R Code

Jeremiah Lewis

2022-04-13

Table of contents

Multi-Language Stats Book Demo

1 NOTE: Link to PDF is here

2 Motivation

Quarto is a promising successor to RMarkdown, in part because it isn't tied to a specific programming language. For publishing introductory (statistics) textbooks, one often wants to show code examples in multiple programming languages (usually Python, R, and Julia) as the field (and applications) are currently split amongst them. This site demonstrates different strategies for incorporating multiple programming languages into a single book.

3 Example 1: Crossreferencing examples in other languages

With this approach, the book is written in a main language, shown here as Python, and links to snippets in other languages are provided below each code block.

One advantage here is that each notebook file is dedicated to a separate programming language and can be executed in a dedicated environment. In terms of open source contributorship, it also makes ownership and maintenance responsibilities clearly separated on a per-file basis. The clear downside is that it means the book is not able to put all programming languages on equal footing and on mobile / tablet, the book will be harder to read as you will have to toggle between tabs or go forward / back between pages.

3.1 Example: How to Plot a Figure

Snippet from: https://quarto.org/docs/computations/python.html#code-blocks

```python
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
  subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

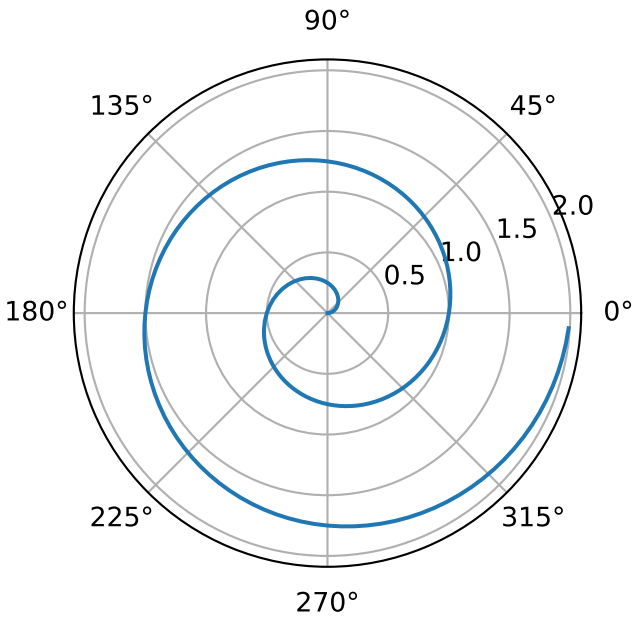

Figure 3.1: A line plot on a polar axis

See Section 5.1 for Julia code and Section 5.1 for R code.

4 Python and Julia in Separate Tabs, using Python Kernel and Julia 'Magic'

In this example, the book is written in multiple languages in parallel, with all languages rendered in a single Jupyter notebook, using the Python Kernel and pyjulia extension (or other extensions like rpy2 as necessary). The clear downsides here are that non-Python code is called through Python, debugging is more difficult, and reproducibility is diminished. The number of dependencies required is also unfortunately maximized.

4.0.0.1 Python

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
  subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

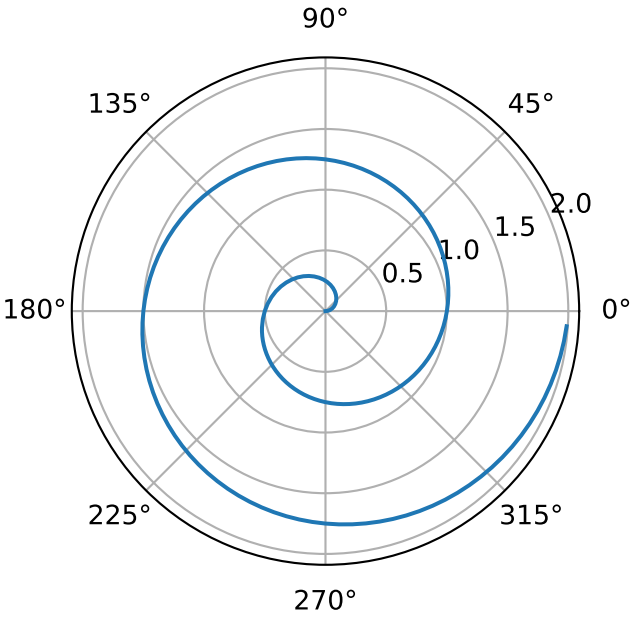

Figure 4.1: A line plot on a polar axis

4.0.0.2 Julia

See Section 5.1.

4.0.0.3 R

See Section 6.1.

Julia Code

5 Julia Supporting Code

5.1 Chapter 1

Snippet from: https://quarto.org/docs/computations/julia.html#code-blocks

