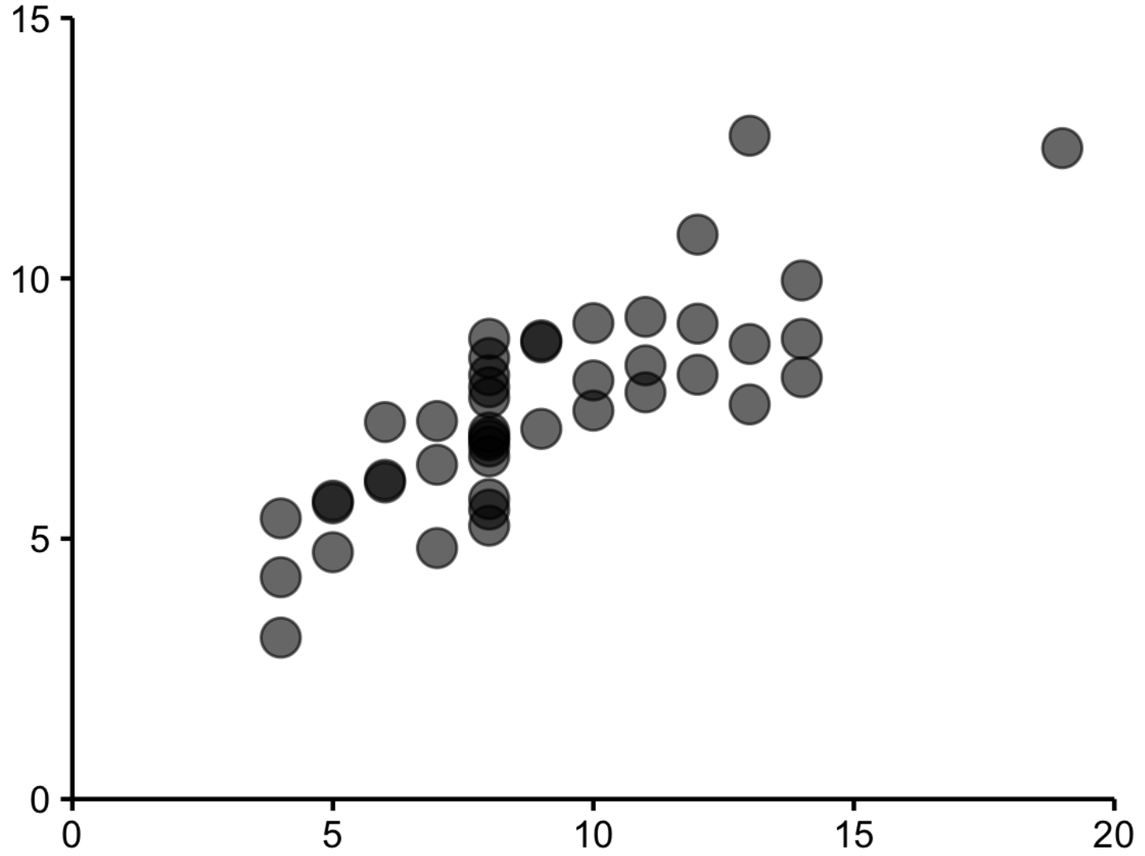
The facets layer

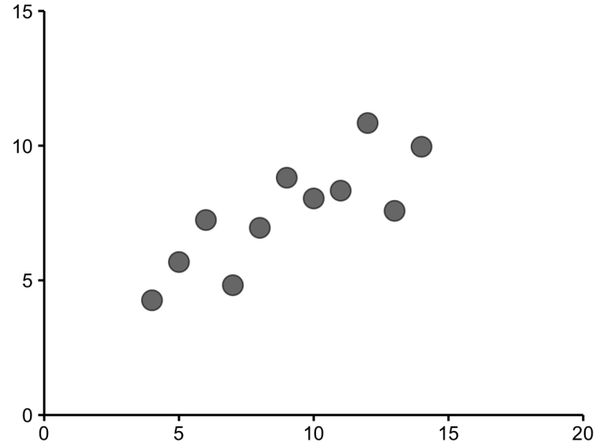
INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

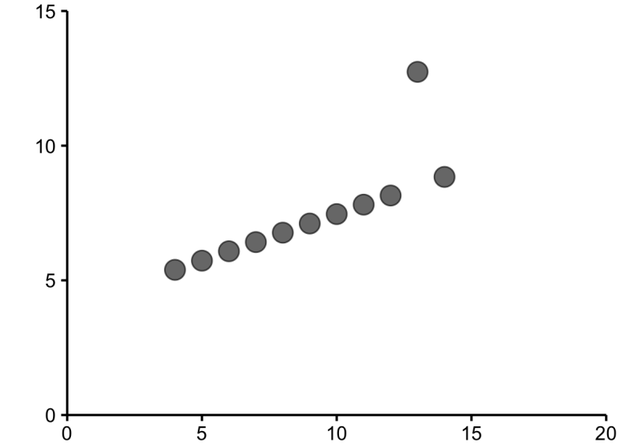
Rick Scavetta
Founder, Scavetta Academy

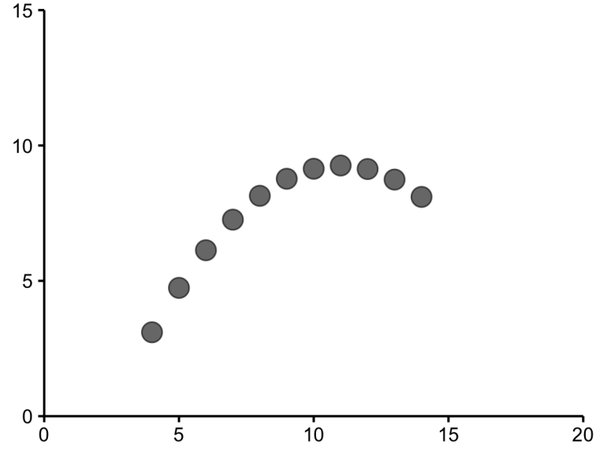
Facets

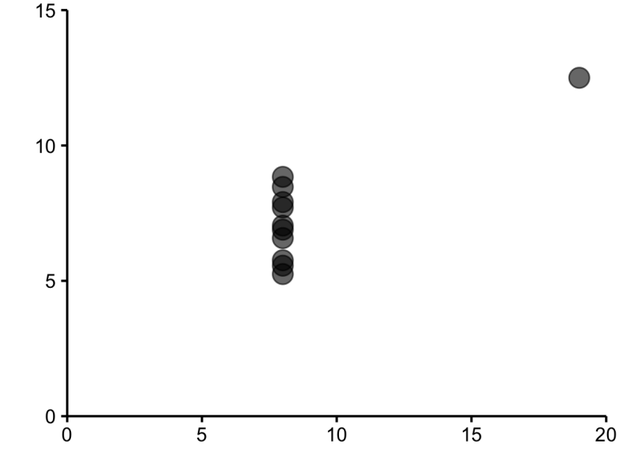
- Straight-forward yet useful
- Concept of Small Multiples
  - Popularized by Edward Tufte
  - Visualization of Quantitative Information, 1983


15
10
5
0
0
5
10
15
20


15
10
5
0
0
5
10
15
20


15
10
5
0
0
5
10
15
20

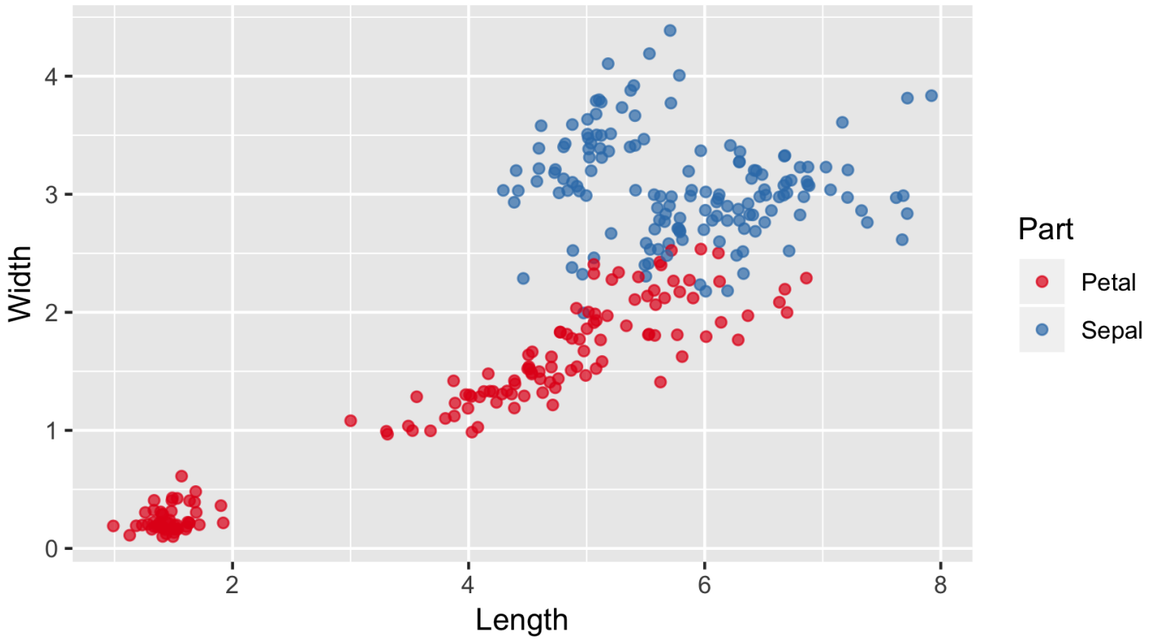

15
10
5
0
0
5
10
15
20


15
10
5
0
0
5
10
15
20

iris.wide

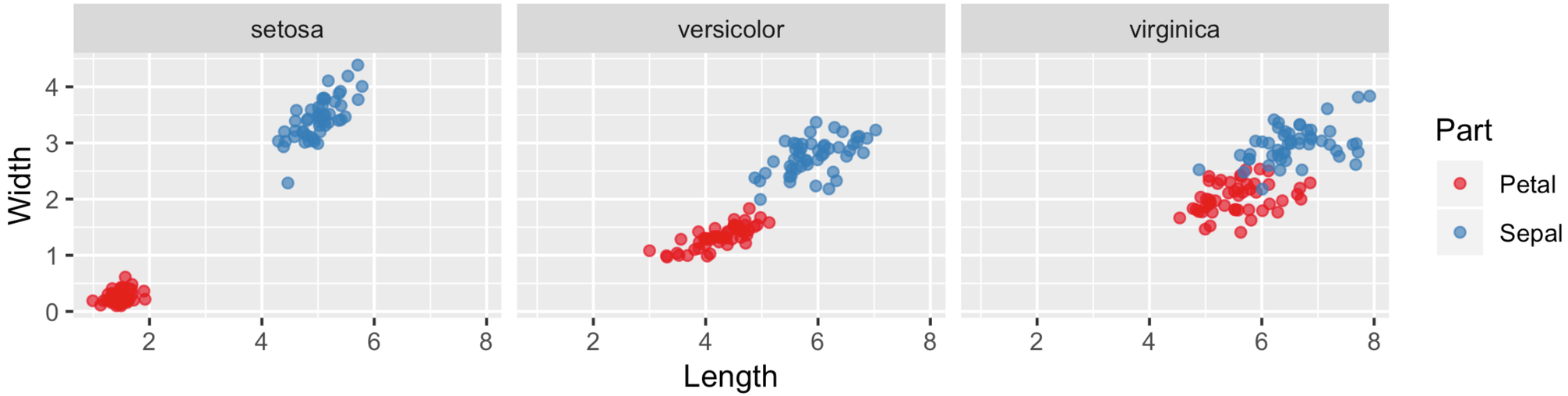
```
p <- ggplot(iris.wide, aes(x = Length,
                           y = Width,
                           ccolorol = Part)) +
  geom_jitter(alpha = 0.7) +
  scale_color_brewer(palette = "Set1") +
  coord_fixed()

p
```



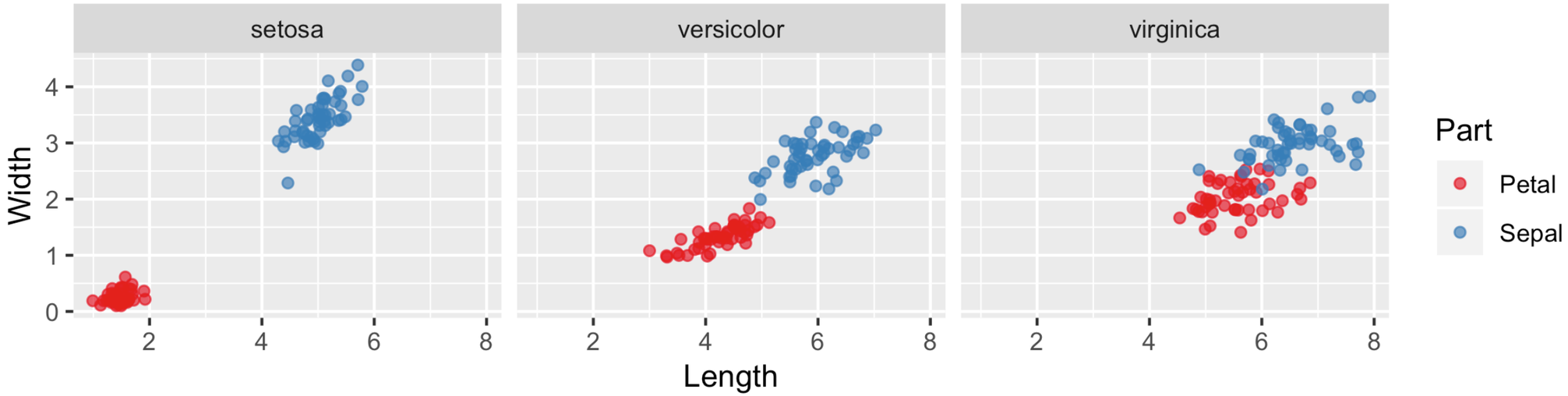
iris.wide & facet_grid()

```
p <- ggplot(iris.wide, aes(x = Length, y = Width, color = Part)) +
  geom_jitter(alpha = 0.7) +
  scale_color_brewer(palette = "Set1") +
  coord_fixed()
p + facet_grid(cols = vars(Species))
```



