Reporting Research Study Results with {gtsummary}

Daniel D. Sjoberg

November 16, 2021

Memorial Sloan Kettering
Cancer Center

@statistishdan

@ddsjoberg

Motivation

The reproducibility crisis

- Quality of medical research is often low
- **Low quality code** in medical research part of the problem
- Low quality code is more likely to **contain errors**
- Reproducibility is often **cumbersome** and **time-consuming**

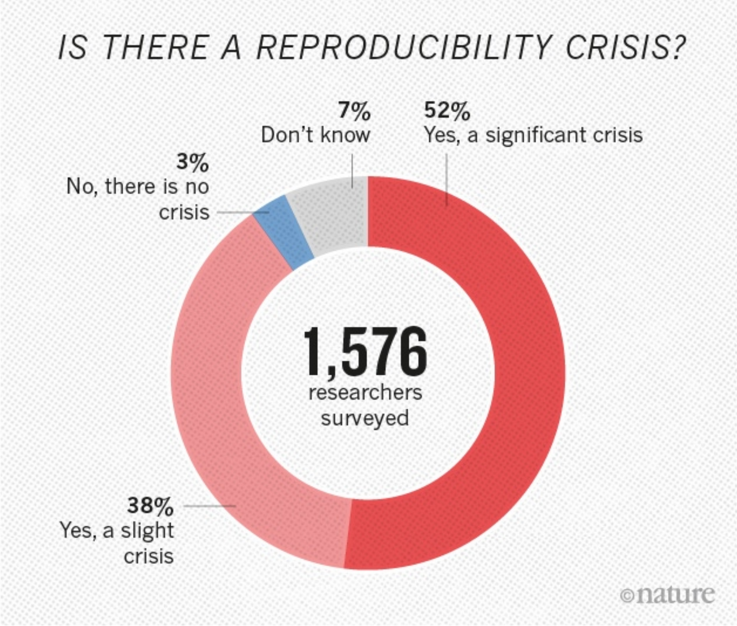


Image source: https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970

{gtsummary} overview

- Create **tabular summaries** with sensible defaults but highly customizable

- Types of summaries:

    - "Table 1"-types
    - Cross-tabulation
    - Regression models
    - Survival data
    - Survey data
    - Custom tables

- Report statistics from {gtsummary} tables **inline** in R Markdown

- Stack and/or merge any table type
- Use **themes** to standardize across tables
- Choose from different **print**

{gtsummary} example dataset

- The `trial` dataset is included with {gtsummary}
- Simulated dataset of baseline characteristics for 200 patients who receive Drug A or Drug B
- Variables were assigned labels using the `labelled` package

```r
library(gtsummary)
library(tidyverse)

head(trial)
```

```
## # A tibble: 6 x 8
##   trt      age marker stage grade response death ttdeath
##   <chr>  <dbl>  <dbl> <fct> <fct>    <int> <int>   <dbl>
## 1 Drug A    23  0.16  T1    II           0     0    24  
## 2 Drug B     9  1.11  T2    I            1     0    24  
## 3 Drug A    31  0.277 T1    II           0     0    24  
## 4 Drug A    NA  2.07  T3    III          1     1    17.6
## 5 Drug A    51  2.77  T4    III          1     1    16.4
## 6 Drug B    39  0.613 T4    I            0     1    15.6
```

{gtsummary} example dataset

This presentation will use a subset of the variables.

```
sm_trial <-
  trial %>%
  select(trt, age, grade, response)
```

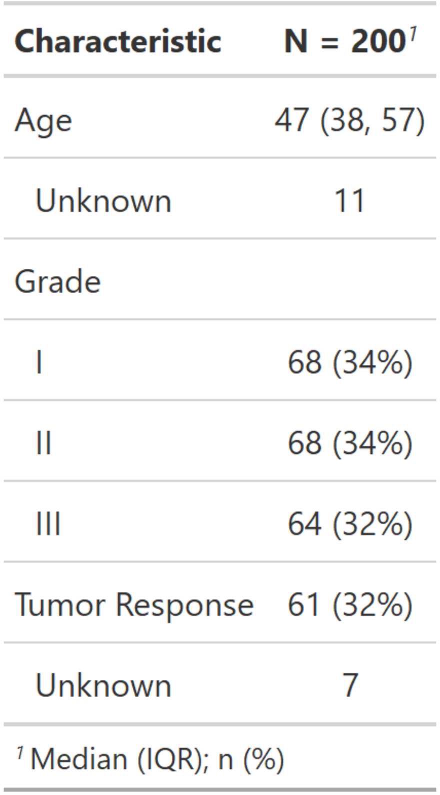
Variable	Label
trt	Chemotherapy Treatment
age	Age
grade	Grade
response	Tumor Response

tbl_summary()

Basic tbl_summary()

```
tbl_summary_1 <-
  sm_trial %>%
  select(-trt) %>%
  tbl_summary()
```

- Four types of summaries: `continuous`, `continuous2`, `categorical`, and `dichotomous`
- Statistics are `median (IQR)` for continuous, `n (%)` for categorical/dichotomous
- Variables coded `0/1`, `TRUE/FALSE`, `Yes/No` treated as dichotomous
- Lists `NA` values under "Unknown"
- Label attributes are printed automatically



Characteristic	**N = 200**[1]
Age	47 (38, 57)
Unknown	11
Grade	
I	68 (34%)
II	68 (34%)
III	64 (32%)
Tumor Response	61 (32%)
Unknown	7

[1] Median (IQR); n (%)

Customize tbl_summary() output

```
tbl_summary_2 <-  sm_trial %>%
  tbl_summary(
    by = trt,
    type = age ~ "continuous2",
    statistic =
      list(age ~ c("{mean} ({sd})",
           response ~ "{n} / {N} ({
    label = grade ~ "Pathologic tum
    digits = age ~ 1
  )
```

Characteristic	**Drug A**, N = 98	**Drug B**, N = 102
Age		
Mean (SD)	47.0 (14.7)	47.4 (14.0)
Range	6.0, 78.0	9.0, 83.0
Unknown	7	4
Pathologic tumor grade		
I	35 (36%)	33 (32%)
II	32 (33%)	36 (35%)
III	31 (32%)	33 (32%)
Tumor Response	28 / 95 (29%)	33 / 98 (34%)
Unknown	3	4

- `by`: specifies a column variable for cross-tabulation
- `type`: specifies the summary type
- `statistic`: customize the reported statistics
- `label`: change or customize

{gtsummary} + formulas

select variables

give instructions

```
sm_trial %>%
tbl_summary(
  label       =        age         ~  "Patient Age",
  type        =    c(age, marker)  ~   "continuous",
  digits      =  starts_with("age") ~        0,
  statistic   =   all_continuous() ~  "{mean} ({sd})"
)
```

`Use lists` to pass ≥2 sets of instruction:

```
label = list(age ~ "Patient Age", marker ~ "Marker Level")
```

Add-on functions in {gtsummary}

`tbl_summary()` objects can also be updated using related functions.

- `add_*()` add additional column of statistics or information, e.g. p-values, q-values, overall statistics, treatment differences, N obs., and more
- `modify_*()` modify table headers, spanning headers, footnotes, and more
- `bold_*()`/`italicize_*()` style labels, variable levels, significant p-values

Update tbl_summary() with add_*()

```
tbl_summary_3a <-
  sm_trial %>%
  tbl_summary(
    by = trt
    ) %>%
  add_p() %>%
  add_q(method = "bonferroni")
```

- `add_p()`: adds a column of p-values
- `add_q()`: adds a column of p-values adjusted for multiple comparisons through a call to `p.adjust()`

Characteristic	**Drug A**, N = 98[1]	**Drug B**, N = 102[1]	**p-value**[2]	**q-value**[3]
Age	46 (37, 59)	48 (39, 56)	0.7	>0.9
Unknown	7	4		
Grade			0.9	>0.9
I	35 (36%)	33 (32%)		
II	32 (33%)	36 (35%)		
III	31 (32%)	33 (32%)		
Tumor Response	28 (29%)	33 (34%)	0.5	>0.9
Unknown	3	4		

[1] Median (IQR); n (%)

[2] Wilcoxon rank sum test; Pearson's Chi-squared test

[3] Bonferroni correction for multiple testing

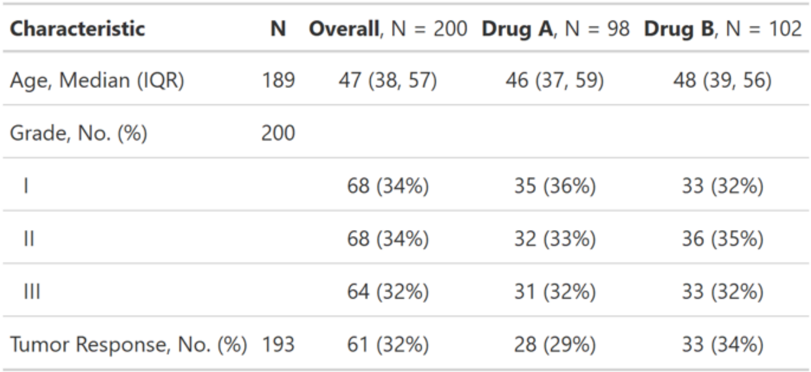
Update tbl_summary() with add_*()

```
tbl_summary_3ab <-
  trial %>%
  select(trt, marker, response) %>%
  tbl_summary(
    by = trt,
    statistic = list(marker ~ "{mea
                     response ~ "{p
    missing = "no"
    ) %>%
  add_difference()
```

- `add_difference()`: mean and rate differences between two groups. Can also be adjusted differences

Update tbl_summary() with add_*()

```
tbl_summary_3b <-
  sm_trial %>%
  tbl_summary(
    by = trt,
    missing = "no"
    ) %>%
  add_overall() %>%
  add_n() %>%
  add_stat_label(
    label = all_categorical() ~ "Nc
  )
```

