



DATA VISUALIZATION WITH GGPLOT2

Introduction

Your Instructor - Rick Scavetta

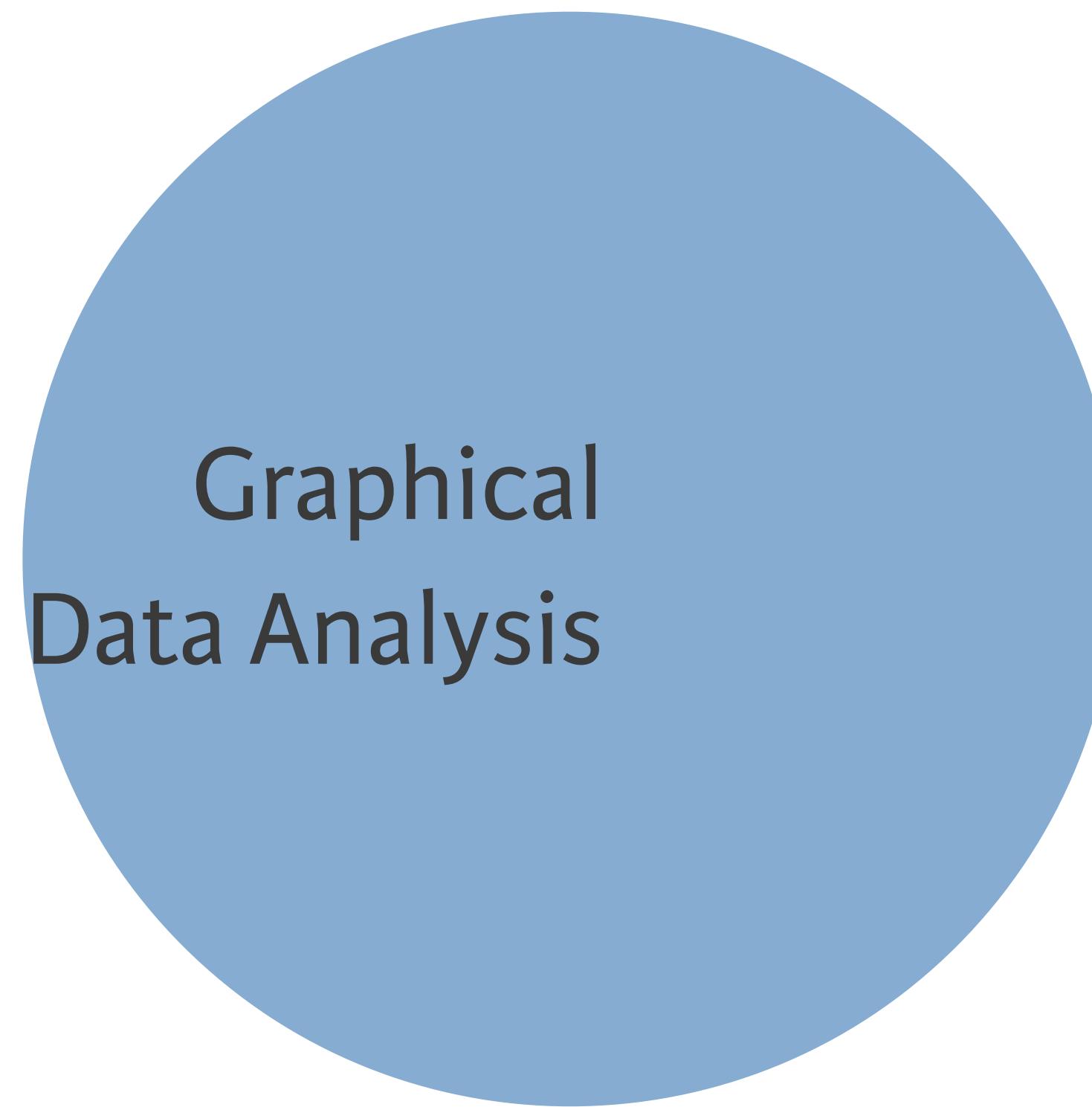
- rick.scavetta@science-craft.com
- @Rick_Scavetta

Data Visualization

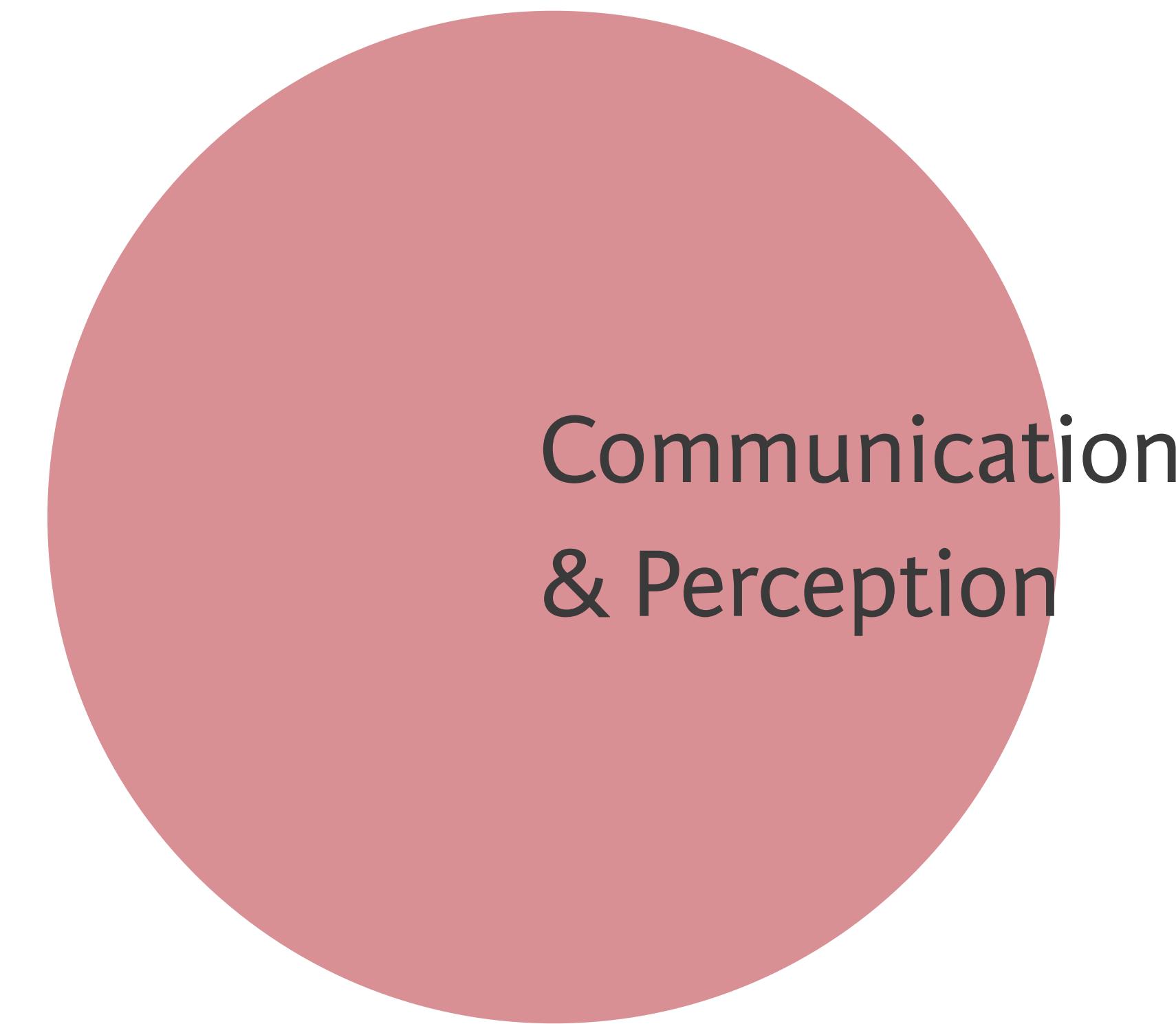
Data Visualization & Data Science

Data Visualization & Data Science

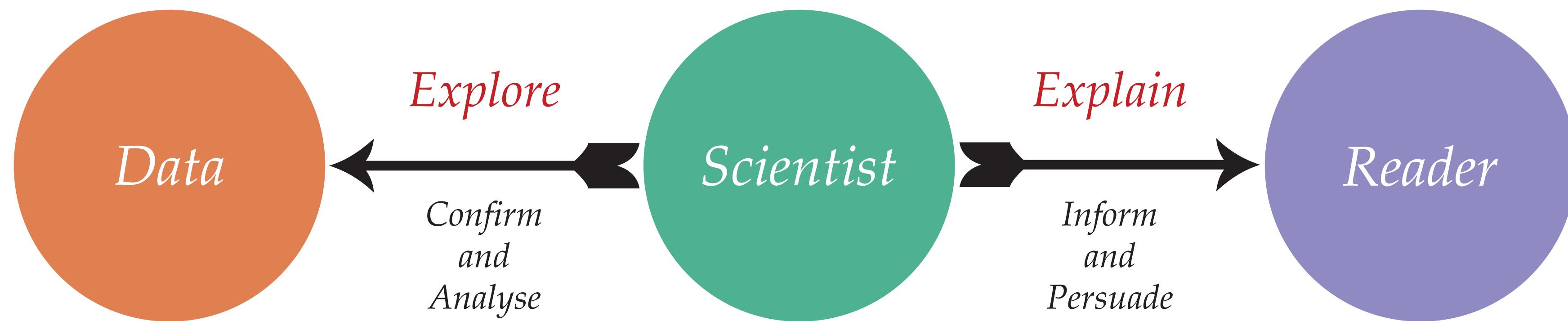
Statistics



Design



Exploratory versus Explanatory



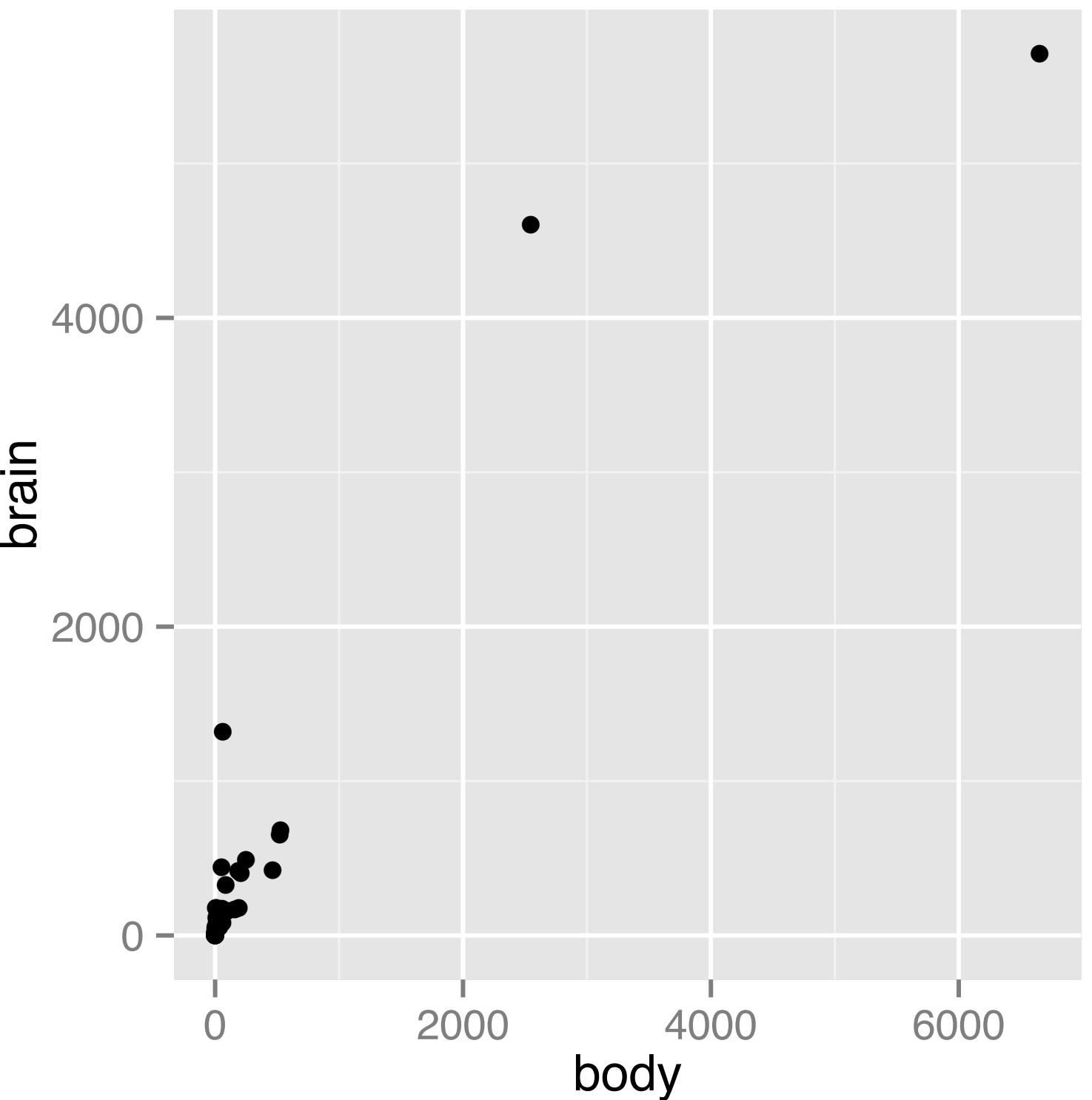
MASS::mammals

```
> library(MASS)
> mammals
```

	body	brain
Arctic fox	3.385	44.50
Owl monkey	0.480	15.50
Mountain beaver	1.350	8.10
Cow	465.000	423.00
Grey wolf	36.330	119.50
Goat	27.660	115.00
Roe deer	14.830	98.20
...		
Pig	192.000	180.00
Echidna	3.000	25.00
Brazilian tapir	160.000	169.00
Tenrec	0.900	2.60
Phalanger	1.620	11.40
Tree shrew	0.104	2.50
Red fox	4.235	50.40

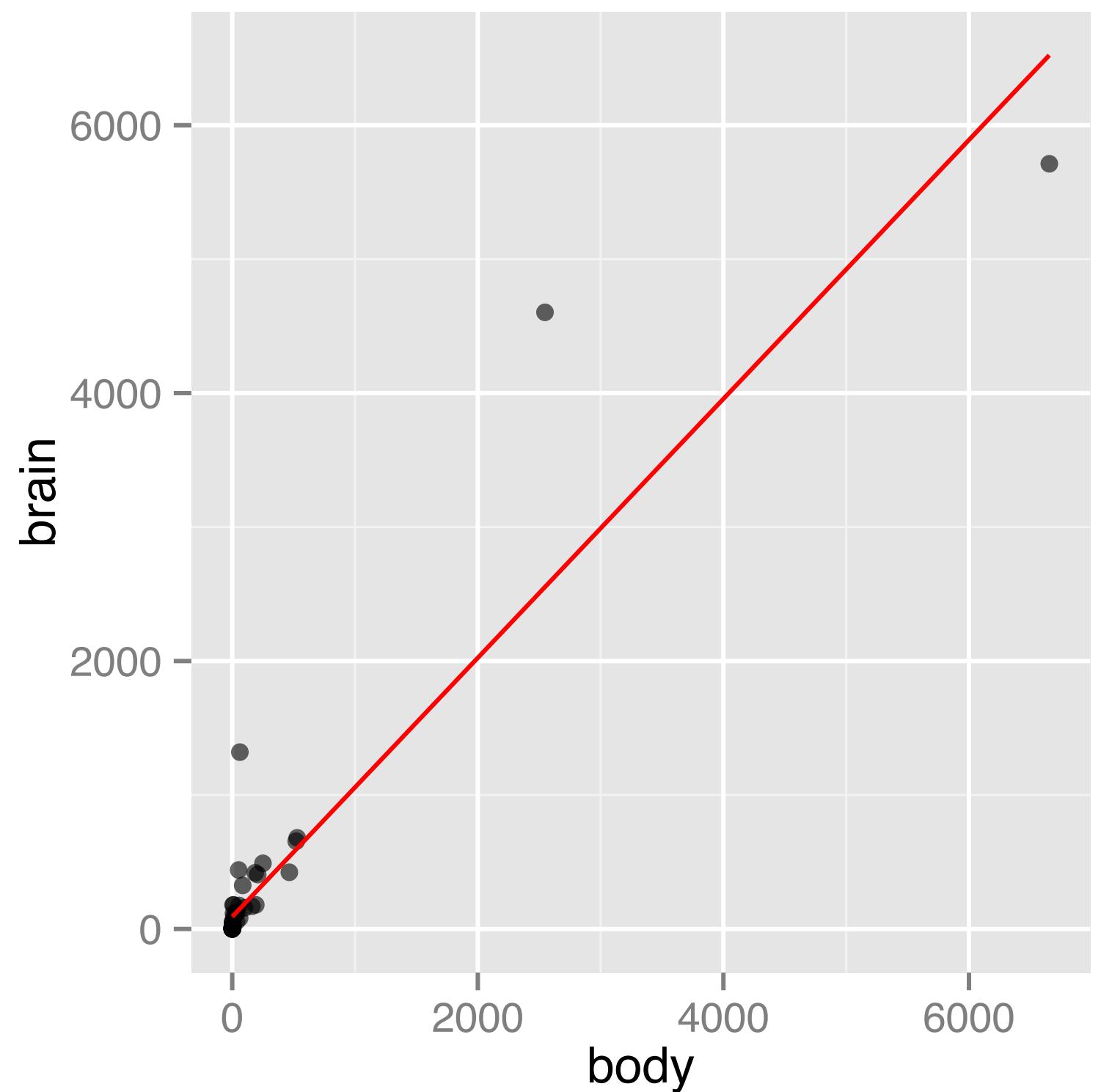
Scatter Plot

```
> library(ggplot2)
> ggplot(mammals, aes(x = body, y = brain)) +
  geom_point()
```



