

# Models for each category

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Data Evangelist at DataCamp

# 4 categories

```
unique(fish$species)
```

```
"Bream" "Roach" "Perch" "Pike"
```

# Splitting the dataset

## The smart way

- **base-R**: `split()` + `lapply()`
- **dplyr**: `nest_by()` + `mutate()`

## The simple way

```
bream <- fish %>%  
  filter(species == "Bream")  
perch <- fish %>%  
  filter(species == "Perch")  
pike <- fish %>%  
  filter(species == "Pike")  
roach <- fish %>%  
  filter(species == "Roach")
```

# 4 models

```
mdl_bream <- lm(mass_g ~ length_cm, data = bream)
```

```
Call:
lm(formula = mass_g ~ length_cm, data = bream)

Coefficients:
(Intercept)    length_cm
   -1035.35         54.55
```

```
mdl_pike <- lm(mass_g ~ length_cm, data = pike)
```

```
Call:
lm(formula = mass_g ~ length_cm, data = pike)

Coefficients:
(Intercept)    length_cm
   -1540.82         53.19
```

```
mdl_perch <- lm(mass_g ~ length_cm, data = perch)
```

```
Call:
lm(formula = mass_g ~ length_cm, data = perch)

Coefficients:
(Intercept)    length_cm
   -619.18         38.91
```

```
mdl_roach <- lm(mass_g ~ length_cm, data = roach)
```

```
Call:
lm(formula = mass_g ~ length_cm, data = roach)

Coefficients:
(Intercept)    length_cm
   -329.38         23.32
```

# Explanatory data

```
explanatory_data <- tibble(  
  length_cm = seq(5, 60, 5)  
)
```

# Making predictions

```
prediction_data_bream <- explanatory_data %>%  
  mutate(  
    mass_g = predict mdl_bream, explanatory_data),  
    species = "Bream"  
  )
```

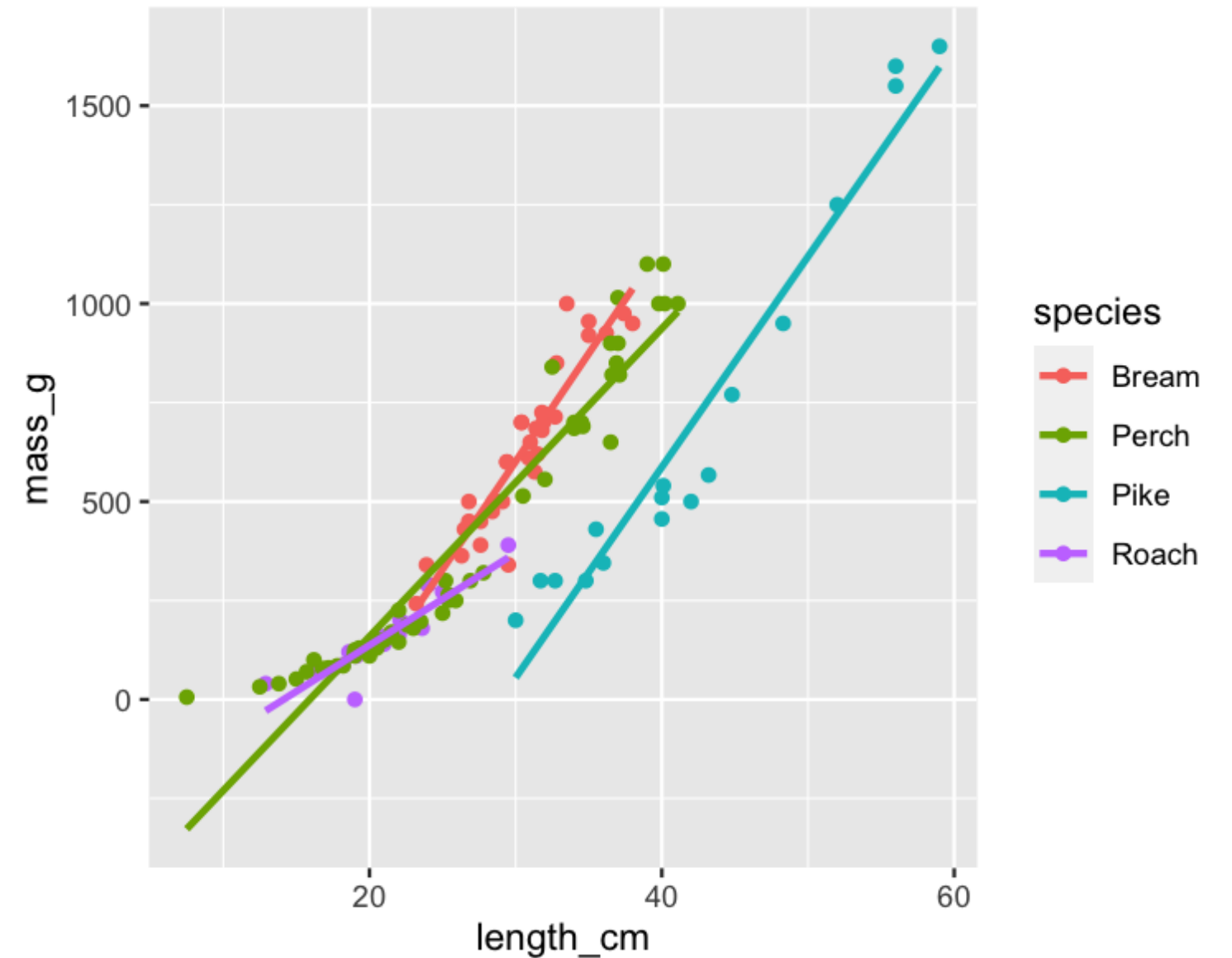
```
prediction_data_pike <- explanatory_data %>%  
  mutate(  
    mass_g = predict mdl_perch, explanatory_data),  
    species = "Perch"  
  )
```

```
prediction_data_perch <- explanatory_data %>%  
  mutate(  
    mass_g = predict mdl_pike, explanatory_data),  
    species = "Pike"  
  )
```

```
prediction_data_roach <- explanatory_data %>%  
  mutate(  
    mass_g = predict mdl_roach, explanatory_data),  
    species = "Roach"  
  )
```