```
import CairoMakie

x = range(0, 10, length=100)
y = sin.(x)
CairoMakie.lines(x, y)
```

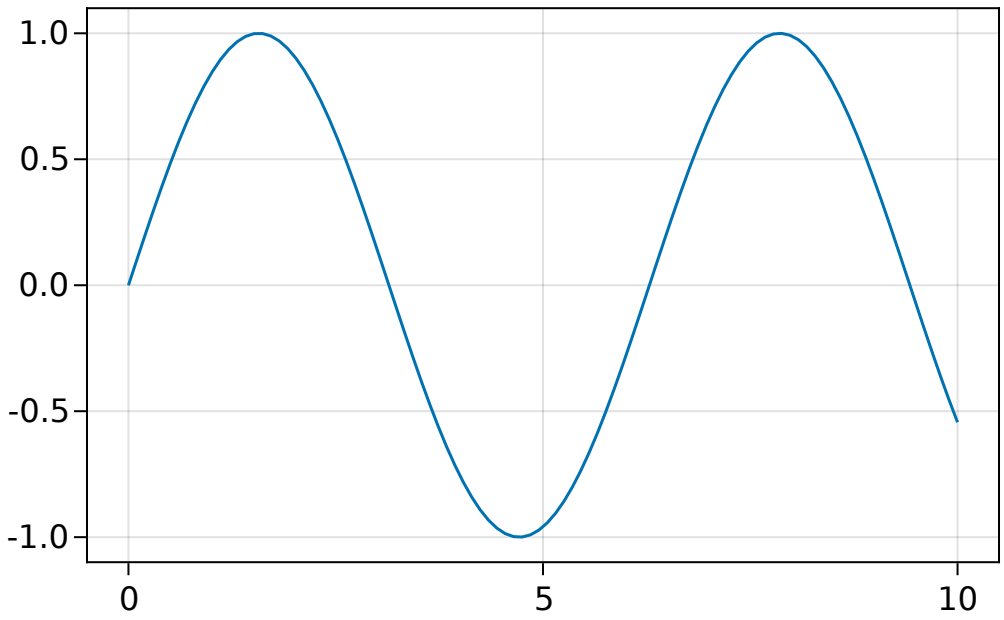

Figure 5.1: Covariance ellipses

