Two numeric explanatory variables

INTERMEDIATE REGRESSION IN R

Richie Cotton
Learning Solutions Architect at DataCamp

Visualizing 3 numeric variables

- 3D scatter plot
- 2D scatter plot with response as color

Another column for the fish dataset

species	mass_g	length_cm	height_cm
Bream	1000	33.5	18.96
Bream	925	36.2	18.75
Roach	290	24.0	8.88
Roach	390	29.5	9.48
Perch	1100	39.0	12.80
Perch	1000	40.2	12.60
Pike	1250	52.0	10.69
Pike	1650	59.0	10.81

3D scatter plot

```
library(plot3D)

scatter3D(fish$length_cm, fish$height_cm, fish$mass_g)
```

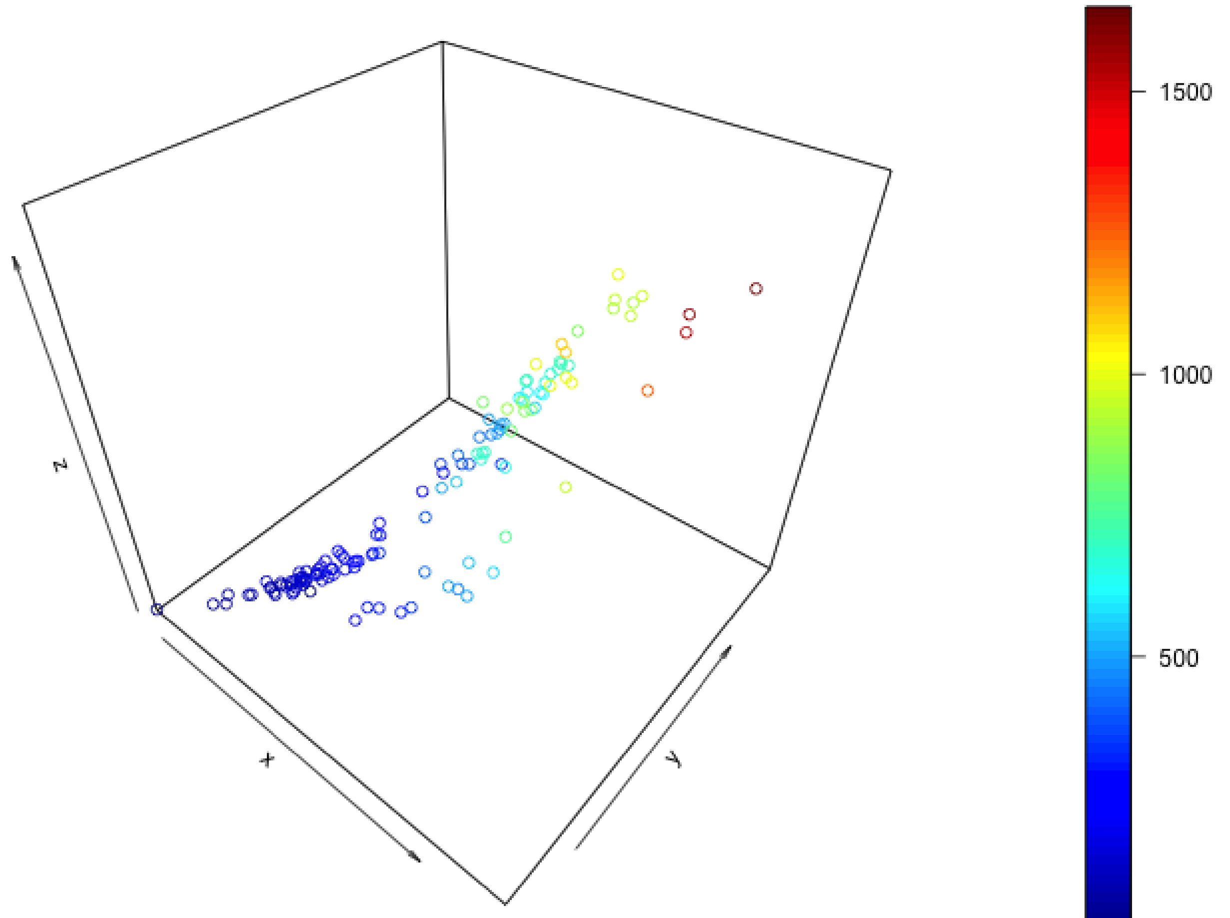
```
library(plot3D)
library(magrittr)

fish %$%
  scatter3D(length_cm, height_cm, mass_g)
```

3D scatter plot

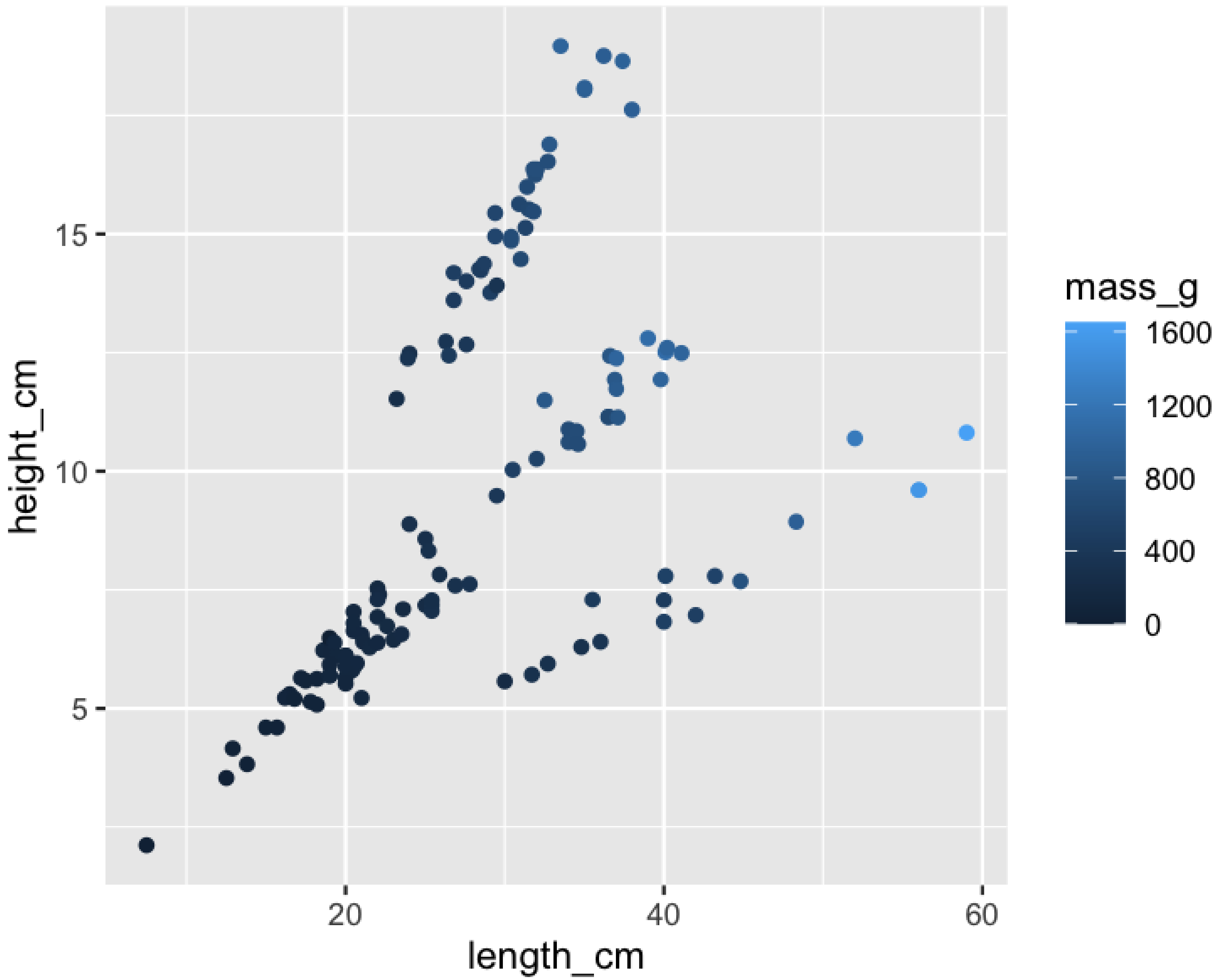
```
library(plot3D)
library(magrittr)

fish %$%
  scatter3D(length_cm, height_cm, mass_g)
```



