Confounding variables

EXPERIMENTAL DESIGN IN PYTHON

Luke Hayden
Instructor

Confounding variables

Confounding variable

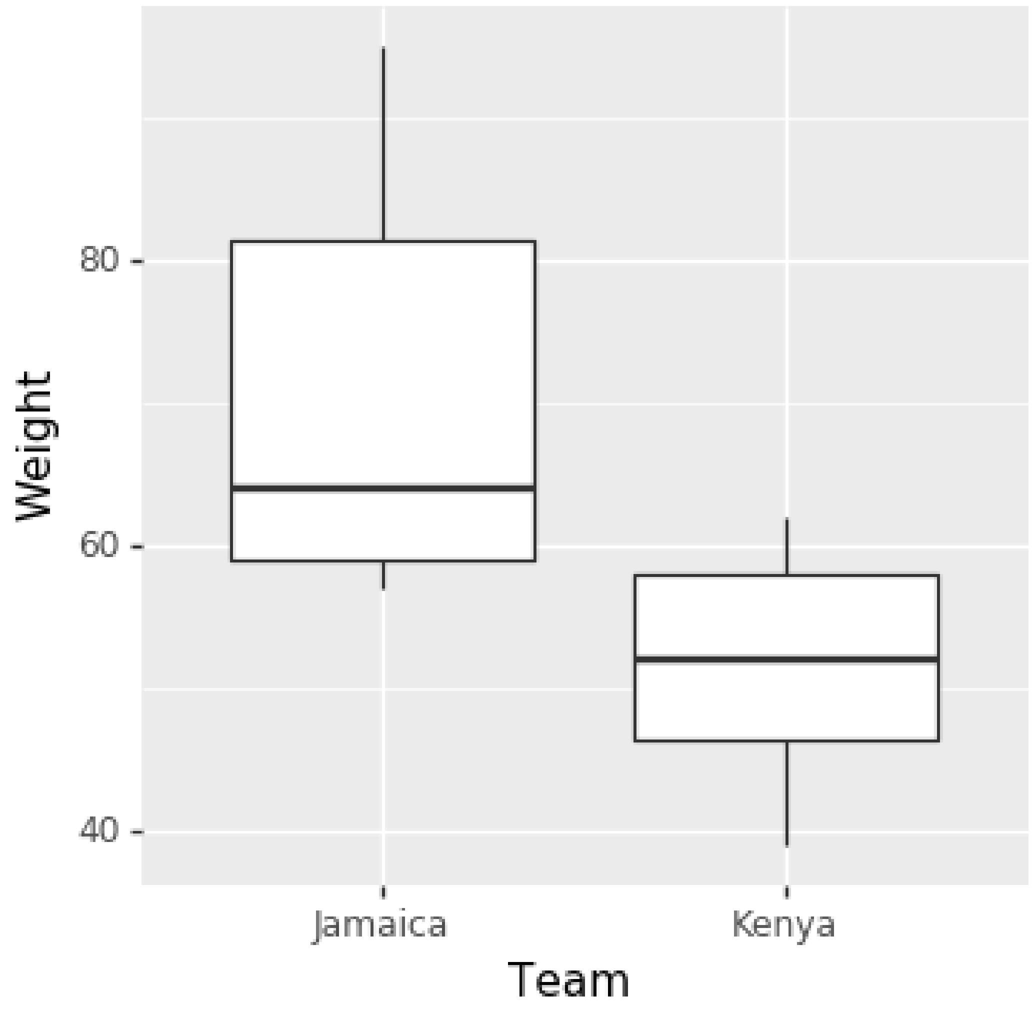
- Additional variable not accounted for in study design
- Alters the independent and dependent variables

Example

- Examining children's test scores
- Expensive cars and higher test scores in school correlate
- Reliable?
- Actually due to confounding
- Both linked to family income

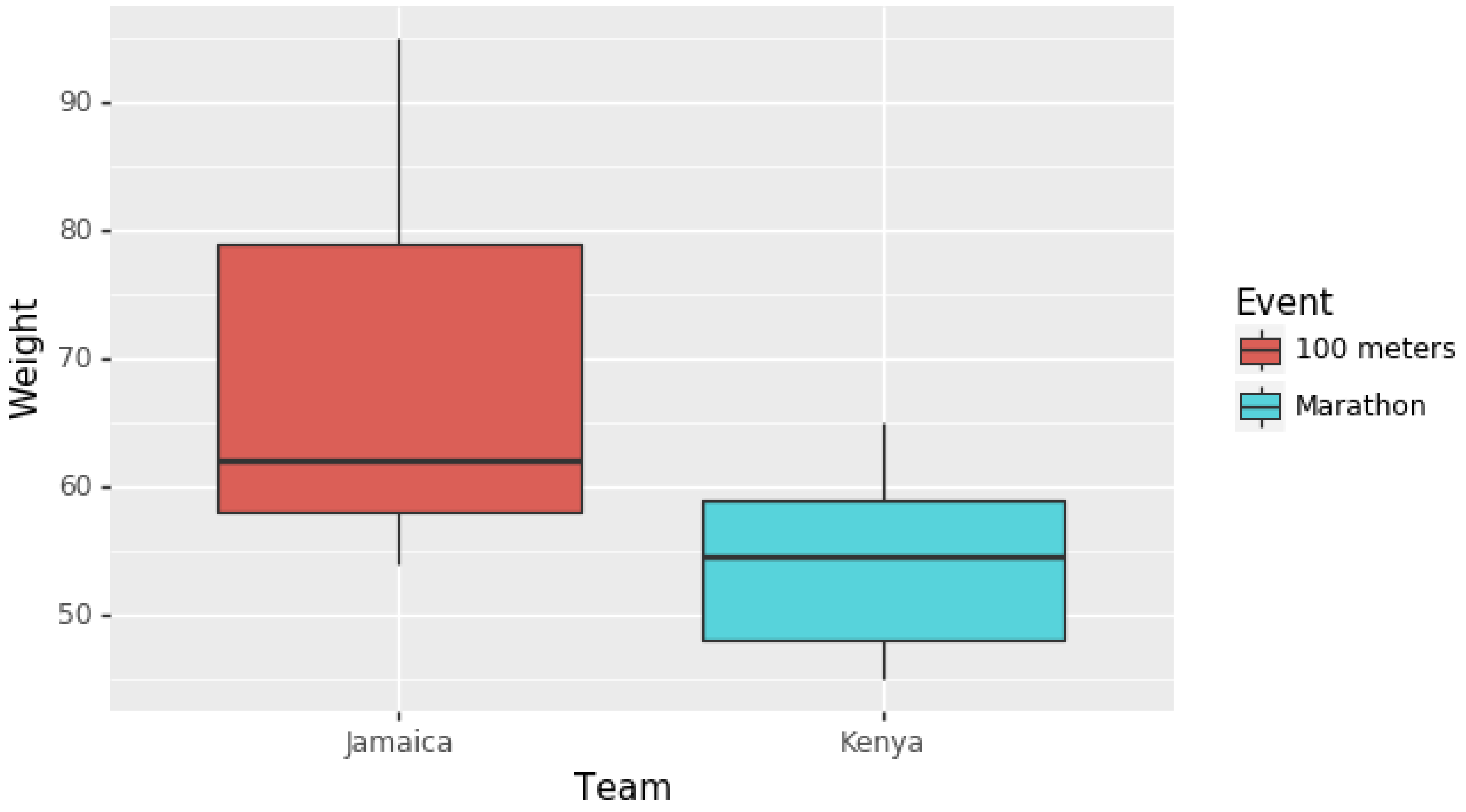
Obvious conclusion?

```
print(p9.ggplot(df)+ p9.aes(x= 'Team', y= 'Weight')+ p9.geom_boxplot())
```



