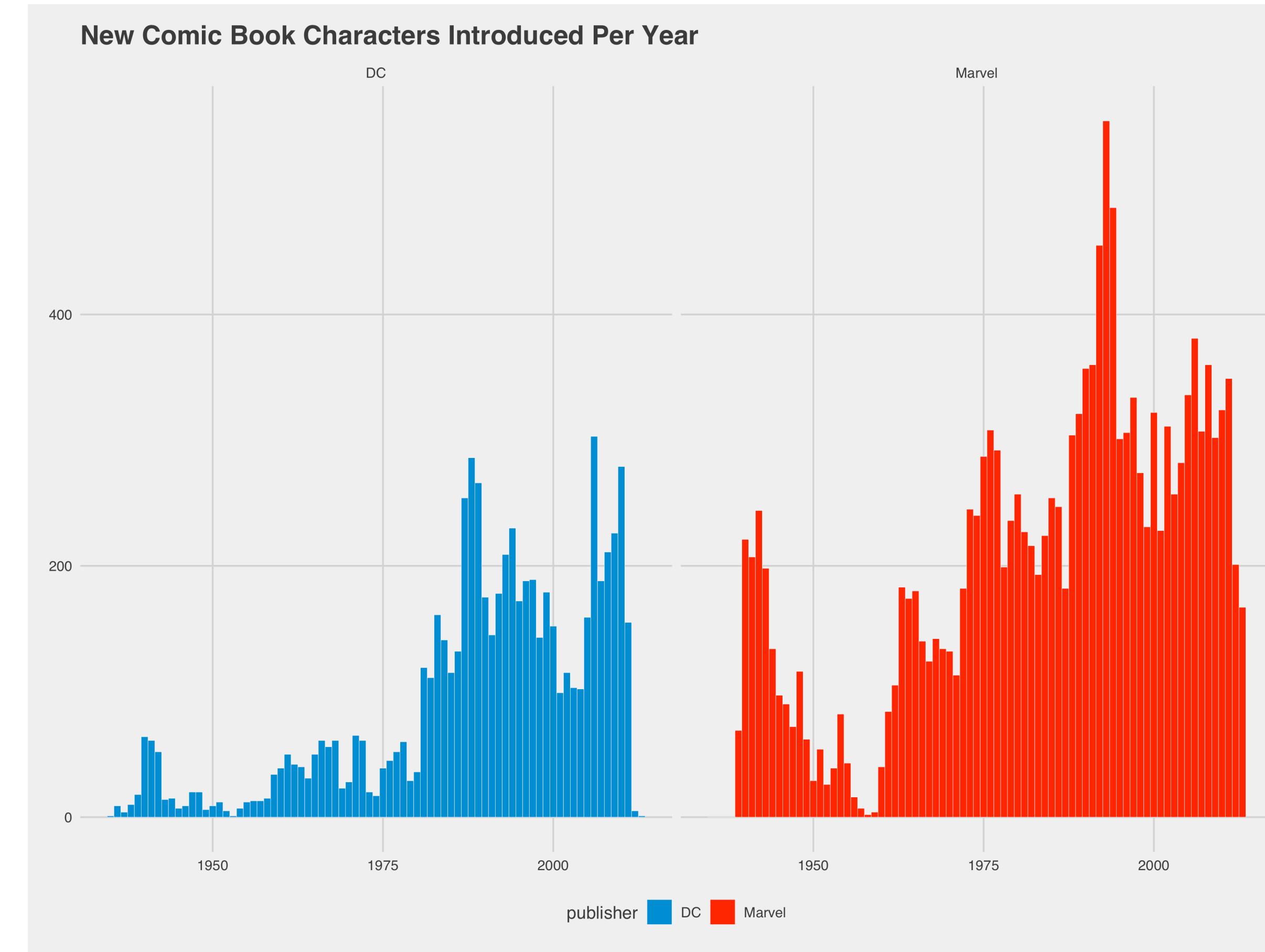


more ggplot2

```
p1 <- ggplot(comic_characters) +  
  geom_histogram(aes(x = year, fill = publisher), binwidth = 1, color =  
"white", lwd = 0.1) +  
  facet_wrap(~publisher) +  
  theme_fivethirtyeight() +  
  scale_fill_manual(values = c("#008fd5", "#ff2700")) +  
  labs(title = "New Comic Book Characters Introduced Per Year")
```

p1



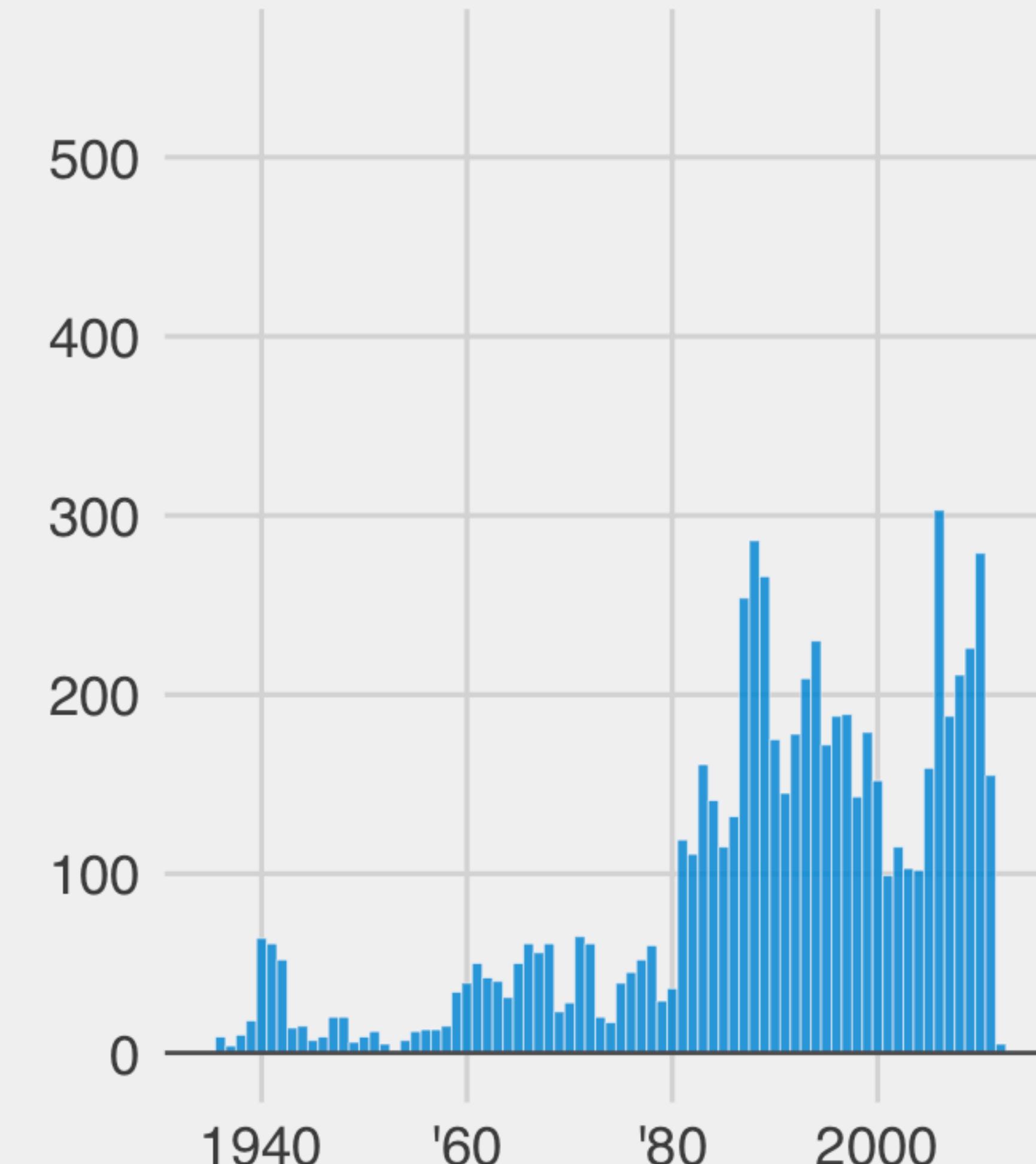
```
publisher_labels <- c(DC = "DC, New Earth continuity",
                      Marvel = "Marvel, Earth-616 continuity")
p1 <- ggplot(comic_characters) +
  geom_histogram(aes(x = year, fill = publisher), binwidth = 1,
                 color = "white", lwd = 0.1,
                 show.legend = FALSE, alpha = 0.9) +
  facet_wrap(~publisher, labeller = labeller(publisher = publisher_labels)) +
  scale_x_continuous(breaks = seq(1940, 2000, 20),
                     labels = c("1940", "'60", "'80", "2000")) +
  scale_y_continuous(limits = c(0, 555), breaks = seq(0, 500, 100)) +
  scale_fill_manual(values = c("#008fd5", "#ff2700"))+
  geom_hline(yintercept = 0, color = "grey31", size = 0.5) +
  theme_fivethirtyeight() +
  theme(axis.text.y = element_text(size = 13),
        axis.text.x = element_text(size = 13),
        plot.title = element_text(size = 26, hjust = 0.5),
        strip.text.x = element_text(size = 18, hjust = 0, face ="bold"),
        panel.spacing = unit(2, "lines")) +
  labs(title = "New Comic Book Characters Introduced Per Year")
```