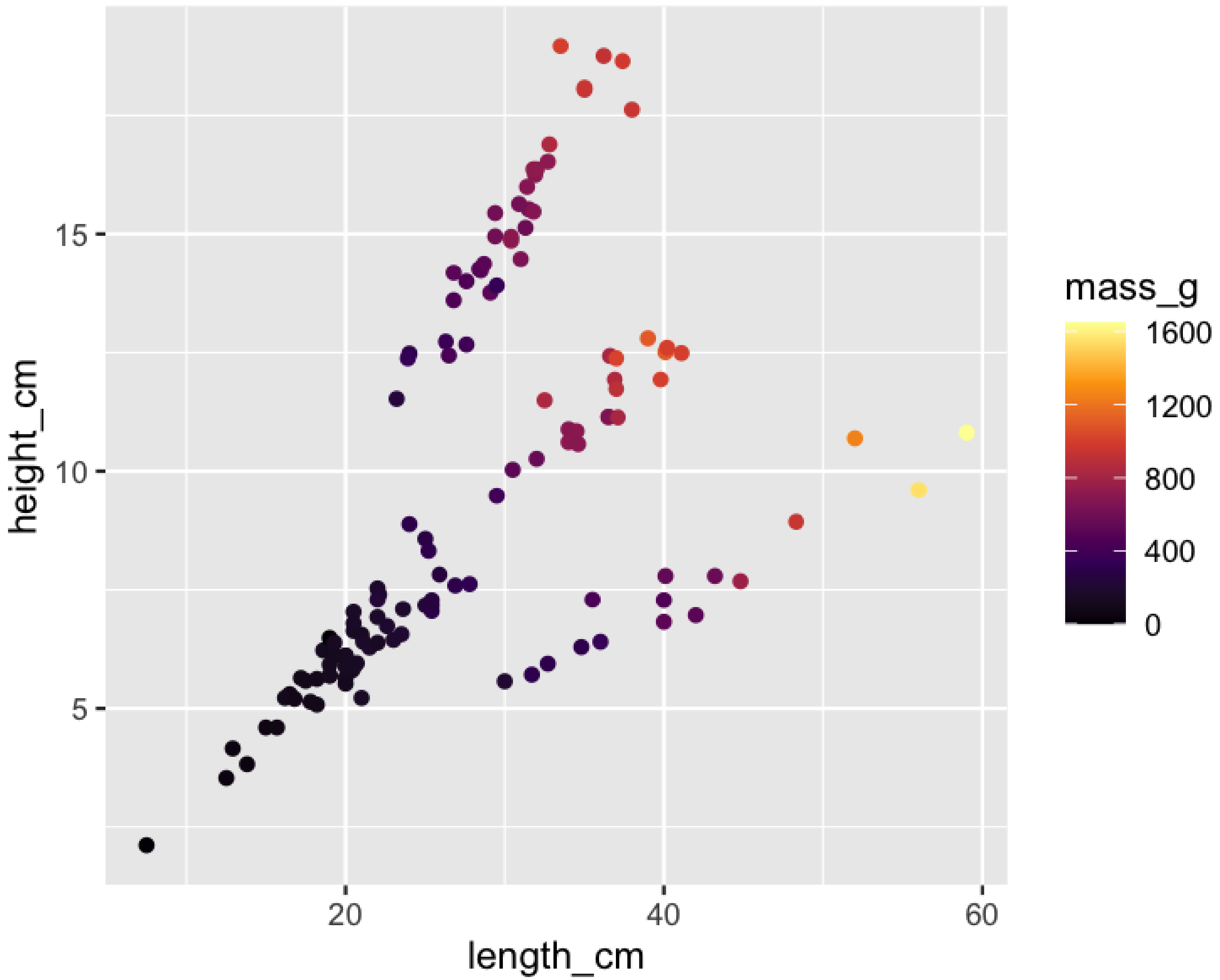
2D scatter plot, color for response

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point()
```



Viridis color scales

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno")
```



Modeling with 2 numeric explanatory variables

```
mdl_mass_vs_both <- lm(mass_g ~ length_cm + height_cm, data = fish)
```

```
Call:
lm(formula = mass_g ~ length_cm + height_cm, data = fish)


Coefficients:
(Intercept)    length_cm    height_cm
    -622.16        28.97        26.34
```

