THE GRAMMAR OF GRAPHICS AND GGPLOT2

A GGPLOT2 PRIMER

ggplot2 is a set of tools that map data to visual display elements, and that allow the user to control the fine details of plot display.

Most important aspect: *ggplot2* can be used to think about the **logical structure** of the plot.

A *ggplot2* graph has 2 main components (and optional terms):

- aesthetic mappings (**aes** – connections between data and plot elems.)
- plot geometry (**geom** – specifies the type of plot)
- *facets, *coordinates, *scales, *labels, *guides, etc.

[Healey, K., Data Visualization: A Practical Introduction]

GGPLOT2 GRAMMAR

1. Tidy Data

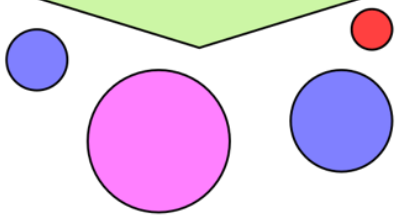
```
p ← ggplot(data = gapminder, ...
```

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

2. Mapping

```
p ← ggplot(data = gapminder, mapping =
      aes(x = gdp, y = lifexp, size = pop,
        color = continent))
```

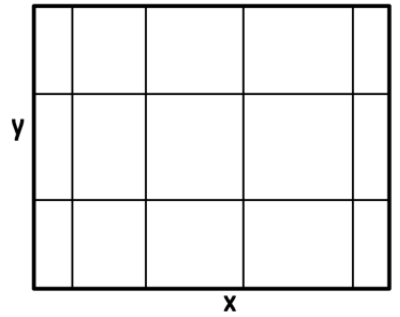
3. Geom



```
p + geom_point()
```

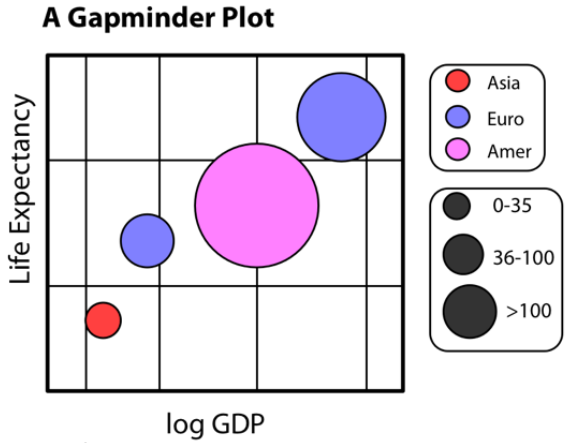
4. Co-Ordinates & Scales

```
p + coord_cartesian() + scale_x_log10()
```



5. Labels & Guides

```
p + labs(x = "log GDP", y = "Life
Expectancy", title = "A Gapminder Plot")
```



GGPLOT2 GRAMMAR – GEOMS

The data source and variables to be plotted are specified *via* ggplot().

The various geom functions specify **how** these variables are to be visually represented

- using points, bars, lines, shaded regions, etc.

There are currently 37+ available geoms.

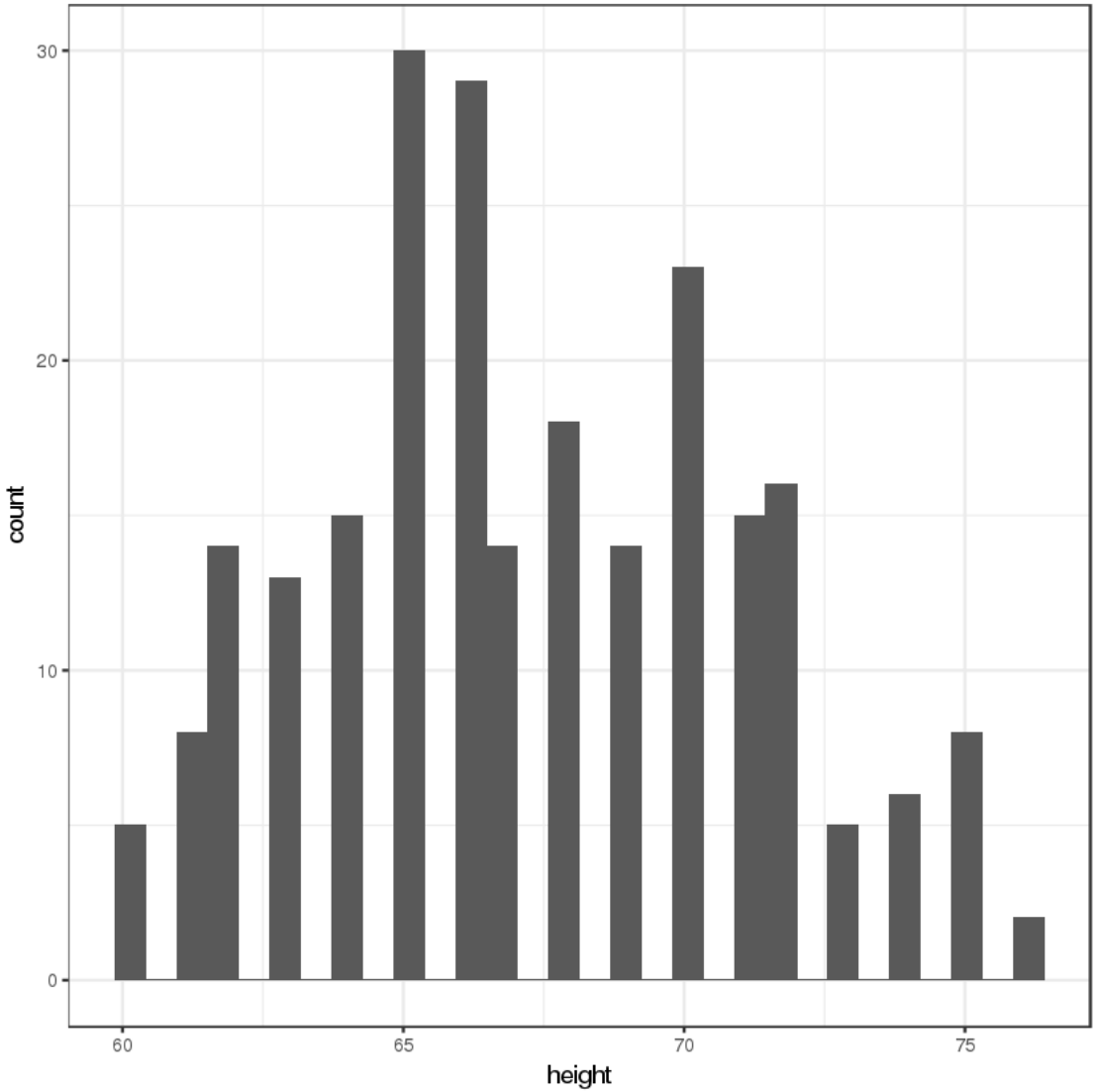
GGPLOT2 GRAMMAR – GEOM()

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
  geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
  geom_boxplot()
```

What do you expect the output to be?

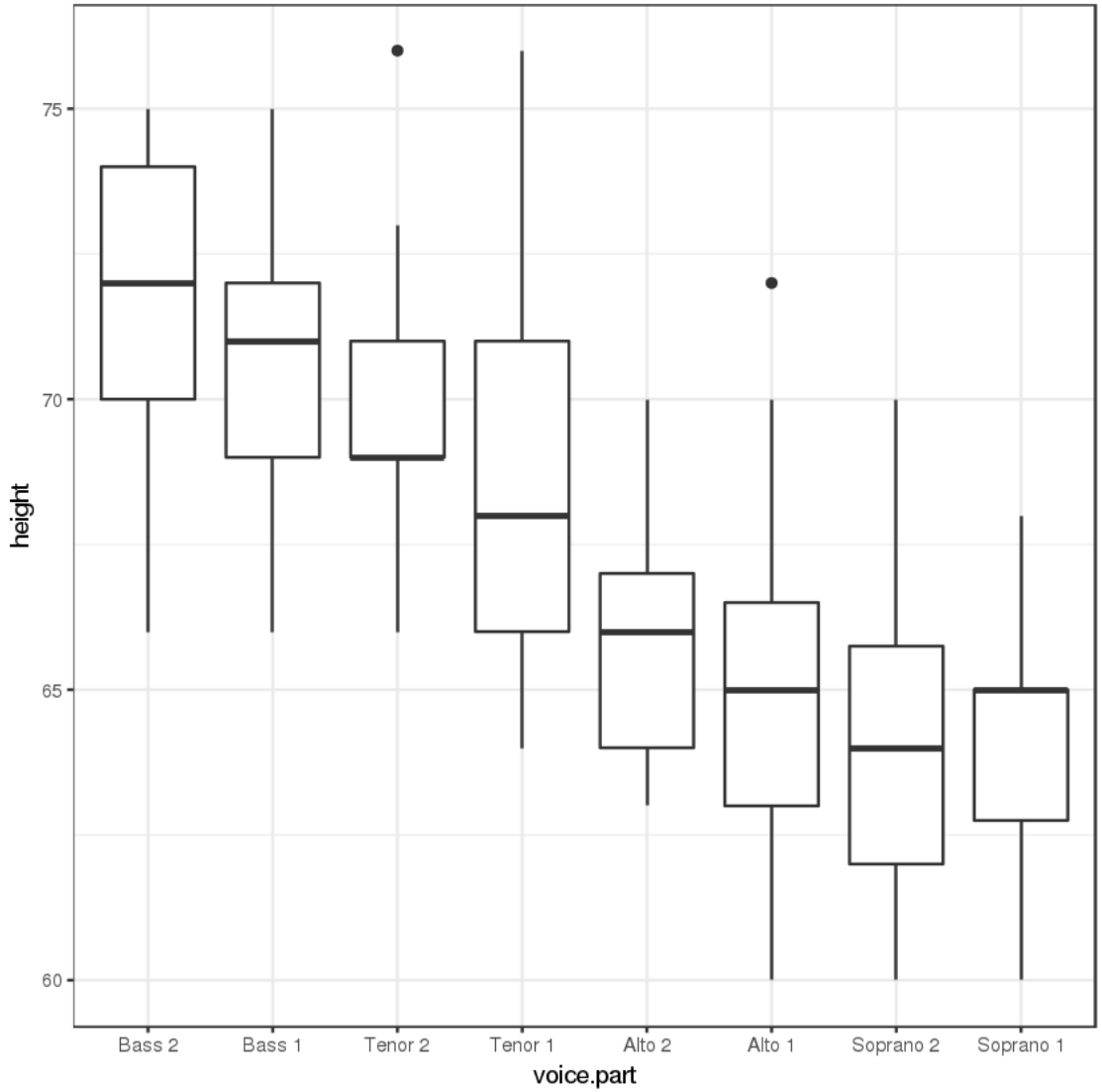
GGPLOT2 GRAMMAR – GEOM()

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
   geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
   geom_boxplot()
```



GGPLOT2 GRAMMAR – GEOM()

```
library("ggplot2")
data(singer, package="lattice")
# Using data from the 1979 ed. of the
# New York Choral Society

# Histogram of heights
ggplot(singer, aes(x=height)) +
   geom_histogram()

# Boxplot of heights by voice part
ggplot(singer, aes(x=voice.part, y=height)) +
   geom_boxplot()
```



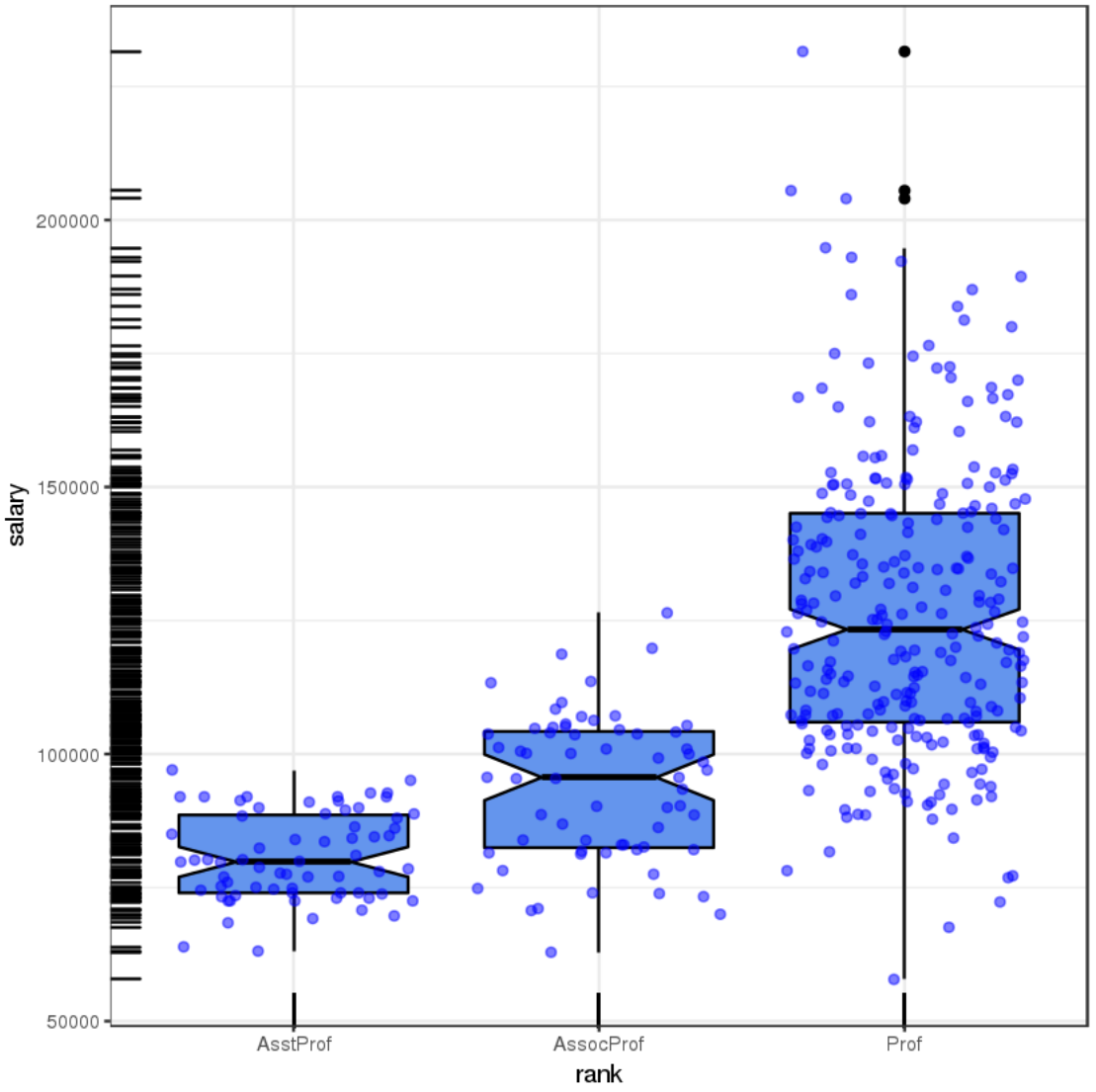
GGPLOT2 GRAMMAR – GEOM()

```
library(ggplot2)
data(Salaries, package="car")
# Using data on salaries of a sample of
# US university professors (2018-2019)
# var: rank, sex, yrs.since.phd, yrs.service, salary

ggplot(Salaries, aes(x=rank, y=salary)) +
  geom_boxplot(fill="cornflowerblue",color="black", notch=TRUE) +
  geom_point(position="jitter", color="blue", alpha=.5) +
  geom_rug(side="l", color="black")
```

What do you expect the output to be?

GGPLOT2 GRAMMAR – GEOM()

```
library(ggplot2)
data(Salaries, package="car")
# Using data on salaries of a sample of
# US university professors (2018-2019)
# var: rank, sex, yrs.since.phd, yrs.service, salary

ggplot(Salaries, aes(x=rank, y=salary)) +
   geom_boxplot(fill="cornflowerblue",color="black", notch=TRUE) +
   geom_point(position="jitter", color="blue", alpha=.5) +
   geom_rug(side="l", color="black")
```



GGPLOT2 GRAMMAR – AESTHETICS

Aesthetics refer to the displayed attributes of the data.

They map the data to an attribute (such as the size or shape of a marker) and generate an appropriate legend.

Aesthetics are specified with the `aes()` function.

Aesthetics can be specified within the data function or within a geom. If they're specified within the data function then they apply to all specified geoms.

GGPLOT2 GRAMMAR – AESTHETICS

The aesthetics available to `geom_point()` (scatterplot), as an example, are:

- `x`, `y`, `alpha`, `color`, `fill`, `shape`, `size`

Important difference between specifying characteristics (like colour and shape) inside and outside the `aes()` function

- inside: assigned colour or shape automatically based on the data.
- outside: not mapped to data.

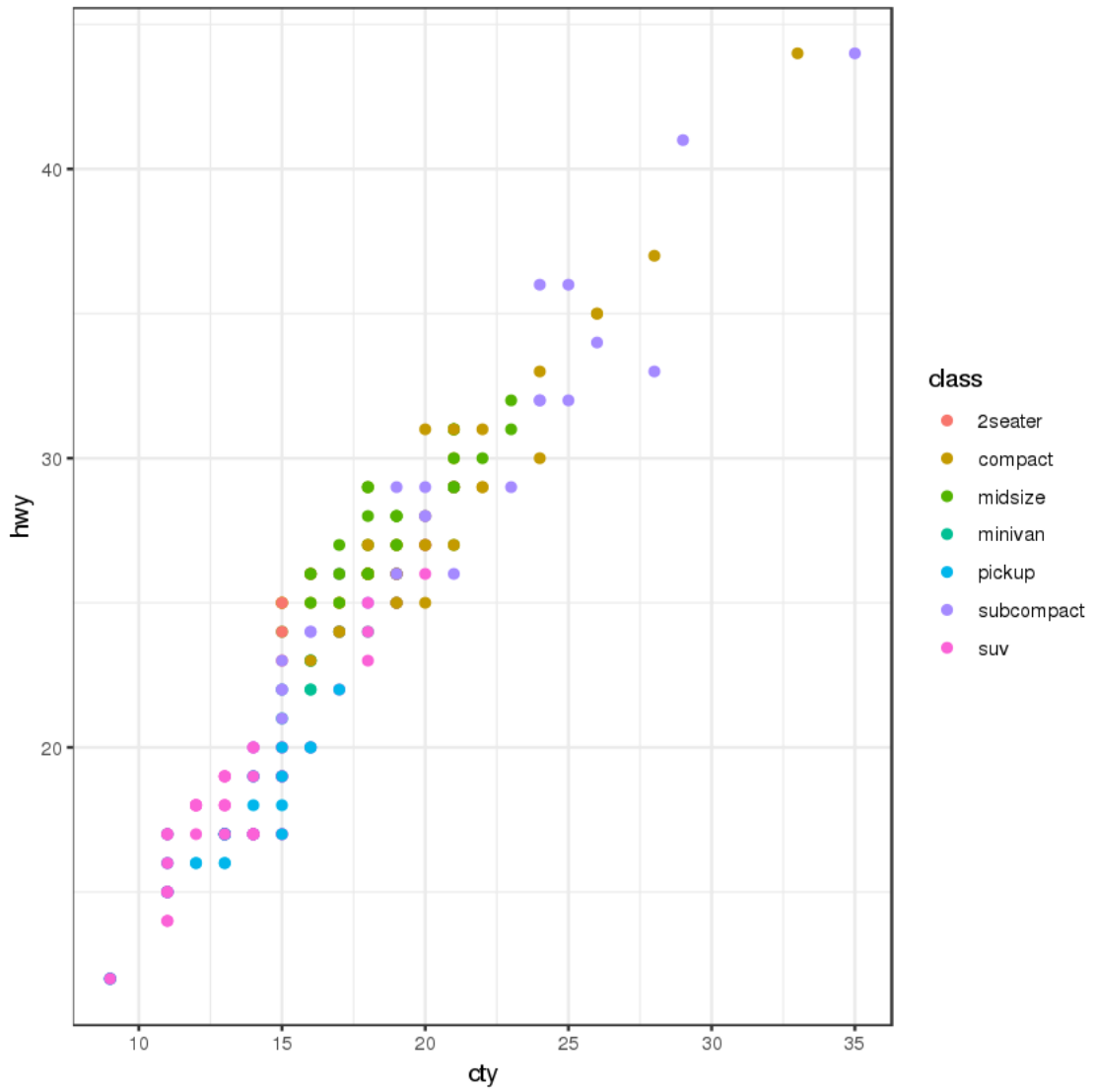
GGPLOT2 GRAMMAR – AES()

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```

What do you expect the output to be?

