The Coordinates Layer

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

datacamp

Coordinates layer

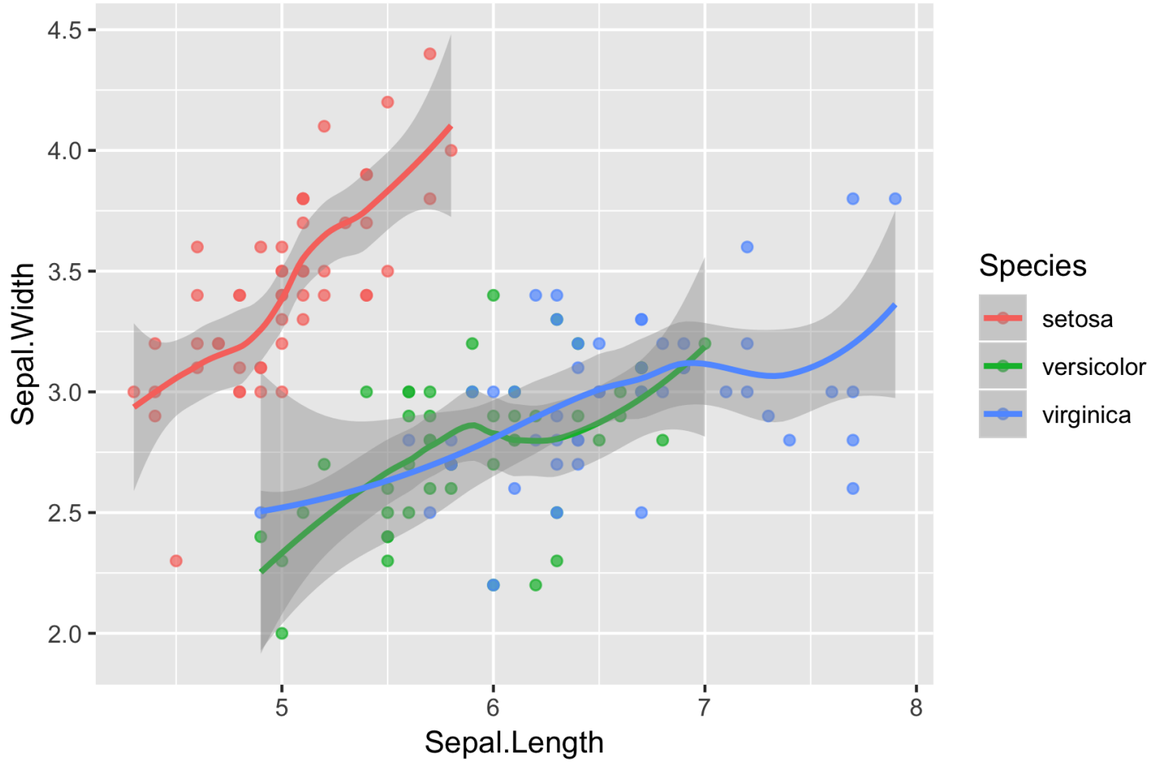
- Controls plot dimensions
- `coord_`
  - e.g. `coord_cartesian()`

Zooming in

- `coord_cartesian(xlim = ...)`
- `scale_x_continuous(limits = ...)`
- `xlim(...)`

Original plot

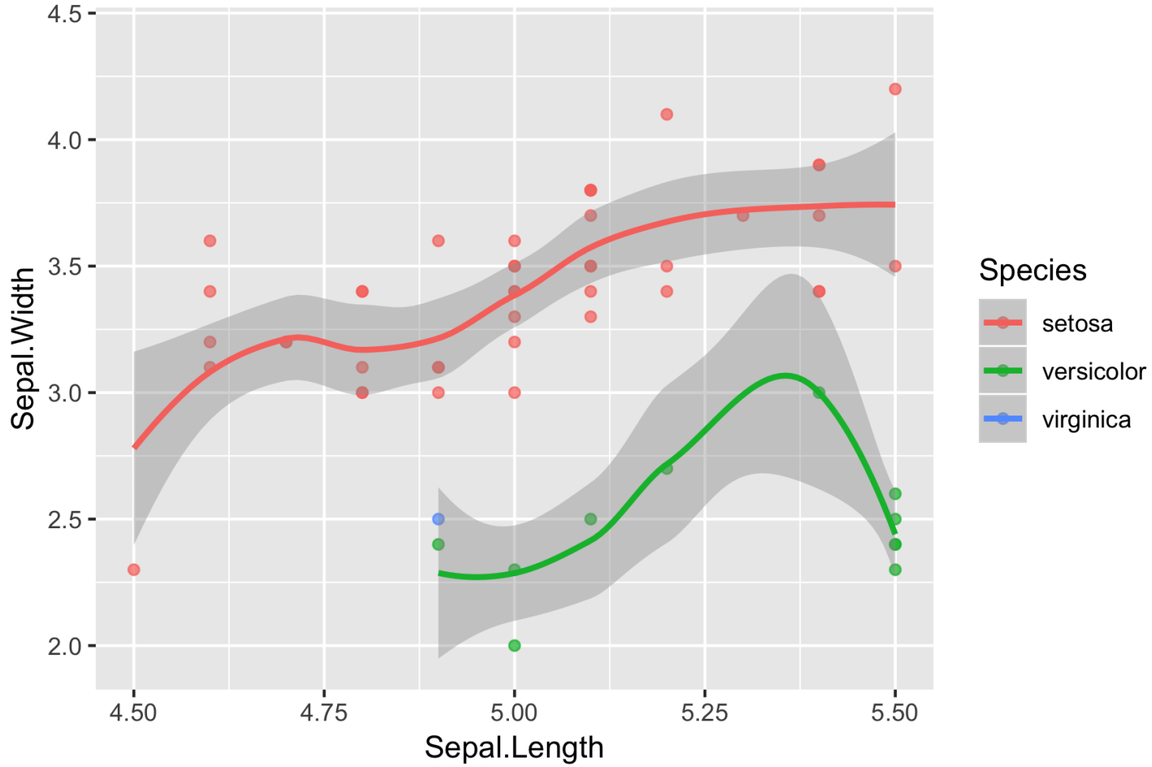
```
iris.smooth <- ggplot(
  iris,
  aes(x = Sepal.Length,
      y = Sepal.Width,
      color = Species)
  ) +
  geom_point(alpha = 0.7) +
  geom_smooth()

iris.smooth
```



scale_x_continuous()

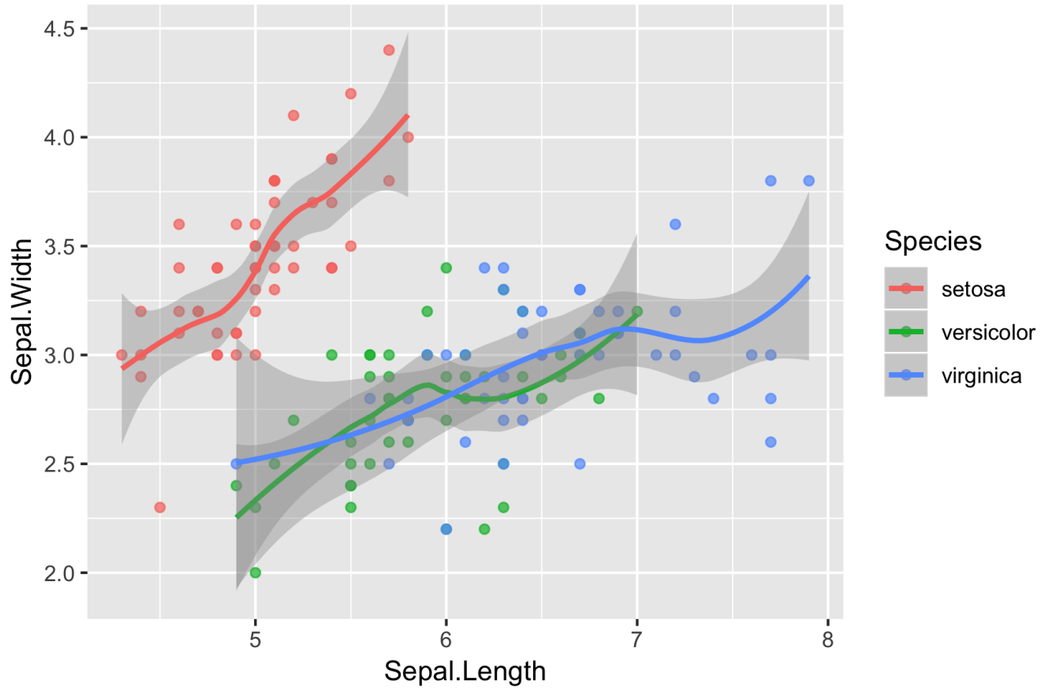
```
iris.smooth +
  scale_x_continuous(limits = c(4.5, 5.5))
```

```
Removed 95 rows containing non-finite values
(stat_smooth).
Removed 95 rows containing missing values
(geom_point).
```