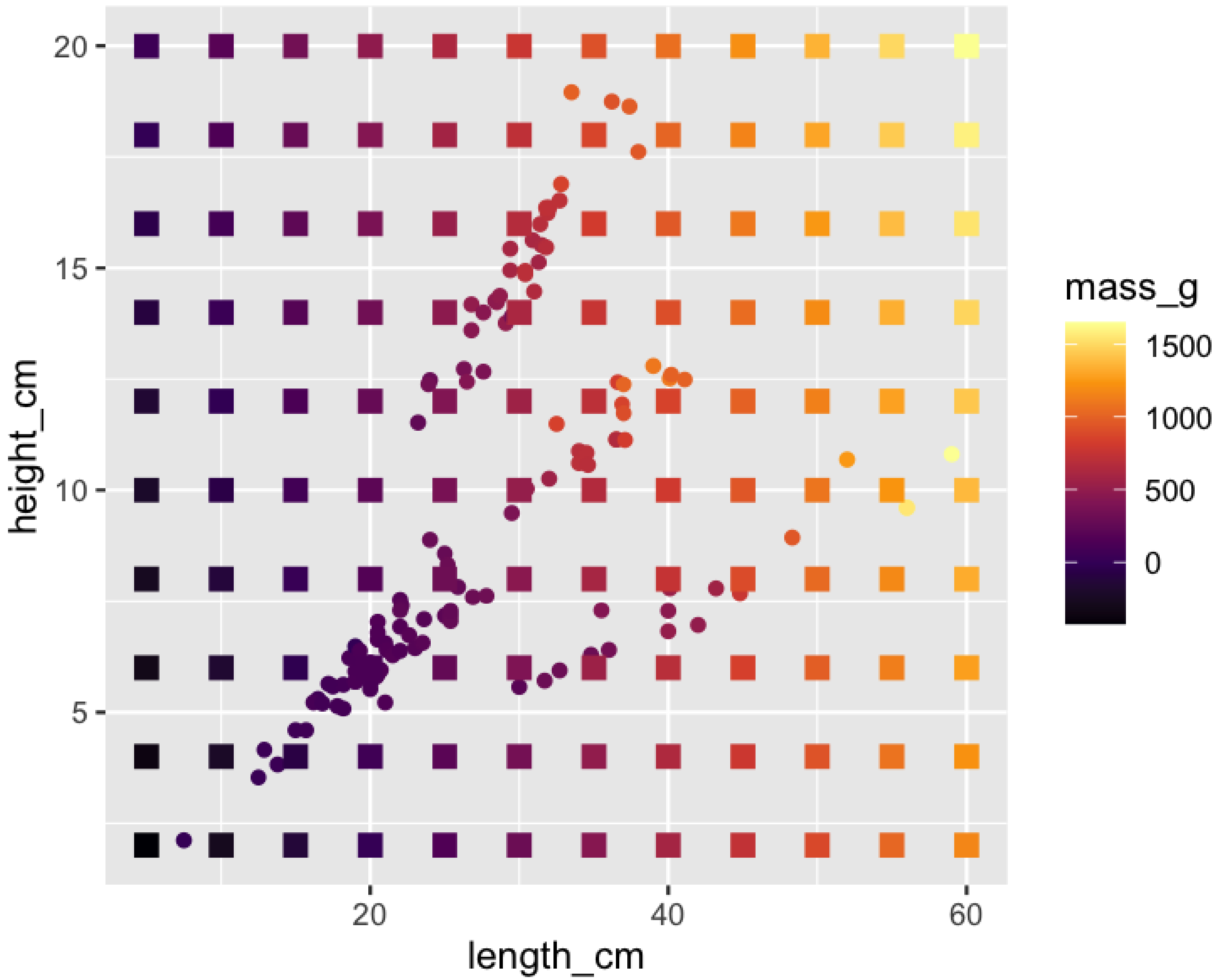
The prediction flow

```r
explanatory_data <- expand_grid(
  length_cm = seq(5, 60, 5),
  height_cm = seq(2, 20, 2)
)

prediction_data <- explanatory_data %>%
  mutate(
    mass_g = predict(mdl_mass_vs_both, explanatory_data)
  )
```

Plotting the predictions

```r
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno") +
  geom_point(
    data = prediction_data, shape = 15, size = 3
  )
```



Including an interaction

```
mdl_mass_vs_both_inter <- lm(mass_g ~ length_cm * height_cm, data = fish)
```

```
Call:
lm(formula = mass_g ~ length_cm * height_cm, data = fish)


Coefficients:
        (Intercept)            length_cm            height_cm  length_cm:height_cm
           159.1144               0.3001             -78.1234               3.5455
```

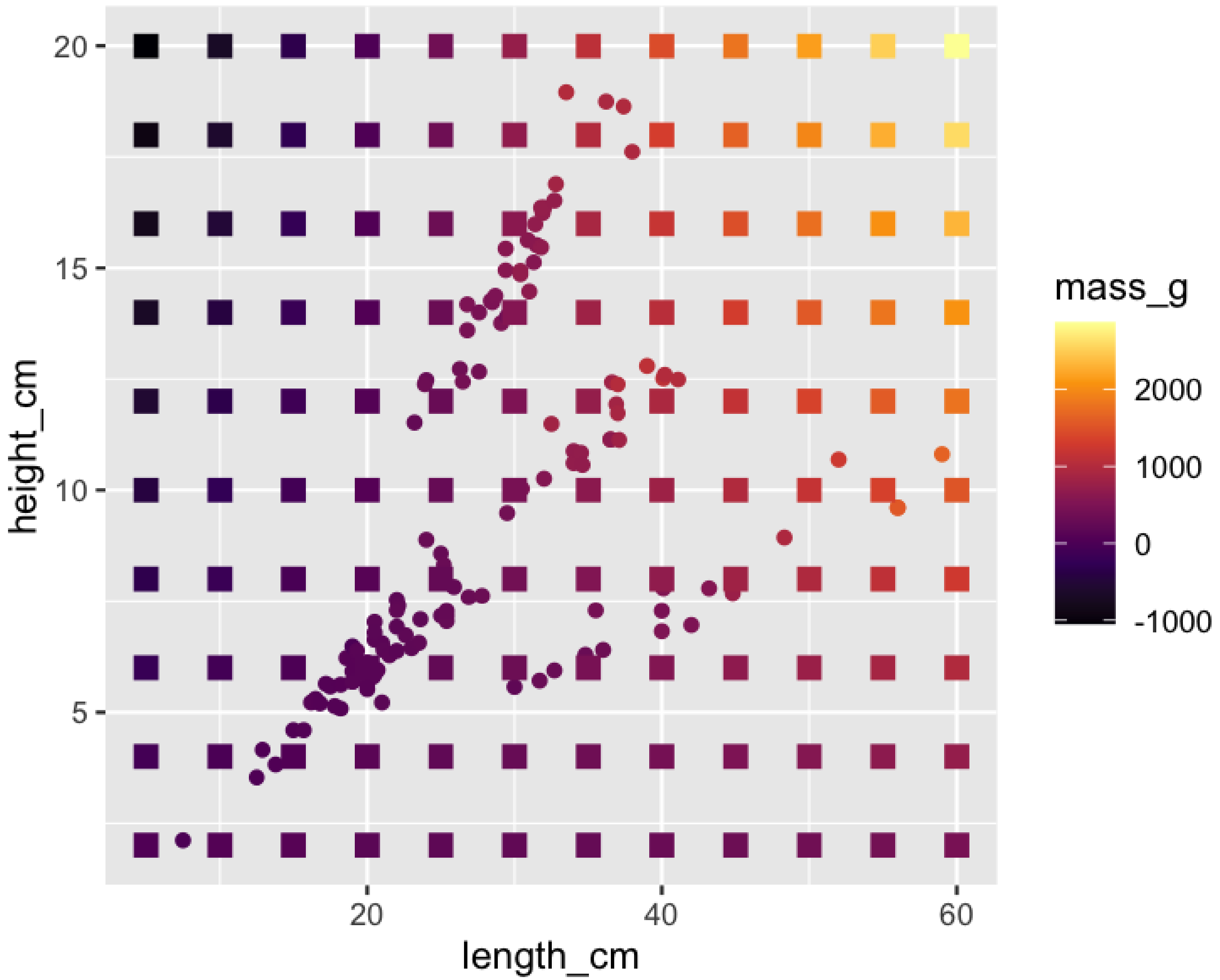
The prediction flow again

```
explanatory_data <- expand_grid(
  length_cm = seq(5, 60, 5),
  height_cm = seq(2, 20, 2)
)

prediction_data <- explanatory_data %>%
  mutate(
    mass_g = predict(mdl_mass_vs_both_inter, explanatory_data)
  )
```

