Data Visualisation

Statistical Theory and Methods (HT19)

Nils Reimer

What is
data visualisation?

How do we
visualise data?

Why should we
visualise data?

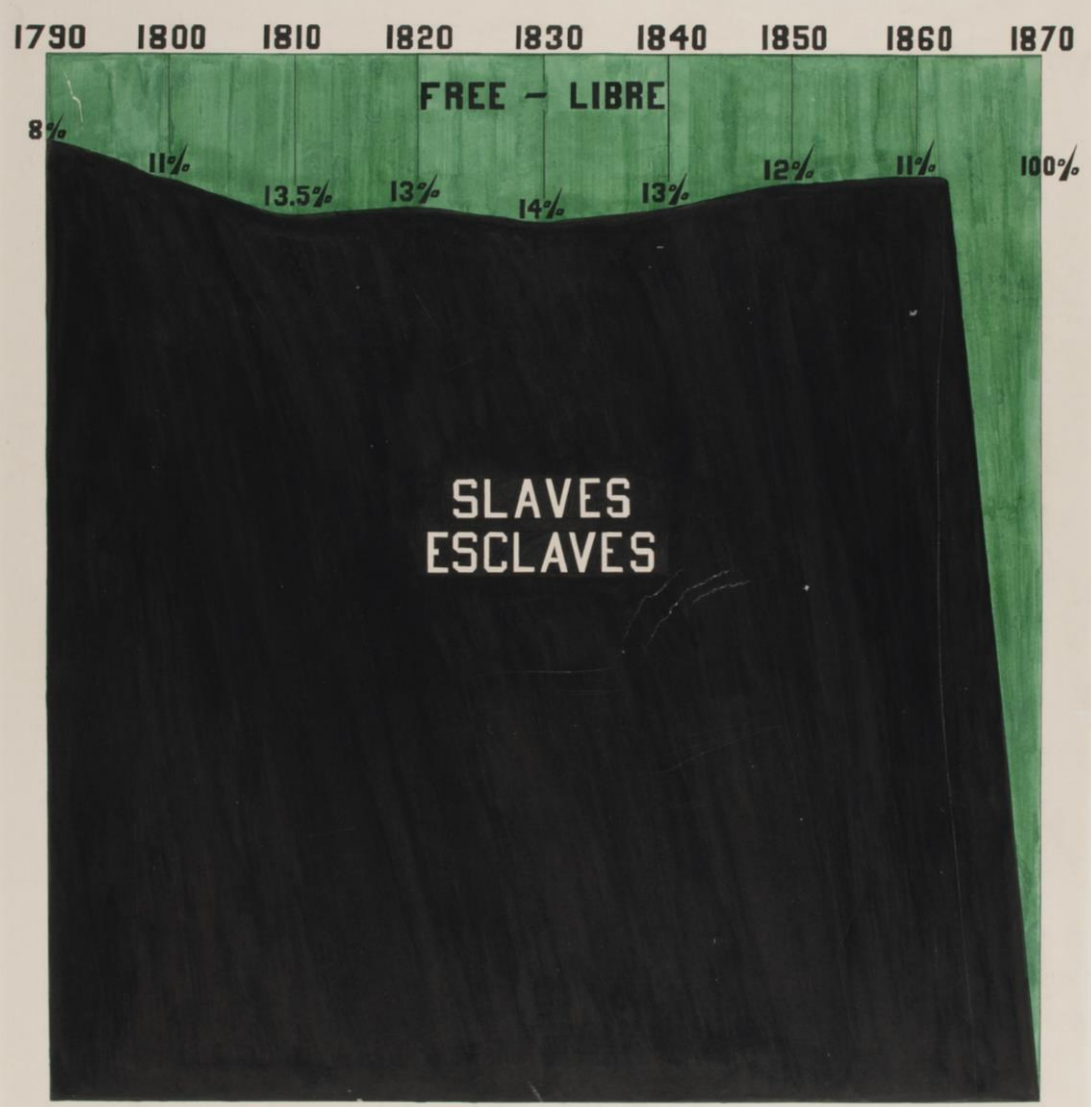
What makes a good
data visualisation?

What is data visualisation?

PROPORTION OF FREEMEN AND SLAVES AMONG AMERICAN NEGROES .

PROPORTION DES NÈGRES LIBRES ET DES ESCLAVES EN AMÉRIQUE .

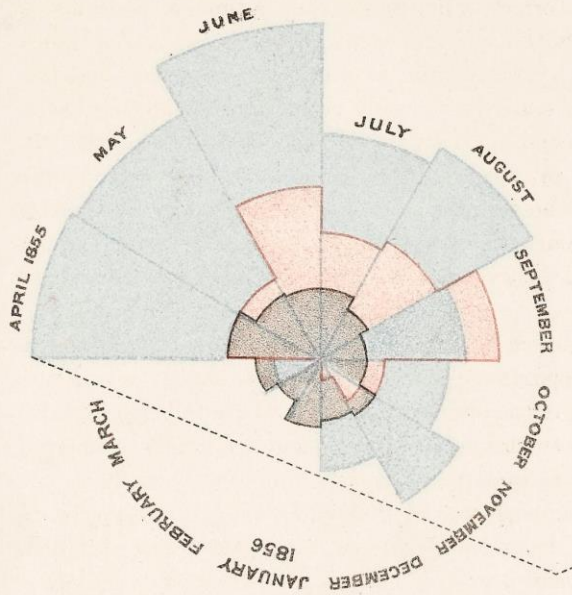
DONE BY ATLANTA UNIVERSITY .



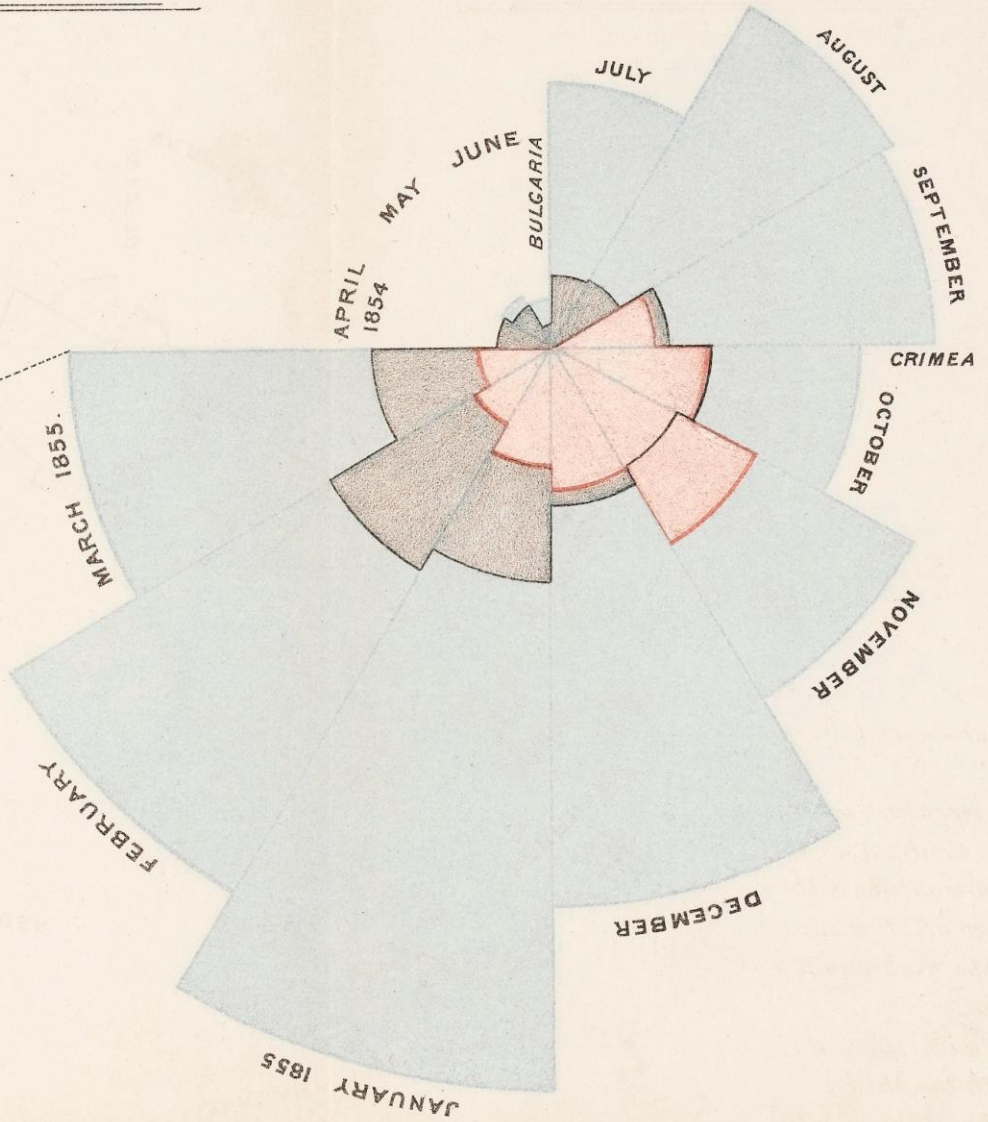
CITY AND RURAL POPULATION.
1890.

DIAGRAM of the CAUSES of MORTALITY IN THE ARMY IN THE EAST.

2.
APRIL 1855 to MARCH 1856.



1.
APRIL 1854 to MARCH 1855.



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

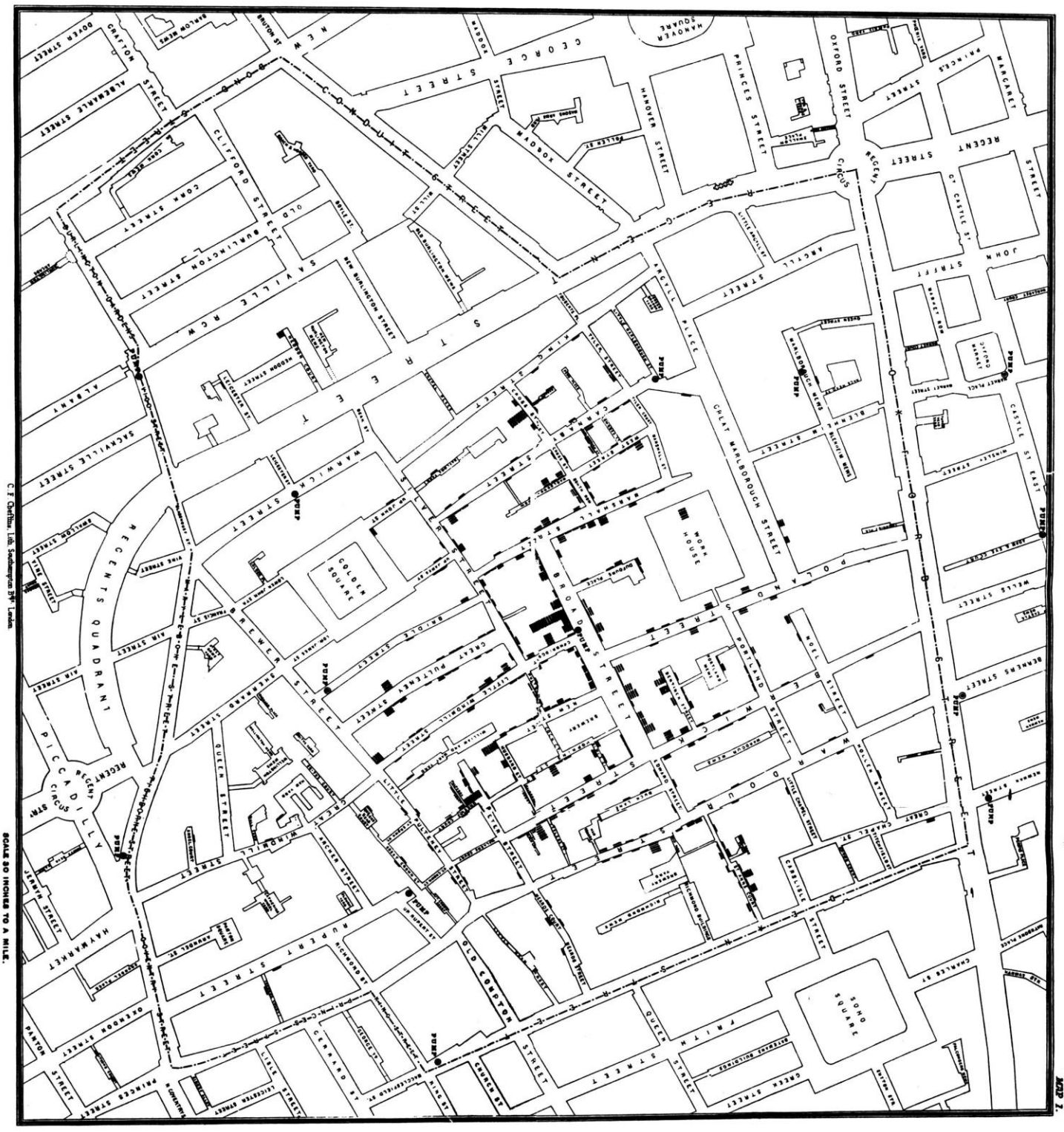
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Miligable Zymotic diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes.

The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.

In October 1854, & April 1855, the black area coincides with the red, in January & February 1855, the blue coincides with the black.

The entire areas may be compared by following the blue, the red & the black lines enclosing them.

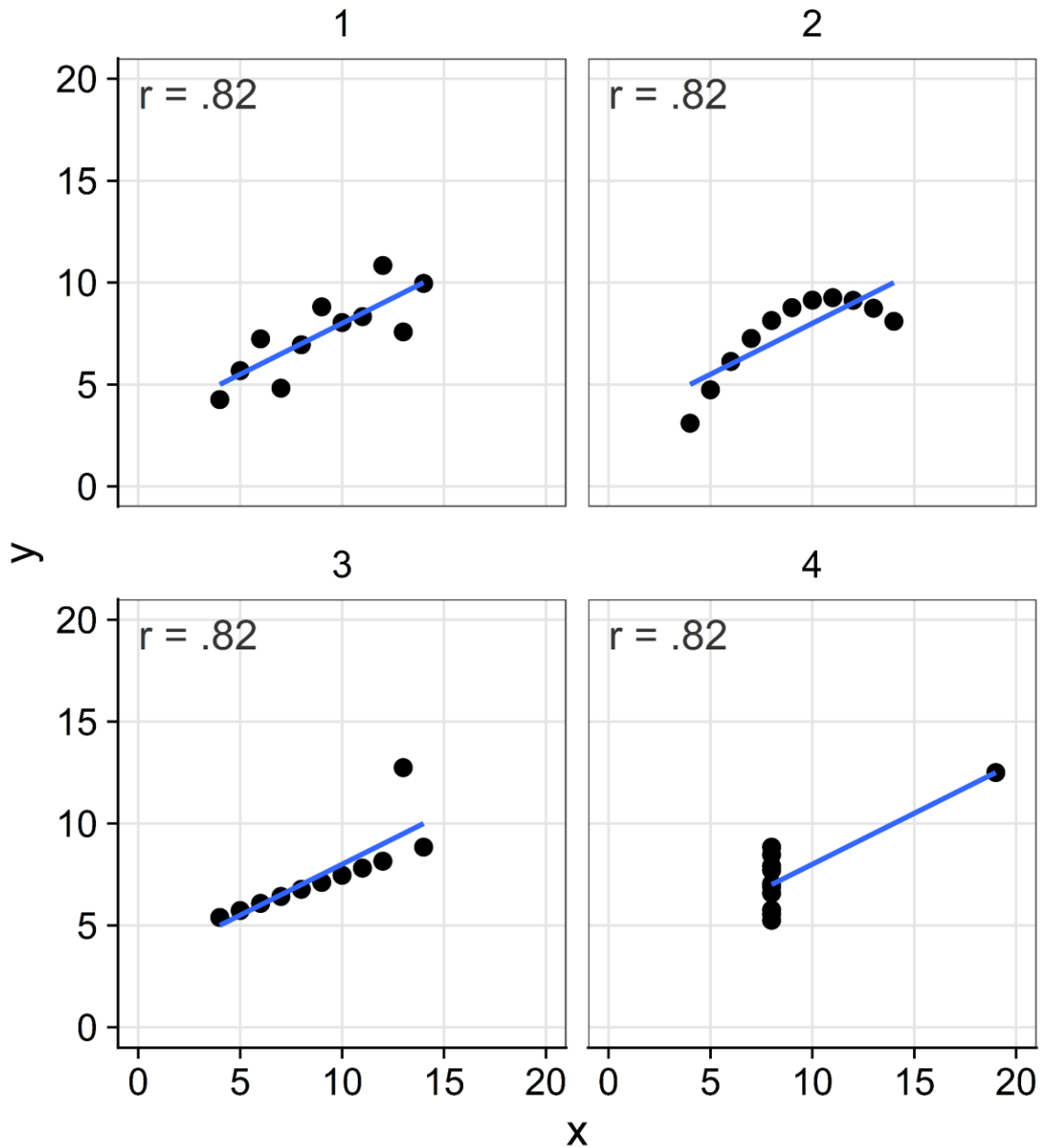
Harrison & Sons, St. Martin's Lane.



Why should we visualise data?

Anscombe's Quartet

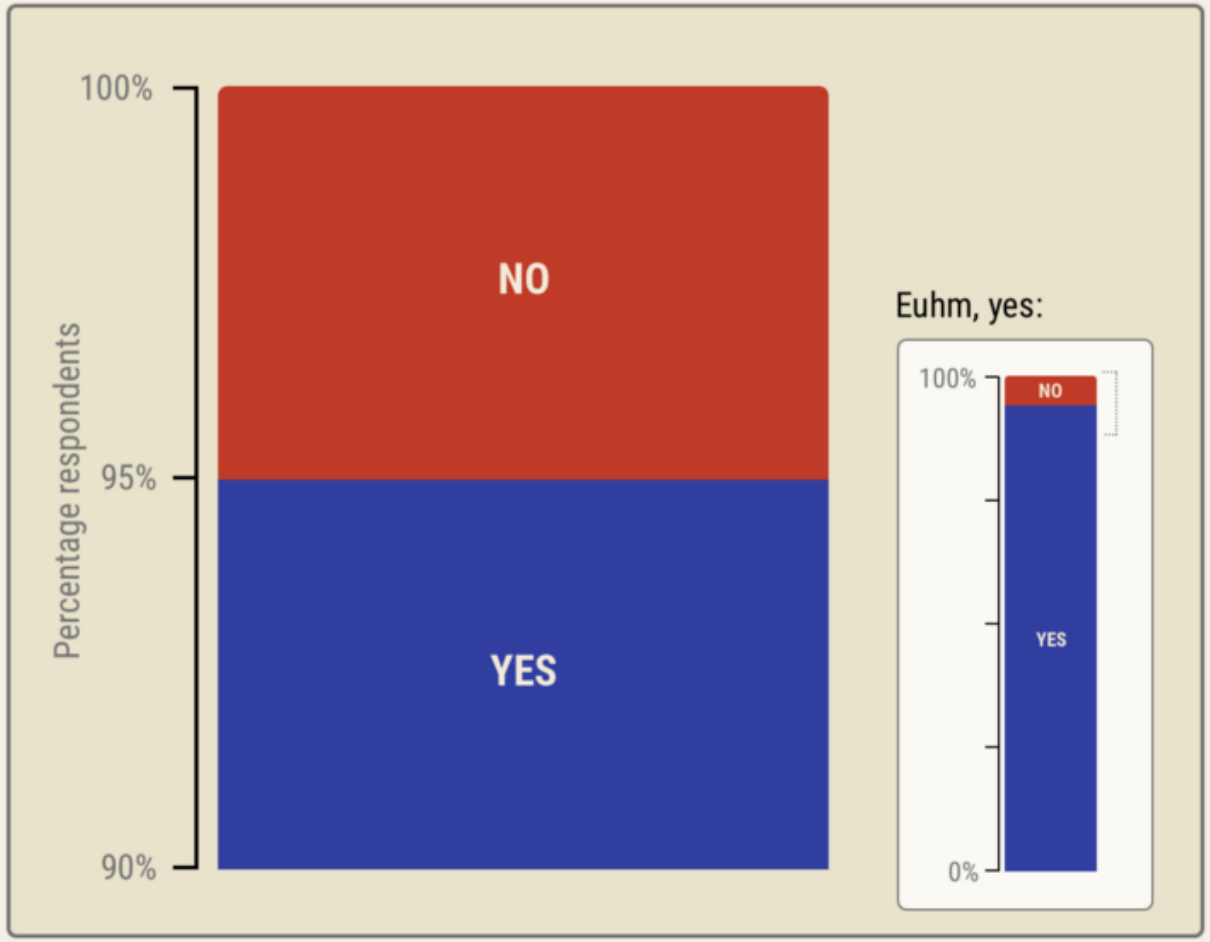
Four reasons why we should plot our data.

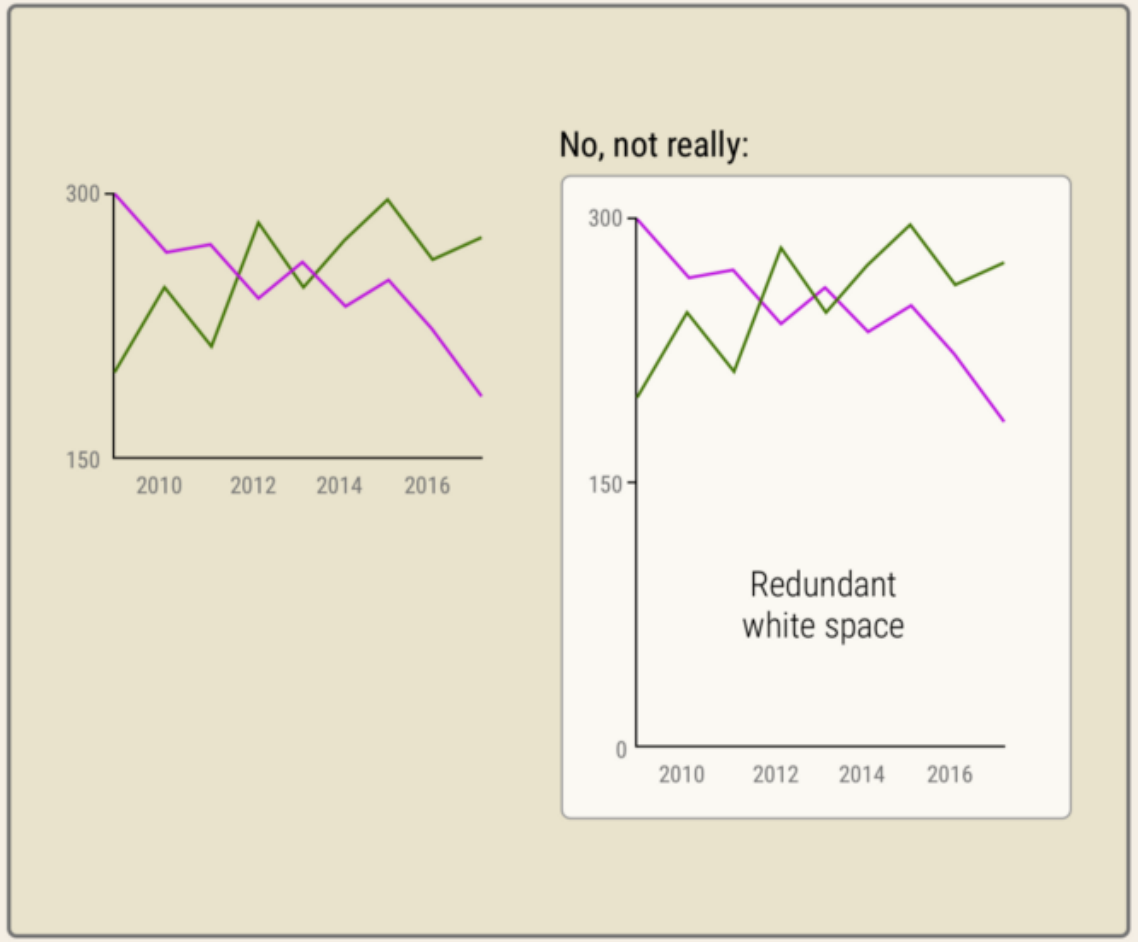


What makes a good data visualisation?

What makes a ~~good~~ bad data visualisation?

Should the y-axis in cases like these start at zero?





de Beus, T. (May 30, 2017). Retrieved from https://medium.com/tdebeus/data-visualisations-redesigned-for-the-better-75c707770d0e

TV staat bekend om slechte grafieken

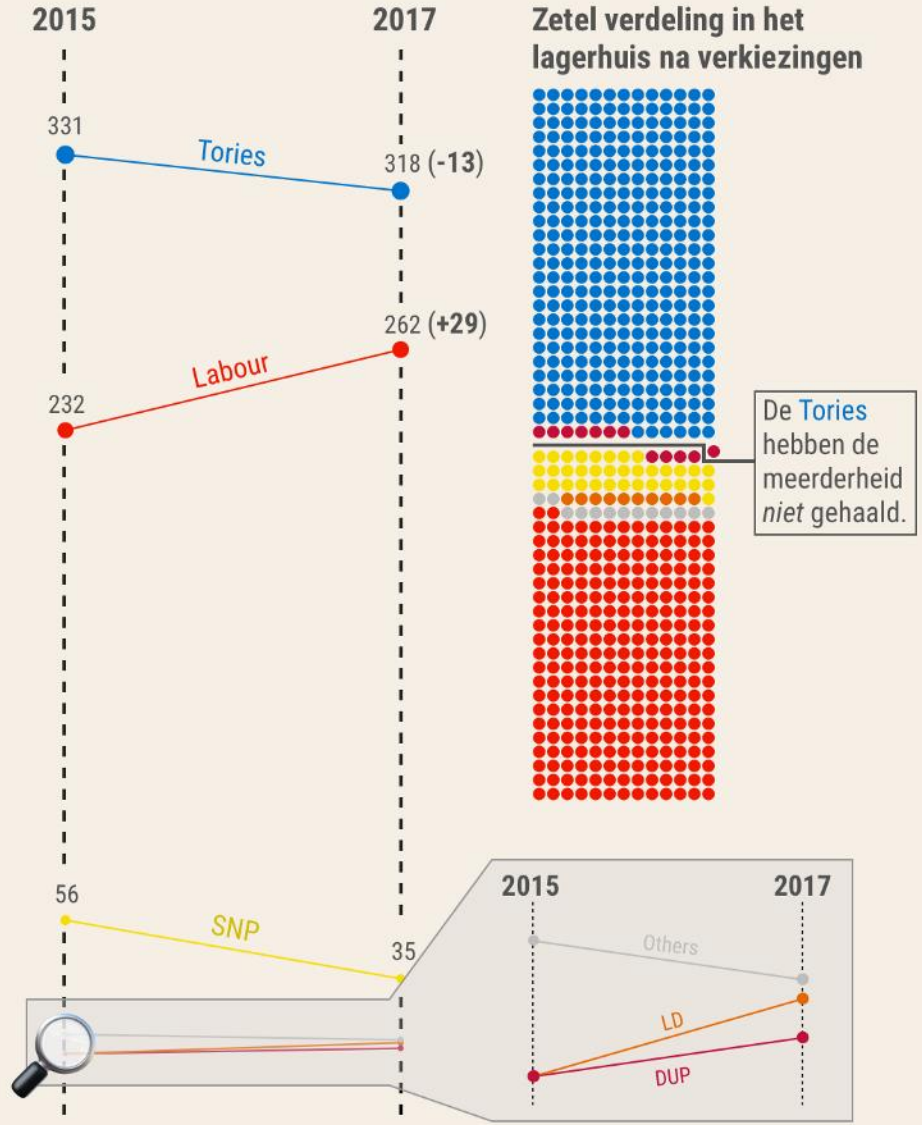
In de TV show Jinek vroeg zij aan één van de tafelgasten: “Hier zie je de exit-polls, wat is jouw eerste reactie op deze uitslag?” Wat gaat hier mis?

Ten eerste lijken tv-programmas een fetisj te hebben met 3D graphics. Omvangrijk research heeft bewezen dat deze heel moeilijk te lezen zijn. Ten tweede is de reactie moeilijker te geven als van de resultaten het verschil niet wordt laten zien van voor en na de verkiezingen.

