Effect sizes

`ges` is generalized eta squared. We can change `effect.size = "pes"` for partial eta squared.

```
college %>%
  anova_test(exam_1 ~ grade_class,
             effect.size = "ges")
```

```
college %>%
  anova_test(exam_1 ~ grade_class,
             effect.size = "pes")
```

	Effect <chr>	DFn <dbl>	DFd <dbl>	F <dbl>
1	grade_class	3	296	0.716

1 row | 1-5 of 8 columns

	Effect <chr>	DFn <dbl>	DFd <dbl>	F <dbl>
1	grade_class	3	296	0.716

1 row | 1-5 of 8 columns

Your Turn

1. Perform a one-way ANOVA to test whether there is a difference in `hs_gpa` by `grade_class`. Is there a difference? What is the p-value?
2. Perform a one-way ANOVA to test whether there is a difference in `act_english` by `race`. Is there a difference? What is the p-value?

Post hoc comparisons

Post hoc comparisons

The one-way ANOVA tests the null hypothesis that there is no mean difference between the 3+ groups. If the one-way ANOVA is statistically significant, it does not tell us *where* the differences lie.

For that, we need post hoc comparisons.*

There are many post hoc comparisons possible (e.g., Fisher's LSD, Bonferroni, Newman-Keuls, Scheffe), but we're going to focus on **Tukey's HSD** because it and does a good job protecting against Type I errors without being too conservative.

Tukey's HSD

```
dat %>%
  tukey_hsd(DV ~ IV)
```

An example

Using our previous example, let's run Tukey's HSD by testing whether exam 1 scores differ by grade rank (`grade_class`).

```
dat %>%
  tukey_hsd(DV ~ IV)
```

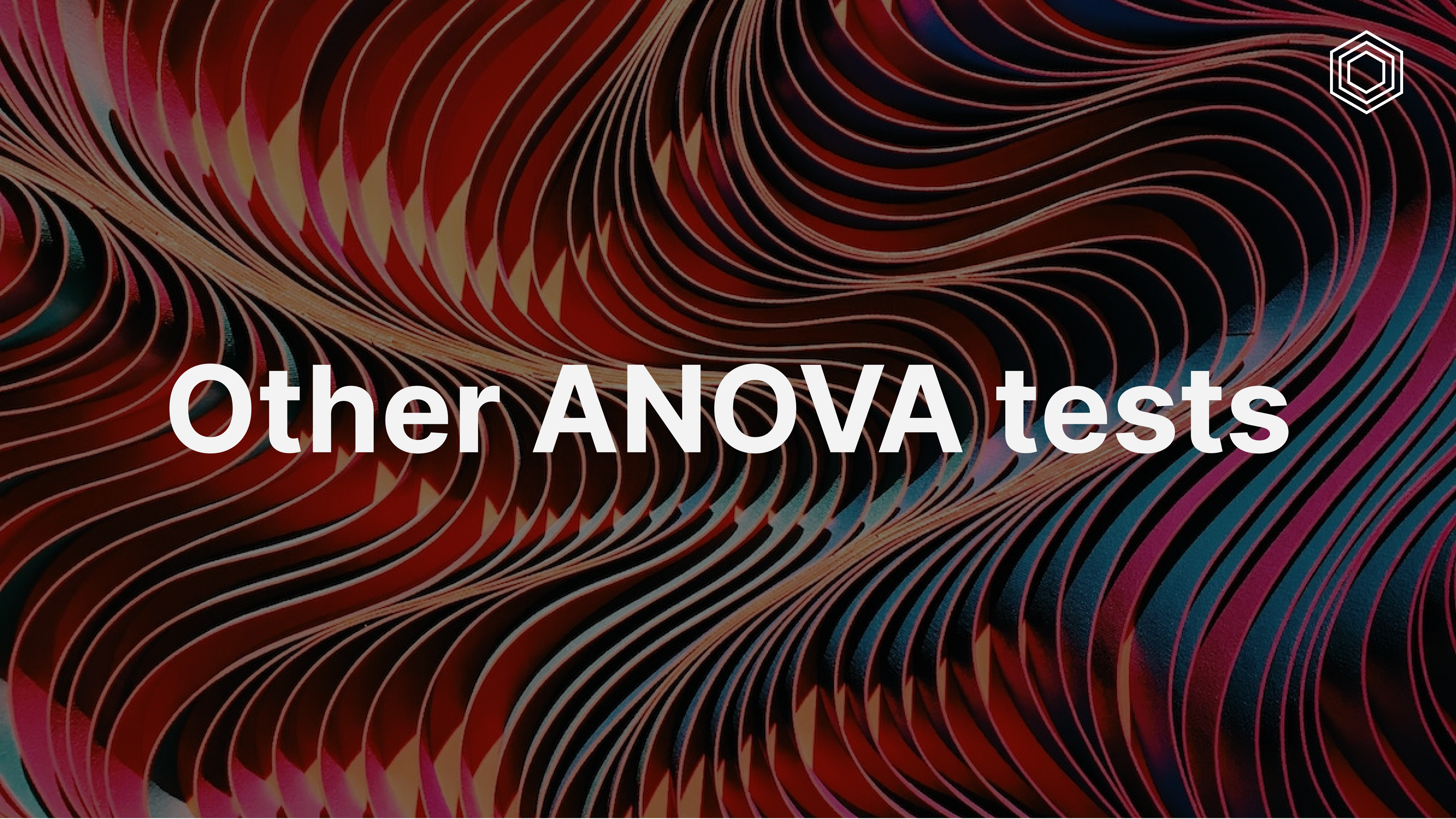
```
college %>%
  tukey_hsd(exam_1 ~ grade_class)
```

	term <chr>	group1 <chr>	group2 <chr>	null.value <dbl>	estimate <dbl>
1	grade_class	Freshman	Junior	0	-0.39591206
2	grade_class	Freshman	Senior	0	3.02662315
3	grade_class	Freshman	Sophomore	0	3.06739409
4	grade_class	Junior	Senior	0	3.42253521
5	grade_class	Junior	Sophomore	0	3.46330615
6	grade_class	Senior	Sophomore	0	0.04077094

Your Turn

1. Perform one-way ANOVA to test whether there is a difference in `hs_gpa` by `grade_class`. Is there a difference? If so, perform a Tukey's HSD to test where the differences are. Describe the differences.

2. Perform one-way ANOVA to test whether there is a difference in `weight_2` by `gender`. Is there a difference? If so, perform a Tukey's HSD to test where the differences are. Describe the differences.



Other ANOVA tests

Repeated measures ANOVA

Repeated Measures ANOVA allows us to examine designs in which the same people or samples contributed to the different means (within-groups design).

This means that all participants or cases participated in all conditions of an experiment or IV.

Repeated measures ANOVA in R

There are two ways to write the R code for a repeated measures ANOVA using `anova_test`: by the equation, or by specifying which variables are the DV, between- or within-subjects IVs, and wid (within-subjects identifier).

Option 1:

```
dat %>%
  anova_test(DV ~ IV + Error(id/IV))
```

Option 2:

```
dat %>%
  anova_test(dv = DV, wid = id, within = IV)
```

An example

Let's examine how exam scores (`exam_1`, `exam_2`, and `exam_3`) differ within individuals across time.

Notice that we'll need to make our data into *long format* to make this work, since we need the DV into into variable.

pivot_longer:

```
college %>%
  select(starts_with("exam"), id) %>%
  pivot_longer(-id,
               names_to = "exam",
               values_to = "score",
               names_prefix = "exam_")
```

An example: Option 1

```r
college %>%
  select(exam_1:exam_3, id) %>%
  pivot_longer(-id, names_to = "exam", values_to = "score", names_prefix = "exam_") %>%
  anova_test(score ~ exam + Error(id/exam))
```

```
## ANOVA Table (type III tests)
##
## $ANOVA
##   Effect DFn DFd      F        p p<.05   ges
## 1   exam   2 598 87.773 3.77e-34     * 0.024
##
## $Mauchly's Test for Sphericity
##   Effect     W        p p<.05
## 1   exam 0.771 1.58e-17     *
##
## $Sphericity Corrections
##   Effect   GGe      DF[GG]    p[GG] p[GG]<.05   HFe       DF[HF]    p[HF]
## 1   exam 0.814 1.63, 486.7 2.58e-28         * 0.818 1.64, 488.99 1.96e-28
##   p[HF]<.05
## 1         *
```

An example: Option 2

```
college %>%
  select(exam_1:exam_3, id) %>%
  pivot_longer(-id, names_to = "exam", values_to = "score", names_prefix = "exam_") %>%
  anova_test(dv = score, wid = id, within = exam)
```

```
## ANOVA Table (type III tests)
##
## $ANOVA
##   Effect DFn DFd      F        p p<.05   ges
## 1   exam   2 598 87.773 3.77e-34     * 0.024
##
## $Mauchly's Test for Sphericity
##   Effect     W        p p<.05
## 1   exam 0.771 1.58e-17     *
##
## $Sphericity Corrections
##   Effect   GGe      DF[GG]    p[GG] p[GG]<.05   HFe        DF[HF]    p[HF]
## 1   exam 0.814 1.63, 486.7 2.58e-28         * 0.818 1.64, 488.99 1.96e-28
##   p[HF]<.05
## 1         *
```

Factorial ANOVA

Factorial ANOVA allows us to examine two or more independent variables (IVs) simultaneously. There are several types of factorial designs:

- **Independent factorial design**: 2+ between-group (independent) IVs
- **Repeated measures factorial design**: 2+ within-group (repeated-measures) IVs
- **Mixed factorial design**: 1+ between-group and 1+ within-group IVs

Factorial ANOVA

To perform a factorial ANOVA in R, all you need to do is add more variables to the right hand of the tilde (~) in the ANOVA equation using either a + (does not add an interaction term) or * (adds an interaction term).

Example independent factorial design

Formula without interaction:

```
college %>%
  anova_test(hs_gpa ~ grade_class + smokes) # NO INTERACTION
```

	Effect	DFn	DFd	F	p	p<.05	ges
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	grade_class	3	295	0.841	0.473		0.008000
2	smokes	1	295	0.157	0.692		0.000533

2 rows

Example independent factorial design

Formula with interaction:

```
college %>%
  anova_test(hs_gpa ~ grade_class * smokes) #  INTERACTION
```

	Effect	DFn	DFd	F	p	p<.05	ges
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	grade_class	3	292	0.843	0.471		0.00900
2	smokes	1	292	0.158	0.692		0.00054
3	grade_class:smokes	3	292	1.271	0.284		0.01300

3 rows

Your Turn

Read more about how to conduct repeated measures and factorial ANOVA designs using the resources on the R for the Rest of Us website.

Chi-square

Chi-square

The chi-square test is used to analyze categorical data (data that describe categories of entities). We can't examine categorical data using things like means or medians, so we examine counts instead.

```
college %>%
  tabyl(smokes, athlete)
```

smokes	**0**	**1**
<dbl>	<dbl>	<dbl>
0	156	80
1	53	11

2 rows

Chi-square in R: rstatix

```
table(college$smokes, college$athlete) %>%
  chisq_test()
```

```
        n   statistic        p    df method              p.signif
    <int>       <dbl>    <dbl> <int> <chr>               <chr>
1     300    5.885766   0.0153     1 Chi-square test     *
1 row
```

Chi-square in R: janitor

```
college %>%
  tabyl(smokes, athlete) %>%
  janitor::chisq.test()
```

```
##
##      Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 5.8858, df = 1, p-value = 0.01526
```

Solution! Use the `tidy()` function in the *broom* package!

Image by [Allison Horst]

Chi-square in R: janitor

```
college %>%
  tabyl(smokes, athlete) %>%
  janitor::chisq.test() %>%
  tidy()
```

statistic <dbl>	p.value <dbl>	parameter <int>
5.885766	0.01526375	1

1 row | 1-3 of 4 columns

Standardized residuals

The standardized residuals tell us for that cell if the expected and observed frequency are independent. They are z-scores so values greater than 1.96 are statistically significant (*p* < .05).

```r
tab <- college %>%
  tabyl(smokes, athlete) %>%
  janitor::chisq.test()

tab$stdres
```

	smokes	**0**	**1**
	<chr>	<dbl>	<dbl>
0	0	-2.579349	2.579349
1	1	2.579349	-2.579349

2 rows

Observed and expected counts

Similarly, we can ask for the observed and expected counts.

Observed

```r
tab$observed
```

	smokes	0	1
	<chr>	<dbl>	<dbl>
0	0	156	80
1	1	53	11

2 rows

Expected

```r
tab$expected
```

	smokes	0	1
	<chr>	<dbl>	<dbl>
0	0	164.41333	71.58667
1	1	44.58667	19.41333

2 rows

Your Turn

1. Perform a chi-square to examine how `grade_class` relates to `live_on_campus`. What is the p-value? Is there a relationship?
2. If there is a significant difference, examine standardized residuals and the observed/expected frequencies to determine what grade class is more or less likely to live on campus. Interpret the results.

Dealing with small cells

Dealing with small cells

Sometimes when dealing with small cells, you will get an error that the Chi-square approximation may not be correct.

```
college %>%
  filter(smokes == 1) %>%
  tabyl(race, gender) %>%
  chisq.test()
```

```
## Warning in stats::chisq.test(., ...): Chi-squared approximation may be incorrect
```

```
##
##      Pearson's Chi-squared test
##
## data:  .
## X-squared = 5.2355, df = 8, p-value = 0.7321
```

There are a few things we can do to fix this problem.