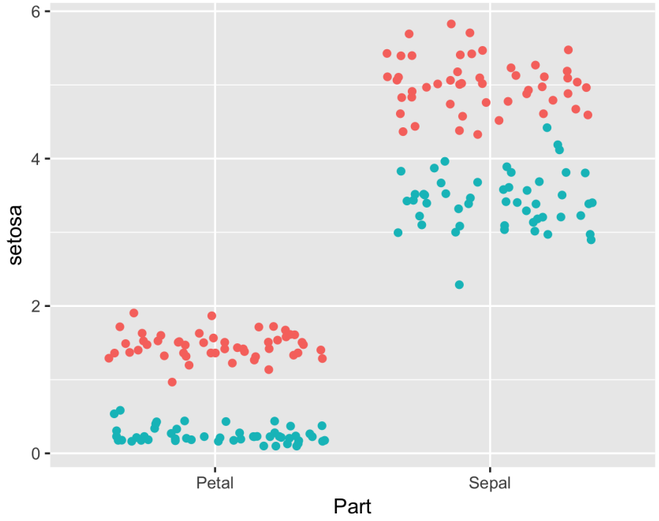
Formula notation

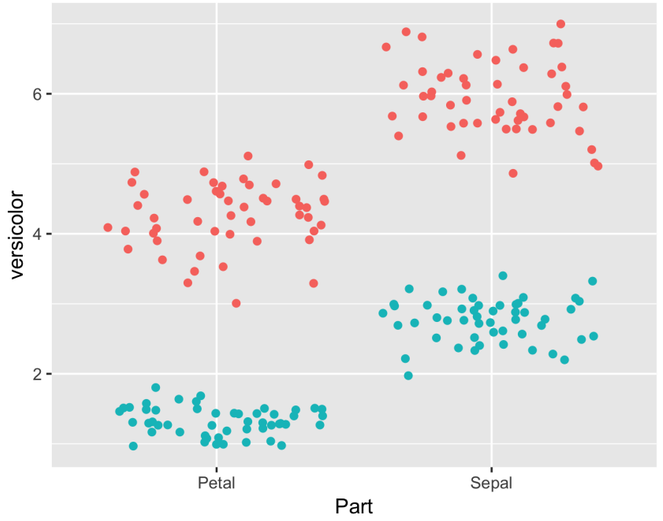
```
p <- ggplot(iris.wide, aes(x = Length, y = Width, color = Part)) +
  geom_jitter(alpha = 0.7) +
  scale_color_brewer(palette = "Set1") +
  coord_fixed()
p + facet_grid(. ~ Species)
```

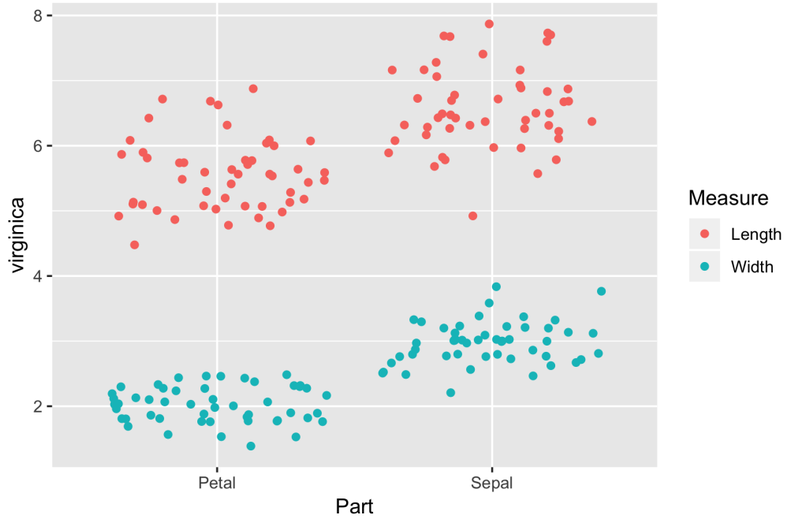


iris.wide2

```
ggplot(iris.wide2, aes(x = Part, y = setosa, color = Measure)) +
  geom_jitter()
ggplot(iris.wide2, aes(x = Part, y = versicolor, color = Measure)) +
  geom_jitter()
ggplot(iris.wide2, aes(x = Part, y = virginica, color = Measure)) +
  geom_jitter()
```

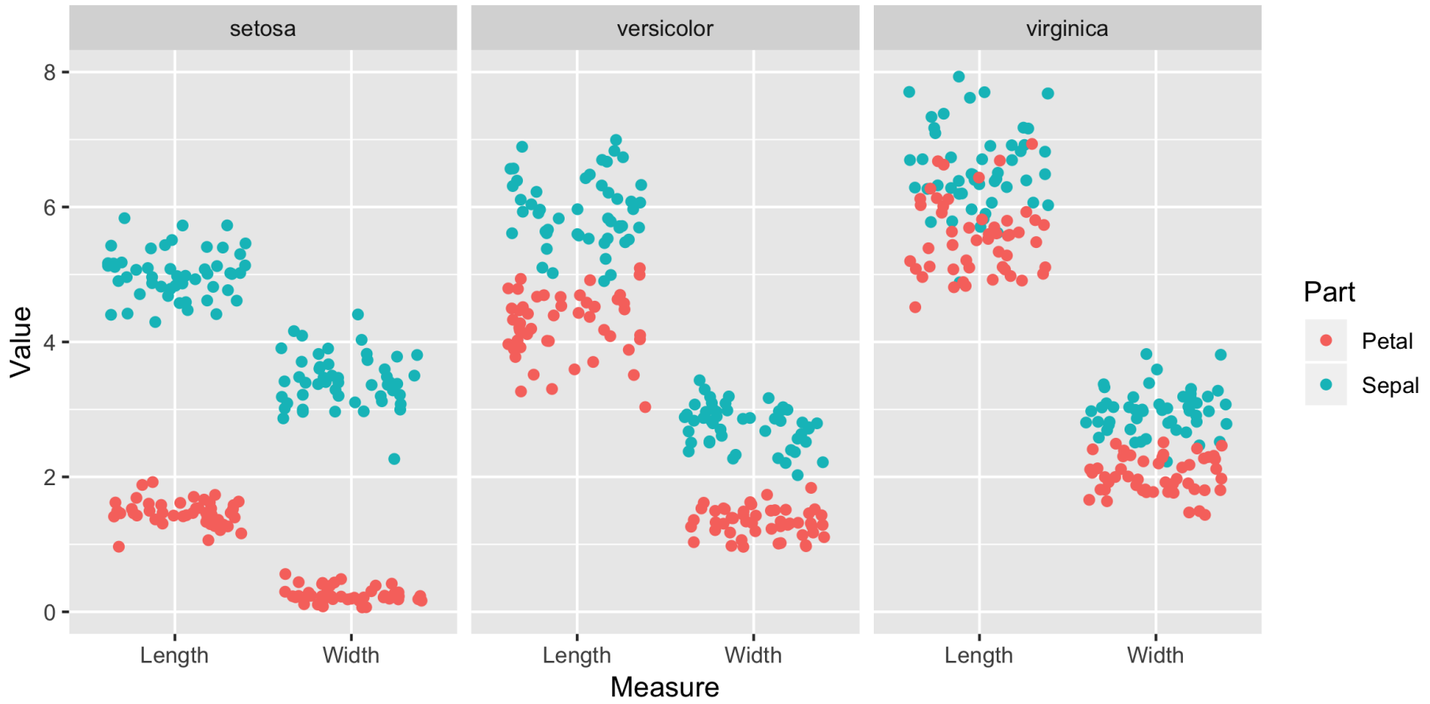






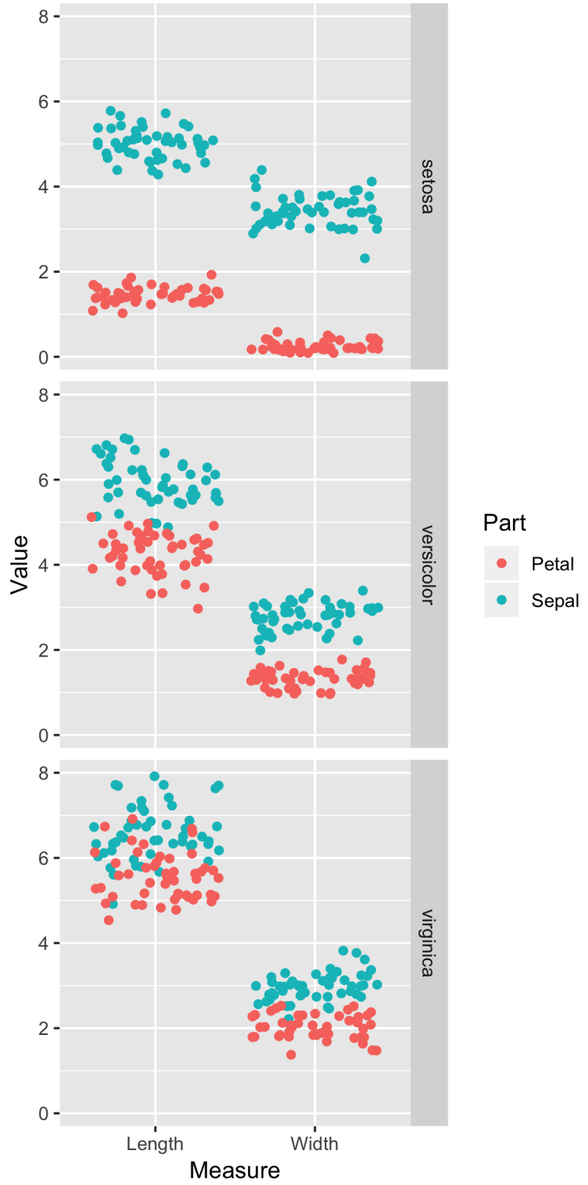
iris.tidy

```
ggplot(iris.tidy, aes(x = Measure, y = Value, color = Part)) +
  geom_jitter() +
  facet_grid(cols = vars(Species))
```



`iris.tidy` faceting done wrong:

```
ggplot(iris.tidy, aes(x = Measure,
                      y = Value,
                      color = Part)) +
  geom_jitter() +
  facet_grid(rows = vars(Species))
```



Other options

- Split according to rows and columns

Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Facet labels and order

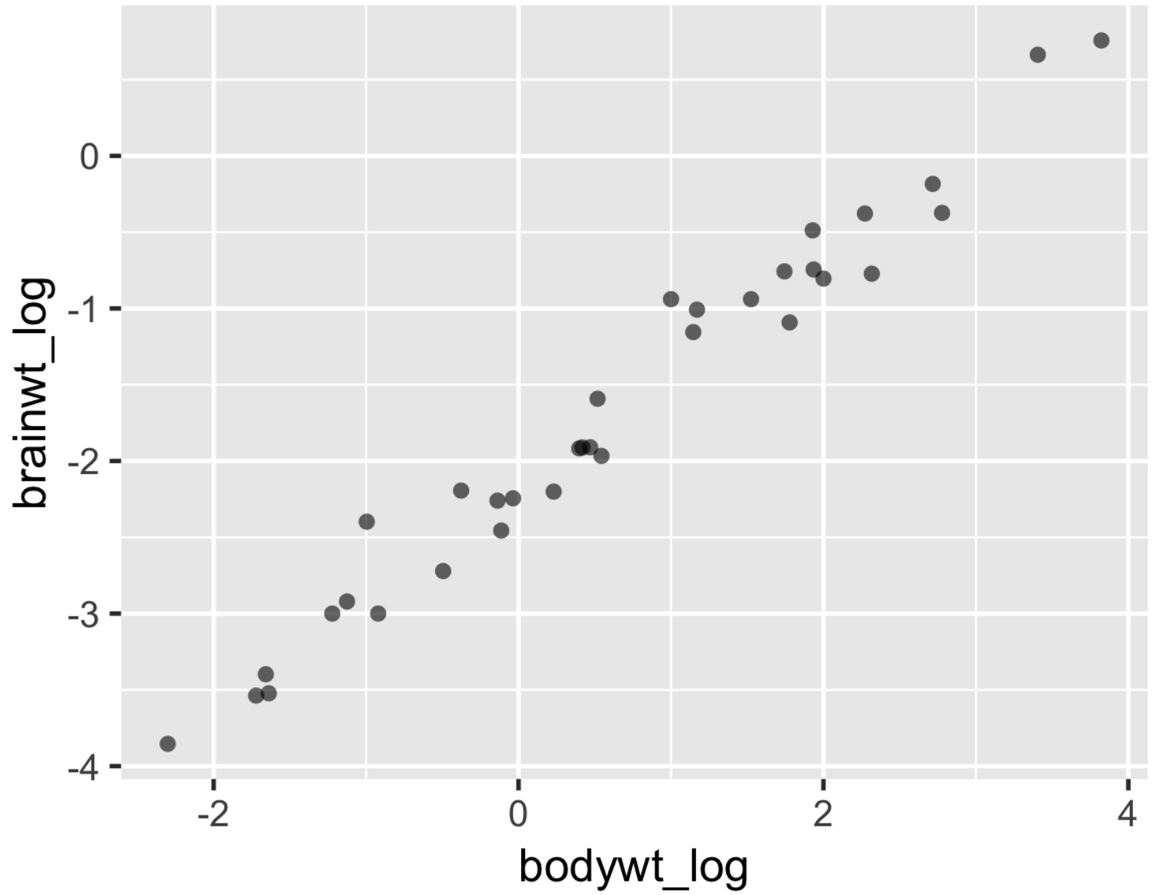
INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

