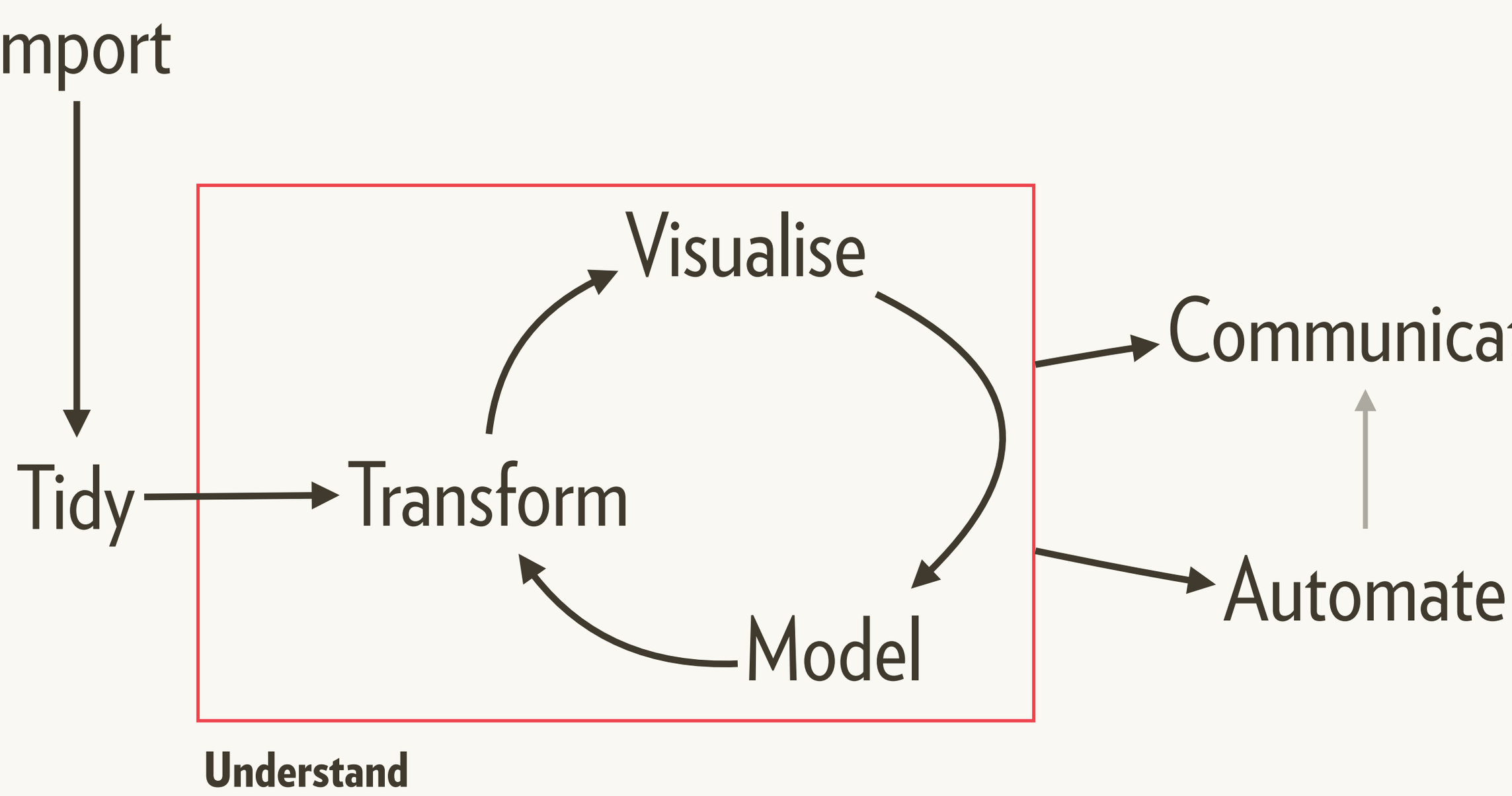
Managing many models

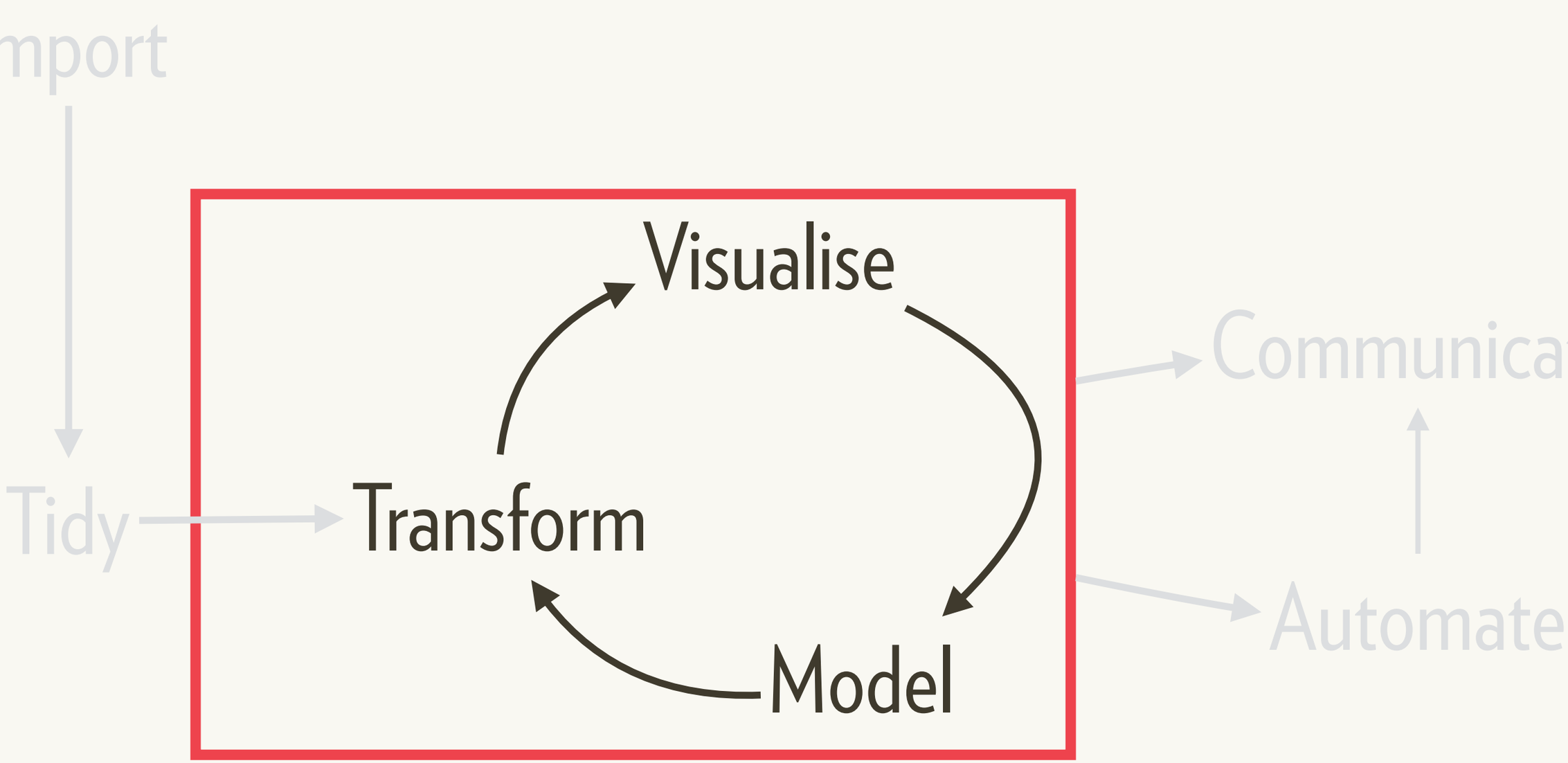
February 2016

Hadley Wickham
@hadleywickham
Chief Scientist, **RStudio**

There are 7 key components of data science



Today I want to focus on **understanding**



Exploratory data analysis

Gapminder data

142 countries

80

60

40

lifeExp

1950

1960

1970

1980

1990

2000

year

One way to handle is to fit a model to each country

New Zealand

year	lifeEx
1952	69.4
1957	70.3
1962	71.2
1967	71.5
…	…

```
lm(lifeExp ~ year, data = nz)
```

R^2=0.95

