Models for each category

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

Maarten Van den Broeck
Content Developer at DataCamp

Four categories

```
print(fish["species"].unique())
```

```
array(['Bream', 'Roach', 'Perch', 'Pike'], dtype=object)
```

Splitting the dataset

```
bream = fish[fish["species"] == "Bream"]
perch = fish[fish["species"] == "Perch"]
pike = fish[fish["species"] == "Pike"]
roach = fish[fish["species"] == "Roach"]
```

Four models

```
mdl_bream = ols("mass_g ~ length_cm", data=bream).fit()
print(mdl_bream.params)
```

```
Intercept   -1035.3476
length_cm      54.5500
```

```
mdl_perch = ols("mass_g ~ length_cm", data=perch).fit()
print(mdl_perch.params)
```

```
Intercept   -619.1751
length_cm     38.9115
```

```
mdl_pike = ols("mass_g ~ length_cm", data=pike).fit()
print(mdl_pike.params)
```

```
Intercept   -1540.8243
length_cm      53.1949
```

```
mdl_roach = ols("mass_g ~ length_cm", data=roach).fit()
print(mdl_roach.params)
```

```
Intercept   -329.3762
length_cm     23.3193
```

Explanatory data

```python
explanatory_data = pd.DataFrame(
    {"length_cm": np.arange(5, 61, 5)})

print(explanatory_data)
```

```
    length_cm
0           5
1          10
2          15
3          20
4          25
5          30
6          35
7          40
8          45
9          50
10         55
11         60
```

Making predictions

```python
prediction_data_bream = explanatory_data.assign(
    mass_g = mdl_bream.predict(explanatory_data),
    species = "Bream")
```

```python
prediction_data_perch = explanatory_data.assign(
    mass_g = mdl_perch.predict(explanatory_data),
    species = "Perch")
```

```python
prediction_data_pike = explanatory_data.assign(
    mass_g = mdl_pike.predict(explanatory_data),
    species = "Pike")
```

```python
prediction_data_roach = explanatory_data.assign(
    mass_g = mdl_roach.predict(explanatory_data),
    species = "Roach")
```

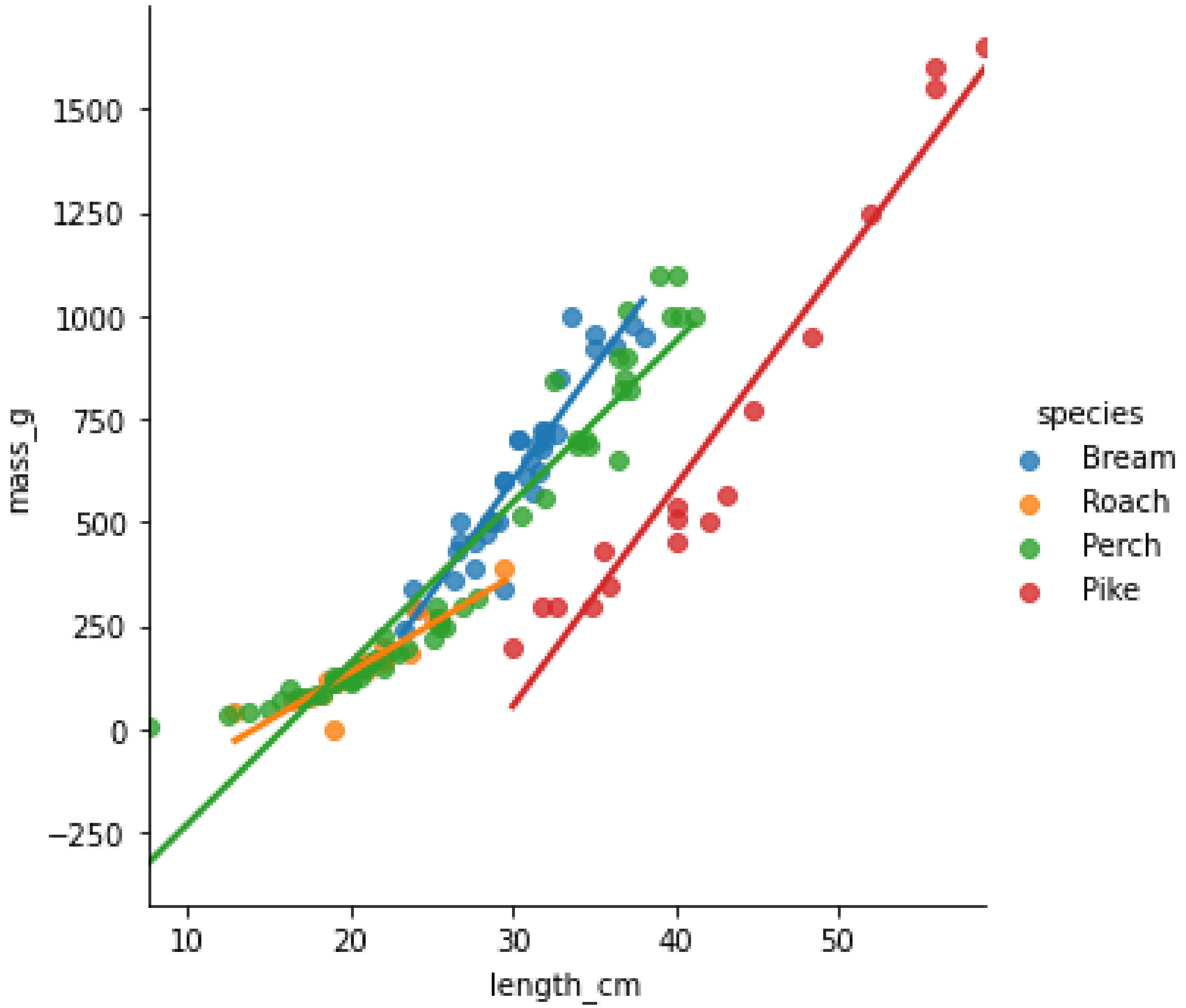
Concatenating predictions

```
prediction_data = pd.concat([prediction_data_bream,
                             prediction_data_roach,
                             prediction_data_perch,
                             prediction_data_pike])
```

```
    length_cm      mass_g species
0           5 -762.597660   Bream
1          10 -489.847756   Bream
2          15 -217.097851   Bream
3          20   55.652054   Bream
4          25  328.401958   Bream
5          30  601.151863   Bream
...
3          20 -476.926955    Pike
4          25 -210.952626    Pike
5          30   55.021703    Pike
6          35  320.996032    Pike
7          40  586.970362    Pike
8          45  852.944691    Pike
9          50 1118.919020    Pike
10         55 1384.893349    Pike
11         60 1650.867679    Pike
```

