

Grain yields and unit conversion

INTRODUCTION TO WRITING FUNCTIONS IN R



Richie Cotton

Curriculum Architect at DataCamp

USDA NASS



Corn and wheat



- Soon these will be food
- Hmm, delicious

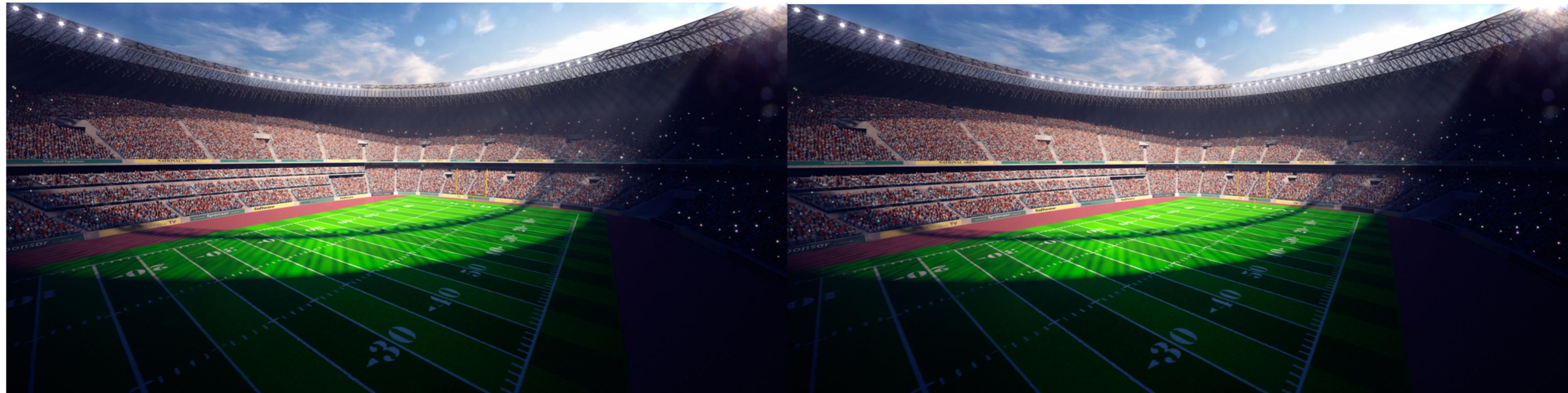
1 acre = area of land 2 oxen can plough in a day



Not the 100 Acre Wood



1 hectare = 2 football fields



1 hectare = 150 New York apartments



1 bushel = 2 baskets of peaches



1 kilogram = 1 squirrel monkey



magrittr's pipeable operator replacements

operator	functional alternative
<code>x * y</code>	<code>x %>% multiply_by(y)</code>
<code>x ^ y</code>	<code>x %>% raise_to_power(y)</code>
<code>x[y]</code>	<code>x %>% extract(y)</code>

Let's practice!