glance

tidy

Intercept	-307.7
Slope	0.19

augment

year	resid
1952	0.70
1957	0.61
1962	0.63
1967	-0.05
…	…

Broom, by David Robinson, makes this easy!

To do that for many countries, we need a list of data frames

	Year	LifeEx
Afghanistan	1952	28.9
Afghanistan	1957	30.3
Afghanistan	...	...
Albania	1952	55.2
Albania	1957	59.3
Albania	...	...
Algeria	...	...
...	...	

A nested data frame has one row per group

	Data
Afghanistan	‹data›
Albania	‹data›
Algeria	‹data›
...	‹data›

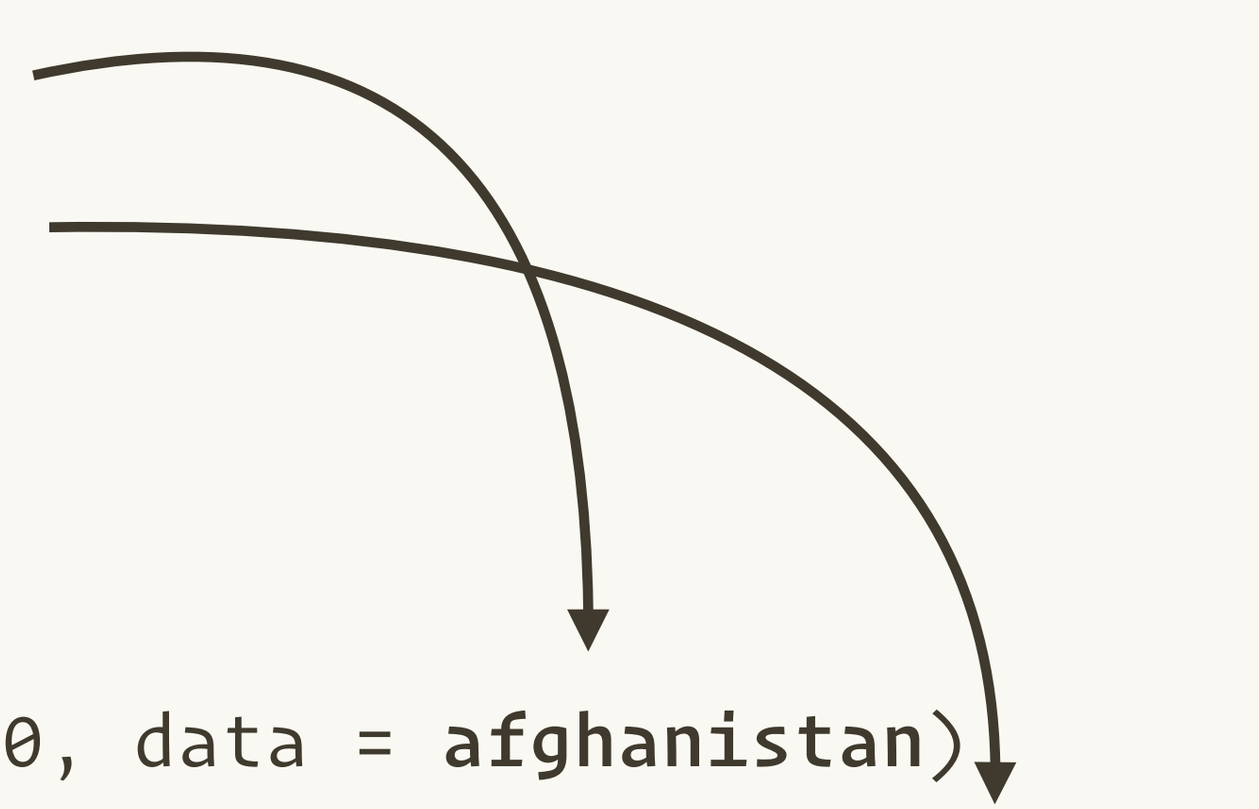
Year	LifeExp
1952	28.9
1957	30.3
...	...