Visualizing predictions

```
sns.lmplot(x="length_cm",
           y="mass_g",
           data=fish,
           hue="species",
           ci=None)
plt.show()
```

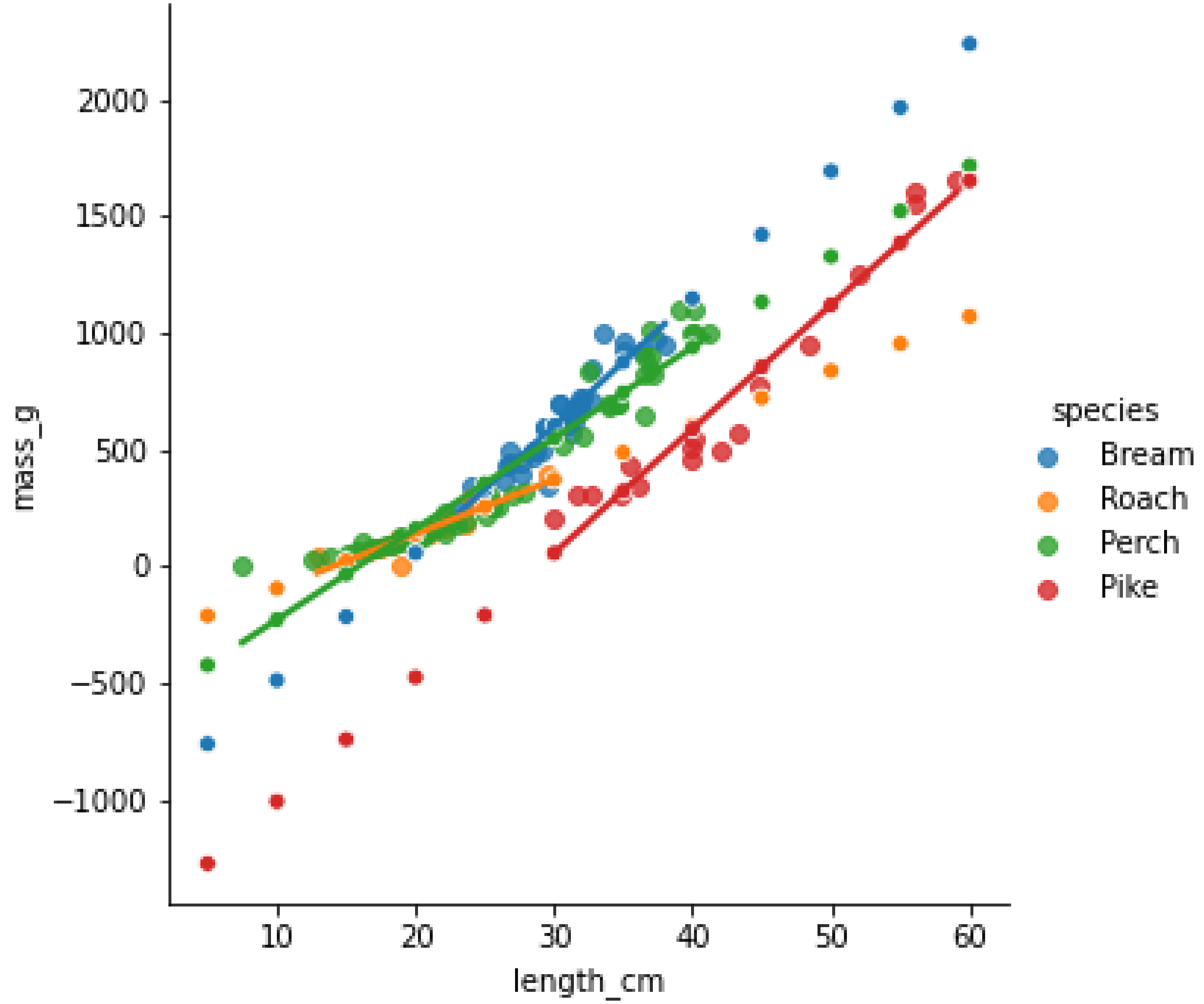


Adding in your predictions

```
sns.lmplot(x="length_cm",
           y="mass_g",
           data=fish,
           hue="species",
           ci=None)

sns.scatterplot(x="length_cm",
                y="mass_g",
                data=prediction_data,
                hue="species",
                ci=None,
                legend=False)

plt.show()
```



Coefficient of determination

```
mdl_fish = ols("mass_g ~ length_cm + species",
               data=fish).fit()

print(mdl_fish.rsquared_adj)
```

```
0.917
```

```
print(mdl_bream.rsquared_adj)
```

```
0.874
```

```
print(mdl_perch.rsquared_adj)
```

```
0.917
```

```
print(mdl_pike.rsquared_adj)
```

```
0.941
```

```
print(mdl_roach.rsquared_adj)
```

```
0.815
```

Residual standard error

```
print(np.sqrt(mdl_fish.mse_resid))
```

```
103
```

```
print(np.sqrt(mdl_bream.mse_resid))
```

```
74.2
```

```
print(np.sqrt(mdl_perch.mse_resid))
```

```
100
```

```
print(np.sqrt(mdl_pike.mse_resid))
```

```
120
```

```
print(np.sqrt(mdl_roach.mse_resid))
```

```
38.2
```

Let's practice!

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

One model with an interaction

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

Maarten Van den Broeck
Content Developer at DataCamp

What is an interaction?

