Standardizing R usage to improve student focus on statistical concepts

USCOTS 2025

Aaron Rendahl, PhD, University of Minnesota College of Veterinary Medicine

Motivation

- Students struggle to learn both R and statistics, in part due to the cognitive load to handle R idiosyncrasies.
- Standard R output often doesn't model best practices.

Since all of my students will need to understand statistics, but only some will need R…

Can I write a package to reduce the cognitive load of learning R and also demonstrate best practices, to improve student learning of statistical ideas?

Goals

Simpler input:

- Consistent formula notation
- Analyses by group and for multiple variables
- Simplify exploration of fitted models
- Hide most package usage
- More reasonable default behavior

Simpler output:

- Consistent output, in both console and Quarto
- Always show how variables were used
- Nice looking tables, with more useful labels and reasonable rounding
- Handle backtransformation more simply
- Continue to use tidyverse verbs for data manipulation and
  ggplot2 for graphics, adding helper functions as needed.

Fall 2024 Assessment

- Notably fewer students struggled with R
- Office hours were more focused on conceptual questions
- Less class time spent on R idiosyncrasies
- Improved understanding of statistical concepts and statistical reasoning and thinking skills
- However, no formal evaluation or comparison with past years

Concerns

- Maybe it does too much? Is figuring out the output useful for understanding?
- Not as easy for students to build on R skills later
- Yet another package with different notation and usage?
  - **mosaic**: my initial inspiration: it uses formula notation, but doesn't standardize output
  - **broom/gt**: makes nice tables, but adds coding complexity
  - **emmeans**: simplifies working with models, mostly nice syntax
  - **tidymodels**: some nice elements, but not traditional enough for my audience.

https://aaronrendahl.github.io/umncvmstats/

Included features:

One and Two Group Inference

- Standardized functions for one-sample, two-sample, paired, and pairwise inference for…
  - Proportions (automatically choosing a reasonable test)
  - Means, with possible log-transformed response
  - Non-parametric tests (Wilcoxon and Kruskal-Wallis)
- Correlation tests (Pearson, Spearman, Kendall)
- Allow these to be done for subgroups and for multiple responses and/or predictors without creating subsetted data frames or looping

Linear and Logistic Models

- Output for anova tables, summary statistics, coefficients
- Estimated model means, slopes, and pairwise differences
- Model means and predictions use similar syntax, and allow for back-transformation from both log responses and logistic models
- Diagnostic plots

Summary Statistics

- Incorporate selected gtsummary functionality

Power Calculations

- Power calculations for two-sample t-tests, for traditional power, equivalence tests, and desired margin of error

Output

- Combine results from multiple tests
- Control formatting of output, including rounding using either decimals or significant digits
- Can convert output tables to tibbles for plotting or saving
- Includes blank Quarto template with all necessary setup code, and also R version and citation information

Graphics

- Incorporate beeswarm graphics
- Simplify plots of data with a binary response
- Model diagnostic plots using ggplot graphics

Documentation

- Vignettes with examples for all major functions
- Explanation of how to get started with R and Quarto

Bonus

- A correlation guessing game
- Demonstrate regression diagnostics on randomly created data sets

Example 1: One sample proportion inference

What proportion of cars have a straight engine?

A traditional one-sample analysis…

```r
xtabs(~vs, data=mtcars2)
```

```
vs
V-shaped straight
      18       14
```

```r
xtabs(~vs, data=mtcars2) |> prop.test(correct = FALSE)
```

```

	1-sample proportions test without continuity correction

data:  xtabs(~vs, data = mtcars2), null probability 0.5
X-squared = 0.5, df = 1, p-value = 0.4795
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.3932559 0.7183467
sample estimates:
     p
0.5625
```

```r
xtabs(~vs, data=mtcars2) |> prop.test(correct = FALSE) |> broom::tidy()
```

```
# A tibble: 1 × 8
  estimate statistic p.value parameter conf.low conf.high method     alternative
     <dbl>     <dbl>   <dbl>     <int>    <dbl>     <dbl> <chr>      <chr>
1    0.562       0.5   0.480         1    0.393     0.718 1-sample … two.sided
```

Why multiple steps?

What's the proportion of?

Why are we doing a hypothesis test?