Year	LifeExp
1952	55.2
1957	59.3
...	...

We can use purrr::map() to fit each model

```
map(by_country$data, ~ lm(year1950 ~ year, data = .))
```

	Data
Afghanistan	<data>
Albania	<data>
Algeria	<data>
...	<data>



```
lm(lifeExp ~ year1950, data = afghanistan)
```

```
lm(lifeExp1950 ~ year, data = albania)
```

Why for loops are bad

An digression with cupcakes

Why for loops

are ~~bad~~ suboptimal

An digression with cupcakes

Vanilla cupcakes

The hummingbird bakery cookbook

1 cup flour
a scant ¾ cup sugar
1 ½ t baking powder
3 T unsalted butter
½ cup whole milk
1 egg
¼ t pure vanilla extract

Preheat oven to 350°F.

Put the flour, sugar, baking powder, salt, and butter in a freestanding electric mixer with a paddle attachment and beat on slow speed until you get a sandy consistency and everything is combined.

Whisk the milk, egg, and vanilla together in a pitcher, then slowly pour about half into the flour mixture, beat to combine, and turn the mixer up to high speed to get rid of any lumps.

Turn the mixer down to a slower speed and slowly pour in the remaining milk mixture. Continue mixing for a couple of more minutes until the batter is smooth but do not overmix.

Spoon the batter into paper cases until 2/3 full and bake in the preheated oven for 20-25 minutes, or until the cake bounces back when touched.

Chocolate cupcakes

The hummingbird bakery cookbook

¾ cup + 2T flour
2 ½ T cocoa powder
a scant ¾ cup sugar
1 ½ t baking powder
3 T unsalted butter
½ cup whole milk
1 egg
¼ t pure vanilla extract

Preheat oven to 350°F.

Put the flour, cocoa, sugar, baking powder, salt, and butter in a freestanding electric mixer with a paddle attachment and beat on slow speed until you get a sandy consistency and everything is combined.

Whisk the milk, egg, and vanilla together in a pitcher, then slowly pour about half into the flour mixture, beat to combine, and turn the mixer up to high speed to get rid of any lumps.

Turn the mixer down to a slower speed and slowly pour in the remaining milk mixture. Continue mixing for a couple of more minutes until the batter is smooth but do not overmix.

Spoon the batter into paper cases until 2/3 full and bake in the preheated oven for 20-25 minutes, or until the cake bounces back when touched.

Chocolate cupcakes

The hummingbird bakery cookbook

¾ cup + 2T flour
2 ½ T cocoa powder
a scant ¾ cup sugar
1 ½ t baking powder
3 T unsalted butter
½ cup whole milk
1 egg
¼ t pure vanilla extract

Preheat oven to 350°F.

Put the flour, cocoa, sugar, baking powder, salt, and butter in a freestanding electric mixer with a paddle attachment and beat on slow speed until you get a sandy consistency and everything is combined.

Whisk the milk, egg, and vanilla together in a pitcher, then slowly pour about half into the flour mixture, beat to combine, and turn the mixer up to high speed to get rid of any lumps.

Turn the mixer down to a slower speed and slowly pour in the remaining milk mixture. Continue mixing for a couple of more minutes until the batter is smooth but do not overmix.

Spoon the batter into paper cases until 2/3 full and bake in the preheated oven for 20-25 minutes, or until the cake bounces back when touched.

For loops bury the lede

```
df <- data.frame(...)

means <- double(ncol(df))
for(i in seq_along(df)) {
  means[[i]] <- mean(x[[i]], na.rm = TRUE)
}

medians <- double(ncol(df))
for(i in seq_along(df)) {
  median[[i]] <- median(x[[i]], na.rm = TRUE)
}
```

For loops bury the lede

```
df <- data.frame(...)

means <- double(ncol(df))
for(i in seq_along(df)) {
  means[[i]] <- mean(x[[i]], na.rm = TRUE)
}

medians <- double(ncol(df))
for(i in seq_along(df)) {
  median[[i]] <- median(x[[i]], na.rm = TRUE)
}
```

Vanilla cupcakes

1 cup flour
a scant ¾ cup sugar
1 ½ t baking powder
3 T unsalted butter
½ cup whole milk
1 egg
¼ t pure vanilla extract

Preheat oven to 350°F.

Put the flour, sugar, baking powder, salt, and butter in a freestanding electric mixer with a paddle attachment and beat on slow speed until you get a sandy consistency and everything is combined.

Whisk the milk, egg, and vanilla together in a pitcher, then slowly pour about half into the flour mixture, beat to combine, and turn the mixer up to high speed to get rid of any lumps.

Turn the mixer down to a slower speed and slowly pour in the remaining milk mixture. Continue mixing for a couple of more minutes until the batter is smooth but do not overmix.