A new dataframe

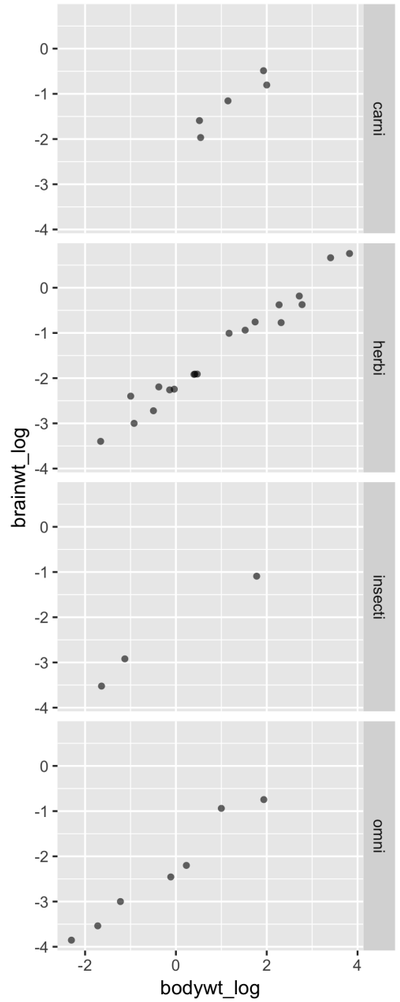
```
# Plot
p <- ggplot(msleep2, aes(bodywt_log,
                         brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  coord_fixed()

p
```



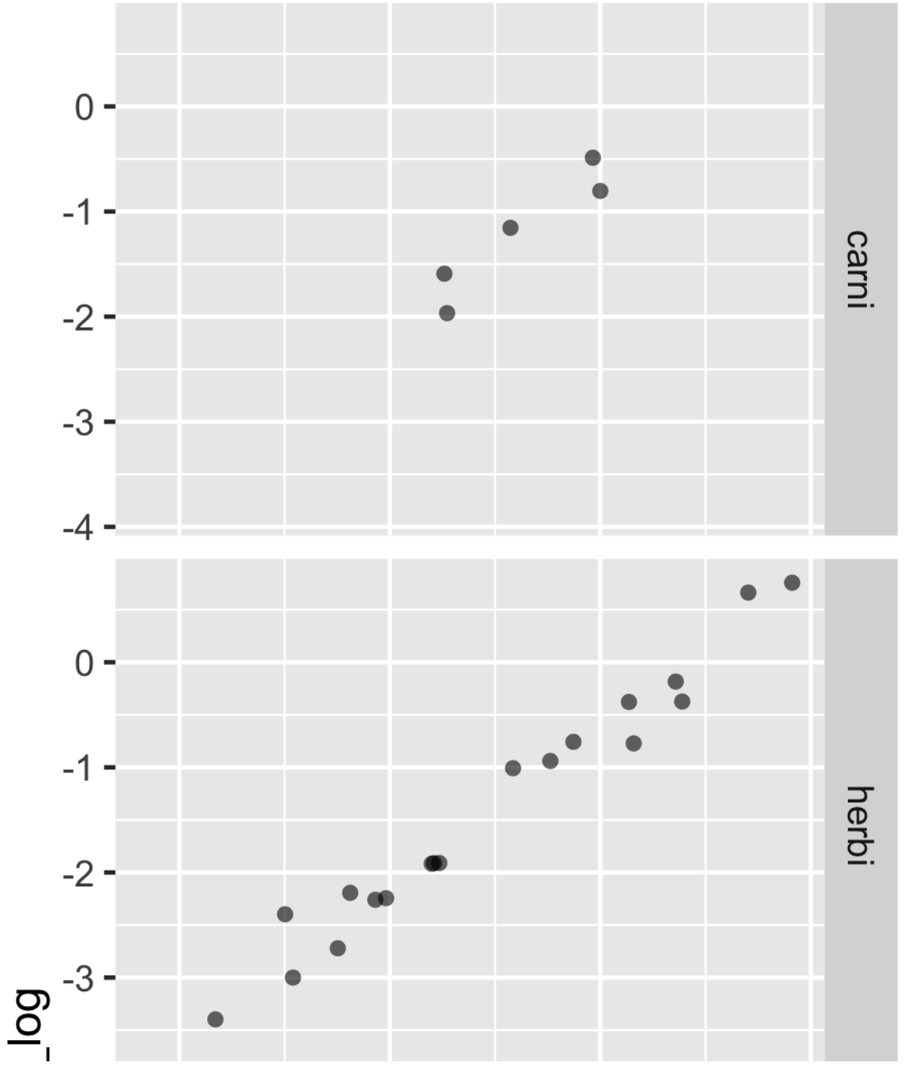
A new dataframe, with facets

```
p +
  facet_grid(rows = vars(vore))
```



A new dataframe, with facets

```
p +
  facet_grid(rows = vars(vore))
```

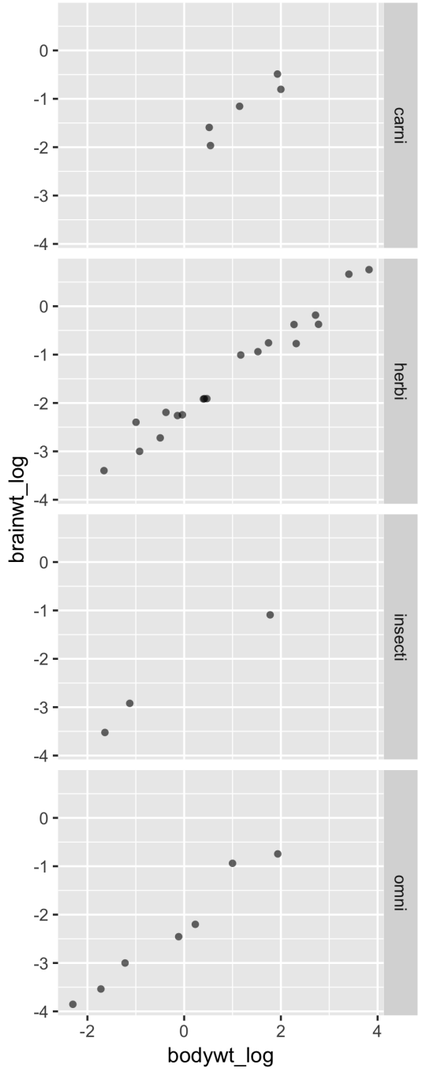


Poor labels and order

```
p +
  facet_grid(rows = vars(vore))
```

Two typical problems with facets:

- Poorly labeled (e.g. non descriptive)
- Wrong or inappropriate order

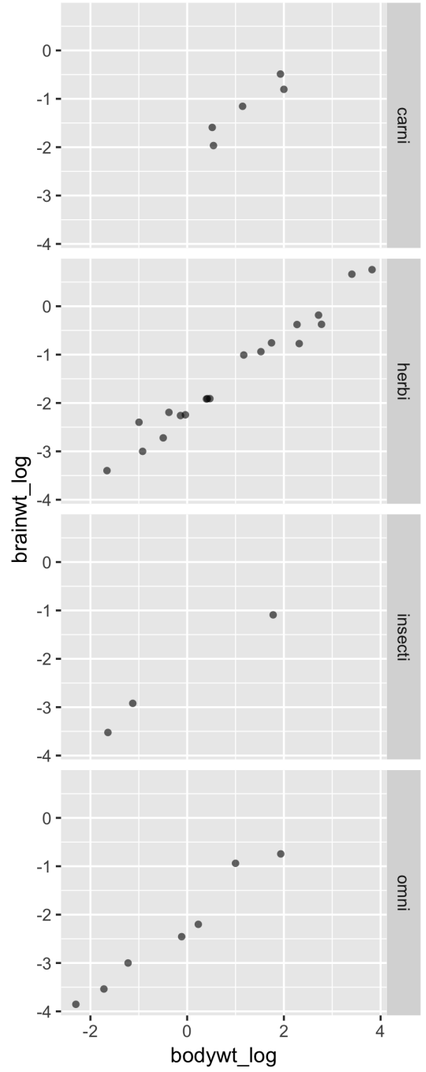


Poor labels and order

```
p +
  facet_grid(rows = vars(vore))
```

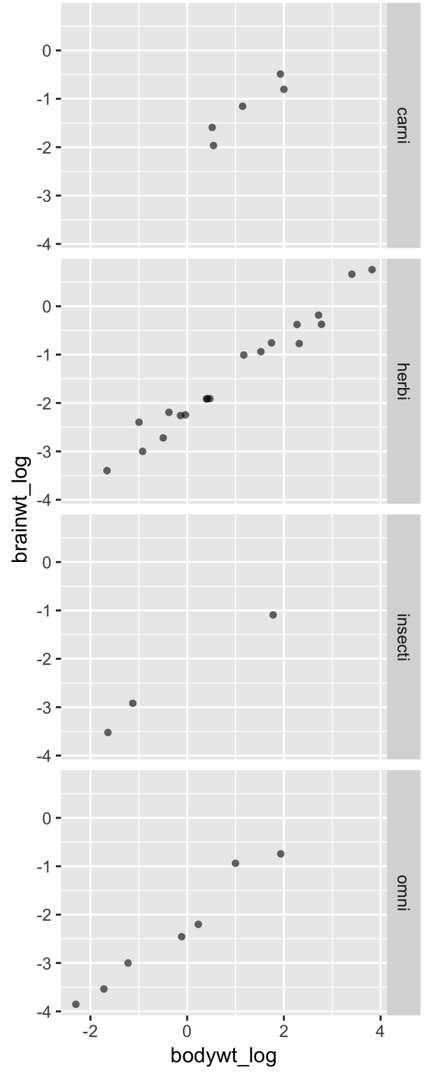
Solutions:

- Easy: Add labels in ggplot
- Better: Relabel and rearrange factor variables in your dataframe



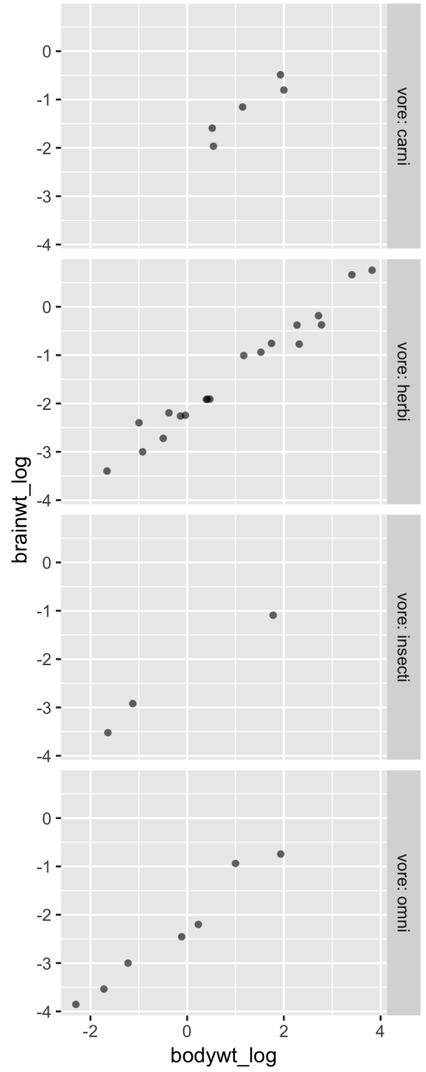
The labeller argument

```
# Default is to label the value
p +
  facet_grid(rows = vars(vore),
             labeller = label_value)
```



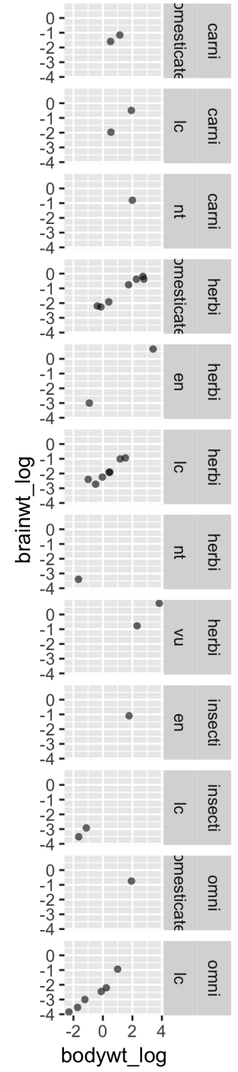
Using label_both adds the variable name

```
# Print variable name also
p +
  facet_grid(rows = vars(vore),
                 labeller = label_both)
```



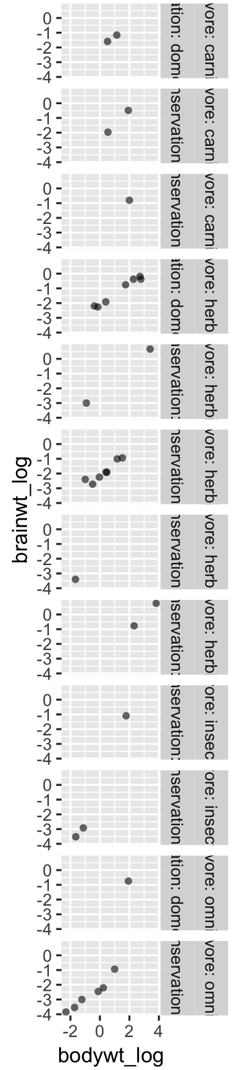
Two variables on one side

```
p +
  facet_grid(rows = vars(vore,
                         conservation))
```



