QStep — Week 4 Lab: Homework Solutions

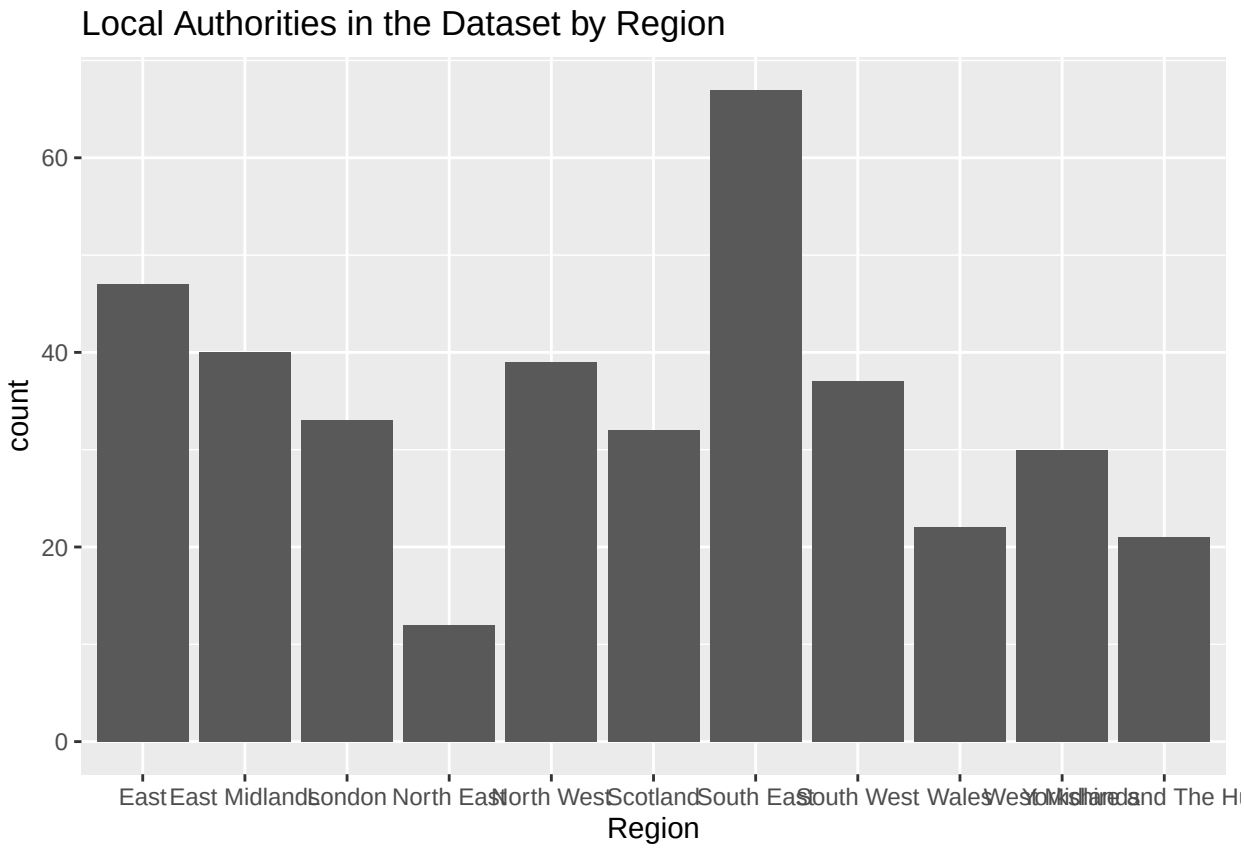
```
library(tidyverse) # load package at start of session
```

- Load the `brexit` data from week 2's lab. Choose and implement a suitable data visualisation for describing...

```
brexit <- read.csv("brexit.csv") # get the brexit data
```