Original

Redesign

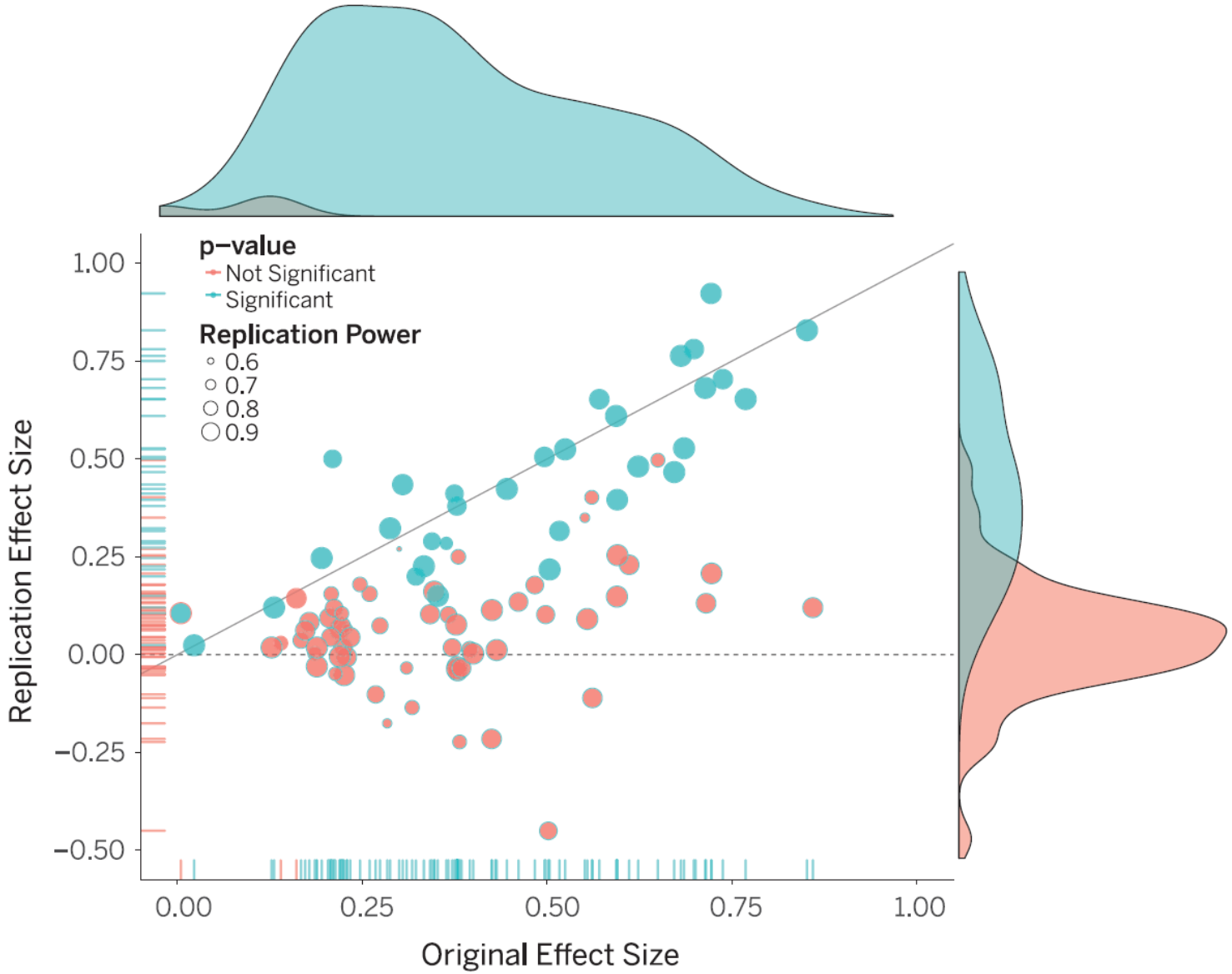
UK Verkiezingsuitslagen 2017 in Zetels



@TdeBeus

Data: BBC, *election results*. 9 juni 2017

de Beus, T. (2017, June 24). Retrieved from https://medium.com/tdebeus/tv-is-pre-eminently-the-medium-for-unreadable-3d-charts-9715c18d8edf



Original study effect size versus replication effect size (correlation coefficients). Diagonal line represents replication effect size equal to original effect size. Dotted line represents replication effect size of 0. Points below the dotted line were effects in the opposite direction of the original. Density plots are separated by significant (blue) and nonsignificant (red) effects.

How do we visualise data?

Variables:

Chart preview uses example data

Treatment
Cholesterol Concentrat...
Diet
Exercise

Drag a Gallery chart here to use it as your starting point

OR

Click on the Basic Elements tab to build a chart element by element

TYPOLOGY

Gallery | Basic Elements | Groups/Point ID | Titles/Footnotes

Choose from:

Favorites
Bar
Line
Area
Pie/Polar
Scatter/Dot
Histogram
High-Low
Boxplot
Dual Axes

Element Properties...

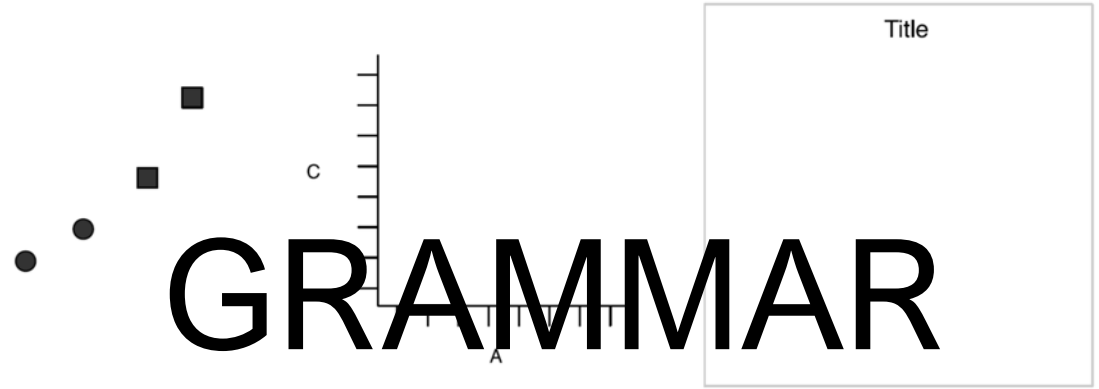
Options...

OK | Paste | Reset | Cancel | Help

A Layered Grammar of Graphics

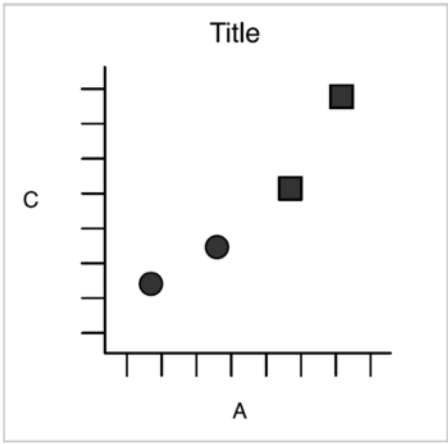
Hadley WICKHAM

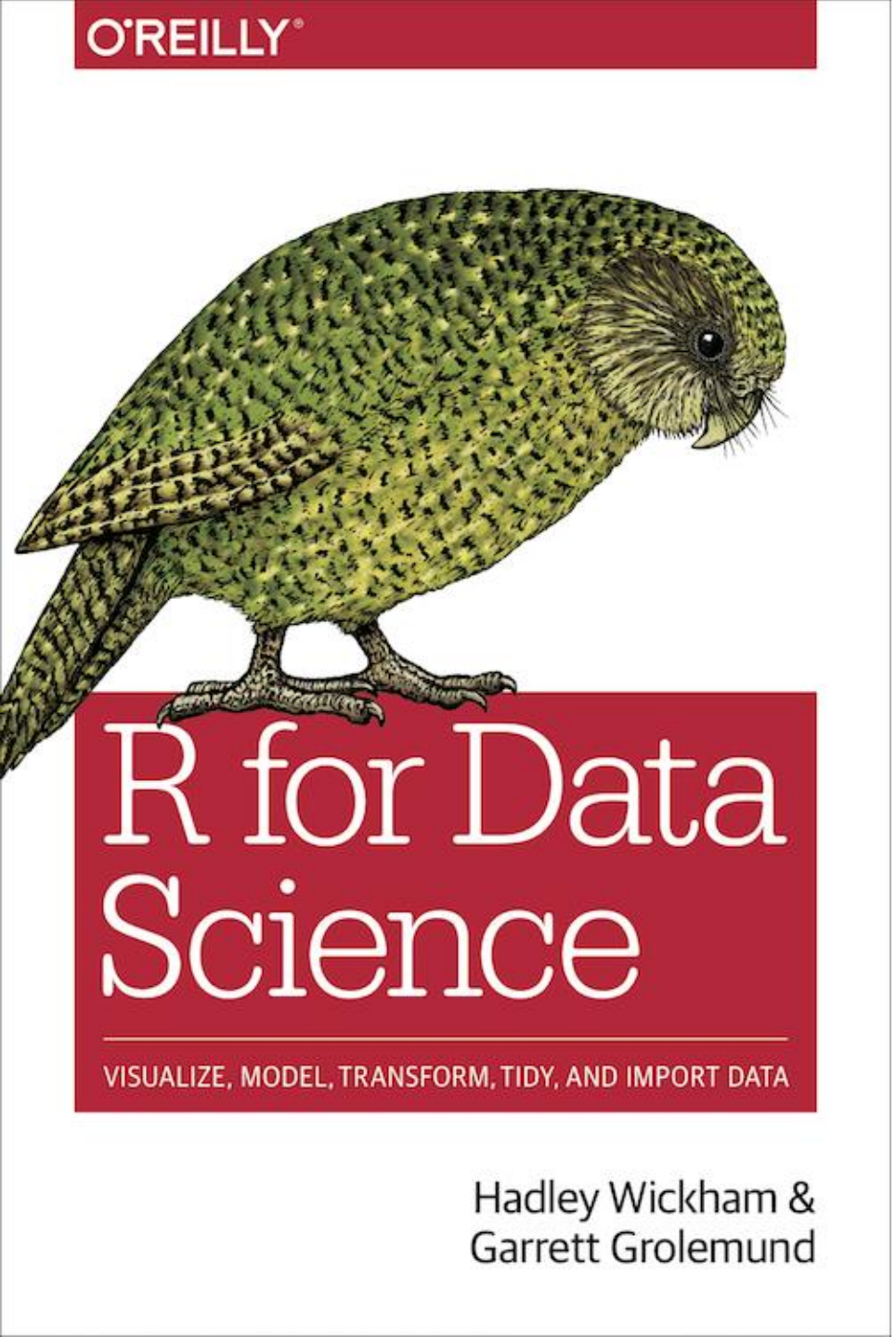
A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the "scatterplot") and gain insight into the deep structure that underlies statistical graphics. This article builds on Wilkinson, Anand, and Grossman (2005), describing extensions and refinements developed while building an open source implementation of the grammar of graphics for R, `ggplot2`.



GRAMMAR

Figure 1. Graphics objects produced by (from left to right): geometric objects, scales and coordinate system, plot annotations.





The tidyverse

Components

The tidyverse is a collection of R packages that share common philosophies and are designed to work together. This site is a work-in-progress guide to the tidyverse and its packages.

If you are new to the tidyverse, the best place to learn the complete philsophy and how everything fits together is the R for data science book. This book is available for free online, and can you order a physical copy from Amazon (currently taking pre-orders, the book should be out by the end of the year).

```
install.packages("tidyverse")
```

```
library(tidyverse)
```

`ggplot()` initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

```
ggplot(data, mapping = aes(x, y))
```

Default list of aesthetic mappings to use for plot. If not specified, must be suppled in each layer added to the plot.

Default dataset to use for plot. If not already a `data.frame`, will be converted to one by `fortify`. If not specified, must be suppled in each layer added to the plot.

```r
# Example 1: A First Plot ------------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")
```

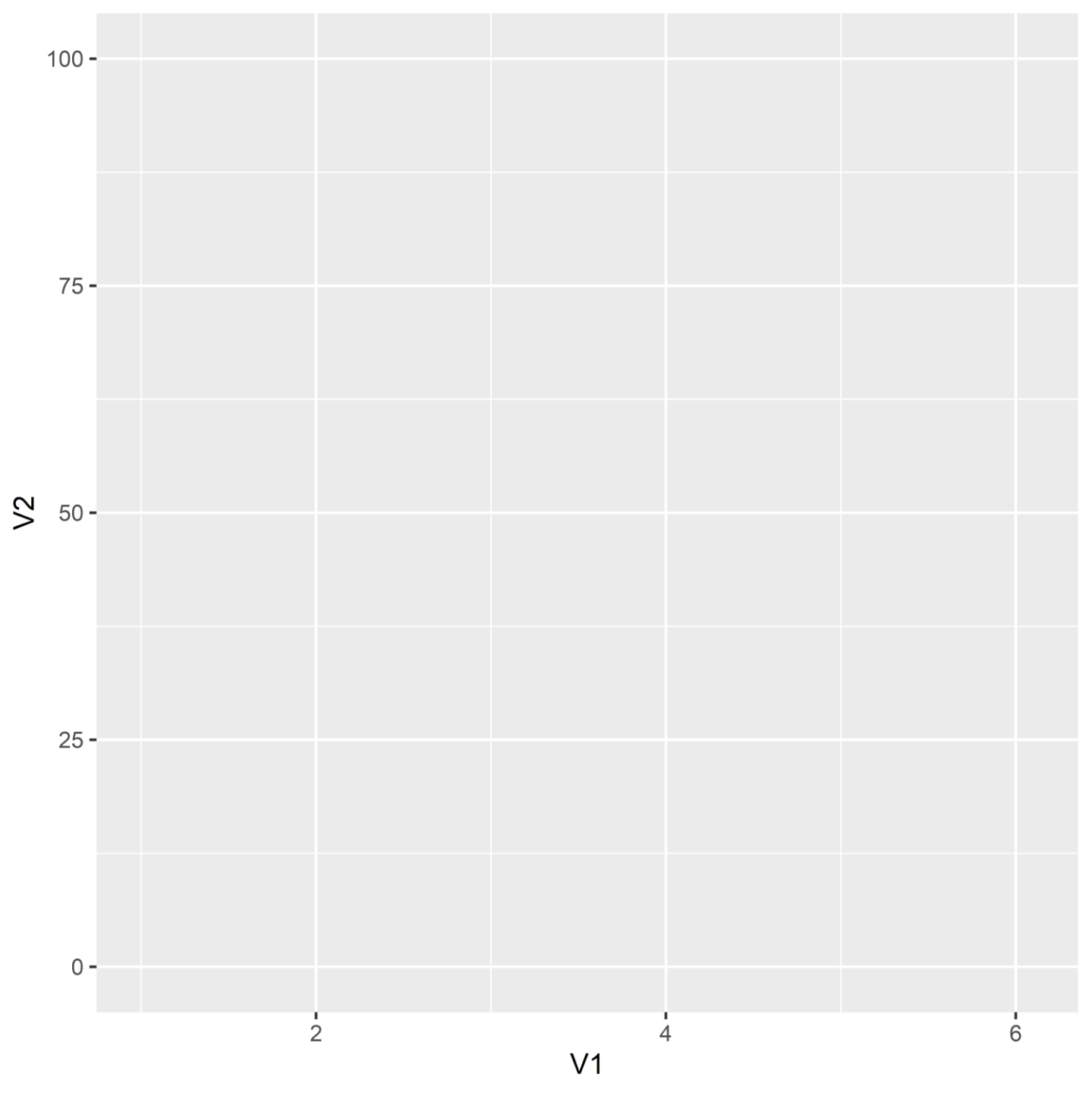
```
> print(d1, n = 5)
# A tibble: 161 x 4
     V1    V2           V3       V4
  <dbl> <int>         <chr>    <chr>
1   3.7    59 Experimental Group 1
2   3.4    45      Control Group 2
3   3.5    49      Control Group 1
4   2.8    48 Experimental Group 2
5   4.2    90 Experimental Group 2
# ... with 156 more rows
```

```
> count(d1, V3, V4)
# A tibble: 4 x 3
             V3       V4     n
          <chr>    <chr> <int>
1       Control Group 1    29
2       Control Group 2    51
3  Experimental Group 1    57
4  Experimental Group 2    24
```

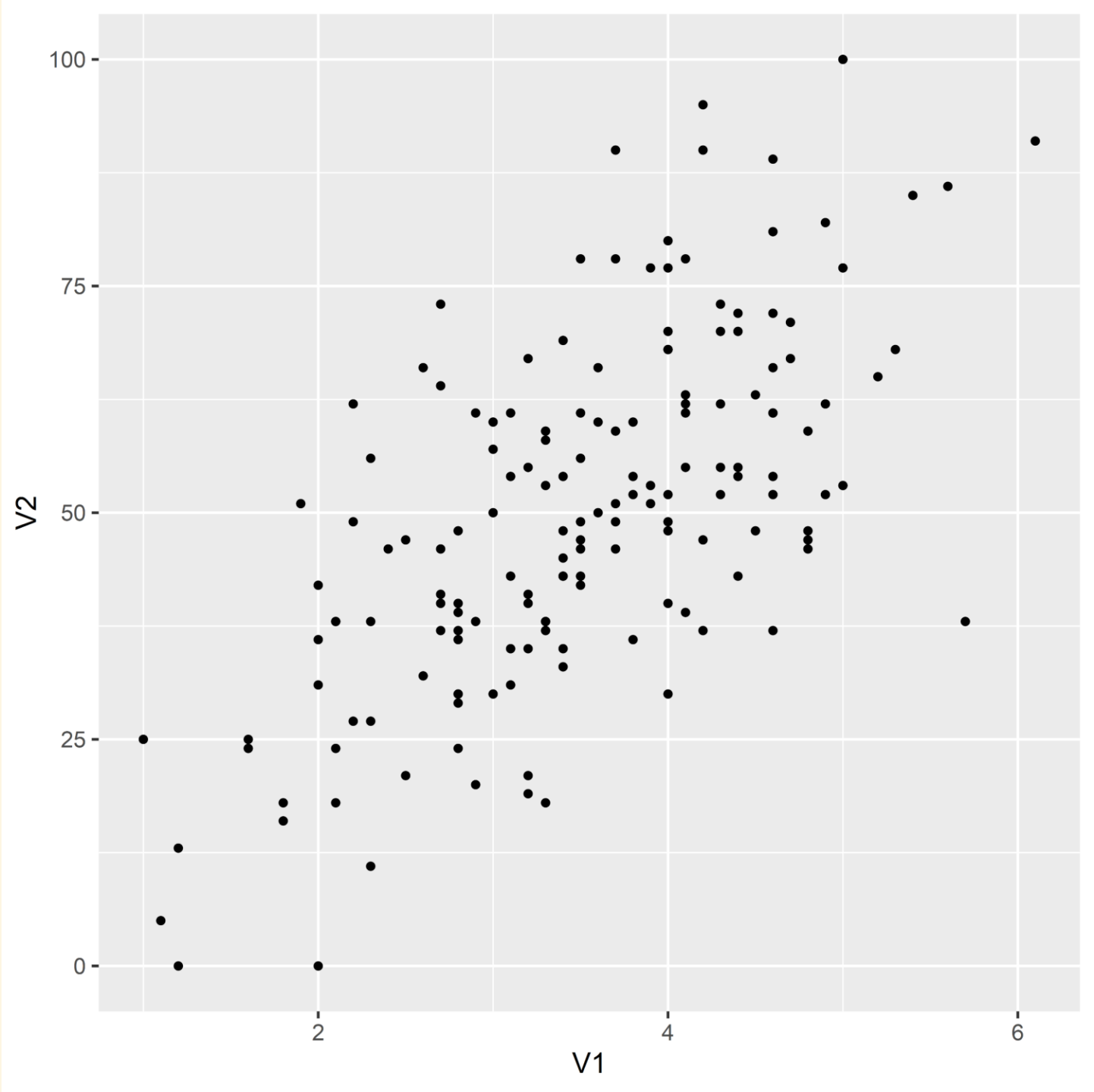
```r
# Example 1: A First Plot ------------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, mapping = aes(x = V1, y = V2))
```

```r
# Example 1: A First Plot ------------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point()
```

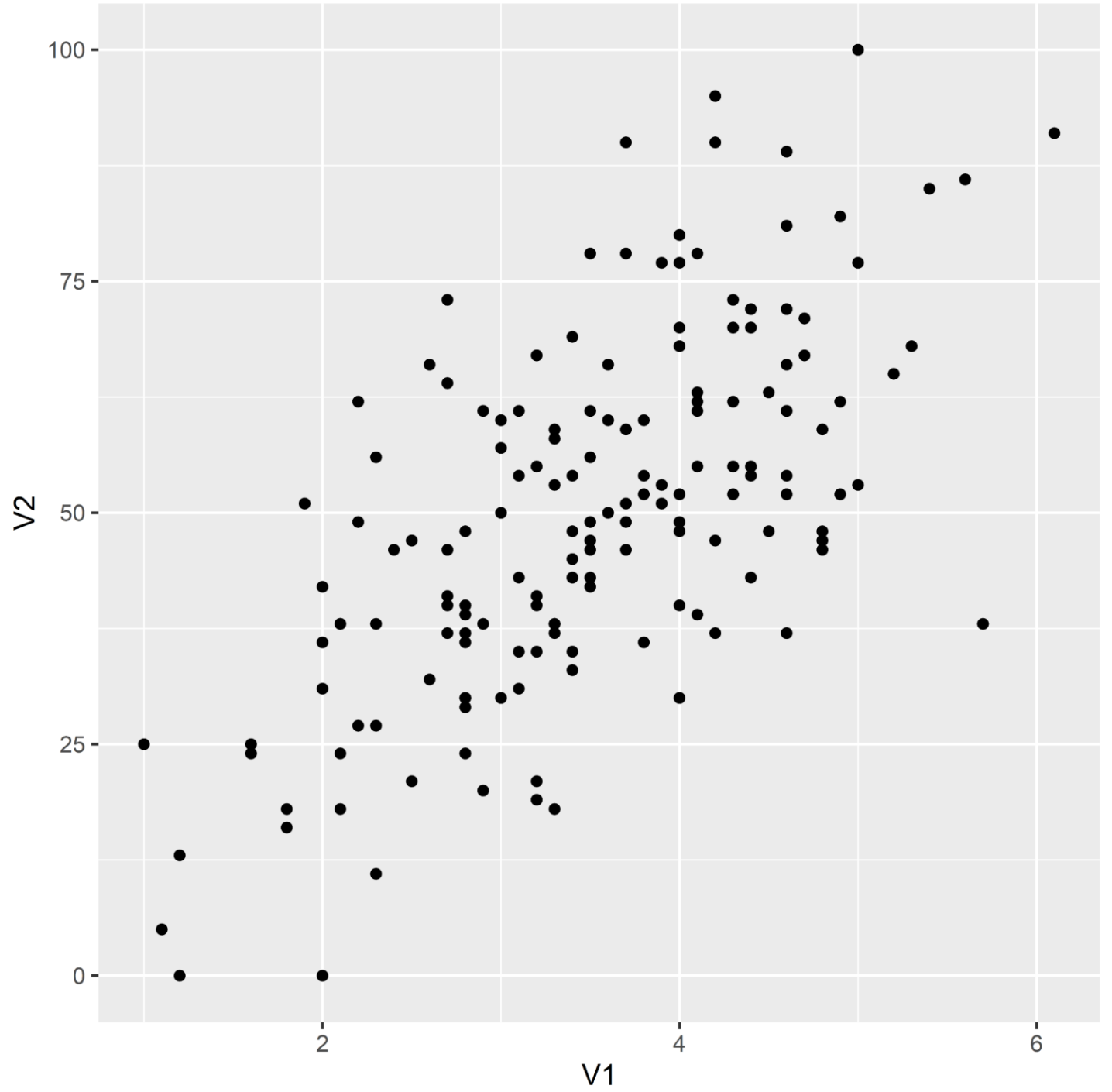
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2))
```



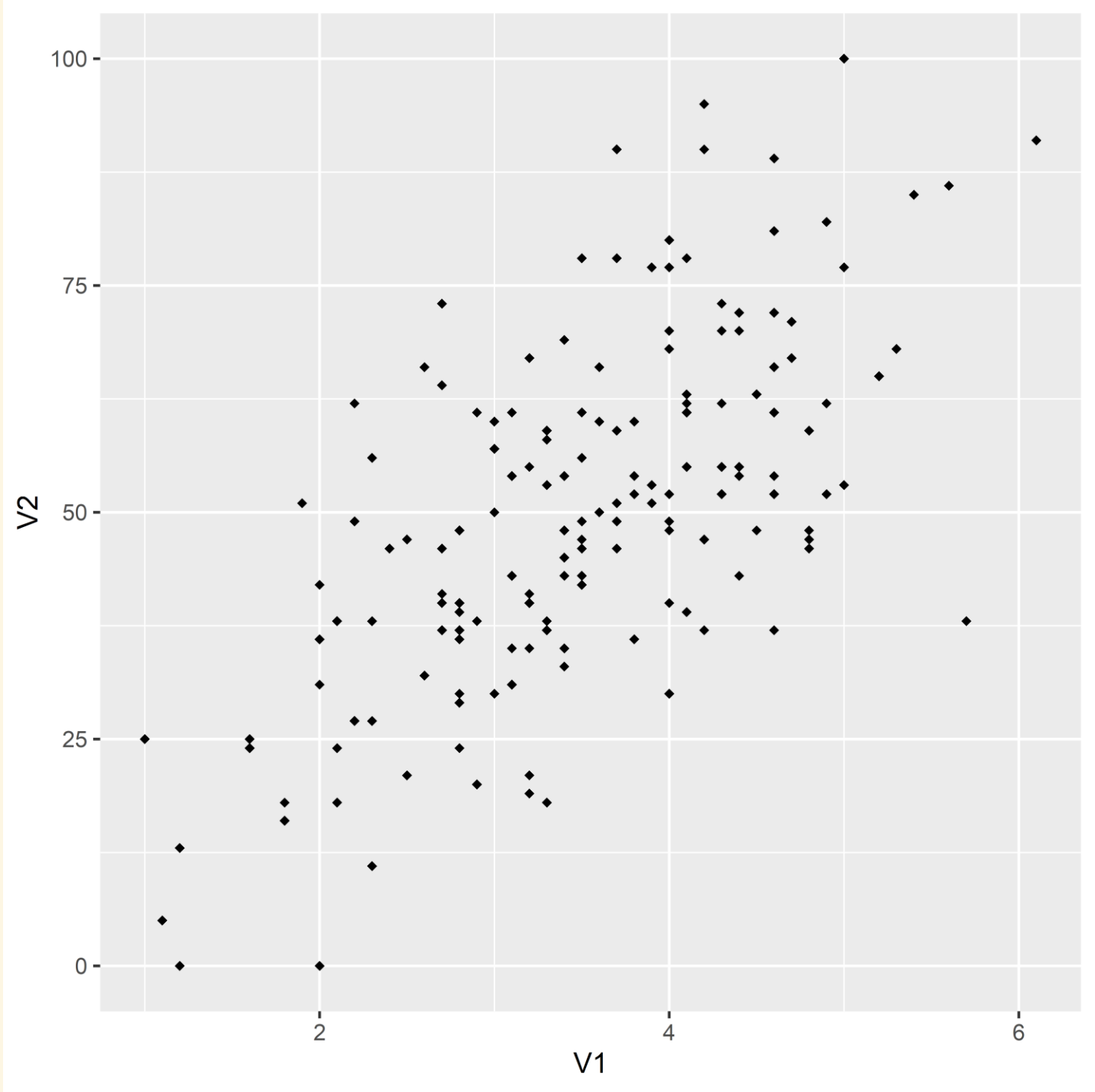
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point()
```



