Designing and training models

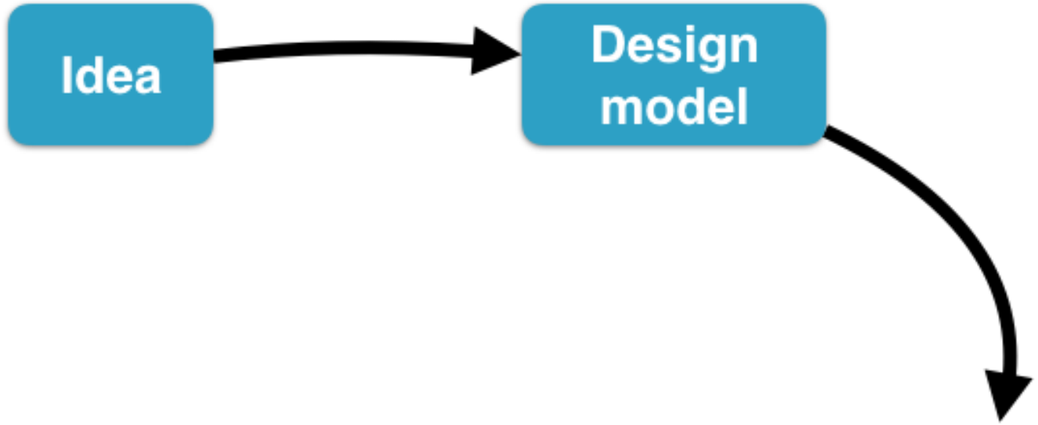
INTRODUCTION TO STATISTICAL MODELING IN R

Danny Kaplan
Instructor

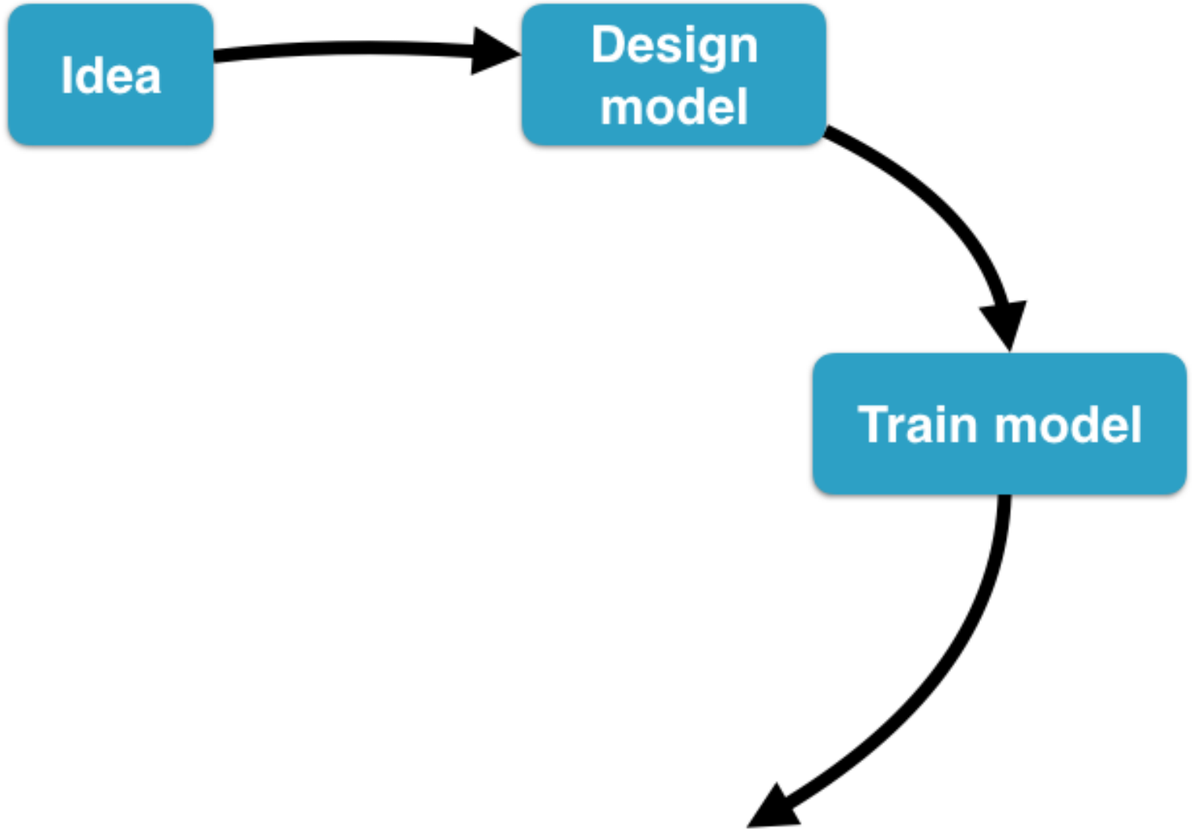
datacamp

Modeling is a process

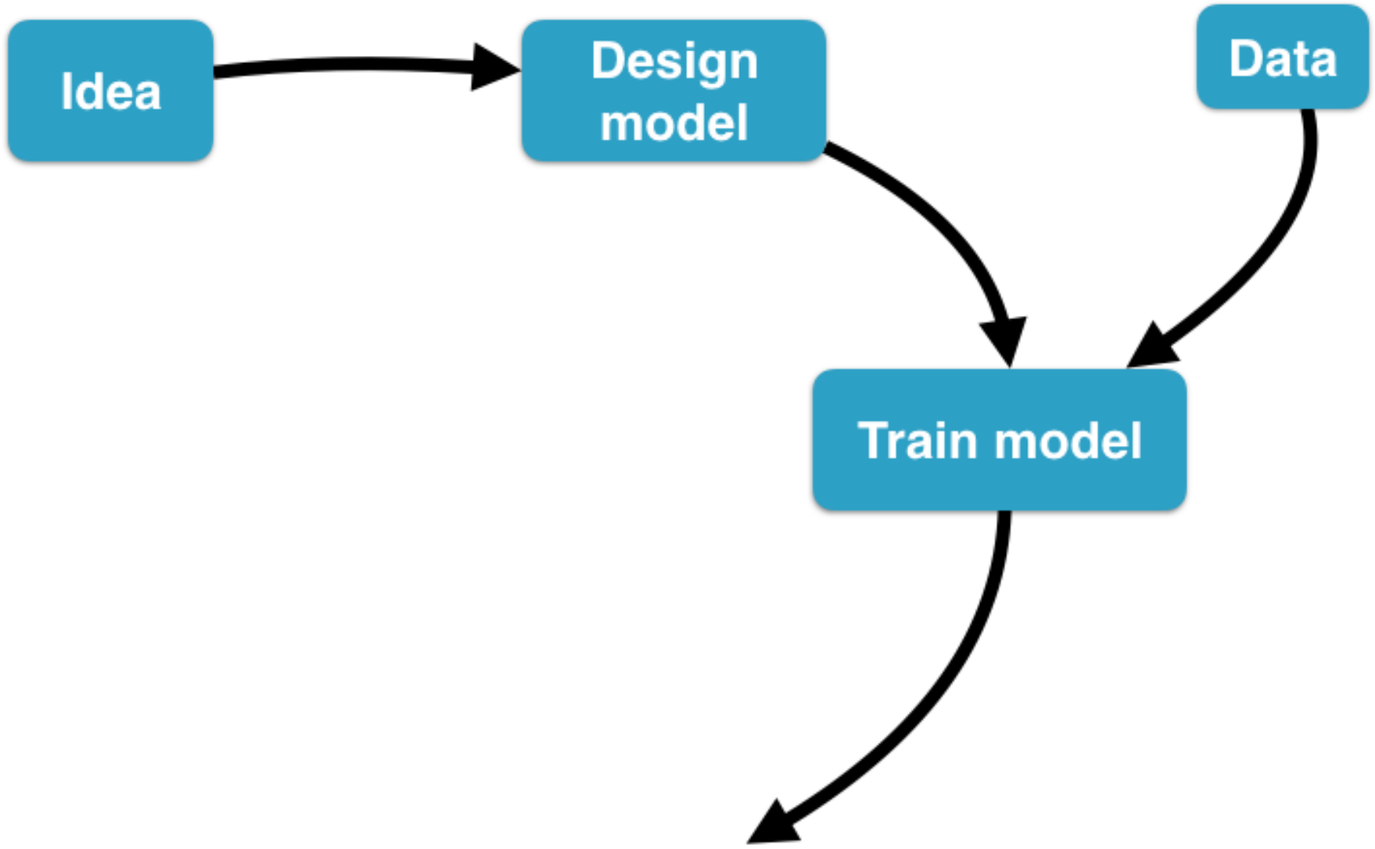
Modeling is a process



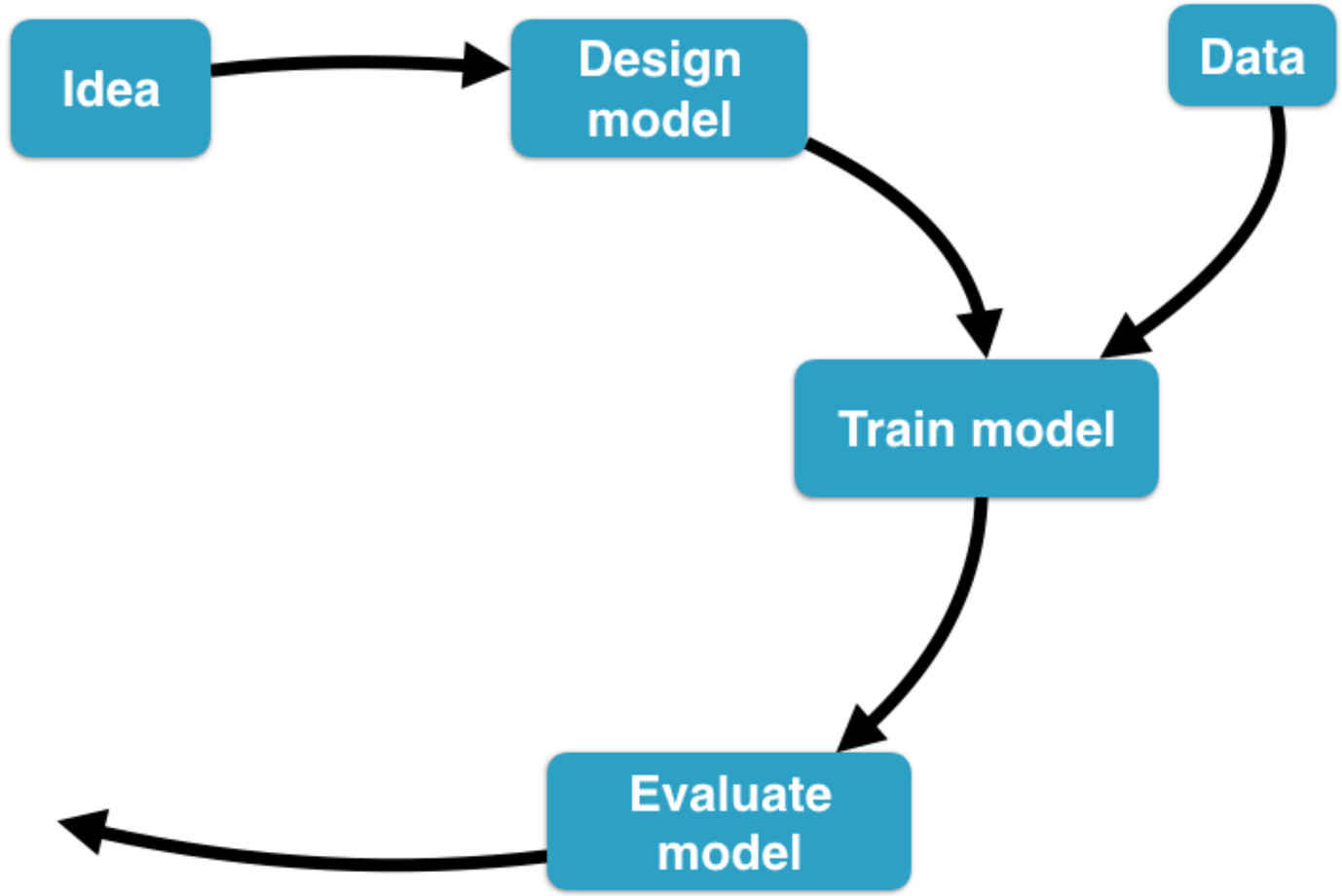
Modeling is a process



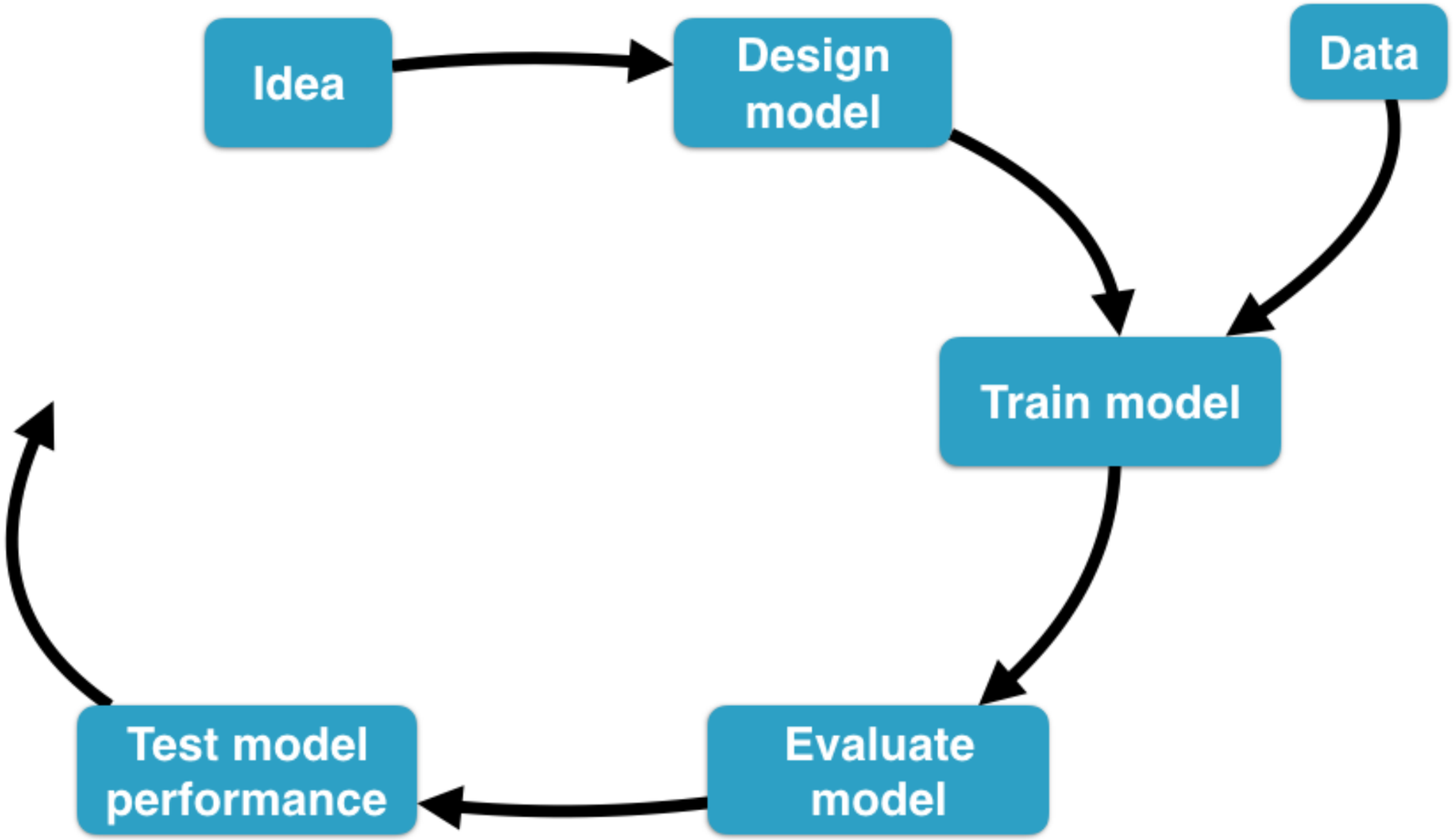
Modeling is a process