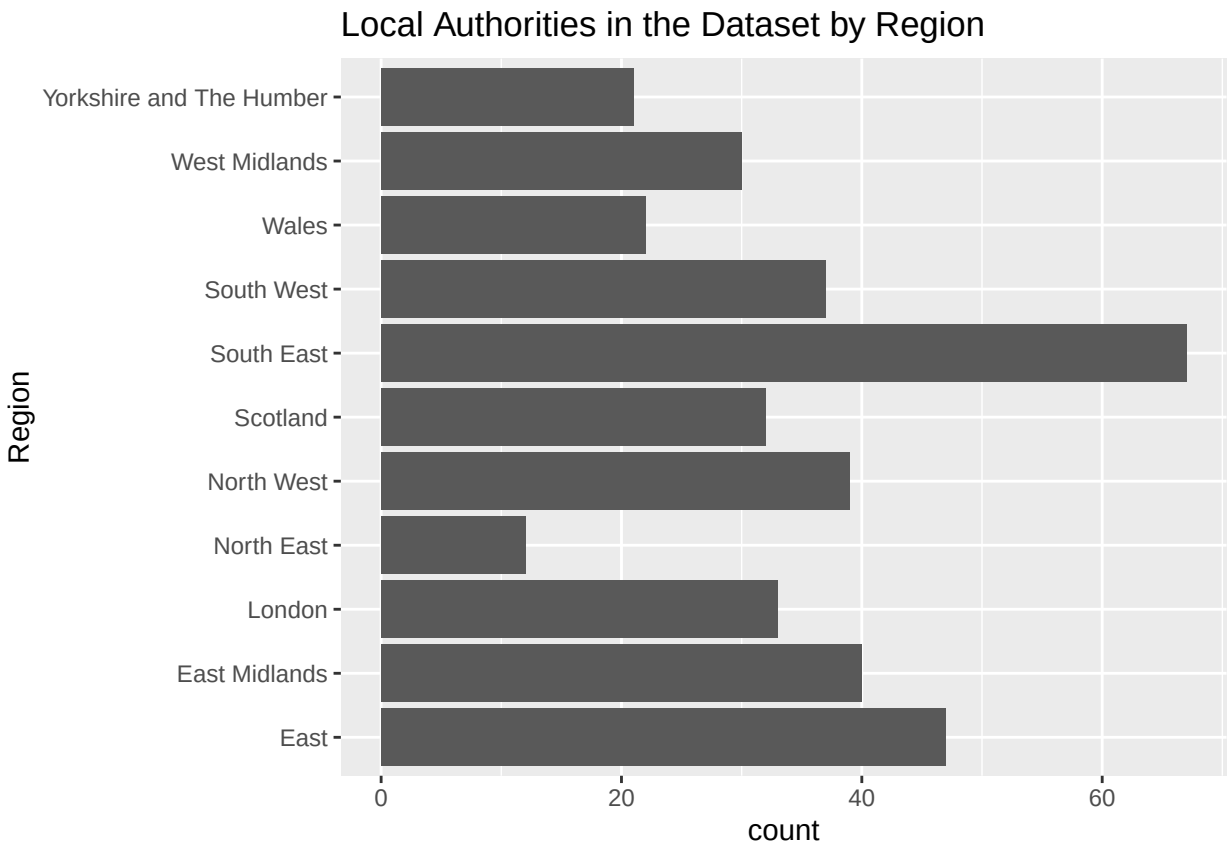
1. the `region` variable

```
ggplot(data = brexit) +
  geom_bar(mapping = aes(x = region)) +
  xlab("Region") + ggtitle("Local Authorities in the Dataset by Region")
```



Bonus: he bar plot is more legible if we put the region variable on the y-axis:

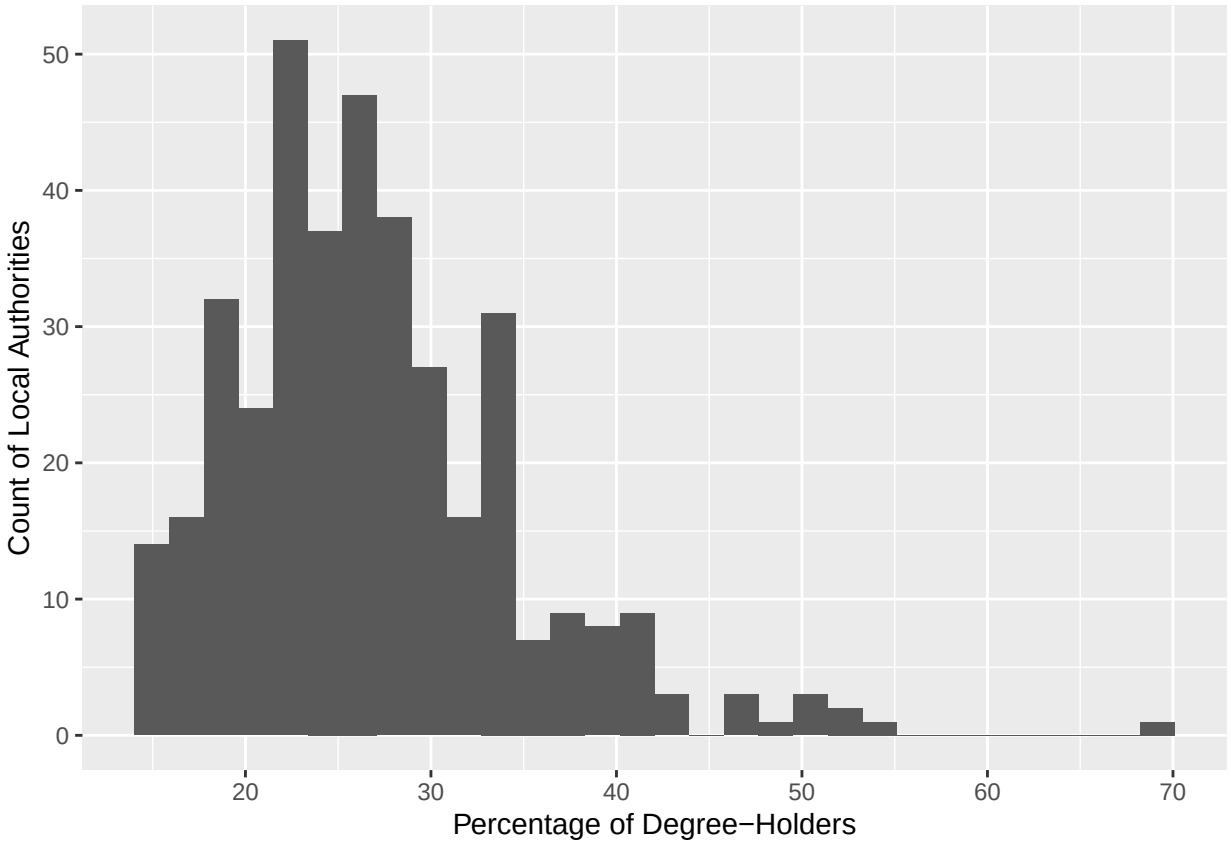
```
ggplot(data = brexit) +
  geom_bar(mapping = aes(y = region)) +
  ylab("Region") + ggtitle("Local Authorities in the Dataset by Region")
```



