Bar charts

Some tips and tricks

Liza Bolton

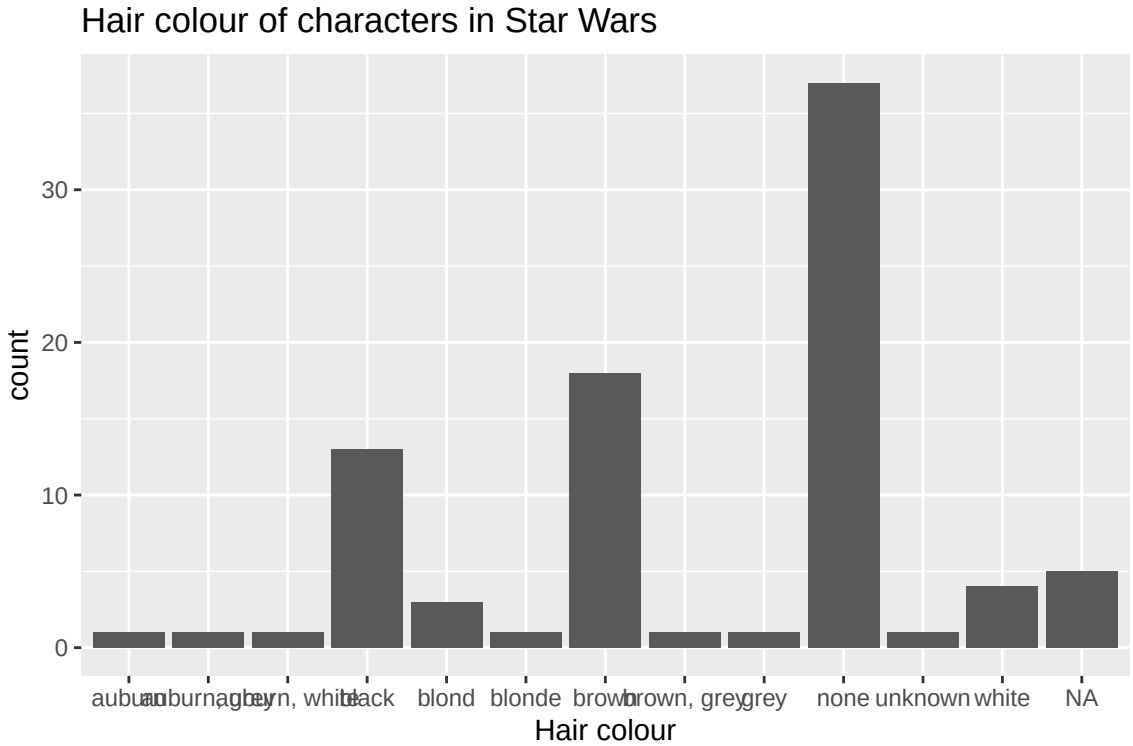
Fall 2020

```
library(tidyverse)
knitr::opts_chunk$set(fig.width = 6, fig.height = 4, message = FALSE)
```

This example uses the `starwars` dataset from `dplyr`.

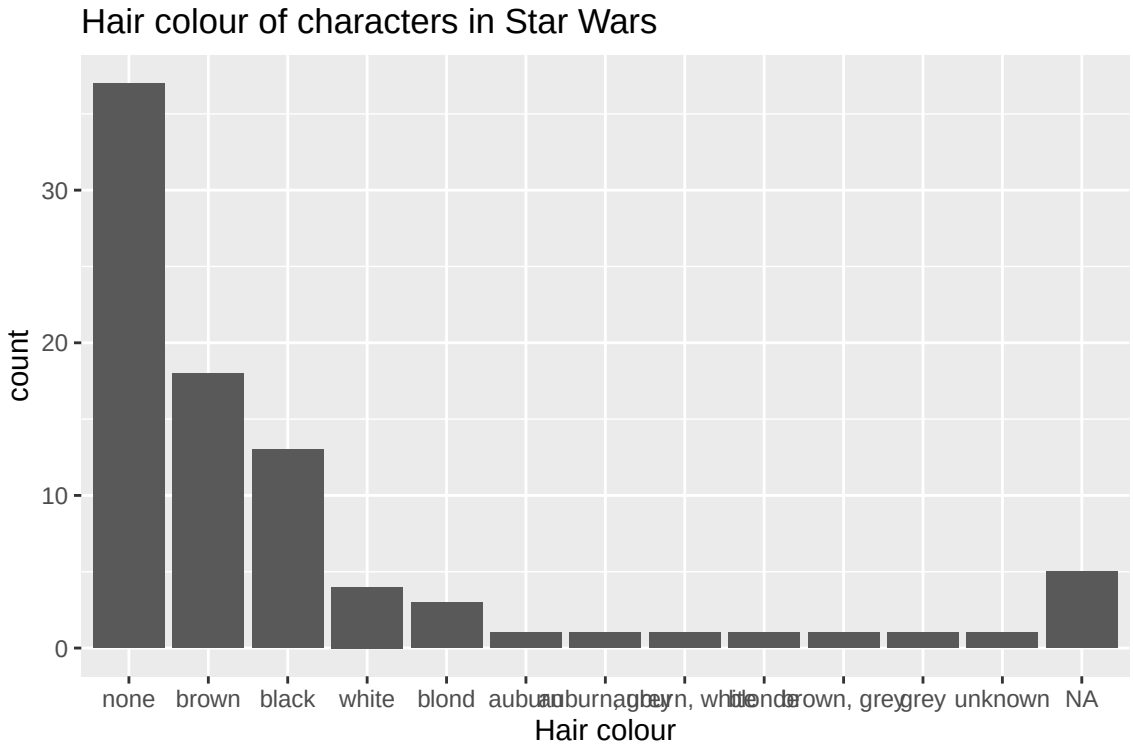
Simple barchart

```
starwars %>%
  ggplot(aes(x = hair_color)) +
  geom_bar() +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour")
```