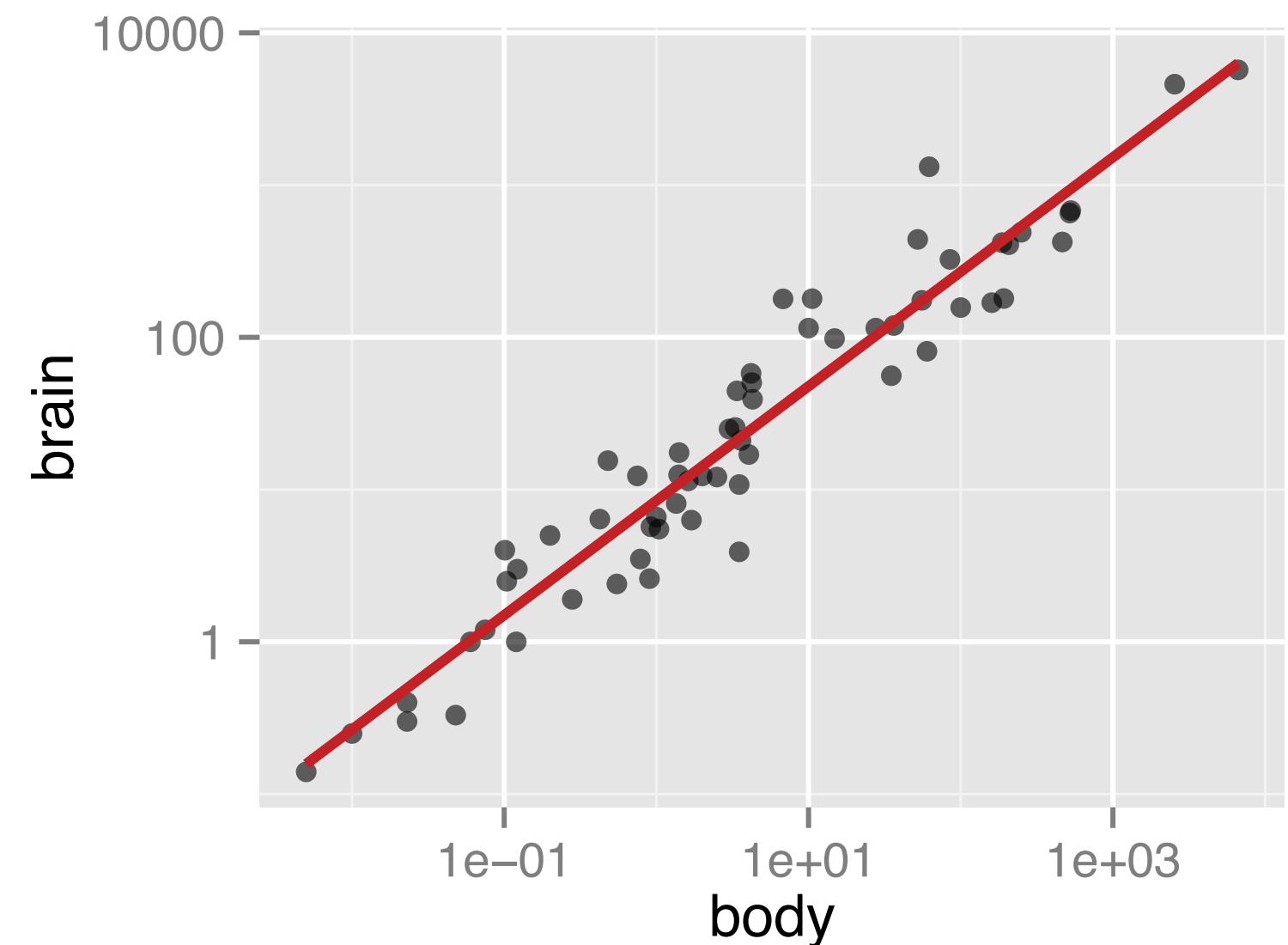
Explore - Statistics

```
> ggplot(mammals, aes(x = body, y = brain)) +  
  geom_point(alpha = 0.6) +  
  stat_smooth(method = "lm", col = "red", se = FALSE)
```



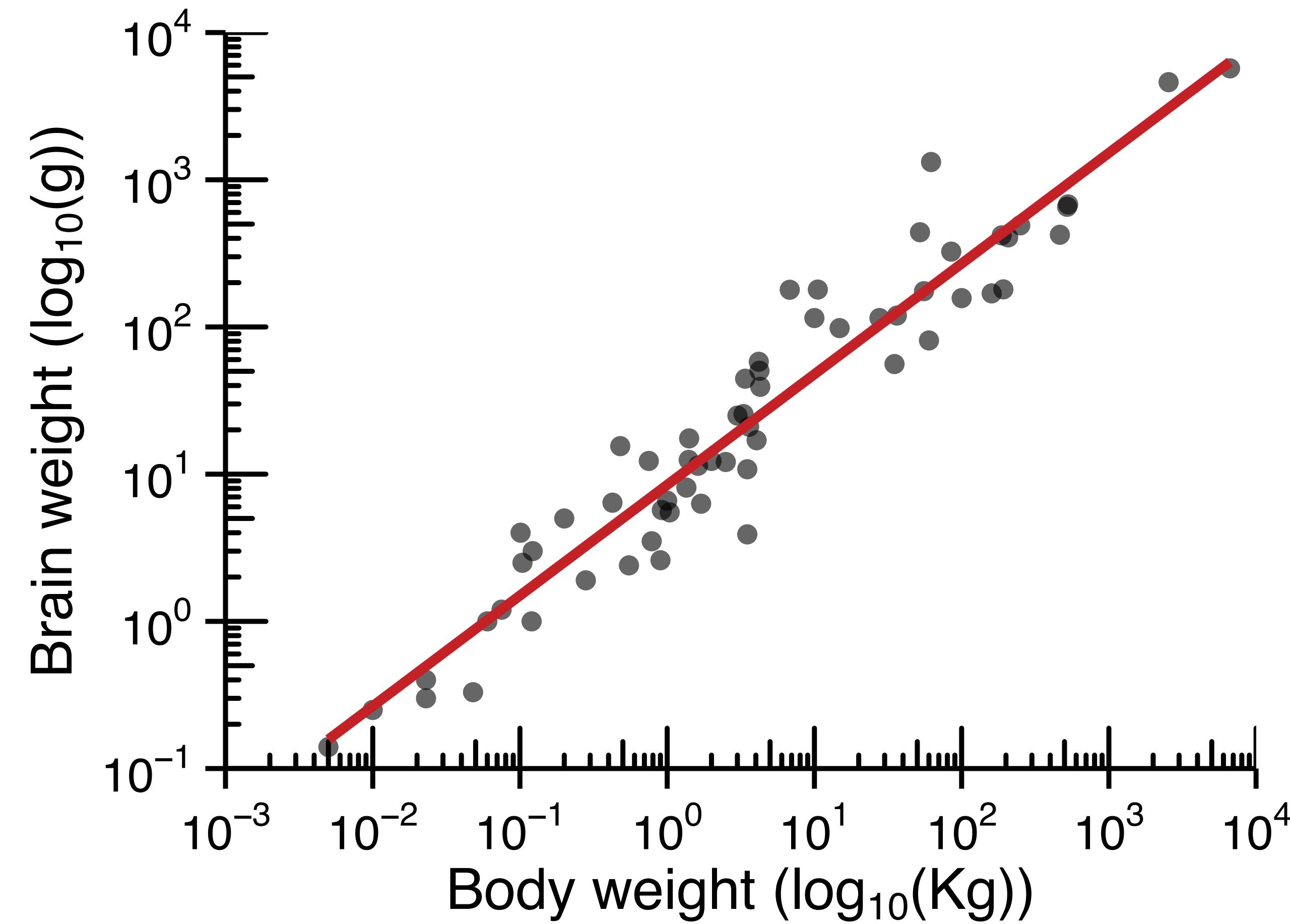
Explore - Fine-tuning

```
> ggplot(mammals, aes(x = body, y = brain)) +  
  geom_point(alpha = 0.6) +  
  coord_fixed() +  
  scale_x_log10() +  
  scale_y_log10() +  
  stat_smooth(method = "lm",  
              col = "#C42126",  
              se = FALSE, size = 1)
```

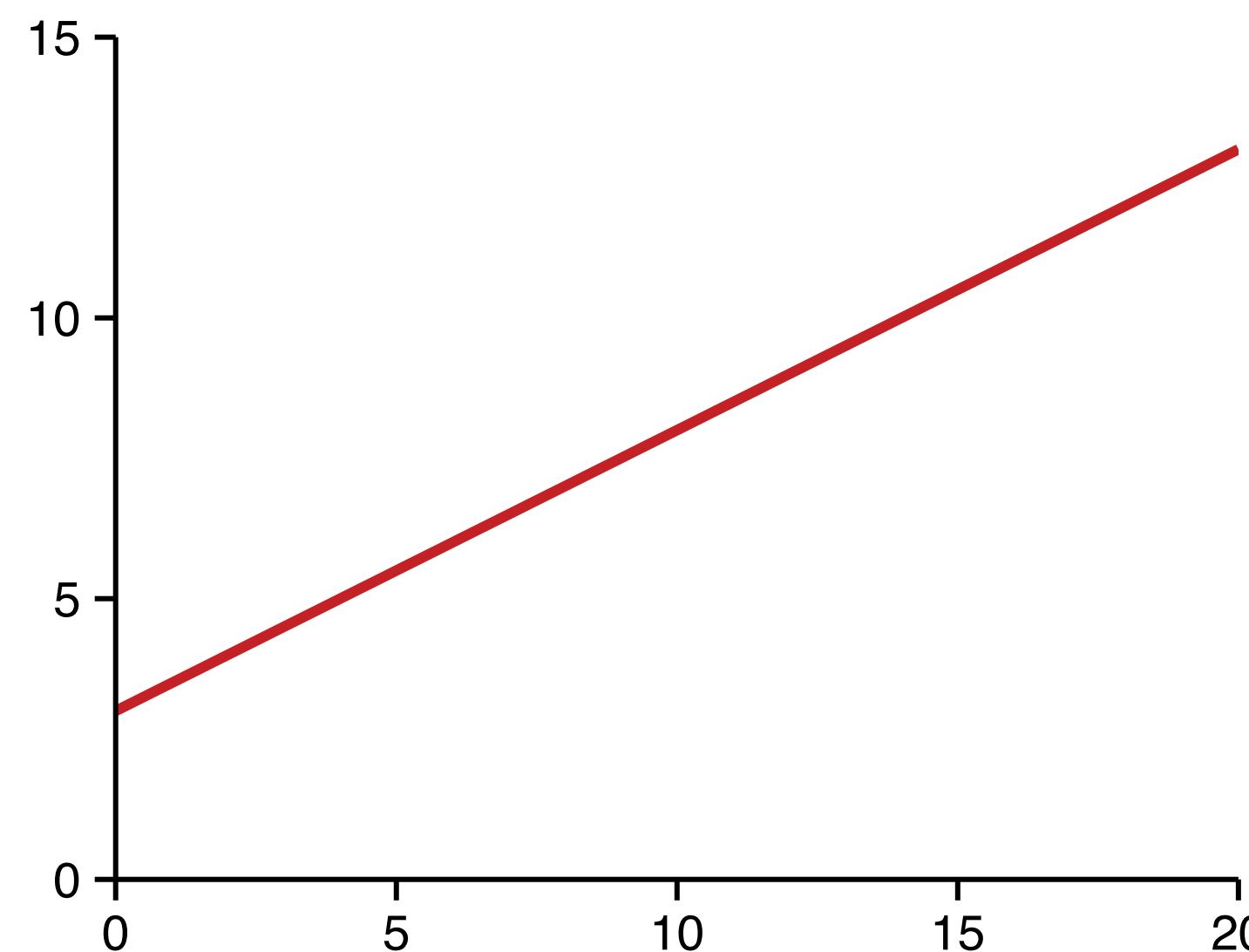


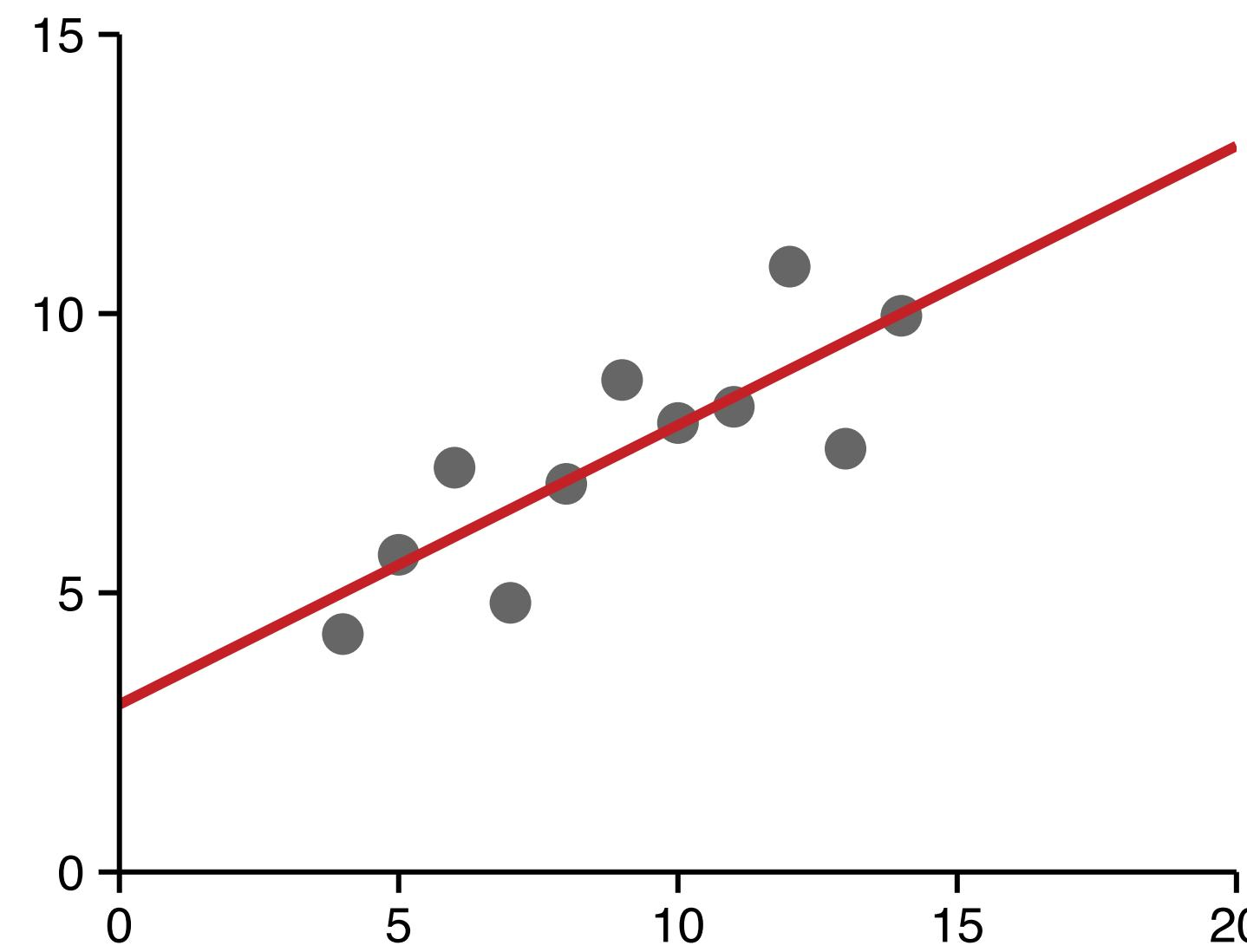
Publication-ready plot

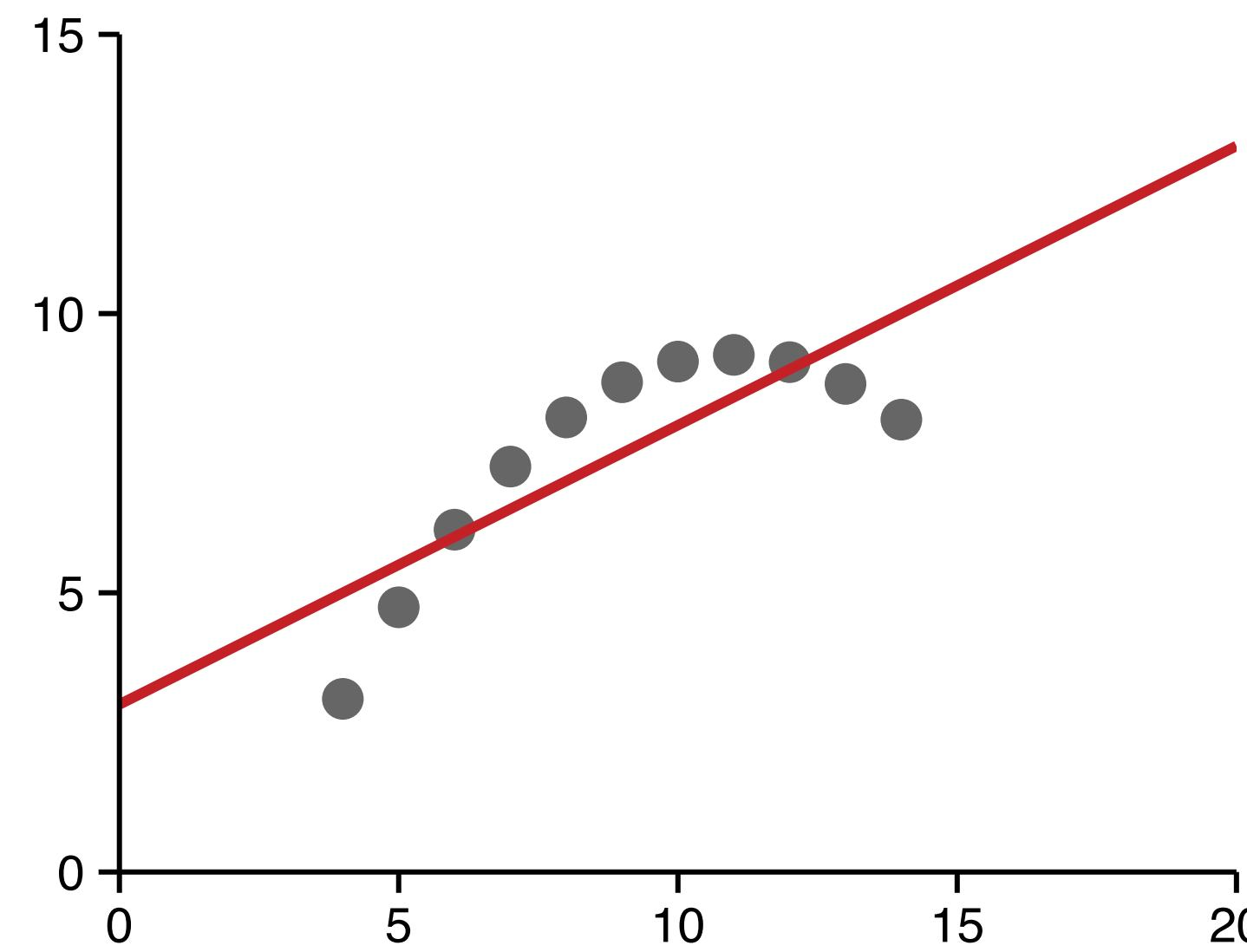
```
> library(scales) # functions trans_breaks and trans_format
> ggplot(mammals, aes(x = body, y = brain)) +
  annotation_logticks() +
  geom_point(alpha = 0.6) +
  coord_fixed(xlim = c(10^-3, 10^4), ylim = c(10^-1, 10^4)) +
  scale_x_log10(expression("Body weight (log"[10]"*(Kg))"),
    breaks = trans_breaks("log10", function(x) 10^x),
    labels = trans_format("log10", math_format(10^.x))) +
  scale_y_log10(expression("Brain weight (log"[10]"*(g))"),
    breaks = trans_breaks("log10", function(x) 10^x),
    labels = trans_format("log10", math_format(10^.x))) +
  stat_smooth(method = "lm", col = "#C42126", se = FALSE, size = 1) +
  theme_classic()
```