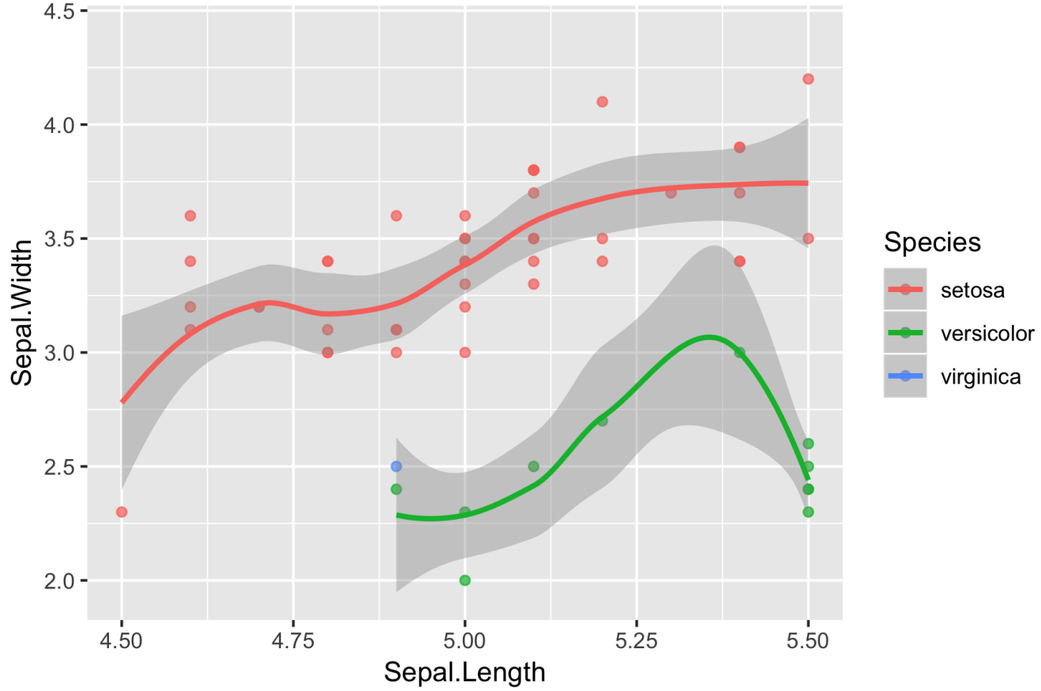


scale_x_continuous()

Original plot



Zoom in with `scale_x_continuous()`

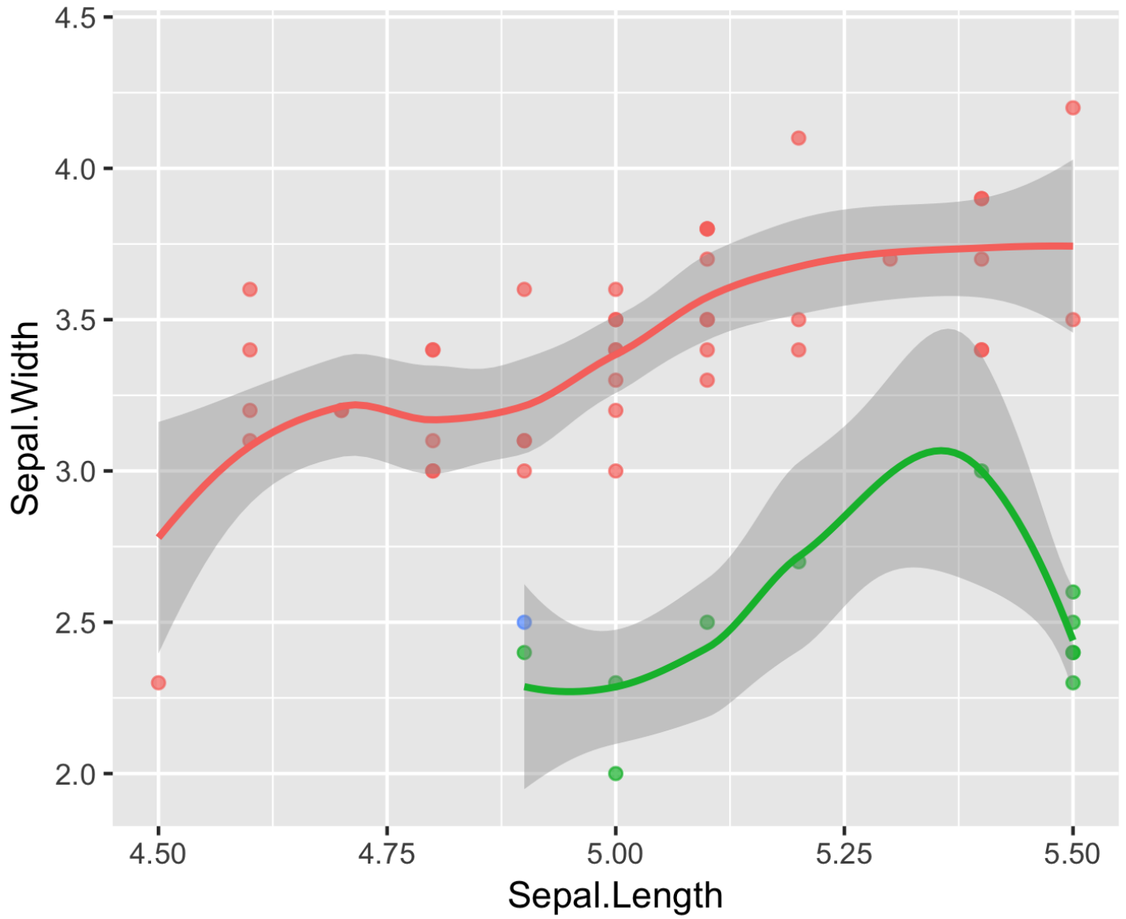


Part of original data is filtered out!

xlim()

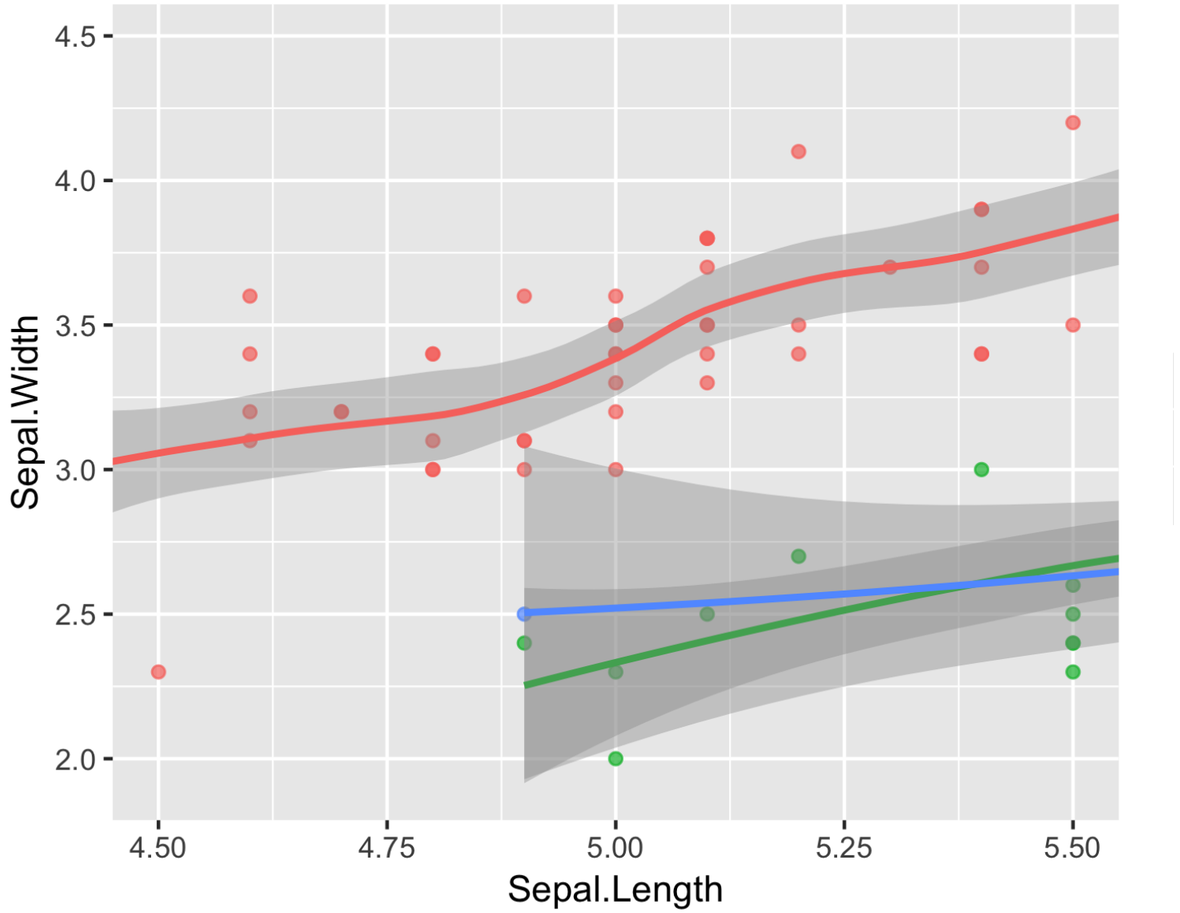
```
iris.smooth +
  xlim(c(4.5, 5.5))
```

```
Removed 95 rows containing non-finite values
(stat_smooth).
Removed 95 rows containing missing values
(geom_point).
```



