

# R-Funktionen

## 1 R-Funktionen

Diese Seite listet all die R-Funktionen auf die ich schonmal benutzt habe. Ich versuche so weit es geht die ausgeführten Beispiele mit anzuzeigen. \*\*\*

Mit ? können wird die Dokumentation von R aufgerufen und weitere zusätzliche Informationen werden angezeigt.

```
# zum Beispiel
?row.names
```

c, nchar, data, str, dim, names, head, and tail.

```
# zeigt alle Observations an
row.names(mtcars)
```

```
## [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
## [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
## [7] "Duster 360"          "Merc 240D"           "Merc 230"
## [10] "Merc 280"            "Merc 280C"           "Merc 450SE"
## [13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
## [16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
## [22] "Dodge Challenger"    "AMC Javelin"         "Camaro Z28"
## [25] "Pontiac Firebird"     "Fiat X1-9"           "Porsche 914-2"
## [28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
## [31] "Maserati Bora"       "Volvo 142E"
```

---

Es folgen mehrere Beispiel mit R-Code

```
# Gibt die Werte der Variable mpg des Datenframes mtcars aus. (Also $ als Symbol).
mtcars$mpg
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

### 1.1 Funktionen die das Zentrum beschreiben

```
# Gibt den Mittelwert der Daten an.
mean(mtcars$mpg)
```

```
## [1] 20.09
```

```
# Gibt den Median der Daten an.
median(mtcars$mpg)
```

```
## [1] 19.2
```

```
# Überblick über alle Daten
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0   Min.   : 2
## 1st Qu.:12.0   1st Qu.: 26
## Median :15.0   Median : 36
## Mean   :15.4   Mean    : 43
## 3rd Qu.:19.0   3rd Qu.: 56
## Max.   :25.0   Max.    :120
```

## 1.2 Funktionen um Datenframes zu laden

```
#Zeigt das Verzeichnis an in welchen wir uns befinden.
getwd()
```

```
## [1] "/Users/user/AllGitHub/ProgrammingKnowledge"
```

```
#Wechsel des Verzeichnisses setwd('~Downloads') immer in '**' - Zeichen**.
setwd('~Downloads')
#csv-Datei einlesen.
read.csv('reddit.csv')
```

## 1.3 Andere Funktionen

```
# zeigt ein Subset
subset(mtcars, mtcars$mpg<=25 & mtcars$wt<=2.581 )
```

```
##           mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710  22.8   4 108.0  93 3.85  2.320 18.61  1  1    4    1
## Toyota Corona 21.5   4 120.1  97 3.70  2.465 20.01  1  0    3    1
```

```
# Zeigt einen Überblick an
summary(mtcars)
```

```
##      mpg      cyl      disp      hp
## Min.   :10.4   Min.   :4.00   Min.   : 71.1   Min.   : 52.0
## 1st Qu.:15.4   1st Qu.:4.00   1st Qu.:120.8   1st Qu.: 96.5
## Median :19.2   Median :6.00   Median :196.3   Median :123.0
## Mean   :20.1   Mean    :6.19   Mean    :230.7   Mean    :146.7
## 3rd Qu.:22.8   3rd Qu.:8.00   3rd Qu.:326.0   3rd Qu.:180.0
```

```
## Max. :33.9 Max. :8.00 Max. :472.0 Max. :335.0
##      drat      wt      qsec      vs
## Min. :2.76 Min. :1.51 Min. :14.5 Min. :0.000
## 1st Qu.:3.08 1st Qu.:2.58 1st Qu.:16.9 1st Qu.:0.000
## Median :3.69 Median :3.33 Median :17.7 Median :0.000
## Mean :3.60 Mean :3.22 Mean :17.8 Mean :0.438
## 3rd Qu.:3.92 3rd Qu.:3.61 3rd Qu.:18.9 3rd Qu.:1.000
## Max. :4.93 Max. :5.42 Max. :22.9 Max. :1.000
##      am      gear      carb
## Min. :0.000 Min. :3.00 Min. :1.00
## 1st Qu.:0.000 1st Qu.:3.00 1st Qu.:2.00
## Median :0.000 Median :4.00 Median :2.00
## Mean :0.406 Mean :3.69 Mean :2.81
## 3rd Qu.:1.000 3rd Qu.:4.00 3rd Qu.:4.00
## Max. :1.000 Max. :5.00 Max. :8.00
```