Spoon the batter into paper cases until 2/3 full and bake in the preheated oven for 20-25 minutes, or until the cake bounces back when touched.

Vanilla cupcakes

The hummingbird bakery cookbook

120g flour
140g sugar
1.5 t baking powder
40g unsalted butter
120ml milk
1 egg
0.25 t pure vanilla extract

Preheat oven to 170°C.

Put the flour, sugar, baking powder, salt, and butter in a freestanding electric mixer with a paddle attachment and beat on slow speed until you get a sandy consistency and everything is combined.

Whisk the milk, egg, and vanilla together in a pitcher, then slowly pour about half into the flour mixture, beat to combine, and turn the mixer up to high speed to get rid of any lumps.

Turn the mixer down to a slower speed and slowly pour in the remaining milk mixture. Continue mixing for a couple of more minutes until the batter is smooth but do not overmix.

Spoon the batter into paper cases until 2/3 full and bake in the preheated oven for 20-25 minutes, or until the cake bounces back when touched.

1. Convert units

Vanilla cupcakes

The hummingbird bakery cookbook

120g flour
140g sugar
1.5 t baking powder
40g butter
120ml milk
1 egg
0.25 t vanilla

Beat flour, sugar, baking powder, salt, and butter until sandy.

Whisk milk, egg, and vanilla. Mix half into flour mixture until smooth (use high speed). Beat in remaining half. Mix until smooth.

Bake 20-25 min at 170°C.

2. Rely on domain knowledge

For loops emphasise the data

```
df <- data.frame(...)

means <- double(ncol(df))
for(i in seq_along(df)) {
  means[[i]] <- mean(x[[i]], na.rm = TRUE)
}

medians <- double(ncol(df))
for(i in seq_along(df)) {
  median[[i]] <- median(x[[i]], na.rm = TRUE)
}
```

Purrr emphasises the action

```
library(purrr)
means <- map_dbl(df, mean)
medians <- map_dbl(df, median)
```

Vanilla cupcakes

The hummingbird bakery cookbook

120g flour
140g sugar
1.5 t baking powder
40g butter
120ml milk
1 egg
0.25 t vanilla

Beat dry ingredients + butter until sandy.

Whisk together wet ingredients. Mix half into dry until smooth (use high speed). Beat in remaining half. Mix until smooth.

Bake 20-25 min at 170°C.

3. Use variables

Cupcakes

Beat dry ingredients + butter until sandy.

Whisk together wet ingredients. Mix half into dry until smooth (use high speed). Beat in remaining half. Mix until smooth.

Bake 20-25 min at 170°C.

Vanilla

120g flour

140g sugar

1.5t baking powder

40g butter

120ml milk

1 egg

0.25 t vanilla

Chocolate

100g flour

20g cocoa

140g sugar

1.5t baking powder

40g butter

120ml milk

1 egg

0.25 t vanilla

4. Extract out common code

Similarly, purrr lets you create more complex recipes

```
df <- data.frame(...)

col_sum <- function(df, f) {
  df %>%
    keep(is_numeric) %>%
    map_dbl(f)
}

means <- col_sum(df, mean)
medians <- col_sum(df, median)
```

Similarly, purrr lets you create more complex recipes

```
df <- data.frame(...)

col_sum <- function(df, f) {
  map_dbl(keep(df, is_numeric), f)
}

means <- col_sum(df, mean)
medians <- col_sum(df, median)
```

Cupcakes

	Flour	Baking powder	Sugar	Butter	Egg	Extra
Vanilla	120	1.5	140	40	1	0.25t vanilla
Chocolate	100	1.5	140	40	1	20g cocoa • 0.25t vanilla
Lemon	120	1.5	140	40	1	2T lemon zest
Red velvet	150	0	150	60	1	10g cocoa • 20ml red colouring • 1.5t vinegar • 0.5 t baking soda

5. Store as data

In R, we can store functions in lists

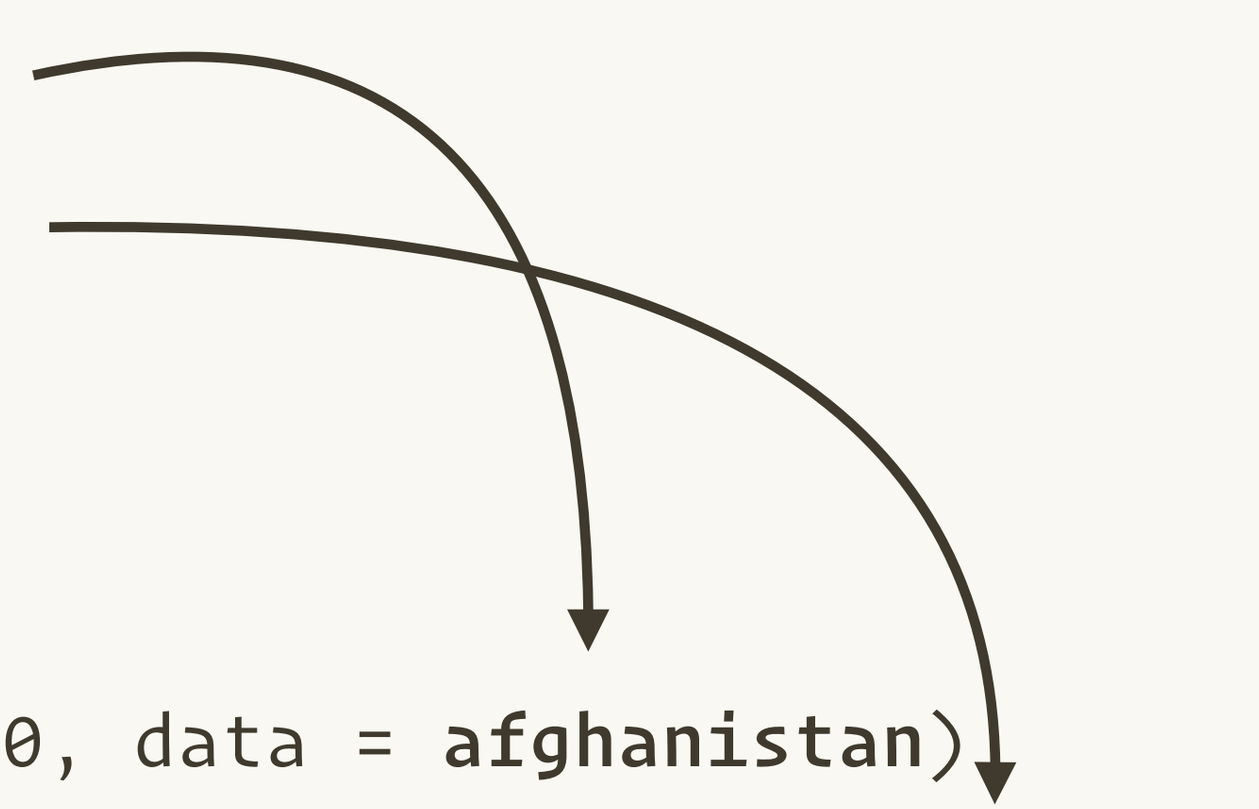
```
funs <- list(
  mean = mean,
  median = median,
  sd = sd
)

map(funs, col_sum, df = df)
```

