

Correlation and Regression

PSYC 2020-A01 / PSYC 6022-A01 | 2025-11-14 | Lab 13

Jessica Helmer

Outline

- Assignment 12 Review
- Correlation
- Regression

Learning objectives:

R: Correlation, regression

Assignment 12 Review

[placeholder for Assignment 12 review]

Correlation

Correlation

Measure of association: how strongly are two variables related?

Indexes the linear relationship between two variables

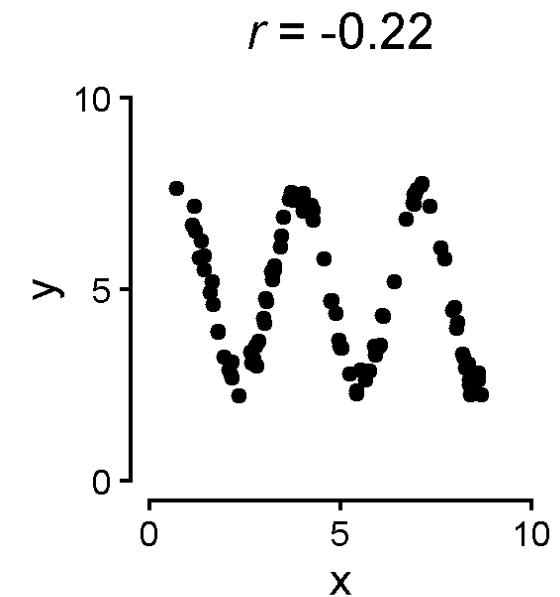
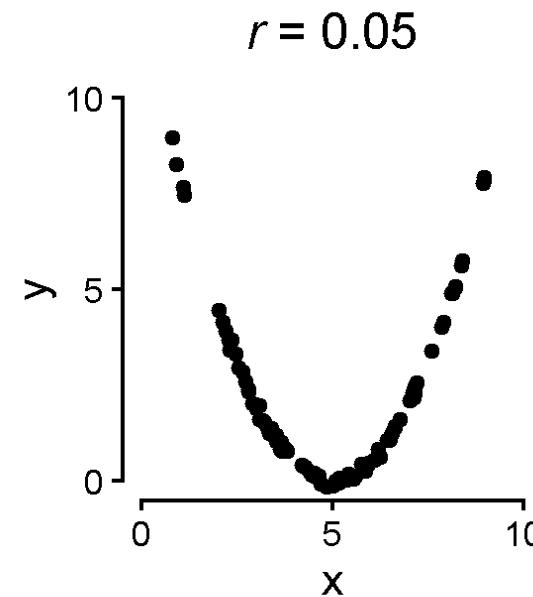
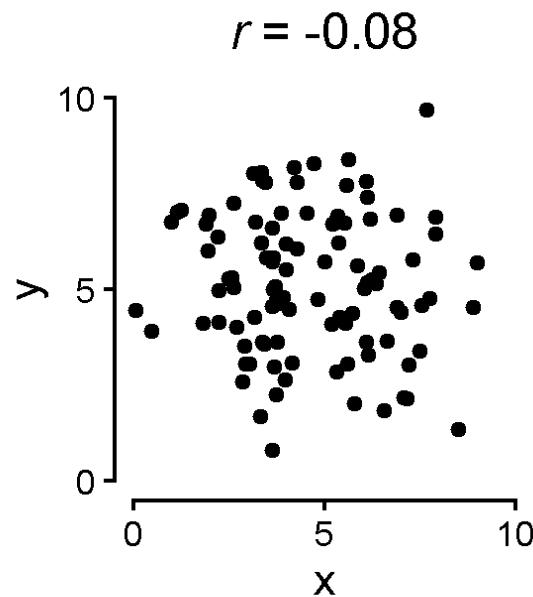
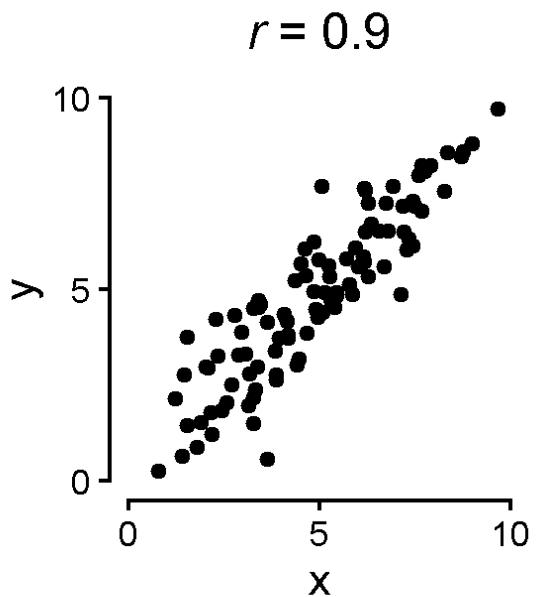
We will cover correlation for two continuous variables

Correlation

Only measures **linear** relationships

Plot

Code



Correlation Generally

$$r_{xy} = \frac{\sum_{i=1}^I (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

where

i = index of observation i out of I total observations

\bar{x} = mean of x

\bar{y} = mean of y

s_x = standard deviation of x

s_y = standard deviation of y

Correlation Generally

$$r_{xy} = \frac{\sum_{i=1}^I (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

By subtracting the mean and dividing by the standard deviation, this formula is converting the observations to z-scores.