1. Drop a row or column

```
college %>%
  filter(smokes == 1) %>%
  tabyl(race, gender)
```

race <chr>	Female <dbl>	Male <dbl>	Trans* <dbl>
Asian	3	1	0
Bi-Racial	2	1	0
Black	5	1	0
Hispanic	7	3	1
White	20	19	1

1-5 of 5 rows

1. Drop a row or column

```
college %>%
  filter(smokes == 1) %>%
  filter(gender != "Trans*") %>%
  tabyl(race, gender)
```

race	Female	Male
<chr>	<dbl>	<dbl>
Asian	3	1
Bi-Racial	2	1
Black	5	1
Hispanic	7	3
White	20	19

1-5 of 5 rows

1. Drop a row or column

```
college %>%
  filter(smokes == 1) %>%
  filter(gender != "Trans*") %>%
  tabyl(race, gender) %>%
  chisq.test()
```

```
## Warning in stats::chisq.test(., ...): Chi-squared approximation may be incorrect

##
##      Pearson's Chi-squared test
##
## data:  .
## X-squared = 3.4316, df = 4, p-value = 0.4884
```

2. Combine rows or columns so there are larger cells

```
college %>%
  filter(smokes == 1) %>%
  tabyl(race, gender)
```

race <chr>	Female <dbl>	Male <dbl>	Trans* <dbl>
Asian	3	1	0
Bi-Racial	2	1	0
Black	5	1	0
Hispanic	7	3	1
White	20	19	1

1-5 of 5 rows

2. Combine rows or columns so there are larger cells

```
college %>%
  filter(smokes == 1) %>%
  mutate(URM = recode(race,
                      "Bi-Racial" = 1,
                      "Asian" = 0,
                      "White" = 0,
                      "Black" = 1,
                      "Hispanic" = 1)) %>%
  tabyl(URM, gender)
```

URM	Female	Male	Trans*
<dbl>	<dbl>	<dbl>	<dbl>
0	23	20	1
1	14	5	1

2 rows

2. Combine rows or columns so there are larger cells

```
college %>%
  filter(smokes == 1) %>%
  mutate(URM = recode(race,
                      "Bi-Racial" = 1,
                      "Asian" = 0,
                      "White" = 0,
                      "Black" = 1,
                      "Hispanic" = 1)) %>%
  tabyl(URM, gender) %>%
  chisq.test()
```

```
## 
##      Pearson's Chi-squared test
## 
## data:  .
## X-squared = 2.5474, df = 2, p-value = 0.2798
```

3. Both!

```
college %>%
  filter(smokes == 1) %>%
  filter(gender != "Trans*") %>%
  mutate(URM = recode(race,
                      "Bi-Racial" = 1,
                      "Asian" = 0,
                      "White" = 0,
                      "Black" = 1,
                      "Hispanic" = 1)) %>%
  tabyl(URM, gender)
```

URM	Female	Male
<dbl>	<dbl>	<dbl>
0	23	20
1	14	5

2 rows

3. Both!

```
college %>%
  filter(smokes == 1) %>%
  filter(gender != "Trans*") %>%
  mutate(URM = recode(race,
                      "Bi-Racial" = 1,
                      "Asian" = 0,
                      "White" = 0,
                      "Black" = 1,
                      "Hispanic" = 1)) %>%
  tabyl(URM, gender) %>%
  chisq.test()
```

```
##
##      Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 1.4731, df = 1, p-value = 0.2249
```

Correlation

Correlation

Tests the relationship between two (usually) continuous variables.

Correlation in R

```
dat %>%
  cor_test(var1, var2)
```

An example

Let's look at the correlation between `exam_1` and `exam_2`.

```
dat %>%
  cor_test(var1, var2)
```

```
college %>%
  cor_test(exam_1, exam_2)
```

var1	var2	cor	statistic	p	conf.low	conf.high	method
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
exam_1	exam_2	0.9	34.93659	2.17e-107	0.8717947	0.9167018	Pearson

1 row

Multiple correlations in R

```
college %>%
  cor_test(starts_with("exam"))
```

var1	var2	cor	statistic	p	conf.low	conf.high
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
exam_1	exam_1	1.00	Inf	0.00e+00	1.0000000	1.0000000
exam_1	exam_2	0.90	3.493659e+01	2.17e-107	0.8717947	0.9167018
exam_1	exam_3	0.81	2.397949e+01	1.59e-71	0.7690092	0.8469762
exam_2	exam_1	0.90	3.493659e+01	2.17e-107	0.8717947	0.9167018
exam_2	exam_2	1.00	1.158479e+09	0.00e+00	1.0000000	1.0000000
exam_2	exam_3	0.91	3.861393e+01	5.27e-118	0.8918864	0.9300190
exam_3	exam_1	0.81	2.397949e+01	1.59e-71	0.7690092	0.8469762
exam_3	exam_2	0.91	3.861393e+01	5.27e-118	0.8918864	0.9300190
exam_3	exam_3	1.00	1.158479e+09	0.00e+00	1.0000000	1.0000000

9 rows | 1-7 of 8 columns

Correlation Matrix in R

```r
college %>%
  cor_mat(starts_with("exam"))
```

```
     rowname      exam_1   exam_2   exam_3
     <chr>         <dbl>    <dbl>    <dbl>
1    exam_1         1.00     0.90     0.81
2    exam_2         0.90     1.00     0.91
3    exam_3         0.81     0.91     1.00
3 rows
```

Correlation Matrix in R: upper or lower triangle

```r
college %>%
  cor_mat(starts_with("exam")) %>%
  pull_lower_triangle()
```

	rowname	**exam_1**	**exam_2**	**exam_3**
	<chr>	<chr>	<chr>	<chr>
1	exam_1			
2	exam_2	0.9		
3	exam_3	0.81	0.91	

3 rows

Correlation Matrix in R: p-values

```r
college %>%
  cor_pmat(starts_with("exam"))
```

rowname	exam_1	exam_2	exam_3
<chr>	<dbl>	<dbl>	<dbl>
exam_1	0.00e+00	2.17e-107	1.59e-71
exam_2	2.17e-107	0.00e+00	5.27e-118
exam_3	1.59e-71	5.27e-118	0.00e+00

3 rows

Correlation Matrix in R: combined

```
college %>%
  cor_mat(starts_with("exam")) %>%
  cor_mark_significant()
```

rowname <chr>	exam_1 <chr>	exam_2 <chr>	exam_3 <chr>
exam_1			
exam_2	0.9****		
exam_3	0.81****	0.91****	

3 rows

Your Turn

1. Perform a correlation among the four ACT scores. Which two scales of the ACT have the highest correlation? Which two scales have the lowest correlation?

Regression

Three basic types of regression

1. **Linear regression** has a single independent variable (IV)
2. **Multiple regression** has multiple IVs
3. **Hierarchical regression** has multiple IVs across multiple "steps"

All three use the `lm()` function in the **`stats`** package.

Linear regression

```r
college %>%
  lm(exam_1 ~ iq,
     data = .)
```

```
##
## Call:
## lm(formula = exam_1 ~ iq, data = .)
##
## Coefficients:
## (Intercept)           iq
##      50.089        0.016
```

Linear regression

```r
college %>%
  lm(exam_1 ~ iq,
     data = .) %>%
  summary()
```

```
##
## Call:
## lm(formula = exam_1 ~ iq, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.721 -12.965   0.399  13.323  48.215
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  50.0891    11.8456   4.228 3.13e-05 ***
## iq            0.0160     0.1134   0.141    0.888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.17 on 298 degrees of freedom
## Multiple R-squared:  6.681e-05,    Adjusted R-squared:  -0.003289
## F-statistic: 0.01991 on 1 and 298 DF,  p-value: 0.8879
```

Standardized Coefficients

```r
college %>%
  lm(exam_1 ~ iq,
     data = .) %>%
  lm.beta() %>%
  summary()
```

`lm.beta()` is from the **lm.beta** package.

```
##
## Call:
## lm(formula = exam_1 ~ iq, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.721 -12.965   0.399  13.323  48.215
##
## Coefficients:
##              Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept) 50.089114     0.000000  11.845612   4.228 3.13e-05
***
## iq           0.015997     0.008174   0.113369   0.141    0.888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.17 on 298 degrees of freedom
## Multiple R-squared:  6.681e-05,    Adjusted R-squared:  -0.003289
## F-statistic: 0.01991 on 1 and 298 DF,  p-value: 0.8879
```

Tidy results

```
college %>%
  lm(exam_1 ~ iq,
     data = .) %>%
  lm.beta() %>%
  tidy()
```

term <chr>	estimate <dbl>	std_estimate <dbl>	std.error <dbl>	statistic <dbl>
(Intercept)	50.08911424	0.00000000	11.8456116	4.2284954
iq	0.01599749	0.00817398	0.1133694	0.1411095

2 rows | 1-5 of 6 columns

Your Turn

1. Perform a linear regression examining how `iq` predicts `act_reading`. Ask for standardized coefficients before calling for the summary of results. Is IQ a significant predictor of ACT reading scores? If so, what is the standardized coefficient?

Multiple regression

Multiple regression in R

To perform a multiple regression, add the IVs to the right of the ~ (tilde).

Example: DV ~ IV1 + IV2

Just like with ANOVAs, you can request interactions by replacing the + with a *.

Example: DV ~ IV1 * IV2

Multiple regression in R

```
college %>%
  lm(exam_2 ~ iq + exam_1, data = .) %>%
  lm.beta() %>%
  summary()
```

```
##
## Call:
## lm(formula = exam_2 ~ iq + exam_1, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.9816  -6.2163   0.1579   5.6591  24.5116
##
## Coefficients:
##              Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  2.956034     0.000000   5.576883   0.530    0.596
## iq          -0.007941    -0.003937   0.051843  -0.153    0.878
## exam_1       0.923936     0.896560   0.026489  34.879   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Multiple regression in R

```
college %>%
  lm(exam_2 ~ iq * exam_1, data = .) %>%
  lm.beta() %>%
  summary()
```

```
##
## Call:
## lm(formula = exam_2 ~ iq * exam_1, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.9536  -6.3219   0.2314   5.6324  24.6062
##
## Coefficients:
##               Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  6.6708634    0.0000000 16.6424406   0.401  0.68883
## iq          -0.0437046   -0.0216694  0.1596091  -0.274  0.78441
## exam_1       0.8549552    0.8296235  0.2923125   2.925  0.00371 **
## iq:exam_1    0.0006639    0.0696530  0.0028019   0.237  0.81285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Categorical predictors

```r
college %>%
  lm(exam_2 ~ iq + gender, data = .) %>%
  lm.beta() %>%
  summary()
```

```
##
## Call:
## lm(formula = exam_2 ~ iq + gender, data = .)
##
## Residuals:
##     Min     1Q Median      3Q     Max
## -48.82 -13.80   0.23  14.91  49.77
##
## Coefficients:
##               Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  48.806287     0.000000  12.142610   4.019 7.41e-05 ***
## iq           -0.004699    -0.002330   0.115918  -0.041   0.9677
## genderMale    4.525041     0.114100   2.323535   1.947   0.0524 .
## genderTrans* -7.751819    -0.083056   5.467970  -1.418   0.1573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Changing the reference level

```r
college %>%
  mutate(gender = fct_relevel(gender, levels = "Trans*", "Female", "Male")) %>%
  lm(exam_2 ~ iq + gender, data = .) %>%
  lm.beta() %>%
  summary()
```

```
##
## Call:
## lm(formula = exam_2 ~ iq + gender, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -48.82 -13.80   0.23  14.91  49.77
##
## Coefficients:
##               Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  41.054468     0.000000  13.013549   3.155  0.00177 **
## iq           -0.004699    -0.002330   0.115918  -0.041  0.96769
## genderFemale  7.751819     0.196817   5.467970   1.418  0.15734
## genderMale   12.276860     0.309564   5.507382   2.229  0.02655 *
## ---
```

Your Turn

1. Perform a multiple regression examining how both `iq` and `act_science` predict `act_reading`. Ask for standardized coefficients before calling for the summary of results. Which (if any) of the IVs are significant predictors? Based on the standardized coefficient, which is the stronger predictor of ACT readings cores?

Hierarchical regression

Hierarchical regression

Hierarchical regression takes a *little* more work. Here are the steps:

1. Perform a regression with your first set of predictors.
2. Perform a regression *adding* your second set of predictors.
3. Run `anova()` comparing the two models.

Hierarchical regression in R

Step 1:

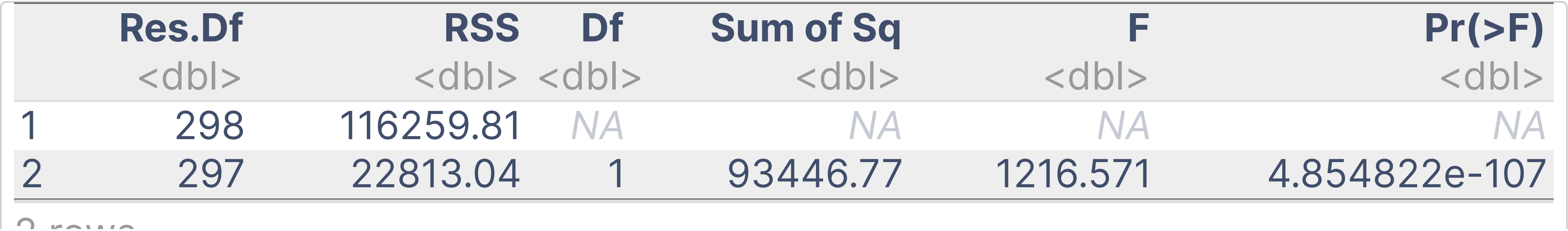
```
lm1 <-
  college %>%
  lm(exam_2 ~ iq, data = .) %>%
  lm.beta()
```

Step 2:

```
lm2 <-
  college %>%
  lm(exam_2 ~ iq + exam_1, data = .) %>%
  lm.beta()
```

Step 3:

```
anova(lm1, lm2)
```



	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	298	116259.81	*NA*	*NA*	*NA*	*NA*
2	297	22813.04	1	93446.77	1216.571	4.854822e-107

2 rows

Your Turn

Read more about hierarchical regression with the resources below.

Reliability

Reliability

When working with scales, we want to see how well items "hang together." This is called internal consistency, which is a type of reliability.

You are likely familiar with Cronbach's alpha as a measure of internal consistency.

However, Cronbach's alpha has assumptions your data *may not* meet. For that reason, I recommend using omega instead.

Assumptions of Cronbach's alpha

1. Scale is unidimensional
2. Scale items are continuous and normally distributed
3. Tau equivalence
4. Errors of the items do not covary

Tau equivalence

Tau equivalent

Item	Std. Loading
Q1	.711
Q2	.714
Q3	.716
Q4	.709
Q5	.721

Not Tau equivalent

Item	Std. Loading
Q1	.806
Q2	.790
Q3	.725
Q4	.578
Q5	.523

In my experience running lots of factor analyses, I *almost never* have tau equivalence.

Covarying item errors

Some items may be more related to other items in the scale for a variety of reasons:

- The items are very similarly worded
- The items measure essentially the same thing
- Order of items
- Speeded tests
- Transient responses where feelings or opinions may change over the course of the scale
- Unmodeled multidimensionality of the scale

Omega

Omega is designed for scales that are not tau equivalent (also called congeneric). Most of our scales are congeneric!

Therefore, I recommend you always use omega instead of alpha!

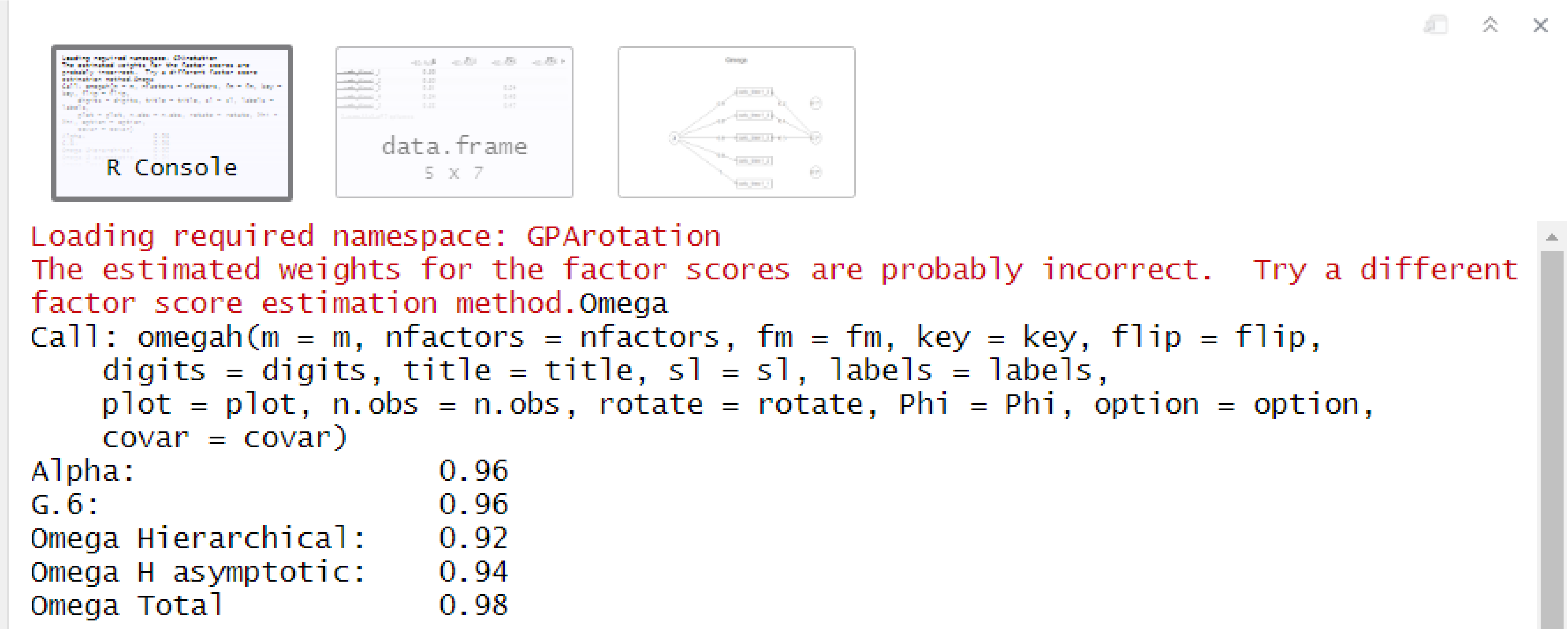
The values are interpreted just the same as you would alpha. It will just be a less biased estimate of the internal consistency of your scale, especially if you do not meet the assumptions of alpha.

Omega and alpha in R

```
library(psych)

omega(dat)

alpha(dat)
```

An example: omega

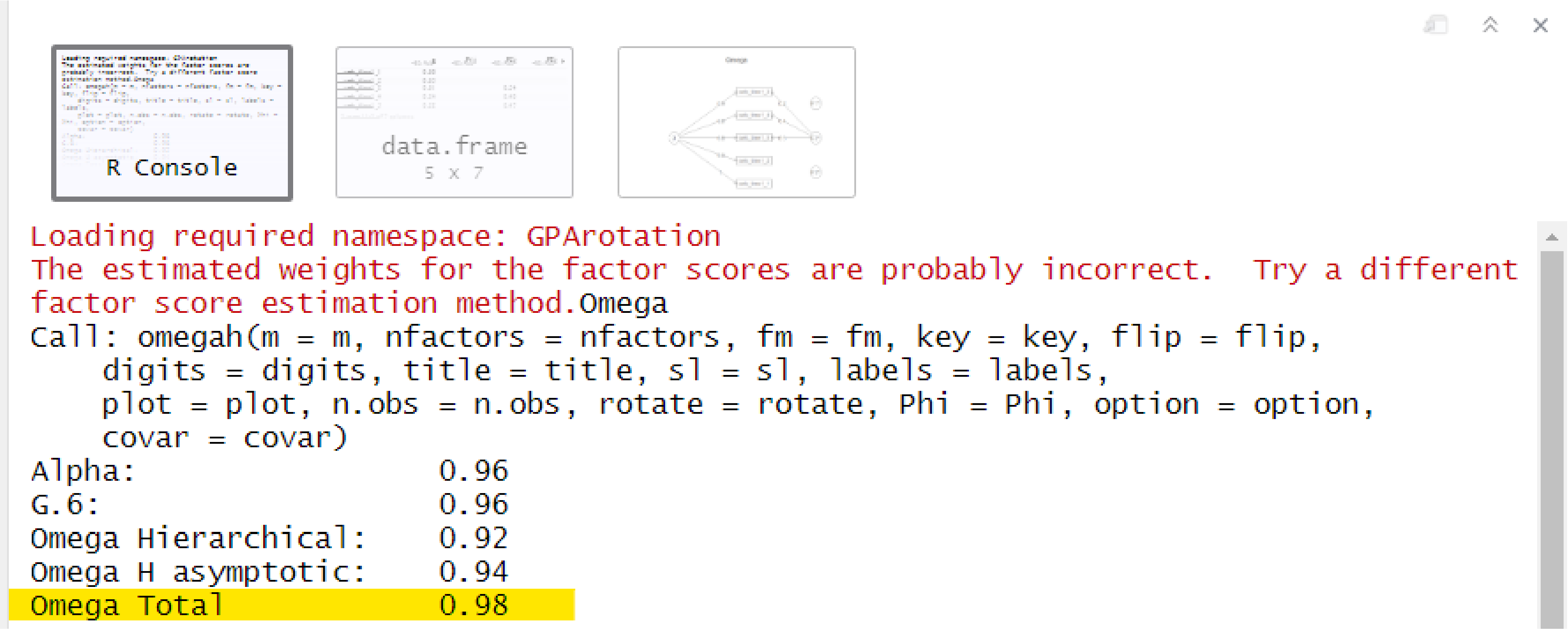
```r
college %>%
  select(starts_with("swls_time1")) %>%
  omega()
```



```
Loading required namespace: GPArotation
The estimated weights for the factor scores are probably incorrect.  Try a different
factor score estimation method.Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.96
G.6:                   0.96
Omega Hierarchical:    0.92
Omega H asymptotic:    0.94
Omega Total            0.98
```

An example: omega

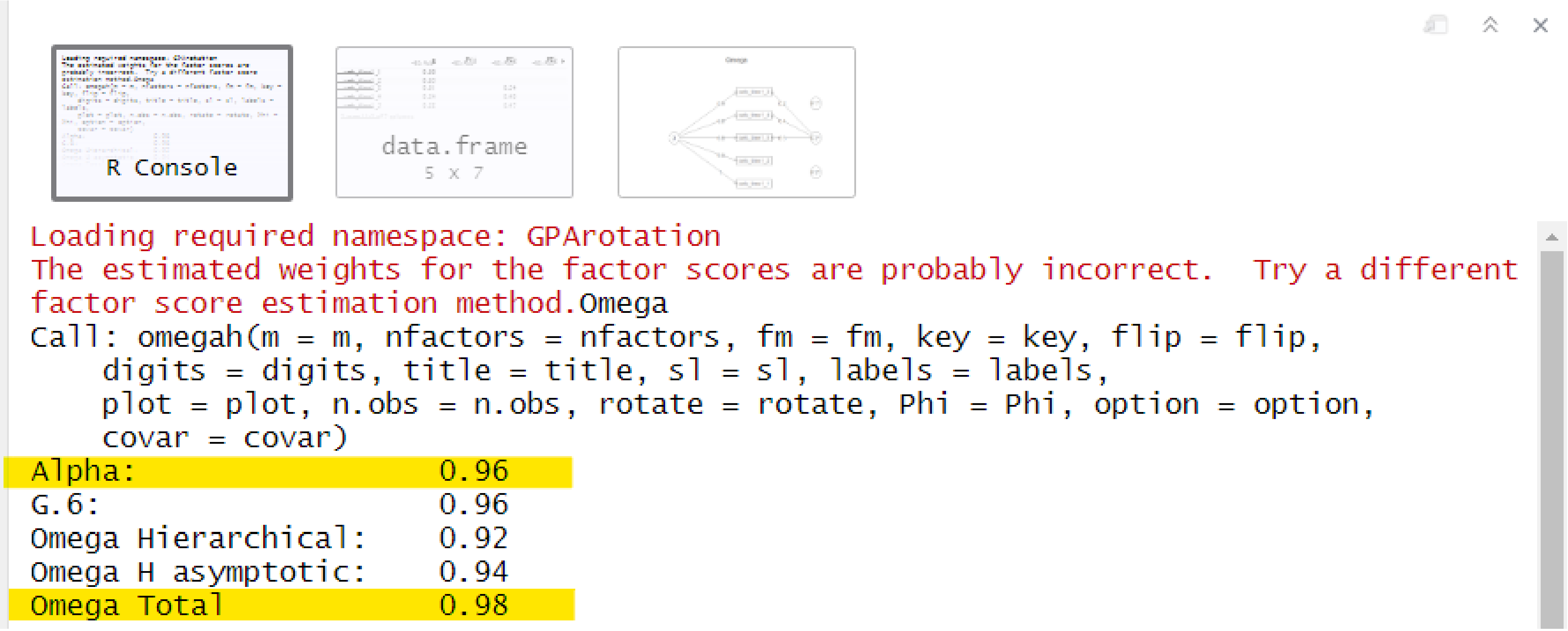
```
college %>%
  select(starts_with("swls_time1")) %>%
  omega()
```



```
Loading required namespace: GPArotation
The estimated weights for the factor scores are probably incorrect.  Try a different
factor score estimation method.Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.96
G.6:                   0.96
Omega Hierarchical:    0.92
Omega H asymptotic:    0.94
Omega Total            0.98
```

An example: omega

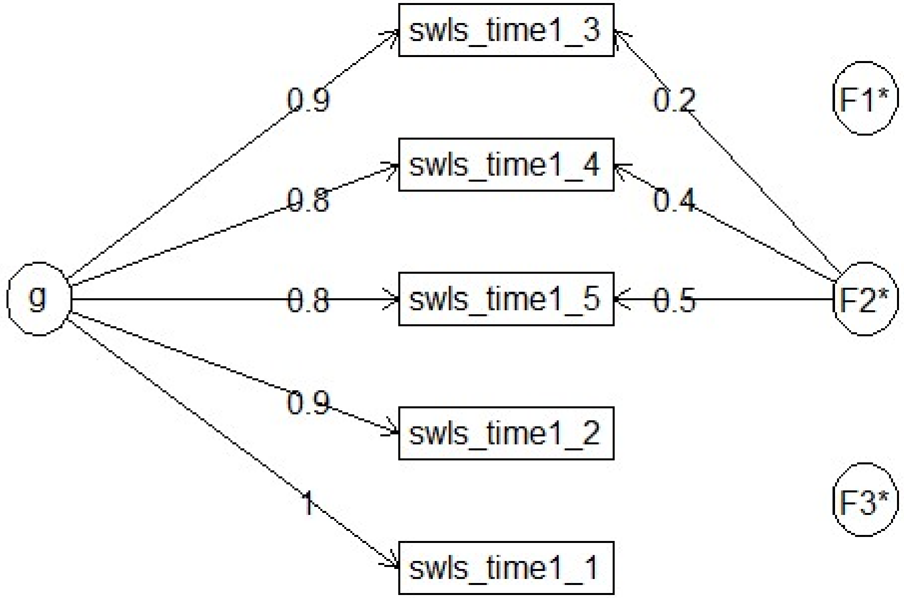
```
college %>%
  select(starts_with("swls_time1")) %>%
  omega()
```