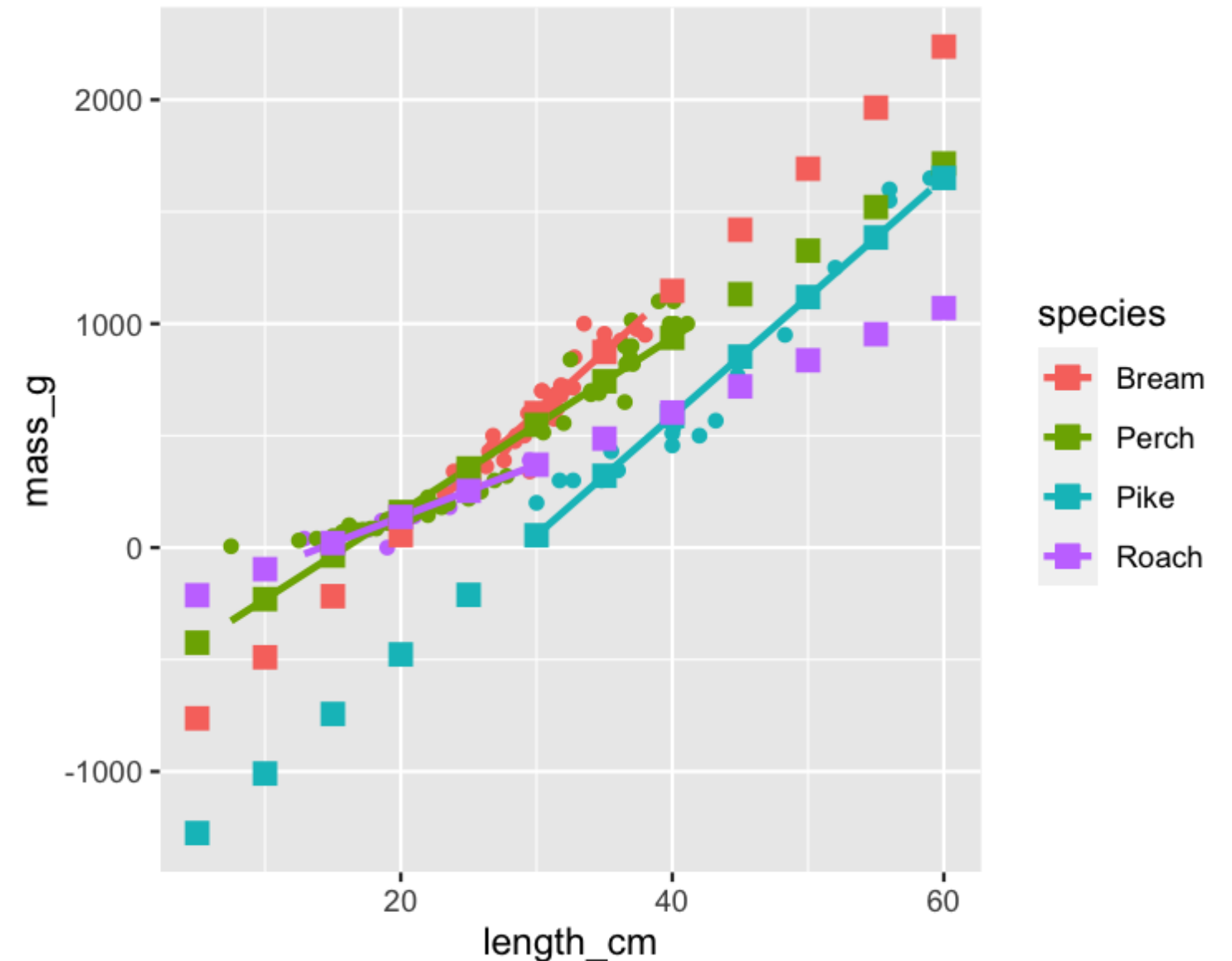
# Visualizing predictions

```
ggplot(fish, aes(length_cm, mass_g, color = species)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```



# Adding in your predictions

```
ggplot(fish,aes(length_cm, mass_g, color = species)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE) +  
  geom_point(data = prediction_data_bream, size = 3, shape = 15) +  
  geom_point(data = prediction_data_perch, size = 3, shape = 15) +  
  geom_point(data = prediction_data_pike, size = 3, shape = 15) +  
  geom_point(data = prediction_data_roach, size = 3, shape = 15)
```





# Coefficient of determination

```
mdl_fish <- lm(mass_g ~ length_cm + species, data = fish)
```

```
mdl_fish %>%  
  glance() %>%  
  pull(adj.r.squared)
```

0.917

```
mdl_bream %>% glance() %>% pull(adj.r.squared)
```

0.874

```
mdl_perch %>% glance() %>% pull(adj.r.squared)
```

0.917

```
mdl_pike %>% glance() %>% pull(adj.r.squared)
```

0.941

```
mdl_roach %>% glance() %>% pull(adj.r.squared)
```

0.815

# Residual standard error