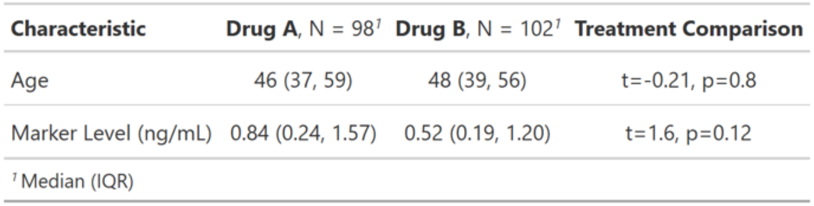


Characteristic	**N**	**Overall**, N = 200	**Drug A**, N = 98	**Drug B**, N = 102
Age, Median (IQR)	189	47 (38, 57)	46 (37, 59)	48 (39, 56)
Grade, No. (%)	200			
I		68 (34%)	35 (36%)	33 (32%)
II		68 (34%)	32 (33%)	36 (35%)
III		64 (32%)	31 (32%)	33 (32%)
Tumor Response, No. (%)	193	61 (32%)	28 (29%)	33 (34%)

- `add_overall()`: adds a column of overall statistics
- `add_n()`: adds a column with the sample size
- `add_stat_label()`: adds a

Update tbl_summary() with add_*()

```
tbl_summary_3b <-
  sm_trial %>%
  select(age, marker, trt)
  tbl_summary(
    by = trt,
    missing = "no"
    ) %>%
  add_stat(...)
```



Characteristic	**Drug A**, N = 98[1]	**Drug B**, N = 102[1]	**Treatment Comparison**
Age	46 (37, 59)	48 (39, 56)	t=-0.21, p=0.8
Marker Level (ng/mL)	0.84 (0.24, 1.57)	0.52 (0.19, 1.20)	t=1.6, p=0.12

[1] Median (IQR)

- Customize statistics presented with `add_stat()`
    - Added statistics can be placed on the label or the level rows
    - Added statistics may be a single column or multiple

Update with bold_*()/italicize_*()

```
tbl_summary_4 <-
  sm_trial %>%
  tbl_summary(
    by = trt
    ) %>%
  add_p() %>%
  bold_labels() %>%
  italicize_levels() %>%
  bold_p(t = 0.8)
```

- `bold_labels()`: bold the variable labels
- `italicize_levels()`: italicize the variable levels
- `bold_p()`: bold p-values according a specified threshold

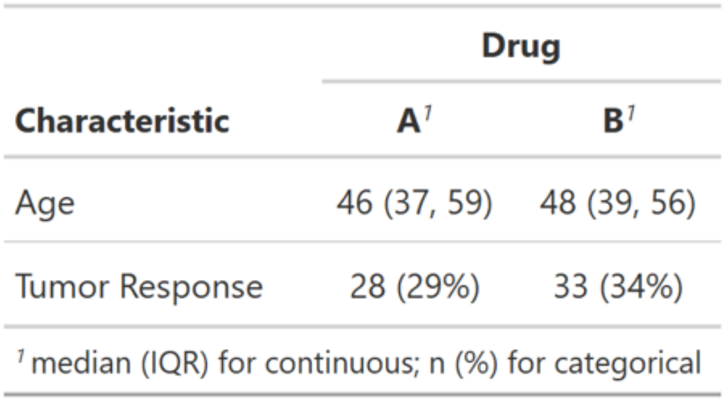
Characteristic	**Drug A**, N = 98[1]	**Drug B**, N = 102[1]	**p-value**[2]
Age	46 (37, 59)	48 (39, 56)	**0.7**
Unknown	7	4	
Grade			0.9
I	35 (36%)	33 (32%)	
II	32 (33%)	36 (35%)	
III	31 (32%)	33 (32%)	
Tumor Response	28 (29%)	33 (34%)	**0.5**
Unknown	3	4	

[1] Median (IQR); n (%)

[2] Wilcoxon rank sum test; Pearson's Chi-squared test

Update tbl_summary() with modify_*()

```
tbl_summary_5 <-
  sm_trial %>% select(age, response
  tbl_summary(
    by = trt, missing = "no"
    ) %>%
  modify_header(
    update = list(
      stat_1 ~ "**A**",
      stat_2 ~ "**B**"
    )) %>%
  modify_spanning_header(
    update = all_stat_cols() ~ "**D
  modify_footnote(
    update = all_stat_cols() ~
      "median (IQR) for continuous;
  )
```



	Drug	
Characteristic	**A**[1]	**B**[1]
Age	46 (37, 59)	48 (39, 56)
Tumor Response	28 (29%)	33 (34%)

[1] median (IQR) for continuous; n (%) for categorical

- Use `show_header_names()` to see the internal header names available for use in

Add-on functions in {gtsummary}

And many more!

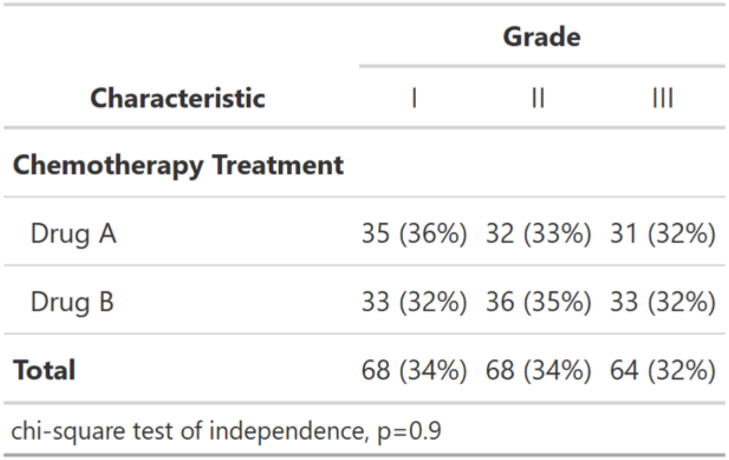
See the documentation at
http://www.danieldsjoberg.com/gtsummary/reference/index.html

And a detailed `tbl_summary()` vignette at
http://www.danieldsjoberg.com/gtsummary/articles/tbl_summary.html

Cross-tabulation with tbl_cross()

`tbl_cross()` is a wrapper for `tbl_summary()` for **n x m** tables

```
tbl_cross_1 <-
  sm_trial %>%
  tbl_cross(
    row = trt,
    col = grade,
    percent = "row",
    margin = "row"
  ) %>%
  add_p(source_note = TRUE)
```



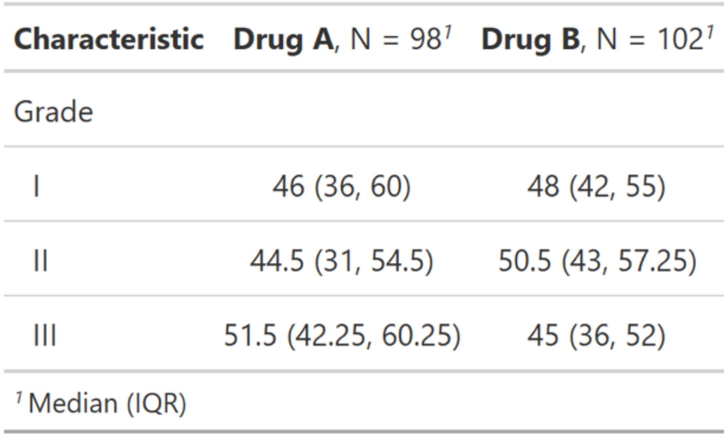
	Grade		
Characteristic	I	II	III
Chemotherapy Treatment			
Drug A	35 (36%)	32 (33%)	31 (32%)
Drug B	33 (32%)	36 (35%)	33 (32%)
Total	68 (34%)	68 (34%)	64 (32%)

chi-square test of independence, p=0.9

Continuous Summaries with tbl_continuous()

`tbl_continuous()` summarizes a continuous variable by 1, 2, or more categorical variables

```
tbl_continuous_1 <-
  sm_trial %>%
  tbl_continuous(
    data = trial,
    variable = age,
    by = trt,
    include = grade
  )
```

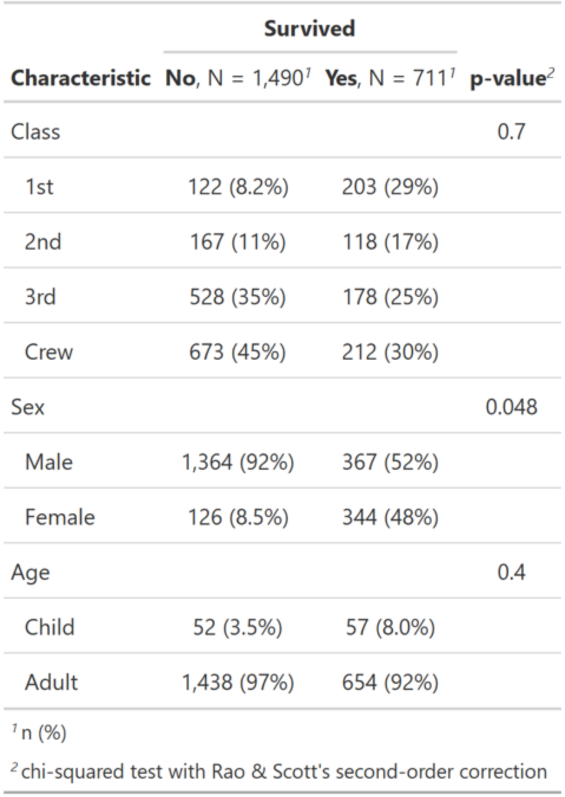


Characteristic	**Drug A**, N = 98[1]	**Drug B**, N = 102[1]
Grade		
I	46 (36, 60)	48 (42, 55)
II	44.5 (31, 54.5)	50.5 (43, 57.25)
III	51.5 (42.25, 60.25)	45 (36, 52)

[1] Median (IQR)

Survey data with tbl_svysummary()

```r
tbl_svysummary <-
  survey::svydesign(
    ids = ~1,
    data = as.data.frame(Titanic),
    weights = ~Freq
  ) %>%
  tbl_svysummary(by = Survived) %>%
  add_p() %>%
  modify_spanning_header(
    all_stat_cols() ~ "**Survived**
```



