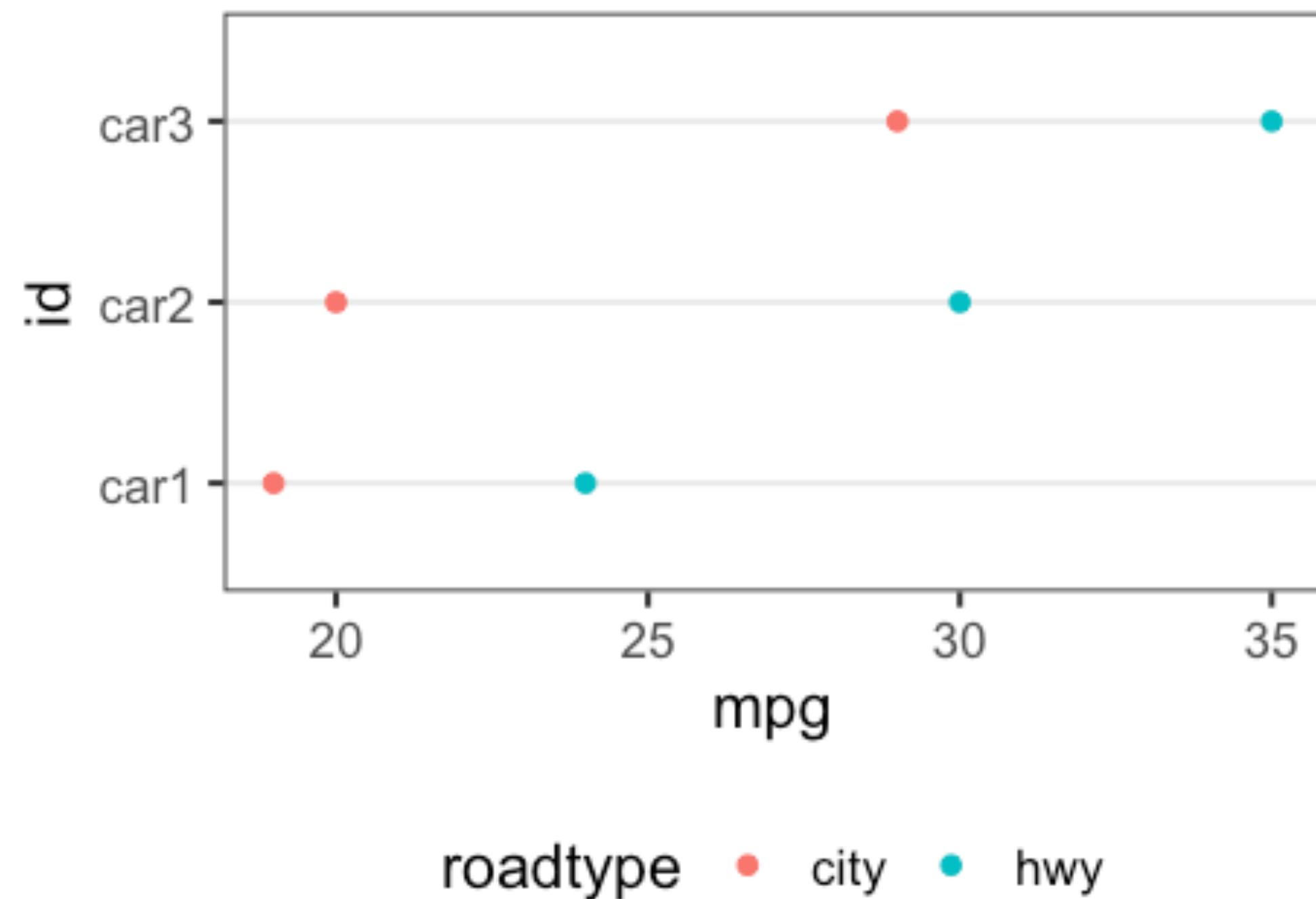


Pivot longer

`slides/03b_pivot_longer.pdf`

Problem: missing categorical column for mapping

How can you make this graph...