```
Loading required namespace: GPArotation
The estimated weights for the factor scores are probably incorrect.  Try a different
factor score estimation method.Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.96
G.6:                   0.96
Omega Hierarchical:    0.92
Omega H asymptotic:    0.94
Omega Total            0.98
```

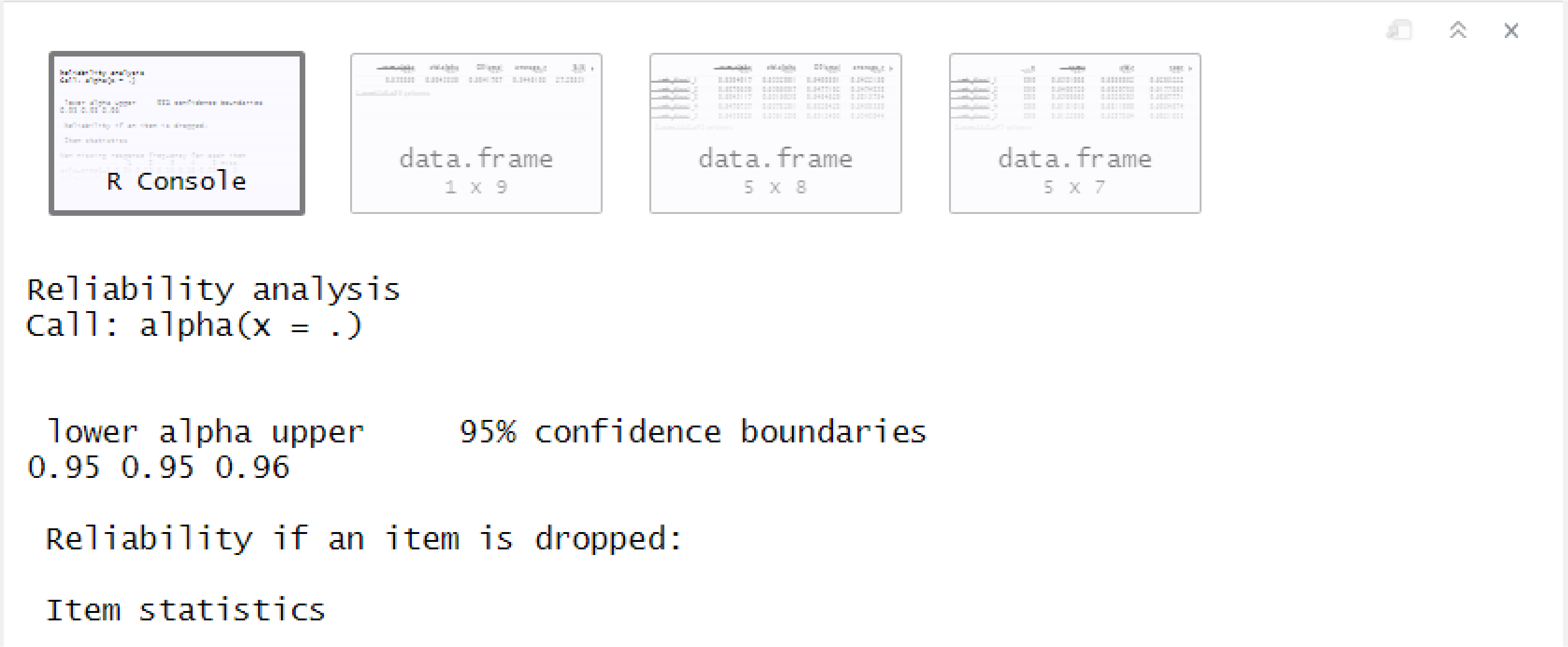
An example: omega

```
college %>%
  select(starts_with("swls_time1")) %>%
  omega()
```



An example: alpha

```
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```



```
Reliability analysis
Call: alpha(x = .)


 lower alpha upper     95% confidence boundaries
0.95 0.95 0.96

 Reliability if an item is dropped:

 Item statistics
```

An example: alpha

```
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```

```
##                        1         2         3         4          5 miss
## swls_time1_1 0.236666667 0.1900000 0.1900000 0.1833333 0.20000000    0
## swls_time1_2 0.110000000 0.2400000 0.3700000 0.1933333 0.08666667    0
## swls_time1_3 0.033333333 0.4333333 0.1600000 0.3333333 0.04000000    0
## swls_time1_4 0.010000000 0.3166667 0.4033333 0.2533333 0.01666667    0
## swls_time1_5 0.006666667 0.4433333 0.2066667 0.3333333 0.01000000    0
```

An example: alpha

```
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```

raw_alpha <dbl>	std.alpha <dbl>	G6(smc) <dbl>	average_r <dbl>	S/N <dbl>	ase <dbl>
0.953686	0.9645886	0.9641767	0.8449108	27.23951	0.003575771

1 row | 1-7 of 10 columns

An example: alpha

```
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```

	raw_alpha	std.alpha	G6(smc)	average_r	S/N
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
swls_time1_1	0.9504017	0.9552601	0.9466861	0.8422180	21.35144
swls_time1_2	0.9370956	0.9569307	0.9477102	0.8474353	22.21838
swls_time1_3	0.9345117	0.9519053	0.9494920	0.8318784	19.79230
swls_time1_4	0.9470757	0.9573201	0.9526423	0.8486580	22.43021
swls_time1_5	0.9453626	0.9591266	0.9512468	0.8543644	23.46581

1-5 of 5 rows | 1-6 of 9 columns

An example: alpha

```
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```

	n <dbl>	raw.r <dbl>	std.r <dbl>	r.cor <dbl>	r.drop <dbl>
swls_time1_1	300	0.9531068	0.9393632	0.9268222	0.9093061
swls_time1_2	300	0.9400729	0.9326738	0.9177395	0.9035852
swls_time1_3	300	0.9509995	0.9526205	0.9397771	0.9236305
swls_time1_4	300	0.9181619	0.9311060	0.9084674	0.8850635
swls_time1_5	300	0.9122396	0.9237894	0.9021638	0.8718080

1-5 of 5 rows | 1-6 of 8 columns

Your Turn

1. Perform a test of internal consistency using both omega and alpha on the SWLS scale at time 2. Does the scale have good internal consistency at time 2, as well?

Extracting output

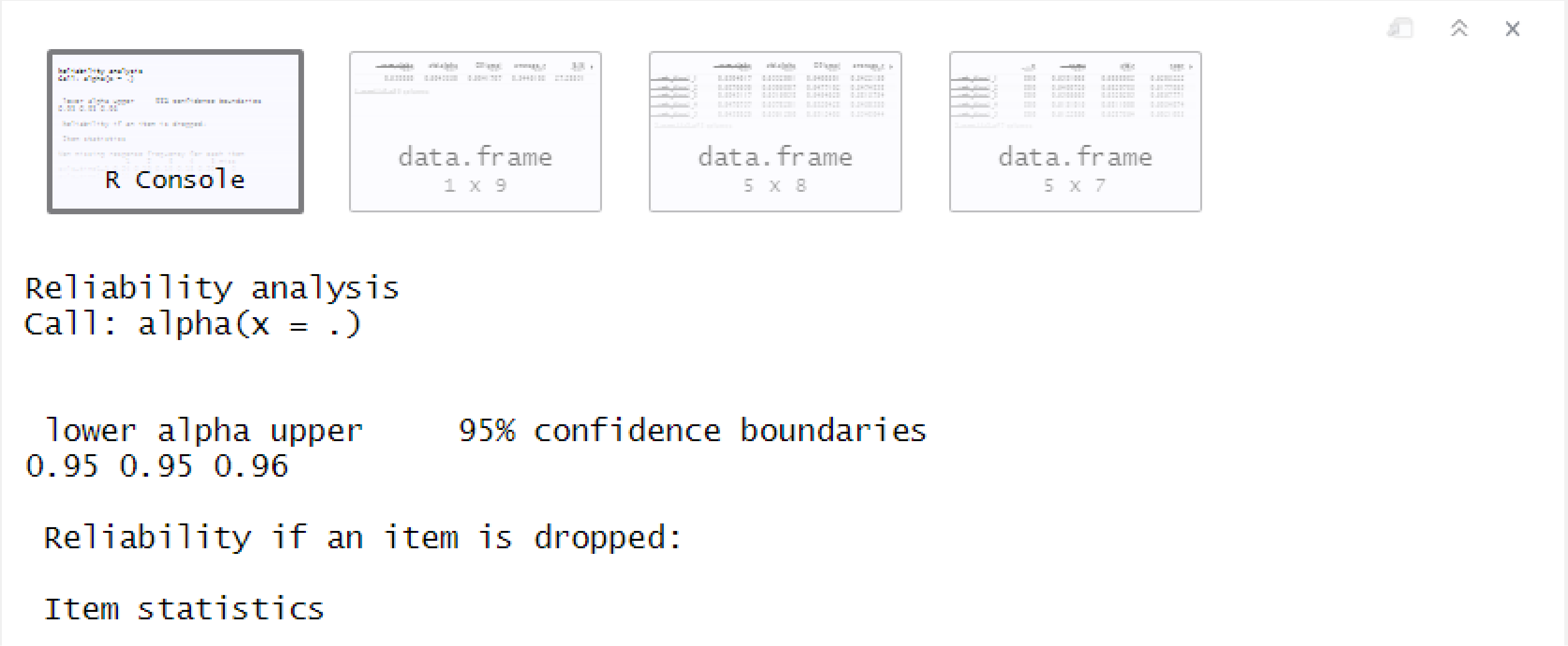
Extracting output

So you've got your output...

what next?

An example: alpha()

```r
college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```



```
Reliability analysis
Call: alpha(x = .)


 lower alpha upper     95% confidence boundaries
0.95 0.95 0.96

 Reliability if an item is dropped:

 Item statistics
```

An example: alpha()

```
rel <- college %>%
  select(starts_with("swls_time1")) %>%
  alpha()
```

An example: alpha()

```
rel$alpha.drop
```

	raw_alpha	std.alpha	G6(smc)	average_r	S/N
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
swls_time1_1	0.9504017	0.9552601	0.9466861	0.8422180	21.35144
swls_time1_2	0.9370956	0.9569307	0.9477102	0.8474353	22.21838
swls_time1_3	0.9345117	0.9519053	0.9494920	0.8318784	19.79230
swls_time1_4	0.9470757	0.9573201	0.9526423	0.8486580	22.43021
swls_time1_5	0.9453626	0.9591266	0.9512468	0.8543644	23.46581

1-5 of 5 rows | 1-6 of 9 columns

An example: alpha()

```
rel$alpha.drop %>%
  select(std.alpha) %>%
  rownames_to_column("var") %>%
  arrange(std.alpha)
```

var <chr>	std.alpha <dbl>
swls_time1_3	0.9519053
swls_time1_1	0.9552601
swls_time1_2	0.9569307
swls_time1_4	0.9573201
swls_time1_5	0.9591266

1-5 of 5 rows

An example: alpha()

```
drop <-
  rel$alpha.drop %>%
  select(std.alpha) %>%
  rownames_to_column("var") %>%
  arrange(std.alpha)
```

An example: alpha()

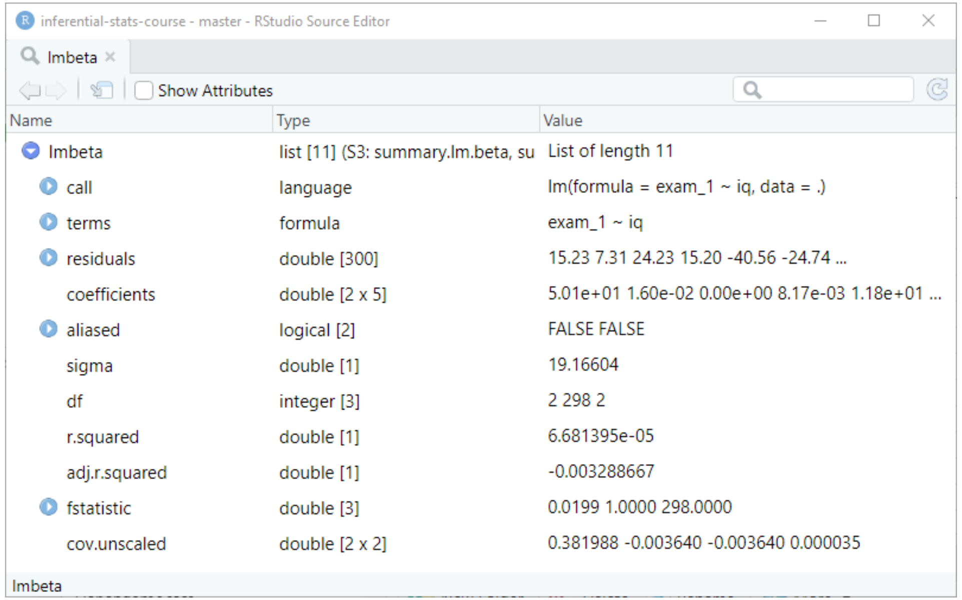
```r
drop <- rel$alpha.drop %>%
  select(std.alpha) %>%
  rownames_to_column("var") %>%
  arrange(std.alpha)

rel$total %>%
  select(std.alpha) %>%
  rownames_to_column("var") %>%
  bind_rows(drop) %>%
  mutate(std.alpha = round(std.alpha, 3))
```

var <chr>	std.alpha <dbl>
	0.965
swls_time1_3	0.952
swls_time1_1	0.955
swls_time1_2	0.957
swls_time1_4	0.957
swls_time1_5	0.959

6 rows

Another example: lm()

```
lmbeta <- college %>%
  lm(exam_1 ~ iq, data = .) %>%
  lm.beta() %>%
  summary()
```



Name	Type	Value
lmbeta	list [11] (S3: summary.lm.beta, su	List of length 11
call	language	lm(formula = exam_1 ~ iq, data = .)
terms	formula	exam_1 ~ iq
residuals	double [300]	15.23 7.31 24.23 15.20 -40.56 -24.74 ...
coefficients	double [2 x 5]	5.01e+01 1.60e-02 0.00e+00 8.17e-03 1.18e+01 ...
aliased	logical [2]	FALSE FALSE
sigma	double [1]	19.16604
df	integer [3]	2 298 2
r.squared	double [1]	6.681395e-05
adj.r.squared	double [1]	-0.003288667
fstatistic	double [3]	0.0199 1.0000 298.0000
cov.unscaled	double [2 x 2]	0.381988 -0.003640 -0.003640 0.000035

Another example: lm()

```
lmbeta$coefficients
```

```
##                Estimate Standardized Std. Error   t value     Pr(>|t|)
## (Intercept) 50.08911424   0.00000000 11.8456116 4.2284954 3.133076e-05
## iq           0.01599749   0.00817398  0.1133694 0.1411095 8.878788e-01
```

Another example: lm()