Plotting the predictions

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno") +
  geom_point(
    data = prediction_data, shape = 15, size = 3
  )
```



Let's practice!

INTERMEDIATE REGRESSION IN R

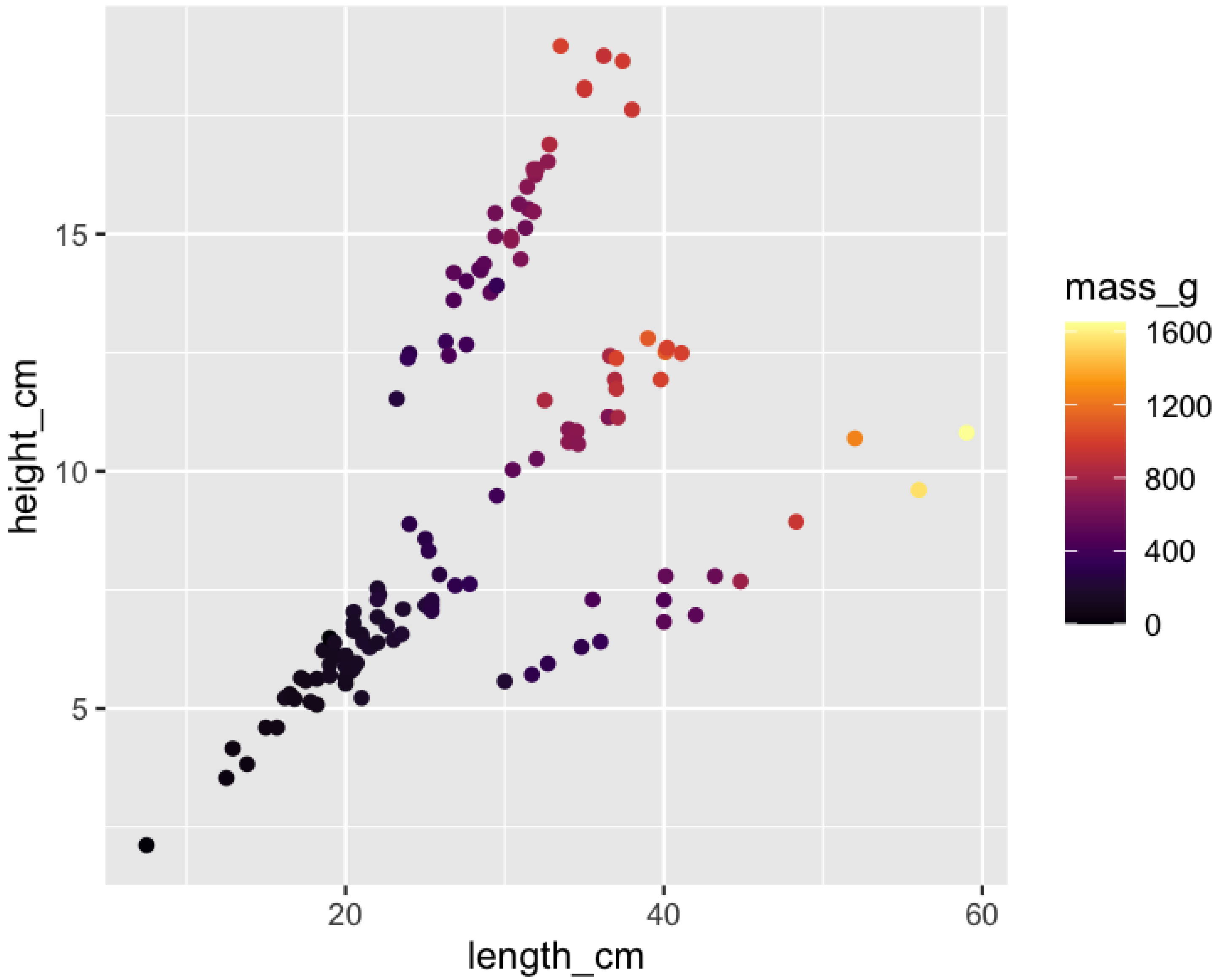
More than 2 explanatory variables

INTERMEDIATE REGRESSION IN R

Richie Cotton
Learning Solutions Architect at DataCamp

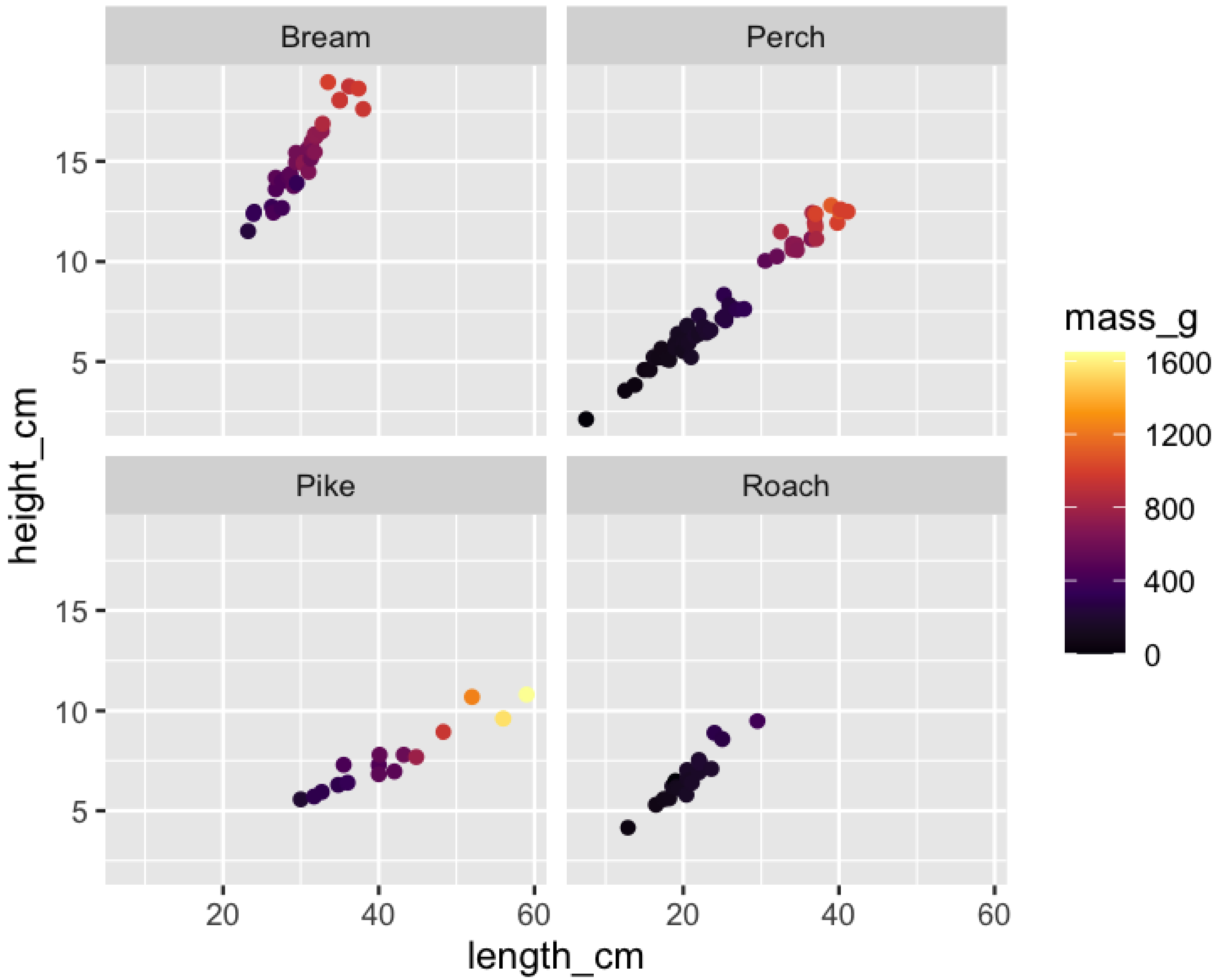
From last time

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno")
```



