Dot plot basics

Richard Layton

2017-09-11

Dot plots can be multivariate

Dot plots begin with bivariate data. In this way they replace nearly every pie chart and bar chart, ever.

- x-axis: quantitative
- y-axis: categorical

Using *conditioning* we can display additional variables. In today's examples, we have

- no conditioning
- one additional categorical variable
- two additional categorical variables

Example 1

1970's population of US states by region

Dot plot with one conditioning variable

Base R includes data(state)

```
library(ggplot2)
library(tibble)
data(state)
```

```
glimpse(state.name)
##  chr [1:50] "Alabama" "Alaska" "Arizona" ...
```

```
glimpse(state.region)
##  Factor w/ 4 levels "Northeast","South",..: 2 4 4 2 4 4
```

```
glimpse(state.x77)
##  num [1:50, 1:8] 3615 365 2212 2110 21198 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansa
##   ..$ : chr [1:8] "Population" "Income" "Illiteracy" "Li
```

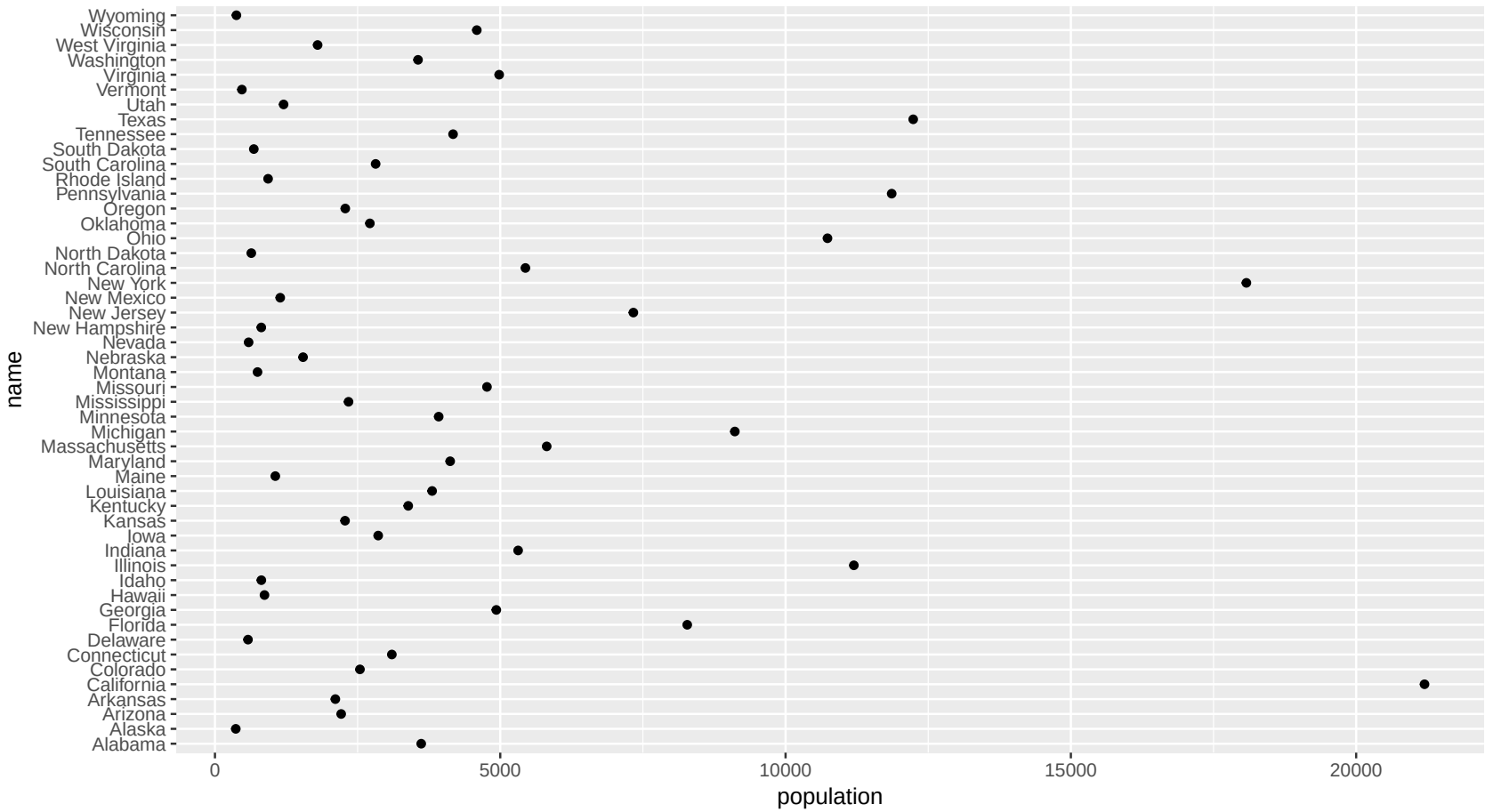
Create data frame for example

```
name       <- state.name
region     <- state.region
population <- state.x77[ , "Population"]
state_df   <- data_frame(name, population, region)
```

```
glimpse(state_df)
## Observations: 50
## Variables: 3
## $ name       <chr> "Alabama", "Alaska", "Arizon...
## $ population <dbl> 3615, 365, 2212, 2110, 21198...
## $ region     <fctr> South, West, West, South, W...
```

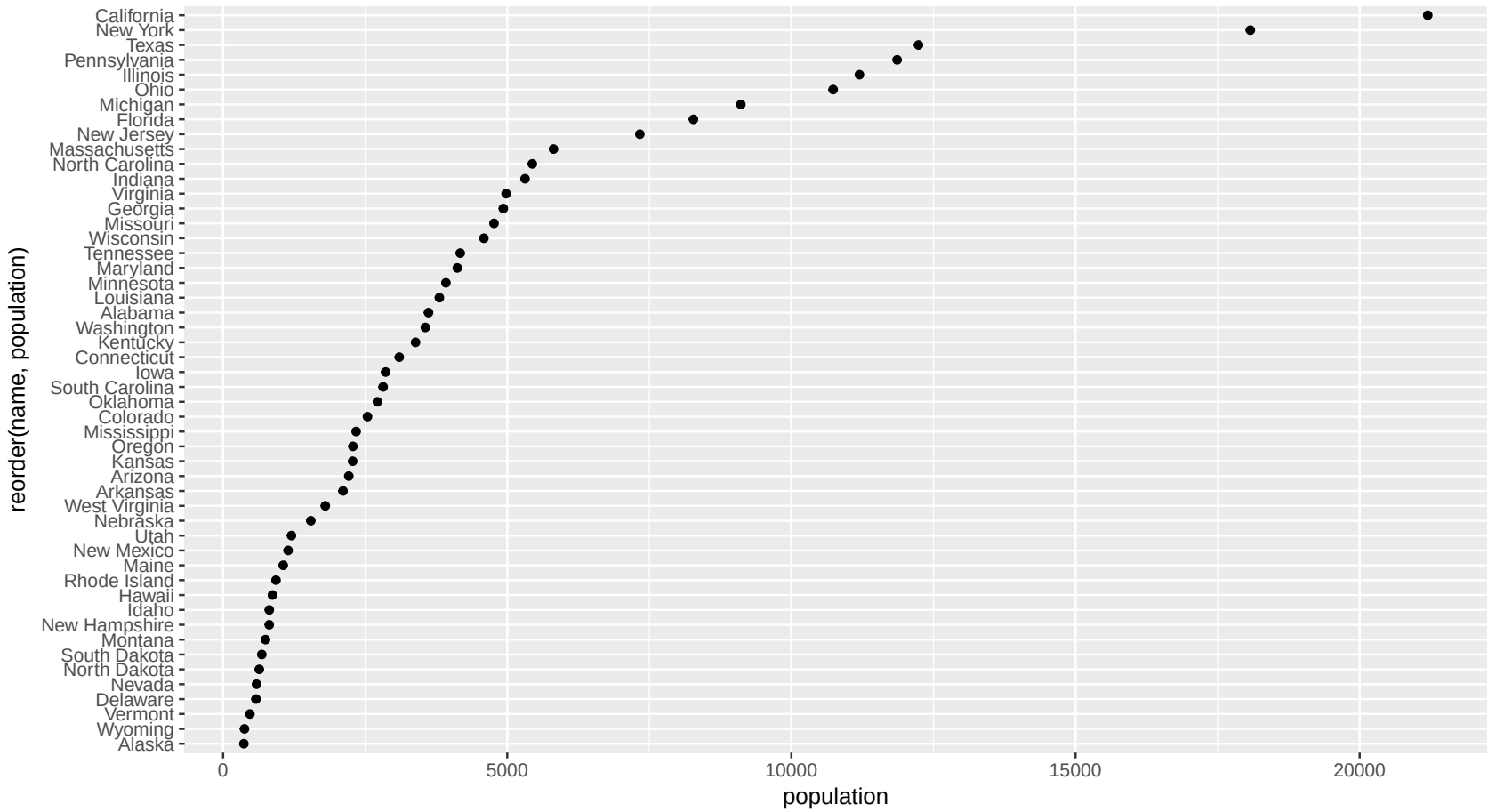
Similar to a scatterplot but with y = categorical variable

```
ggplot(data = state_df, aes(
    x = population,
    y = name
    )) + geom_point()
```



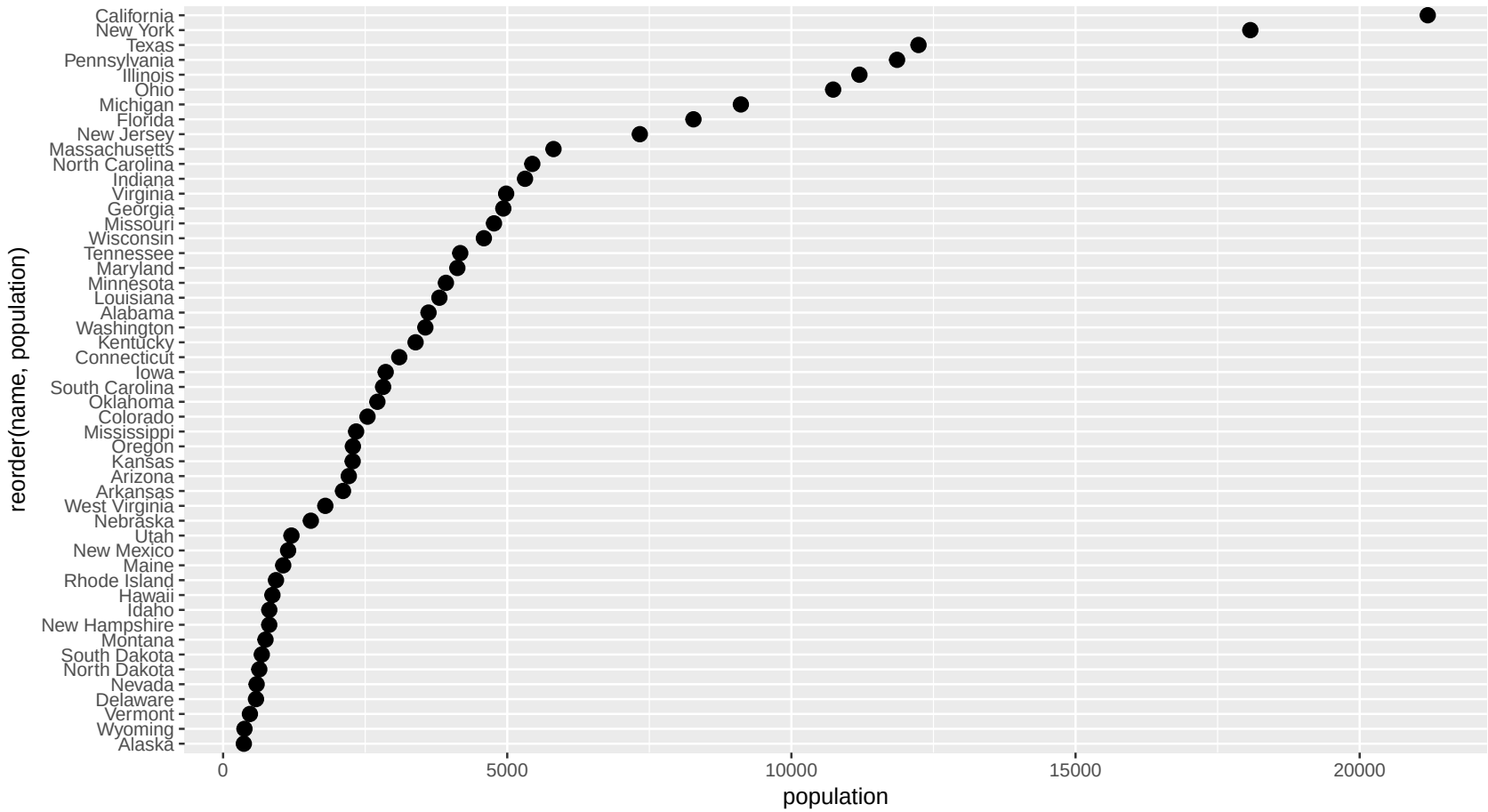
Change row-order in *aes()* via *reorder()*

```
ggplot(data = state_df, aes(
    x = population,
    y = reorder(name, population)
    )) + geom_point()
```