```
# neue Spalte mit year als überschriift und 1974 für alle
mtcars$year <- 1974
```

```
# Spalte löschen
mtcars <- subset(mtcars, select = -year)
```

```
# Ein Beispiel für Konditionen
mtcars$wt
```

```
## [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440
## [12] 4.070 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520
## [23] 3.435 3.840 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780
```

```
cond <- mtcars$wt < 3
cond
```

```
## [1] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE FALSE
## [23] FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE
```

```
mtcars$weight_class <- ifelse(cond, 'light', 'average')
mtcars$weight_class
```

```
## [1] "light" "light" "light" "average" "average" "average" "average"
## [8] "average" "average" "average" "average" "average" "average" "average"
## [15] "average" "average" "average" "light" "light" "light" "light"
## [22] "average" "average" "average" "average" "light" "light" "light"
## [29] "average" "light" "average" "light"
```

```
cond <- mtcars$wt > 3.5
mtcars$weight_class <- ifelse(cond, 'heavy', mtcars$weight_class)
mtcars$weight_class
```

```
## [1] "light" "light" "light" "average" "average" "average" "heavy"
## [8] "average" "average" "average" "average" "heavy" "heavy" "heavy"
```

```
## [15] "heavy" "heavy" "heavy" "light" "light" "light" "light"
## [22] "heavy" "average" "heavy" "heavy" "light" "light" "light"
## [29] "average" "light" "heavy" "light"
```

```
# entfernt code aus dem arbeitsbereich
rm(cond)
rm(efficient)
```

```
## Warning: Objekt 'efficient' nicht gefunden
```

```
# zeigt die Anzahl der Fahrzeuge mit bestimmten Werten an
table(mtcars$mpg)
```

```
##
## 10.4 13.3 14.3 14.7 15 15.2 15.5 15.8 16.4 17.3 17.8 18.1 18.7 19.2 19.7
## 2 1 1 1 1 2 1 1 1 1 1 1 1 2 1
## 21 21.4 21.5 22.8 24.4 26 27.3 30.4 32.4 33.9
## 2 2 1 2 1 1 1 2 1 1
```