Faceting by species

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno") +
  facet_wrap(vars(species))
```



Different levels of interaction

No interactions

```
lm(mass_g ~ length_cm + height_cm + species + 0, data = fish)
```

2-way interactions between pairs of variables

```
lm(
  mass_g ~ length_cm + height_cm + species + length_cm:height_cm + length_cm:species + height_cm:species + 0,
  data = fish
)
```

3-way interaction between all three variables

```
lm(
  mass_g ~ length_cm + height_cm + species + length_cm:height_cm + length_cm:species + height_cm:species + length_cm:height_cm:species + 0,
  data = fish
)
```

All the interactions

```
lm(
  mass_g ~ length_cm + height_cm + species + length_cm:height_cm + length_cm:species + height_cm:species + length_cm:height_cm:species + 0,
  data = fish
)
```

```
lm(
  mass_g ~ length_cm * height_cm * species + 0,
  data = fish
)
```

Only 2-way interactions

```
lm(
  mass_g ~ length_cm + height_cm + species + length_cm:height_cm + length_cm:species + height_cm:species + 0,
  data = fish
)
```

```
lm(
  mass_g ~ (length_cm + height_cm + species) ^ 2 + 0,
  data = fish
)
```

```
lm(
  mass_g ~ I(length_cm) ^ 2 + height_cm + species + 0,
  data = fish
)
```

[1] To square explanatory variables, see "Introduction to Regression in R", Chapter 2, "Transforming variables"

The prediction flow

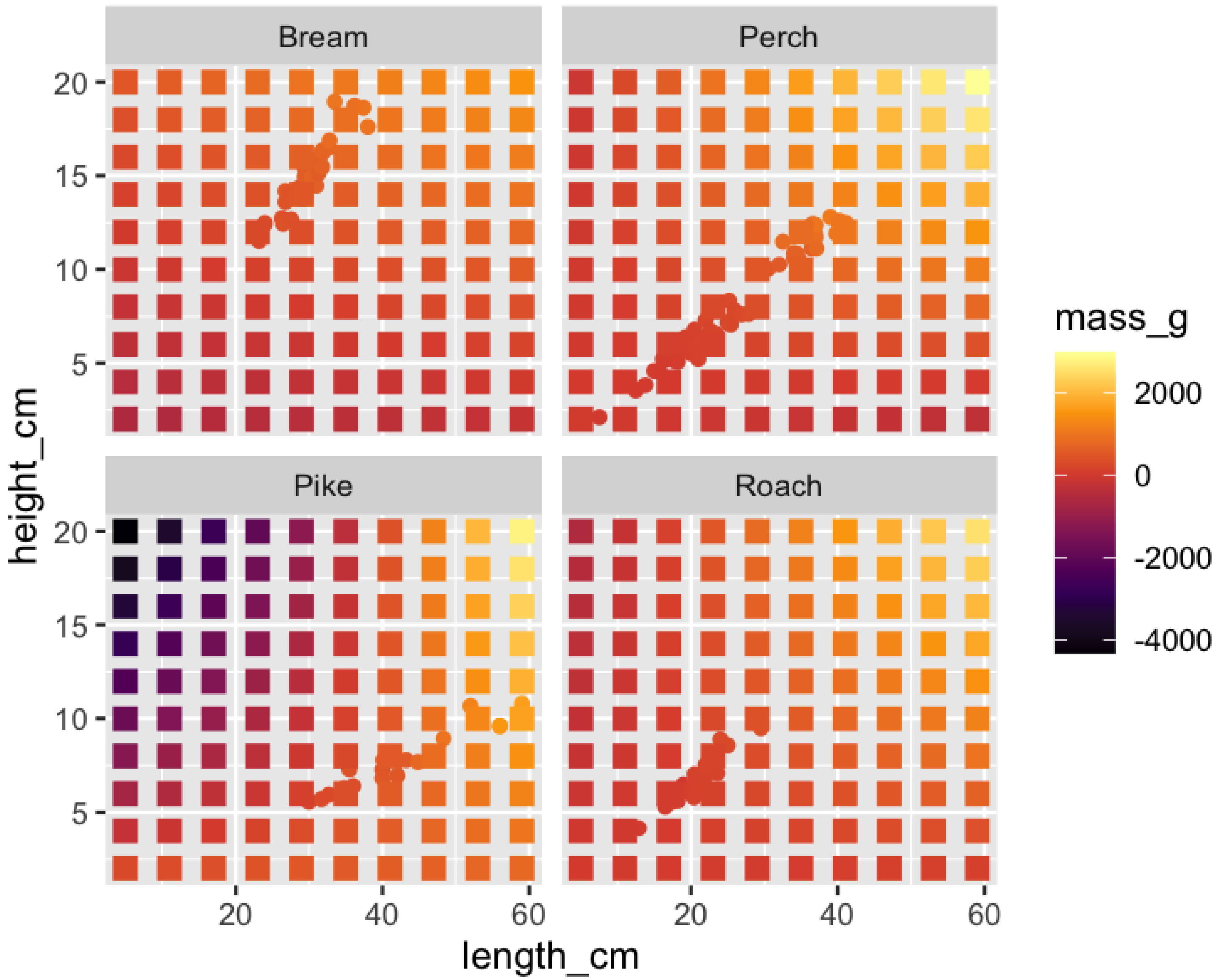
```
mdl_mass_vs_all <- lm(mass_g ~ length_cm * height_cm * species * 0, data = fish)