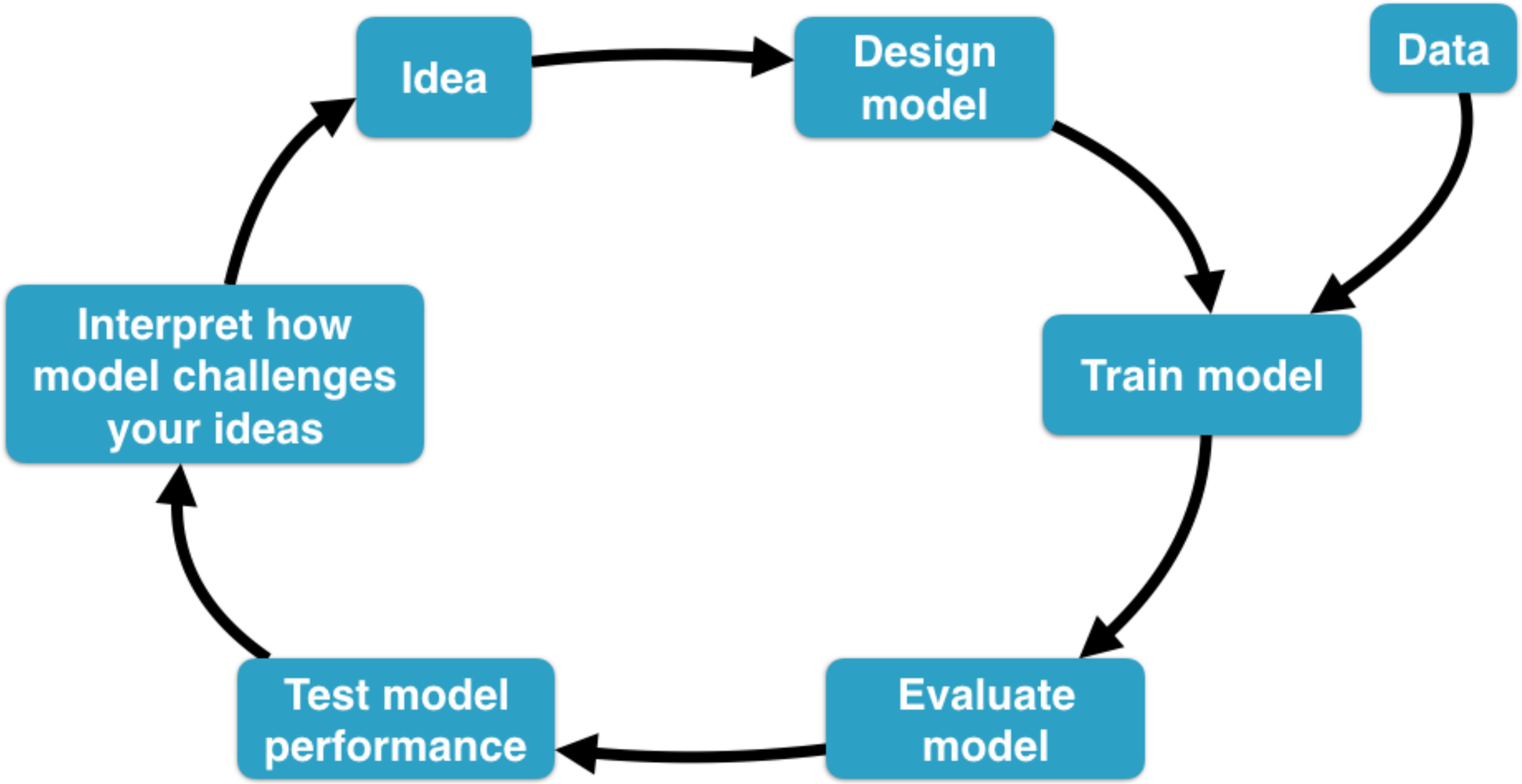
Modeling is a process



Modeling is a process



Modeling is a process



Choices in model design

- A suitable training data set
- Specify response and explanatory variables
- Select a model architecture
  - Linear model: `lm()`
  - Recursive partitioning: `rpart()`

Training a model

- Automatic process carried out by the computer
- Tailors (i.e. "fits") the model to the data