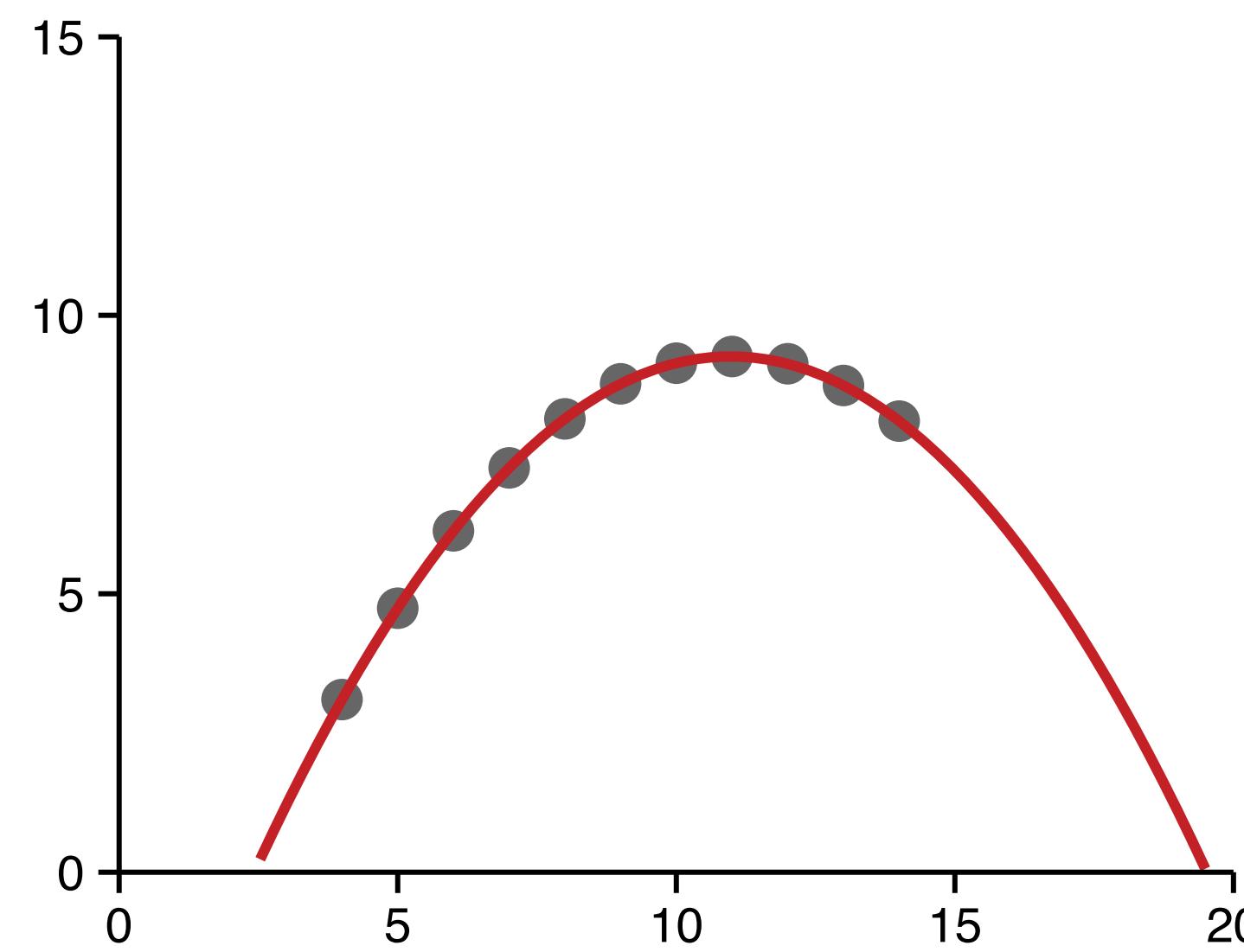


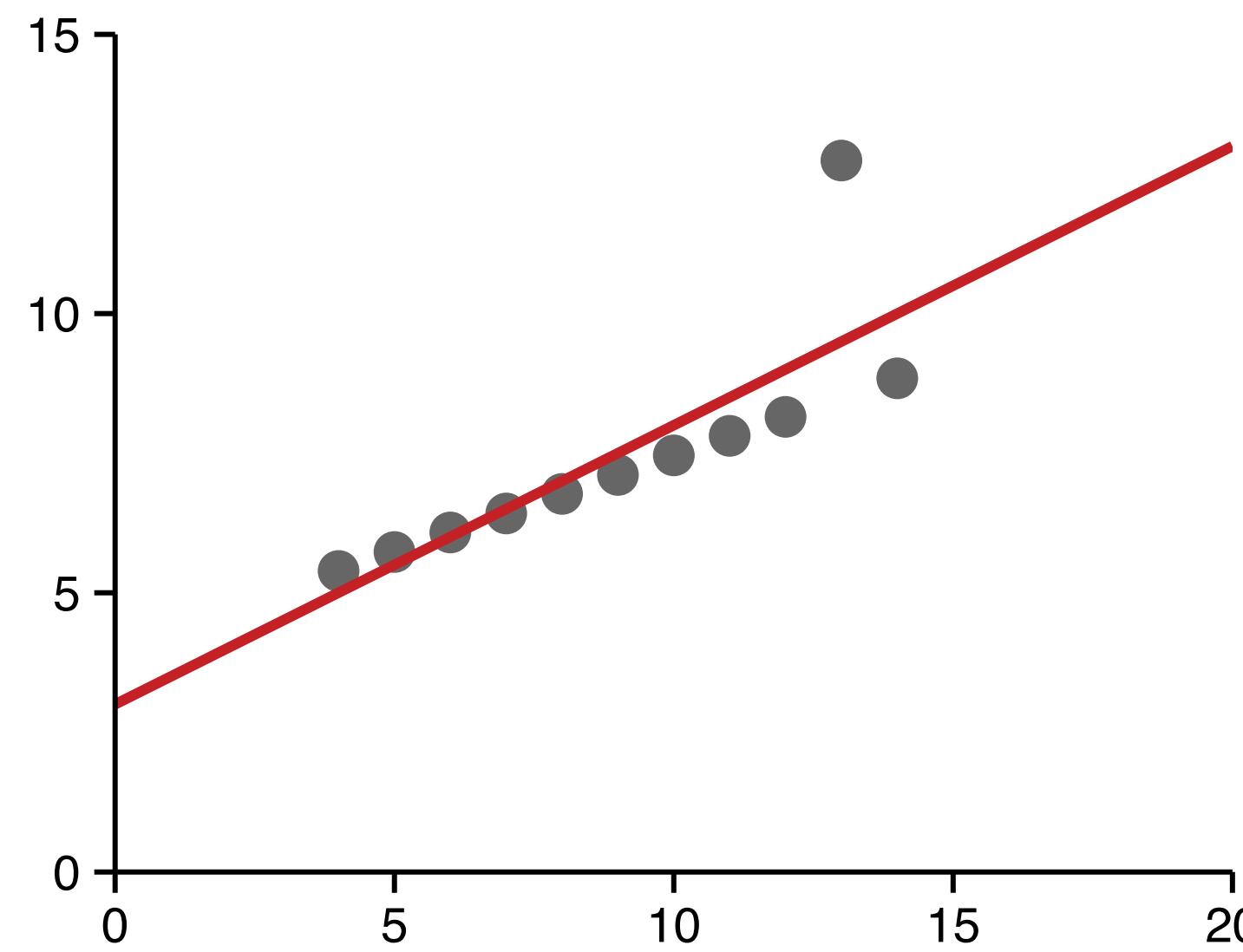
Anscombe Plots

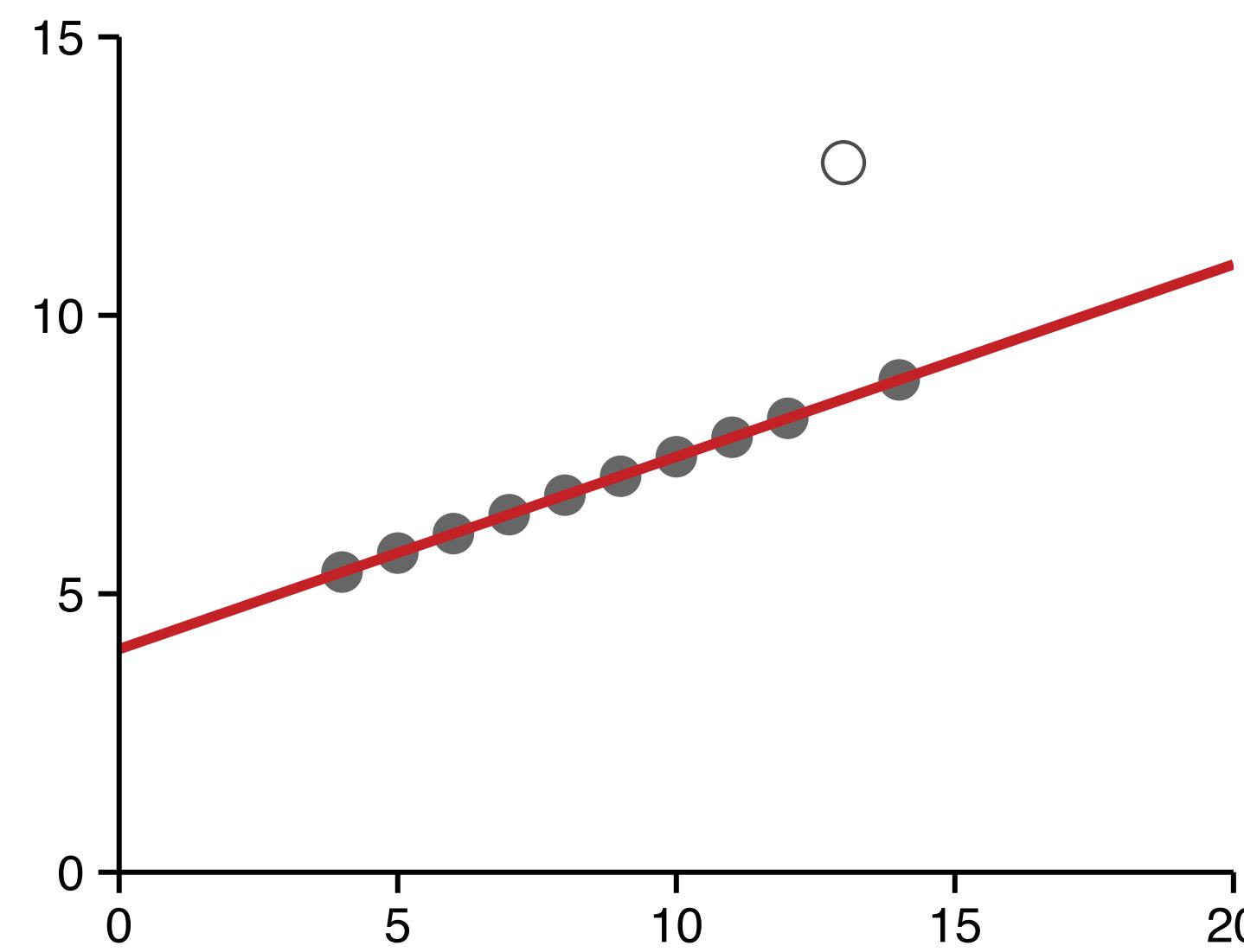


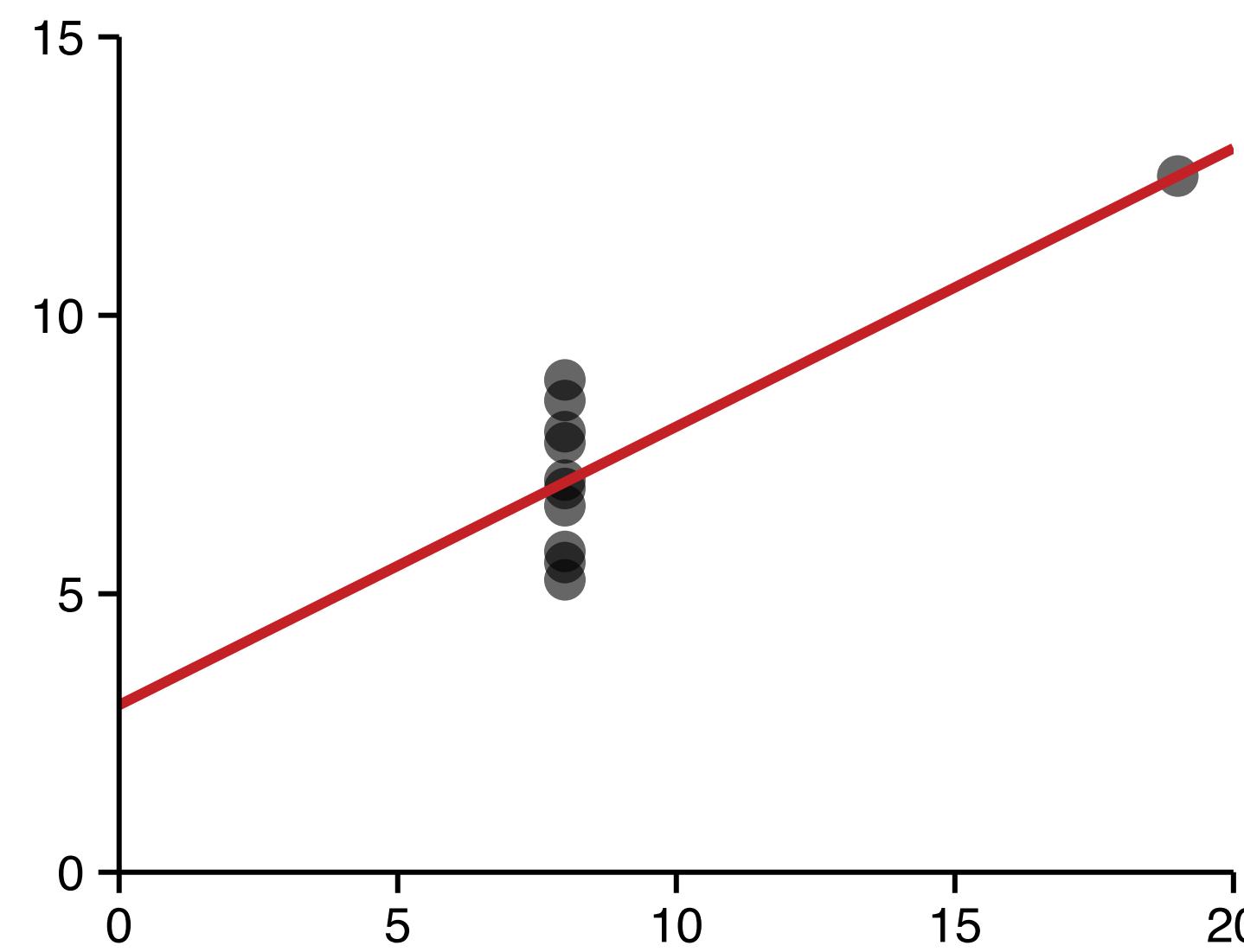