Correlation Example

$$r_{xy} = \frac{\sum_{i=1}^I (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

Let's find the correlation between iris' sepal length and sepal width. For demonstration, let's only use the first four rows.

```
1 iris_small <- iris |>
2   select(Sepal.Length, Sepal.Width) |>
3   head(4)
4 iris_small
```

	Sepal.Length	Sepal.Width
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1

```
1 x_bar <- mean(iris_small$Sepal.Length)
2 x_sd <- sd(iris_small$Sepal.Length)
3
4 y_bar <- mean(iris_small$Sepal.Width)
5 y_sd <- sd(iris_small$Sepal.Width)
6
7 paste(x_bar, x_sd, y_bar, y_sd, sep = ", ")
```

```
[1] "4.825, 0.221735578260834, 3.2,
0.216024689946929"
```

Correlation Example

$$r_{xy} = \frac{\sum_{i=1}^I (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

Let's find the correlation between iris' sepal length and sepal width. For demonstration, let's only use the first four rows.

```
iris_small <- iris_small |>  
  mutate(sq_diff_x = Sepal.Length - x_bar,  
        sq_diff_y = Sepal.Width - y_bar,  
        product = sq_diff_x * sq_diff_y)  
  
iris_small
```

	Sepal.Length	Sepal.Width	sq_diff_x	sq_diff_y	product
1	5.1	3.5	0.275	0.3	0.0825
2	4.9	3.0	0.075	-0.2	-0.0150
3	4.7	3.2	-0.125	0.0	0.0000
4	4.6	3.1	-0.225	-0.1	0.0225

```
sum(iris_small$product) /  
  ((nrow(iris_small) - 1) * x_sd * y_sd)  
[1] 0.6263001
```

Correlation in R

$$r_{xy} = \frac{\sum_{i=1}^I (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y}$$

Let's find the correlation between iris' sepal length and sepal width. For demonstration, let's only use the first four rows.

```
1 iris_small <- iris |>
2   select(Sepal.Length, Sepal.Width) |>
3   head(4)
4 iris_small
```

	Sepal.Length	Sepal.Width
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1

```
1 cor(iris_small$Sepal.Length,
2     iris_small$Sepal.Width)
[1] 0.6263001
```

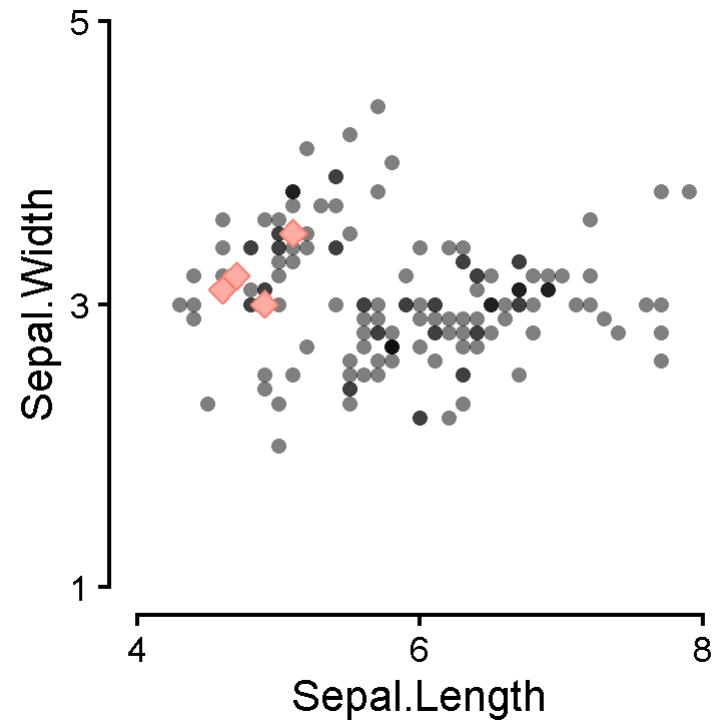
Correlation in R

As a note, those four points were not actually representative of the general trend!

Plot

Code

```
1 cor(iris$Sepal.Length, iris$Sepal.Width)
[1] -0.1175698
```



Regression

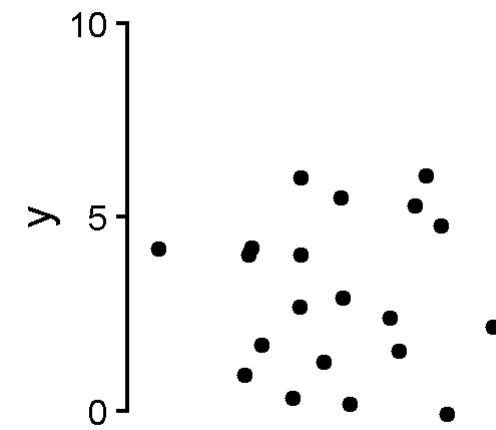
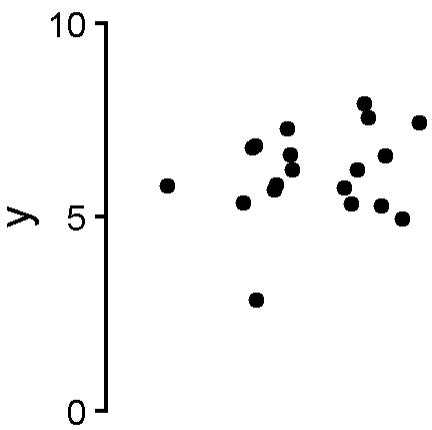
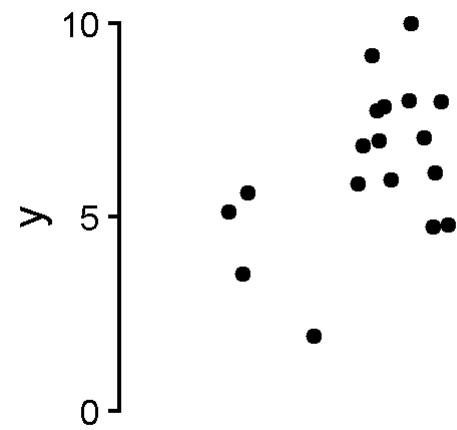
Regression

If you are working with just one variable, what is the best way you can represent the data?