What output do I care about?

How many decimal places should I report?

A standardized one-sample analysis…

```r
one_proportion_inference(vs ~ 1, data=mtcars2)
```

response	x	n	proportion	SE	conf.low	conf.high
vs = straight	14	32	0.438	0.088	0.282	0.607

Wilson's proportion test (two.sided), with 95% confidence intervals.

Counts, proportions, and CI in one table

Clear what level the proportion is for

Makes better default choices

No hypothesis test

Chooses between Wilson's and Clopper-Pearson

Round to have two significant digits in SE

Example 2: Two sample proportion inference

How does engine type depend on transmission type?

A traditional two-sample analysis…

```
xtabs(~ am + vs, data=mtcars2)
```

```
           vs
am          V-shaped straight
  automatic       12        7
  manual           6        7
```

```
xtabs(~ am + vs, data=mtcars2) |> prop.test()
```

```

	2-sample test for equality of proportions with continuity correction

data:  xtabs(~am + vs, data = mtcars2)
X-squared = 0.34754, df = 1, p-value = 0.5555
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.2418423  0.5819233
sample estimates:
   prop 1    prop 2 
0.6315789 0.4615385 
```

Which groups are prop 1 and prop 2?

What direction was the comparison done?

What's the difference in proportion?

What output do I care about? How many decimals?

A standardized two-sample analysis…

```
one_proportion_inference(vs ~ 1 + am, data=mtcars2)
```

response	variable	x	n	proportion	SE	conf.low	conf.high
vs = straight		14	32	0.438	0.088	0.282	0.607
vs = straight	am = automatic	7	19	0.37	0.11	0.19	0.59
vs = straight	am = manual	7	13	0.54	0.14	0.29	0.77

Wilson's proportion test (two.sided), with 95% confidence intervals.

```
two_proportion_inference(vs ~ am, data=mtcars2)
```

response	variable	difference	SE	conf.low	conf.high	chisq.value	p.value
vs = straight	am: automatic - manual	−0.17	0.18	−0.58	0.24	0.348	0.56

2-sample test for equality of proportions with continuity correction (two.sided), with 95% confidence intervals.

Easily get proportions and CIs overall and by group

Output in clear tables (optionally combined into one)

Clear what level the proportion is for, and which direction the difference is

Round to have two significant digits in SE

Example 3: One and two-sample t inference, with log transformation

How does the car weight depend on transmission type?

A possible traditional analysis…

```
t.test(log(wt) ~ am, data=mtcars2) |> broom::tidy()
```

```
# A tibble: 1 × 10
  estimate estimate1 estimate2 statistic   p.value parameter conf.low conf.high
     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1    0.459      1.31     0.849      5.40 0.0000242      20.8    0.282     0.636
# ℹ 2 more variables: method <chr>, alternative <chr>
```

Which groups are these estimates for?

Would you really report that p-value?

```
t.test(log(wt) ~ am, data=mtcars2) |> broom::tidy() |>
  mutate(across(c(starts_with("estimate"), starts_with("conf")), exp))
```

```
# A tibble: 1 × 10
  estimate estimate1 estimate2 statistic   p.value parameter conf.low conf.high
     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1     1.58      3.70      2.34      5.40 0.0000242      20.8     1.33      1.89
# ℹ 2 more variables: method <chr>, alternative <chr>
```

How would you have your students code this back-transformation?

```
mtcars2 |> nest(data=-am) |>
  mutate(map_dfr(data, \(x)
    t.test(log(wt)~1, data=x) |> broom::tidy())) |>
  select(-data)
```

```
# A tibble: 2 × 9
  am        estimate statistic  p.value parameter conf.low conf.high method
  <fct>        <dbl>     <dbl>    <dbl>     <dbl>    <dbl>     <dbl> <chr>
1 manual       0.849      11.7 6.28e- 8        12    0.691      1.01 One Sample…
2 automatic    1.31       29.4 1.17e-16        18    1.21       1.40 One Sample…
# ℹ 1 more variable: alternative <chr>
```

Best practice is to report estimates for each group as well as the difference; how would you have them code this?

