Workbook_3

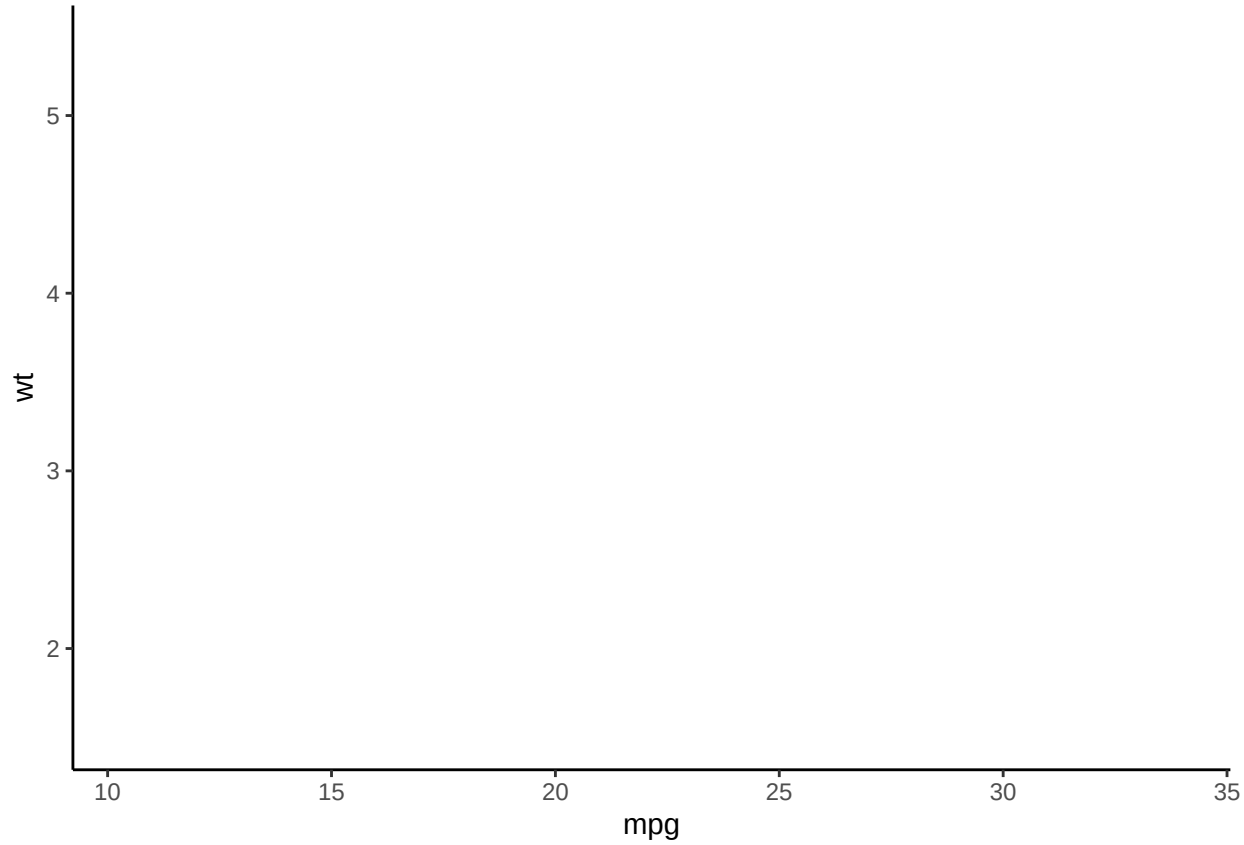
Paul Edwards

10/03/2021

Question 1

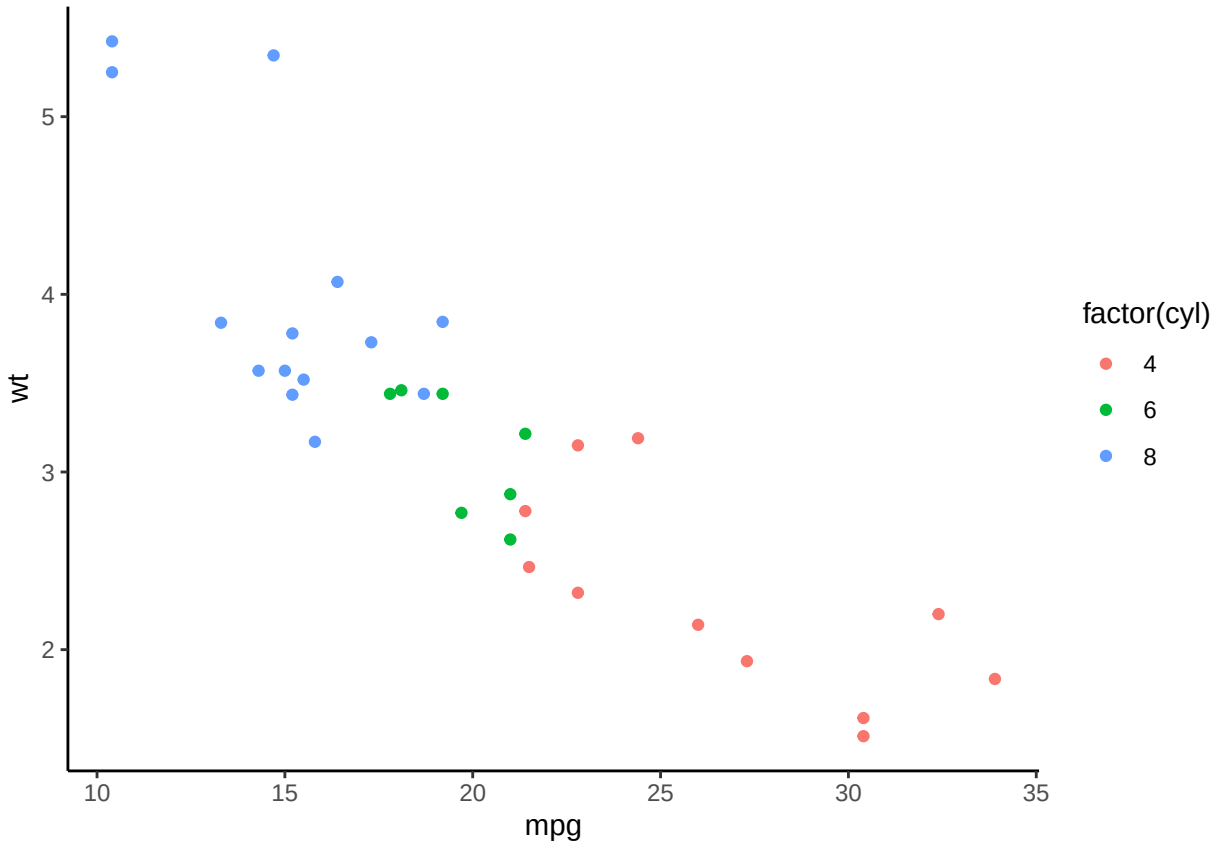
The following code has a error preventing the data from being graphed correctly:

```
ggplot(data = mtcars) +
  aes(mpg, wt, colour = factor(cyl))
```



Here, the `aes()` function must be inside a `geom_` function such as `geom_point()`.