p1

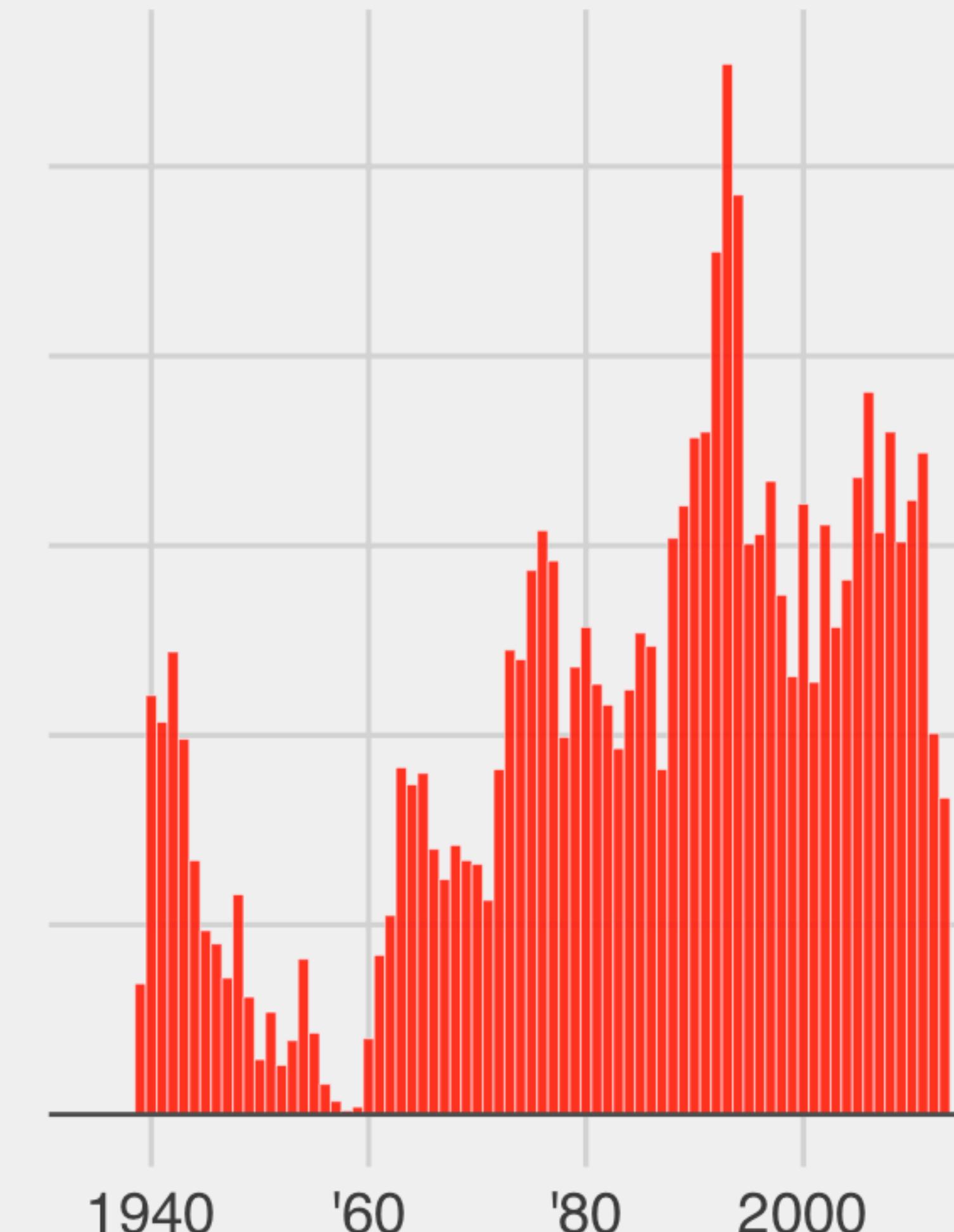
From RMarkdown

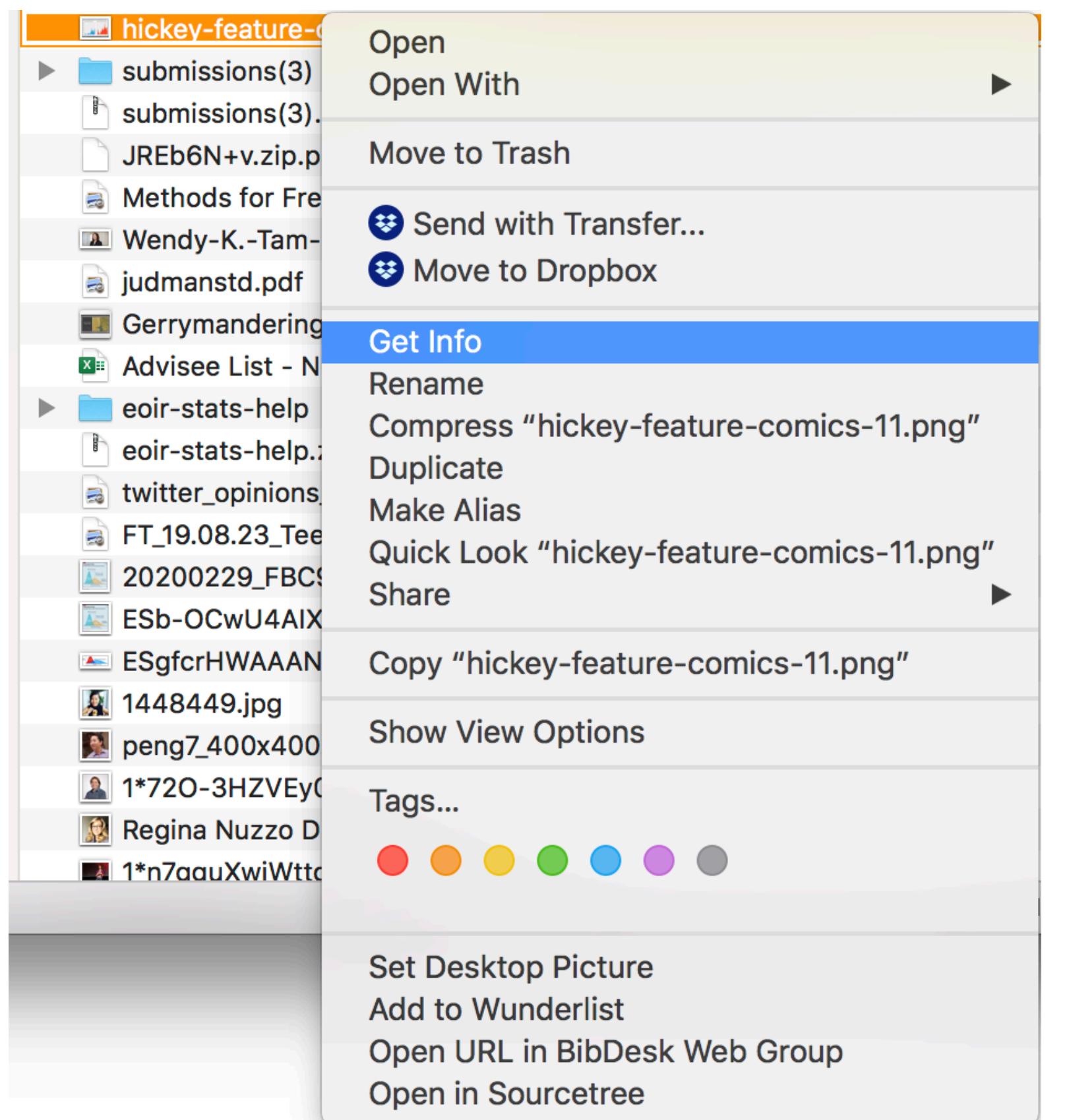
N Comic Book Characters Introduced Per Year

DC, New Earth continuu