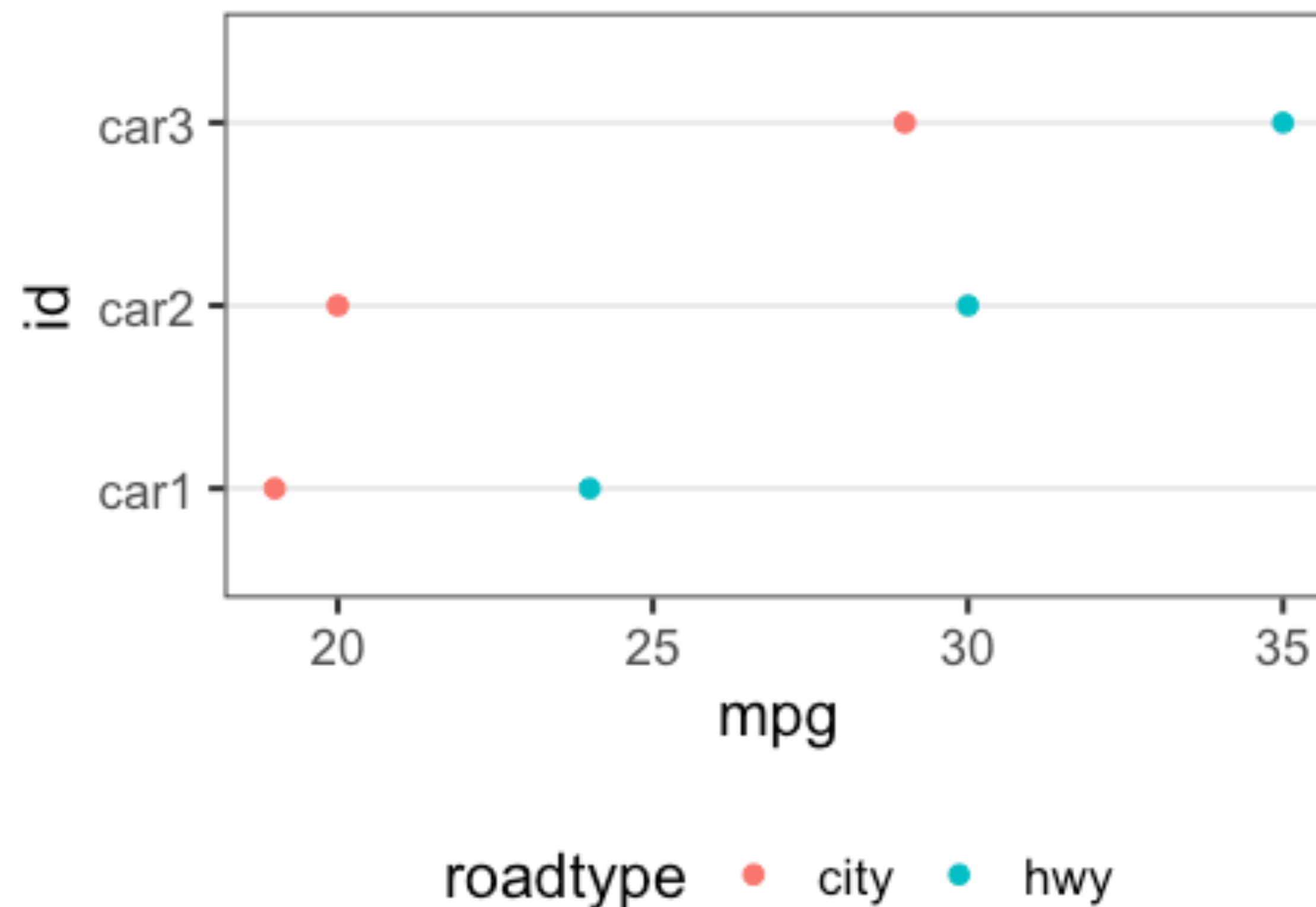
with this data?