INTRODUCTION TO WRITING FUNCTIONS IN R

Visualizing grain yields

INTRODUCTION TO WRITING FUNCTIONS IN R



Richie Cotton

Curriculum Architect at DataCamp

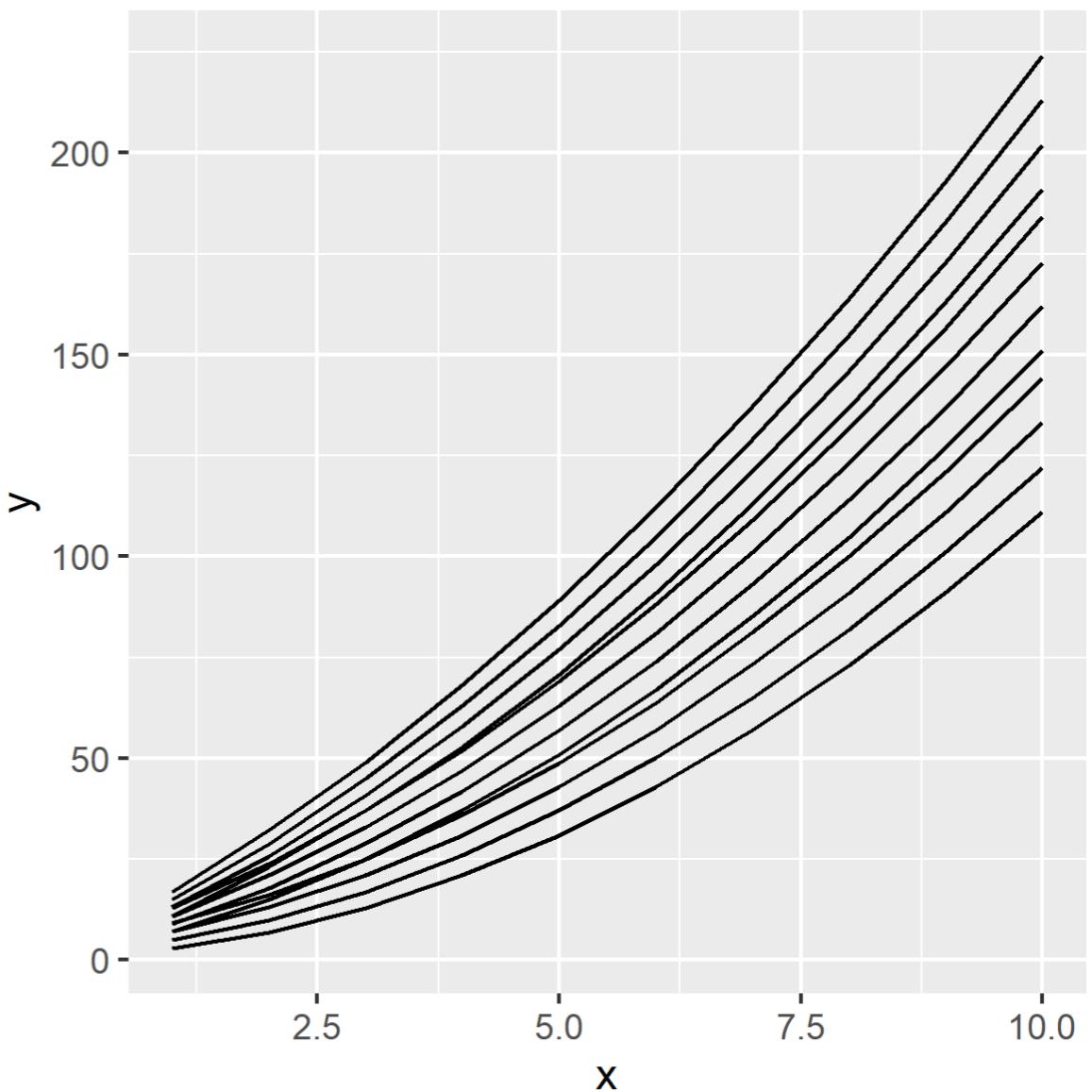
The corn dataset

```
glimpse(corn)
```

```
Observations: 6,381
Variables: 6
$ year                  <int> 1866, 1866, 1866, 1866, 1866, 1866...
$ state                 <chr> "Alabama", "Arkansas", "California...
$ farmed_area_acres    <dbl> 1050000, 280000, 42000, 57000, 200...
$ yield_bushels_per_acre <dbl> 9.0, 18.0, 28.0, 34.0, 23.0, 9.0, ...
$ farmed_area_ha        <dbl> 424919.92, 113311.98, 16996.80, 23...
$ yield_kg_per_ha       <dbl> 79.29892, 158.59784, 246.70776, 29...
```

