# Pivot longer

# wider vs. longer

wider

| | id | city | hwy |
|---|---|---|---|
| 1 | car1 | 19 | 24 |
| 2 | car2 | 20 | 30 |
| 3 | car3 | 29 | 35 |

goal: lengthen

longer

| | id | roadtype | mpg |
|---|---|---|---|
| 1 | car1 | city | 19 |
| 2 | car2 | city | 20 |
| 3 | car3 | city | 29 |
| 4 | car1 | hwy | 24 |
| 5 | car2 | hwy | 30 |
| 6 | car3 | hwy | 35 |

# Why?

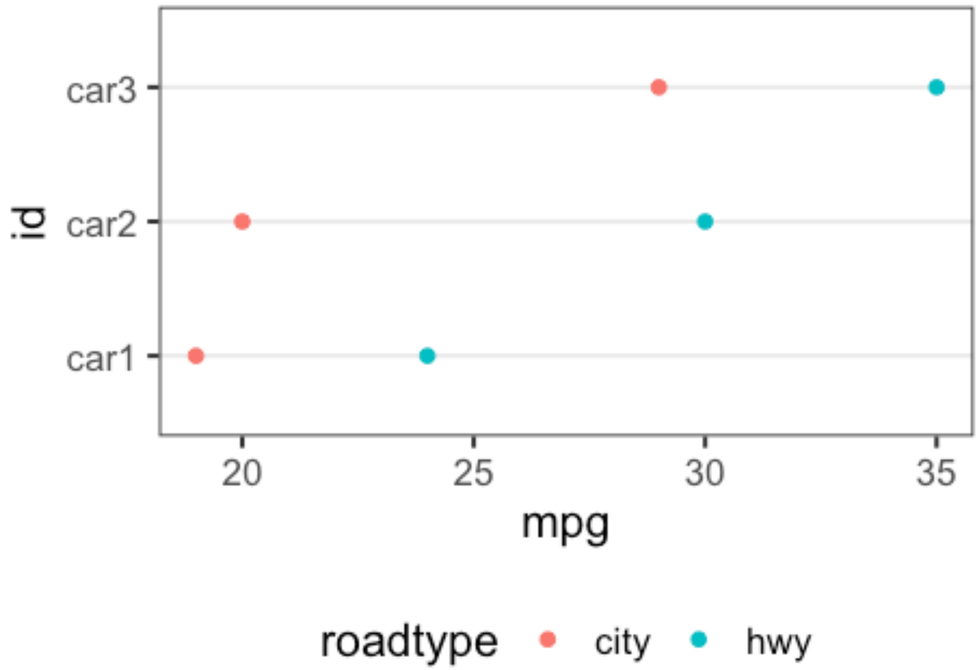
Because we want to create a graph in which "city" and "hwy" are different colors.

longer

| | id | roadtype | mpg |
|---|---|---|---|
| 1 | car1 | city | 19 |
| 2 | car2 | city | 20 |
| 3 | car3 | city | 29 |
| 4 | car1 | hwy | 24 |
| 5 | car2 | hwy | 30 |
| 6 | car3 | hwy | 35 |

# Why?

```
1 ggplot(df, aes(x = mpg, y = id, color = roadtype)) +
2   geom_point() + ...
```



# wider vs. longer

wider

| | id | city | hwy |
|---|---|---|---|
| 1 | car1 | 19 | 24 |
| 2 | car2 | 20 | 30 |
| 3 | car3 | 29 | 35 |

goal: lengthen

longer

| | id | roadtype | mpg |
|---|---|---|---|
| 1 | car1 | city | 19 |
| 2 | car2 | city | 20 |
| 3 | car3 | city | 29 |
| 4 | car1 | hwy | 24 |
| 5 | car2 | hwy | 30 |
| 6 | car3 | hwy | 35 |

# Step 1: picture the new data frame

**What columns should the new data frame have?**

| id | city | hwy |
|---|---|---|
| car1 | 19 | 24 |
| car2 | 20 | 30 |
| car3 | 29 | 35 |

| id | name | value |
|---|---|---|
| car1 | city | 19 |
| car2 | city | 20 |
| car3 | city | 29 |
| car1 | hwy | 24 |
| car2 | hwy | 30 |
| car3 | hwy | 35 |

# Step 2: identify the columns to be pivoted

**city and hwy will become values of a new column**

| id | city | hwy |
|---|---|---|
| car1 | 19 | 24 |
| car2 | 20 | 30 |
| car3 | 29 | 35 |

| id | name | value |
|---|---|---|
| car1 | city | 19 |
| car2 | city | 20 |
| car3 | city | 29 |
| car1 | hwy | 24 |
| car2 | hwy | 30 |
| car3 | hwy | 35 |