If you had to pick just one statistic, what might you choose?

Plot

Code



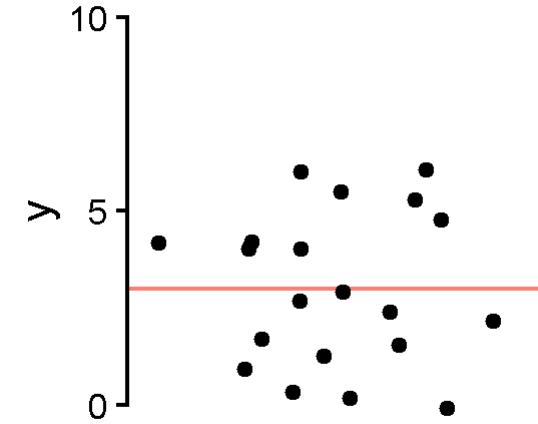
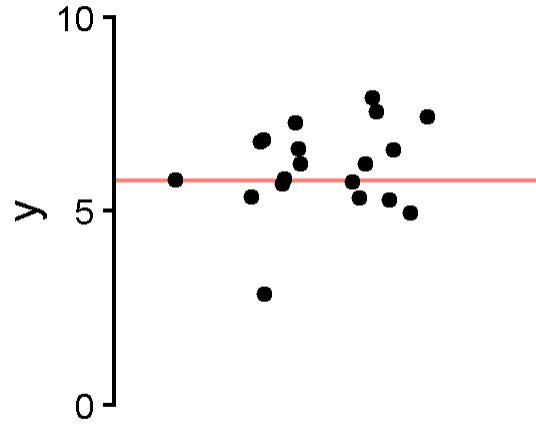
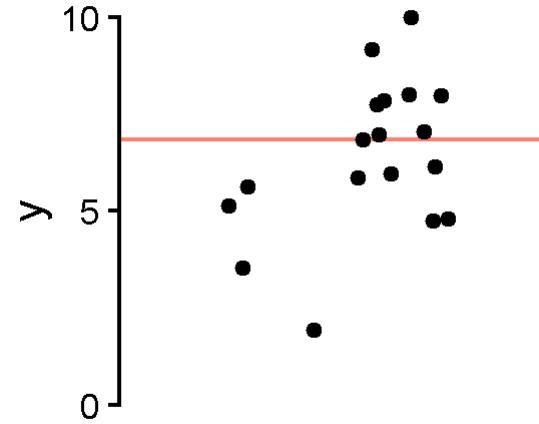
Regression

If we consider “best” to mean *minimizing the squared distances from the data*, the **mean** is the best fit.

The mean has the smallest squared errors from the observations.

[Plot](#)

[Code](#)



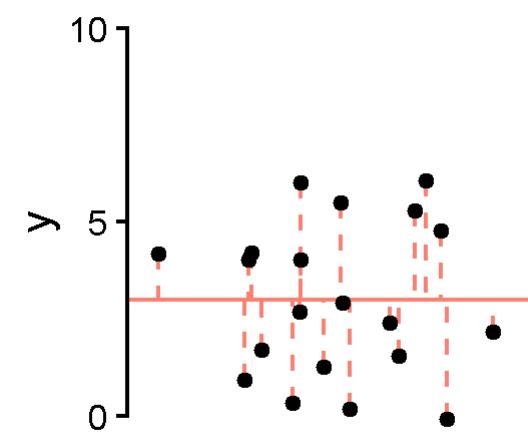
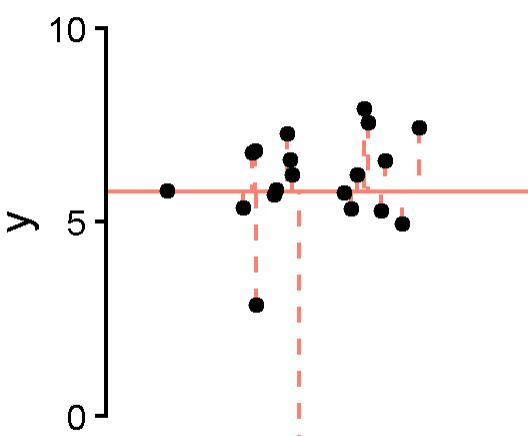
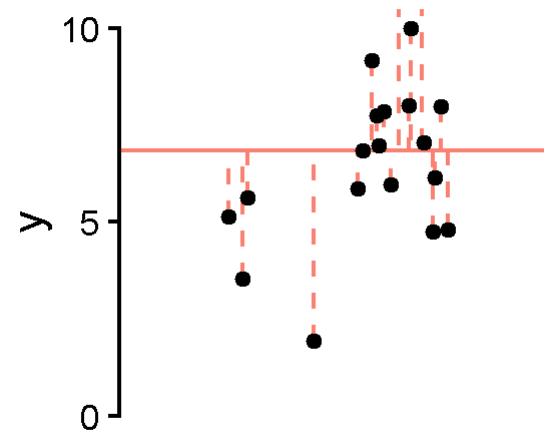
Regression

If we consider “best” to mean *minimizing the squared distances from the data*, the **mean** is the best fit.