Standardized analysis on next page…

Example 3 continued…

A standardized analysis…

```
combine_tests(
  one_t_inference(log(wt) ~ am, data = mtcars2, backtransform = FALSE),
  two_t_inference(log(wt) ~ am, data = mtcars2, backtransform = FALSE))
```

response	variable	n	mean	difference	SE	df	conf.low	conf.high	null	t.value	p.value	footn
log(wt)	am = automatic	19	1.308		0.045	18.0	1.215	1.402				
log(wt)	am = manual	13	0.849		0.072	12.0	0.691	1.007				
log(wt)	am: automatic - manual			0.459	0.085	20.8	0.282	0.636	0.000	5.40	< 0.0001	

[1] One Sample t-test (two.sided), with 95% confidence intervals.
[2] Welch Two Sample t-test (two.sided), with 95% confidence intervals.

Use the same formula notation to get estimates and CIs for each group, and for the difference.

```
combine_tests(
  one_t_inference(log(wt) ~ am, data = mtcars2),
  two_t_inference(log(wt) ~ am, data = mtcars2))
```

response	variable	n	mean	ratio	SE	df	conf.low	conf.high	null	t.value	p.value	footnote
wt	am = automatic	19	3.70		0.16	18.0	3.37	4.06				1,2
wt	am = manual	13	2.34		0.17	12.0	2.00	2.74				1,2
wt	am: automatic / manual			1.58	0.13	20.8	1.33	1.89	1.00	5.40	< 0.0001	3,4

[1] One Sample t-test (two.sided), with 95% confidence intervals.
[2] Results are backtransformed from the log scale (that is, the geometric mean is reported), and the standard error is estimated using the delta method.
[3] Welch Two Sample t-test (two.sided), with 95% confidence intervals.
[4] Results are backtransformed from the log scale (that is, the ratio is reported), and the standard error is estimated using the delta method.

Back-transformation is built in, to keep the focus on what it means, not how to code it.

Example 4: Pairwise t-tests, ANOVA

How does the mpg depend the number of cylinders?

Pairwise t-tests:

```
combine_tests(
  one_t_inference(mpg ~ cyl, data=mtcars2),
  pairwise_t_inference(mpg ~ cyl, data=mtcars2)) |>
  as_gt() |> tab_compact()
```

response	variable	n	mean	difference	SE	df	conf.low	conf.high	null	t.value	p.value	p.adjust	footnote
mpg	cyl = 4	11	26.7		1.4	10.0	23.6	29.7					[1]
mpg	cyl = 6	7	19.74		0.55	6.00	18.40	21.09					[1]
mpg	cyl = 8	14	15.10		0.68	13.0	13.62	16.58					[1]
mpg	cyl: 4 - 6			6.9	1.5	13.0	2.9	10.9	0.000	4.72	0.0004	0.0012	[2,3]
mpg	cyl: 4 - 8			11.6	1.5	15.0	7.5	15.7	0.000	7.60	< 0.0001	< 0.0001	[2,3]
mpg	cyl: 6 - 8			4.64	0.88	18.5	2.33	6.95	0.000	5.29	< 0.0001	0.0001	[2,3]

[1] One Sample t-test (two.sided), with 95% confidence intervals.
[2] Welch Two Sample t-test (two.sided), with 95% confidence intervals, adjusted for 3 comparisons using the Bonferroni method.
[3] p-values adjusted for 3 multiple comparisons using the Bonferroni method.

ANOVA, with model means and predictions:

```
mpg_model <- lm(mpg ~ cyl, data=mtcars2)
model_anova(mpg_model)
```

mpg ~ cyl

term	df	sumsq	meansq	F	p.value
cyl	2	825	412	39.7	< 0.0001
Residuals	29	301	10.4		

```
combine_tests(
  model_means(mpg_model, ~ cyl),
  pairwise_model_means(mpg_model, ~ cyl))
```

mpg ~ cyl

cyl	contrast	emmean	estimate	SE	df	conf.low	conf.high	t.ratio	p.value	cld.group	footnote
8		15.10		0.86	29	13.34	16.86			a	[1,2,3]
6		19.7		1.2	29	17.3	22.2			b	[1,2,3]
4		26.66		0.97	29	24.68	28.65			c	[1,2,3]
	cyl4 - cyl6		6.9	1.6	29	3.1	10.8	4.44	0.0003		[1,4,2]
	cyl4 - cyl8		11.6	1.3	29	8.4	14.8	8.90	< 0.0001		[1,4,2]
	cyl6 - cyl8		4.6	1.5	29	1.0	8.3	3.11	0.011		[1,4,2]