	Survived		
Characteristic	**No**, N = 1,490[1]	**Yes**, N = 711[1]	**p-value**[2]
Class			0.7
1st	122 (8.2%)	203 (29%)	
2nd	167 (11%)	118 (17%)	
3rd	528 (35%)	178 (25%)	
Crew	673 (45%)	212 (30%)	
Sex			0.048
Male	1,364 (92%)	367 (52%)	
Female	126 (8.5%)	344 (48%)	
Age			0.4
Child	52 (3.5%)	57 (8.0%)	
Adult	1,438 (97%)	654 (92%)	

[1] n (%)

[2] chi-squared test with Rao & Scott's second-order correction

Survival outcomes with tbl_survfit()

```r
library(survival)
fit <- survfit(Surv(ttdeath, death) ~ trt, trial)

tbl_survfit_1 <-
  tbl_survfit(
    fit,
    times = c(12, 24),
    label_header = "**{time} Month**"
  ) %>%
  add_p()
```

Characteristic	**12 Month**	**24 Month**	**p-value**[1]
Chemotherapy Treatment			0.2
Drug A	91% (85%, 97%)	47% (38%, 58%)	
Drug B	86% (80%, 93%)	41% (33%, 52%)	

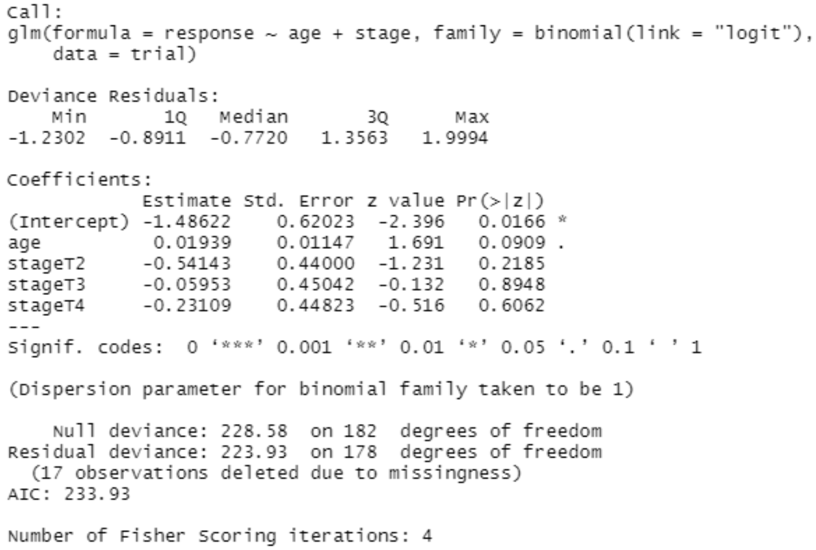
[1] Log-rank test

tbl_regression()

Traditional model summary()

```
m1 <-
  glm(
    response ~ age + stage,
    data = trial,
    family = binomial(link = "logit
  )

summary(m1)
```

Looks **messy** and it's not easy for others to understand.

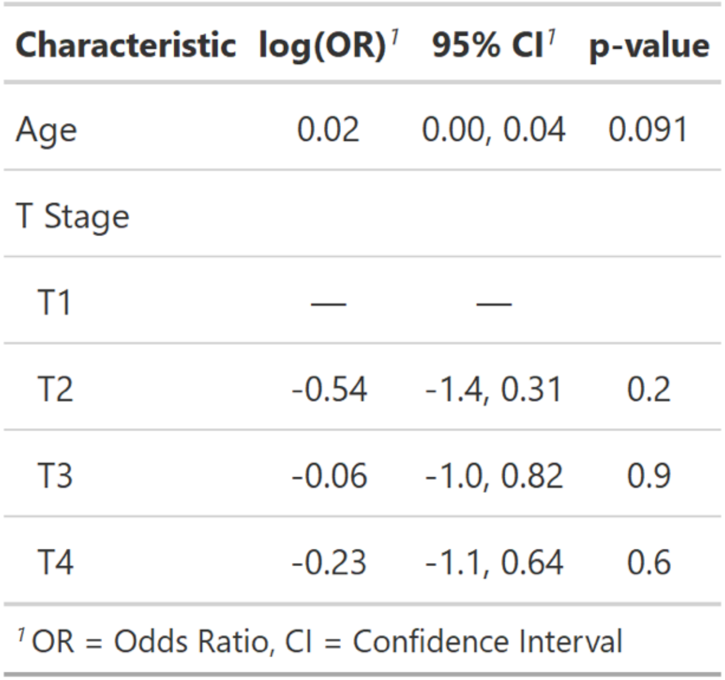


```
Call:
glm(formula = response ~ age + stage, family = binomial(link = "logit"),
    data = trial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2302  -0.8911  -0.7720   1.3563   1.9994

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.48622    0.62023  -2.396   0.0166 *
age          0.01939    0.01147   1.691   0.0909 .
stageT2     -0.54143    0.44000  -1.231   0.2185
stageT3     -0.05953    0.45042  -0.132   0.8948
stageT4     -0.23109    0.44823  -0.516   0.6062
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 228.58  on 182  degrees of freedom
Residual deviance: 223.93  on 178  degrees of freedom
  (17 observations deleted due to missingness)
AIC: 233.93

Number of Fisher Scoring iterations: 4
```

Basic tbl_regression()

```
m1_tbl_1 <-
  tbl_regression(
    m1
  )
```

- Displays **p-values** for covariates
- Shows **reference levels** for categorical variables



Characteristic	log(OR)[1]	95% CI[1]	p-value
Age	0.02	0.00, 0.04	0.091
T Stage			
T1	—	—	
T2	-0.54	-1.4, 0.31	0.2
T3	-0.06	-1.0, 0.82	0.9
T4	-0.23	-1.1, 0.64	0.6

[1] OR = Odds Ratio, CI = Confidence Interval

Customize tbl_regression() output

```
m1_tbl_2 <-
  tbl_regression(
    m1,
    exponentiate = TRUE
  ) %>%
  add_global_p() %>%
  add_glance_table(
    include = c(nobs, logLik, AIC,
```

- Display **odds ratio** estimates and **confidence intervals**
- Add global p-values
- Add various model statistics

Characteristic	OR[1]	95% CI[1]	p-value
Age	1.02	1.00, 1.04	0.087
T Stage			0.6
T1	—	—	
T2	0.58	0.24, 1.37	
T3	0.94	0.39, 2.28	
T4	0.79	0.33, 1.90	
No. Obs.	183		
Log-likelihood	-112		
AIC	234		
BIC	250		

[1] OR = Odds Ratio, CI = Confidence Interval

Supported models in tbl_regression()

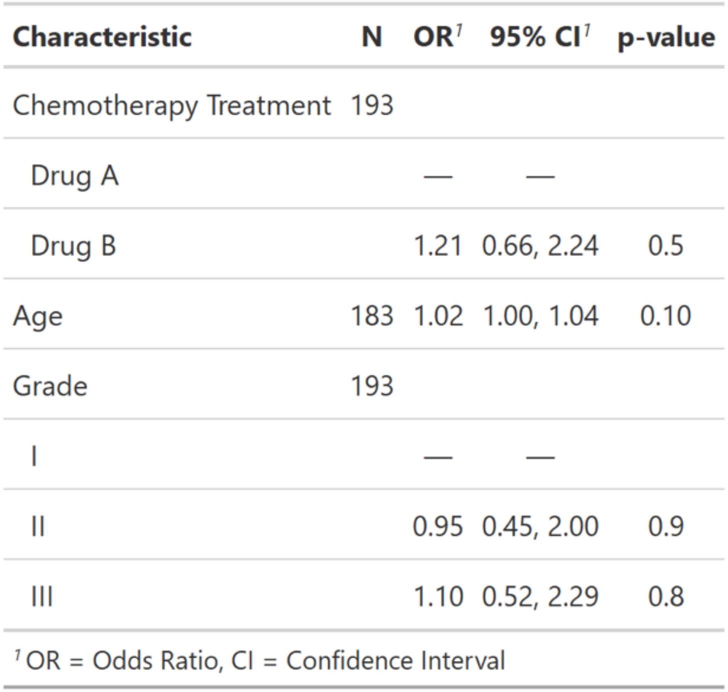
`stats::lm()`, `stats::glm()`, `stats::aov()`, `ordinal::clm()`, `ordinal::clmm()`, `survival::coxph()`, `survival::survreg()`, `survival::clogit()`, `lme4::lmer()`, `lme4::glmer()`, `lme4::glmer.nb()`, `brms::brm()`, `geepack::geeglm()`, `gam::gam()`, `mgcv::gam()`, `nnet::multinom()`, `survey::svyglm()`, `survey::svycoxph()`, `survey::svyolr()`, `MASS::polr()`, `MASS::glm.nb()`, `mice::mira`, `lavaan::lavaan()`, `cmprsk::crr()`, `stats::nls()`, `lfe::felm()`, `rstanarm::stan_glm()`, `VGAM::vglm()`, and more!

Custom tidiers can be written and passed to `tbl_regression()` using the `tidy_fun=` argument.

Univariate models with tbl_uvregression()

```
tbl_uvreg <-
  sm_trial %>%
  tbl_uvregression(
    method = glm,
    y = response,
    method.args = list(family = bir
    exponentiate = TRUE
  )
```

- Specify model `method`, `method.args`, and the `response` variable
- Arguments and helper functions like `exponentiate`, `bold_*()`, `add_global_p()` can also be used with `tbl_uvregression()`



Characteristic	**N**	**OR**[1]	**95% CI**[1]	**p-value**
Chemotherapy Treatment	193			
Drug A		—	—	
Drug B		1.21	0.66, 2.24	0.5
Age	183	1.02	1.00, 1.04	0.10
Grade	193			
I		—	—	
II		0.95	0.45, 2.00	0.9
III		1.10	0.52, 2.29	0.8

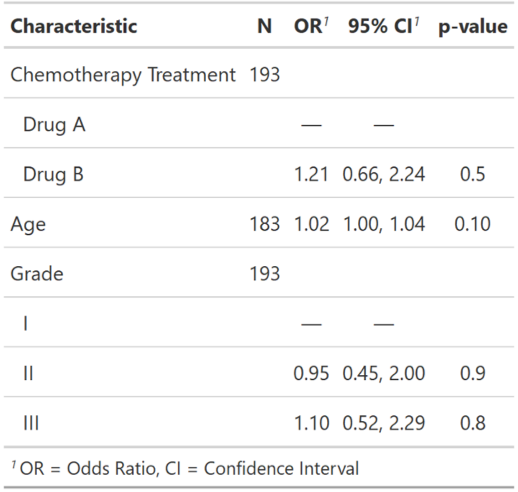
[1] OR = Odds Ratio, CI = Confidence Interval

inline_text()

{gtsummary} reporting with inline_text()

- Tables are important, but we often need to report results in-line in a report.
- Any statistic reported in a {gtsummary} table can be extracted and reported in-line in an R Markdown document with the `inline_text()` function.
- The pattern of what is reported can be modified with the `pattern =` argument.
- Default is `pattern = "{estimate} ({conf.level*100}% CI {conf.low}, {conf.high}; {p.value})"`.