```
mdl_fish %>%  
  glance() %>%  
  pull(sigma)
```

103

```
mdl_bream %>% glance() %>% pull(sigma)
```

74.2

```
mdl_perch %>% glance() %>% pull(sigma)
```

100

```
mdl_pike %>% glance() %>% pull(sigma)
```

120

```
mdl_roach %>% glance() %>% pull(sigma)
```

38.2

# Let's practice!

INTERMEDIATE REGRESSION IN R

# One model with an interaction

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Data Evangelist at DataCamp

# What is an interaction?

## In the fish dataset

The effect of length on the expected mass is different for different species.

## More generally

The effect of one explanatory variable on the expected response changes depending on the value of another explanatory variable.

# Specifying interactions

## No interactions

```
response ~ explntry1 + explntry2
```

## With interactions (implicit)

```
response_var ~ explntry1 * explntry2
```

## With interactions (explicit)

```
response ~ explntry1 + explntry2 + explntry1:explntry2
```

## No interactions

```
mass_g ~ length_cm + species
```

## With interactions (implicit)

```
mass_g ~ length_cm * species
```

## With interactions (explicit)

```
mass_g ~ length_cm + species + length_cm:species
```

# Running the model

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

Call:

```
lm(formula = mass_g ~ length_cm * species, data = fish)
```

Coefficients:

(Intercept)	length_cm	speciesPerch	speciesPike
-1035.348	54.550	416.172	-505.477
speciesRoach	length_cm:speciesPerch	length_cm:speciesPike	length_cm:speciesRoach
705.971	-15.639	-1.355	-31.231

# Easier to understand coefficients

```
mdl_inter <- lm(mass_g ~ species + species:length_cm + 0, data = fish)
```

Call:

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

Coefficients:

speciesBream	speciesPerch	speciesPike	speciesRoach
-1035.35	-619.18	-1540.82	-329.38
speciesBream:length_cm	speciesPerch:length_cm	speciesPike:length_cm	speciesRoach:length_cm
54.55	38.91	53.19	23.32



# Familiar numbers

```
      speciesBream      speciesPerch      speciesPike      speciesRoach
      -1035.35      -619.18      -1540.82      -329.38
speciesBream:length_cm speciesPerch:length_cm speciesPike:length_cm speciesRoach:length_cm
      54.55      38.91      53.19      23.32
```

```
coefficients mdl_bream)
```

```
(Intercept) length_cm
-1035.34757    54.54998
```

# Let's practice!

INTERMEDIATE REGRESSION IN R

# Making predictions with interactions

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Data Evangelist at DataCamp

# The model with the interaction

```
mdl_mass_vs_both_inter <- lm(mass_g ~ species + species:length_cm + 0, data = fish)
```

Call:

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

Coefficients:

speciesBream	speciesPerch	speciesPike	speciesRoach
-1035.35	-619.18	-1540.82	-329.38
speciesBream:length_cm	speciesPerch:length_cm	speciesPike:length_cm	speciesRoach:length_cm
54.55	38.91	53.19	23.32

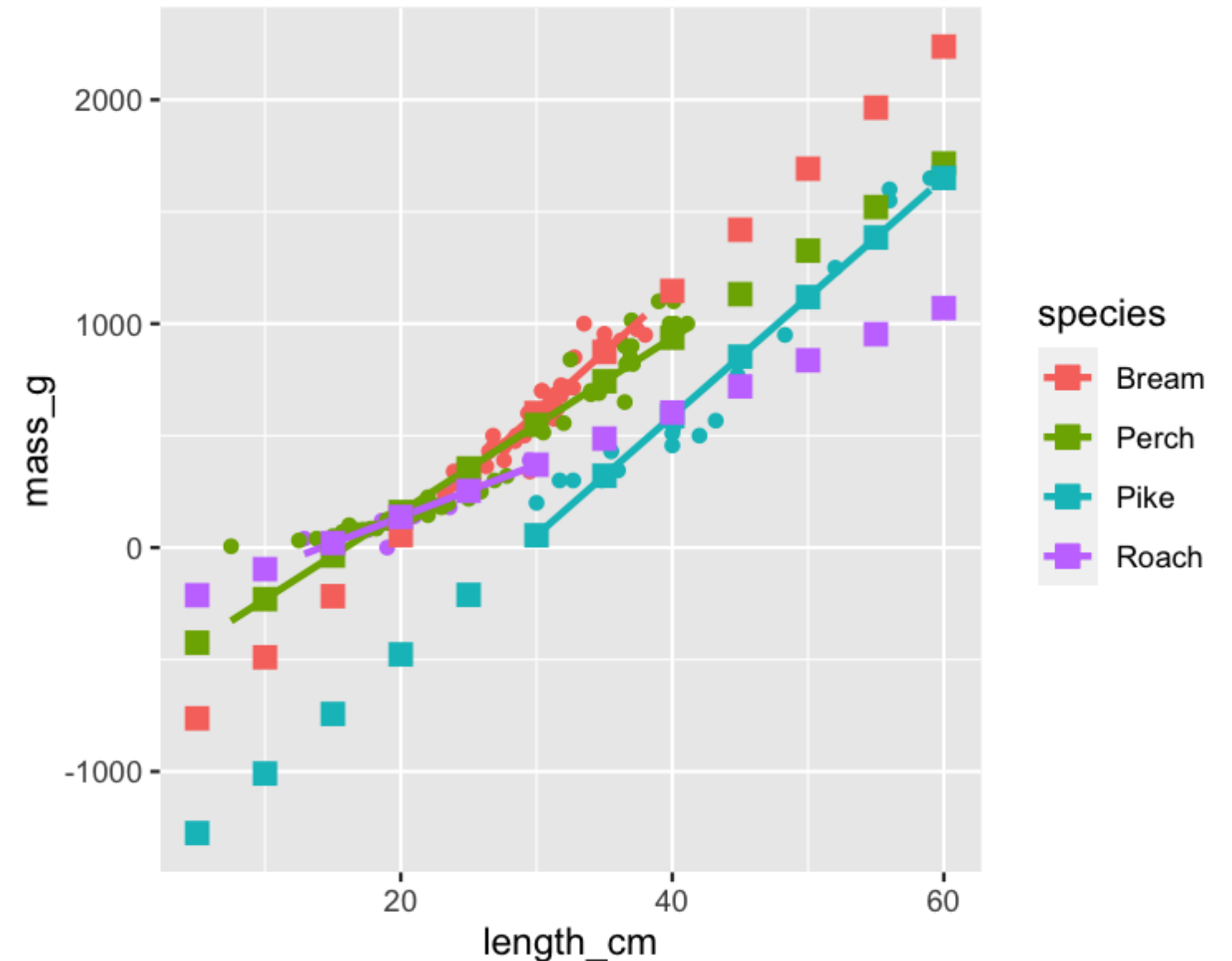
# The prediction flow, again