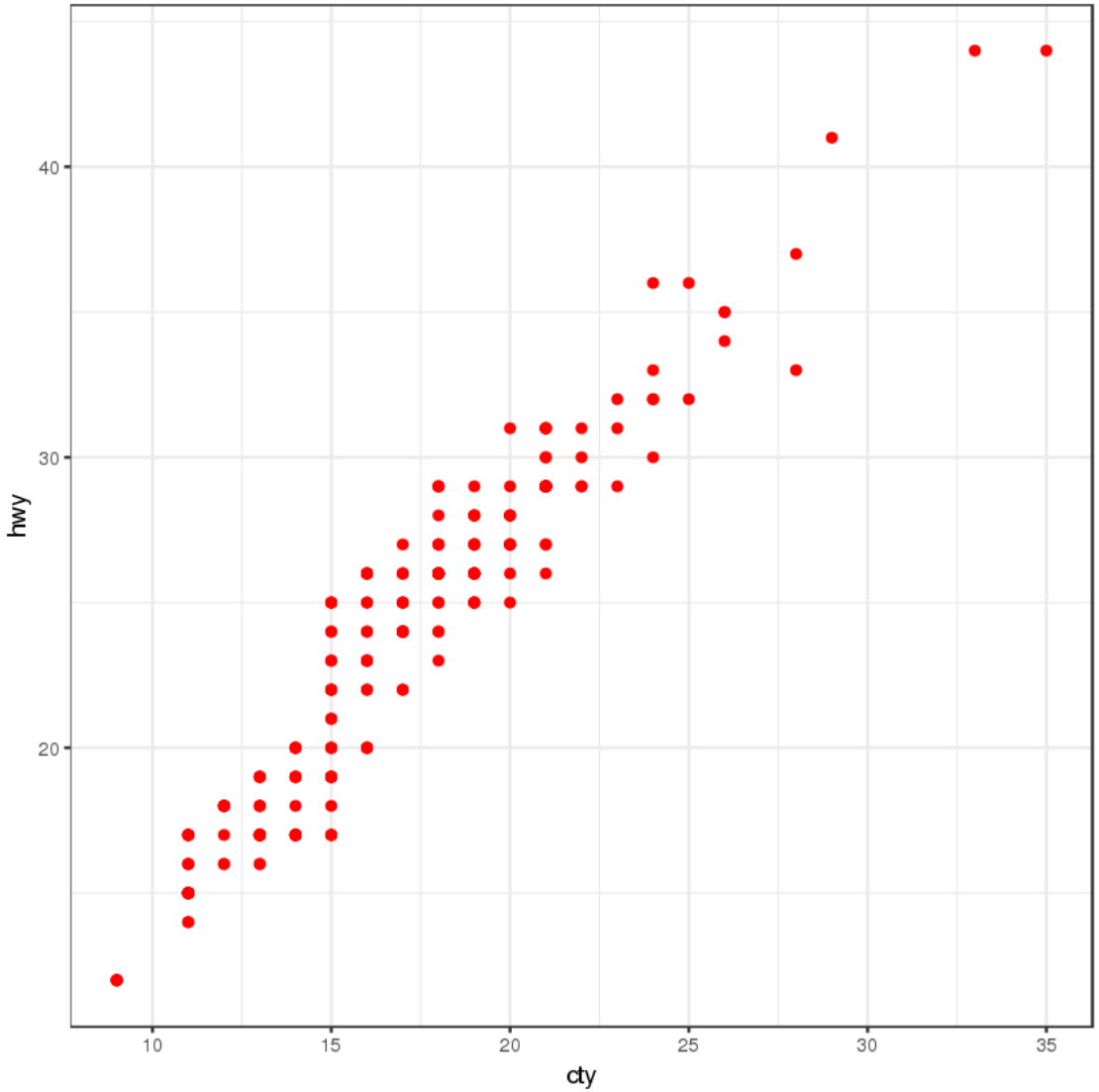
GGPLOT2 GRAMMAR – AES()

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```



GGPLOT2 GRAMMAR – AES()

```
library(ggplot2)
# Using the mpg dataset

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(aes(colour = class))

# specifying characteristics inside aes()
ggplot(mpg, aes(cty, hwy)) +
  geom_point(colour = "red")
```



GGPLOT2 GRAMMAR – FACET

In *ggplot2* parlance, small multiples are referred to as **facets**

- `facet_wrap()`,`facet_grid()`

By default, all panels (one for each factor) share the same axes (scale-wise).

Separating the graph into a sequence of smaller, side-by-side plots makes it easier to enact comparisons.

Wraps only display those small multiples for which there is data, grids display all multiples, even the empty ones.

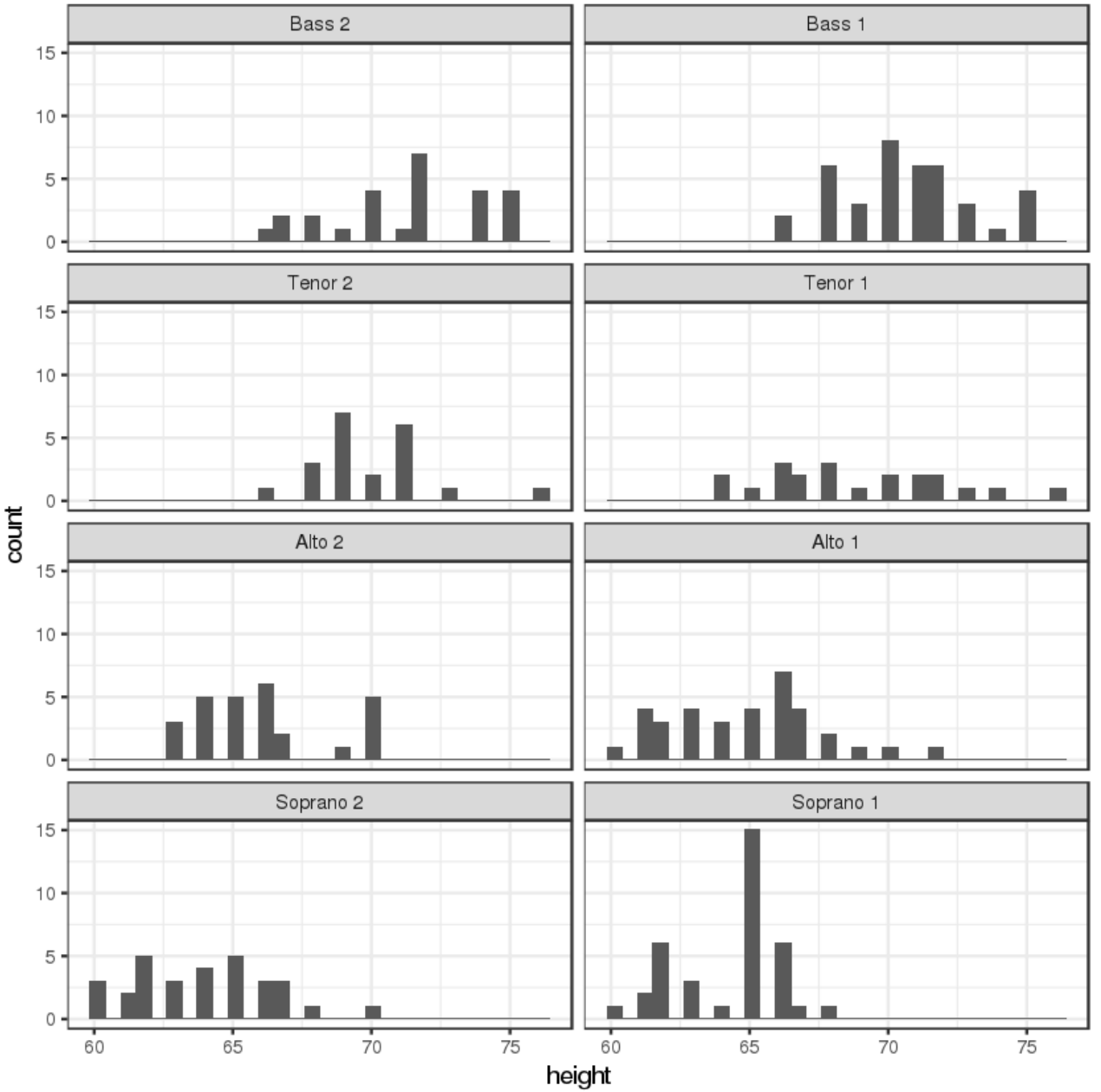
GGPLOT2 GRAMMAR – FACET

```
data(singer, package="lattice")
library(ggplot2)
ggplot(data=singer, aes(x=height)) +
      geom_histogram() +
      facet_wrap(~voice.part, nrow=4)
```

What do you expect the output to be?

GGPLOT2 GRAMMAR – FACET

```
data(singer, package="lattice")
library(ggplot2)
ggplot(data=singer, aes(x=height)) +
      geom_histogram() +
      facet_wrap(~voice.part, nrow=4)
```



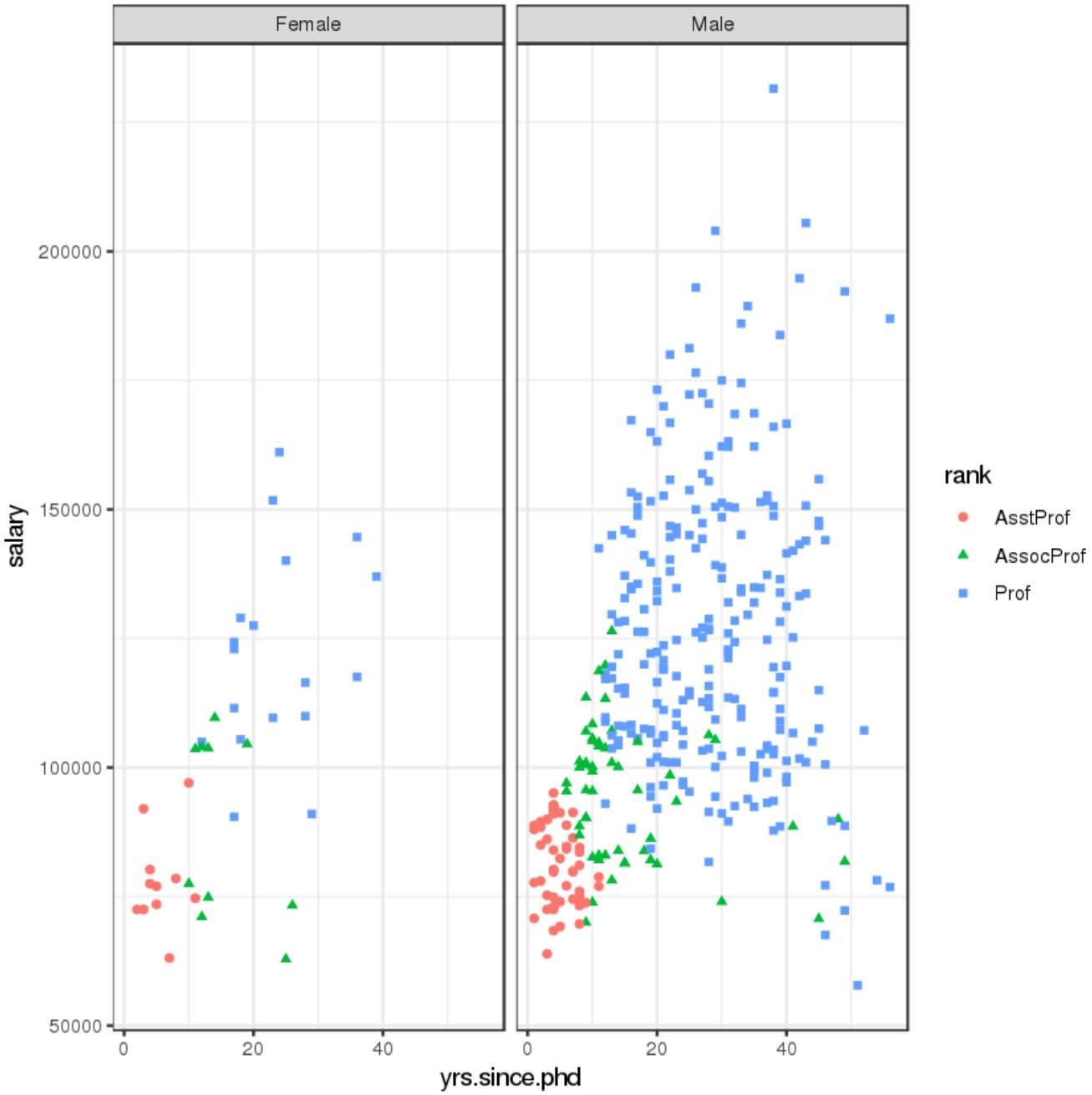
GGPLOT2 GRAMMAR – FACET

```
data(Salaries, package="car")
library(ggplot2)
ggplot(Salaries, aes(x=yrs.since.phd,
 y=salary, color=rank, shape=rank)) +
   geom_point() +
   facet_grid(~sex)
```

What do you expect the output to be?

GGPLOT2 GRAMMAR – FACET

```
data(Salaries, package="car")
library(ggplot2)
ggplot(Salaries, aes(x=yrs.since.phd,
 y=salary, color=rank, shape=rank)) +
   geom_point() +
   facet_grid(~sex)
```



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

The `custdata.tsv` file (in the `Data` folder) is derived from Census PUMS data.

The business objective is to predict whether a customer has health insurance. This synthetic dataset contains customer information for individuals whose health insurance status is **known**.

We start by importing the data into a data frame using the `read.delim()` function (to handle the odd file format).

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

```
> df <- read.delim("Data/custdata.tsv")

> class(df)

## [1] "data.frame"

> dim(df)

## [1] 1000   11
```

We see that we have 1000 observations of 11 variables.

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

```
> head(df)
##   custid sex is.employed income  marital.stat health.ins
## 1   2068   F          NA  11300       Married       TRUE
## 2   2073   F          NA      0       Married       TRUE
## 3   2848   M        TRUE   4500 Never Married      FALSE
## 4   5641   M        TRUE  20000 Never Married      FALSE
## 5   6369   F        TRUE  12000 Never Married       TRUE
## 6   8322   F        TRUE 180000 Never Married       TRUE
##                    housing.type recent.move num.vehicles age state.of.res
## 1     Homeowner free and clear       FALSE            2  49     Michigan
## 2                      Rented        TRUE            3  40      Florida
## 3                      Rented        TRUE            3  22      Georgia
## 4        Occupied with no rent       FALSE            0  22   New Mexico
## 5                      Rented        TRUE            1  31      Florida
## 6 Homeowner with mortgage/loan       FALSE            1  40     New York
```

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

```
> summary(df)
##      custid         sex     is.employed         income                   marital.stat      health.ins
##  Min.   :   2068   F:440   Mode :logical   Min.   : -8700     Divorced/Separated:155   Mode :logical
##  1st Qu.: 345667   M:560   FALSE:73        1st Qu.: 14600     Married            :516   FALSE:159
##  Median : 693403           TRUE :599       Median : 35000     Never Married      :233   TRUE :841
##  Mean   : 698500           NA's :328       Mean   : 53505     Widowed            : 96
##  3rd Qu.:1044606                           3rd Qu.: 67000
##  Max.   :1414286                           Max.   :615000
##                      housing.type     recent.move    num.vehicles        age              state.of.res
##  Homeowner free and clear    :157   Mode :logical   Min.   :0.000   Min.   :  0.0   California  :100
##  Homeowner with mortgage/loan:412   FALSE:820       1st Qu.:1.000   1st Qu.: 38.0   New York    : 71
##  Occupied with no rent       : 11   TRUE :124       Median :2.000   Median : 50.0   Pennsylvania: 70
##  Rented                      :364   NA's :56        Mean   :1.916   Mean   : 51.7   Texas       : 56
##  NA's                        : 56                   3rd Qu.:2.000   3rd Qu.: 64.0   Michigan    : 52
##                                                     Max.   :6.000   Max.   :146.7   Ohio        : 51
##                                                     NA's   :56                      (Other)     :600
```

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

The fact that three of the variables have the same number of missing values means that it is possible that the observations that are missing measurements for **`housing.type`**, **`recent.move`**, and **`num.vehicles`** are the same.

We still need to check (try it!).

As per the ranges, something is definitely fishy with **`income`** and **`age`**:

What does it mean for income to be negative? For a customer to be 0 or 146.7 y.o.?

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

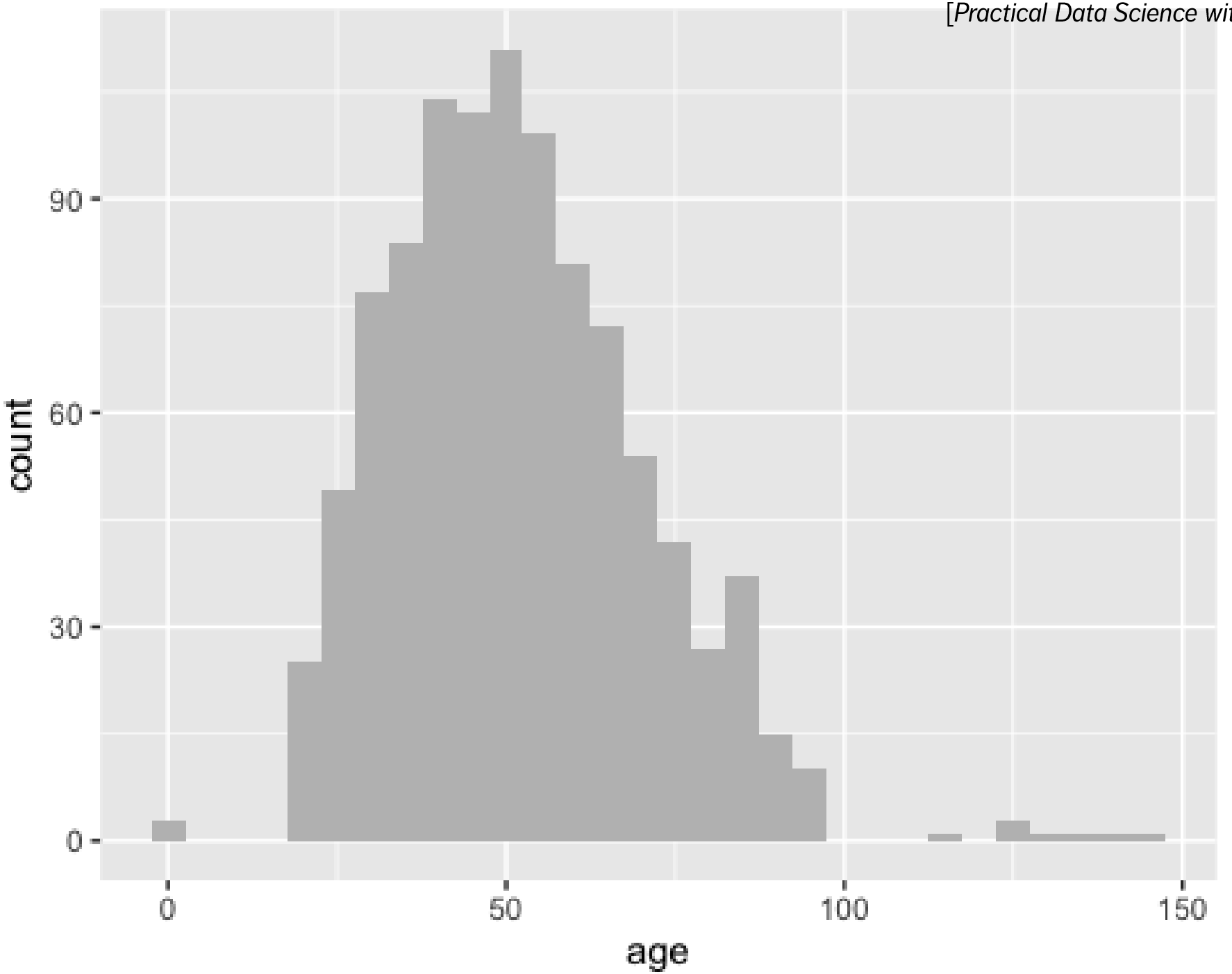
We use `ggplot2` to visually explore the data. Note that:

- the `ggplot()` function works only on data frames (or on tibbles)
- it does not create a graph, it creates an object
- graphs are produced from layers, which are *added* to the object
- aesthetics are visual elements of the graph, e.g., x and y position, marker size, colours, etc.