2. the `percent_degree` variable,

```
ggplot(data = brexit) +
  geom_histogram(mapping = aes(x = percent_degree)) +
  xlab("Percentage of Degree-Holders") + ylab("Count of Local Authorities")
```

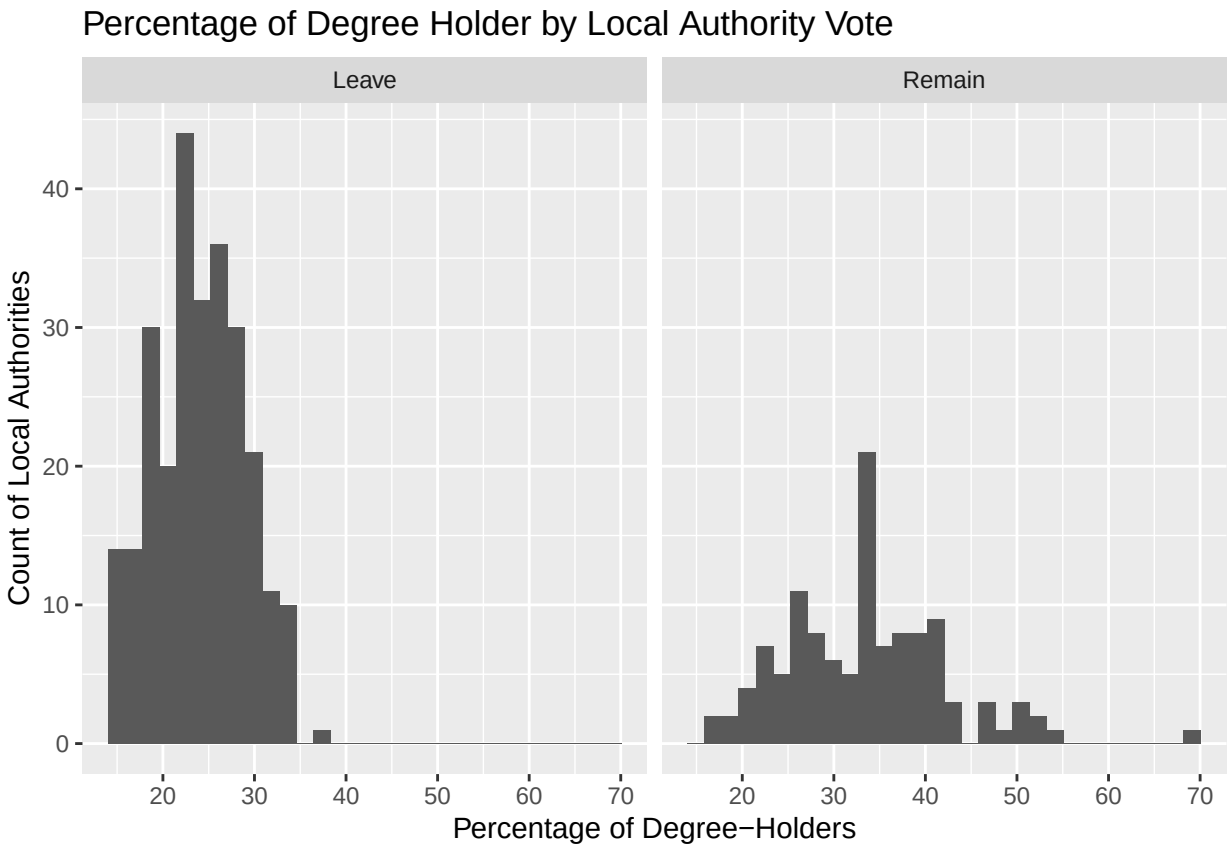
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3. the `percent_degree` variable separately for majority-Leave and majority-Remain areas,