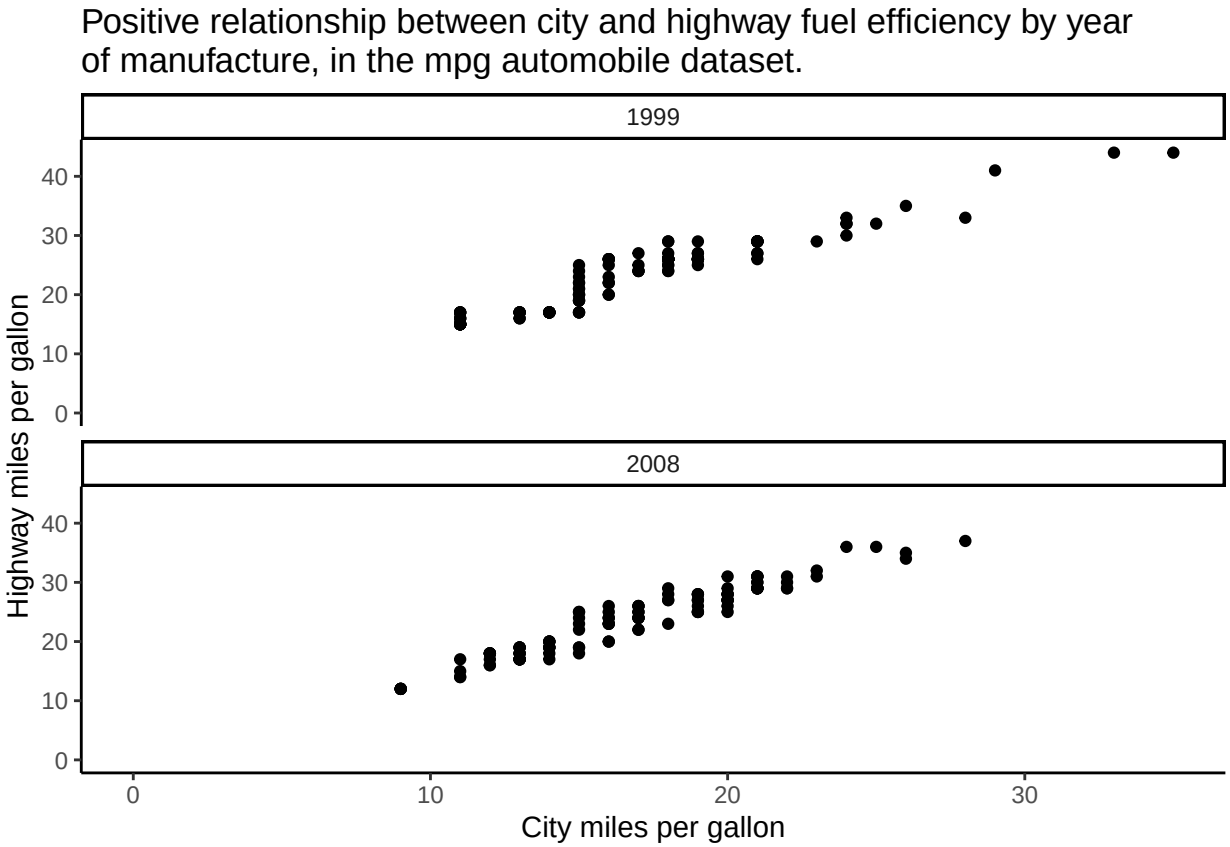
```
#Plot mtcars dataset using scatterplot
ggplot(data = mtcars) +
  geom_point(mapping = aes(mpg, wt, colour = factor(cyl)))
```



Questions 2 and 3

We can use the `facet()` function to compare different classes, such as year of manufacture. We can also set the plot axes start values to zero using `expand_limits()` and setting x and y to zero.

```
#Plot mpg dataset and set axes start values to zero
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy)) +
  labs(title = str_wrap("Positive relationship between city and highway fuel efficiency by year of manu
  facet_wrap(~ year, nrow = 2) +
  xlab("City miles per gallon") +
  ylab("Highway miles per gallon") +
  expand_limits(x = 0, y = 0)
```



Question 4

The benefit of setting the x and y axes to zero is that this avoids misleading readers about the scale of the changes represented by the data in the graph. Truncating axes effectively zooms into the graph, potentially making small variations appear larger than they actually are.