We start by providing univariate visualizations, starting with the **`age`** variable.

```
> library(ggplot2)
> ggplot(df) +
    geom_histogram(aes(x=age), binwidth=5, fill="gray")
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

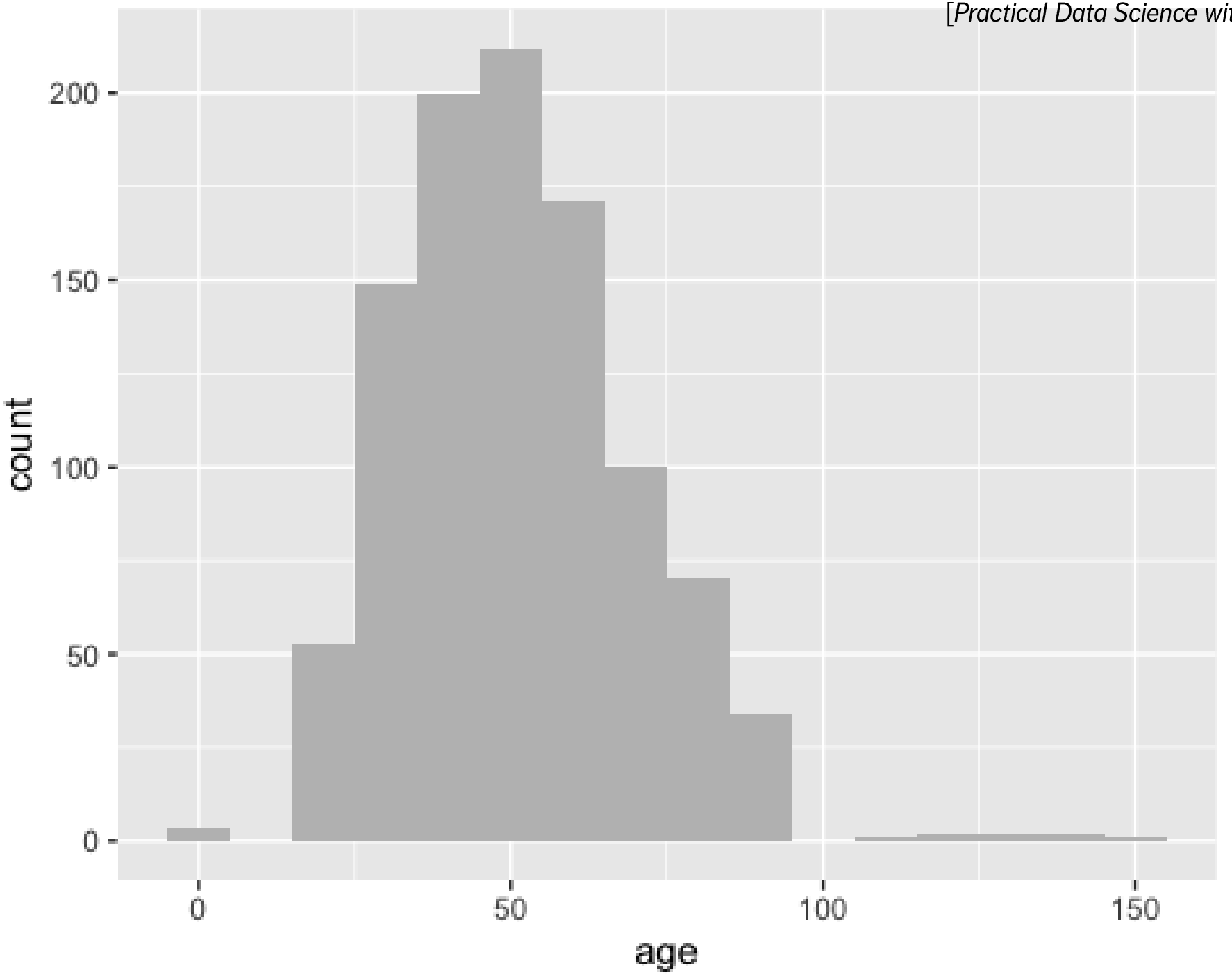


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

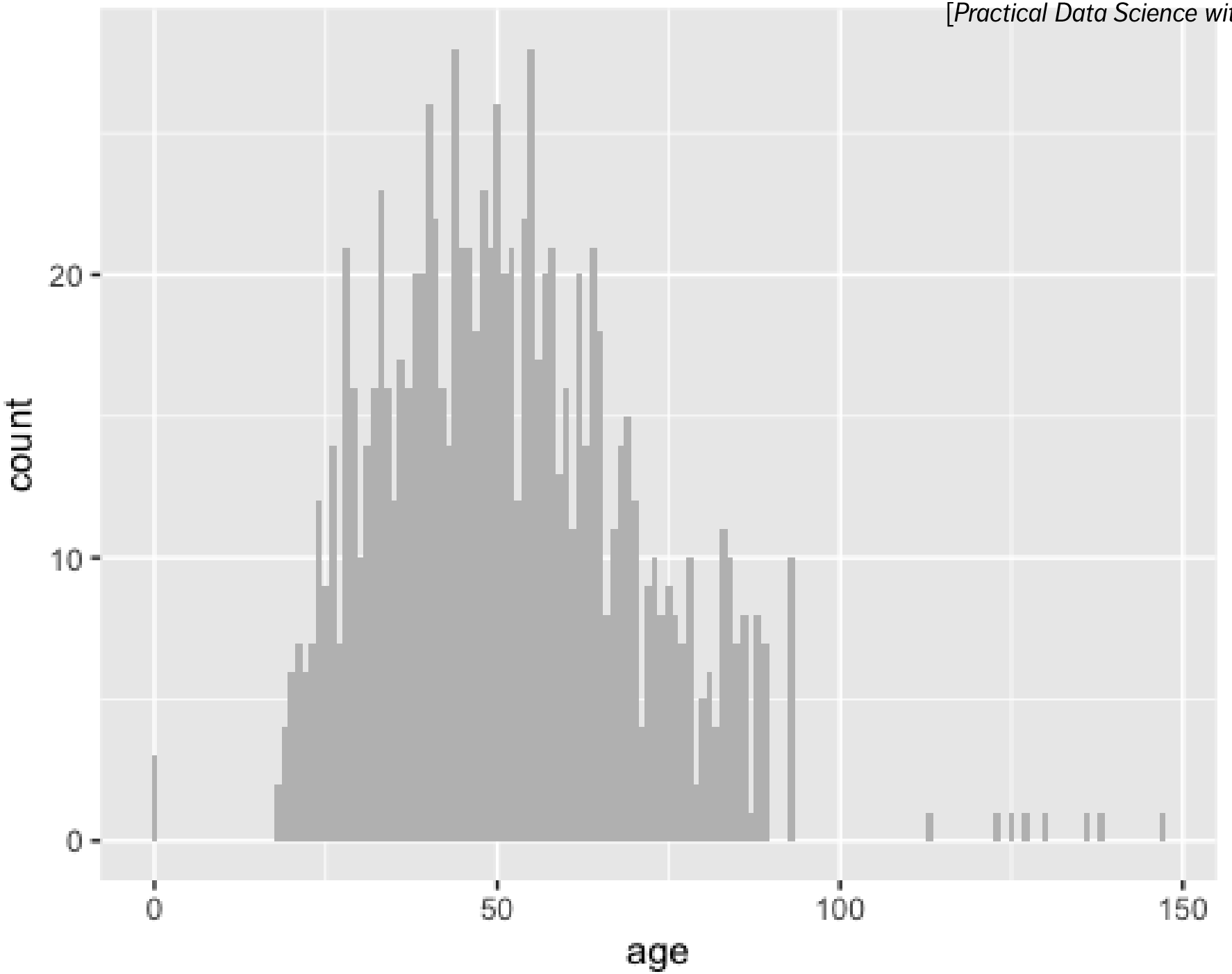
What happens if we use a different bin width?

```
> ggplot(df) +
   geom_histogram(aes(x=age), binwidth=10, fill="gray")

> ggplot(df) +
   geom_histogram(aes(x=age), binwidth=1, fill="gray")
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]

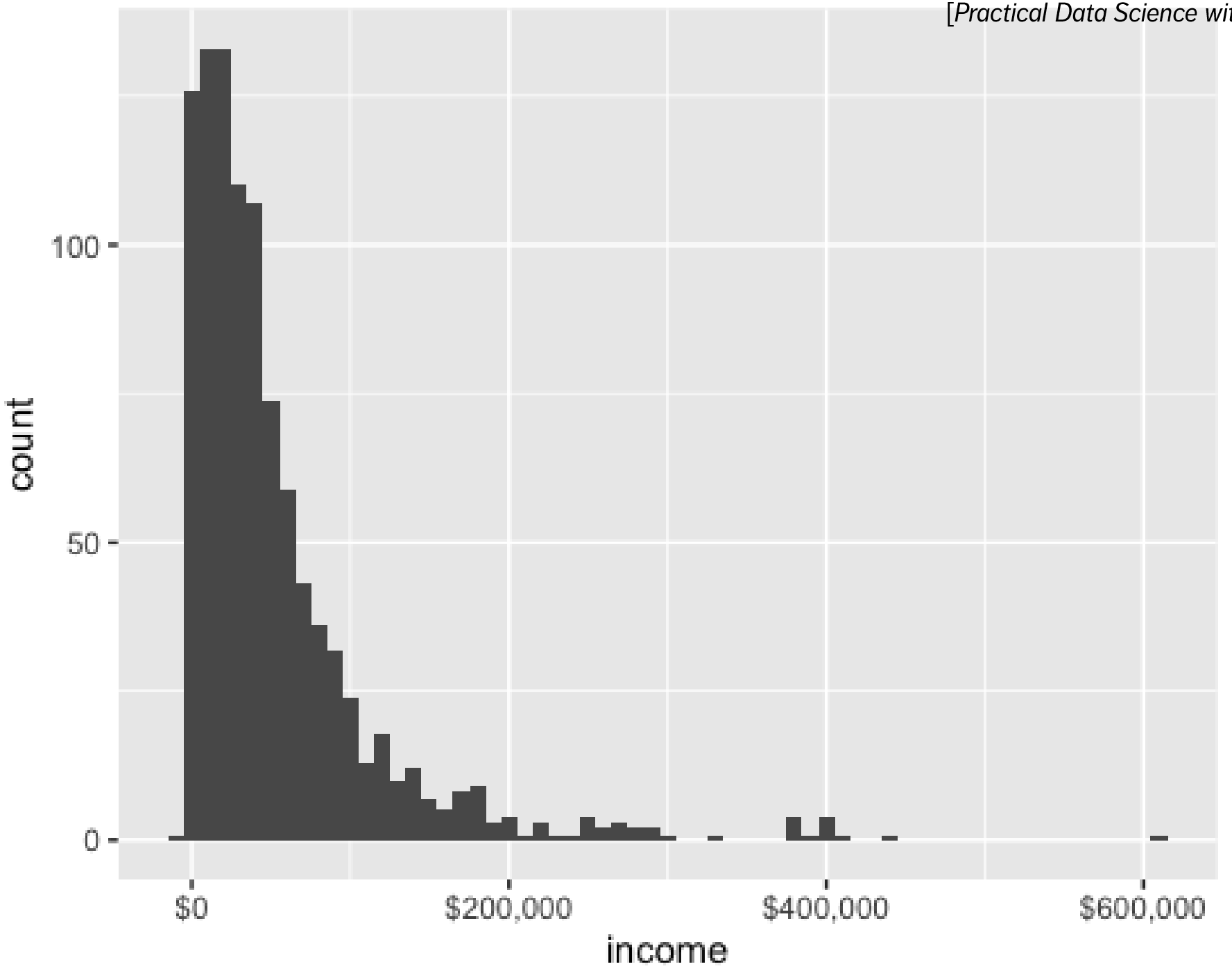


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

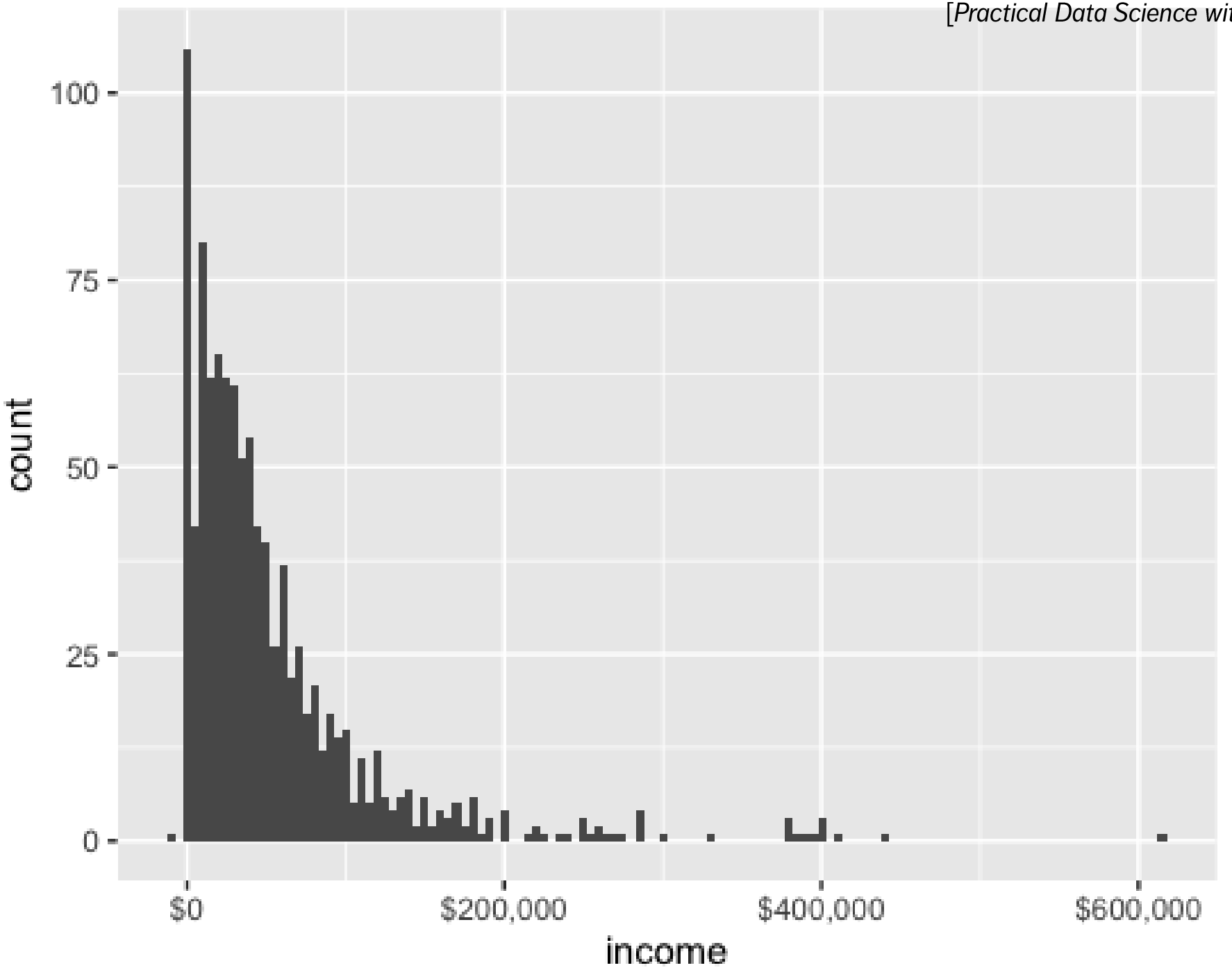
We can also get some (univariate) information about the `income` variable:

```
> library(scales) # to label as dollars

> ggplot(df) +
    geom_histogram(aes(x=income), binwidth = 10000) +
    scale_x_continuous(labels=dollar)

> ggplot(df) +
    geom_histogram(aes(x=income), binwidth = 5000) +
    scale_x_continuous(labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]

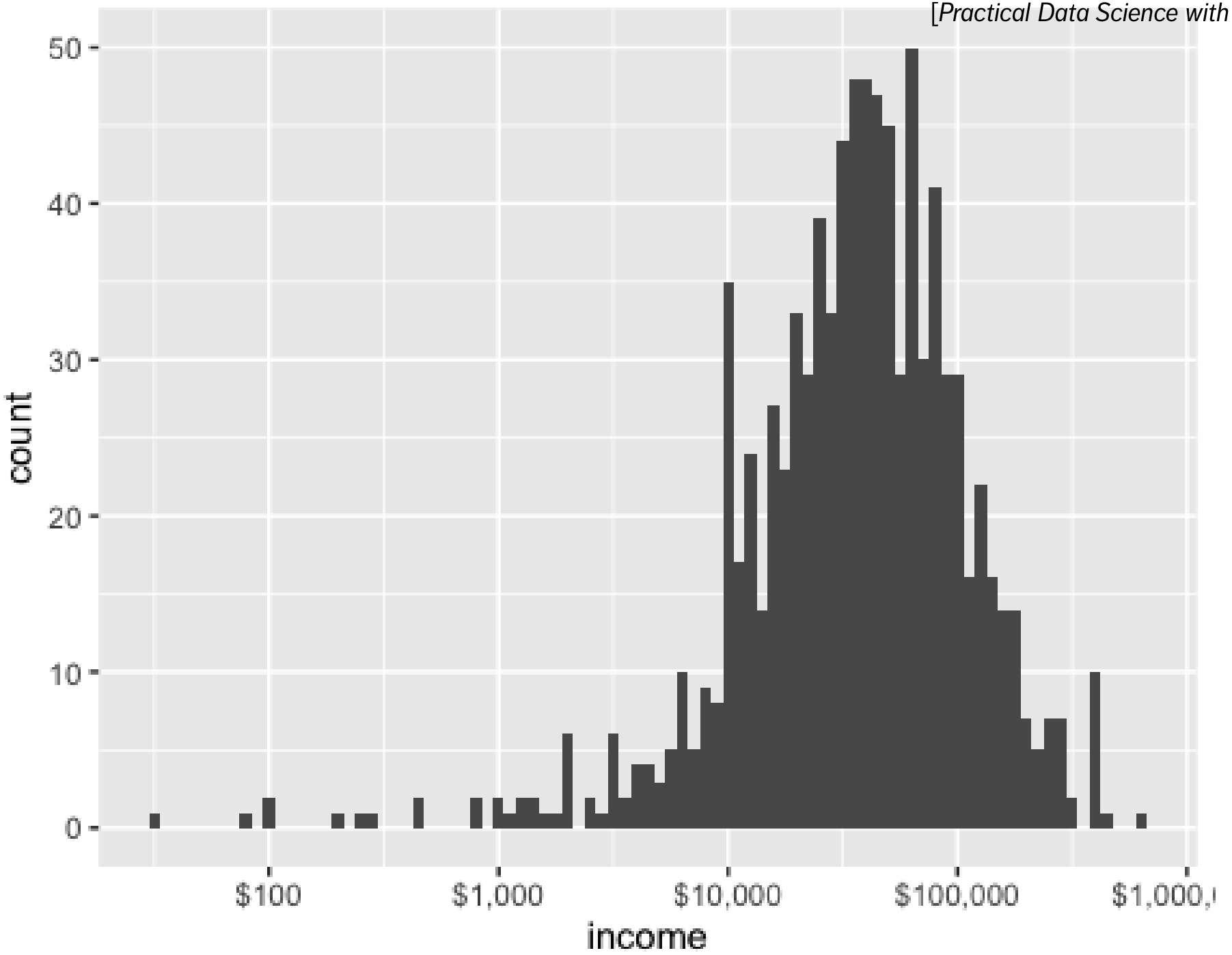


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

We have already seen that there are negative income values in the dataset. Let's restrict the data to those customers with positive values, and display using a logarithmic scale.