explanatory_data <- expand_grid(
  length_cm = seq(5, 60, 6),
  height_cm = seq(2, 20, 2),
  species = unique(fish$species)
)

prediction_data <- explanatory_data %>%
  mutate(mass_g = predict(mdl_mass_vs_all, explanatory_data))
```

Visualizing predictions

```
ggplot(
  fish,
  aes(length_cm, height_cm, color = mass_g)
) +
  geom_point() +
  scale_color_viridis_c(option = "inferno") +
  facet_wrap(vars(species)) +
  geom_point(
    data = prediction_data,
    size = 3, shape = 15
  )
```



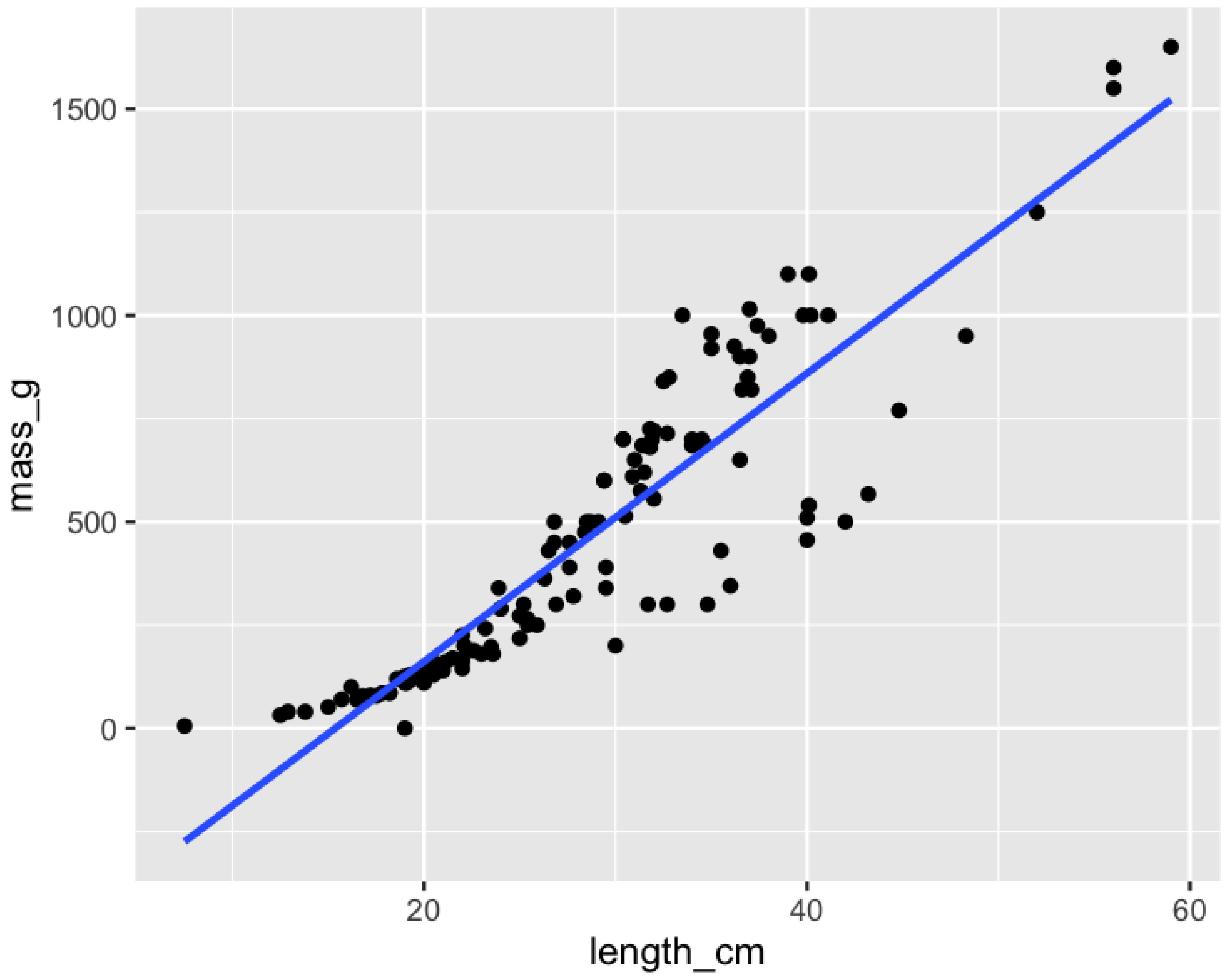
Let's practice!

How linear regression works