{gtsummary} reporting with inline_text()



Characteristic	N	OR[1]	95% CI[1]	p-value
Chemotherapy Treatment	193			
Drug A		—	—	
Drug B		1.21	0.66, 2.24	0.5
Age	183	1.02	1.00, 1.04	0.10
Grade	193			
I		—	—	
II		0.95	0.45, 2.00	0.9
III		1.10	0.52, 2.29	0.8

[1] OR = Odds Ratio, CI = Confidence Interval

In Code: The odds ratio for age is `'r inline_text(tbl_uvreg, variable = age)'`

In Report: The odds ratio for age is 1.02 (95% CI 1.00, 1.04; p=0.10)

tbl_merge()/tbl_stack()

tbl_merge() for side-by-side tables

A **univariable** table:

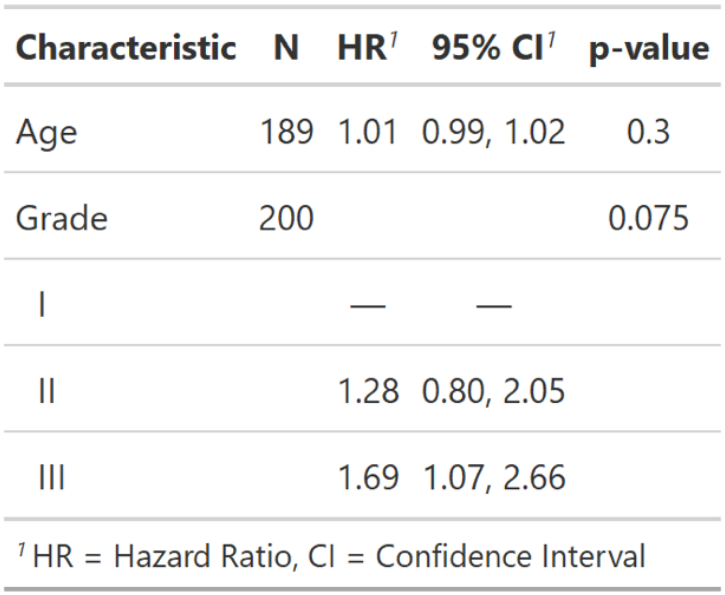
```
library(survival)

tbl_uvsurv <-
  trial %>%
  select(age, grade, death, ttdeath
  tbl_uvregression(
    method = coxph,
    y = Surv(ttdeath, death),
    exponentiate = TRUE
  ) %>%
  add_global_p()
```

A **multivariable** table:

```
library(survival)

tbl_mvsurv <-
  coxph(
    Surv(ttdeath, death) ~ age + gr
    data = trial
  ) %>%
  tbl_regression(
    exponentiate = TRUE
  ) %>%
  add_global_p()
```

tbl_merge() for side-by-side tables

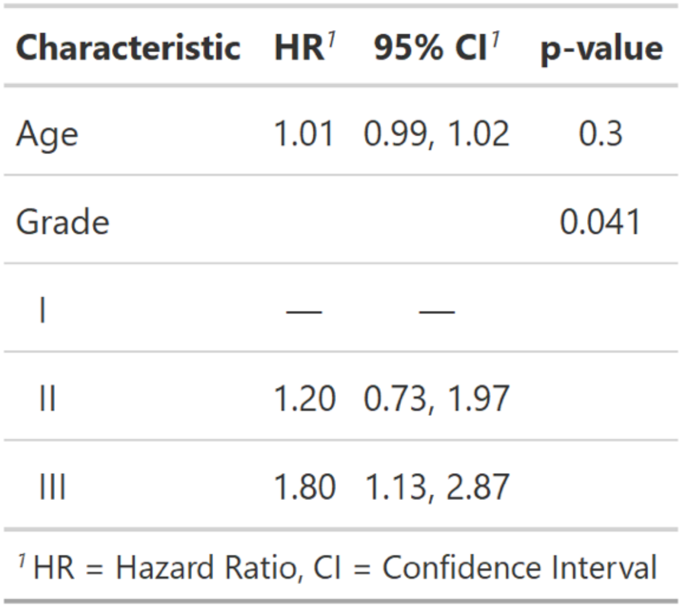
A **univariable** table:



Characteristic	N	HR[1]	95% CI[1]	p-value
Age	189	1.01	0.99, 1.02	0.3
Grade	200			0.075
I		—	—	
II		1.28	0.80, 2.05	
III		1.69	1.07, 2.66	

[1] HR = Hazard Ratio, CI = Confidence Interval

A **multivariable** table:



Characteristic	HR[1]	95% CI[1]	p-value
Age	1.01	0.99, 1.02	0.3
Grade			0.041
I	—	—	
II	1.20	0.73, 1.97	
III	1.80	1.13, 2.87	

[1] HR = Hazard Ratio, CI = Confidence Interval

tbl_merge() for side-by-side tables

```
tbl_surv_merge <-
  tbl_merge(
    list(tbl_uvsurv, tbl_mvsurv),
    tab_spanner = c("**Univariable**", "**Multivariable**")
  )
```

	Univariable				**Multivariable**		
Characteristic	**N**	**HR**[1]	**95% CI**[1]	**p-value**	**HR**[1]	**95% CI**[1]	**p-value**
Age	189	1.01	0.99, 1.02	0.3	1.01	0.99, 1.02	0.3
Grade	200			0.075			0.041
I		—	—		—	—	
II		1.28	0.80, 2.05		1.20	0.73, 1.97	
III		1.69	1.07, 2.66		1.80	1.13, 2.87	

[1] HR = Hazard Ratio, CI = Confidence Interval

tbl_stack() to combine vertically

An **unadjusted** model:

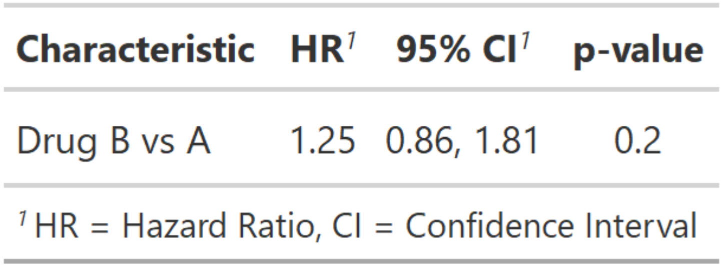
```
t3 <-
  coxph(Surv(ttdeath, death) ~ trt,
        data = trial) %>%
  tbl_regression(
    show_single_row = trt,
    label = trt ~ "Drug B vs A",
    exponentiate = TRUE
  )
```

An **adjusted** model:

```
t4 <-
  coxph(Surv(ttdeath, death) ~
          trt + grade + stage + mar
        data = trial) %>%
  tbl_regression(
    show_single_row = trt,
    label = trt ~ "Drug B vs A",
    exponentiate = TRUE,
    include = "trt"
  )
```

tbl_stack() to combine vertically

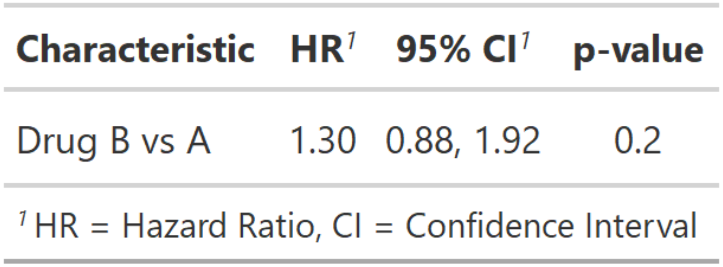
An **unadjusted** model:



Characteristic	HR[1]	95% CI[1]	p-value
Drug B vs A	1.25	0.86, 1.81	0.2

[1] HR = Hazard Ratio, CI = Confidence Interval

An **adjusted** model:

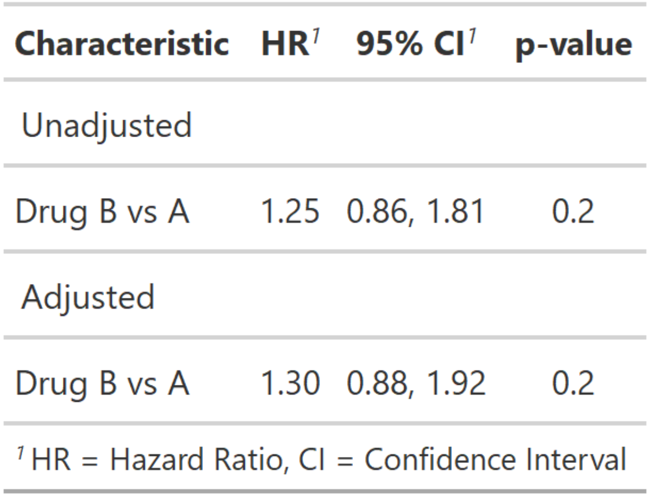


Characteristic	HR[1]	95% CI[1]	p-value
Drug B vs A	1.30	0.88, 1.92	0.2

[1] HR = Hazard Ratio, CI = Confidence Interval

tbl_stack() to combine vertically

```
tbl_surv_stack <-
  tbl_stack(
    list(t3, t4),
    group_header = c("Unadjusted", "Adjusted")
  )
```



Characteristic	HR[1]	95% CI[1]	p-value
Unadjusted			
Drug B vs A	1.25	0.86, 1.81	0.2
Adjusted			
Drug B vs A	1.30	0.88, 1.92	0.2

