LinearModelReporting.R

GlennTattersall

2019-04-22

```r
# Summary of Regression / Linear Models as HTML Table

# Source: https://strengejacke.github.io/sjPlot/articles/tab_model_estimates.html#a-simple-html-table-f

# tab_model() is the pendant to plot_model(), however, instead of creating
# plots, tab_model() creates HTML-tables that will be displayed either in your
# IDE's viewer-pane, in a web browser or in a knitr-markdown-document

# HTML is the only output-format, you can't (directly) create a LaTex or PDF
# output from tab_model() and related table-functions. However, it is possible
# to easily export the tables into Microsoft Word or Libre Office Writer.

# This vignette shows how to create table from regression models with tab_model().

# Note. Due to the custom CSS, the layout of the table inside a
# knitr-document differs from the output in the viewer-pane and web browser.

# Install packages in this order:
# sjlabelled -> sjmisc -> sjstats -> ggeffects -> sjPlot


# load packages
library(sjPlot)
```

```
## #refugeeswelcome
```

```r
library(sjmisc)
library(sjlabelled)

## sample data
data(efc)
efc <- as_factor(efc, c161sex, c172code)

# A simple HTML table from regression results
# First, we fit two linear models to demonstrate the tab_model()-function.

m1 <- lm(barthtot ~ c160age + c12hour + c161sex + c172code, data = efc)
m2 <- lm(neg_c_7 ~ c160age + c12hour + c161sex + e17age, data = efc)

# The simplest way of producing the table output is by passing the fitted model
# as parameter. By default, estimates, confidence intervals (CI) and p-values
# (p) are reported. As summary, the numbers of observations as well as the
# R-squared values are shown.

summary(m1)
```

```
##
## Call:
```

```
## lm(formula = barthtot ~ c160age + c12hour + c161sex + c172code,
##     data = efc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.144 -14.944   4.401  18.661  72.393
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 87.14994    4.68009  18.621  < 2e-16 ***
## c160age     -0.20716    0.07211  -2.873  0.00418 **
## c12hour     -0.27883    0.01865 -14.950  < 2e-16 ***
## c161sex2    -0.39402    2.08893  -0.189  0.85044
## c172code2    1.36596    2.28440   0.598  0.55004
## c172code3   -1.64045    2.84037  -0.578  0.56373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.35 on 815 degrees of freedom
##   (87 observations deleted due to missingness)
## Multiple R-squared:  0.2708, Adjusted R-squared:  0.2664
## F-statistic: 60.54 on 5 and 815 DF,  p-value: < 2.2e-16
```

```r
# compare summary to tab_model:
tab_model(m1)
```

Total score BARTHEL INDEX

Predictors

Estimates

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

carer'age

-0.21

-0.35 – -0.07

0.004

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

Female

-0.39

-4.49 – 3.70

0.850

intermediate level ofeducation

1.37

-3.11 – 5.84

0.550

high level of education

-1.64

-7.21 – 3.93

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```r
# Automatic labelling
colnames(efc)
```

```
##  [1] "c12hour"  "e15relat" "e16sex"   "e17age"   "e42dep"   "c82cop1"
##  [7] "c83cop2"  "c84cop3"  "c85cop4"  "c86cop5"  "c87cop6"  "c88cop7"
## [13] "c89cop8"  "c90cop9"  "c160age"  "c161sex"  "c172code" "c175empl"
## [19] "barthtot" "neg_c_7"  "pos_v_4"  "quol_5"   "resttotn" "tot_sc_e"
## [25] "n4pstu"   "nur_pst"
```

```r
# columns look like quite unremarkable features, but look closely:
str(efc$c160age)
```

```
##  num [1:908] 56 54 80 69 47 56 61 67 59 49 ...
##  - attr(*, "label")= chr "carer' age"
```

```r
str(efc$c12hour)
```

```
##  num [1:908] 16 148 70 168 168 16 161 110 28 40 ...
##  - attr(*, "label")= chr "average number of hours of care per week"
```

```r
# As the sjPlot-packages features labelled data, the coefficients in the table
# are already labelled in this example. The name of the dependent variable(s)
# is used as main column header for each model. For non-labelled data, the
# coefficient names are shown.