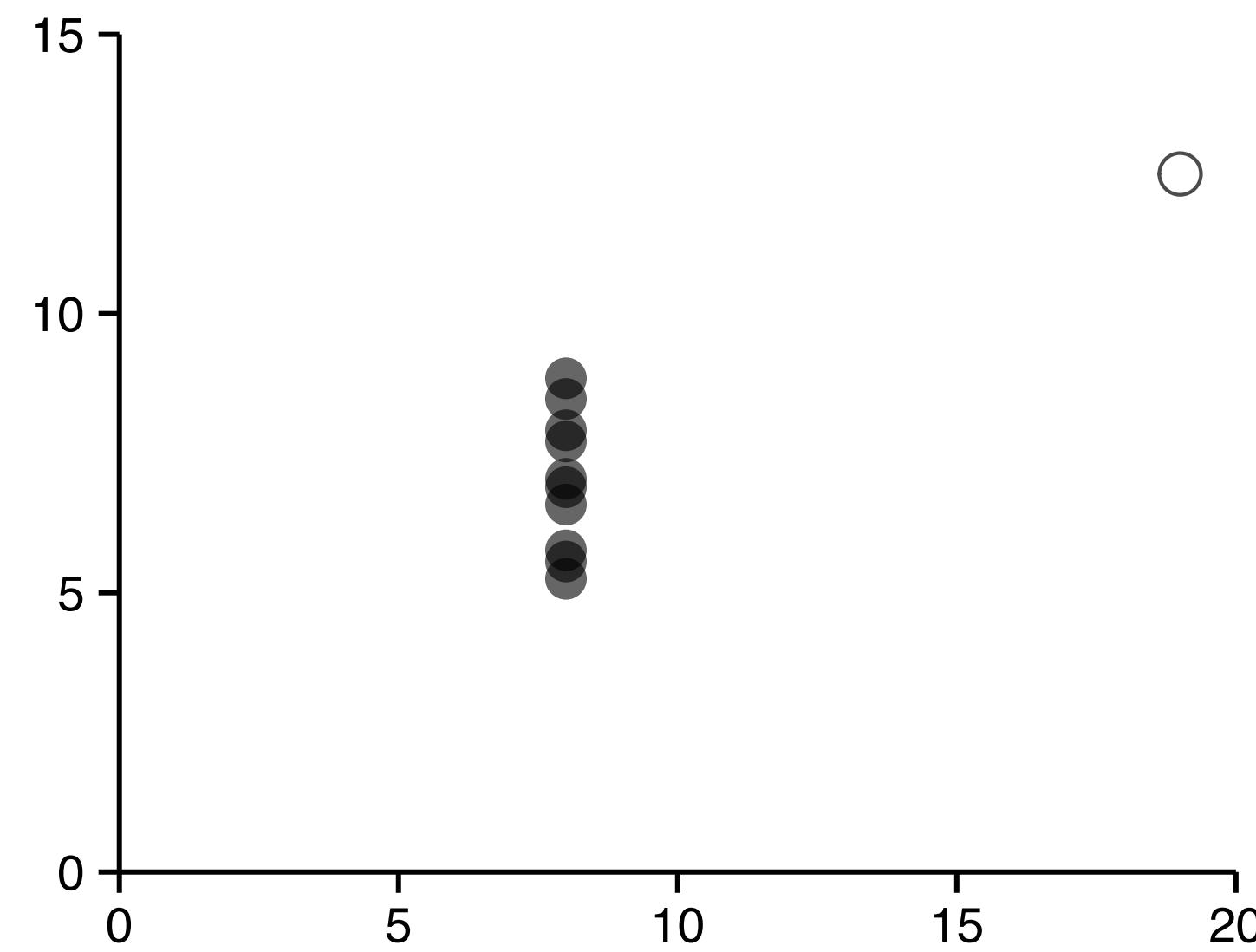


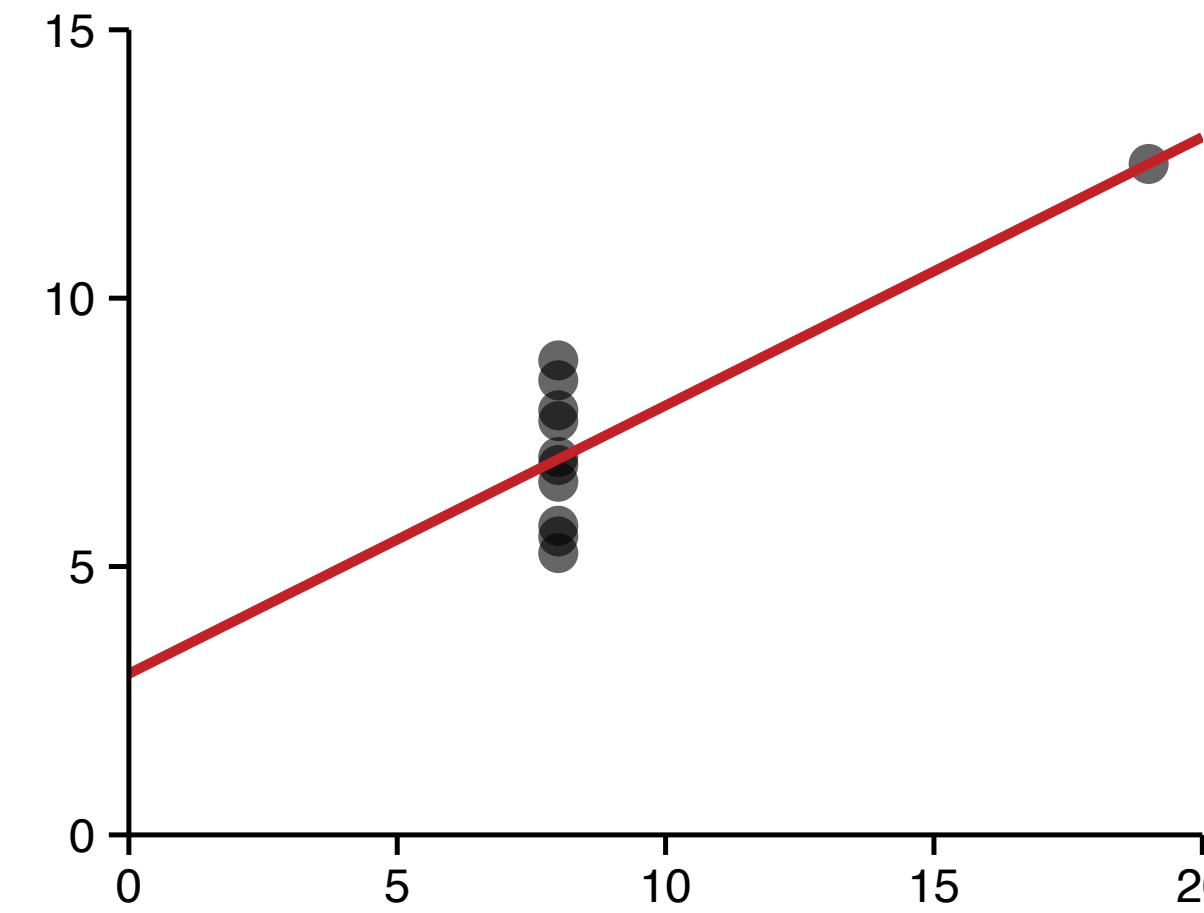
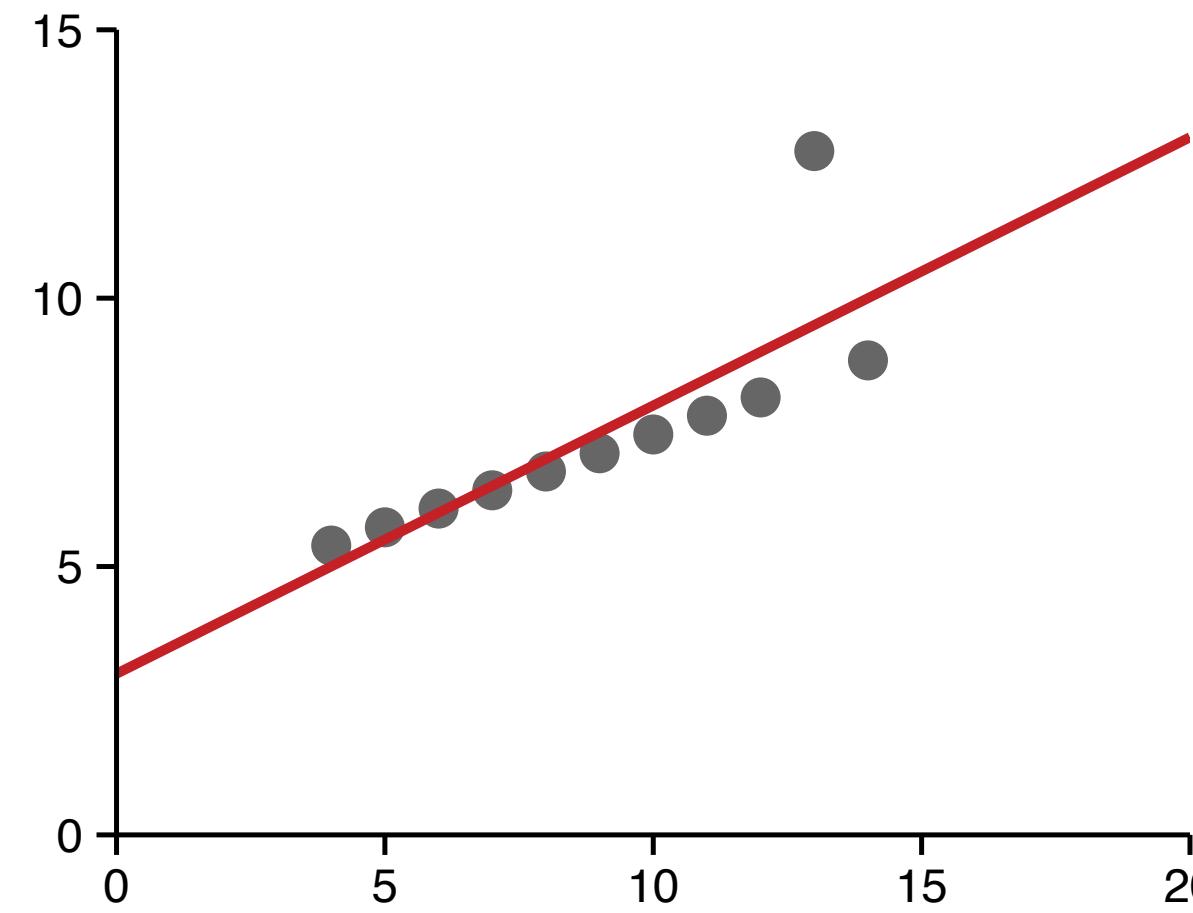
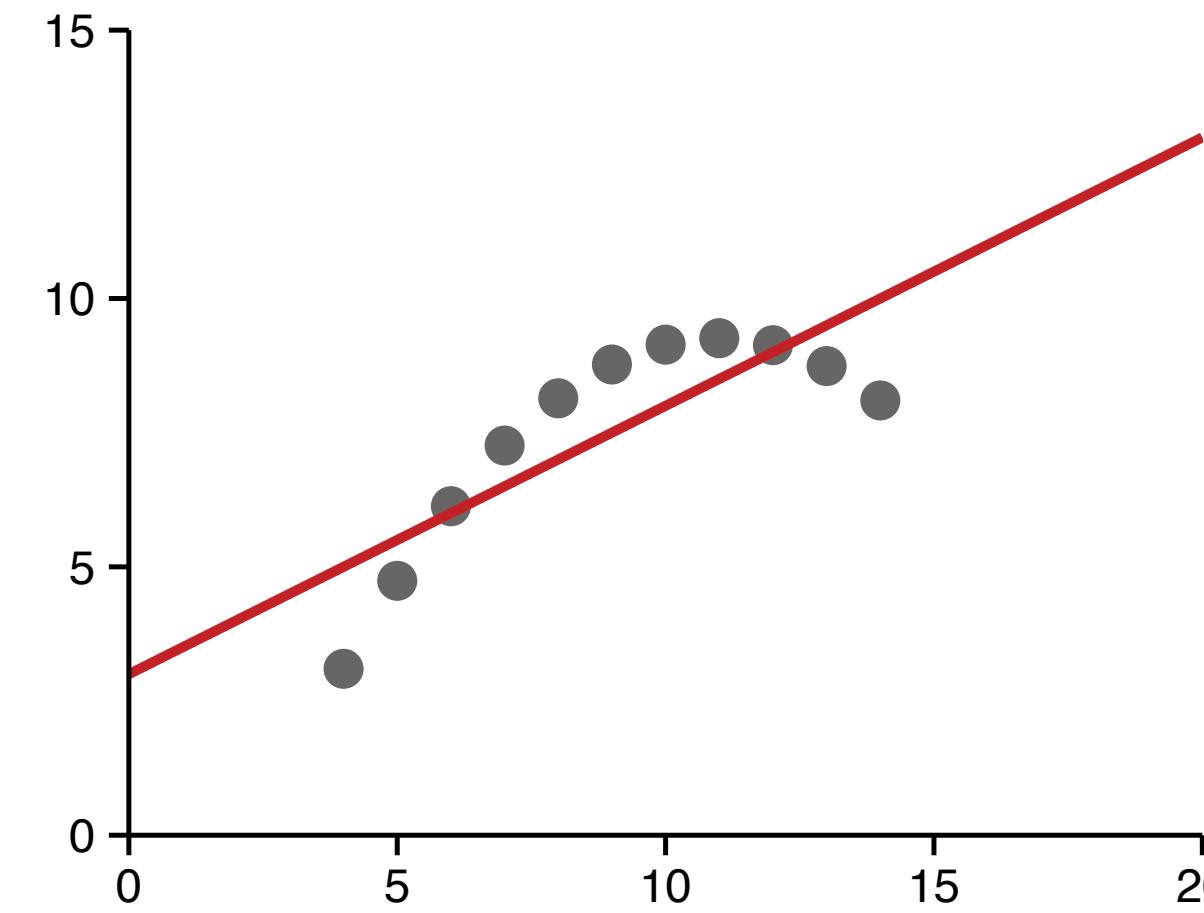
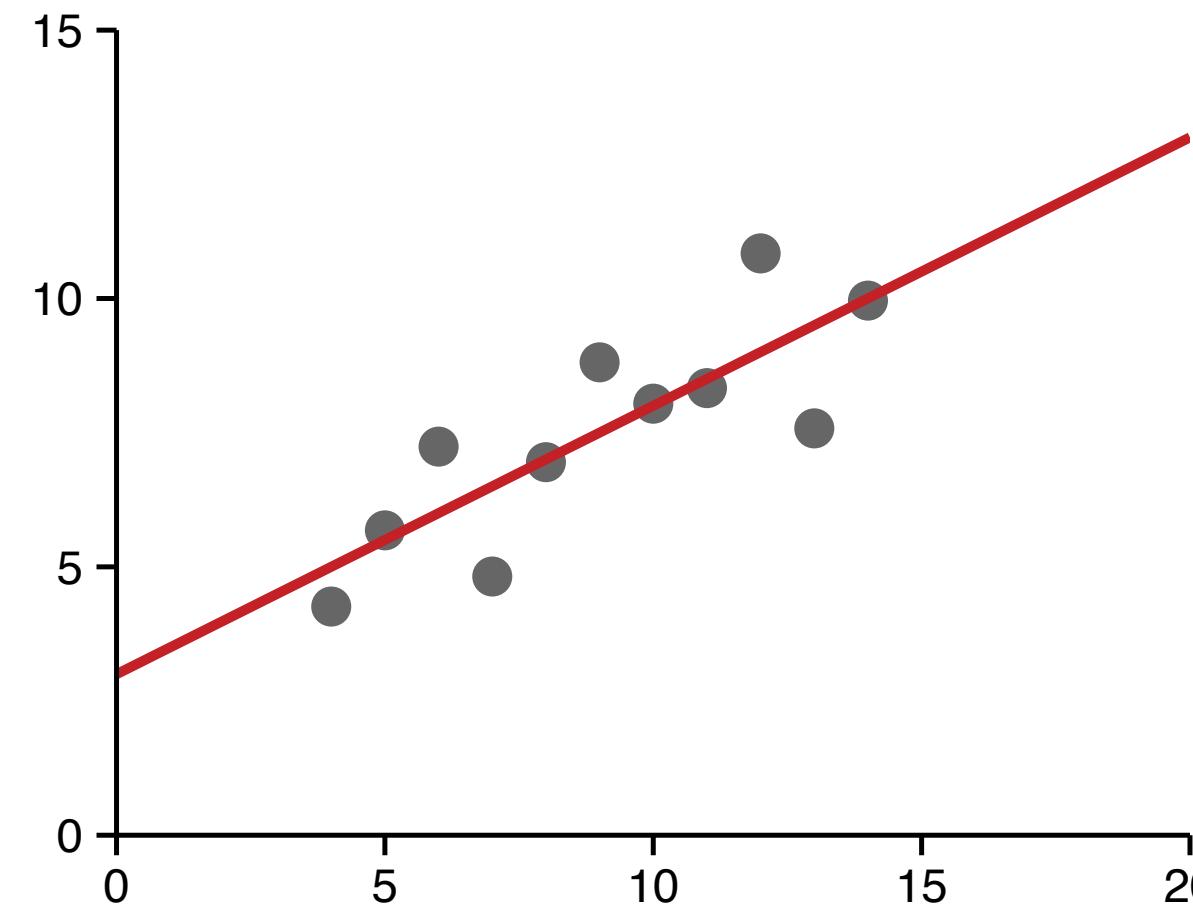