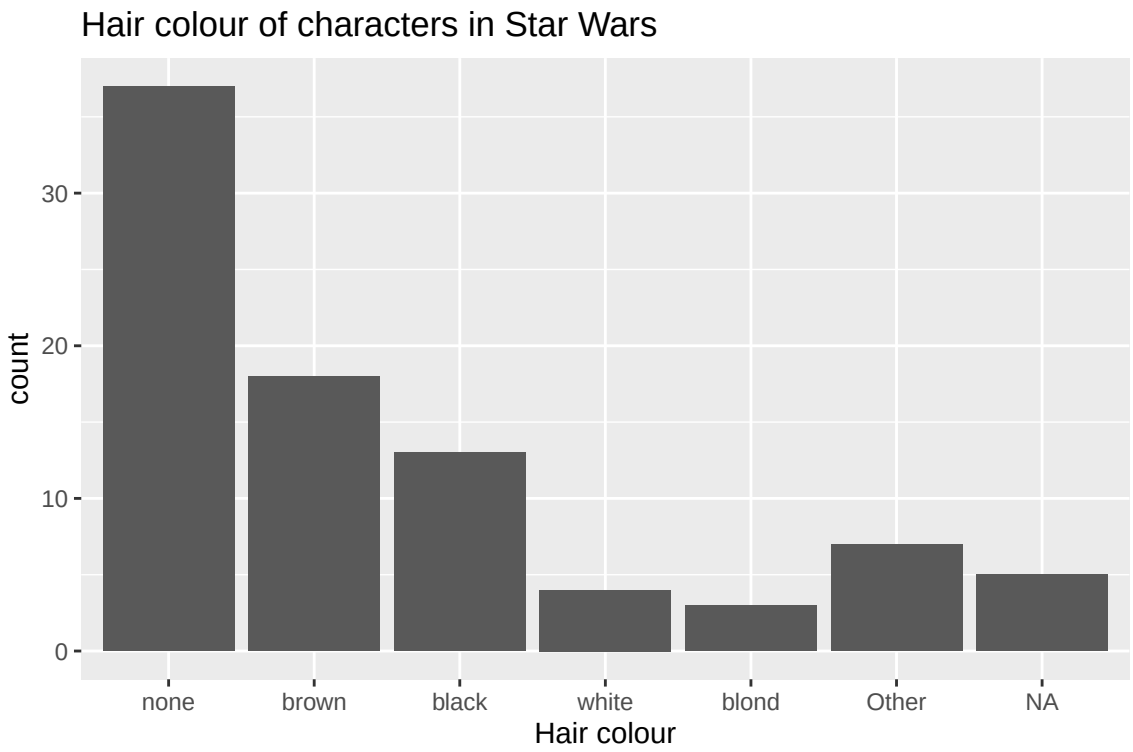
If I want the bars to be in order of frequency instead of alphabetical, I can make use of factors.

```
starwars %>%
  mutate(hair_col_fact = fct_infreq(hair_color)) %>%
  ggplot(aes(x = hair_col_fact)) +
  geom_bar() +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour")
```