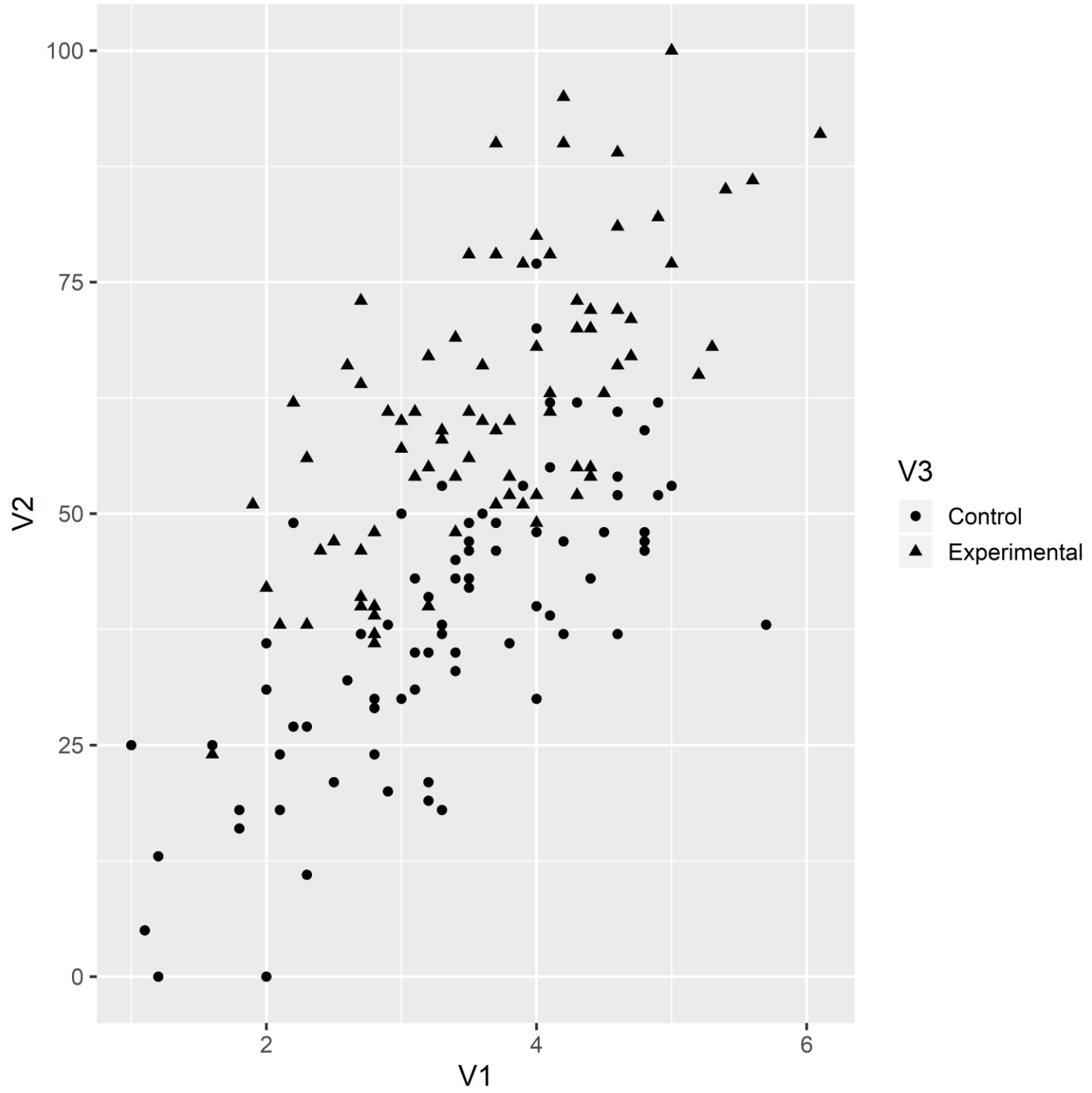
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(size = 2)
```



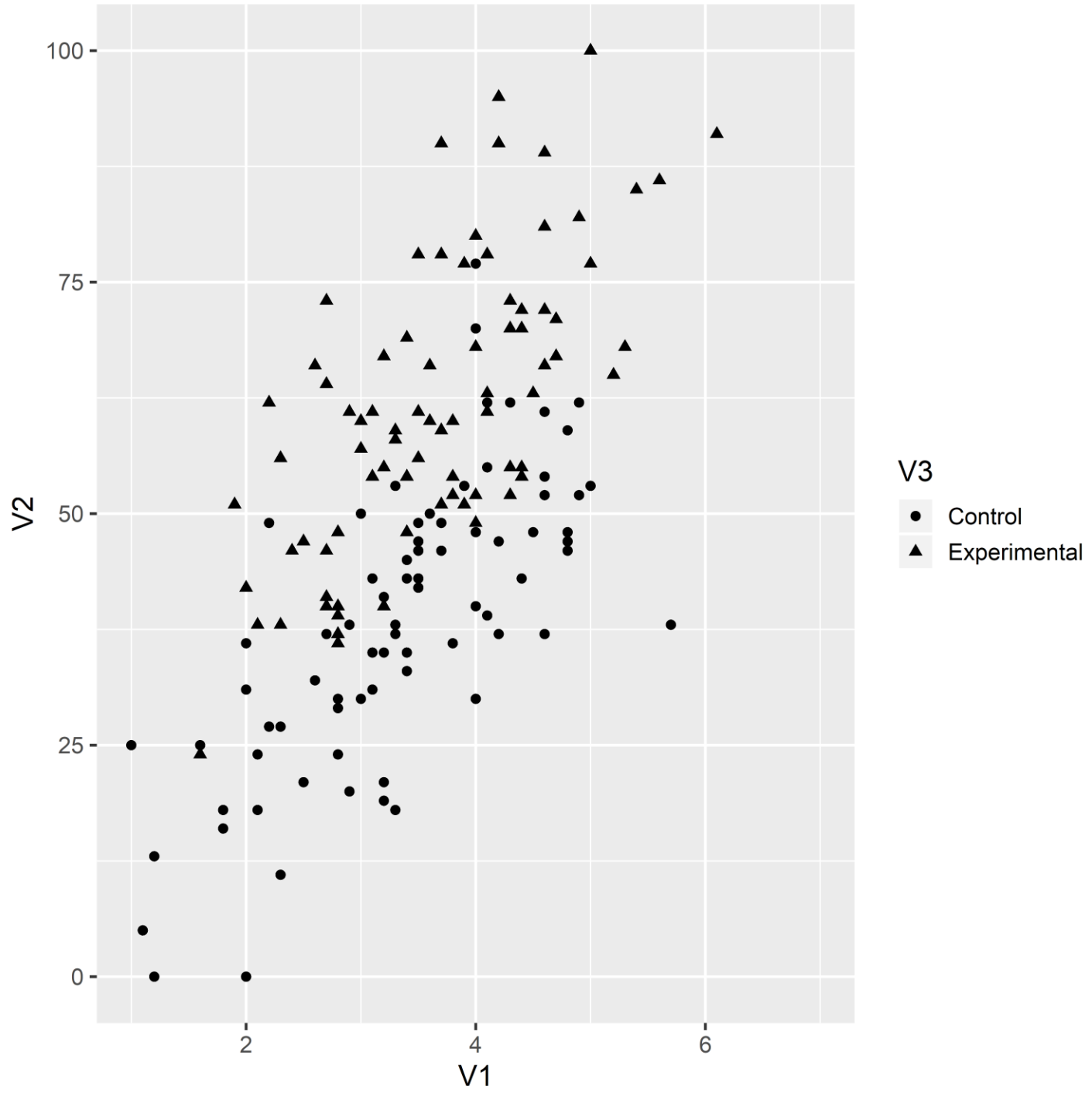
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(shape = "diamond", size = 2)
```



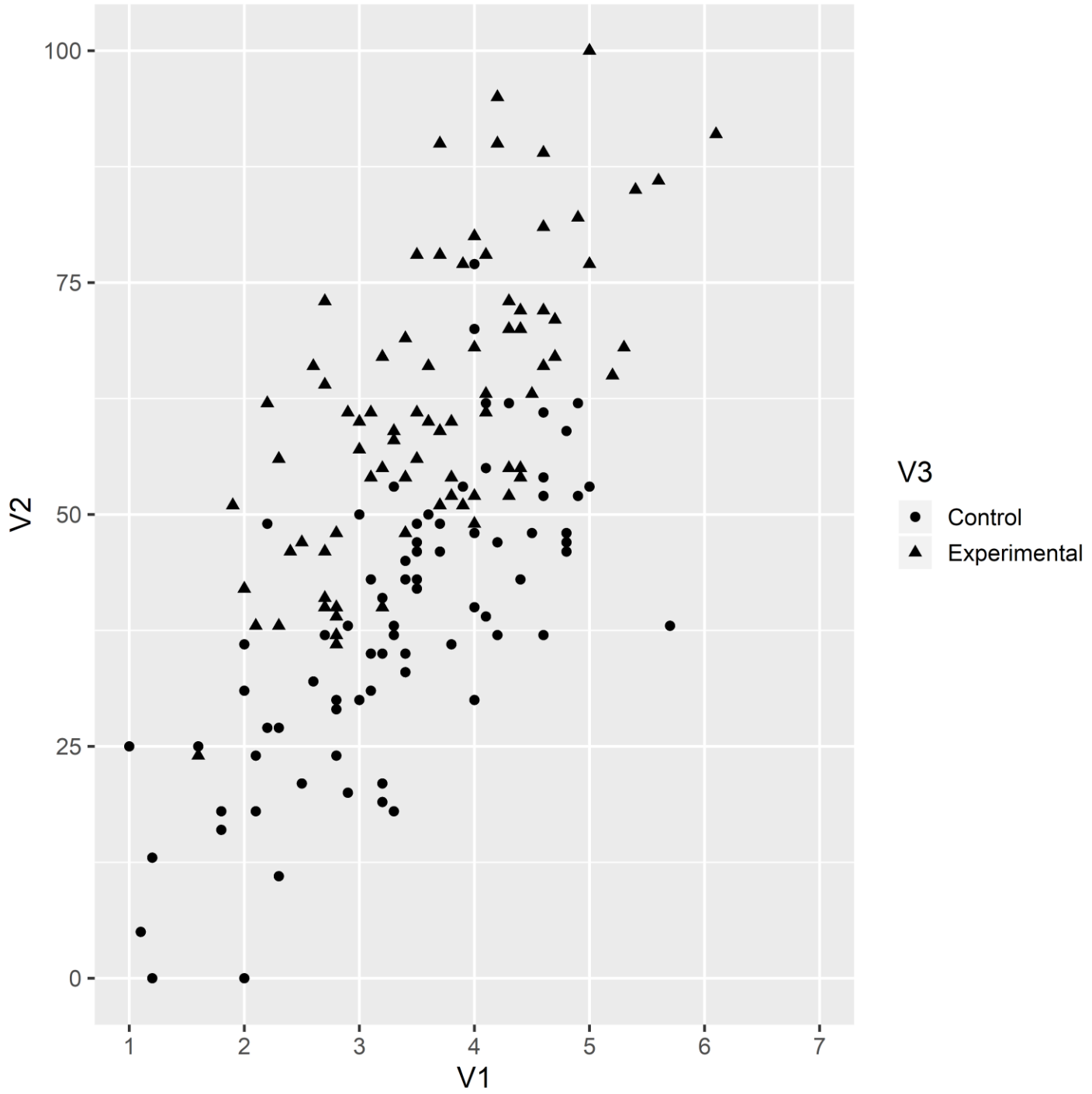
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(aes(shape = V3), size = 2)
```



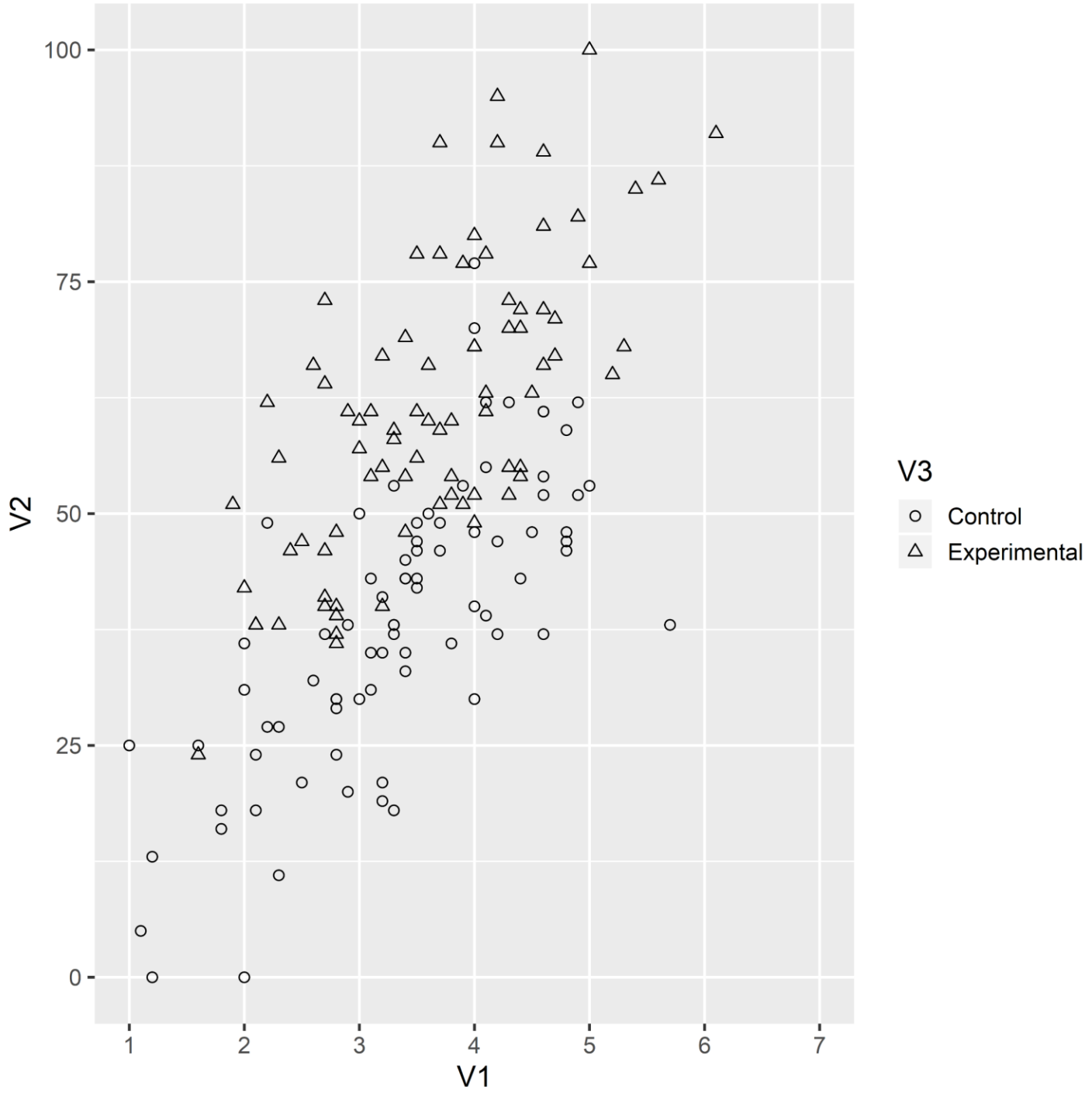
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(aes(shape = V3), size = 2) +
    scale_x_continuous(limits = c(1, 7)) +
    scale_y_continuous(limits = c(0, 100))
```



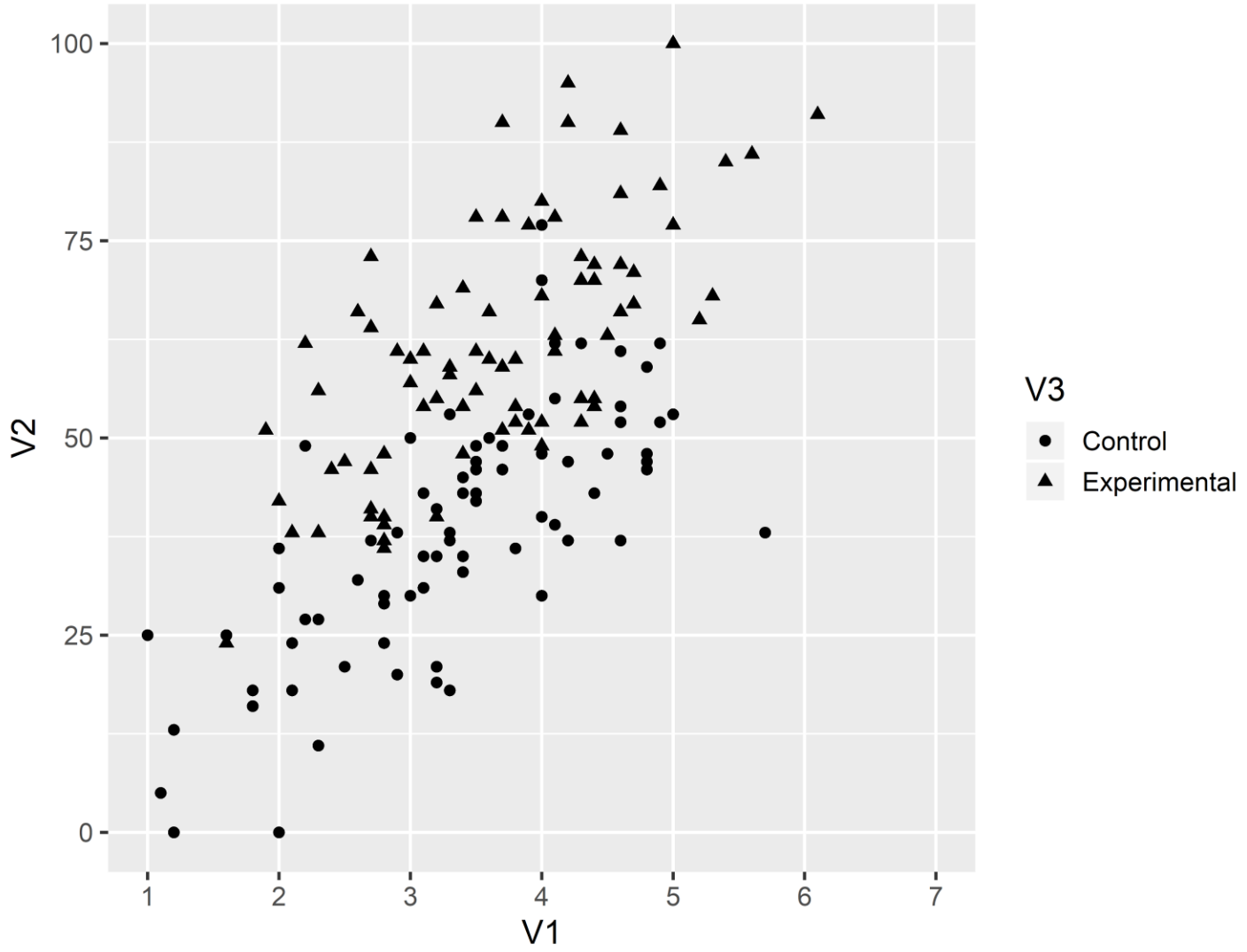
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(aes(shape = V3), size = 2) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100))
```



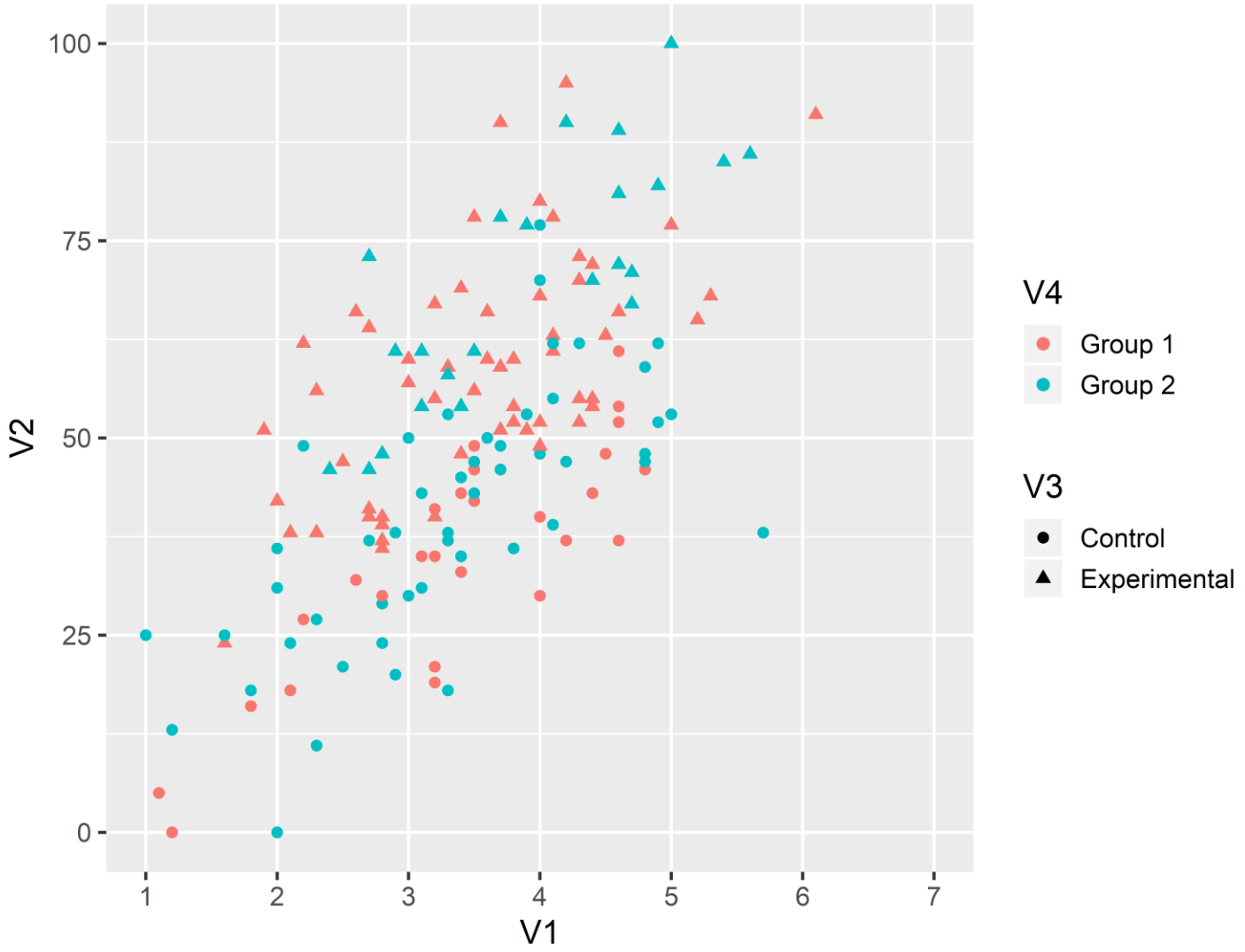
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(aes(shape = V3), size = 2) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    scale_shape_discrete(solid = FALSE)
```



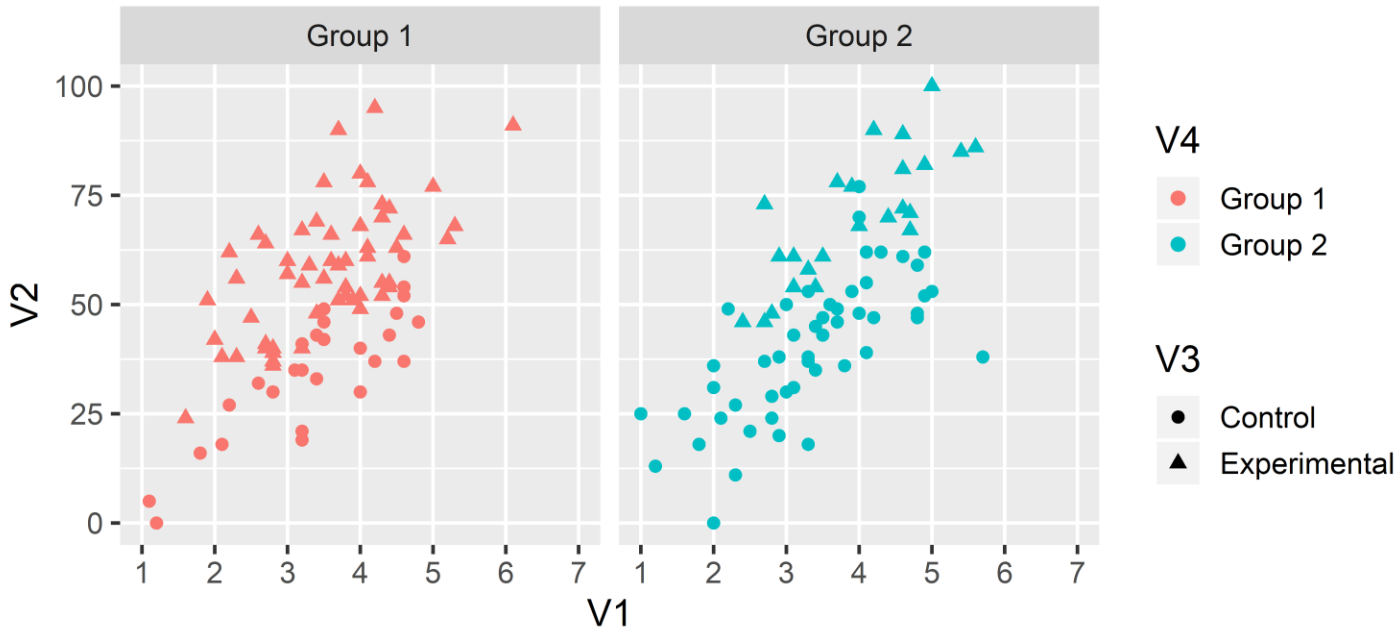
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(aes(shape = V3), size = 2) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100)
```



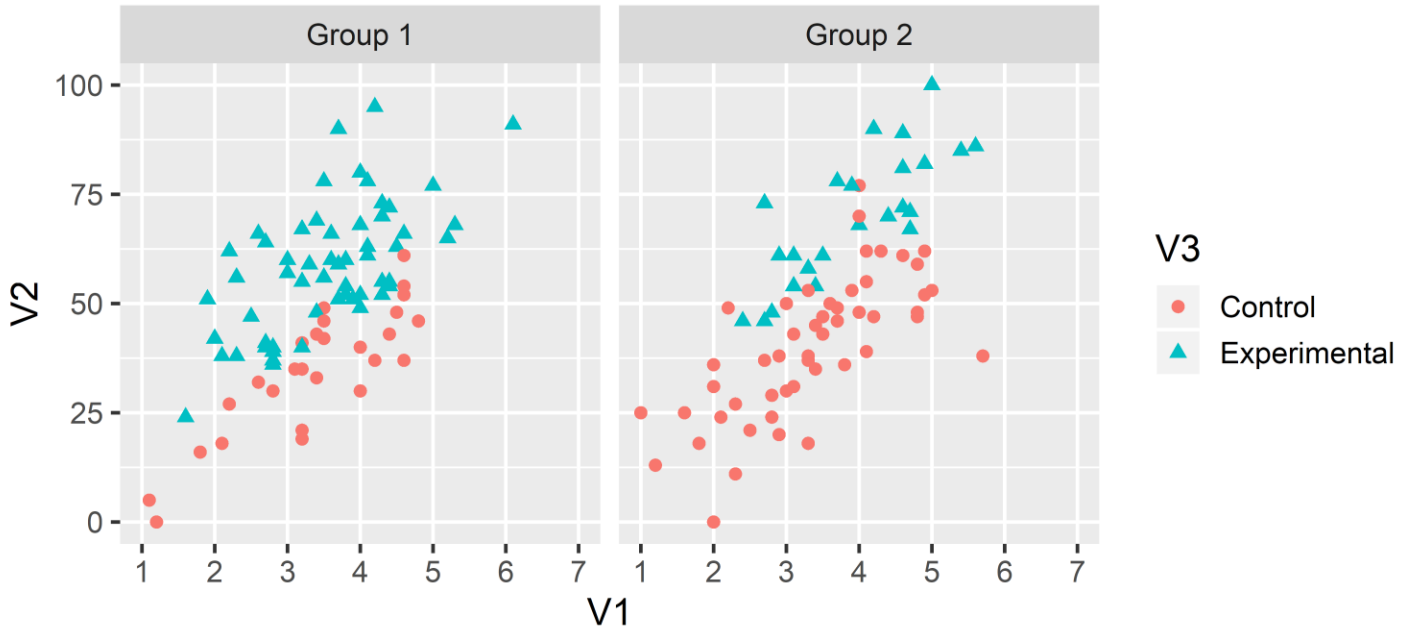
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100)
```



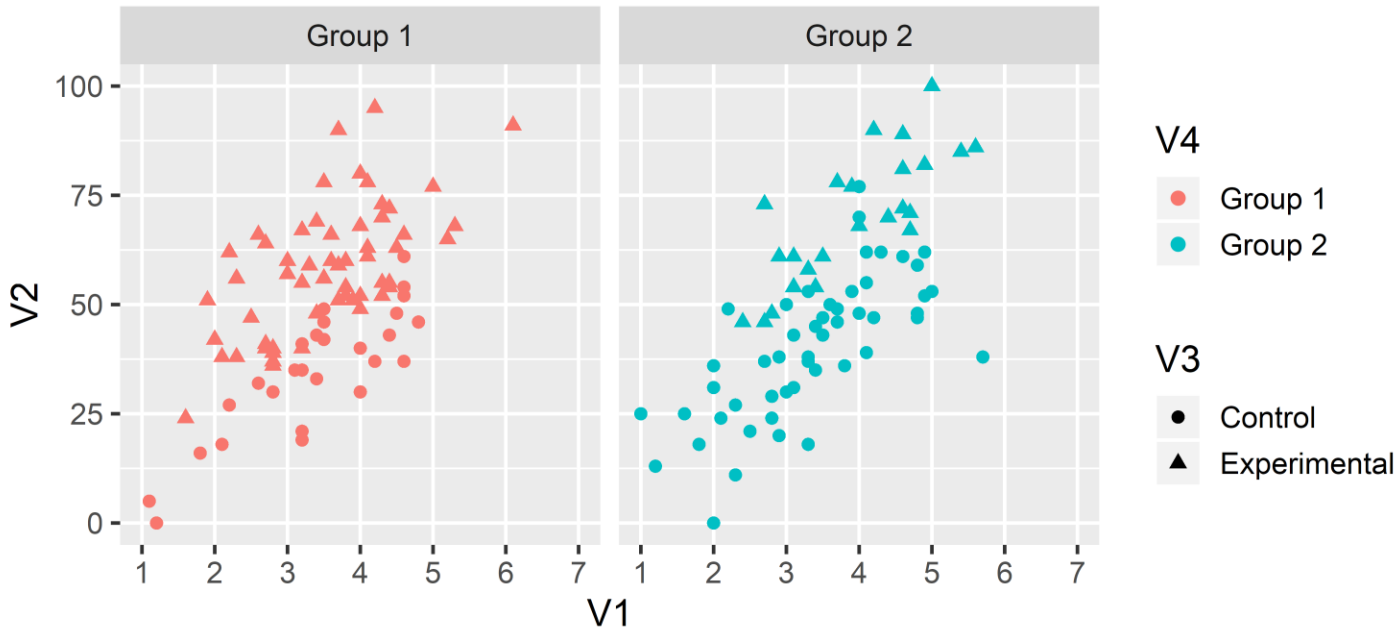
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



