SOC 4015/5050: Lecture 11 Functions

Christopher Prener, Ph.D.

Fall 2018

Packages

- `base`
- `ggplot2`
- `ggstatsplot`
- `grdevices`
- `Hmisc`
- `naniar`
- `pwr`
- `stats`

In-line `R` Code

Rounding Values

```
base::round(x, digits = val)
```

Implementing In-line `R` Code

````
```{r load-data}
library(ggplot2)
auto <- mpg
```

The average highway fuel efficiency in the data set is
`r round(mean(auto$hwy), digits = 3)`.
````

Scatterplots

Basic Scatterplot

```
ggplot2::geom_point(mapping = aes(x = xvar, y = yvar))
```

Scatterplot with Smoothed Line

```
ggplot2::geom_smooth(method = "lm",
  mapping = aes(color = "#hex"))
```
[1]

[1] Use http://colorhexa.com to selection hex values!

Scatterplot with Grouping Variable

```
ggplot2::geom_point(mapping = aes(x = xvar, y = yvar,
  color = groupVar))
```

Scatterplot with Smoothed Line by Grouping Variable

```
ggplot2::geom_smooth(method = "lm",
  mapping = aes(color = groupVar, linetype = groupVar))
```

Scatterplot with Facet

```
ggplot2::facet_grid(.  ~ facetVar)
```

Statistical Scatterplot

```
ggstatsplot::ggscatterstats(data = data, x = xvar,
  y = yvar)
```

Create a File to Save a Statistical Scatterplot

```
grdevices::png(filepath, width = val, height = val)
```
[2]

[2] Values for `width` and `height` should be specified in points.

Writing the Statistical Scatterplot to File

```
grdevices::dev.off()
```

Creating Data Objects

Creating a Vector

```
base::c(element, element, element)
```

Creating a Data Frame

```
base::c(vector, vector, stringsAsFactors = FALSE)
```

Creating a Matrix

```
base::as.matrix(objectName)
```

Missing Data

Missing Data Anlysis

```
naniar::miss_var_summary(data)
```

Remove All Missing Data

```
stats::na.omit(data)
```

Pearson's r

Basic Approach

```
stats::cor(dataFrame, use, method = "pearson")
```

Hmisc Approach

```
Hmisc::rcorr(matrix, type = "pearson")
```

Full Table Approach

```
corrTable(dataFrame, coef = "pearson", listwise = TRUE,
  round = 3, pStar = 3, ...)
```

Sample Size Estimate

```
pwr::pwr.r.test(r = rVal, sig.level = .05, power = powerVal,
  alernative = "two.sided")
```