In the fish dataset

- Different fish species have different mass to length ratios.
- 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

```
mdl_mass_vs_both = ols("mass_g ~ length_cm * species", data=fish).fit()


print(mdl_mass_vs_both.params)
```

```
Intercept                    -1035.3476
species[T.Perch]               416.1725
species[T.Pike]               -505.4767
species[T.Roach]               705.9714
length_cm                       54.5500
length_cm:species[T.Perch]     -15.6385
length_cm:species[T.Pike]       -1.3551
length_cm:species[T.Roach]     -31.2307
```

Easier to understand coefficients

```
mdl_mass_vs_both_inter = ols("mass_g ~ species + species:length_cm + 0", data=fish).fit()


print(mdl_mass_vs_both_inter.params)
```

```
species[Bream]             -1035.3476
species[Perch]              -619.1751
species[Pike]              -1540.8243
species[Roach]              -329.3762
species[Bream]:length_cm      54.5500
species[Perch]:length_cm      38.9115
species[Pike]:length_cm       53.1949
species[Roach]:length_cm      23.3193
```

Familiar numbers

```
print(mdl_mass_vs_both_inter.params)
```

```
species[Bream]             -1035.3476
species[Perch]              -619.1751
species[Pike]              -1540.8243
species[Roach]              -329.3762
species[Bream]:length_cm      54.5500
species[Perch]:length_cm      38.9115
species[Pike]:length_cm       53.1949
species[Roach]:length_cm      23.3193
```

```
print(mdl_bream.params)
```

```
Intercept   -1035.3476
length_cm      54.5500
```

Let's practice!

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

Making predictions with interactions

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

Maarten Van den Broeck
Content Developer at DataCamp

The model with the interaction

```
mdl_mass_vs_both_inter = ols("mass_g ~ species + species:length_cm + 0",
                             data=fish).fit()


print(mdl_mass_vs_both_inter.params)
```

```
species[Bream]                -1035.3476
species[Perch]                 -619.1751
species[Pike]                 -1540.8243
species[Roach]                 -329.3762
species[Bream]:length_cm         54.5500
species[Perch]:length_cm         38.9115
species[Pike]:length_cm          53.1949
species[Roach]:length_cm         23.3193
```

The prediction flow

```
from itertools import product

length_cm = np.arange(5, 61, 5)

species = fish["species"].unique()

p = product(length_cm, species)
```

```
explanatory_data = pd.DataFrame(p,
                                columns=["length_cm",
                                         "species"])


prediction_data = explanatory_data.assign(
 mass_g = mdl_mass_vs_both_inter.predict(explanatory_data))
```

```
print(prediction_data)
```

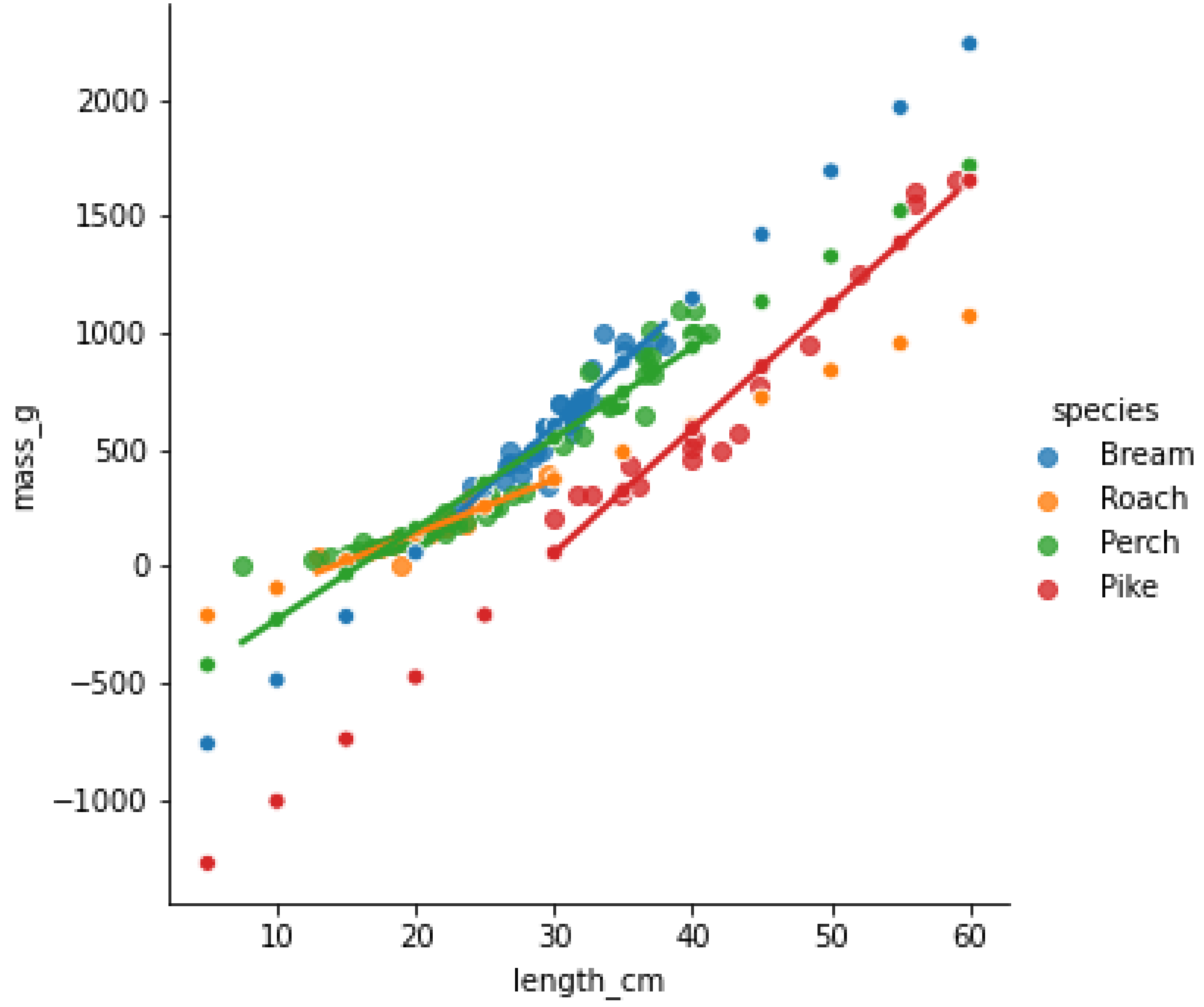
```
    length_cm species     mass_g
0           5   Bream  -762.5977
1           5   Roach  -212.7799
2           5   Perch  -424.6178
3           5    Pike -1274.8499
4          10   Bream  -489.8478
5          10   Roach   -96.1836
6          10   Perch  -230.0604
7          10    Pike -1008.8756
8          15   Bream  -217.0979
...
40         55   Bream  1964.9014
41         55   Roach   953.1833
42         55   Perch  1520.9556
43         55    Pike  1384.8933
44         60   Bream  2237.6513
45         60   Roach  1069.7796
46         60   Perch  1715.5129
47         60    Pike  1650.8677
```