[1] Confidence level used: 0.95
[2] P value adjustment: tukey method for comparing a family of 3 estimates
[3] significance level used: alpha = 0.05
[4] Conf-level adjustment: tukey method for comparing a family of 3 estimates

Continued on next page…

```
model_predictions(mpg_model, at=list(cyl=c('8', '6', '4')))
```

mpg ~ cyl			
cyl	prediction	predict.low	predict.high
8	15.1	8.3	21.9
6	19.7	12.7	26.8
4	26.7	19.8	33.5
Prediction level used: 0.95			

Example 5: Multiple tests

How does the car weight AND mpg depend on transmission type?

```
combine_tests(
  one_t_inference(wt + mpg ~ am, data = mtcars2),
  two_t_inference(wt + mpg ~ am, data = mtcars2))
```

response	variable	n	mean	difference	SE	df	conf.low	conf.high	null	t.value	p.value	footnot
mpg	am = automatic	19	17.15		0.88	18.0	15.30	19.00				
mpg	am = manual	13	24.4		1.7	12.0	20.7	28.1				
mpg	am: automatic - manual			−7.2	1.9	18.3	−11.3	−3.2	0.000	-3.77	0.0014	
wt	am = automatic	19	3.77		0.18	18.0	3.39	4.14				
wt	am = manual	13	2.41		0.17	12.0	2.04	2.78				
wt	am: automatic - manual			1.36	0.25	29.2	0.85	1.86	0.000	5.49	< 0.0001	

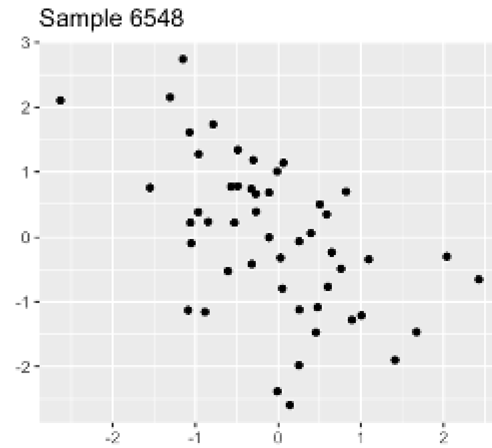
[1] One Sample t-test (two.sided), with 95% confidence intervals.
[2] Welch Two Sample t-test (two.sided), with 95% confidence intervals.

Bonus Features:

Correlation Guessing Game:

- Various random patterns to build intuition about correlation
- Strength, direction, linearity, and shape all randomly vary

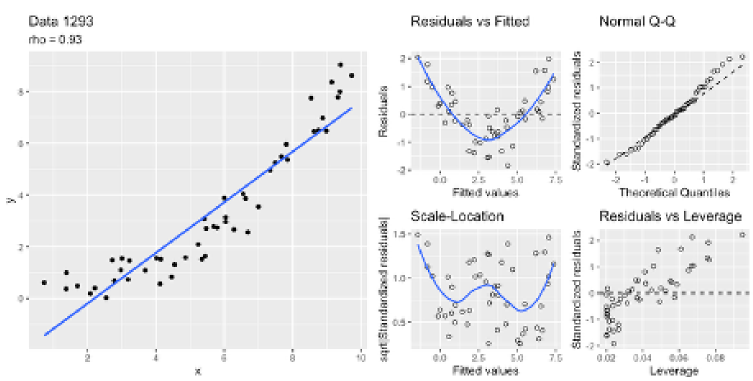
```
> guess_cor()
Sample 6548
What is the strength, direction,
   linearity, and shape?
What do you think the correlation is?
...
You guessed ___, it was -0.54.
Hit enter for another random sample.
  [Type a number for that sample.
  Type X to quit.]
```



Model Diagnostics Sampler:

- What do the patterns in the diagnostic plots really mean?
- Try a bunch of models with data of various patterns and build your intuition.
- For discussion, specific samples can be recreated using the sample code.

```
> sample_lm()
```




```
> sample_lm()
```