```
# Für Faktrone als Datentypen
levels(reddit$age.range)
```

## 1.4 qplot

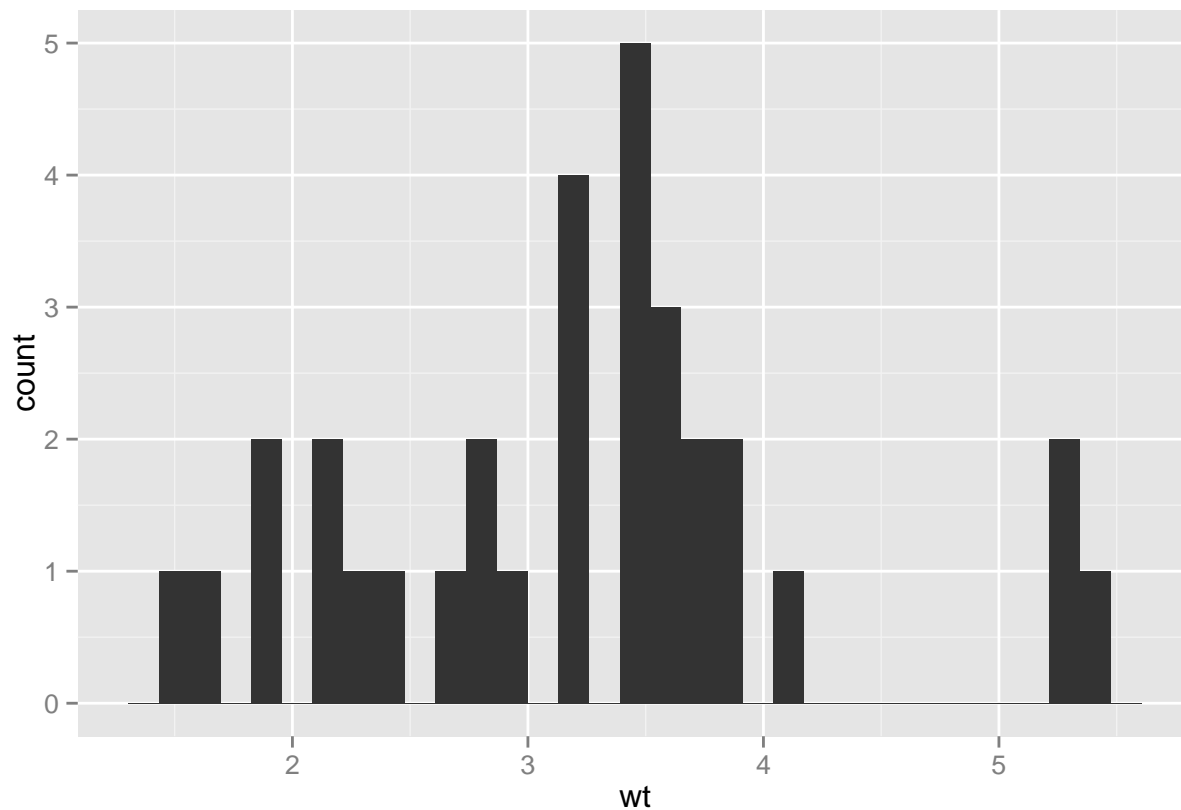
qplot is the basic plotting function in the ggplot2 package, designed to be familiar if you're used to plot from the base package. Parameter für qplot:

```
x = Variabelname data = Datename xlim = Vektor(Von, Bis) binwidth = Balkendicke facet_wrap(~gender,
ncol = 2) -- Aufteilen in Einzelne kleine Fenster
```

### 1.4.1 Um eine Plot zu zeichnen

```
#install.packages('ggplot2', dependencies = T)
library(ggplot2)
qplot(data= mtcars, x=wt)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



## 1.5 Datentypen

### 1. Vektoren

Ein Beispiel für Vektoren

```
a <- c(1,2,5.3,6,-2,4) # numeric vector
b <- c("one","two","three") # character vector
c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) #logical vector
```

### 2. Matrizen

Ein Beispiel für Matrizen

```
# generates 5 x 4 numeric matrix
y<-matrix(1:20, nrow=5,ncol=4)

# another example
cells <- c(1,26,24,68)
rnames <- c("R1", "R2")
cnames <- c("C1", "C2")
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,
  dimnames=list(rnames, cnames))
mymatrix[]
```

```
##      C1 C2
## R1   1 26
## R2  24 68
```

### (3) Arrays

Sind wie Matrizen aufgebaut, nur sind mehrere Dimensionen möglich Arrays are similar to matrices but can have more than two dimensions. See `help(array)` for details.