id	city	hwy
<chr>	<dbl>	<dbl>
car1	19	24
car2	20	30
car3	29	35

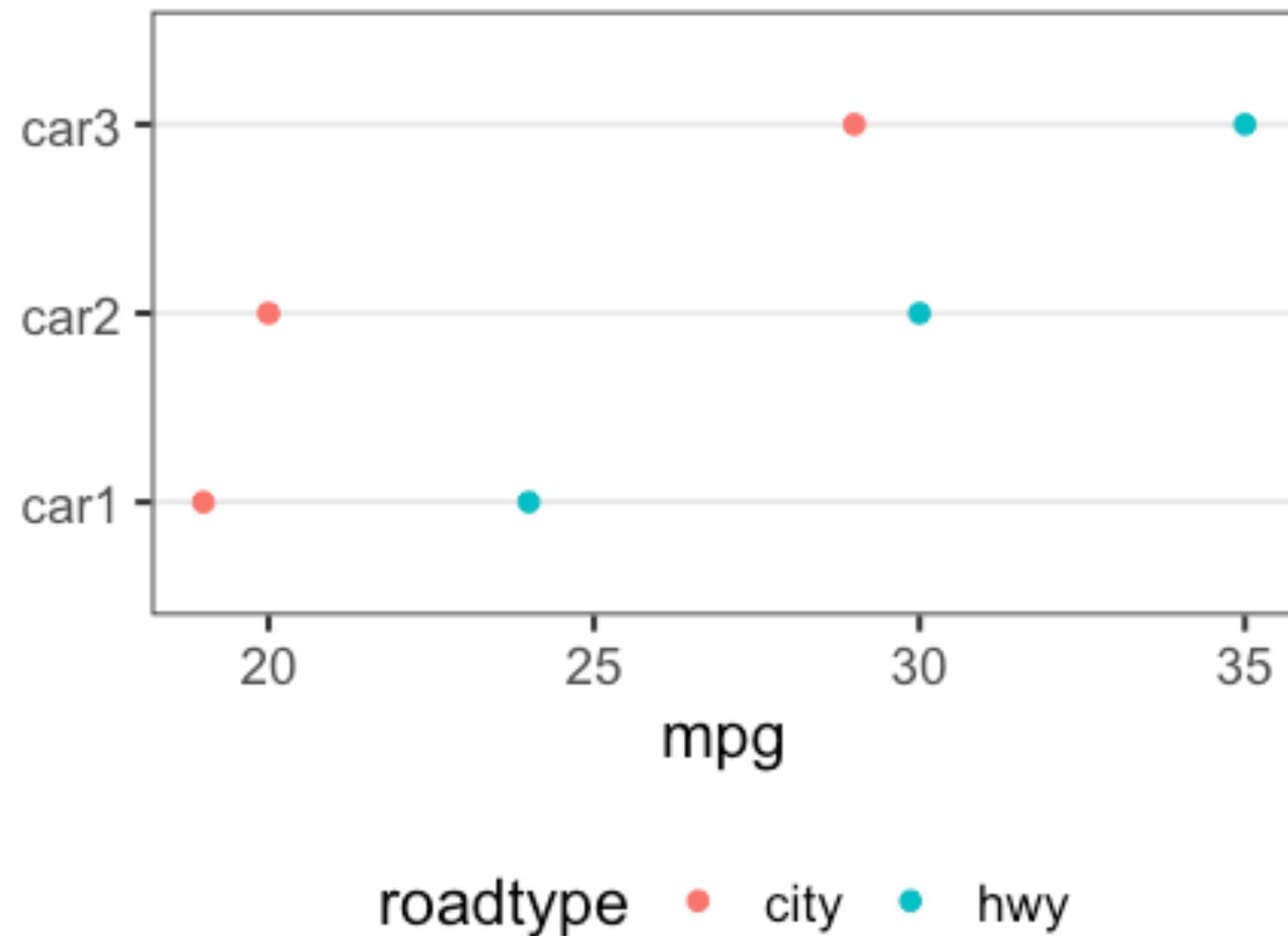
The missing column

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

Ideally we
would have a
column to
which we could
map **color**



The missing column



id	roadtype	value
<chr>	<chr>	<dbl>
car1	city	19
car2	city	20
car3	city	29
car1	hwy	24
car2	hwy	30
car3	hwy	35

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

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

	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 2: identify the columns to be pivoted

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

	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

pivot_longer

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

```
pivot_longer(dfwide, cols = city:hwy)
```

columns to be
pivoted



pivot_longer

```
pivot_longer(dfwide, 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(dfwide, 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
1	car1	19	24
2	car2	20	30
3	car3	29	35

	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

old column names
become *values of*
name column

What happened?

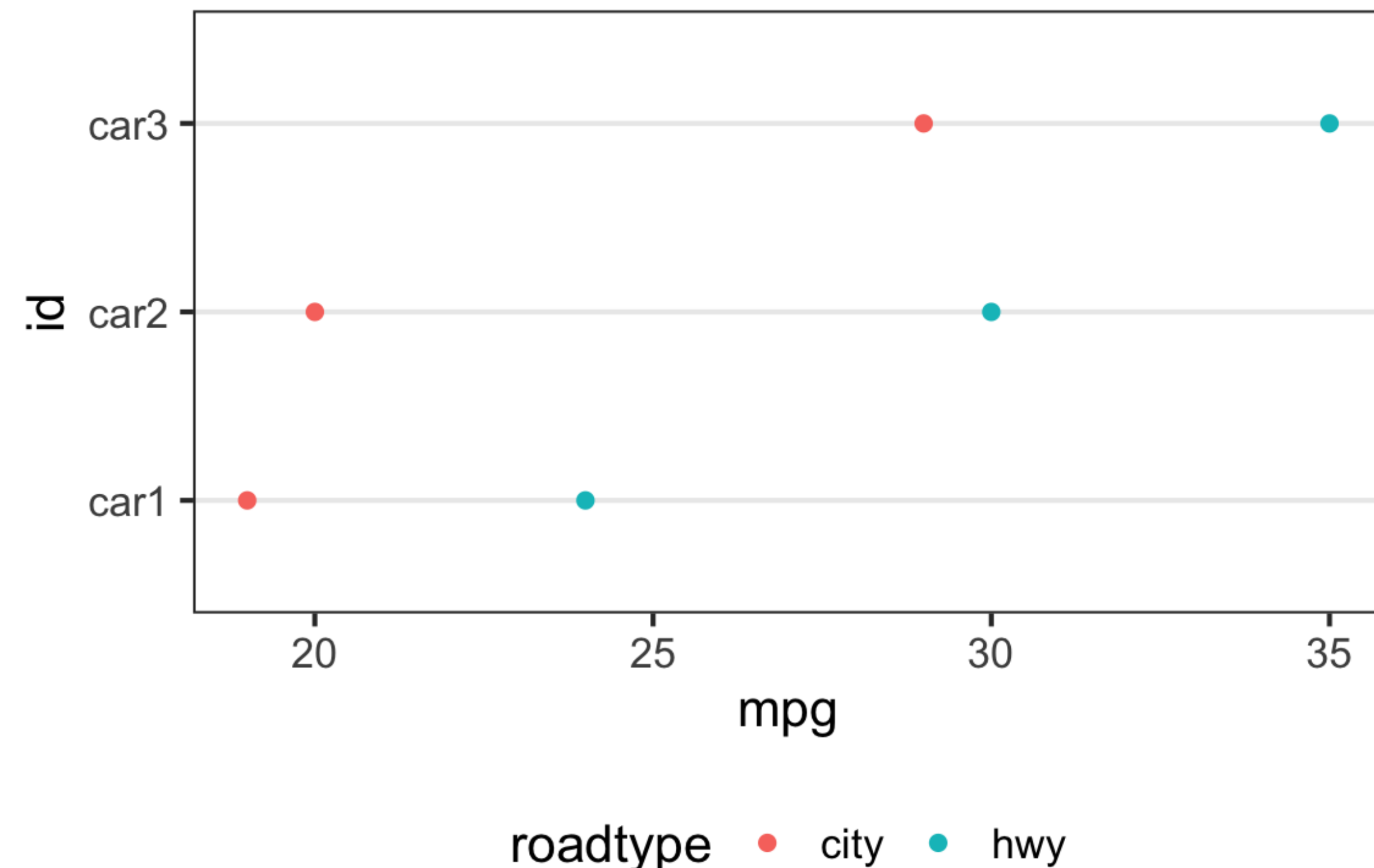
	id	city	hwy
1	car1	19	24
2	car2	20	30
3	car3	29	35

	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

old cell values
move to single
value column

Graph