Back to gapminder

We can use purrr::map() to fit each model

```
map(by_country$data, ~ lm(year1950 ~ year, data = .))
```

	Data
Afghanistan	<data>
Albania	<data>
Algeria	<data>
...	<data>

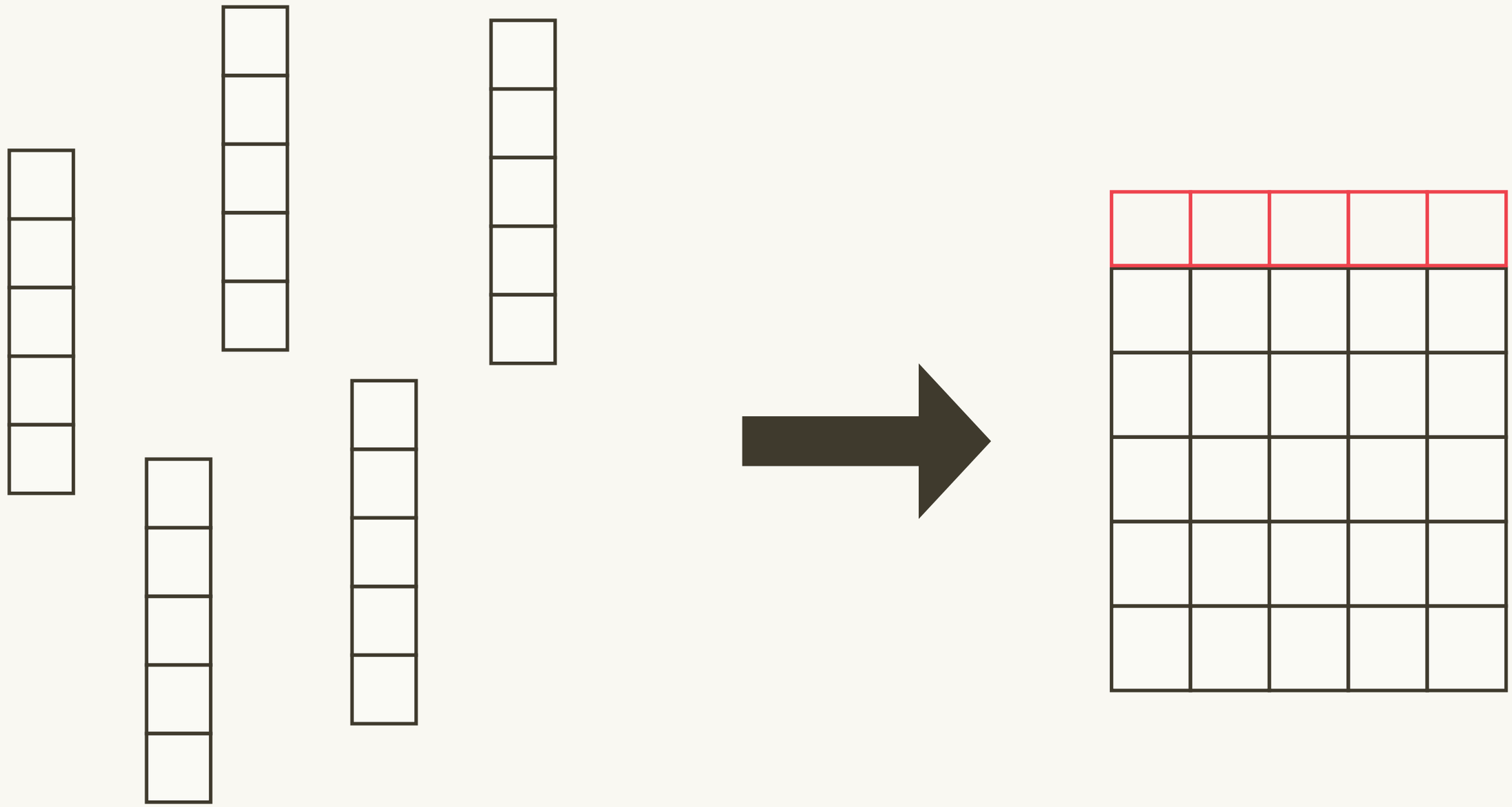


```
lm(lifeExp ~ year1950, data = afghanistan)
```

```
lm(lifeExp1950 ~ year, data = albania)
```

```r
map(by_country$data, ~ lm(year1950 ~ year, data = .))

# same as

out <- vector("list", length(by_country$data))
for (i in seq_along(by_country$data)) {
  df <- by_country$data[[i]]
  out[[i]] <- lm(year1950 ~ year, data = df)
}
```

Multiple lists make it easy to lose context



So use a data frame!