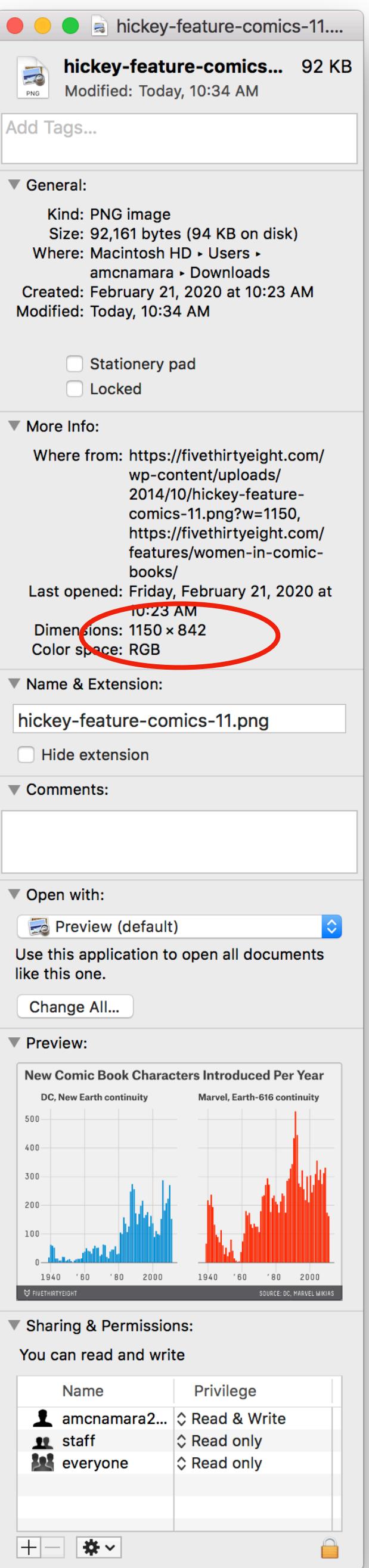
Marvel, Earth-616 continuu





$$1150/842 = 1.365796$$

$$12/9 = 1.33333$$



Saving plots

ggsave() saves the last plot.

Uses size on screen:

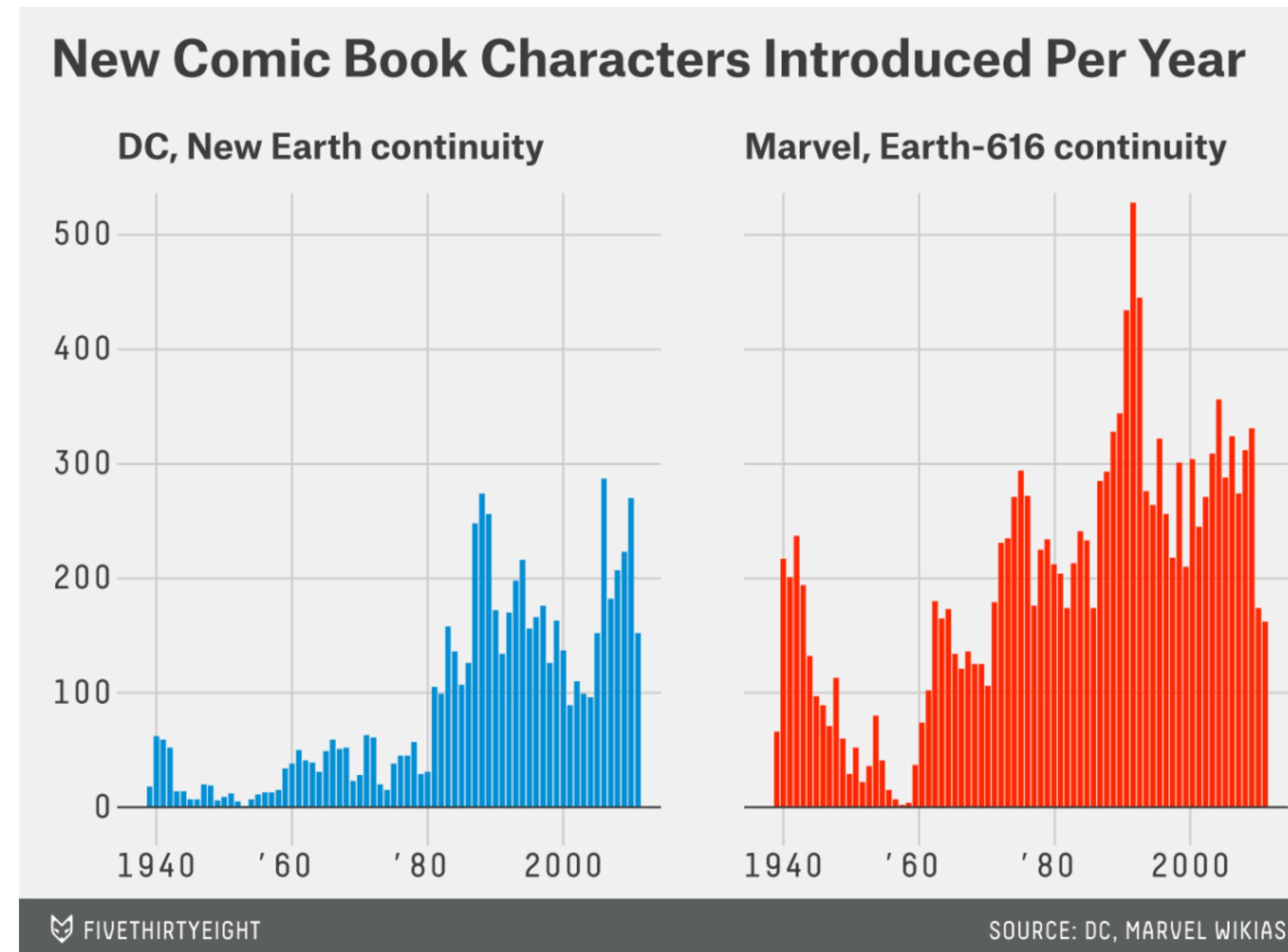
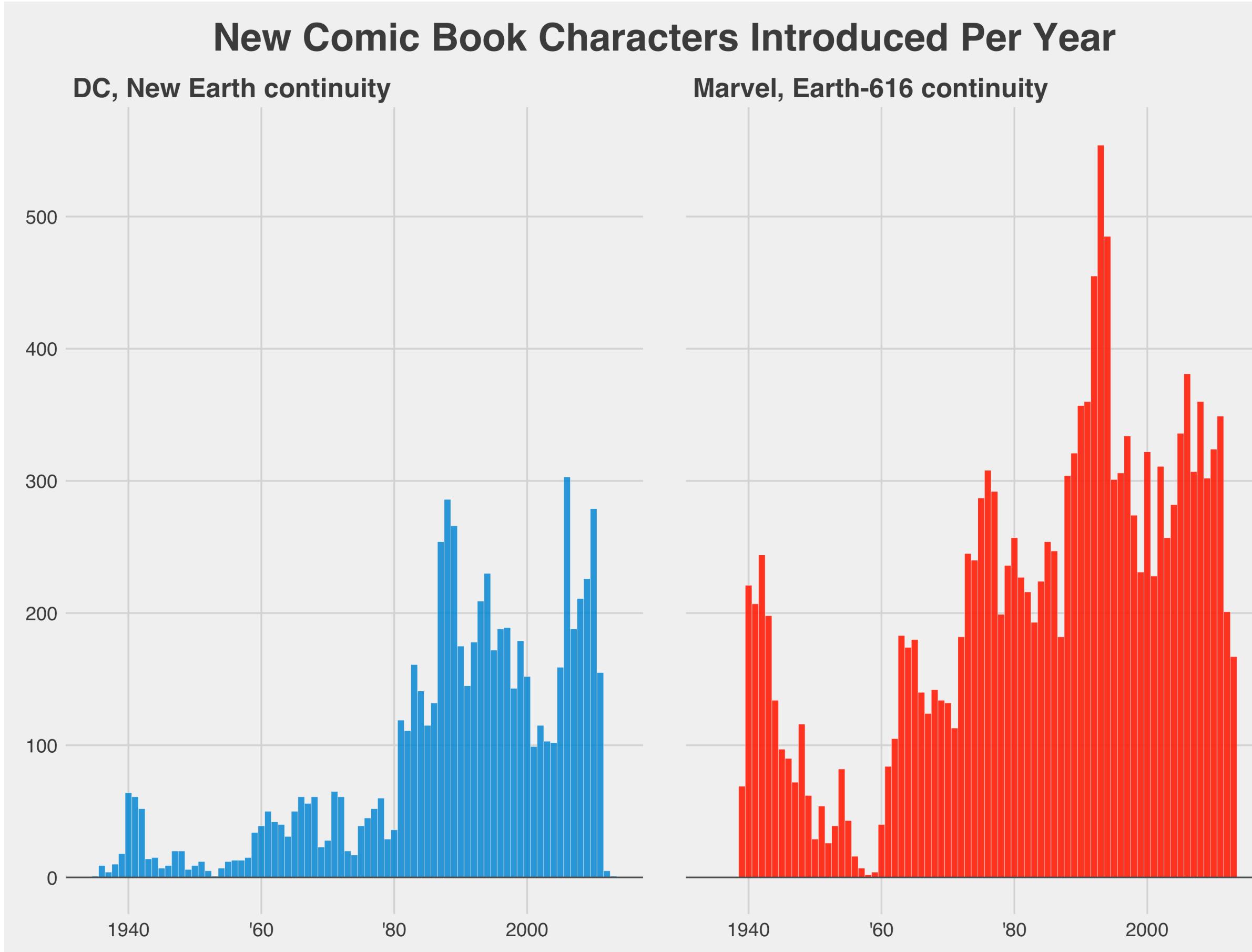
```
ggsave("my-plot.pdf")  
ggsave("my-plot.png")
```

Specify size in inches

```
ggsave("my-plot.pdf", width = 6, height = 6)
```

```
ggsave(p1, filename = "comic_final.png", width = 12, height=9)
```

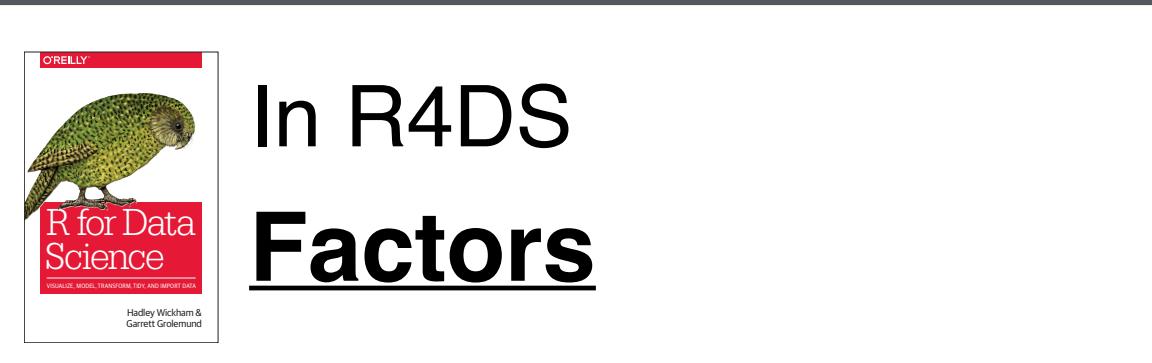
Better, but still not quite!