[1] HR = Hazard Ratio, CI = Confidence Interval

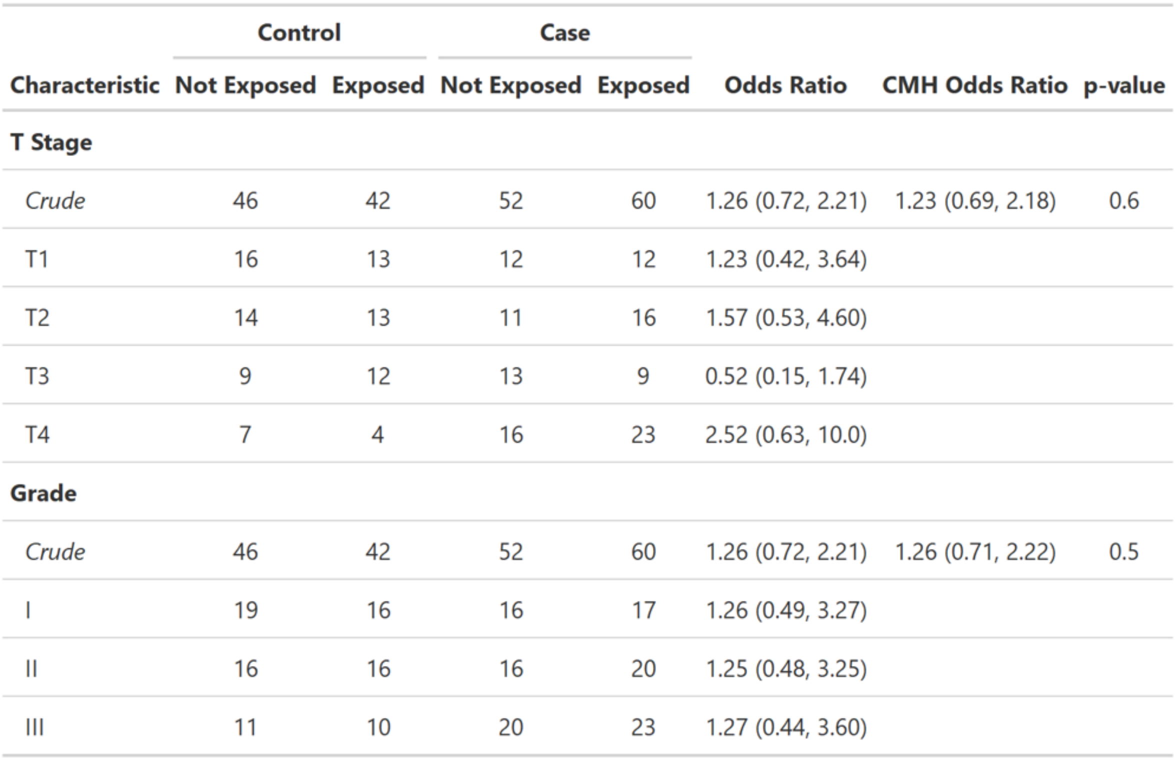
tbl_strata() for stratified tables

```
tbl_strata <-
  sm_trial %>%
  mutate(grade = paste("Grade", grade)) %>%
  tbl_strata(
    strata = grade,
    ~tbl_summary(.x, by = trt, missing = "no") %>%
      modify_header(all_stat_cols() ~ "**{level}**")
  )
```

	Grade I		**Grade II**		**Grade III**	
Characteristic	**Drug A**[1]	**Drug B**[1]	**Drug A**[1]	**Drug B**[1]	**Drug A**[1]	**Drug B**[1]
Age	46 (36, 60)	48 (42, 55)	44 (31, 54)	50 (43, 57)	52 (42, 60)	45 (36, 52)
Tumor Response	8 (23%)	13 (41%)	7 (23%)	12 (36%)	13 (43%)	8 (24%)

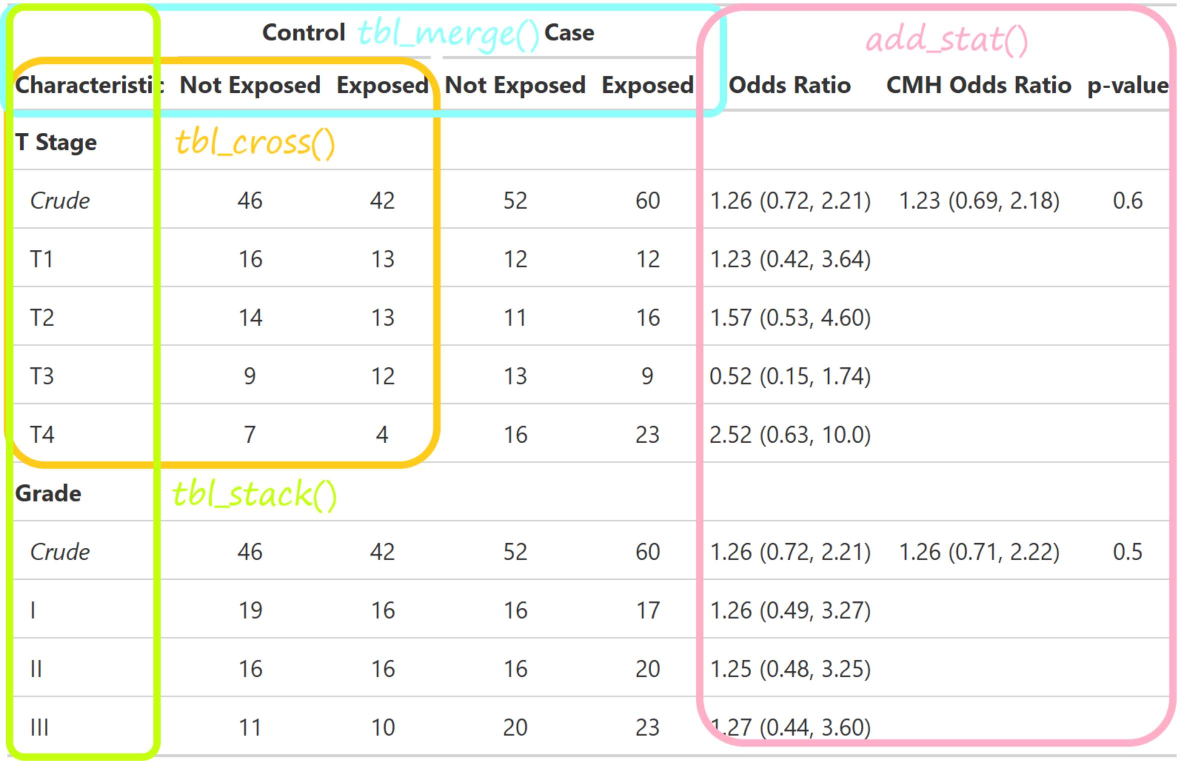
[1] Median (IQR); n (%)

Define custom function `tbl_cmh()`



	Control		Case				
Characteristic	**Not Exposed**	**Exposed**	**Not Exposed**	**Exposed**	**Odds Ratio**	**CMH Odds Ratio**	**p-value**
T Stage							
Crude	46	42	52	60	1.26 (0.72, 2.21)	1.23 (0.69, 2.18)	0.6
T1	16	13	12	12	1.23 (0.42, 3.64)		
T2	14	13	11	16	1.57 (0.53, 4.60)		
T3	9	12	13	9	0.52 (0.15, 1.74)		
T4	7	4	16	23	2.52 (0.63, 10.0)		
Grade							
Crude	46	42	52	60	1.26 (0.72, 2.21)	1.26 (0.71, 2.22)	0.5
I	19	16	16	17	1.26 (0.49, 3.27)		
II	16	16	16	20	1.25 (0.48, 3.25)		
III	11	10	20	23	1.27 (0.44, 3.60)		

Define custom functiont `tbl_cmh()`



	Control		Case				
Characteristi	**Not Exposed**	**Exposed**	**Not Exposed**	**Exposed**	**Odds Ratio**	**CMH Odds Ratio**	**p-value**
T Stage							
Crude	46	42	52	60	1.26 (0.72, 2.21)	1.23 (0.69, 2.18)	0.6
T1	16	13	12	12	1.23 (0.42, 3.64)		
T2	14	13	11	16	1.57 (0.53, 4.60)		
T3	9	12	13	9	0.52 (0.15, 1.74)		
T4	7	4	16	23	2.52 (0.63, 10.0)		
Grade							
Crude	46	42	52	60	1.26 (0.72, 2.21)	1.26 (0.71, 2.22)	0.5
I	19	16	16	17	1.26 (0.49, 3.27)		
II	16	16	16	20	1.25 (0.48, 3.25)		
III	11	10	20	23	1.27 (0.44, 3.60)		

tbl_merge()
add_stat()
tbl_cross()
tbl_stack()

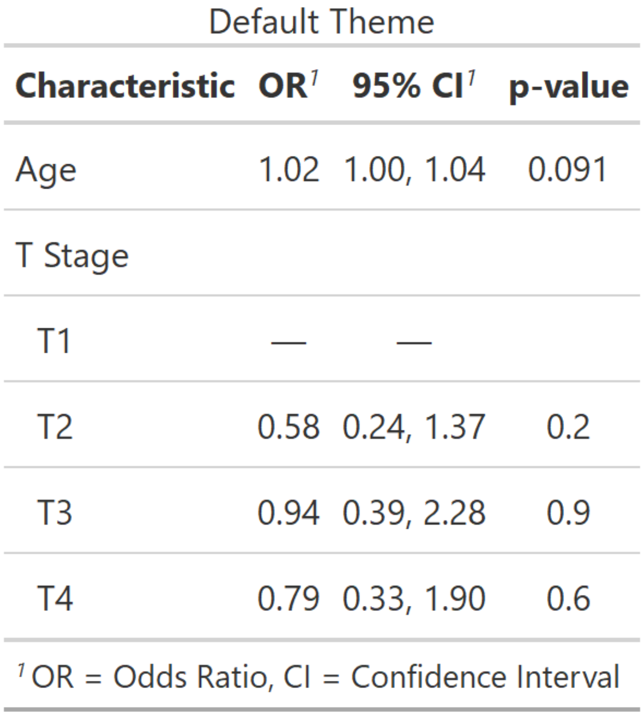
{gtsummary} themes

{gtsummary} theme basics

- A **theme** is a set of customization preferences that can be easily set and reused.
- Themes control **default settings for existing functions**
- Themes control more **fine-grained customization** not available via arguments or helper functions
- Easily use one of the **available themes**, or **create your own**