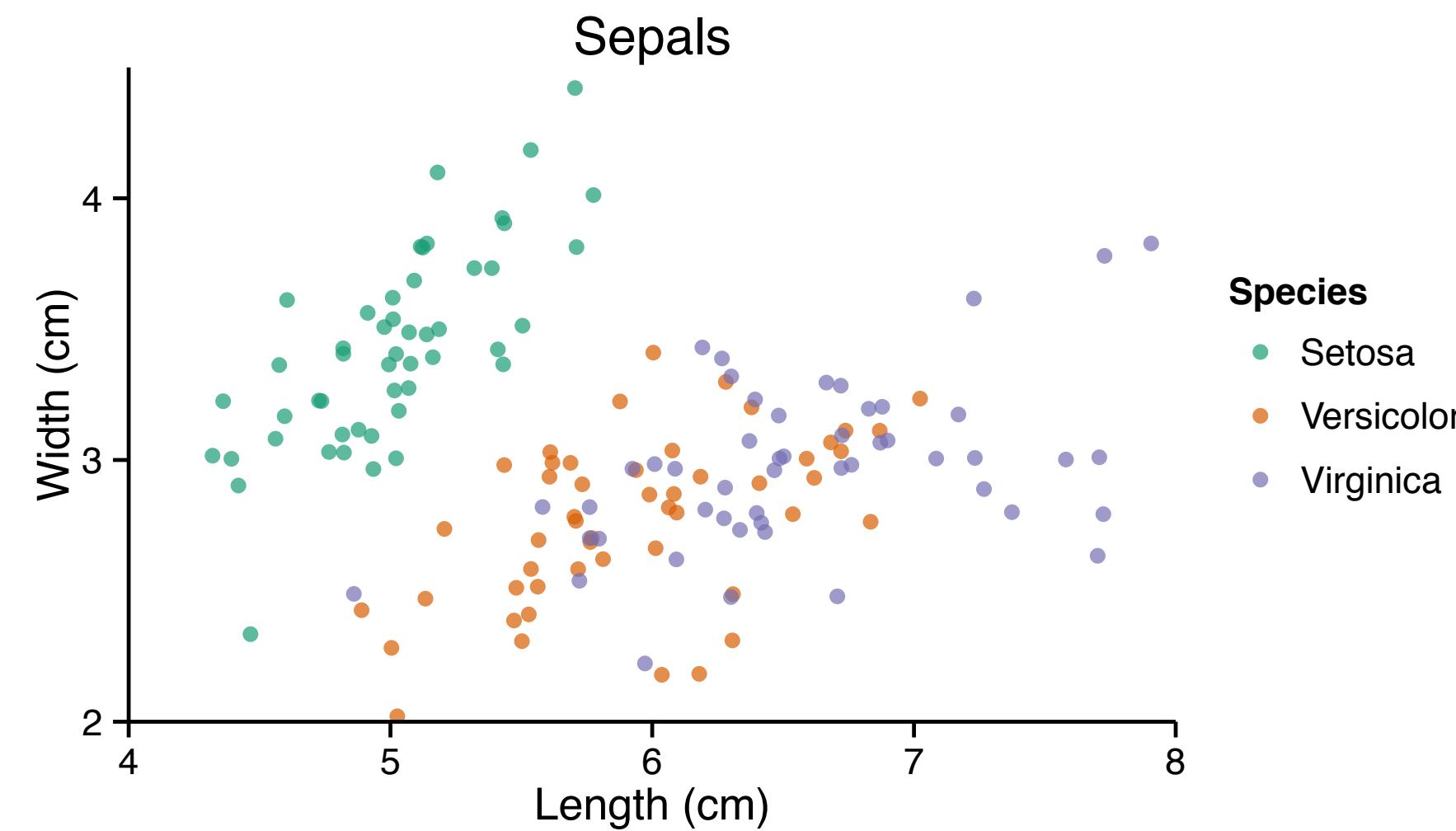


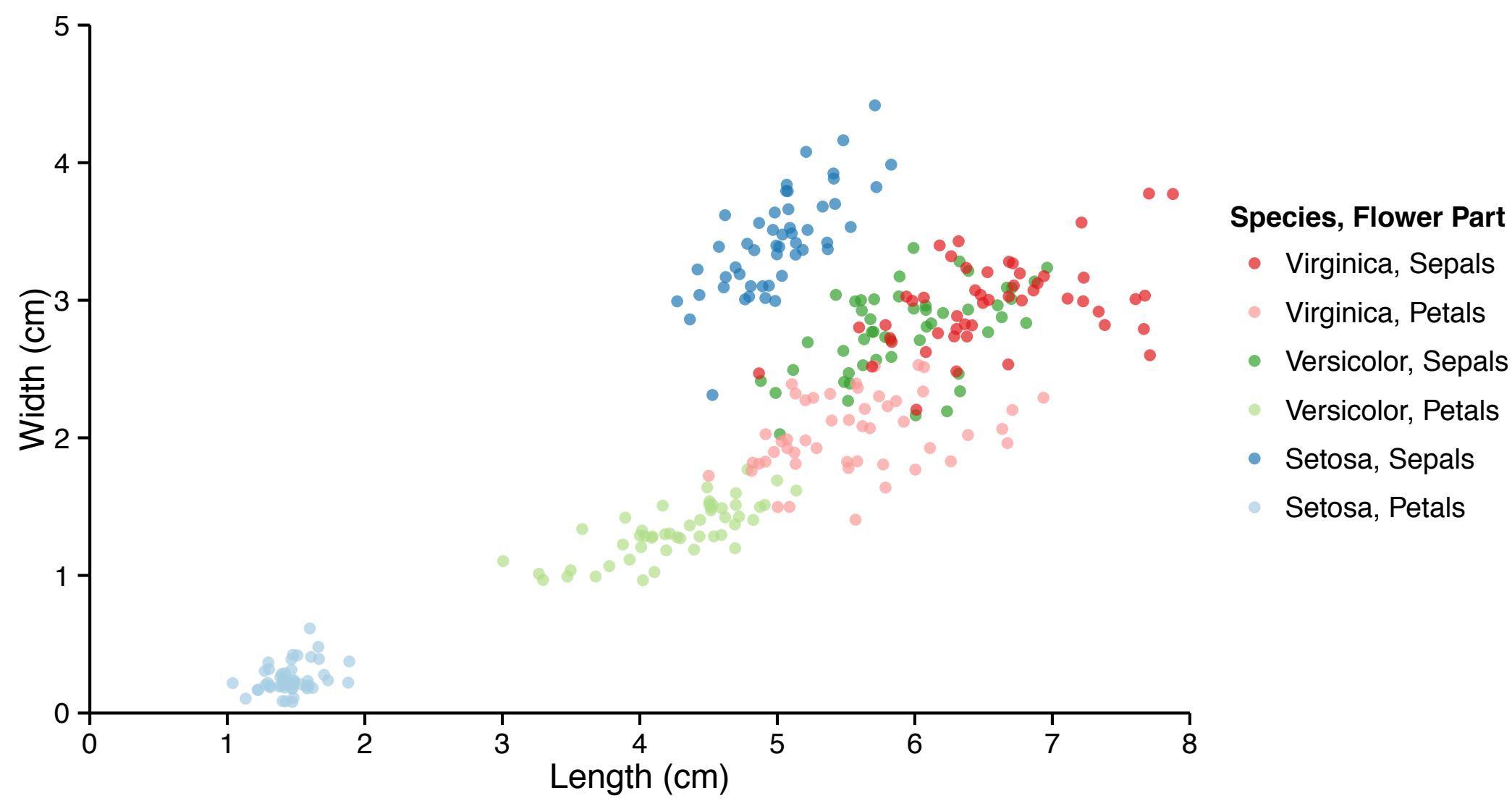
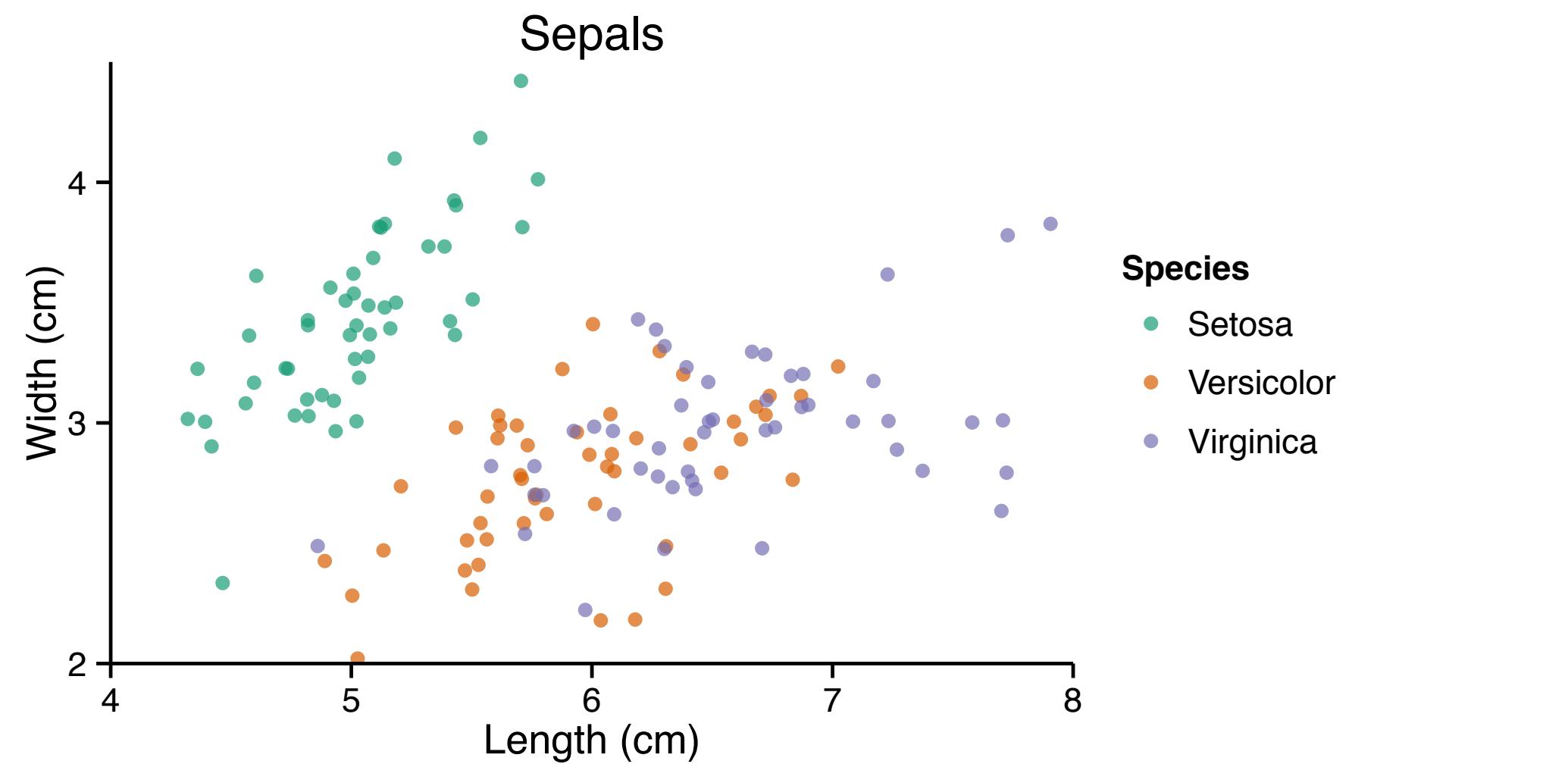


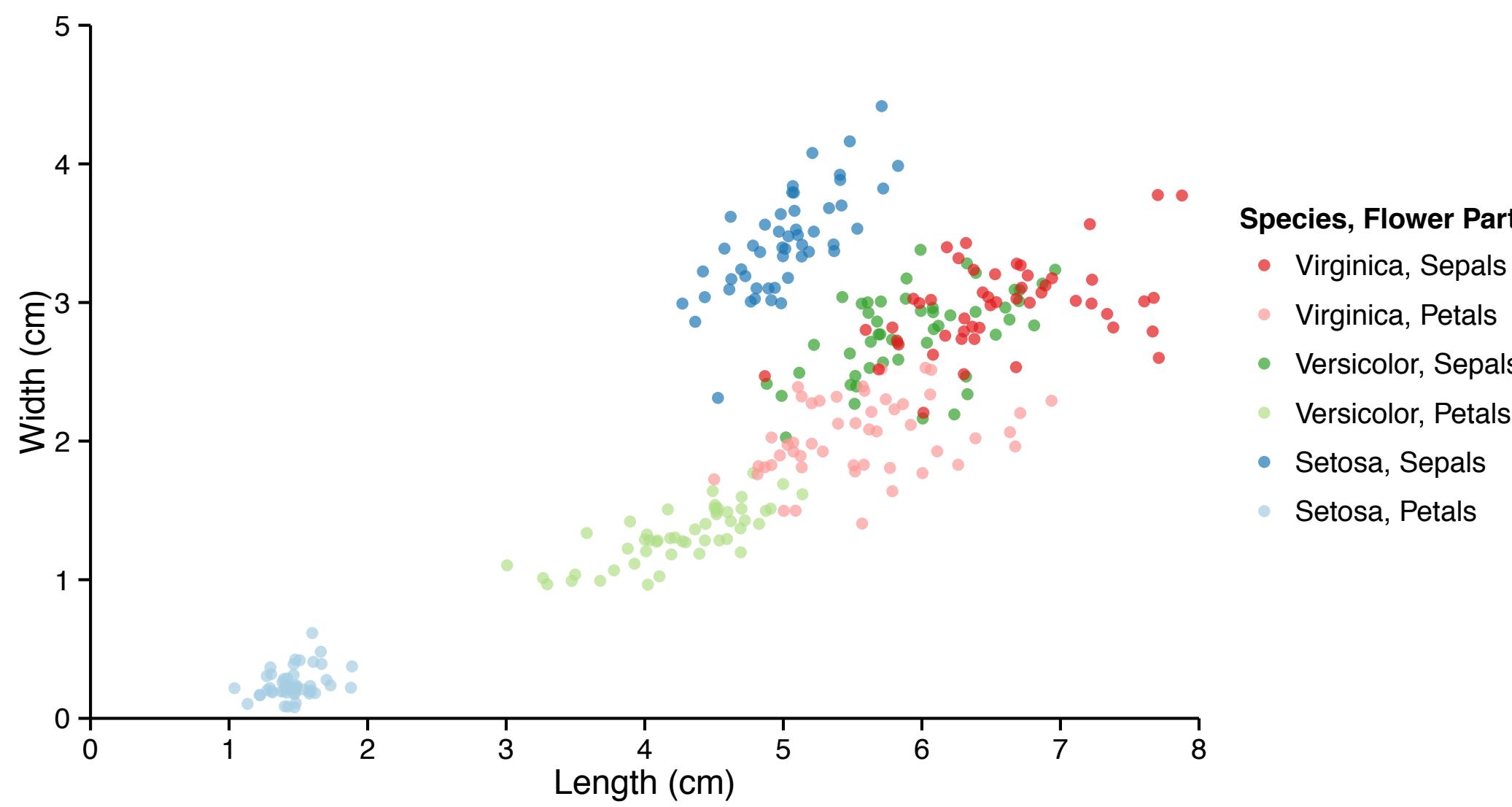
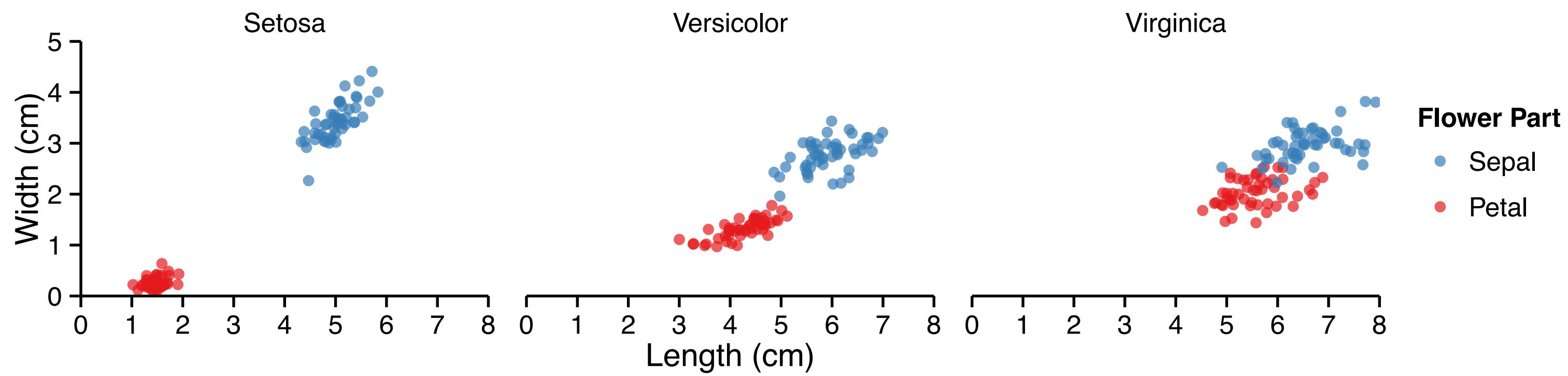


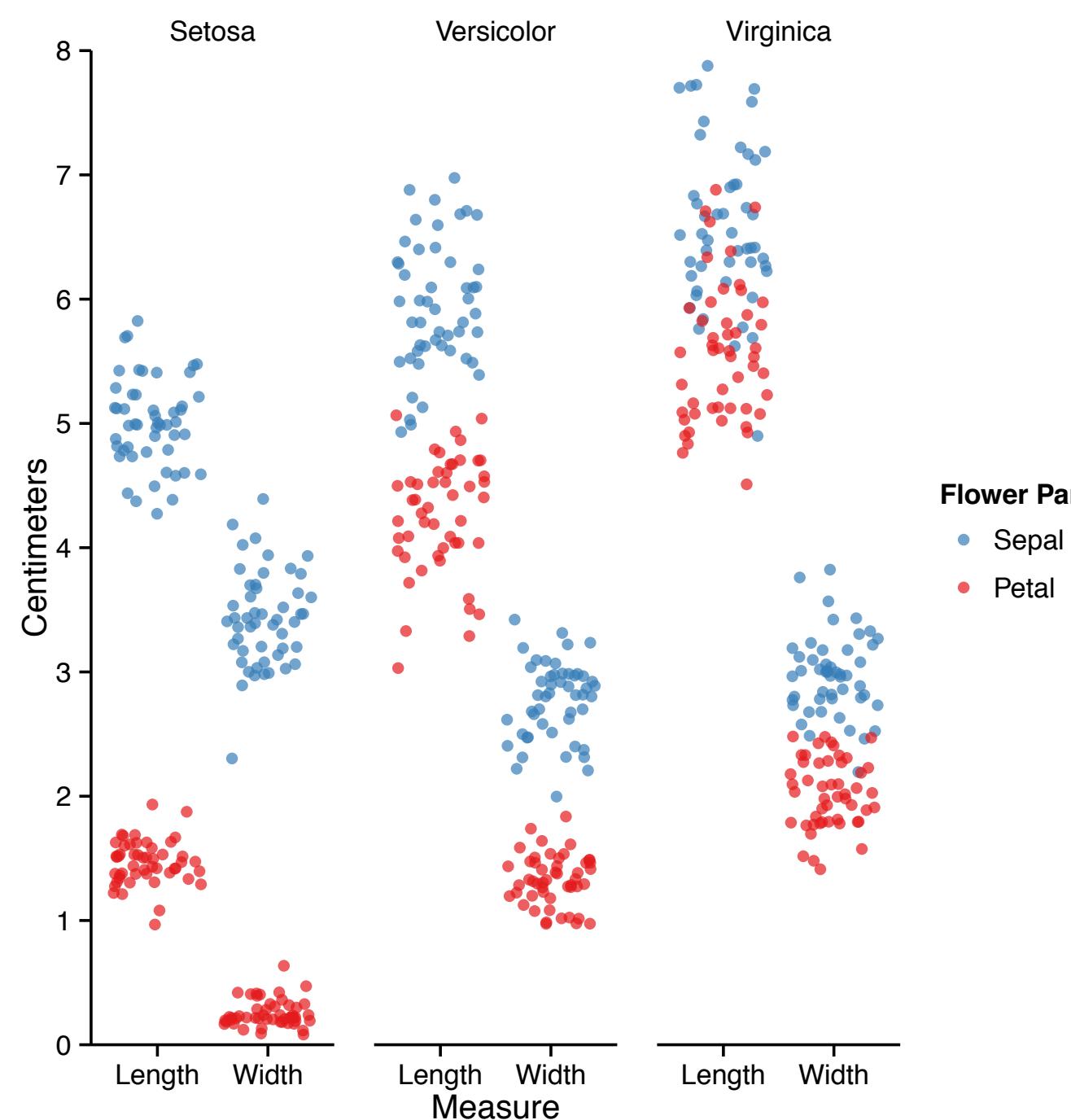
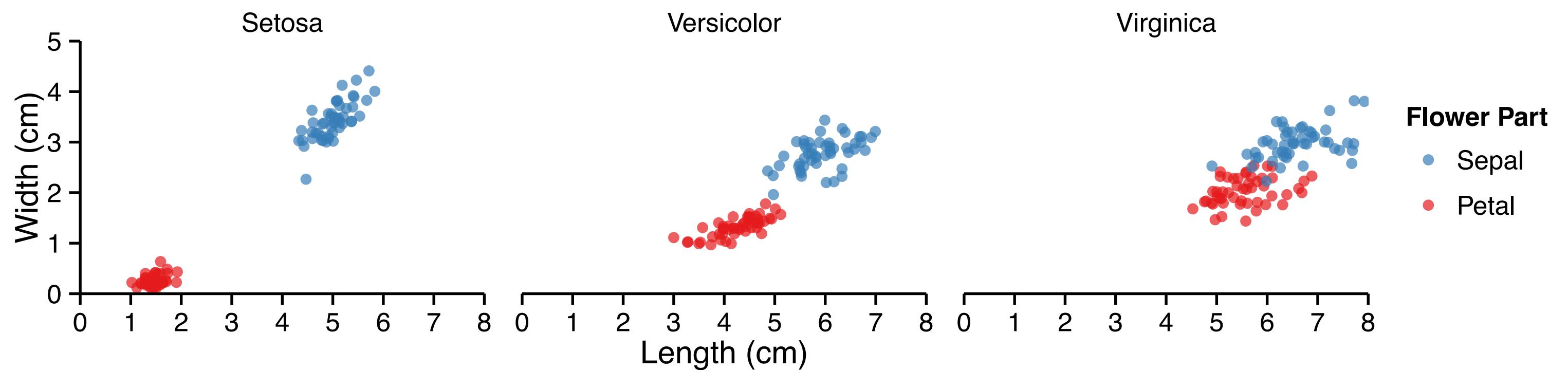


