

# Dynamic documents in R

## Making tables in R Markdown

2021-06-04

# Working with descriptive tables

```
descriptives <- diabetes %>%  
  filter(!is.na(glyhb)) %>%  
  mutate(  
    diabetic = case_when(  
      glyhb >= 6.5 ~ "Diabetic",  
      glyhb < 6.5 ~ "Healthy",  
      NA ~ NA_character_  
    ),  
    bmi = (weight / height^2) * 703  
  ) %>%  
  group_by(diabetic) %>%  
  summarise(n = n(), across(c(glyhb, bmi, age), mean, na.rm = TRUE))
```

# Descriptive statistics table

## Standard tibble display

```
descriptives
```

```
## # A tibble: 2 x 5
##   diabetic      n glyhb    bmi    age
##   <chr>    <int> <dbl> <dbl> <dbl>
## 1 Diabetic     65  9.89  30.8  58.4
## 2 Healthy    325  4.73  28.4  44.4
```

# Descriptive statistics table

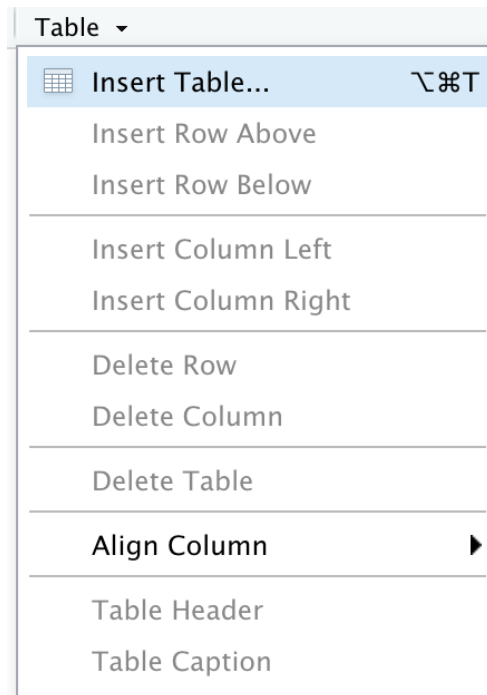
## Standard tibble display

```
descriptives
```

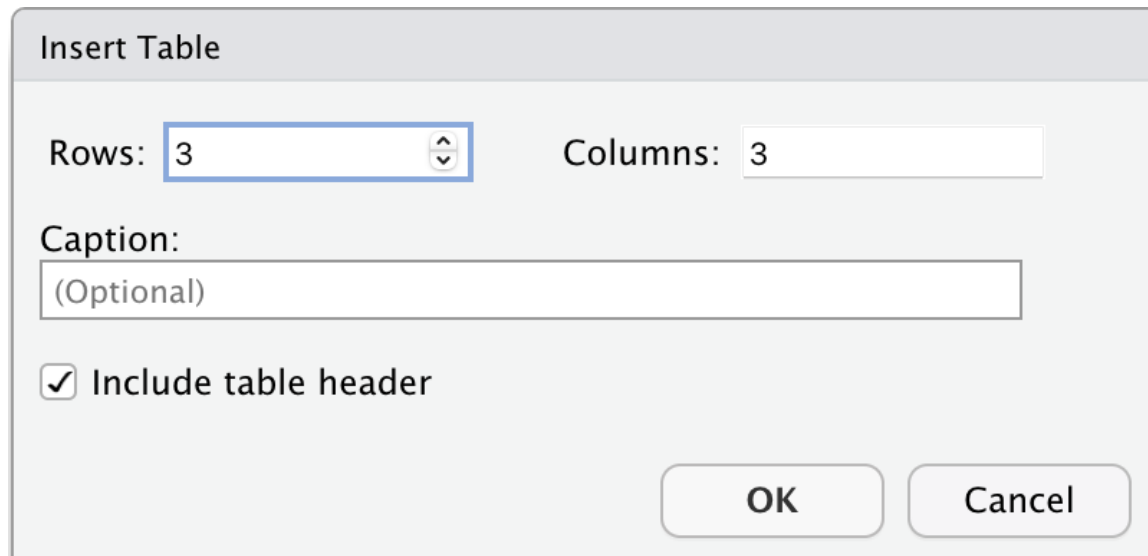
```
## # A tibble: 2 x 5
##   diabetic      n glyhb    bmi    age
##   <chr>    <int> <dbl> <dbl> <dbl>
## 1 Diabetic     65  9.89  30.8  58.4
## 2 Healthy    325  4.73  28.4  44.4
```