coord_cartesian()

```
iris.smooth +
  coord_cartesian(xlim = c(4.5, 5.5))
```

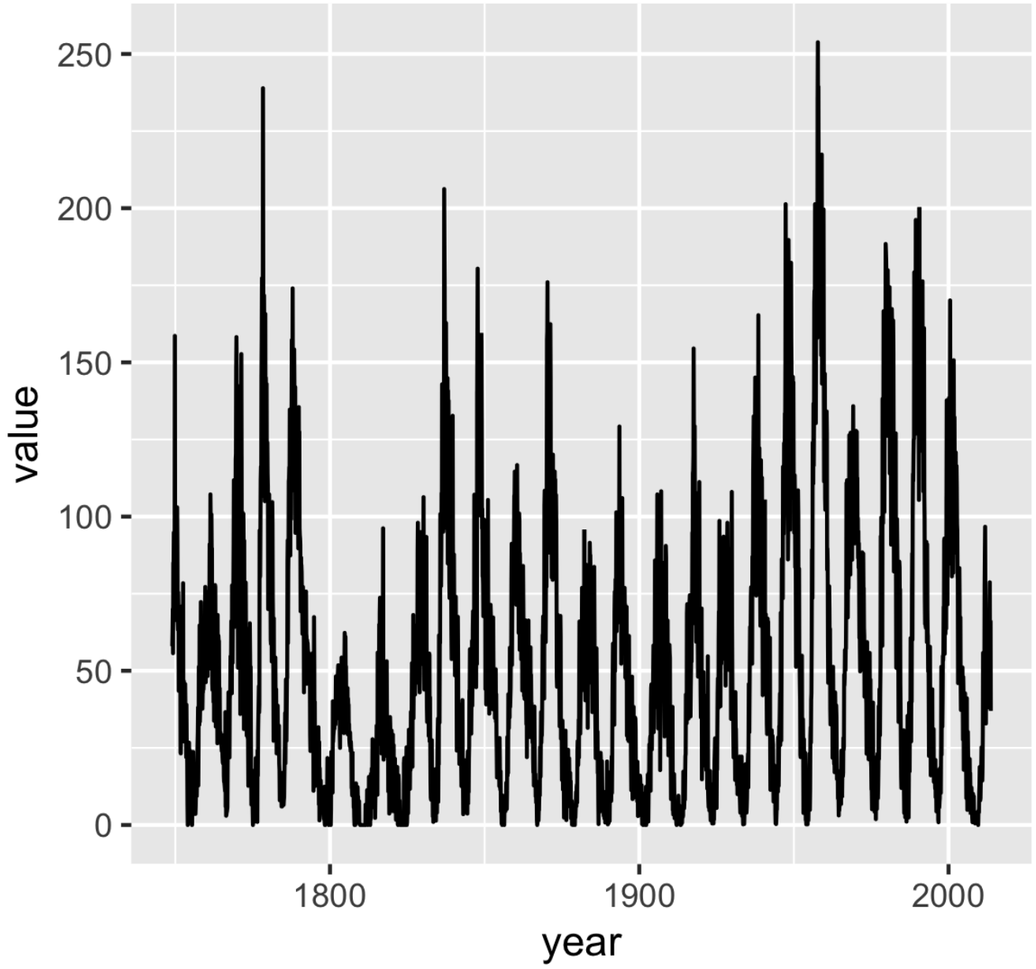


Aspect ratio

- Height-to-width ratio
- Watch out for deception!
- No universal standard so far
- Typically use 1:1 if data is on the same scale

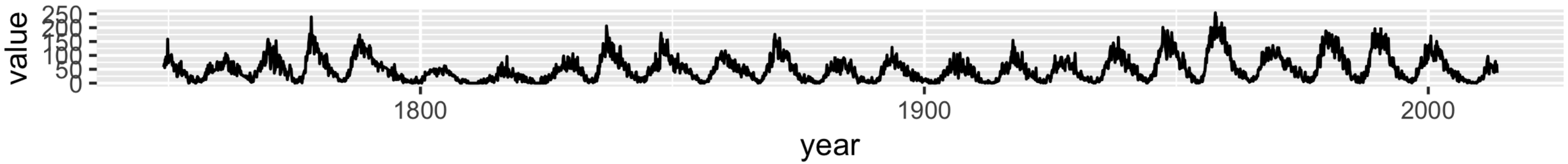
Sunspots

```
library(zoo)
sunspots.m <- data.frame(
    year = index(sunspot.month),
    value = reshape2::melt(sunspot.month)$value)
)

ggplot(sunspots.m, aes(x = year, y = value)) +
  geom_line() +
  coord_fixed() # default to 1:1 aspect ratio
```



Sunspots

```
ggplot(sunspots.m, aes(x = year, y = value)) +
  geom_line() +
  coord_fixed(0.055)
```



Practice time!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

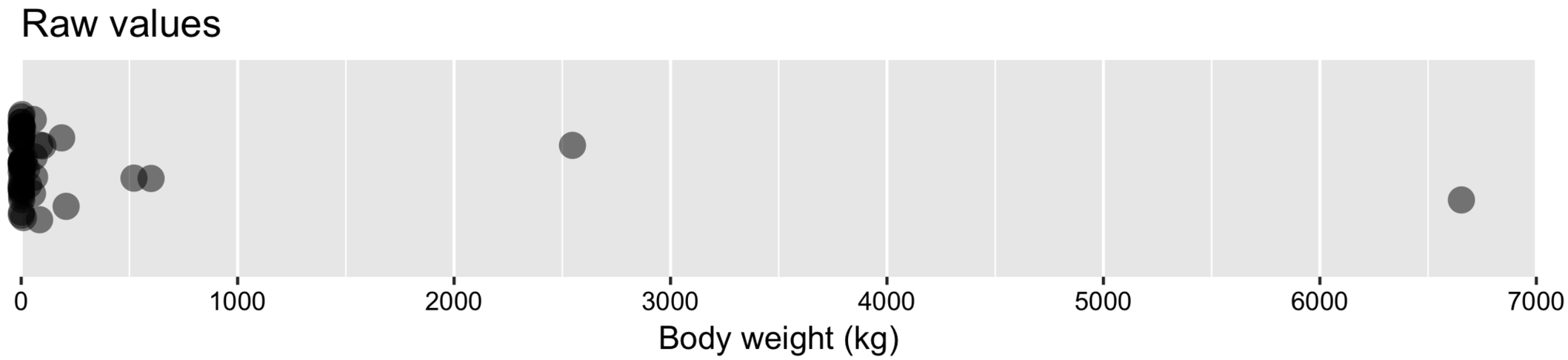
Coordinates vs. scales

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

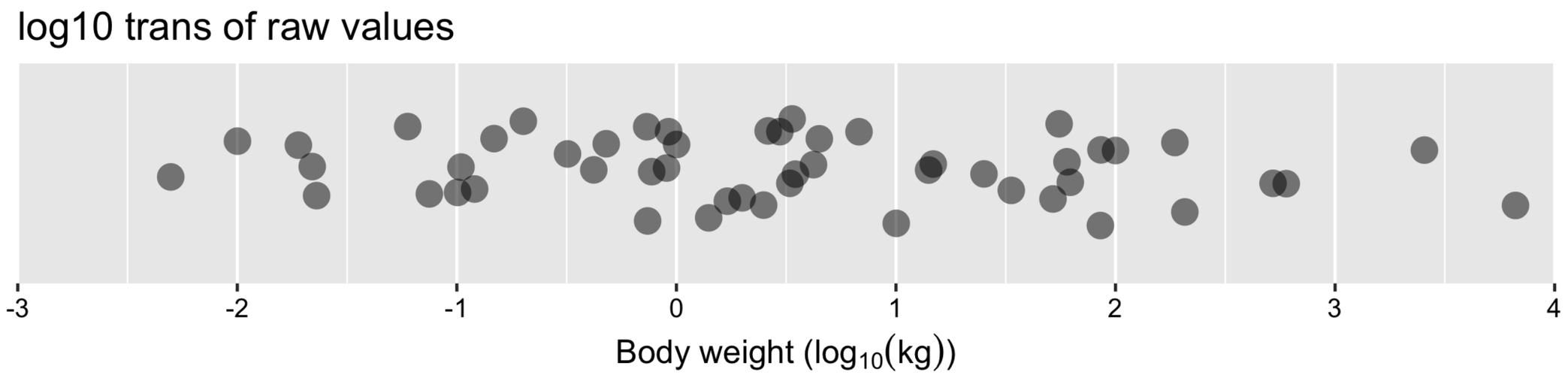
Plot the raw data

```
ggplot(msleep, aes(bodywt, y = 1)) +
  geom_jitter() +
  scale_x_continuous(limits = c(0, 7000),
                     breaks = seq(0, 7000, 1000))
```