Maybe not...

```
print(p9.ggplot(df)+ p9.aes(x= 'Team', y= 'Weight', fill="Event")+ p9.geom_boxplot())
```



Interpretation

Differences could be due to:

1. Country
2. Event
3. Country & event

Difficult to choose between these

- Event is a confounding variable

Let's practice!

EXPERIMENTAL DESIGN IN PYTHON

Blocking and randomization

EXPERIMENTAL DESIGN IN PYTHON

Luke Hayden
Instructor

Making comparisons

Compare like with like

- Only variable of interest should differ between groups

Remove sources of variation

- See variation of interest

Random sampling

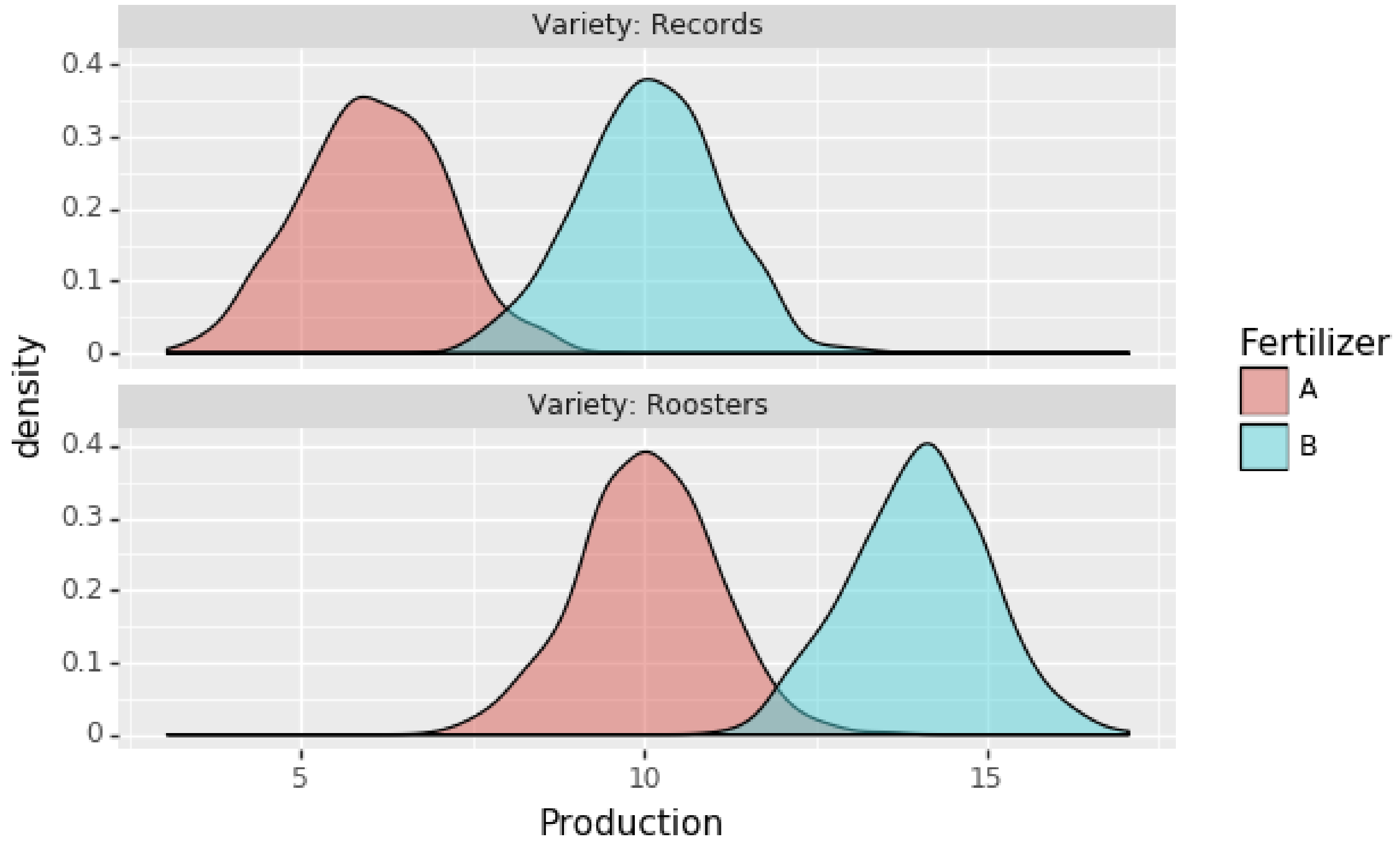
- Simple way to assign to treatments

```
import pandas as pd
from scipy import stats


seed= 1916
subset_A = df[df.Sample == "A"].sample(n= 30, random_state= seed)
subset_B = df[df.Sample == "B"].sample(n= 30, random_state= seed)


t_result = stats.ttest_ind(subset_A.value, subset_B.value)
```