**Useful in console, but less so for reports or presentation**

# Visual R Markdown tables



# Visual R Markdown tables



The image shows a 'Insert Table' dialog box with a light gray background and a thin gray border. At the top, the title 'Insert Table' is in a darker gray font. Below the title, there are two input fields: 'Rows:' with a value of '3' and a small up/down arrow icon, and 'Columns:' with a value of '3'. Below these, there is a 'Caption:' label followed by a text input field containing '(Optional)'. At the bottom left, there is a checked checkbox labeled 'Include table header'. At the bottom right, there are two buttons: 'OK' and 'Cancel'.

Insert Table

Rows: 3 Columns: 3

Caption:  
(Optional)

☒ Include table header

OK Cancel

## Your turn 1

**Using Visual R Markdown, create a markdown table that represents these data:**

**The Physicians' Health Study enrolled over 22,000 male physicians to study the effect of low-dose aspirin on myocardial infarctions (heart attacks). Of those who took aspirin, 129 had heart attacks, while 10,898 did not. Of those who took the placebo, 239 had heart attacks and 10,795 did not.**

# knitr::kable()



**kable() creates formatted tables from rectangular objects (data.frames, matrices, and tibbles)**

```
kable(descriptives)
```

diabetic	n	glyhb	bmi	age
Diabetic	65	9.886615	30.80235	58.43077
Healthy	325	4.730400	28.38114	44.44308

**Formats include** pipe ('|'), **HTML** (html), **LaTeX** (latex), and **reStructuredText** (rst)



# kable(): underlying structure

## Simple and pipe work in most output document formats

```
knitr::kable(x = descriptives, format = "simple")
```

diabetic	n	glyhb	bmi	age
Diabetic	65	9.886615	30.80235	58.43077
Healthy	325	4.730400	28.38114	44.44308

```
knitr::kable(x = descriptives, format = "pipe")
```

diabetic	n	glyhb	bmi	age
:-----	---:	-----:	-----:	-----:
Diabetic	65	9.886615	30.80235	58.43077
Healthy	325	4.730400	28.38114	44.44308

## kable() options: column names

**Supply a vector of new names with col.names argument**

```
kable(descriptives,  
      col.names = c("Diabetes Status", "N", "A1c", "BMI", "Age")  
)
```

Diabetes Status	N	A1c	BMI	Age
Diabetic	65	9.886615	30.80235	58.43077
Healthy	325	4.730400	28.38114	44.44308

# kable() options: number format

## Use digits for decimal place

```
kable(descriptives,  
  digits = 1  
)
```

diabetic	n	glyhb	bmi	age
Diabetic	65	9.9	30.8	58.4
Healthy	325	4.7	28.4	44.4

# kable() options: add caption

## Add caption with caption argument

```
kable(descriptives,  
      caption = "Descriptive statistics"  
)
```

Table: Descriptive statistics

diabetic	n	glyhb	bmi	age
Diabetic	65	9.886615	30.80235	58.43077
Healthy	325	4.730400	28.38114	44.44308

# Your turn 2

**Using the tidy() function from broom, turn response\_model and marker\_model into dataframes, binding them together with bind\_rows()**

**Turn the resulting dataframe into a table using kable()**

**Set the eval chunk option to TRUE**

**Knit**

```
response_model <- glm(response ~ age + stage, data = trial,  
                      family = binomial)  
marker_model <- lm(marker ~ trt + stage + age, data = trial)
```

## Your turn 2

```
models <- bind_rows(tidy(response_model), tidy(marker_model))  
kable(models)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-1.4862242	0.6202284	-2.3962530	0.0165637
age	0.0193911	0.0114681	1.6908683	0.0908620
stageT2	-0.5414264	0.4400027	-1.2305071	0.2185073
stageT3	-0.0595348	0.4504203	-0.1321761	0.8948450
stageT4	-0.2310863	0.4482284	-0.5155549	0.6061653
(Intercept)	0.8336800	0.2497219	3.3384338	0.0010317
trtDrug B	-0.2009397	0.1290364	-1.5572330	0.1212427
stageT2	0.3841096	0.1734658	2.2143255	0.0281131
stageT3	0.2838452	0.1895362	1.4975777	0.1360653
stageT4	0.1987877	0.1818389	1.0932073	0.2758227
age	-0.0004592	0.0044877	-0.1023233	0.9186185

# gtsummary



Create publication-ready analytic and summary tables

Good support for wide variety of output formats

Built on the larger **gt** package

# tbl\_cross()

Creates a **cross-tabulation** of two categorical variables