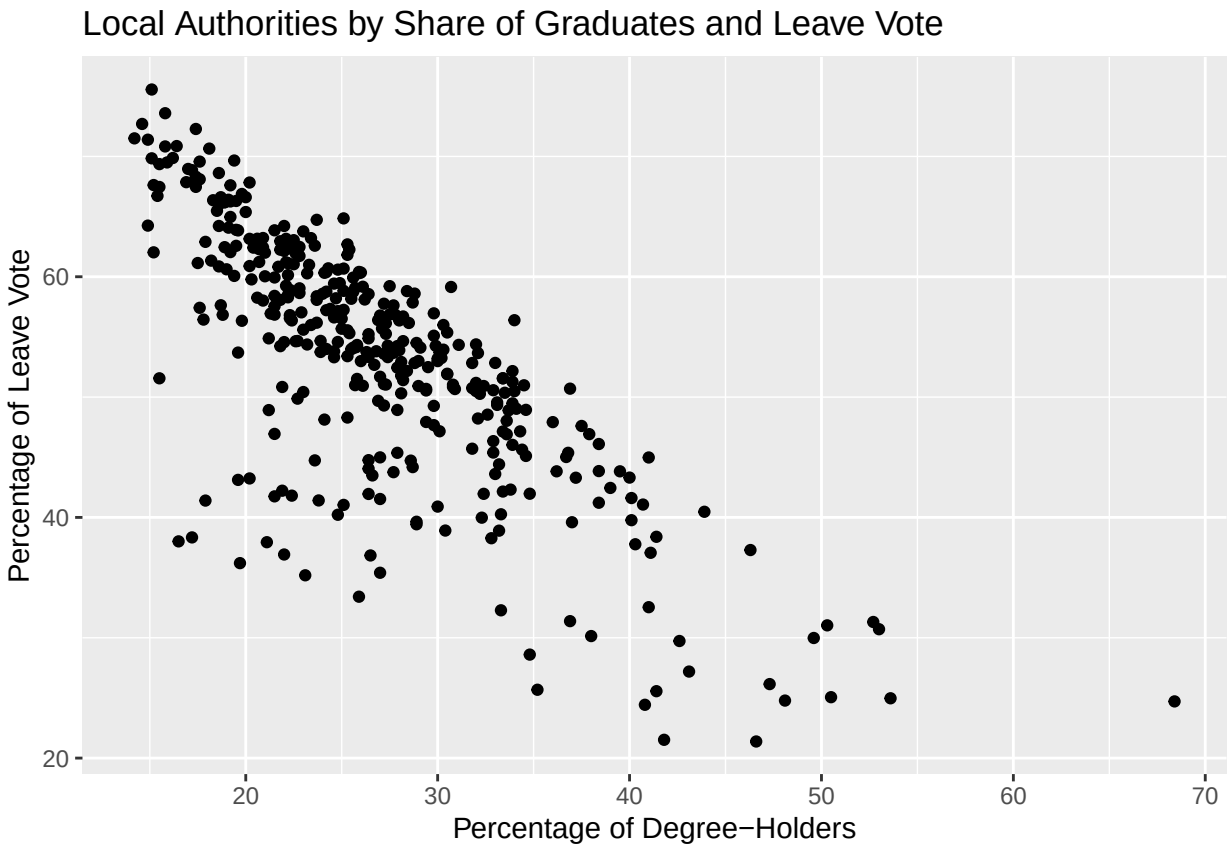
```
#create a 'winner' variable recording whether the LA is Leave- or Remain- majority
brexit$winner <- NA
brexit$winner[brexit$percent_leave > 50] <- "Leave"
brexit$winner[brexit$percent_leave < 50] <- "Remain"

# use a histogram and faceting to plot
ggplot(data = brexit) +
  geom_histogram(mapping = aes(x = percent_degree)) +
  facet_wrap(~winner) +
  xlab("Percentage of Degree-Holders") +
  ylab("Count of Local Authorities") +