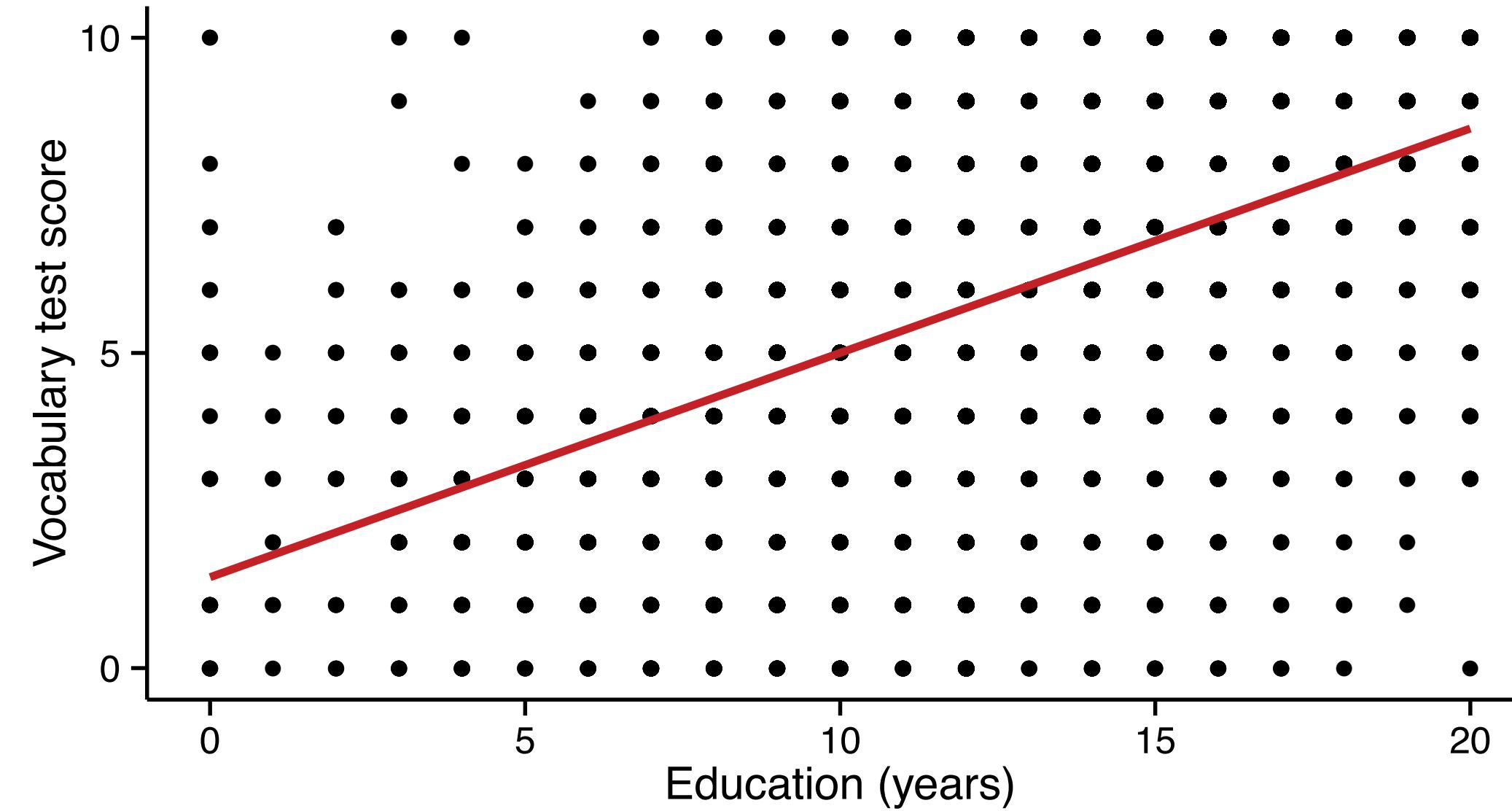


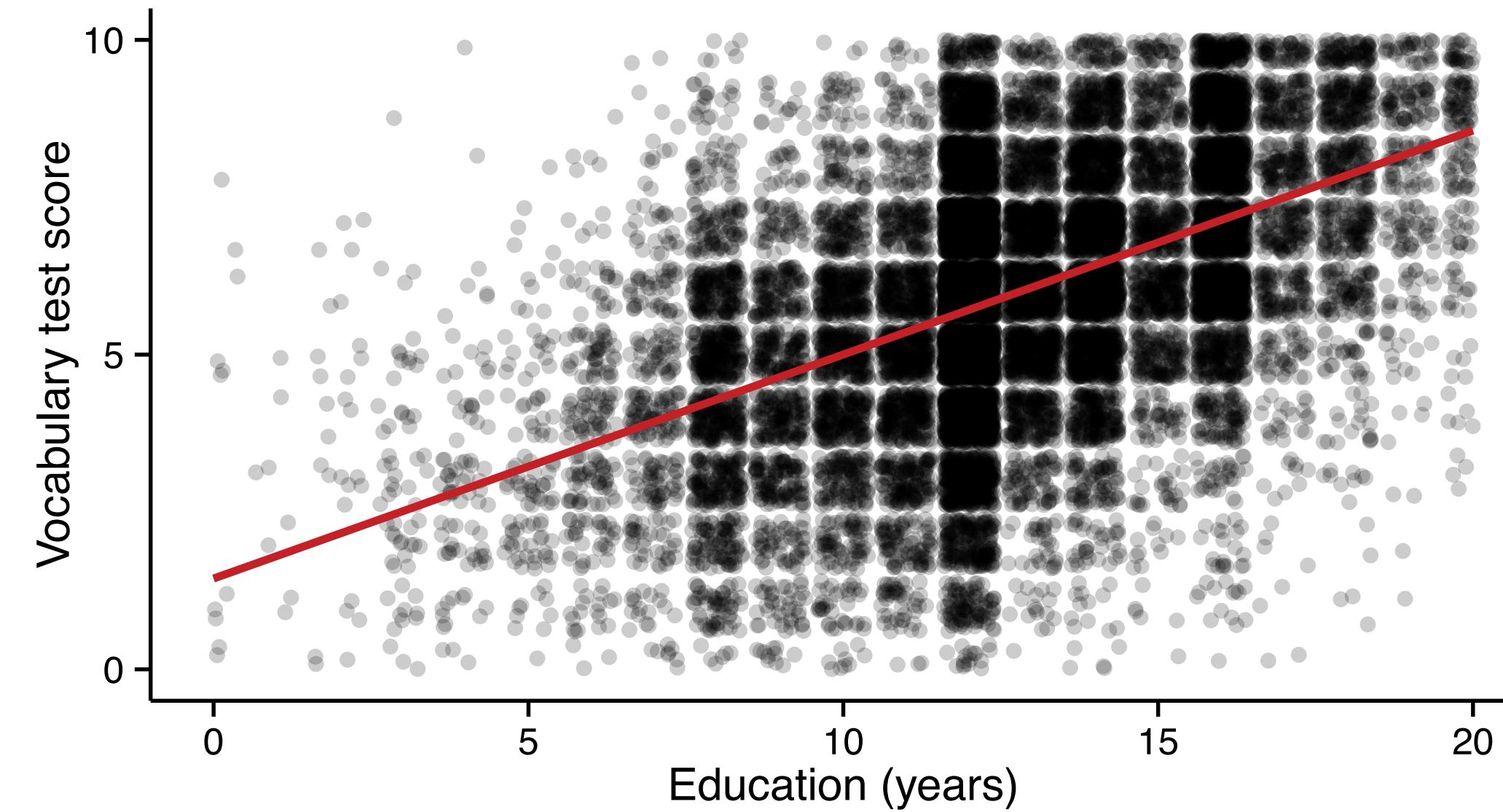


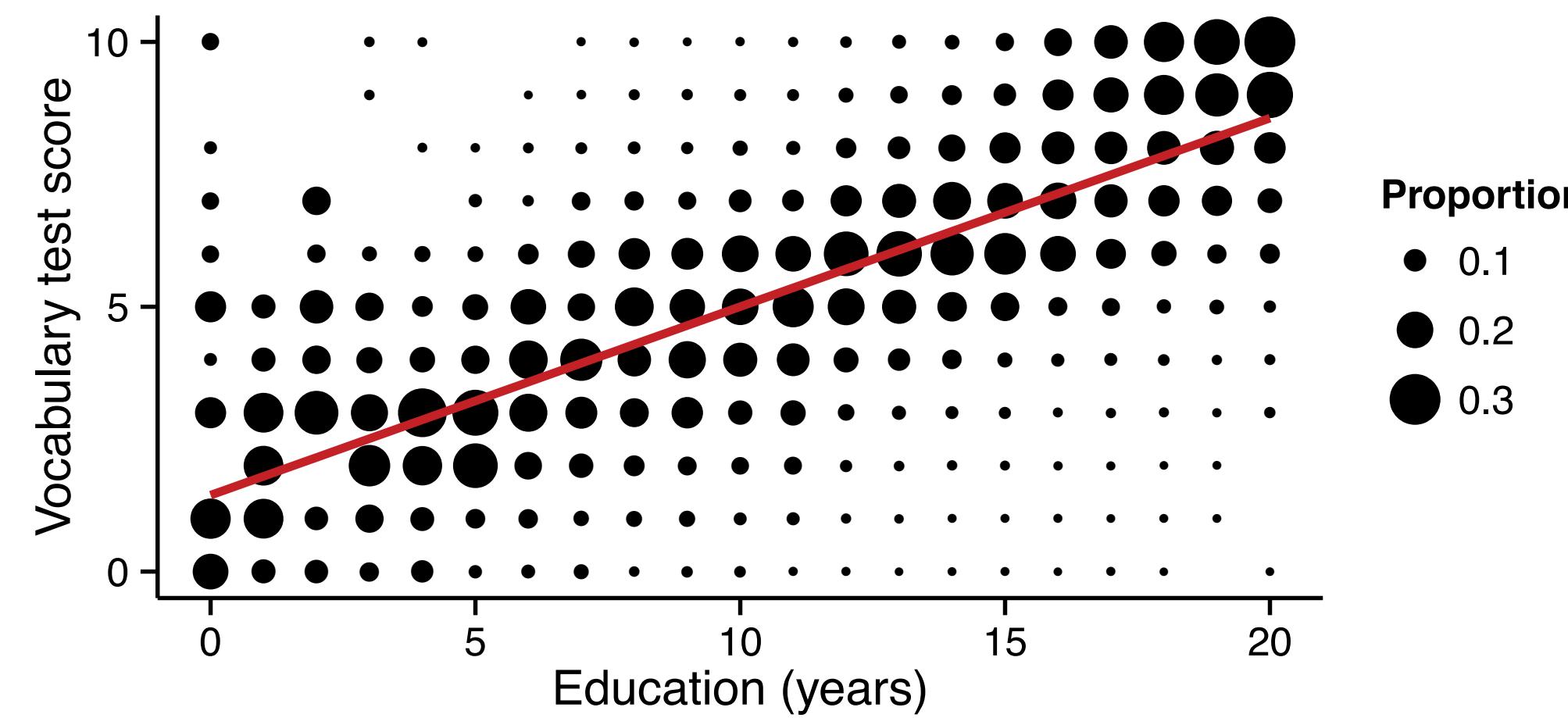
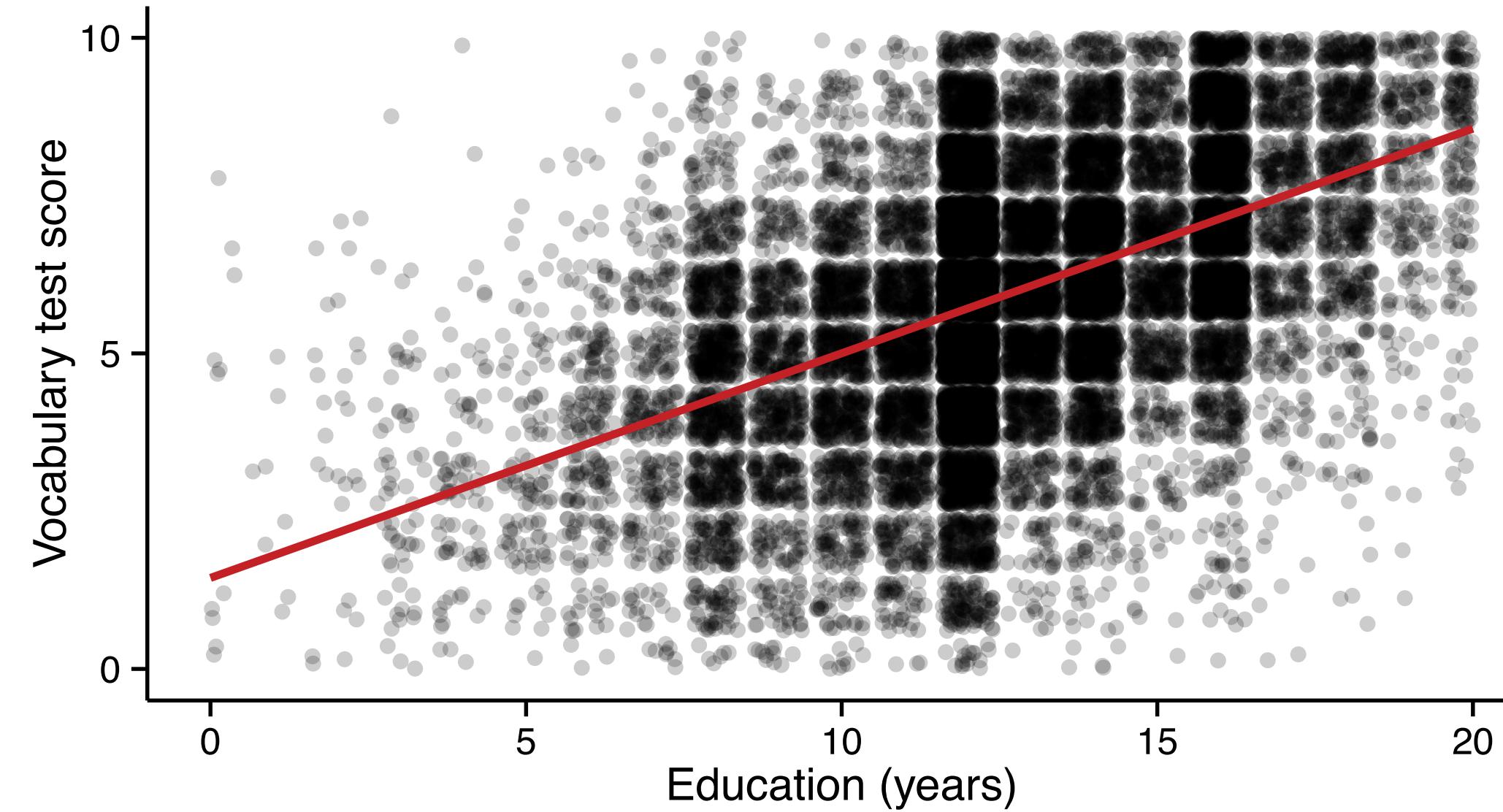