```
> df.2 <- subset(df, income > 0)

> ggplot(df.2) +
    geom_histogram(aes(x=income), binwidth = 0.05) +
    scale_x_log10(breaks=10^(1:6), labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

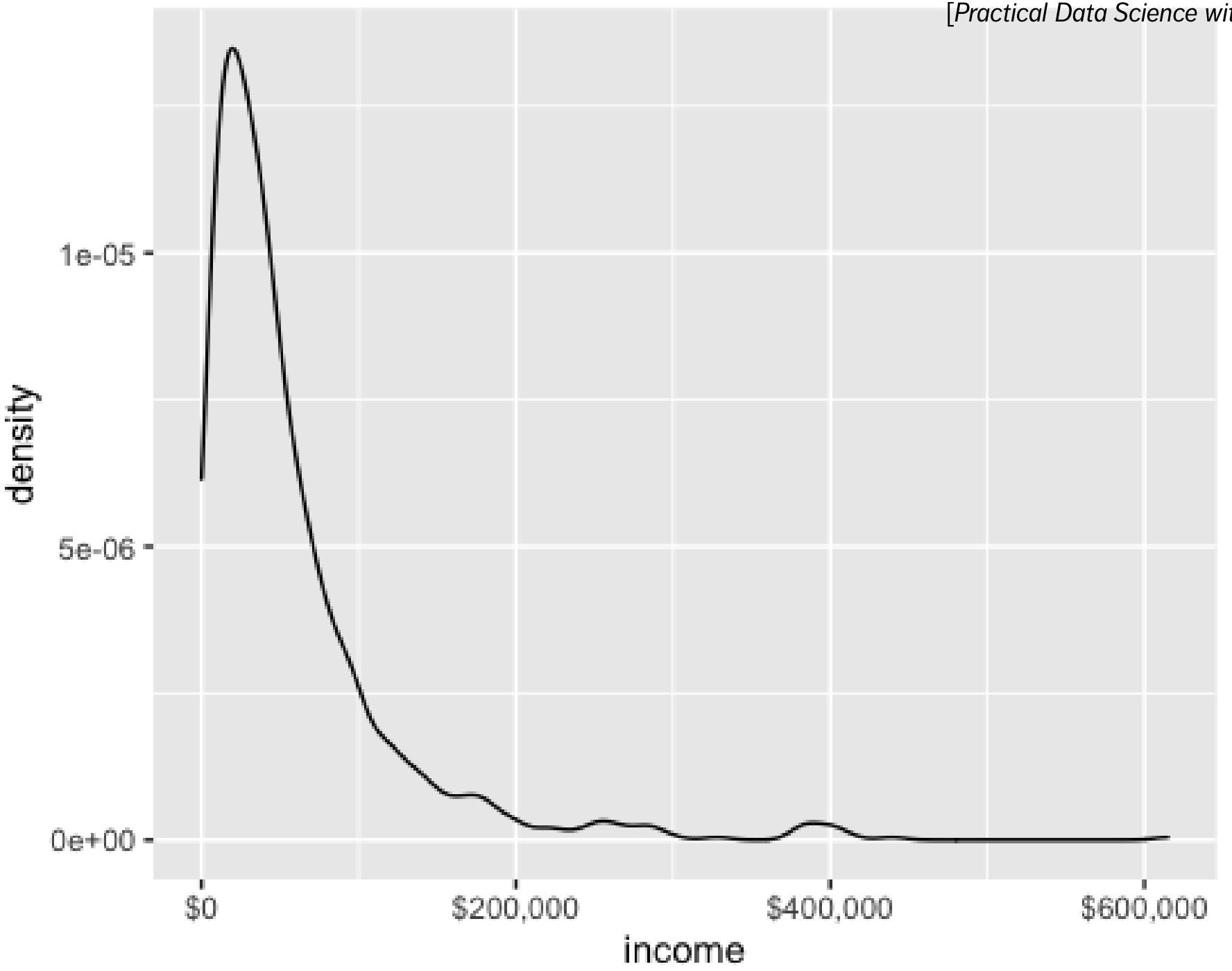


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

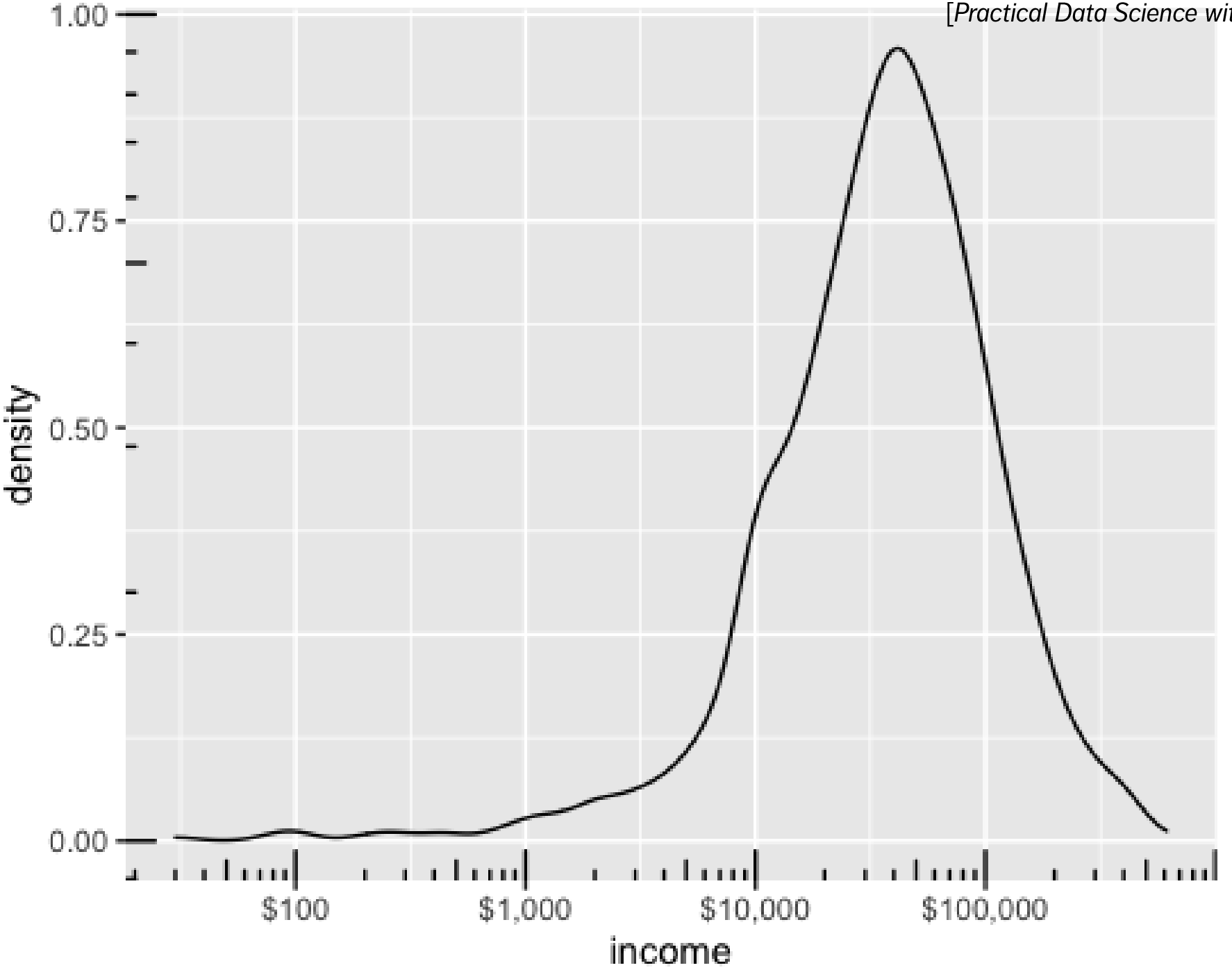
The density plot estimates the probability distribution function.

```
> library(scales) # to label as dollars

> ggplot(df.2) +
    geom_density(aes(x=income)) +
    scale_x_continuous(labels=dollar)  # heavy tails

> ggplot(df.2) +
    geom_density(aes(x=income)) +
    scale_x_log10(breaks=10^(2:5), labels=dollar) +
    annotation_logticks()
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



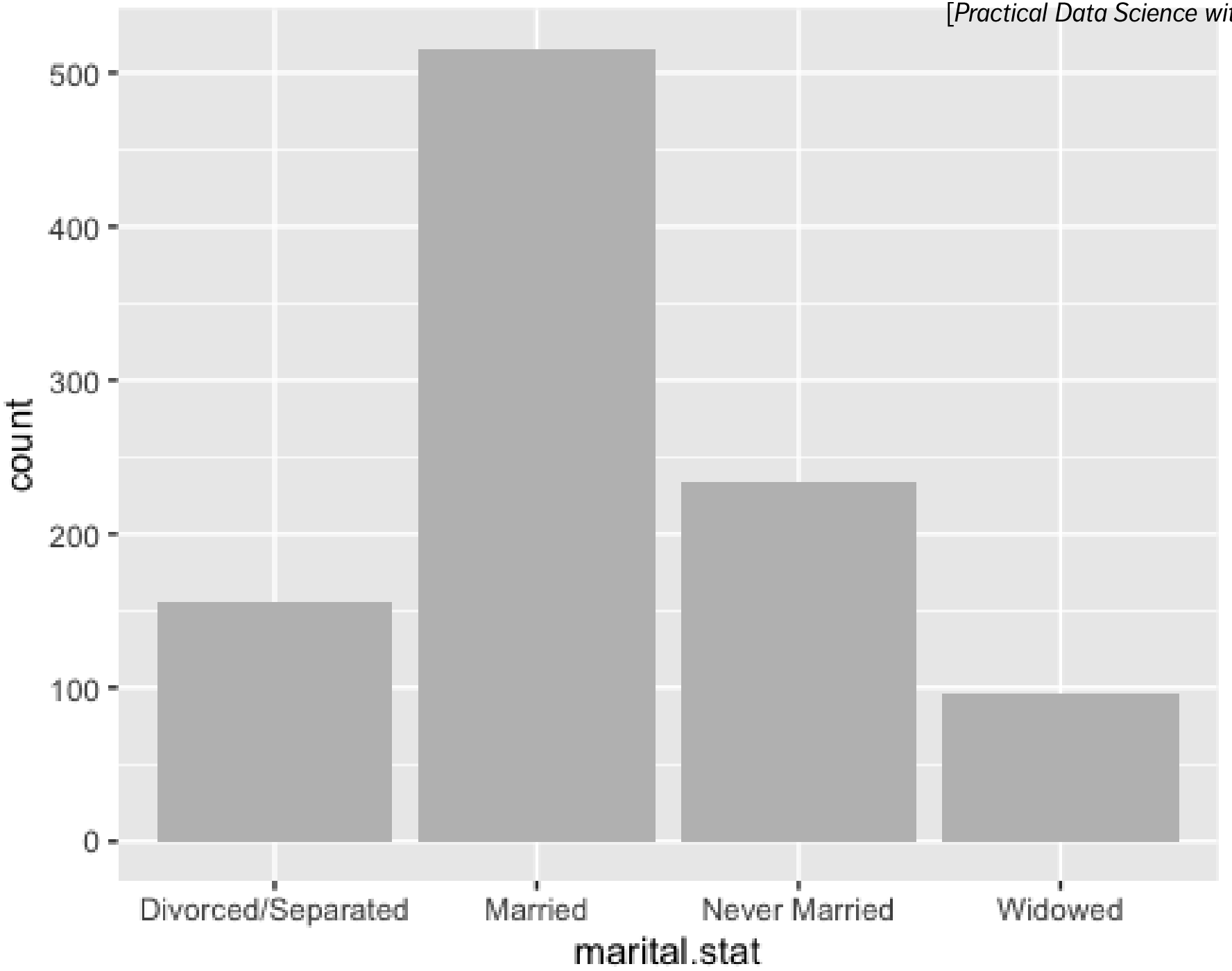
USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

What can we say about the distribution of marital status?

```
> ggplot(df) +
   geom_bar(aes(x=marital.stat), fill="gray")
```

Anything surprising so far?

[*Practical Data Science with R*, N. Zumel, J. Mount]

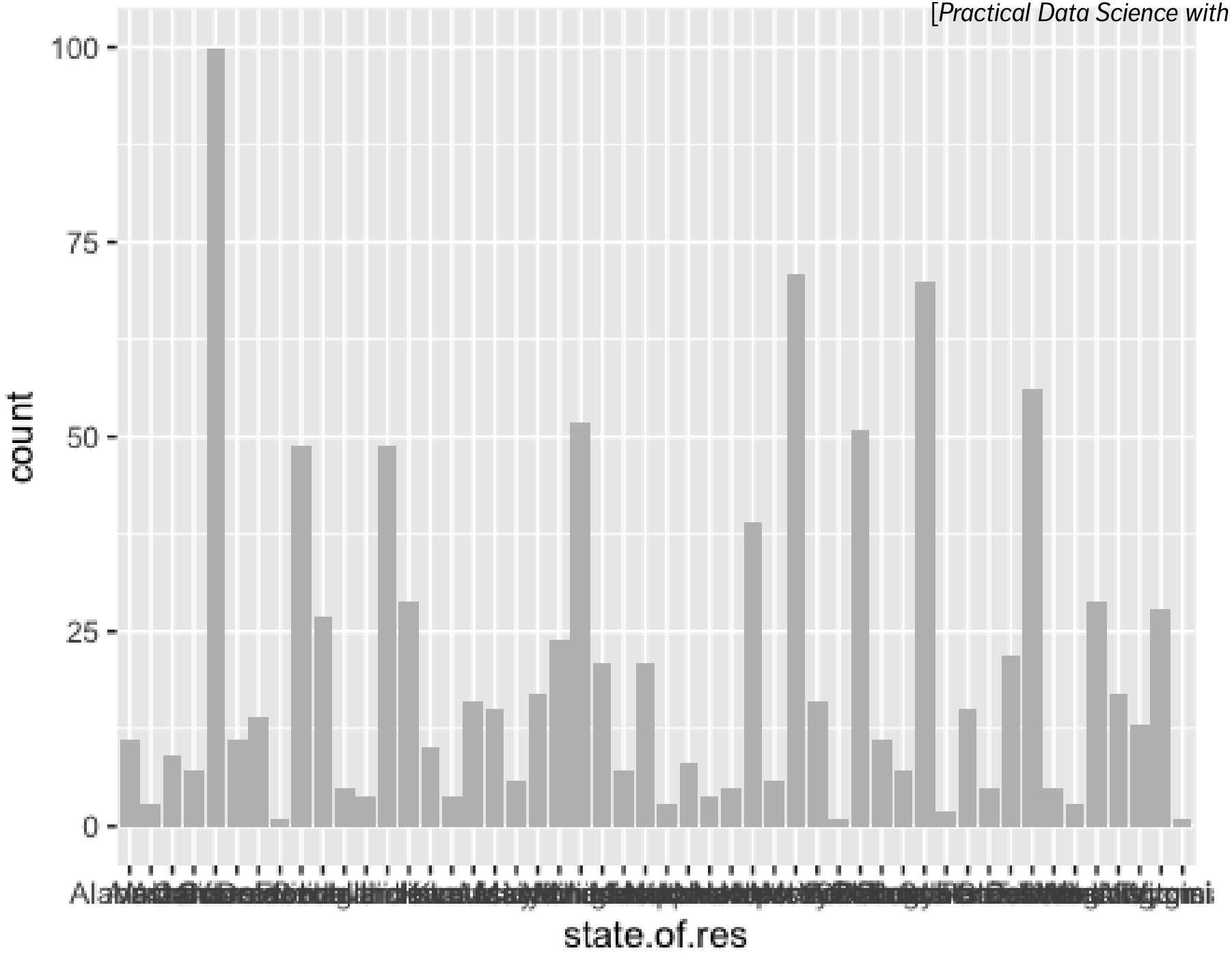


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

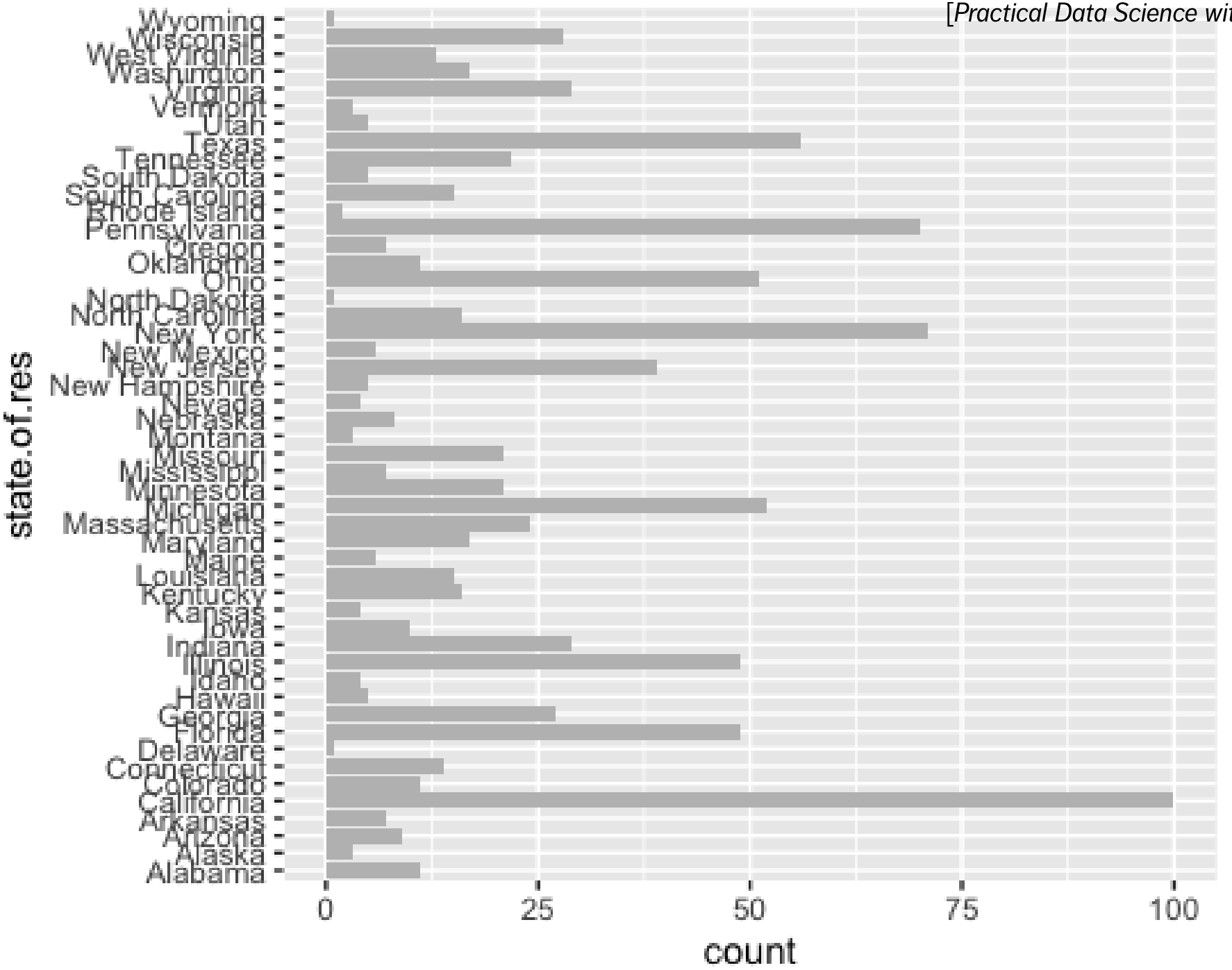
For variables with 10+ levels (**`state.of.res`**), the charts can get busy:

```
> ggplot(df) +
    geom_bar(aes(x=state.of.res), fill="gray")

> ggplot(df) +
    geom_bar(aes(x=state.of.res), fill="gray") +
    coord_flip()

> ggplot(df) +
    geom_bar(aes(x=state.of.res), fill="gray") +
    coord_flip() +
    theme(axis.text.y=element_text(size=rel(0.6)))
```

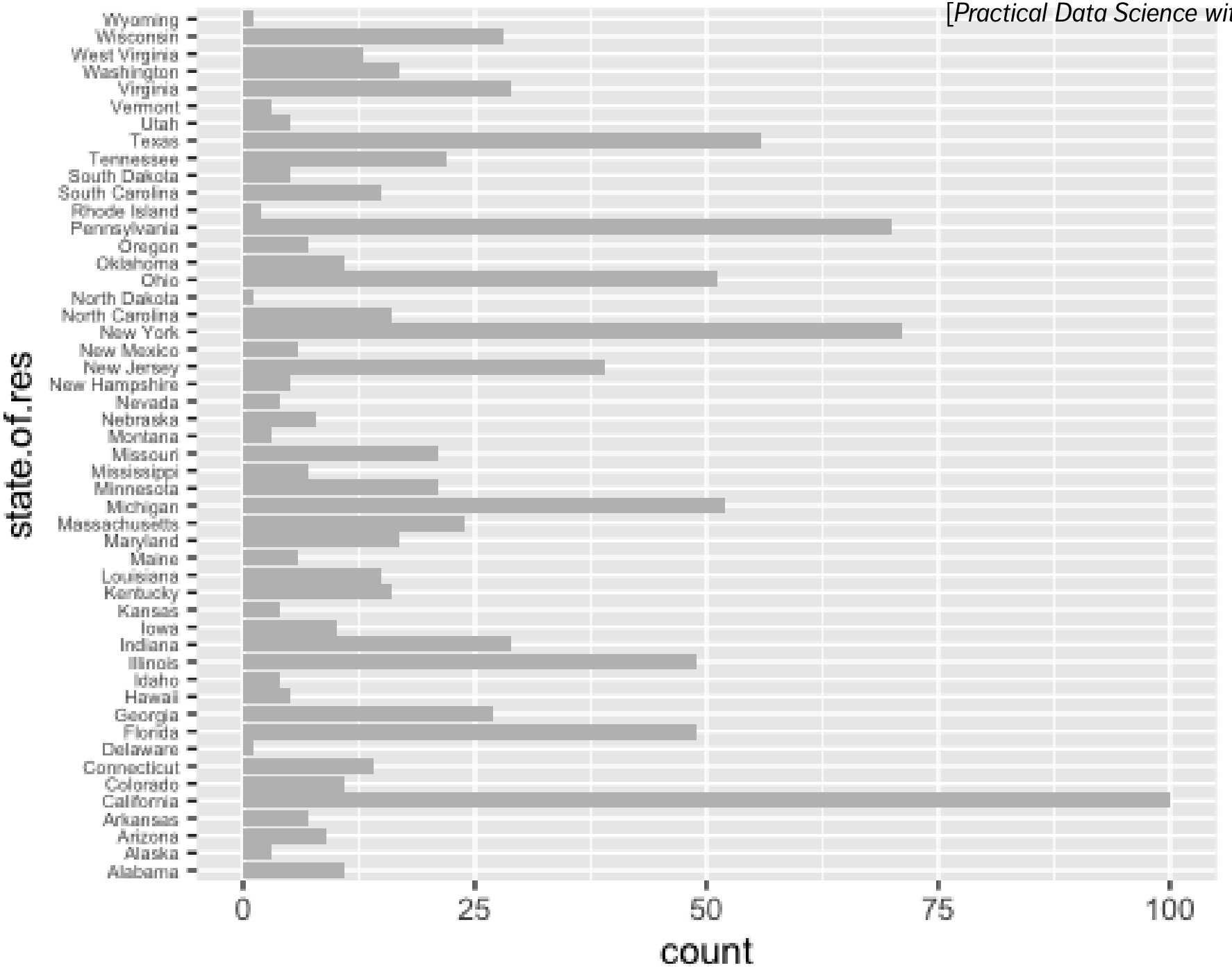
[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

Currently, the chart displays the states ordered alphabetically.

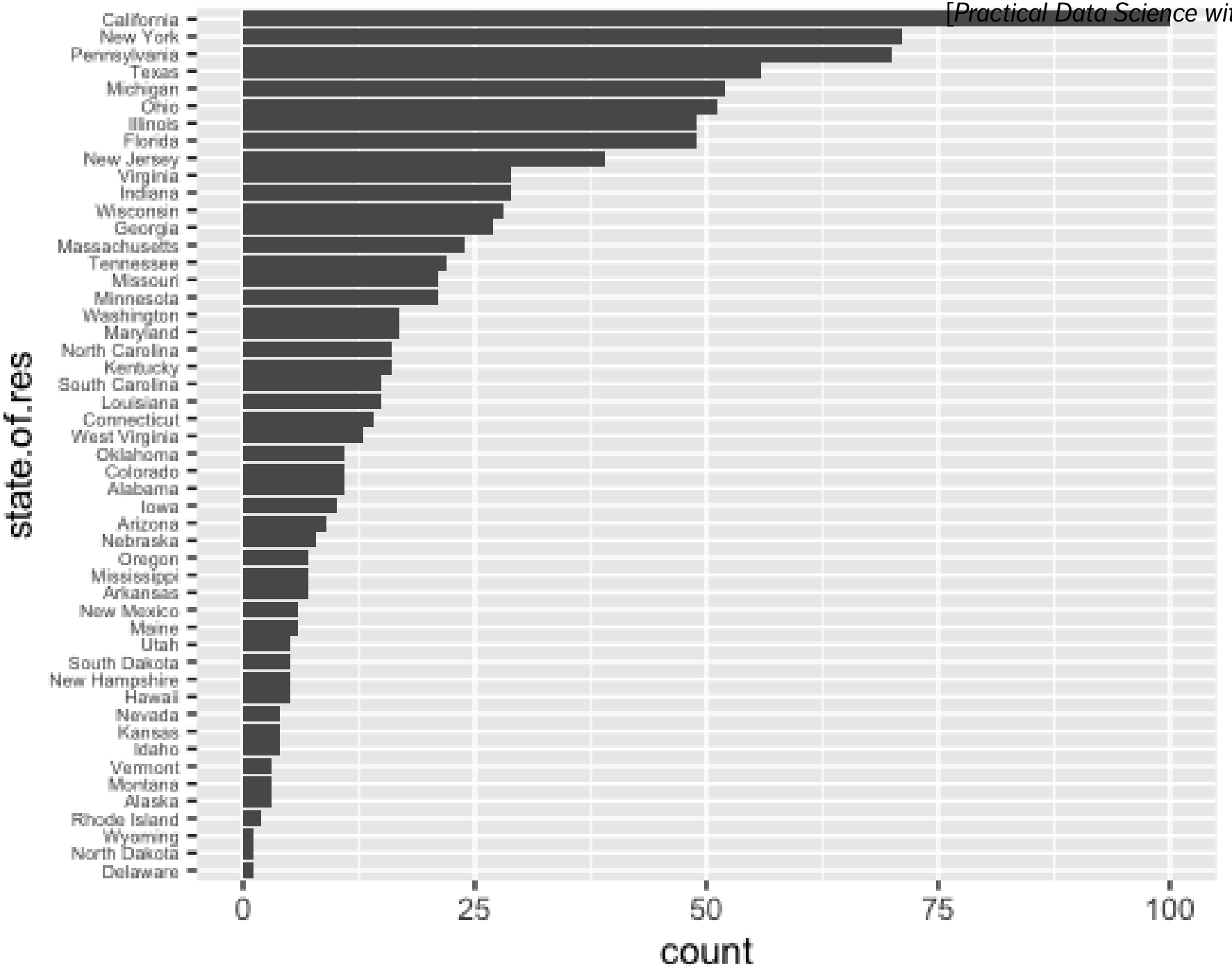
To order according to the number of observations in each state, we first need to modify the data using `transform()`, which will actually re-order the levels for **`state.of.res`** by population in the dataset.

If the dataset is representative, presumably that order will match the order of the states' populations.

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

```
> tbl <- as.data.frame(table(df$state.of.res))

> colnames(tbl) <- c("state.of.res", "count")

> tbl <- transform(tbl,
   state.of.res=reorder(state.of.res, count))

> ggplot(tbl) +
   geom_bar(aes(x=state.of.res, y=count),
   stat="identity") + coord_flip() +
   theme(axis.text.y=element_text(size=rel(0.6)))
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

When it comes to **univariate** representations:

- use a **histogram** or **density plot** to look for outliers, incorrect values, in numerical variables
- which will also give a sense of the distribution – is it symmetric, normal, log-normal, etc.
- use a **bar chart** to compare frequencies for categorical variables

USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

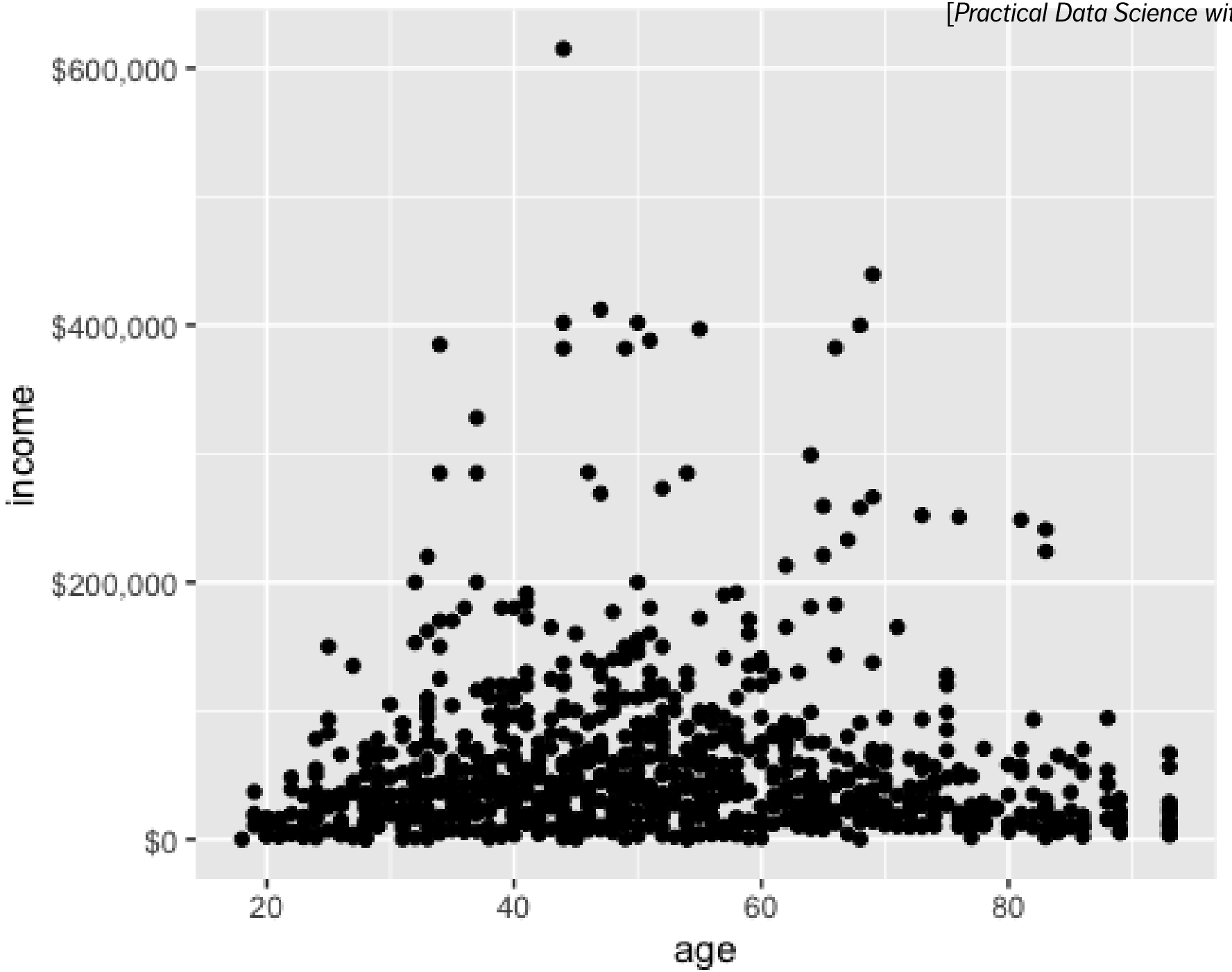
We can also look at bivariate charts, say involving **`age`** and **`income`**. Let's start by removing the weird observations.

```
> df.3 <- with(df, subset(df, age > 0 & age < 100 &
income > 0))
```

We can prepare a scatterplot:

```
> ggplot(df.3, aes(x=age, y=income)) +
   geom_point() +  # scatterplot geometry
   scale_y_continuous(labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

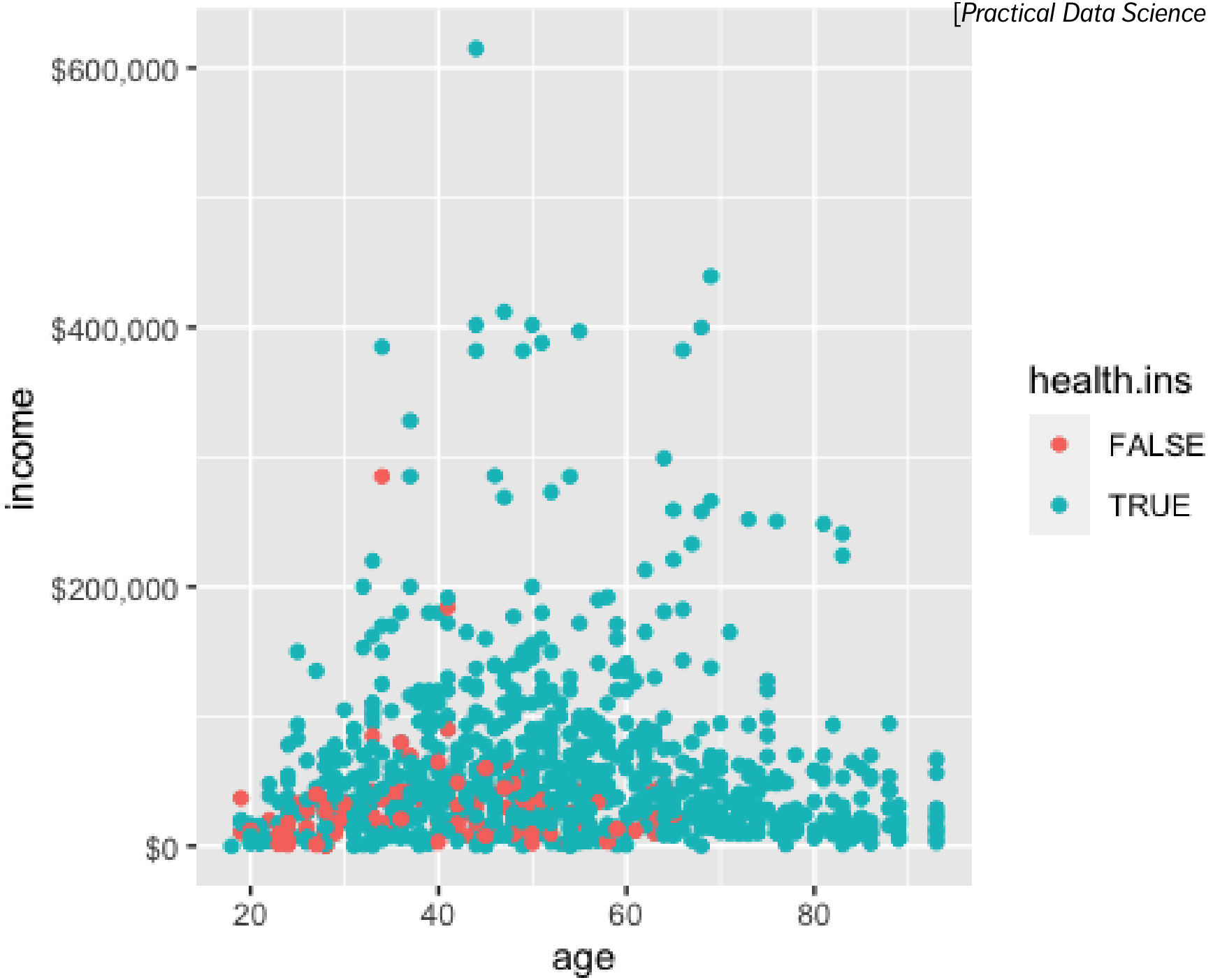


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

We can also colour the dots according to the **health insurance status:**

```
> ggplot(df, aes(x=age, y=income, colour=health.ins)) +
   geom_point() +
   scale_y_continuous(labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