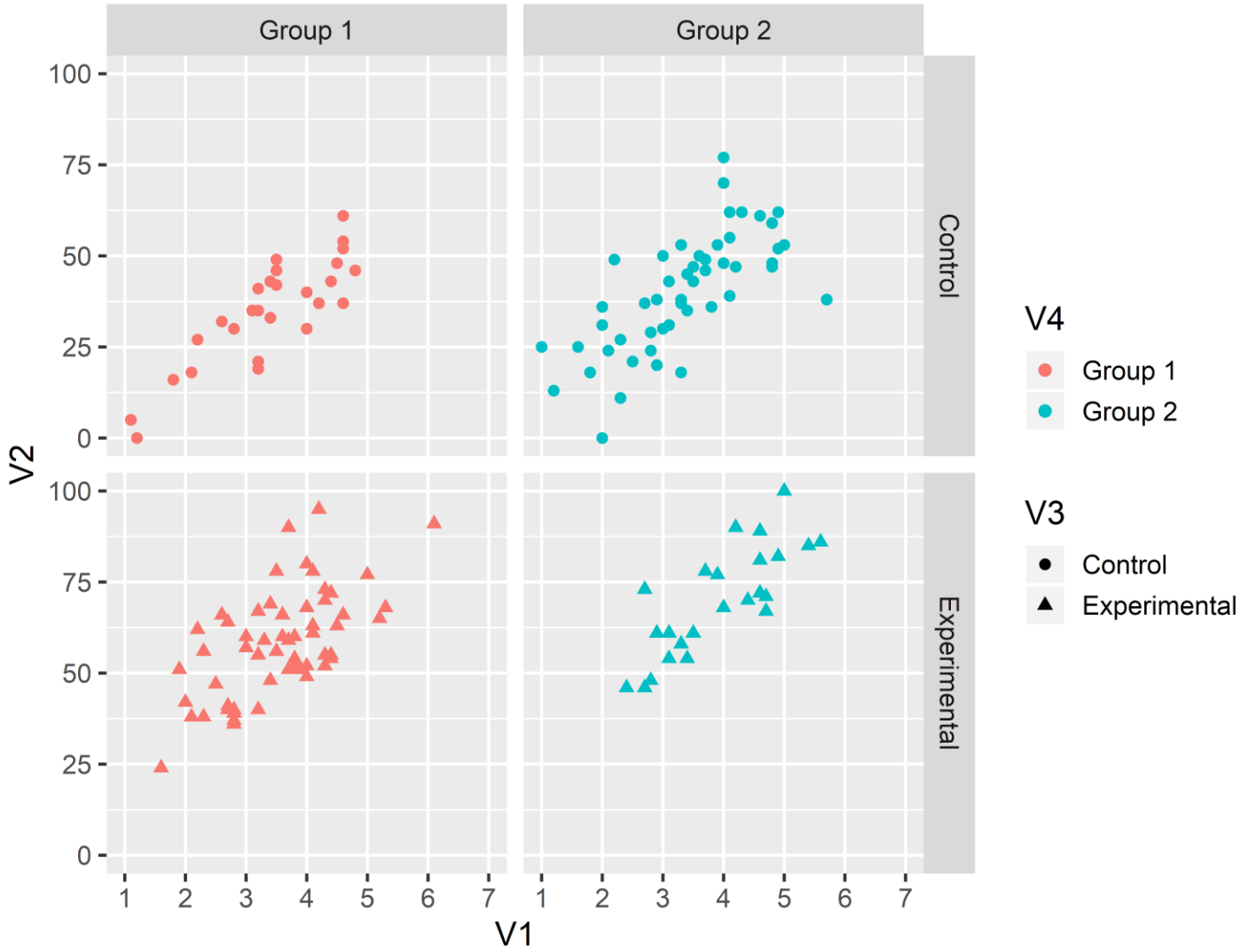
```r
# Example 1: A First Plot ----------------------

  # Load packages
  library(tidyverse)

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 1
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V3),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



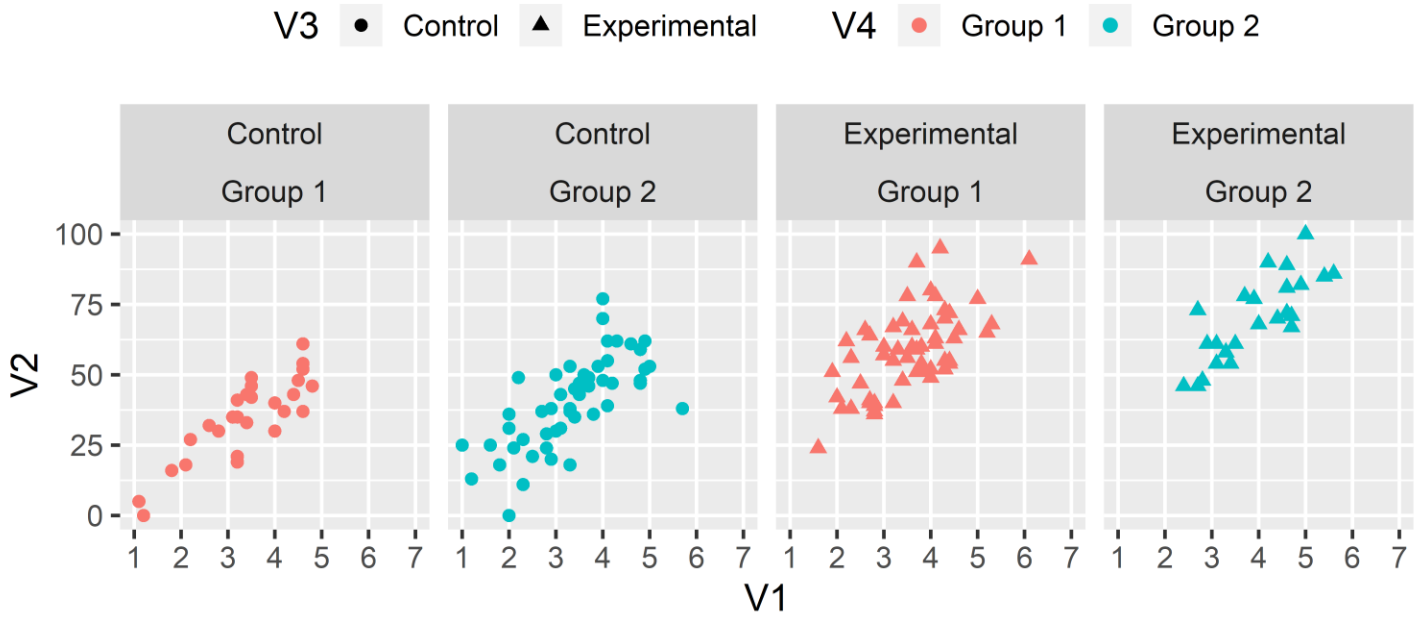
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



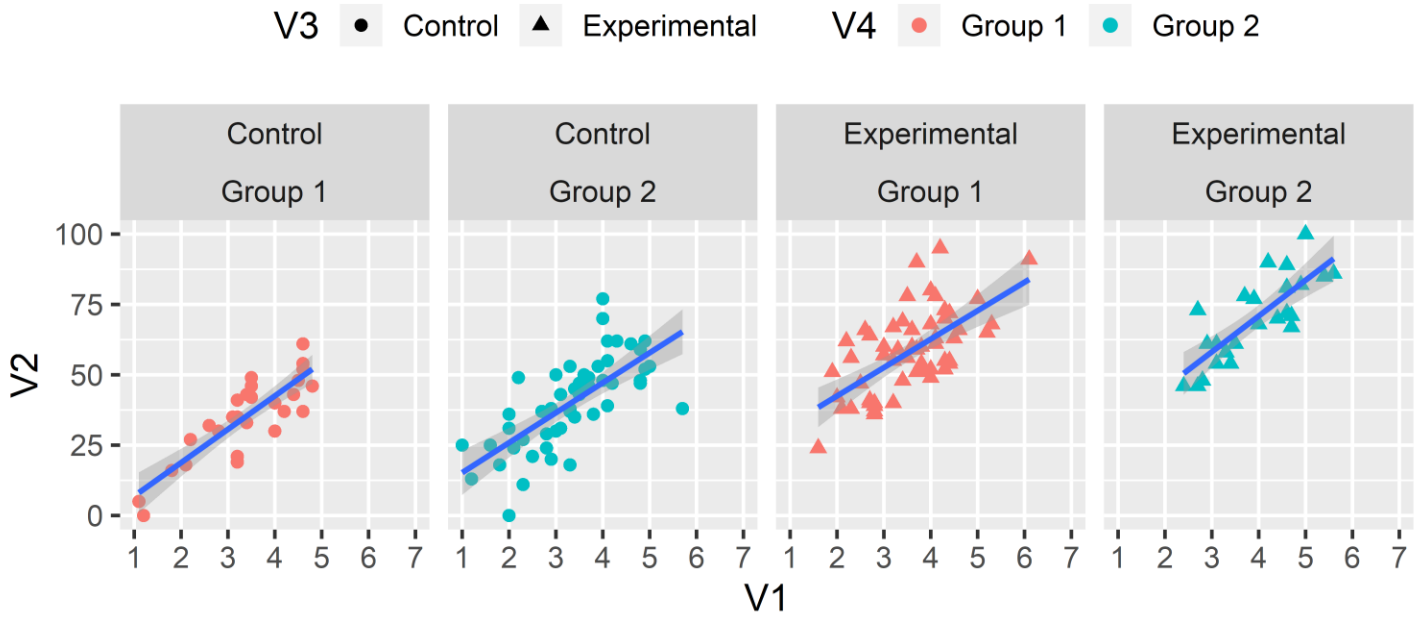
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(V3 ~ V4)
```



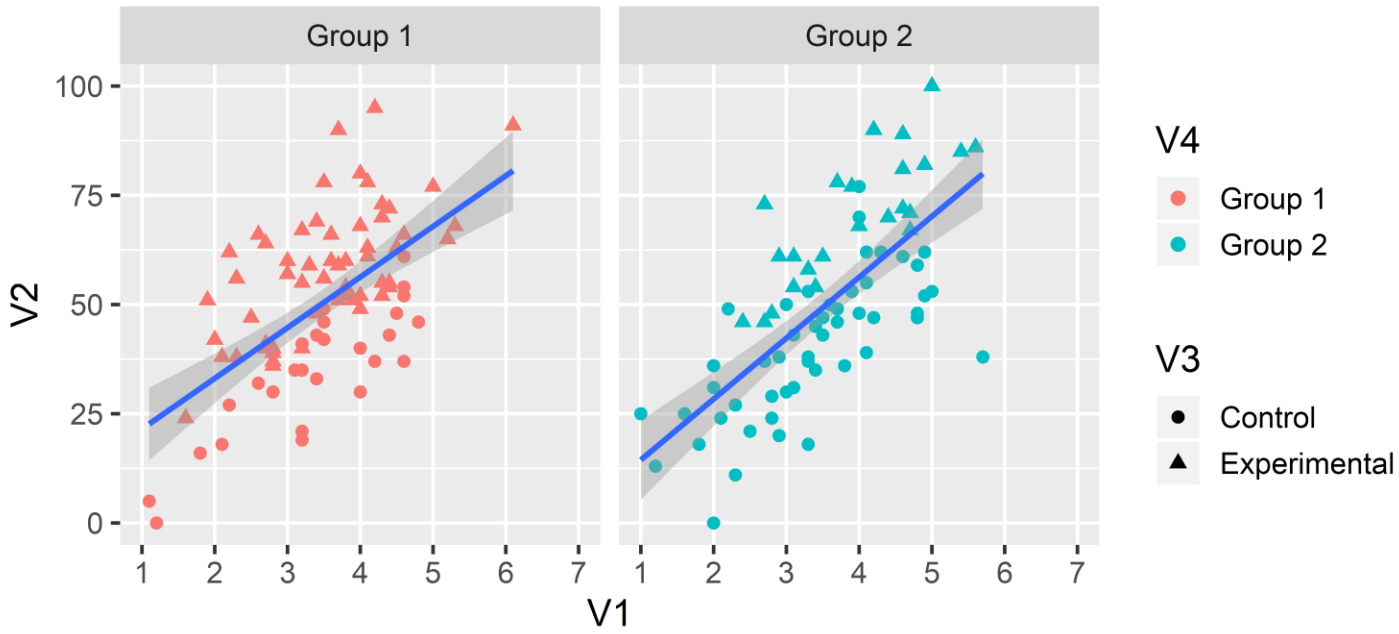
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V3 + V4) +
    theme(legend.position = "top")
```



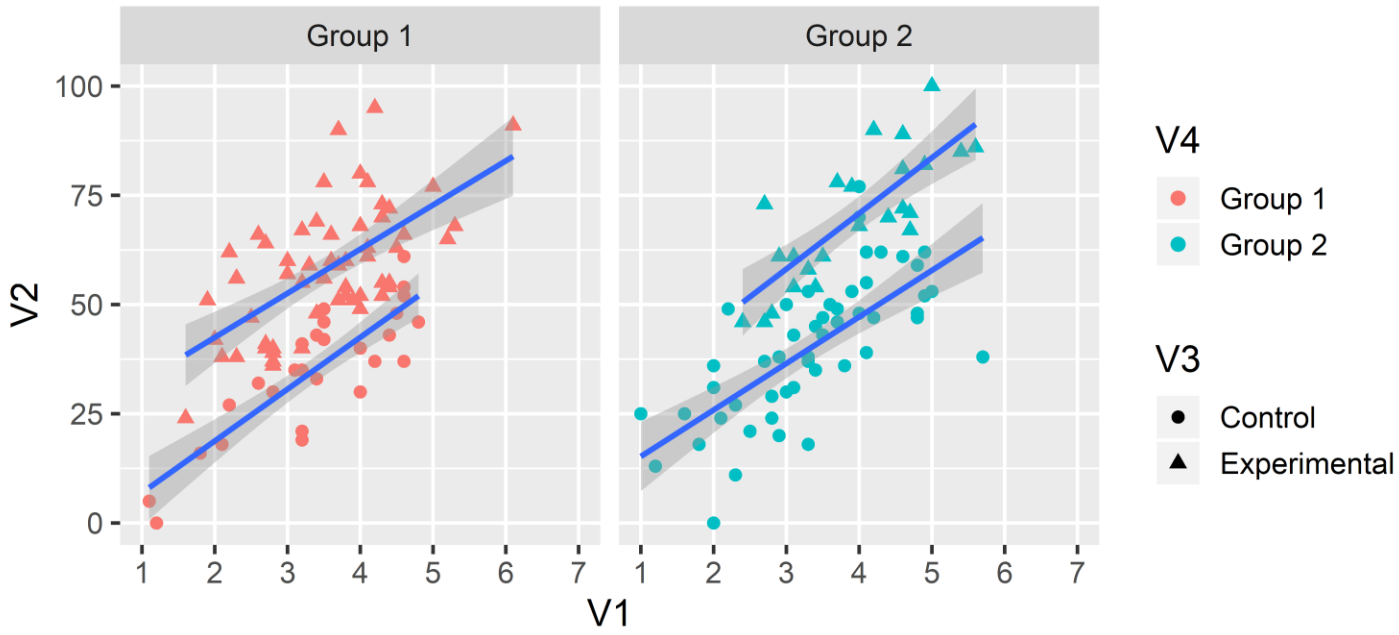
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    geom_smooth(method = "lm") +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V3 + V4) +
    theme(legend.position = "top")
```



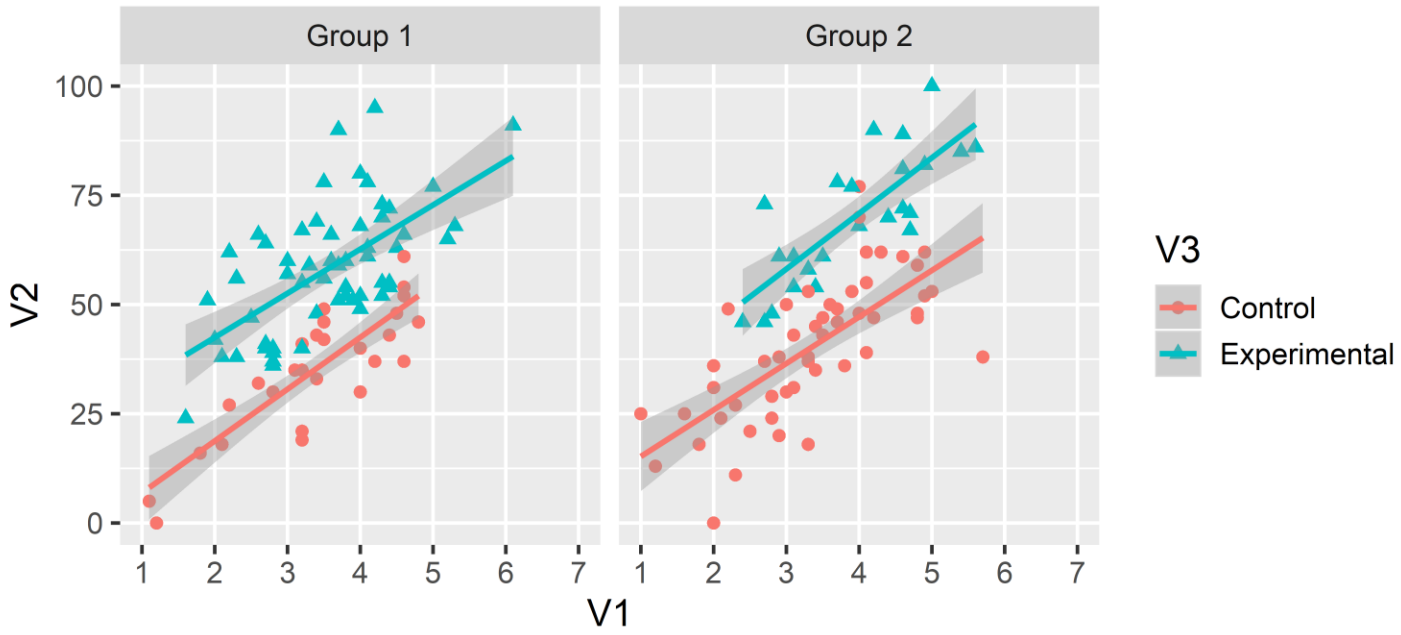
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    geom_smooth(method = "lm") +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



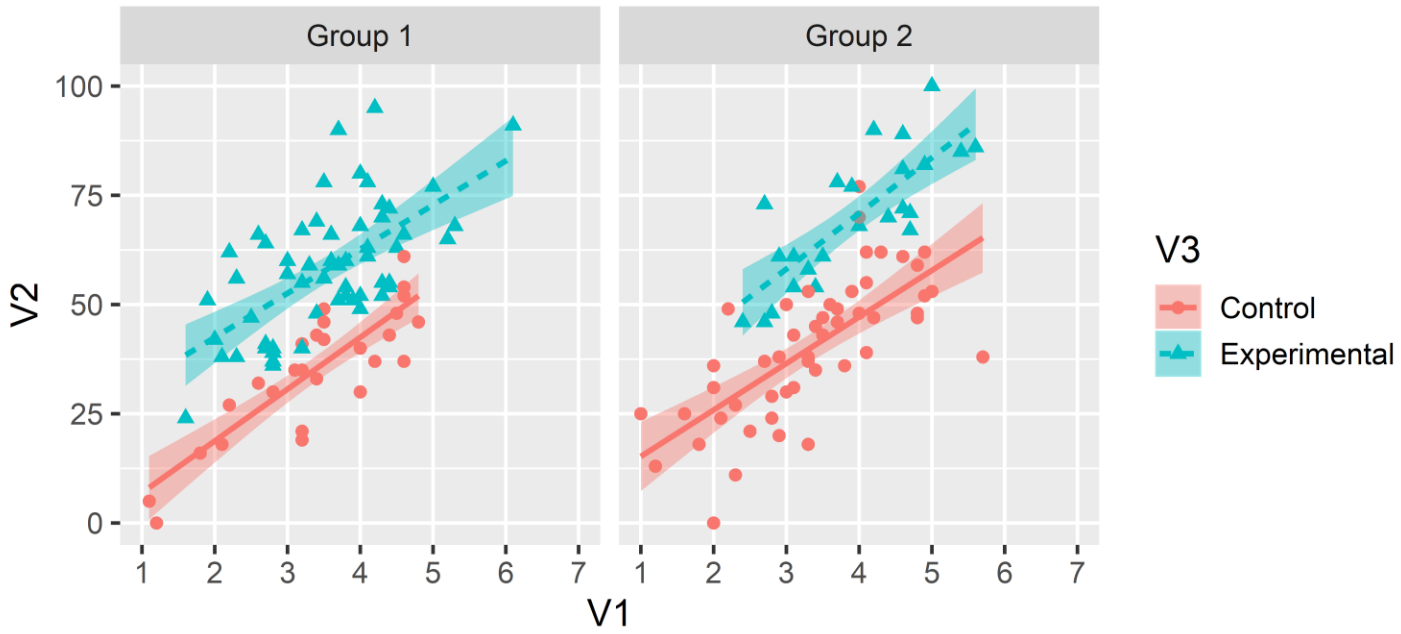
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2)) +
    geom_point(
      aes(shape = V3, colour = V4),
      size = 2
    ) +
    geom_smooth(
      aes(group = V3),
      method = "lm"
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2, colour = V3)) +
    geom_point(
      aes(shape = V3),
      size = 2
    ) +
    geom_smooth(
      aes(group = V3),
      method = "lm"
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



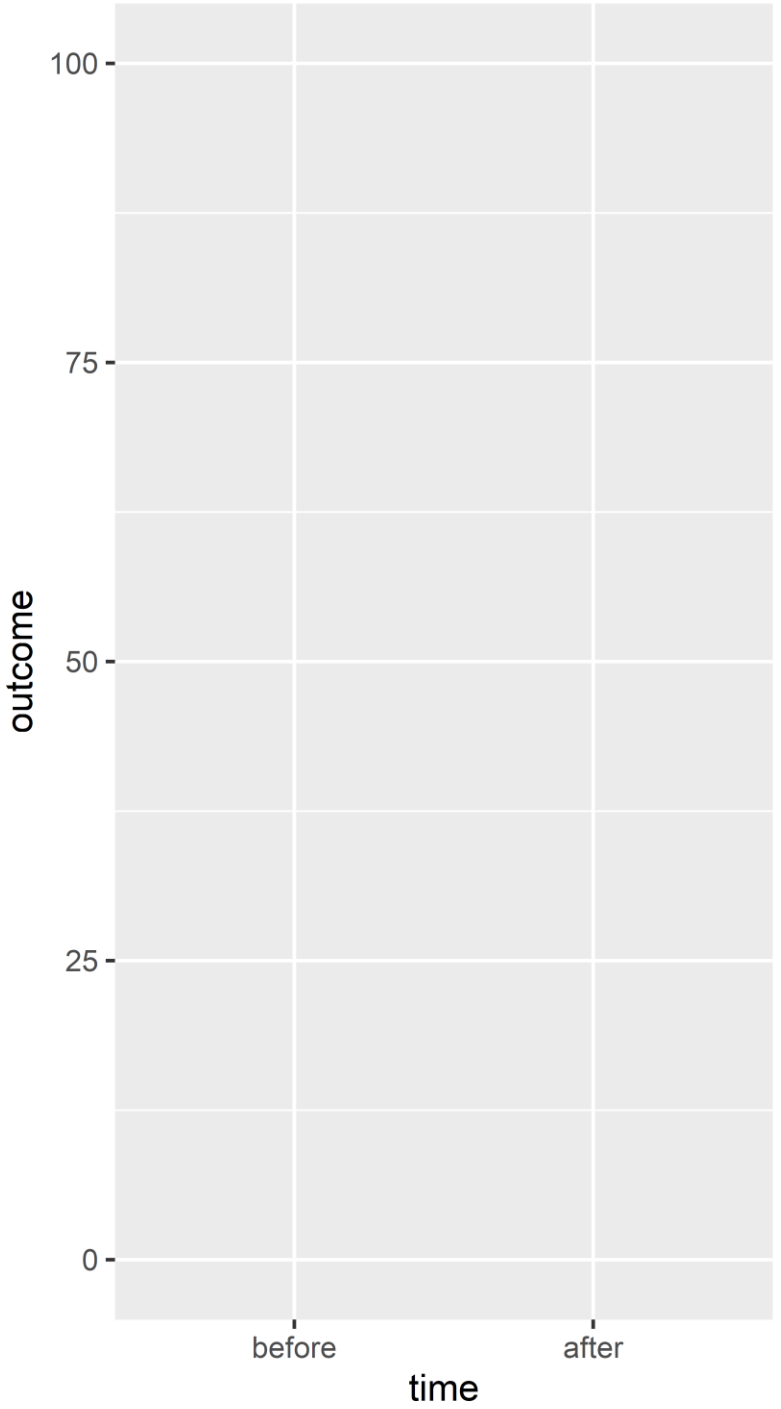
```r
# Example 2: Facets and Curves ----------------

  # Import data
  d1 <- read_rds("materials/d1.rds")

  # Figure 2
  ggplot(d1, aes(x = V1, y = V2, colour = V3)) +
    geom_point(
      aes(shape = V3),
      size = 2
    ) +
    geom_smooth(
      aes(fill = V3, linetype = V3),
      method = "lm"
    ) +
    scale_x_continuous(
      limits = c(1, 7),
      breaks = 1:7,
      minor_breaks = NULL
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(ratio = 6/100) +
    facet_grid(. ~ V4)
```



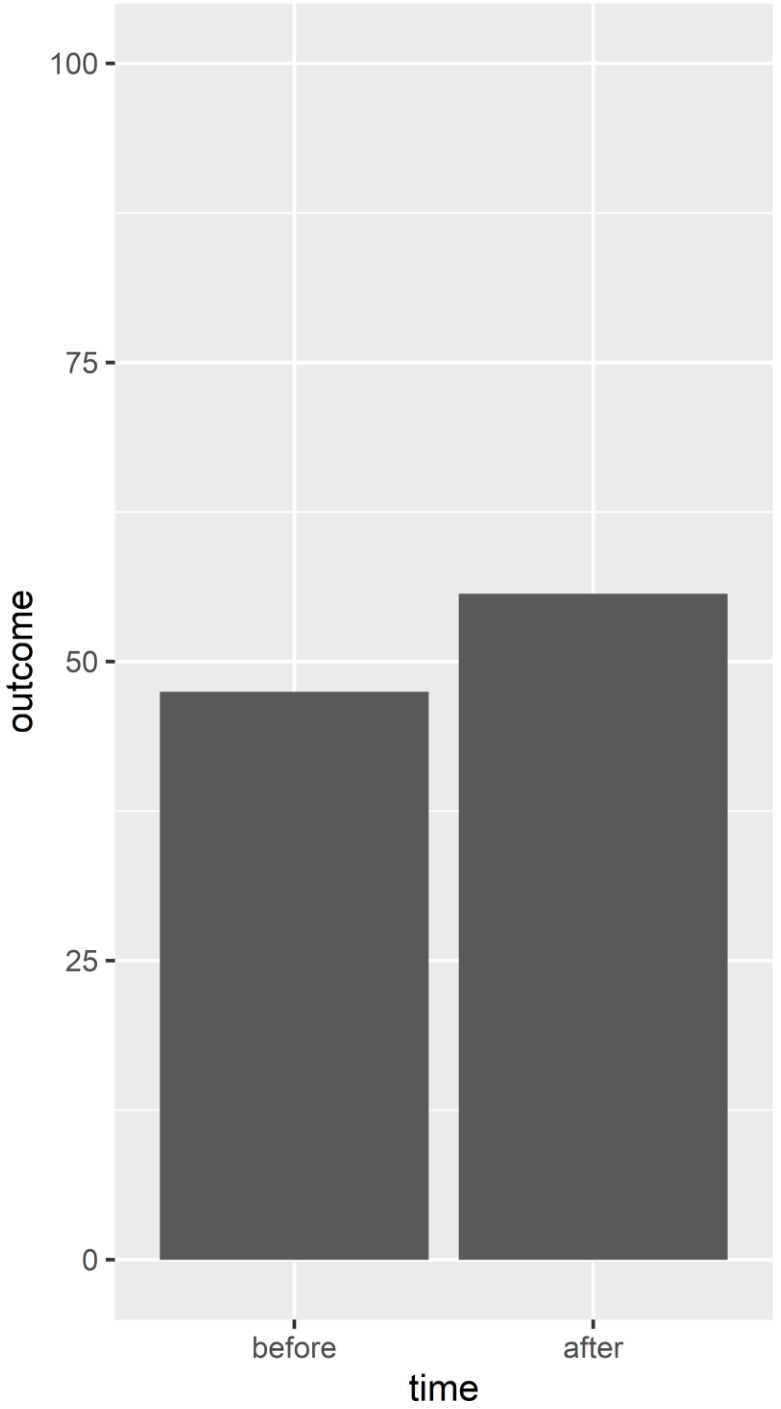
```r
# Example 3: Within-Subjects Plots ----------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")
```

```
> print(dl, n = 5)
# A tibble: 160 x 4
  person group   time   outcome
   <int> <ord>  <ord>    <int>
1      1   low before       41
2      1   low  after       NA
3      2   low before       30
4      2   low  after       20
5      3  high before       55
# ... with 155 more rows
```