```
lmbeta$adj.r.squared
```

```
## [1] -0.003288667
```

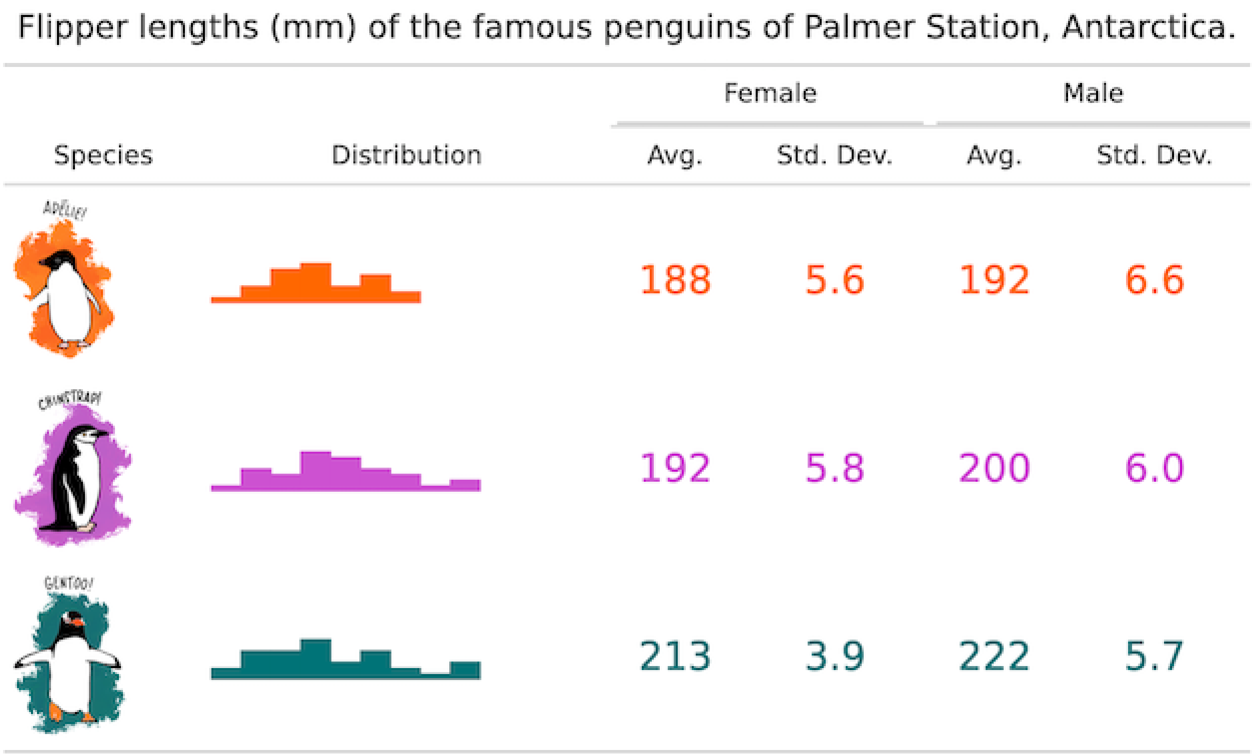
Your Turn

1. Perform a test of internal consistency using omega on the SWLS scale at time 1.
2. Extract the `omega.tot` information from the list. Round it to two decimals. What is your omega value?

Reporting results

modelsummary package

Creates highly customizable tables and plots to summarize statistical models and data in R.



Flipper lengths (mm) of the famous penguins of Palmer Station, Antarctica.

		Female		Male	
Species	Distribution	Avg.	Std. Dev.	Avg.	Std. Dev.
		188	5.6	192	6.6
		192	5.8	200	6.0
		213	3.9	222	5.7

apa package

The **apa** package functions help you format statistical tests according to the American Psychological Association (APA) guidelines. It supports the following tests from the **stats** package:

- t-test: `t.test()` → **t_apa()**
- ANOVA: `aov()` → **anova_apa()**
- chi-square: `chisq.test()` → **chisq_apa()**
- correlation: `cor.test()` → **cor_apa()**

```
college %>%
  t.test(exam_1 ~ athlete, data = ., var.equal = TRUE) %>%
  t_apa()
```