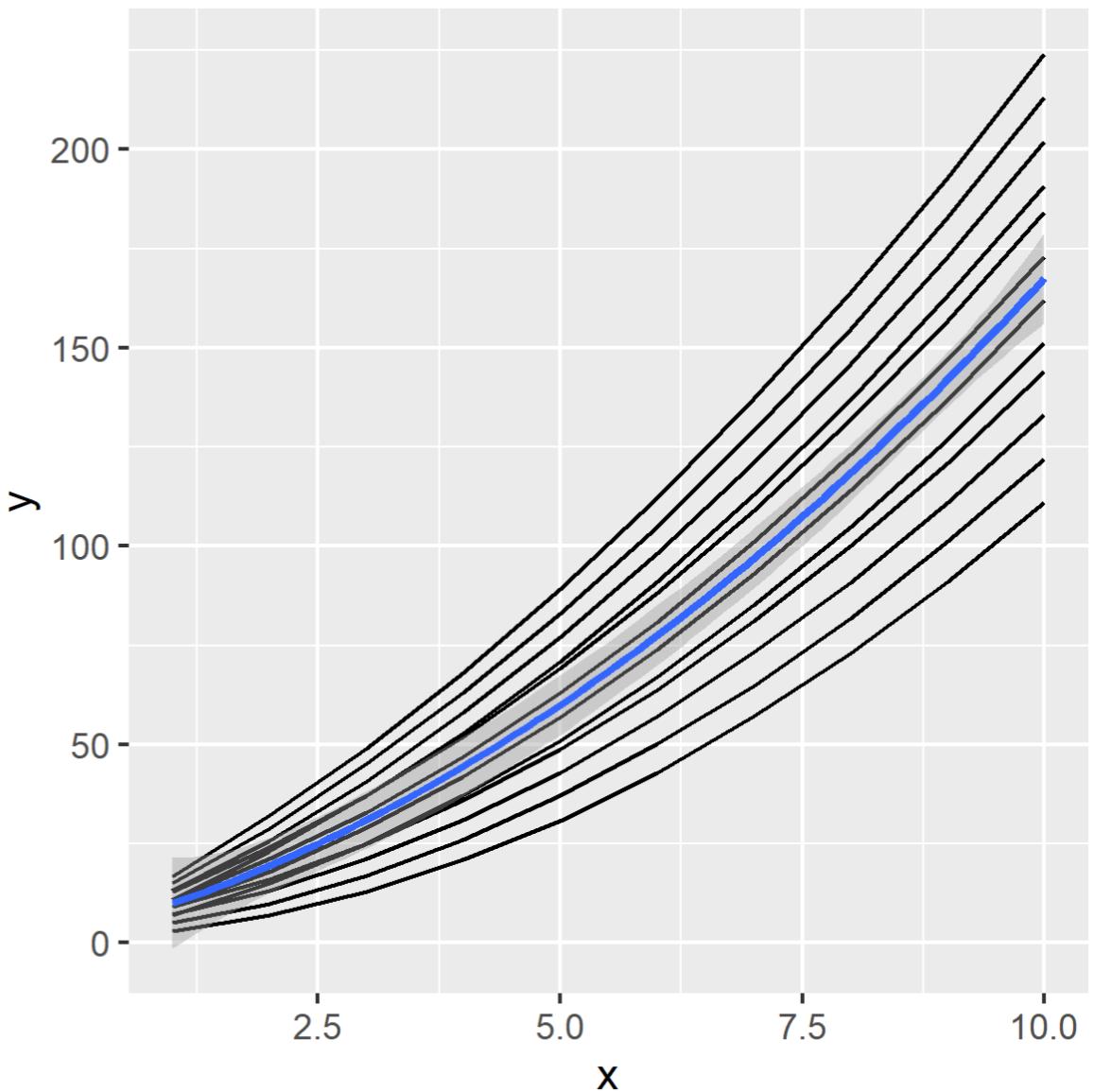
ggplot2: drawing multiple lines

```
ggplot(dataset, aes(x, y)) +  
  geom_line(aes(group = group))
```