  ggtitle("Percentage of Degree Holder by Local Authority Vote")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



4. the relationship between `percent_degree` and `percent_leave`.

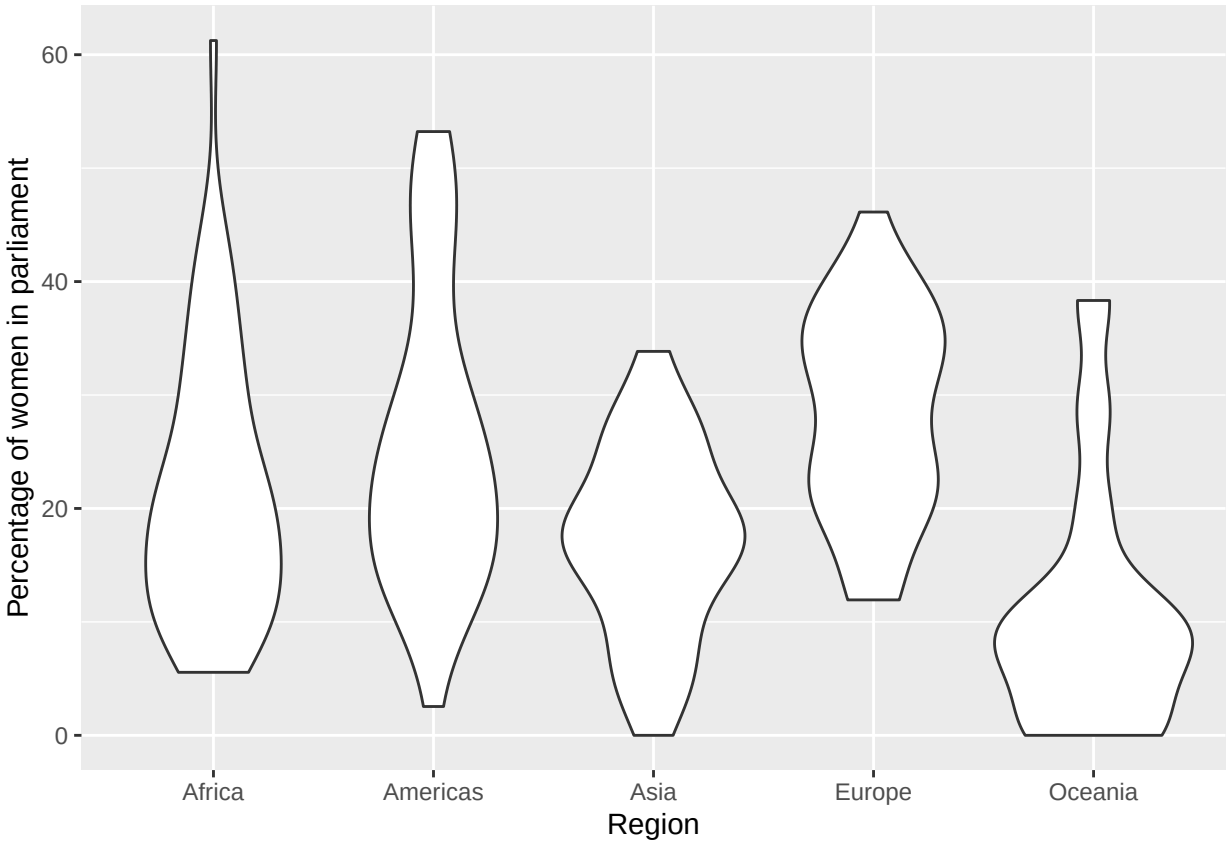
```
# use a scatterplot
ggplot(data = brexit) +
  geom_point(mapping = aes(x = percent_degree, y = percent_leave)) +
  xlab("Percentage of Degree-Holders") +
  ylab("Percentage of Leave Vote") +
  ggtitle("Local Authorities by Share of Graduates and Leave Vote")
```



- Violin Plot