```
df <- pivot_longer(dfwide, cols = city:hwy,  
                   names_to = "roadtype", values_to = "mpg")  
ggplot(df, aes(x = mpg, y = id, color = roadtype)) + ...
```

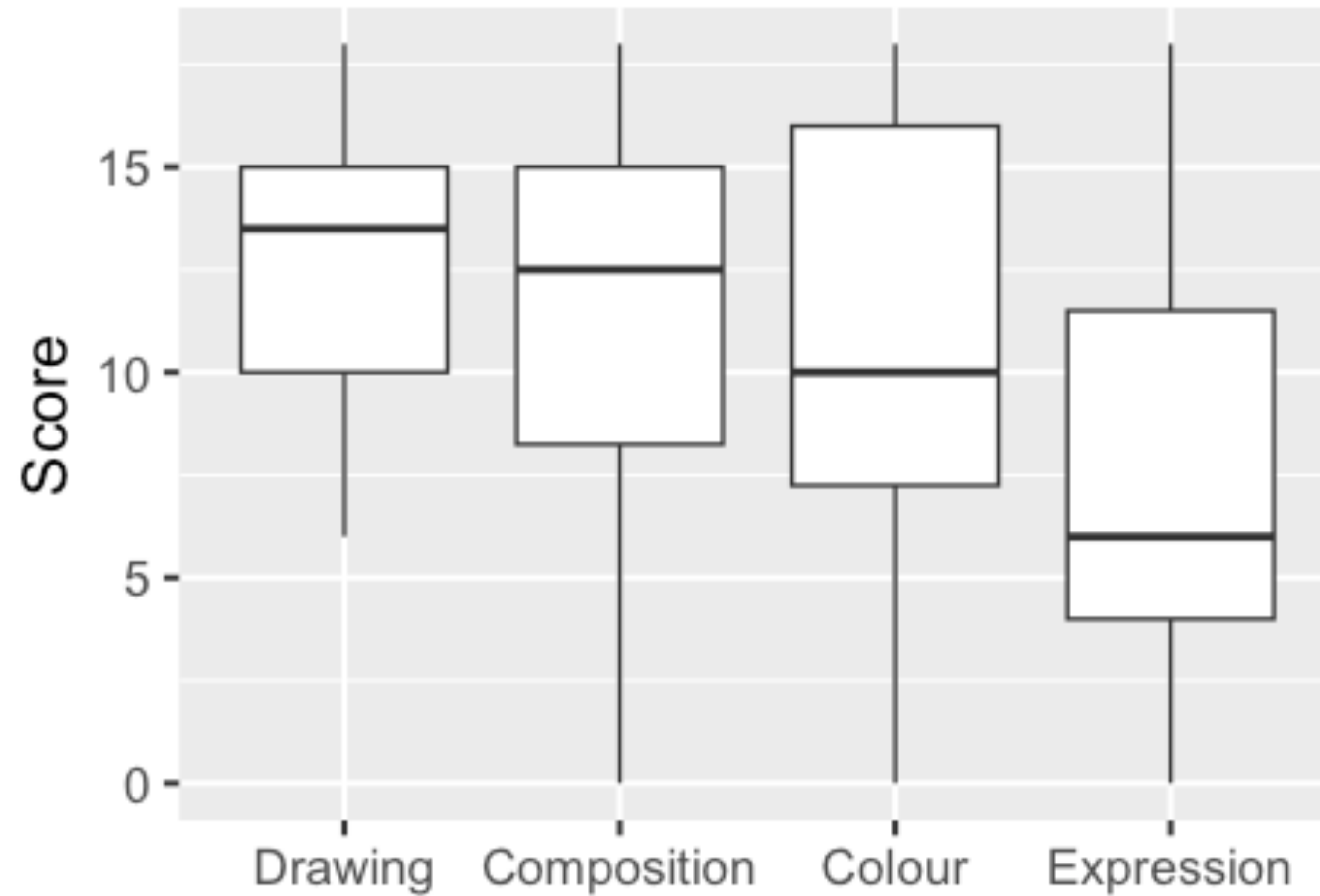


Exercise

```
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



What do we want our data to look like?