```
library(gtsummary)
tbl_cross(
  data,
  row = x,
  col = y
)
```



# tbl\_cross()

```
table_data <- diabetes %>%  
  mutate(  
    # create diabetic category based on A1c and calculate bmi  
    diabetic = case_when(  
      glyhb >= 6.5 ~ "Diabetic",  
      glyhb < 6.5 ~ "Healthy",  
      NA ~ NA_character_  
    ), bmi = (weight / height^2) * 703  
  ) %>%  
  select(diabetic, age, gender, bmi)
```

# tbl\_cross()

```
tbl_cross(table_data, row = diabetic, col = gender)
```

# tbl\_cross()

```
tbl_cross(table_data, row = diabetic, col = gender)
```

Characteristic	gender		Total
	female	male	
diabetic			
Diabetic	36	29	65
Healthy	192	133	325
Unknown	6	7	13
Total	234	169	403

# tbl\_cross() output

Characteristic	gender		Total
	female	male	
diabetic			
Diabetic	36	29	65
Healthy	192	133	325
Unknown	6	7	13
Total	234	169	403

## tbl\_cross(): variable names

gtsummary **uses the** labelled package **to create labels**

```
library(labelled)
var_label(table_data$diabetic) <- "Diabetes Status"
var_label(table_data$diabetic)
```

## tbl\_cross(): variable names

gtsummary **uses the** labelled package **to create labels**

```
library(labelled)
var_label(table_data$diabetic) <- "Diabetes Status"
var_label(table_data$diabetic)
```

```
## [1] "Diabetes Status"
```

## tbl\_cross(): variable names

### Set multiple variable labels with list()

```
var_label(table_data) <- list(  
  gender = "Gender",  
  diabetic = "Diabetes Status"  
)
```

## tbl\_cross(): variable names

```
tbl_cross(table_data, diabetic, gender)
```

Characteristic	Gender		Total
	female	male	
Diabetes Status			
Diabetic	36	29	65
Healthy	192	133	325
Unknown	6	7	13
Total	234	169	403



## Your turn 3

**Create a contingency table of trial (a dataset from gtsummary) using `tbl_cross()`: set trt to the rows and response to the columns.**

**Add a p-value with `add_p()`**

**Knit**

# Your turn 3

```
trial %>%  
  tbl_cross(row = trt, col = response) %>%  
  add_p()
```

Characteristic	Tumor Response			Total	p-value <sup>1</sup>
	0	1	Unknown		
Chemotherapy Treatment					0.7
Drug A	67	28	3	98	
Drug B	65	33	4	102	
Total	132	61	7	200	

<sup>1</sup>Fisher's exact test

`tbl_summary()`

**Calculates descriptive statistics**

`tbl_summary()`

Calculates descriptive statistics

**Can split calculations by groups (i.e. categorical or dichotomous variables)**

# `tbl_summary()`

Calculates descriptive statistics

Can split calculations by groups (i.e. categorical or dichotomous variables)

**Sensible defaults; easily customized**

# tbl\_summary(): basic use

```
var_label(table_data) <- list(  
  gender = "Gender",  
  diabetic = "Diabetes Status",  
  age = "Age",  
  bmi = "BMI"  
)
```