Other sources of variation

Example

- Two potato varieties: Roosters & Records
- Two fertilizers: A & B
- Variety could be a confounder



Blocking

- Solution to confounding
- Control for confounding by balancing with respect to other variable

Example

- Equal proportions of each variety treated with each fertilizer

Design

Variety	Fertilizer A	Fertilizer B
Records	10	10
Roosters	10	10

Implementing a blocked design

```
import pandas as pd

block1 = df[(df.Variety == "Roosters") ].sample(n=15, random_state= seed)
block2 = df[(df.Variety == "Records") ].sample(n=15, random_state= seed)

fertAtreatment = pd.concat([block1, block2])
```

Paired samples

Special case

- Control for individual variation
- Increase statistical power by reducing noise

Example

- Yield of 5 fields before/after change of fertilizer

2017 yield (tons/hectare)	2018 yield (tons/hectare)
60.2	63.2
12	15.6
13.8	14.8
91.8	96.7
50	53

Implementing a paired t-test

```
from scipy import stats

yields2018= [60.2, 12, 13.8, 91.8, 50]
yields2019 = [63.2, 15.6, 14.8, 96.7, 53]

ttest = stats.ttest_rel(yields2018,yields2019)

print(ttest[1])
```

p-value:

```
0.007894143467973484
```

Let's practice!

EXPERIMENTAL DESIGN IN PYTHON

ANOVA

EXPERIMENTAL DESIGN IN PYTHON

Luke Hayden
Instructor

Variable types

Independent (Factors)

- Manipulate experimentally

Dependent

- Try to understand their patterns

t-test

- One discrete independent variable with two levels
- One dependent variable

ANOVA

- Analysis of variance
- Generalize t-test to broader set of cases
- Examine multiple factors/levels

Approach

- Partition variation into separate components
- Multiple simultaneous tests

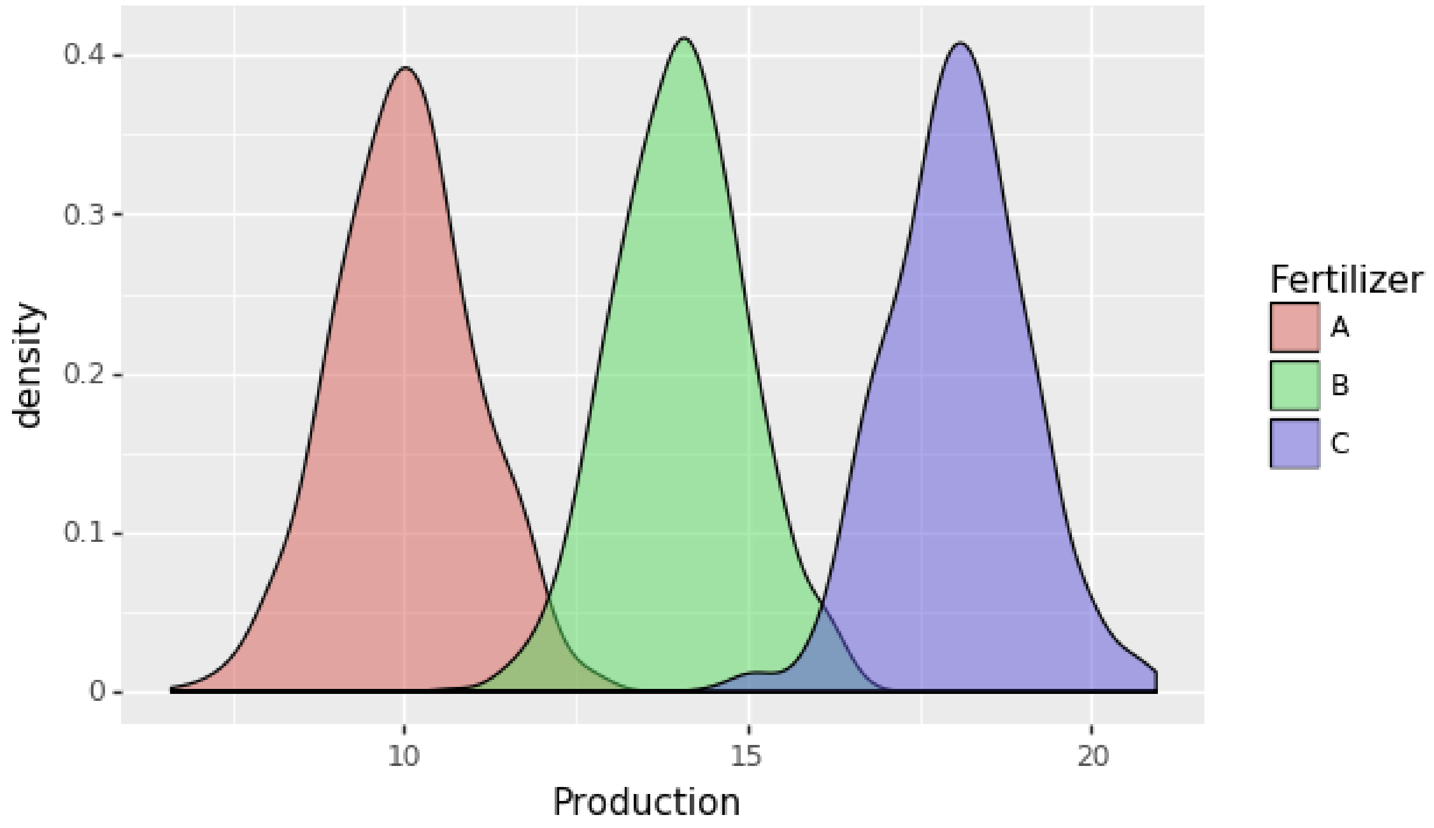
One-way ANOVA

Use

- One factor with 3+ levels
- Does factor affect sample mean?

Example:

- Does potato production differ between three fertilizers?