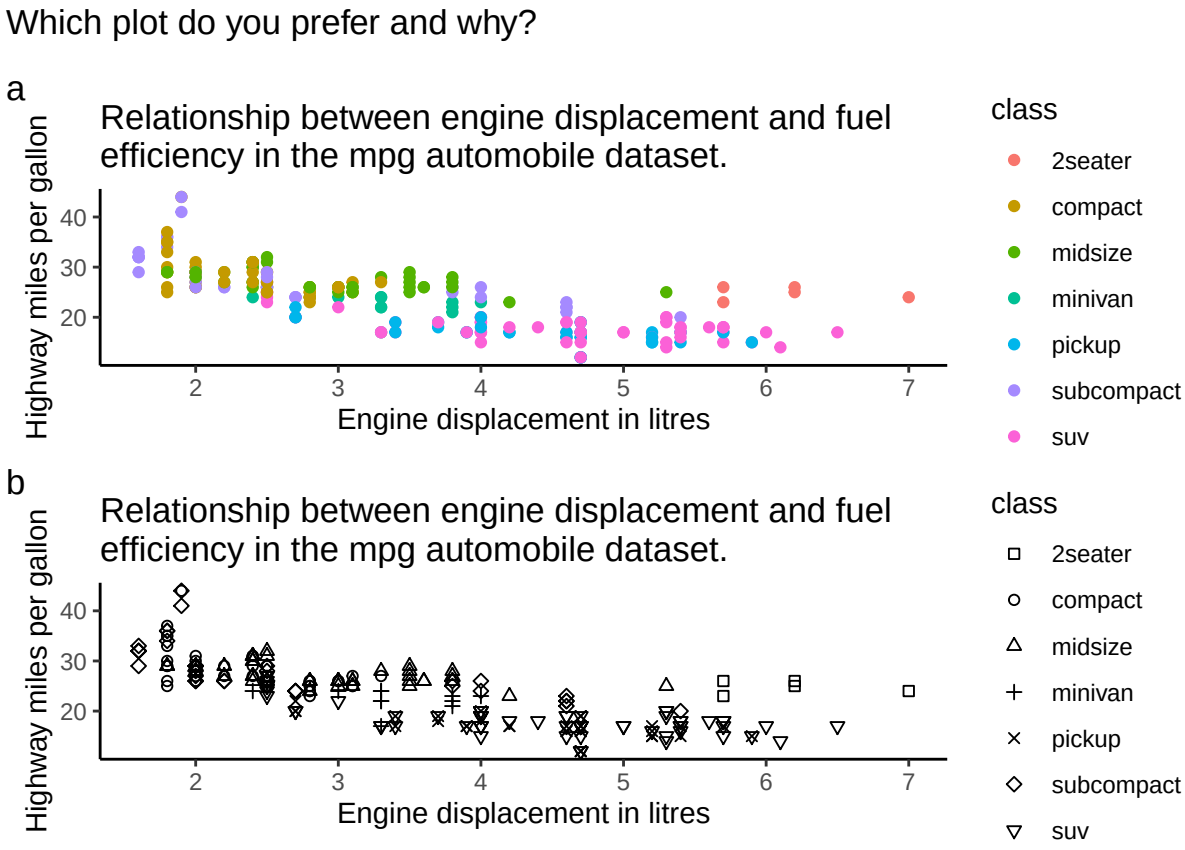
A limitation of this approach is that sometimes small variations can be practically significant and therefore important to highlight. If we compare global temperature over a very long period of time, minute variations in temperature may be obscured by the relative stability over time. A plot of global temperature starting at zero may therefore underemphasise the effect of global warming.

Question 5

Here are two plots:

```
#Store plot of mpg dataset using colours to differentiate car type
g1 <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, colour = class)) +
  labs(title = str_wrap("Relationship between engine displacement and fuel efficiency in the mpg automob
  xlab("Engine displacement in litres") +
  ylab("Highway miles per gallon")

#Store plot of mpg dataset using shapes to differentiate car type
g2 <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class)) +
  scale_shape_manual(values=c(0, 1, 2, 3, 4, 5, 6, 7)) +
```

```
  labs(title = str_wrap("Relationship between engine displacement and fuel efficiency in the mpg automob
  xlab("Engine displacement in litres") +
  ylab("Highway miles per gallon")

#Call plots with annotation and tags
g1 / g2 + plot_annotation(title = "Which plot do you prefer and why?", tag_levels = 'a')
```