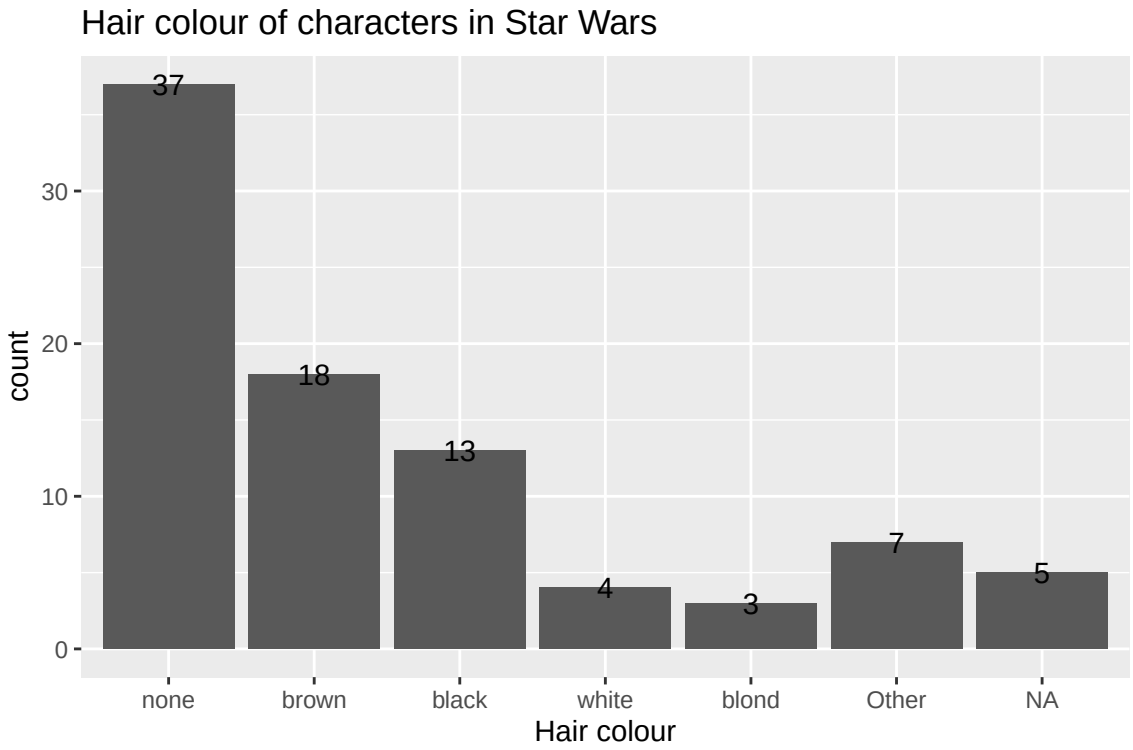
Infact, I can collapse some levels I don't care about!

```
p <- starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact)) +
  geom_bar() +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour")
p
```



Let's add some labels.

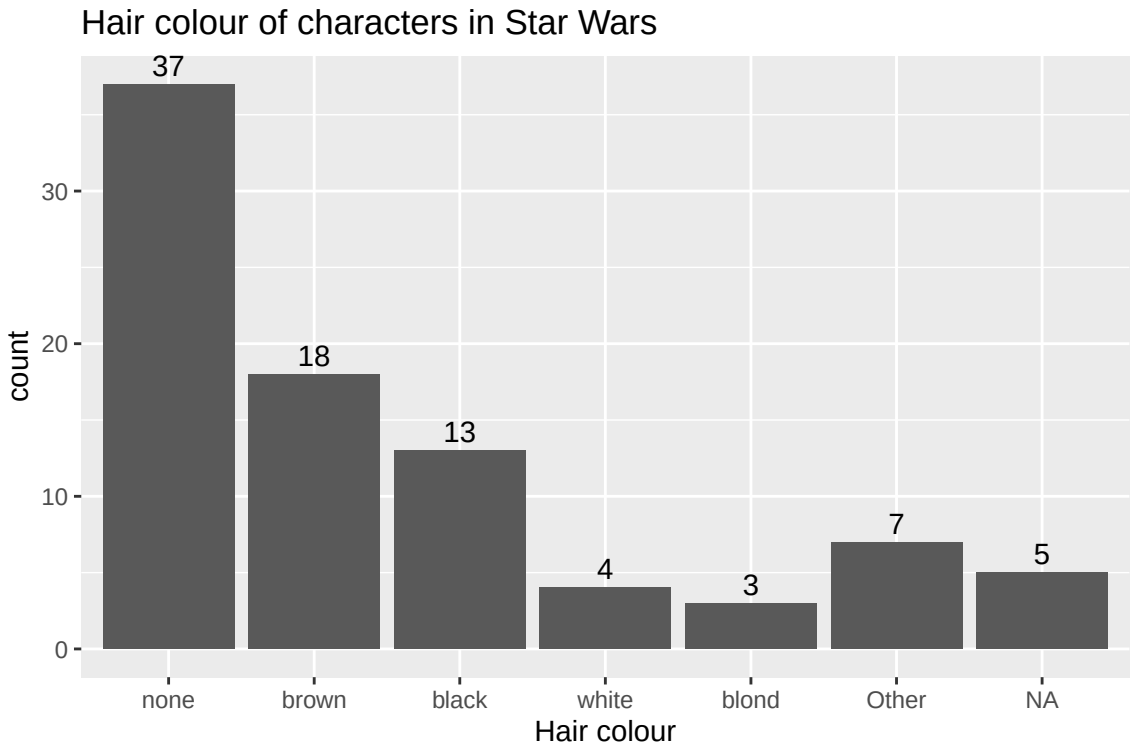
```
p +
  geom_text(aes(label=..count..), stat="count")
```



Eww, those are hard to read!