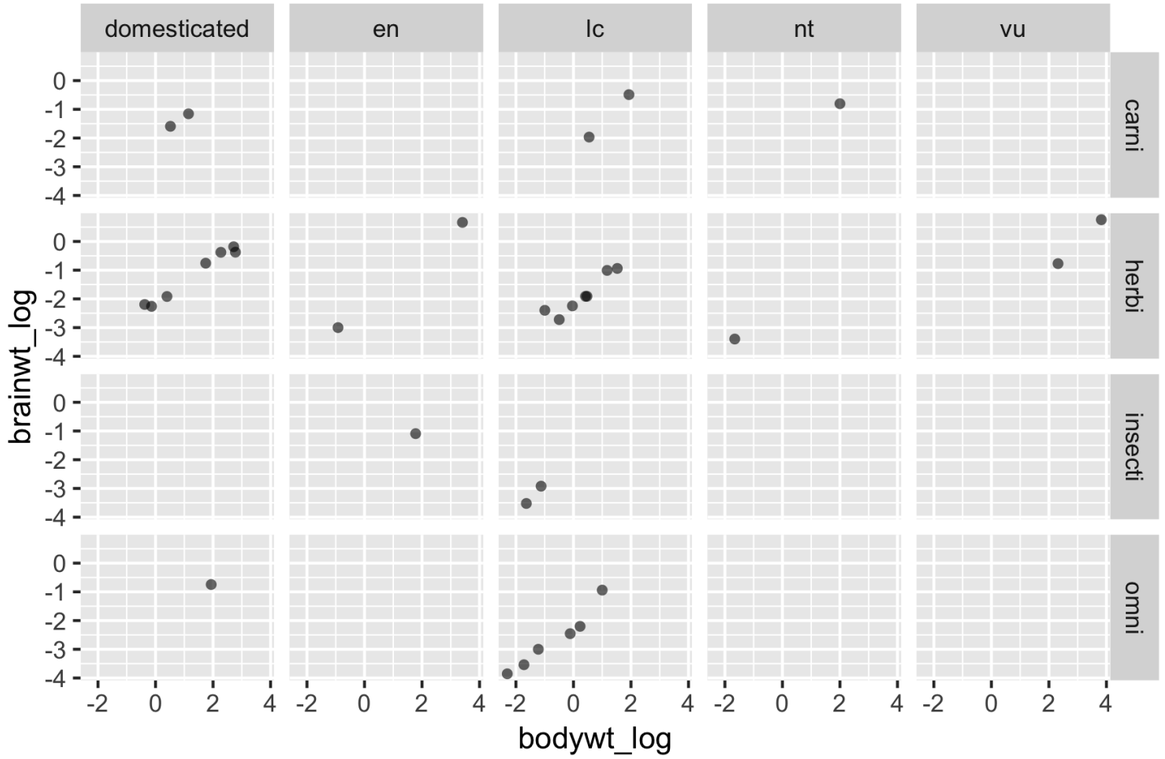
Using label_context avoids ambiguity

```
p +
  facet_grid(rows = vars(vore,
                         conservation),
             labeller = label_context)
```



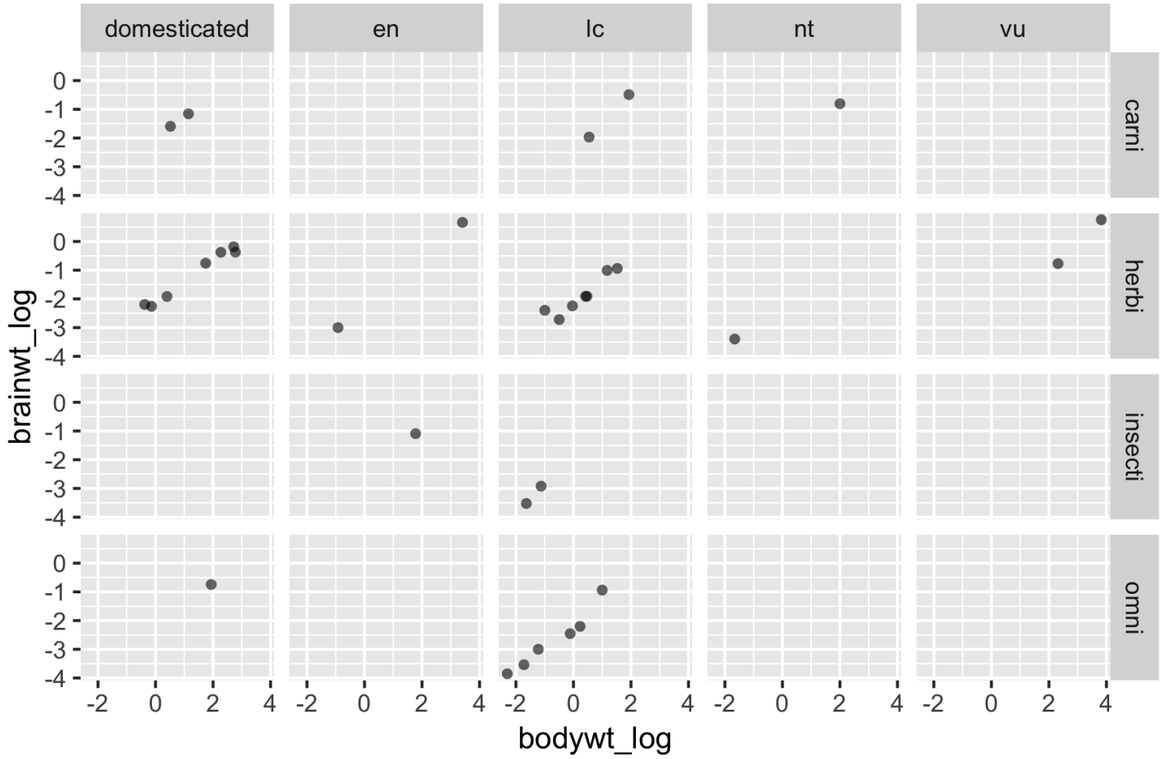
Use rows and columns when appropriate

```
p +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             labeller = label_context)
```

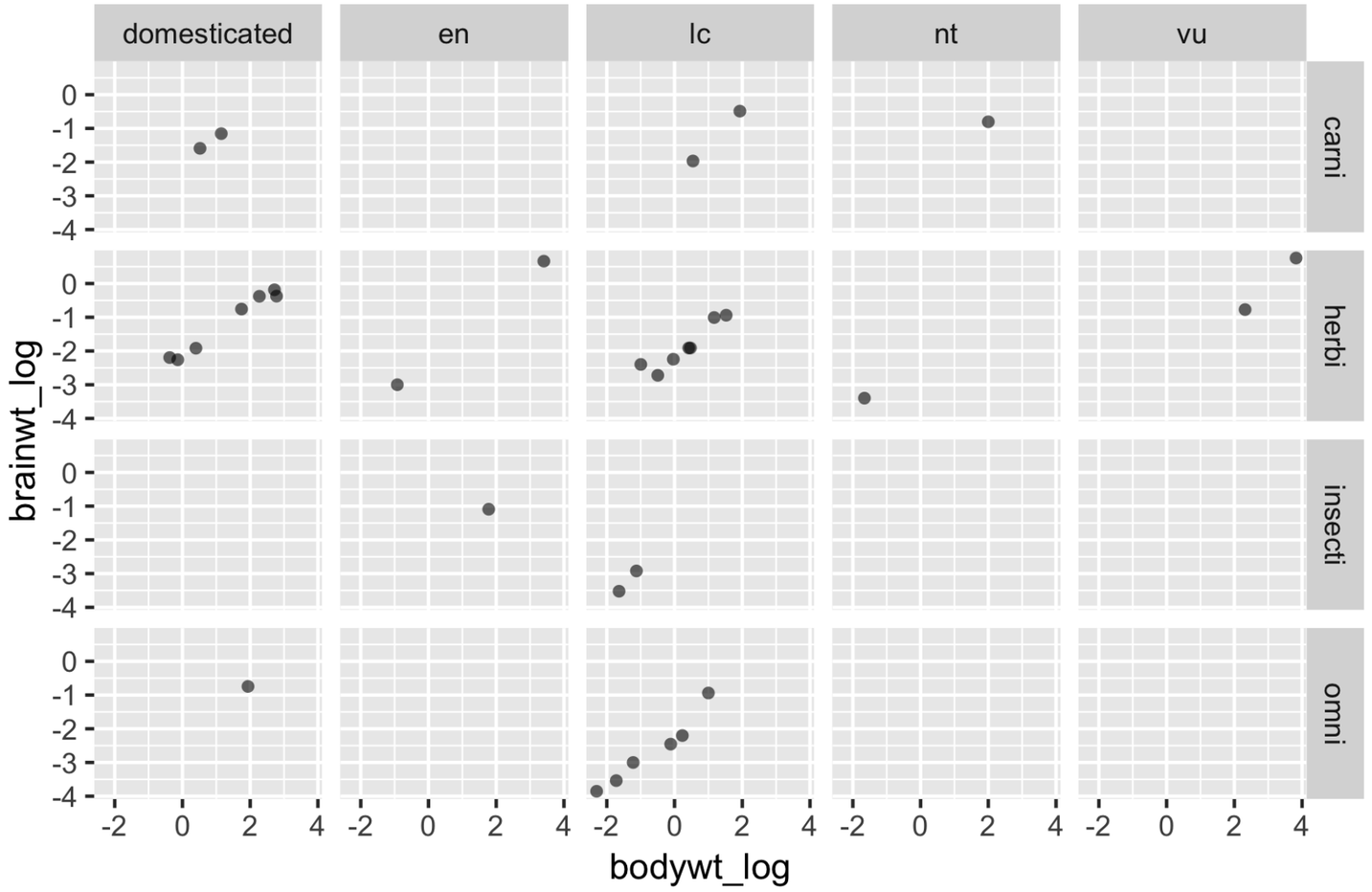


Use rows and columns when appropriate

```
p +
  facet_grid(rows = vars(vore),
             cols = vars(conservation))
```



Use rows and columns when appropriate



Relabeling and reordering factors

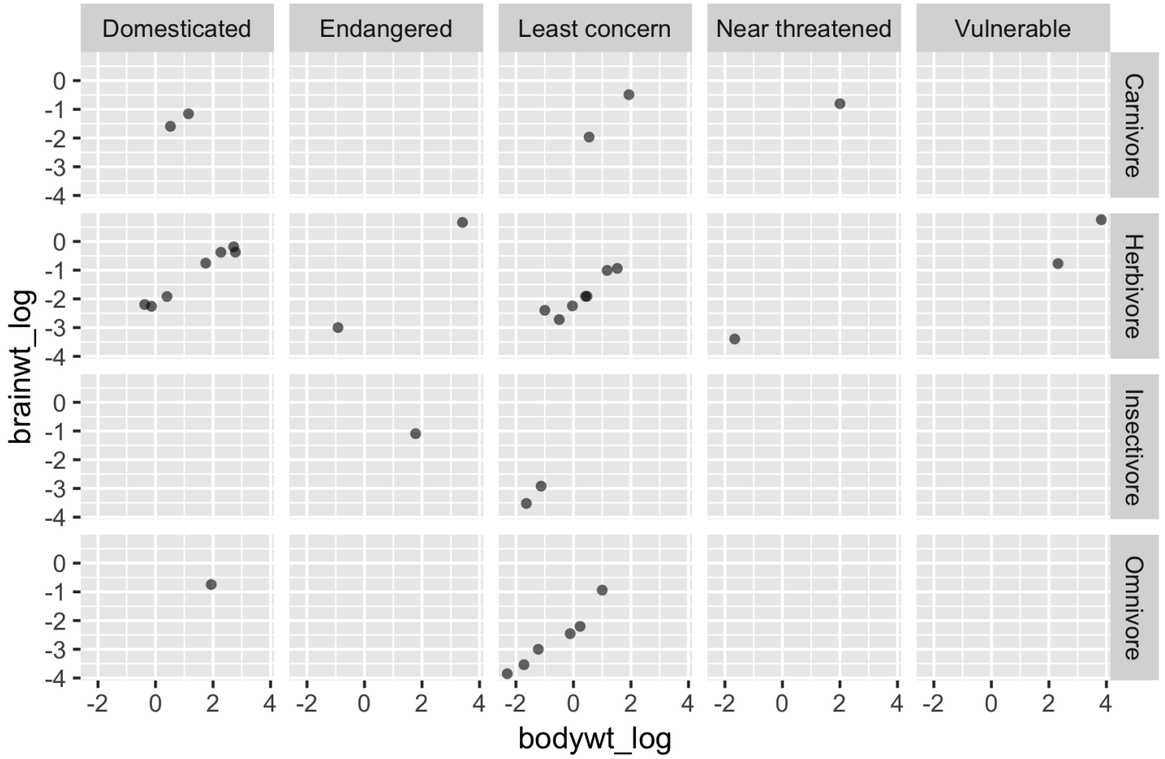
```
msleep2$conservation <- fct_recode(msleep2$conservation,
                                   Domesticated = "domesticated",
                                   `Least concern` = "lc",
                                   `Near threatened` = "nt",
                                   Vulnerable = "vu",
                                   Endangered = "en")

msleep2$vore = fct_recode(msleep2$vore,
                          Carnivore = "carni",
                          Herbivore = "herbi",
                          Insectivore = "insecti",
                          Omnivore = "omni")
```