- Model represents both your choices and data

The CPS85 data

```
library(mosaicData)
head(CPS85)
```

```
  wage educ race sex hispanic south married exper union age   sector
1  9.0   10    W   M       NH    NS Married    27   Not  43    const
2  5.5   12    W   M       NH    NS Married    20   Not  38    sales
3  3.8   12    W   F       NH    NS  Single     4   Not  22    sales
4 10.5   12    W   F       NH    NS Married    29   Not  47 clerical
5 15.0   12    W   M       NH    NS Married    40 Union  58    const
6  9.0   16    W   F       NH    NS Married    27   Not  49 clerical
```

Modeling wage and education

- Choose `wage` as the response variable
- Choose `educ` and `exper` as explanatory variables
- Primary interest is `educ`, so `exper` is a *covariate*

```
model_1 <- lm(wage ~ educ + exper, data = CPS85)
model_2 <- rpart(wage ~ educ + exper, data = CPS85)
```

Let's practice!

INTRODUCTION TO STATISTICAL MODELING IN R

Evaluating models

INTRODUCTION TO STATISTICAL MODELING IN R

Danny Kaplan
Instructor

datacamp

From the last lesson

```
model_1 <- lm(wage ~ educ + exper, data = CPS85)


model_2 <- rpart(wage ~ educ + exper, data = CPS85)
```

Don't worry about the internals

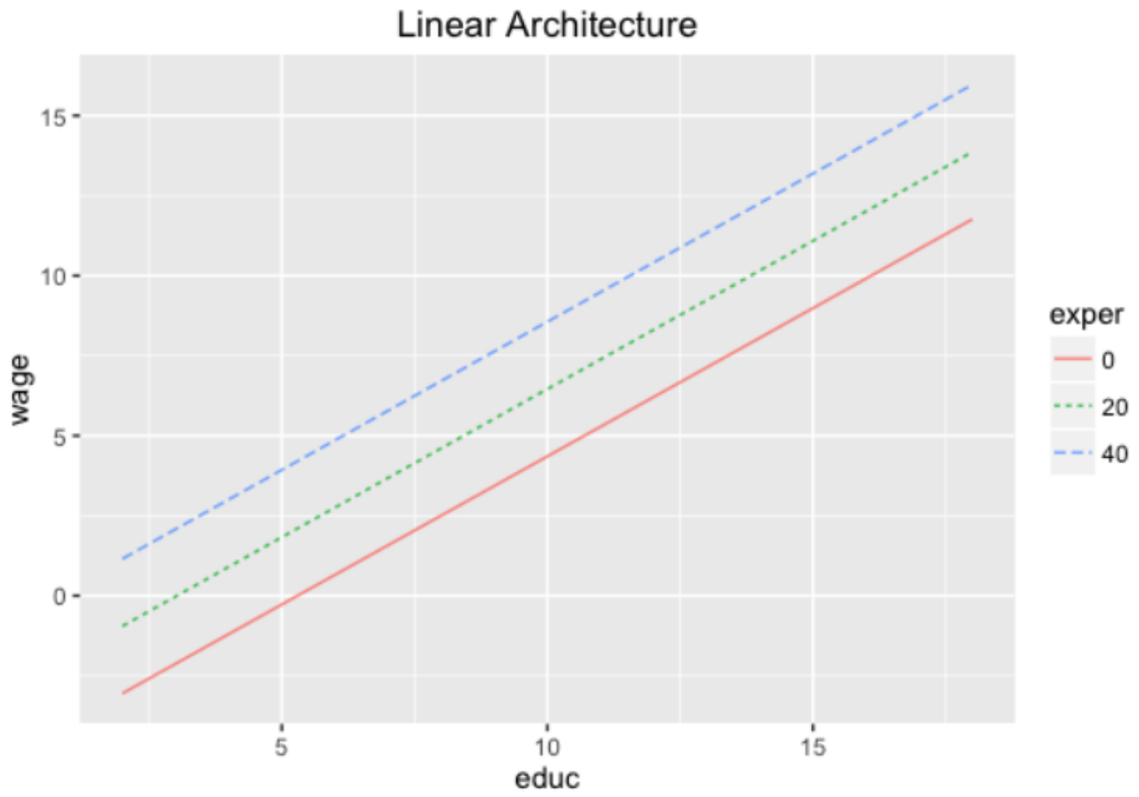
```
model_1 <- lm(wage ~ educ + exper, data = CPS85)
```