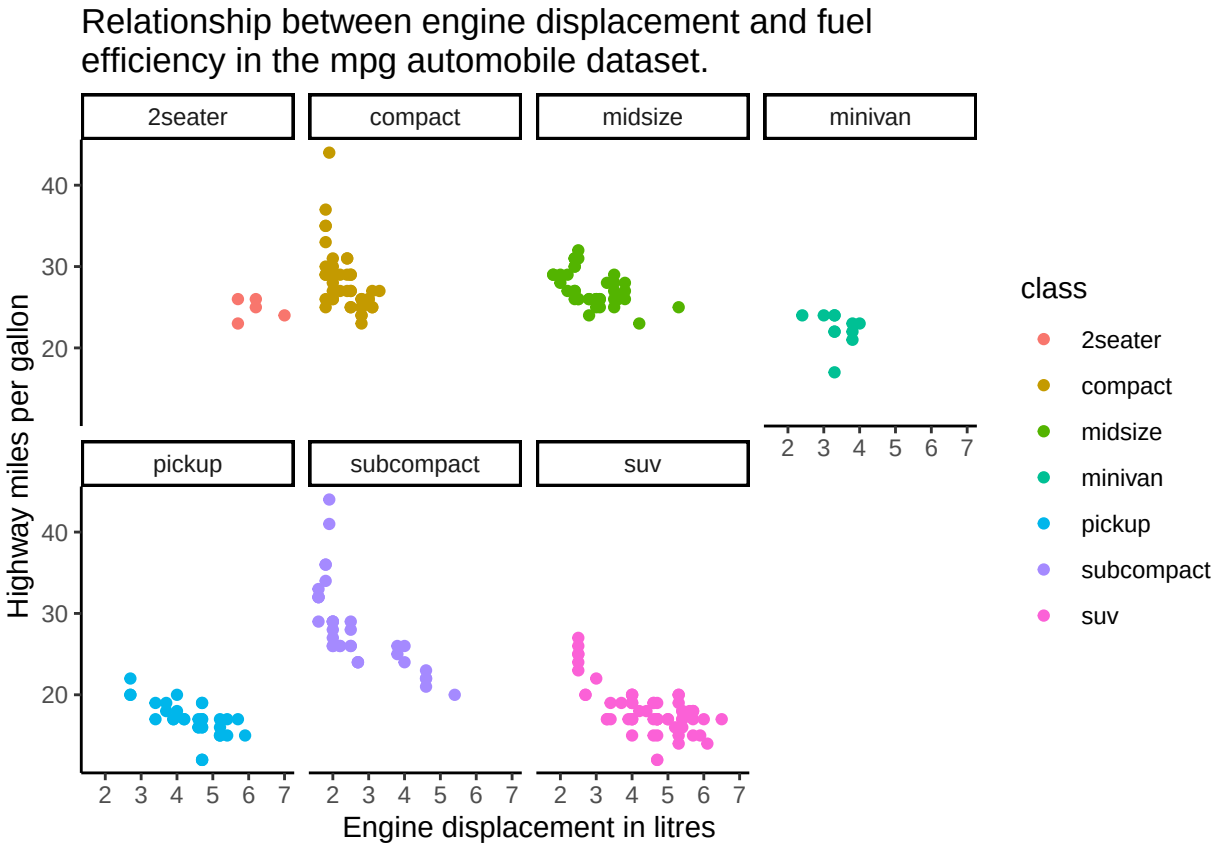


Scatterplot (a) classifies car type by colour, while scatterplot (b) classifies by shape. I find that colour is much easier to intuit, for the simple reason that colours are easier to distinguish than shapes.

Question 6

As previously described, we can add the `facet()` function to split our plot by a variable:

```
#Store plot of mpg dataset faceted by car type
g3 <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, colour = class)) +
  labs(title = str_wrap("Relationship between engine displacement and fuel efficiency in the mpg automob
  facet_wrap(~ class, nrow = 2) +
  xlab("Engine displacement in litres") +
  ylab("Highway miles per gallon")
#Call plot
g3
```