```
tbl_summary(table_data)
```

# tbl\_summary(): basic use

Characteristic	N = 403 <sup>1</sup>
Diabetes Status	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
Age	45 (34, 60)
Gender	
female	234 (58%)
male	169 (42%)
BMI	28 (24, 32)
Unknown	6
<sup>1</sup> n (%); Median (IQR)	

# tbl\_summary(): basic use

Automatically adds header:

- **Characteristic** = variable
- **N** = nrow(diabetes)

Characteristic N = 403 <sup>1</sup>	
diabetic	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
age	45 (34, 60)
gender	
female	234 (58%)
male	169 (42%)
bmi	28 (24, 32)
Unknown	6
<sup>1</sup> n (%); Median (IQR)	



# tbl\_summary(): basic use

**Missing = Unknown**

Characteristic	N = 403 <sup>1</sup>
diabetic	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
age	45 (34, 60)
gender	
female	234 (58%)
male	169 (42%)
bmi	28 (24, 32)
Unknown	6
<sup>1</sup> n (%); Median (IQR)	

# tbl\_summary(): basic use

## Source note lists summary statistics:

- **n(%)** = !is.na(var)/403
- **Median** = median(var)
- **IQR** = IQR(var)

Characteristic	N = 403 <sup>1</sup>
diabetic	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
age	45 (34, 60)
gender	
female	234 (58%)
male	169 (42%)
bmi	28 (24, 32)
Unknown	6
<sup>1</sup> n (%); Median (IQR)	

# tbl\_summary(): variable levels

## Variable levels are indented

```
table(table_data$diabetic)
```

```
###  
### Diabetic   Healthy  
###         65      325
```

```
table(table_data$gender)
```

```
###  
### female    male  
###      234    169
```

Characteristic	N = 403 <sup>1</sup>
Diabetes Status	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
Age	45 (34, 60)
Gender	
female	234 (58%)
male	169 (42%)
BMI	28 (24, 32)
Unknown	6
<sup>1</sup> n (%); Median (IQR)	

# tbl\_summary(): split by groups

## Split the summary table by a grouping variable

```
table_data %>%  
  select(diabetic, age, gender, bmi) %>%  
  tbl_summary(by = diabetic)
```

Missing values removed

**Characteristic** Diabetic, N = 65<sup>†</sup> Healthy, N = 325<sup>†</sup>

Age	59 (51, 65)	41 (32, 55)
-----	-------------	-------------

Gender		
--------	--	--

female	36 (55%)	192 (59%)
--------	----------	-----------

male	29 (45%)	133 (41%)
------	----------	-----------

BMI	30 (26, 33)	27 (23, 32)
-----	-------------	-------------

Unknown	2	4
---------	---	---

<sup>†</sup> Median (IQR); n (%)

# tbl\_summary(): digits

Change number formats with digits

```
table_data %>%  
  select(diabetic, age, gender, bmi) %>%  
  tbl_summary(  
    by = diabetic,  
    digits = all_continuous() ~ 1  
  )
```

Characteristic	Diabetic, N = 65 <sup>1</sup>	Healthy, N = 325 <sup>1</sup>
age	59.0 (51.0, 65.0)	41.0 (32.0, 55.0)
gender		
female	36 (55%)	192 (59%)
male	29 (45%)	133 (41%)
bmi	29.9 (26.4, 33.2)	27.5 (23.4, 31.8)
Unknown	2	4
<sup>1</sup> Median (IQR); n (%)		

# tbl\_summary(): missing\_text

**Rename missing value label with** missing\_text

```
table_data %>%  
  select(diabetic, age, gender, bmi) %>%  
  tbl_summary(  
    by = diabetic,  
    missing_text = "(Missing)"  
  )
```

Characteristic	Diabetic, N = 65 <sup>1</sup>	Healthy, N = 325 <sup>1</sup>
age	59 (51, 65)	41 (32, 55)
gender		
female	36 (55%)	192 (59%)
male	29 (45%)	133 (41%)
bmi	30 (26, 33)	27 (23, 32)
(Missing)	2	4

<sup>1</sup> Median (IQR); n (%)

# tbl\_summary(): tests & p-values

Add tests and p-values (with one function!)

```
table_data %>%  
  select(diabetic, age, gender, bmi) %>%  
  tbl_summary(by = diabetic) %>%  
  add_p()
```

p-value column added

Characteristic	Diabetic, N = 65 <sup>1</sup>	Healthy, N = 325 <sup>1</sup>	p-value <sup>2</sup>
Age	59 (51, 65)	41 (32, 55)	<0.001
Gender			0.6
female	36 (55%)	192 (59%)	
male	29 (45%)	133 (41%)	
BMI	30 (26, 33)	27 (23, 32)	0.003
Unknown	2	4	

<sup>1</sup> Median (IQR); n (%)

<sup>2</sup> Wilcoxon rank sum test; Pearson's Chi-squared test

tests included in footnote

## tbl\_summary(): statistic