```
qog <- read.csv("qog2022.csv") # get the data

ggplot(data = qog) +
  geom_violin(mapping = aes(x = region, y = wdi_wip)) +
  xlab("Region") + ylab("Percentage of women in parliament")
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_ydensity()‘).
```

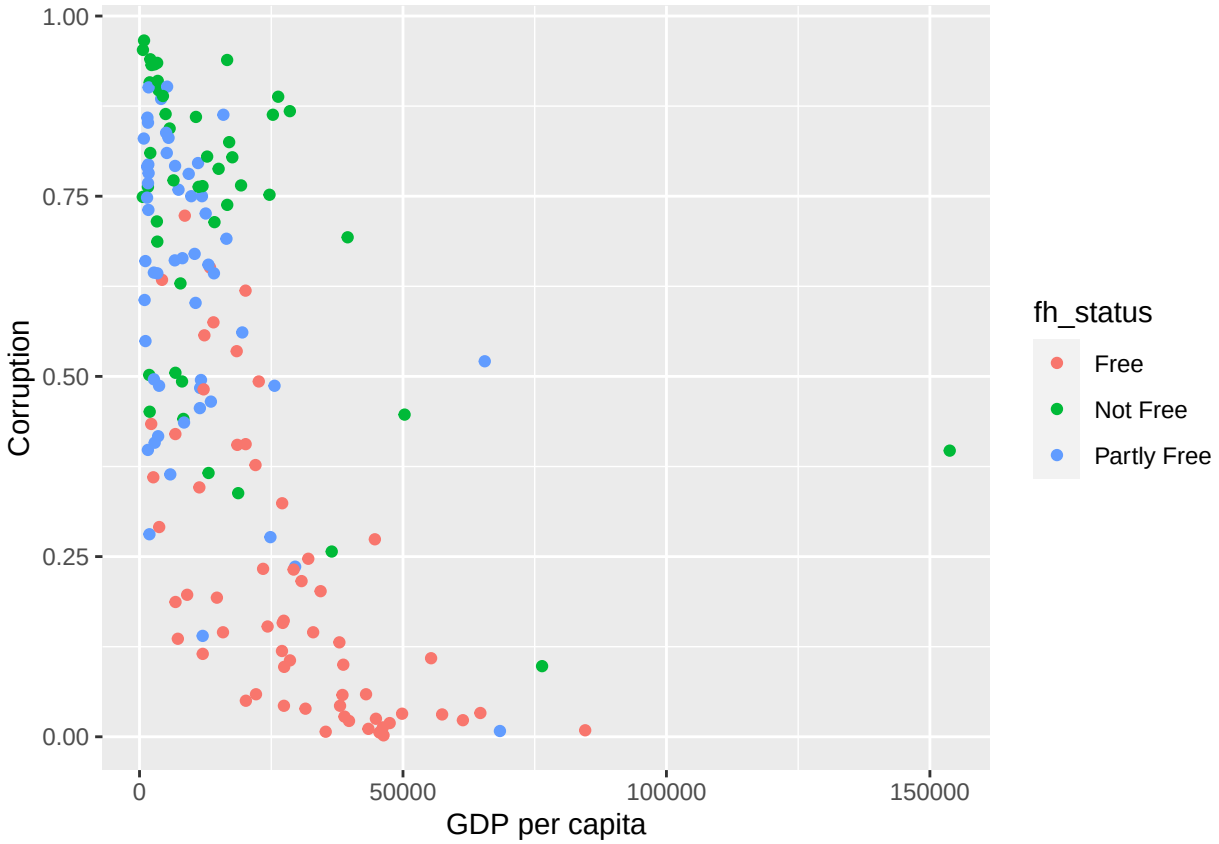


- Colour scales in `ggplot`

1. Use the data in the `qog` database to build a scatterplot with GDP per capita (`mad_gdppc`) on the x-axis, the V-Dem corruption indicator (`vdem_corr`) on the y-axis, color-coded by Freedom House rating (`fh_status`).

```
ggplot(data = qog) +
  geom_point(mapping = aes(x = mad_gdppc, y = vdem_corr, colour = fh_status)) +
  xlab("GDP per capita") + ylab("Corruption")
```

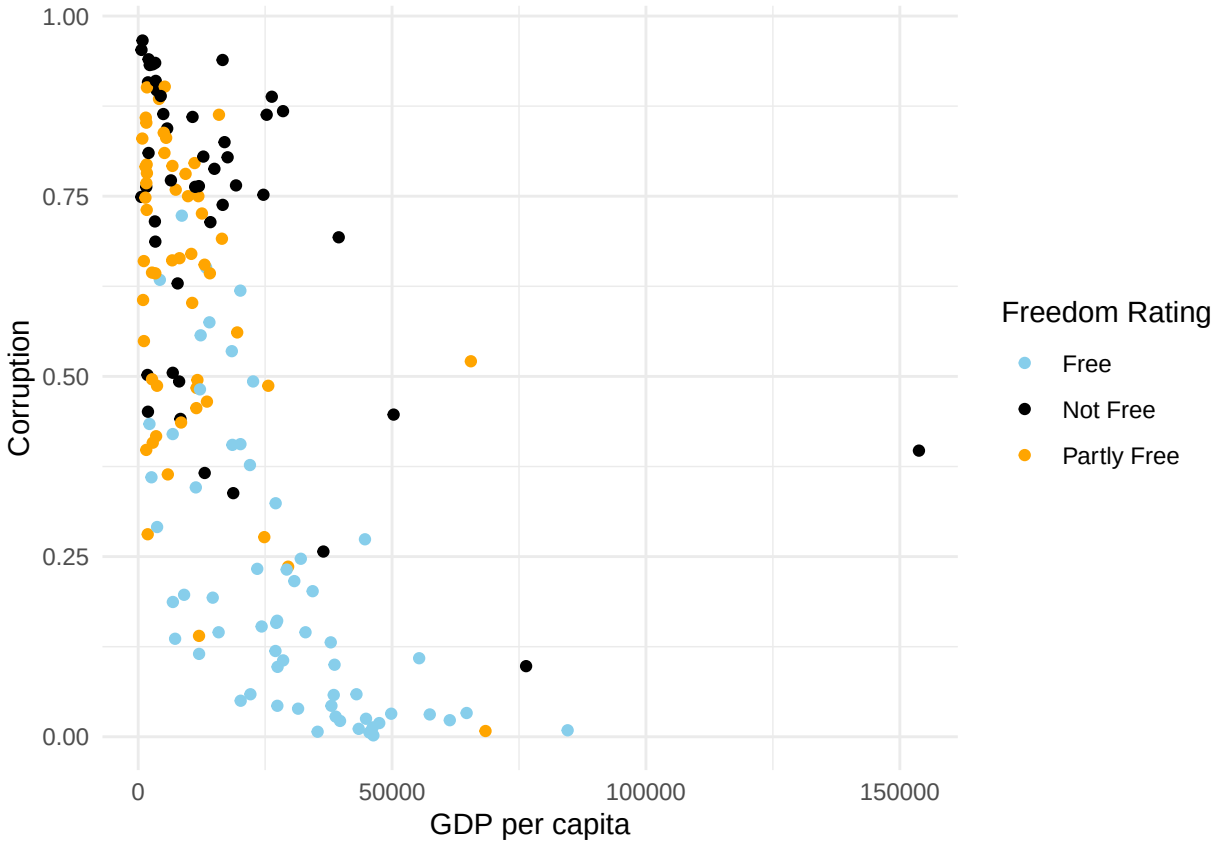
```
## Warning: Removed 33 rows containing missing values (‘geom_point()‘).
```



Bonus: you can use `scale_colour_manual()` to select the colours you want to use and label the legend; you can use `theme_minimal()` to remove the ugly grey background