**That's it!**

# tidyr code

df

| id | city | hwy |
|---|---|---|
| car1 | 19 | 24 |
| car2 | 20 | 30 |
| car3 | 29 | 35 |

```
library(tidyr)
pivot_longer(df, cols = city:hwy))
```

# tidyr code

```
library(tidyr)
pivot_longer(df, cols = city:hwy)
#> # A tibble: 6 × 3
#>   id    name  value
#>   <chr> <chr> <dbl>
#> 1 car1  city     19
#> 2 car1  hwy      24
#> 3 car2  city     20
#> 4 car2  hwy      30
#> 5 car3  city     29
#> 6 car3  hwy      35
```

# Optional: choose names for the new columns

```
pivot_longer(df, cols = city:hwy,
             names_to = "roadtype", values_to = "mpg")
#> # A tibble: 6 × 3
#>   id    roadtype   mpg
#>   <chr> <chr>    <dbl>
#> 1 car1  city        19
#> 2 car1  hwy         24
#> 3 car2  city        20
#> 4 car2  hwy         30
#> 5 car3  city        29
#> 6 car3  hwy         35
```

"roadtype" and "mpg" do not exist as columns in the original data frame

# What happened?

| id | city | hwy |
|---|---|---|
| car1 | 19 | 24 |
| car2 | 20 | 30 |
| car3 | 29 | 35 |

| id | name | value |
|---|---|---|
| car1 | city | 19 |
| car2 | city | 20 |
| car3 | city | 29 |
| car1 | hwy | 24 |
| car2 | hwy | 30 |
| car3 | hwy | 35 |

old column names become *values of* name

old cell values move to single value column

# No id column


```
df <- data.frame(city = c(19, 20, 29),
                 hwy = c(24, 30, 35))
df
#>   city hwy
#> 1   19  24
#> 2   20  30
#> 3   29  35
```

# No id column


```
pivot_longer(df, cols = city:hwy)
#> # A tibble: 6 × 2
#>   name  value
#>   <chr> <dbl>
#> 1 city     19
#> 2 hwy      24
#> 3 city     20
#> 4 hwy      30
#> 5 city     29
#> 6 hwy      35
```

We have no way of knowing which rows are connected

# Solution: turn the rownames into a column

```
df <- data.frame(city = c(19, 20, 29),
                 hwy = c(24, 30, 35))
df$id <- rownames(df)
df
#>   city hwy id
#> 1   19  24  1
#> 2   20  30  2
#> 3   29  35  3
```

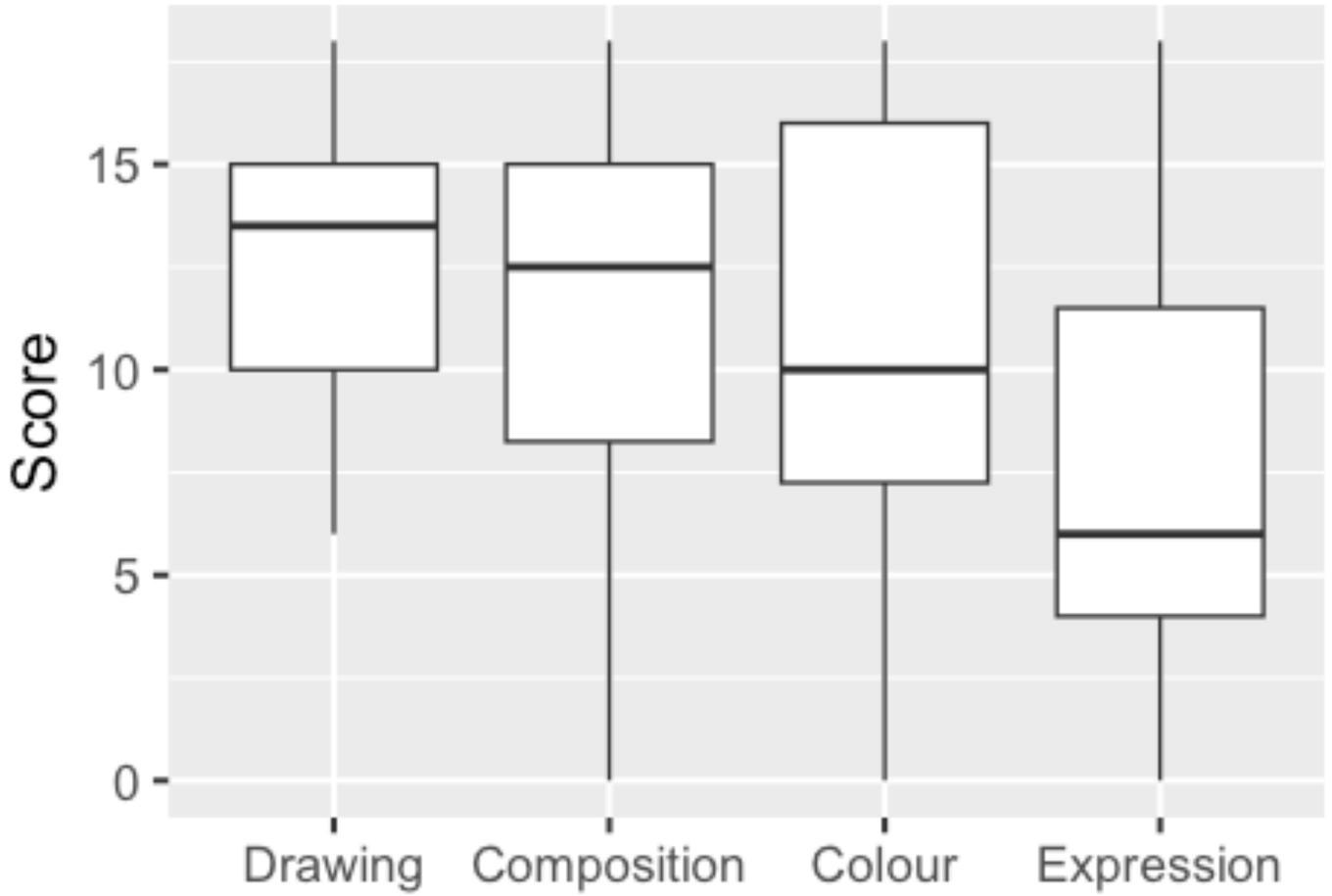
# id column from rownames

```
pivot_longer(df, cols = city:hwy)
#> # A tibble: 6 × 3
#>   id    name  value
#>   <chr> <chr> <dbl>
#> 1 1     city     19
#> 2 1     hwy      24
#> 3 2     city     20
#> 4 2     hwy      30
#> 5 3     city     29
#> 6 3     hwy      35
```