Current columns:

Composition Colour Drawing Expression School

New columns:

(name) (value)
School Skill Score

What do we want our data to look like?

	(name)	(value)
School	Skill	Score

A	Composition	10
A	Composition	15
A	Composition	8
A	Composition	12

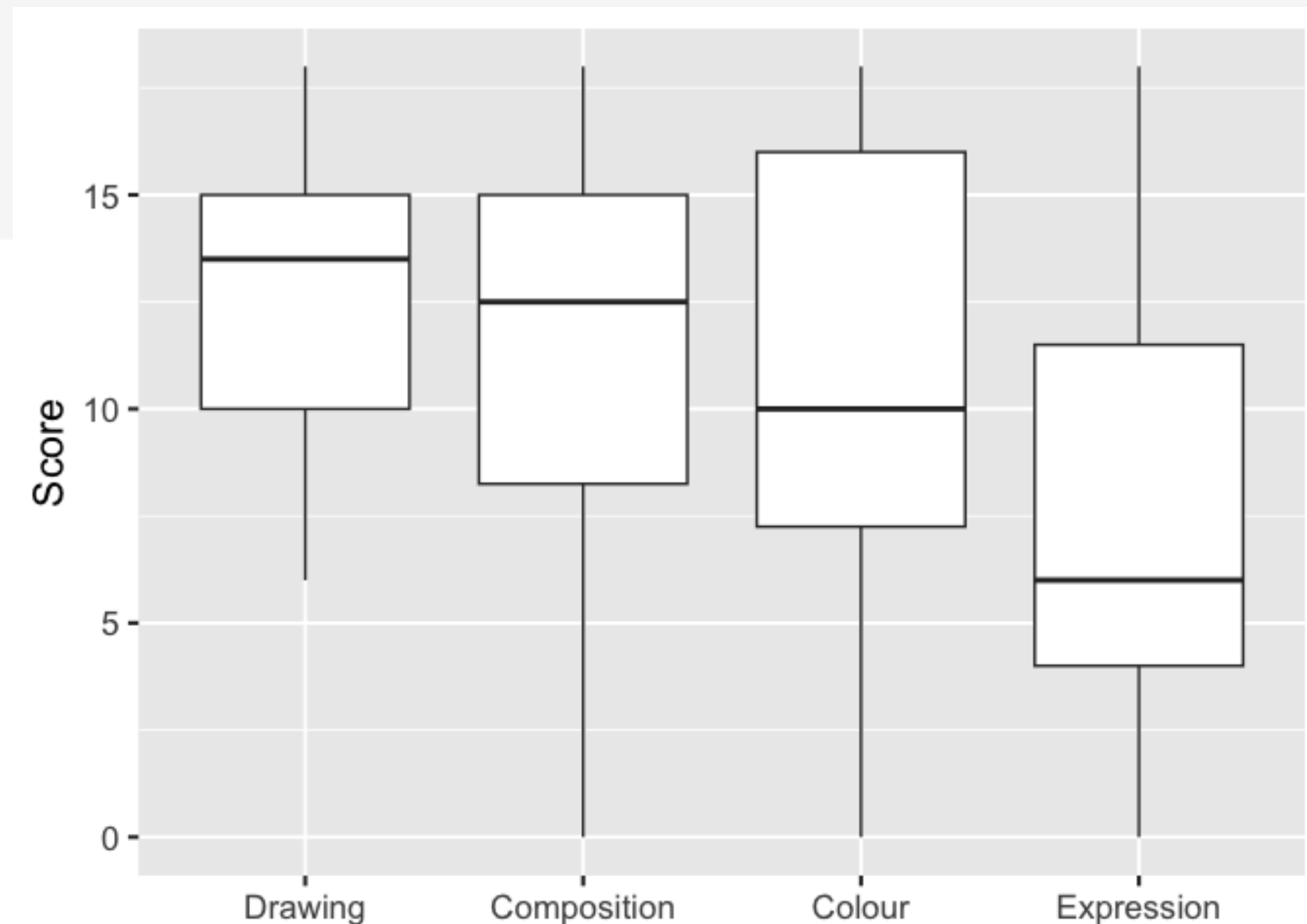
Solution: the transformed data

```
painters |>
  pivot_longer(cols = Composition:Expression,
               names_to = "Skill", values_to = "Score") |>
  head()
```

