Getting started

To work with R Markdown:

- Install R
- Install the lastest version of RStudio
- Install the latest version of the knitr package: `install.packages("knitr")`

To knit a PDF file, you need to install TeX.

- Easy way is to install the tinytex package by running the following lines:
  ```r
  install.packages("tinytex")
  tinytex::install_tinytex()
  ```

- If you want full version of TeX: For Mac install MacTeX. For Windows install TeX Live.
- More info:
  - R Markdown Reference Guide
  - R Markdown Cheat Sheet
Prepare for analyses

```r
set.seed(1234)

#install.packages("tidyverse")
#install.packages("stargazer")
#install.packages("pander")

library(tidyverse)
library(stargazer)
library(pander)
```

Without specify the options of chunk, you could get `warning` or package messages.

Basic console output

To insert an R code chunk, you can type or insert it manually. You can also use the shortcut key (Windows: Ctrl + Alt + I; OS X: Cmd + Option + I). This will produce the following code chunk:

You can label a code chunk with a name (no space). Pressing tab when inside the braces will show code chunk options.

The following R code chunk is labelled `basic-df` and will be displayed as follow:

```r
x <- 1:10
y <- round(rnorm(10, x, 1), 2)
df <- data.frame(x, y)
df
```

```r
##   x  y
## 1 1 -0.21
## 2 2  2.28
## 3 3  4.08
## 4 4  1.65
## 5 5  5.43
## 6 6  6.51
## 7 7  6.43
## 8 8  7.45
## 9 9  8.44
##10 10 9.11
```
R Code chunk features

Frequently used chunk options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>include</code></td>
<td>If FALSE, knitr will run the code but prevent the code chunk AND results from appearing.</td>
</tr>
<tr>
<td><code>echo</code></td>
<td>If FALSE, knitr will run the code, show the results but prevent the code chunk from appearing (useful for embedding figures or tables).</td>
</tr>
<tr>
<td><code>error</code></td>
<td>If FALSE, knitr will not display any error messages generated by the code.</td>
</tr>
<tr>
<td><code>message</code></td>
<td>If FALSE, knitr will not display any messages generated by the code.</td>
</tr>
<tr>
<td><code>warning</code></td>
<td>If FALSE, knitr will not display any warning messages generated by the code.</td>
</tr>
</tbody>
</table>

Echo and Results

The following code hides the command input (i.e., `echo=FALSE`) but displays the table output.

```r
## x  y
## 1  1  0.52
## 2  2  1.00
## 3  3  2.22
## 4  4  4.06
## 5  5  5.96
## 6  6  5.89
## 7  7  6.49
## 8  8  7.09
## 9  9  8.16
## 10 10 12.42
```

You can also display a `r` object with `backtick` `r` object-name `backtick` : The first element of `y` is 0.52.

Message and Warning

A code chunk without any specification of options show all warnings and messages which might be unnecessary for readers:

```r
df %>%
  summarize_at(vars(y), funs(mean))
```

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(.), trim = .2), ~ median(.), na.rm = TRUE))
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

```r
y
## 1 5.381
```

This code does not output warnings:

```r
df %>%
  summarize_at(vars(y), funs(mean))
```
Basic markdown functionality

List items

Simple dot points:
- Point 1
- Point 2
- Point 3

and numeric dot points:
  1. Number 1
  2. Number 2
  3. Number 3

and nested dot points:
- A
  - A.1
  - A.2
- B
  - B.1
  - B.2

Tables

Manual tables can be included using the following notation:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>Purple</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
<td>Gold</td>
</tr>
<tr>
<td>3</td>
<td>Non-binary</td>
<td>White</td>
</tr>
</tbody>
</table>

For displaying `data.frame` as table, you can create prettier output by using `pander` or `kable` functions.