```
library(dplyr)
library(tidyr)
explanatory_data <- expand_grid(
  length_cm = seq(5, 60, 5),
  species = unique(fish$species)
)

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

# Visualizing the predictions

```
ggplot(fish, aes(length_cm, mass_g, color = species)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE) +  
  geom_point(data = prediction_data, size = 3, shape = 15)
```



# Manually calculating the predictions

```
coeffs <- coefficients mdl_mass_vs_both_inter)
```

speciesBream	speciesPerch	speciesPike	speciesRoach
-1035.34757	-619.17511	-1540.82427	-329.37621
speciesBream:length_cm	speciesPerch:length_cm	speciesPike:length_cm	speciesRoach:length_cm
54.54998	38.91147	53.19487	23.31926

```
intercept_bream <- coeffs[1]  
intercept_perch <- coeffs[2]  
intercept_pike <- coeffs[3]  
intercept_roach <- coeffs[4]
```

```
slope_bream <- coeffs[5]  
slope_perch <- coeffs[6]  
slope_pike <- coeffs[7]  
slope_roach <- coeffs[8]
```

# Manually calculating the predictions

```
explanatory_data %>%  
  mutate(  
    mass_g = case_when(  
  
    )  
  )
```



# Manually calculating the predictions

```
explanatory_data %>%  
  mutate(  
    mass_g = case_when(  
      species == "Bream" ~  
  
    )  
  )
```

# Manually calculating the predictions

```
explanatory_data %>%  
  mutate(  
    mass_g = case_when(  
      species == "Bream" ~ intercept_bream + slope_bream * length_cm  
    )  
  )
```

# Manually calculating the predictions

```
explanatory_data %>%  
  mutate(  
    mass_g = case_when(  
      species == "Bream" ~ intercept_bream + slope_bream * length_cm,  
      species == "Perch" ~ intercept_perch + slope_perch * length_cm,  
      species == "Pike" ~ intercept_pike + slope_pike * length_cm,  
      species == "Roach" ~ intercept_roach + slope_roach * length_cm  
    )  
  )
```

# Let's practice!

INTERMEDIATE REGRESSION IN R

# Simpson's Paradox

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Data Evangelist at DataCamp

# A most ingenious paradox!

Simpson's Paradox occurs when the trend of a model on the whole dataset is very different from the trends shown by models on subsets of the dataset.

*trend = slope coefficient*

# Synthetic Simpson data

x	y	group
62.24344	70.60840	D
52.33499	14.70577	B
56.36795	46.39554	C
66.80395	66.17487	D
66.53605	89.24658	E
62.38129	91.45260	E

- 5 groups of data, labeled "A" to "E"

<sup>1</sup> [https://www.rdocumentation.org/packages/datasauRus/topics/simpsons\\_paradox](https://www.rdocumentation.org/packages/datasauRus/topics/simpsons_paradox)