Implementing a one-way ANOVA

```python
from scipy import stats

array_fertA = df[df.Fertilizer == "A"].Production
array_fertB = df[df.Fertilizer == "B"].Production
array_fertC = df[df.Fertilizer == "C"].Production

anova = stats.f_oneway(array_fertA, array_fertB, array_fertC)

print(anova[1])
```

```
0.00
```

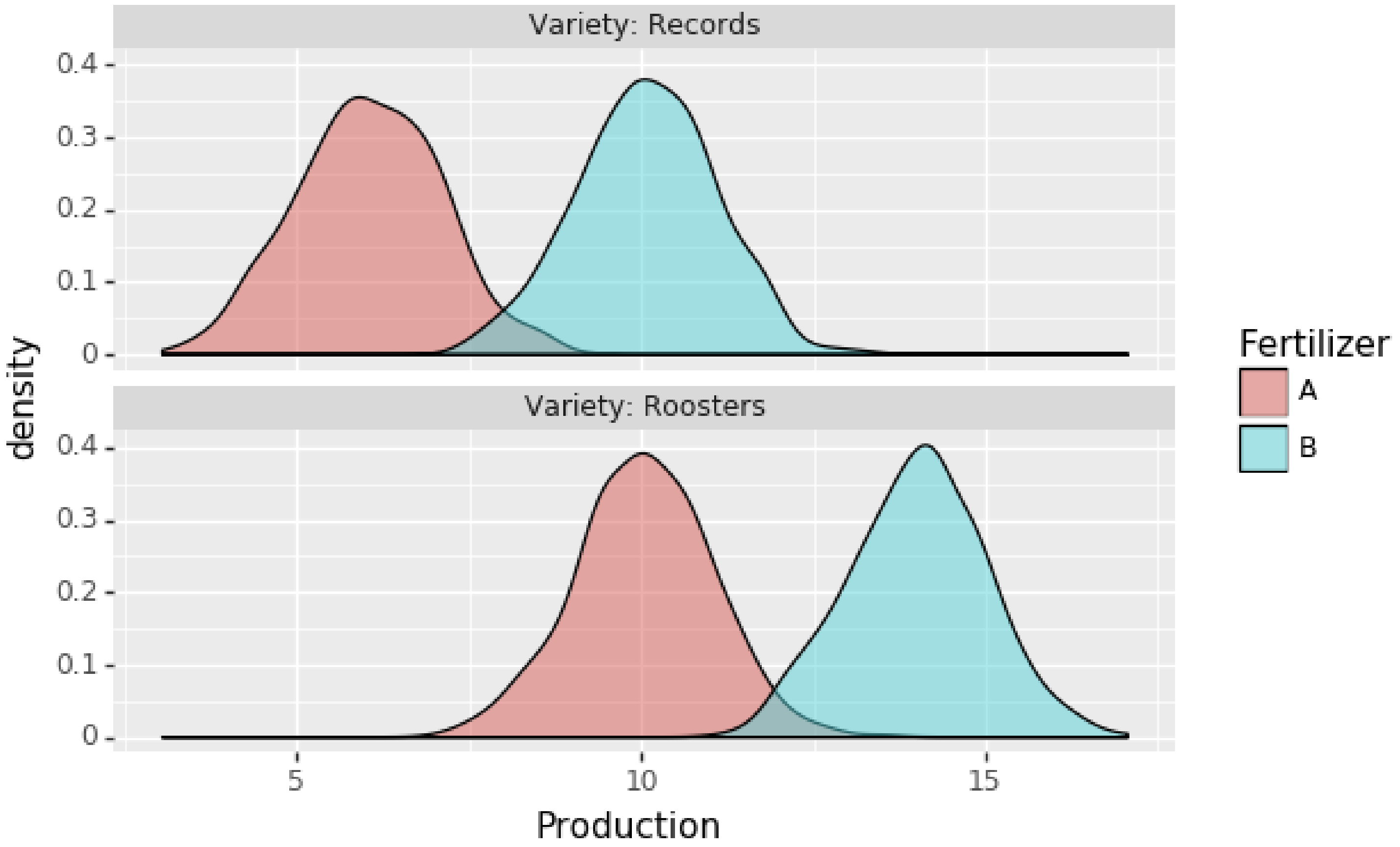
Two-way ANOVA

Use

- Two factors with 2+ levels
- Does each factor explain variation in the dependent variable?

Example

- 2 fertilizers, 2 potato varieties
- Potato production (dependent variable)