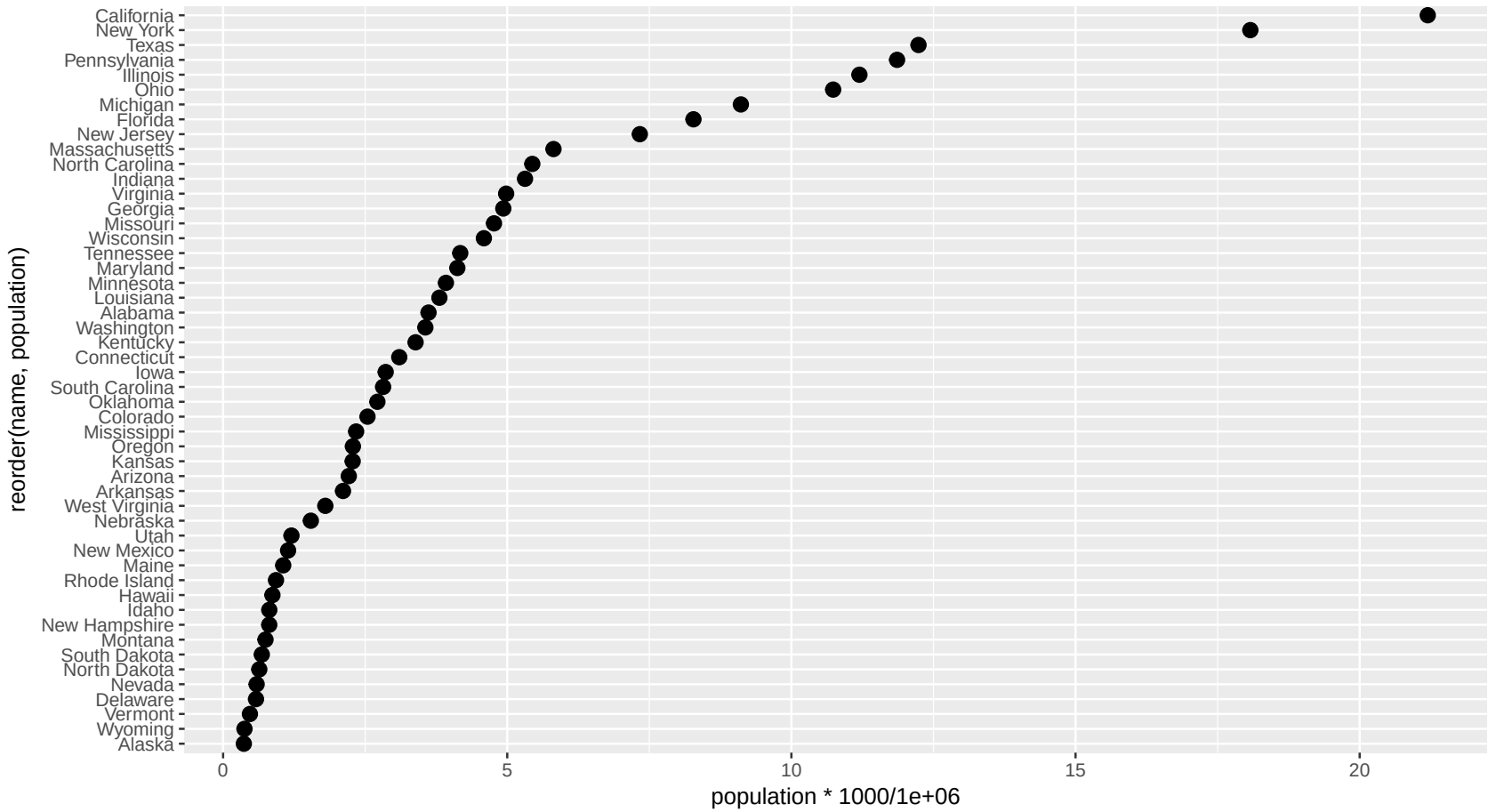
Edit dot size with *geom_point()*

```
ggplot(data = state_df, aes(
    x = population,
    y = reorder(name, population)
    )) + geom_point(size = 3)
```



Change population units to millions inside *aes()*

```
f3 <- ggplot(data = state_df, aes(
    x = population * 1000 / 1e6,
    y = reorder(name, population)
    )) + geom_point(size = 3)
```

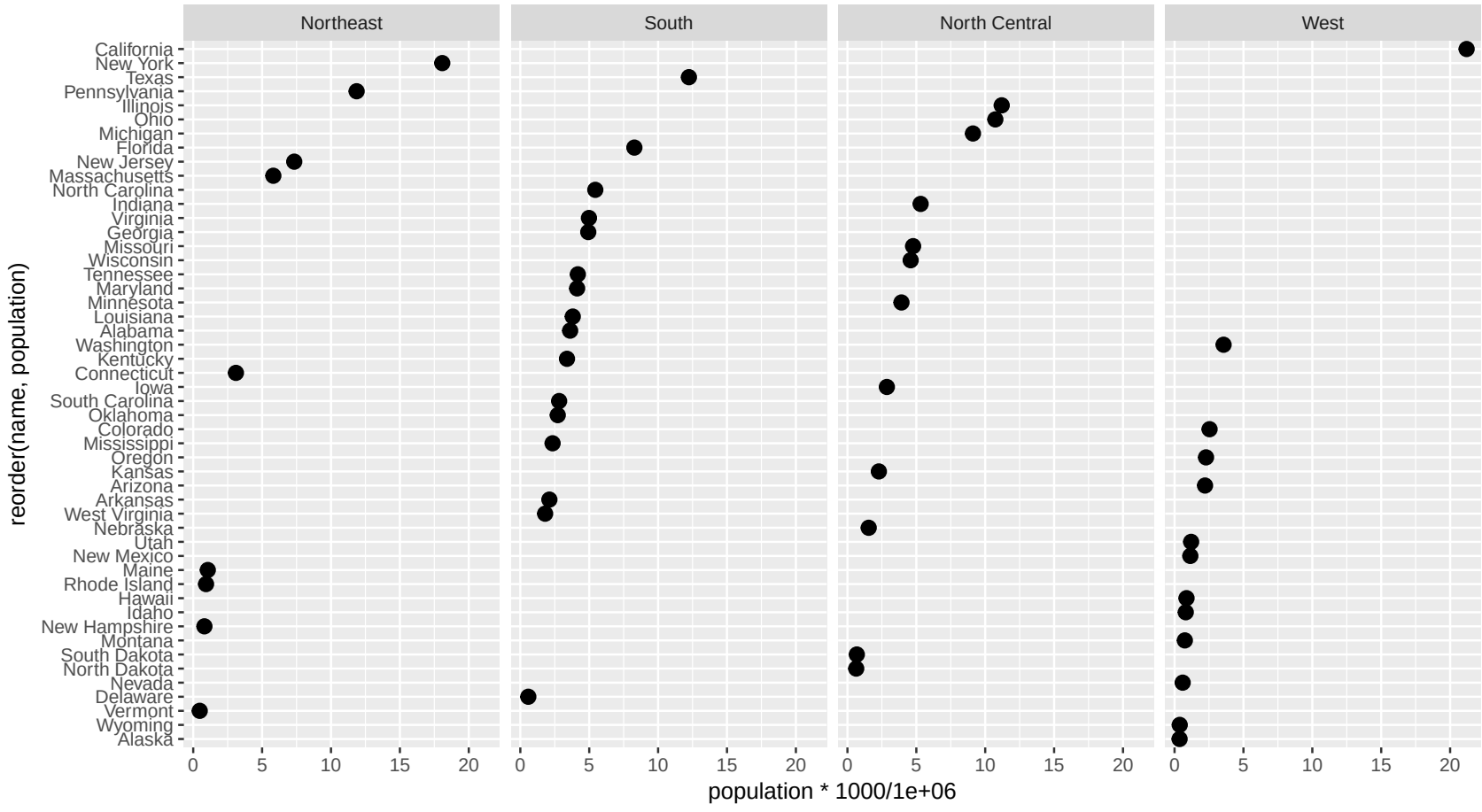


That's the basic dot plot

- We can still edit axis labels and the theme
- Dot plots are effective substitutes for most pies and bars

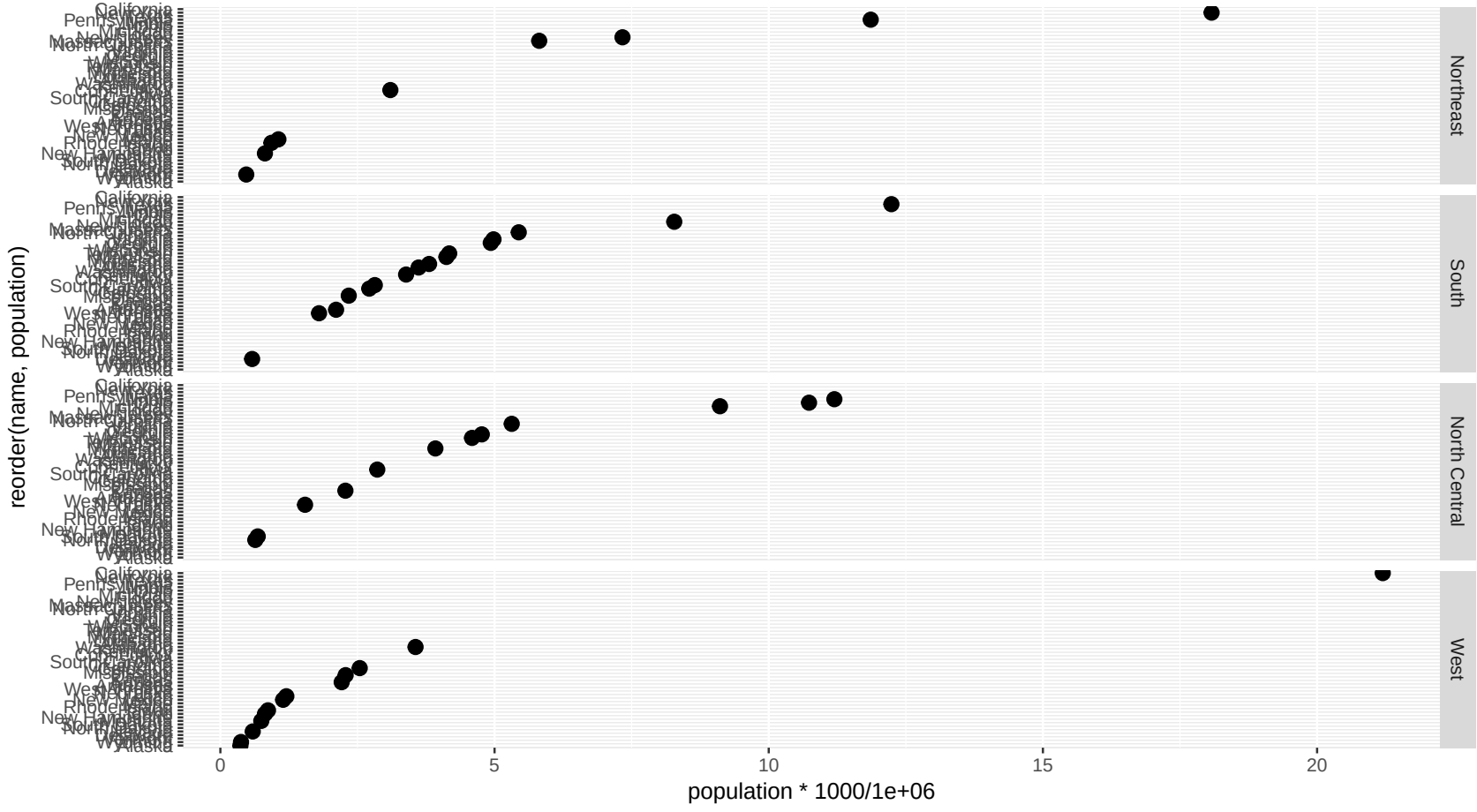
Conditioning by a categorical variable *region*

```
f4 <- f3 + facet_grid(~ region)
```



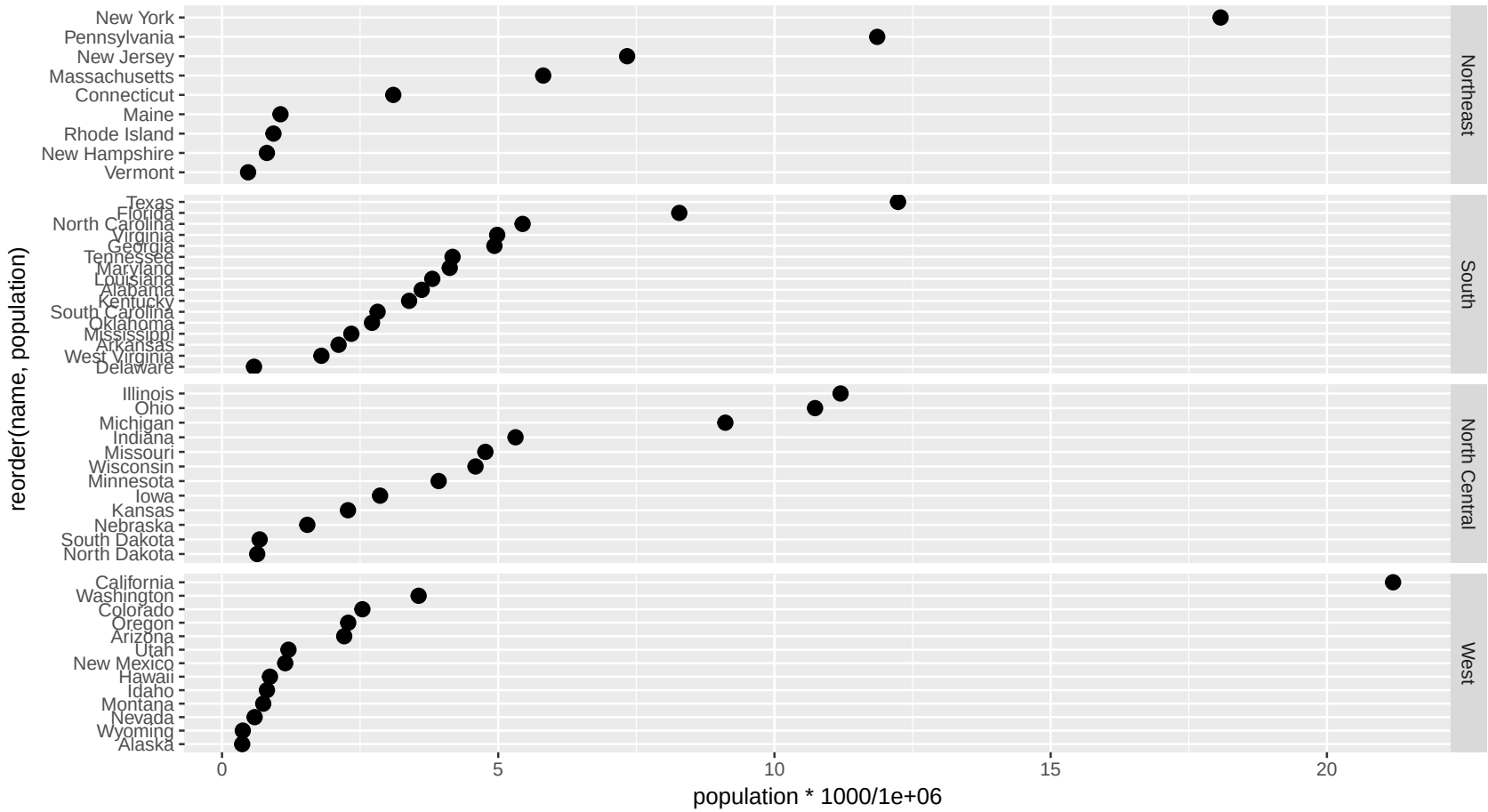
Facets in a column by swapping the position of ~

```
f5 <- f3 + facet_grid(region ~.)
```