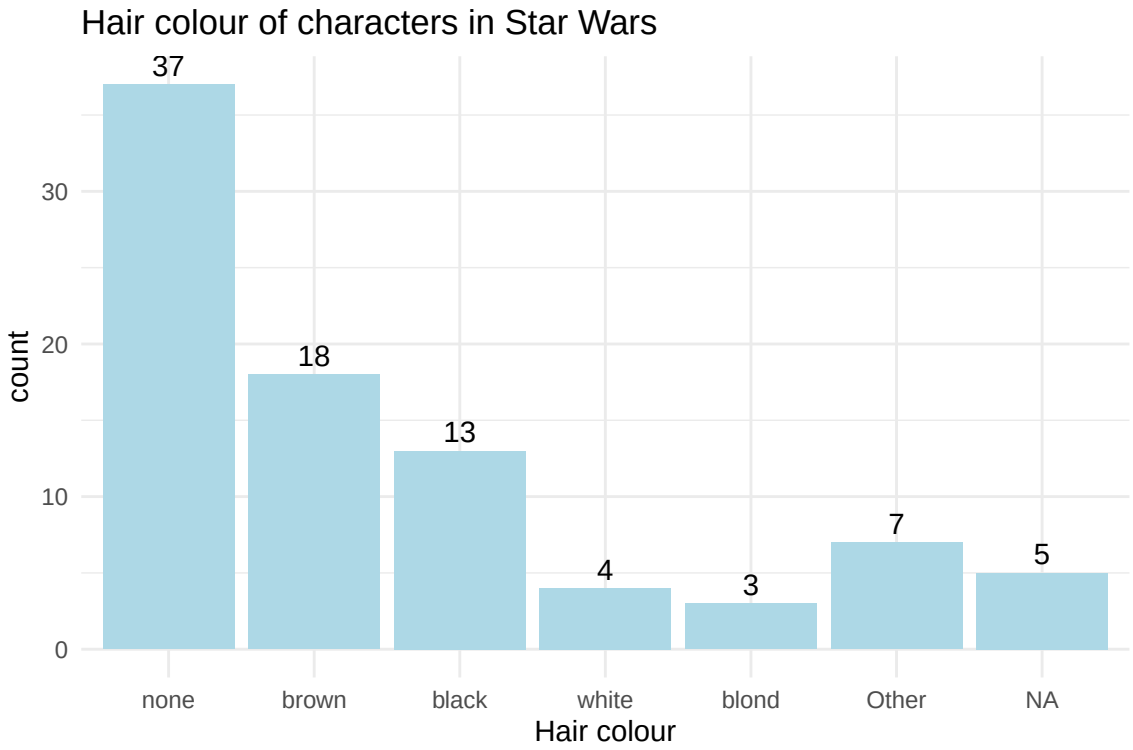
Let's do some **vertical adjustment** (vjust) to try to get them clear of the bars. How do you pick this value for vjust? I just play around until I like it.

```
p +
  geom_text(aes(label=..count..), stat="count", vjust=-0.4)
```



Let's make a few more little cosmetic tweaks.

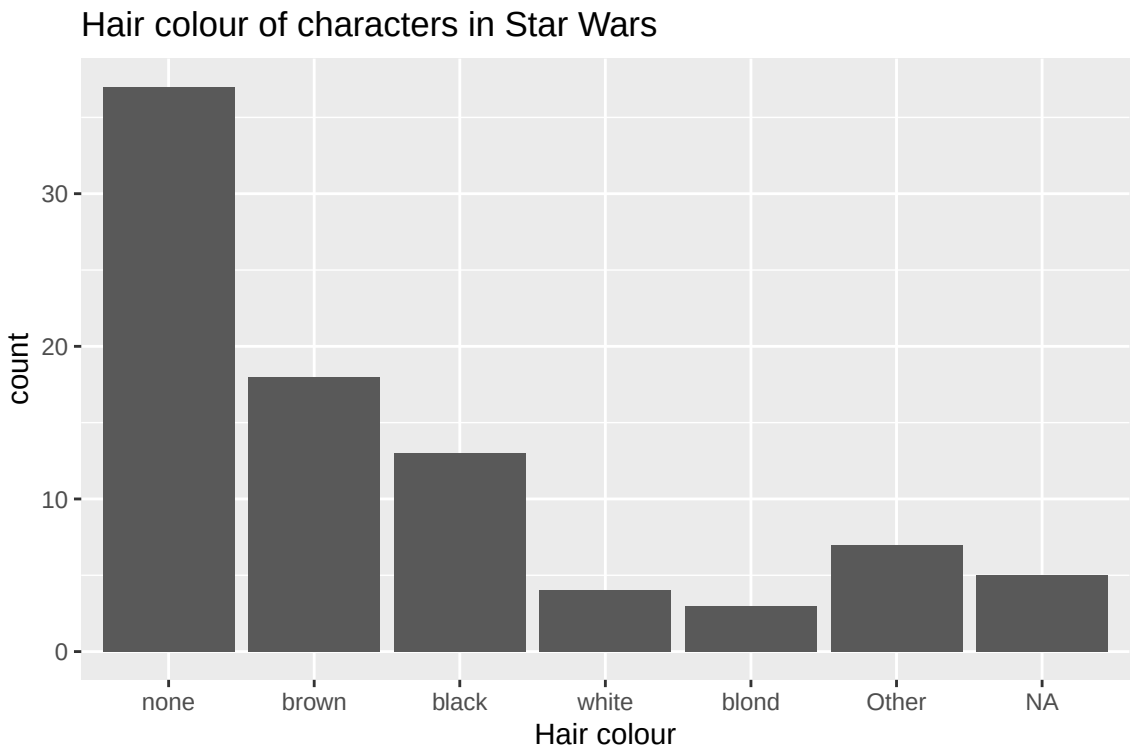
```
starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact)) +
  geom_bar(fill = "lightblue") +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  geom_text(aes(label=..count..), stat="count", vjust=-0.4) +
  theme_minimal()
```



Grouped bar charts

If you want to group your bars, you need to set `group` in the `aes`.