### (4) Data Frames

A data frame is more general than a matrix, in that different columns can have different modes (numeric, character, factor, etc.). This is similar to SAS and SPSS datasets.

```
d <- c(1,2,3,4)
e <- c("red", "white", "red", NA)
f <- c(TRUE,TRUE,TRUE,FALSE)
mydata <- data.frame(d,e,f)
names(mydata) <- c("ID","Color","Passed") # variable names
mydata
```

```
##      ID Color Passed
## 1     1   red    TRUE
## 2     2 white    TRUE
## 3     3   red    TRUE
## 4     4  <NA> FALSE
```

### 5. List

An ordered collection of objects (components). A list allows you to gather a variety of (possibly unrelated) objects under one name.

```
# example of a list with 4 components -
# a string, a numeric vector, a matrix, and a scalar
w <- list(name="Fred", mynumbers=a, mymatrix=y, age=5.3)
```

```
# example of a list containing two lists
v <- c(list1,list2)
```

### 6. Factors

Tell R that a variable is nominal by making it a factor. The factor stores the nominal values as a vector of integers in the range  $[1 \dots k]$  (where  $k$  is the number of unique values in the nominal variable), and an internal vector of character strings (the original values) mapped to these integers.

- Nominale Variablen

```
# variable gender with 20 "male" entries and
# 30 "female" entries
gender <- c(rep("male",20), rep("female", 30))
gender <- factor(gender)
# stores gender as 20 1s and 30 2s and associates
# 1=female, 2=male internally (alphabetically)
# R now treats gender as a nominal variable
summary(gender)
```

```
## female    male
##         30     20
```

- Ordinale Variablen

An ordered factor is used to represent an ordinal variable.

```
# variable rating coded as "large", "medium", "small"
rating <- c(rep("large"), rep("medium"), rep("small"))
rating <- ordered(rating)

summary(rating)
```

```
## large medium small
##      1      1      1
```

```
# recodes rating to 1,2,3 and associates
# 1=large, 2=medium, 3=small internally
# R now treats rating as ordinal
```