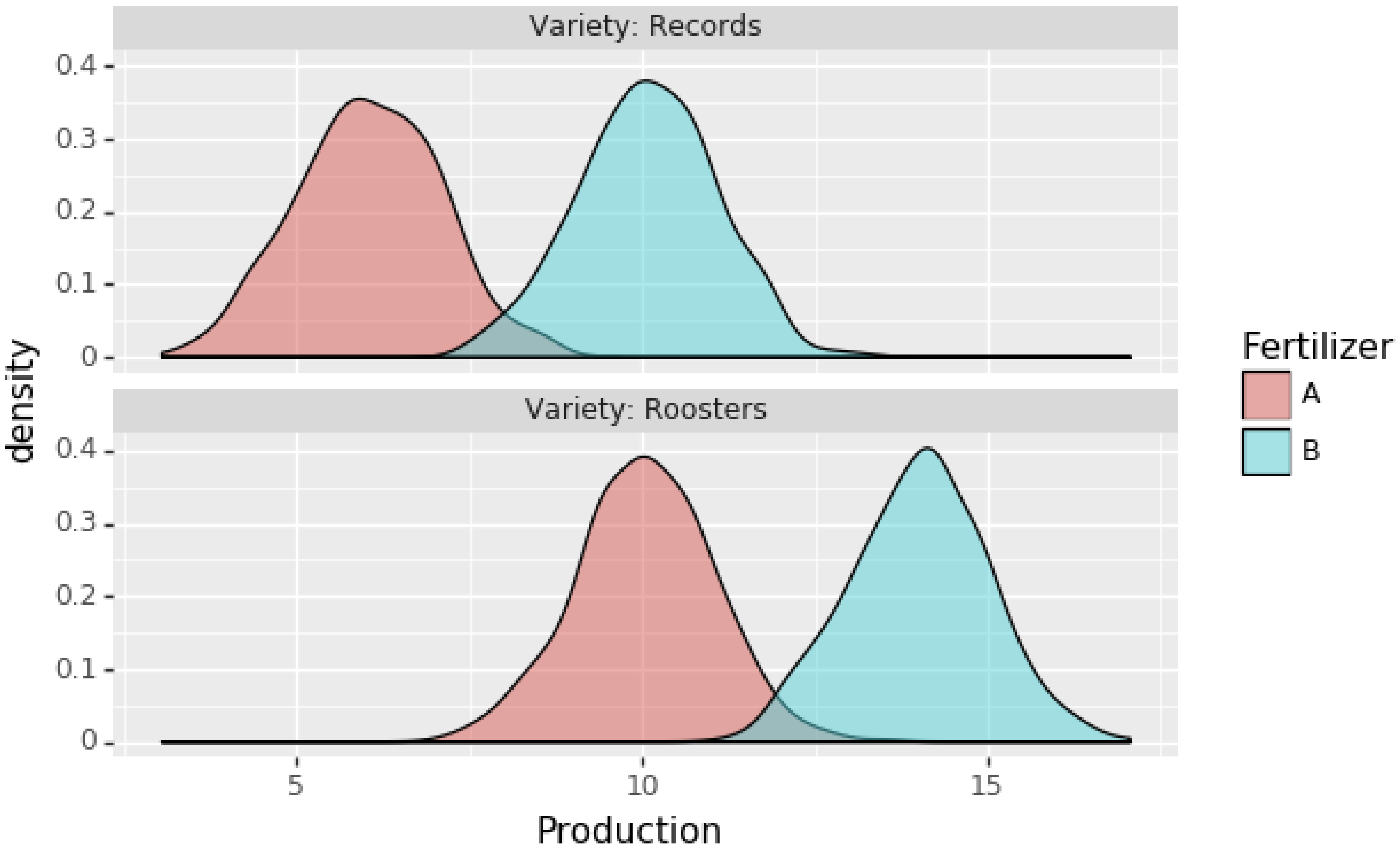


Implementing a two-way ANOVA

```
import statsmodels as sm

formula = 'Production ~ Fertilizer + Variety'
model = sm.api.formula.ols(formula, data=df).fit()

aov_table = sm.api.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                          sum_sq      df             F   PR(>F)
Fertilizer                           1.0                   p-value
Variety                              1.0                   p-value
Residual                                           NaN       NaN
```

Example


Variety: Records
Variety: Roosters
density
0.4
0.3
0.2
0.1
0
5
10
15
Production
Fertilizer
A
B

Example output

```python
import statsmodels as sm

formula = 'Production ~ Fertilizer + Variety'
model = sm.api.formula.ols(formula, data=df).fit()

aov_table = sm.api.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                 sum_sq      df             F  PR(>F)
Fertilizer  16247.966193     1.0  16347.749306     0.0
Variety     15881.785333     1.0  15979.319631     0.0
Residual     3972.603180  3997.0           NaN     NaN
```

Let's practice!