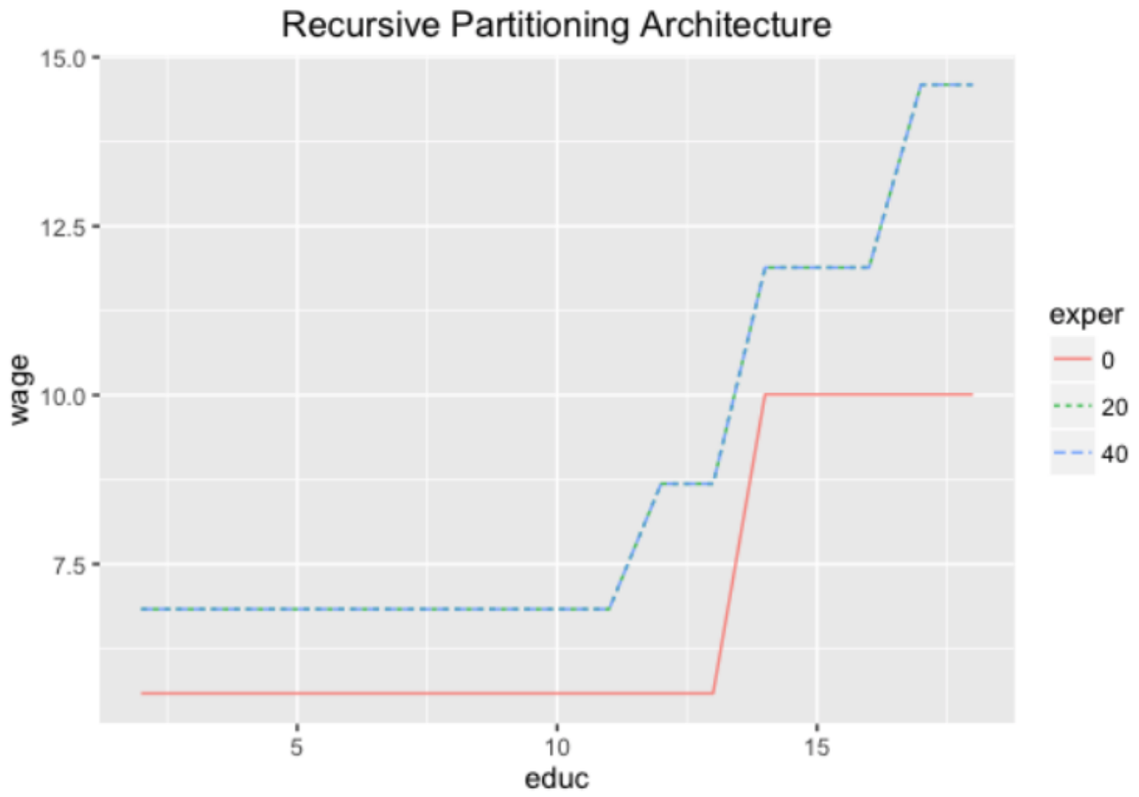
```
Call:
lm(formula = wage ~ educ + exper, data = CPS85)
Coefficients:
(Intercept)         educ        exper
    -4.9045       0.9260       0.1051
```

```
model_2 <- rpart(wage ~ educ + exper, data = CPS85)
```

```
n= 534
node), split, n, deviance, yval
      * denotes terminal node
...
```