FIVETHIRTYEIGHT

SOURCE: DC, MARVEL WIKIAS

Factors



factors

R's representation of categorical data. Consists of:

1. A set of **values**
2. An ordered set of **possible levels**

```
eyes <- factor(x = c("blue", "green", "green"),
                 levels = c("blue", "brown", "green"))

eyes
## [1] blue  green green
## Levels: blue brown green
```



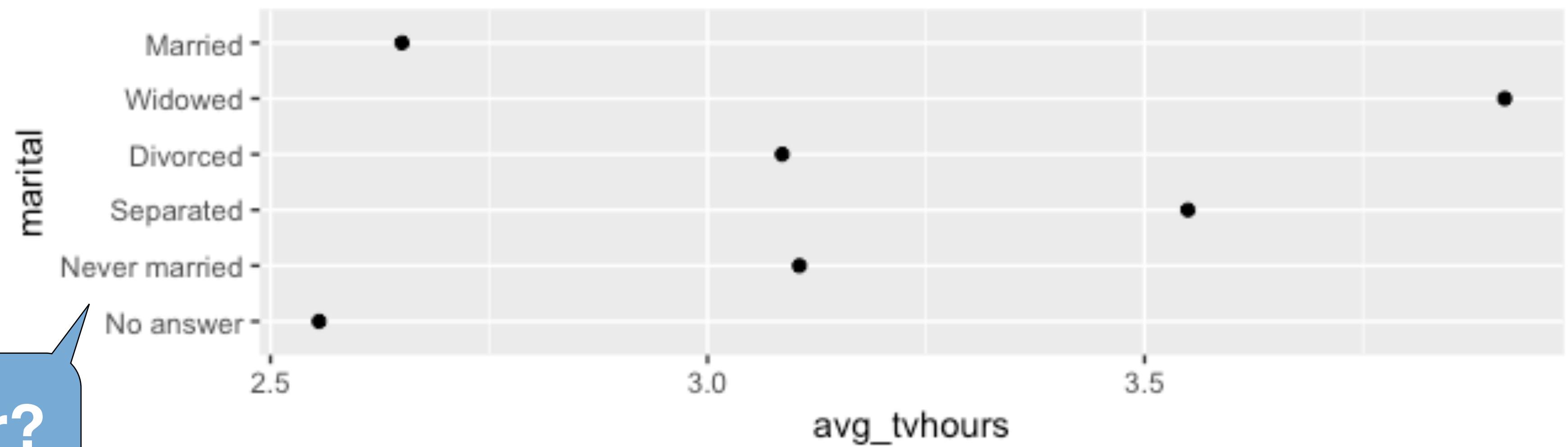
forcats



Simple functions for working with factors.

```
library(forcats)
```

```
gss_cat %>%  
  drop_na(tvhours) %>%  
  group_by(marital) %>%  
  summarise(avg_tvhours = mean(tvhours)) %>%  
  ggplot() +  
  geom_point(aes(avg_tvhours, marital))
```

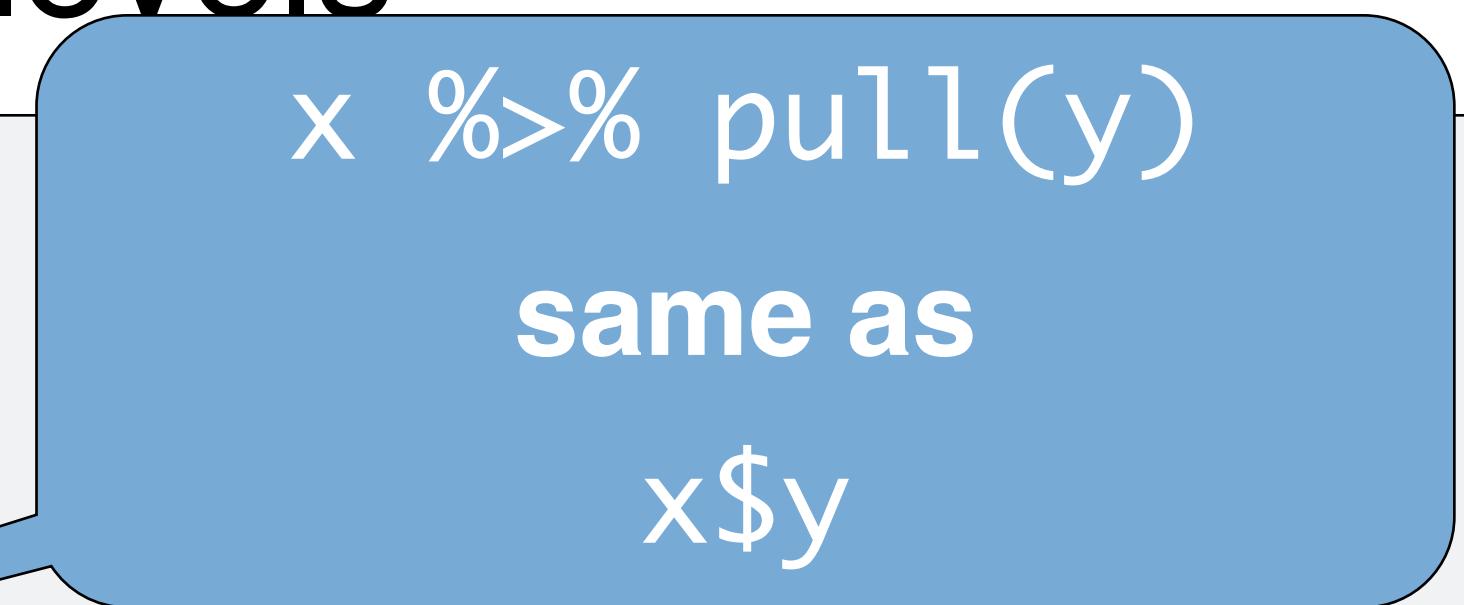


Why this order?

levels()

Use **levels()** to access a factor's levels

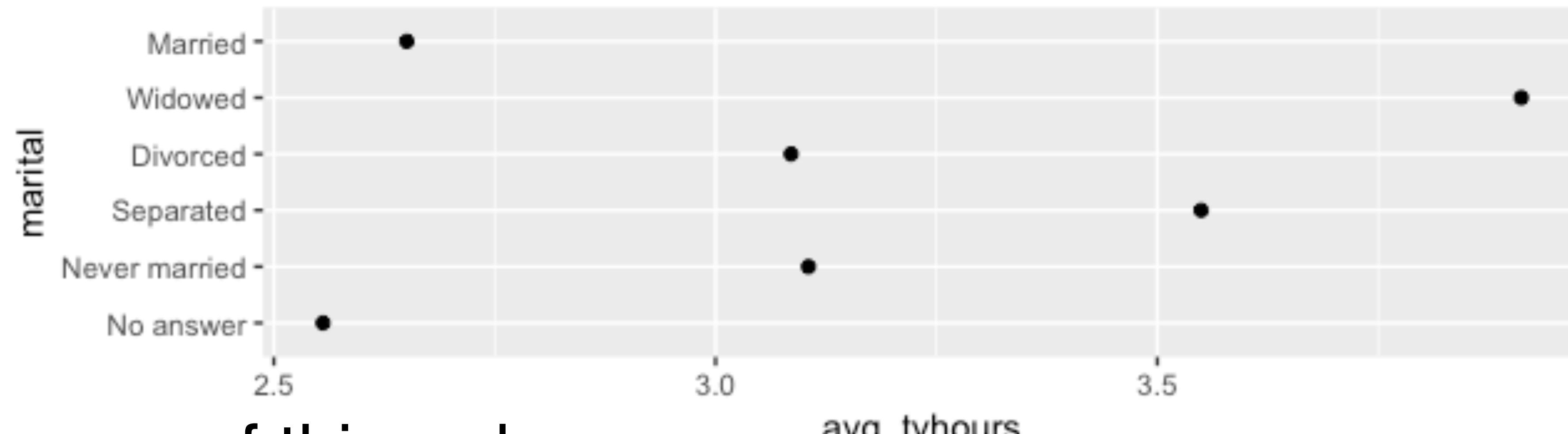
```
levels(eyes)
# [1] "blue"   "brown"  "green"
gss_cat %>% pull(marital) %>% levels()
# [1] "No answer"      "Never married" "Separated"
# [4] "Divorced"        "Widowed"       "Married"
```



x %>% pull(y)
same as
x\$y



Why this order?