```
tbl_summary(  
  table_data,  
  statistic = list(all_continuous() ~ "{mean} ({sd})")  
)
```

Supply **statistic** arguments in **list()**  
(separate with ~)



## tbl\_summary(): statistic

```
tbl_summary(  
  table_data,  
  statistic = list(all_continuous() ~ "{mean} ({sd})")  
)
```

**Access variables by name or type**  
(**all\_continuous()**, **all\_categorical()**,  
**all\_dichotomous()**, etc.)

## tbl\_summary(): statistic

```
tbl_summary(  
  table_data,  
  statistic = list(all_continuous() ~ "{mean} ({sd})")  
)
```

Access statistic with curly brackets (i.e. {mean}, {sd}, etc.). See ?tbl\_summary for options

# tbl\_summary(): statistic

Characteristic	N = 403 <sup>1</sup>
Diabetes Status	
Diabetic	65 (17%)
Healthy	325 (83%)
Unknown	13
Age	47 (16)
Gender	
female	234 (58%)
male	169 (42%)
BMI	29 (7)
Unknown	6
<sup>1</sup> n (%); Mean (SD)	

# Add other elements to table

Function	Description
<code>add_p()</code>	add p-values to the output comparing values across groups
<code>add_overall()</code>	add a column with overall summary statistics
<code>add_n()</code>	add a column with N (or N missing) for each variable
<code>add_difference()</code>	add column for difference between two group, confidence interval, and p-value
<code>add_stat_label()</code>	add label for the summary statistics shown in each row
<code>add_stat()</code>	generic function to add a column with user-defined values
<code>add_q()</code>	add a column of q values to control for multiple comparisons

# gtsummary functions to format table

Function	Description
modify_header()	update column headers
modify_footnote()	update column footnote
modify_spanning_header()	update spanning headers
modify_caption()	update table caption/title
bold_labels()	bold variable labels
bold_levels()	bold variable levels
italicize_labels()	italicize variable labels
italicize_levels()	italicize variable levels
bold_p()	bold significant p-values

# Your Turn 4

**Create a summary table of trial by the trt variable.**

**Modify the label for grade to say "Tumor Grade"**

**Add an overall column and a p-value**

**Modify the table to use Mean (SD) and n (%) via the statistic argument:** `list(all_continuous() ~ "{mean} ({sd})", all_categorical() ~ "{n} ({p}%")`

**Bold the labels**

**Modify the header with** `modify_spanning_header(c("stat_1", "stat_2") ~ "**Treatment Received**")`

# Your Turn 4

```
trial %>%
  tbl_summary(
    by = trt,
    label = grade ~ "Tumor Grade",
    statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} ({p}%) "
    )
  ) %>%
  bold_labels() %>%
  modify_spanning_header(
    c("stat_1", "stat_2") ~ "**Treatment Received**"
  ) %>%
  add_overall() %>%
  add_p()
```

# Your Turn 4

Characteristic	Overall, N = 200 <sup>1</sup>	Treatment Received		p-value <sup>2</sup>
		Drug A, N = 98 <sup>1</sup>	Drug B, N = 102 <sup>1</sup>	
<b>Age</b>	47 (14)	47 (15)	47 (14)	0.7
Unknown	11	7	4	
<b>Marker Level (ng/mL)</b>	0.92 (0.86)	1.02 (0.89)	0.82 (0.83)	0.085
Unknown	10	6	4	
<b>T Stage</b>				0.9
T1	53 (26%)	28 (29%)	25 (25%)	
T2	54 (27%)	25 (26%)	29 (28%)	
T3	43 (22%)	22 (22%)	21 (21%)	
T4	50 (25%)	23 (23%)	27 (26%)	
<b>Tumor Grade</b>				0.9
I	68 (34%)	35 (36%)	33 (32%)	
II	68 (34%)	32 (33%)	36 (35%)	
III	64 (32%)	31 (32%)	33 (32%)	
<b>Tumor Response</b>	61 (32%)	28 (29%)	33 (34%)	0.5
Unknown	7	3	4	
<b>Patient Died</b>	112 (56%)	52 (53%)	60 (59%)	0.4
<b>Months to Death/Censor</b>	19.6 (5.3)	20.2 (5.0)	19.0 (5.5)	0.14

<sup>1</sup> Mean (SD); n (%)

<sup>2</sup> Wilcoxon rank sum test; Pearson's Chi-squared test



# Other `tbl_summary()` arguments

See the other options with `?tbl_summary()`

Argument	Description
label	specify the variable labels printed in table
type	specify the variable type (e.g. continuous, categorical, etc.)
statistic	change the summary statistics presented
digits	number of digits the summary statistics will be rounded to
missing	whether to display a row with the number of missing observations
missing_text	text label for the missing number row
sort	change the sorting of categorical levels by frequency
percent	print column, row, or cell percentages
include	list of variables to include in summary table

# tbl\_regression()