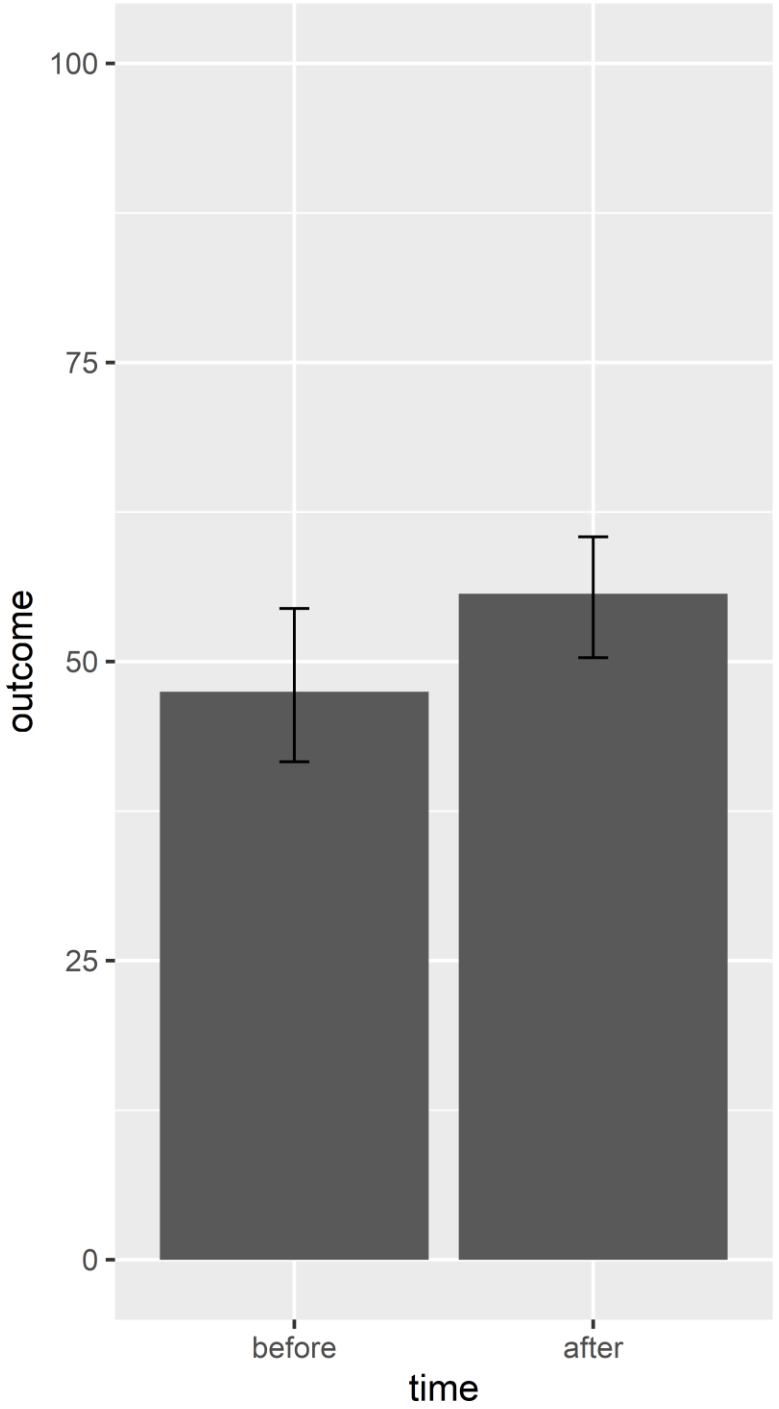
```r
# Example 3: Within-Subjects Plots -------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



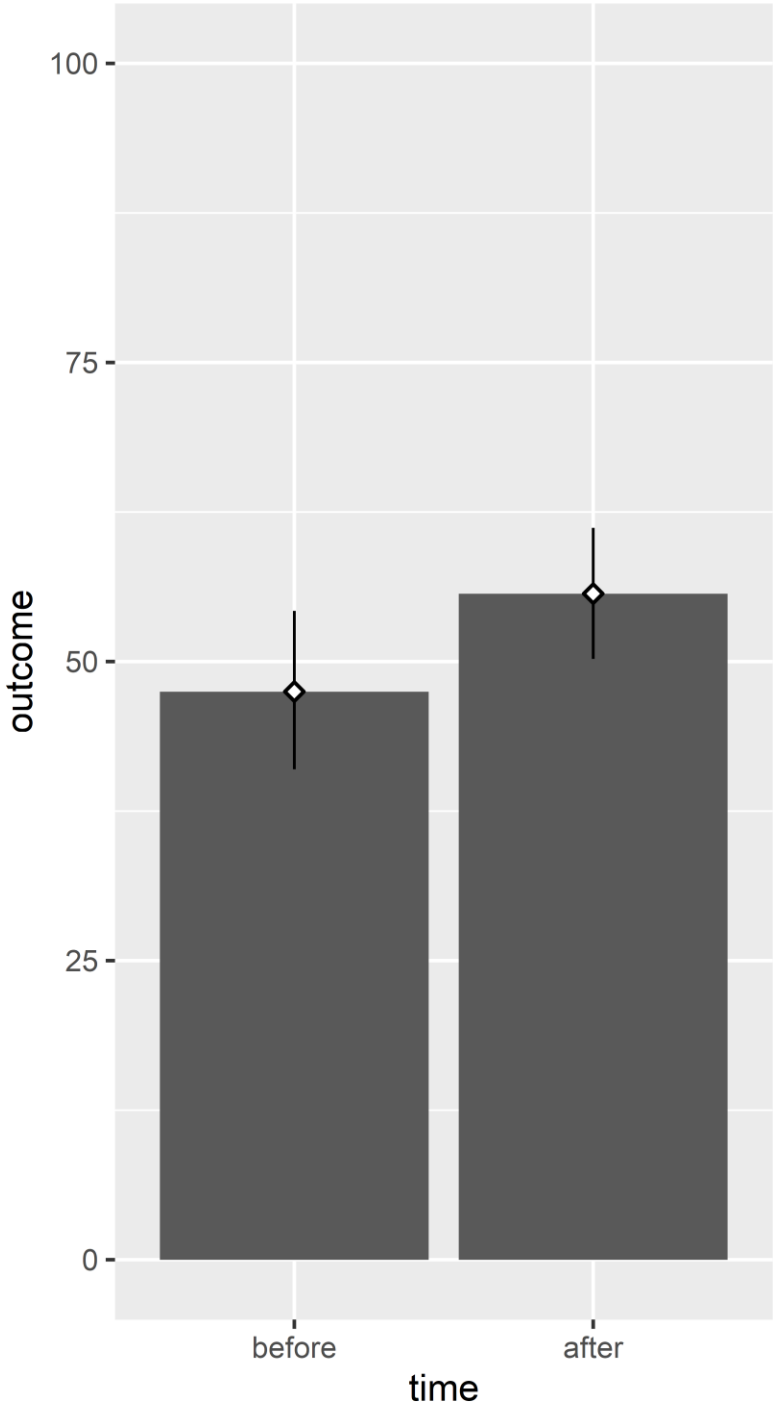
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    stat_summary(geom = "bar", fun.y = "mean") +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



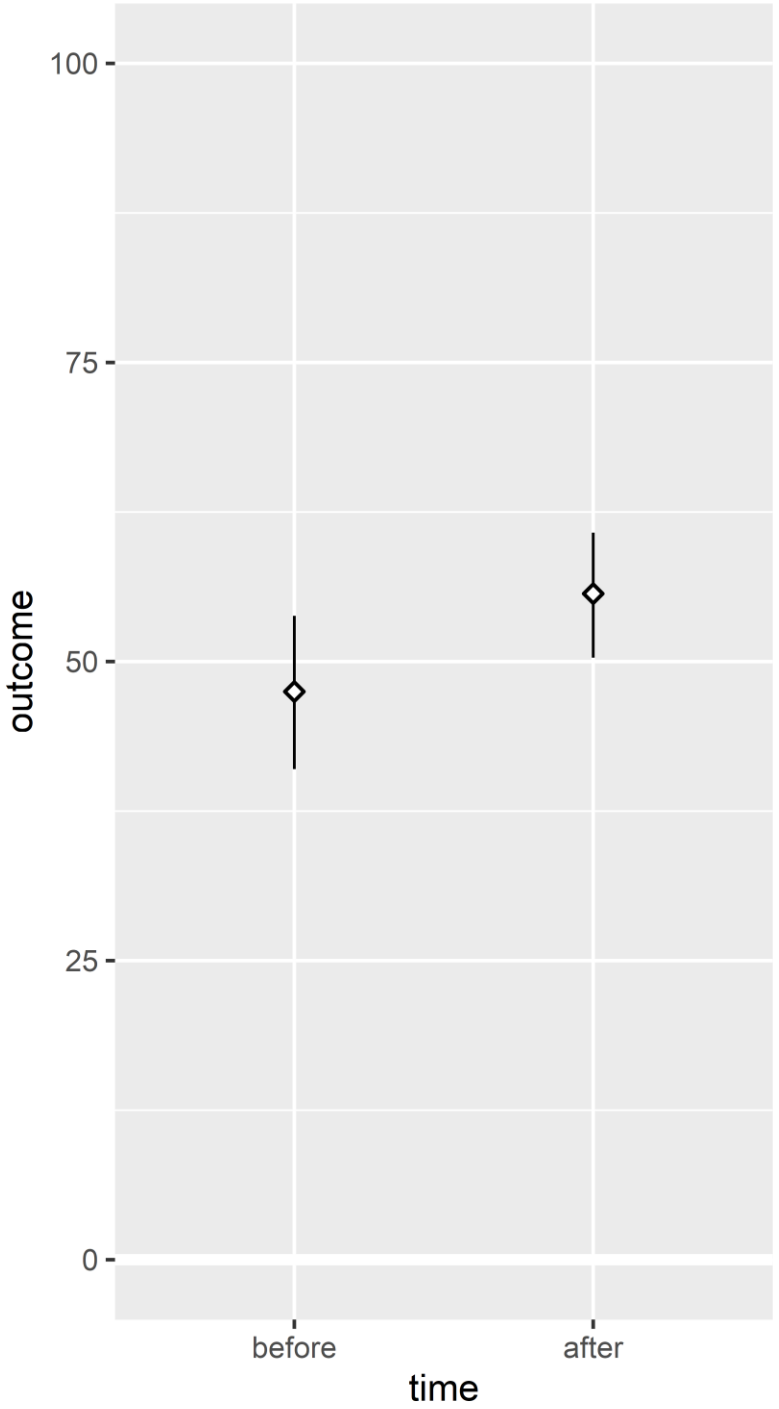
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    stat_summary(geom = "bar", fun.y = "mean") +
    stat_summary(
      geom = "errorbar",
      fun.data = "mean_cl_boot",
      width = 0.1
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



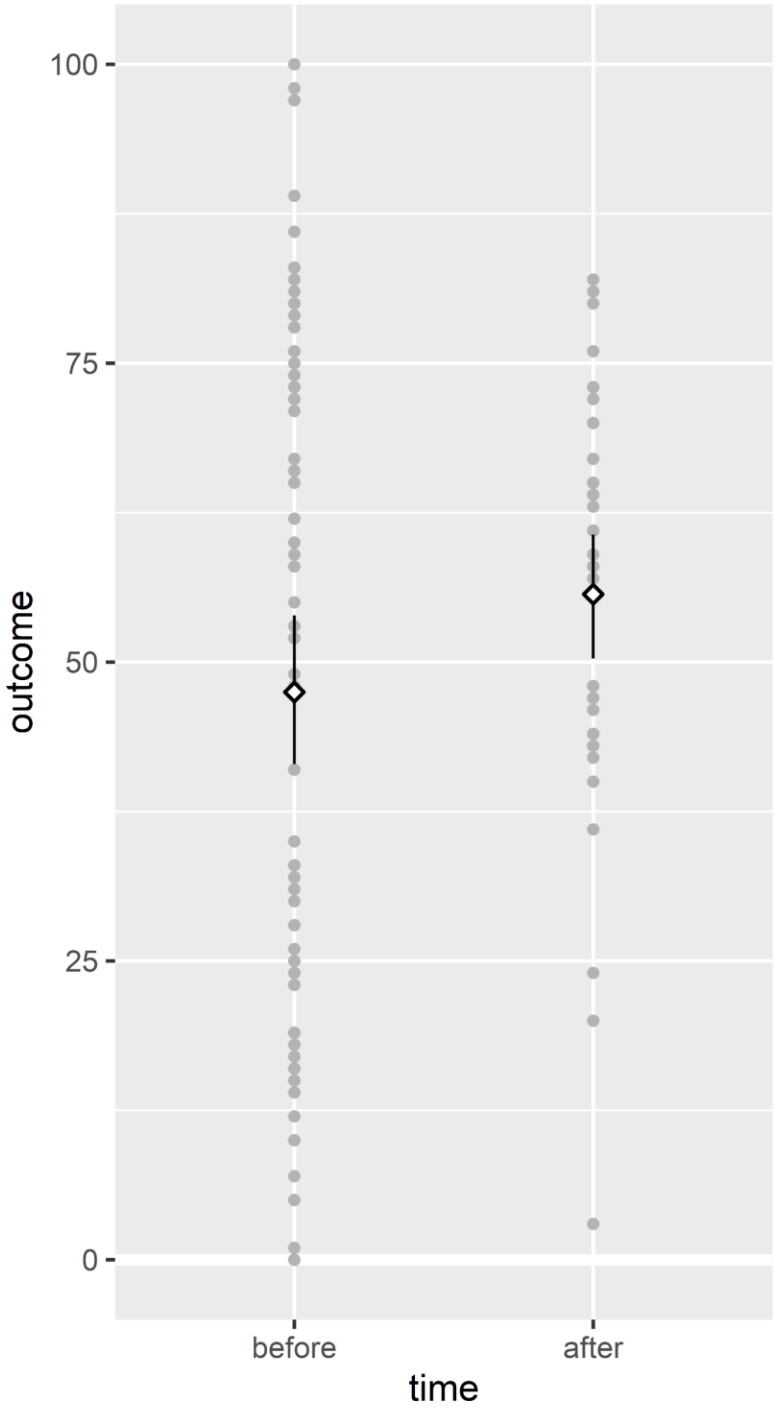
```r
# Example 3: Within-Subjects Plots -------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    stat_summary(geom = "bar", fun.y = "mean") +
    stat_summary(
      geom = "pointrange",
      fun.data = "mean_cl_boot",
      shape = "diamond filled",
      fill = "white"
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



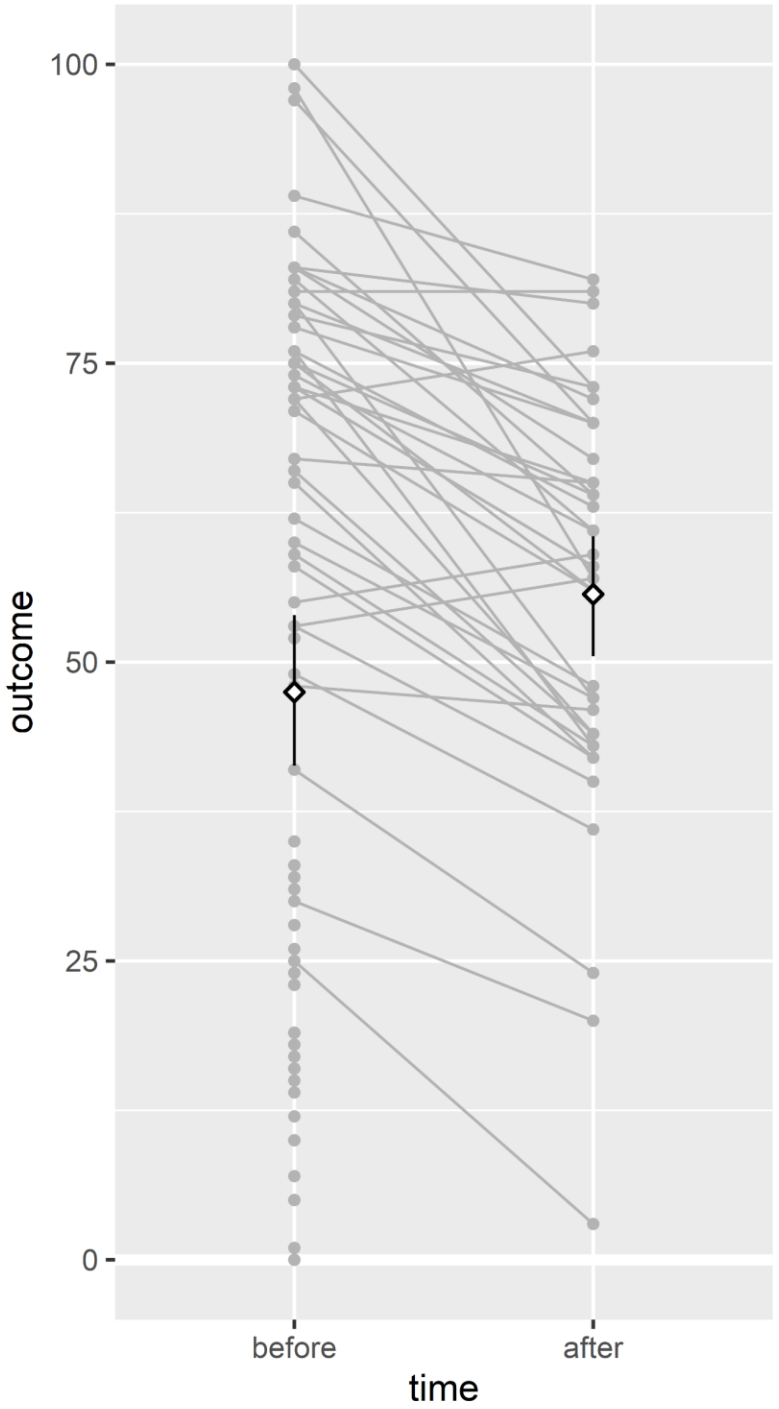
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    geom_hline(
      yintercept = 0,
      colour = "white",
      size = 2
    ) +
    stat_summary(
      geom = "pointrange",
      fun.data = "mean_cl_boot",
      shape = "diamond filled",
      fill = "white"
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



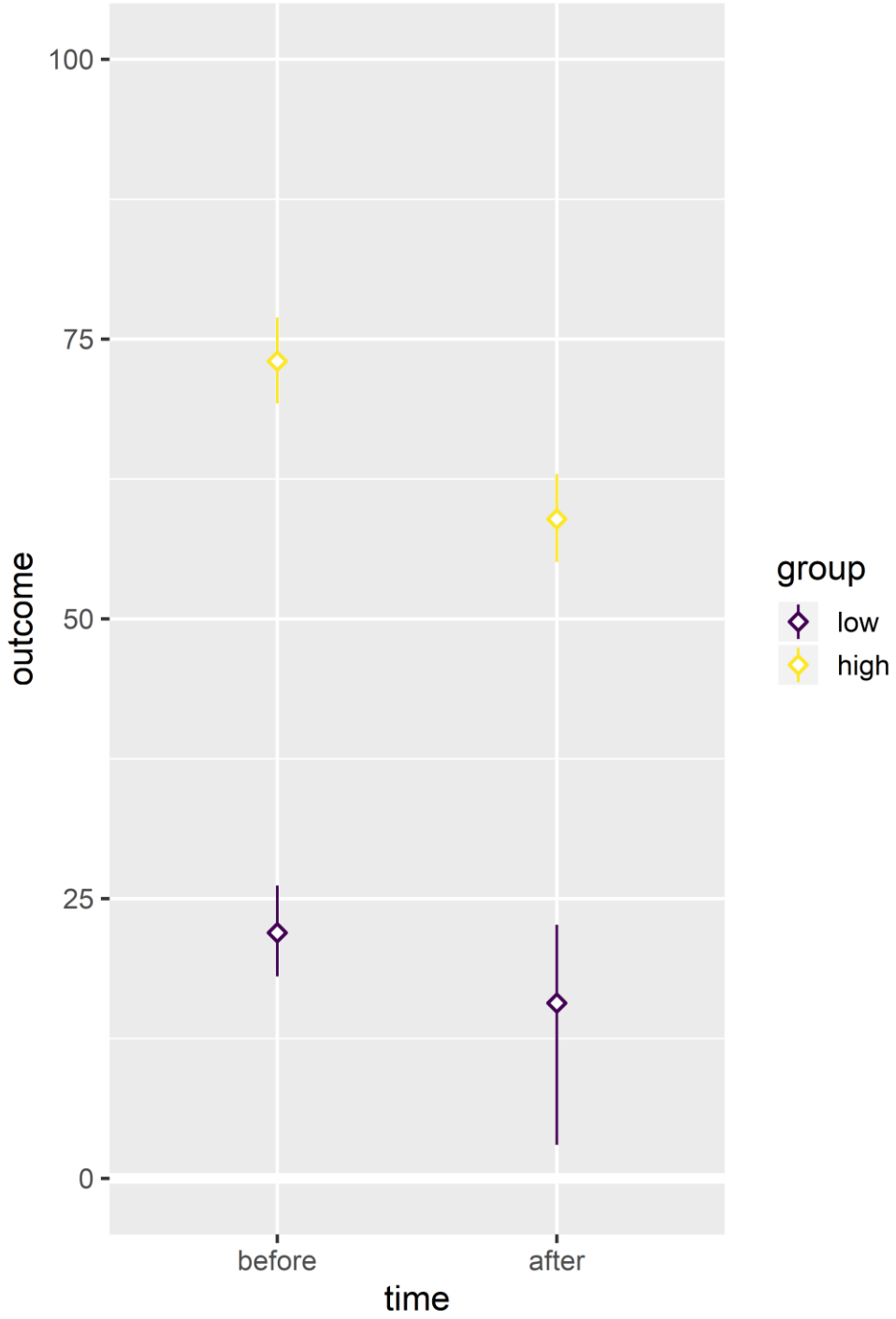
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    geom_hline(
      yintercept = 0,
      colour = "white",
      size = 2
    ) +
    geom_point(colour = "grey70") +
    stat_summary(
      geom = "pointrange",
      fun.data = "mean_cl_boot",
      shape = "diamond filled",
      fill = "white"
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



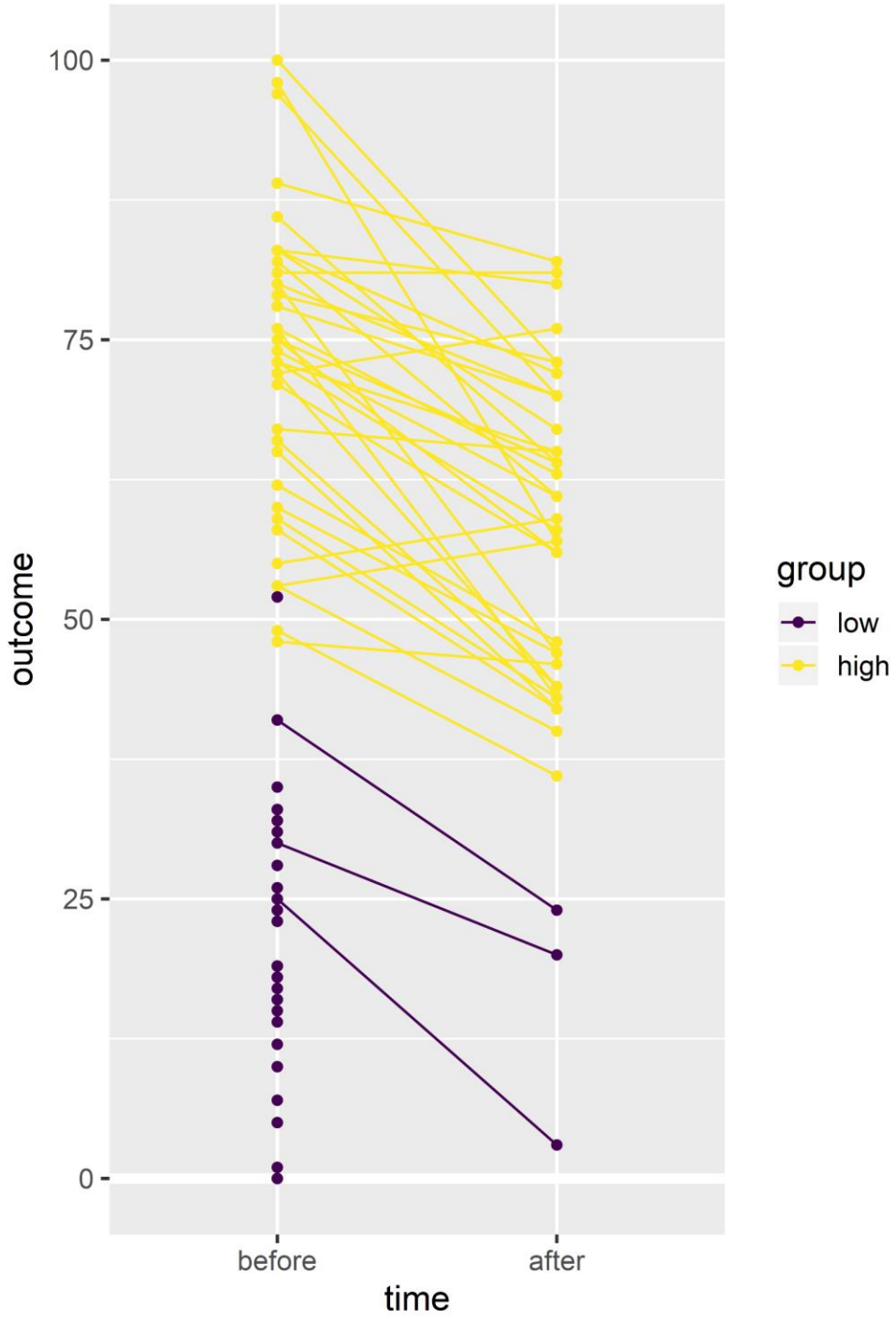
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(x = time, y = outcome)) +
    geom_hline(
      yintercept = 0,
      colour = "white",
      size = 2
    ) +
    geom_line(
      aes(group = person),
      colour = "grey70"
    ) +
    geom_point(colour = "grey70") +
    stat_summary(
      geom = "pointrange",
      fun.data = "mean_cl_boot",
      shape = "diamond filled",
      fill = "white"
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