Visualizing the predictions

```
sns.lmplot(x="length_cm",
           y="mass_g",
           data=fish,
           hue="species",
           ci=None)

sns.scatterplot(x="length_cm",
                y="mass_g",
                data=prediction_data,
                hue="species")

plt.show()
```



Manually calculating the predictions

```
coeffs = mdl_mass_vs_both_inter.params
```

```
species[Bream]             -1035.3476
species[Perch]              -619.1751
species[Pike]              -1540.8243
species[Roach]              -329.3762
species[Bream]:length_cm      54.5500
species[Perch]:length_cm      38.9115
species[Pike]:length_cm       53.1949
species[Roach]:length_cm      23.3193
```

```
ic_bream, ic_perch, ic_pike, ic_roach,
slope_bream, slope_perch, slope_pike, slope_roach = coeffs
```

Manually calculating the predictions

```
conditions = [
    explanatory_data["species"] == "Bream",
    explanatory_data["species"] == "Perch",
    explanatory_data["species"] == "Pike",
    explanatory_data["species"] == "Roach"
]

ic_choices = [ic_bream, ic_perch, ic_pike, ic_roach]
intercept = np.select(conditions, ic_choices)

slope_choices = [slope_bream, slope_perch, slope_pike, slope_roach]
slope = np.select(conditions, slope_choices)
```

Manually calculating the predictions

```
prediction_data = explanatory_data.assign(
 mass_g = intercept + slope * explanatory_data["length_cm"])

print(prediction_data)
```

```
    length_cm species     mass_g
0           5   Bream  -762.5977
1           5   Roach  -212.7799
2           5   Perch  -424.6178
3           5    Pike -1274.8499
4          10   Bream  -489.8478
5          10   Roach   -96.1836
...
43         55    Pike  1384.8933
44         60   Bream  2237.6513
45         60   Roach  1069.7796
46         60   Perch  1715.5129
47         60    Pike  1650.8677
```

```
prediction_data = explanatory_data.assign(
 mass_g = mdl_mass_vs_both_inter.predict(explanatory_data))

print(prediction_data)
```

```
    length_cm species     mass_g
0           5   Bream  -762.5977
1           5   Roach  -212.7799
2           5   Perch  -424.6178
3           5    Pike -1274.8499
4          10   Bream  -489.8478
5          10   Roach   -96.1836
...
43         55    Pike  1384.8933
44         60   Bream  2237.6513
45         60   Roach  1069.7796
46         60   Perch  1715.5129
47         60    Pike  1650.8677
```

Let's practice!

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

Simpson's Paradox

INTERMEDIATE REGRESSION WITH STATSMODELS IN PYTHON

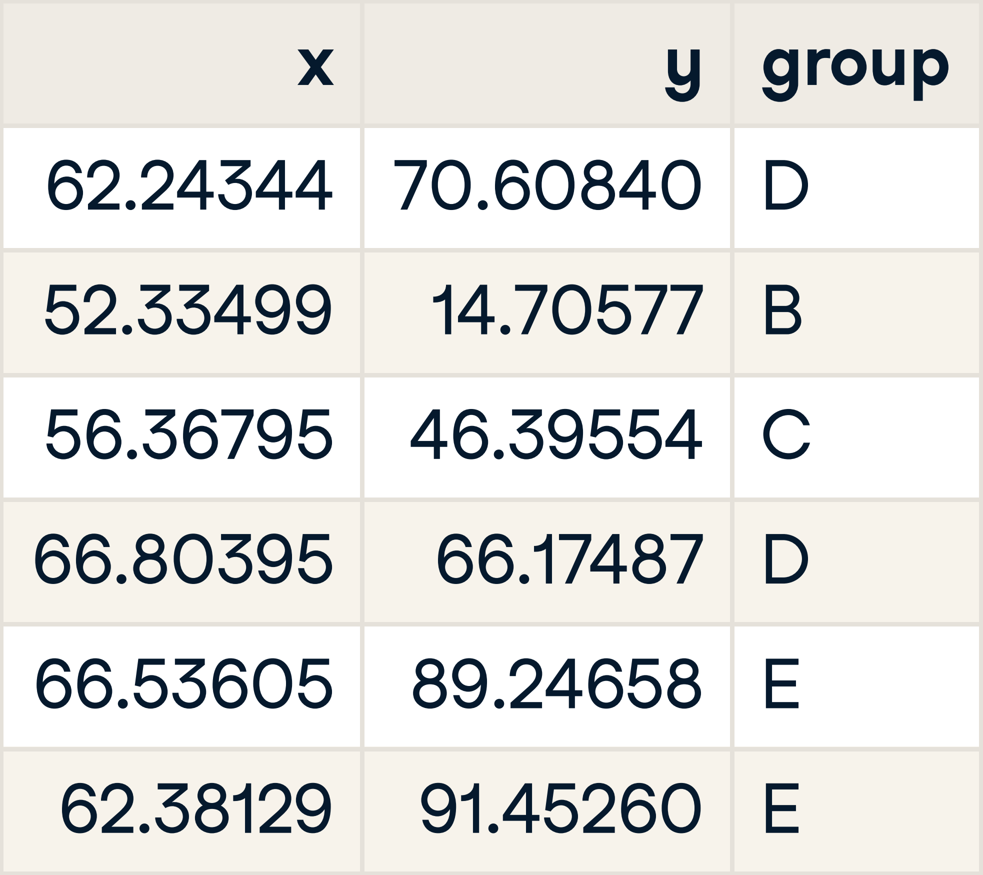
Maarten Van den Broeck
Content Developer at DataCamp