Unnesting is reverse of nesting

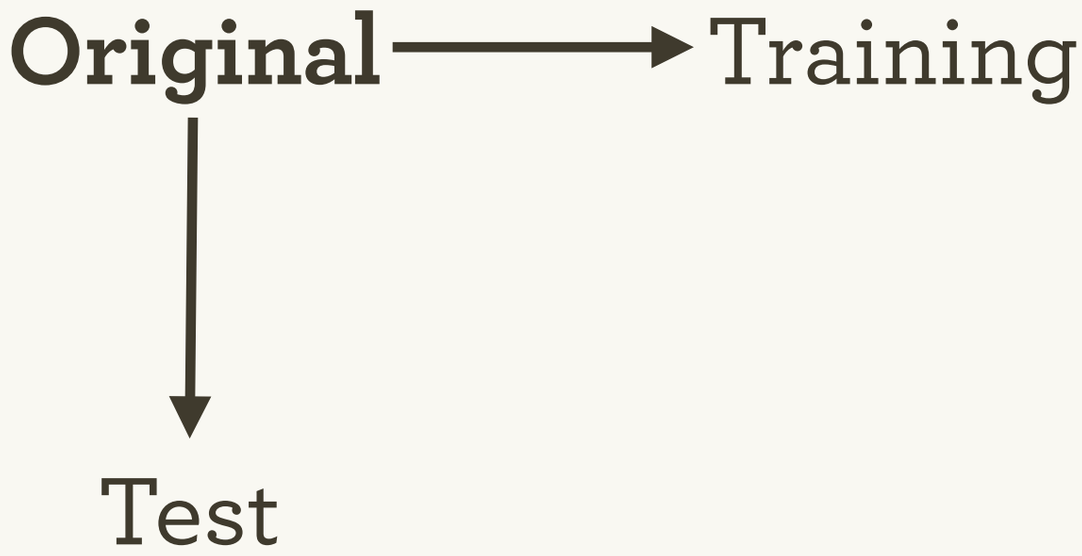
nest()

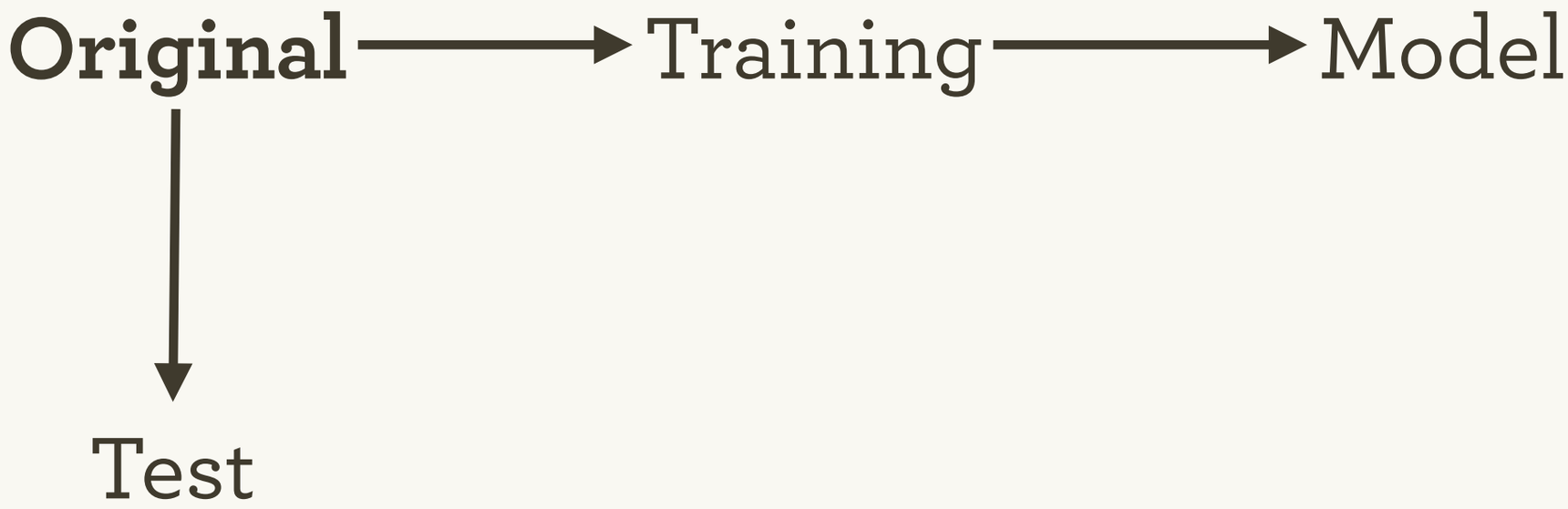
	Data
Afghanistan	<data>
Albania	<data>
Algeria	<data>
...	<data>