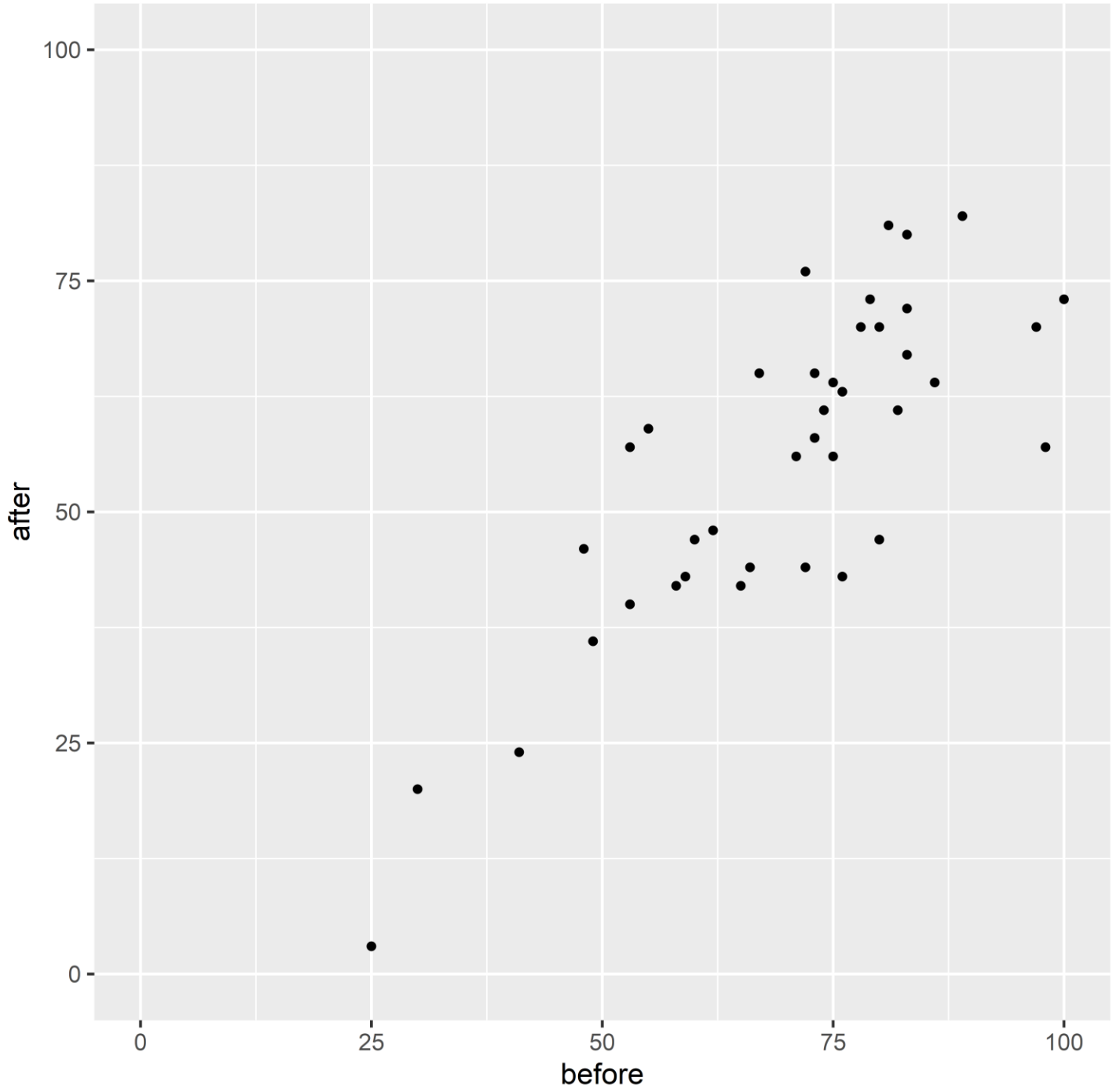
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(
      x = time, y = outcome, colour = group
    )) +
    geom_hline(
      yintercept = 0,
      colour = "white",
      size = 2
    ) +
    stat_summary(
      geom = "pointrange",
      fun.data = "mean_cl_boot",
      shape = "diamond filled",
      fill = "white"
    ) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



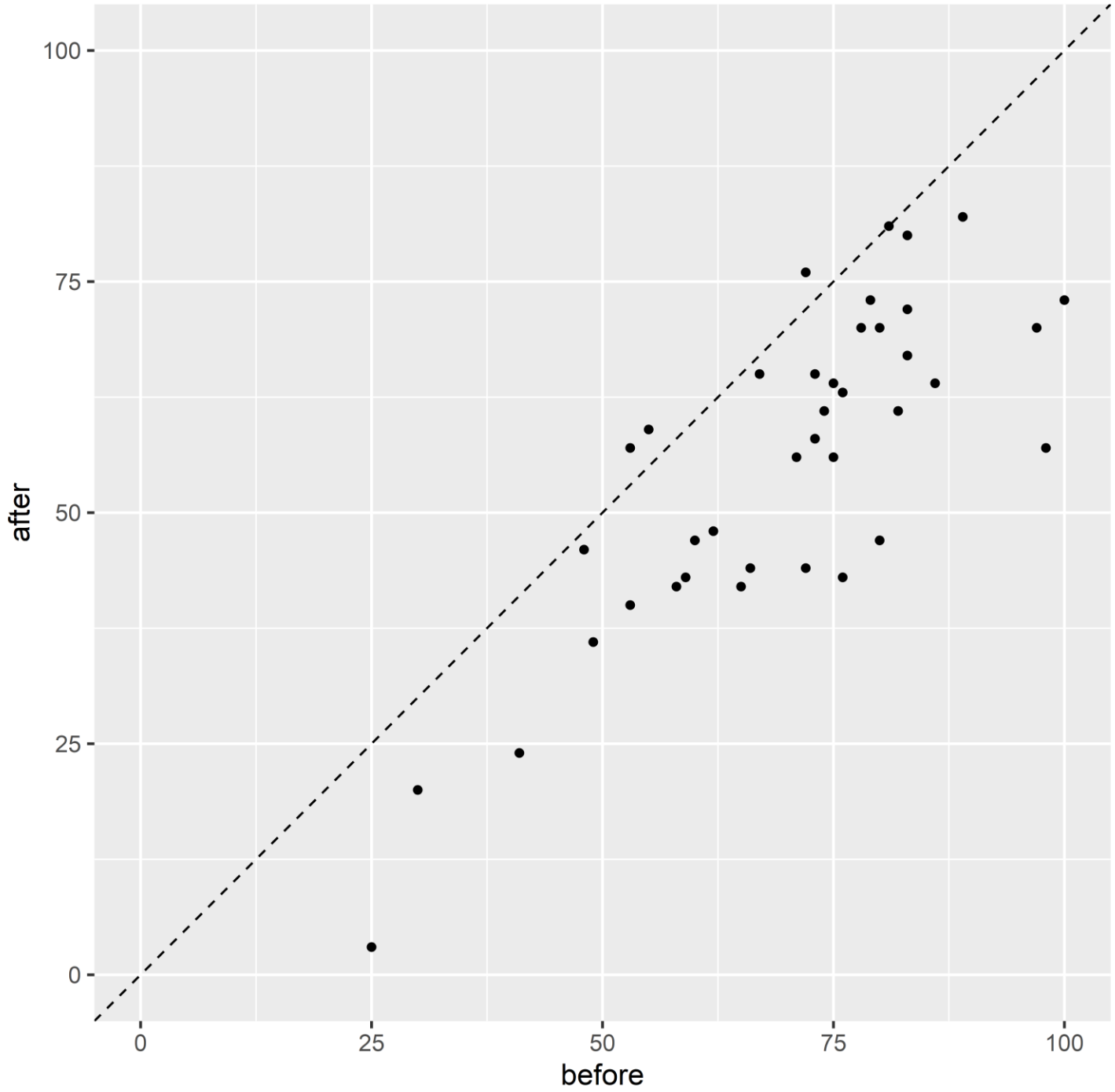
```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Figure 3
  ggplot(dl, aes(
      x = time, y = outcome, colour = group
    )) +
    geom_hline(
      yintercept = 0,
      colour = "white",
      size = 2
    ) +
    geom_line(aes(group = person)) +
    geom_point() +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1/25)
```



```r
# Example 3: Within-Subjects Plots -------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Transform data
  dw <- dl %>% spread(time, outcome)

  # Figure 3
  ggplot(dw, aes(x = before, y = after)) +
    geom_point() +
    scale_x_continuous(limits = c(0, 100)) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1)
```

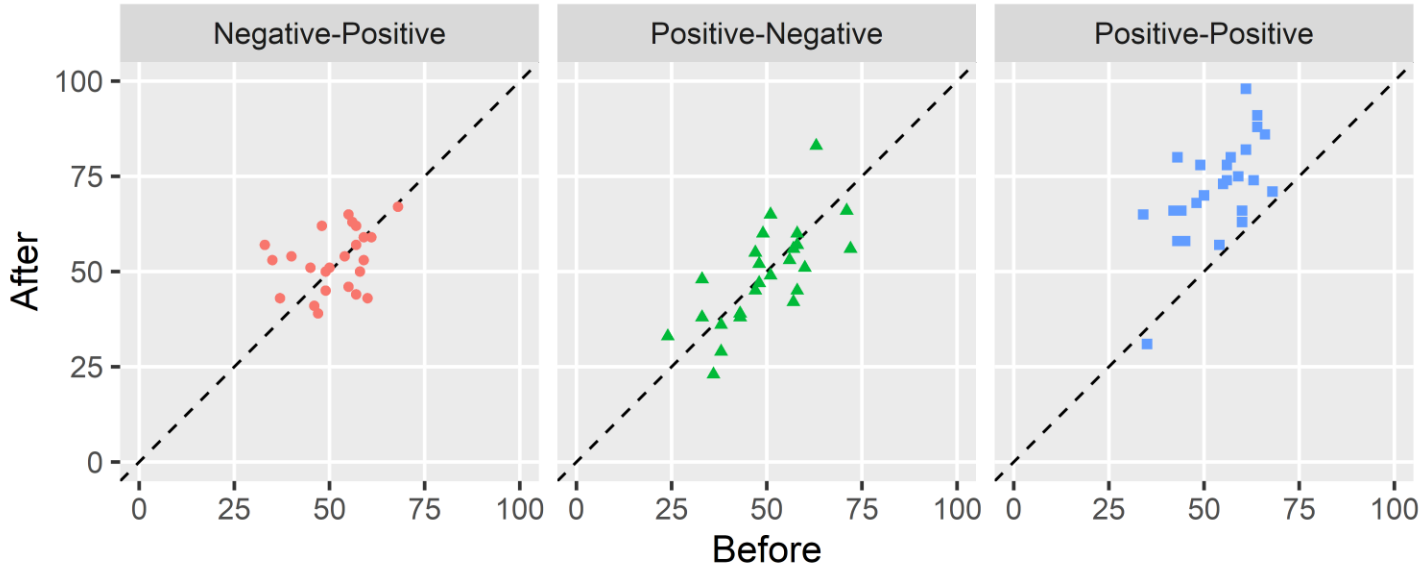


```r
# Example 3: Within-Subjects Plots ------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk3.rds")

  # Transform data
  dw <- dl %>% spread(time, outcome)

  # Figure 3
  ggplot(dw, aes(x = before, y = after)) +
    geom_abline(
      intercept = 0,
      slope = 1,
      linetype = "dashed“
    ) +
    geom_point() +
    scale_x_continuous(limits = c(0, 100)) +
    scale_y_continuous(limits = c(0, 100)) +
    coord_fixed(1)
```

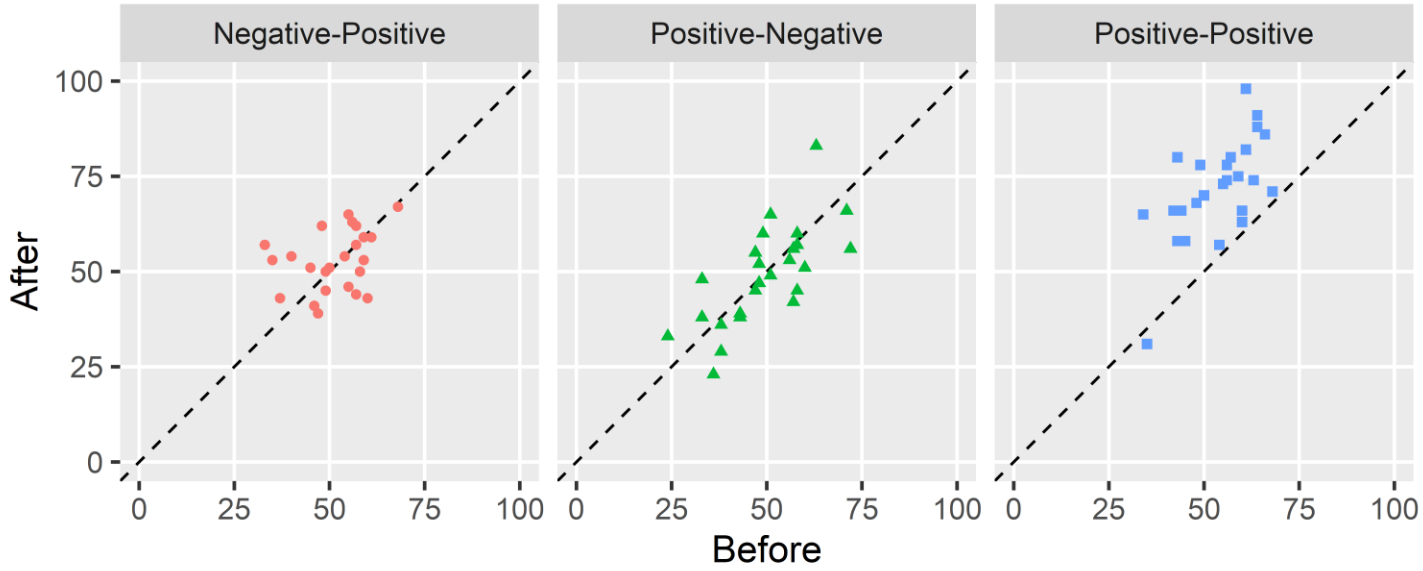


```r
# Exercise (workshop.R) ----------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")




  ##########################################################
  # If you have trouble seeing colour, add                 #
  # scale_colour_viridis_d() and                           #
  # scale_fill_viridis_d() to your plot; or use            #
  # scale_colour_grey() and scale_fill_grey().             #
  ##########################################################
```

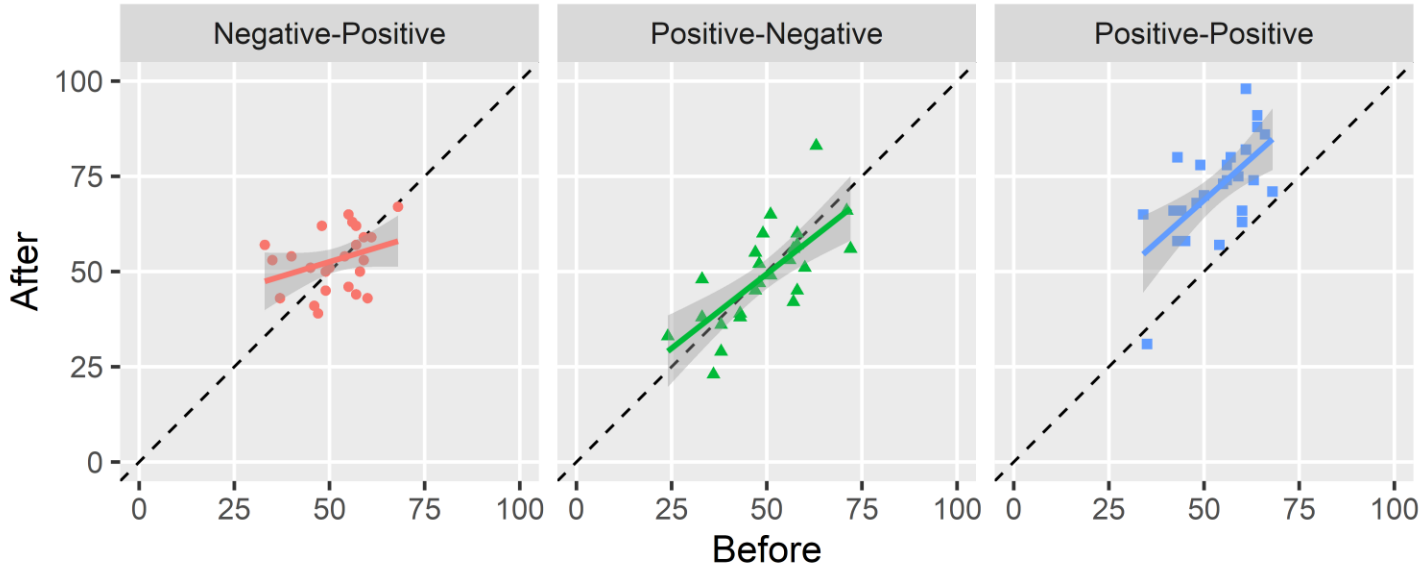
```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Figure 4
  ggplot(dw, aes(
      x = Before,
      y = After,
      colour = condition
    )) +
    geom_abline(
      intercept = 0,
      slope = 1,
      linetype = "dashed"
    ) +
    geom_point(aes(shape = condition)) +
    scale_x_continuous(
      limits = c(0, 100), minor_breaks = NULL
    ) +
    scale_y_continuous(
      limits = c(0, 100), minor_breaks = NULL
    ) +
    facet_grid(. ~ condition) +
    coord_fixed(1) +
    theme(legend.position = "none")
```



```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Figure 4
  fig <- ggplot(dw, aes(
      x = Before,
      y = After,
      colour = condition
    )) +
    geom_abline(
      intercept = 0,
      slope = 1,
      linetype = "dashed"
    ) +
    geom_point(aes(shape = condition)) +
    scale_x_continuous(
      limits = c(0, 100), minor_breaks = NULL
    ) +
    scale_y_continuous(
      limits = c(0, 100), minor_breaks = NULL
    ) +
    facet_grid(. ~ condition) +
    coord_fixed(1) +
    theme(legend.position = "none")
```



```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Figure 4
  fig + geom_smooth(method = "lm")
```

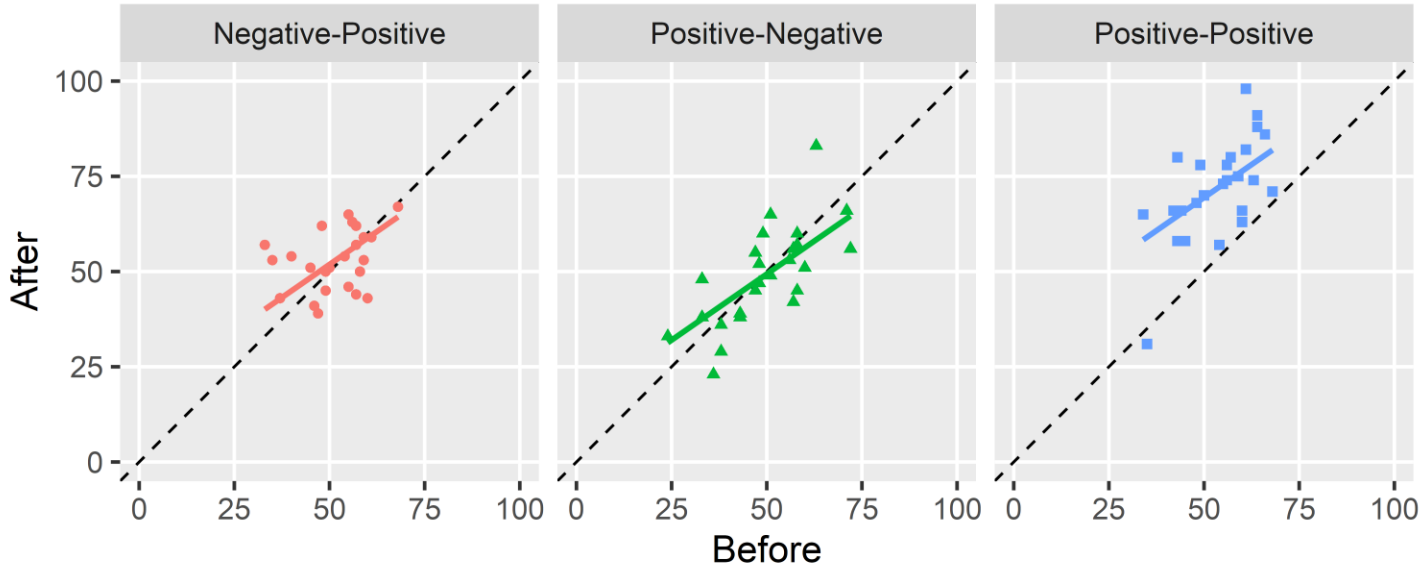


```r
# Example 4: Models ------------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Estimate model
  fit <- lm(After ~ Before + condition, data = dw)

  # Load package
  library(broom)
```

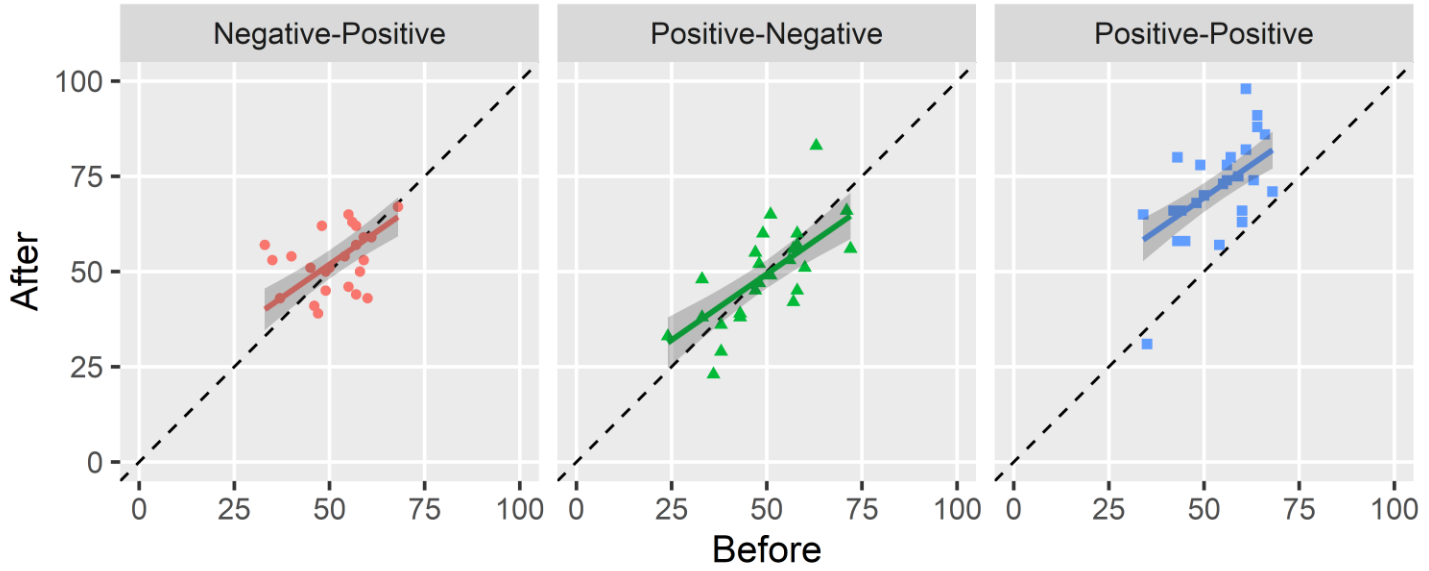
```
> tidy(fit)
# A tibble: 4 x 5
                                term estimate std.error
                               <chr>    <dbl>     <dbl>
1                       (Intercept)   17.2        5.98
2                            Before    0.694      0.110
3 conditionPositive-Negative          -2.51       2.69
4 conditionPositive-Positive          17.6        2.69
```

```
> augment(fit)
# A tibble: 75 x 10
  After Before          condition .fitted .se.fit .resid   .hat .sigma  .cooksd
  <int>  <int>              <chr>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>
1    29     38 Positive-Negative    41.0    2.28  -12.0 0.0579   9.42 0.0263
2    39     43 Positive-Negative    44.5    2.03   -5.5 0.0458   9.52 0.00422
3    52     48 Positive-Negative    48.0    1.90    4.0 0.0403   9.53 0.00199
4    55     47 Positive-Negative    47.3    1.92    7.7 0.0409   9.49 0.00741
5    49     51 Positive-Negative    50.0    1.90   -1.0 0.0403   9.54 0.000132
# ... with 70 more rows
```

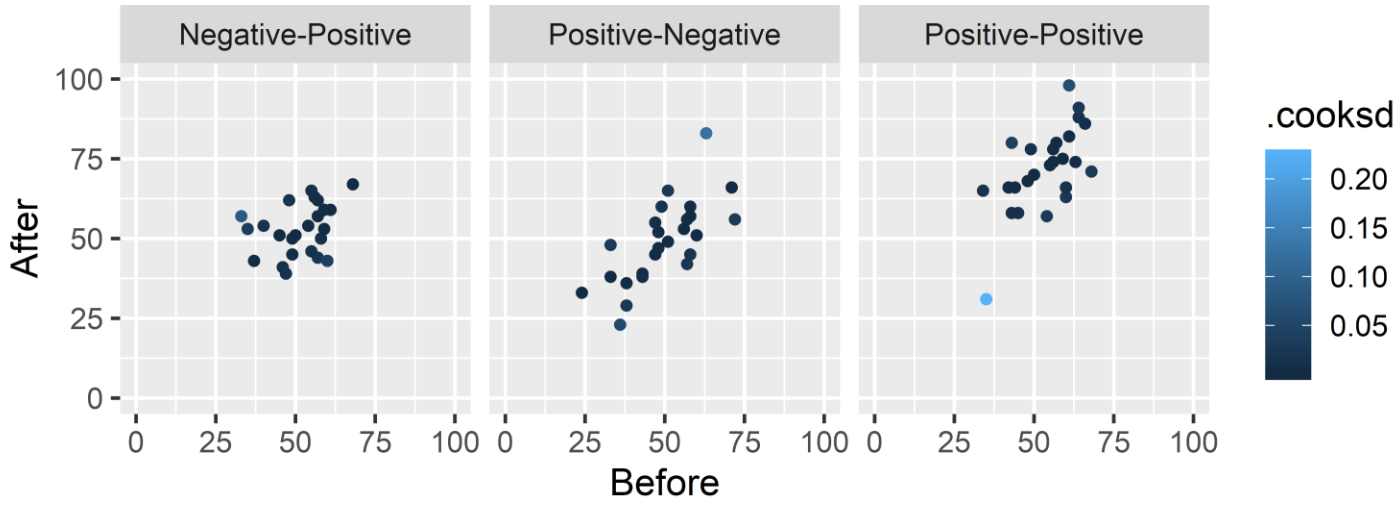
```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Estimate model
  fit <- lm(After ~ Before + condition,
            data = dw)

  # Load package
  library(broom)

  # Figure 4
  fig +
    geom_line(
      data = augment(fit),
      aes(y = .fitted),
      size = 1
     )
```



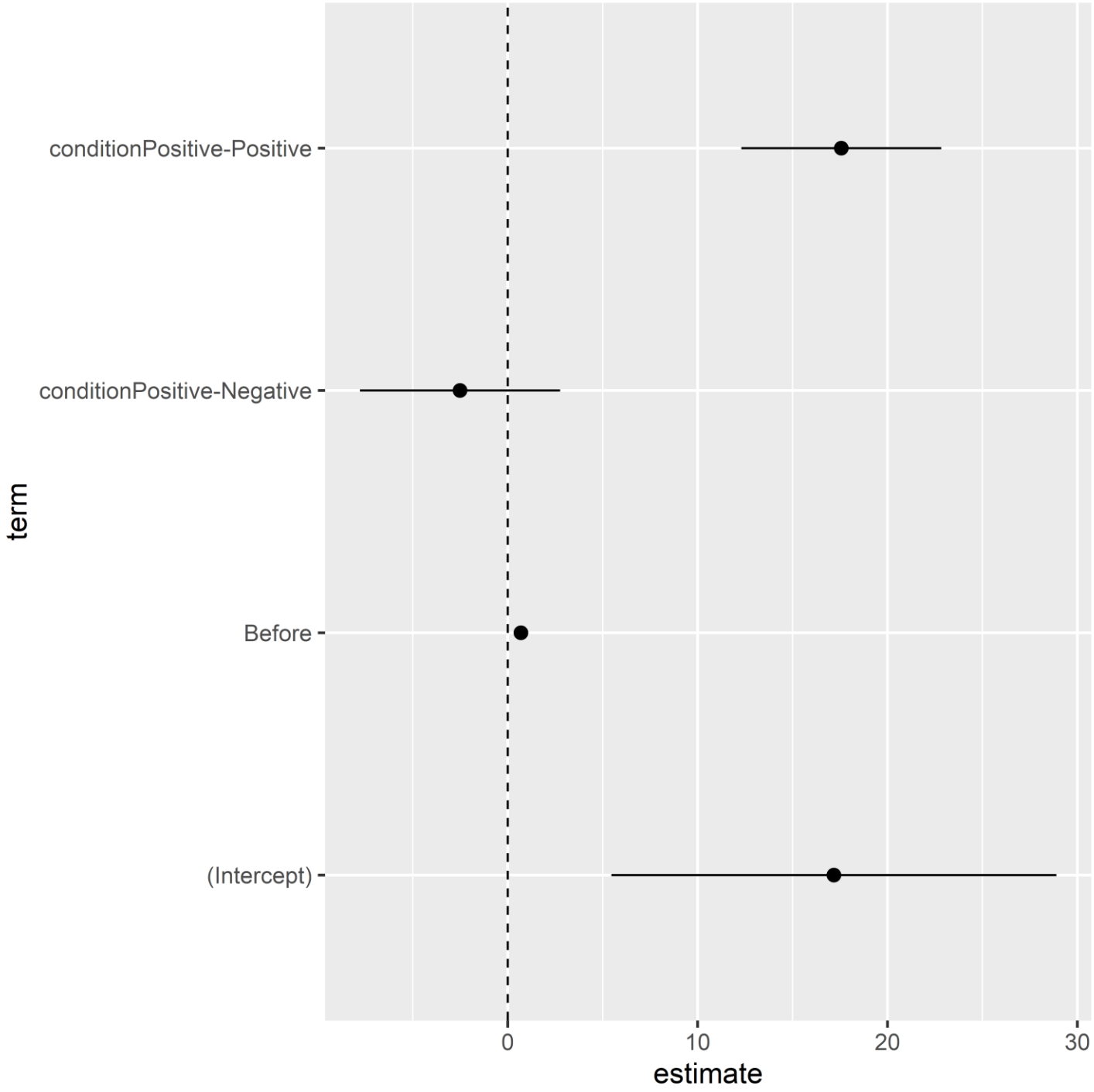
```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Estimate model
  fit <- lm(After ~ Before + condition,
            data = dw)

  # Load package
  library(broom)

  # Figure 4
  fig +
    geom_line(
      data = augment(fit),
      aes(y = .fitted),
      size = 1
    ) +
    geom_ribbon(
      data = augment(fit),
      aes(ymin = .fitted - 1.96*.se.fit,
          ymax = .fitted + 1.96*.se.fit),
      size = 1,
      colour = NA,
      alpha = 0.25
    )
```