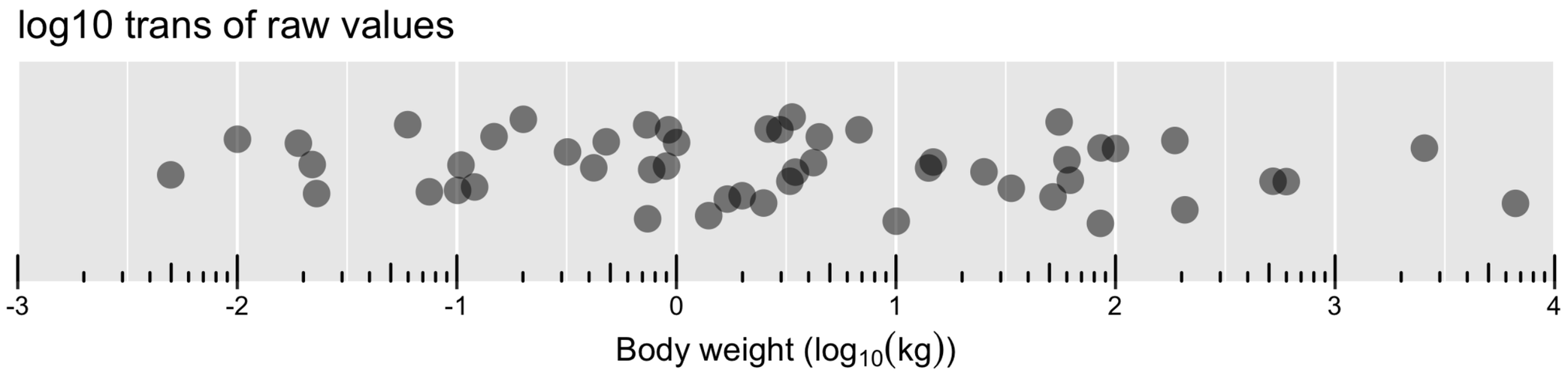
Transform the raw data

```
ggplot(msleep, aes(log10(bodywt), y = 1)) +
  geom_jitter() +
  scale_x_continuous(limits = c(-3, 4),
                     breaks = -3:4)
```



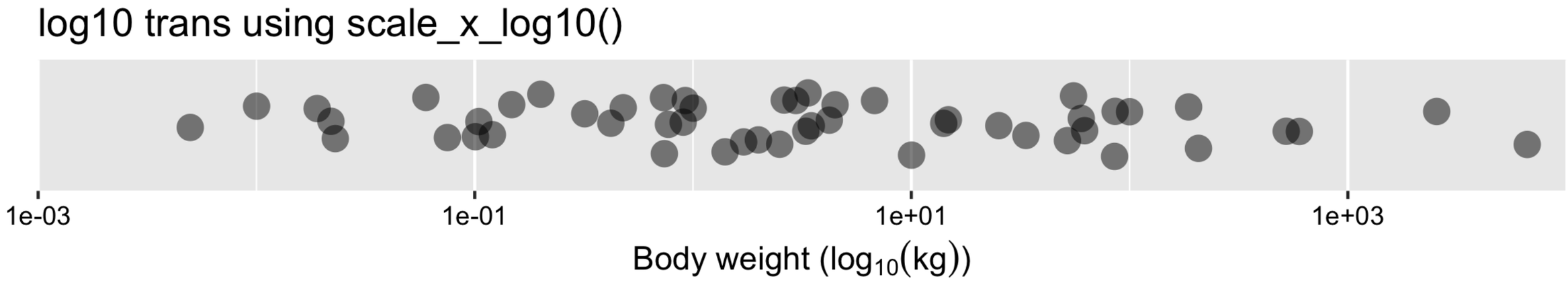
Add logtick annotation

```
ggplot(msleep, aes(log10(bodywt), y = 1)) +
  geom_jitter() +
  scale_x_continuous(limits = c(-3, 4),
                     breaks = -3:4) +
  annotation_logticks(sides = "b")
```

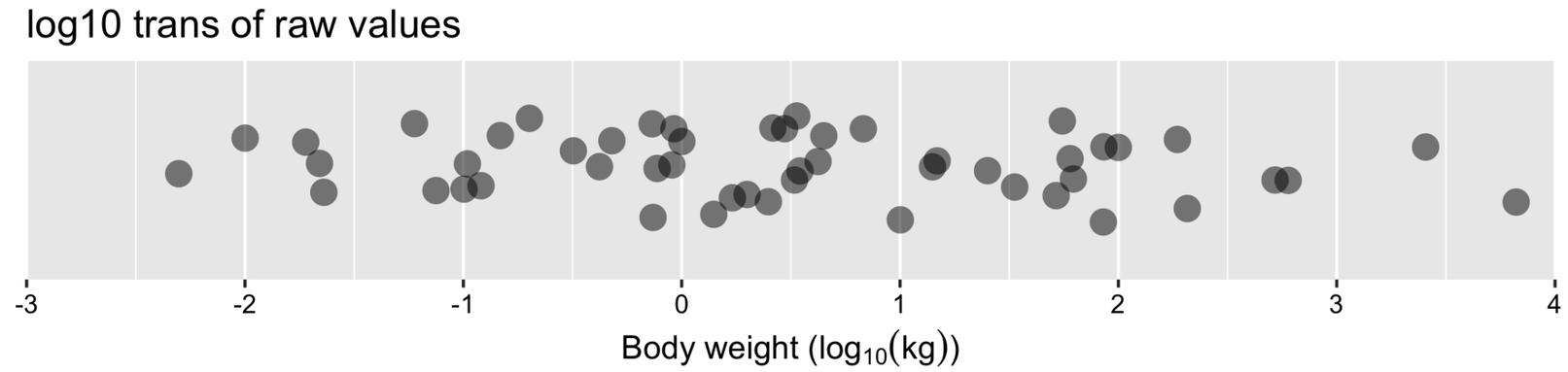


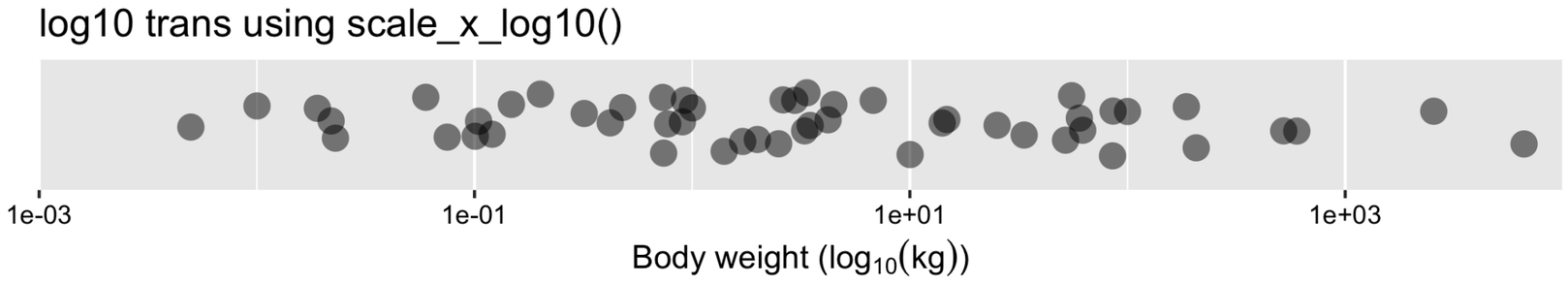
Use scale_*_log10()

```
ggplot(msleep, aes(bodywt, y = 1)) +
  geom_jitter() +
  scale_x_log10(limits = c(1e-03, 1e+04))
```



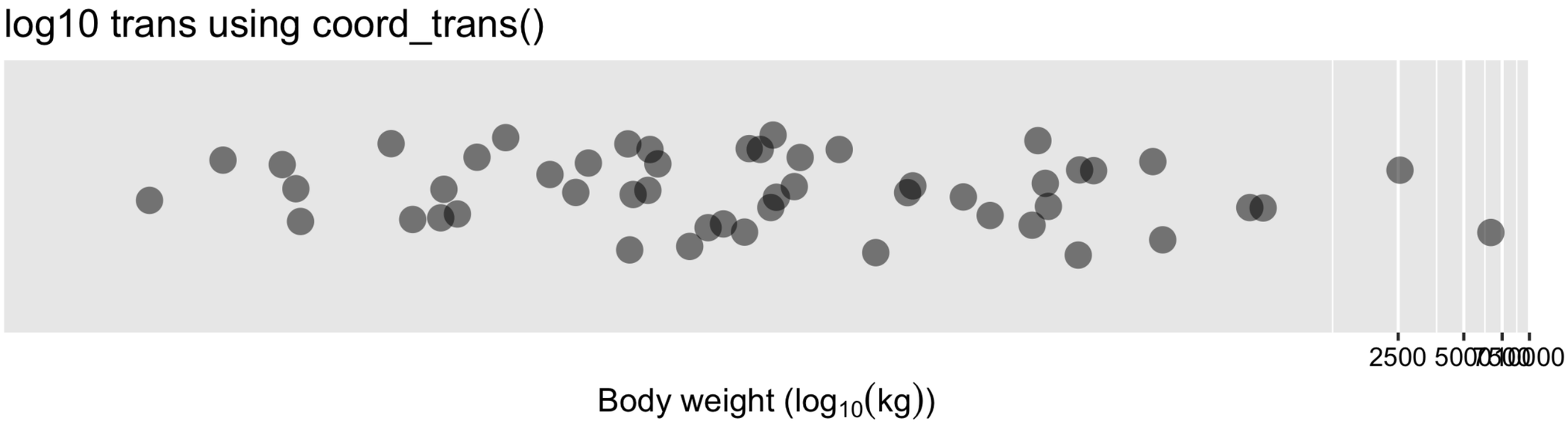
Compare direct transform and scale_*_log10() output



