

RStudio Cheat Sheet

T.Scofield

September 14, 2021

Making a list of numbers, calculating statistics from it

Suppose I want things like the mean, sd, median, etc. for the list of numbers 9, 11, 7, 13, 10.

```
c(9, 11, 7, 13, 10) -> x
mean(~x)
```

```
## [1] 10
```

Tables

To make a frequency table

```
ssurv <- read.csv("http://scofield.site/teaching/data/csv/ssurv.csv")
tally(~ selfhandedness, data=ssurv)
```

```
## selfhandedness
##      L      R
##    1  31 248
```

If I want a relative frequency table instead

```
tally(~ selfhandedness, data=ssurv, format="proportion")
```

```
## selfhandedness
##                  L          R
## 0.003571429 0.110714286 0.885714286
```

For bivariate data (i.e., two-way tables)

```
tally(~ selfhandedness | sex, data=ssurv)
```

```
##           sex
## selfhandedness  F  M
##                0  1
##           L  15  16
##           R 124 124
```

Adding the “format” switch

```
tally(~ selfhandedness | sex, data=ssurv, format="percent")
```

```
##           sex
## selfhandedness  F      M
##                0 0.0000000 0.7092199
##           L 10.7913669 11.3475177
##           R 89.2086331 87.9432624
```

gives us percentages out of the whole, done for each column. If we want to reverse the roles of `sex` and `handedness`,

```
tally(~ sex | selfhandedness, data=ssurv, format="percent")
```

```
##      selfhandedness
## sex                L          R
## F    0.0000  48.3871  50.0000
## M 100.0000  51.6129  50.0000
```

Now, to make the total of all combined cells make up 100 percent,

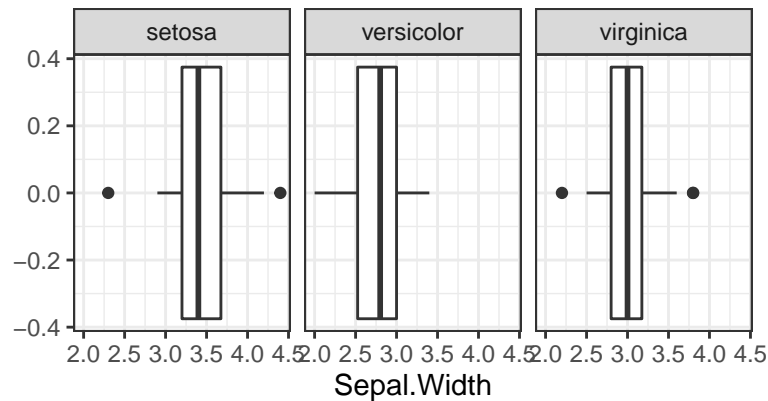
```
tally(~selfhandedness | sex, data=ssurv) %>% prop.table()
```

```
##              sex
## selfhandedness      F      M
##              0.00000000 0.003571429
##              L 0.053571429 0.057142857
##              R 0.442857143 0.442857143
```

Plotting bivariate data

In the `iris` data frame, there is a column (categorical) called `Species` and another column (quantitative) called `Sepal.Width`. If I want side-by-side boxplots for the quantitative variable broken down by the categorical one

```
gf_boxplot(~ Sepal.Width | Species, data=iris)
```



Now, try out this modification

```
gf_boxplot(Species ~ Sepal.Width, data=iris)
```