```
## t(298) = -2.25, p = .025, d = -0.26
```

apa package: inline code

If you're writing your results directly in R, you can use inline code to embed your statistics in your text wrapping the statistic with `apa()`.

For example

This: There was a strong, positive relationship between exam 1 and exam 2 scores, `'r apa(cor.test(college$exam_1, college$exam_2))'`.

Note: use backticks, not apostrophes.

Turns into this: There was a strong, positive relationship between exam 1 and exam 2 scores, $r(298) = .90$, $p < .001$.

papaja package

papaja stands for Preparing APA Journal Articles. It is an R Markdown template that produces APA manuscripts.

It works with the **apa**, **APAstats**, and **apaTables** packages for easy reporting in accordance with APA guidelines.

Testing assumptions

Testing assumptions

Our parametric statistics all have four main assumptions that we should test and verify *before* running our statistical analyses:

1. Our data is **normally distributed**.
2. Data between groups have **equal (homogeneous) variances**.
3. Scores are **independent** from one another. *Note*: this is not an assumption of within subjects designs like the dependent t-test or repeated measures ANOVA.
4. Data is at the **interval or ratio** level. *Note*: likert-scale data is *technically* at the ordinal level, and there is research to suggest it shouldn't be analyzed at the interval/ratio level.

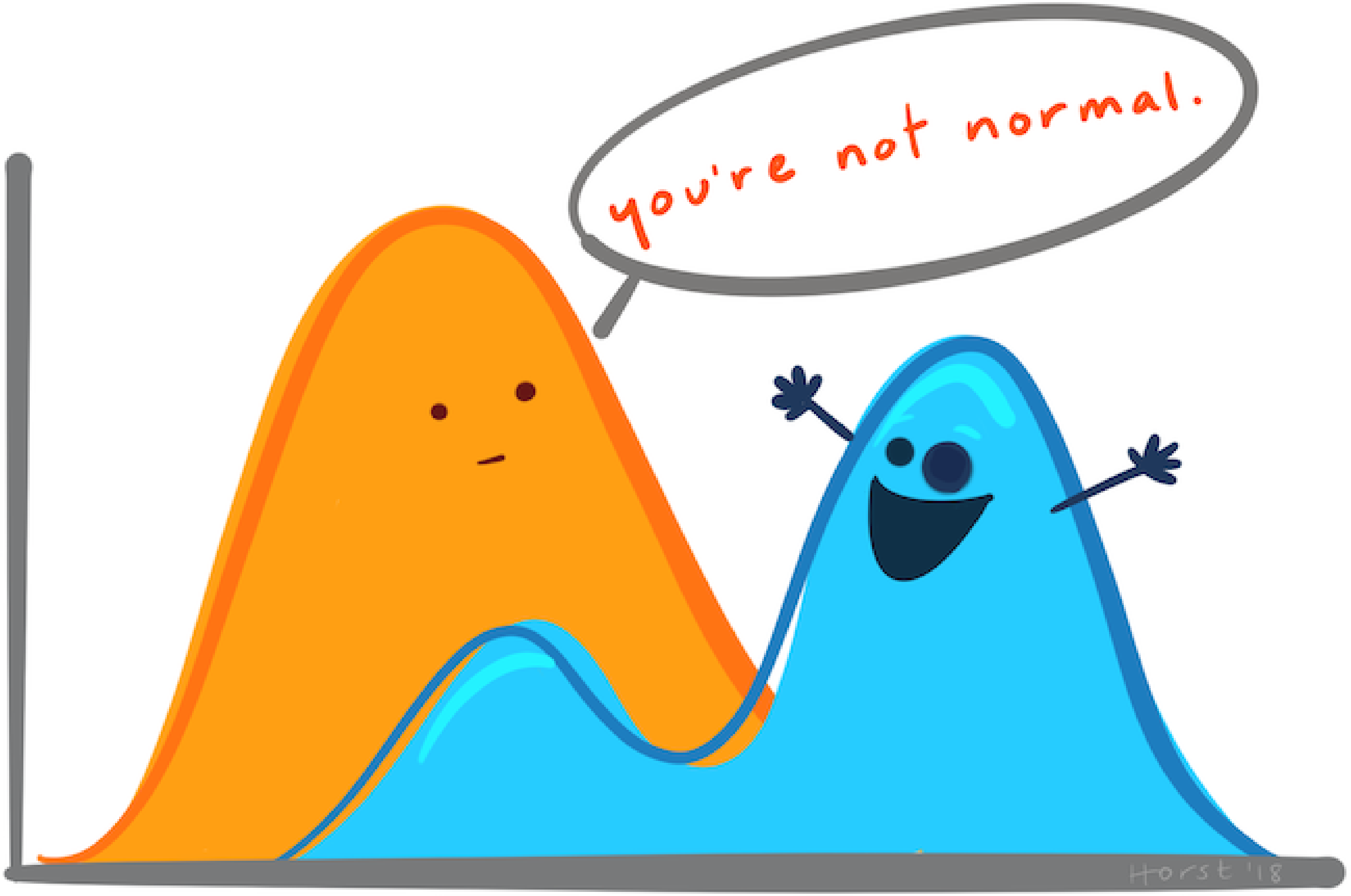
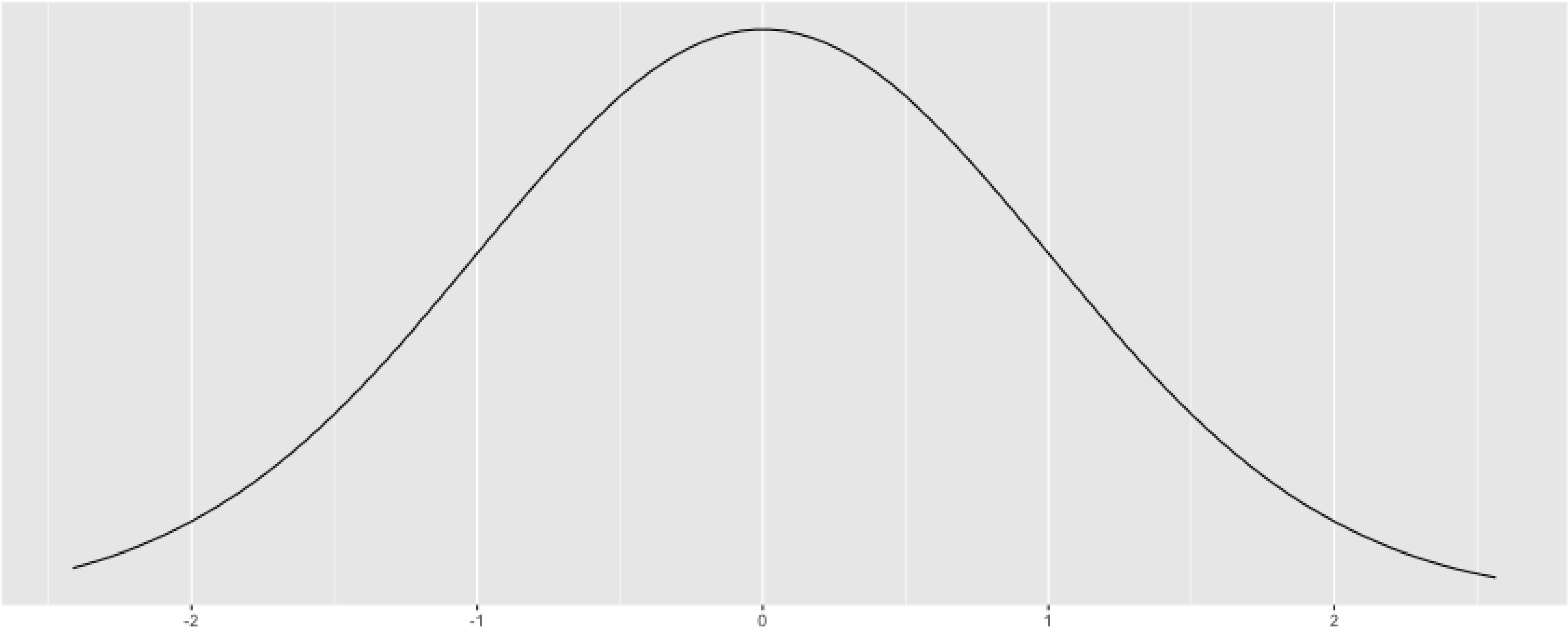


Image by [Allison Horst]

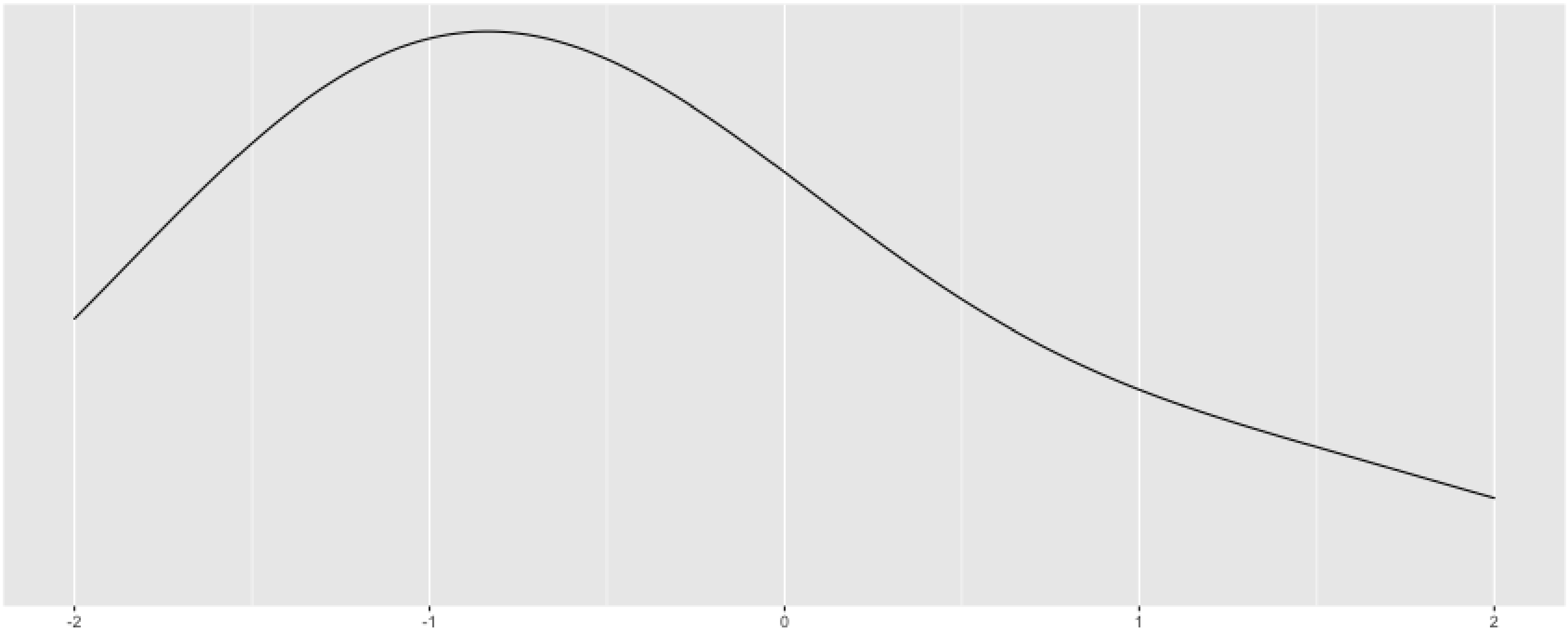
Normality

Normally distributed:



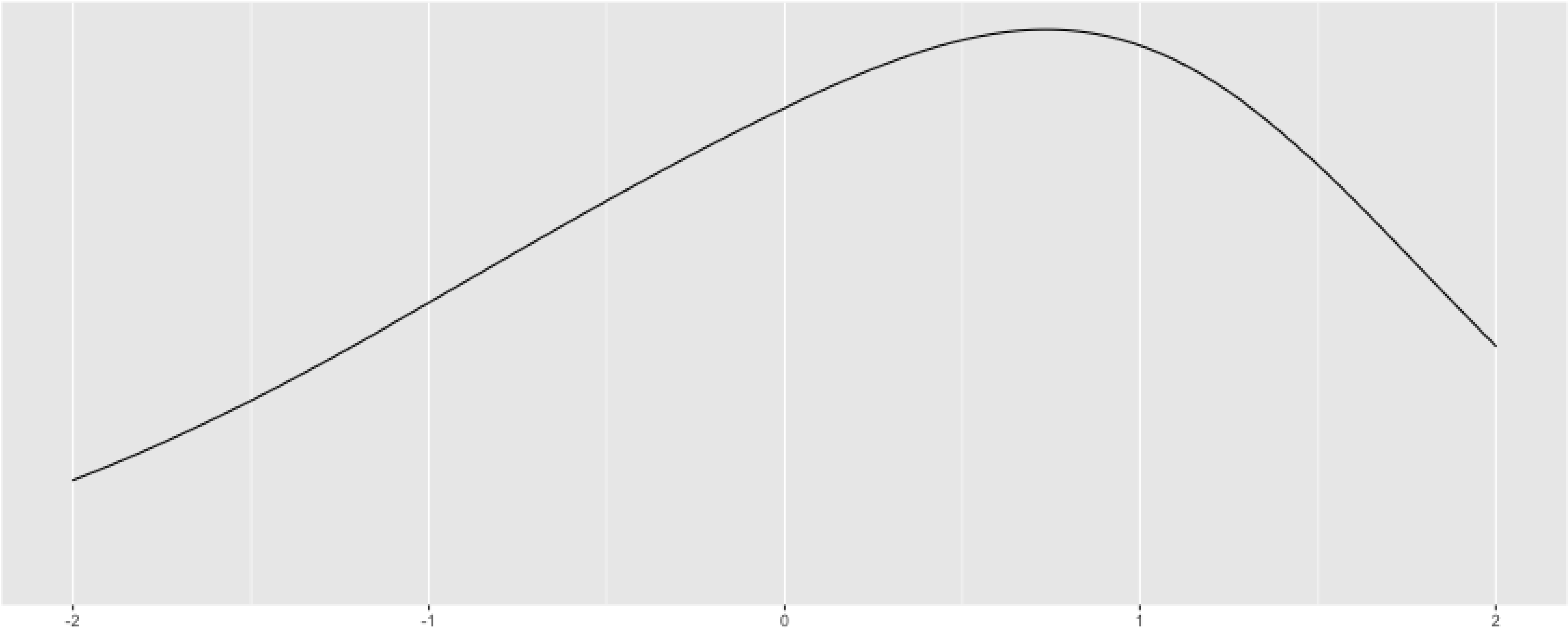
Normality

Positively skewed:



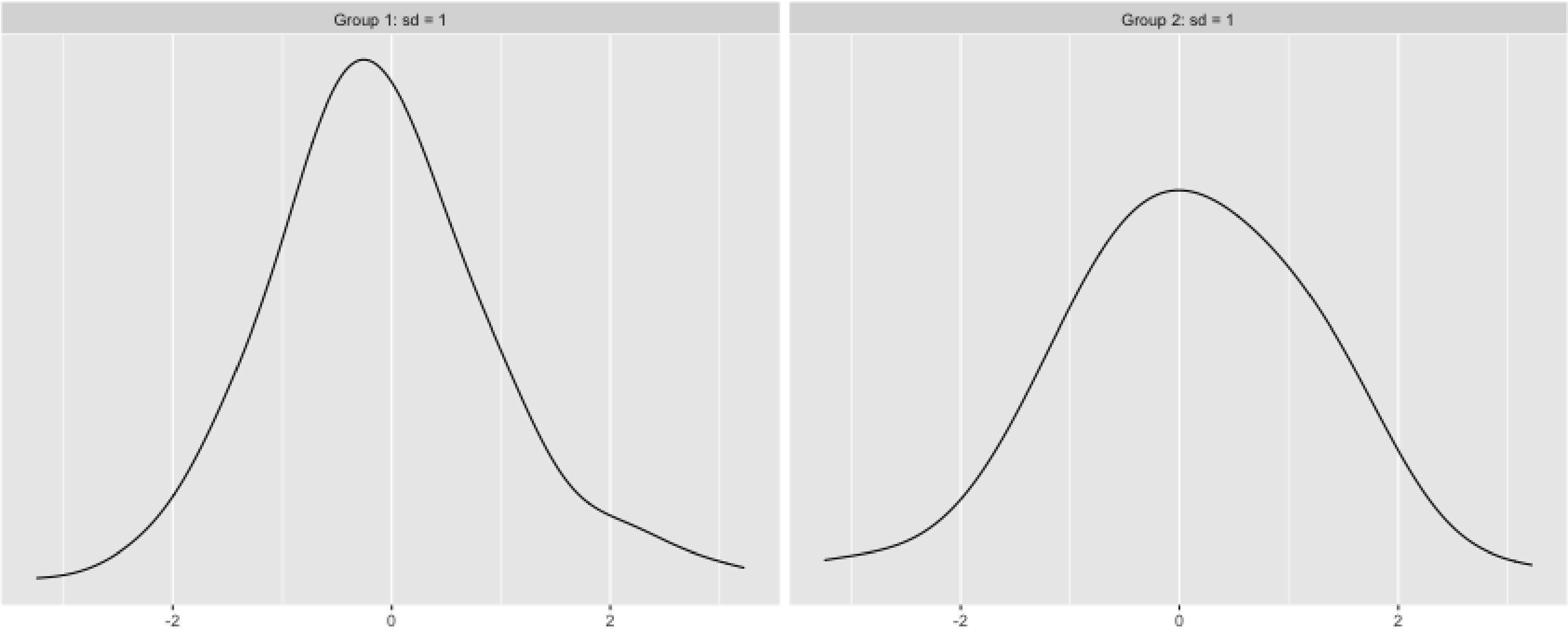
Normality

Negatively skewed:



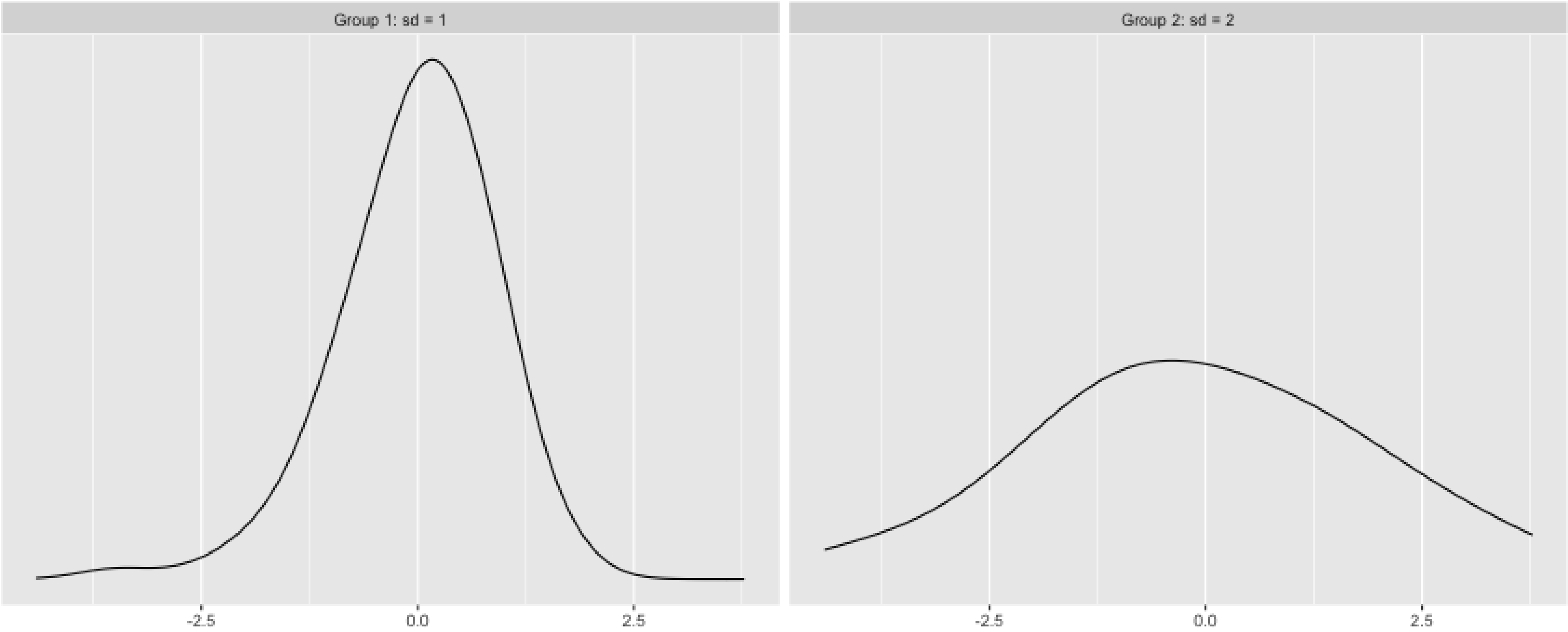
Homogeneity of variances

Equal variances or homogeneity of variances (assumption met):



Homogeneity of variances

Unequal variances or heterogeneity of variances (assumption not met):



Independent data

This cannot be explicitly tested, but is a function of you knowing your data.

- Are some data points more related to one another?
- Are some participants related to one another more than with other participants?
- Do you have nested data?

If you said "yes" to any of these questions, you likely violate this assumption.

Interval or ratio data

Interval and ratio data are:

- Ordered variables
- That have proportionate intervals between levels
- (Ratio variables can accommodate an absolute zero)

Ordinal variables *do not have proportionate intervals between levels*.

People working with likert-scales (e.g., on a 1-7 scale) often *assume* that people respond to them as if response options are proportionate, but research suggests they are *not* proportionate.

Your Turn

1. Read through the documents for more information on the importance of testing the assumptions of the statistics you perform.

Testing for normality

Testing for normality

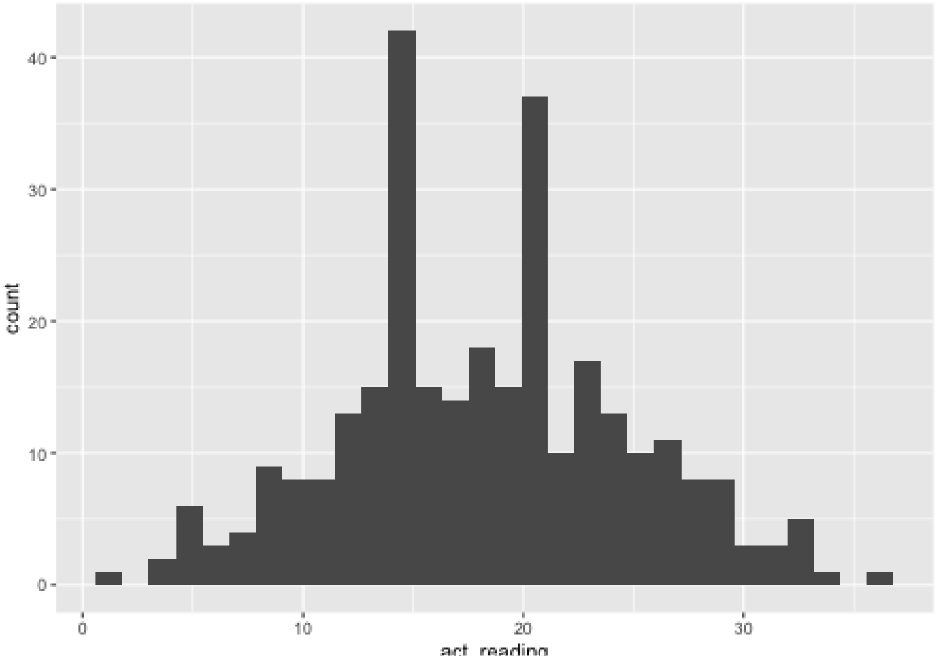
There are a few basic ways to test whether your DV is normally distributed:

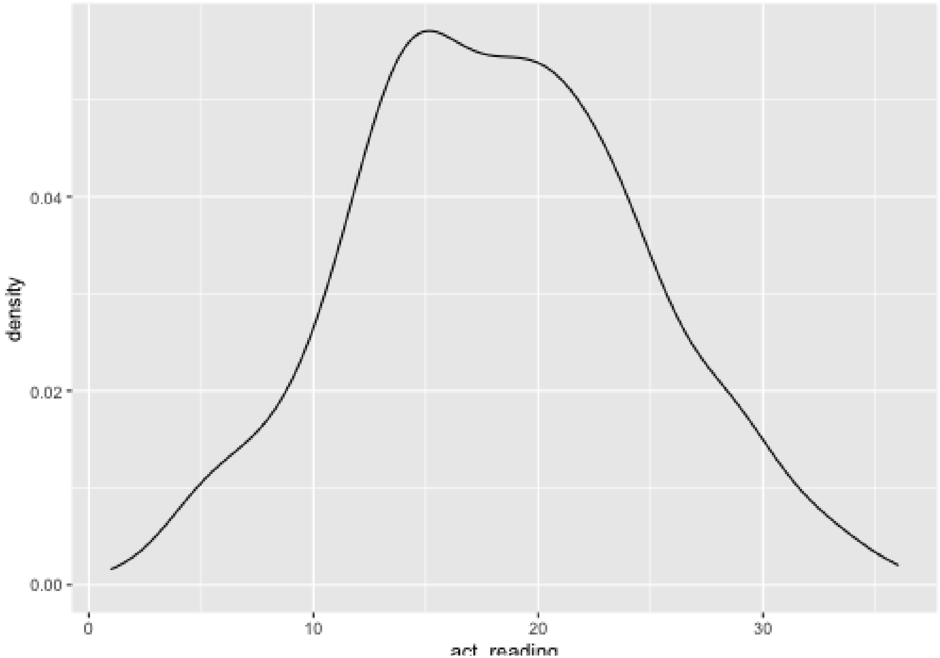
1. Histogram or density plots
2. q-q plots
3. Shapiro-Wilks test

Histogram and density plots