The mean has the smallest squared errors from the observations.

[Plot](#)

[Code](#)



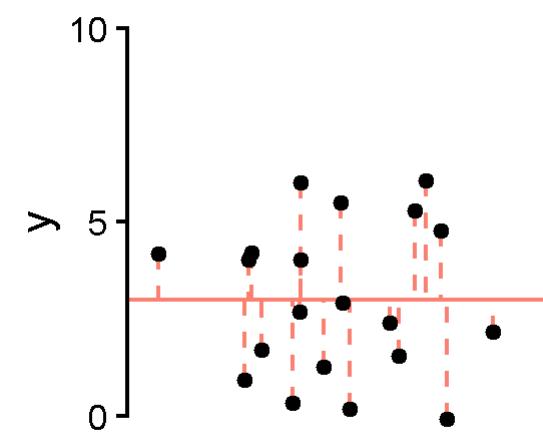
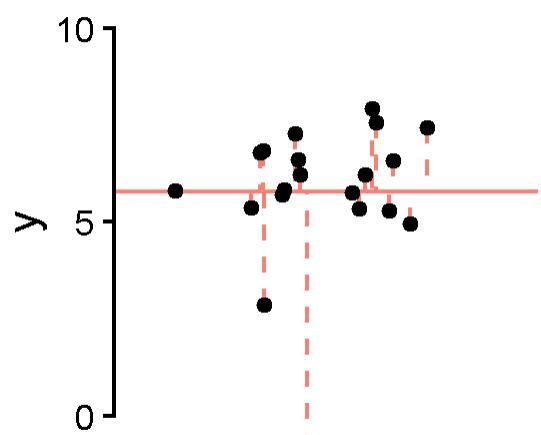
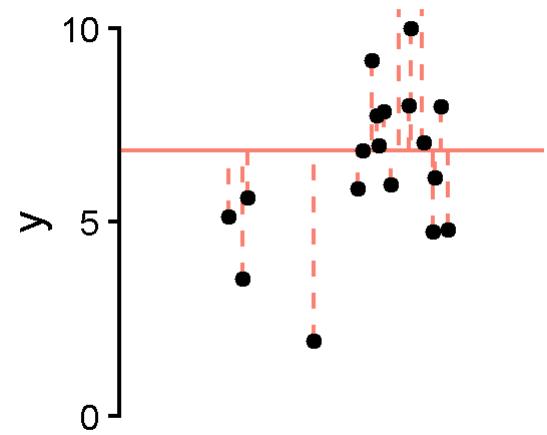
Regression

This is equivalent to the “mean intercept model” regression equation

$$y_i = \beta_0 + \epsilon_i$$

Plot

Code



Intercept Model to Regression

When we have *one* variable of interest, the mean is the best single estimator.

- $y = \beta_0 = \bar{y}$ would be the line of best fit

If we have *two* variables and we want to predict one from the other, we now may need a *slope* to find a line of best fit.

- This model will typically have both an intercept and a slope

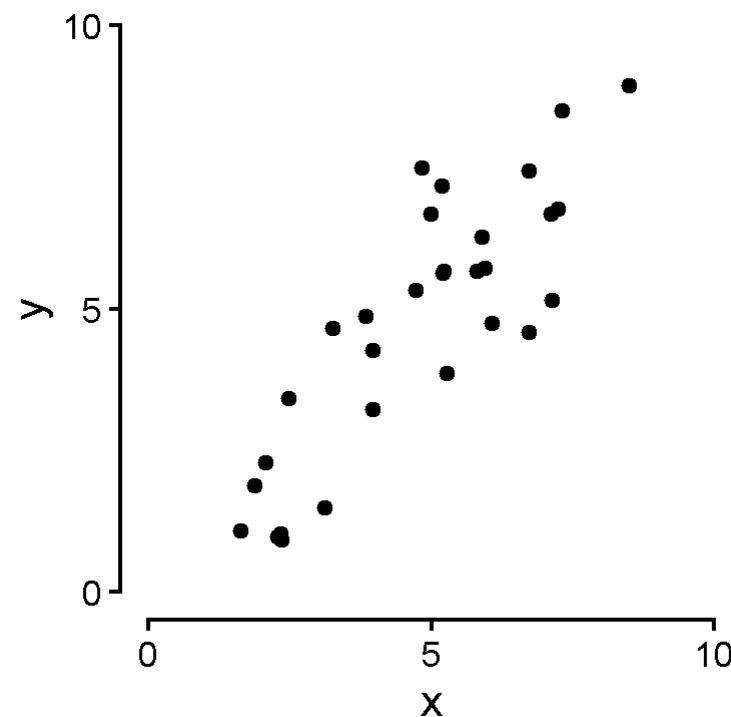
Intercept Model to Regression

We have the same question: which line can represent points most effectively?