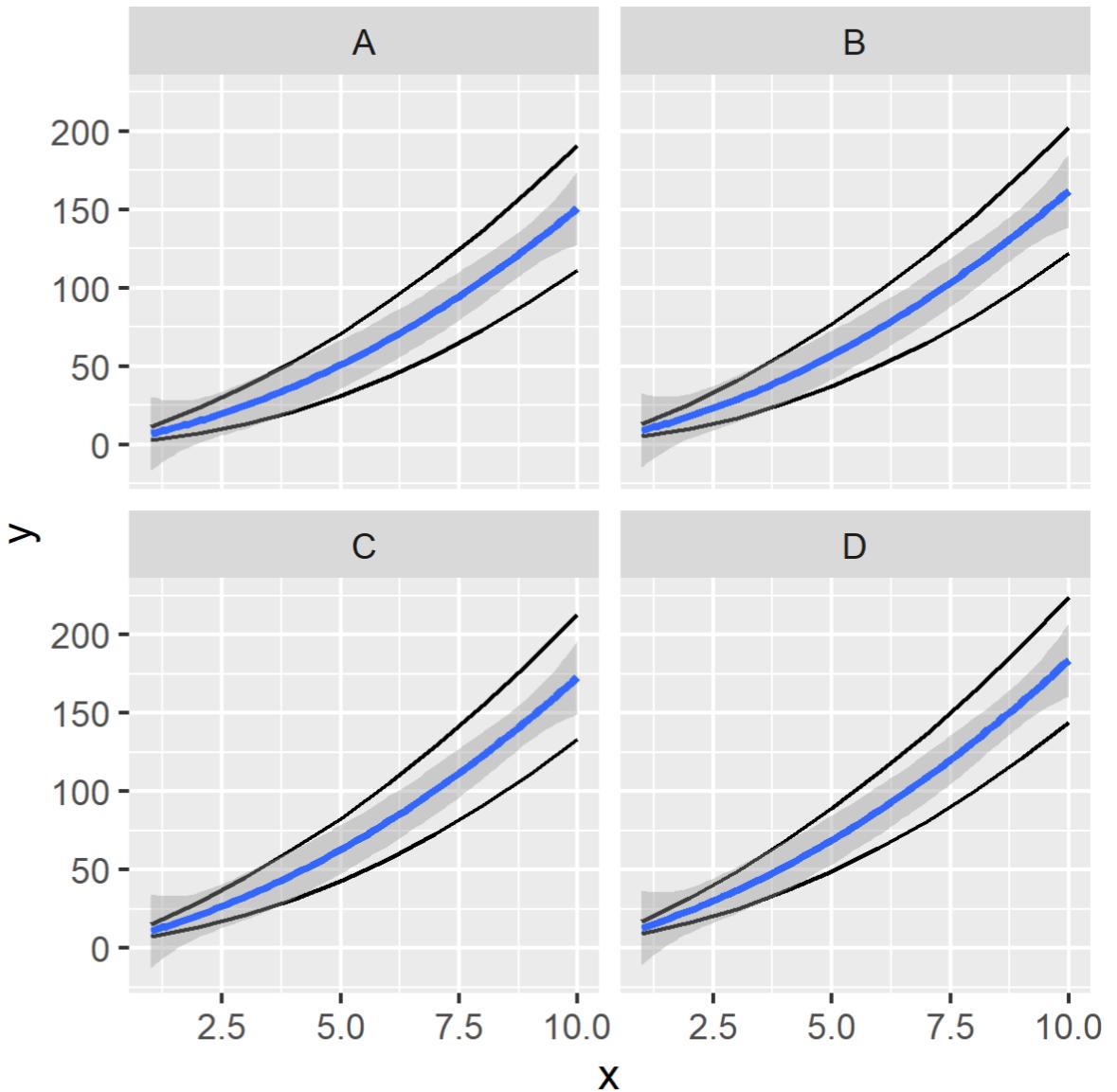
ggplot2: smooth trends

```
ggplot(dataset, aes(x, y)) +  
  geom_line(aes(group = group)) +  
  geom_smooth()
```