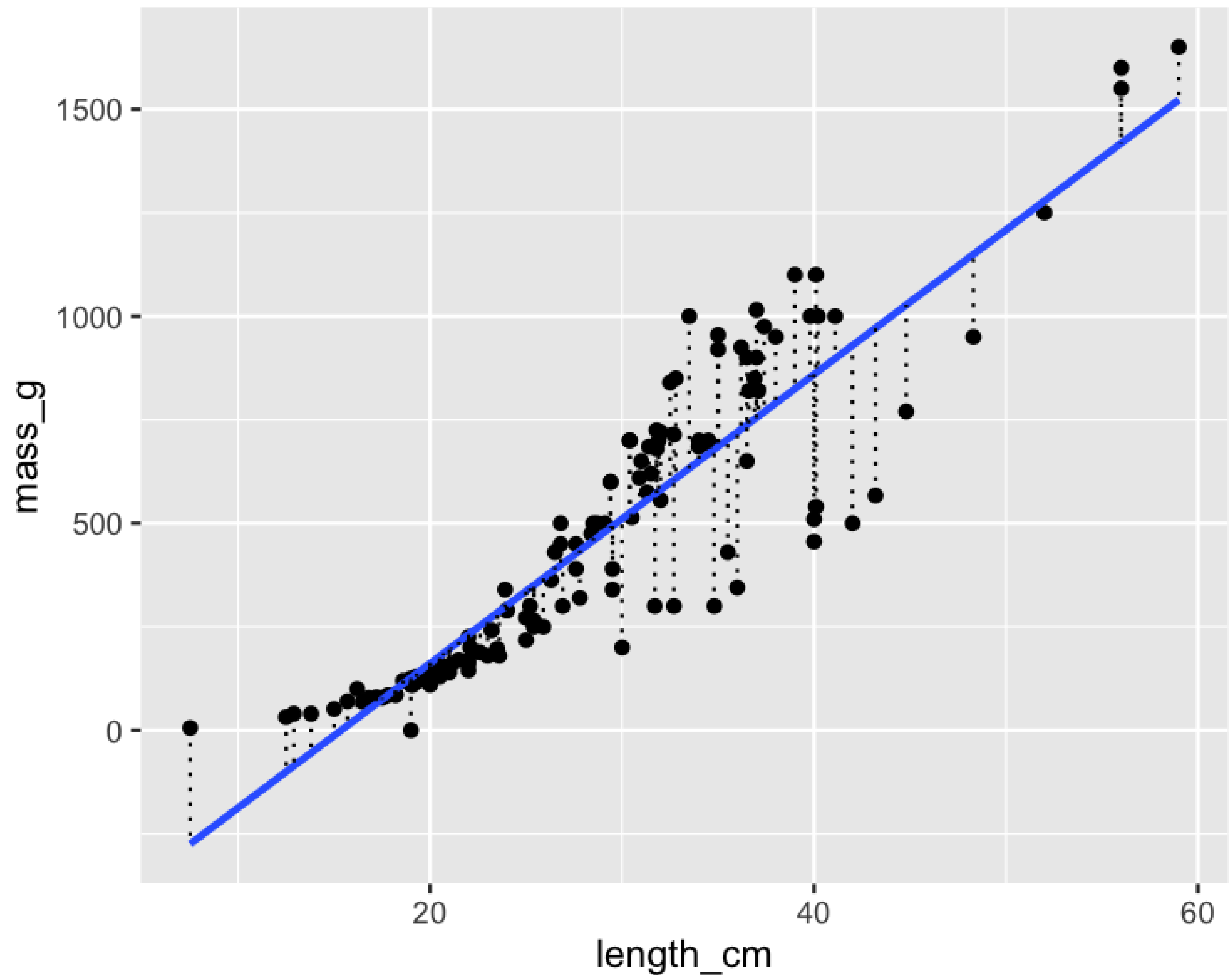
INTERMEDIATE REGRESSION IN R

Richie Cotton
Learning Solutions Architect at DataCamp

The standard simple linear regression plot



Visualizing residuals



A metric for the best fit

The simplest idea (which doesn't work)

- Take the sum of all the residuals.
- Some residuals are negative.

The next simplest idea (which does work)

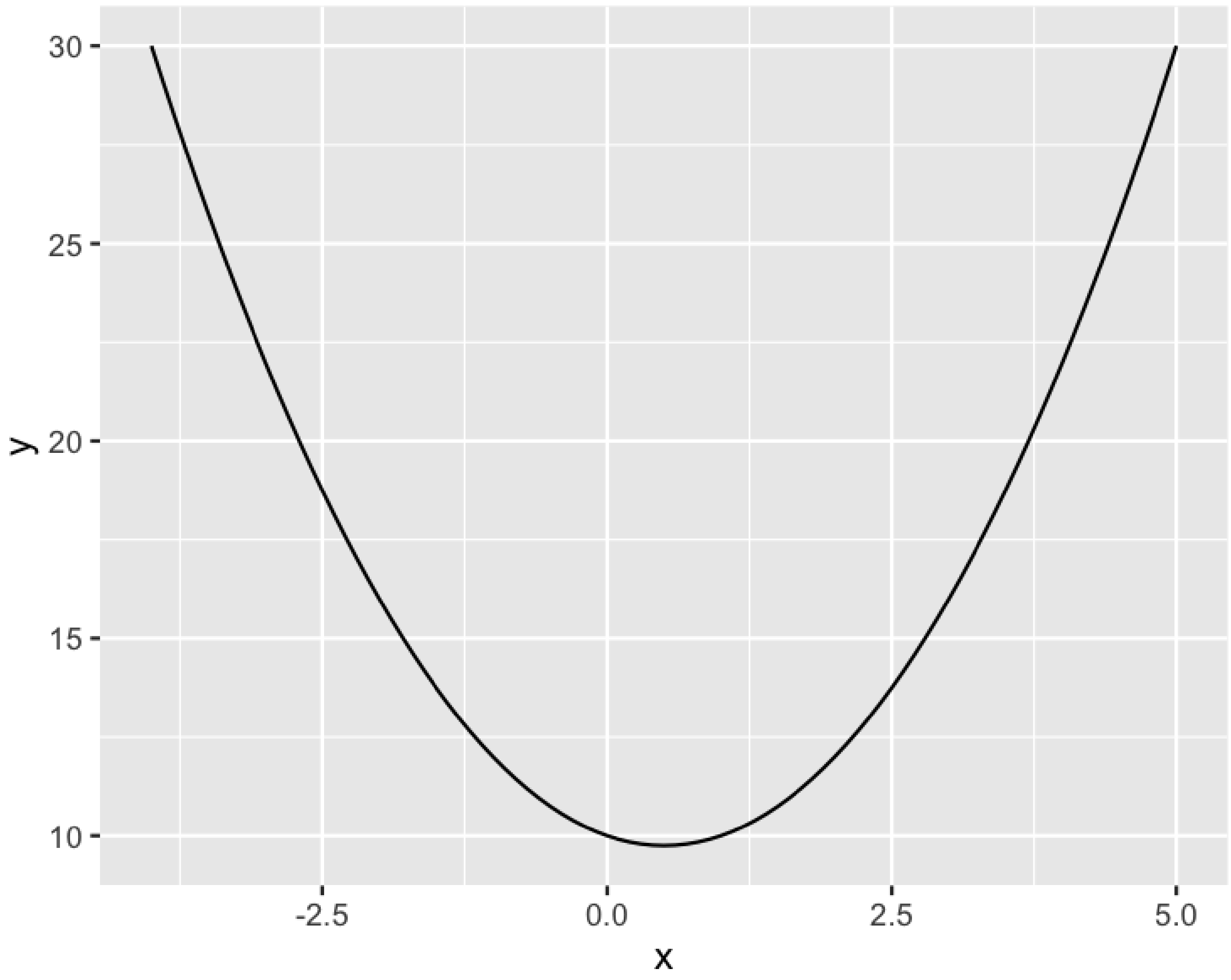
- Take the square of each residual, and add up those squares.
- This is called the *sum of squares*.

A detour into numerical optimization

A line plot of a quadratic equation