```
college %>%
  ggplot(aes(act_reading)) +
  geom_histogram()
```

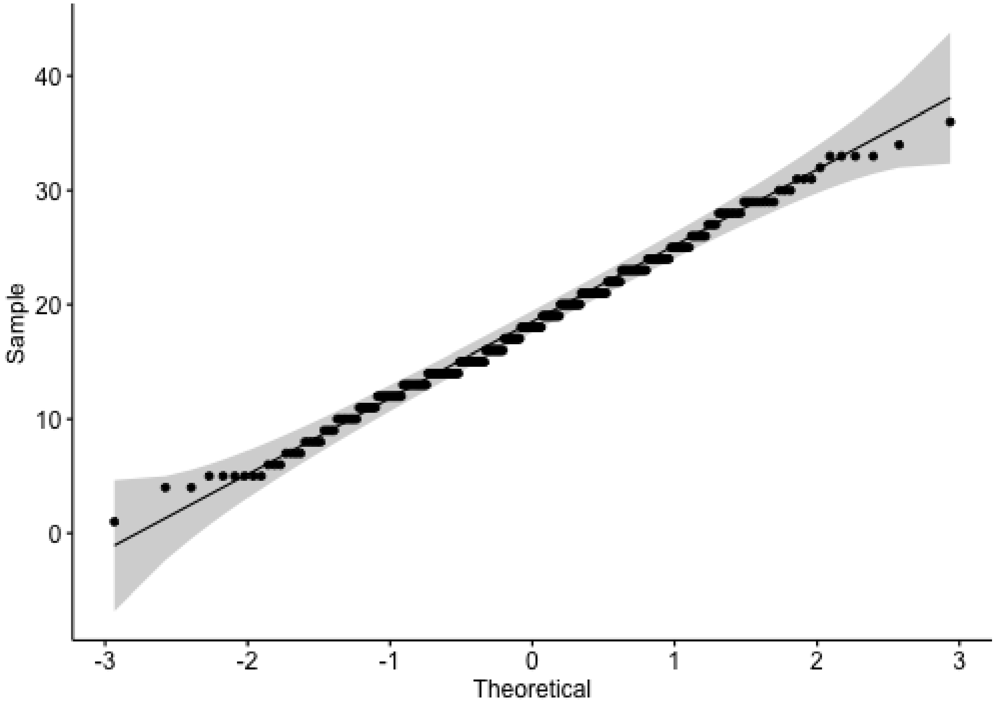
```
college %>%
  ggplot(aes(act_reading)) +
  geom_density()
```





q-q plots

```
college %>%
  ggqqplot("act_reading") ## ggpubr package
```



Shapiro-Wilks test

```
college %>%
  shapiro_test(act_reading) ## rstatix package
```

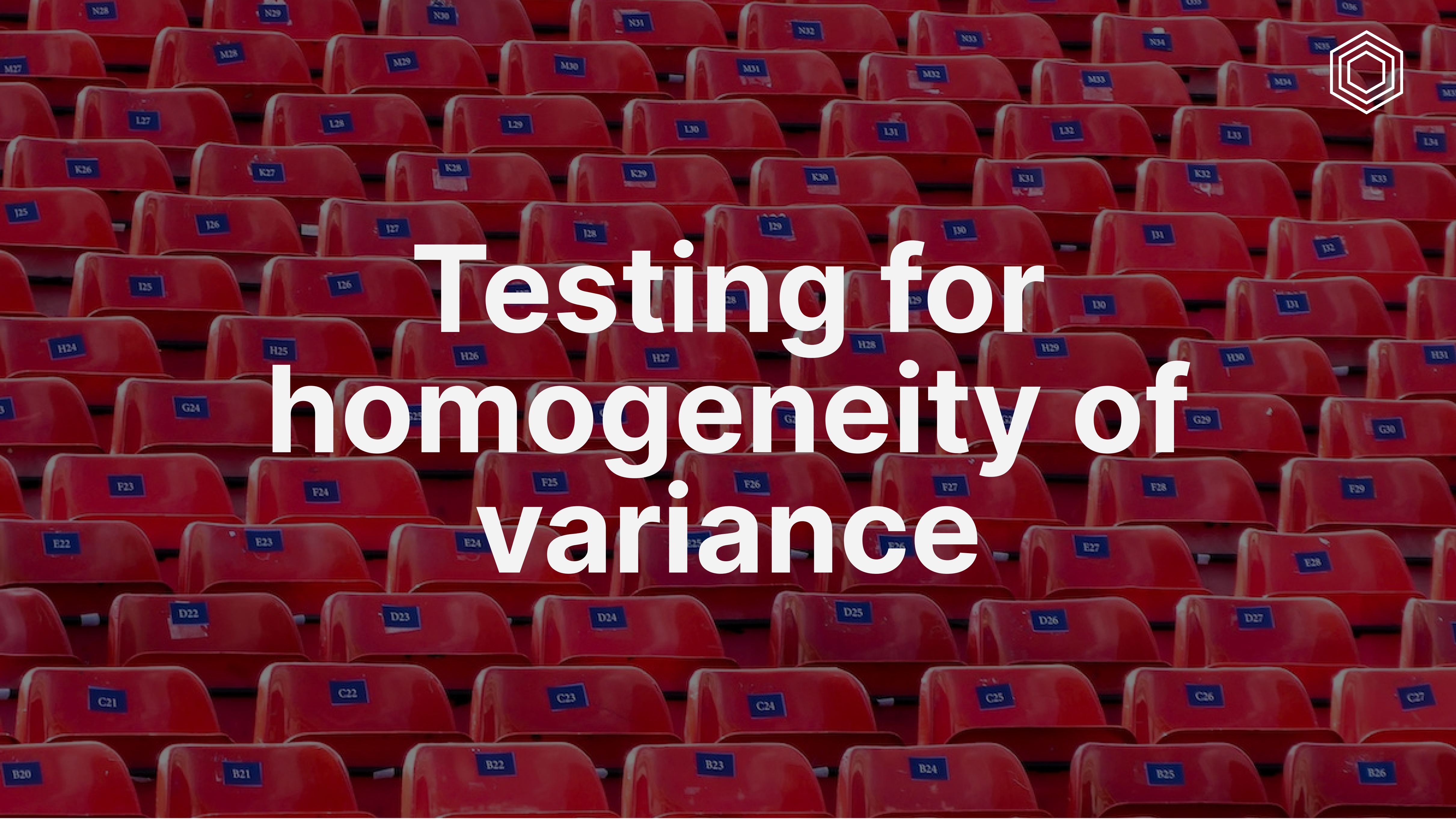
variable <chr>	statistic <dbl>	p <dbl>
act_reading	0.9937468	0.2503583

1 row

When Shapiro-Wilks is *not statistically significant* then you have evidence that the data is normally distributed.

Your Turn

1. Using each of the three methods, test whether `age` is normally distributed. Come to a conclusion: is age normally distributed?



Testing for homogeneity of variance

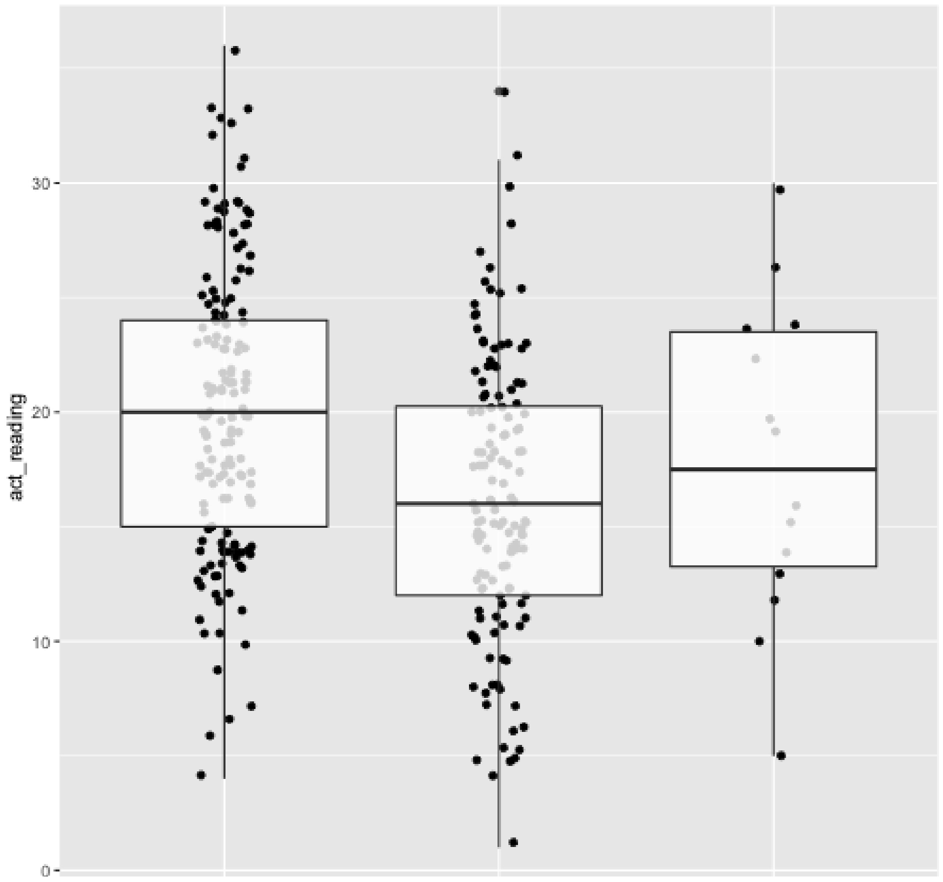
Testing for homogeneity of variance

Two basic methods for testing for homogeneity of variance:

- Graph the data by groups
- Levene's test

Graph the data by groups

```
college %>%
  ggplot(aes(gender, act_reading)) +
  geom_jitter(width = .1) +
  geom_boxplot(alpha = .8)
```



Levene's test