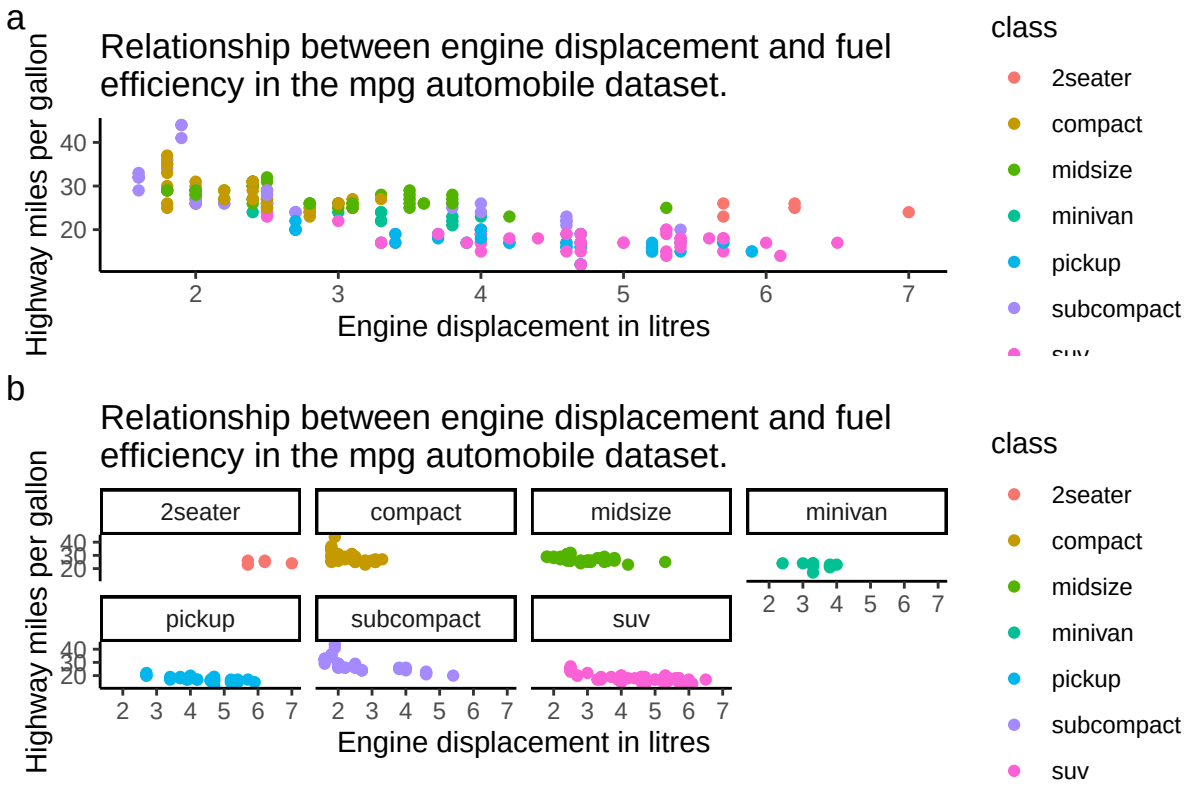


Question 7

Here are two plots:

```
#Call plots with annotation and tags
g1 / g3 + plot_annotation(title = "Which plot do you prefer and why?", tag_levels = 'a')
```

Which plot do you prefer and why?