```
using Plots

plot([0; 1.0], [ 0.12000000000; 0.12000000003])
```

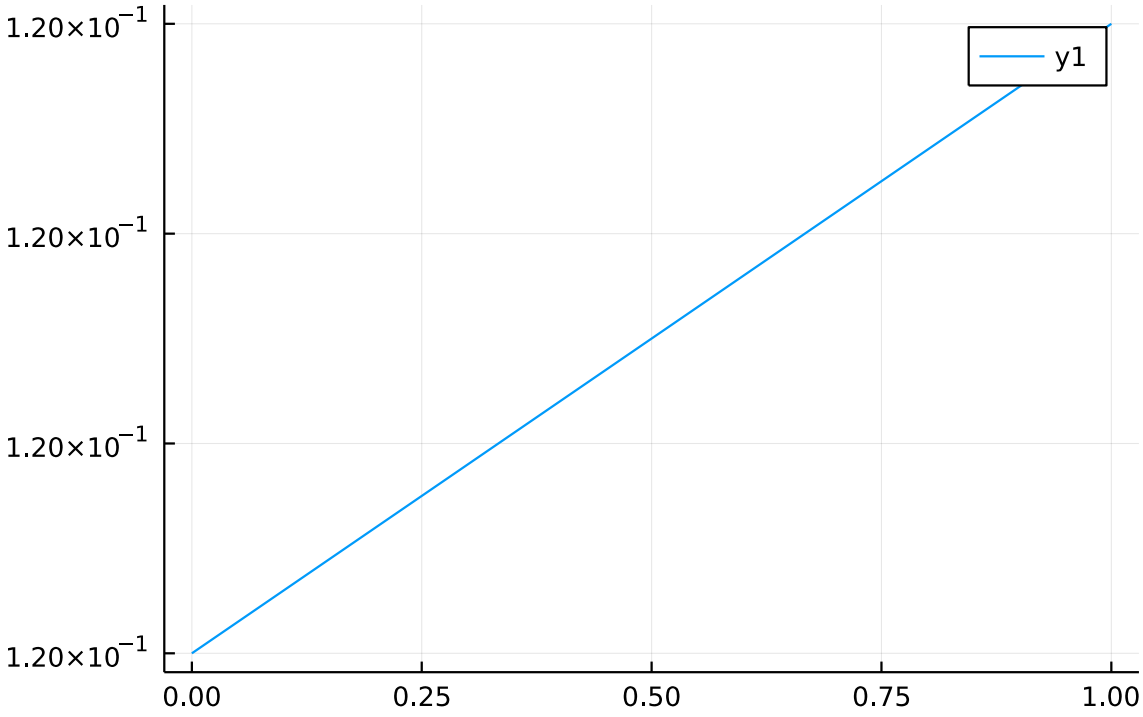

Figure 5.2: Covariance ellipses

5.2 Chapter 2

```
using StatsPlots

covellipse([0,2], [2 1; 1 4], n_std=2,
           aspect_ratio=1, label="cov1")
covellipse!([1,0], [1 -0.5; -0.5 3],
            showaxes=true, label="cov2")
```

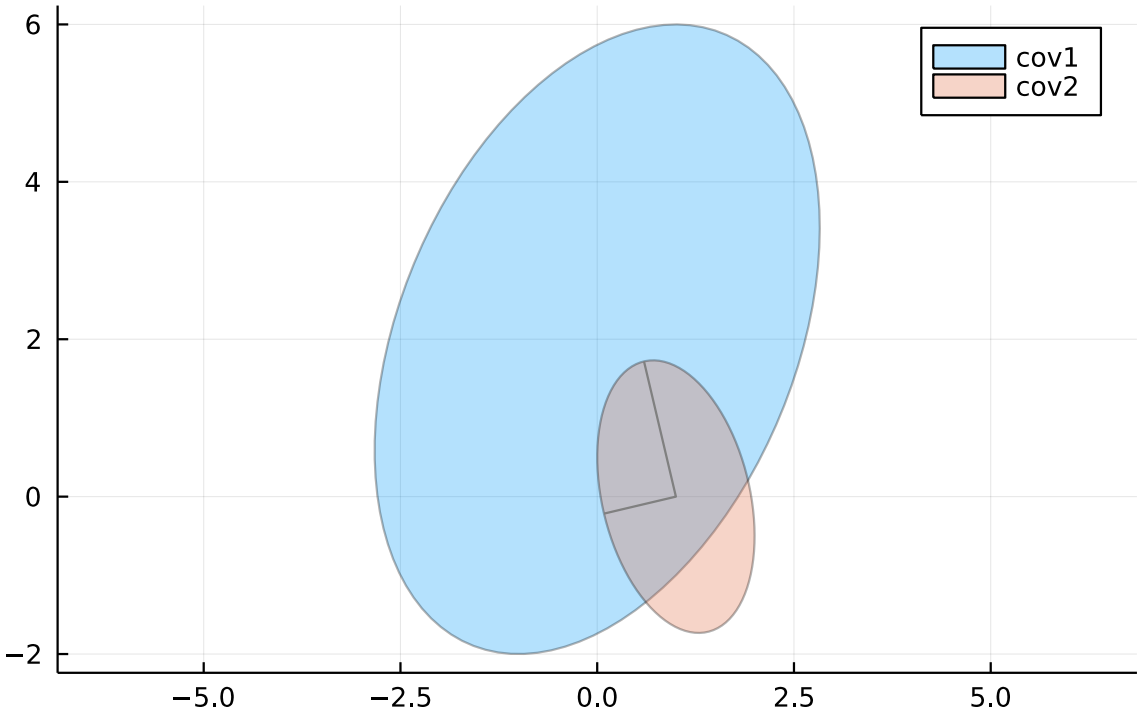

Figure 5.3: Covariance ellipses

5.3 Chapter 3