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"

[1] https://www.rdocumentation.org/packages/datasauRus/topics/simpsons_paradox

Linear regressions

Whole dataset

```
mdl_whole = ols("y ~ x",
                data=simpsons_paradox).fit()


print(mdl_whole.params)
```

```
Intercept           -38.554
x                     1.751
```

By group

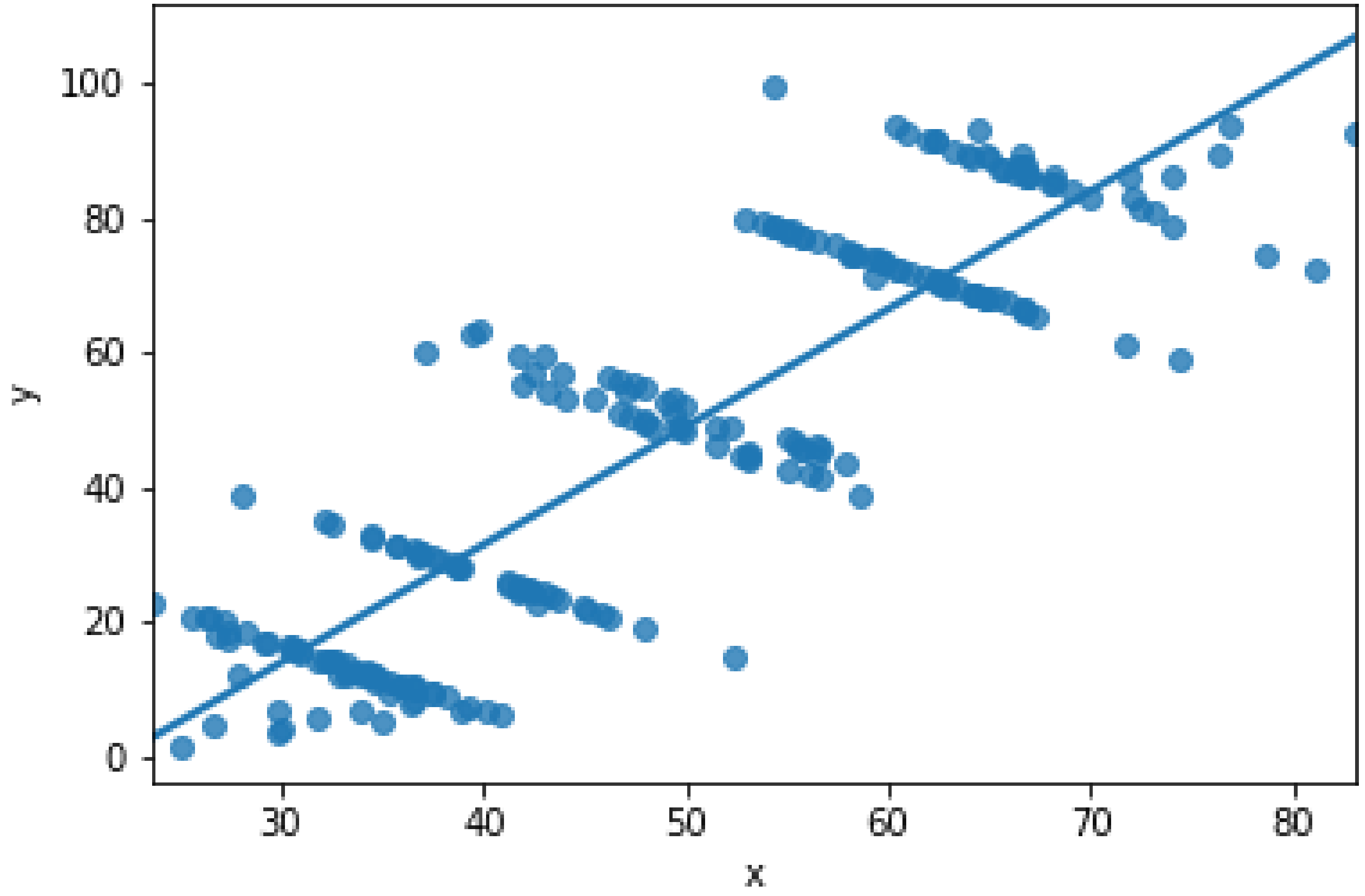
```
mdl_by_group = ols("y ~ group + group:x + 0",
                   data = simpsons_paradox).fit()


print(mdl_by_group.params)
```

```
  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
```