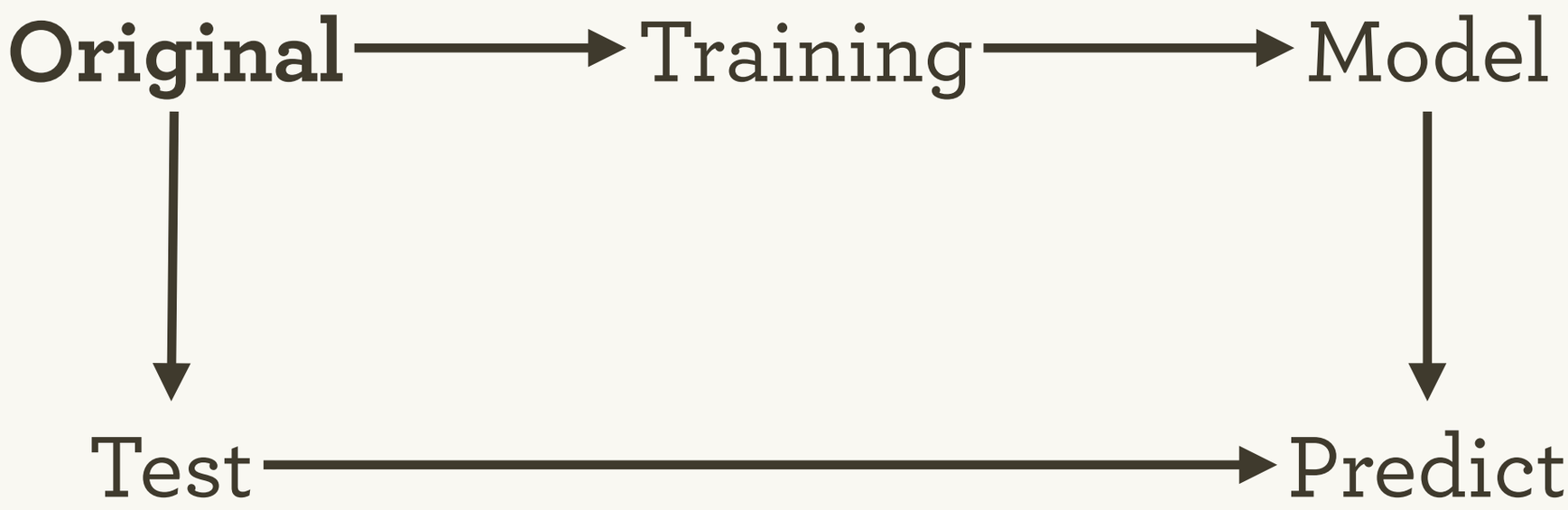
unnest()

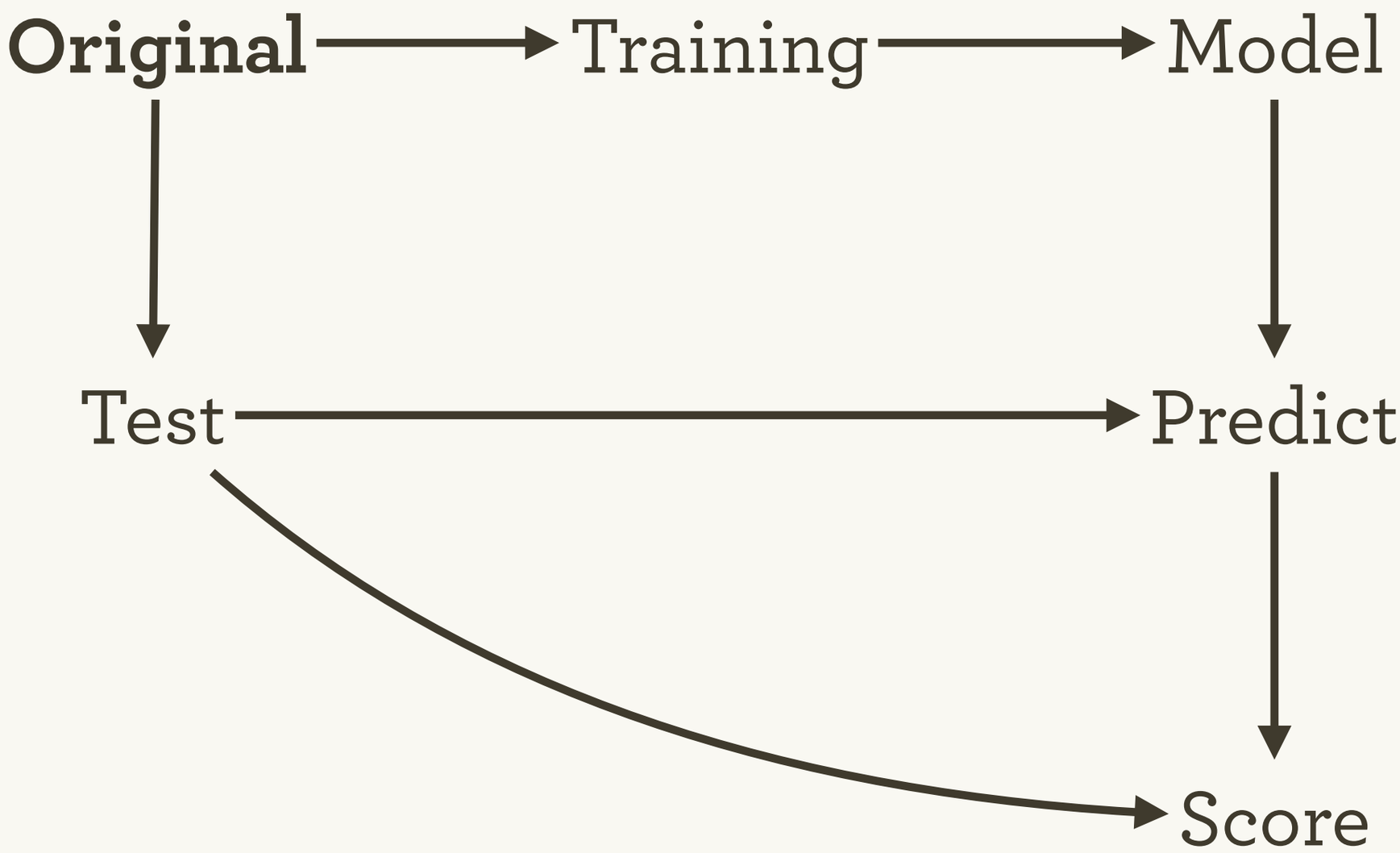
	Year	LifeEx
Afghanistan	1952	28.9
Afghanistan	1957	30.3
Afghanistan	...	...
Albania	1952	55.2
Albania	1957	59.3
Albania	...	...
Algeria	...	...
...	...	