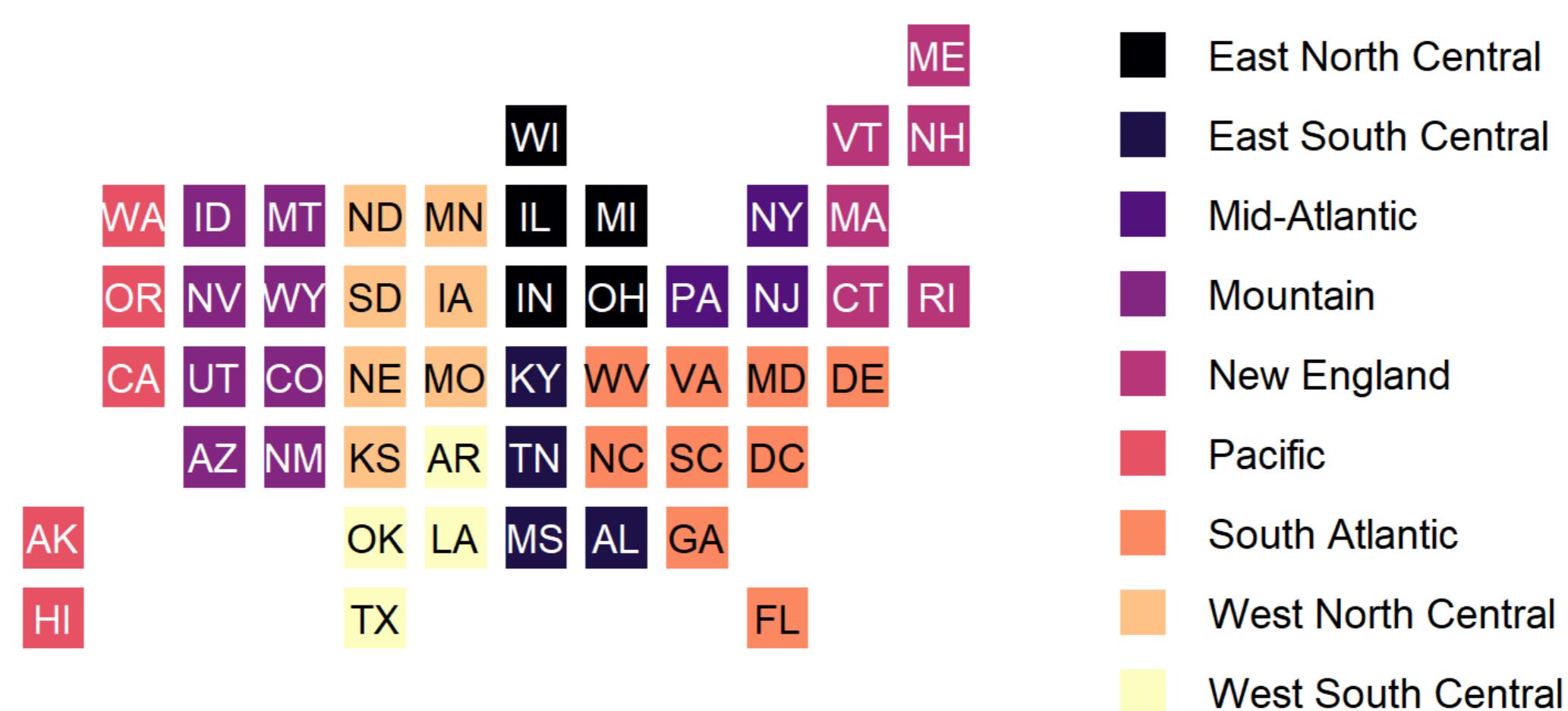


ggplot2: facetting

```
ggplot(dataset, aes(x, y)) +  
  geom_line(aes(group = group)) +  
  geom_smooth() +  
  facet_wrap(vars(facet))
```



USA Census regions



dplyr inner joins

```
dataset1 %>%  
  inner_join(dataset2, by = "column_to_join_on")
```

Let's practice!