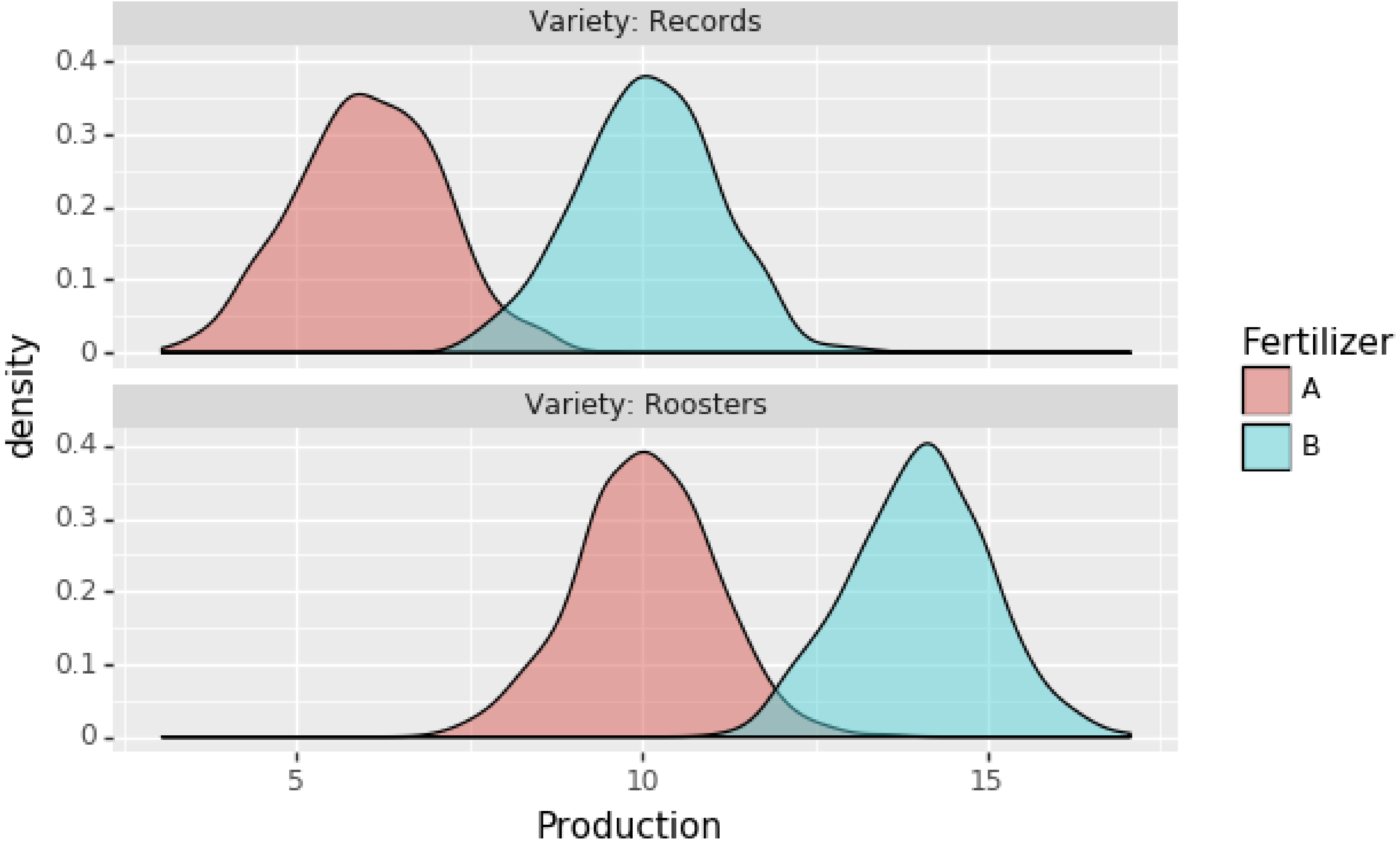
EXPERIMENTAL DESIGN IN PYTHON

Interactive effects

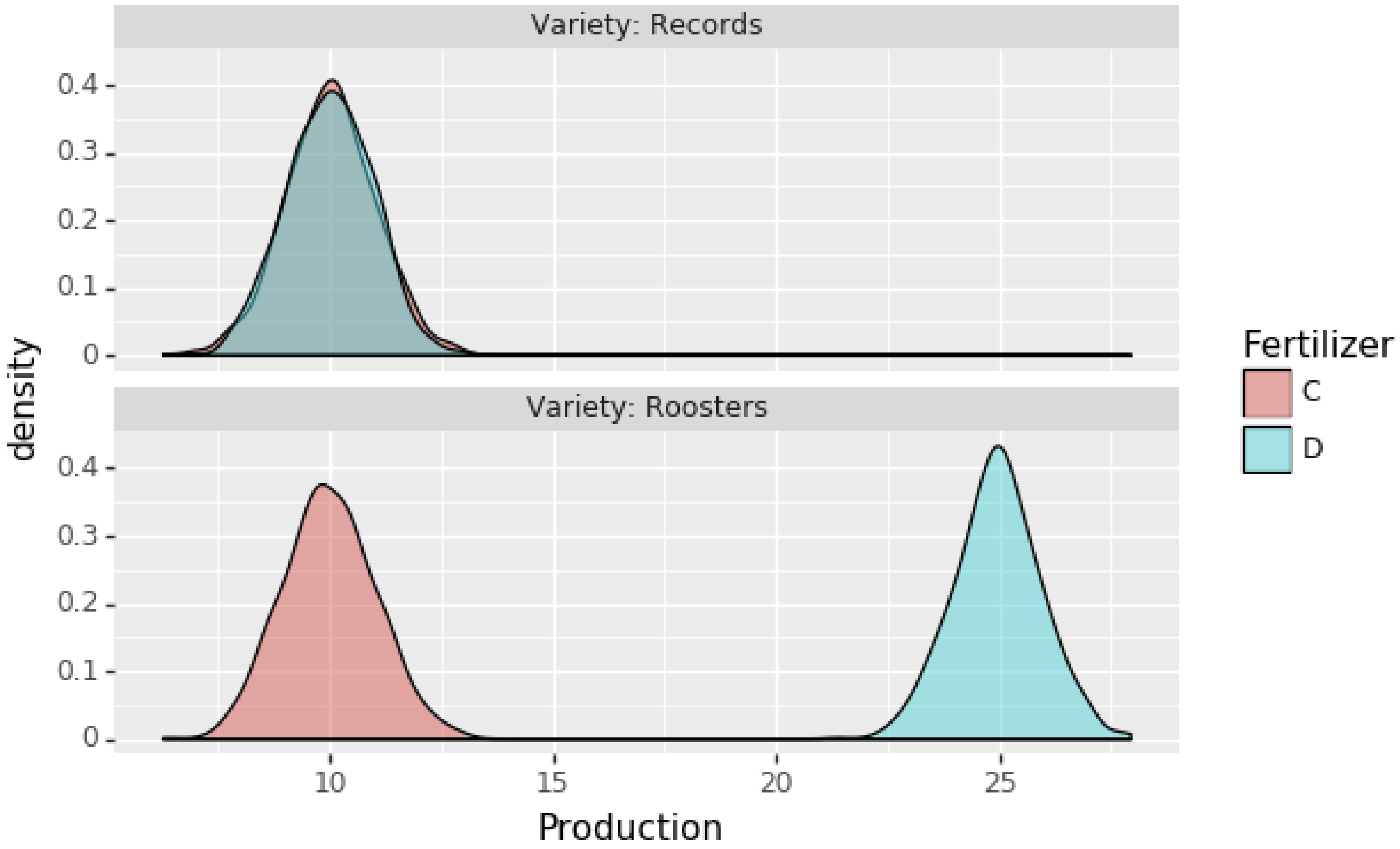
EXPERIMENTAL DESIGN IN PYTHON

Luke Hayden
Instructor

Additive model



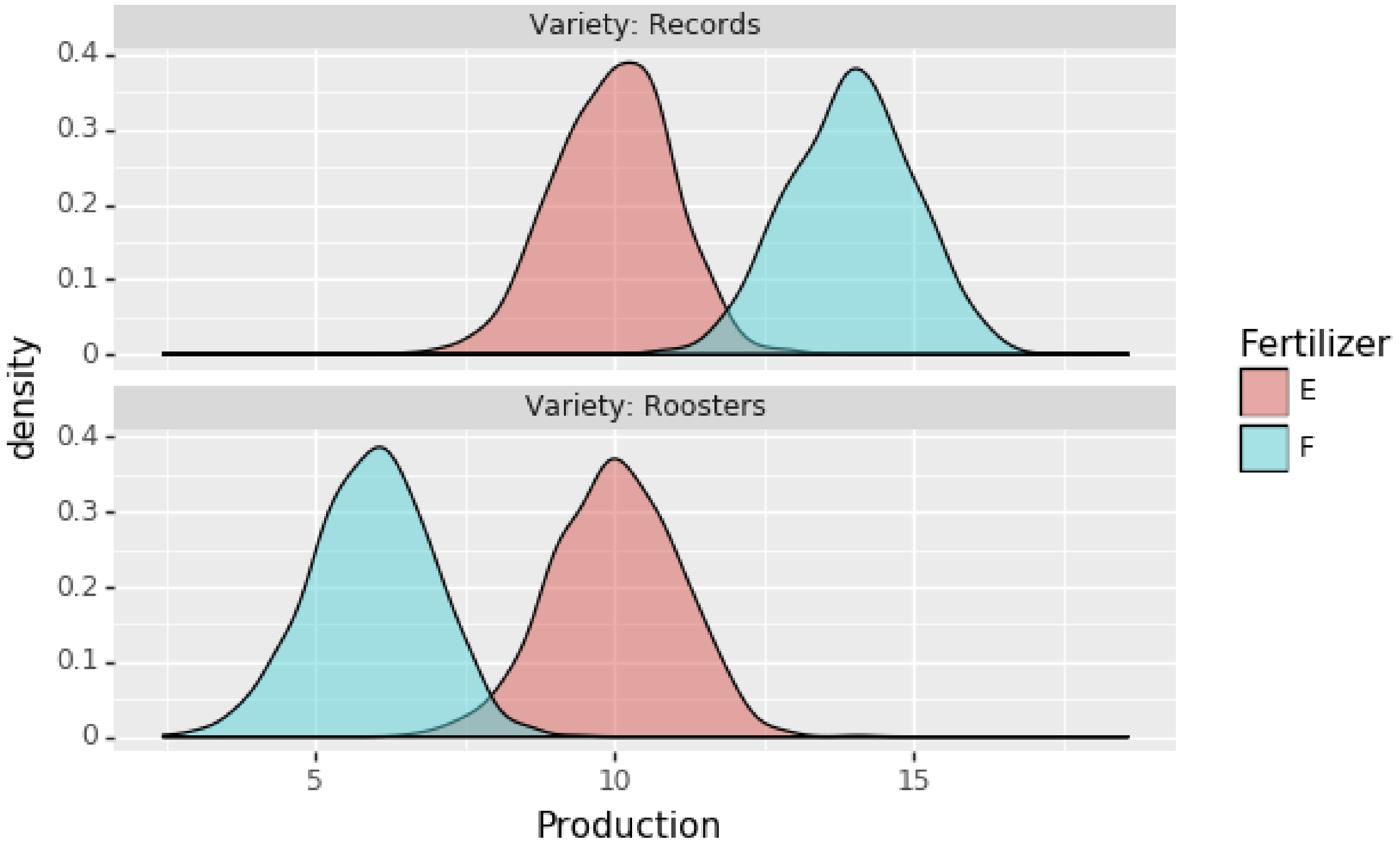