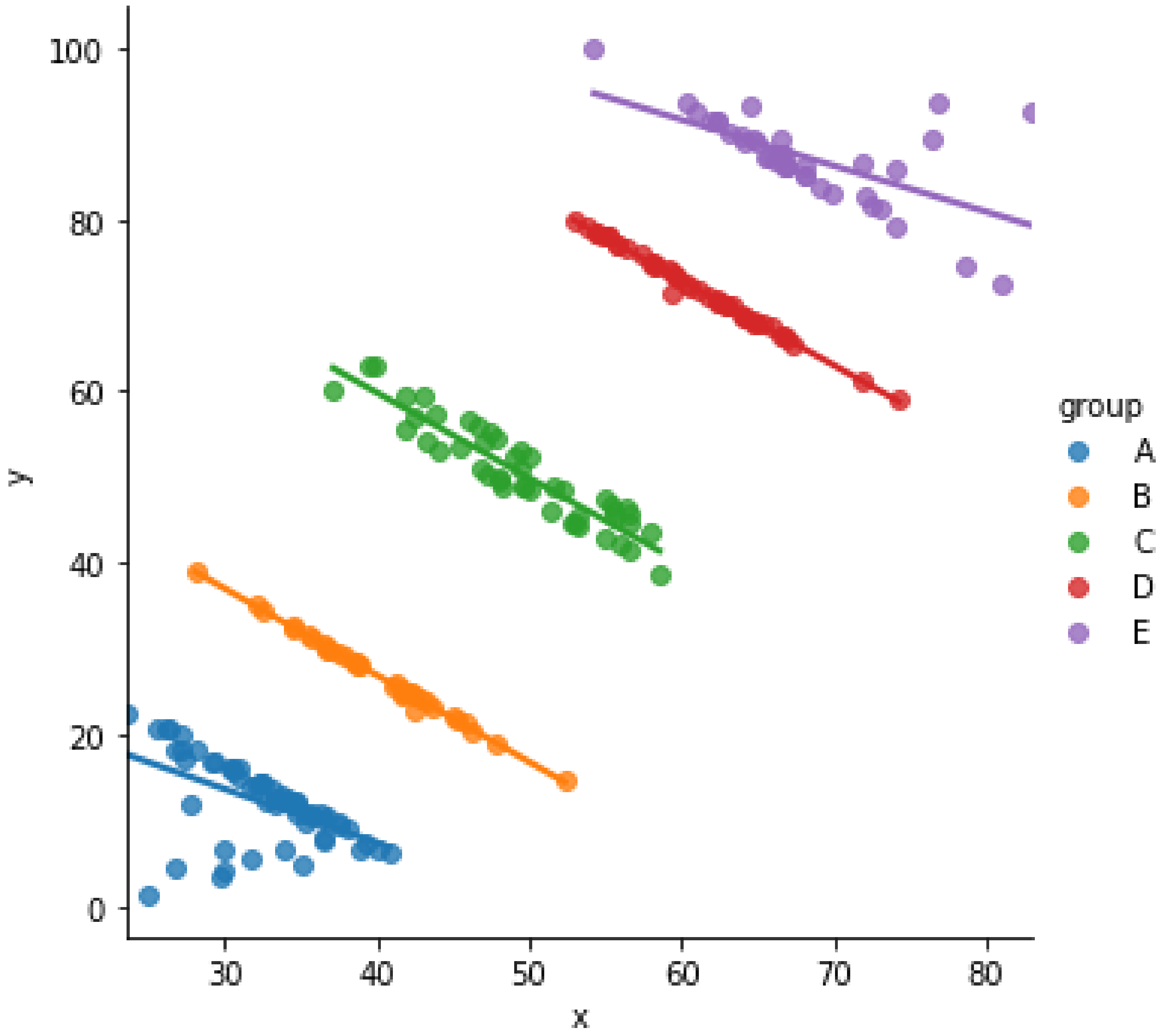
Plotting the whole dataset

```
sns.regplot(x="x",
            y="y",
            data=simpsons_paradox,
            ci=None)
```



Plotting by group

```
sns.lmplot(x="x",
           y="y",
           data=simpsons_paradox,
           hue="group",
           ci=None)
```



Reconciling the difference

Good advice

If possible, try to plot the dataset.