# Turn off automatic labelling
# To turn off automatic labelling, use auto.label = FALSE, or provide an empty
# character vector for pred.labels and dv.labels.

tab_model(m1, auto.label = FALSE)
```

barthtot

Predictors

Estimates

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

c160age

-0.21

-0.35 – -0.07

0.004

c12hour

-0.28

-0.32 – -0.24

<0.001

c161sex2

-0.39

-4.49 – 3.70

0.850

c172code2

1.37

-3.11 – 5.84

0.550

c172code3

-1.64

-7.21 – 3.93

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```r
# some categorical data are already sufficient
data(mtcars)
m.mtcars <- lm(mpg ~ cyl + hp + wt, data = mtcars)
tab_model(m.mtcars)
```

mpg

Predictors

Estimates

CI

p

(Intercept)

38.75

35.25 – 42.25

<0.001

cyl

-0.94

-2.02 – 0.14

0.098

hp

-0.02

-0.04 – 0.01

0.140

wt

-3.17

-4.62 – -1.72

<0.001

Observations

32

R2 / adjusted R2

0.843 / 0.826

```
# but maybe you want to add details, you can do so manually. Note you need to
# specify the intercept predictor as well in a linear model:
tab_model(m.mtcars,
          pred.labels=c("(Intercept)", "Cylinders", "Horse Power", "Weight"))
```

mpg

Predictors

Estimates

CI

p

(Intercept)

38.75

35.25 – 42.25

<0.001

Cylinders

-0.94

-2.02 – 0.14

0.098

Horse Power

-0.02

-0.04 – 0.01

0.140

Weight

-3.17

-4.62 – -1.72

<0.001

Observations

32

R2 / adjusted R2

0.843 / 0.826

```r
# What to do about model intercept?
# You can forcibly remove the intercept, at which point, the intercept effect
# simply becomes encapsulated into one of the main categorical variables.
m1.0 <- lm(barthtot ~ c160age + c12hour + c161sex + c172code - 1, data = efc)
tab_model(m1)
```

Total score BARTHEL INDEX

Predictors

Estimates

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

carer'age

-0.21

-0.35 – -0.07

0.004

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

Female

-0.39

-4.49 – 3.70

0.850

intermediate level ofeducation

1.37

-3.11 – 5.84

0.550

high level of education

-1.64

-7.21 – 3.93

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```
tab_model(m1.0)
```

Total score BARTHEL INDEX

Predictors

Estimates

CI

p

carer'age

-0.21

-0.35 – -0.07

0.004

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

Male

87.15

77.98 – 96.32

<0.001

Female

86.76

78.00 – 95.51

<0.001

intermediate level ofeducation

1.37

-3.11 – 5.84

0.550

high level of education

-1.64

-7.21 – 3.93

0.564

Observations

821

R2 / adjusted R2

0.874 / 0.873

```r
# More than one model
# tab_model() can print multiple models at once, which are then printed
# side-by-side. Identical predictor coefficients are matched in a row.
tab_model(m1, m2)
```

Total score BARTHEL INDEX

Negative impact with 7items

Predictors

Estimates

CI

p

Estimates

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

9.83

7.34 – 12.33

<0.001

carer'age

-0.21

-0.35 – -0.07

0.004

0.01

-0.01 – 0.03

0.359

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

0.02

0.01 – 0.02

<0.001

Female

-0.39

-4.49 – 3.70

0.850

0.43

-0.15 – 1.01

0.147

intermediate level ofeducation

1.37

-3.11 – 5.84

0.550

high level of education

-1.64

-7.21 – 3.93

0.564

elder'age

0.01

-0.03 – 0.04

0.741

Observations

821

879

R2 / adjusted R2

0.271 / 0.266

0.067 / 0.063

```r
# Generalized linear models
# For generalized linear models, the ouput is slightly adapted.
# Instead of Estimates, the column is named Odds Ratios, Incidence Rate Ratios
# etc., depending on the model.
# The coefficients are, by default, automatically
# converted (exponentiated). Furthermore, pseudo R-squared statistics are
# shown in the summary.