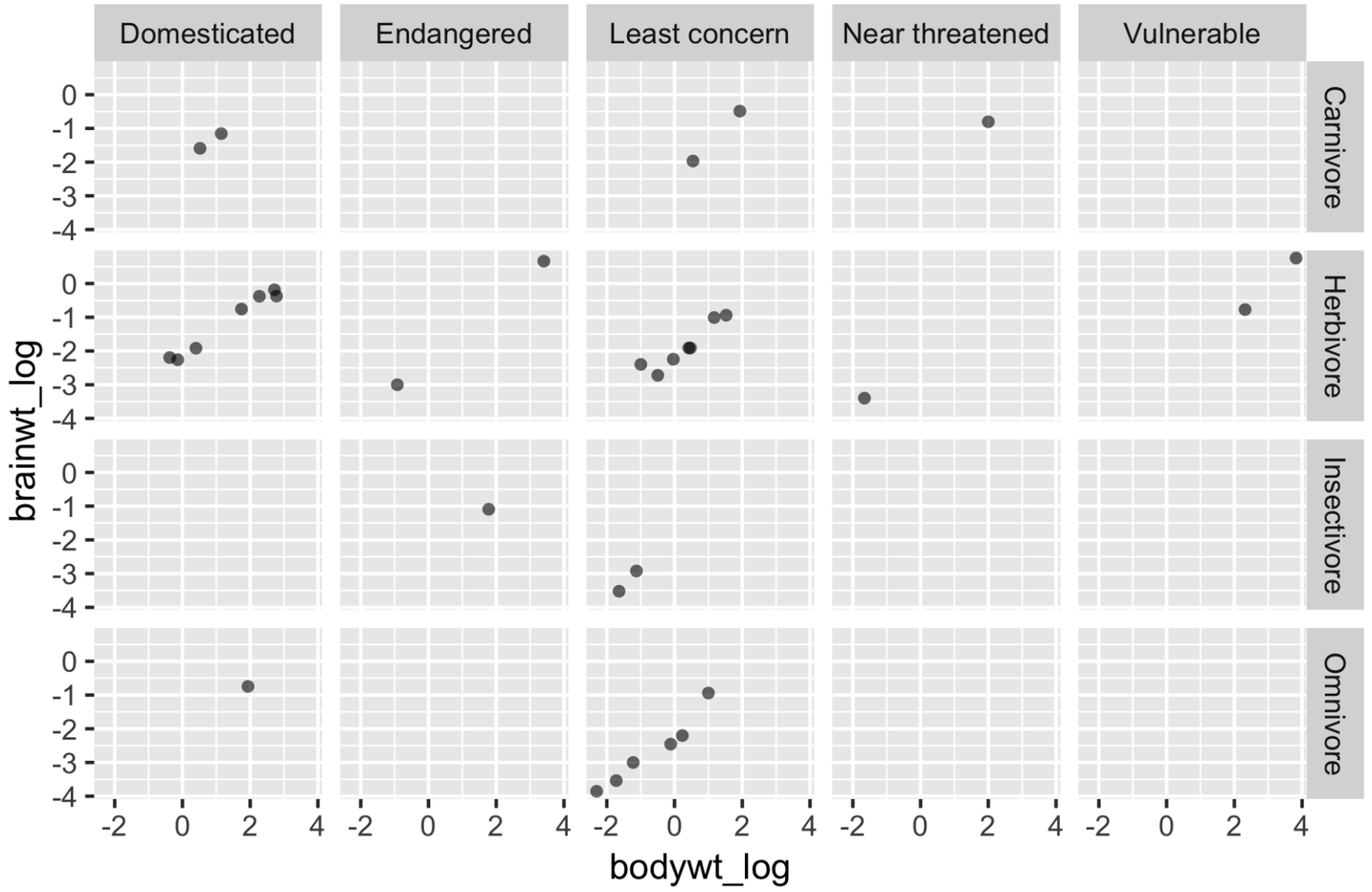
Reinitialize plot with new labels

```
# Plot
p <- ggplot(msleep2, aes(bodywt_log,
                         brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  coord_fixed()

p +
  facet_grid(rows = vars(vore),
             cols = vars(conservation))
```



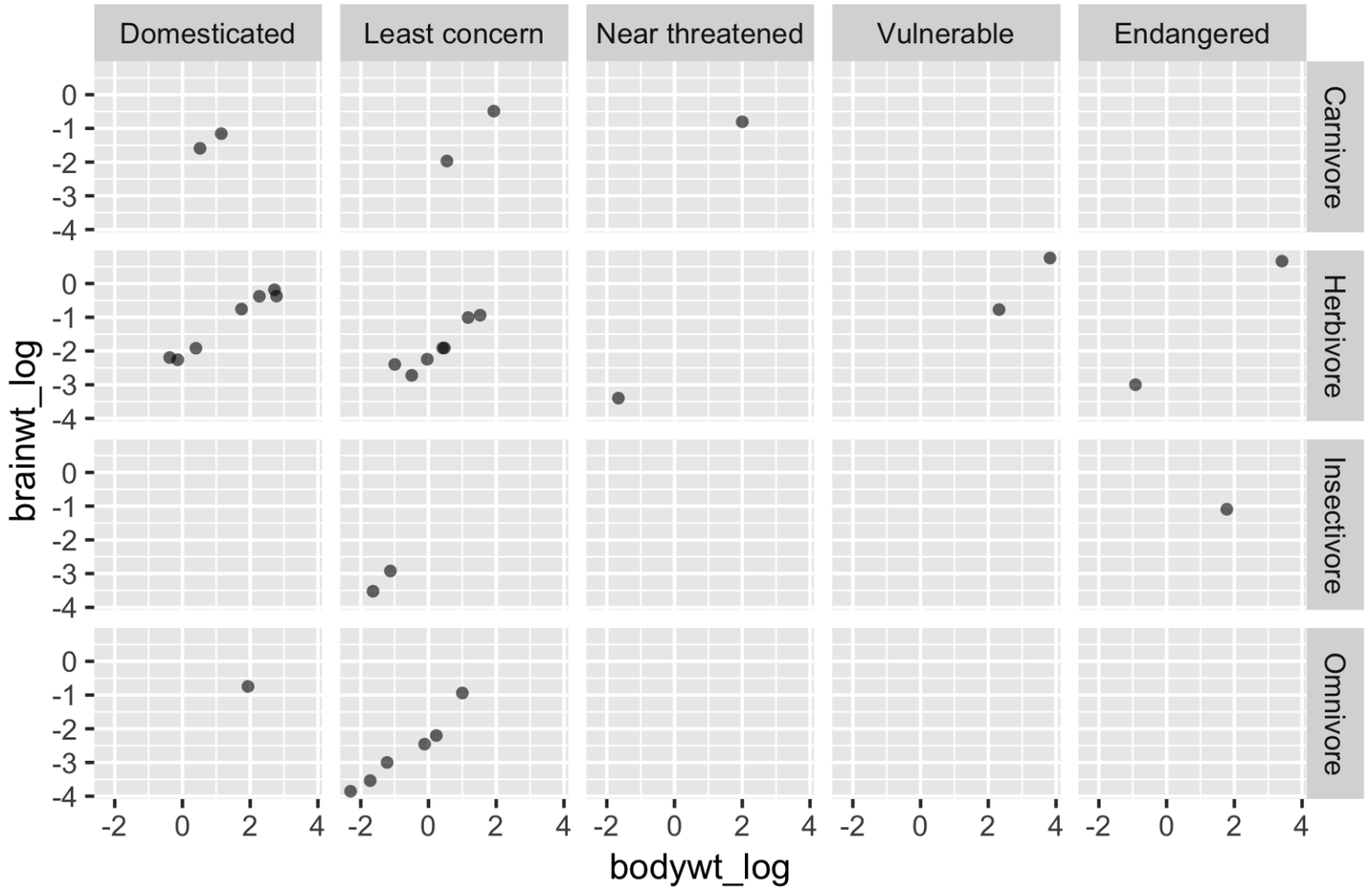
Reinitialize plot with new labels



Changing the order of levels

```
# Change order of levels:
msleep2$conservation = fct_relevel(msleep2$conservation,
                                   c("Domesticated",
                                     "Least concern",
                                     "Near threatened",
                                     "Vulnerable",
                                     "Endangered"))
```

Reinitialize plot with new order



Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Facet plotting spaces

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

datacamp

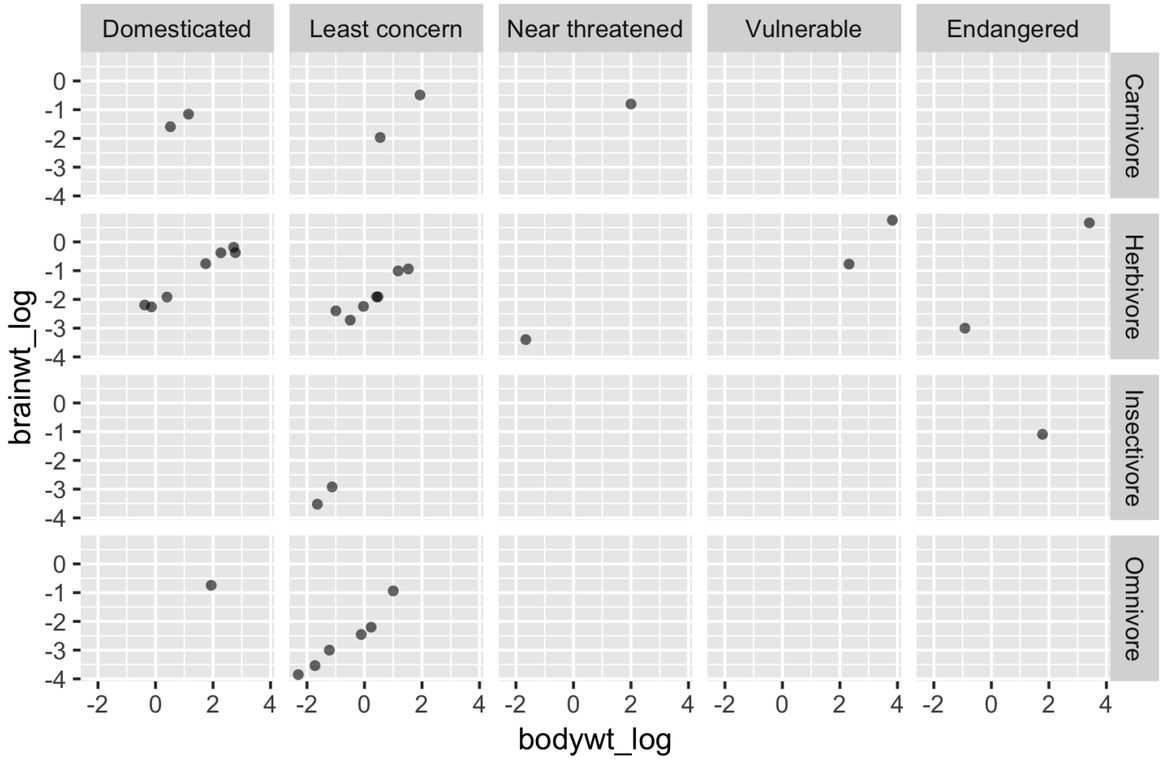
Facets and variable plotting spaces

Reasons to not use consistent plotting spaces:

Variable type	Subsets contains
Continuous	Wildly different ranges
Categorical	Different groups

Adjusting the plotting space...

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  coord_fixed() +
  facet_grid(rows = vars(vore),
             cols = vars(conservation))
```



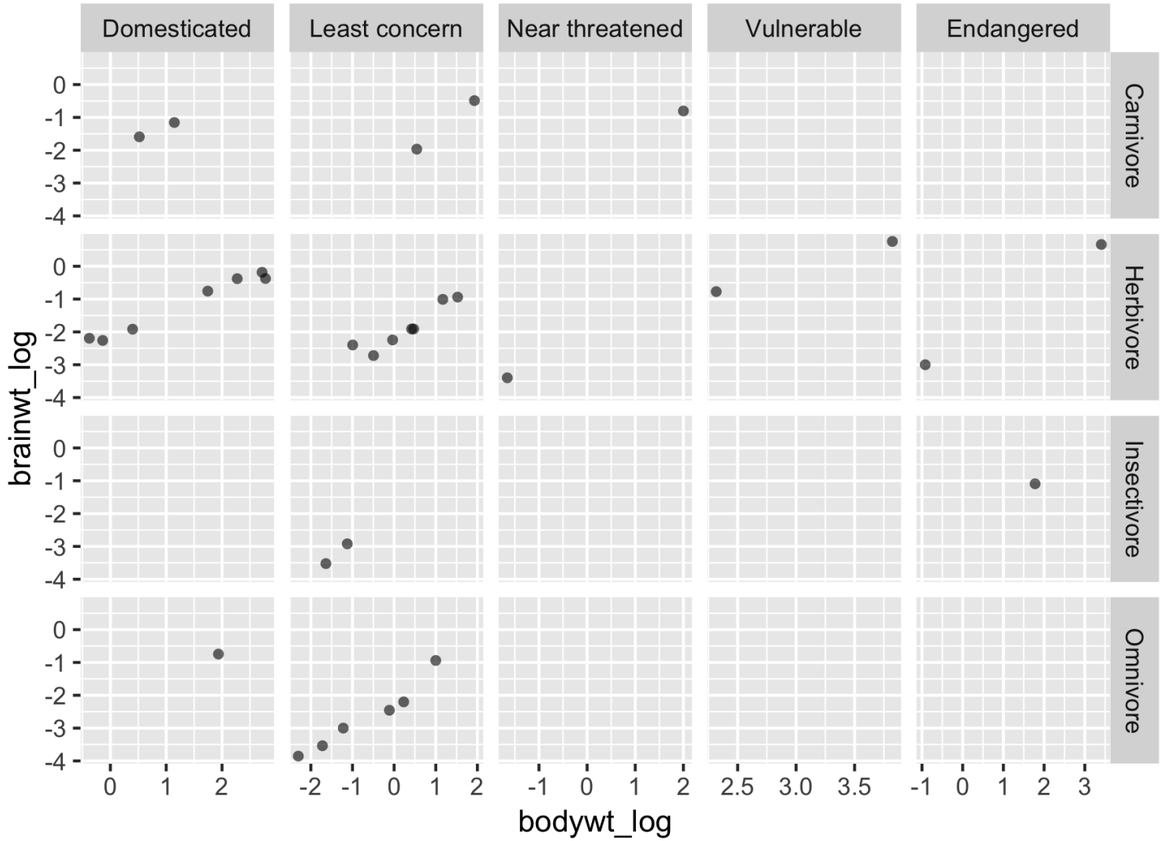
... but not with fixed scales

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  coord_fixed() +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free_x")
```

```
Error: coord_fixed doesn't support free scales
```