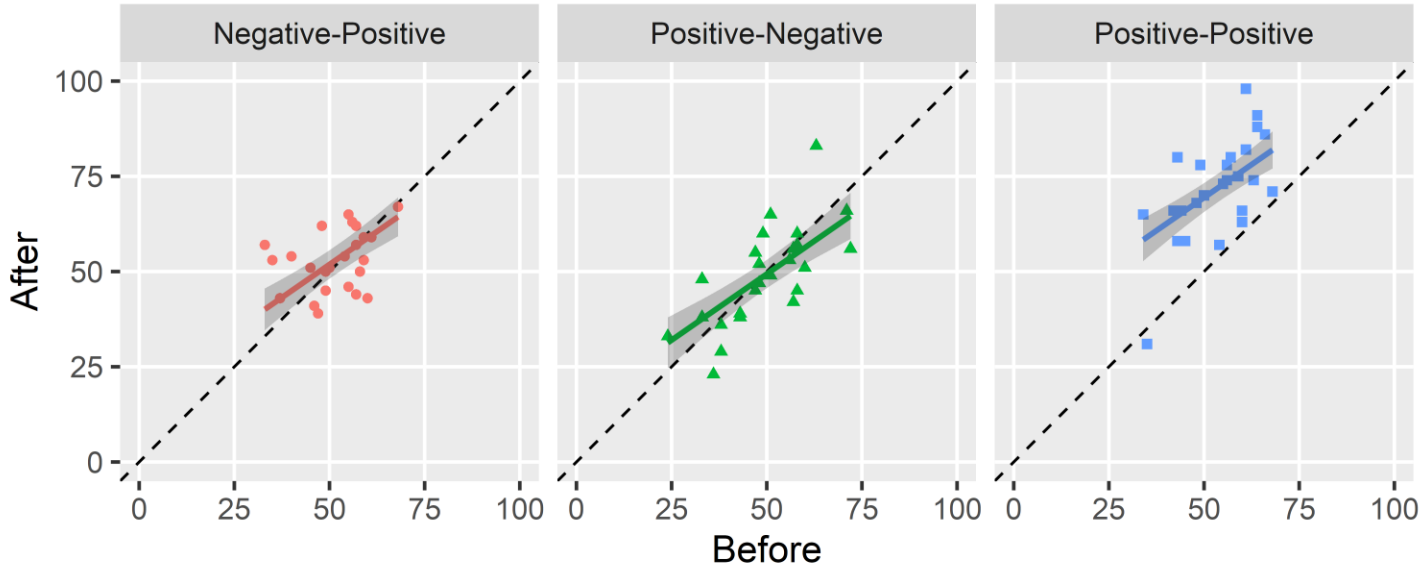
```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Estimate model
  fit <- lm(After ~ Before + condition,
            data = dw)

  # Load package
  library(broom)

  # Figure 4
  ggplot(augment(fit), aes(Before, After)) +
    geom_point(aes(colour = .cooksd)) +
    scale_x_continuous(limits = c(0, 100)) +
    scale_y_continuous(limits = c(0, 100)) +
    facet_grid(. ~ condition) +
    coord_fixed(1)
```



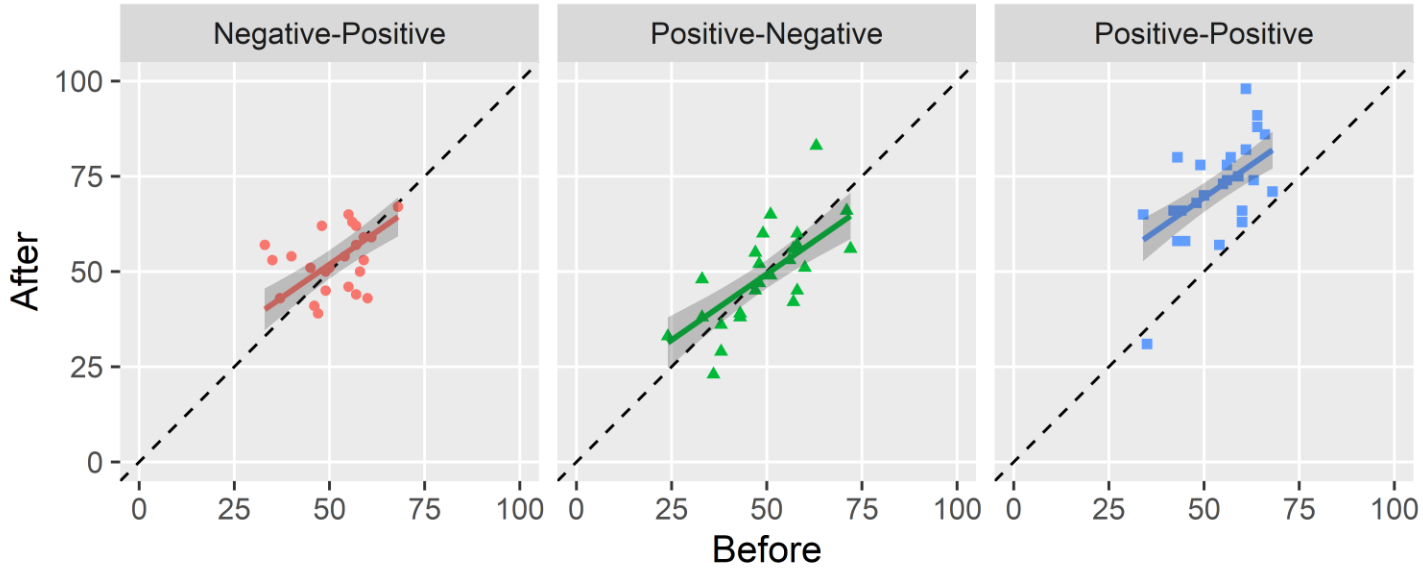
```r
# Example 4: Models ---------------------------

  # Import data
  dl <- read_rds("materials/gwtp/dl_wk4.rds")

  # Transform data
  dw <- dl %>% spread(time, attitudes)

  # Estimate model
  fit <- lm(After ~ Before + condition,
            data = dw)

  # Load package
  library(broom)

  # Figure 4
  ggplot(tidy(fit), aes(
      x = term,
      y = estimate,
      ymin = estimate - 1.96*std.error,
      ymax = estimate + 1.96*std.error)
    ) +
    geom_hline(
      yintercept = 0,
      linetype = "dashed“
    ) +
    geom_pointrange() +
    coord_flip()
```



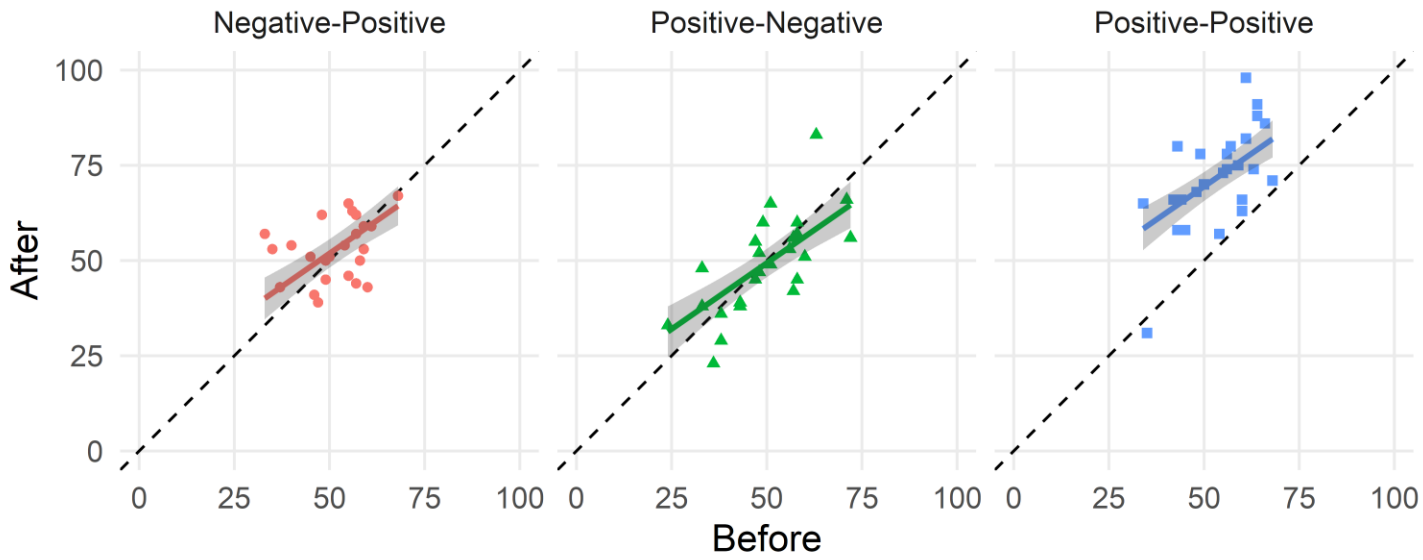
```r
# Example 5: Themes and Labels ----------------

  # Figure 5
  fig <- fig +
    geom_line(
      data = augment(fit),
      aes(y = .fitted),
      size = 1
    ) +
    geom_ribbon(
      data = augment(fit),
      aes(ymin = .fitted - 1.96*.se.fit,
          ymax = .fitted + 1.96*.se.fit),
      size = 1,
      colour = NA,
      alpha = 0.25
    )
```



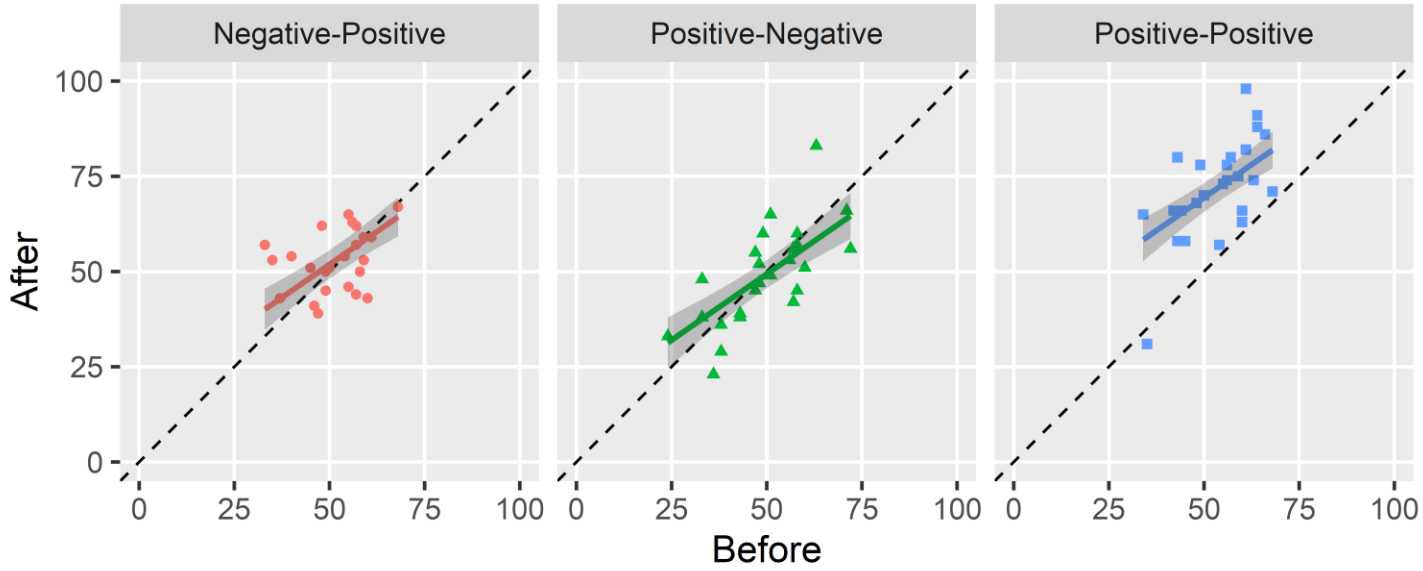
```r
# Example 5: Themes and Labels -----------------

  # Figure 5
  fig +
    theme_grey(
      base_size = 14, base_line_size = 0.5
    ) +
  theme(legend.position = "none")
```



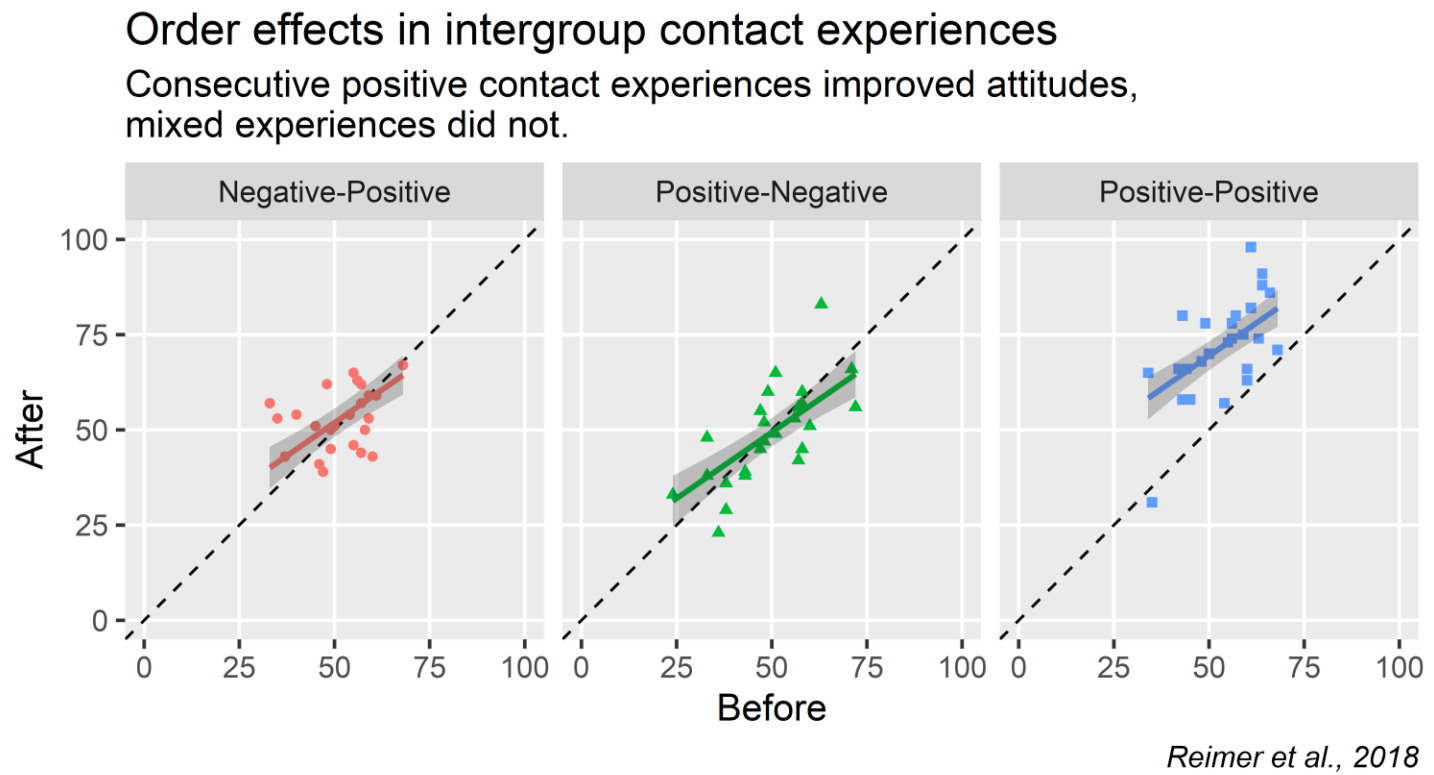
```r
# Example 5: Themes and Labels -----------------

  # Figure 5
  fig +
    theme_minimal(
      base_size = 14, base_line_size = 0.5
    ) +
  theme(legend.position = "none")
```



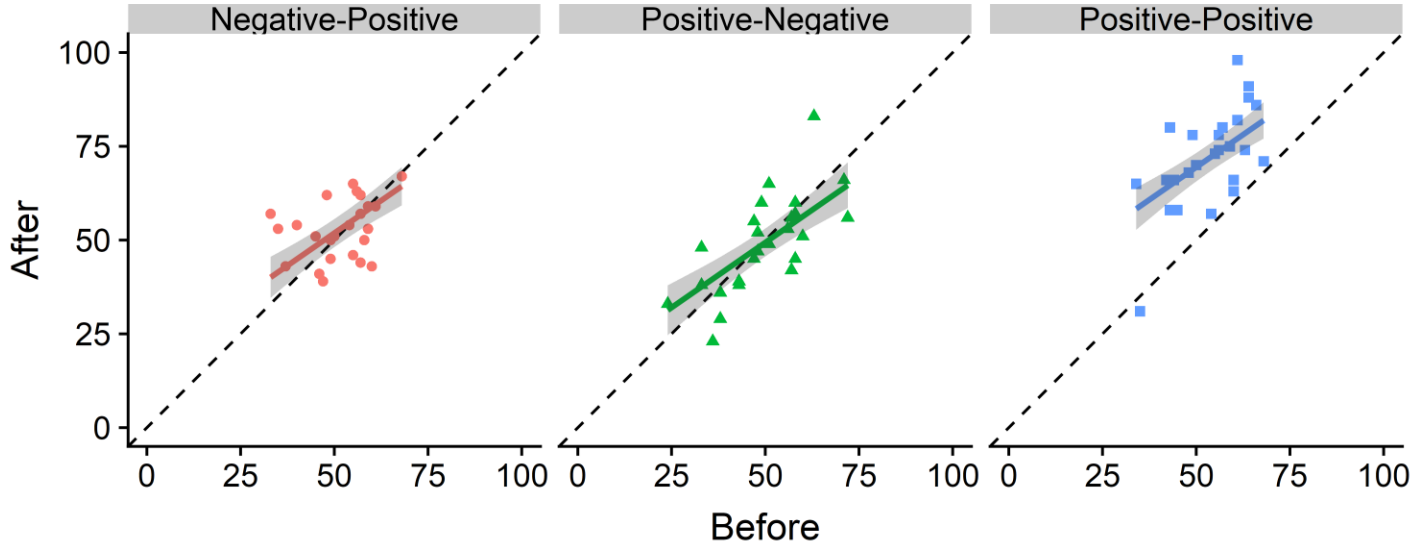
```r
# Example 5: Themes and Labels ----------------

  # Figure 5
  fig +
    labs(
      x = "Before",
      y = "After"
    ) +
    theme(legend.position = "none")
```



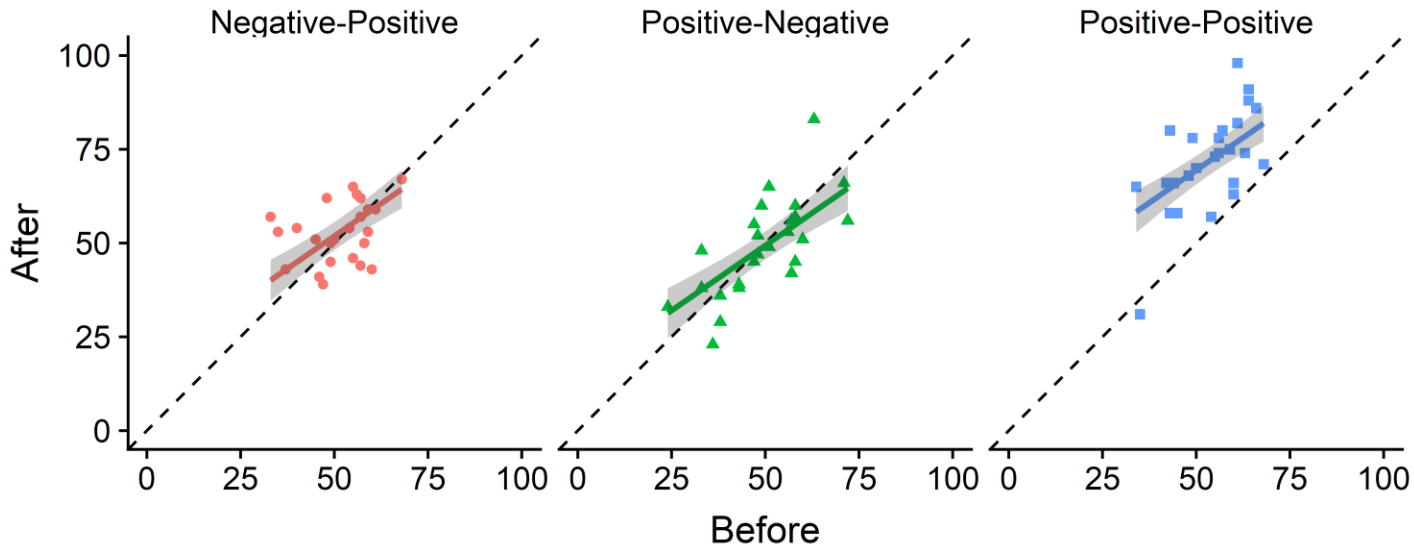
```r
# Example 5: Themes and Labels ----------------

  # Figure 5
  fig +
    labs(
      title = "Order effects in intergroup
               contact experiences",
      subtitle = "Consecutive positive contact
                  experiences improved
                  attitudes,\nmixed experiences
                  did not.",
      caption = expression(italic("Reimer et
                                  al., 2018"))
    ) +
    theme(legend.position = "none")
```



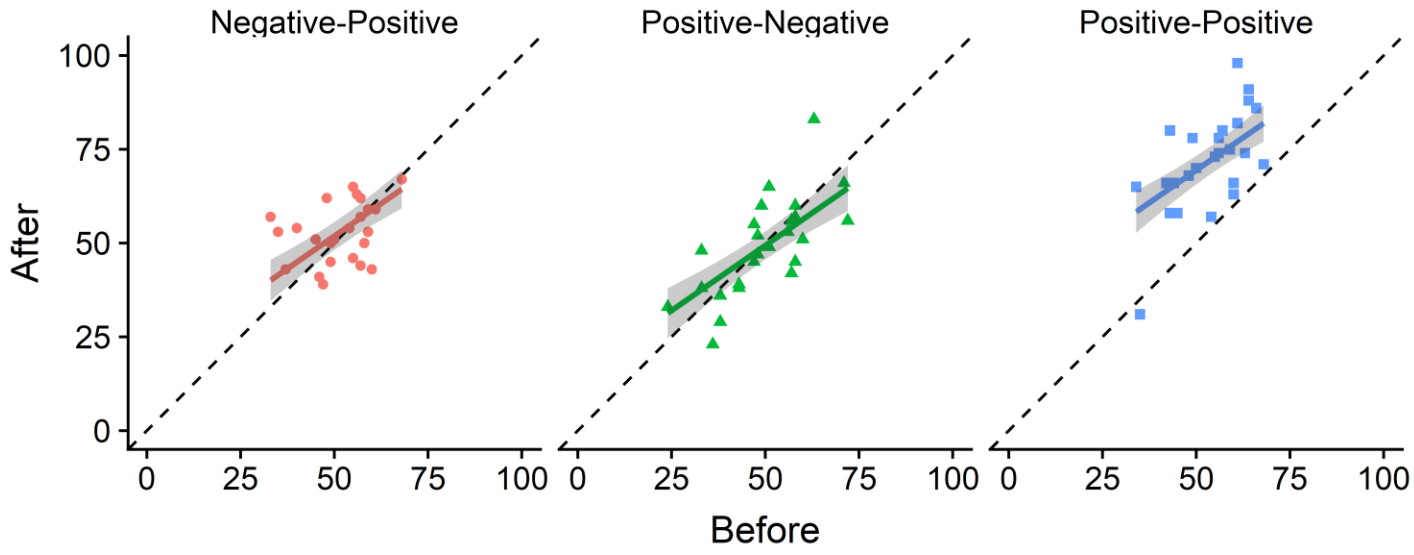
```r
# Example 5: Themes and Labels -----------------

  # Figure 5
  fig +
    cowplot::theme_cowplot(font_size = 14) +
    theme(legend.position = "none")
```



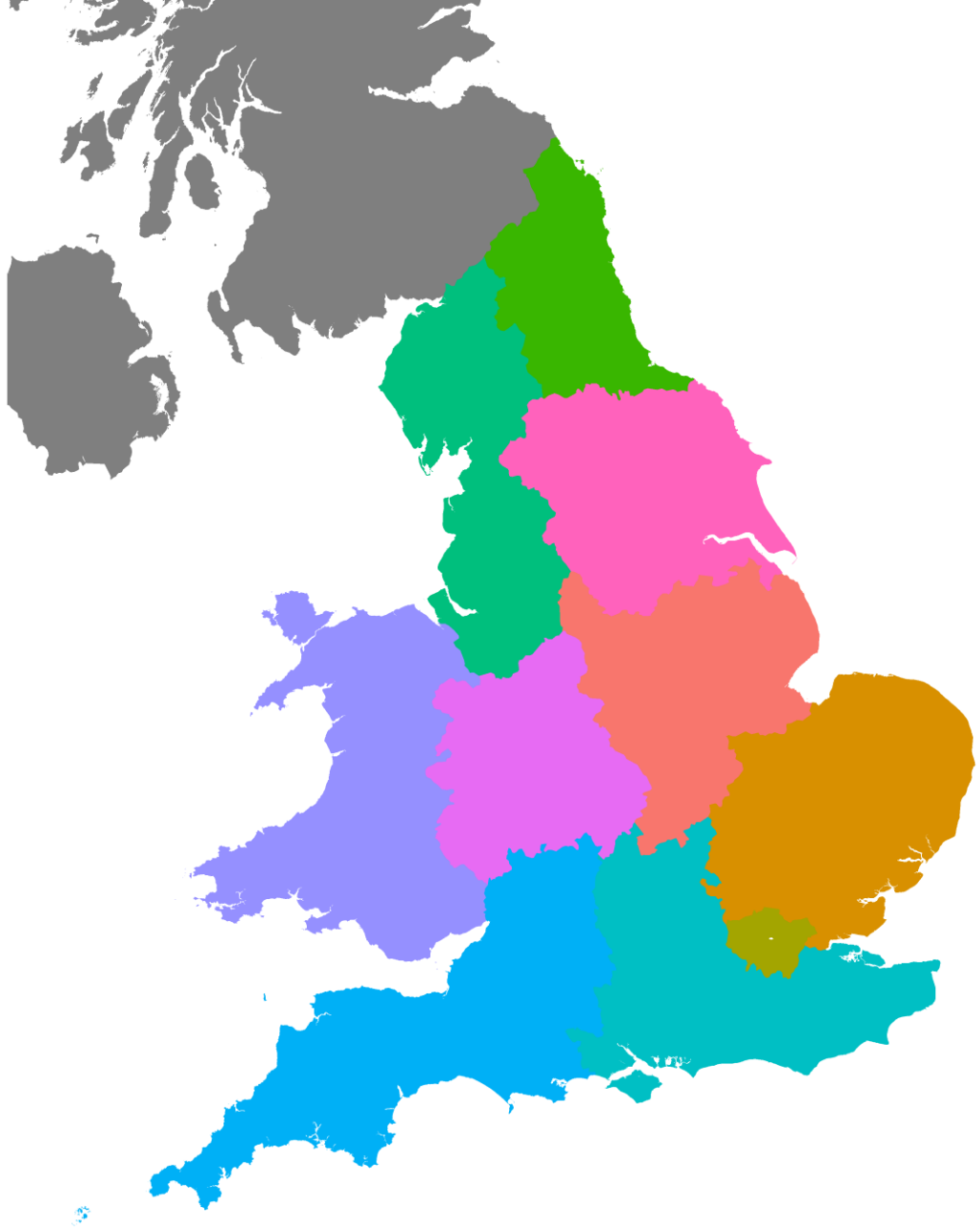
```r
# Example 5: Themes and Labels -----------------

  # Figure 5
  fig +
    cowplot::theme_cowplot(font_size = 14) +
    theme(
      legend.position = "none",
      strip.background = element_blank()
    )
```