m3 <- glm(
  tot_sc_e ~ c160age + c12hour + c161sex + c172code,
  data = efc, family = poisson(link = "log")
)

efc$neg_c_7d <- ifelse(efc$neg_c_7 < median(efc$neg_c_7, na.rm = TRUE), 0, 1)

m4 <- glm(
  neg_c_7d ~ c161sex + barthtot + c172code,
  data = efc, family = binomial(link = "logit")
)

tab_model(m3, m4)
```

Services for elderly

neg c 7 d

Predictors

Incidence Rate Ratios

CI

p

Odds Ratios

CI

p

(Intercept)

0.30

0.21 – 0.45

<0.001

6.54

3.62 – 11.81

<0.001

carer'age

1.01

1.01 – 1.02

<0.001

average number of hoursof care per week

1.00

1.00 – 1.00

<0.001

Female

1.01

0.86 – 1.19

0.867

1.87

1.30 – 2.68

0.001

intermediate level ofeducation

1.47

1.21 – 1.78

<0.001

1.23

0.84 – 1.82

0.288

high level of education

1.90

1.52 – 2.37

<0.001

1.37

0.84 – 2.23

0.204

Total score BARTHEL INDEX

0.97

0.96 – 0.97

<0.001

Observations

840

815

Cox & Snell's R2 / Nagelkerke's R2

0.083 / 0.106

0.184 / 0.247

```r
# Untransformed estimates on the linear scale
# To plot the estimates on the linear scale, use transform = NULL.
tab_model(m3, m4, transform = NULL, auto.label = T)
```

Services for elderly

neg c 7 d

Predictors

Log-Mean

CI

p

Log-Odds

CI

p

(Intercept)

-1.19

-1.58 – -0.80

<0.001

1.88

1.29 – 2.47

<0.001

carer'age

0.01

0.01 – 0.02

<0.001

average number of hoursof care per week

0.00

0.00 – 0.00

<0.001

Female

0.01

-0.15 – 0.17

0.867

0.63

0.26 – 0.99

0.001

intermediate level ofeducation

0.39

0.19 – 0.58

<0.001

0.21

-0.18 – 0.60

0.288

high level of education

0.64

0.42 – 0.86

<0.001

0.31

-0.17 – 0.80

0.204

Total score BARTHEL INDEX

-0.03

-0.04 – -0.03

<0.001

Observations

840

815

Cox & Snell's R2 / Nagelkerke's R2

0.083 / 0.106

0.184 / 0.247

```r
# More complex models
# Other models, like hurdle- or zero-inflated models, also work with tab_model().
# In this case, the zero inflation model is indicated in the table.
# Use show.zeroinf = FALSE to hide this part from the table.

library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
data(bioChemists)

m5 <- zeroinfl(art ~ . | ., data = bioChemists)
tab_model(m5)
```

Dependent variable

Predictors

Incidence Rate Ratios

CI

p

(Intercept)

1.90

1.50 – 2.41

<0.001

femWomen

0.81

0.72 – 0.92

0.001

marMarried

1.11

0.97 – 1.28

0.145

kid5

0.87

0.79 – 0.95

0.003

phd

0.99

0.94 – 1.06

0.842

ment

1.02

1.01 – 1.02

<0.001

Zero-Inflated Model

(Intercept)

0.56

0.21 – 1.52

0.257

femWomen

1.12

0.64 – 1.93

0.695

marMarried

0.70

0.38 – 1.31

0.265

kid5

1.24

0.85 – 1.83

0.269

phd

1.00

0.75 – 1.33

0.993

ment

0.87

0.80 – 0.96

0.003

```
tab_model(m5, show.zeroinf = F)
```

Dependent variable

Predictors

Incidence Rate Ratios

CI

p

(Intercept)

1.90

1.50 – 2.41

<0.001

femWomen

0.81

0.72 – 0.92

0.001

marMarried

1.11

0.97 – 1.28

0.145

kid5

0.87

0.79 – 0.95

0.003

phd

0.99

0.94 – 1.06

0.842

ment

1.02

1.01 – 1.02

<0.001

```r
# You can combine any model in one table.

tab_model(m1, m3, auto.label = FALSE)
```

barthtot

tot_sc_e

Predictors

Estimates

CI

p

Incidence Rate Ratios

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

0.30

0.21 – 0.45

<0.001

c160age

-0.21

-0.35 – -0.07

0.004

1.01

1.01 – 1.02

<0.001

c12hour

-0.28

-0.32 – -0.24

<0.001

1.00

1.00 – 1.00

<0.001

c161sex2

-0.39

-4.49 – 3.70

0.850

1.01

0.86 – 1.19

0.867

c172code2

1.37

-3.11 – 5.84

0.550

1.47

1.21 – 1.78

<0.001

c172code3

-1.64

-7.21 – 3.93

0.564

1.90

1.52 – 2.37

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```r
# Show or hide further columns
# tab_model() has some argument that allow to show or hide specific columns
# from the output:

# show.est to show/hide the column with model estimates.
tab_model(m1, m3, auto.label = FALSE,  show.est=FALSE)
```

barthtot

tot_sc_e

Predictors

p

p

(Intercept)