Because of this order

```
gss_cat %>% pull(marital) %>% levels()  
# [1] "No answer"          "Never married" "Separated"  
# [4] "Divorced"           "Widowed"       "Married"
```

Most useful skills

1. Reorder the levels
2. Manipulate the levels



reordering levels



fct_reorder()

Reorders the levels of a factor based on the result of `fun(x)` applied to each group of cases (grouped by level).

```
fct_reorder(f, x, fun = median, ..., .desc = FALSE)
```

factor to
reorder

variable to
reorder by
(in conjunction
with fun)

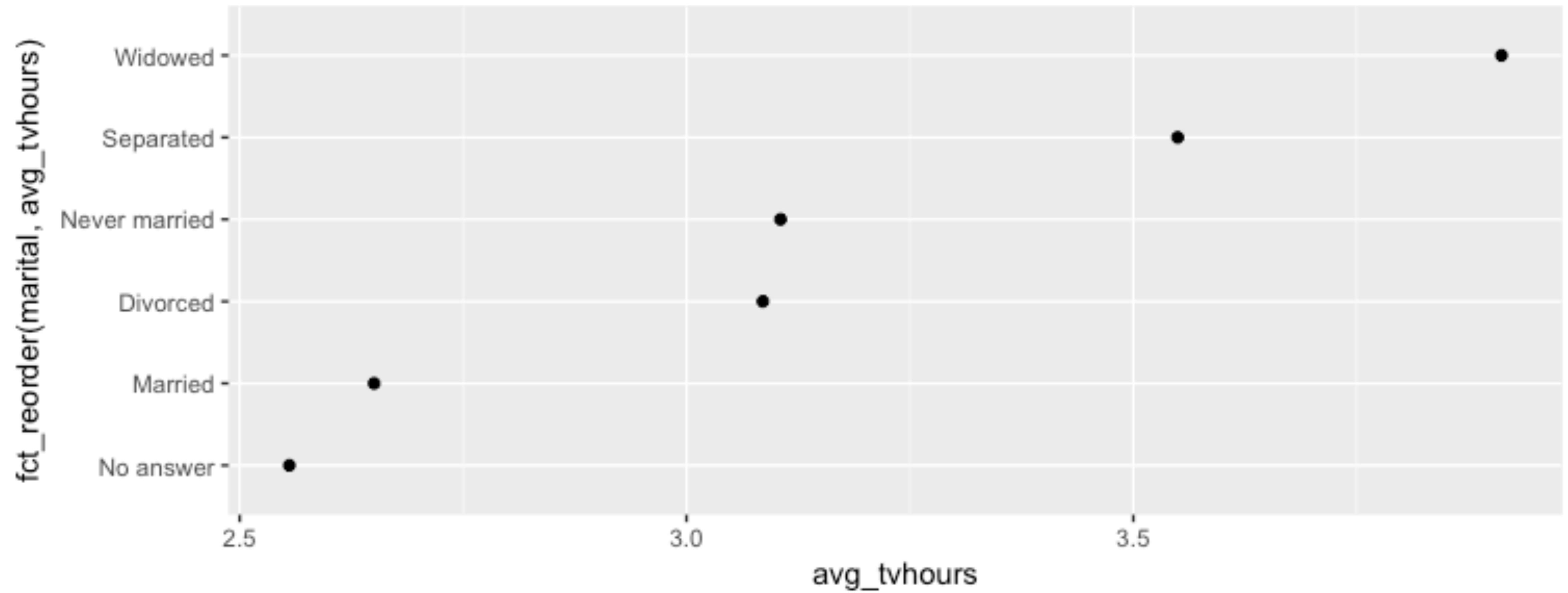
function to
reorder by
(in conjunction
with x)

put in
descending
order?



```
gss_cat %>%  
  drop_na(tvhours) %>%  
  group_by(marital) %>%  
  summarise(avg_tvhours = mean(tvhours)) %>%  
  ggplot() +  
    geom_point(aes(x = avg_tvhours,  
                  y = fct_reorder(marital, avg_tvhours)))
```





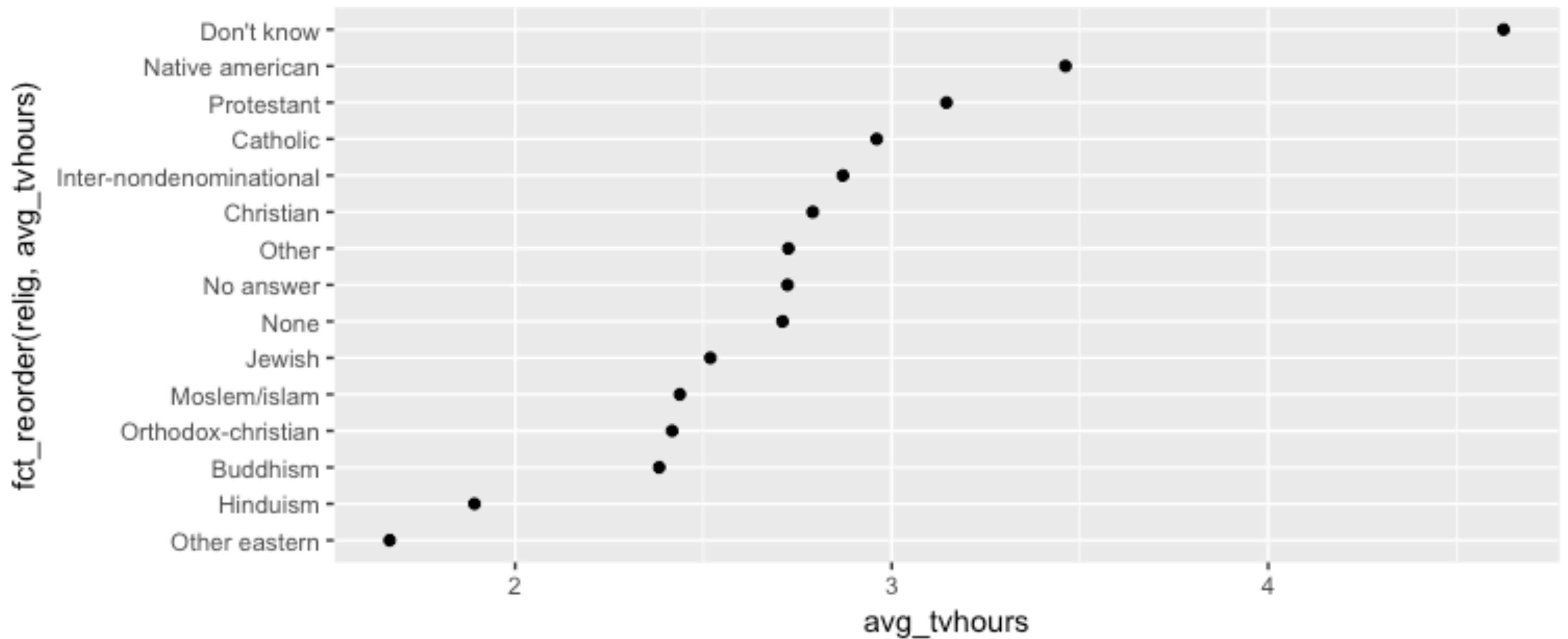
Your Turn 3

Fill in the blanks (____) to explore the average hours of tv watched by religion.

```
gss_cat %>%  
  drop_na(____) %>%  
  group_by(____) %>%  
  summarise(____) %>%  
  ggplot() +  
  geom_point(mapping = aes(x = ____, y = _____))
```

```
gss_cat %>%  
  drop_na(tvhours) %>%  
  group_by(relig) %>%  
  summarise(avg_tvhours = mean(tvhours)) %>%  
  ggplot() +  
  geom_point(mapping = aes(x = avg_tvhours,  
                            y = fct_reorder(relig, avg_tvhours)))
```