```
using StatsPlots

covellipse([0,2], [2 1; 1 4], n_std=2,
           aspect_ratio=1, label="cov1")
covellipse!([1,0], [1 -0.5; -0.5 3],
            showaxes=true, label="cov2")
```

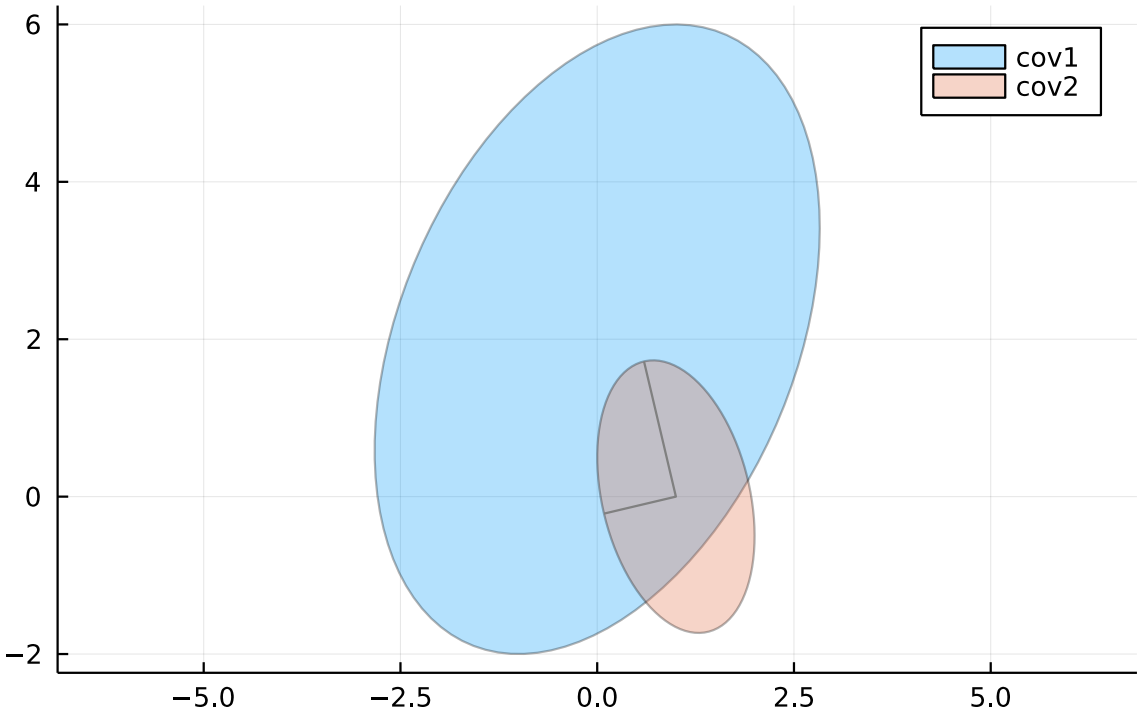

Figure 5.4: Covariance ellipses

5.4 Chapter 4