R will treat factors as nominal variables and ordered factors as ordinal variables in statistical procedures and graphical analyses. You can use options in the `factor()` and `ordered()` functions to control the mapping of integers to strings (overriding the alphabetical ordering). You can also use factors to create value labels. For more on factors see the [UCLA page](#).

```
length(mtcars) # number of the variables or components
```

```
## [1] 12
```

```
str(mtcars) # structure of an object
```

```
## 'data.frame':   32 obs. of  12 variables:
## $ mpg      : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl      : num   6  6  4  6  8  6  8  4  4  6 ...
## $ disp     : num  160 160 108 258 360 ...
## $ hp       : num  110 110  93 110 175 105 245  62  95 123 ...
## $ drat     : num   3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt       : num   2.62 2.88 2.32 3.21 3.44 ...
## $ qsec     : num  16.5 17 18.6 19.4 17 ...
## $ vs       : num   0  0  1  1  0  1  0  1  1  1 ...
## $ am       : num   1  1  1  0  0  0  0  0  0  0 ...
## $ gear     : num   4  4  4  3  3  3  3  4  4  4 ...
## $ carb     : num   4  4  1  1  2  1  4  2  2  4 ...
## $ weight_class: chr  "light" "light" "light" "average" ...
```

```
class(mtcars$wt) # class or type of an object
```

```
## [1] "numeric"
```

```
class(mtcars) # many options
```

```
## [1] "data.frame"
```

```
names(mtcars) # names of the Variables
```

```
## [1] "mpg"      "cyl"      "disp"     "hp"
## [5] "drat"     "wt"       "qsec"     "vs"
## [9] "am"       "gear"     "carb"     "weight_class"
```

```
mtcars # prints the object mtcars
```

```
##      mpg  cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160.0  110 3.90 2.620 16.46 0 1   4   4
## Mazda RX4 Wag  21.0   6  160.0  110 3.90 2.875 17.02 0 1   4   4
## Datsun 710     22.8   4  108.0   93 3.85 2.320 18.61 1 1   4   1
## Hornet 4 Drive  21.4   6  258.0  110 3.08 3.215 19.44 1 0   3   1
## Hornet Sportabout 18.7   8  360.0  175 3.15 3.440 17.02 0 0   3   2
## Valiant       18.1   6  225.0  105 2.76 3.460 20.22 1 0   3   1
## Duster 360    14.3   8  360.0  245 3.21 3.570 15.84 0 0   3   4
## Merc 240D     24.4   4  146.7   62 3.69 3.190 20.00 1 0   4   2
## Merc 230      22.8   4  140.8   95 3.92 3.150 22.90 1 0   4   2
## Merc 280      19.2   6  167.6  123 3.92 3.440 18.30 1 0   4   4
## Merc 280C     17.8   6  167.6  123 3.92 3.440 18.90 1 0   4   4
## Merc 450SE    16.4   8  275.8  180 3.07 4.070 17.40 0 0   3   3
## Merc 450SL    17.3   8  275.8  180 3.07 3.730 17.60 0 0   3   3
## Merc 450SLC   15.2   8  275.8  180 3.07 3.780 18.00 0 0   3   3
## Cadillac Fleetwood 10.4   8  472.0  205 2.93 5.250 17.98 0 0   3   4
## Lincoln Continental 10.4   8  460.0  215 3.00 5.424 17.82 0 0   3   4
## Chrysler Imperial 14.7   8  440.0  230 3.23 5.345 17.42 0 0   3   4
## Fiat 128      32.4   4   78.7   66 4.08 2.200 19.47 1 1   4   1
## Honda Civic   30.4   4   75.7   52 4.93 1.615 18.52 1 1   4   2
## Toyota Corolla 33.9   4   71.1   65 4.22 1.835 19.90 1 1   4   1
## Toyota Corona 21.5   4  120.1   97 3.70 2.465 20.01 1 0   3   1
## Dodge Challenger 15.5   8  318.0  150 2.76 3.520 16.87 0 0   3   2
## AMC Javelin   15.2   8  304.0  150 3.15 3.435 17.30 0 0   3   2
## Camaro Z28    13.3   8  350.0  245 3.73 3.840 15.41 0 0   3   4
## Pontiac Firebird 19.2   8  400.0  175 3.08 3.845 17.05 0 0   3   2
## Fiat X1-9     27.3   4   79.0   66 4.08 1.935 18.90 1 1   4   1
## Porsche 914-2 26.0   4  120.3   91 4.43 2.140 16.70 0 1   5   2
## Lotus Europa  30.4   4   95.1  113 3.77 1.513 16.90 1 1   5   2
## Ford Pantera L 15.8   8  351.0  264 4.22 3.170 14.50 0 1   5   4
## Ferrari Dino  19.7   6  145.0  175 3.62 2.770 15.50 0 1   5   6
## Maserati Bora  15.0   8  301.0  335 3.54 3.570 14.60 0 1   5   8
## Volvo 142E    21.4   4  121.0  109 4.11 2.780 18.60 1 1   4   2
##
##      weight_class
## Mazda RX4      light
```



```
## Mazda RX4 Wag          light
## Datsun 710             light
## Hornet 4 Drive         average
## Hornet Sportabout      average
## Valiant                average
## Duster 360             heavy
## Merc 240D              average
## Merc 230               average
## Merc 280               average
## Merc 280C              average
## Merc 450SE             heavy
## Merc 450SL             heavy
## Merc 450SLC            heavy
## Cadillac Fleetwood     heavy
## Lincoln Continental    heavy
## Chrysler Imperial      heavy
## Fiat 128               light
## Honda Civic            light
## Toyota Corolla         light
## Toyota Corona          light
## Dodge Challenger       heavy
## AMC Javelin            average
## Camaro Z28             heavy
## Pontiac Firebird       heavy
## Fiat X1-9              light
## Porsche 914-2          light
## Lotus Europa           light
## Ford Pantera L         average
## Ferrari Dino           light
## Maserati Bora          heavy
## Volvo 142E             light
```

```
ls() # list current objects
```

```
## [1] "a"      "b"      "c"      "cells"  "cnames" "d"
## [7] "e"      "f"      "gender" "mtcars" "mydata"  "mymatrix"
## [13] "rating" "rnames" "w"      "y"
```

```
c(object,object,...)      # combine objects into a vector
cbind(object, object, ...) # combine objects as columns
rbind(object, object, ...) # combine objects as rows
```

```
newobject <- edit(object) # edit copy and save as newobject
fix(object)               # edit in place
```