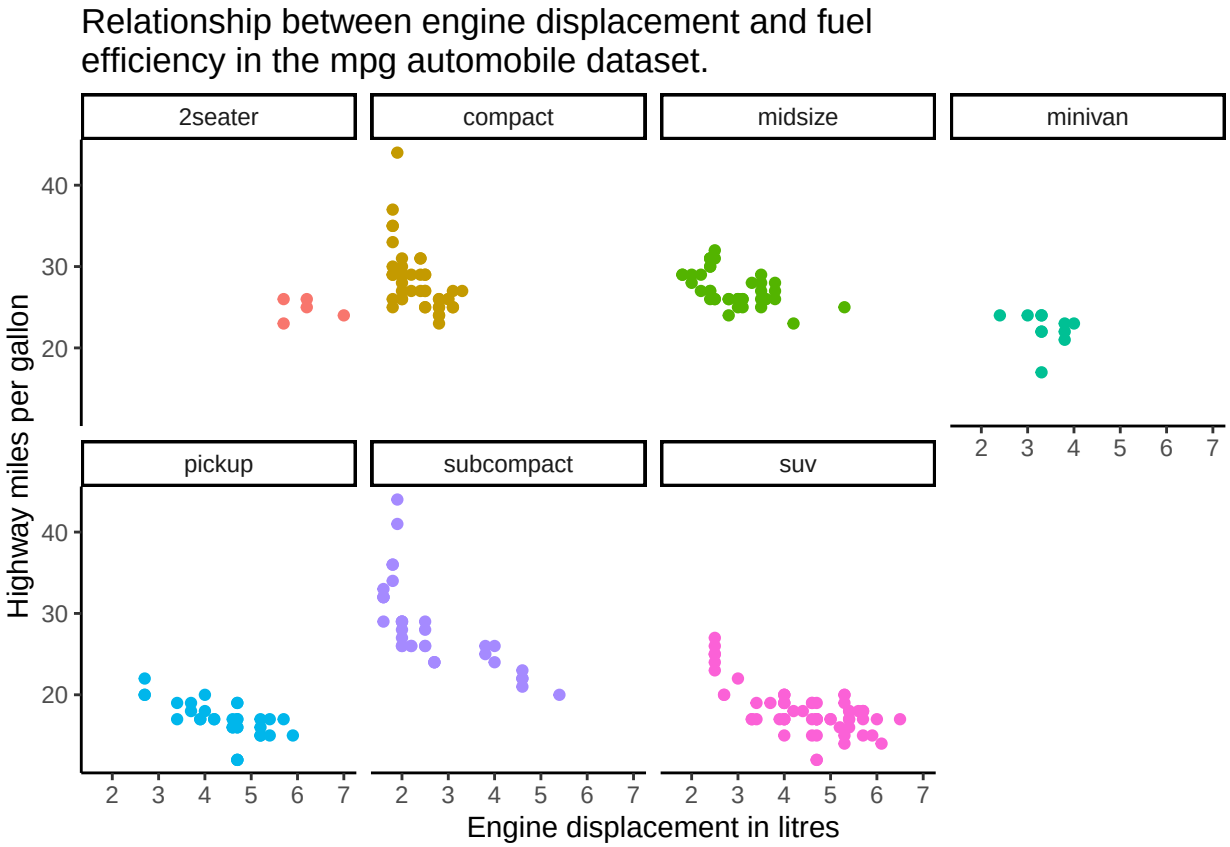


Scatterplot (a) displays all car types in one plot, while scatterplot (b) facets each car type into separate plots. I would perhaps use these graphs in different situations - if it were necessary to compare car types, I would use the non-faceted view, while if it were necessary to examine the shape of the data for each individual car type then the faceted view would be easiest.

Question 8

We can remove the legend by setting the legend.position to "none":

```
#Call plot and remove legend
g3 + theme(legend.position = "none")
```



Question 9

It is important to note which variables are listed in a dataset as factors or numerical values, as this affects how R will

```
#Read dataset and scope data
issp <- readr::read_csv2(url("https://raw.githubusercontent.com/go-bayes/psych-447/main/data/issp.csv"))
head(issp)
```

```
## # A tibble: 6 x 21
##      id   age male   eduyears nzeuro rightwing neg_ath   neg_bd  neg_ch  neg_hd
##   <dbl> <dbl> <chr>     <dbl>  <dbl>     <dbl> <chr>     <chr>   <chr>   <chr>
## 1     1    52 Not M~       16      1         4 Neither ~ Somewh~ Somewh~ Somewh~
## 2     1    53 Not M~       13      1         3 Neither ~ Neithe~ Neithe~ Somewh~
## 3     2    63 Not M~       10      1         4 Neither ~ Neithe~ Somewh~ Neithe~
## 4     2    64 Not M~       12      1         1 Neither ~ Neithe~ Somewh~ Neithe~
## 5     3    64 Male         NA      1         8 Neither ~ Neithe~ Neithe~ Neithe~
## 6     3    65 Male         16      1         7 Neither ~ Neithe~ Neithe~ Neithe~
## # ... with 11 more variables: neg_jw <chr>, neg_ms <chr>, thr_ath <dbl>,
## #   thr_bd <chr>, thr_ch <chr>, thr_hd <chr>, thr_jw <chr>, rural <chr>,
## #   thr_ms <chr>, wave <dbl>, religiosity <dbl>
```