```
using StatsPlots

covellipse([0,2], [2 1; 1 4], n_std=2,
           aspect_ratio=1, label="cov1")
covellipse!([1,0], [1 -0.5; -0.5 3],
            showaxes=true, label="cov2")
```

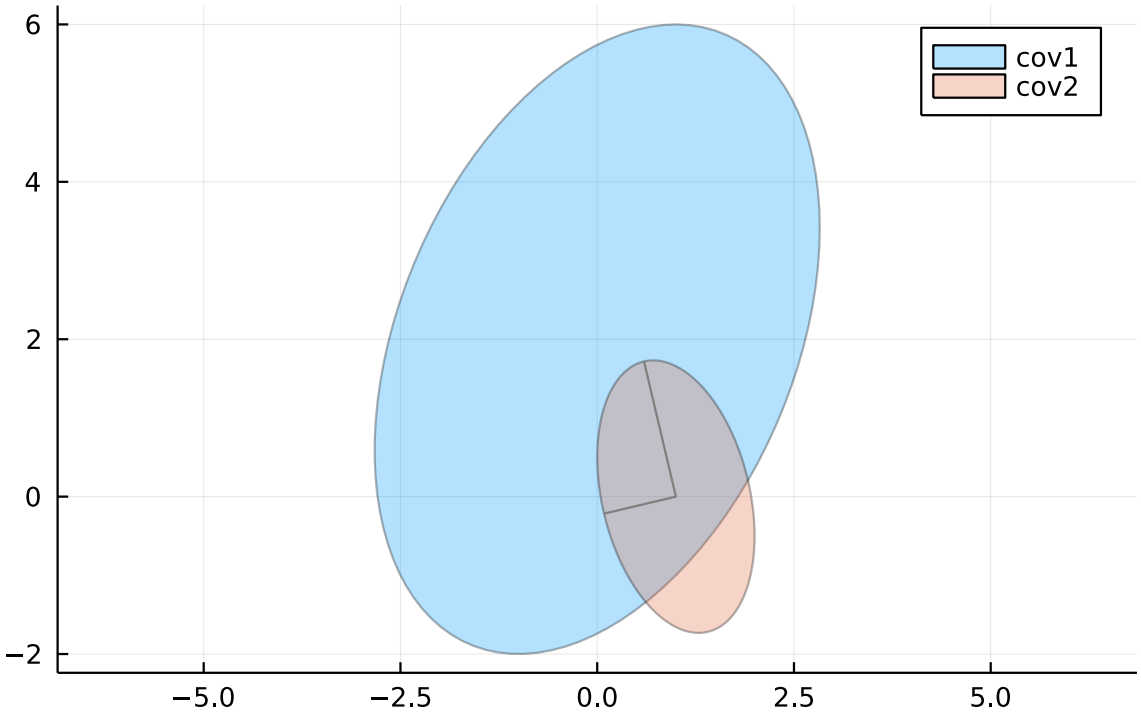

Figure 5.5: Covariance ellipses

5.5 Chapter 5

```
using StatsPlots

covellipse([0,2], [2 1; 1 4], n_std=2,
           aspect_ratio=1, label="cov1")
covellipse!([1,0], [1 -0.5; -0.5 3],
            showaxes=true, label="cov2")
```

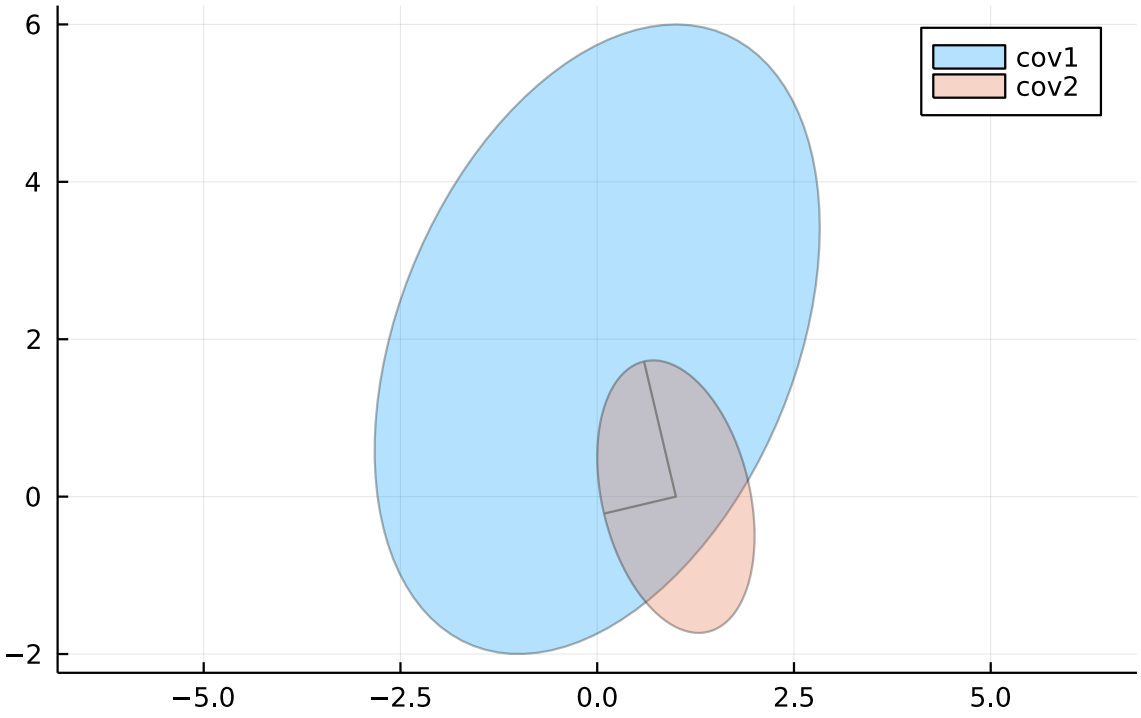

Figure 5.6: Covariance ellipses

R Code

6 R Supporting Code

6.1 Chapter 1

Snippet from: https://quarto.org/docs/computations/julia.html#code-blocks