Cross-validation


Original
Training
Test


Original
Training
Model
Test


Original
Training
Model
Test
Predict

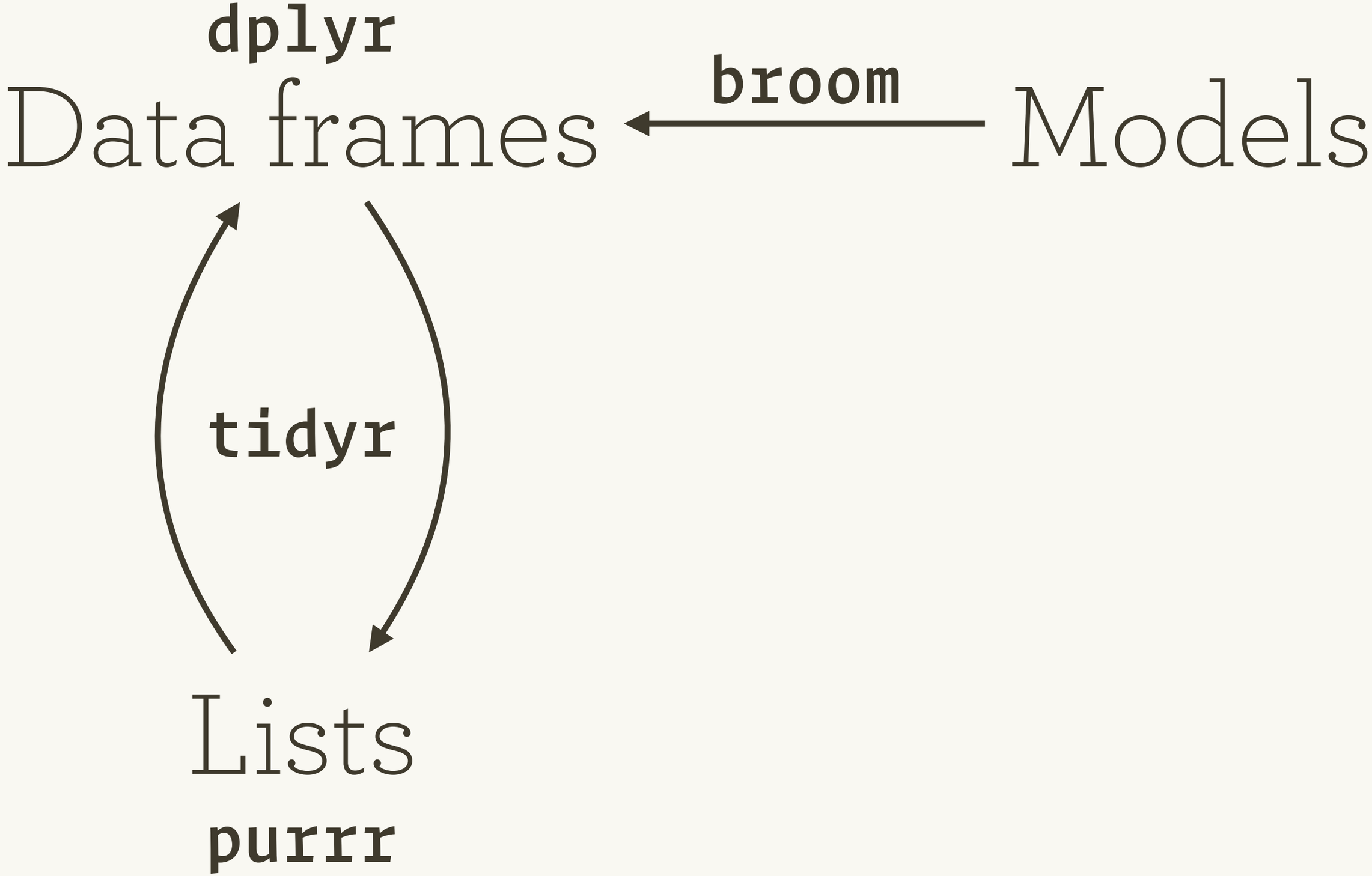

Original
Training
Model
Test
Predict
Score

	Test	Training	Model	Prediction	Score
1	df	df	lm	vector	number
2	df	df	lm	vector	number
3	df	df	lm	vector	number
4	df	df	lm	vector	number
	...	...	...	...	...

```r
crossv <- partition(mtcars, 100, c(
  test = 0.2,
  training = 0.8
))

crossv <- crossv %>% mutate(
  # Fit the models
  model = map(training, ~ lm(mpg ~ wt, data = .)),
  # Make predictions on test data
  pred =  map2(model, test, predict),
  # Evaluate difference between predicted
  diff =  map2_dbl(pred, test %>% map("mpg"), msd)
)
```

Conclusion

1. Store related objects in **list-columns**.
2. Learn **FP** so you can focus on verbs, not objects.
3. Use **broom** to convert models to tidy data.


dplyr
Data frames
broom
Models
tidyr
Lists
purrr