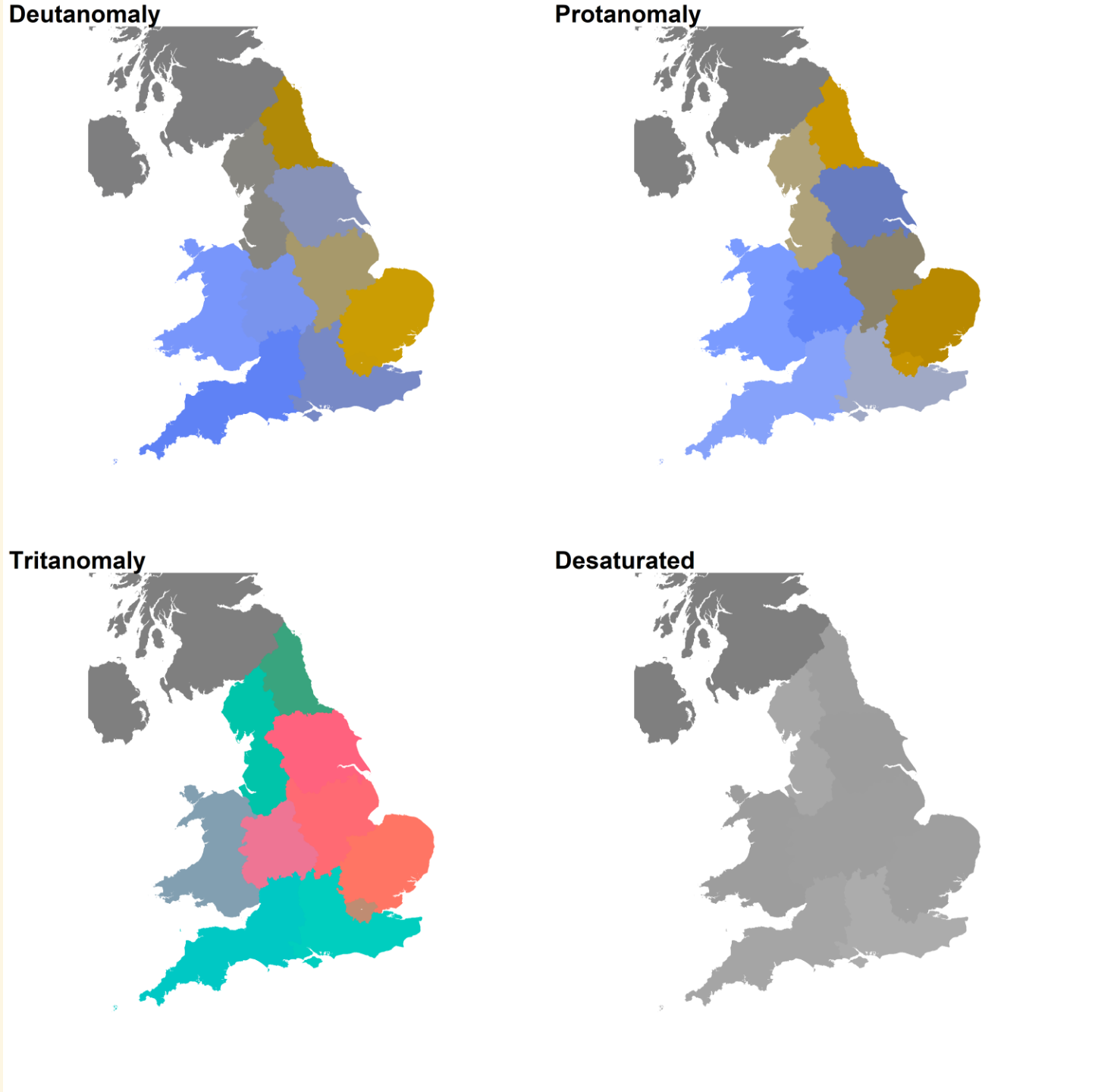
```r
# Example 5: Themes and Labels ----------------

  # Figure 5
  fig +
    cowplot::theme_cowplot(font_size = 14) +
    theme(
      legend.position = "none",
      strip.background = element_blank()
    )

  # Export
  ggsave(
    "figures/f5.png",
    width = 19.05, height = 19.05, units = "cm",
    dpi = 600,
    type = "cairo-png"
  )
```



```r
# Example 6: Colours --------------------------

  # Import data
  dm <- read_rds("materials/d5.rds")

  # Figure 6
  ggplot(dm, aes(
      x = long, y = lat, group = group
    )) +
    geom_polygon(fill = "grey50") +
    geom_polygon(aes(fill = region)) +
    coord_map(
      xlim = c(-7, 2.5),
      ylim = c(49, 56.5)
    ) +
    theme_void() +
    theme(legend.position = "none")
```

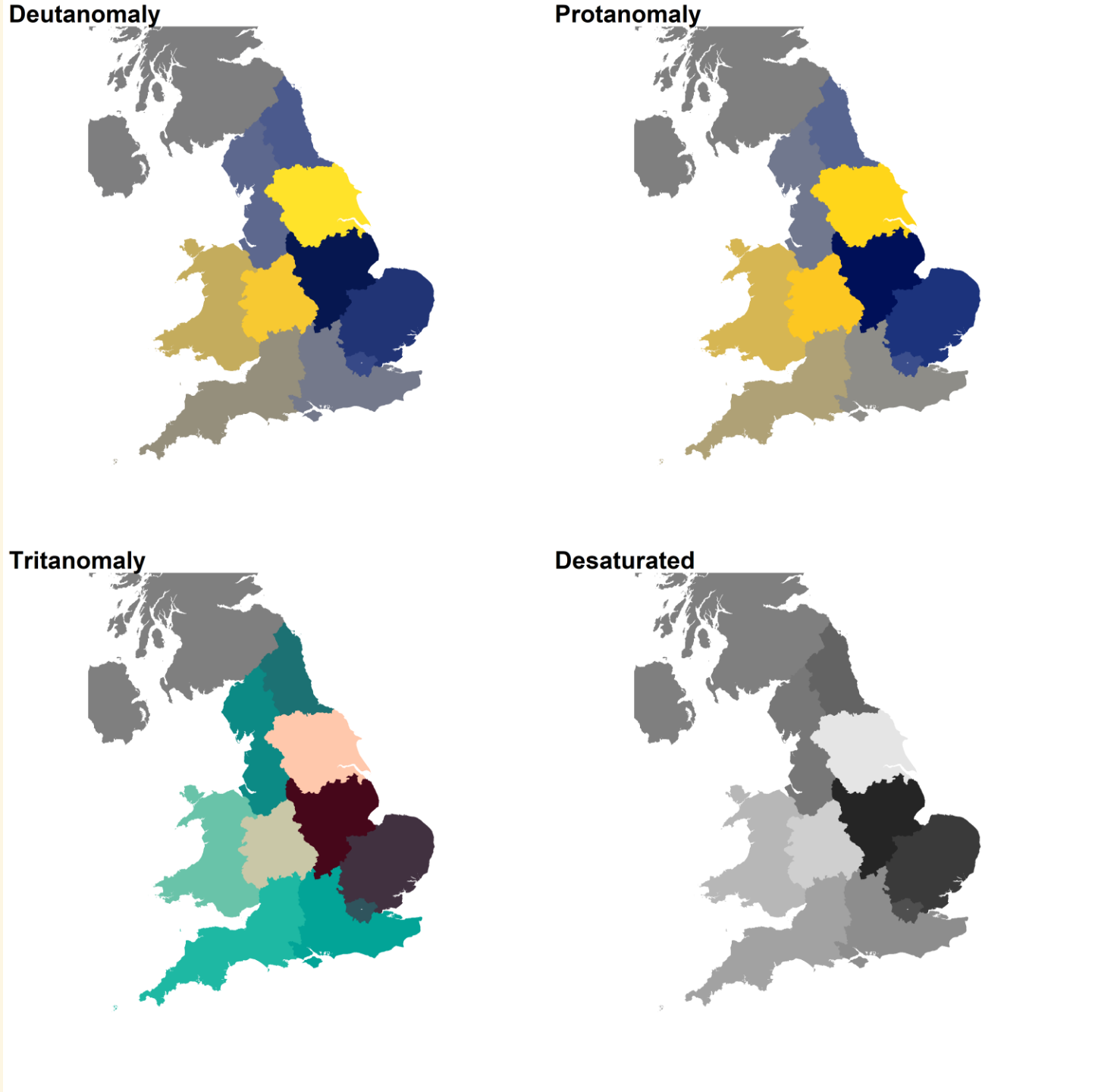


```r
# Example 6: Colours ---------------------------

  # Import data
  dm <- read_rds("materials/d5.rds")

  # Figure 6
  ggplot(dm, aes(
      x = long, y = lat, group = group
    )) +
    geom_polygon(fill = "grey50") +
    geom_polygon(aes(fill = region)) +
    coord_map(
      xlim = c(-7, 2.5),
      ylim = c(49, 56.5)
    ) +
    theme_void() +
    theme(legend.position = "none")

  # Simulate colour-vision deficiencies
  colorblindr::cvd_grid()
```

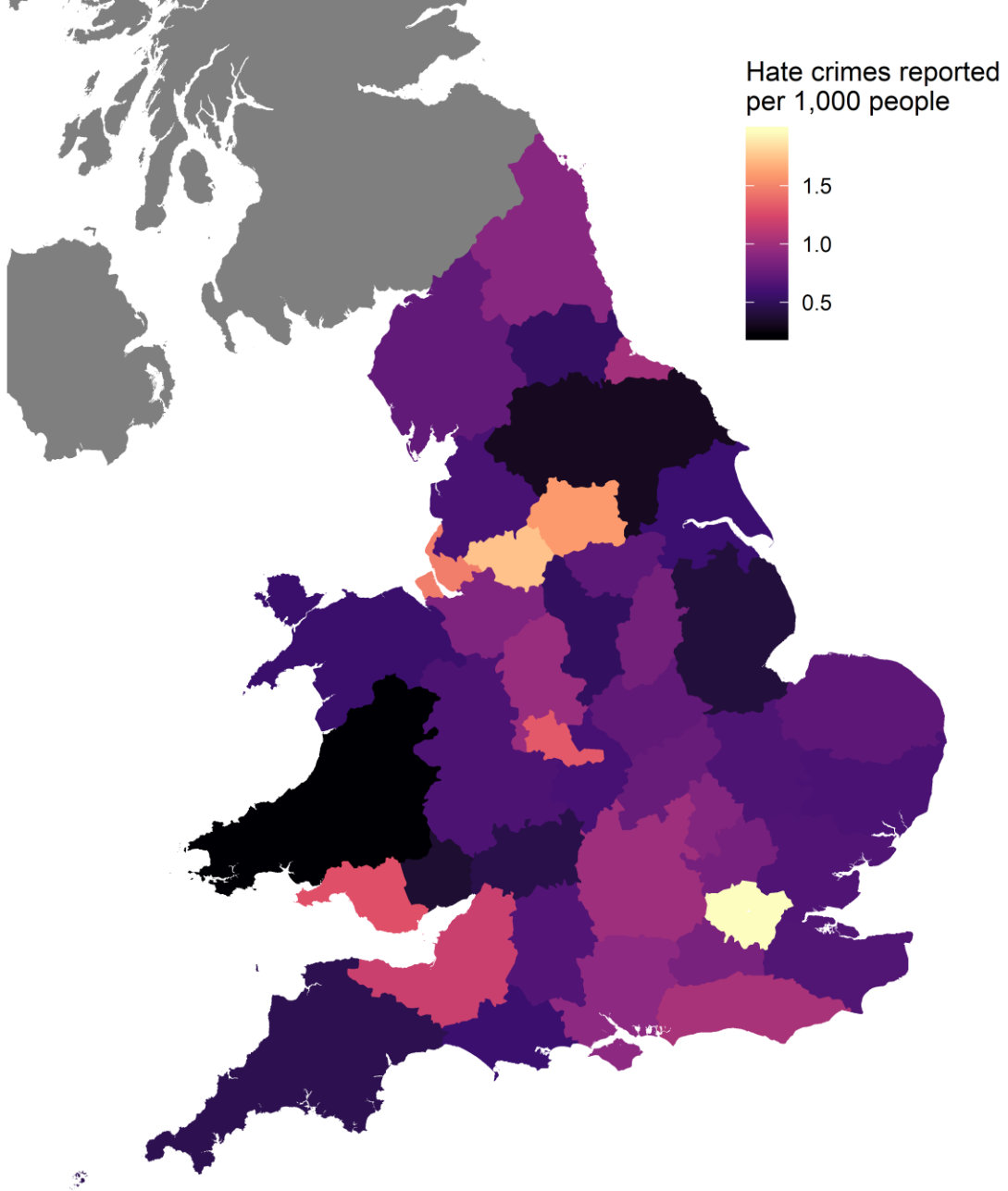


```r
# Example 6: Colours --------------------------

  # Import data
  dm <- read_rds("materials/d5.rds")

  # Figure 6
  ggplot(dm, aes(
      x = long, y = lat, group = group
    )) +
    geom_polygon(fill = "grey50") +
    geom_polygon(aes(fill = region)) +
    scale_fill_viridis_d() +
    coord_map(
      xlim = c(-7, 2.5),
      ylim = c(49, 56.5)
    ) +
    theme_void() +
    theme(legend.position = "none")
```

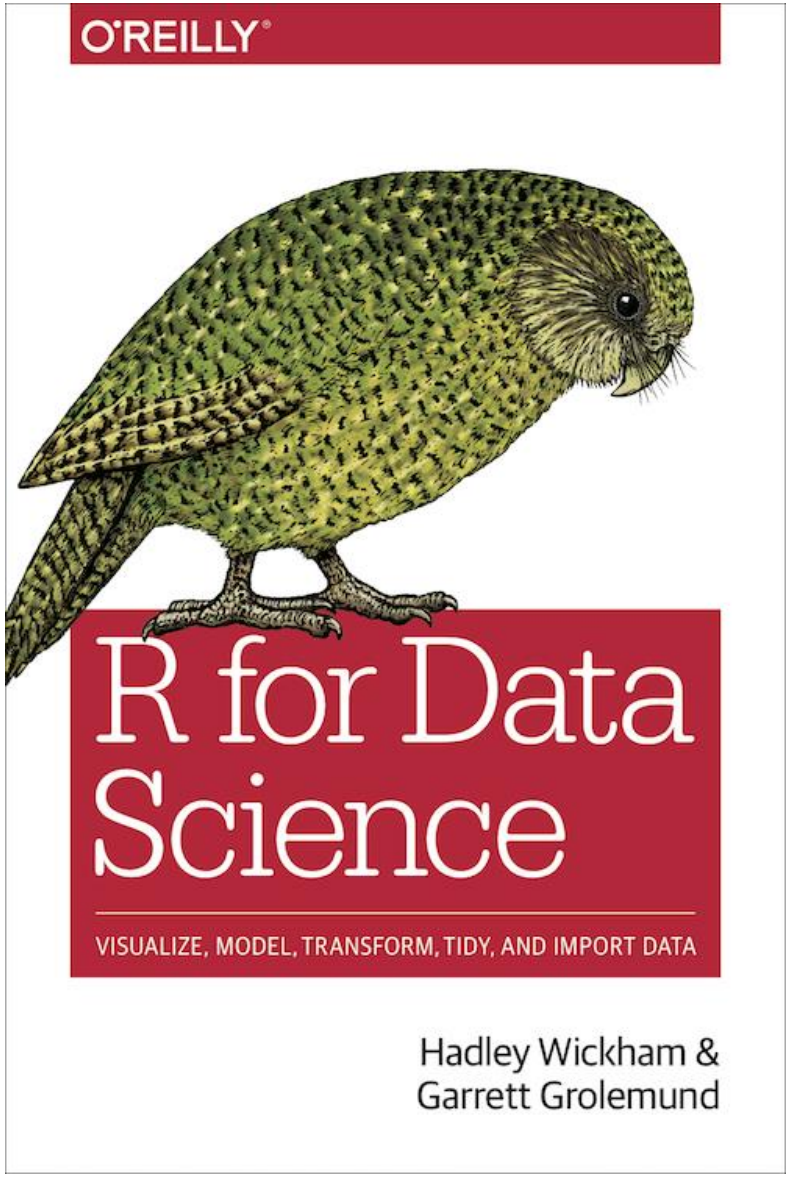
```r
# Example 6: Colours ---------------------------

  # Import data
  dm <- read_rds("materials/d5.rds")

  # Figure 6
  ggplot(dm, aes(
      x = long, y = lat, group = group
    )) +
    geom_polygon(fill = "grey50") +
    geom_polygon(aes(fill = region)) +
    scale_fill_viridis_d() +
    coord_map(
      xlim = c(-7, 2.5),
      ylim = c(49, 56.5)
    ) +
    theme_void() +
    theme(legend.position = "none")

  # Simulate color-vision deficiencies
  colorblindr::cvd_grid()
```



```r
# Example 6: Colours ---------------------------

  # Import data
  dm <- read_rds("materials/d5.rds")

  # Figure 6
  ggplot(dm, aes(
      x = long, y = lat, group = group
    )) +
    geom_polygon(aes(fill = prop)) +
    scale_fill_viridis_c(option = "A") +
    coord_map(
      xlim = c(-7, 2.5),
      ylim = c(49, 56.5)
    ) +
    theme_void() +
    theme(legend.position = c(.85, .85)) +
    labs(
      fill = "Hate crimes reported\nper 1,000
people"
    ) +
```



```
?ggplot
?geom_point
```

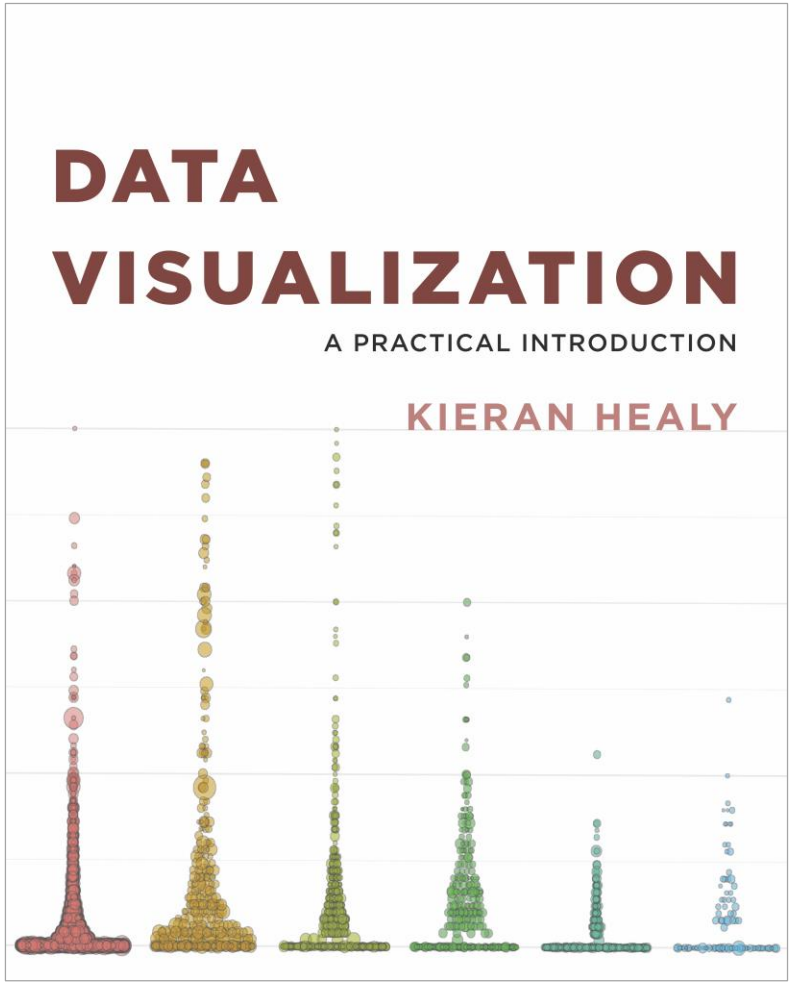


http://r4ds.had.co.nz/

Series: Get with the Plot!

09/10/2018	Week 1: A First Plot
16/10/2018	Week 2: Facets and Curves
23/10/2018	Week 3: #BarBarPlots
30/10/2018	Week 4: Themes and Labels
13/11/2018	Week 5: Annotations

https://nilsreimer.com/



http://socviz.co/

```
citation("ggplot2")
```

Deviation

Emphasise variations (+/-) from a fixed reference point. Typically the reference point is zero but it can also be a target or a long-term average. Can also be used to show sentiment (positive/neutral/negative).

Example FT uses
Trade surplus/deficit, climate change

Diverging bar

A simple standard bar chart that can handle both negative and positive magnitude values.

Diverging stacked bar

Perfect for presenting survey results which involve sentiment (eg disagree/neutral/agree).

Spine chart

Splits a single value into two contrasting components (eg male/female).

Surplus/deficit filled line

The shaded area of these charts allows a balance to be shown – either against a baseline or between two series.

Correlation

Show the relationship between two or more variables. Be mindful that unless you tell them otherwise, many readers will assume the relationships you show them to be causal (i.e. one causes the other).

Example FT uses
Inflation & unemployment, income & life expectancy

Scatterplot

The standard way to show the relationship between two continuous variables, each of which has its own axis.

Line + Column

A good way of showing the relationship between an amount (columns) and a rate (line).

Connected scatterplot

Usually used to show how the relationship between 2 variables has changed over time.

Bubble

Like a scatterplot, but adds additional detail by sizing the circles according to a third variable.

XY heatmap

A good way of showing the patterns between 2 categories of data, less good at showing fine differences in amounts.

Ranking

Use where an item's position in an ordered list is more important than its absolute or relative value. Don't be afraid to highlight the points of interest.

Example FT uses
Wealth, deprivation, league tables, constituency election results

Ordered bar

Standard bar charts display the ranks of values much more easily when sorted into order.

Ordered column

See above.

Ordered proportional symbol

Use when there are big variations between values and/or seeing fine differences between data is not so important.

Dot strip plot

Dots placed in order on a strip are a space-efficient method of laying out ranks across multiple categories.

Slope

Perfect for showing how ranks have changed over time or vary between categories.

Lollipop chart

Lollipops draw more attention to the data value than standard bar/column and can also show rank and value effectively.

Distribution

Show values in a dataset and how often they occur. The shape (or 'skew') of a distribution can be a memorable way of highlighting the lack of uniformity or equality in the data.

Example FT uses
Income distribution, population (age/sex) distribution

Histogram

The standard way to show a statistical distribution - keep the gaps between columns small to highlight the 'shape' of the data.

Boxplot

Summarise multiple distributions by showing the median (centre) and range of the data

Violin plot

Similar to a box plot but more effective with complex distributions (data that cannot be summarised with simple average).

Population pyramid

A standard way for showing the age and sex breakdown of a population distribution, effectively, back to back histograms.

Dot strip plot

Good for showing individual values in a distribution, can be a problem when too many dots have the same value.

Dot plot

A simple way of showing the change or range (min/max) of data across multiple categories.

Barcode plot

Like dot strip plots, good for displaying all the data in a table, they work best when highlighting individual values.

Cumulative curve

A good way of showing how unequal a distribution is; y axis is always cumulative frequency, x axis is always a measure.

Change over Time

Give emphasis to changing trends. These can be short (intra-day) movements or extended series traversing decades or centuries: Choosing the correct time period is important to provide suitable context for the reader.

Example FT uses
Share price movements, economic time series

Line

The standard way to show a changing time series. If data are irregular, consider markers to represent data points.

Column

Columns work well for showing change over time - but usually best with only one series of data at a time.

Line + column

A good way of showing the relationship over time between an amount (columns) and a rate (line).

Stock price

Usually focused on day-to-day activity, these charts show opening/closing and high/low points of each day

Slope

Good for showing changing data as long as the data can be simplified into 2 or 3 points without missing a key part of story.

Area chart

Use with care – these are good at showing changes to total, but seeing change in components can be very difficult

Fan chart (projections)

Use to show the uncertainty in future projections - usually this grows the further forward to projection.

Connected scatterplot

A good way of showing changing data for two variables whenever there is a relatively clear pattern of progression.

Calendar heatmap

A great way of showing temporal patterns (daily, weekly, monthly) – at the expense of showing precision in quantity.

Priestley timeline

Great when date and duration are key elements of the story in the data.

Circle timeline

Good for showing discrete values of varying size across multiple categories (eg earthquakes by continent).

Vertical timeline

Presents time on the Y axis. Good for displaying detailed time series that work especially well when scrolling on mobile.

Seismogram

Another alternative to the circle timeline for showing series where there are big variations in the data.

Magnitude

Show size comparisons. These can be relative (just being able to see larger/bigger) or absolute (need to see fine differences). Usually these show a 'counted' number (for example, barrels, dollars or people) rather than a calculated rate or per cent.

Example FT uses
Commodity production, market capitalisation

Column

The standard way to compare the size of things. Must always start at 0 on the axis.

Bar

See above. Good when the data are not time series and labels have long category names.

Paired column

As per standard column but allows for multiple series. Can become tricky to read with more than 2 series.

Paired bar

See above.

Proportional stacked bar

A good way of showing the size and proportion of data at the same time – as long as the data are not too complicated.

Proportional symbol

Use when there are big variations between values and/or seeing fine differences between data is not so important.

Isotype (pictogram)

Excellent solution in some instances – use only with whole numbers (do not slice off an arm to represent a decimal).

Lollipop chart

Lollipop charts draw more attention to the data value than standard bar/column – does not have to start at zero (but preferable).

Radar chart

A space-efficient way of showing value of multiple variables – but make sure they are organised in a way that makes sense to reader.

Parallel coordinates

An alternative to radar charts – again, the arrangement of the variables is important. Usually benefits from highlighting values.

Bullet chart

Good for showing a measurement against the context of a target or performance range.

Grouped symbol

An alternative to bar/column charts when being able to count data or highlight individual elements is useful.

Part-to-whole

Show how a single entity can be broken down into its component elements. If the reader's interest is solely in the size of the components, consider a magnitude-type chart instead.

Example FT uses
Fiscal budgets, company structures, national election results

Stacked column

A simple way of showing part-to-whole relationships but can be difficult to read with more than a few components.

Proportional stacked bar

A good way of showing the size and proportion of data at the same time – as long as the data are not too complicated.

Pie

A common way of showing part-to-whole data – but be aware that it's difficult to accurately compare the size of the segments.

Donut

Similar to a pie chart – but the centre can be a good way of making space to include more information about the data (eg total).

Treemap

Use for hierarchical part-to-whole relationships can be difficult to read when there are many small segments.

Voronoi

A way of turning points into areas – any point within each area is closer to the central point than any other centroid.

Arc

A hemicycle, often used for visualising political results in parliaments.

Gridplot

Good for showing % information, they work best when used on whole numbers and work well in multiple layout form.

Venn

Generally only used for schematic representation.

Waterfall

Can be useful for showing part-to-whole relationships where some of the components are negative.

Spatial

Aside from locator maps only used when precise locations or geographical patterns in data are more important to the reader than anything else.

Example FT uses
Population density, natural resource locations, natural disaster risk/impact, catchment areas, variation in election results

Basic choropleth (rate/ratio)

The standard approach for putting data on a map – should always be rates rather than totals and use a sensible base geography.

Proportional symbol (count/magnitude)

Use for totals rather than rates – be wary that small differences in data will be hard to see.

Flow map

For showing unambiguous movement across a map.

Contour map

For showing areas of equal value on a map. Can use deviation colour schemes for showing +/- values

Equalised cartogram

Converting each unit on a map to a regular and equally-sized shape – good for representing voting regions with equal value.

Scaled cartogram (value)

Stretching and shrinking a map so that each area is sized according to a particular value.

Dot density

Used to show the location of individual events/locations – make sure to annotate any patterns the reader should see.

Heat map

Grid-based data values mapped with an intensity colour scale. As choropleth map – but not snapped to an admin/political unit.

Flow

Show the reader volumes or intensity of movement between two or more states or conditions. These might be logical sequences or geographical locations.

Example FT uses
Movement of funds, trade, migrants, lawsuits, information; relationship graphs.

Sankey

Shows changes in flows from one condition to at least one other; good for tracing the eventual outcome of a complex process.

Waterfall

Designed to show the sequencing of data through a flow process, typically budgets. Can include +/- components.

Chord

A complex but powerful diagram which can illustrate 2-way flows (and net winner) in a matrix.

Network

Used for showing the strength and inter-connectedness of relationships of varying types.

Visual vocabulary

Designing with data

There are so many ways to visualise data - how do we know which one to pick? Use the categories across the top to decide which data relationship is most important in your story, then look at the different types of chart within the category to form some initial ideas about what might work best. This list is not meant to be exhaustive, nor a wizard, but is a useful starting point for making informative and meaningful data visualisations.

FT graphic: Alan Smith; Chris Campbell; Ian Bott; Liz Faunce; Graham Parrish; Billy Ehrenberg; Paul McCallum; Martin Stabe
Inspired by the Graphic Continuum by Jon Schwabish and Severino Ribecca

ft.com/vocabulary