Age	Frequency
18-25	15
26-35	33
36-45	22

## 2 Tabellen

Zum Einbinden von Tabellen eignet sich R-Markdown ebenfalls, es muss nur die Tabelle in der Datei haben ein Beispiel:

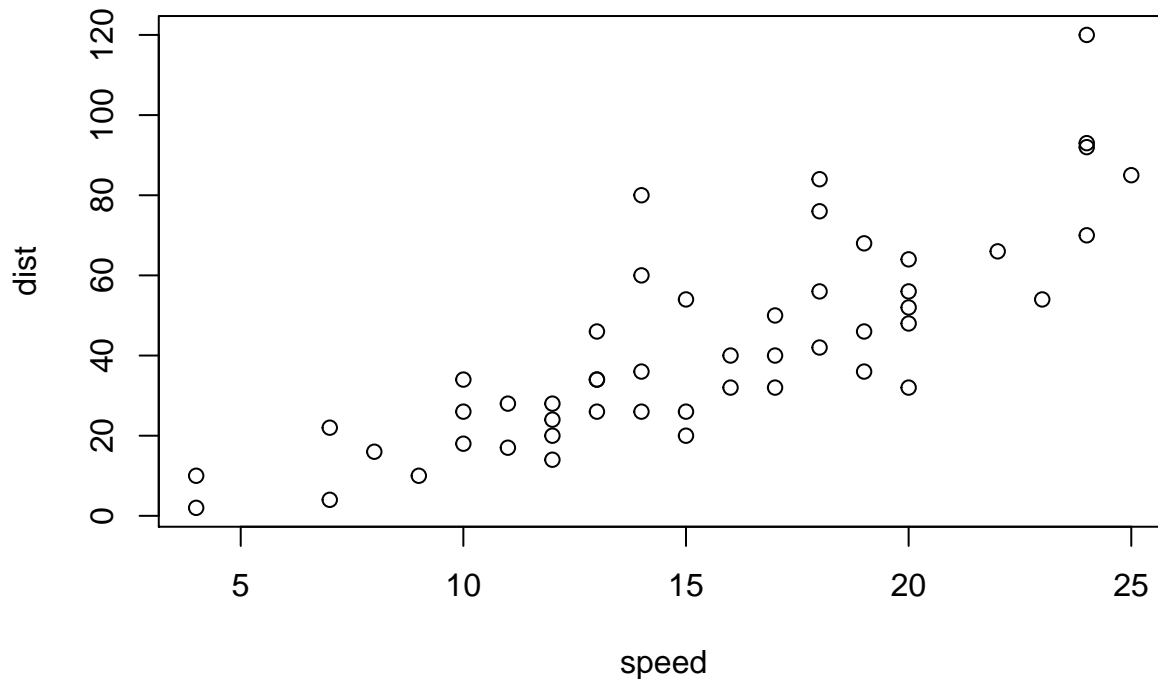
man kann die Layouts verändern, wenn man dies in die Meta daten der Datei schreibt

```
rmarkdown::tufte_handout:  
  highlight: zenburn
```

wenn man die Folgenden Daten im Chunkout schreibt wird die Tabele im R ausgeführt.

```
library(xtable)  
options(xtable.comment = FALSE)  
options(xtable.booktabs = TRUE)  
xtable(head(mtcars[, 1:6]), caption = "First rows of mtcars")
```

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot. ##Funktionen für das Setup des Pdf's unter R-Studio

### 2.0.0.1 So macht man links

[Udacity website] (<https://www.udacity.com/course/viewer#!/c-ud651/l-729069797/e-804129319/m-811719066>)

*This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.*