Plot

Code

Lots of potential lines we could use



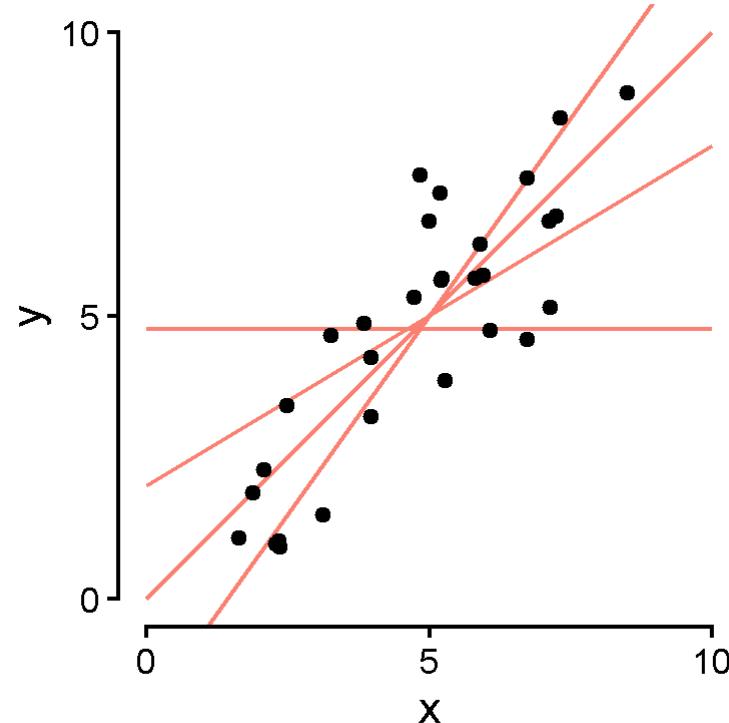
Intercept Model to Regression

We have the same question: which line can represent points most effectively?

Plot

Code

Lots of potential lines we could use



Intercept Model to Regression

Considering the relationship of y with another variable x

$$y_i = \beta_0 + \beta_1 \times x_i + \epsilon_i$$

where

i = index of observation i out of I total observations

y = outcome

x = predictor

β_0 = intercept

β_1 = slope for x

Regression Example in R

```
lm(y ~ x, data = data)
```

- `y` = outcome
- `x` = predictor, use `1` for just intercept
- `data` = dataframe that includes `x` and `y`

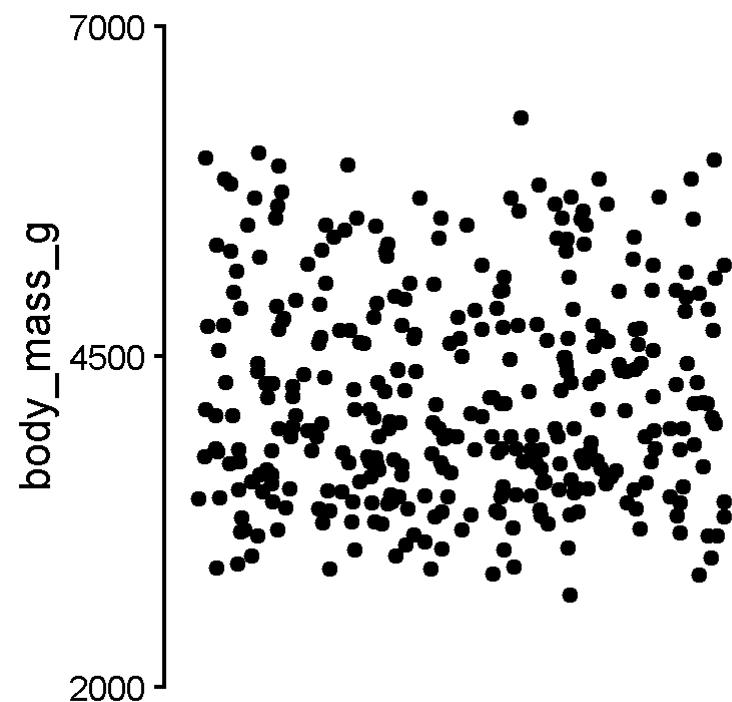
```
summary(model) to see results
```

Regression Example: penguins

Let's predict penguins' body mass!

Plot

Code



With just this one variable, what would be our best prediction for these data?

Regression Example in R: Intercept only

```
1 penguins_m0 <- lm(body_mass_g ~ 1, data = penguins)
2 summary(penguins_m0)
```

Call:

```
lm(formula = body_mass_g ~ 1, data = penguins)
```

Residuals:

Min	1Q	Median	3Q	Max
-1501.8	-651.8	-151.8	548.2	2098.2

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4201.75	43.36	96.89	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Regression Example in R: Predictions

We can extract predicted values to create our line of best fit with
`predict(model, newdata)`

- `newdata` = dataframe containing theoretical values of the predictor(s)]
 - Should have the same column name(s)
 - Don't need for intercept-only

```
1 predict(penguins_m0) |>  
2 head(16)
```

1	2	3	5	6	7	8	9
4201.754	4201.754	4201.754	4201.754	4201.754	4201.754	4201.754	4201.754
10	11	12	13	14	15	16	17
4201.754	4201.754	4201.754	4201.754	4201.754	4201.754	4201.754	4201.754

```
penguins_m0 |> predict() |>  
length()
```