Interactive effects



In this example:

- Fertilizer D only better for Rooster potatoes

Interactive effects



In this example:

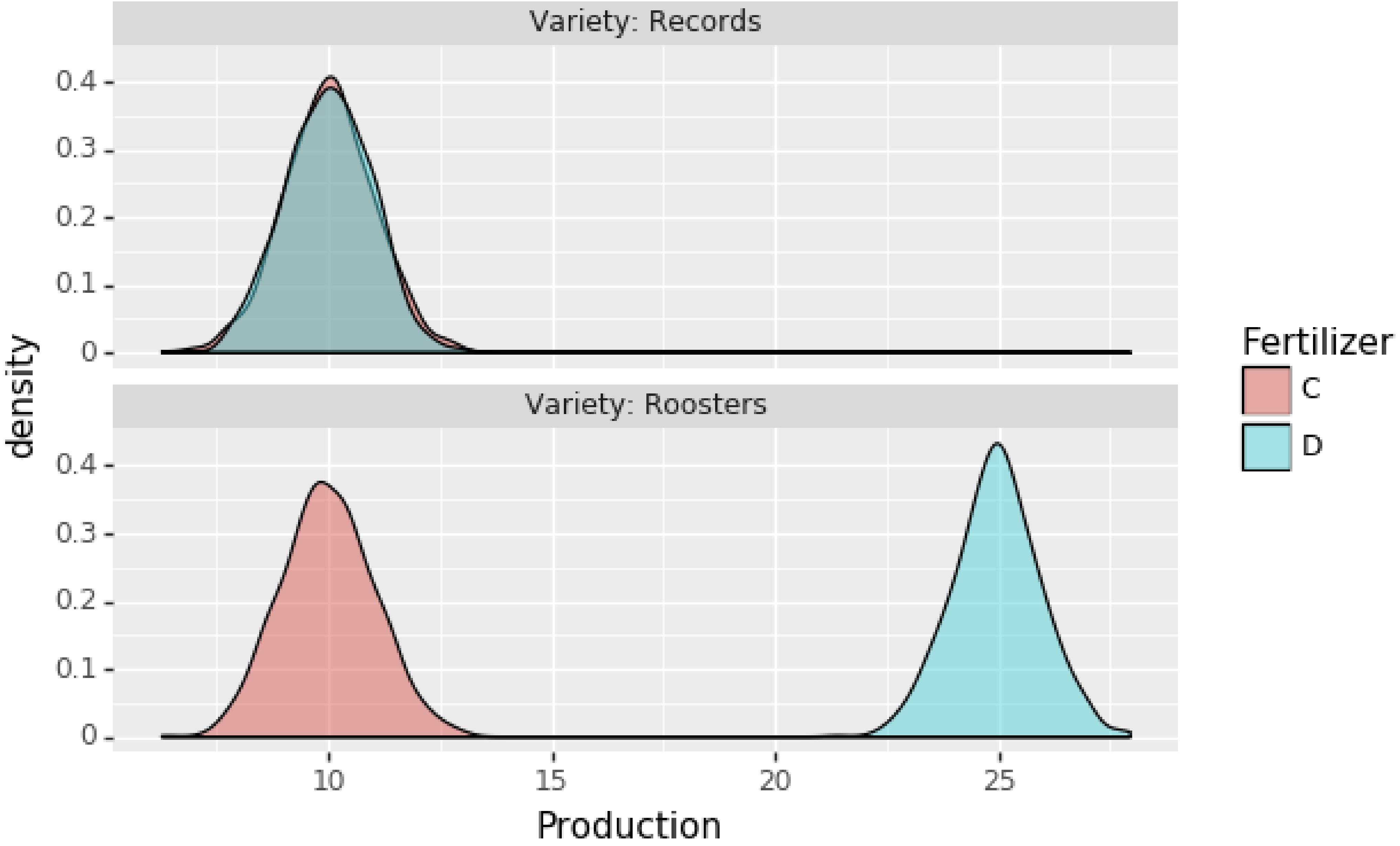
- Fertilizer E is best for Roosters
- Fertilizer F is best for Records