# Linear regressions

## Whole dataset

```
mdl_whole <- lm(  
  y ~ x,  
  data = simpsons_paradox  
)  
coefficients(mdl_whole)
```

(Intercept)	x
-38.554	1.751

## By group

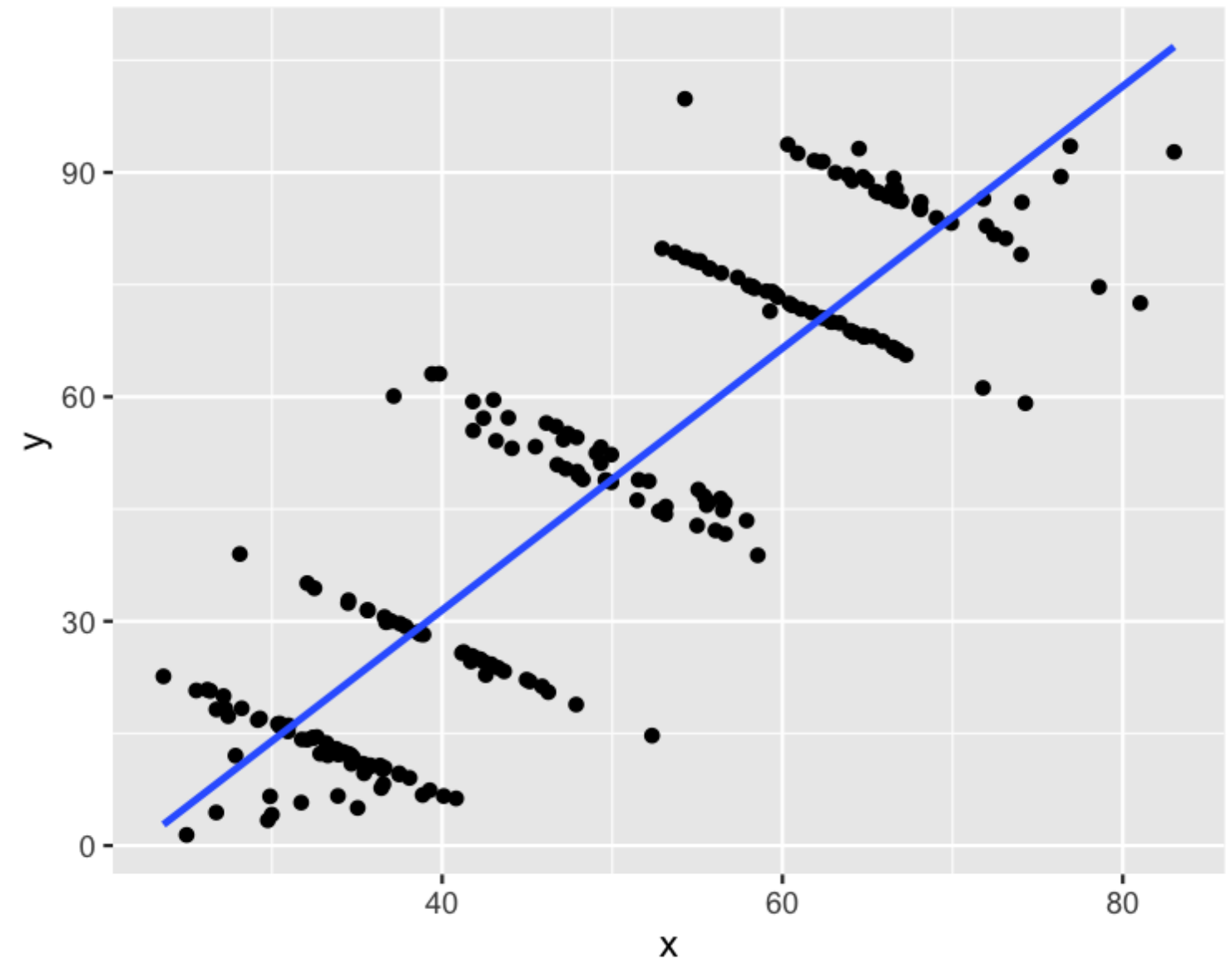
```
mdl_by_group <- lm(  
  y ~ group + group:x + 0,  
  data = simpsons_paradox  
)  
coefficients(mdl_by_group)
```