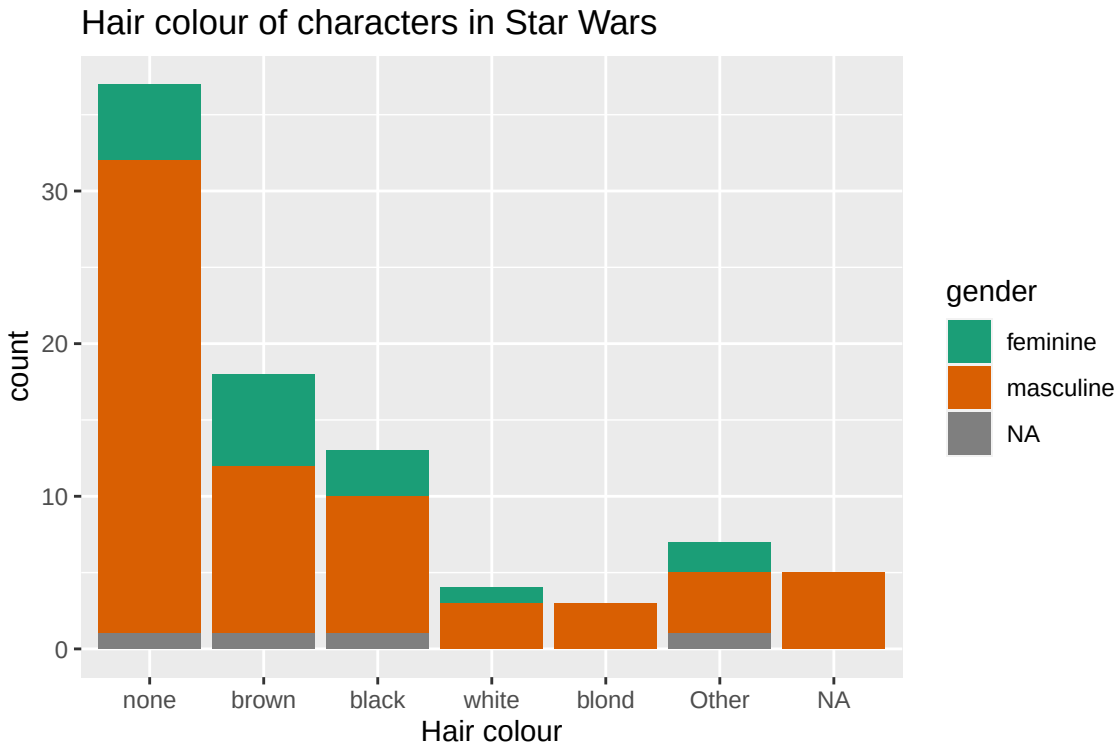
```
starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact, group=gender)) +
  geom_bar() +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour")
```



Hmmm. . . .that looks horrible.

Let's add some fill colour to understand what is wrong (adding in the aesthetics).

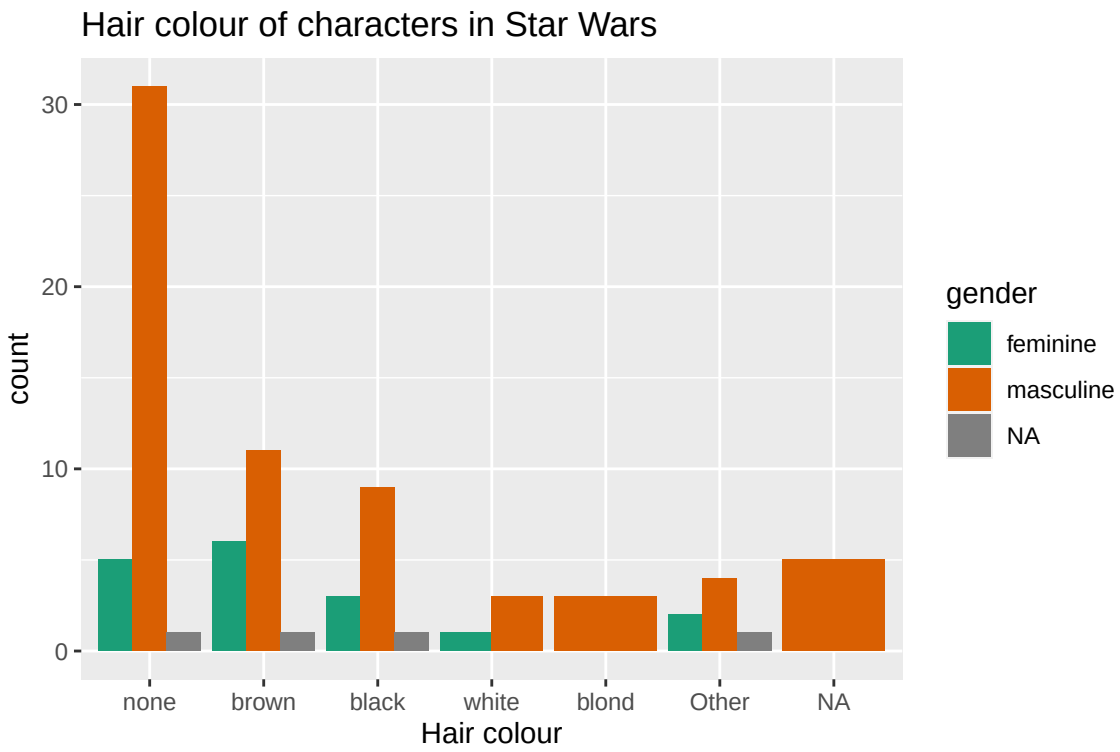
```
starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact, group=gender, fill = gender)) +
    # ^ addign fill = gender colours by that var
  geom_bar() +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  scale_fill_brewer(palette="Dark2", na.value = "grey50") # custom color
```