```
ggplot(data = qog) +
  geom_point(mapping = aes(x = mad_gdppc, y = vdem_corr, colour = fh_status)) +
  xlab("GDP per capita") + ylab("Corruption") +
  scale_colour_manual(values = c("skyblue", "black", "orange"), name = "Freedom Rating") +
  theme_minimal()
```

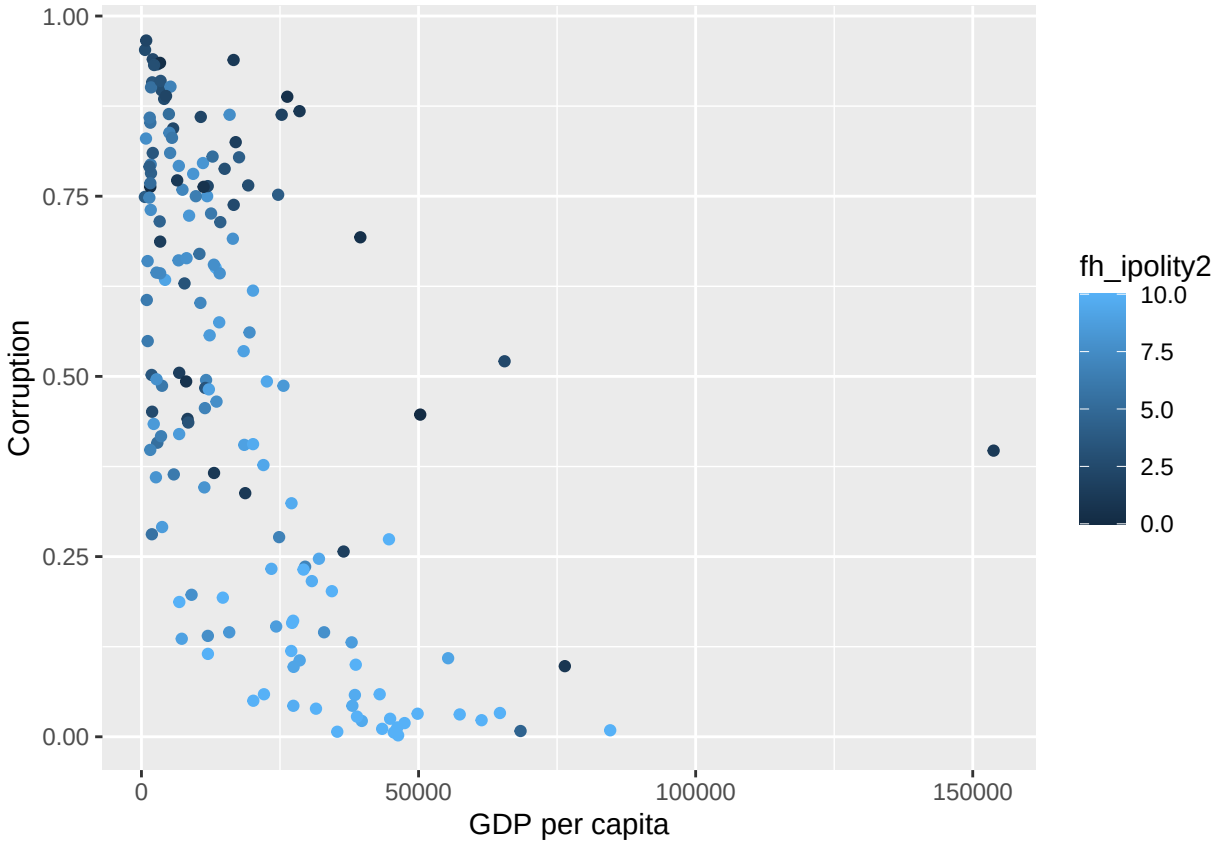
```
## Warning: Removed 33 rows containing missing values (‘geom_point()‘).
```



2. Using the same scatterplot, swap `fh_status` (a categorical variable) for `fh_ipolity2` (a continuous indicator of democracy).