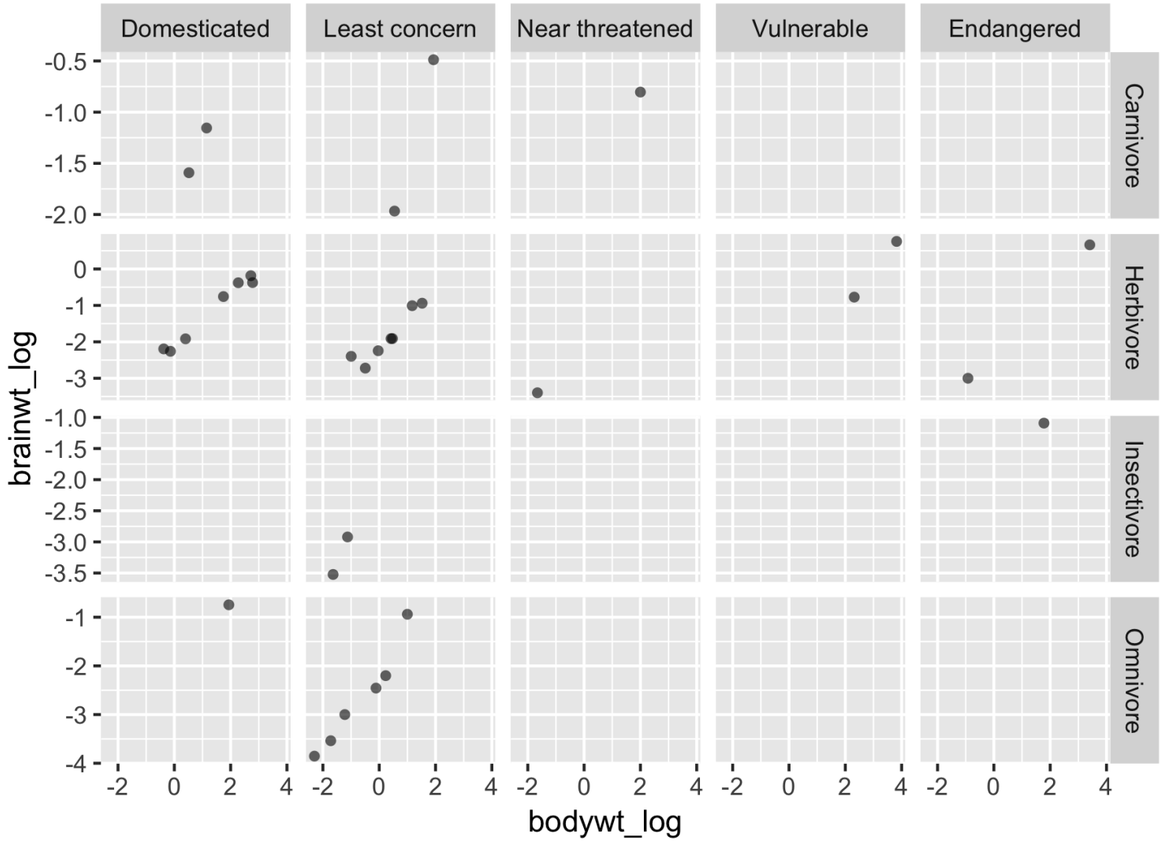
Adjusting the plotting space

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free_x")
```



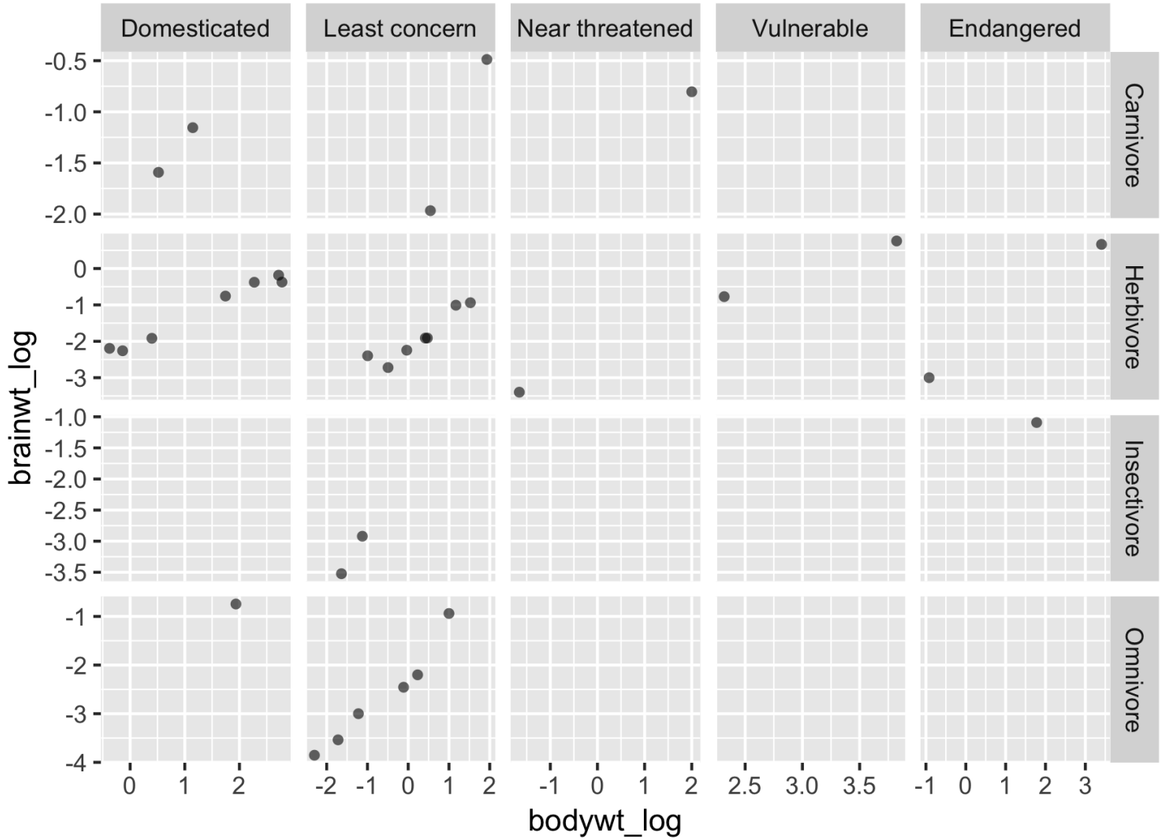
Adjusting the plotting space

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free_y")
```



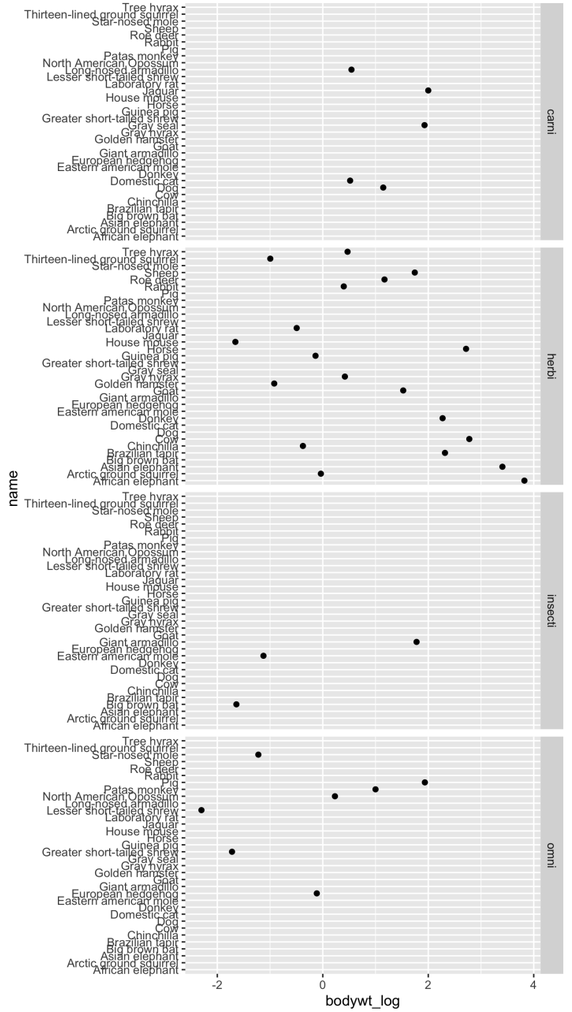
Adjusting the plotting space

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free")
```



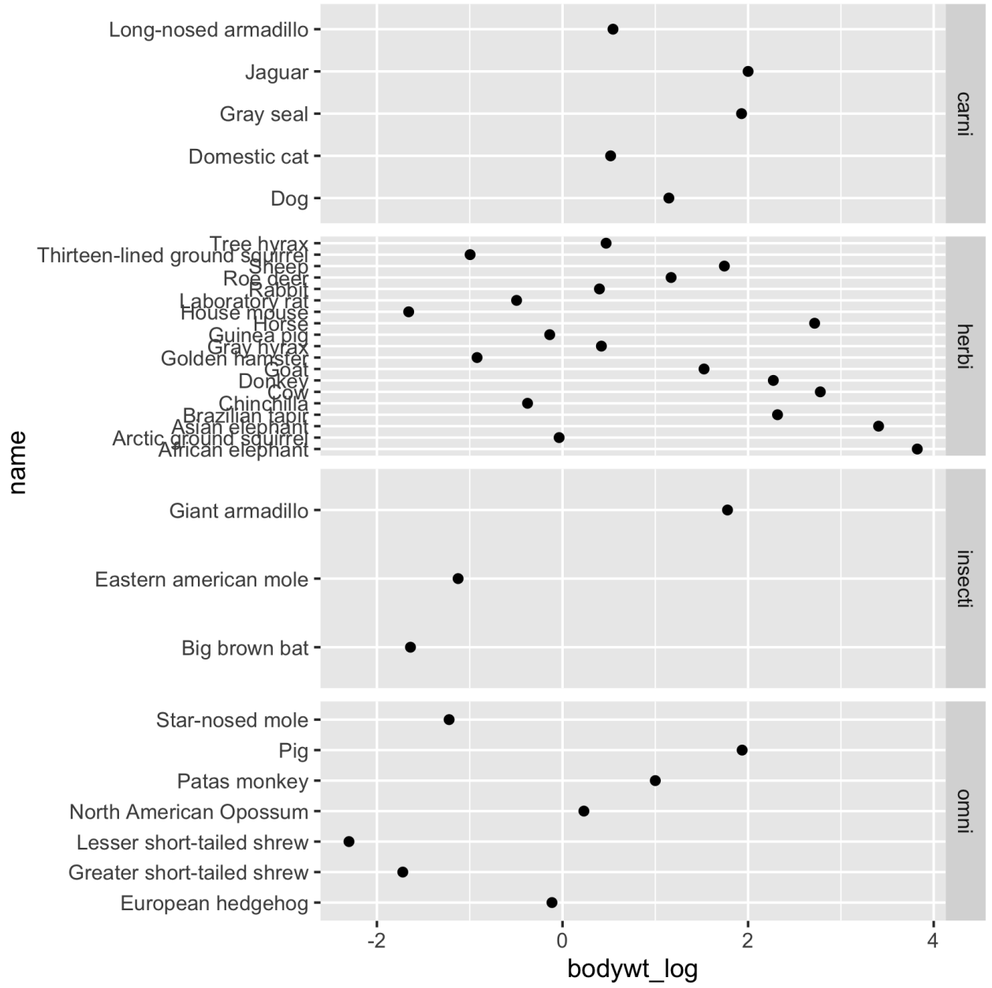
Adjusting the plotting space