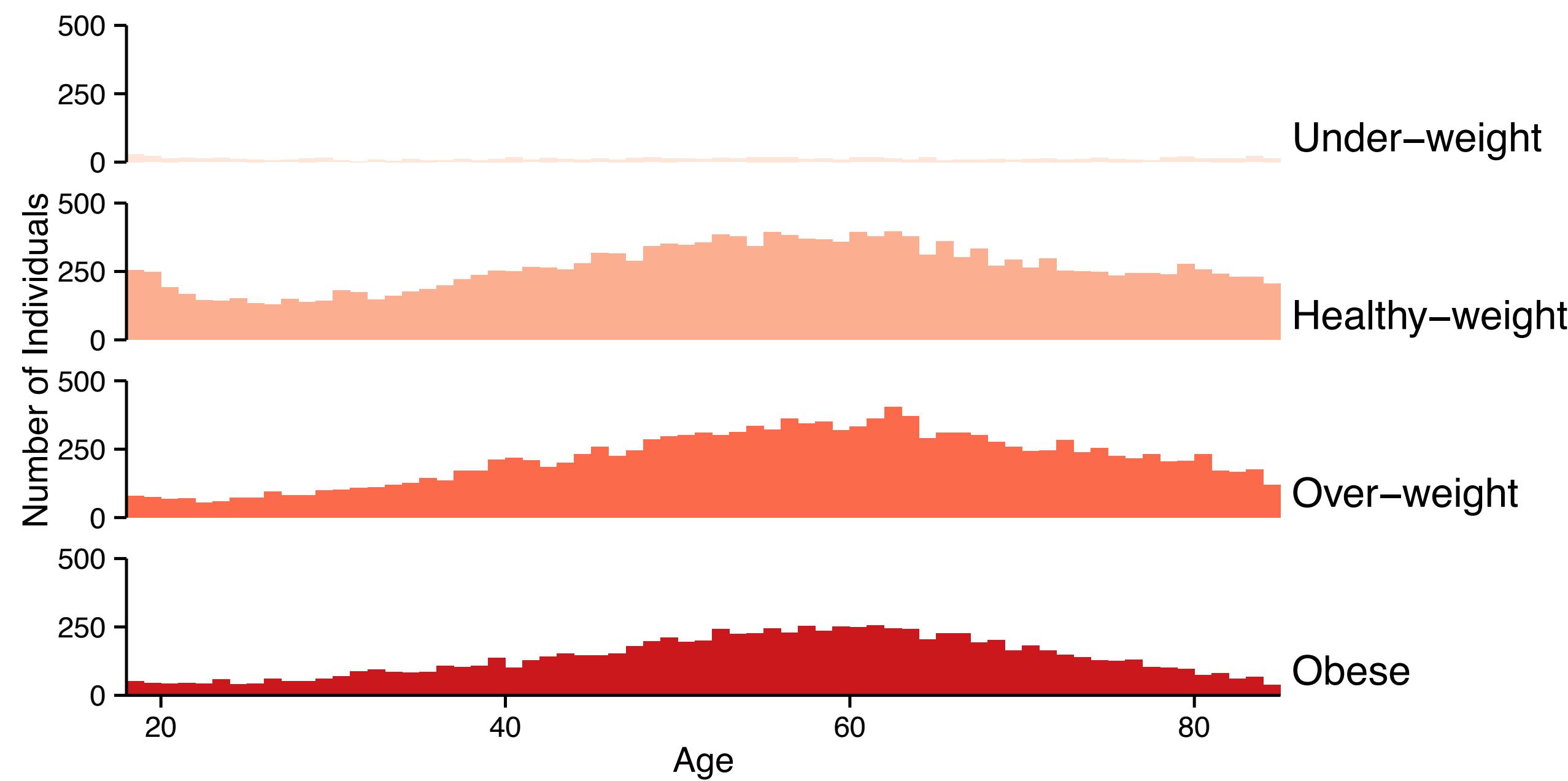




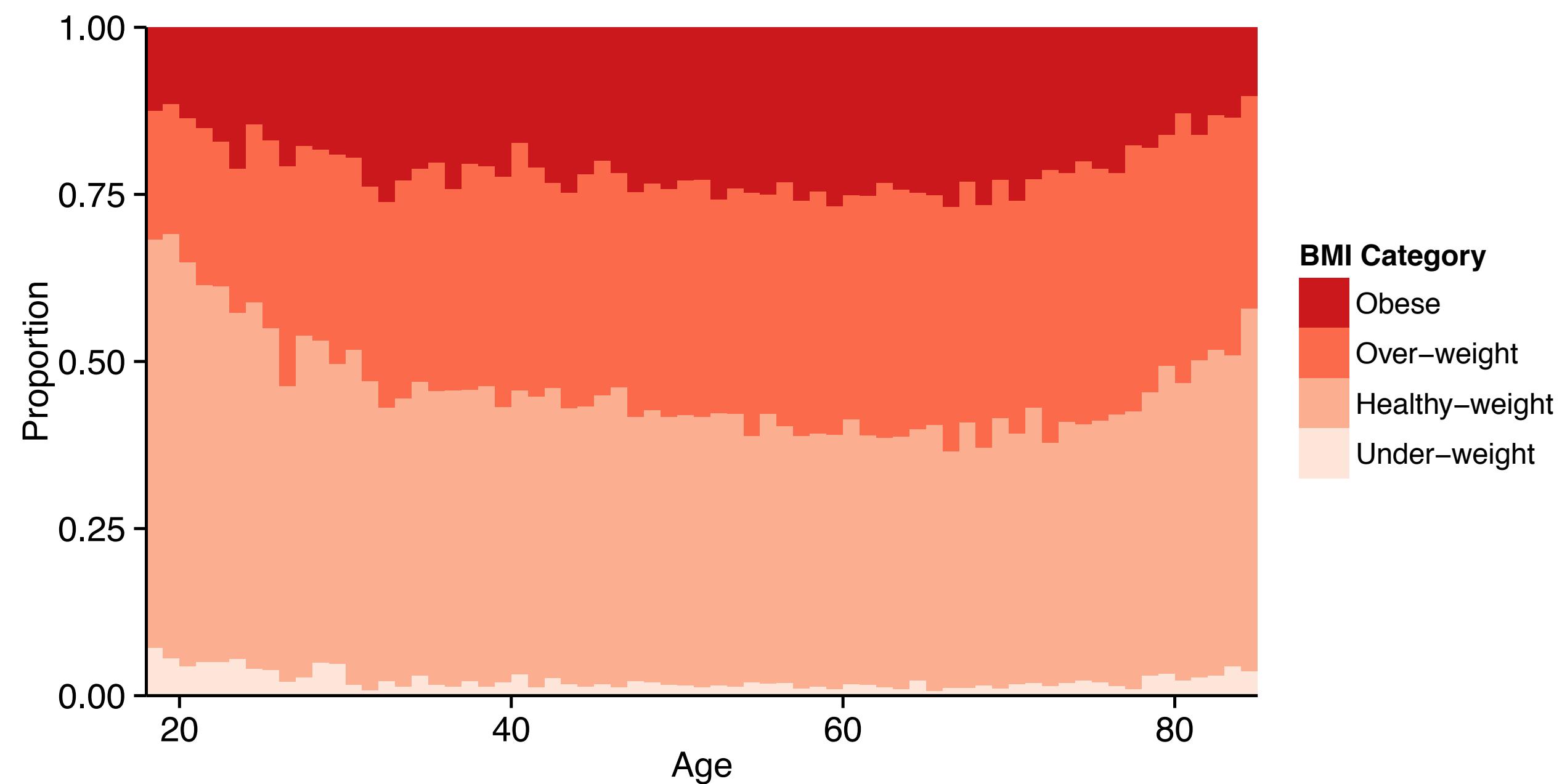




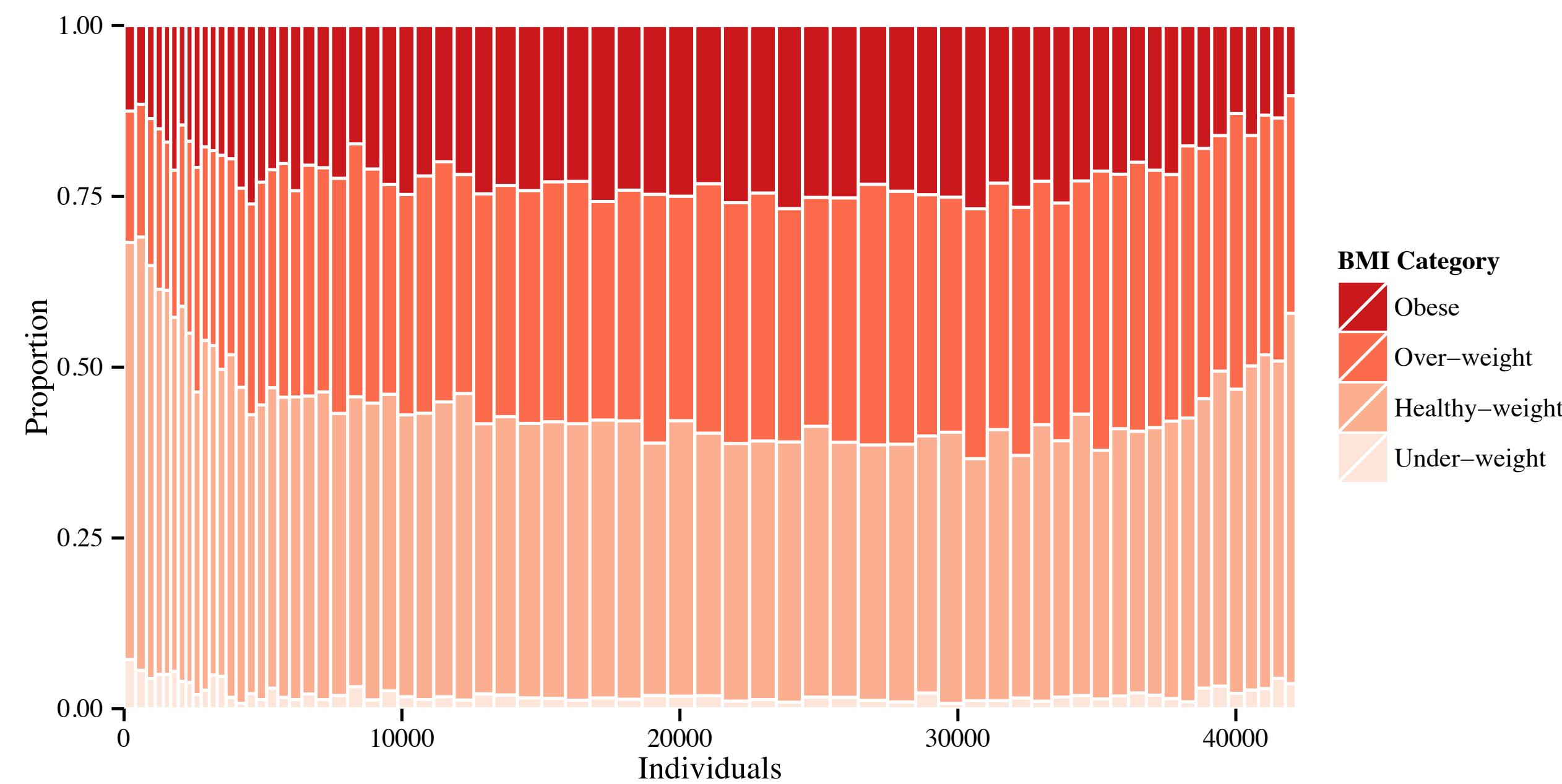
Case Study: CHIS



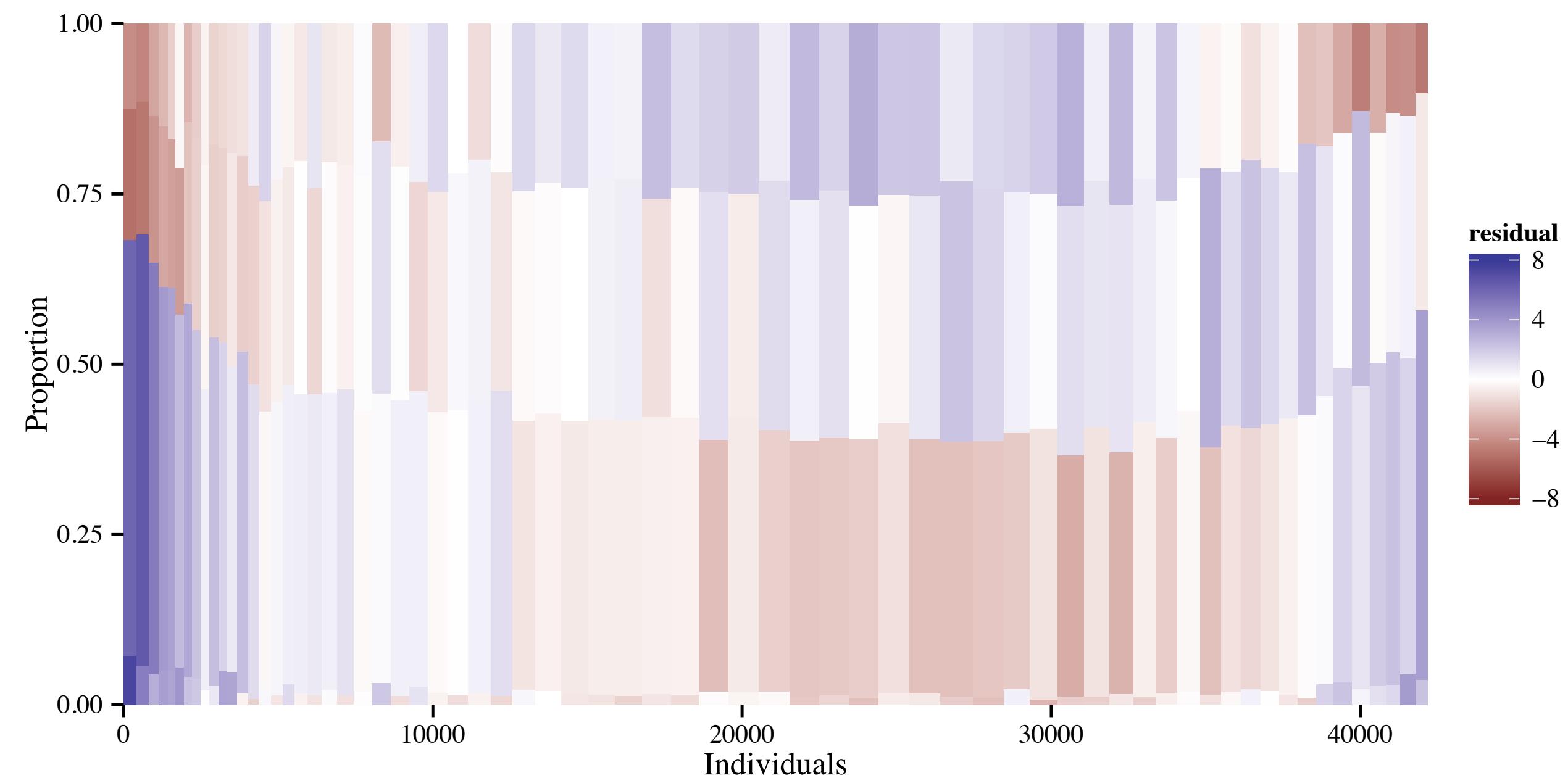
Case Study: CHIS



Case Study: CHIS



Case Study: CHIS



Outline

- Course 1
 - Concepts, data, aesthetics, geometries
- Course 2
 - Statistics, coordinates, facets, themes
 - Best practices
 - Case study
- Course 3
 - Advanced plots and ggplot2 internals



DATA VISUALIZATION WITH GGPLOT2

Let's practice!



DATA VISUALIZATION WITH GGPLOT2

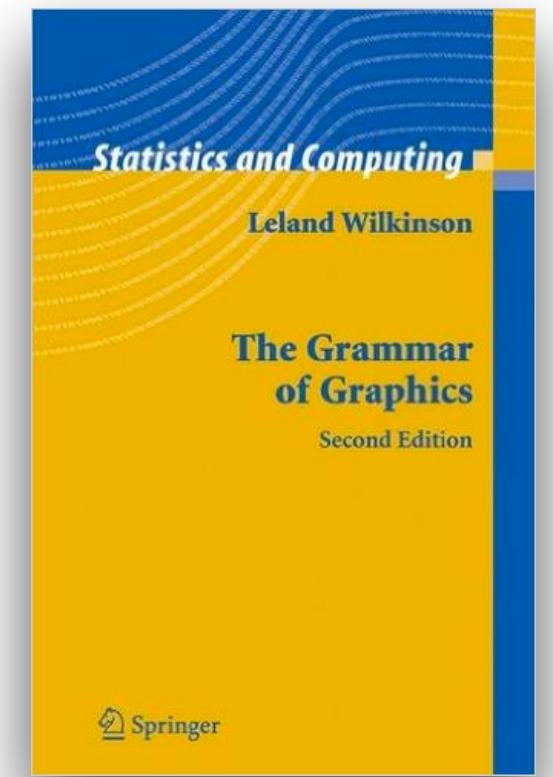
Grammar of Graphics

The quick brown fox jumps over the lazy dog

<i>Article</i>	<i>The</i>	<i>A</i>	<i>The</i>
<i>Adjective</i>	<i>quick brown</i>	<i>rabid red</i>	
<i>Noun</i>	<i>fox</i>	<i>fox</i>	<i>Hunter</i>
<i>Verb</i>	<i>jumps</i>	<i>bit</i>	<i>shot</i>
<i>Preposition</i>	<i>over</i>		
<i>Article</i>	<i>the</i>	<i>the</i>	<i>the</i>
<i>Adjective</i>	<i>lazy</i>	<i>friendly</i>	<i>rabid red</i>
<i>Noun</i>	<i>dog.</i>	<i>dog.</i>	<i>fox.</i>

Grammar of Graphics

- Plotting Framework
- Leland Wilkinson, Grammar of Graphics, 1999
- 2 principles
 - Graphics = distinct layers of grammatical elements
 - Meaningful plots through aesthetic mapping



Essential Grammatical Elements

Element	Description
Data	The dataset being plotted.
Aesthetics	The scales onto which we <i>map</i> our data.
Geometries	The visual elements used for our data.

All Grammatical Elements

Element	Description
Data	The dataset being plotted.
Aesthetics	The scales onto which we <i>map</i> our data.
Geometries	The visual elements used for our data.
Facets	Plotting small multiples.
Statistics	Representations of our data to aid understanding.
Coordinates	The space on which the data will be plotted.
Themes	All non-data ink.

Diagram

<i>Data</i>	{variables of interest}				
<i>Aesthetics</i>	<i>x-axis</i>	<i>colour</i>	<i>size</i>	<i>alpha</i>	<i>line width</i>
	<i>y-axis</i>	<i>fill</i>	<i>labels</i>	<i>shape</i>	<i>line type</i>
<i>Geometries</i>	<i>point</i> <i>line</i> <i>histogram</i> <i>bar</i> <i>boxplot</i>				
<i>Facets</i>	<i>columns</i>	<i>rows</i>			
<i>Statistics</i>	<i>binning</i>	<i>smoothing</i>	<i>descriptive</i>	<i>inferential</i>	
<i>Coordinates</i>	<i>cartesian</i>	<i>fixed</i>	<i>polar</i>	<i>limits</i>	
<i>Themes</i>	<i>non-data ink</i>				

Grammar of Graphics

- Building blocks
- Solid, creative, meaningful visualizations
- Course 1: First 3 layers
- Course 2: Remaining 4 layers



DATA VISUALIZATION WITH GGPLOT2

Let's practice!



DATA VISUALIZATION WITH GGPLOT2

ggplot2

ggplot2

- Hadley Wickham
- Layer grammatical elements
- Aesthetic Mappings

ggplot2 Layers - Data

Data



Iris dataset

- Edgar Anderson
- R.A. Fischer



setosa



versicolor



virginica

Iris dataset

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1        3.5         1.4       0.2    setosa
2          4.9        3.0         1.4       0.2    setosa
3          4.7        3.2         1.3       0.2    setosa
...
50         5.0        3.3         1.4       0.2    setosa
51         7.0        3.2         4.7      1.4 versicolor
52         6.4        3.2         4.5      1.5 versicolor
53         6.9        3.1         4.9      1.5 versicolor
...
100        5.7        2.8         4.1      1.3 versicolor
101        6.3        3.3         6.0      2.5  virginica
102        5.8        2.7         5.1      1.9  virginica
103        7.1        3.0         5.9      2.1  virginica
...
150        5.9        3.0         5.1      1.8  virginica
```

ggplot2 Layers - Aesthetics

Aesthetics
Data



ggplot2 Layers - Aesthetics

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
	X		Y	

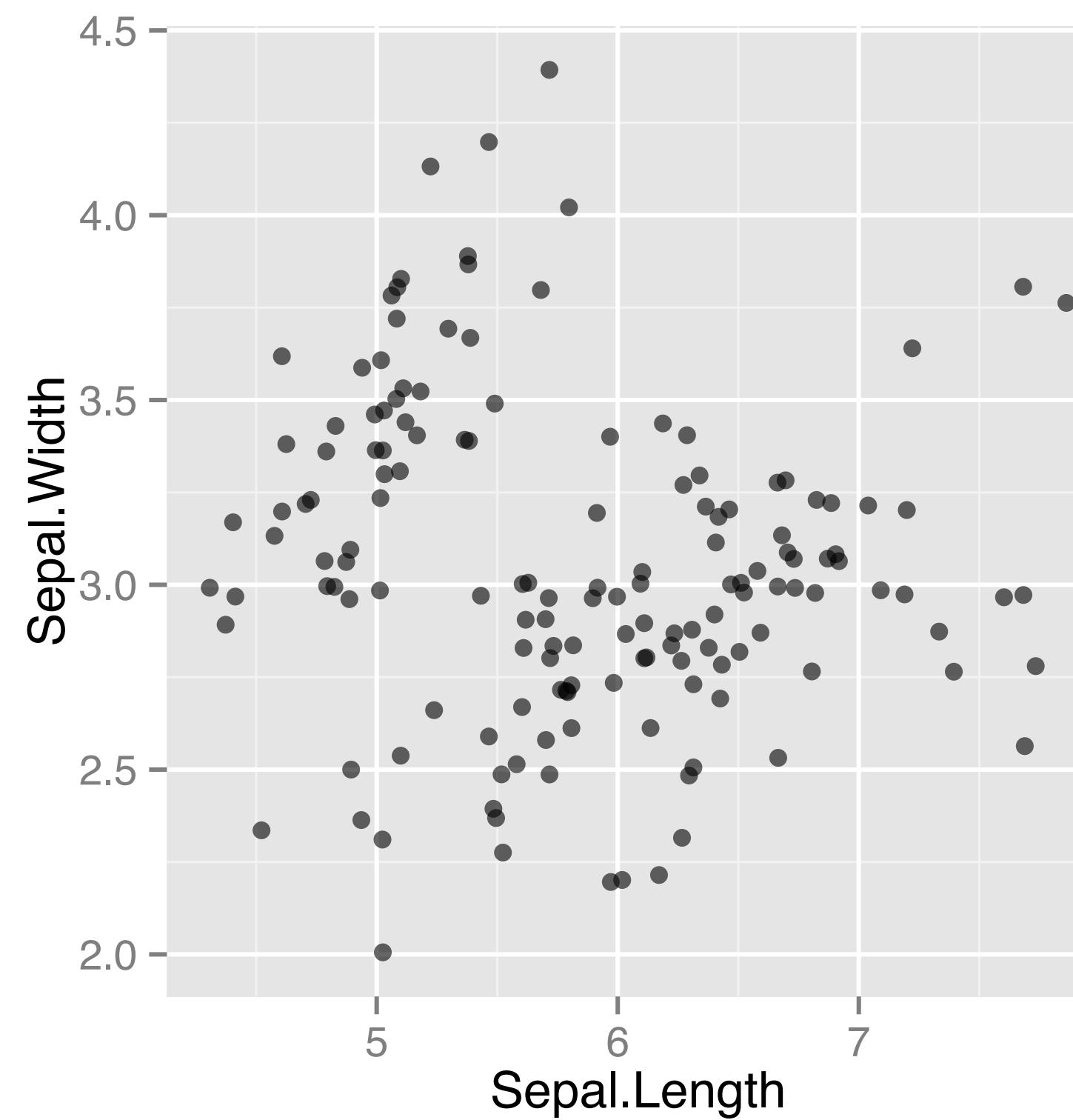
ggplot2 Layers - Geometries

Geometries
Aesthetics
Data



ggplot2 Layers - Geometries

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_jitter(alpha = 0.6)
```