```
ggplot(data = qog) +
  geom_point(mapping = aes(x = mad_gdppc, y = vdem_corr, colour = fh_ipolity2)) +
  xlab("GDP per capita") + ylab("Corruption")
```

```
## Warning: Removed 33 rows containing missing values (‘geom_point()‘).
```

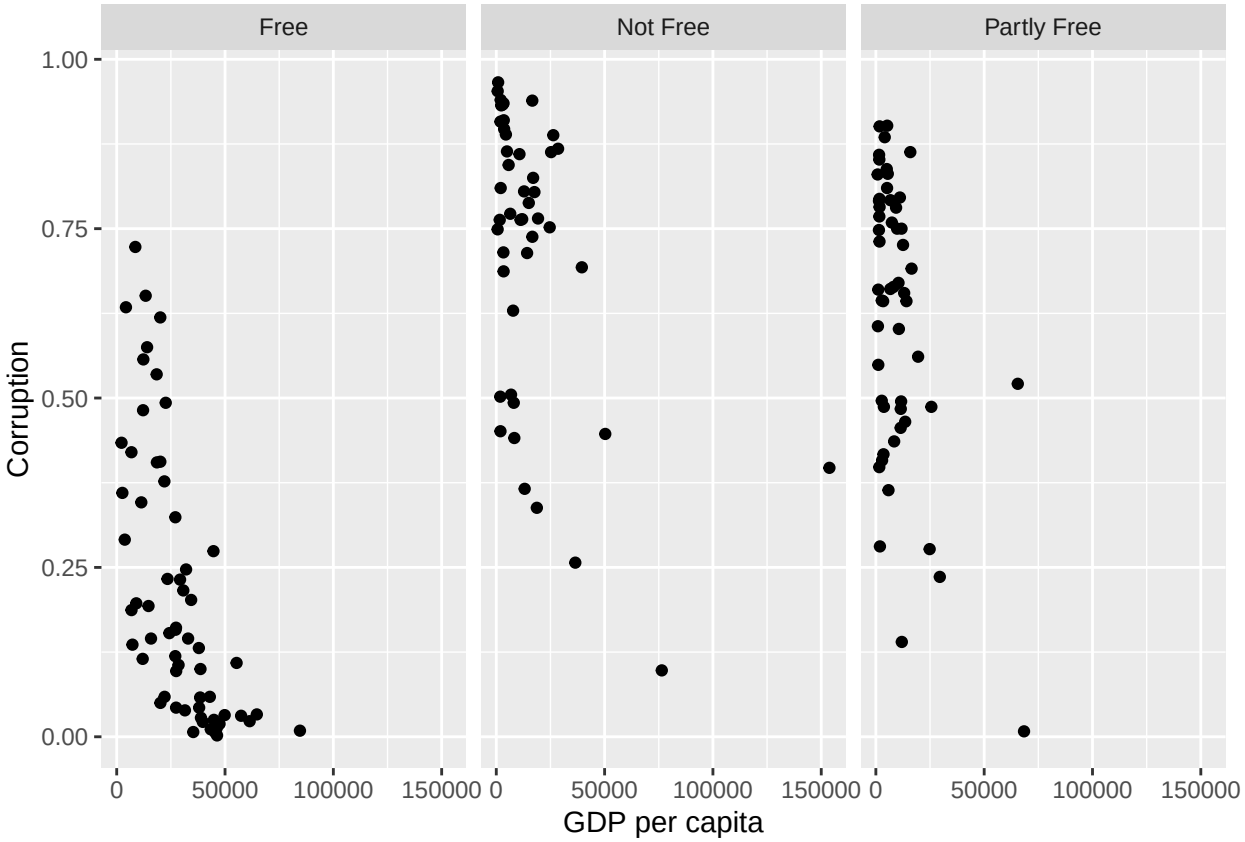


How would you convey the information in the scatterplots above without relying on colour?

With faceting.

```
ggplot(data = qog) +
  geom_point(mapping = aes(x = mad_gdppc, y = vdem_corr)) +
  xlab("GDP per capita") + ylab("Corruption") + facet_wrap(~fh_status)
```

```
## Warning: Removed 33 rows containing missing values (‘geom_point()‘).
```


Free
Not Free
Partly Free
Corruption
1.00
0.75
0.50
0.25
0.00
0
50000
100000
150000
GDP per capita