Implementing ANOVA with interactive effects

```python
import statsmodels as sm
formula = 'Production ~ Fertilizer + Variety + Fertilizer:Variety'
model = sm.api.formula.ols(formula, data=df).fit()
aov_table = sm.api.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                     sum_sq      df             F   PR(>F)
Fertilizer                      1.0                    p-value
Variety                         1.0                    p-value
Fertilizer:Variety              1.0                    p-value
Residual                                      NaN      NaN
```

Example 1



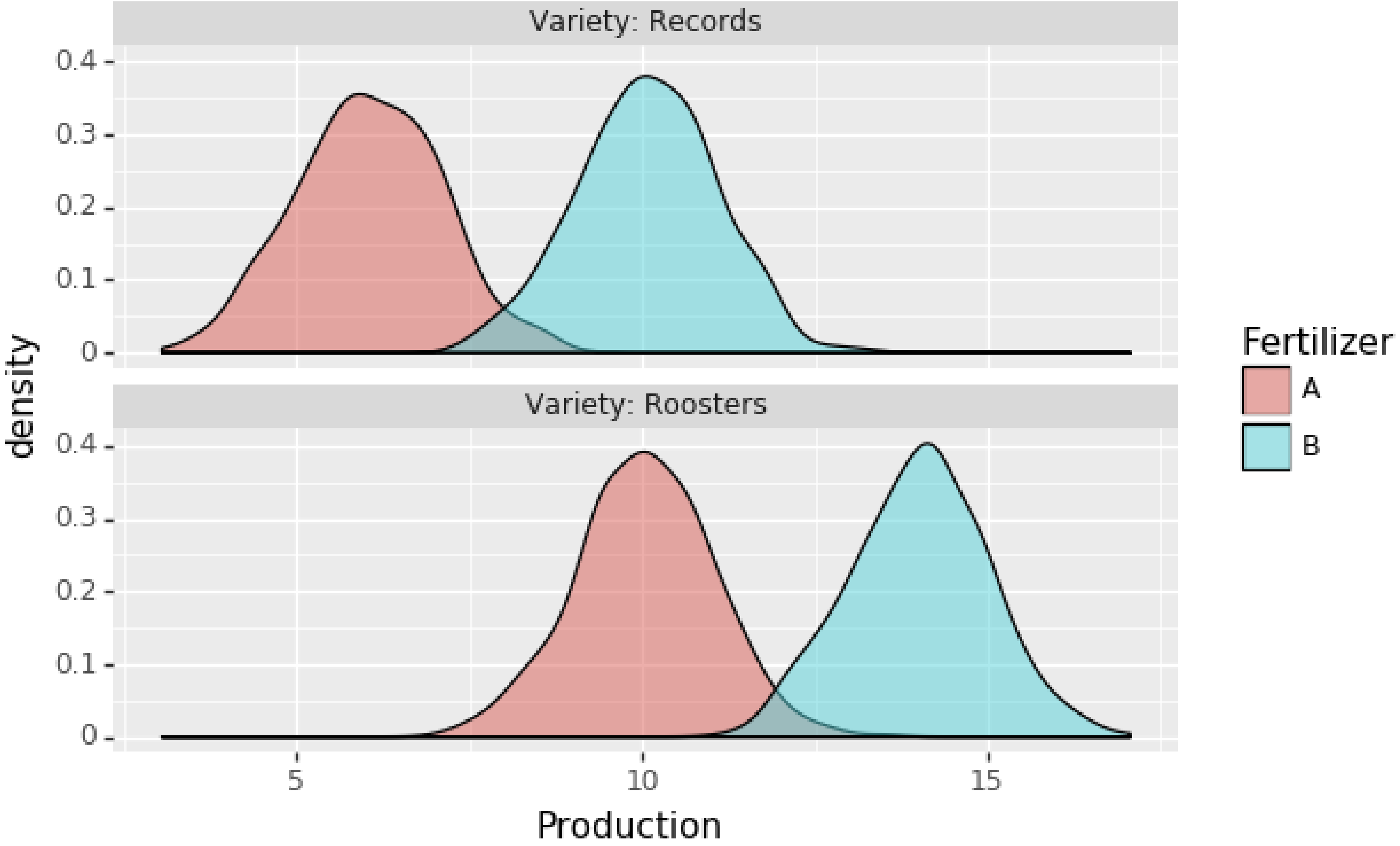
Variety: Records
Variety: Roosters
0.4
0.3
0.2
0.1
0
density
10
15
20
25
Production
Fertilizer
C
D

Interactive effect

```
import statsmodels as sm
formula = 'Production ~ Fertilizer + Variety + Fertilizer:Variety'
model = sm.api.formula.ols(formula, data=df).fit()
aov_table = sm.api.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                          sum_sq      df             F  PR(>F)
Fertilizer          56425.833205     1.0  60222.992593     0.0
Variety             56049.056459     1.0  59820.860770     0.0
Fertilizer:Variety  55385.556078     1.0  59112.710332     0.0
Residual             3744.045584  3996.0           NaN     NaN
```

Example 2


Variety: Records
Variety: Roosters
0.4
0.3
0.2
0.1
0
density
5
10
15
Production
Fertilizer
A
B

No interactive effect

```
import statsmodels as sm
formula = 'Production ~ Fertilizer + Variety + Fertilizer:Variety'
model = sm.api.formula.ols(formula, data=df).fit()
aov_table = sm.api.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                         sum_sq      df             F    PR(>F)
Fertilizer         15468.395105     1.0  15172.001139  0.000000
Variety            16010.275045     1.0  15703.497977  0.000000
Fertilizer:Variety     1.464654     1.0      1.436589  0.230763
Residual            4074.064210  3996.0           NaN       NaN
```

Beyond 2-way ANOVA

Two-way ANOVA

```
formula = 'Production ~
Fertilizer + Variety + Fertilizer:Variety'
```

- 3 variables, 3 p-values
- 2 factors, 1 interaction

Three-way ANOVA

```
formula = 'Production ~ Fertilizer +
Variety + Season + Fertilizer:Variety +
Fertilizer:Season + Variety:Season +
Fertilizer:Variety:Season'
```

- 7 variables, 7 p-values
- 3 factors, 4 interactions

Let's practice!

EXPERIMENTAL DESIGN IN PYTHON