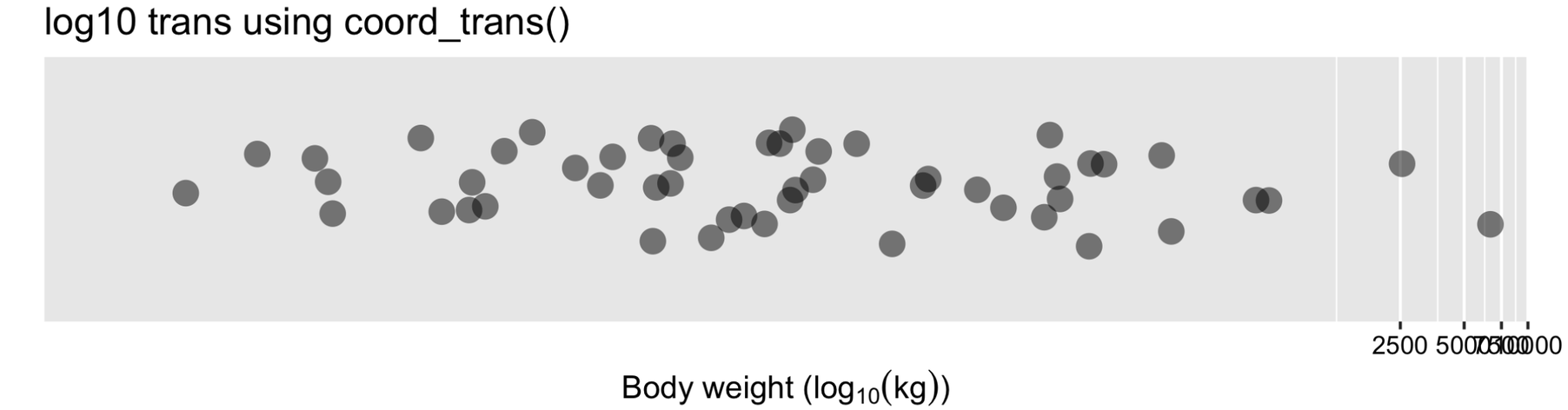


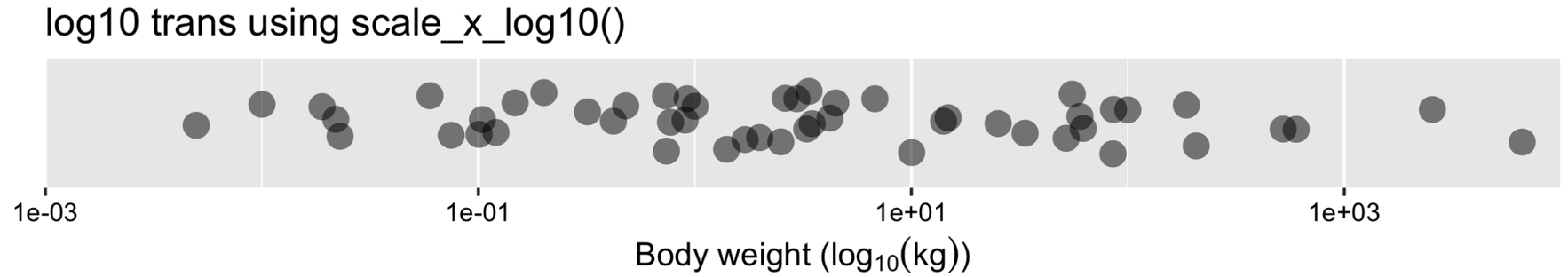
Use coord_trans()

```
ggplot(msleep, aes(bodywt, y = 1)) +
  geom_jitter() +
  coord_trans(x = "log10")
```



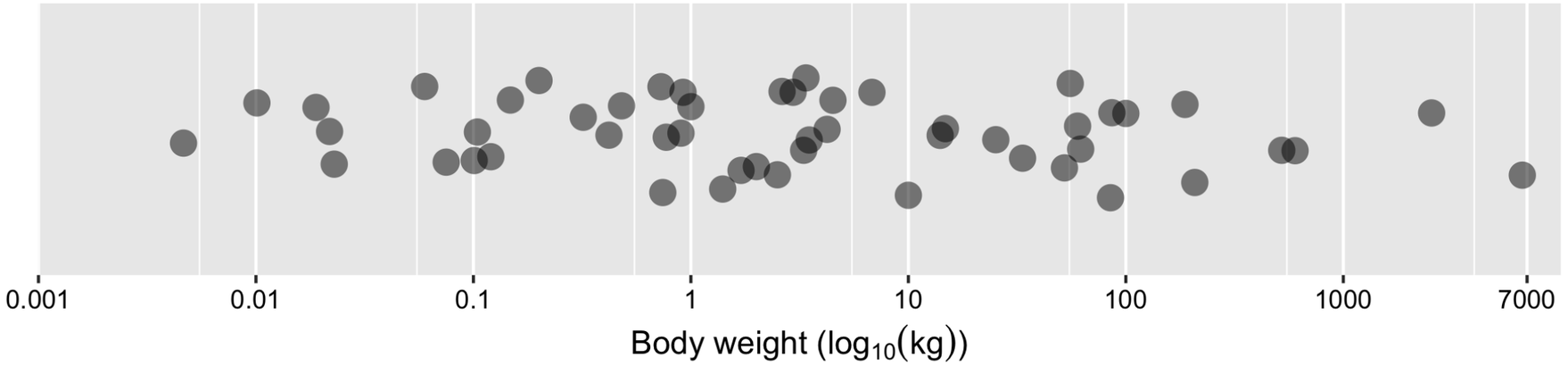
Compare scale_*_log10() and coord_trans() output



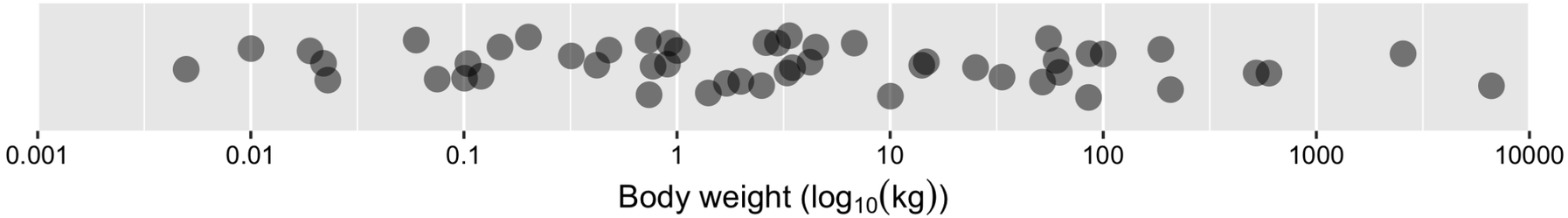


Adjusting labels

log10 trans using coord_trans()



log10 trans using scale_x_log10()



Time for exercises

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Double and flipped axes

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

Typical axis modifications

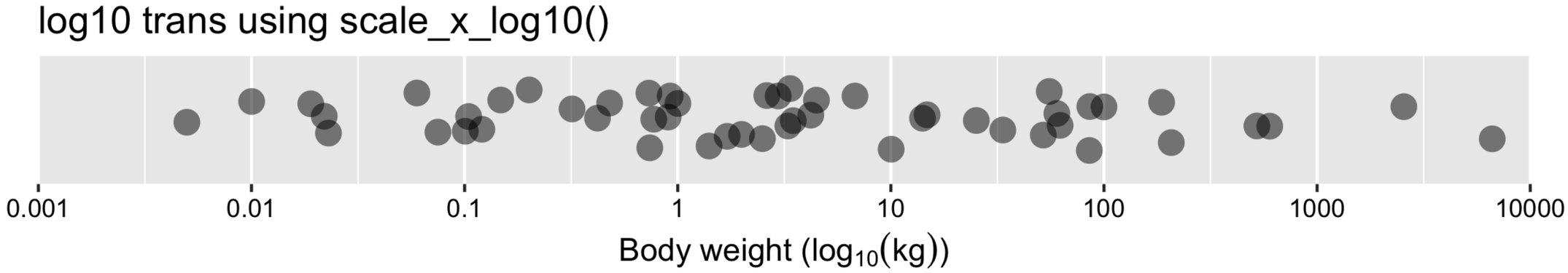
- Aspect ratios (see video 1)
  - Adjust for best perspective
- Transformation functions (e.g. log, see video 2)
  - Adjust if original scale is inappropriate
- Double x or y axes
  - Add raw and transformed values
- Flipped axes
  - Change direction of dependencies
  - Change geometry orientation