With `pander`:

Table 3: Fancy table from pander

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2.22</td>
</tr>
<tr>
<td>4</td>
<td>4.06</td>
</tr>
<tr>
<td>5</td>
<td>5.96</td>
</tr>
<tr>
<td>6</td>
<td>5.89</td>
</tr>
<tr>
<td>7</td>
<td>6.49</td>
</tr>
<tr>
<td>8</td>
<td>7.09</td>
</tr>
<tr>
<td>9</td>
<td>8.16</td>
</tr>
<tr>
<td>10</td>
<td>12.42</td>
</tr>
</tbody>
</table>
With **kable:**

### Table 4: Fancy table from kable

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.52</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>2.22</td>
</tr>
<tr>
<td>4</td>
<td>4.06</td>
</tr>
<tr>
<td>5</td>
<td>5.96</td>
</tr>
<tr>
<td>6</td>
<td>5.89</td>
</tr>
<tr>
<td>7</td>
<td>6.49</td>
</tr>
<tr>
<td>8</td>
<td>7.09</td>
</tr>
<tr>
<td>9</td>
<td>8.16</td>
</tr>
<tr>
<td>10</td>
<td>12.42</td>
</tr>
</tbody>
</table>

For regression tables, the default output is not very pretty:

```r
mod1 <- y ~ x
res1 <- lm(formula = mod1, data = df)
mod2 <- y ~ x^2
res2 <- lm(formula = mod2, data = df)

summary(res1)
```

```r
##
## Call:
## lm(formula = mod1, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.2256 -0.5495 -0.1818  0.3684  1.8902
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.9120    0.7070  -1.29  0.233
## x            1.1442    0.1139  10.04 8.23e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.035 on 8 degrees of freedom
## Multiple R-squared: 0.9265, Adjusted R-squared: 0.9173
## F-statistic: 100.8 on 1 and 8 DF, p-value: 8.232e-06
```

```r
summary(res2)
```

```r
##
## Call:
## lm(formula = mod2, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.2256 -0.5495 -0.1818  0.3684  1.8902
```
## Coefficients:

|                | Estimate | Std. Error | t value | Pr(>|t|) |
|----------------|----------|------------|---------|----------|
| (Intercept)    | -0.9120  | 0.7070     | -1.29   | 0.233    |
| x              | 1.1442   | 0.1139     | 10.04   | 8.23e-06 *** |

---

**Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

**Residual standard error: 1.035 on 8 degrees of freedom**

**Multiple R-squared: 0.9265, Adjusted R-squared: 0.9173**

**F-statistic: 100.8 on 1 and 8 DF, p-value: 8.232e-06**

Use `stargazer` package to display format regression tables:

Table 5:

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependent variable:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>1.144***</td>
<td>1.144***</td>
</tr>
<tr>
<td></td>
<td>(0.114)</td>
<td>(0.114)</td>
</tr>
<tr>
<td>Constant</td>
<td>-0.912</td>
<td>-0.912</td>
</tr>
<tr>
<td></td>
<td>(0.707)</td>
<td>(0.707)</td>
</tr>
<tr>
<td>Observations</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>R²</td>
<td>0.926</td>
<td>0.926</td>
</tr>
<tr>
<td>Adjusted R²</td>
<td>0.917</td>
<td>0.917</td>
</tr>
<tr>
<td>Residual Std. Error (df = 8)</td>
<td>1.035</td>
<td>1.035</td>
</tr>
<tr>
<td>F Statistic (df = 1; 8)</td>
<td>100.826***</td>
<td>100.826***</td>
</tr>
</tbody>
</table>

**Note:** *p<0.1; **p<0.05; ***p<0.01**

More info: Cheat Sheet
Plots

You can also display plots from `ggplot2` or other graphic packages.

Sample Plot

Images

Images can be embedded using `knitr::include_graphics()`:

Source: Statistics How To “Misleading Graphs: Real Life Examples”
Equations

Equations are included by using LaTeX notation and including them either between single dollar signs (inline equations) or double dollar signs (displayed equations). If you hang around the Q&A site CrossValidated you’ll be familiar with this idea.

There are inline equations such as $y_i = \alpha + \beta x_i + e_i$.

And displayed formulas:

$$
\frac{1}{1 + \exp(-x)}
$$

$$
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
$$

$$
X = (x + a)(x - b)
= x(x - b) + a(x - b)
= x^2 + x(a - b) - ab
$$

More info: LaTeX wiki