```
plot(1:5, 6:10)
```

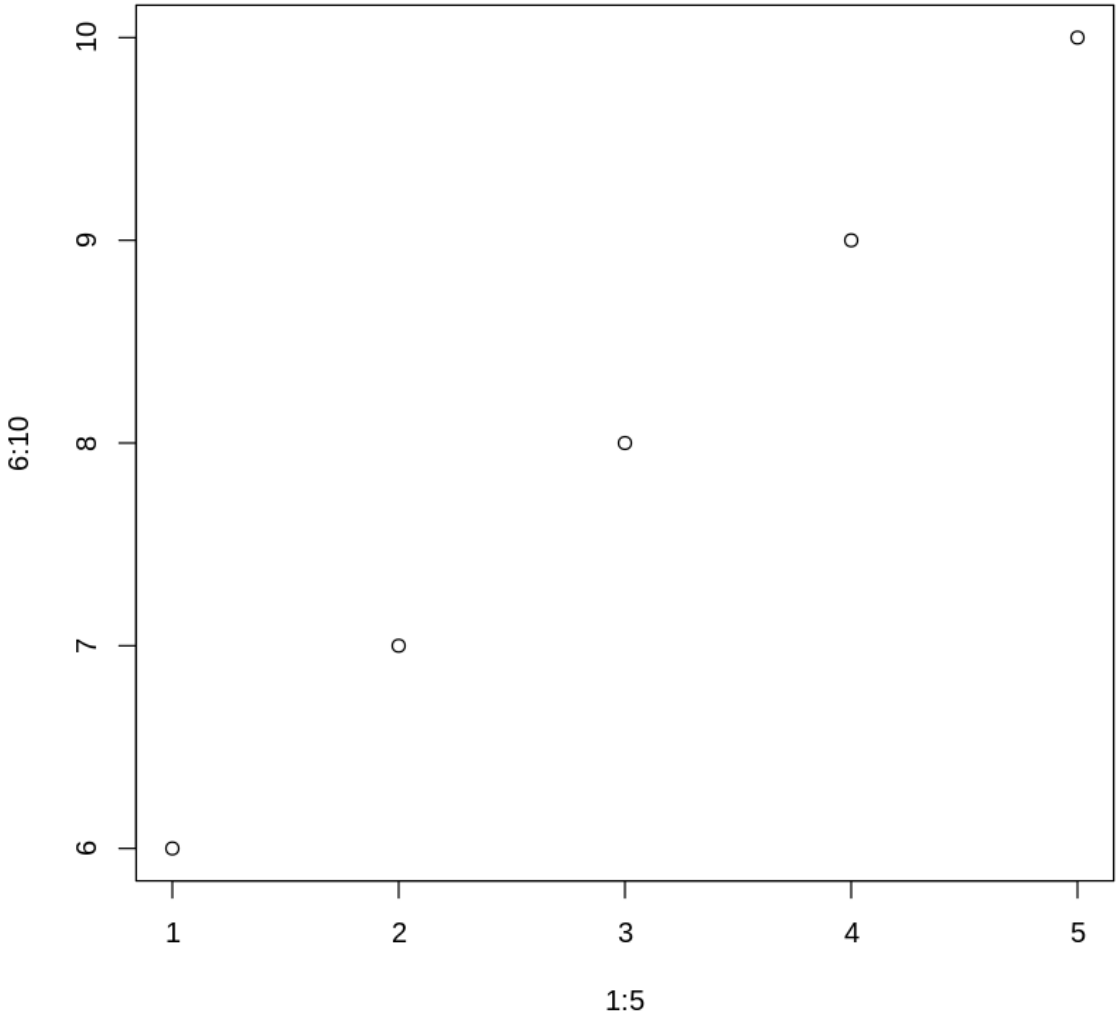

Figure 6.1: Covariance ellipses

6.2 Chapter 2