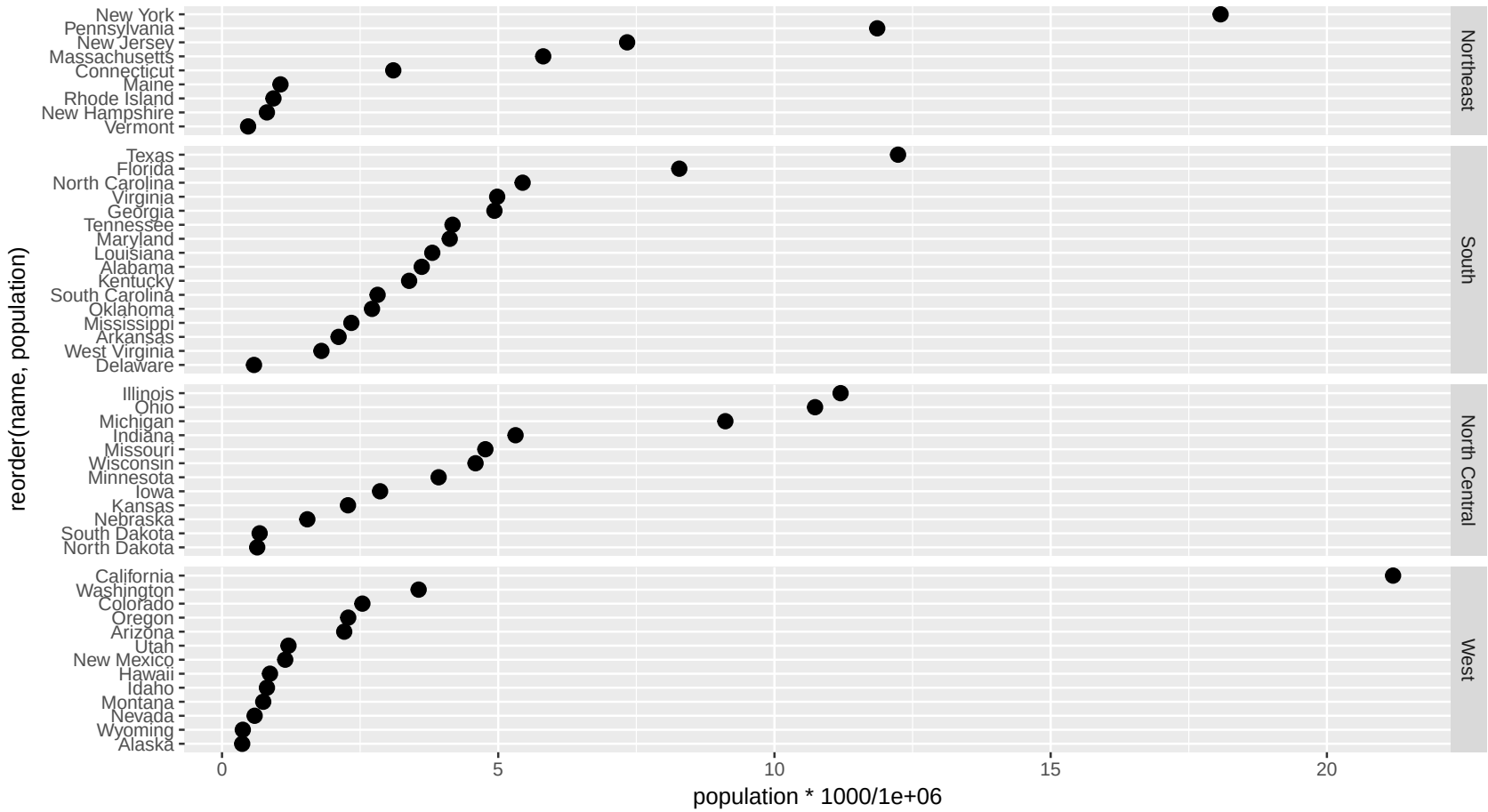
Delete unused rows with `scales = "free_y"`

```
f6 <- f3 + facet_grid(
    region ~. ,
    scales = "free_y")
```



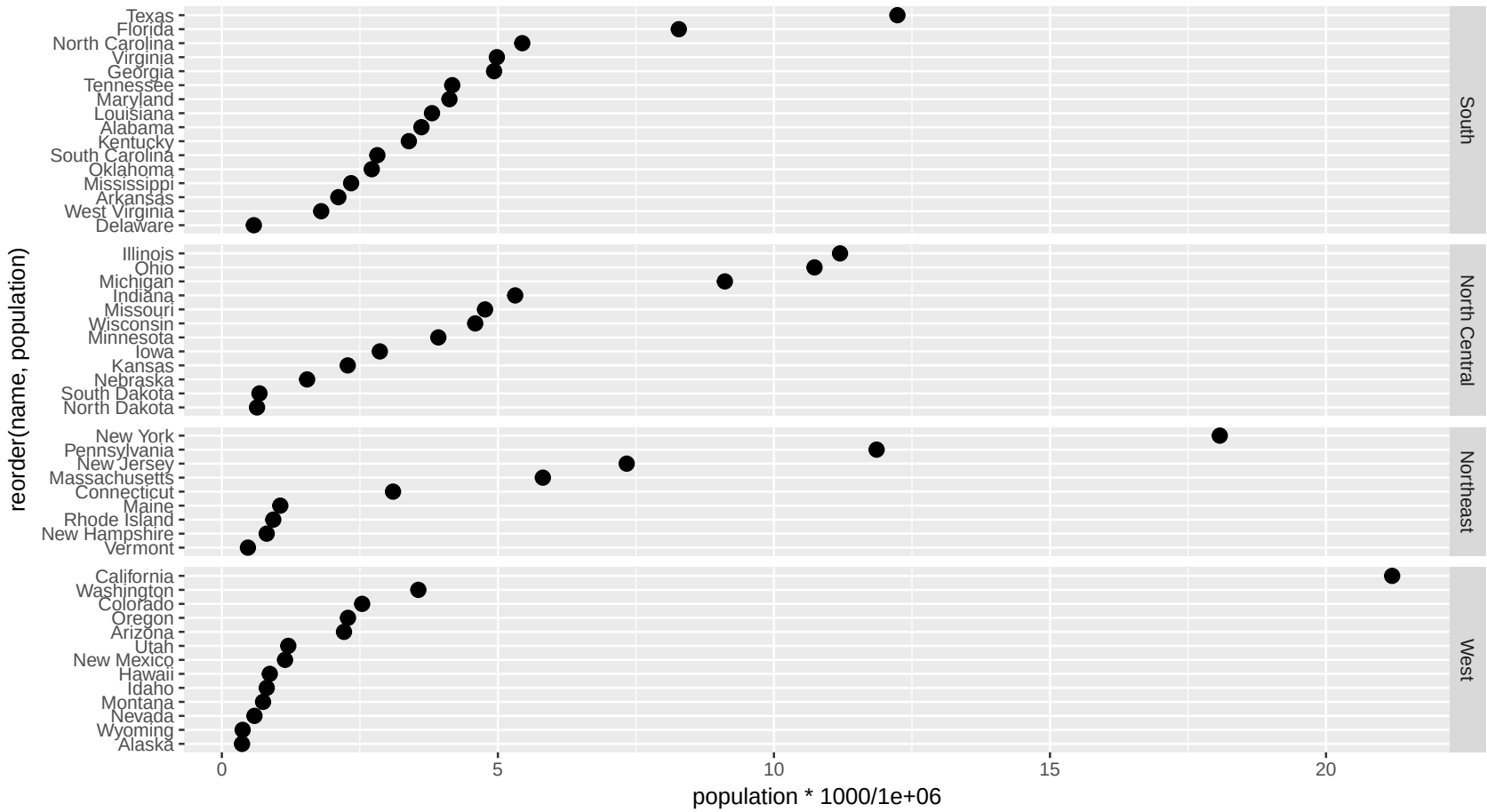
Make row intervals equal with space = "free_y"

```
f7 <- f3 + facet_grid(
    region ~. ,
    scales = "free_y", space = "free_y")
```



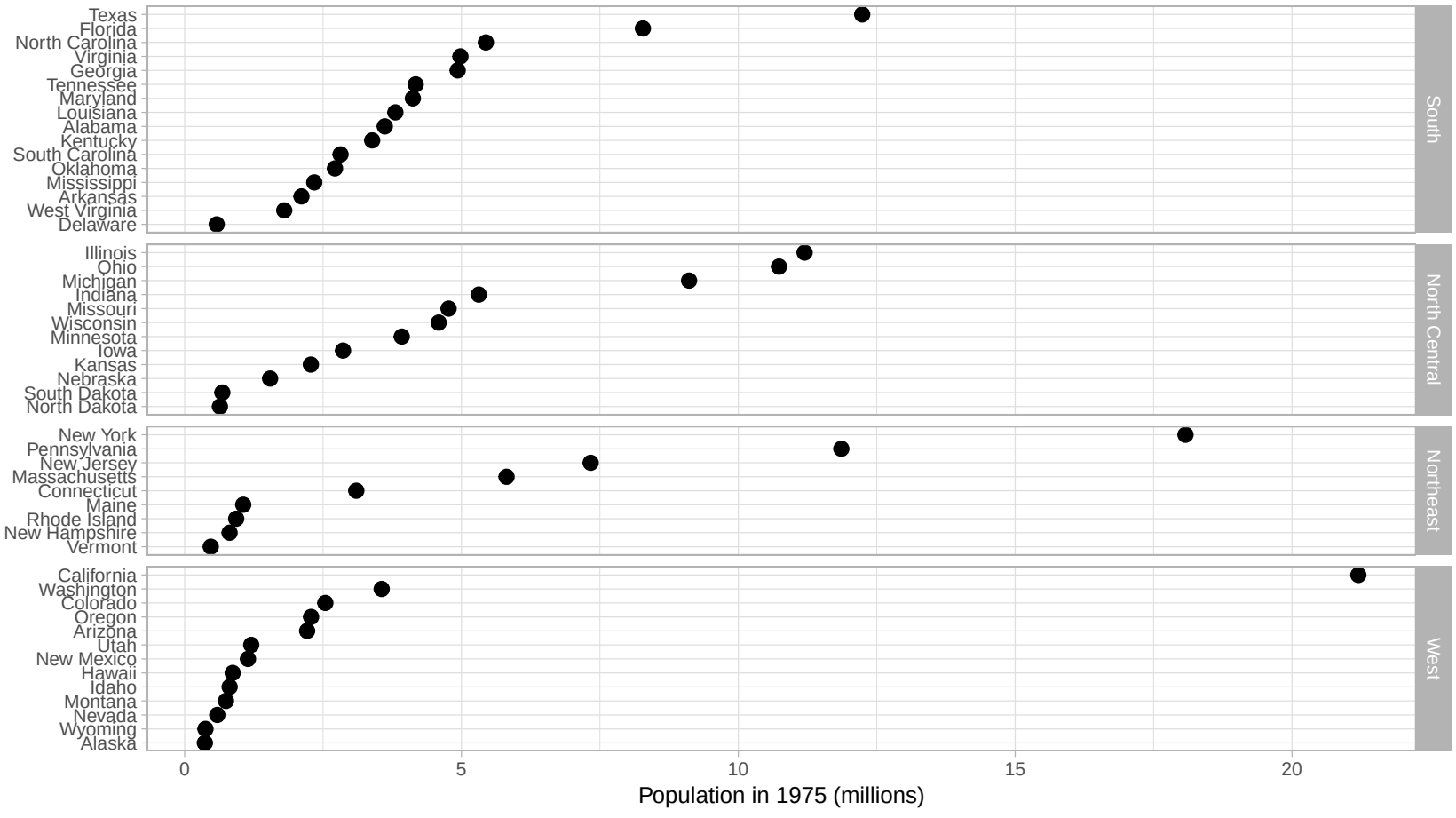
Reorder facets by sum of population in a facet

```
f8 <- f3 + facet_grid(
    reorder(region, -population, sum) ~. ,
    scales = "free_y", space = "free_y")
```