<0.001

<0.001

c160age

0.004

<0.001

c12hour

<0.001

<0.001

c161sex2

0.850

0.867

c172code2

0.550

<0.001

c172code3

0.564

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```r
# show.ci to show/hide the column with confidence intervals.
tab_model(m1, m3, auto.label = FALSE, show.ci=FALSE)
```

barthtot

tot_sc_e

Predictors

Estimates

p

Incidence Rate Ratios

p

(Intercept)

87.15

<0.001

0.30

<0.001

c160age

-0.21

0.004

1.01

<0.001

c12hour

-0.28

<0.001

1.00

<0.001

c161sex2

-0.39

0.850

1.01

0.867

c172code2

1.37

0.550

1.47

<0.001

c172code3

-1.64

0.564

1.90

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```r
# show.se to show/hide the column with standard errors.
tab_model(m1, m3,  auto.label = FALSE, show.se=FALSE)
```

barthtot

tot_sc_e

Predictors

Estimates

CI

p

Incidence Rate Ratios

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

0.30

0.21 – 0.45

<0.001

c160age

-0.21

-0.35 – -0.07

0.004

1.01

1.01 – 1.02

<0.001

c12hour

-0.28

-0.32 – -0.24

<0.001

1.00

1.00 – 1.00

<0.001

c161sex2

-0.39

-4.49 – 3.70

0.850

1.01

0.86 – 1.19

0.867

c172code2

1.37

-3.11 – 5.84

0.550

1.47

1.21 – 1.78

<0.001

c172code3

-1.64

-7.21 – 3.93

0.564

1.90

1.52 – 2.37

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```
# show.std to show/hide the column with standardized estimates
# (and their standard errors).
tab_model(m1, m3, auto.label = FALSE, show.std=T, show.ci=F)
```

barthtot

tot_sc_e

Predictors

Estimates

std. Beta

p

Incidence Rate Ratios

p

(Intercept)

87.15

<0.001

0.30

<0.001

c160age

-0.21

-0.09

0.004

1.01

<0.001

c12hour

-0.28

-0.48

<0.001

1.00

<0.001

c161sex2

-0.39

-0.01

0.850

1.01

0.867

c172code2

1.37

0.02

0.550

1.47

<0.001

c172code3

-1.64

-0.02

0.564

1.90

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```r
# show.p to show/hide the column with p-values.
tab_model(m1, m3, auto.label = FALSE, show.p=FALSE, show.ci=F)
```

barthtot

tot_sc_e

Predictors

Estimates

Incidence Rate Ratios

(Intercept)

87.15

0.30

c160age

-0.21

1.01

c12hour

-0.28

1.00

c161sex2

-0.39

1.01

c172code2

1.37

1.47

c172code3

-1.64

1.90

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```r
# show.stat to show/hide the column with the coefficients' test statistics.
tab_model(m1, m3, auto.label = FALSE, show.stat=T, show.ci=F)
```

barthtot

tot_sc_e

Predictors

Estimates

Statistic

p

Incidence Rate Ratios

Statistic

p

(Intercept)

87.15

18.62

<0.001

0.30

-5.97

<0.001

c160age

-0.21

-2.87

0.004

1.01

4.41

<0.001

c12hour

-0.28

-14.95

<0.001

1.00

3.72

<0.001

c161sex2

-0.39

-0.19

0.850

1.01

0.17

0.867

c172code2

1.37

0.60

0.550

1.47

3.89

<0.001

c172code3

-1.64

-0.58

0.564

1.90

5.65

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```
tab_model(m1, m3, auto.label = FALSE, show.stat=F, show.ci=F)
```

barthtot

tot_sc_e

Predictors

Estimates

p

Incidence Rate Ratios

p

(Intercept)