```
plot(1:5, 6:10)
```

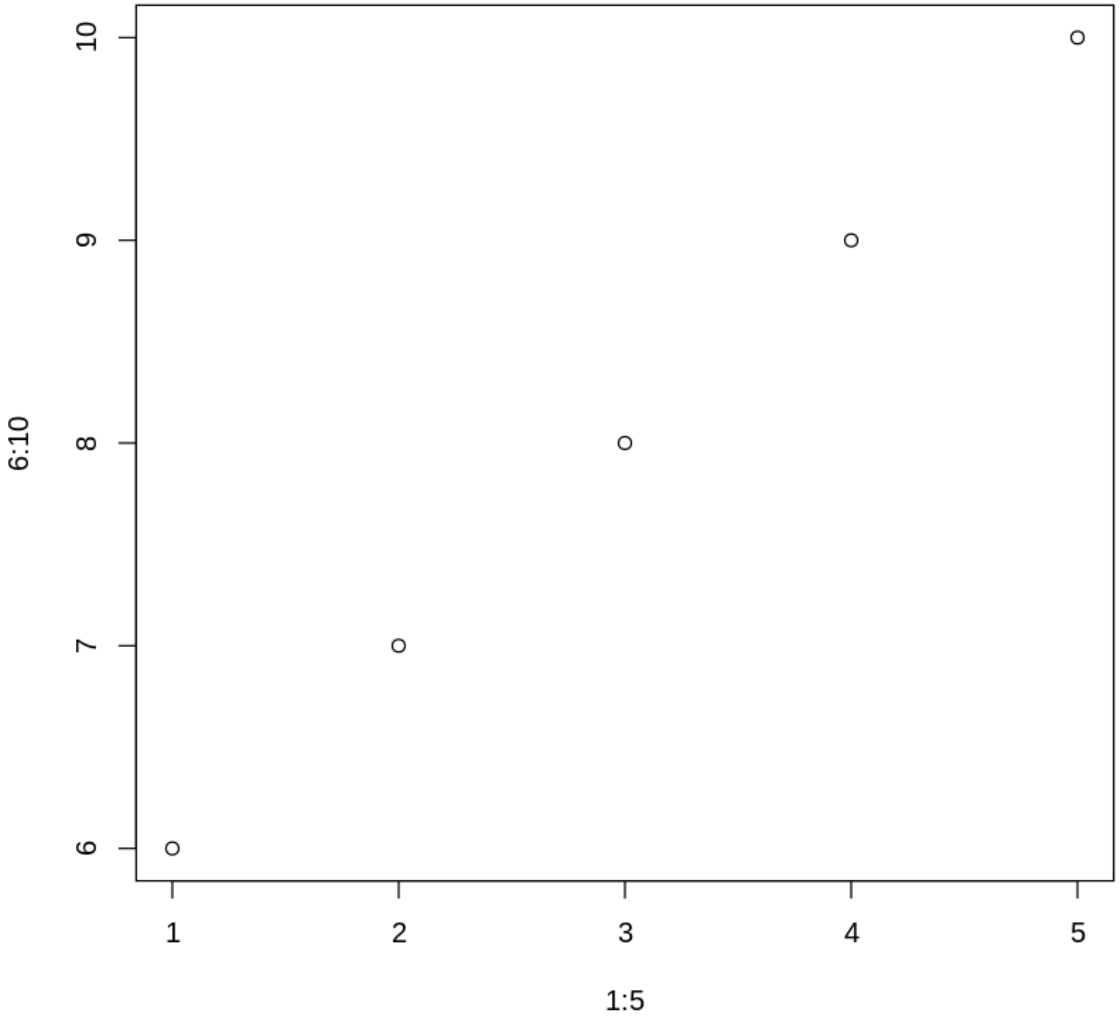

Figure 6.2: Covariance ellipses

6.3 Chapter 3