groupA	groupB	groupC	groupD	groupE
32.5051	67.3886	99.6333	132.3932	123.8242
groupA:x	groupB:x	groupC:x	groupD:x	groupE:x
-0.6266	-1.0105	-0.9940	-0.9908	-0.5364



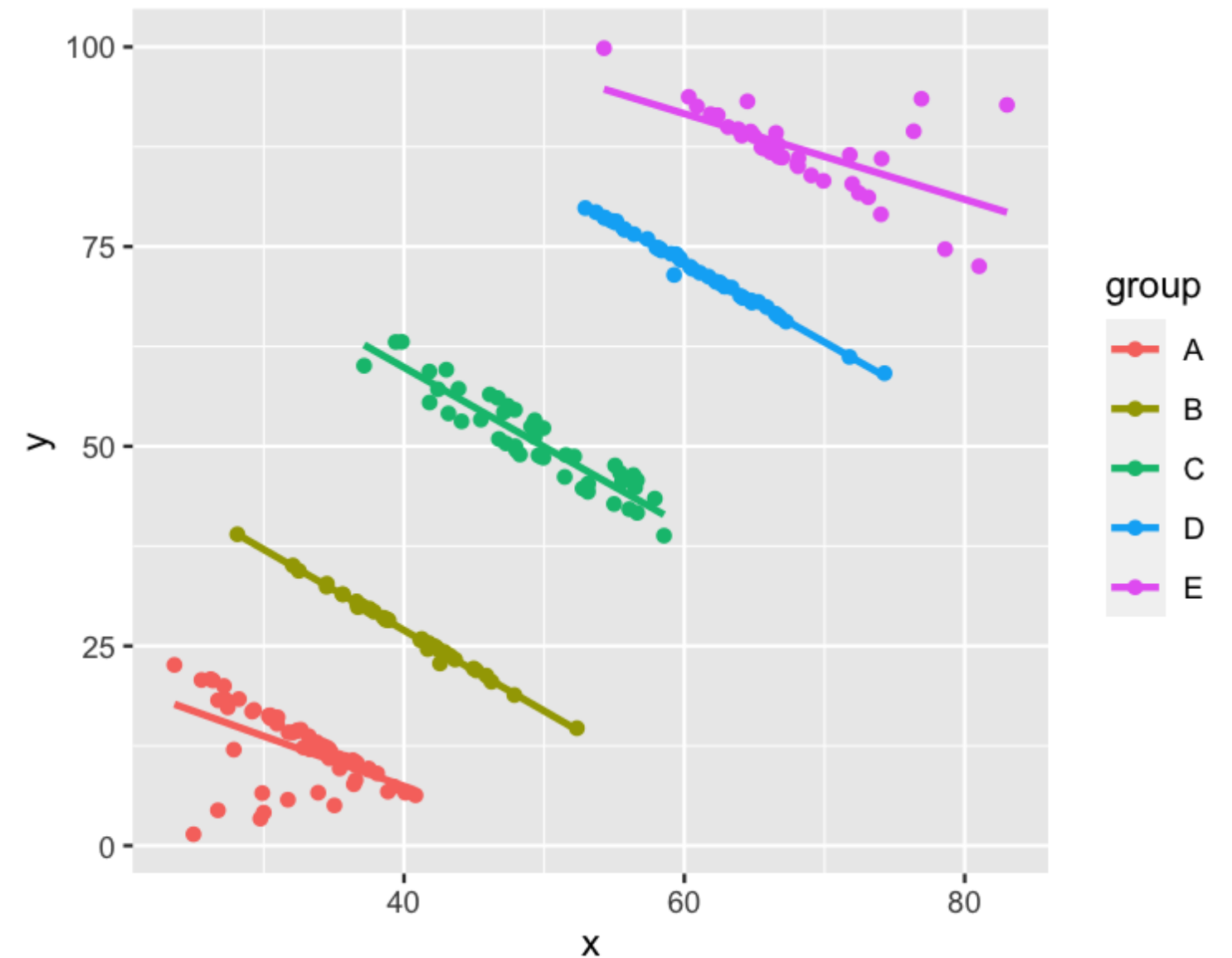
# Plotting the whole dataset

```
ggplot(simpsons_paradox, aes(x, y)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```



# Plotting by group