Oh! The bars are stacked. But I want them side by side. So I need to tell it that.

I can use position = "dodge" to put the bar side-by-side instead of stacking them.

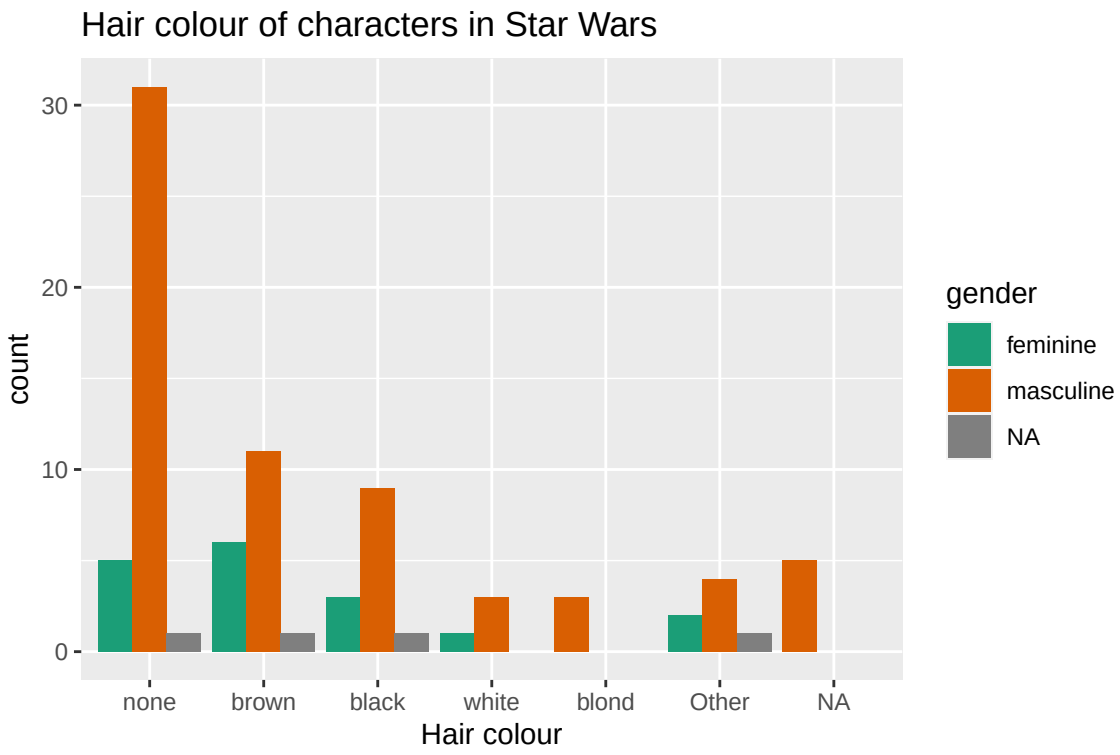
```
starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact, group=gender, fill = gender)) +
  geom_bar(position = "dodge") +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
```



Oh no! Why are my bars different widths?

