SOC 4015/5050: Lecture-02 Functions

Christopher Prener, Ph.D.

Fall 2018

Packages

- `base`
- `knitr`
- `rmarkdown`
- `tidyverse`
  - `dplyr`
  - `ggplot2`
  - `magrittr`
- `utils`

Lecture-02a: Introducing `ggplot2`

Basic Plot

```
ggplot2::ggplot(data = dataFrame) +
   geom(mapping = aes(aesthetic))
```

Geometric Objects

Each of these geoms must be paired with an initial `ggplot` function call.

Line Plot

```
geom_freqpoly(mapping = aes(aesthetic))
```

Bar Plot

```
geom_bar(mapping = aes(aesthetic))
```

Scatter Plot

```
geom_point(mapping = aes(x = var1, y = var2))
```

Smoothed Line

```
geom_smooth(mapping = aes(x = var1, y = var2))
```

Box Plot

```
geom_box(mapping = aes(x = var1, y = var2))
```

The x variable should be the discrete variable.

Lecture-02b: The Grammar of Graphics

Basic Template

```
ggplot2::ggplot(data = dataFrame) +
   geom(mapping = aes(aesthetic),
     stat = statistics,
     position = position,
   ) +
   coordinateFunction +
   facetFunction
```

Geometric Objects

This geom must be paired with an initial `ggplot` function call.

Area Plot

```
geom_area(mapping = aes(aesthetic), stat = "bin")
```

Aesthetics

Use fill for bar plots and histograms, and color for point-based plots like scatter plots.

Add One Color

```
geom(mapping = aes(aesthetic, color = "colorText"))
```

Add Color Based on Additional Variable

```
geom(mapping = aes(aesthetic, color = colorVar))
```

Position Adjustments

Dodge

```
geom(mapping = aes(aesthetic, fill = colorVar),
   position = "dodge")
```

Jitter

```
geom(mapping = aes(x = var1, y = var2, color = colorVar),
   position = "jitter")
```

Coordinate Systems

Flip x and y Axes

```
geom(mapping = aes(aesthetic)) +
   coord_flip()
```

Lecture-02c: Verbs for Cleaning Data

Verbs for Cleaning Data

Rename Variables

```
dplyr::rename(dataFrame, newName = oldName)
```

Reorder Variables, Low to High

```
dplyr::arrange(dataFrame, varlist)
```

varlist items should be separated by commas

Reorder Variables, High to Low

```
dplyr::arrange(dataFrame, desc(varlist))
```

varlist items should be separated by commas

Subset Data, Specific Observations

```
dplyr::filter(dataFrame, expression)
```

Subset Data, Keep Specific Variables

```
dplyr::select(dataFrame, varlist)
```

varlist items should be separated by commas

Subset Data, Drop Specific Variables

```
dplyr::select(dataFrame, -varlist)
```

varlist items should be separated by commas with each item individually labeled with the dash drop symbol

Create New Variables

```
dplyr::mutate(dataFrame, newVar = expression)
```

`ifelse` Outcomes

```
base::ifelse(expression, trueOutcome, falseOutcome)
```

Pipe Operator

Basic Syntax

`%>%` - "then"

Example with Assignment

```
mpg %>%
    select(manufacturer, model, cty, hwy) %>%
    rename(cityMpg = cty) %>%
    rename(hwyMpg = hwy) -> autoData
```

Remember that `dataFrame` names do not need to be included for many functions when included in piped code

Logical Operators

`&` - "and"
`|` - "or"

Relational Operators

`<` - "less than"
`<=` - "less than or equal to"
`>` - "greater than"
`>=` - "greater than or equal to"
`==` - "exactly equal to"
`!=` - "not equal to"

Arithmetic Operators

`+` - "addition"
`-` - "subtraction"
`*` - "multiplication"
`/` - "division"
`^` - "exponentiation"

Viewing Observations

First Six Observations

```
utils::head(dataFrame)
```

Last Six Observations

```
utils::tail(dataFrame)
```

Frequency Tables

```
base::table(dataFrame$var)
```

Lecture-02d: Structuring Notebooks

Create New R Notebook

File ▷ New File ▷ R Notebook

Create New Project

File ▷ New Project ▷ New Directory ▷ New Project