```
ggplot(msleep2, aes(x = bodywt_log,
                    y = name)) +
  geom_point() +
  facet_grid(rows = vars(vore))
```



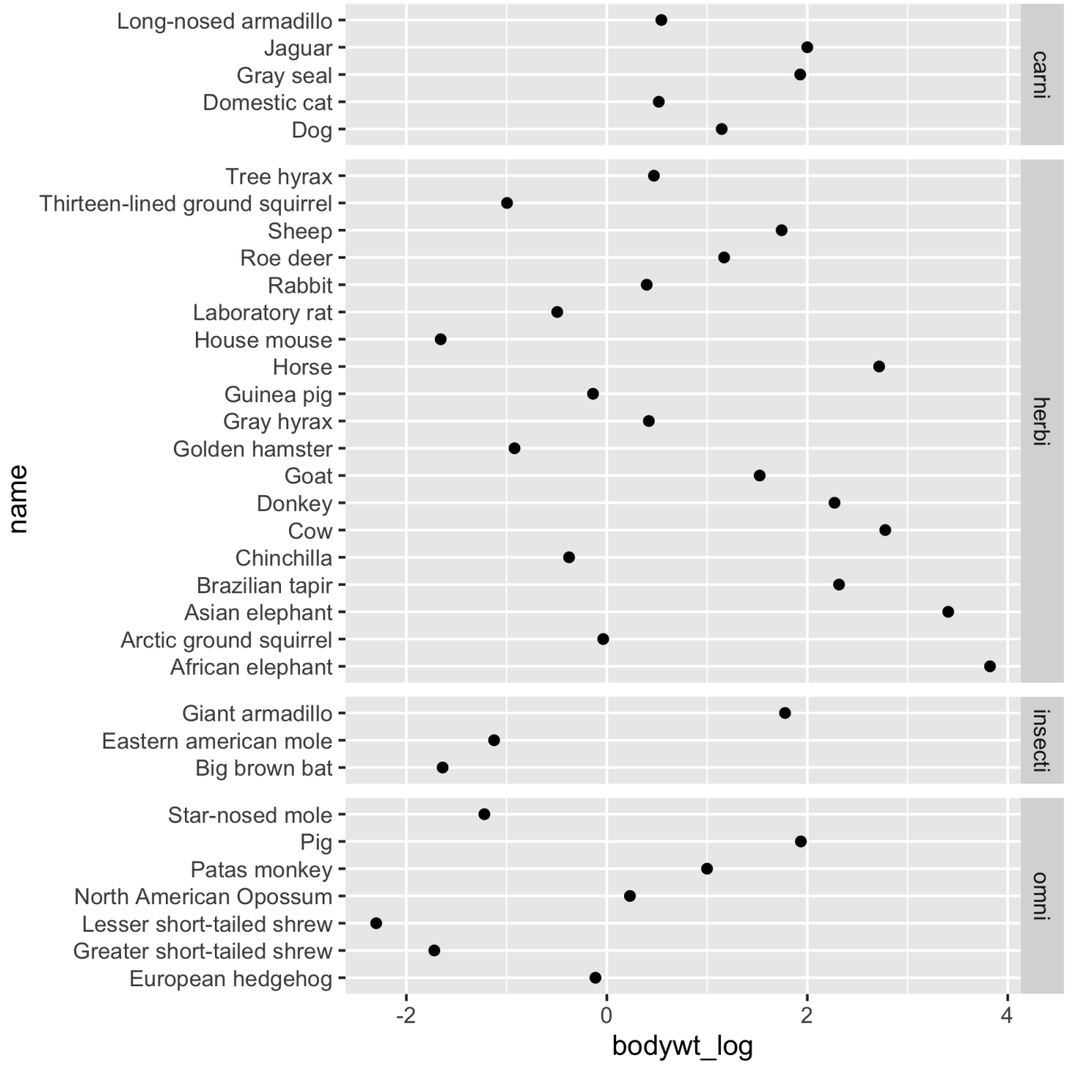
Adjusting the plotting space

```
ggplot(msleep2, aes(x = bodywt_log,
                    y = name)) +
  geom_point() +
  # Free the y scales and space
  facet_grid(rows = vars(vore),
             scales = "free_y")
```



Adjusting the plotting space

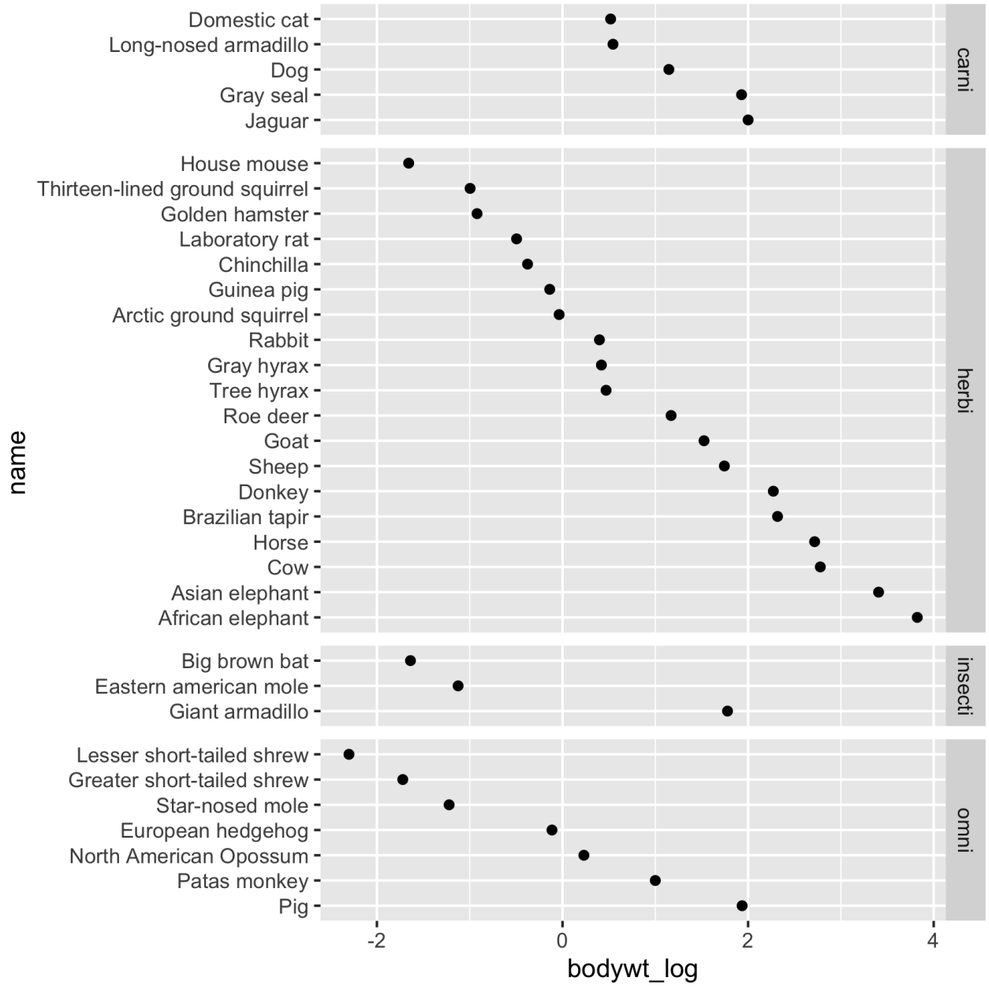
```
ggplot(msleep2, aes(x = bodywt_log,
                    y = name)) +
  geom_point() +
  # Free the y scales and space
  facet_grid(rows = vars(vore),
             scales = "free_y",
             space = "free_y")
```



Final adjustments

```
msleep2 <- msleep2 %>%
  # Arrange from lo to hi weight
  arrange(-bodywt_log) %>%
  # Redefine factor levels in order
  mutate(name = as_factor(name))

# New order is reflected in y axis
ggplot(msleep2, aes(x = bodywt_log,
                    y = name)) +
  geom_point() +
  # Free the y scales and space
  facet_grid(rows = vars(vore),
             scales = "free_y",
             space = "free_y")
```



Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Facet wrap & margins

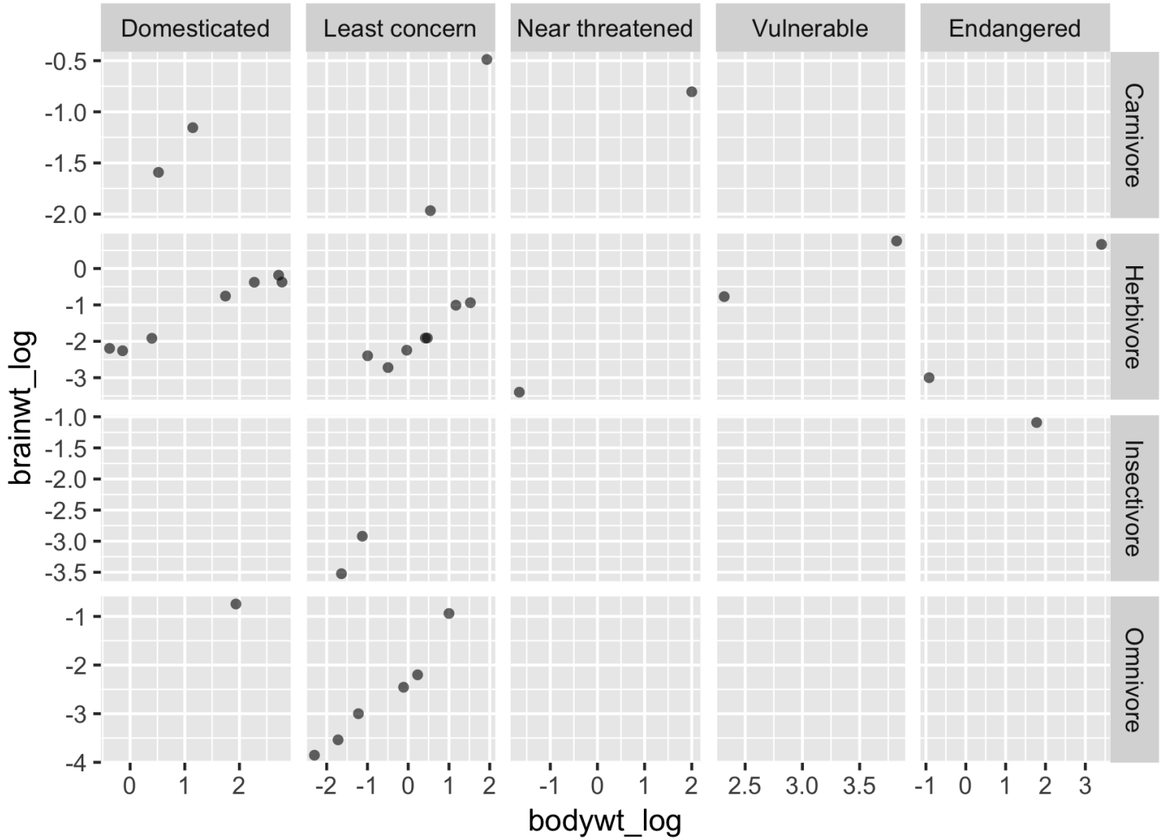
INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2

Rick Scavetta
Founder, Scavetta Academy

datacamp

Adjusting the plotting space

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free")
```



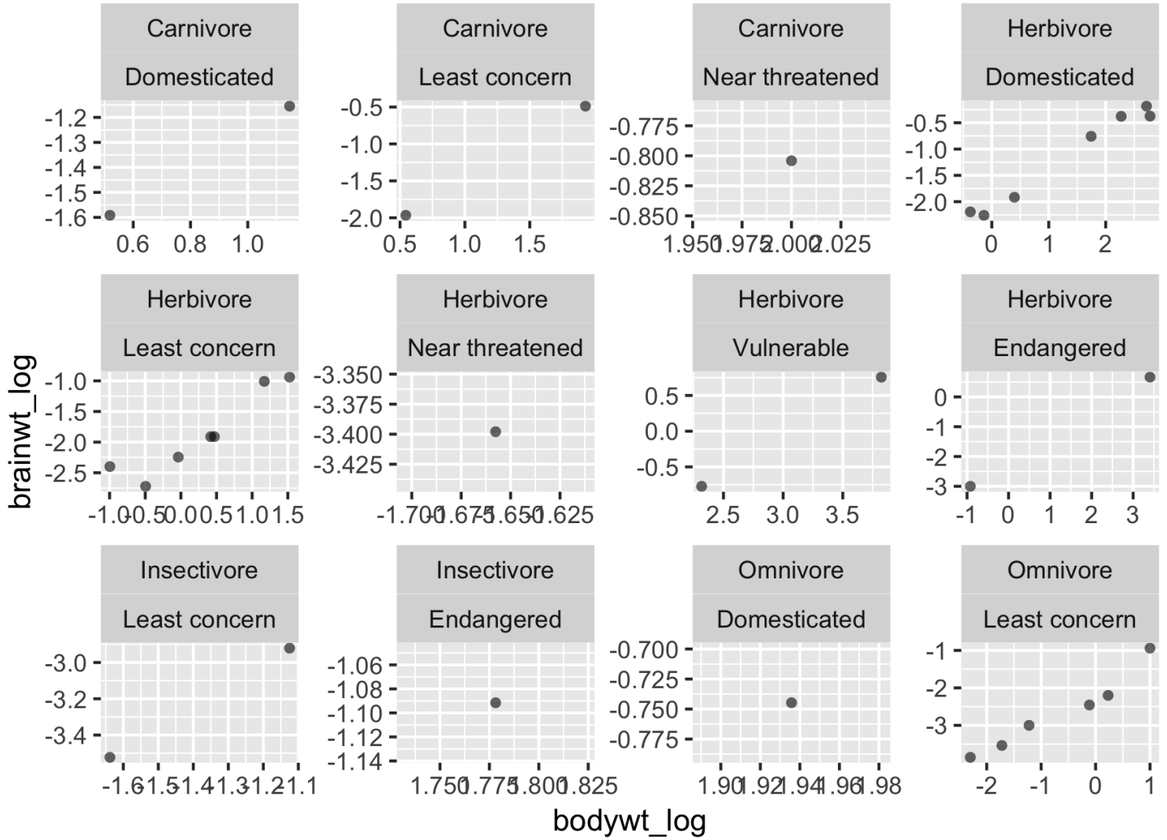
Using facet_wrap()

Use cases:

1. When you want both x and y axes to be free on every individual plot
   - i.e. Not just per row or column as per `facet_grid()`