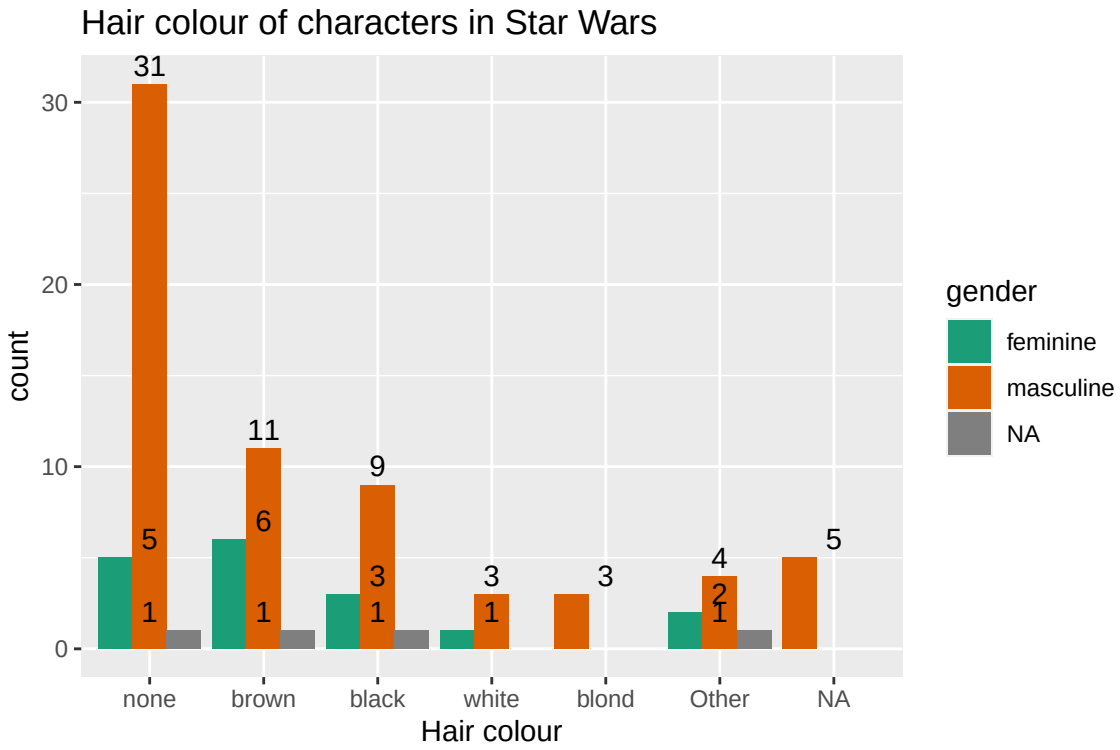
I can add some more commands within position dodge to achieve this. Run `?position_dodge` in your console to learn more about it.

```
p2 <- starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
    # ^ lump low freqs into "Other"
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
    # ^ put in order of frequency
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
    # ^ move "Other" to the end (but NA will be put last by R)
  ggplot(aes(x = hair_col_fact, group=gender, fill = gender)) +
  geom_bar(position = position_dodge(0.9, preserve = "single")) +
    # ^ maing this a little more complicated so it does what I want
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
p2
```



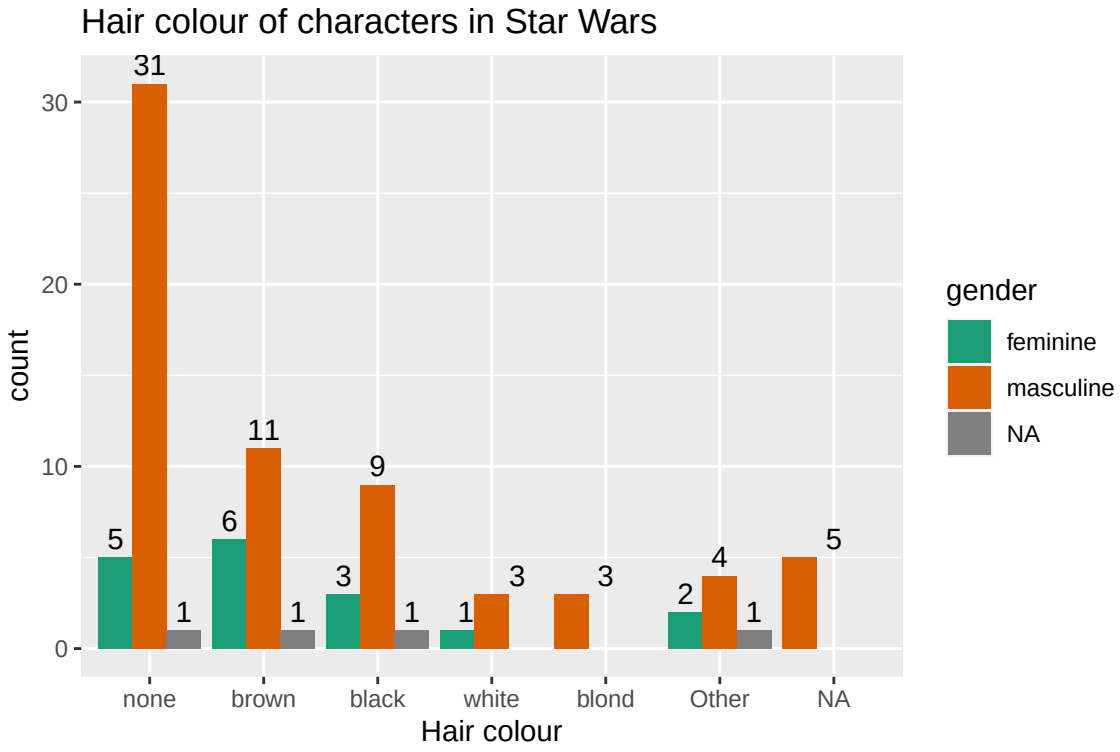
Let's add some labels!

```
p2 +
   geom_text(aes(label=..count..), stat="count", vjust=-0.4) +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
```