INTRODUCTION TO WRITING FUNCTIONS IN R

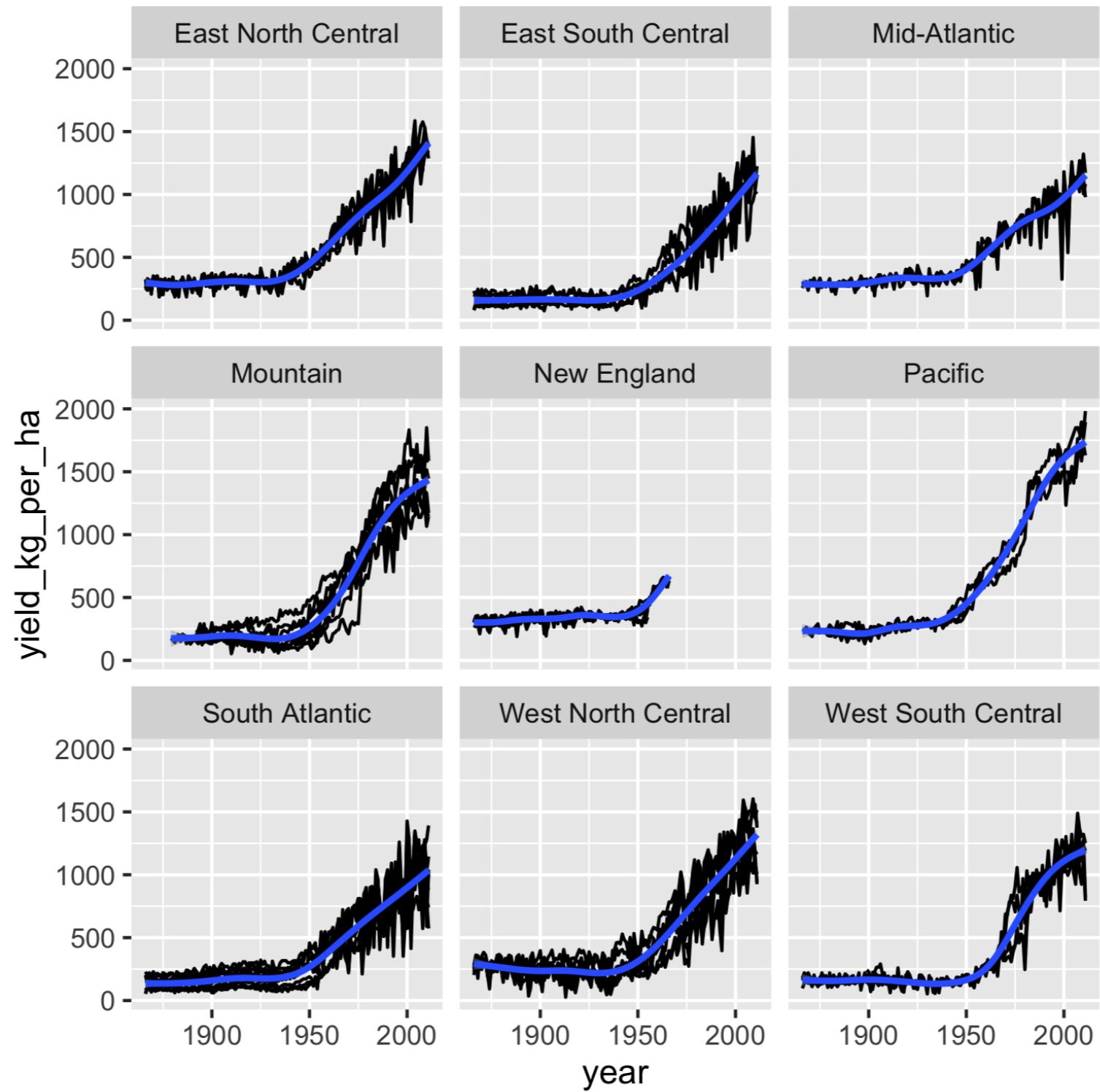
Modeling grain yields

INTRODUCTION TO WRITING FUNCTIONS IN R



Richie Cotton

Curriculum Architect at DataCamp



Linear models vs. generalized additive models

A linear model

```
lm(  
  response_var ~ explanatory_var1 + explanatory_var2,  
  data = dataset  
)
```

A generalized additive model

```
library(mgcv)  
gam(  
  response_var ~ s(explanatory_var1) + explanatory_var2,  
  data = dataset  
)
```

Predicting GAMs

```
predict_this <- data.frame(  
  explanatory_var1 = c("some", "values"),  
  explanatory_var2 = c("more", "values"))  
)
```

```
predicted_responses <- predict(model, predict_this, type = "response")
```

```
predict_this %>%  
  mutate(predicted_responses = predicted_responses)
```

Let's practice!

INTRODUCTION TO WRITING FUNCTIONS IN R

Congratulations

INTRODUCTION TO WRITING FUNCTIONS IN R



Richie Cotton

Curriculum Architect at DataCamp

In Chapter 1 you learned

- Writing your own functions lets you reuse code.
- There is a simple process for turning scripts into functions.
- Data arguments come before detail arguments.

In Chapter 2 you learned

- Defaults can be set using `name = value` syntax.
- Arguments can be passed between functions using their name or `....`.
- Checking user inputs can be done using `assertive`.