87.15

<0.001

0.30

<0.001

c160age

-0.21

0.004

1.01

<0.001

c12hour

-0.28

<0.001

1.00

<0.001

c161sex2

-0.39

0.850

1.01

0.867

c172code2

1.37

0.550

1.47

<0.001

c172code3

-1.64

0.564

1.90

<0.001

Observations

821

840

R2 / adjusted R2

0.271 / 0.266

0.083 / 0.106

```
# show.df for linear mixed models, when p-values are based on degrees of
# freedom with Kenward-Rogers approximation, these degrees of freedom are shown.
# p.val needs to be set to "kr"

library(lme4)
```

```
## Loading required package: Matrix
```

```
data(sleepstudy)
str(sleepstudy)
```

```
## 'data.frame':    180 obs. of  3 variables:
##  $ Reaction: num  250 259 251 321 357 ...
##  $ Days    : num  0 1 2 3 4 5 6 7 8 9 ...
##  $ Subject : Factor w/ 18 levels "308","309","310",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
me1<-lmer(Reaction ~ Days + (1|Subject), data=sleepstudy)
#tab_model(me1, auto.label = FALSE, show.stat=T, show.se=T, show.df=T,
#          p.val="kr")


# Adding columns
# In the following example, standard errors, standardized coefficients
# and test statistics are also shown.

tab_model(m1, show.se = TRUE, show.std = TRUE, show.stat = TRUE)
```

Total score BARTHEL INDEX

Predictors

Estimates

std. Error

std. Beta

standardized std. Error

CI

standardized CI

Statistic

p

(Intercept)

87.15

4.68

77.98 – 96.32

18.62

<0.001

carer'age

-0.21

0.07

-0.09

0.03

-0.35 – -0.07

-0.16 – -0.03

-2.87

0.004

average number of hoursof care per week

-0.28

0.02

-0.48

0.03

-0.32 – -0.24

-0.54 – -0.42

-14.95

<0.001

Female

-0.39

2.09

-0.01

0.03

-4.49 – 3.70

-0.06 – 0.05

-0.19

0.850

intermediate level ofeducation

1.37

2.28

0.02

0.04

-3.11 – 5.84

-0.05 – 0.10

0.60

0.550

high level of education

-1.64

2.84

-0.02

0.04

-7.21 – 3.93

-0.09 – 0.05

-0.58

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```r
# Removing columns
# In the following example, default columns are removed.

tab_model(m3, m4, show.ci = FALSE, show.p = FALSE, auto.label = FALSE)
```

tot_sc_e

neg_c_7d

Predictors

Incidence Rate Ratios

Odds Ratios

(Intercept)

0.30

6.54

c160age

1.01

c12hour

1.00

c161sex2

1.01

1.87

c172code2

1.47

1.23

c172code3

1.90

1.37

barthtot

0.97

Observations

840

815

Cox & Snell's R2 / Nagelkerke's R2

0.083 / 0.106

0.184 / 0.247

```
# Removing and sorting columns
# Another way to remove columns, which also allows to reorder the columns,
# is the col.order-argument. This is a character vector, where each element
# indicates a column in the output. The value est, for instance,
# indicates the estimates, while std.est is the column for standardized
# estimates and so on.

# By default, col.order contains all possible columns. All columns that
# should shown (see previous tables, for example using show.se = TRUE to
# show standard errors, or show.st = TRUE to show standardized estimates) are
# then printed by default. Colums that are excluded from col.order are not
# shown, no matter if the show-arguments are TRUE or FALSE.
# So if show.se = TRUE, but col.order does not contain the element "se",
# standard errors are not shown. On the other hand, if show.est = FALSE,
# but col.order does include the element "est", the columns with estimates
# are not shown.
# In summary, col.order can be used to exclude columns from the table and
# to change the order of colums.


tab_model(
  m1, show.se = TRUE, show.std = TRUE, show.stat = TRUE,
  col.order = c("p", "stat", "est", "std.se", "se", "std.est")
)
```

Total score BARTHEL INDEX

Predictors

p

Statistic

Estimates

standardized std. Error

std. Error

std. Beta

(Intercept)

<0.001

18.62

87.15

4.68

carer’age

0.004

-2.87

-0.21

0.03

0.07

-0.09

average number of hoursof care per week

<0.001

-14.95

-0.28

0.03

0.02

-0.48

Female

0.850

-0.19

-0.39

0.03

2.09

-0.01

intermediate level ofeducation

0.550

0.60

1.37

0.04

2.28

0.02

high level of education

0.564

-0.58

-1.64

0.04

2.84

-0.02

Observations

821

R2 / adjusted R2

0.271 / 0.266