```
model_data <- diabetes %>% mutate(bmi = (weight / height^2) * 703)  
glyhb_model <- lm(glyhb ~ bmi + age, data = model_data)  
tbl_regression(glyhb_model)
```

# tbl\_regression()

```
model_data <- diabetes %>% mutate(bmi = (weight / height^2) * 703)
glyhb_model <- lm(glyhb ~ bmi + age, data = model_data)
tbl_regression(glyhb_model)
```

Characteristic	Beta	95% CI <sup>1</sup>	p-value
bmi	0.04	0.01, 0.08	0.006
age	0.05	0.03, 0.06	<0.001

<sup>1</sup> CI = Confidence Interval

# tbl\_regression(): variable labels

Add labels just like `tbl_summary()`

```
var_label(model_data) <- list(  
  bmi = "BMI",  
  glyhb = "A1c",  
  age = "Age"  
)  
glyhb_model <- lm(glyhb ~ bmi + age, data = model_data)  
tbl_regression(glyhb_model)
```

# tbl\_regression(): variable labels

## Add labels just like tbl\_summary()

```
var_label(model_data) <- list(  
  bmi = "BMI",  
  glyhb = "A1c",  
  age = "Age"  
)  
glyhb_model <- lm(glyhb ~ bmi + age, data = model_data)  
tbl_regression(glyhb_model)
```

Characteristic	Beta	95% CI <sup>1</sup>	p-value
BMI	0.04	0.01, 0.08	0.006
Age	0.05	0.03, 0.06	<0.001

<sup>1</sup> CI = Confidence Interval

# tbl\_regression(): other options

Argument	Description
label	modify variable labels in table
exponentiate	exponentiate model coefficients
`include	names of variables to include in output. Default is all variables
show_single_row	By default, categorical variables are printed on multiple rows. If a variable is dichotomous and you wish to print the regression coefficient on a single row, include the variable name(s) here.
conf.level	confidence level of confidence interval
intercept	indicates whether to include the intercept
estimate_fun	function to round and format coefficient estimates
pvalue_fun	function to round and format p-values
tidy_fun	function to specify/customize tidier function

# tbl\_regression(): Model statistics

**Add model statistics from broom::glance()**

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note()
```

# tbl\_regression(): Model statistics

## Add model statistics from broom::glance()

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note()
```

Characteristic	Beta	95% CI <sup>1</sup>	p-value
BMI	0.04	0.01, 0.08	0.006
Age	0.05	0.03, 0.06	<0.001

<sup>1</sup> CI = Confidence Interval

R<sup>2</sup> = 0.132; Adjusted R<sup>2</sup> = 0.127; Sigma = 2.06; Statistic = 28.9; p-value = <0.001; df = 2; Log-likelihood = -821; AIC = 1,650; BIC = 1,666; Deviance = 1,617; Residual df = 381; No. Obs. = 384



# tbl\_regression(): other add options

Function	Description
add_global_p()	adds the global p-value for a categorical variables
add_glance_source_note()	adds statistics from glance() as source note
add_vif()	adds column of the variance inflation factors (VIF)
add_q()	add a column of q values to control for multiple comparisons

# tbl\_regression(): format options

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note() %>%  
  bold_p() %>%  
  bold_labels()
```