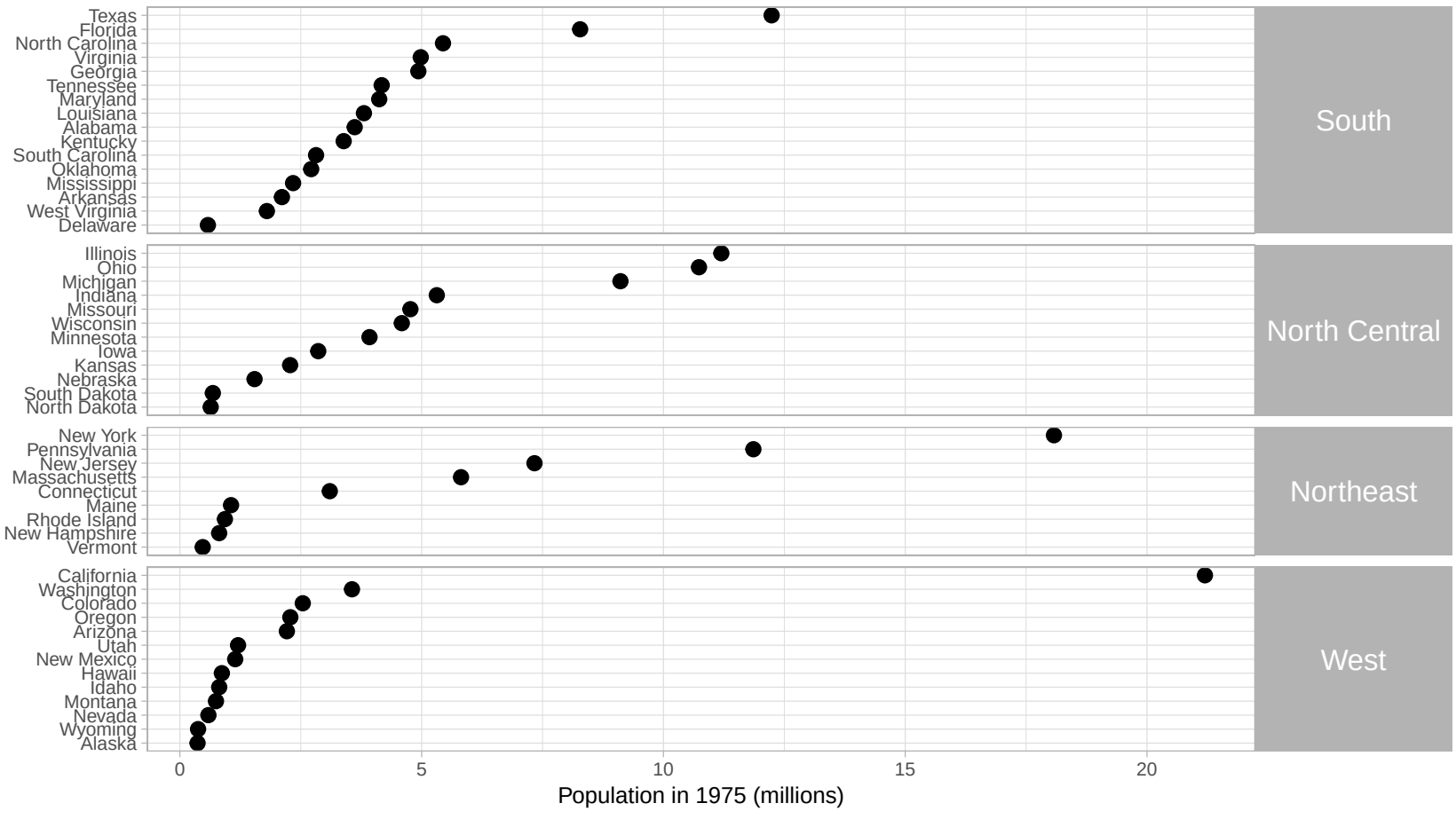
Edit labels and theme

```
f9 <- f8 + theme_light()  +
    labs(x = "Population in 1975 (millions)", y = "")
```



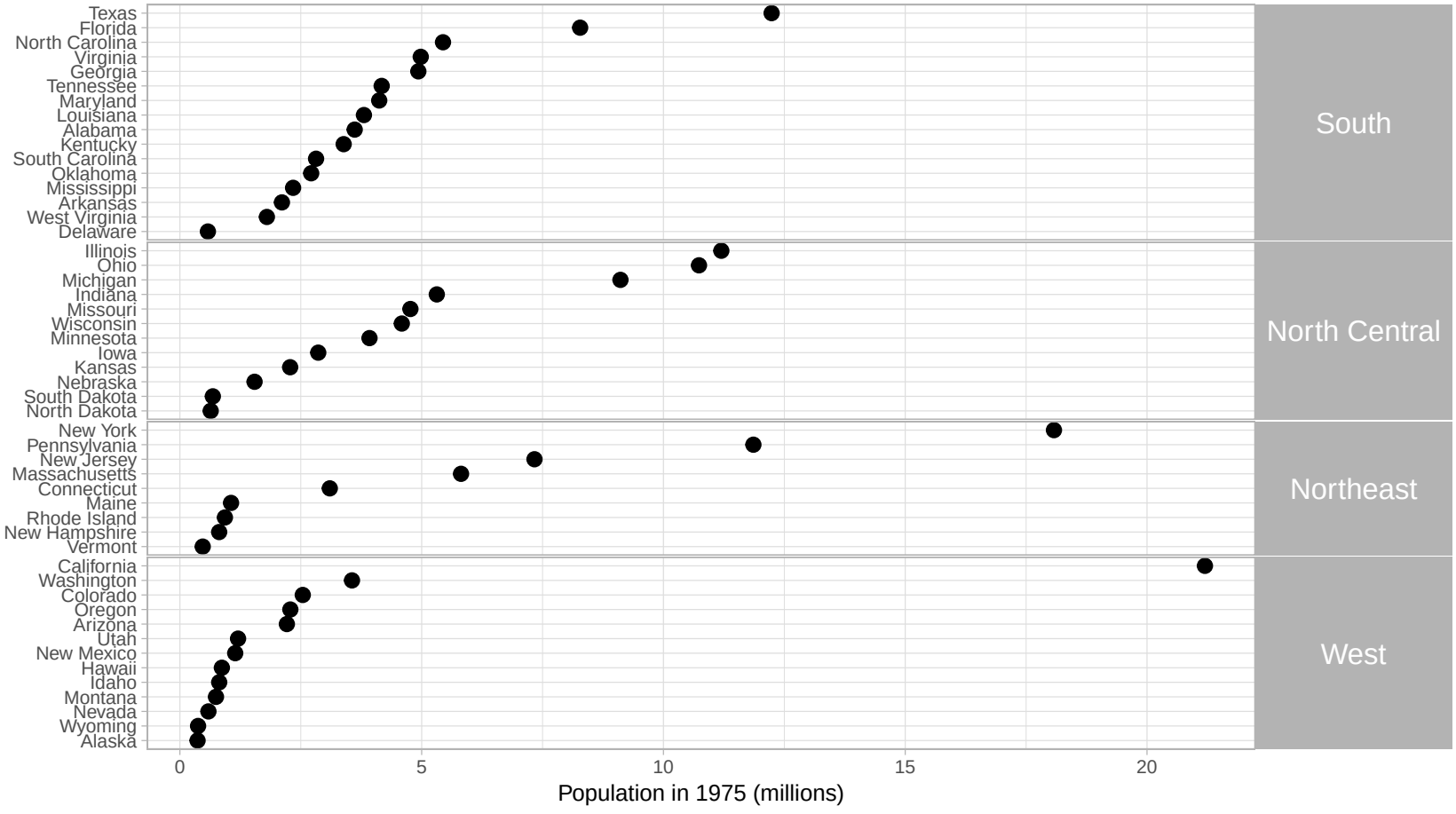
Rotate the facet strip labels

```
f10 <- f9 + theme(
    strip.text.y = element_text(angle = 0, size = 14))
```



Reduce spacing between facets concludes the example

```
f11 <- f10 + theme(panel.spacing = unit(0.3, "mm"))
```



Example 2

1940's death rates in Virginia

Dot plot with two conditioning variables

We manually tidied this data previously

```
library(readxl)
rates <- read_excel(
    path = "data/VADeaths.xlsx", sheet = "tidy"
    )
```

```
tibble::glimpse(rates)
## Observations: 20
## Variables: 4
## $ location   <chr> "rural", "rural", "rural", "...
## $ sex        <chr> "men", "men", "men", "men", ...
## $ age_group  <chr> "50-54", "55-59", "60-64", "...
## $ death_rate <dbl> 11.7, 18.1, 26.9, 41.0, 66.0...
```

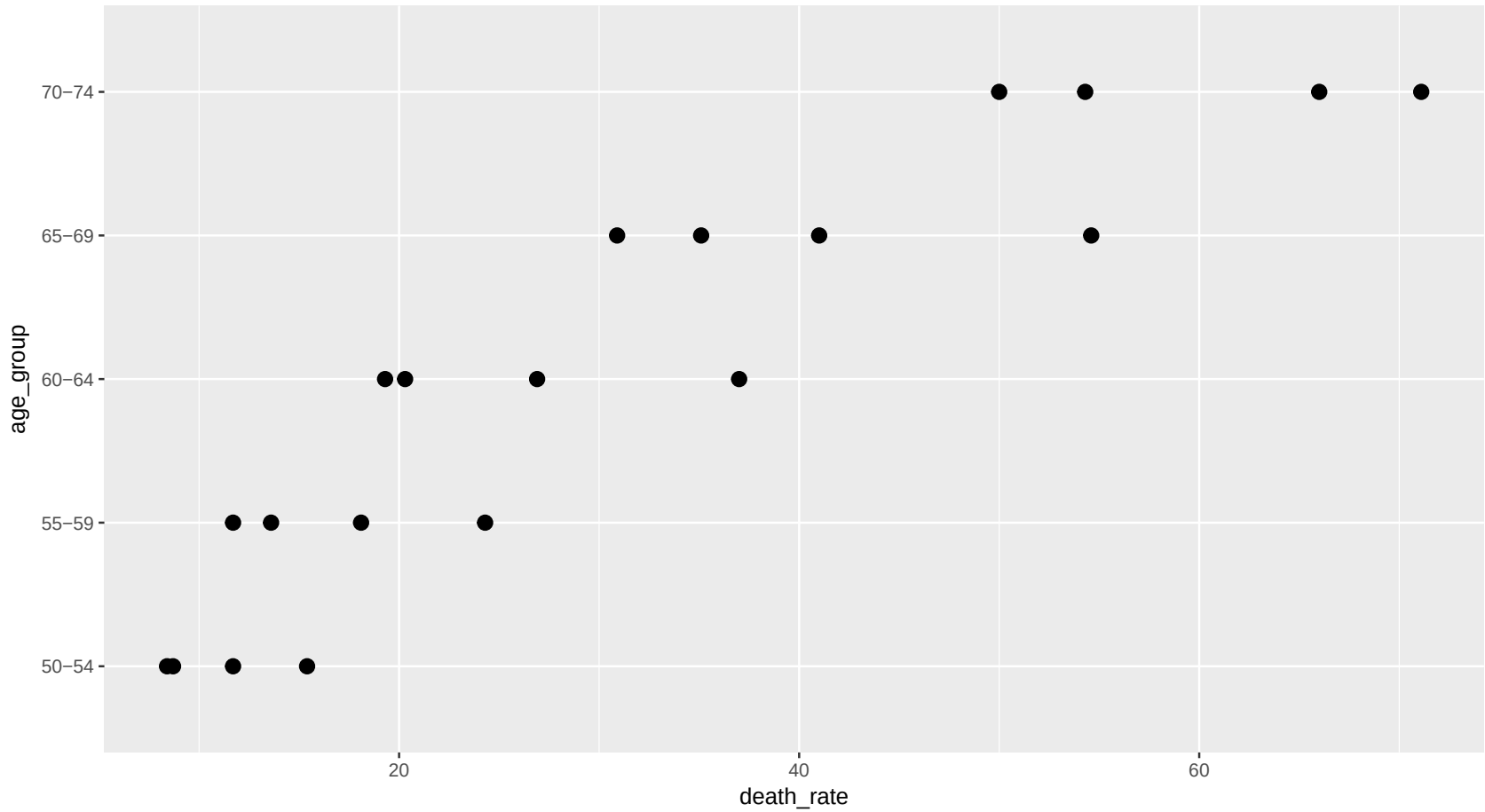
Examine the data in a simple table

```
knitr::kable(rates)
```

location	sex	age_group	death_rate
rural	men	50-54	11.7
rural	men	55-59	18.1
rural	men	60-64	26.9
rural	men	65-69	41.0
rural	men	70-74	66.0
rural	women	50-54	8.7
rural	women	55-59	11.7
rural	women	60-64	20.3
rural	women	65-69	30.9
rural	women	70-74	54.3
urban	men	50-54	15.4
urban	men	55-59	24.3
urban	men	60-64	37.0
urban	men	65-69	54.6