Using facet_wrap() - Scenario 1

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_wrap(vars(vore, conservation),
             scales = "free")
```



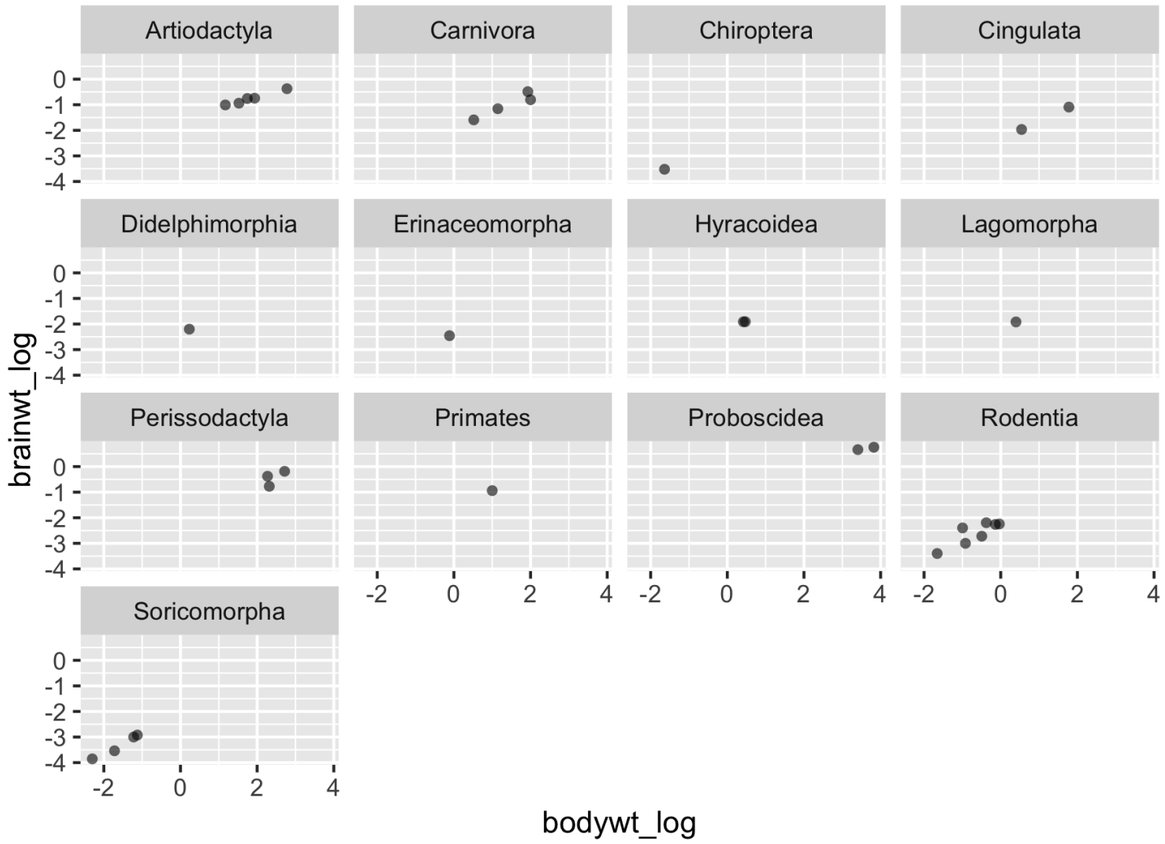
Using facet_wrap()

Use cases:

1. When you want both x and y axes to be free on every individual plot
   - i.e. Not just per row or column as per `facet_grid()`
2. When your categorical (factor) variable has many groups (levels)
   - i.e. too many sub plots for column or row-wise faceting
   - A more typical scenario

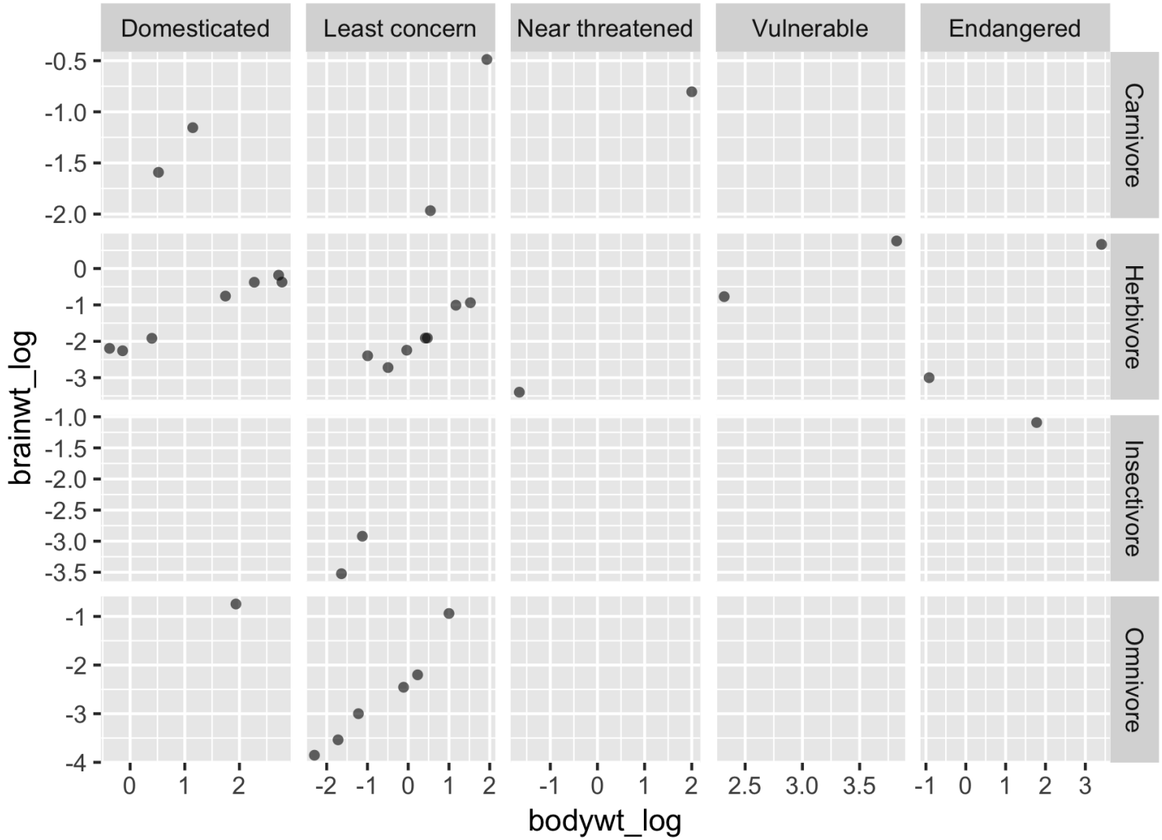
Using facet_wrap() - Scenario 2

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_wrap(vars(order))
```



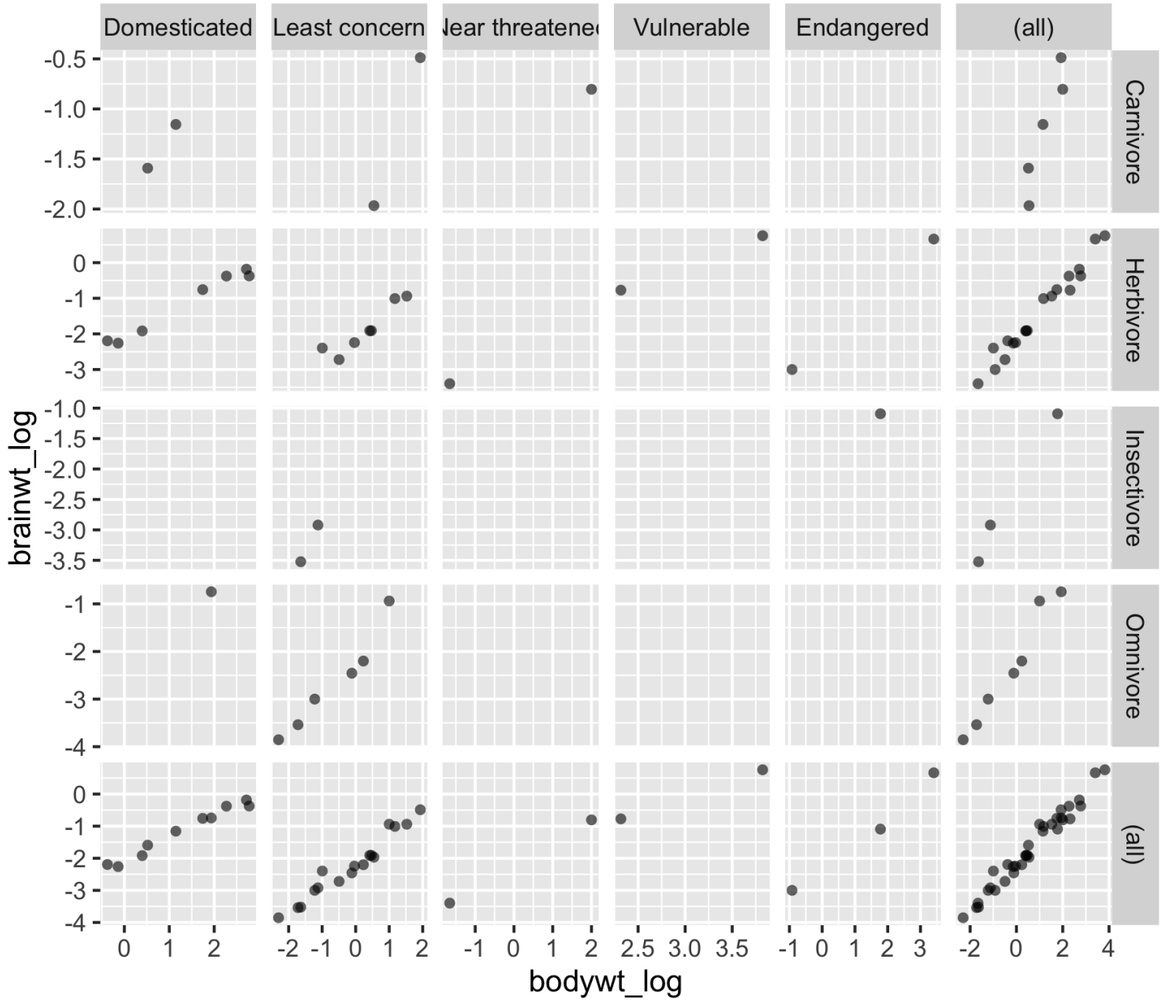
Using margin plots

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free")
```



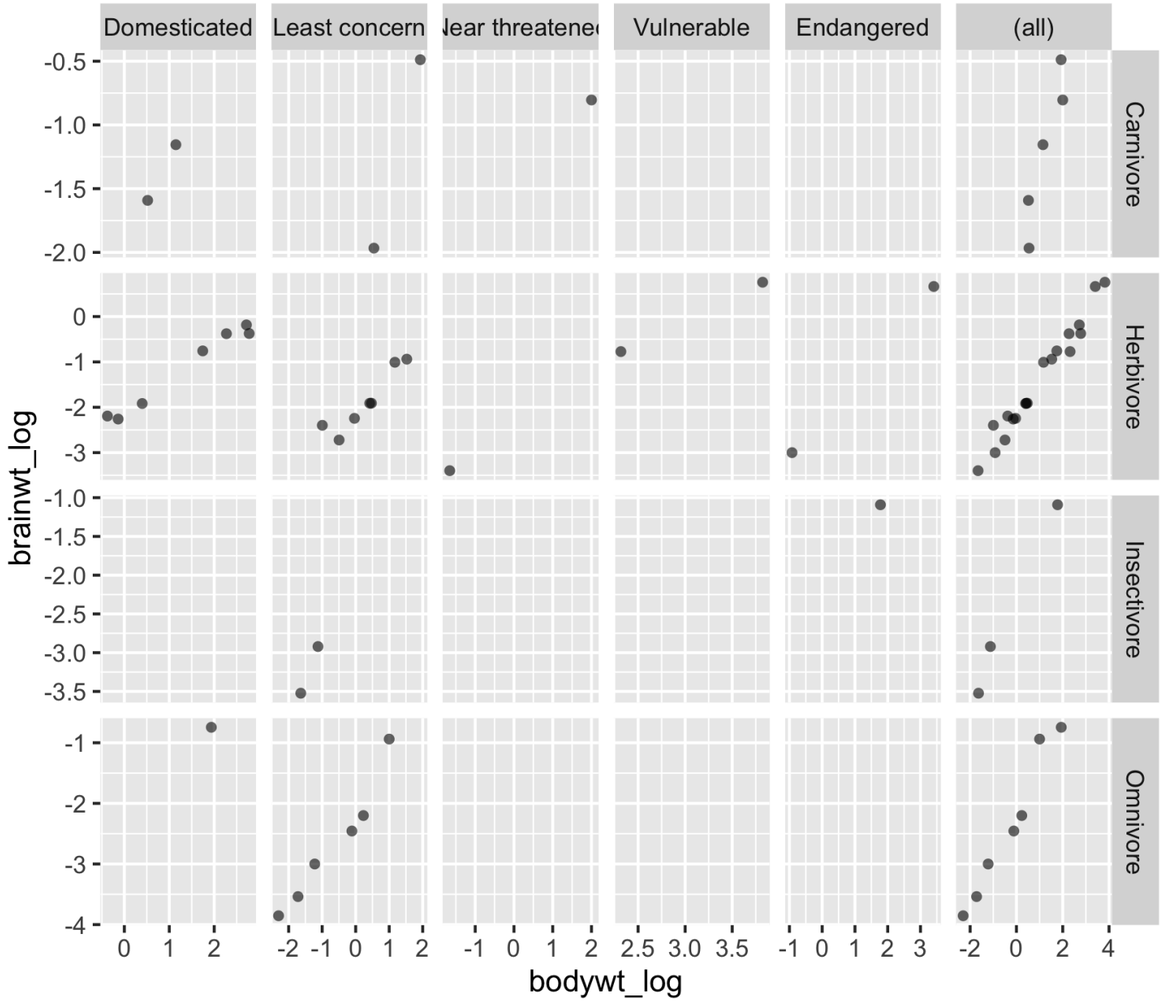
Using margin plots

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free",
             margins = TRUE)
```



Using margin plots

```
ggplot(msleep2, aes(bodywt_log,
                    brainwt_log)) +
  geom_point(alpha = 0.6, shape = 16) +
  facet_grid(rows = vars(vore),
             cols = vars(conservation),
             scales = "free",
             margins = "conservation")
```



Let's practice!

INTERMEDIATE DATA VISUALIZATION WITH GGPLOT2