```
xy_data <- tibble(
  x = seq(-4, 5, 0.1),
  y = x ^ 2 - x + 10
)

ggplot(xy_data, aes(x, y)) +
  geom_line()
```



Using calculus to solve the equation

$y = x^2 - x + 10$

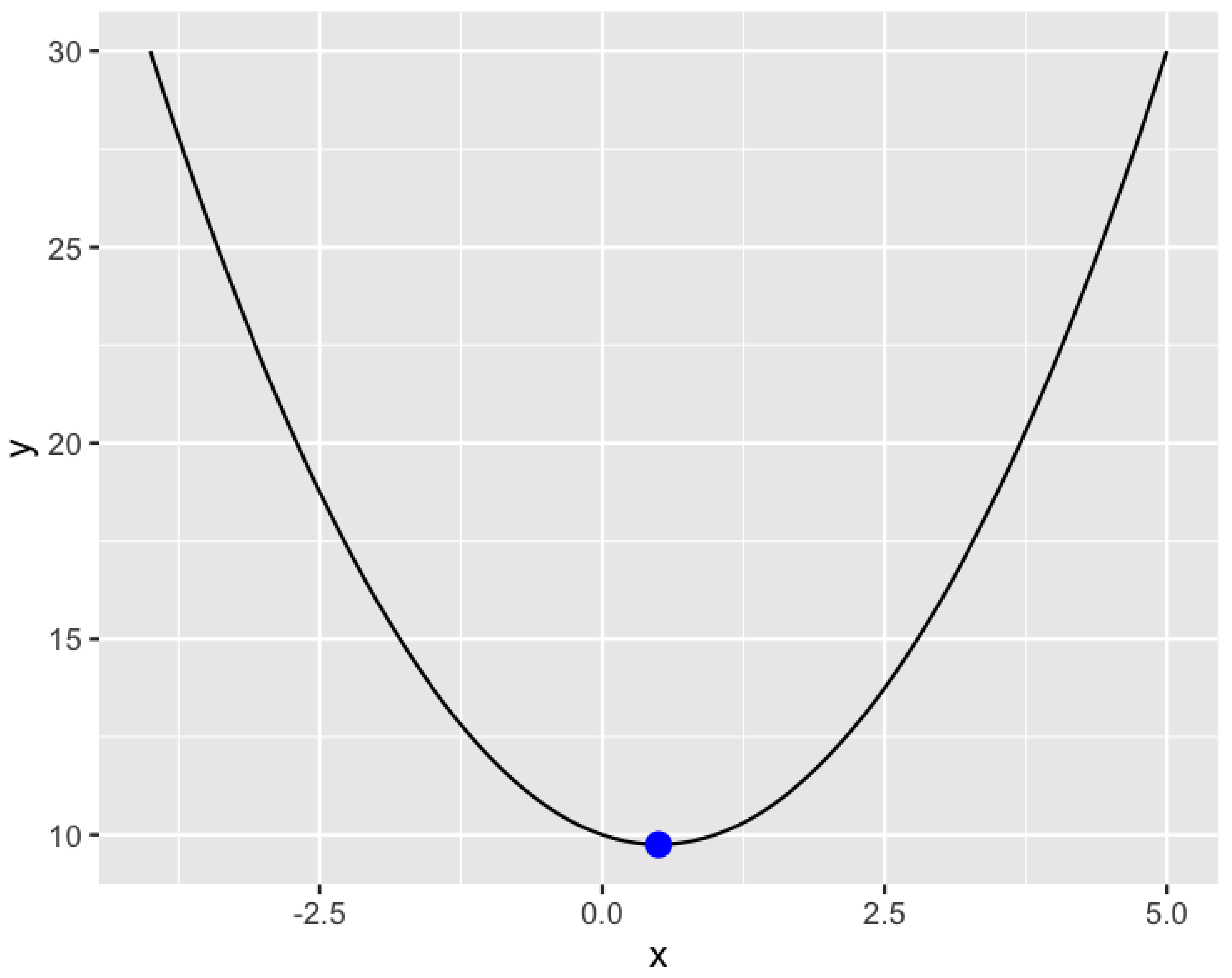
$\frac{\partial y}{\partial x} = 2x - 1$

$0 = 2x - 1$

$x = 0.5$

$y = 0.5^2 - 0.5 + 10 = 9.75$

- Not all equations can be solved like this.
- You can let R figure it out.



optim()

```r
calc_quadratic <- function(x) {
  x ^ 2 - x + 10
}
```

```r
optim(par = 3, fn = calc_quadratic)
```

```
$par
[1] 0.4998047

$value
[1] 9.75

$counts
function gradient
      30       NA

$convergence
[1] 0

$message
NULL
```

Slight refinements

```
calc_quadratic <- function(coeffs) {
  x <- coeffs[1]
  x ^ 2 - x + 10
}
```

```
optim(par = c(x = 3), fn = calc_quadratic)
```

```
$par
        x
0.4998047

$value
[1] 9.75

$counts
function gradient
      30       NA

$convergence
[1] 0

$message
NULL
```

A linear regression algorithm

1. Define a function to calculate the sum of squares metric.
2. Call `optim()` to find coefficients that minimize this function.

```
calc_sum_of_squares <- function(coeffs) {
  intercept <- coeffs[1]
  slope <- coeffs[2]
  # More calculation!
}
```

```
optim(
  par = ???,
  fn = ???
)
```

Let's practice!