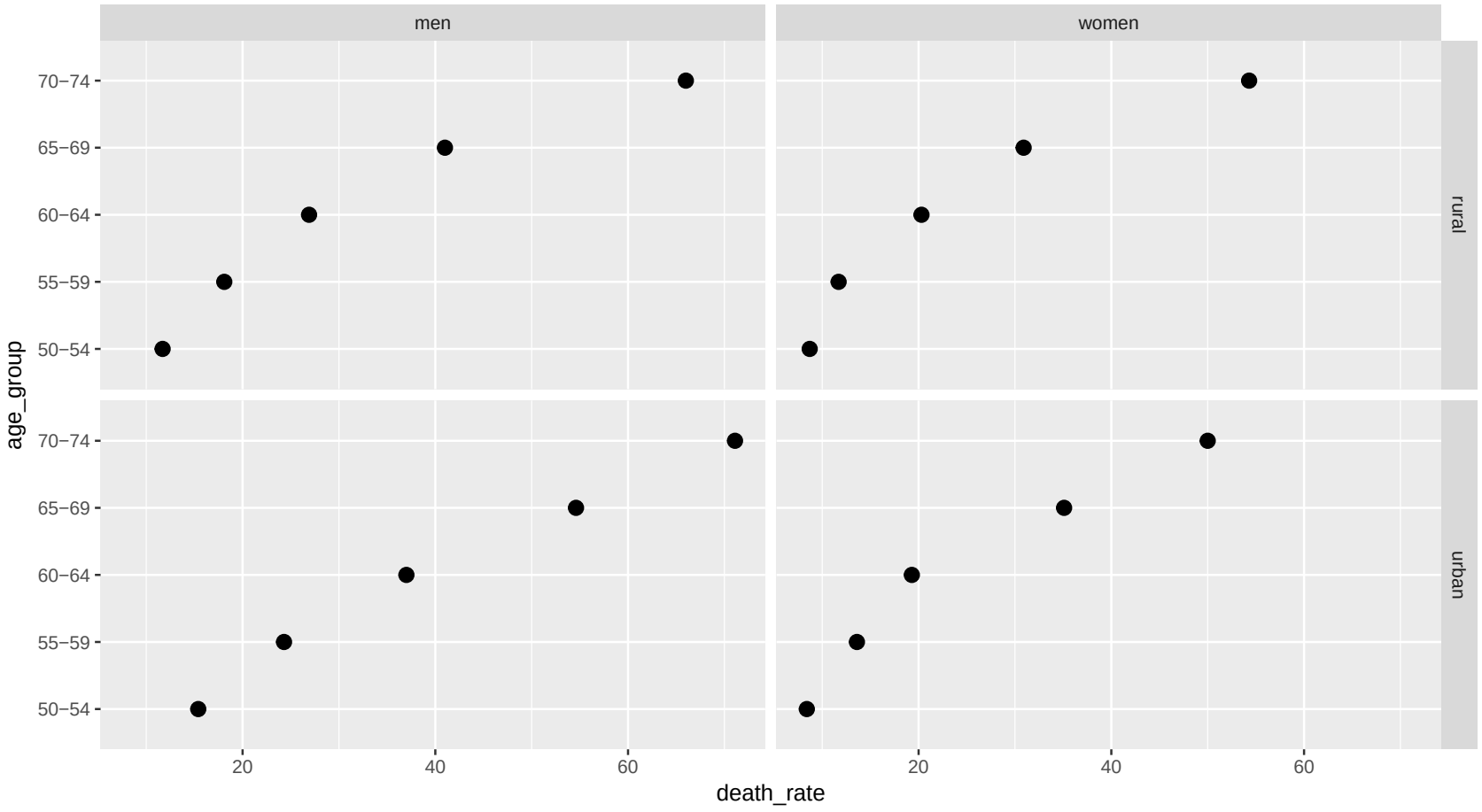
The basic dot plot has 4 dots per row

```
f1 <- ggplot(data = rates,
     aes(x = death_rate, y = age_group)) +
     geom_point(size = 3)
```



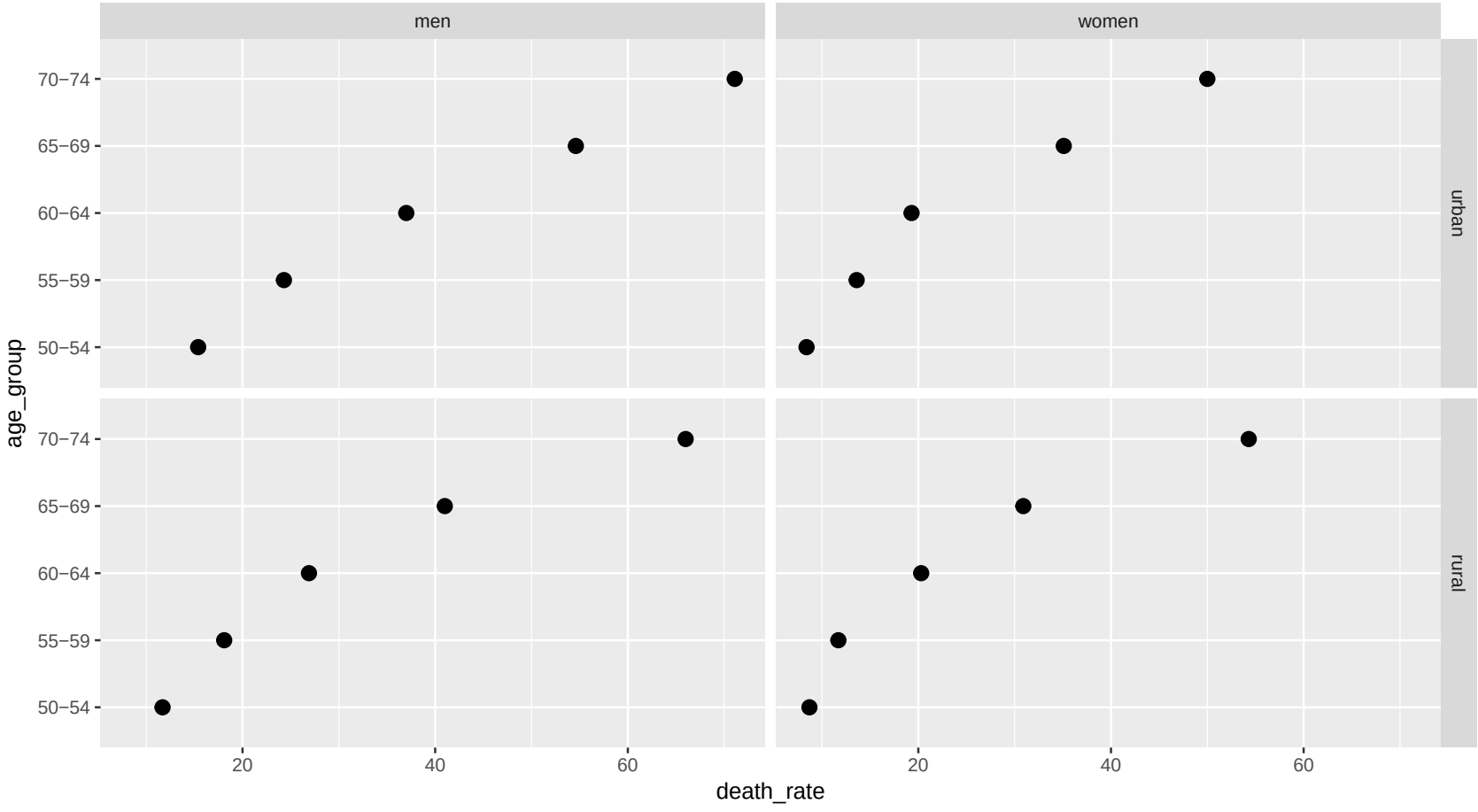
Condition by both categories

```
f2 <- f1 + facet_grid(location ~ sex)
```



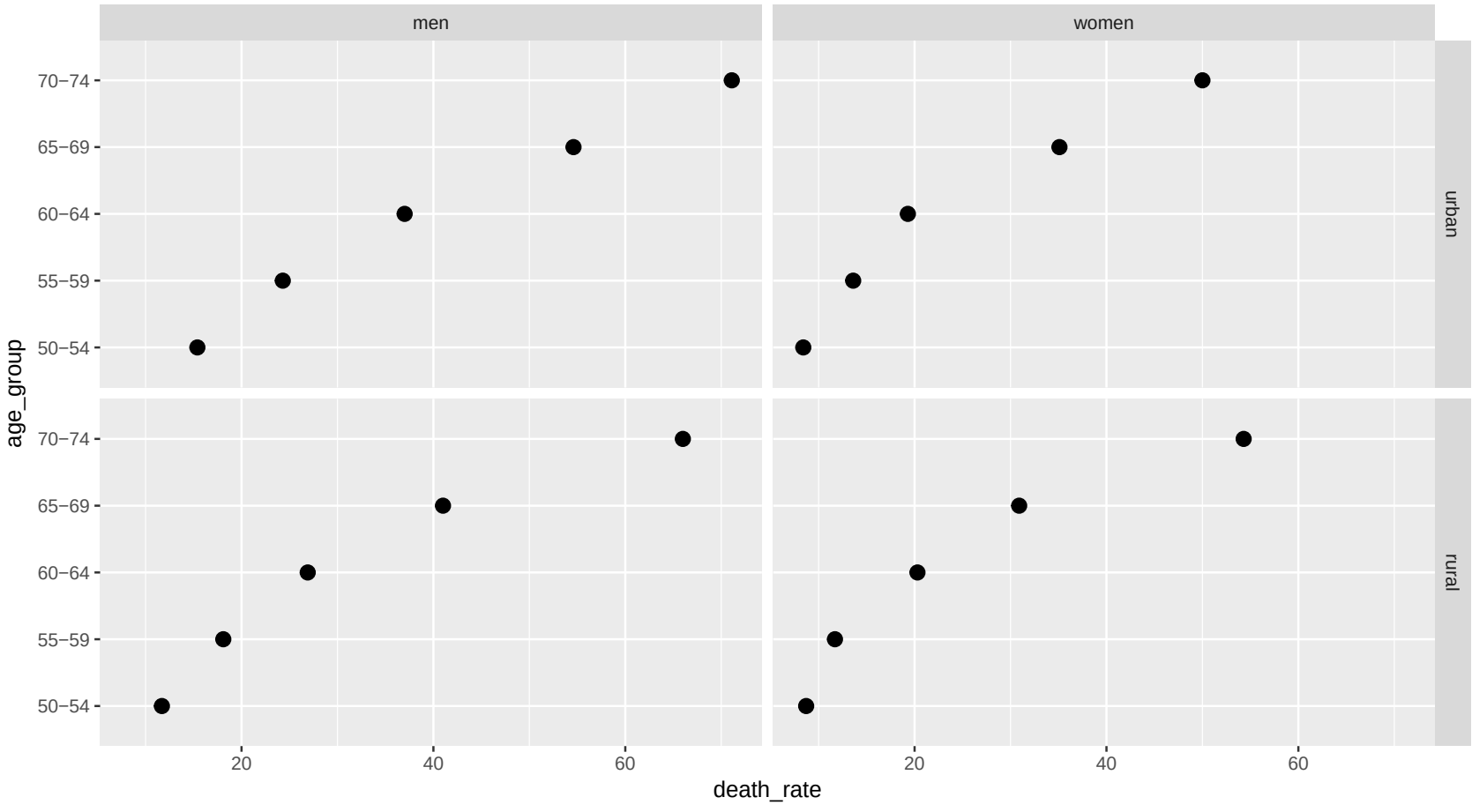
Reorder the panels by the mean rate by location

```
f3 <- f1 + facet_grid(reorder(location, -death_rate, mean)
     ~ sex)
```



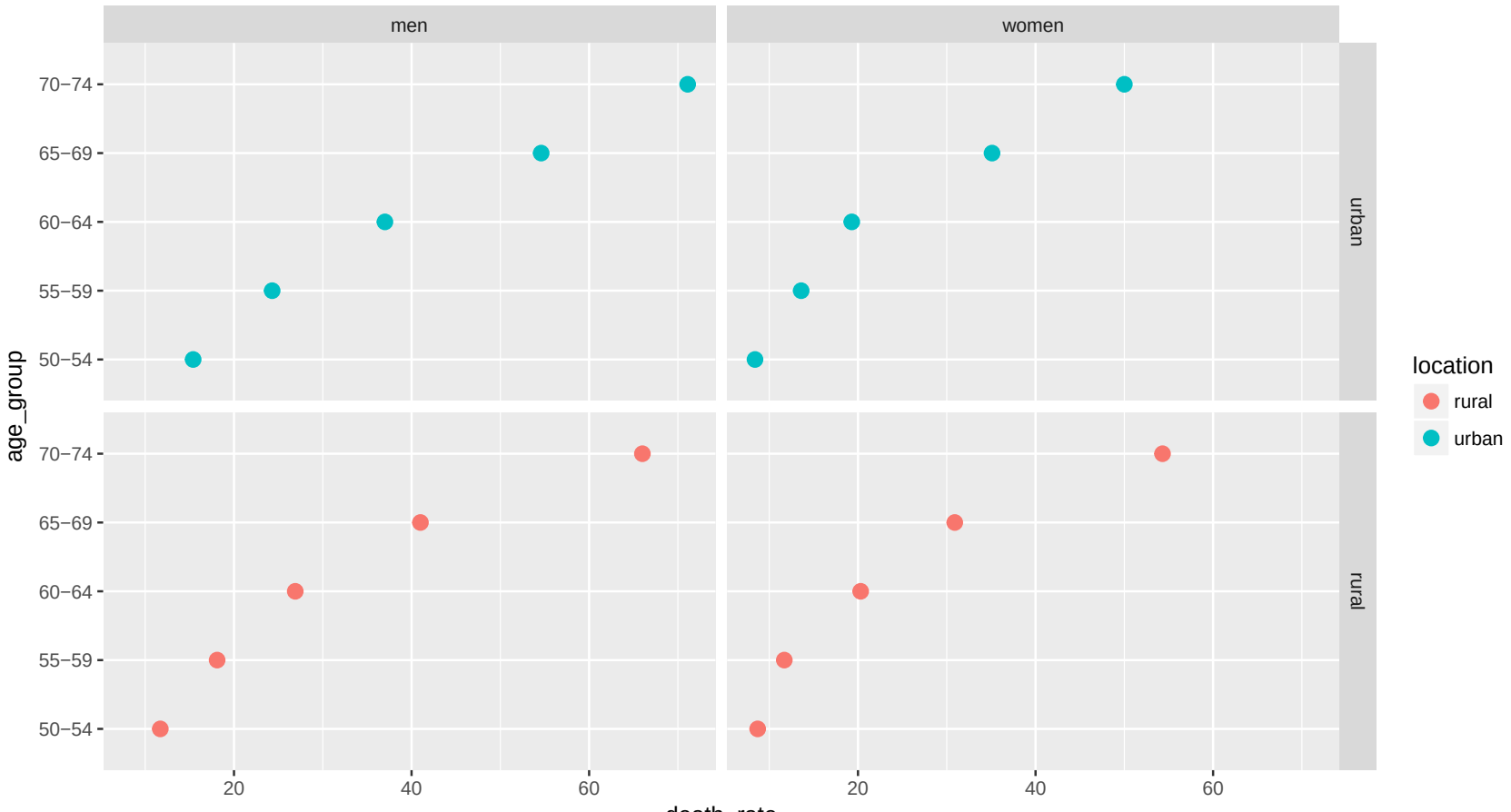
Reorder the panels by the mean rate by sex

```
f5 <- f1 + facet_grid(reorder(location, -death_rate, mean)
     ~ reorder(sex, -death_rate, mean))
```