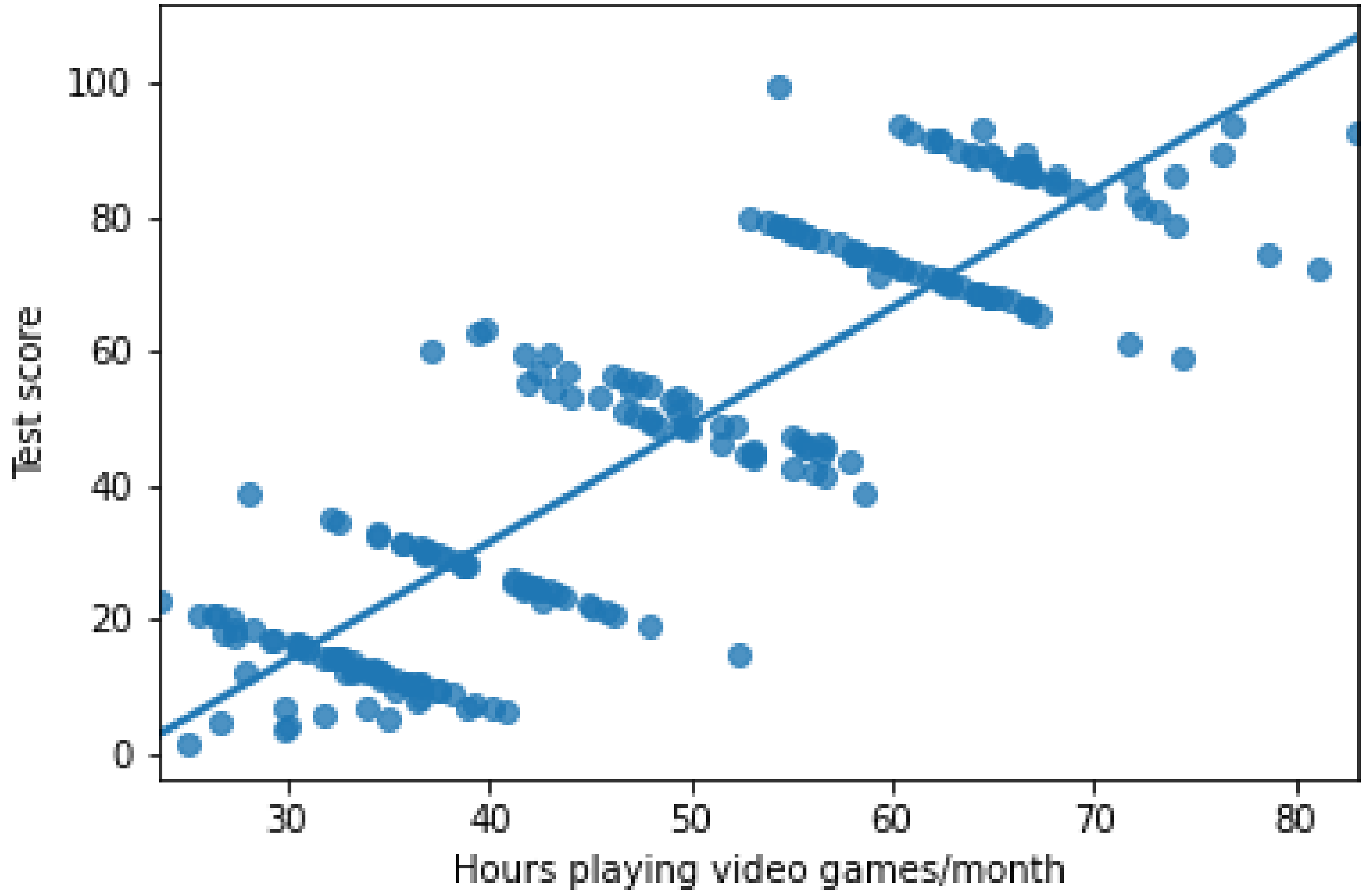
Common advice

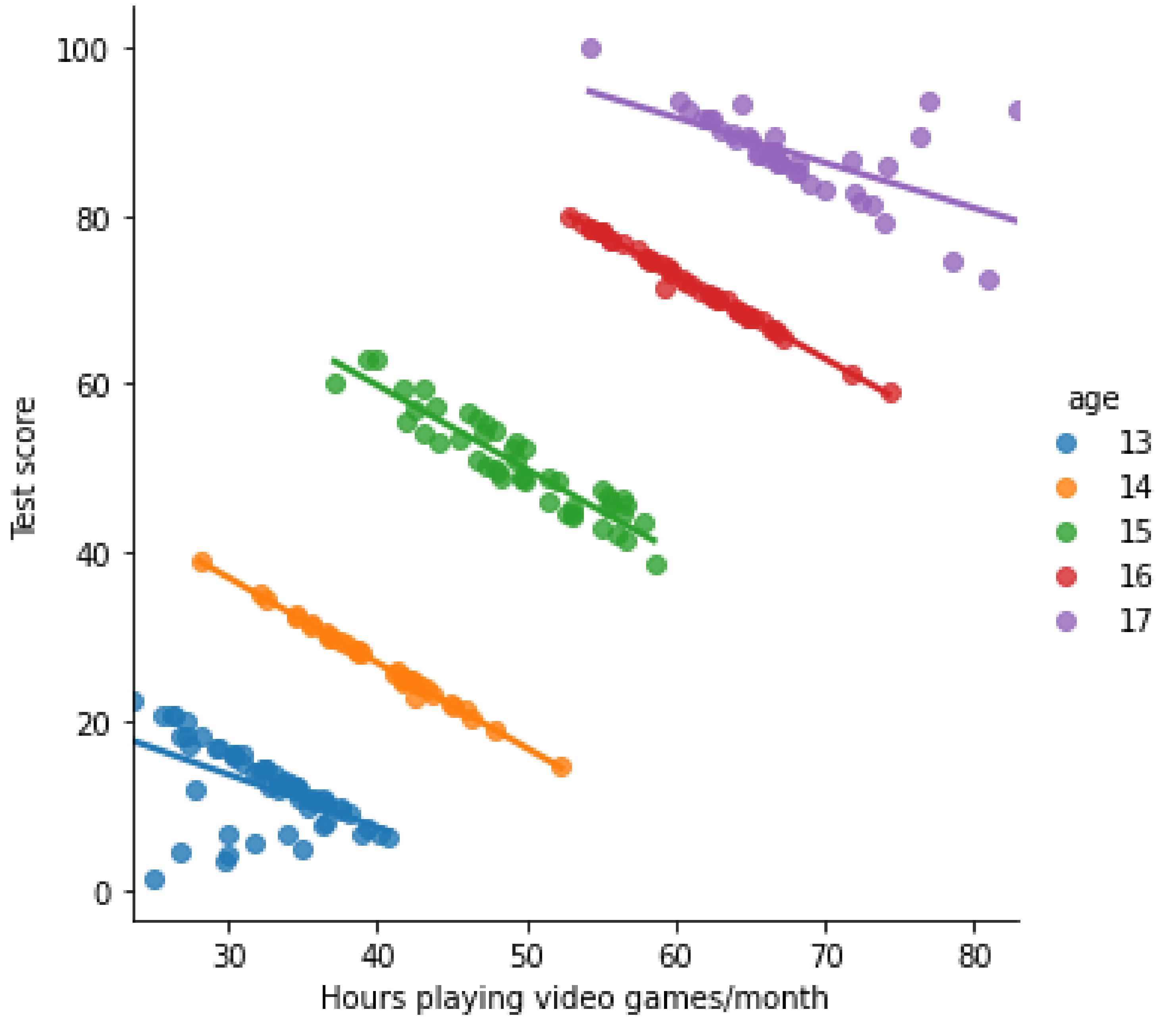
You can't choose the best model in general – it depends on the dataset and the question you are trying to answer.