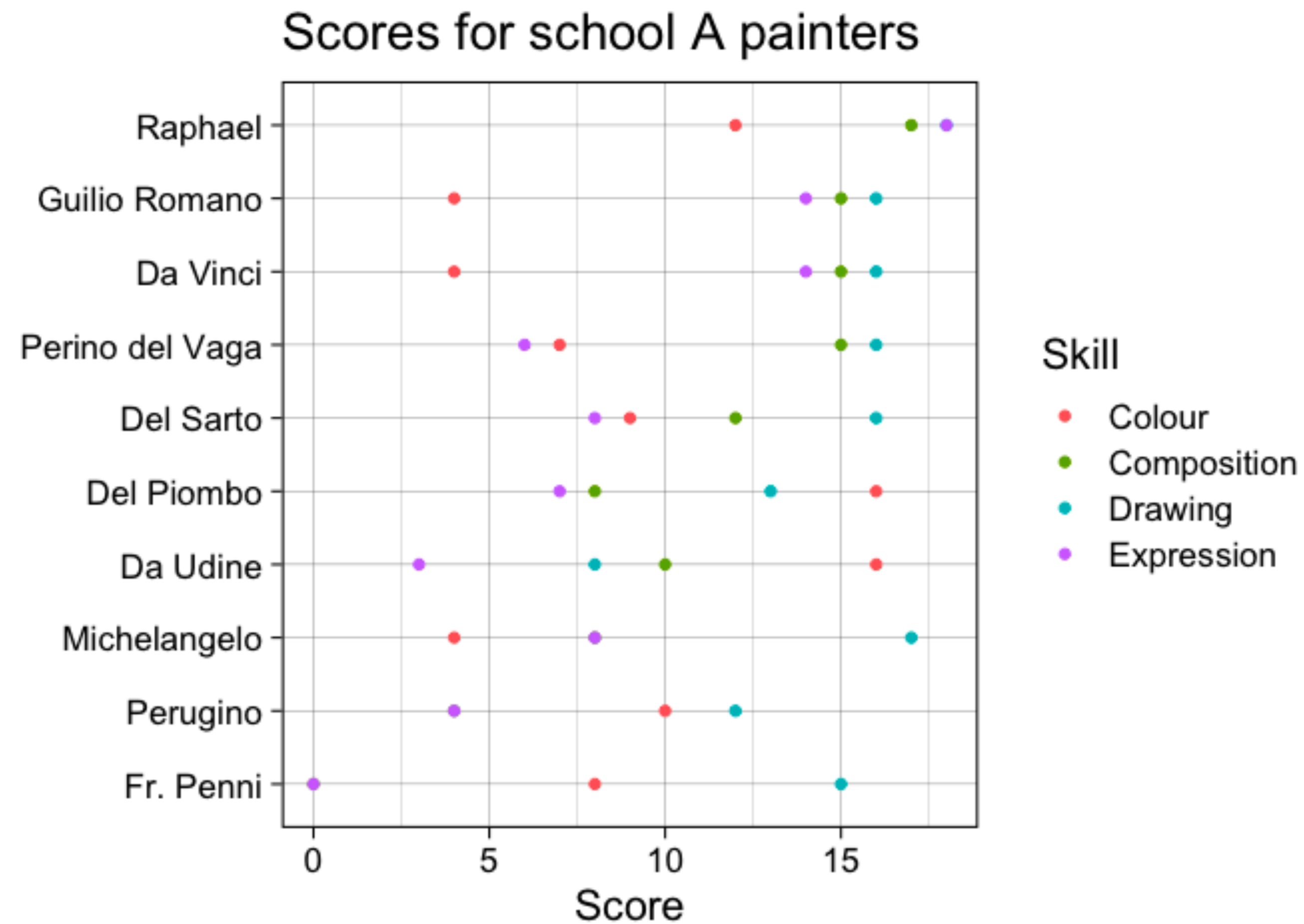
```
# A tibble: 6 × 3
  School Skill      Score
  <fct>  <chr>      <int>
1 A      Composition    10
2 A      Drawing         8
3 A      Colour        16
4 A      Expression     3
5 A      Composition    15
6 A      Drawing        16
```

Solution: pivot_longer

```
painters |>
  pivot_longer(cols = Composition:Expression,
               names_to = "Skill", values_to = "Score") |>
  ggplot(aes(x = reorder(Skill, Score, median, decreasing = TRUE),
             y = Score)) +
  geom_boxplot() +
  labs(x = NULL)
```



Goal: Cleveland dot plot

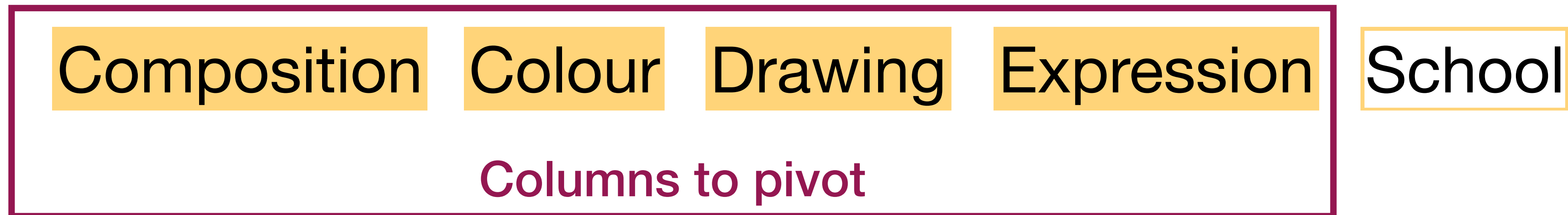


The new data: where are the painter names??

```
painters |>
  pivot_longer(cols = Composition:Expression,
               names_to = "Skill", values_to = "Score") |> head()
#> # A tibble: 6 × 3
#>   School Skill      Score
#>   <fct>  <chr>    <int>
#> 1 A      Composition  10
#> 2 A      Drawing      8
#> 3 A      Colour     16
#> 4 A      Expression   3
#> 5 A      Composition  15
#> 6 A      Drawing     16
```

What do we want our data to look like?