# tbl\_regression(): format options

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note() %>%  
  bold_p() %>%  
  bold_labels()
```

Characteristic	Beta	95% CI <sup>1</sup>	p-value
BMI	0.04	0.01, 0.08	0.006
Age	0.05	0.03, 0.06	<0.001

<sup>1</sup> CI = Confidence Interval

R<sup>2</sup> = 0.132; Adjusted R<sup>2</sup> = 0.127; Sigma = 2.06; Statistic = 28.9; p-value = <0.001; df = 2; Log-likelihood = -821; AIC = 1,650; BIC = 1,666; Deviance = 1,617; Residual df = 381; No. Obs. = 384

# tbl\_regression(): modify caption

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note() %>%  
  bold_p() %>%  
  bold_labels() %>%  
  modify_caption("**Association of BMI and A1c adjusted for age**")
```

# tbl\_regression(): modify caption

```
tbl_regression(glyhb_model) %>%  
  add_glance_source_note() %>%  
  bold_p() %>%  
  bold_labels() %>%  
  modify_caption("**Association of BMI and A1c adjusted for age**")
```

Association of BMI and A1c adjusted for age			
Characteristic	Beta	95% CI <sup>1</sup>	p-value
BMI	0.04	0.01, 0.08	0.006
Age	0.05	0.03, 0.06	<0.001

<sup>1</sup> CI = Confidence Interval

R<sup>2</sup> = 0.132; Adjusted R<sup>2</sup> = 0.127; Sigma = 2.06; Statistic = 28.9; p-value = <0.001; df = 2; Log-likelihood = -821; AIC = 1,650; BIC = 1,666; Deviance = 1,617; Residual df = 381; No. Obs. = 384

## Your turn 5

**Create a regression table for** response model;  
**exponentiate the output with** `exponentiate = TRUE`

**Add a global P-value**

**Add the model statistics via**  
`add_glance_source_note()`

**Italicize the variable levels**

**Merge the table you just made with** `marker_table`  
**using** `tbl_merge()`

## Your turn 5

```
response_table <- response_model %>%  
  tbl_regression(  
    exponentiate = TRUE  
  ) %>%  
  add_global_p() %>%  
  add_glance_source_note() %>%  
  italicize_levels()  
  
marker_table <- marker_model %>%  
  tbl_regression() %>%  
  add_global_p()  
  
both_tables <- list(response_table, marker_table)  
tbl_merge(both_tables)
```

# Your turn 5

Characteristic	Table 1			Table 2		
	OR <sup>1</sup>	95% CI <sup>1</sup>	p-value	Beta	95% CI <sup>1</sup>	p-value
Age	1.02	1.00, 1.04	0.087	0.00	-0.01, 0.01	>0.9
T Stage			0.6			0.2
T1	—	—		—	—	
T2	0.58	0.24, 1.37		0.38	0.04, 0.73	
T3	0.94	0.39, 2.28		0.28	-0.09, 0.66	
T4	0.79	0.33, 1.90		0.20	-0.16, 0.56	
Chemotherapy Treatment						0.12
Drug A				—	—	
Drug B				-0.20	-0.46, 0.05	

<sup>1</sup>OR = Odds Ratio, CI = Confidence Interval

Null deviance = 229; Null df = 182; Log-likelihood = -112; AIC = 234; BIC = 250; Deviance = 224; Residual df = 178; No. Obs. = 183



# Cross-referencing tables

A bookdown output format, e.g.  
`bookdown::html_document2`

# Cross-referencing tables

A bookdown output format, e.g.  
`bookdown::html_document2`

A table caption (`caption = "Table title",  
modify_caption()`)

# Cross-referencing tables

**A bookdown output format, e.g.**  
`bookdown::html_document2`

**A table caption** (`caption = "Table title",  
modify_caption()`)

**A named code chunk** (`{r chunk-name}`)

# Cross-referencing tables

A bookdown output format, e.g.  
`bookdown::html_document2`

A table caption (`caption = "Table title",  
modify_caption()`)

A named code chunk (`{r chunk-name}`)

Reference with `\@ref(tab:chunk-name)`

## Your Turn 6

**Modify the YAML header to use  
bookdown::html\_document2 as the output**

**Add a caption to the merged table above using  
modify\_caption()**

**Reference that table below using this format:  
\@ref(tab:name-of-chunk)**

## Your turn 6

```
tbl_merge(both_tables) %>%  
  modify_caption(  
    "Regression models of participant markets and response"  
  )
```