```
ggplot(simpsons_paradox, aes(x, y, color = group)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```



# Reconciling the difference

## Good advice

If possible, try to plot the dataset.

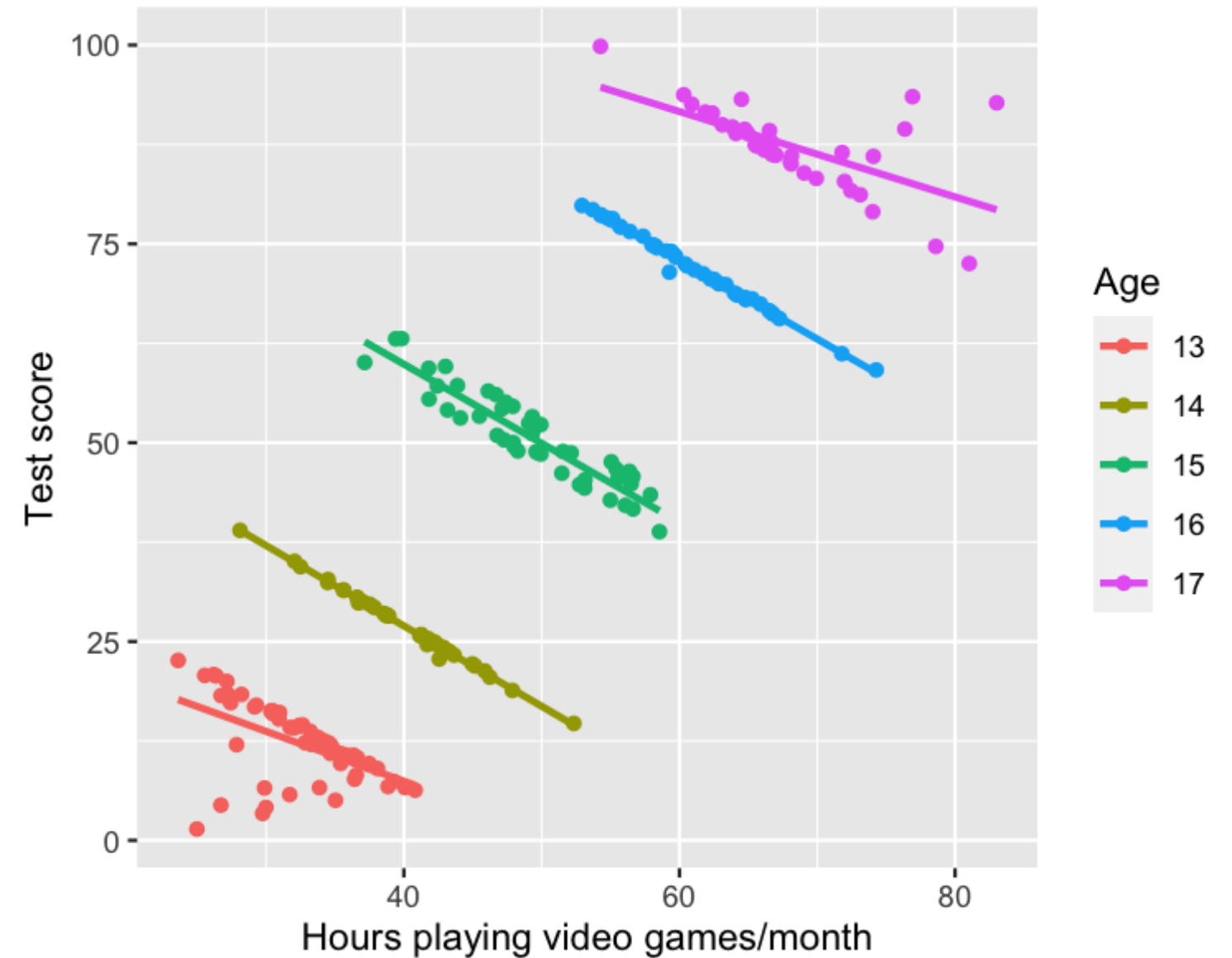
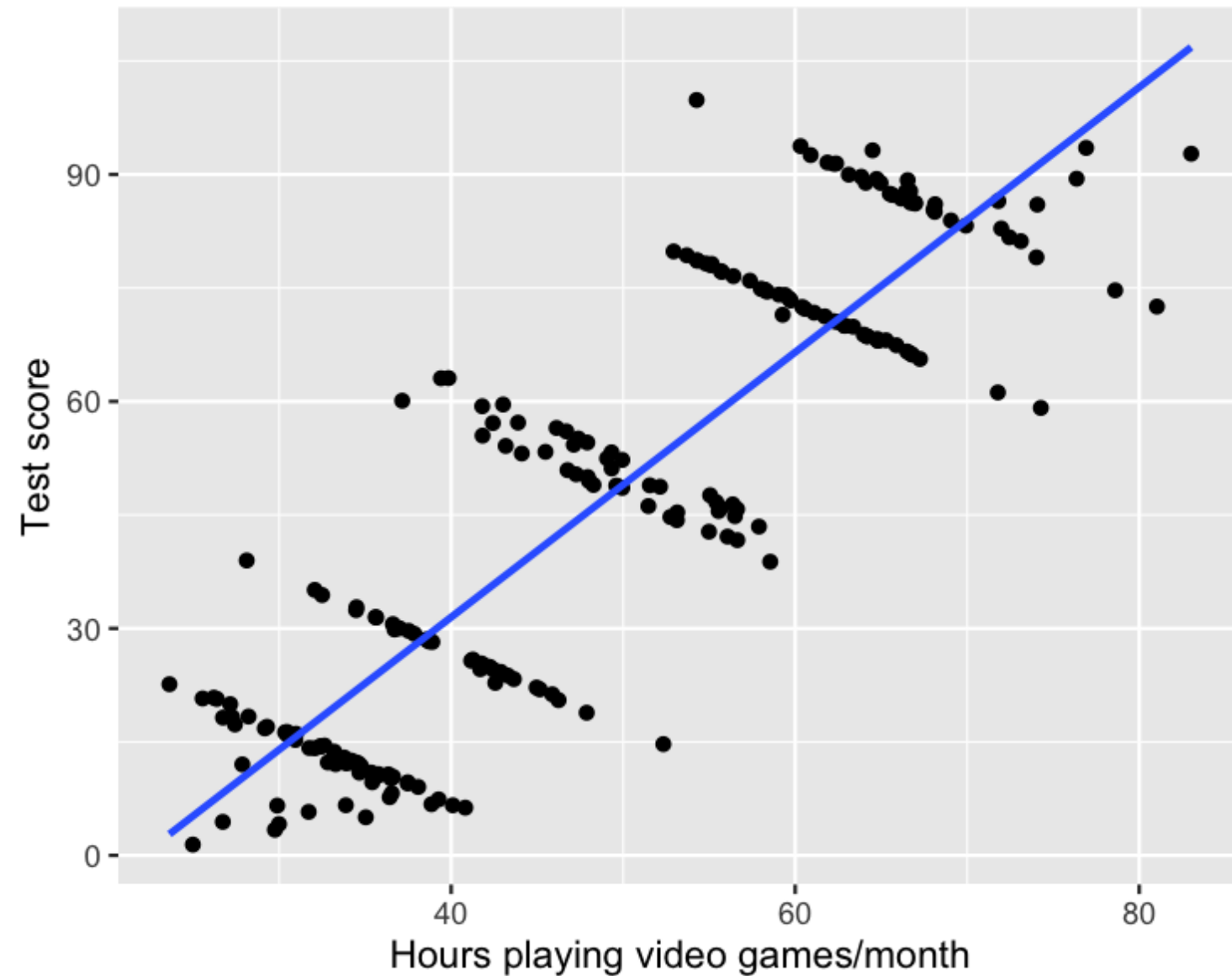
## Common advice