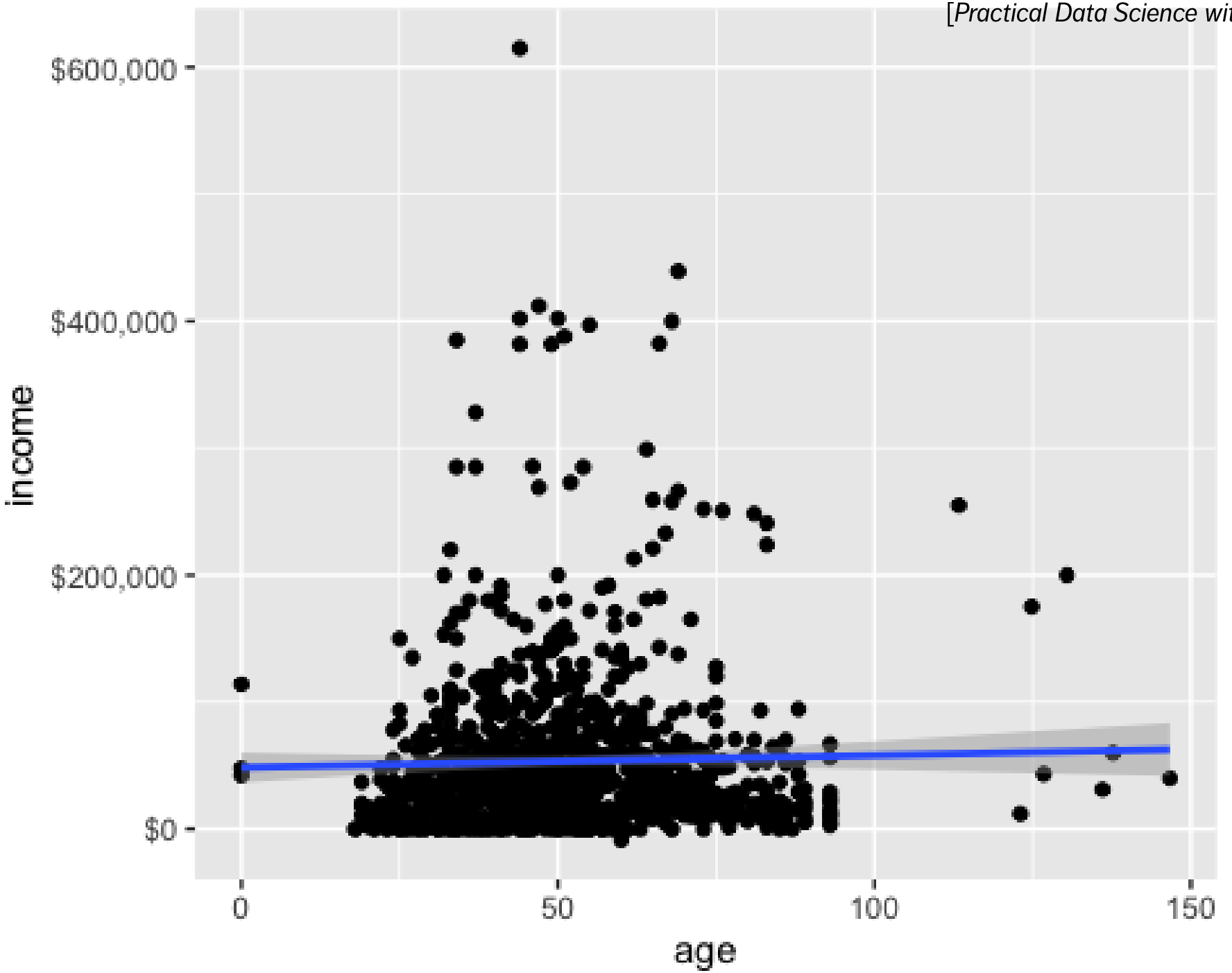


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

The relationship between **age** and **income** is not linear, so adding the line of best-fit might not provide much in the way of insight, but it can be done nonetheless.

```
> ggplot(df, aes(x=age, y=income)) +
   geom_point() +
   geom_smooth(method="lm") +
   scale_y_continuous(labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

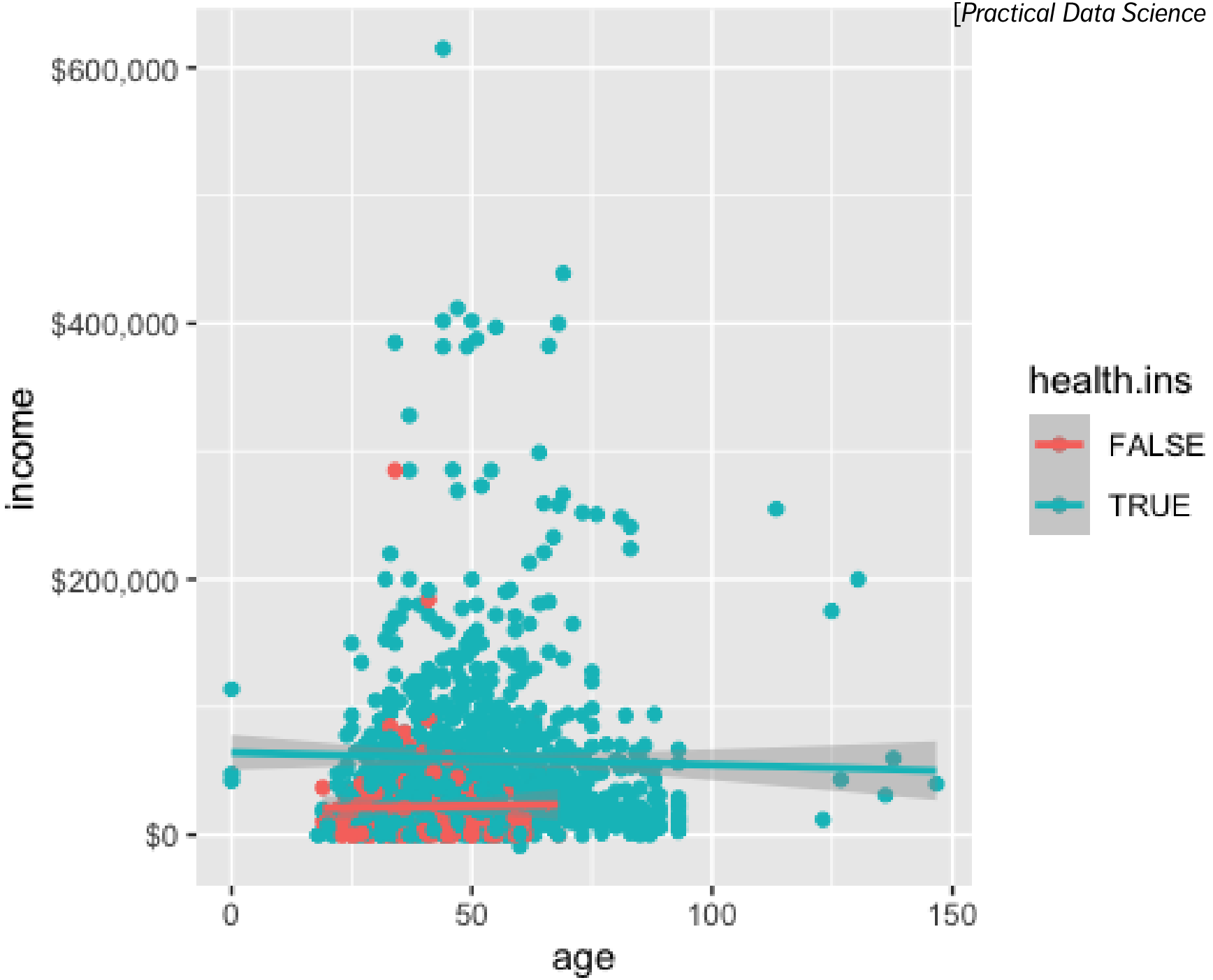


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

We can also plot the lines of best fit for various categories:

```
> ggplot(df, aes(x=age, y=income, colour=health.ins)) +
   geom_point() +
   geom_smooth(method="lm") +
   scale_y_continuous(labels=dollar)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

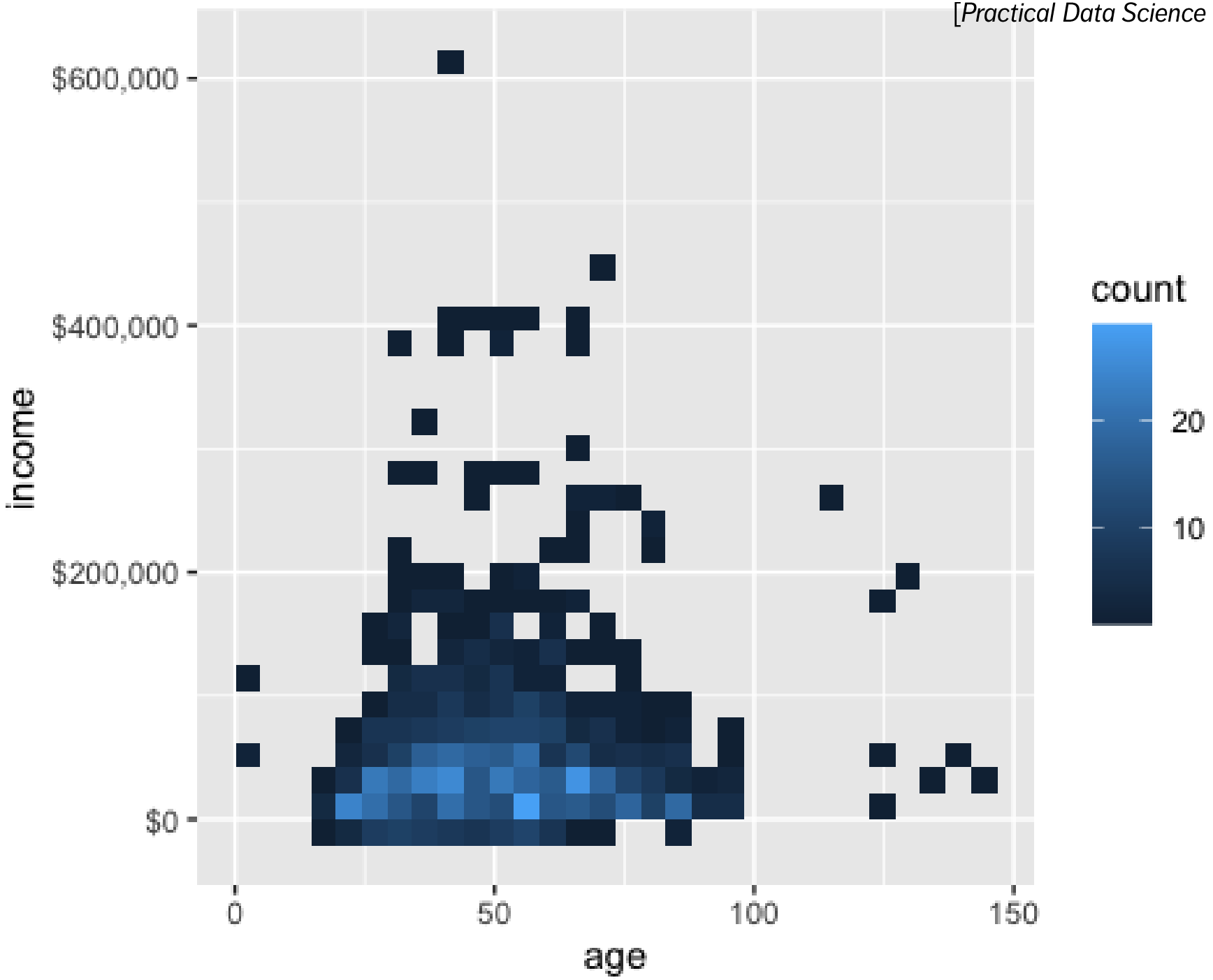


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

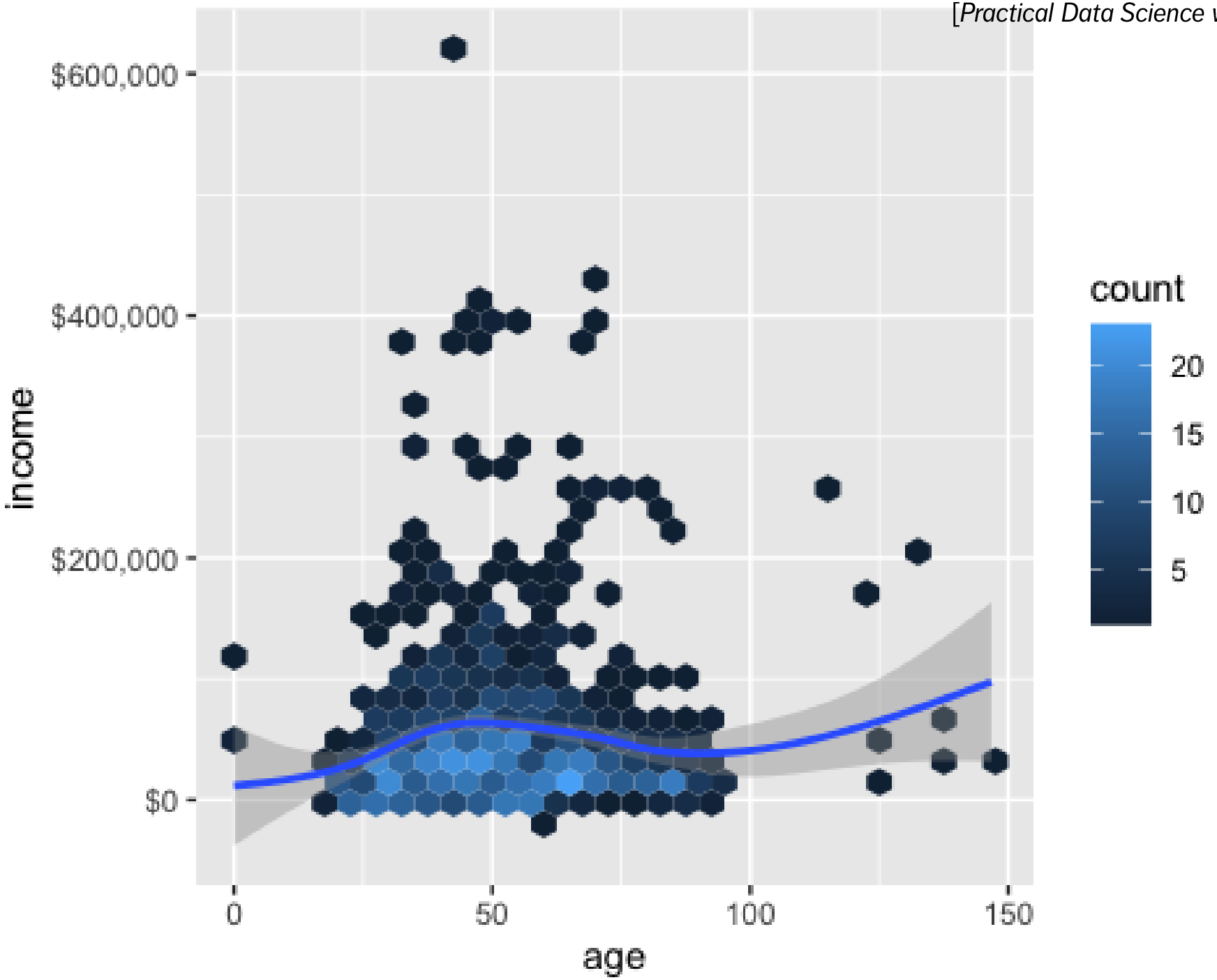
A heat map (colour represents # of observations in the cell) might be more *à propos*.

```
> ggplot(df, aes(x=age, y=income)) +
   geom_bin2d() +
   scale_y_continuous(labels=dollar)

> library(hexbin)

> ggplot(df, aes(x=age, y=income)) +
   geom_hex(binwidth=c(5, 20000)) +
   scale_y_continuous(labels=dollar) +
   geom_smooth() # smooth loess curve of best fit
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

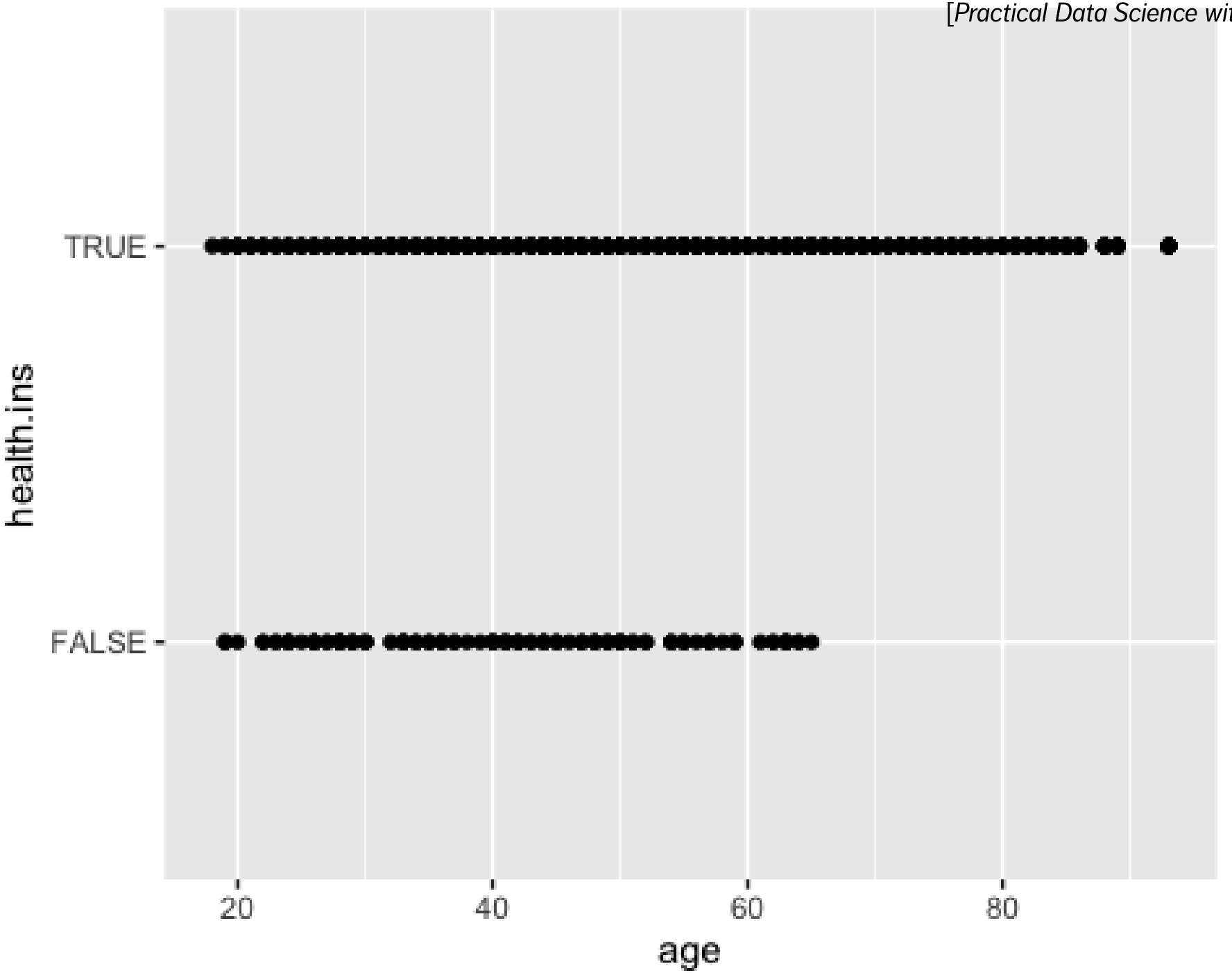
How about a plot of the relationship between **`age`** and **`health.ins`**?