# EXERCISE

```
library(MASS)
head(painters)
#>                Composition Drawing Colour Expression School
#> Da Udine                10       8     16          3      A
#> Da Vinci                15      16      4         14      A
#> Del Piombo               8      13     16          7      A
#> Del Sarto               12      16      9          8      A
#> Fr. Penni                0      15      8          0      A
#> Guilio Romano           15      16      4         14      A
```

# Goal: create boxplots



# Look at the data

```
library(MASS)
str(painters)
#> 'data.frame':    54 obs. of  5 variables:
#>  $ Composition: int  10 15 8 12 0 15 8 15 4 17 ...
#>  $ Drawing    : int  8 16 13 16 15 16 17 16 12 18 ...
#>  $ Colour     : int  16 4 16 9 8 4 4 7 10 12 ...
#>  $ Expression : int  3 14 7 8 0 14 8 6 4 18 ...
#>  $ School     : Factor w/ 8 levels
"A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

# What should the new columns be?

Current columns:

Composition | Colour | Drawing | Expression | School

New columns:

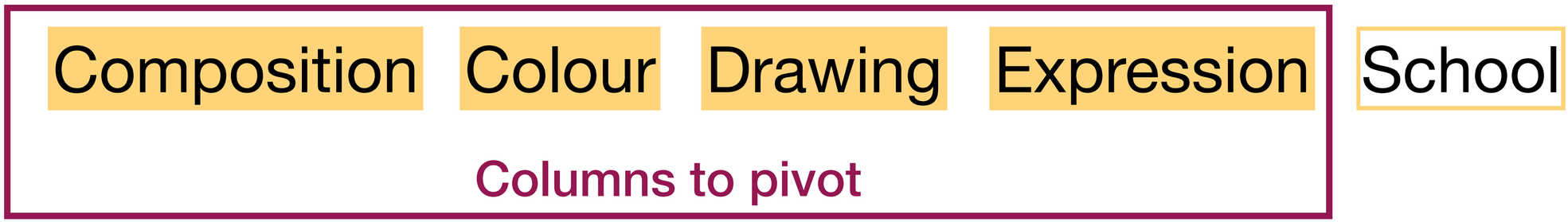
(rownames) → Name | School | Skill (name) | Score (value)

# What will the longer data frame look like?

| (rownames) → | Name | School | (name) Skill | (value) Score |
|---|---|---|---|---|
| Da Udine | | A | Composition | 10 |
| Da Vinci | | A | Composition | 15 |
| Del Piombo | | A | Composition | 8 |
| Del Sarto | | A | Composition | 12 |

# What should the new columns be?

Current columns:




New columns:

# tidyr code

```
painters$Name <- rownames(painters)
pivot_longer(painters,
    cols = Composition:Expression,
    names_to = "Skill", values_to = "Score")
#> # A tibble: 216 × 4
#>    School Name       Skill       Score
#>    <fct>  <chr>      <chr>       <int>
#>  1 A      Da Udine   Composition    10
#>  2 A      Da Udine   Drawing         8
#>  3 A      Da Udine   Colour         16
#>  4 A      Da Udine   Expression      3
```