Typical axis modifications

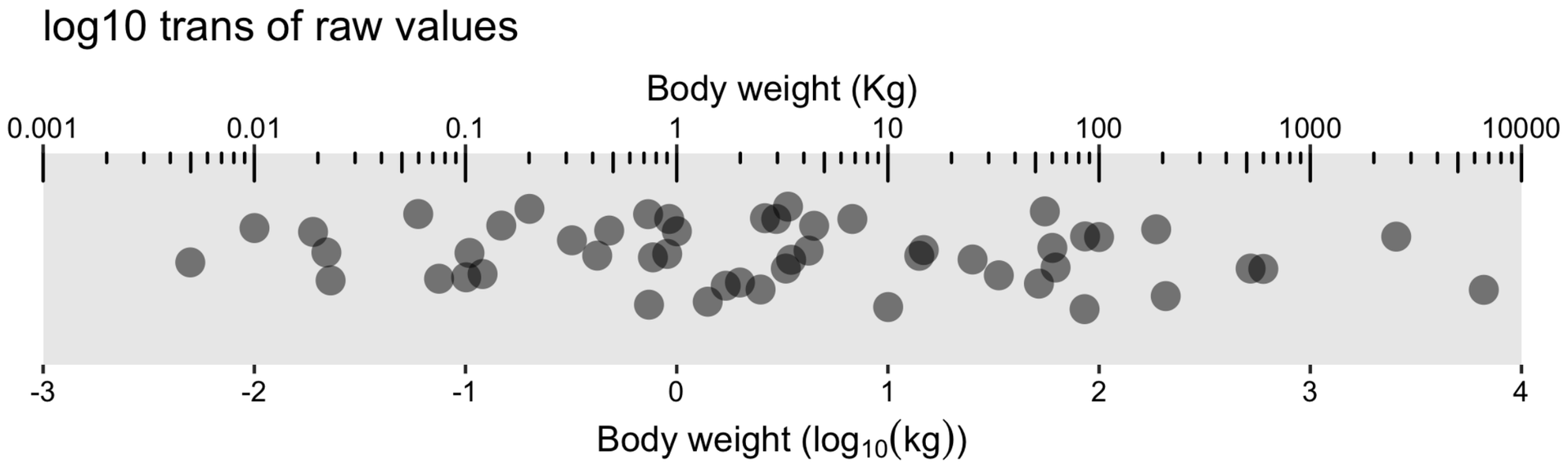
- Aspect ratios (see video 1)
  - Adjust for best perspective
- Transformation functions (e.g. log, see video 2)
  - Adjust if original scale is inappropriate
- **Double x or y axes**
  - **Add raw and transformed values**
- Flipped axes
  - Change direction of dependencies
  - Change geometry orientation

[1] See chapter 4, video 3 for more discussion on double x and y-axes.

Double axes


log10 trans using scale_x_log10()
0.001
0.01
0.1
1
10
100
1000
10000
Body weight (log10(kg))

Adding raw and transformed axes

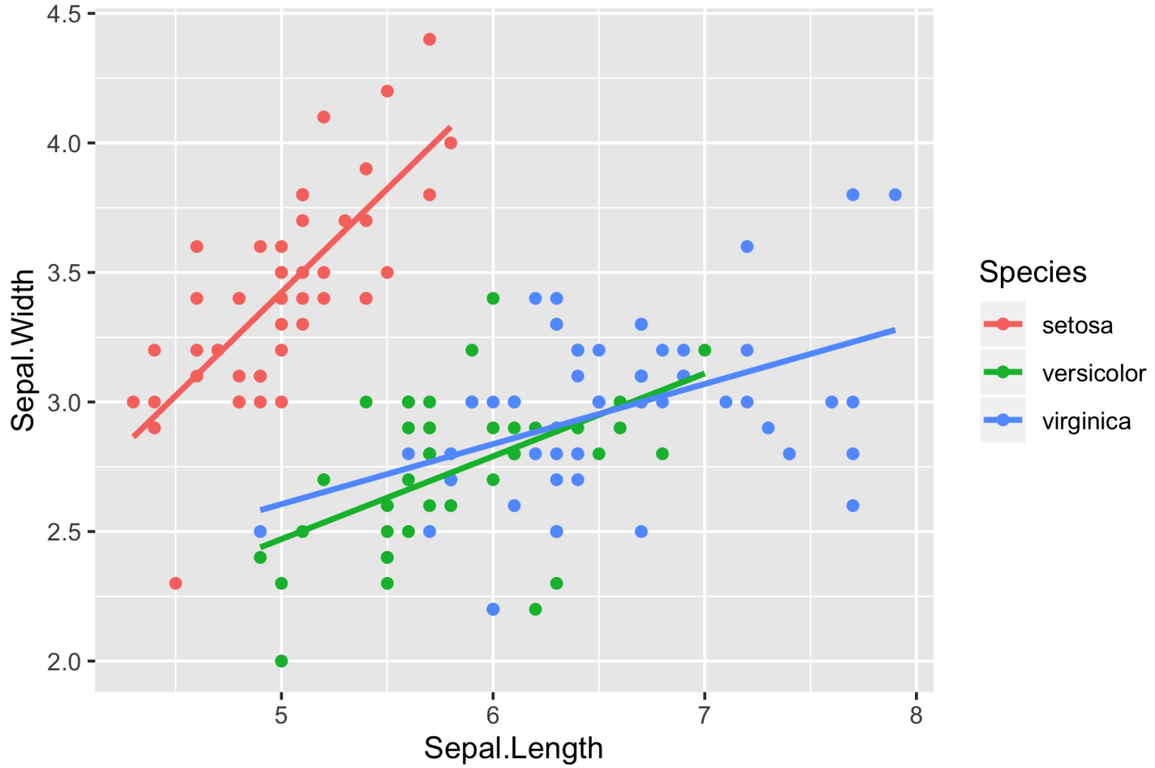


Typical axis modifications

- Aspect ratios (see video 1)
  - Adjust for best perspective
- Transformation functions (e.g. log, see video 2)
  - Adjust if original scale is inappropriate
- Double x or y axes
  - Add raw and transformed values
- **Flipped axes**
  - **Change direction of dependencies**
  - **Change geometry orientation**

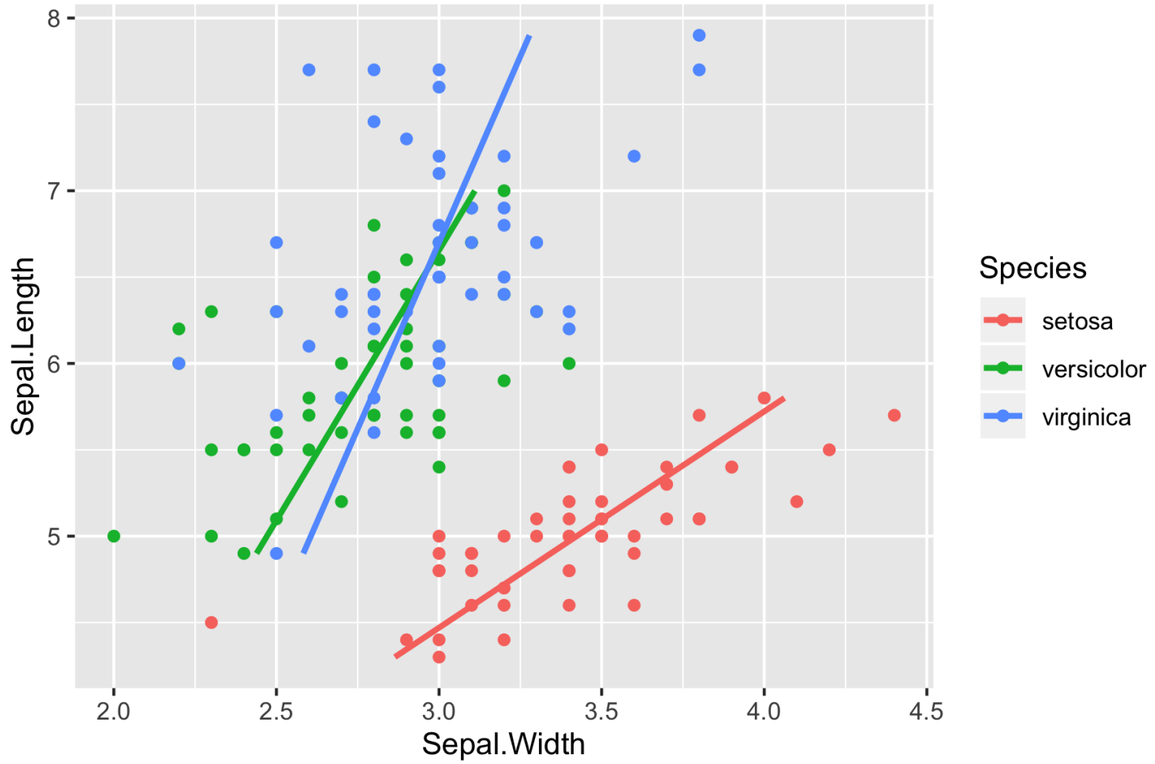
Flipping axes

```
ggplot(iris, aes(x = Sepal.Length,
                 y = Sepal.Width,
                 color = Species)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE)
```



coord_flip()

```
ggplot(iris, aes(x = Sepal.Length,
                 y = Sepal.Width,
                 color = Species)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE) +
  coord_flip()
```



Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Polar coordinates

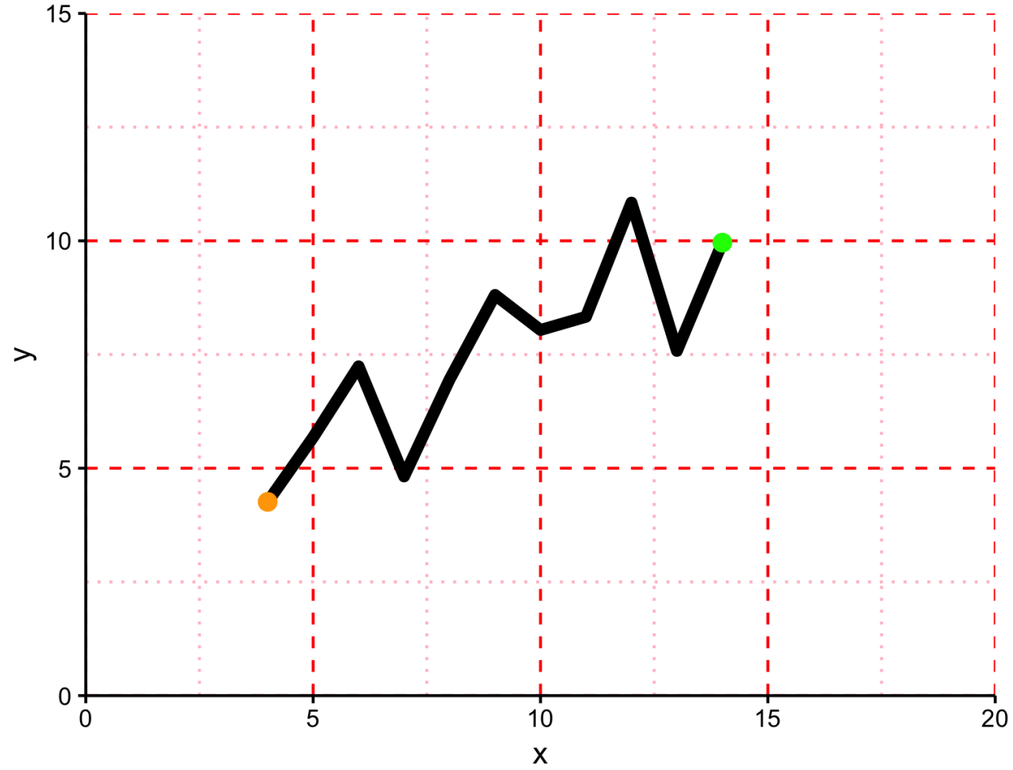
INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

datacamp

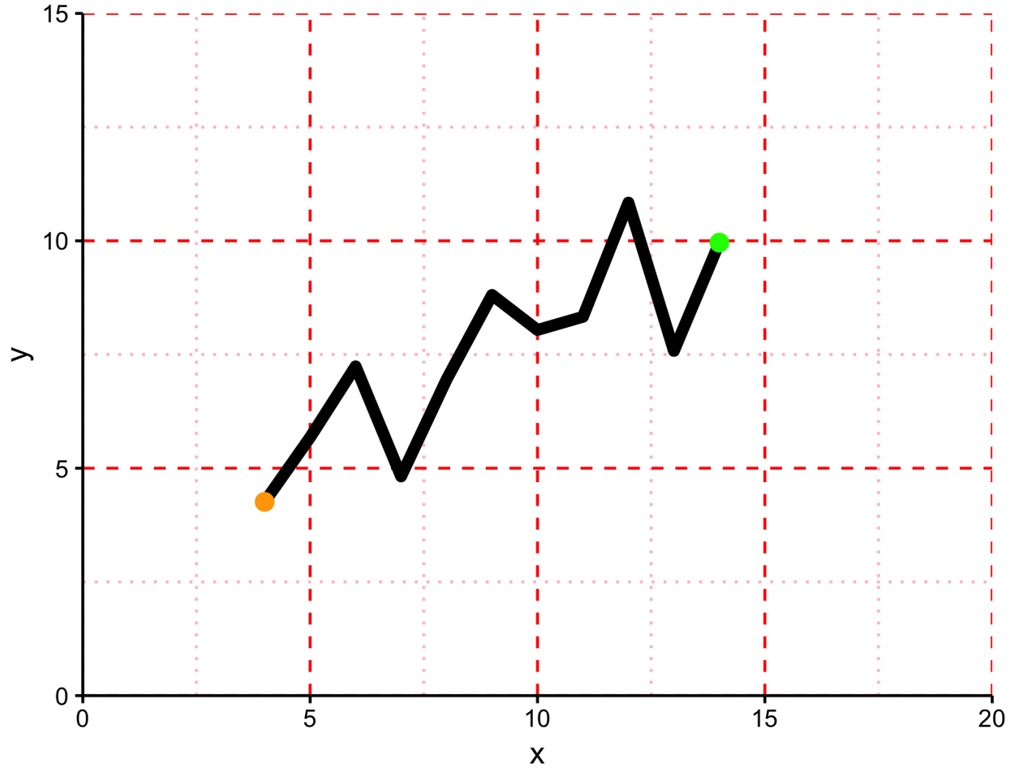
Projections control perception

- Cartesian (2d)
  - Orthogonal x and y-axes
  - Modify axis limits and aspect ratio



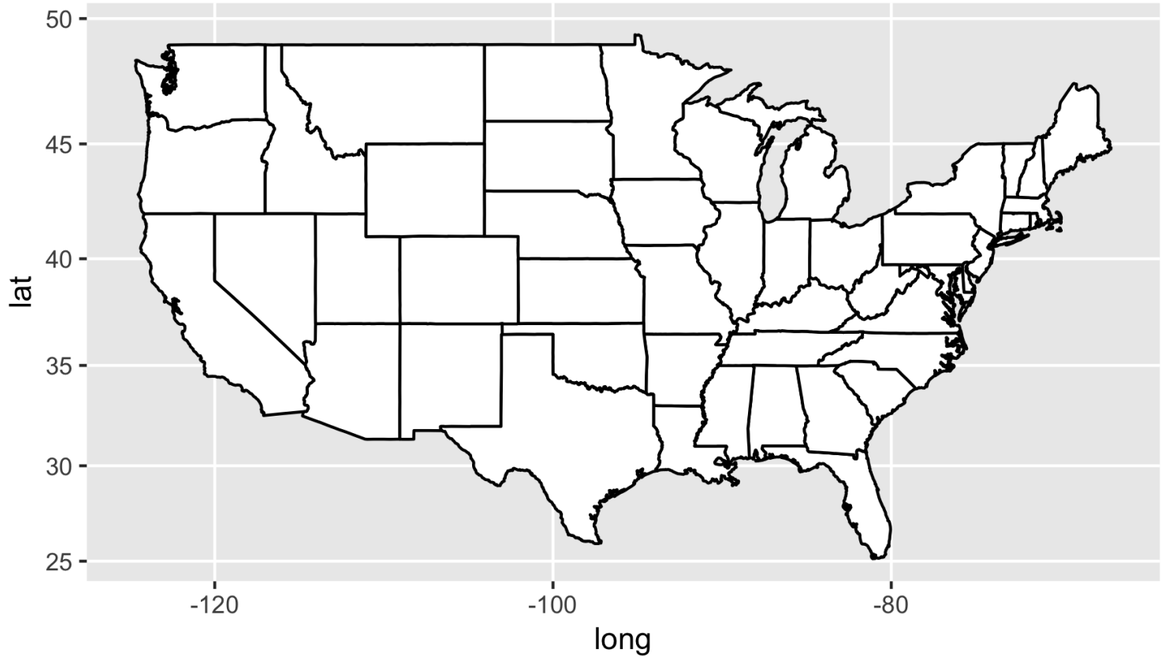
Projections control perception

- Cartesian (2d)
  - Orthogonal x and y-axes
  - Modify axis limits and aspect ratio
- Maps
  - Many possible projections
  - See next course