{gtsummary} default theme

```
reset_gtsummary_theme()

no_theme <-
  tbl_regression(m1, exponentiate =
  modify_caption("Default Theme")
```



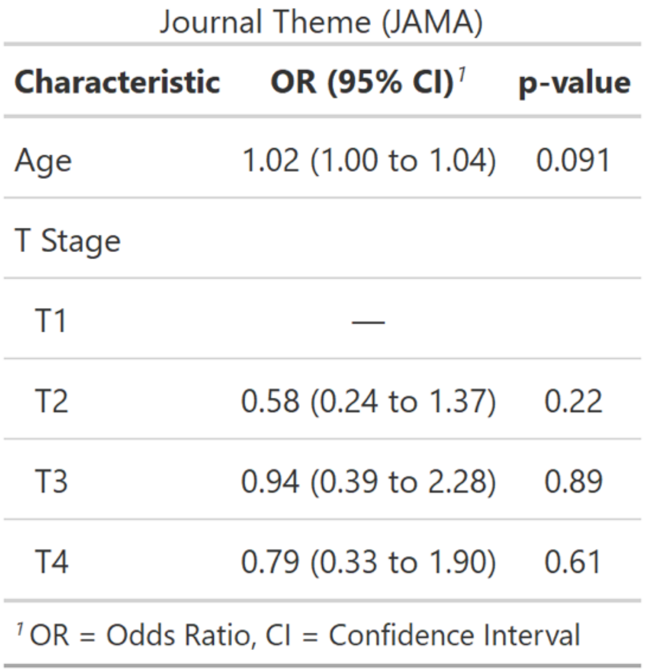
Default Theme

Characteristic	OR[1]	95% CI[1]	p-value
Age	1.02	1.00, 1.04	0.091
T Stage			
T1	—	—	
T2	0.58	0.24, 1.37	0.2
T3	0.94	0.39, 2.28	0.9
T4	0.79	0.33, 1.90	0.6

[1] OR = Odds Ratio, CI = Confidence Interval

{gtsummary} theme_gtsummary_journal()

```
reset_gtsummary_theme()

theme_gtsummary_journal(journal = "

jama_theme <-
  tbl_regression(m1, exponentiate =
  modify_caption("Journal Theme (JA
```



Journal Theme (JAMA)

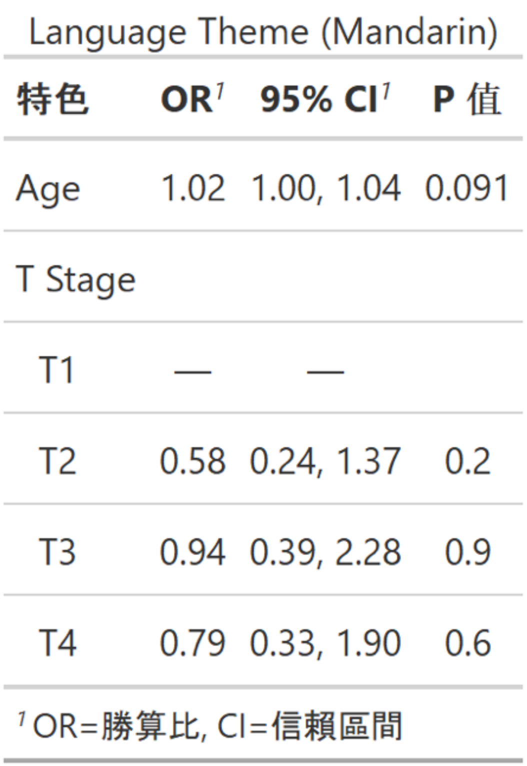
Characteristic	OR (95% CI)[1]	p-value
Age	1.02 (1.00 to 1.04)	0.091
T Stage		
T1	—	
T2	0.58 (0.24 to 1.37)	0.22
T3	0.94 (0.39 to 2.28)	0.89
T4	0.79 (0.33 to 1.90)	0.61

[1] OR = Odds Ratio, CI = Confidence Interval

Contributions welcome!

{gtsummary} theme_gtsummary_language()

```
reset_gtsummary_theme()

theme_gtsummary_language(language =

lang_theme <-
  tbl_regression(m1, exponentiate =
  modify_caption("Language Theme (C
```



Language Theme (Mandarin)

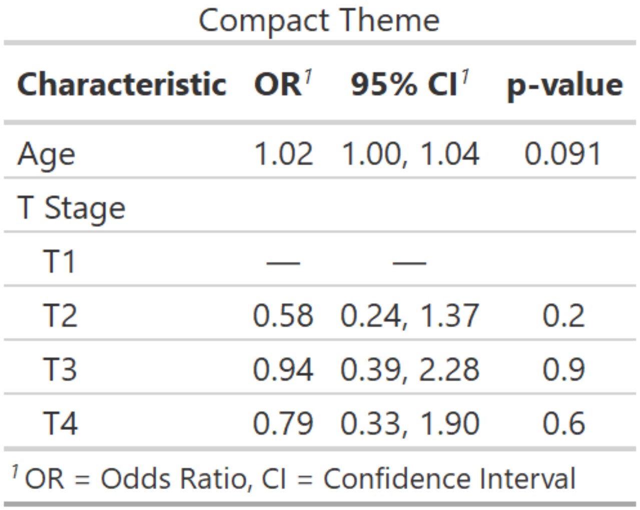
特色	OR[1]	95% CI[1]	P 值
Age	1.02	1.00, 1.04	0.091
T Stage			
T1	—	—	
T2	0.58	0.24, 1.37	0.2
T3	0.94	0.39, 2.28	0.9
T4	0.79	0.33, 1.90	0.6

[1] OR=勝算比, CI=信賴區間

Language options: "de" (German), "en" (English), "es" (Spanish), "fr" (French), "gu" (Gujarati), "hi" (Hindi), "is" (Icelandic),"ja" (Japanese), "kr" (Korean), "mr" (Marathi),

{gtsummary} theme_gtsummary_compact()

```
reset_gtsummary_theme()

theme_gtsummary_compact()

compact_theme <-
  tbl_regression(m1, exponentiate =
  modify_caption("Compact Theme")
```



Compact Theme

Characteristic	OR[1]	95% CI[1]	p-value
Age	1.02	1.00, 1.04	0.091
T Stage			
T1	—	—	
T2	0.58	0.24, 1.37	0.2
T3	0.94	0.39, 2.28	0.9
T4	0.79	0.33, 1.90	0.6

[1] OR = Odds Ratio, CI = Confidence Interval

Reduces padding and font size

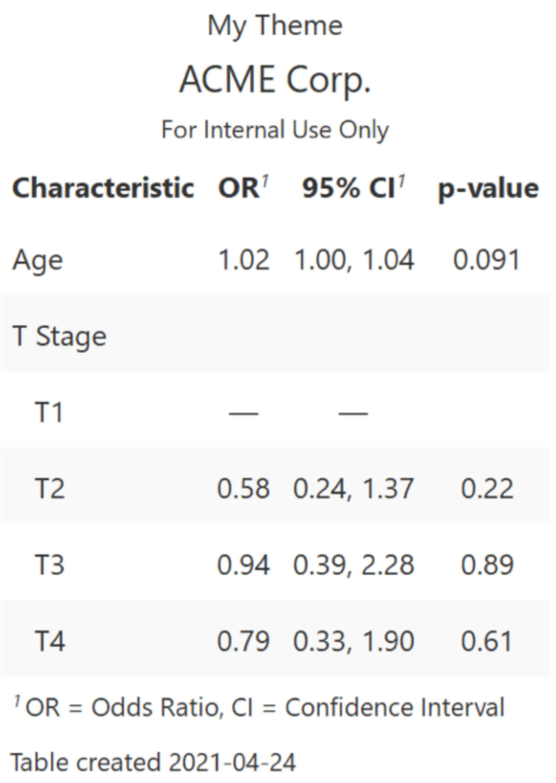
{gtsummary} set_gtsummary_theme()

```r
my_theme <-
  list(
    # round large pvalues to 2 places
    "pkgwide-fn:pvalue_fun" = function(x) gtsummary::style_pvalue(x, digits
    # Some gt customization
    "as_gt-lst:addl_cmds" = list(
      # make the font size small
      tab_spanner = rlang::expr(gt::tab_options(table.font.size = 'small'))
      # add a custom title and subtitle to every table
      user_added1 = rlang::expr(gt::tab_header(
        title = "ACME Corp.", subtitle = "For Internal Use Only")),
      # add a custom data source note
      user_added2 = rlang::expr(gt::tab_source_note(
        source_note = paste("Table created", Sys.Date()))),
      # stripe the table rows
      user_added3 = rlang::expr(gt::opt_row_striping()),
      user_added4 = rlang::expr(gt::opt_table_lines("none"))
    )
  )
```

{gtsummary} set_gtsummary_theme()

```
reset_gtsummary_theme()

set_gtsummary_theme(my_theme)

my_theme_tbl <-
  tbl_regression(m1, exponentiate =
  modify_caption("My Theme")
```

- Made the font size small
- Added custom title, subtitle, source note
- Striped the rows
- Removed all row lines



My Theme

ACME Corp.