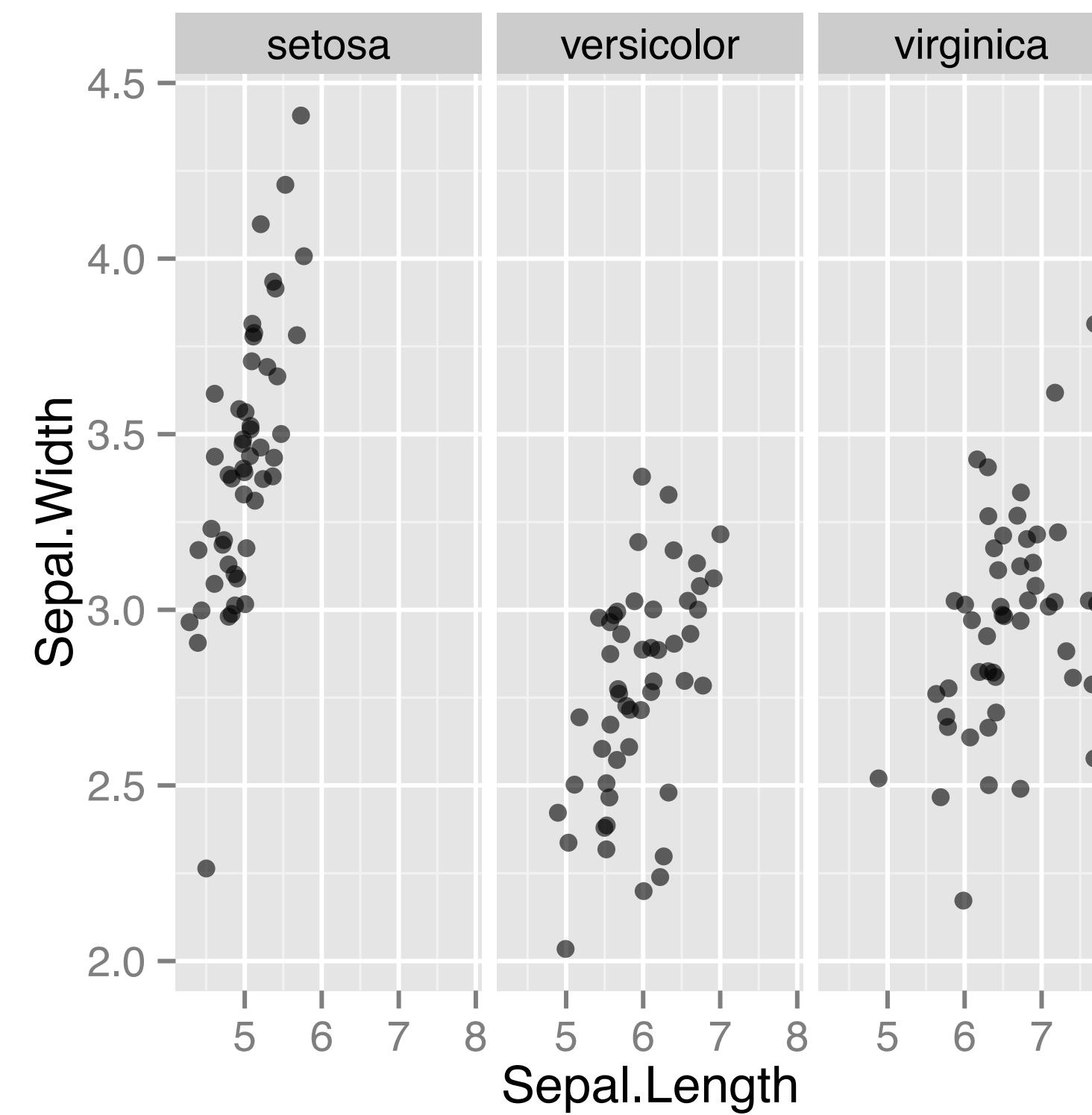
ggplot2 Layers - Facets

Facets
Geometries
Aesthetics
Data

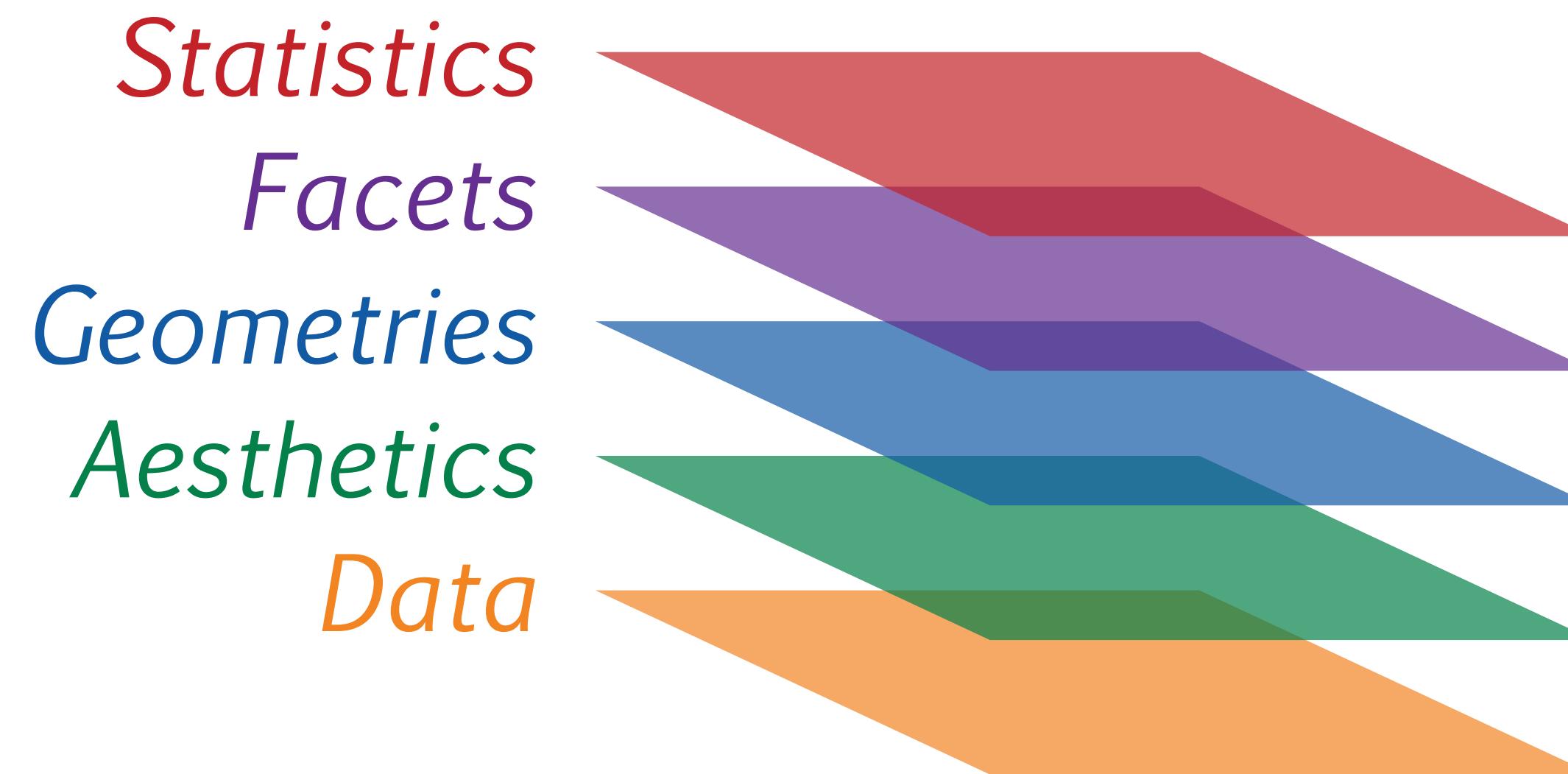


ggplot2 Layers - Facets

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_jitter(alpha = 0.6) +  
  facet_grid(. ~ Species)
```

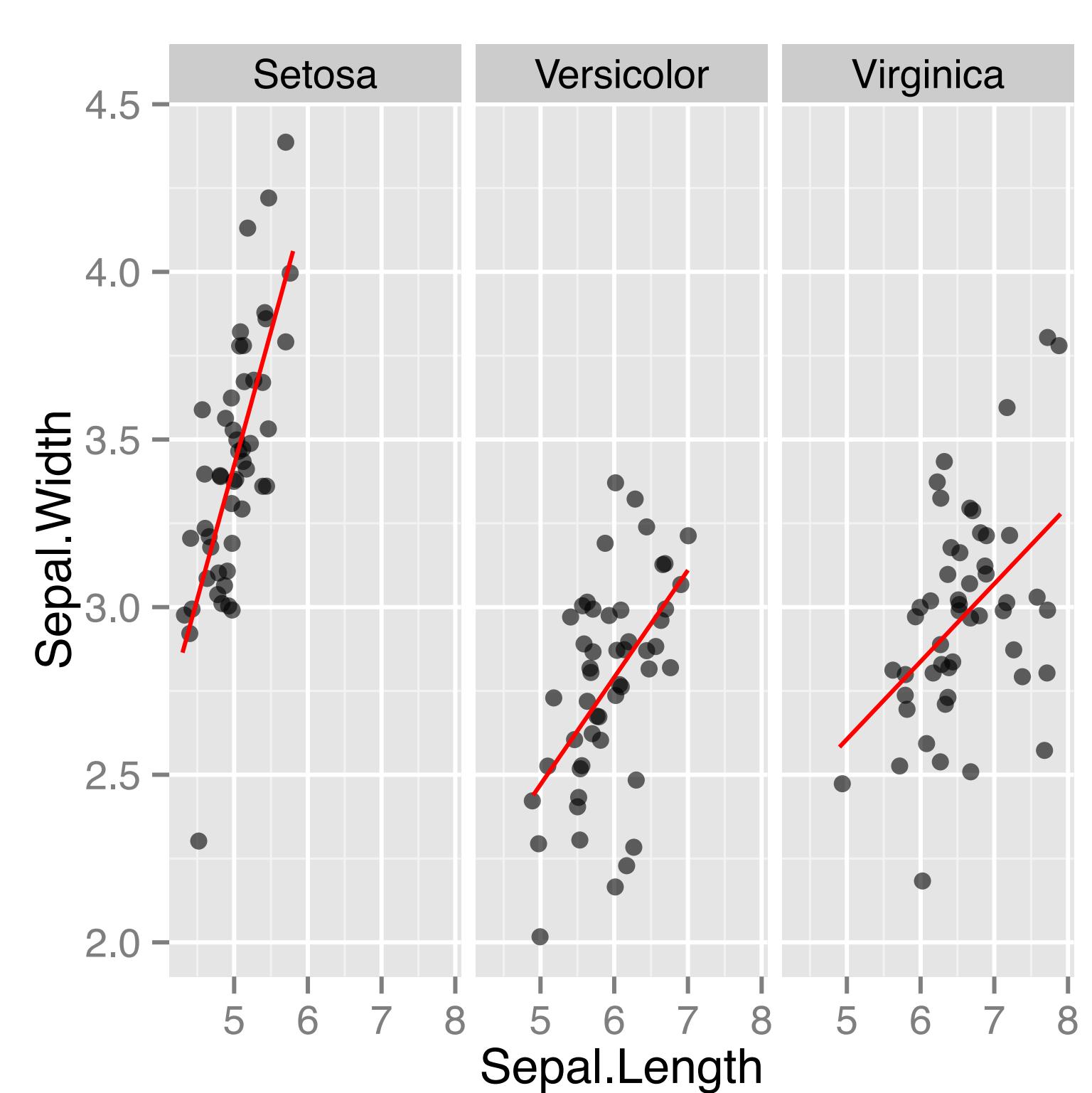


ggplot2 Layers - Statistics



ggplot2 Layers - Statistics

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_jitter(alpha = 0.6) +  
  facet_grid(. ~ Species) +  
  stat_smooth(method = "lm", se = F, col = "red")
```



ggplot2 Layers - Coordinates

Coordinates

Statistics

Facets

Geometries

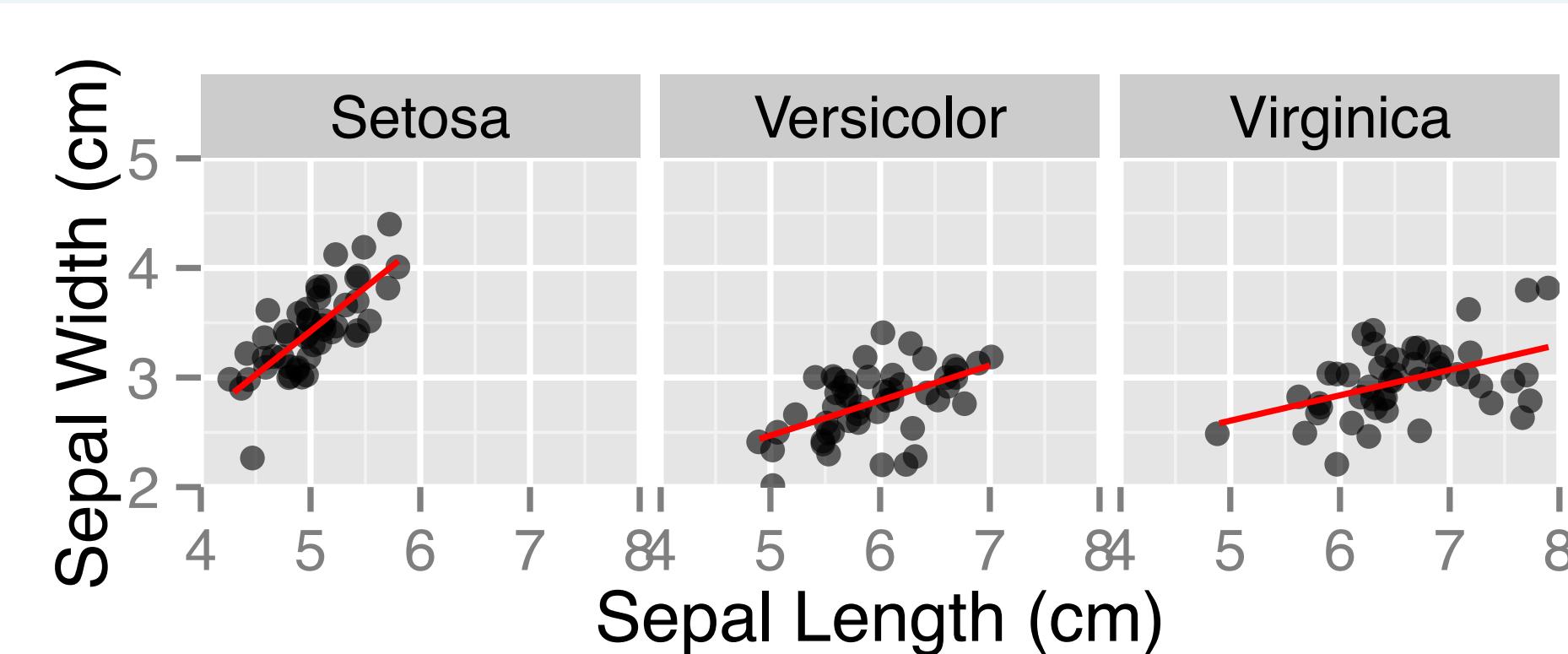
Aesthetics

Data



ggplot2 Layers - Coordinates

```
> levels(iris$Species) <- c("Setosa", "Versicolor", "Virginica")
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_jitter(alpha = 0.6) +
  facet_grid(. ~ Species) +
  stat_smooth(method = "lm", se = F, col = "red") +
  scale_y_continuous("Sepal Width (cm)",
    limits = c(2,5),
    expand = c(0,0)) +
  scale_x_continuous("Sepal Length (cm)",
    limits = c(4,8),
    expand = c(0,0)) +
  coord_equal()
```



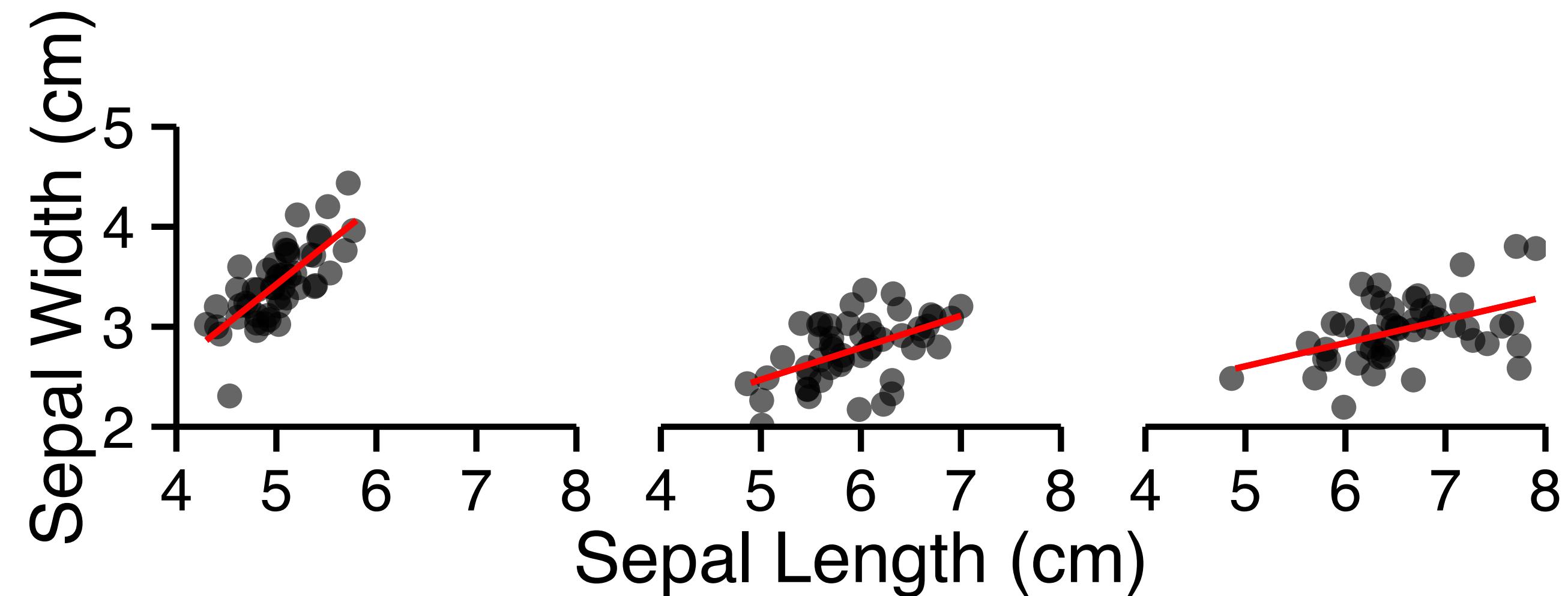
ggplot2 Layers - Themes



ggplot2 Layers - Themes

```
> levels(iris$Species) <- c("Setosa", "Versicolor", "Virginica")
> library(grid)
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  ... # previous code left out
  coord_equal() +
  theme(panel.background = element_blank(),
        plot.background = element_blank(),
        legend.background = element_blank(),
        legend.key = element_blank(),
        strip.background = element_blank(),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"),
        strip.text = element_blank(),
        panel.margin = unit(1, "lines")
  )
```

ggplot2 Layers - Themes





DATA VISUALIZATION WITH GGPLOT2

Let's practice!