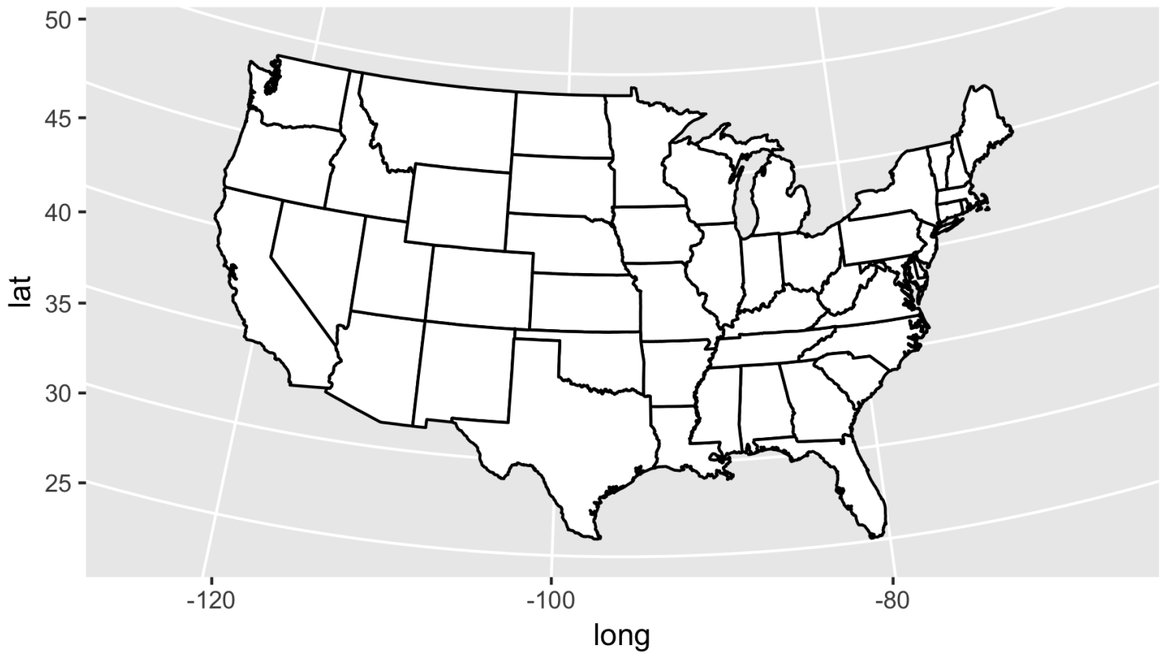


A preview of map projections

The Mercator Projection



The Conic Projection

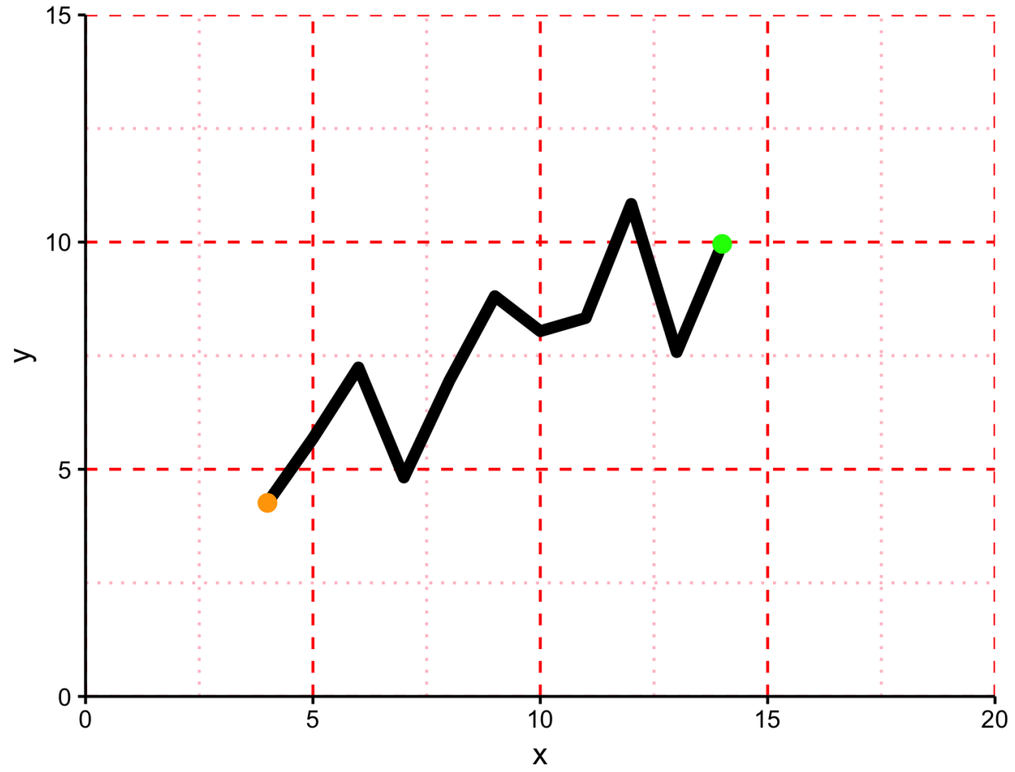


Polar coordinates

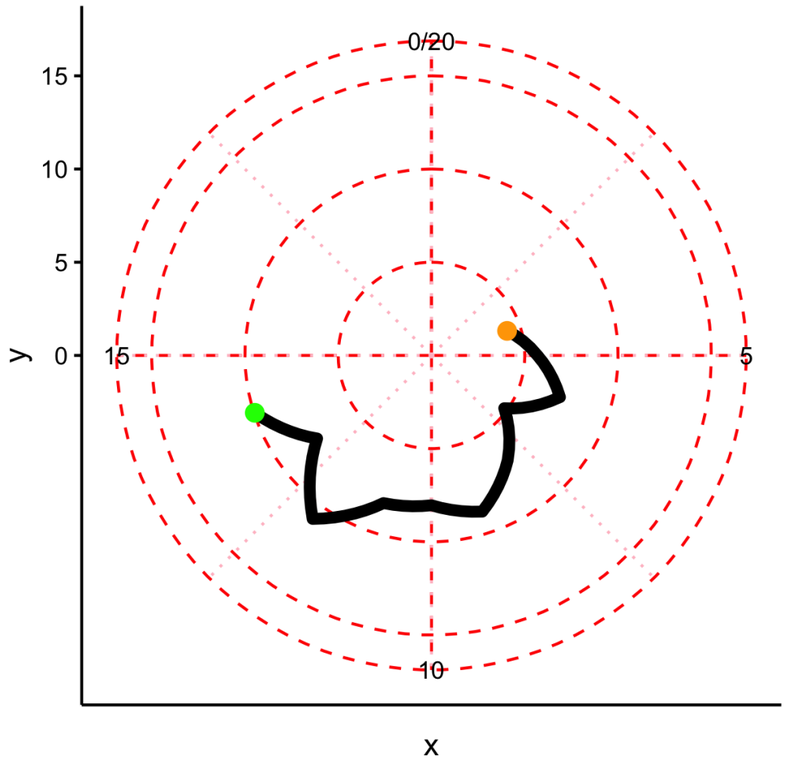
- Cartesian (2d)
  - Orthogonal x and y-axes.
- Maps
  - Many projections, see next course
- Polar
  - Transformed Cartesian space

coord_polar()

```
p + coord_fixed()
```

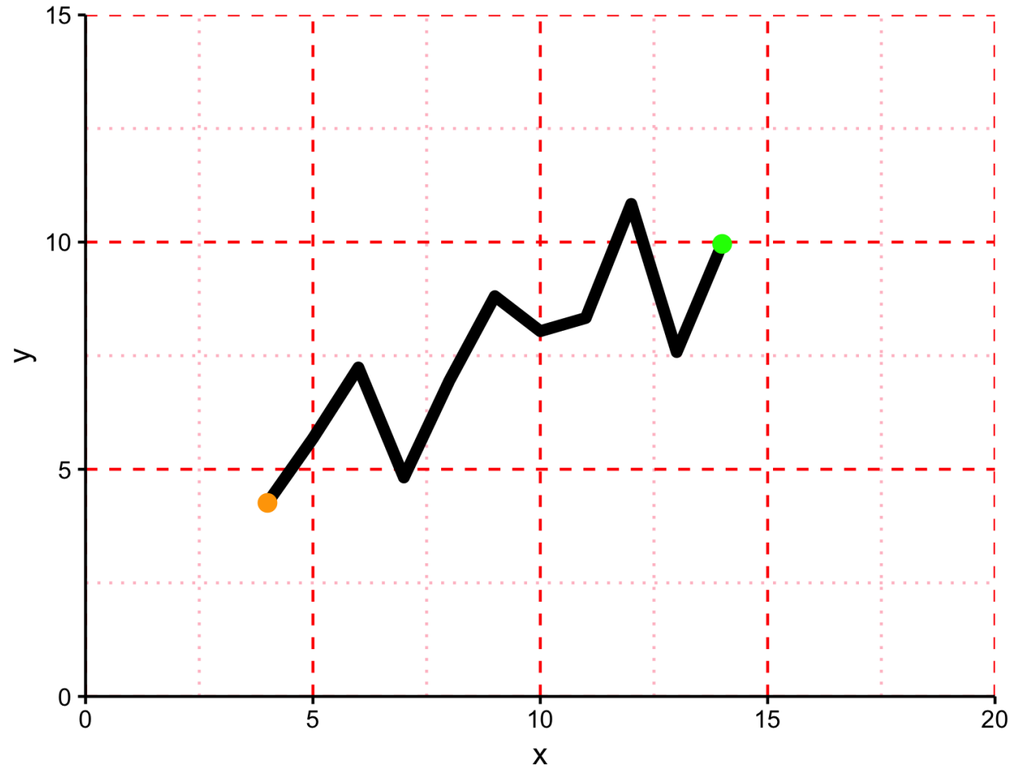



```
p + coord_polar()
```

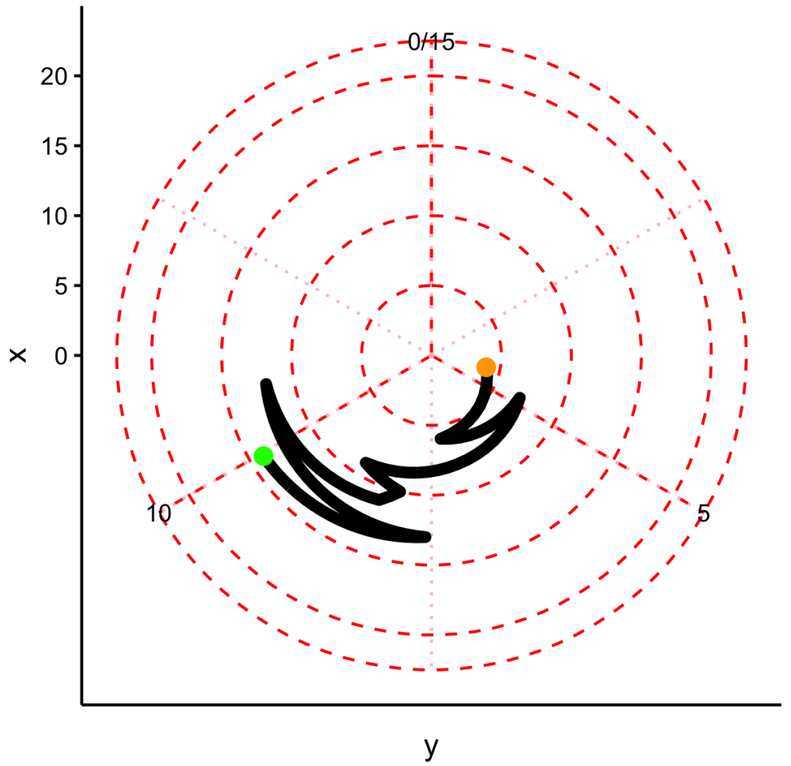


coord_polar(theta = "y")

```
p + coord_fixed()
```




```
p + coord_polar(theta = "y")
```



Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2