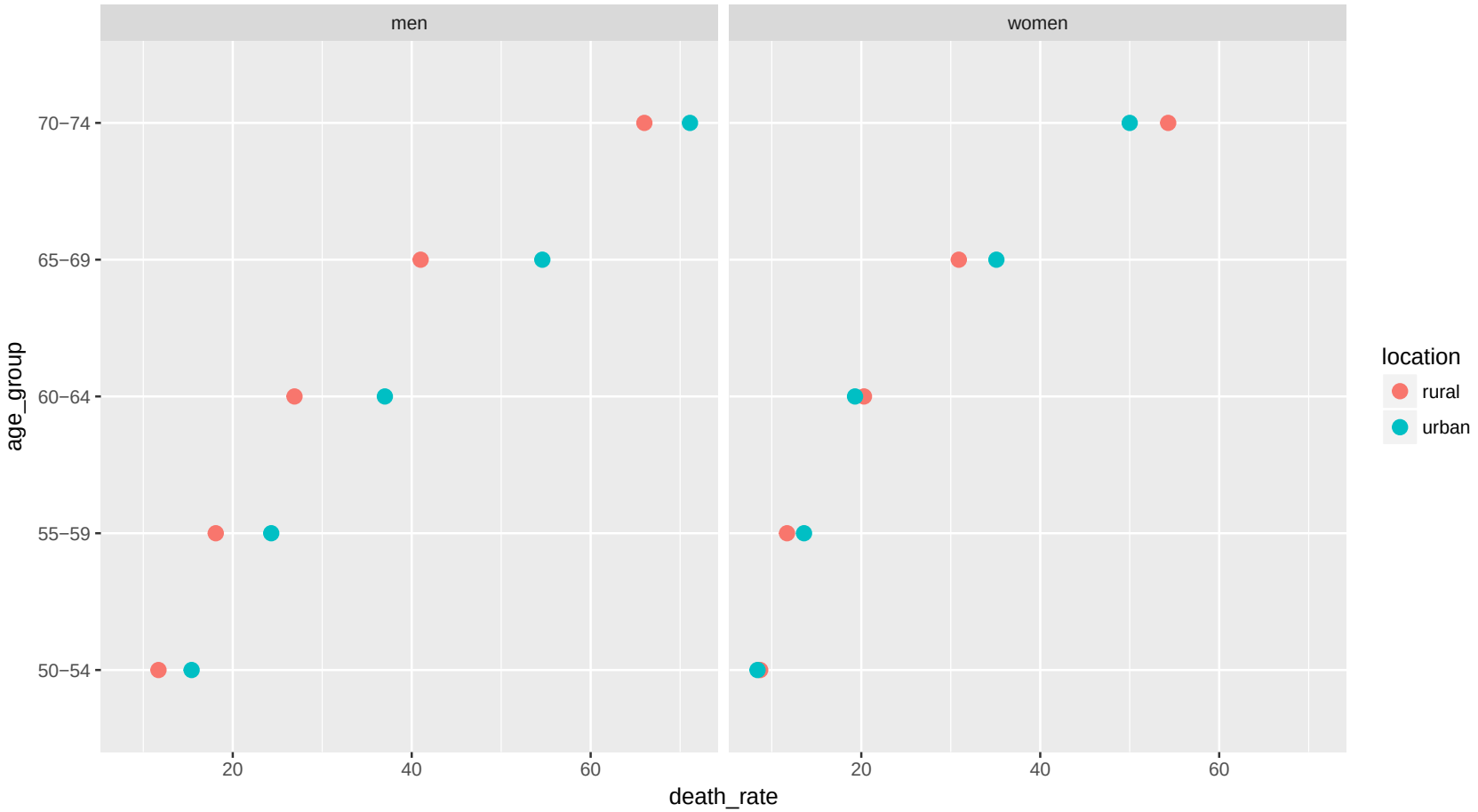
Add color to *aes()* to distinguish urban and rural

```
f6 <- ggplot(data = rates,
    aes(x = death_rate, y = age_group, col = location)) +
    geom_point(size = 3) +
    facet_grid(reorder(location, -death_rate, mean)
        ~ reorder(sex, -death_rate, mean ))
```



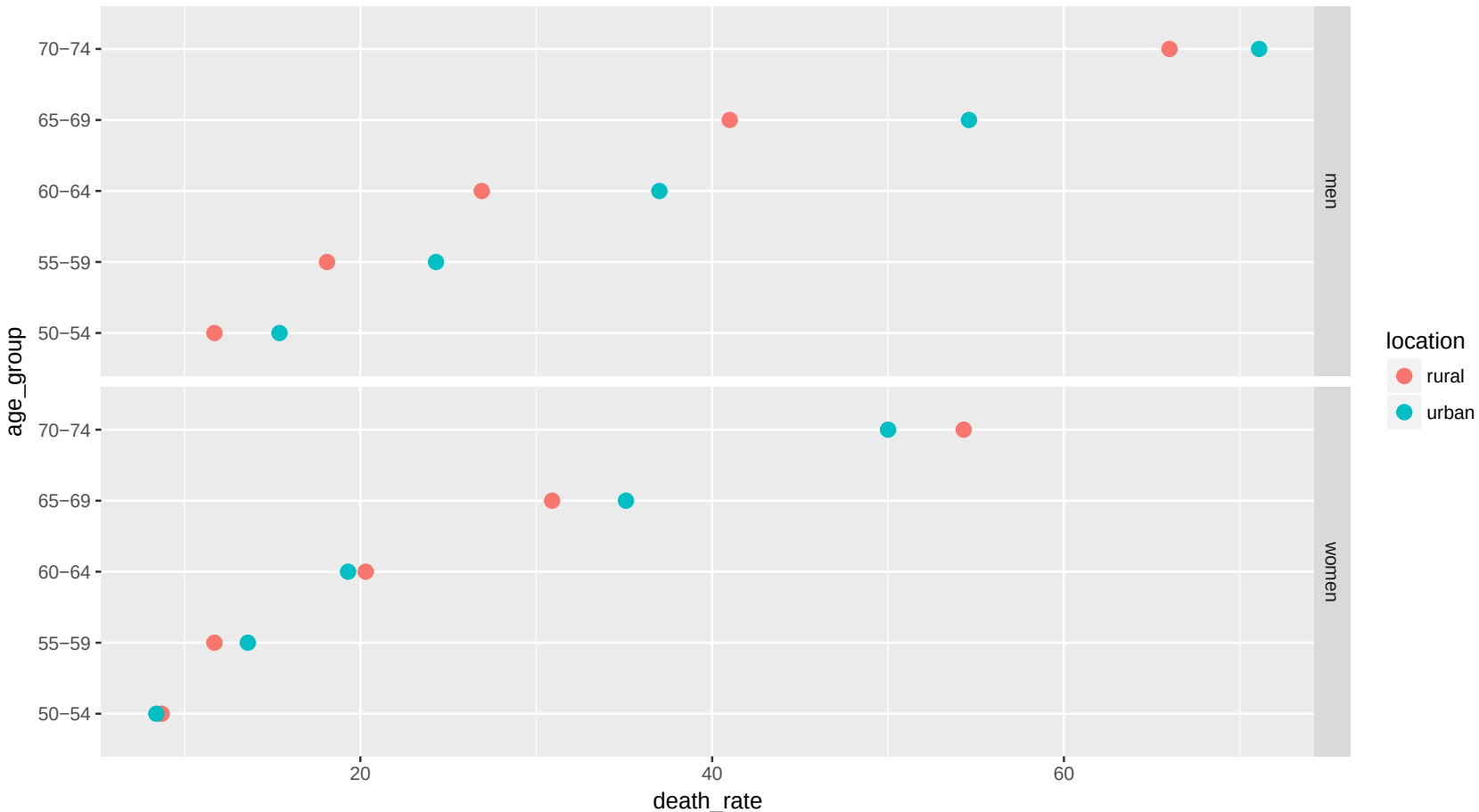
Removing the location condition yields two dots per row

```
f7 <- ggplot(data = rates,
     aes(x = death_rate, y = age_group, col = location)) +
     geom_point(size = 3) +
     facet_grid(.~ reorder(sex, -death_rate, mean))
```



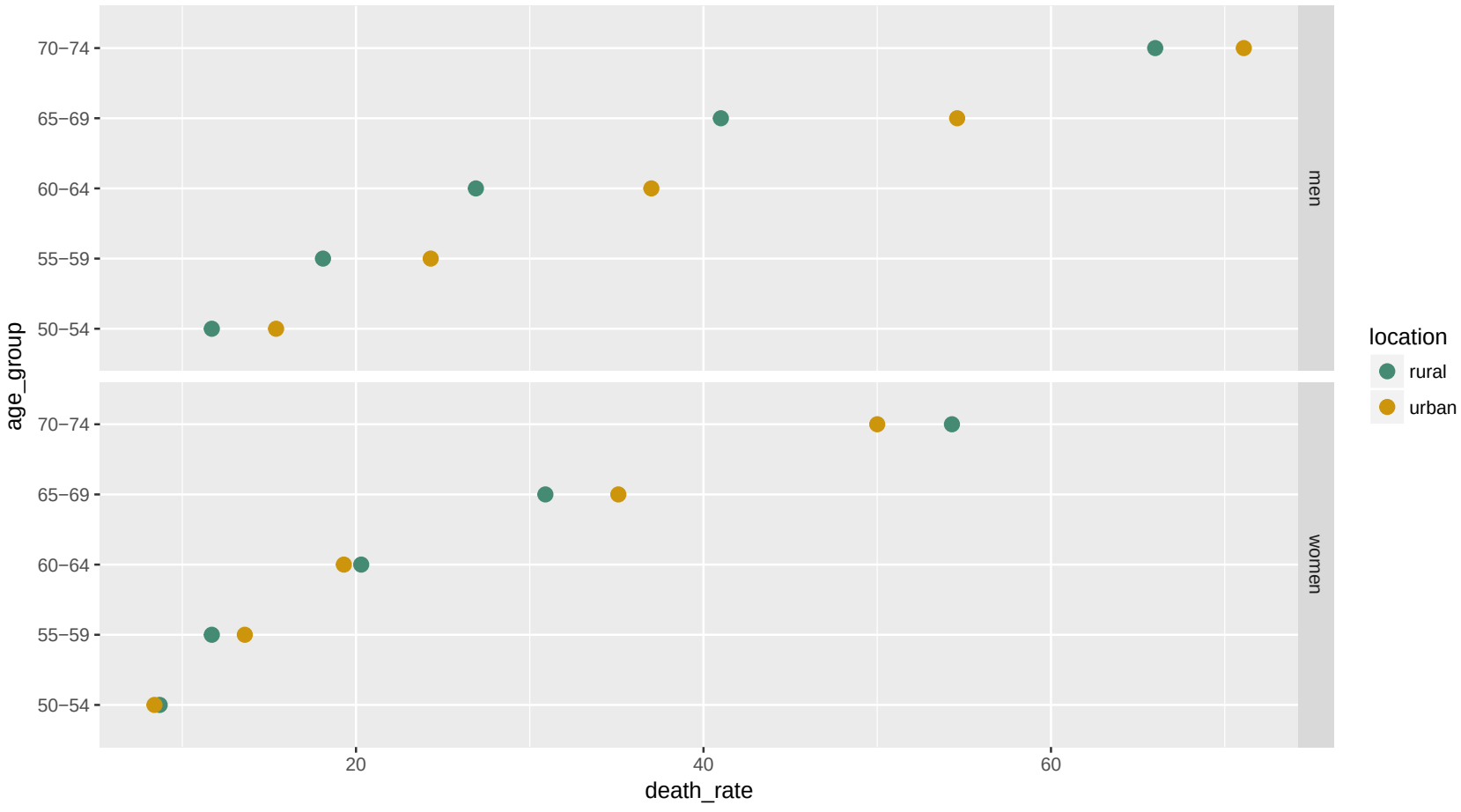
Swap orientation in *facet_grid()*

```
f8 <- ggplot(data = rates,
     aes(x = death_rate, y = age_group, col = location)) +
     geom_point(size = 3) +
     facet_grid(reorder(sex, -death_rate, mean) ~.)
```