# Your turn 6

























Regression models of participant markets and response						
Characteristic	Table 1			Table 2		
	OR <sup>†</sup>	95% CI <sup>†</sup>	p-value	Beta	95% CI <sup>†</sup>	p-value
Age	1.02	1.00, 1.04	0.087	0.00	-0.01, 0.01	>0.9
T Stage			0.6			0.2
<i>T1</i>	—	—		—	—	
<i>T2</i>	0.58	0.24, 1.37		0.38	0.04, 0.73	
<i>T3</i>	0.94	0.39, 2.28		0.28	-0.09, 0.66	
<i>T4</i>	0.79	0.33, 1.90		0.20	-0.16, 0.56	
Chemotherapy Treatment						0.12
<i>Drug A</i>				—	—	
<i>Drug B</i>				-0.20	-0.46, 0.05	

<sup>†</sup> OR = Odds Ratio, CI = Confidence Interval

Null deviance = 229; Null df = 182; Log-likelihood = -112; AIC = 234; BIC = 250; Deviance = 224; Residual df = 178; No. Obs. = 183

In Table 1, we show two regression models: a linear model of participant markets and a logistic regression model of participant response.

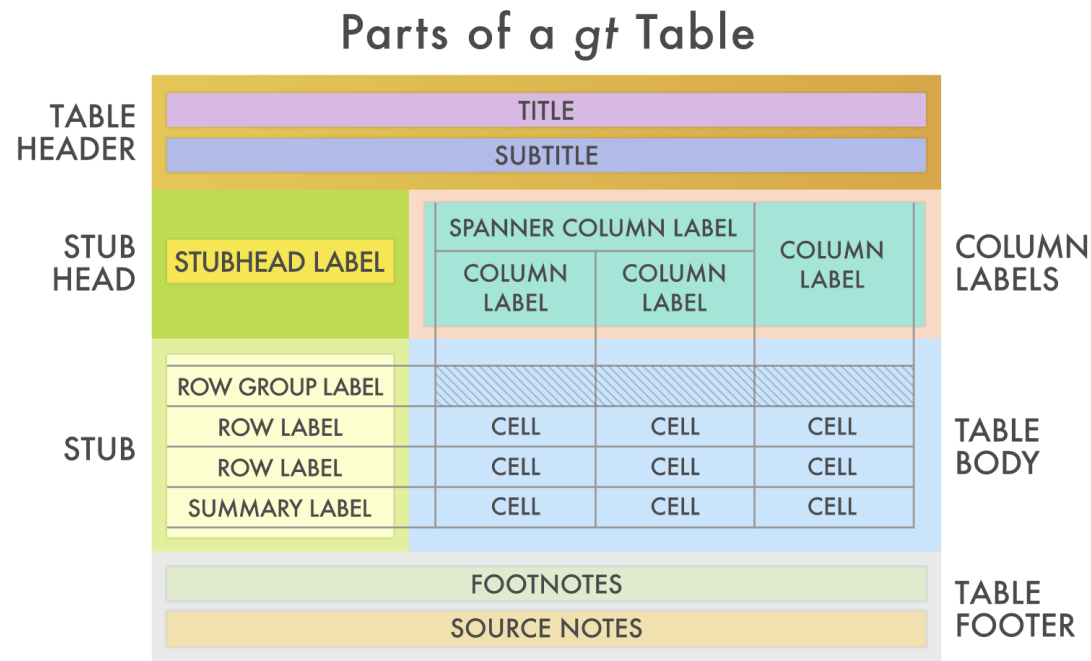
# gtsummary: Output formats

Print Engine	Function	HTML	PDF	RTF	Word
<a href="#">gt</a>	<code>as_gt()</code>				
<a href="#">kable</a>	<code>as_kable()</code>				
<a href="#">flextable</a>	<code>as_flex_table()</code>				
<a href="#">kableExtra</a>	<code>as_kable_extra()</code>				
<a href="#">huxtable</a>	<code>as_hux_table()</code>				
<a href="#">tibble</a>	<code>as_tibble()</code>				



# Also check out the *gt* package!

"construct a wide variety of useful tables with a cohesive set of table parts"



# Resources

**R Markdown Cookbook: Tables:** A series of recipes using `kable()` and friends

**gtsummary Website:** Many vignettes to learn more about gtsummary

**gt Website:** Vignettes and a short course on learning gt