```
> ggplot(df.3, aes(x=age, y=health.ins)) +
   geom_point()
```

Is it surprising that the group of insured customers tends to have older members?

It might seem as though there are roughly as many more people with insurance than people without, but that's an illusion: all the observations that have the same age are represented by a single dot.

[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

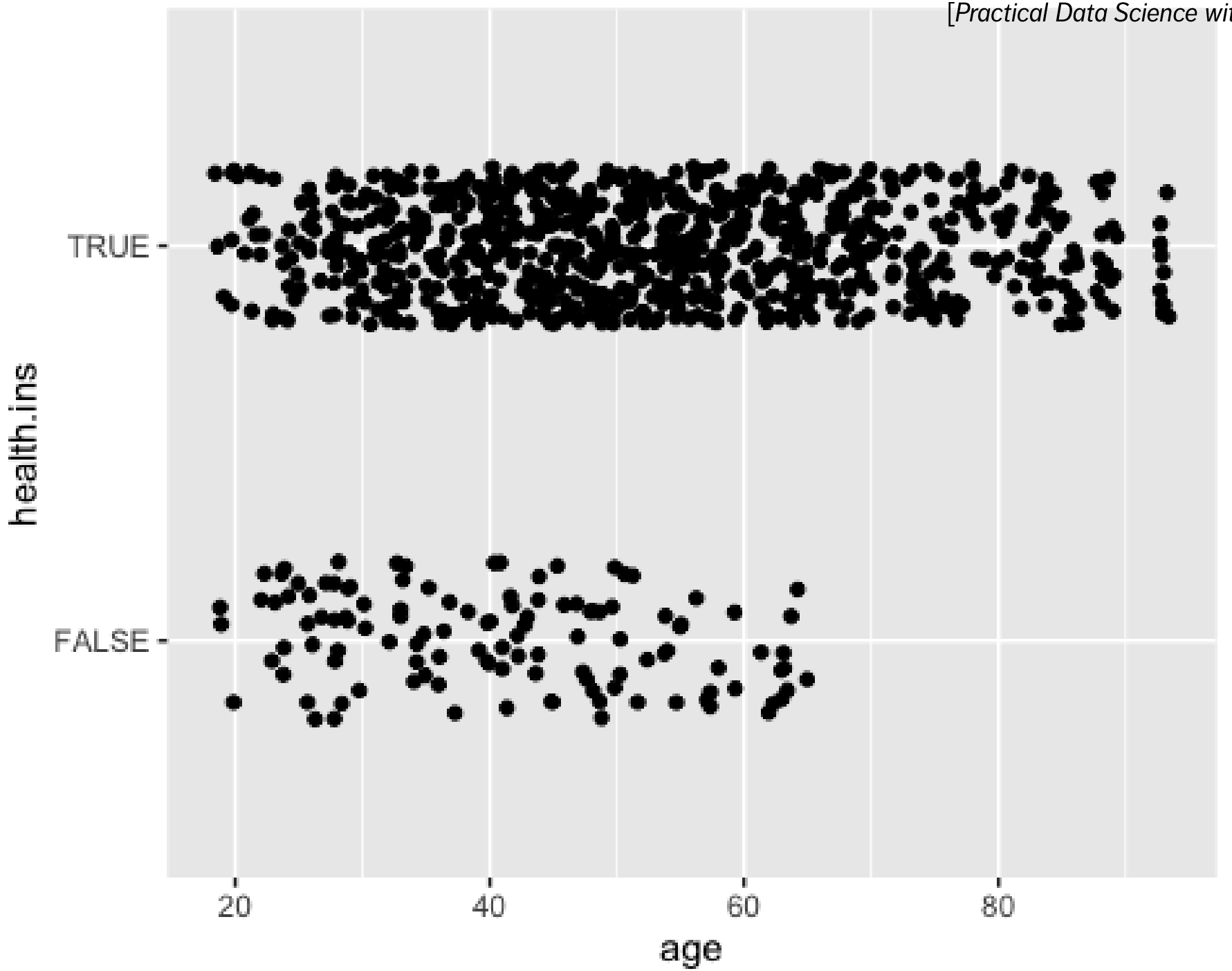
`geom_jitter` can come in handy

```
> ggplot(df.3, aes(x=age, y=health.ins)) +
   geom_jitter(height=0.2)
```

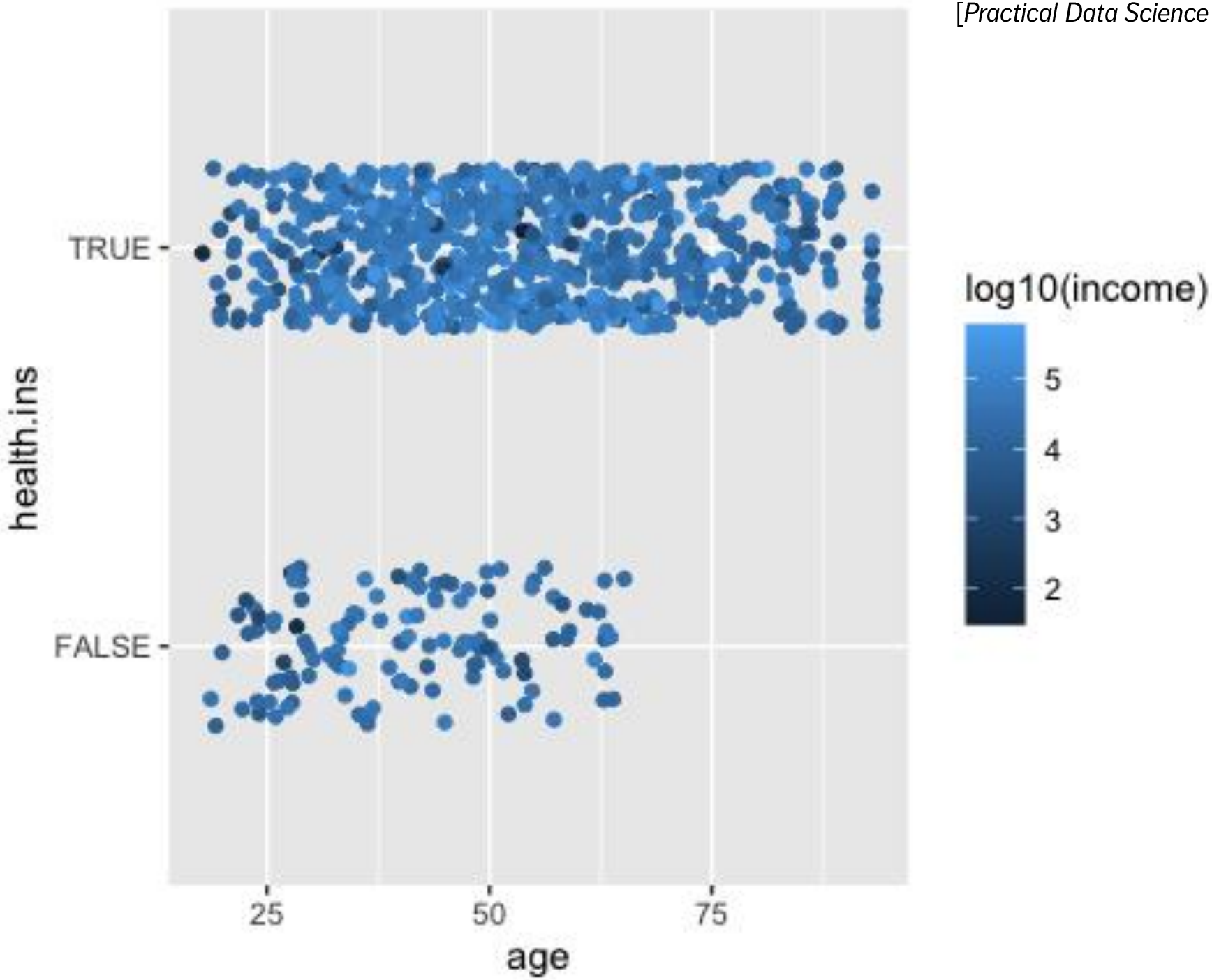
Why stop at only 2 variables when we could add **`income`** to the picture?

```
> ggplot(df.3, aes(x=age, y=health.ins,
   colour=log10(income))) +
   geom_jitter(height=0.2)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

We could also try to link marital status to health insurance status?

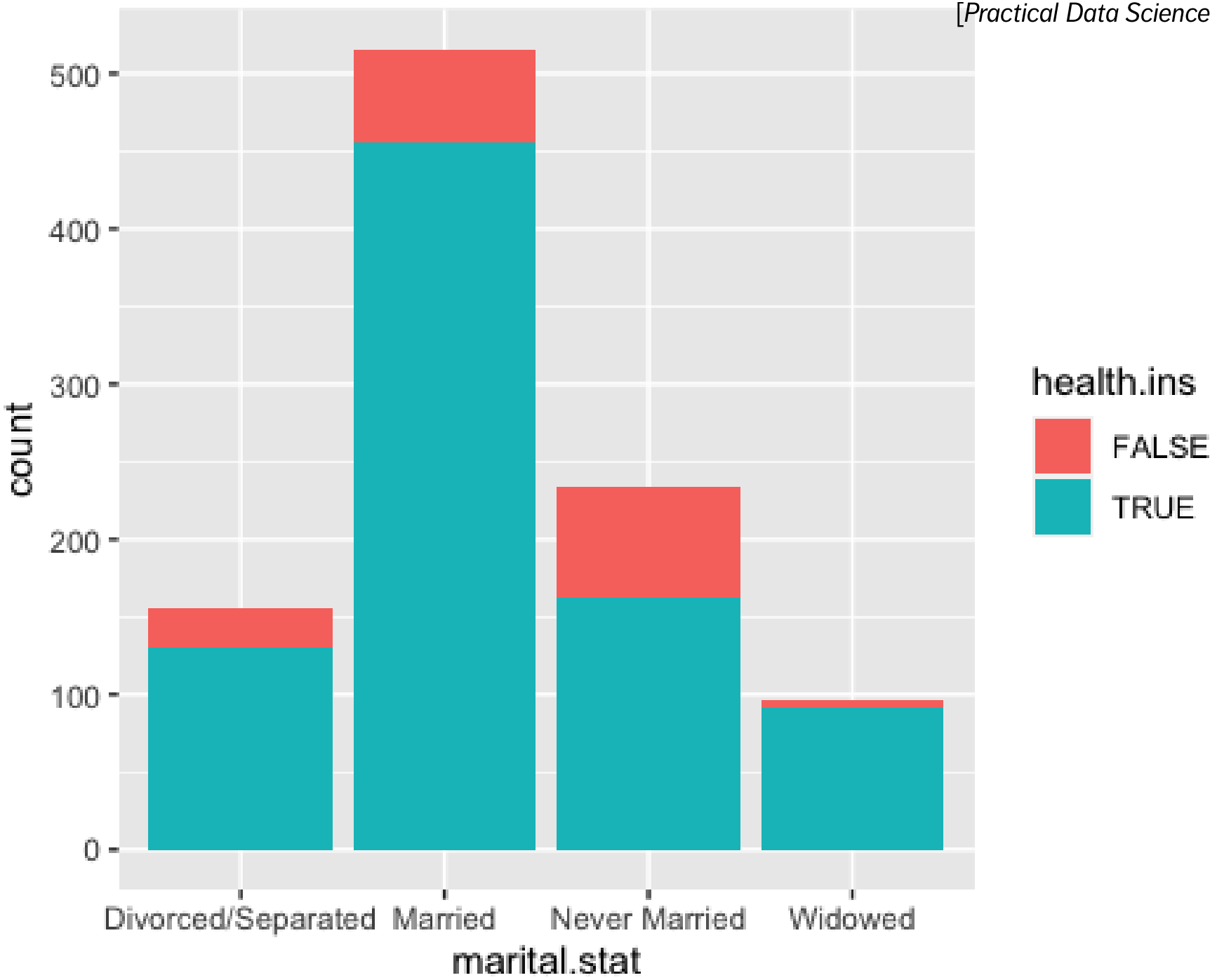
```
> ggplot(df) +
   geom_bar(aes(x=marital.stat, fill=health.ins))
```

Stacked bar charts are the pie charts of bar charts – trying dodging instead:

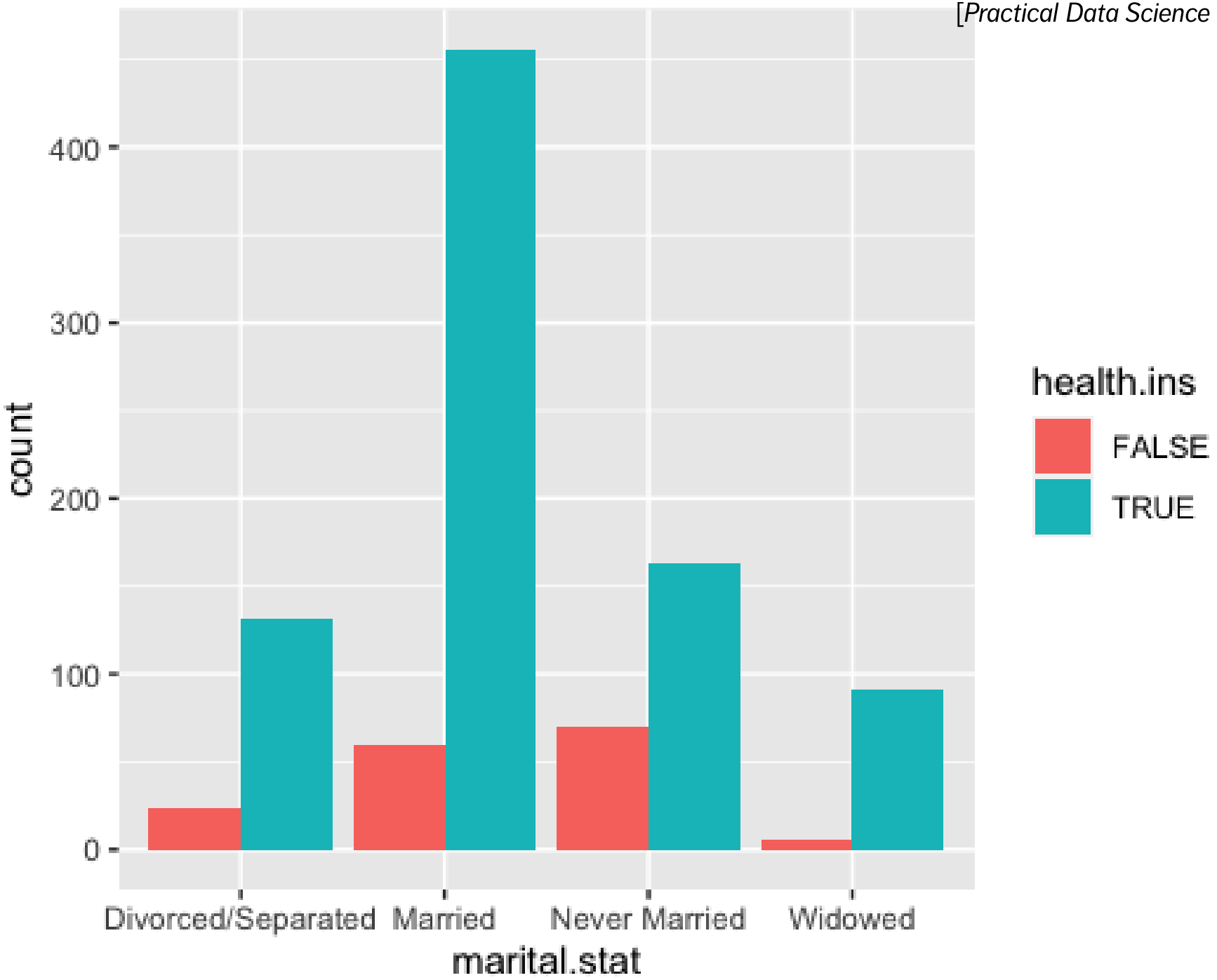
```
> ggplot(df) +
   geom_bar(aes(x=marital.stat, fill=health.ins),
   position="dodge")
```

Is there anything insightful in there?

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]



USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

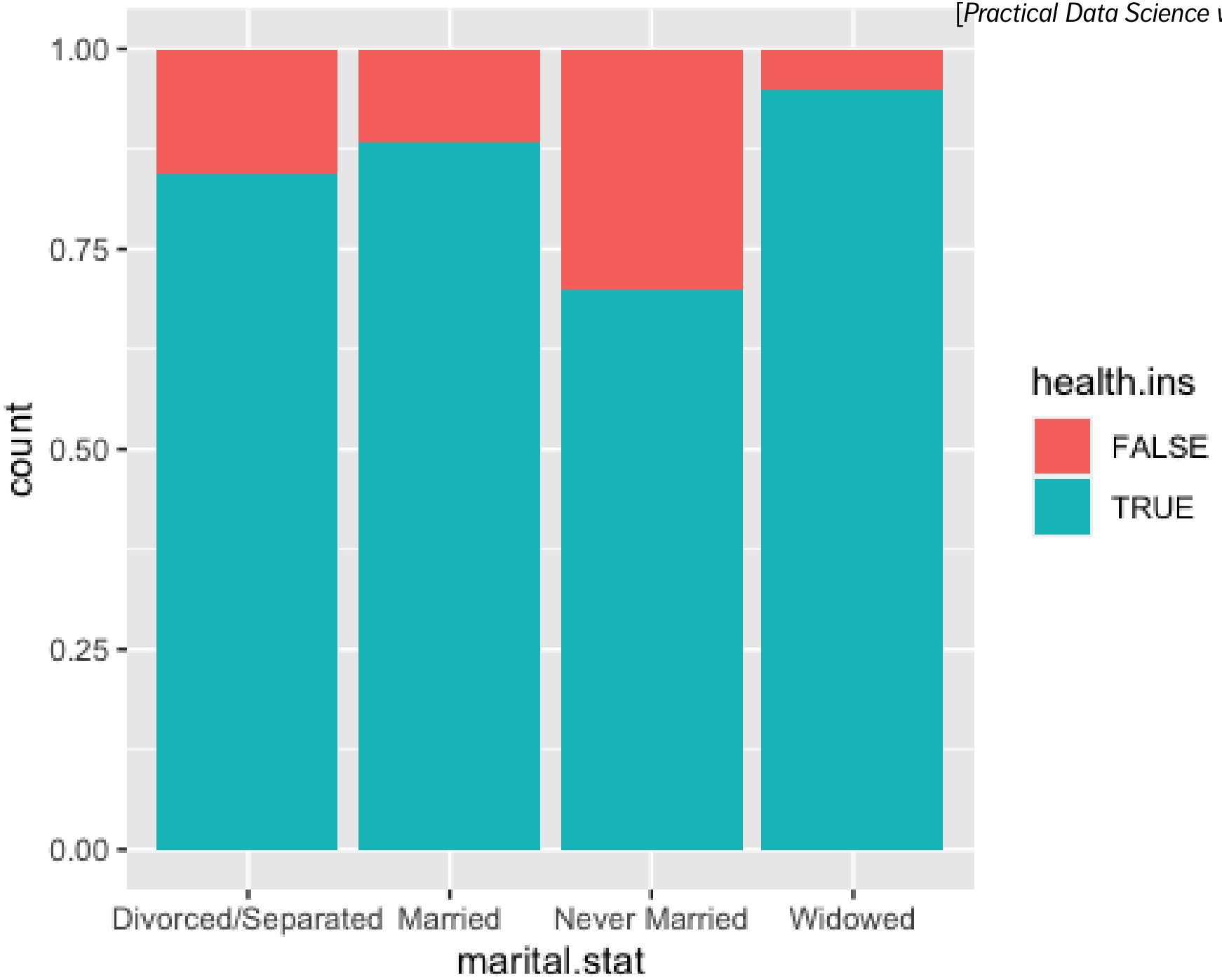
We could probably make an exception for **proportion stacked bar charts**:

```
> ggplot(df, aes(x=marital.stat)) +
    geom_bar(aes(fill=health.ins), position="fill")
```