For Internal Use Only

Characteristic	OR[1]	95% CI[1]	p-value
Age	1.02	1.00, 1.04	0.091
T Stage			
T1	—	—	
T2	0.58	0.24, 1.37	0.22
T3	0.94	0.39, 2.28	0.89
T4	0.79	0.33, 1.90	0.61

[1] OR = Odds Ratio, CI = Confidence Interval

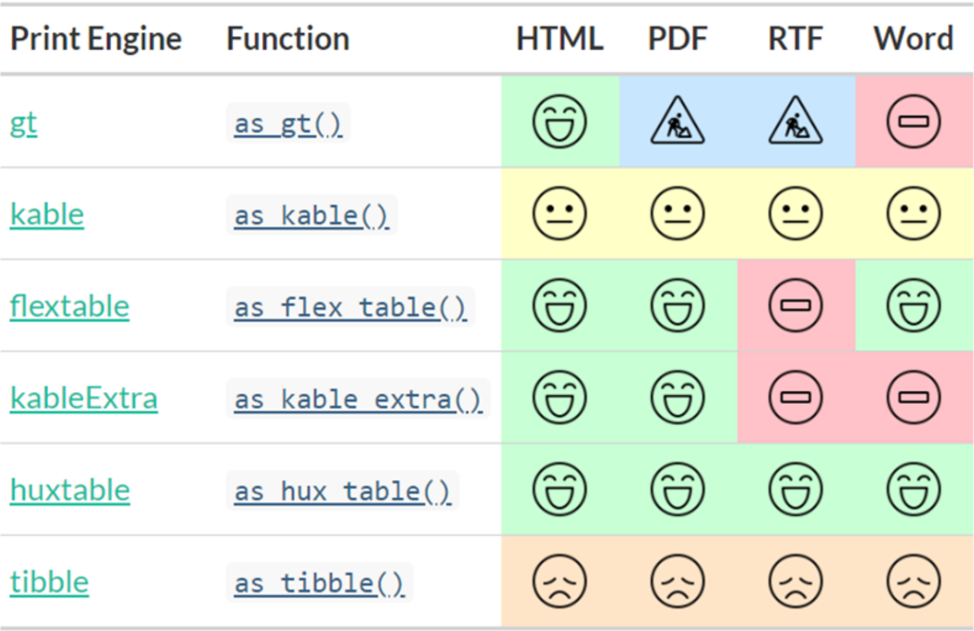
Table created 2021-04-24

And many more options!

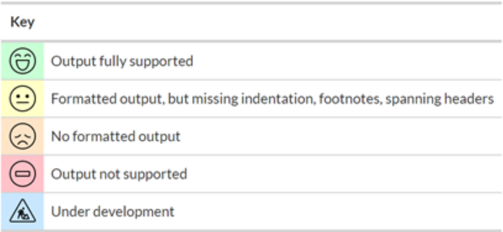
See the {gtsummary} + themes vignette:
http://www.danieldsjoberg.com/gtsummary/articles/themes.html

{gtsummary} print engines

{gtsummary} print engines



Print Engine	Function	HTML	PDF	RTF	Word
gt	`as_gt()`	😁	⚠	⚠	⊖
kable	`as_kable()`	😐	😐	😐	😐
flextable	`as_flex_table()`	😁	😁	⊖	😁
kableExtra	`as_kable_extra()`	😁	😁	⊖	⊖
huxtable	`as_hux_table()`	😁	😁	😁	😁
tibble	`as_tibble()`	☹	☹	☹	☹



Key	
😁	Output fully supported
😐	Formatted output, but missing indentation, footnotes, spanning headers
☹	No formatted output
⊖	Output not supported
⚠	Under development

{gtsummary} print engines

Use any print engine to customize table

```r
library(gt, warn.conflicts = FALSE)

print_engine <-
  trial %>%
  select(age, grade) %>%
  tbl_summary() %>%
  as_gt() %>%
  cols_width(label ~ px(300)) %>%
  cols_align(columns = vars(stat_0)
             align = "left")
```

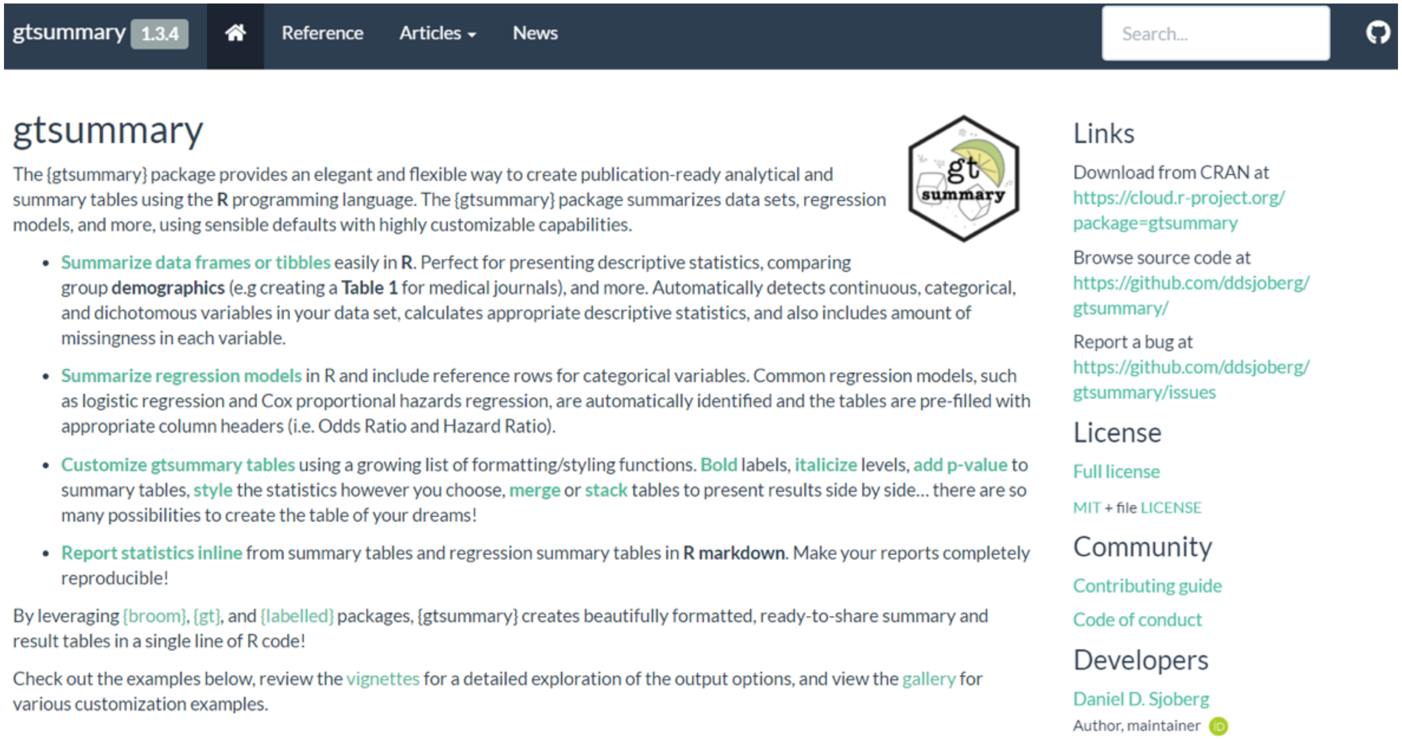
Characteristic	**N = 200**[1]
Age	47 (38, 57)
Unknown	11
Grade	
I	68 (34%)
II	68 (34%)
III	64 (32%)

[1] Median (IQR); n (%)

In Closing

{gtsummary} website

http://www.danieldsjoberg.com/gtsummary/



gtsummary

The {gtsummary} package provides an elegant and flexible way to create publication-ready analytical and summary tables using the **R** programming language. The {gtsummary} package summarizes data sets, regression models, and more, using sensible defaults with highly customizable capabilities.

- Summarize data frames or tibbles easily in **R**. Perfect for presenting descriptive statistics, comparing group **demographics** (e.g creating a **Table 1** for medical journals), and more. Automatically detects continuous, categorical, and dichotomous variables in your data set, calculates appropriate descriptive statistics, and also includes amount of missingness in each variable.
- Summarize regression models in R and include reference rows for categorical variables. Common regression models, such as logistic regression and Cox proportional hazards regression, are automatically identified and the tables are pre-filled with appropriate column headers (i.e. Odds Ratio and Hazard Ratio).
- Customize gtsummary tables using a growing list of formatting/styling functions. Bold labels, italicize levels, add p-value to summary tables, style the statistics however you choose, merge or stack tables to present results side by side... there are so many possibilities to create the table of your dreams!
- Report statistics inline from summary tables and regression summary tables in **R markdown**. Make your reports completely reproducible!

By leveraging {broom}, {gt}, and {labelled} packages, {gtsummary} creates beautifully formatted, ready-to-share summary and result tables in a single line of R code!

Check out the examples below, review the vignettes for a detailed exploration of the output options, and view the gallery for various customization examples.