Other reordering

Values and level labels are unchanged

- `fct_shuffle()` Randomize order
- `fct_relevel()` Order "by hand"
- `fct_infreq()` Order from most to least frequent
- `fct_inorder()` Order from first to last observed
- `fct_rev()` Reverse the current order
- `fct_shift()` Shift the order by 1

Key `forcats` functions
start with `fct_`

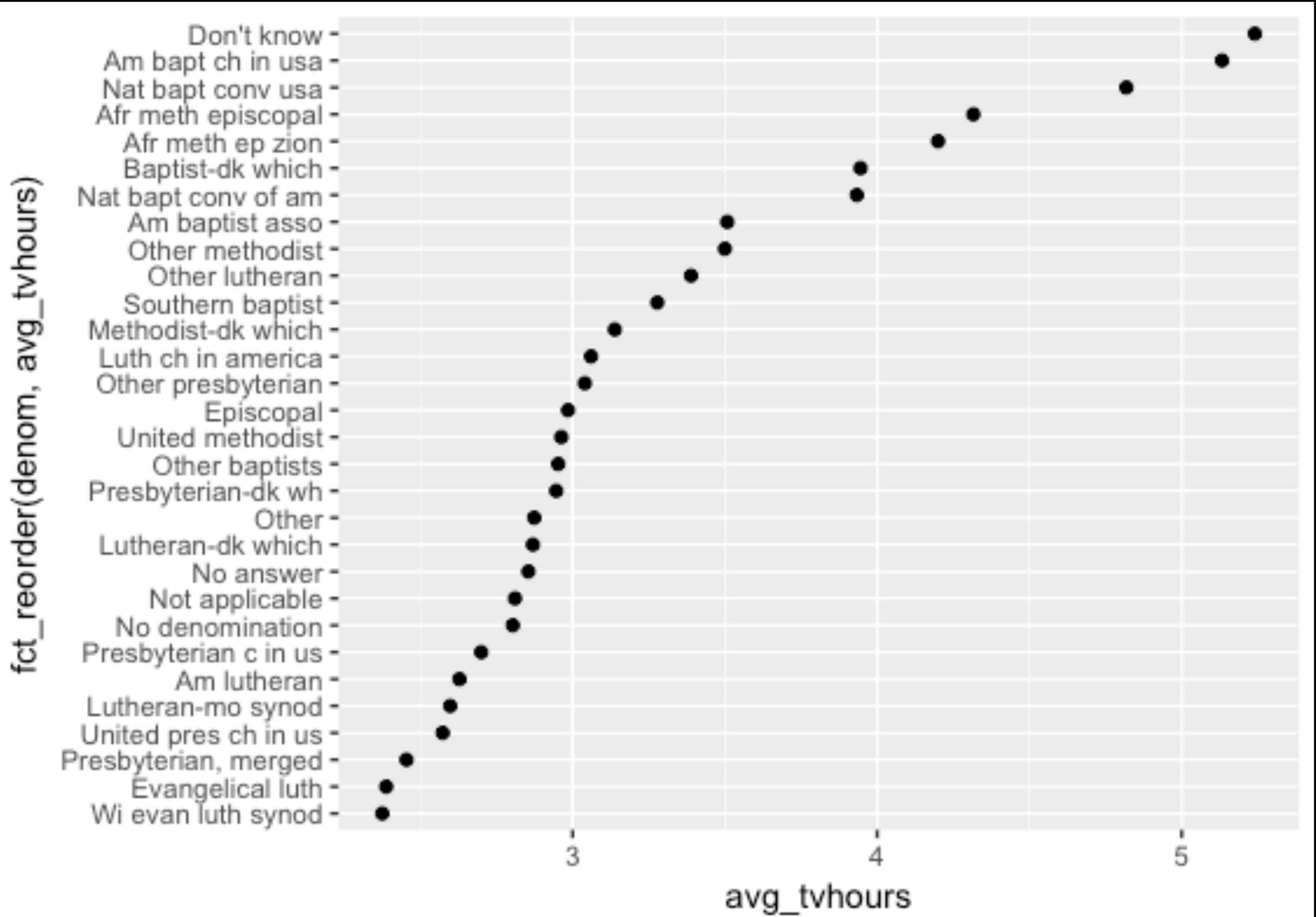


manipulating levels

R

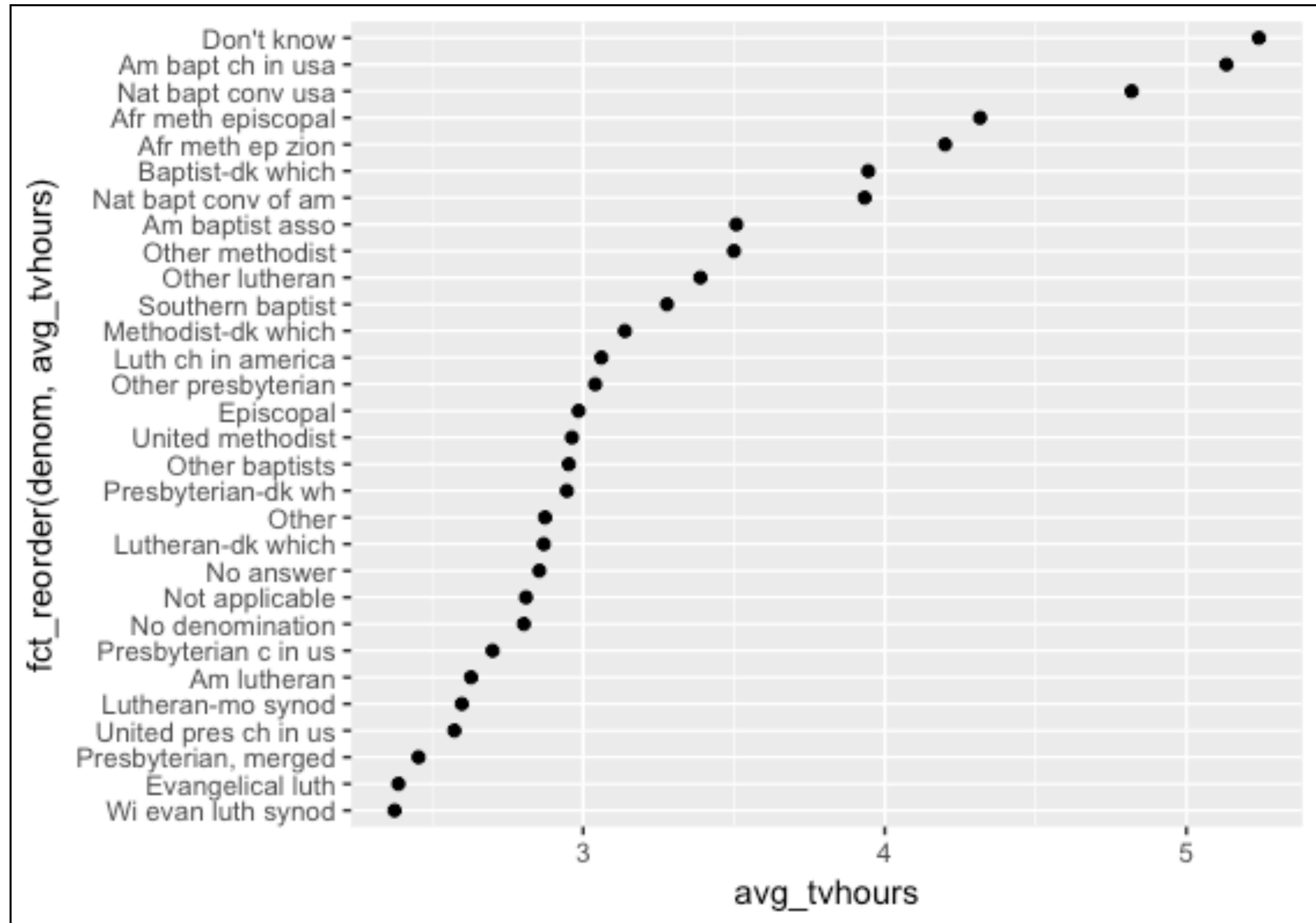
Quiz

Why is this plot not very useful?



Too many categories

Poorly labelled



First 10 rows

Obs.	denom
1	Southern baptist
2	Baptist-dk which
3	No denomination
4	Not applicable
5	Not applicable
6	Southern baptist
7	Not applicable
8	Lutheran-mo
9	Other
10	Southern baptist
...	...

Relabel levels

denom
Baptist - Southern
Baptist - Don't know
No denomination
Not applicable
Not applicable
Baptist - Southern
Not applicable
Lutheran - Missouri
Other
Baptist - Southern
...

Reduce levels

denom
Baptist
Baptist
None
None
None
Baptist
None
Lutheran
Other
Baptist
...

Level manipulation functions

Values change to match levels

Relabel	{	fct_recode()	Relabel levels "by hand"
		fct_anon()	Anonymize levels
	{	fct_relabel()	Relabel using a function
Reduce	{	fct_collapse()	Collapse levels "by hand"
		fct_lump()	Lump levels with small counts <i>together</i>
		fct_other()	Replace levels with "Other"



fct_recode()

Changes values of levels

```
fct_recode(f, ...)
```

factor with
levels

new level = old level
pairs



```
gss_cat %>%  
  pull(denom) %>%  
  levels()
```

[1]	"No answer"	"Don't know"
[3]	"No denomination"	"Other"
[5]	"Episcopal"	"Presbyterian-dk wh"
[7]	"Presbyterian, merged"	"Other presbyterian"
[9]	"United pres ch in us"	"Presbyterian c in us"
[11]	"Lutheran-dk which"	"Evangelical luth"
[13]	"Other lutheran"	"Wi evan luth synod"
[15]	"Lutheran-mo synod"	"Luth ch in america"
[17]	"Am lutheran"	"Methodist-dk which"
[19]	"Other methodist"	"United methodist"
[21]	"Afr meth ep zion"	"Afr meth episcopal"
[23]	"Baptist-dk which"	"Other baptists"
[25]	"Southern baptist"	"Nat bapt conv usa"
[27]	"Nat bapt conv of am"	"Am bapt ch in usa"
[29]	"Am baptist asso"	"Not applicable"



```

gss_cat %>%
  mutate(denom = fct_recode(denom,
    "Baptist - Southern" = "Southern baptist"))
) %>%
  pull(denom) %>%
  levels()

```

[1]	"No answer"	"Don't know"
[3]	"No denomination"	"Other"
[5]	"Episcopal"	"Presbyterian-dk wh"
[7]	"Presbyterian, merged"	"Other presbyterian"
[9]	"United pres ch in us"	"Presbyterian c in us"
[11]	"Lutheran-dk which"	"Evangelical luth"
[13]	"Other lutheran"	"Wi evan luth synod"
[15]	"Lutheran-mo synod"	"Luth ch in america"