```
plot(1:5, 6:10)
```

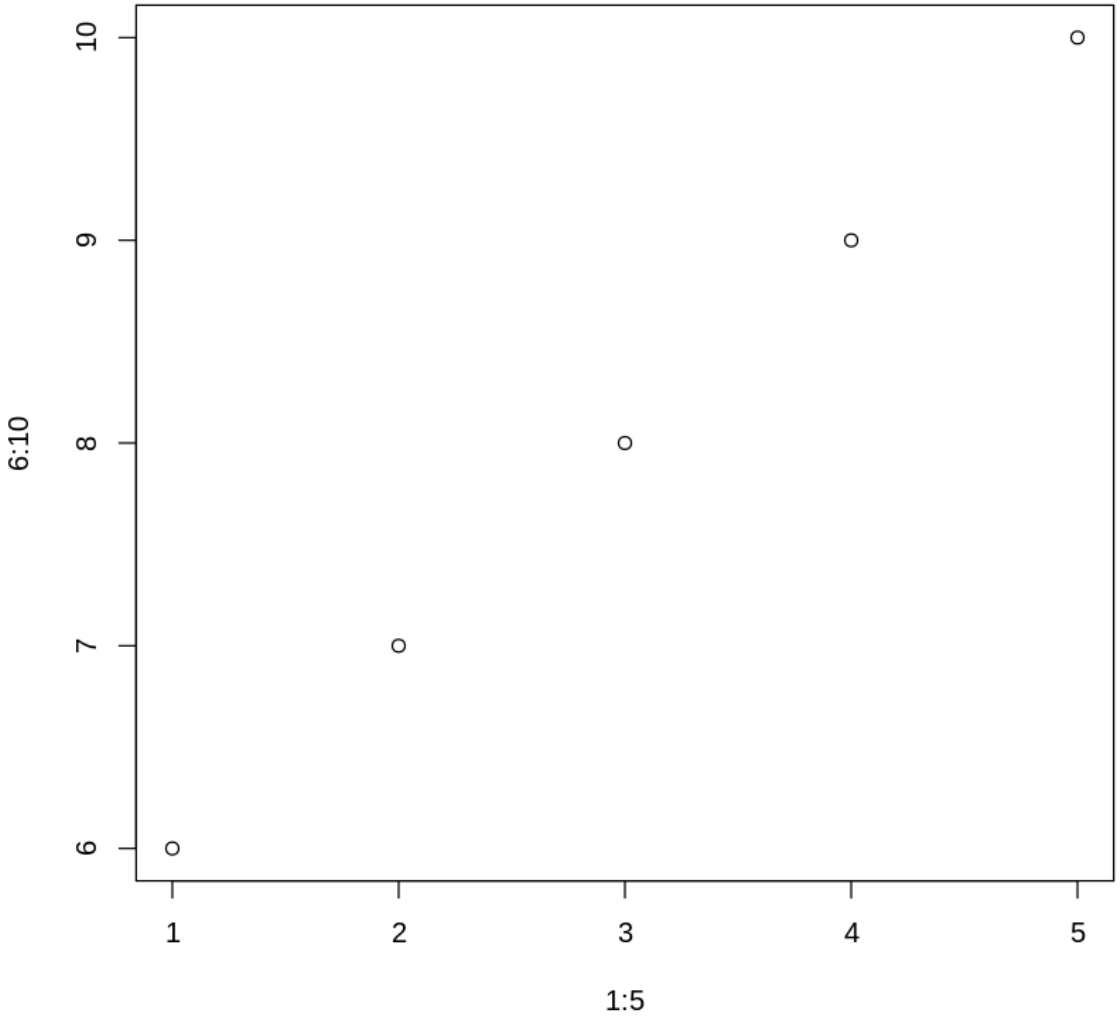

Figure 6.3: Covariance ellipses