Links

Download from CRAN at https://cloud.r-project.org/package=gtsummary

Browse source code at https://github.com/ddsjoberg/gtsummary/

Report a bug at https://github.com/ddsjoberg/gtsummary/issues

License

Full license

MIT + file LICENSE

Community

Contributing guide

Code of conduct

Developers

Daniel D. Sjoberg

Author, maintainer

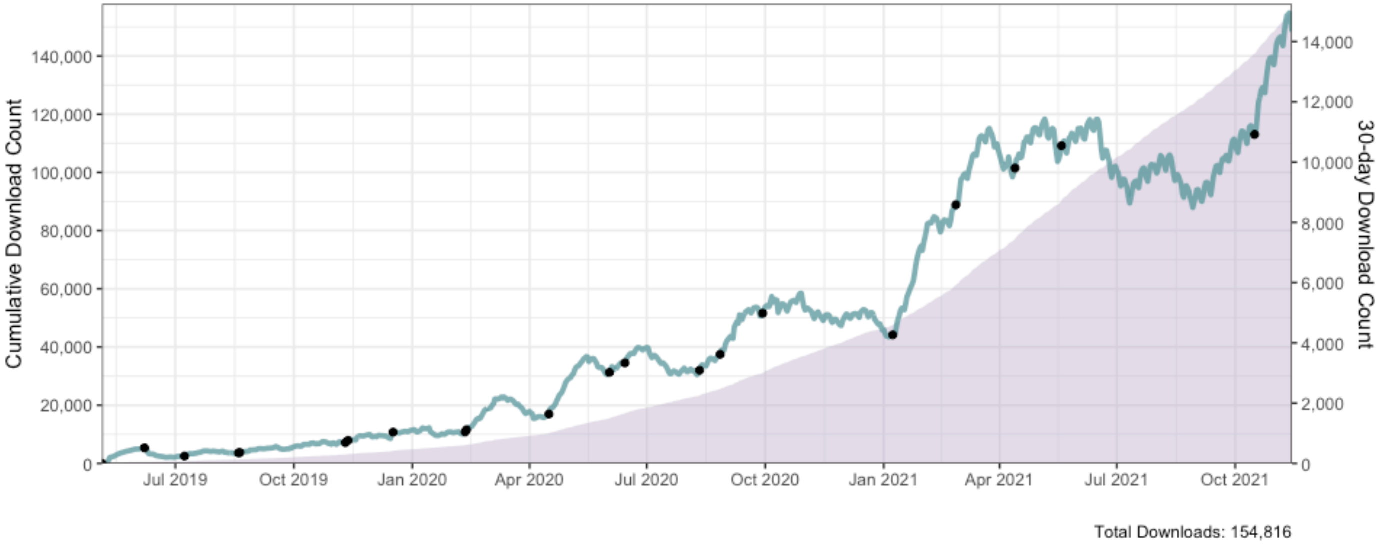
{gtsummary} installation

Install production version from CRAN:

```
install.packages("gtsummary")
```

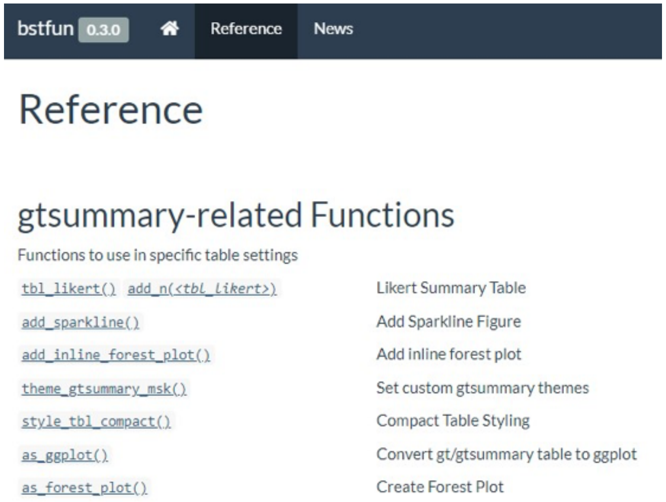
Install development version from GitHub:

```
remotes::install_github("ddsjoberg/
```



{gtsummary} sandbox in {bstfun}

http://www.danieldsjoberg.com/bstfun/

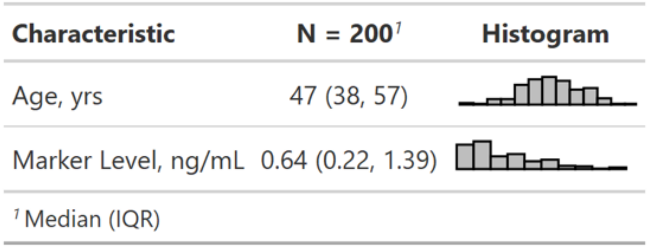


Reference

gtsummary-related Functions

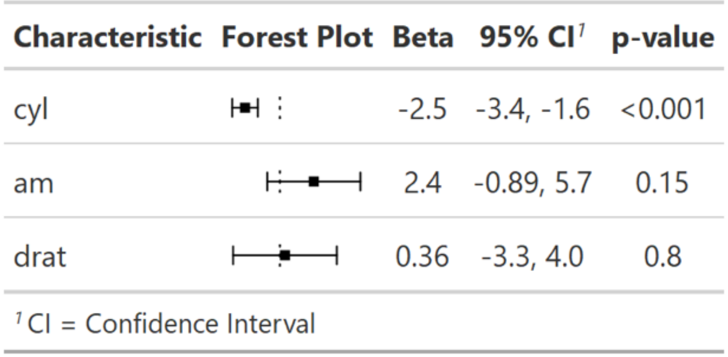
Functions to use in specific table settings

`tbl_likert()` `add_n(<tbl_likert>)`	Likert Summary Table
`add_sparkline()`	Add Sparkline Figure
`add_inline_forest_plot()`	Add inline forest plot
`theme_gtsummary_msk()`	Set custom gtsummary themes
`style_tbl_compact()`	Compact Table Styling
`as_ggplot()`	Convert gt/gtsummary table to ggplot
`as_forest_plot()`	Create Forest Plot



Characteristic	N = 200[1]	Histogram
Age, yrs	47 (38, 57)	
Marker Level, ng/mL	0.64 (0.22, 1.39)	

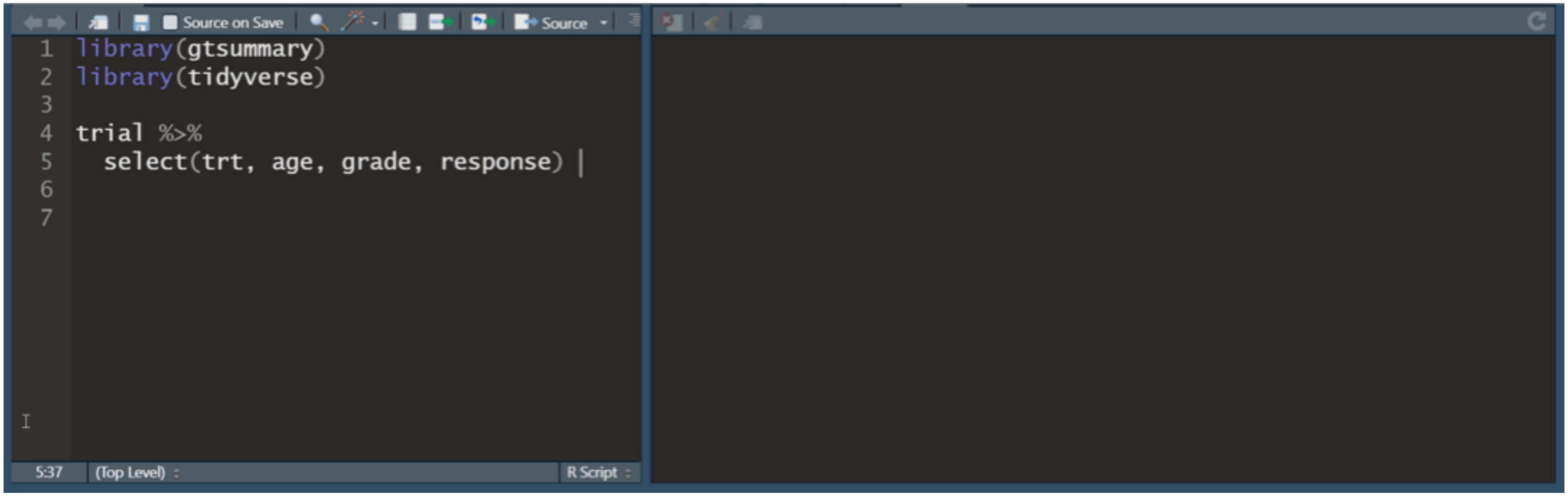
[1] Median (IQR)



Characteristic	Forest Plot	Beta	95% CI[1]	p-value
cyl		-2.5	-3.4, -1.6	<0.001
am		2.4	-0.89, 5.7	0.15
drat		0.36	-3.3, 4.0	0.8

[1] CI = Confidence Interval

Thank you



```
library(gtsummary)
library(tidyverse)

trial %>%
  select(trt, age, grade, response)
```

gtsummary questions?

Ask on stackoverflow.com and *use the gtsummary tag*. Hundreds of Qs already answered!

@statistishdan, @ddsjoberg

gtsummary object

```r
tbl <- tbl_summary(trial, by = trt, include = c(age, trt)) %>% add_p()
tbl$table_body
```

```
## # A tibble: 2 x 12
##   variable test_name var_type var_label row_type label stat_1 stat_2 test_resul
##   <chr>    <chr>     <chr>    <chr>     <chr>    <chr> <chr>  <chr>  <list>
## 1 age      wilcox.t~ continu~ Age       label    Age   46 (3~ 48 (3~ <named lis
## 2 age      wilcox.t~ continu~ Age       missing  Unkn~ 7      4      <NULL>
## # ... with 3 more variables: statistic <dbl>, p.value <dbl>, alternative <chr>
```

gtsummary object

```
names(tbl$table_styling)
```

```
## [1] "header"          "footnote"        "footnote_abbrev" "text_format"
## [5] "fmt_missing"     "fmt_fun"         "cols_merge"
```

```
tbl$table_styling
```

```
## $header
## # A tibble: 12 x 7
##    column  hide  align  interpret_label label  interpret_spanni~ spanning_heade
##    <chr>   <lgl> <chr>  <chr>           <chr>  <chr>             <chr>
##  1 variab~ TRUE  center gt::md          varia~ gt::md            <NA>
##  2 test_n~ TRUE  center gt::md          test_~ gt::md            <NA>
##  3 var_ty~ TRUE  center gt::md          var_t~ gt::md            <NA>
##  4 var_la~ TRUE  center gt::md          var_l~ gt::md            <NA>
##  5 row_ty~ TRUE  center gt::md          row_t~ gt::md            <NA>
##  6 label   FALSE left   gt::md          **Cha~ gt::md            <NA>
##  7 stat_1  FALSE center gt::md          **Dru~ gt::md            <NA>
##  8 stat_2  FALSE center gt::md          **Dru~ gt::md            <NA>
##  9 test_r~ TRUE  center gt::md          test_~ gt::md            <NA>
## 10 statis~ TRUE  center gt::md          stati~ gt::md            <NA>
## 11 p.value FALSE center gt::md          **p-v~ gt::md            <NA>
```

gtsummary object

```
modify_table_styling(
  x,
  columns,
  rows = NULL,
  label = NULL,
  spanning_header = NULL,
  hide = NULL,
  footnote = NULL,
  footnote_abbrev = NULL,
  align = NULL,
  missing_symbol = NULL,
  fmt_fun = NULL,
  text_format = NULL,
  undo_text_format = FALSE,
  text_interpret = c("md", "html"),
  cols_merge_pattern = NULL
)
```

```
modify_header()
modify_footnote()
modify_spanning_header()
modify_caption()
modify_column_hide()
modify_column_unhide()
modify_fmt_fun()

modify_table_body()
```