[17]	"Am lutheran"	"Methodist-dk which"
[19]	"Other methodist"	"United methodist"
[21]	"Afr meth ep zion"	"Afr meth episcopal"
[23]	"Baptist-dk which"	"Other baptists"
[25]	"Baptist - Southern"	"Nat bapt conv usa"
[27]	"Nat bapt conv of am"	"Am bapt ch in usa"
[29]	"Am baptist asso"	"Not applicable"



factor with levels

```
gss_cat %>%  
  mutate(denom = fct_recode(denom,  
    "Baptist - Southern" = "Southern baptist")) %>%  
  pull(denom) %>%  
  levels()
```

new level = old level
pairs

[1]	"No answer"	"Don't know"
[3]	"No denomination"	"Other"
[5]	"Episcopal"	"Presbyterian-dk wh"
[7]	"Presbyterian, merged"	"Other presbyterian"
[9]	"United pres ch in us"	"Presbyterian c in us"
[11]	"Lutheran-dk which"	"Evangelical luth"
[13]	"Other lutheran"	"Wi evan luth synod"
[15]	"Lutheran-mo synod"	"Luth ch in america"
[17]	"Am lutheran"	"Methodist-dk which"
[19]	"Other methodist"	"United methodist"
[21]	"Afr meth ep zion"	"Afr meth episcopal"
[23]	"Baptist-dk which"	"Other baptists"
[25]	"Baptist - Southern"	"Nat bapt conv usa"
[27]	"Nat bapt conv of am"	"Am bapt ch in usa"
[29]	"Am baptist asso"	"Not applicable"



Your Turn

Edit the code to also relabel some other Baptist denominations:

- "Baptist-dk which"
- "Other baptists"

```
gss_cat %>%  
  mutate(denom = fct_recode(denom,  
    "Baptist - Southern" = "Southern baptist",  
    "Baptist - Don't know" = "Baptist-dk which",  
    "Baptist - Other" = "Other baptists")  
  ) %>%  
  pull(denom) %>%  
  levels()
```

```
gss_cat %>%  
  mutate(denom = fct_recode(denom,  
    "Baptist - Southern" = "Southern baptist",  
    "Baptist-dk which" = "Baptist - Don't know",  
    "Baptist - Other" = "Other baptists")  
)  
  
# Unknown levels in `f`: Baptist - Don't know
```

Message, but no
warning or error!

Common mistake

Whoops, around
the wrong way!

gss_cat %>% pull(denom) %>% levels()

```
[1] "No answer"           "Don't know"  
[3] "No denomination"    "Other"  
[5] "Episcopal"          "Presbyterian-dk wh"  
[7] "Presbyterian, merged" "Other presbyterian"  
[9] "United pres ch in us" "Presbyterian c in us"  
[11] "Lutheran-dk which"   "Evangelical luth"  
[13] "Other lutheran"      "Wi evan luth synod"  
[15] "Lutheran-mo synod"    "Luth ch in america"  
[17] "Am lutheran"         "Methodist-dk which"  
[19] "Other methodist"       "United methodist"  
[21] "Afr meth ep zion"     "Afr meth episcopal"  
[23] "Baptist-dk which"     "Other baptists"  
[25] "Southern baptist"      "Nat bapt conv usa"  
[27] "Nat bapt conv of am"   "Am bapt ch in usa"  
[29] "Am baptist asso"       "Not applicable"
```