Original columns:



New columns:



Move the rownames into a column

```
painters |>
  filter(School == "A") |>
  rownames_to_column("Name")
#>      Name Composition Drawing Colour Expression School
#> 1 Da Udine         10      8      16          3      A
#> 2 Da Vinci         15     16       4         14      A
#> 3 Del Piombo        8     13     16          7      A
#> 4 Del Sarto        12     16       9          8      A
#> 5 Fr. Penni         0     15       8          0      A
#> 6 Giulio Romano    15     16       4         14      A
```

Tidyverse principle: don't store important info in rownames

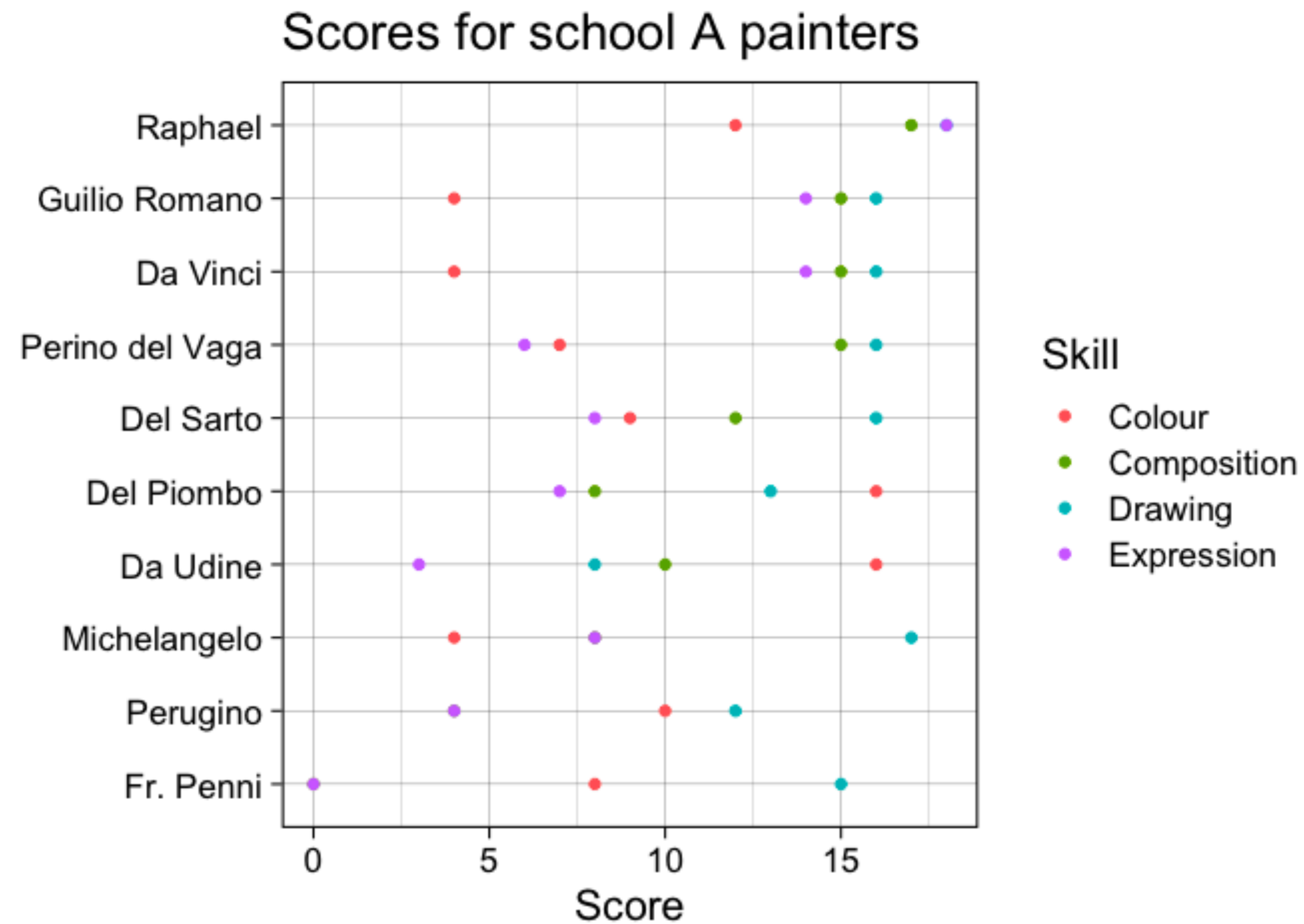
Then pivot and store the result

```
paint_long <- painters |>
  filter(School == "A") |>
  rownames_to_column("Name") |>
  pivot_longer(cols = Composition:Expression,
               names_to = "Skill", values_to = "Score")
```