[1] 342

```
penguins |> select(body_mass_g) |>  
nrow()
```

[1] 344

```
penguins |> select(body_mass_g) |>  
drop_na() |> nrow()
```

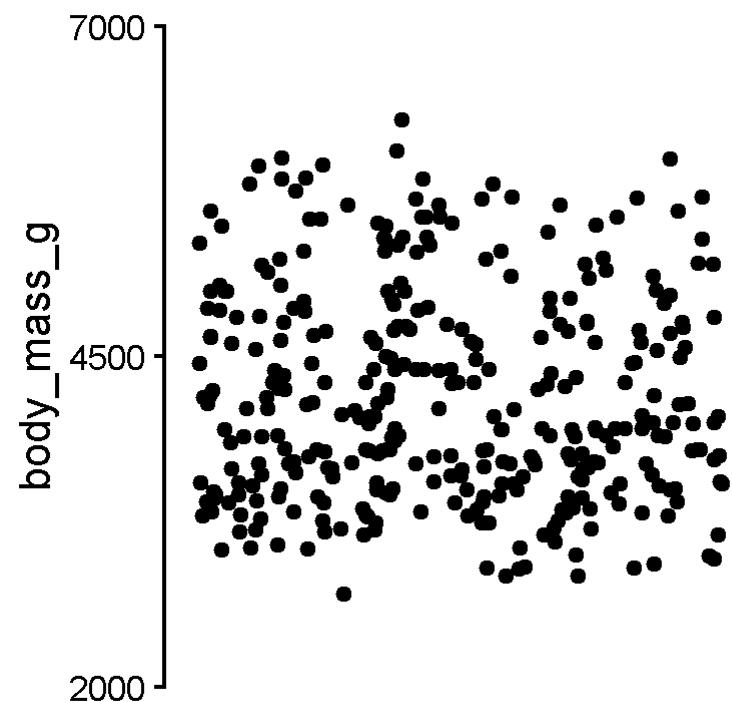
[1] 342

Regression Example: penguins

Let's predict penguins' body mass!

Plot

Code



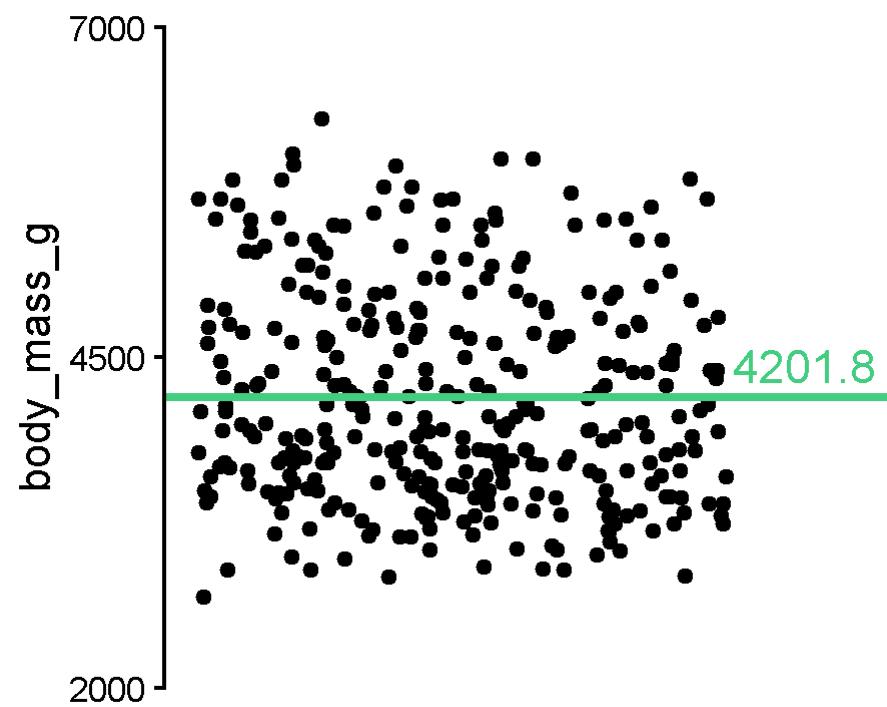
With just this one variable, what would be our best prediction for these data? The mean!

Regression Example: penguins

Let's predict penguins' body mass!

Plot

Code



With just this one variable, what would be our best prediction for these data? The mean!

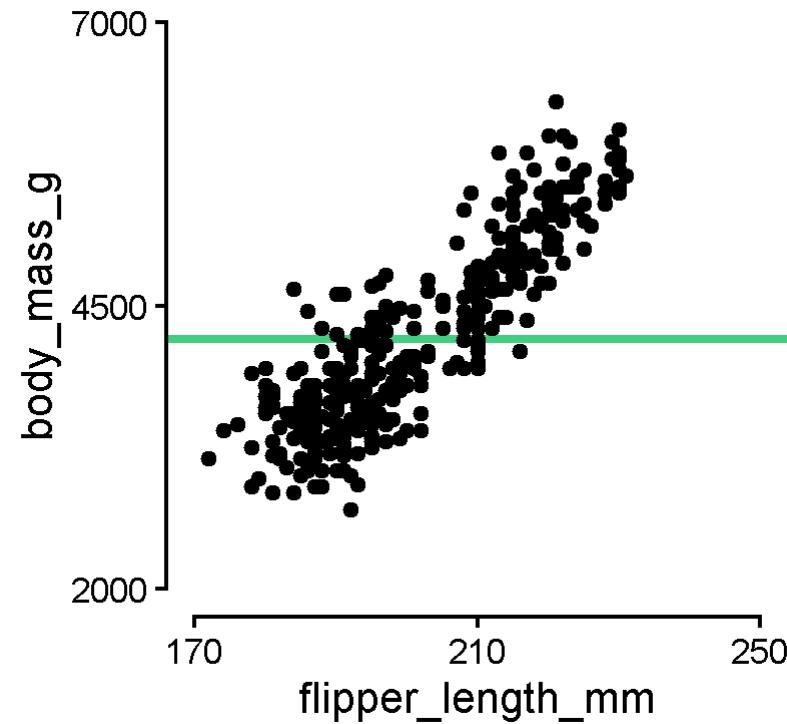
If we needed to predict any random penguin's body mass with only these data, our *best estimate* would be the mean, 4201.8.