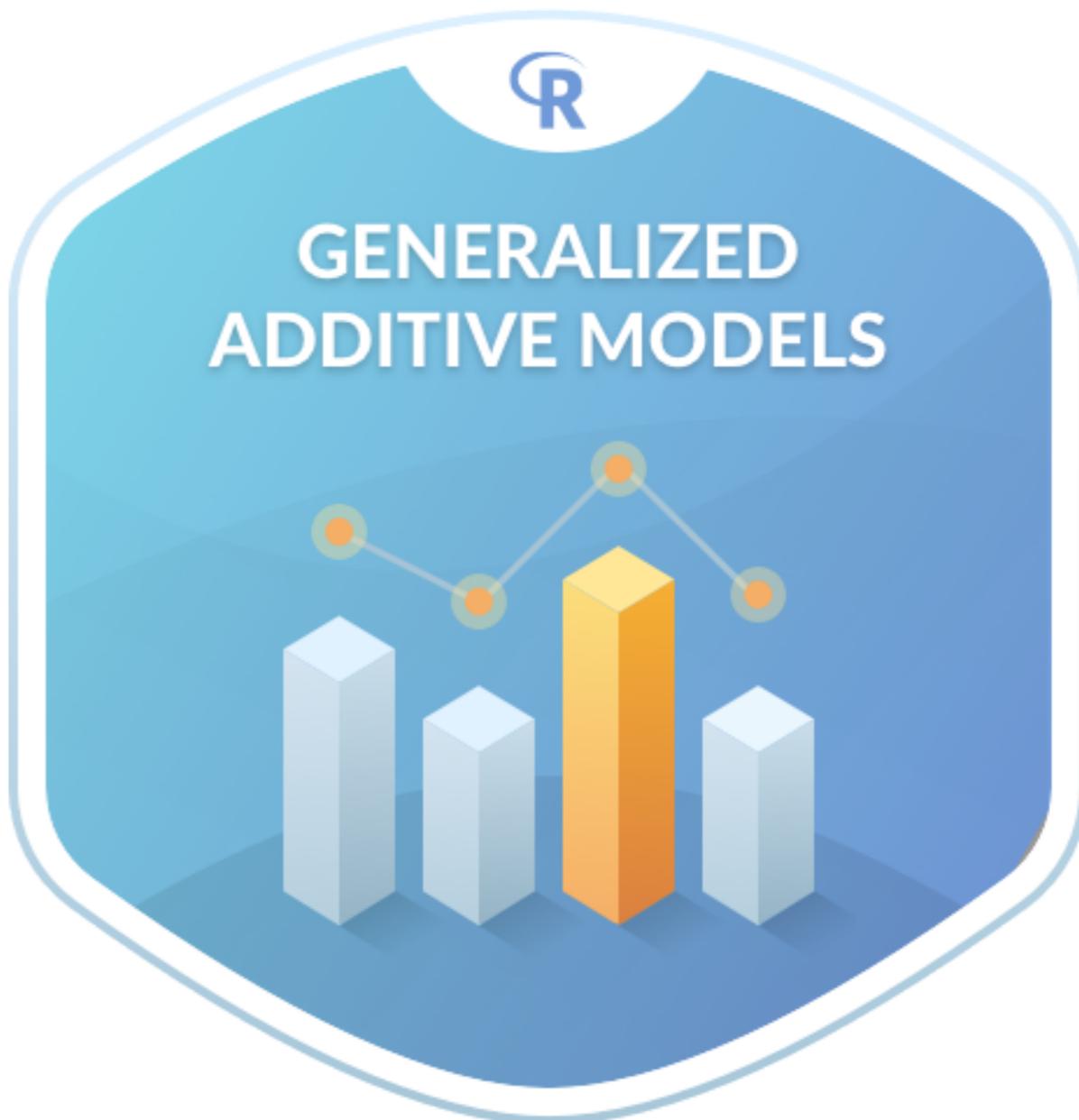
In Chapter 3 you learned

- You can return early from a function using `return()`.
- You can prevent return values being printed with `invisible()`.
- Functions can return multiple values using lists or attributes.
- R has rules about *scope* that determine which variables can be seen.

In Chapter 4 you learned

- Writing your own functions can be useful for your data analyses.
- Even simple, one-line functions can be helpful.

More modeling



Tidying models



Unit testing



Environments



**Thanks for taking the
course!**

INTRODUCTION TO WRITING FUNCTIONS IN R