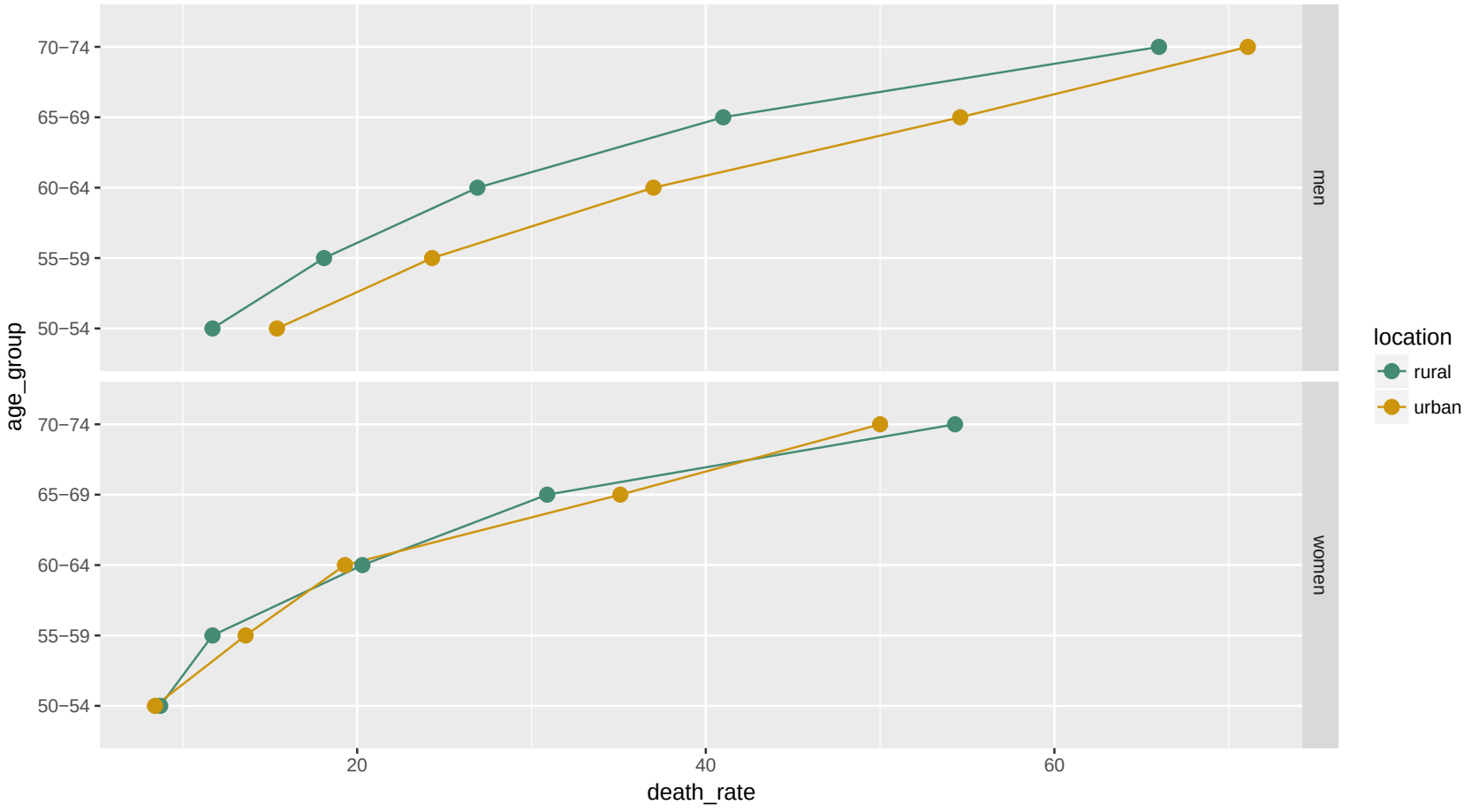
Edit the two colors (See Colors in R)

```
f9 <- f8 + scale_colour_manual(
    values = c("aquamarine4", "darkgoldenrod3"))
```



Add *group* in *aes()* and we can add a *geom_line()*

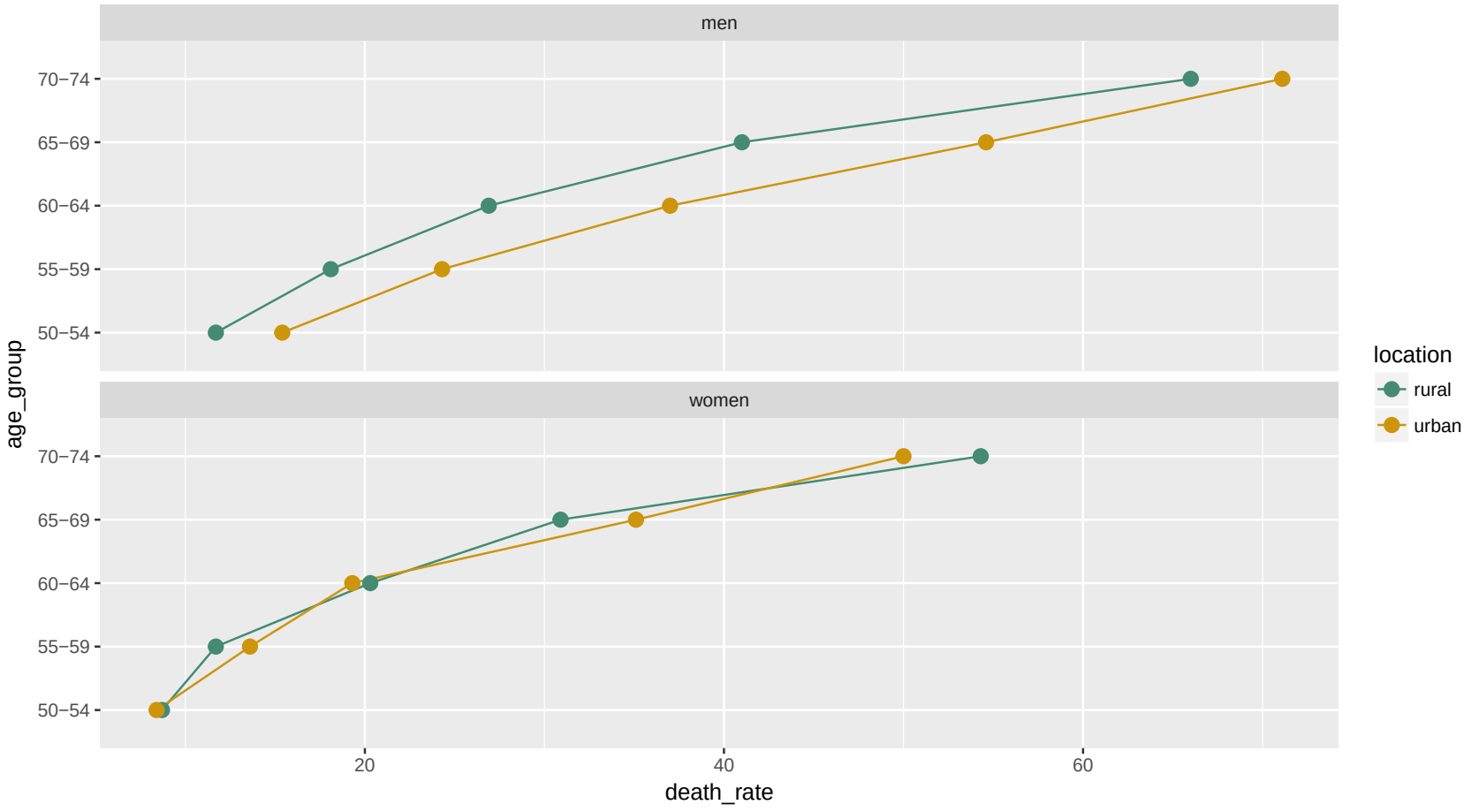
```
f10 <- ggplot(data = rates,
    aes(x = death_rate, y = age_group, col = location,
        group = location)
      ) +
    geom_line() + geom_point(size = 3) +
    facet_grid(reorder(sex, -death_rate) ~.) +
    scale_colour_manual(
        values = c("aquamarine4", "darkgoldenrod3")
        )
```


70–74
65–69
60–64
55–59
50–54
men
women
age_group
20
40
60
death_rate
location
rural
urban

Change *facet_grid()* to *facet_wrap()*

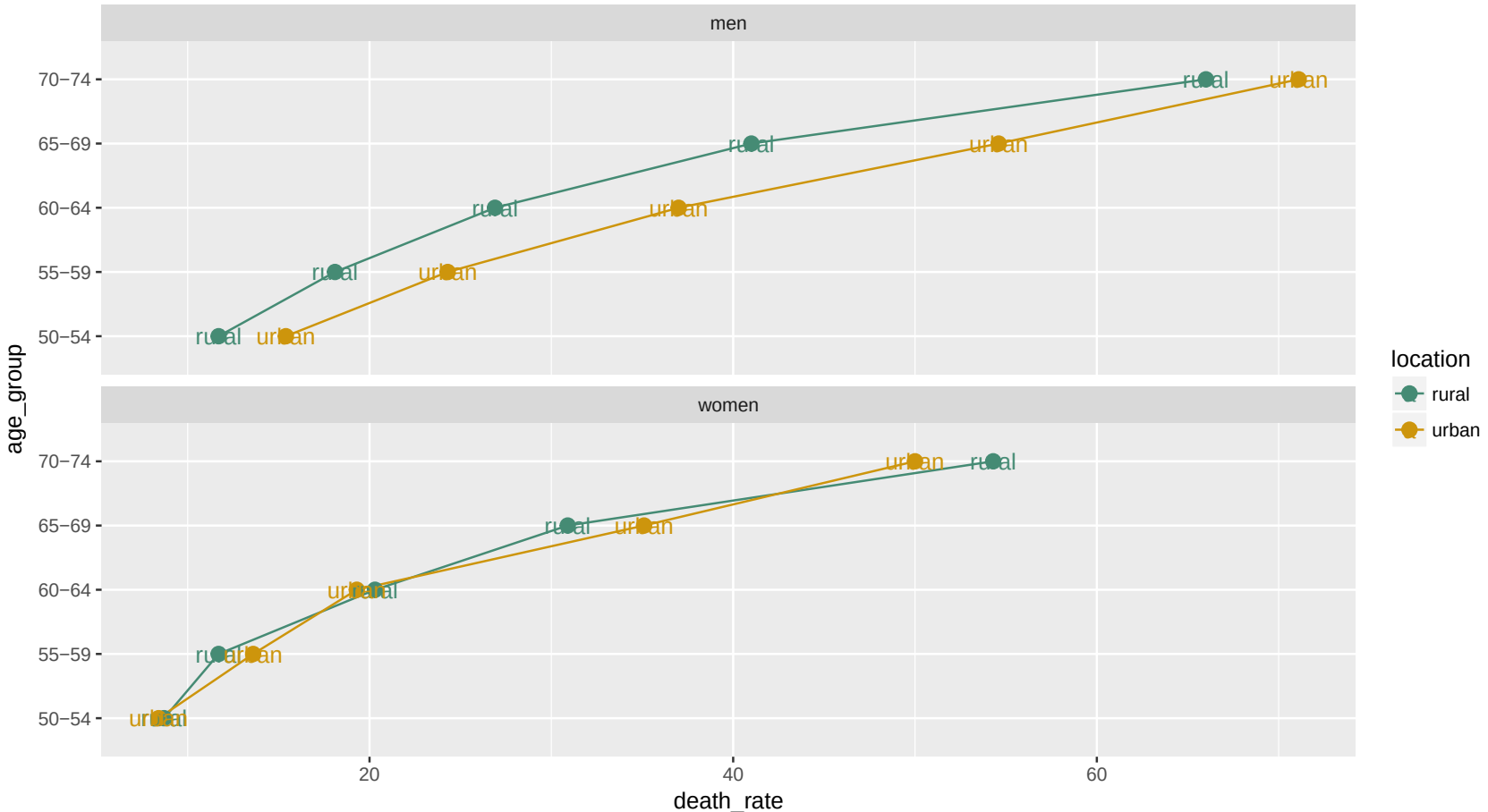
Moves the facet labels to the top of the panel

```
f11 <- ggplot(data = rates,
    aes(x = death_rate, y = age_group, col = location,
        group = location)) +
    geom_line() + geom_point(size = 3) +
    facet_wrap(~ sex, ncol = 1) +
    scale_colour_manual(
        values = c("aquamarine4", "darkgoldenrod3")
        )
```


men
women
70–74
65–69
60–64
55–59
50–54
age_group
20
40
60
death_rate
location
rural
urban

And urbal/rural labels to the lines

```
f12 <- f11 + geom_text(aes(label = location))
```



Create a new data frame for line labels

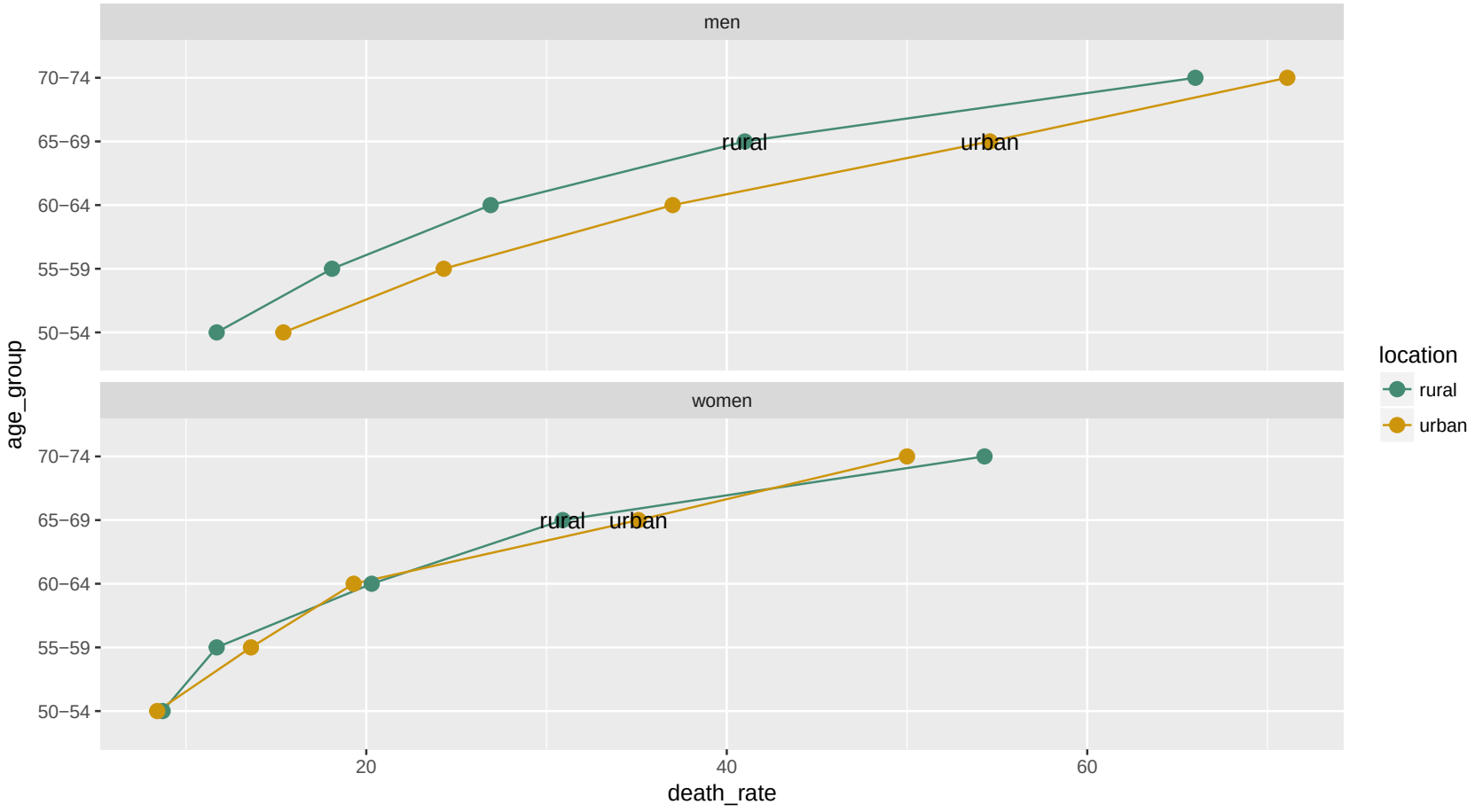
- The new data frame is a subset of the original
- Keeping only those rows for which the age group is "65-69"

```
library(dplyr)
line_labels <- rates %>%
    filter(age_group == "65-69")
```

```
glimpse(line_labels)
## Observations: 4
## Variables: 4
## $ location   <chr> "rural", "rural", "urban", "...
## $ sex        <chr> "men", "women", "men", "women"
## $ age_group  <chr> "65-69", "65-69", "65-69", "...
## $ death_rate <dbl> 41.0, 30.9, 54.6, 35.1
```