```r
college %>%
  levene_test(act_reading ~ gender) ## rstatix package
```

df1	df2	statistic	p
<int>	<int>	<dbl>	<dbl>
2	297	0.3434131	0.7096264

1 row

Just like Shapiro-Wilk's test, when Levene's test is *not statistically significant* then you have evidence that the data has homogeneous (equal) variances.

Your Turn

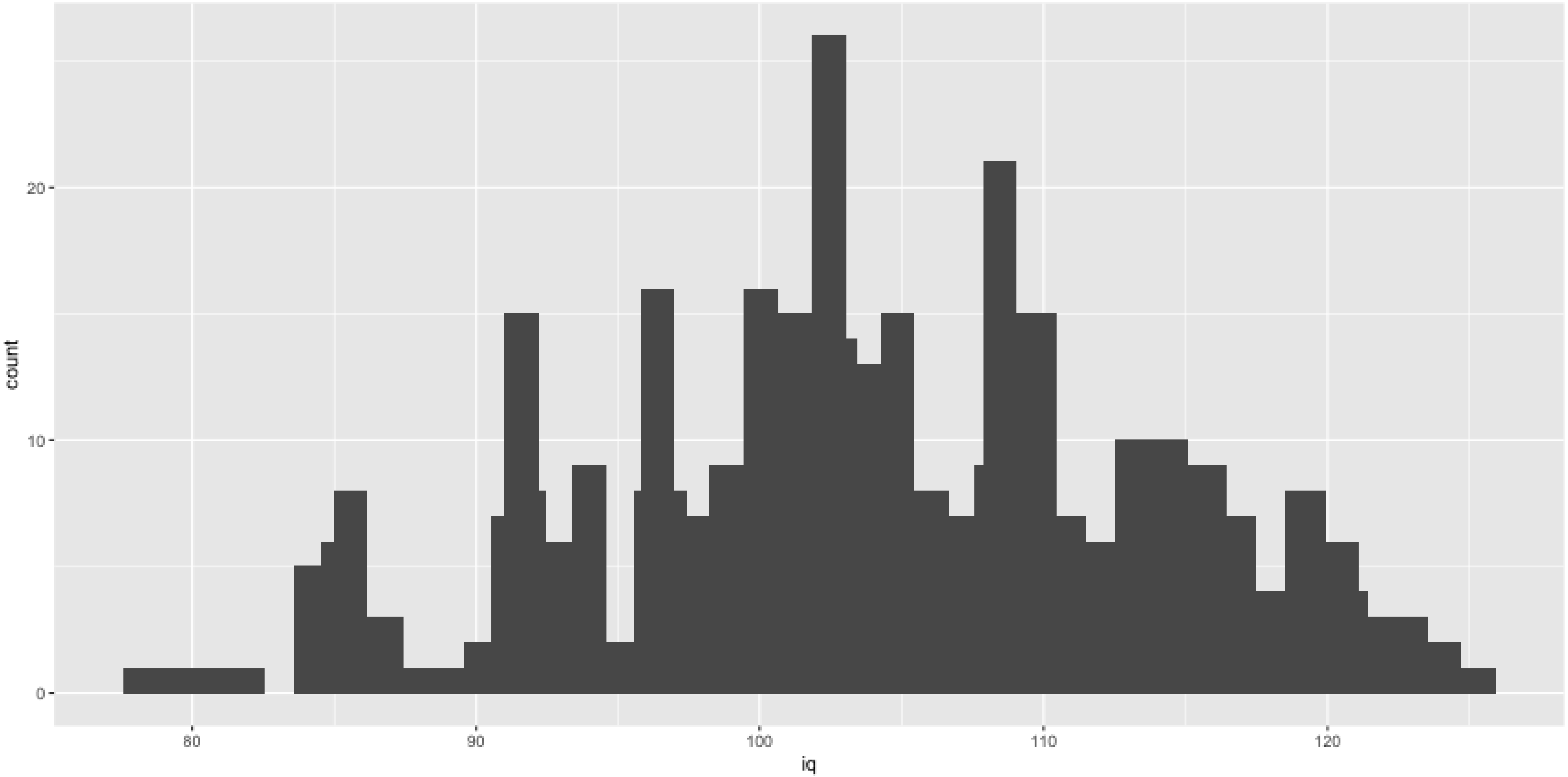
1. Using each of the two methods, test whether the variance of `weight_1` differs by whether someone smokes. Come to a conclusion: is there homogeneity of variance in weight by smoking status?

Violated assumptions

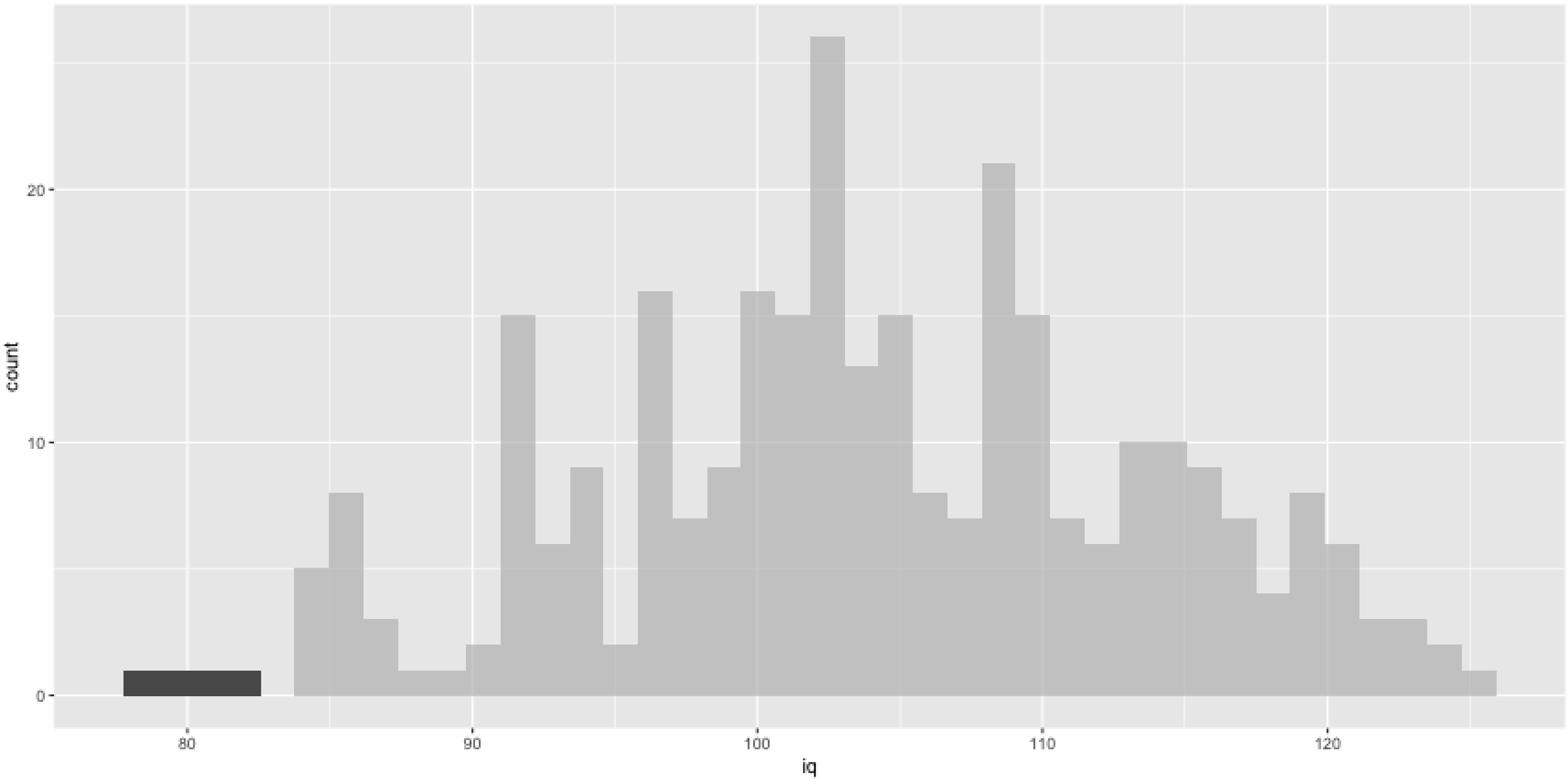
Violated assumptions

1. Winsorize outliers
2. Transform the data
3. Use the non-parametric equivalent statistic

Winsorize or trim

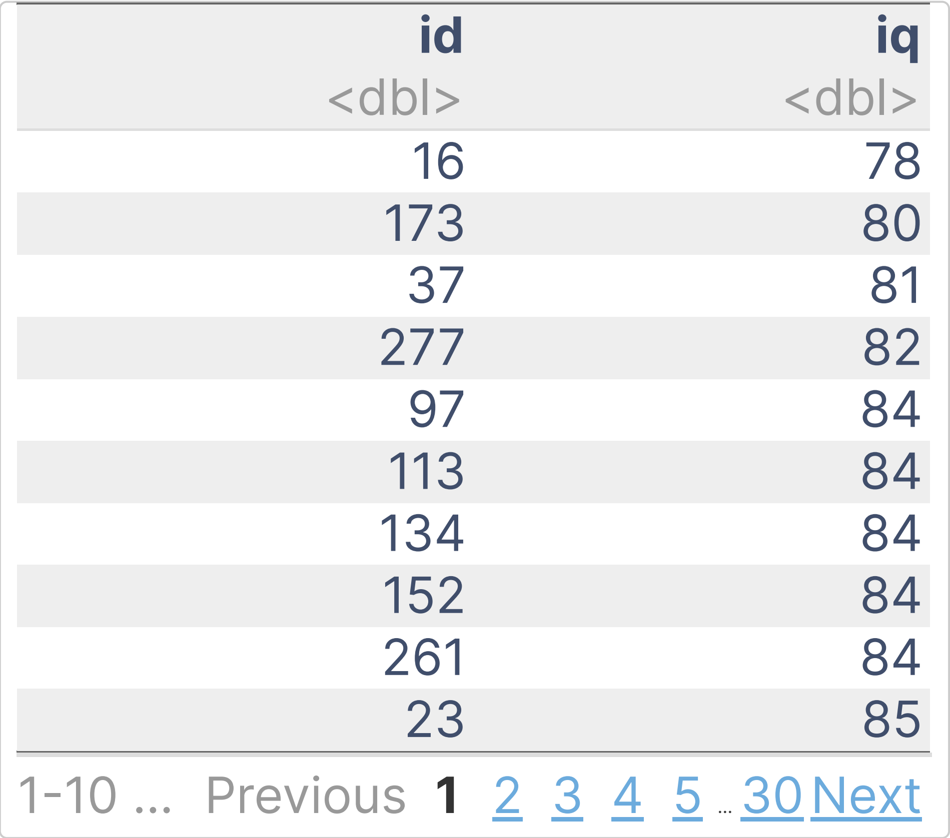


Winsorize or trim



Winsorize or trim

```
college %>%
  select(id, iq) %>%
  arrange(iq)
```



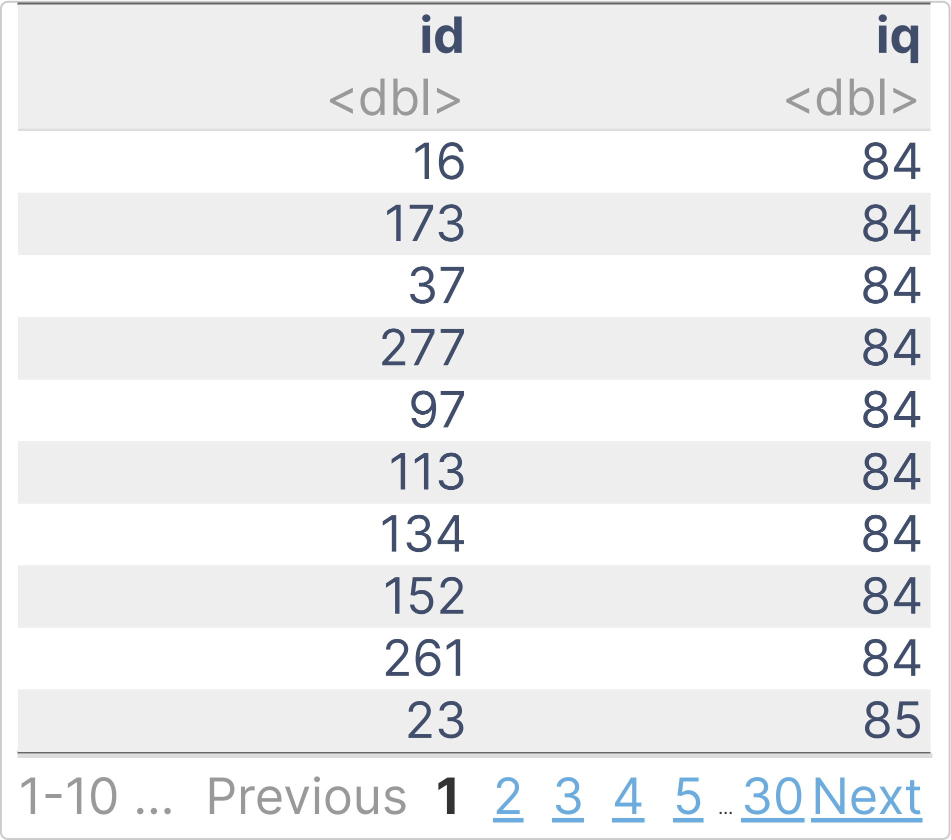
id <dbl>	iq <dbl>
16	78
173	80
37	81
277	82
97	84
113	84
134	84
152	84
261	84
23	85

1-10 ... Previous **1** 2 3 4 5 ... 30 Next

Winsorize or trim

```
college %>%
  select(id, iq) %>%
  arrange(iq) %>%
  mutate(iq = DescTools::Winsorize(iq,
                                   minval = 84,
                                   maxval = 125))
```

Winsorize() is from the DescTools package.



id <dbl>	iq <dbl>
16	84
173	84
37	84
277	84
97	84
113	84
134	84
152	84
261	84
23	85

1-10 ... Previous **1** 2 3 4 5 ... 30 Next

Transform the data

Corrects for positive skew and unequal variances:

- Log: `log(x)`
- Square Root: `sqrt(x)`
- Reciprocal: `(1/x)`

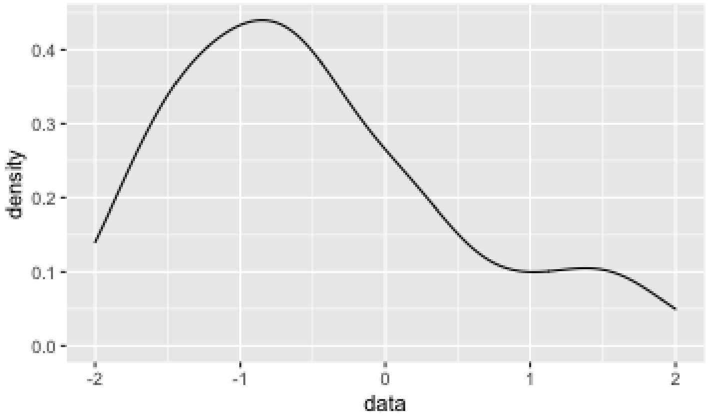
Corrects for negative skew:

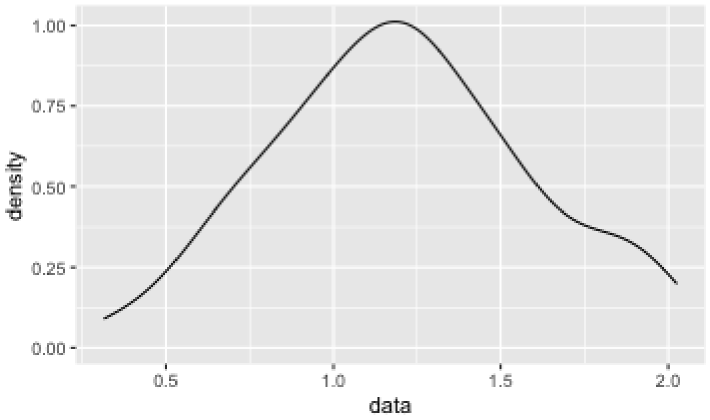
- Reverse score: Reverse the score `((MAX+1)-x)` then do one of the prior transformations.

Transform data: An example

```
posskew %>%
  ggplot(aes(data)) +
  geom_density()
```

```
posskew %>%
  mutate(data = sqrt(2.1 + data)) %>%
  ggplot(aes(data)) +
  geom_density()
```





??? Explain the 2.1 was because can't take square roots of zero or negative values, so needed to shift values up.

Non-parametric tests

- Used when removing outliers or transforming data still does not help you meet the assumptions of the parametric test
- Has less restrictive assumptions than parametric tests
- Does not rely on a parametric (i.e., normal) distribution!

Parametric Test	Non-parametric equivalent
Independent t-test	Mann-Whitney test
Dependent t-test	Wilcoxon Signed Range test
One-way ANOVA	Kruskal-Wallis test
Repeated-measures ANOVA	Friedman's ANOVA test

Your Turn

1. Try the various transformations on the `age` variable. Do any of them improve the normality of the variable? Test using the `shapiro_test()` function as you learned in the "Testing assumptions" lesson.