More good advice

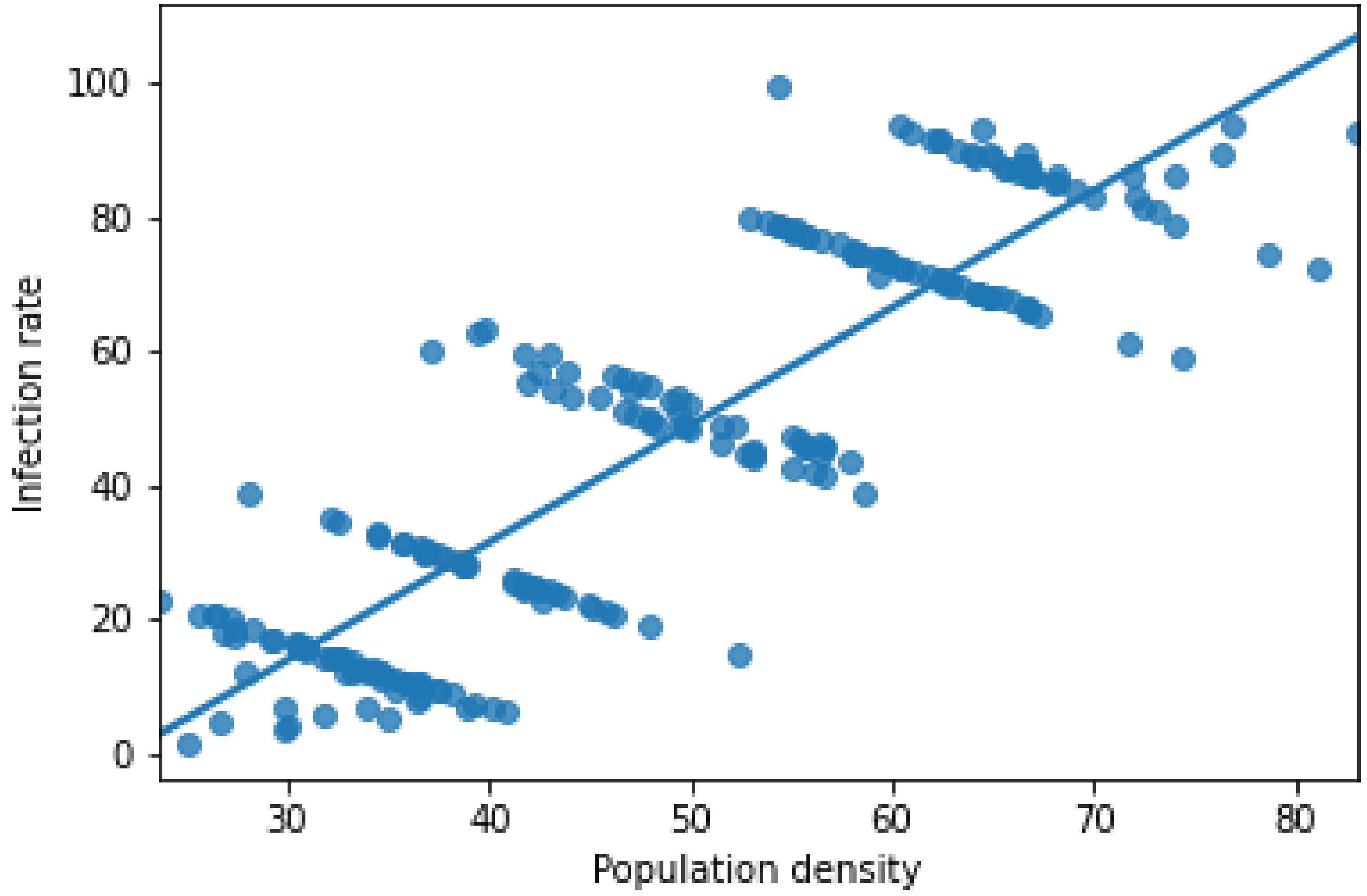
Articulate a question before you start modeling.

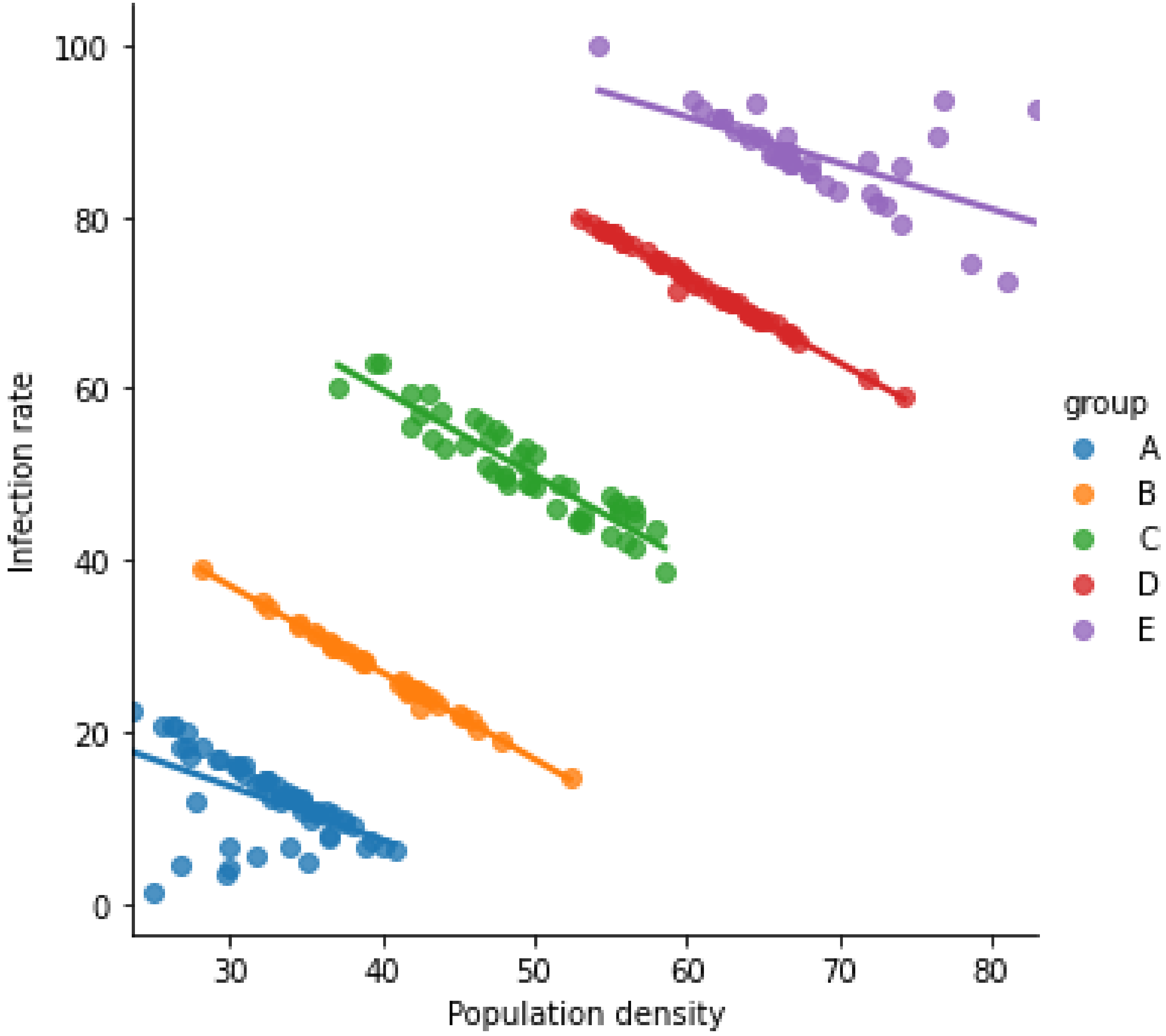
Test score example





Infectious disease example





Reconciling the difference

- 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 WITH STATSMODELS IN PYTHON