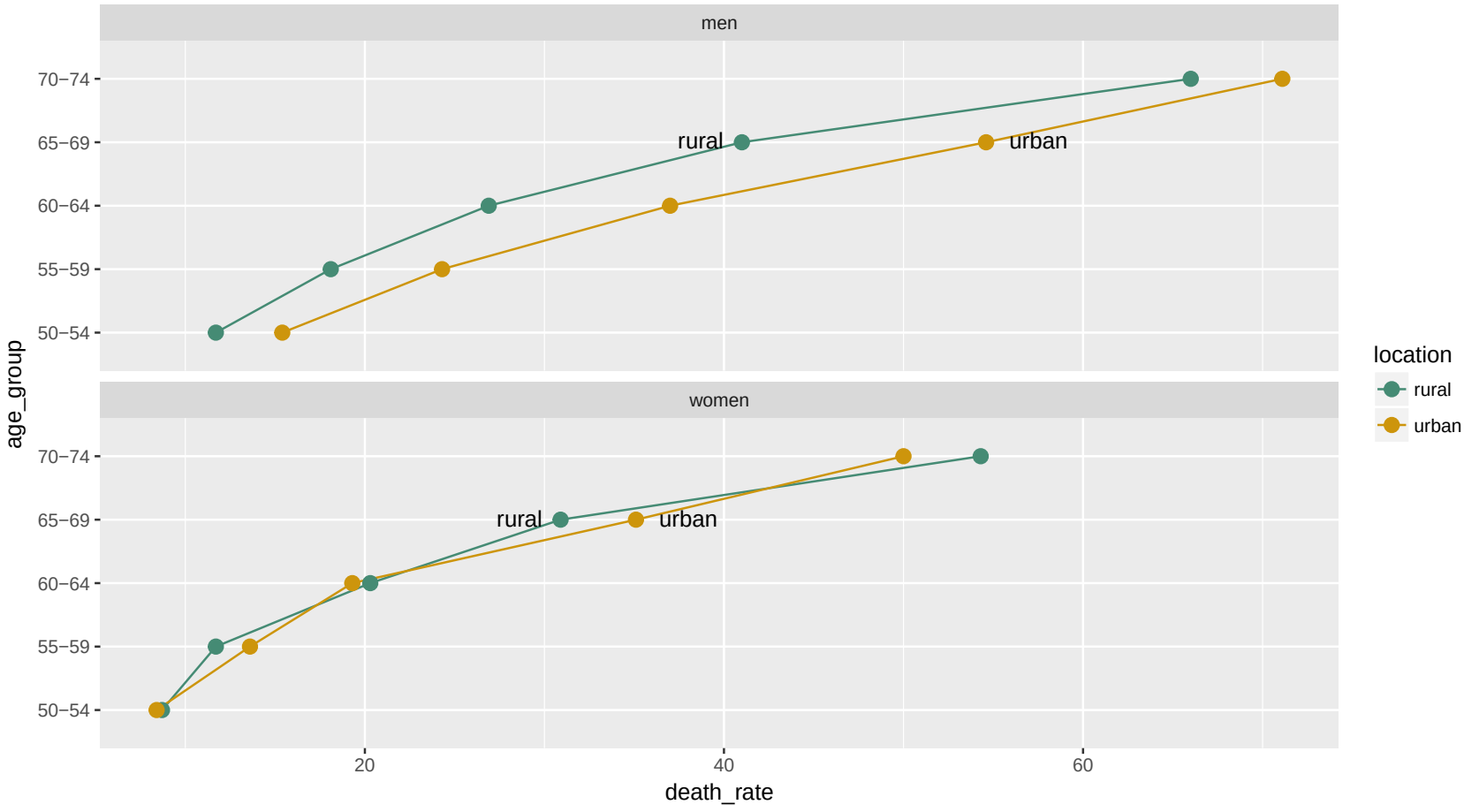
Use *geom_text()* and the new data frame for line labels

```
f13 <- f11 + geom_text(data = line_labels,
    aes(label = location), col = "black")
```



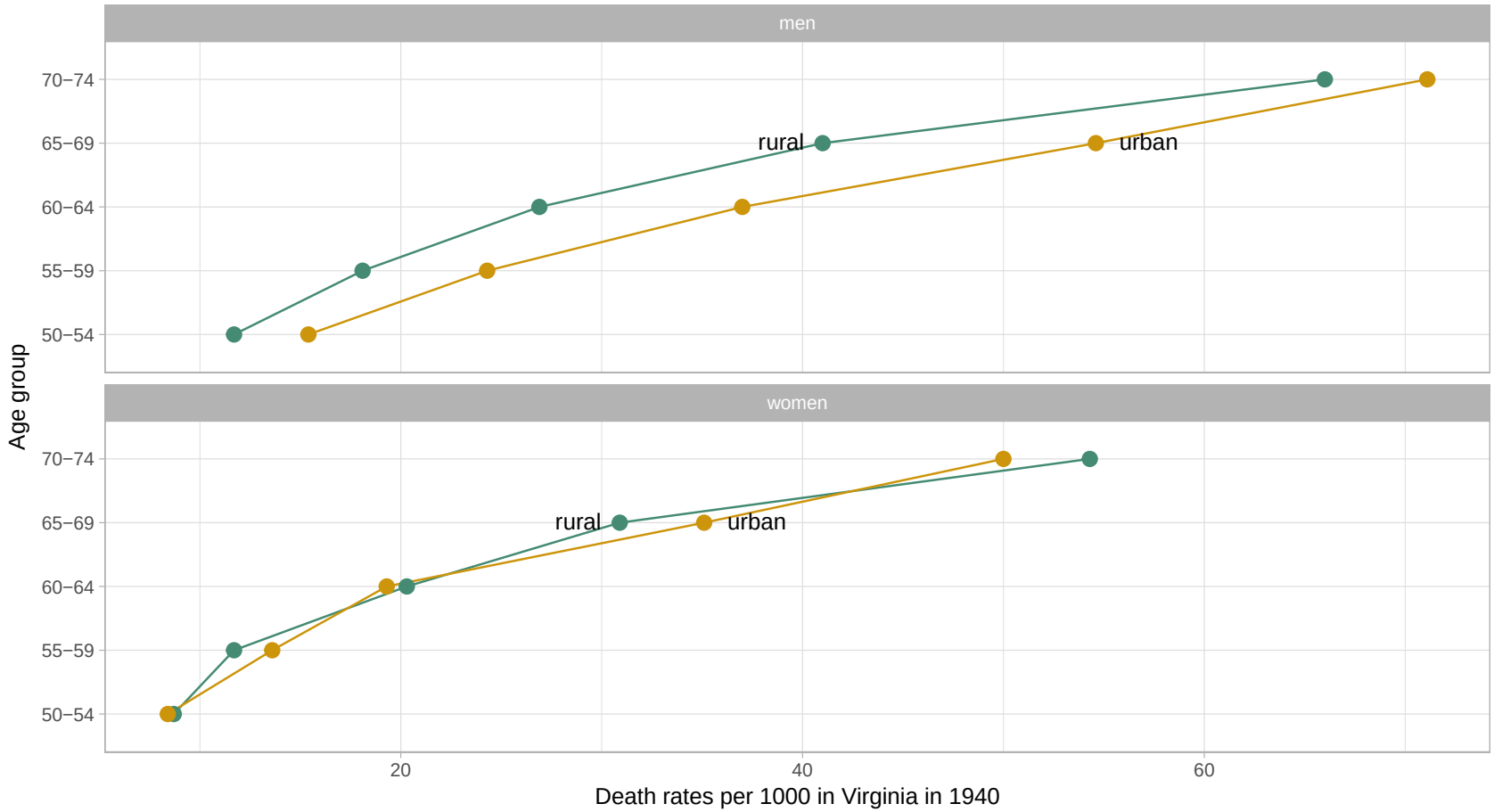
Justify the text labels

```
f14 <- f11 + geom_text(data = line_labels,
    aes(label = location)
    , col = "black"
  , hjust = c(1.4, -0.4, 1.4, -0.4)
  , vjust = 0.4
  )
```


men
women
70–74
65–69
60–64
55–59
50–54
rural
urban
20
40
60
death_rate
age_group
location
rural
urban

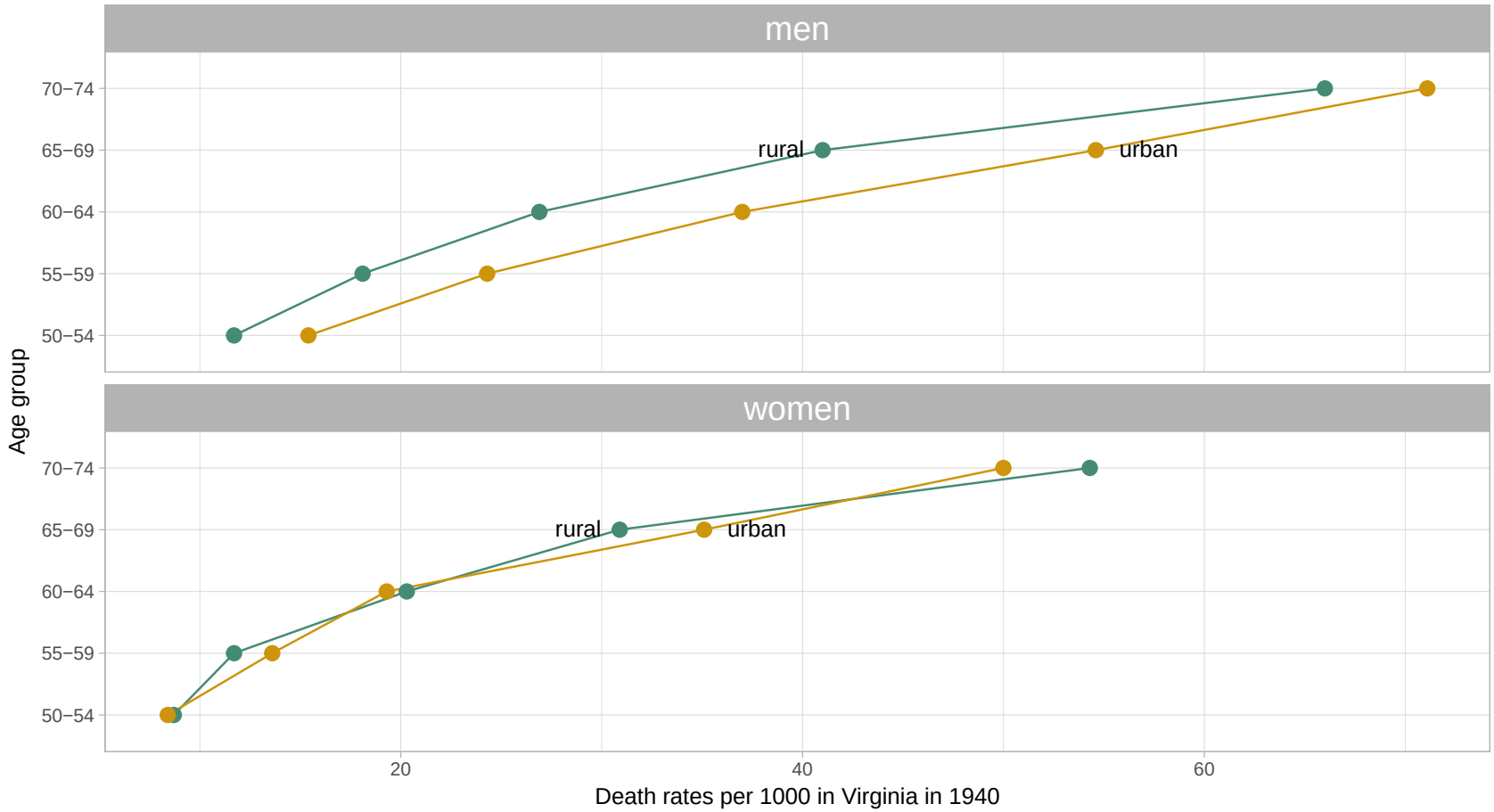
Add labels, edit theme, omit legend

```
f15 <- f14 +
    labs(x = "Death rates per 1000 in Virginia in 1940",
        y = "Age group") +
    theme_light() +
    theme(legend.position = "none")
```


men
women
rural
urban
rural
urban
70–74
65–69
60–64
55–59
50–54
Age group
20
40
60
Death rates per 1000 in Virginia in 1940

Edit text in facet strip

```
f16 <- f15 + theme(strip.text.x = element_text(size = 16))
```



Change the aspect ratio

```
f17 <- f16 + coord_fixed(ratio = 10)
```