```
# Collapsing columns
# With collapse.ci and collapse.se, the columns for confidence intervals
# and standard errors can be collapsed into one column together with the
# estimates. Sometimes this table layout is required.

tab_model(m1, collapse.ci = TRUE)
```

Total score BARTHEL INDEX

Predictors

Estimates

p

(Intercept)

87.15(77.98 – 96.32)

<0.001

carer'age

-0.21(-0.35 – -0.07)

0.004

average number of hoursof care per week

-0.28(-0.32 – -0.24)

<0.001

Female

-0.39(-4.49 – 3.70)

0.850

intermediate level ofeducation

1.37(-3.11 – 5.84)

0.550

high level of education

-1.64(-7.21 – 3.93)

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```
# Defining own labels
# There are different options to change the labels of the column headers
# or coefficients, e.g. with:

# pred.labels to change the names of the coefficients in the Predictors column.
# Note that the length of pred.labels must exactly match the amount of predictors
# in the Predictor column.
# dv.labels to change the names of the model columns, which are labelled with
# the variable labels / names from the dependent variables.
# Furthermore, there are various string-arguments, to change the name of
# column headings.

tab_model(
  m1, m2,
  pred.labels = c("Intercept", "Age (Carer)", "Hours per Week", "Gender (Carer)",
                  "Education: middle (Carer)", "Education: high (Carer)",
                  "Age (Older Person)"),
  dv.labels = c("First Model", "M2"),
  string.pred = "Coefficient",
  string.ci = "Conf. Int (95%)",
  string.p = "P-Value"
)
```

First Model

M2

Coefficient

Estimates

Conf. Int (95%)

P-Value

Estimates

Conf. Int (95%)

P-Value

Intercept

87.15

77.98 – 96.32

<0.001

9.83

7.34 – 12.33

<0.001

Age (Carer)

-0.21

-0.35 – -0.07

0.004

0.01

-0.01 – 0.03

0.359

Hours per Week

-0.28

-0.32 – -0.24

<0.001

0.02

0.01 – 0.02

<0.001

Gender (Carer)

-0.39

-4.49 – 3.70

0.850

0.43

-0.15 – 1.01

0.147

Education: middle (Carer)

1.37

-3.11 – 5.84

0.550

Education: high (Carer)

-1.64

-7.21 – 3.93

0.564

Age (Older Person)

0.01

-0.03 – 0.04

0.741

Observations

821

879

R2 / adjusted R2

0.271 / 0.266

0.067 / 0.063

```
# I don't think there is a way to change the title of the "Estimates" column?

# First Model   M2
# Show asterisks instead of numeric p-values
# You can change the style of how p-values are displayed with the argument
# p.style. With p.style = "asterisk", the p-values are indicated as * in
# the table.

tab_model(m1, m2, p.style = "a")
```

Total score BARTHEL INDEX

Negative impact with 7items

Predictors

Estimates

CI

Estimates

CI

(Intercept)