What? Oh no! That's totally wrong!

```
p2 +
   geom_text(aes(label=..count..), stat="count", vjust=-0.4,
             position = position_dodge(width = 0.9)) +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
```



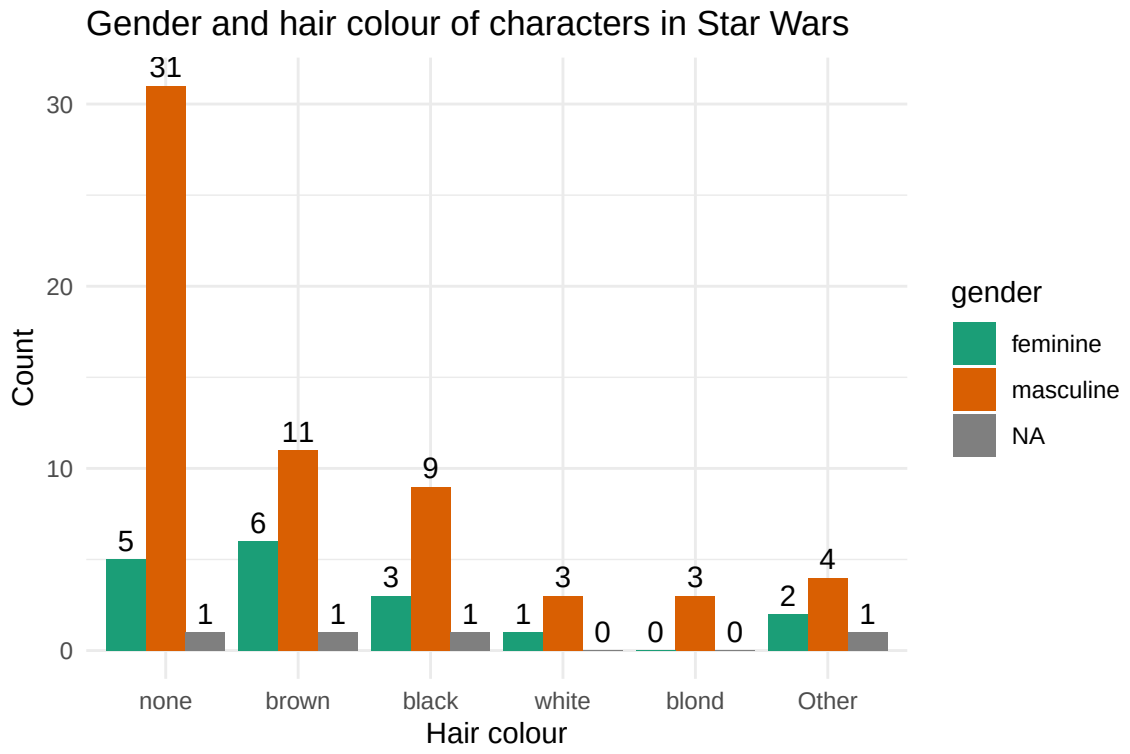
I couldn't actually find a good way to deal with this online. So you have two options, (1) if you want to be a little fancy you could 'expand' the data set so it has all posible combinations and set the missing combos to zero, or (2) just stop caring that the bars were different widths and just go with it.

Option 1

```
my_starwars <- starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf))

# create all possible level combinations
all <- my_starwars %>%
  expand(hair_col_fact, gender)

# create summary of that data and use that
my_starwars %>%
  group_by(hair_col_fact, gender) %>%
  count() %>%
  right_join(all) %>%
  mutate(n = ifelse(is.na(n), 0, n)) %>%
  ggplot(aes(x = hair_col_fact, y = n, group=gender, fill = gender)) +
  geom_bar(stat = "identity", position = position_dodge(0.9, "single")) +
  # ^ making the above a little more complicated so it does what I want
  ggtitle('Gender and hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  ylab("Count") +
  geom_text(aes(label=n), stat="identity", vjust=-0.4,
            position = position_dodge(0.9)) +
  theme_minimal() +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
```



Option 2

```
starwars %>%
  mutate(hair_col_fact = fct_lump_min(as.factor(hair_color), min=3)) %>%
  mutate(hair_col_fact = fct_infreq(hair_col_fact)) %>%
  mutate(hair_col_fact = fct_relevel(hair_col_fact, "Other", after = Inf)) %>%
  ggplot(aes(x = hair_col_fact, group=gender, fill = gender)) +
  geom_bar(position = "dodge") +
  ggtitle('Hair colour of characters in Star Wars') +
  xlab("Hair colour") +
  geom_text(aes(label = stat(count)), stat="count",
            position = position_dodge(width = 0.9), vjust = -0.5) +
  theme_minimal() +
  scale_fill_brewer(palette="Dark2", na.value = "grey50")
```