Pivoted data with painter names

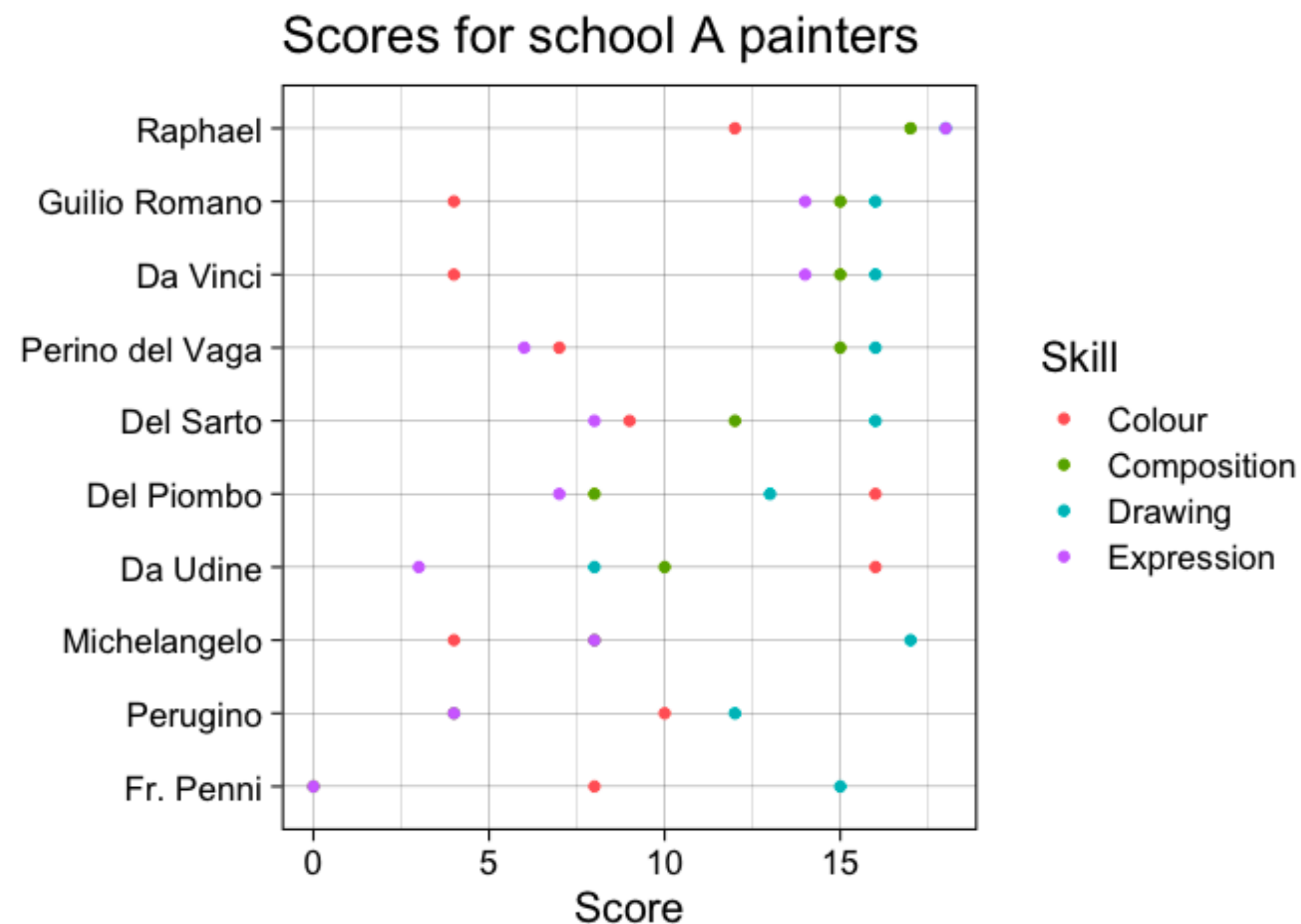
```
head(paint_long)
#> # A tibble: 6 × 4
#>   Name      School Skill      Score
#>   <chr>    <fct>   <chr>    <int>
#> 1 Da Udine A      Composition 10
#> 2 Da Udine A      Drawing      8
#> 3 Da Udine A      Colour     16
#> 4 Da Udine A      Expression   3
#> 5 Da Vinci A      Composition 15
#> 6 Da Vinci A      Drawing     16
```

Exercise 2: Cleveland dot plot



Solution

```
ggplot(paint_long, aes(x = Score, y = fct_reorder(Name, Score),  
                      color = Skill)) + geom_point() +  
  labs(title = "Scores for school A painters", y = NULL) +  
  theme_linedraw(16)
```



mappings
x = Score
y = Name
color = Skill

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

```
library(tidyr)
df <- data.frame(city = c(19, 20, 29),
                  hwy = c(24, 30, 35))
df$id <- paste0("car", rownames(df))
df
#>   city hwy   id
#> 1   19  24 car1
#> 2   20  30 car2
#> 3   29  35 car3
```

id column from rownames

```
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
```

Why?

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

