

{gtsummary}

What is `{gtsummary}`?

- Create tables that are publication-ready
- Highly customizable
- Descriptive tables, regression tables, etc.



gtsummary::tbl_summary()

```
1 library(gtsummary)
2
3 tbl_summary(
4   nlsy,
5   by = sex_cat,
6   include = c(sex_cat, race_ethnicity_cat,
7               eyesight_cat, glaucoma_cat)
```

```
1 tbl_summary(  
2   nlsy,  
3   by = sex_cat,  
4   include = c(sex_cat, race_eth_cat, region_cat,  
5               eyesight_cat, glasses, age_bir)  
6   label = list(  
7     race_eth_cat ~ "Race/ethnicity",  
8     region_cat ~ "Region",  
9     eyesight_cat ~ "Eyesight",  
10    glasses ~ "Wears glasses",  
11    age_bir ~ "Age at first birth"  
12  ),  
13  missing_text = "Missing")
```

```

1  tbl_summary(
2    nlsy,
3    by = sex_cat,
4    include = c(sex_cat, race_eth_cat,
5                eyesight_cat, glasses, age_bir)
6    label = list(
7      race_eth_cat ~ "Race/ethnicity",
8      eyesight_cat ~ "Eyesight",
9      glasses ~ "Wears glasses",
10     age_bir ~ "Age at first birth"
11   ),
12   missing_text = "Missing") |>
13   add_p(test = list(all_continuous() ~ "t.test",
14                    all_categorical() ~ "chi.sq")) |>
15   add_overall(col_label = "**Total**") |>
16   bold_labels() |>
17   modify_footnote(update = everything() ~ NA)
18   modify_header(label = "**Variable**", p.val

```

`tbl_summary()`

- Incredibly customizable
- Really helpful with Table 1
- I often just view in the web browser and copy and paste into a Word document
- Can also be used within quarto/R Markdown¹
- If output is Word, I use `as_flex_table()`
- Make even more customizable with `as_gt()`
 - Then can output to Word with `gt::as_word()`

1. This is where the bug comes in: soon to be fixed!

Univariate regressions

Fit a series of univariate regressions of income on other variables.

```
1 tbl_uvregression(  
2   nlsy,  
3   y = income,  
4   include = c(sex_cat, race_ethnicity_cat, eyesight_cat, income_cat),  
5   method = lm)
```

Can also do logistic regression

```
1 tbl_uvregression(  
2   nlsy,  
3   y = glasses,  
4   include = c(sex_cat, race_ethnicity_cat, eyesight_cat, glaucoma_cat),  
5   method = glm,  
6   method.args = list(family = "binomial"),  
7   exponentiate = TRUE)
```

Customizable just like `tbl_summary()`

Some regressions

```
1 linear_model <- lm(income ~ sex_cat + age_bir + race_eth_cat,  
2                   data = nlsy)
```

```
1 linear_model_int <- lm(income ~ sex_cat*age_bir + race_eth_cat,  
2                   data = nlsy)
```

```
1 logistic_model <- glm(glasses ~ eyesight_cat + sex_cat + income,  
2                   data = nlsy, family = binomial())
```

gtsummary::tbl_regression()

```
1 tbl_regression(  
2   linear_model,  
3   intercept = TRUE,  
4   label = list(  
5     sex_cat ~ "Sex",  
6     race_eth_cat ~ "Race/ethnicity",  
7     age_bir ~ "Age at first birth"  
8   ))
```

```
1 tbl_regression(  
2   logistic_model,  
3   exponentiate = TRUE,  
4   label = list(  
5     sex_cat ~ "Sex",  
6     eyesight_cat ~ "Eyesight",  
7     income ~ "Income"  
8   ))
```

You could put several together

```
1 tbl_no_int <- tbl_regression(  
2   linear_model,  
3   intercept = TRUE,  
4   label = list(  
5     sex_cat ~ "Sex",  
6     race_eth_cat ~ "Race/ethnicity",  
7     age_bir ~ "Age at first birth"  
8   ))  
9  
10 tbl_int <- tbl_regression(  
11   linear_model_int,  
12   intercept = TRUE,  
13   label = list(  
14     sex_cat ~ "Sex"
```

You could put several together

```
1 tbl_merge(list(tbl_no_int, tbl_int),  
2           tab_spanner = c("**Model 1**", "**Model 2**"))
```

Inline text

bonus: `broom::tidy()`

```
1 bind_rows(  
2   broom::tidy(linear_model, conf.int = TRUE),  
3   broom::tidy(linear_model_int, conf.int = TRUE),  
4   .id = "model"  
5 ) |>  
6 mutate(model = factor(model,  
7   labels = c("main terms", "sex-age interact  
8   ggplot(aes(x = term, y = estimate,  
9     ymin = conf.low, ymax = conf.high,  
10    color = model)) +  
11   geom_point(position = position_dodge(width = .5)) +  
12   geom_errorbar(position = position_dodge(width = .5))
```