Now, let's take into account another variable: a penguin's flipper length

Regression Example: penguins

Plot

Code



Now, let's take into account another variable: a penguin's flipper length

Just the mean no longer seems like our line of best fit

Regression Example in R: Adding a predictor

```
1 penguins_m1 <- lm(body_mass_g ~ flipper_length_mm, data = penguins)
2 summary(penguins_m1)
```

Call:

```
lm(formula = body_mass_g ~ flipper_length_mm, data = penguins)
```

Residuals:

Min	1Q	Median	3Q	Max
-1058.80	-259.27	-26.88	247.33	1288.69

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5780.831	305.815	-18.90	<2e-16 ***
flipper_length_mm	49.686	1.518	32.72	<2e-16 ***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

Regression Example in R: Output anatomy

summary()

Coefficients

R^2

F-Statistic

```
1 summary(penguins_m1)
```

Call:

```
lm(formula = body_mass_g ~ flipper_length_mm, data = penguins)
```

Residuals:

Min	1Q	Median	3Q	Max
-1058.80	-259.27	-26.88	247.33	1288.69

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5780.831	305.815	-18.90	<2e-16 ***
flipper_length_mm	49.686	1.518	32.72	<2e-16 ***

Signif. codes:	0 ****	0.001 **	0.01 *	0.05 .
	'	'	'	'
	1			

Regression Example in R: Line of best fit

We can extract predicted values to create our line of best fit with

`predict(model, newdata)`

- `newdata` = dataframe containing theoretical values of the predictor(s)
- Should have the same column name(s)

```
1 penguins |> select(flipper_length_mm) |> range
```

```
[1] 172 231
```

```
1 newdata <- data.frame(flipper_length_mm = 170)
2 head(newdata, 50)
```

```
flipper_length_mm
```

```
1 170
2 171
3 172
4 173
5 174
6 175
7 176
8 177
9 178
10 179
11 180
12 181
13 182
```

```
1 predict(penguins_m1, newdata = newdata) |>
```

```
2 head(32)
```

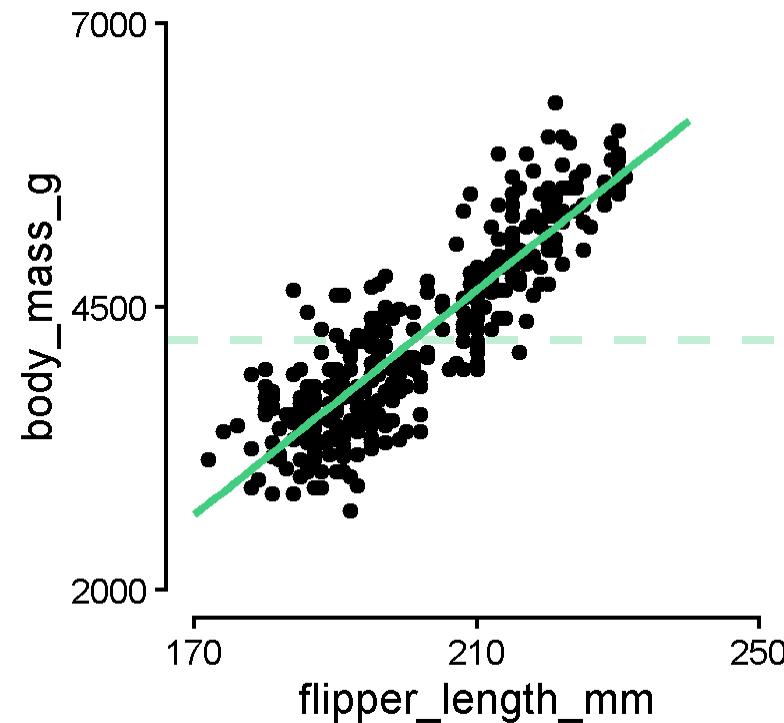
	1	2	3	4	5
6	7	8			
2665.715	2715.400	2765.086	2814.772	2864.457	
2914.143	2963.828	3013.514			
	9	10	11	12	13
	14	15	16		
3063.199	3112.885	3162.571	3212.256	3261.942	
3311.627	3361.313	3410.998			
	17	18	19	20	21
	22	23	24		
3460.684	3510.370	3560.055	3609.741	3659.426	
3709.112	3758.797	3808.483			
	25	26	27	28	29
	30	31	32		

Regression Example in R: Line of best fit

```
1 predicted_data <- data.frame(flipper_length_mm = seq(170, 240, 1)) |>  
2 mutate(predicted_body_mass = predict(penguins_m1, newdata = data.frame(flipper_length_mm)))
```

Plot

Code



Regression Example in R: More predictors

We can add predictors to the right hand side of the formula with `+`.

```
1 penguins_m2 <- lm(body_mass_g ~ flipper_length_mm + bill_length_mm, data = penguins)
2 summary(penguins_m2)
```

