test.R

brianzelip

Wed Jan 20 18:50:10 2016

x <- 4  
y <- 5  
z <- x + y  
z

## [1] 9

example("read.csv")

##   
## rd.csv> ## using count.fields to handle unknown maximum number of fields  
## rd.csv> ## when fill = TRUE  
## rd.csv> test1 <- c(1:5, "6,7", "8,9,10")  
##   
## rd.csv> tf <- tempfile()  
##   
## rd.csv> writeLines(test1, tf)  
##   
## rd.csv> read.csv(tf, fill = TRUE) # 1 column  
## X1  
## 1 2  
## 2 3  
## 3 4  
## 4 5  
## 5 6  
## 6 7  
## 7 8  
## 8 9  
## 9 10  
##   
## rd.csv> ncol <- max(count.fields(tf, sep = ","))  
##   
## rd.csv> read.csv(tf, fill = TRUE, header = FALSE,  
## rd.csv+ col.names = paste0("V", seq\_len(ncol)))  
## V1 V2 V3  
## 1 1 NA NA  
## 2 2 NA NA  
## 3 3 NA NA  
## 4 4 NA NA  
## 5 5 NA NA  
## 6 6 7 NA  
## 7 8 9 10  
##   
## rd.csv> unlink(tf)  
##   
## rd.csv> ## "Inline" data set, using text=  
## rd.csv> ## Notice that leading and trailing empty lines are auto-trimmed  
## rd.csv>   
## rd.csv> read.table(header = TRUE, text = "  
## rd.csv+ a b  
## rd.csv+ 1 2  
## rd.csv+ 3 4  
## rd.csv+ ")  
## a b  
## 1 1 2  
## 2 3 4

help.search("knit")  
  
example("knit")

## Warning in example("knit"): no help found for 'knit'

newV <- 1:10  
newV

## [1] 1 2 3 4 5 6 7 8 9 10

class(newV)

## [1] "integer"

newV \* 2

## [1] 2 4 6 8 10 12 14 16 18 20

newW <- c(1:10, 4)  
newW

## [1] 1 2 3 4 5 6 7 8 9 10 4

help(c)  
write.csv(newW, file="newW.csv")  
list.files()

## [1] "newV.csv" "newW.csv"   
## [3] "R workshop handout.docx" "R workshop handout.pdf"   
## [5] "R workshop handout.pptx" "R workshop slides.pptx"   
## [7] "test.html" "test.R"   
## [9] "test.spin.R" "test.spin.Rmd"

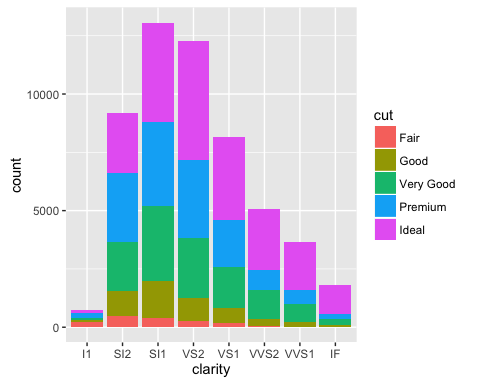
inTheFlesh <- read.csv("newW.csv", header = F)  
View(inTheFlesh)  
  
  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.3

data(diamonds)  
head(diamonds)

## carat cut color clarity depth table price x y z  
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  
## 4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63  
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  
## 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48

p <- ggplot(diamonds, aes(x=clarity, fill=cut))  
p + geom\_bar()



#  
# Begin visualization w/ ggplot2  
#  
  
# Load the ggplot2 package  
library("ggplot2")  
  
# Load the diamonds dataset provided by ggplot2  
data("diamonds")  
  
# Show only the first 6 rows of the diamonds data  
head(diamonds)

## carat cut color clarity depth table price x y z  
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  
## 4 0.29 Premium I VS2 62.4 58 334 4.20 4.23 2.63  
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  
## 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48

# Show all of the diamonds data  
View(diamonds)  
  
# Show metadata about the diamonds data  
help(diamonds)