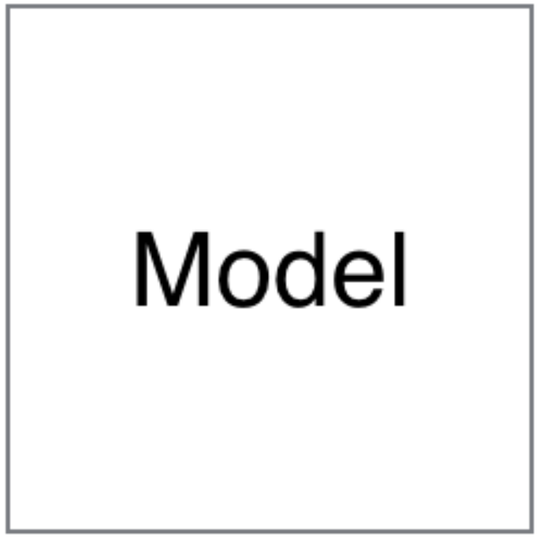
A graphical view





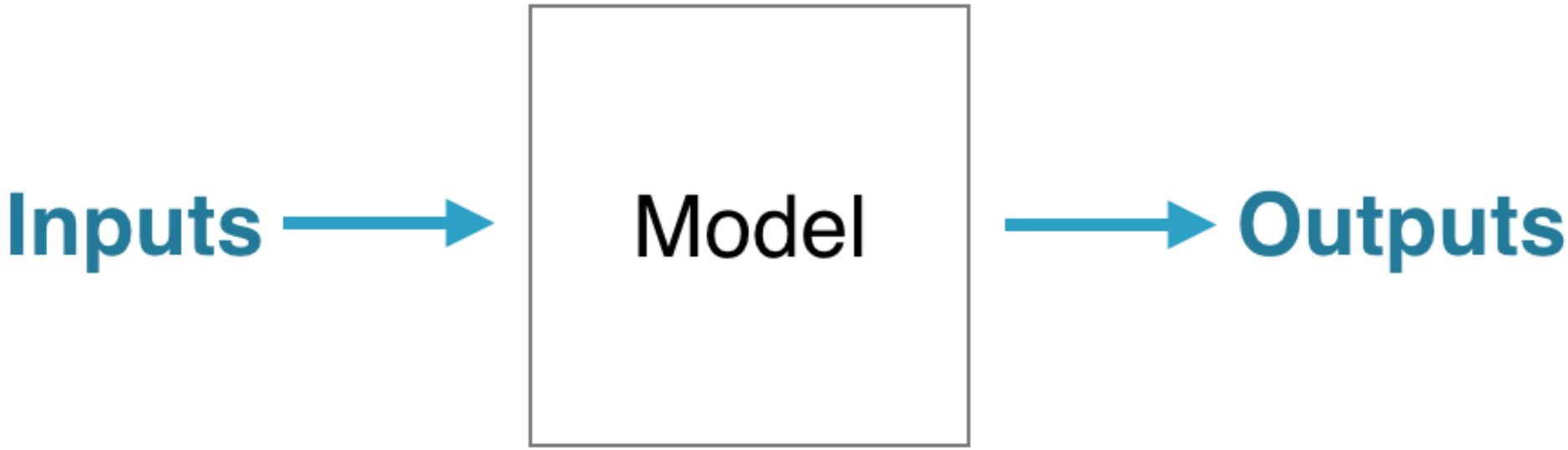
Evaluating a model

- Provide inputs for explanatory variable(s)
- Calculate the corresponding output



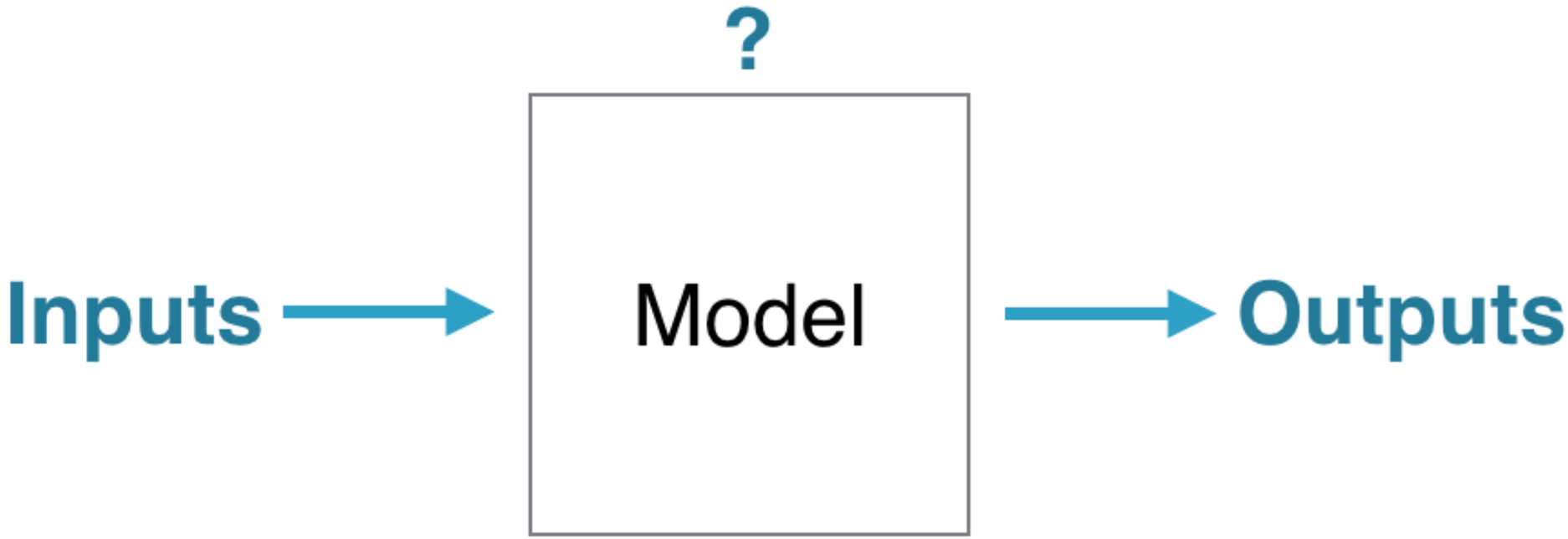
Evaluating a model

- Provide inputs for explanatory variable(s)
- Calculate the corresponding output