Call:

```
lm(formula = body_mass_g ~ flipper_length_mm + bill_length_mm,
   data = penguins)
```

Residuals:

Min	1Q	Median	3Q	Max
-1090.5	-285.7	-32.1	244.2	1287.5

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5736.897	307.959	-18.629	<2e-16 ***
flipper_length_mm	48.145	2.011	23.939	<2e-16 ***
bill_length_mm	6.047	5.180	1.168	0.244

Regression Example in R: More predictors

Can get confidence intervals around predictions with `confint(model)`

```
1 confint(penguins_m2)
```

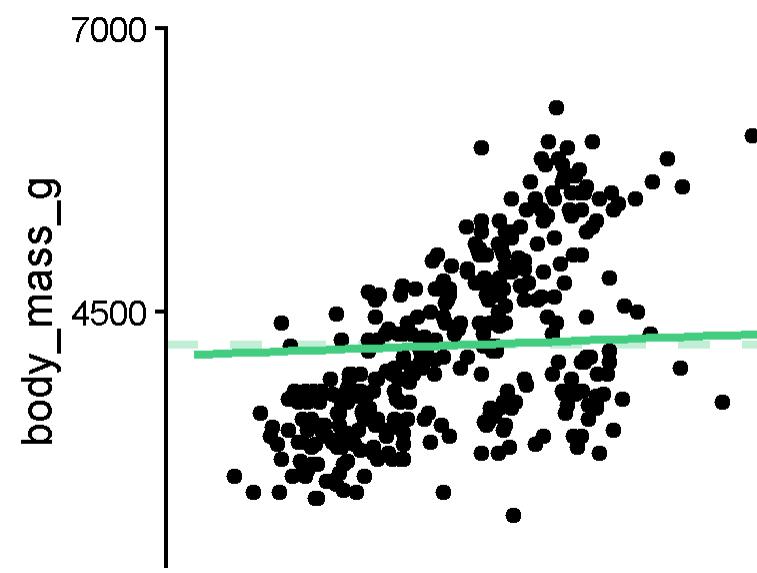
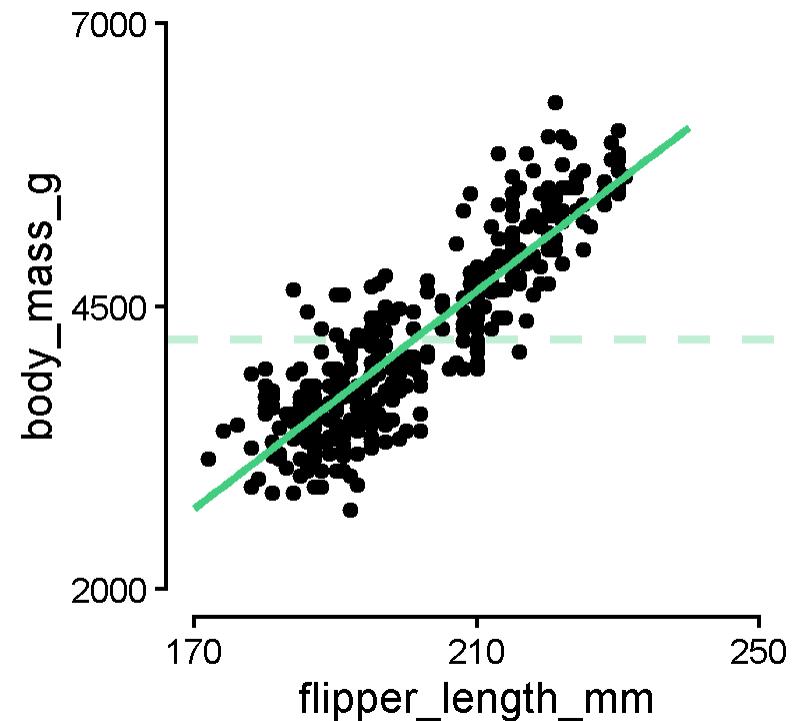
	2.5 %	97.5 %
(Intercept)	-6342.64860	-5131.14573
flipper_length_mm	44.18902	52.10069
bill_length_mm	-4.14117	16.23615

Regression Example in R: More predictions

```
1 predicted_data_f1 <- data.frame(flipper_length_mm = 170:240,  
2                                     bill_length_mm = mean(penguins$bill_length_mm, na.rm = T)) |>  
3   mutate(predicted_body_mass = predict(penguins_m2, newdata = data.frame(flipper_length_mm,  
4                                         bill_length_mm)))  
5  
6 predicted_data_b1 <- data.frame(flipper_length_mm = mean(penguins$flipper_length_mm, na.rm = T)  
7                                     bill_length_mm = 30:60) |>  
8   mutate(predicted_body_mass = predict(penguins_m2, newdata = data.frame(flipper_length_mm,  
9                                         bill_length_mm)))
```

Plot

Code



Regression Example: Writeup

Controlling for bill length, the effect of flipper length on penguins body mass was significant ($p < .001$). For every mm increase in flipper length, we expect to see a 46.15g increase in body mass. Controlling for flipper length, the effect of bill length on penguins' body mass was not significant ($p = .244$). The expected body mass when flipper length and bill length are both zero is -5736.897.

Note

Why is the intercept so crazy? Because flipper length and bill length would never actually be zero!

Assignment 13