You can't choose the best model in general—it depends on the dataset and the question you are trying to answer.

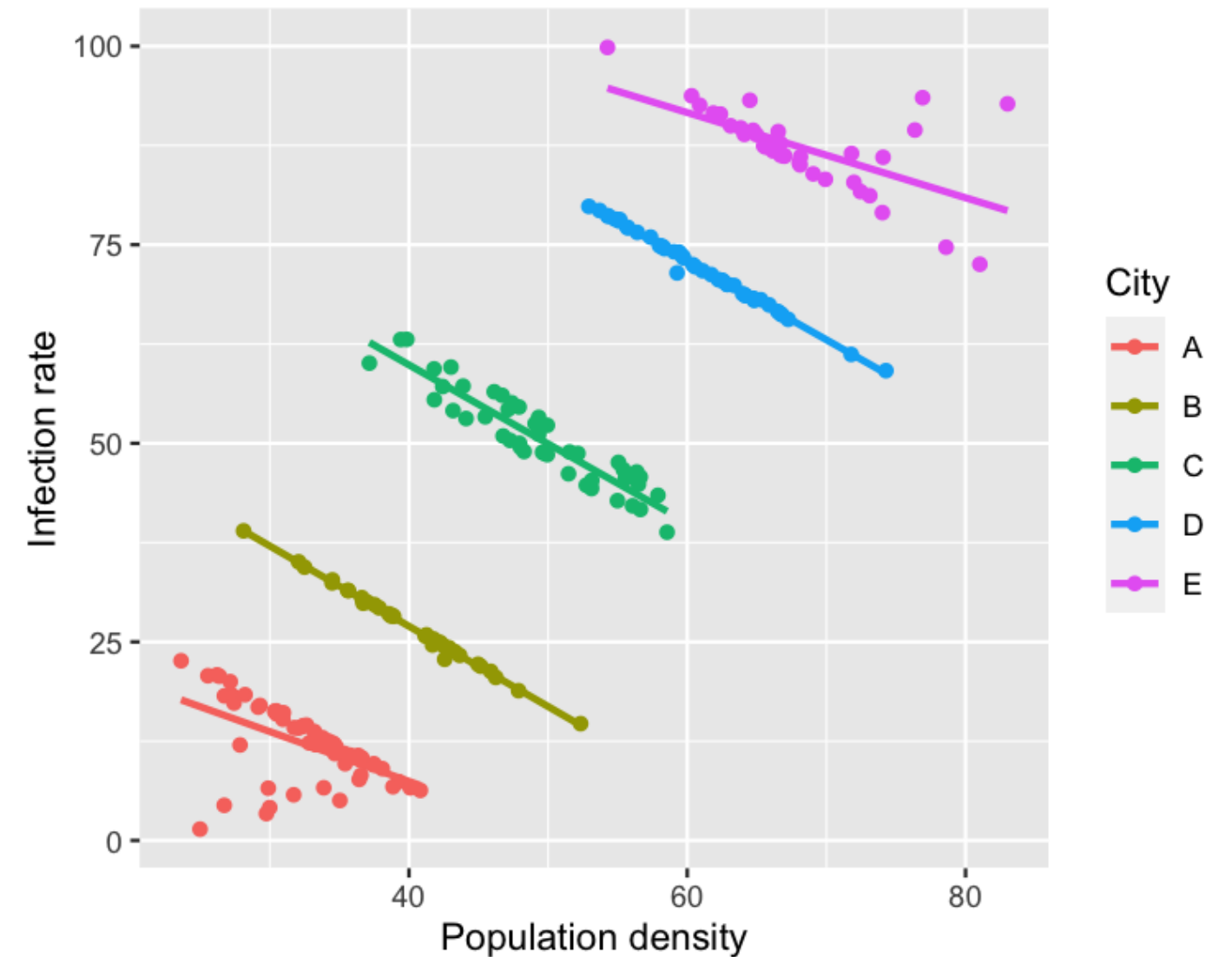
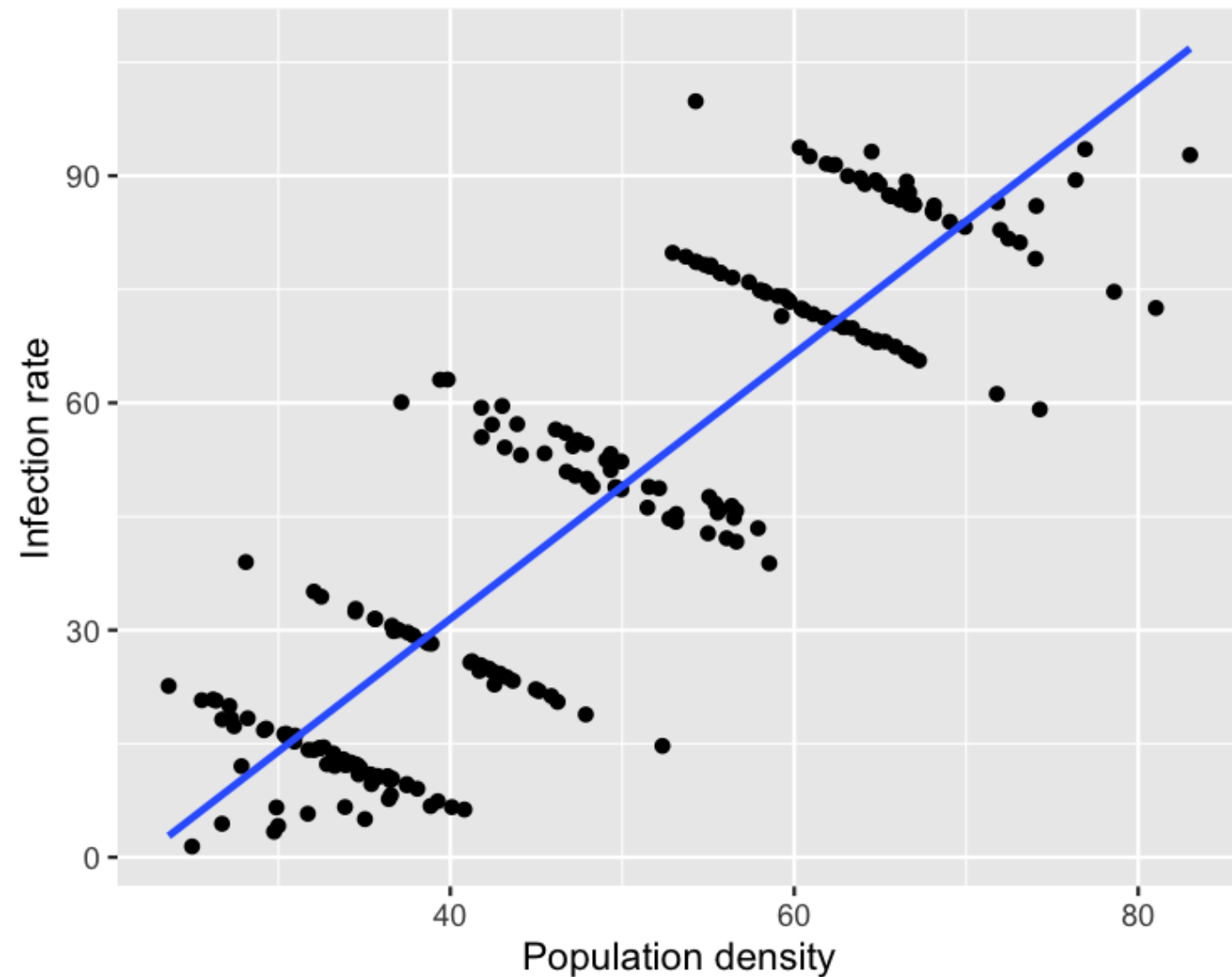
## More good advice

Articulate a question before you start modeling.

# Test score example



# Infectious disease example



<sup>1</sup> <https://stats.stackexchange.com/questions/478463/examples-of-simpsons-paradox-being-resolved-by-choosing-the-aggregate-data>

# Reconciling the difference, again

- Usually (but not always) the grouped model contains more insight.
- Are you missing explanatory variables?
- Context is important.

# Simpson's paradox in real datasets

- The paradox is usually less obvious.
- You may see a zero slope rather than a complete change in direction.
- It may not appear in every group.

# Let's practice!

INTERMEDIATE REGRESSION IN R