87.15 ***

77.98 – 96.32

9.83 ***

7.34 – 12.33

carer'age

-0.21 **

-0.35 – -0.07

0.01

-0.01 – 0.03

average number of hoursof care per week

-0.28 ***

-0.32 – -0.24

0.02 ***

0.01 – 0.02

Female

-0.39

-4.49 – 3.70

0.43

-0.15 – 1.01

intermediate level ofeducation

1.37

-3.11 – 5.84

high level of education

-1.64

-7.21 – 3.93

elder'age

0.01

-0.03 – 0.04

Observations

821

879

R2 / adjusted R2

0.271 / 0.266

0.067 / 0.063

- p<0.05 ** p<0.01 *** p<0.001

```r
# Note: I personally find this annoying as it does not show p values at all but
# gives an impression of importance that may not be warranted.  I.e. when
# do you normally care about the significance of the intercept term?  Or does
# your field really care about p values, so why use *** to inflate or guide
# the reader toward emphasising something that they should discern themselves.

# Automatic matching for named vectors
# Another way to easily assign labels are named vectors. In this case,
# it doesn't matter if pred.labels has more labels than coefficients in the
# model(s), or in which order the labels are passed to tab_model(). The only
# requirement is that the labels' names equal the coefficients names as they
# appear in the summary()-output.

# example, coefficients are "c161sex2" or "c172code3"
summary(m1)
```

```
##
## Call:
## lm(formula = barthtot ~ c160age + c12hour + c161sex + c172code,
##     data = efc)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.144 -14.944   4.401  18.661  72.393
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 87.14994    4.68009  18.621  < 2e-16 ***
## c160age     -0.20716    0.07211  -2.873  0.00418 **
## c12hour     -0.27883    0.01865 -14.950  < 2e-16 ***
## c161sex2    -0.39402    2.08893  -0.189  0.85044
## c172code2    1.36596    2.28440   0.598  0.55004
## c172code3   -1.64045    2.84037  -0.578  0.56373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.35 on 815 degrees of freedom
##   (87 observations deleted due to missingness)
## Multiple R-squared:  0.2708, Adjusted R-squared:  0.2664
## F-statistic: 60.54 on 5 and 815 DF,  p-value: < 2.2e-16
```

```r
# create a named vector, pl:
pl <- c(
  `(Intercept)` = "Intercept",
  e17age = "Age (Older Person)",
  c160age = "Age (Carer)",
  c12hour = "Hours per Week",
  barthtot = "Barthel-Index",
  c161sex2 = "Gender (Carer)",
  c172code2 = "Education: middle (Carer)",
  c172code3 = "Education: high (Carer)",
  a_non_used_label = "We don't care"
)

cbind(pl)
```

```
##                  pl
## (Intercept)      "Intercept"
## e17age           "Age (Older Person)"
## c160age          "Age (Carer)"
## c12hour          "Hours per Week"
## barthtot         "Barthel-Index"
## c161sex2         "Gender (Carer)"
## c172code2        "Education: middle (Carer)"
## c172code3        "Education: high (Carer)"
## a_non_used_label "We don't care"
```

```r
# see how pl is actually named, so you can still use the column names in the
# model call but the pl variable holds more informative information that
# includes words, spaces, capital letters etc..

tab_model(
  m1, m2, m3, m4,
  pred.labels = pl,
  dv.labels = c("Model1", "Model2", "Model3", "Model4"),
```

```
  show.ci = FALSE,
  show.p = FALSE,
  transform = NULL
)
```

Model1

Model2

Model3

Model4

Predictors

Estimates

Estimates

Log-Mean

Log-Odds

Intercept

87.15

9.83

-1.19

1.88

Age (Carer)

-0.21

0.01

0.01

Hours per Week

-0.28

0.02

0.00

Gender (Carer)

-0.39

0.43

0.01

0.63

Education: middle (Carer)

1.37

0.39

0.21

Education: high (Carer)

-1.64

0.64

0.31

Age (Older Person)

0.01

Barthel-Index

-0.03

Observations

821

879

840

815

R2 / adjusted R2

0.271 / 0.266

0.067 / 0.063

0.083 / 0.106

0.184 / 0.247

```r
# Keep or remove coefficients from the table
# Using the terms- or rm.terms-argument allows us to explicitly show or
# remove specific coefficients from the table output.

tab_model(m1, terms = c("c160age", "c12hour"))
```

Total score BARTHEL INDEX

Predictors

Estimates

CI

p

carer'age

-0.21

-0.35 – -0.07

0.004

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

Observations

821

R2 / adjusted R2

0.271 / 0.266

```r
# Note that the names of terms to keep or remove should match the coefficients
# names.

# For categorical predictors, one example would be, which will remove the
# terms c172code2 and c161sex2 from the summary, even though those two
# terms were still used to fit the final model:

tab_model(m1, rm.terms = c("c172code2", "c161sex2"))
```

Total score BARTHEL INDEX

Predictors

Estimates

CI

p

(Intercept)

87.15

77.98 – 96.32

<0.001

carer'age

-0.21

-0.35 – -0.07

0.004

average number of hoursof care per week

-0.28

-0.32 – -0.24

<0.001

high level of education

-1.64

-7.21 – 3.93

0.564

Observations

821

R2 / adjusted R2

0.271 / 0.266

```
# For How to format an Anova table output see:

# http://www.understandingdata.net/2017/05/11/anova-tables-in-r/
```