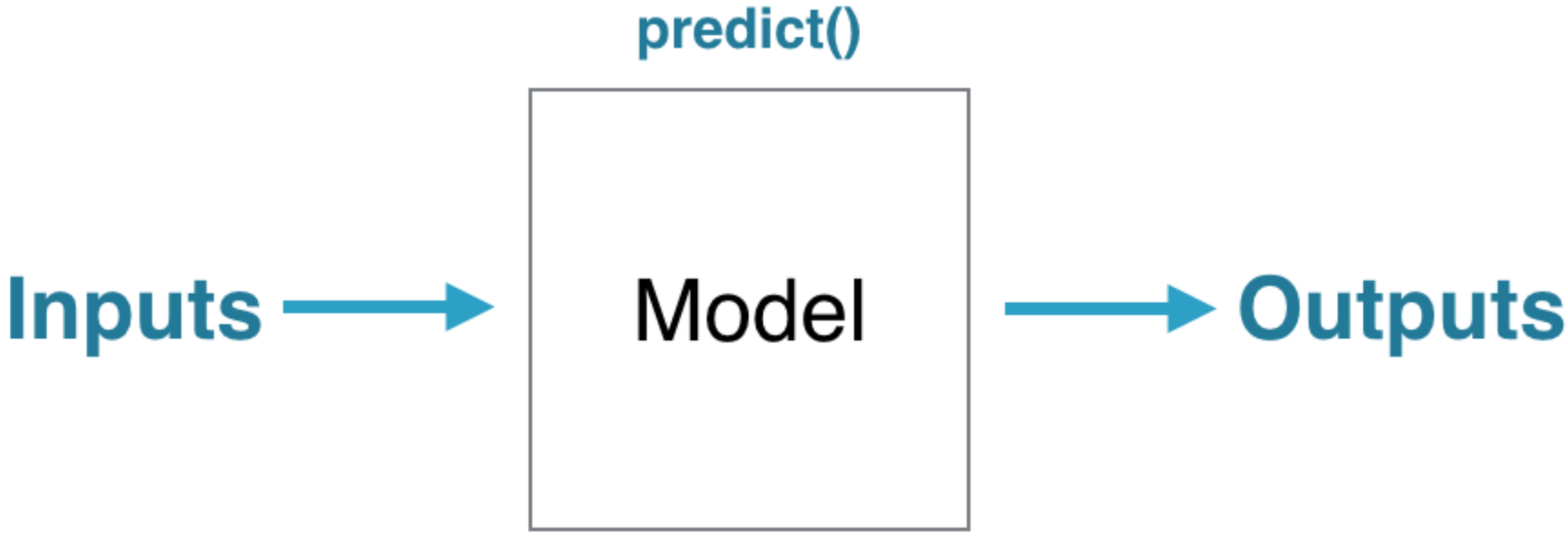
Evaluating a model

- Provide inputs for explanatory variable(s)
- Calculate the corresponding output



Evaluating a model

- Provide inputs for explanatory variable(s)
- Calculate the corresponding output



Evaluating a model

```
new_input <- data.frame(educ = 10:14, exper = 5)
new_input
```

```
  educ exper
1   10     5
2   11     5
3   12     5
4   13     5
5   14     5
```

Evaluating a model

```
predict(model_1, newdata = new_input)
```

```
       1        2        3        4        5
4.880822 5.806787 6.732751 7.658716 8.584680
```

```
predict(model_2, newdata = new_input)
```

```
       1        2        3        4        5
 5.586098  5.586098  5.586098  5.586098 10.009221
```

How good is the model?

One criterion: are the model outputs right?

```
prediction_1 <- predict(model_1, newdata = CPS85)


prediction_2 <- predict(model_2, newdata = CPS85)
```

How close is the model output?

- Actual wage values: `CPS85$wage`
- Compare to find the prediction error

```
output1 <- CPS85$wage - prediction_1 # Linear model
head(output1)
```

```
        1         2         3         4         5         6
 1.806283 -2.809725 -2.827620  1.244090  4.587642 -3.749505
```

```
output2 <- CPS85$wage - prediction_2 # Recursive partitioning
head(output2)
```

```
        1         2         3         4         5         6
 2.166623 -3.188111 -1.786098  1.811889  6.311889 -2.886829
```

Let's practice!

INTRODUCTION TO STATISTICAL MODELING IN R