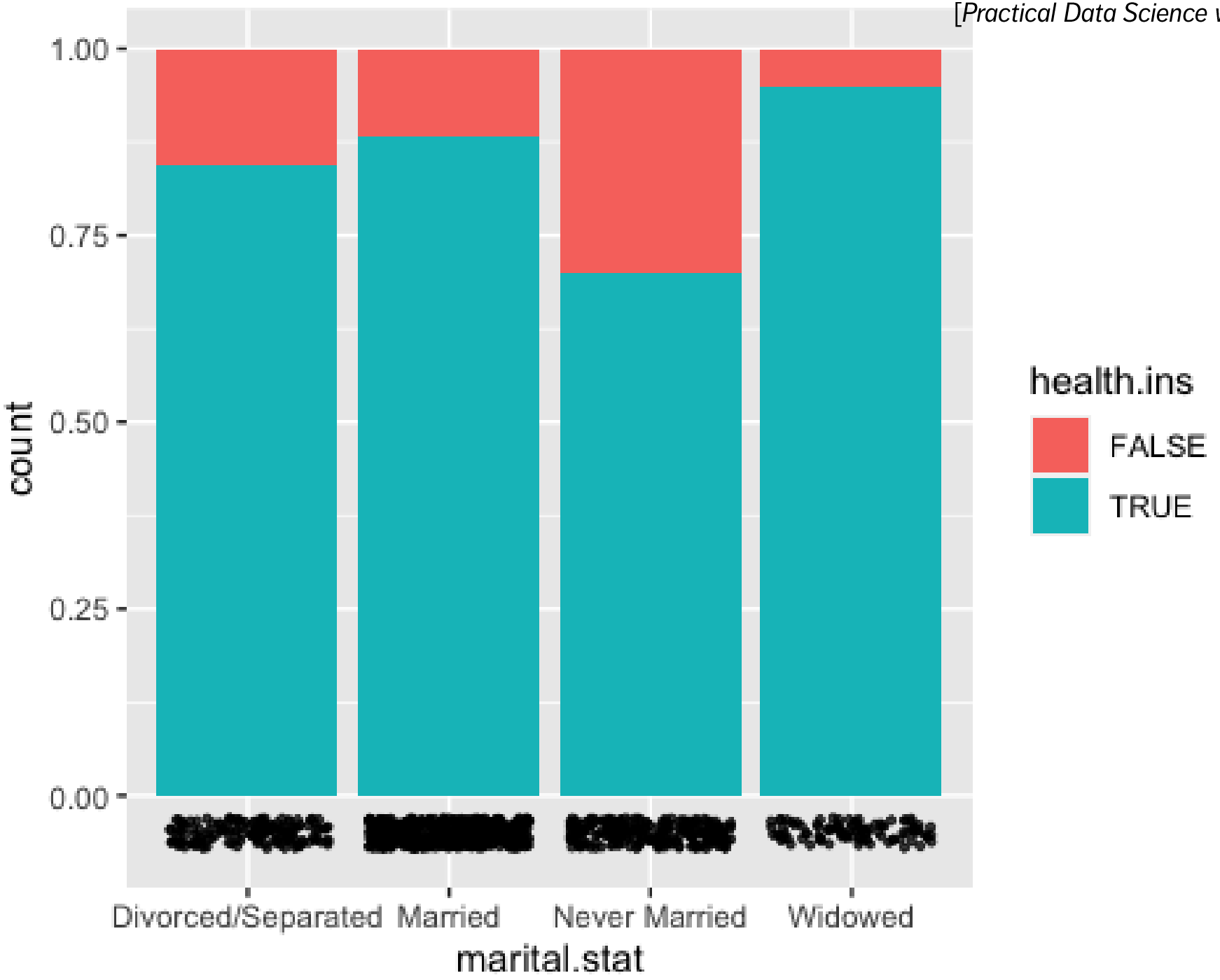
But we do lose the sense of the size of each sub-categories' population:

```
> last_plot() + # useful to modify the previous plot
    geom_point(aes(x=marital.stat, y=-0.05),
    position=position_jitter(h=0.02),
    size=0.75, alpha=0.75)
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]

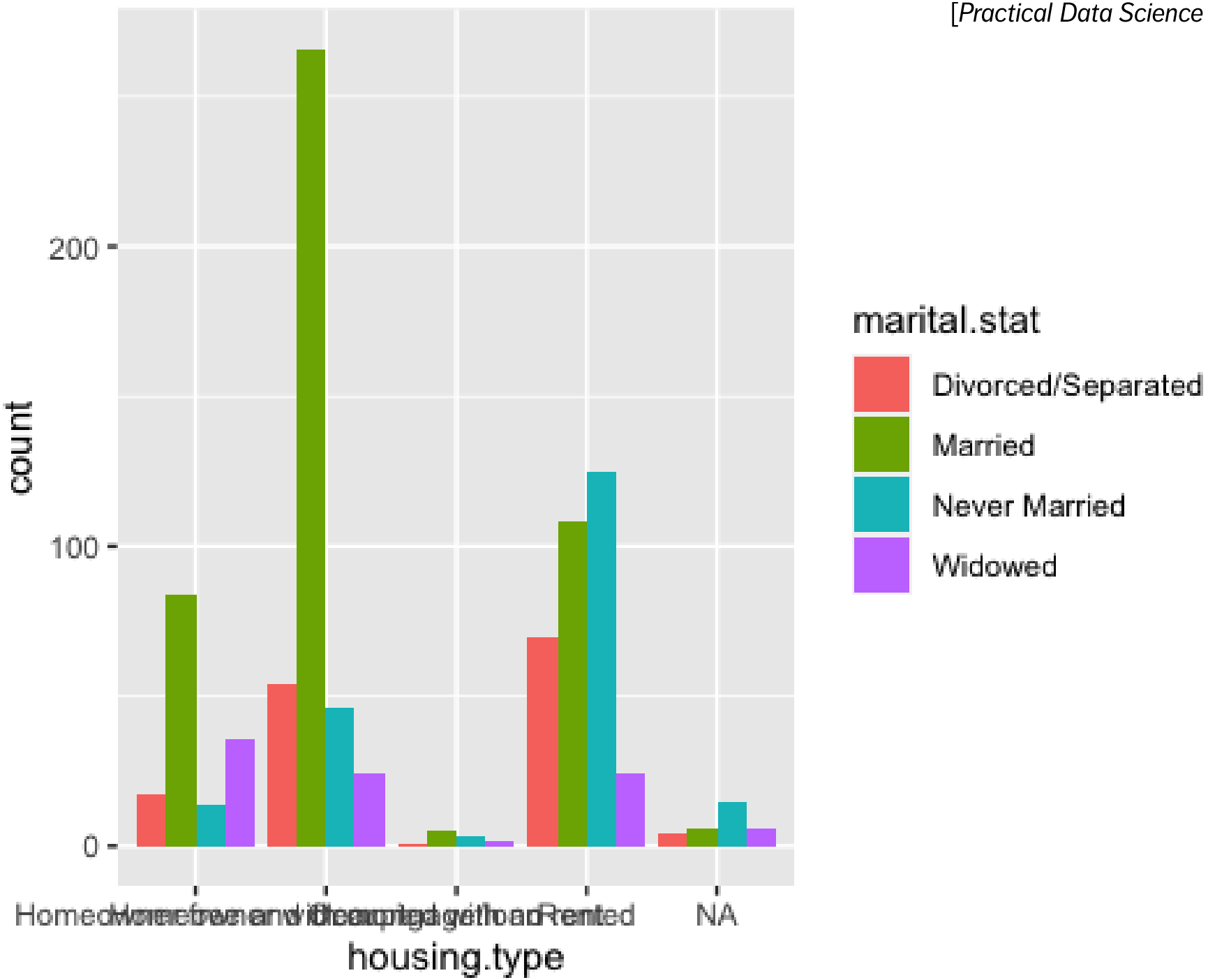


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

Might there be a link between housing type and marital status?

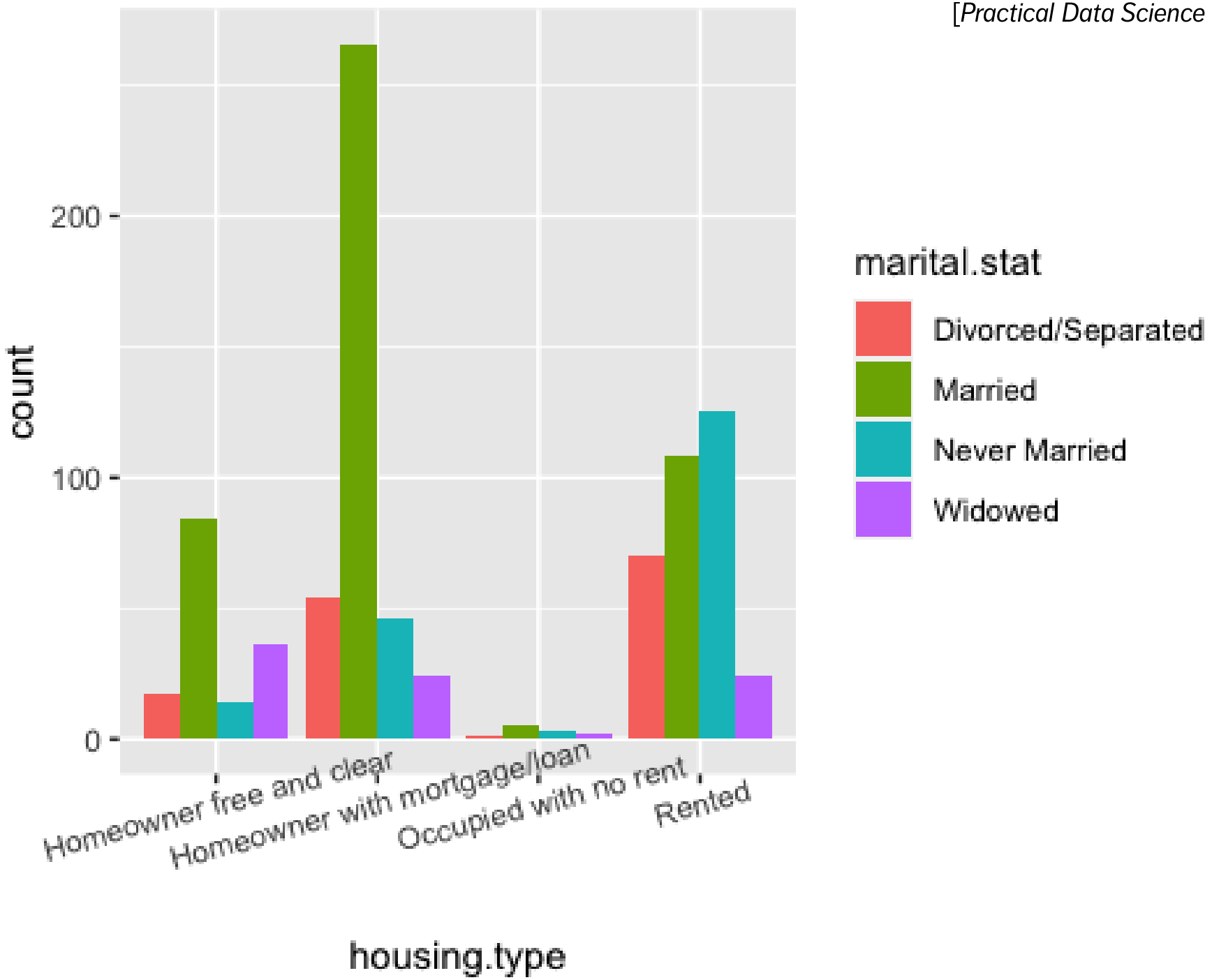
```
> ggplot(df.3) + geom_bar(aes(housing.type,
   fill=marital.stat),position="dodge")

> ggplot(df.3) + geom_bar(aes(housing.type,
   fill=marital.stat), position="dodge") +
   coord_flip()

> ggplot(subset(df.3, !is.na(housing.type))) +
   geom_bar(aes(housing.type, fill=marital.stat),
   position="dodge") +
   theme(axis.text.x=element_text(angle=15))
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



[*Practical Data Science with R*, N. Zumel, J. Mount]

[*Practical Data Science with R*, N. Zumel, J. Mount]



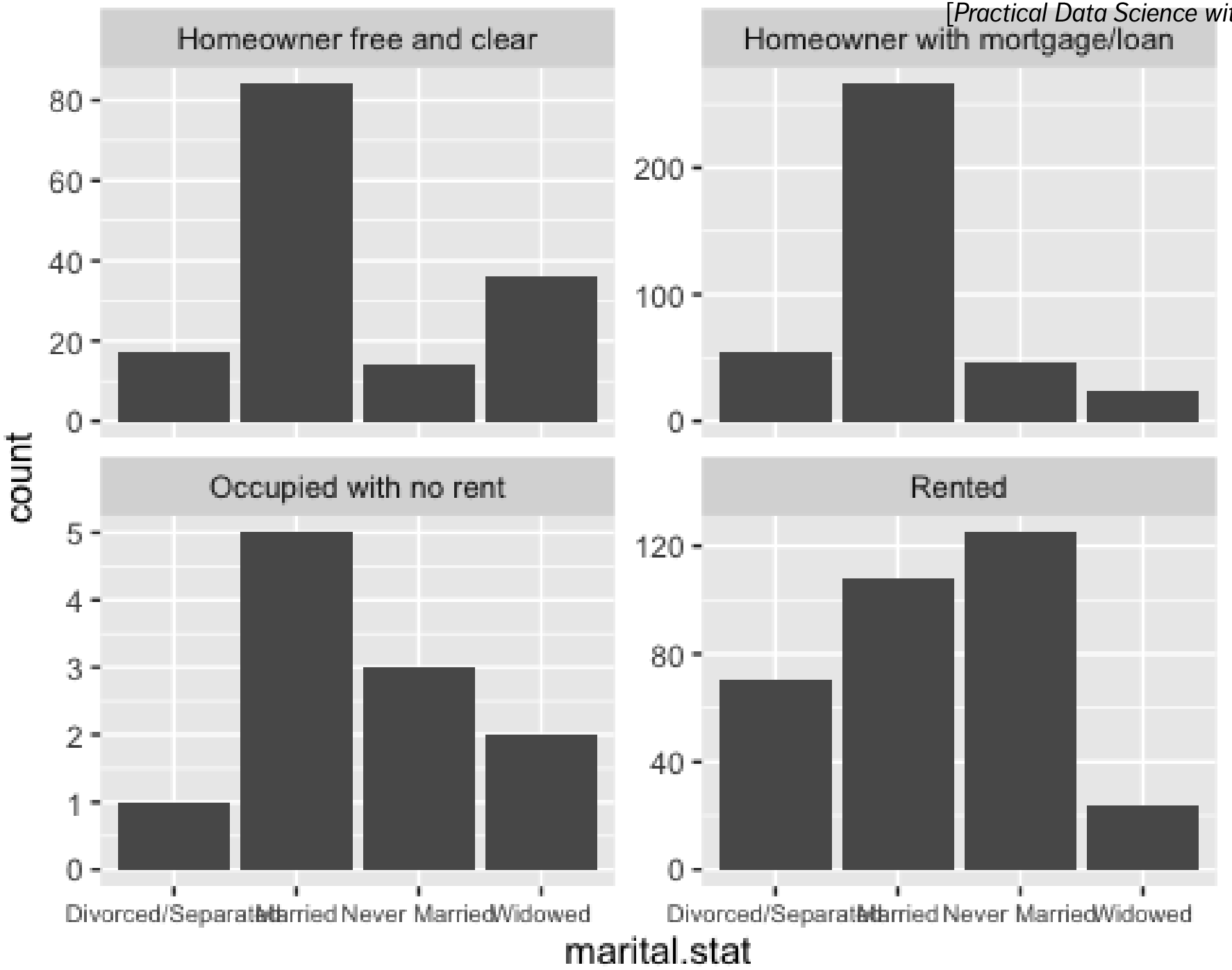
USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

It's easy to see how some fine-tuning can make the charts easier to read (which can only help when it comes to extracting actionable insights).

We wrap up our exploration of `custdata.tsv` by building a **small multiple** chart:

```
> ggplot(subset(df.3, !is.na(housing.type))) +
   geom_bar(aes(marital.stat)) +
   facet_wrap(~housing.type, scales="free_y") +
   theme(axis.text.x=element_text(size=rel(0.8)))
```

[*Practical Data Science with R*, N. Zumel, J. Mount]

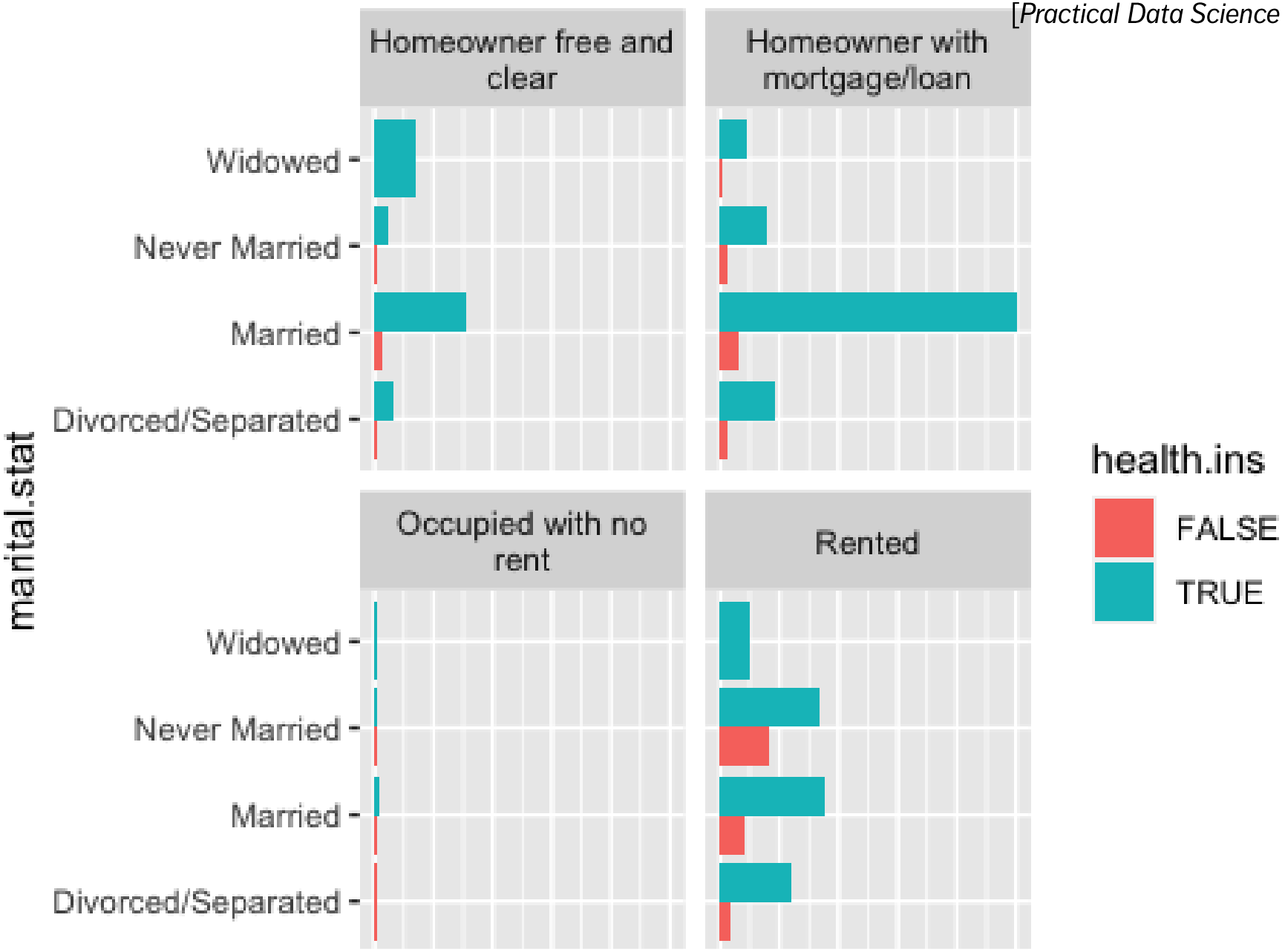


USING GGPLOT2 IN PRACTICE – US CENSUS PUMS DATA

Could there be a link with **health insurance status**?

```
> ggplot(subset(df.3, !is.na(housing.type))) +
   geom_bar(aes(marital.stat, fill=health.ins),
   position="dodge") +
   facet_wrap(~housing.type, ncol=2,
   labeller=label_wrap_gen(width=20, multi_line=TRUE)) +
   theme(axis.text.x=element_text(size=rel(0.8))) +
   coord_flip()
```

[*Practical Data Science with R*, N. Zumel, J. Mount]



EXERCISE

In teams or individually, select a dataset (preferably a "real-world" dataset), and a few questions related to the dataset.

Using `ggplot2`, produce 2-3 "definitive" displays of the dataset (as might be used by the *New York Times* or the *Economist*, say).

Hint: you should first explore the dataset and produce a number of univariate and bivariate charts (using specific and randomly selected variables, and specific and randomly selected chart types).

Once you have a better sense of the data, move on to the definitive charts. You should be prepared to discard many false starts along the way.