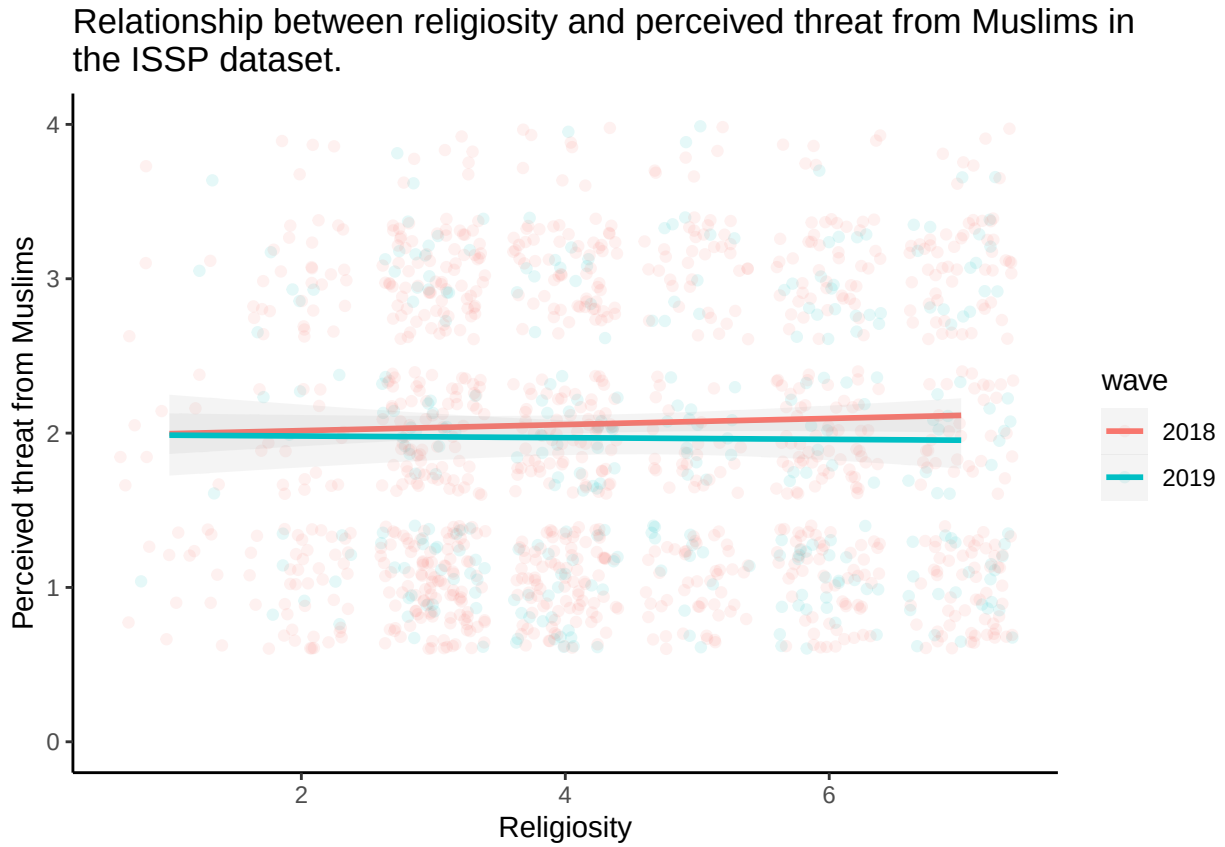
```
str(issp)
```

```
## spec_tbl_df [2,668 x 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id         : num [1:2668] 1 1 2 2 3 3 4 4 5 5 ...
##  $ age        : num [1:2668] 52 53 63 64 64 65 29 30 41 42 ...
```

```
##  $ male       : chr [1:2668] "Not Male" "Not Male" "Not Male" "Not Male" ...
##  $ eduyears   : num [1:2668] 16 13 10 12 NA 16 11 13 11 14 ...
##  $ nzeuro     : num [1:2668] 1 1 1 1 1 1 NA 1 1 1 ...
##  $ rightwing  : num [1:2668] 4 3 4 1 8 7 8 8 6 7 ...
##  $ neg_ath    : chr [1:2668] "Neither negative nor positive" "Neither negative nor positive" "Neithe
##  $ neg_bd     : chr [1:2668] "Somewhat negative" "Neither negative nor positive" "Neither negative n
##  $ neg_ch     : chr [1:2668] "Somewhat negative" "Neither negative nor positive" "Somewhat negative"
##  $ neg_hd     : chr [1:2668] "Somewhat negative" "Somewhat negative" "Neither negative nor positive"
##  $ neg_jw     : chr [1:2668] "Somewhat negative" "Neither negative nor positive" "Neither negative n
##  $ neg_ms     : chr [1:2668] "Somewhat negative" "Somewhat negative" "Somewhat negative" "Somewhat n
##  $ thr_ath    : num [1:2668] 2 2 1 1 2 2 1 1 3 NA ...
##  $ thr_bd     : chr [1:2668] "Not very threatening" "Somewhat threatening" "Not threatening at all"
##  $ thr_ch     : chr [1:2668] "Not very threatening" "Somewhat threatening" "Somewhat threatening" "S
##  $ thr_hd     : chr [1:2668] "Not very threatening" "Somewhat threatening" "Not threatening at all"
##  $ thr_jw     : chr [1:2668] "Not very threatening" "Somewhat threatening" "Not threatening at all"
##  $ rural      : chr [1:2668] "Not Rural" "Not Rural" "Not Rural" "Not Rural" ...
##  $ thr_ms     : chr [1:2668] "Not very threatening" "Somewhat threatening" "Somewhat threatening" "N
##  $ wave       : num [1:2668] 2018 2019 2018 2019 2018 ...
##  $ religiosity: num [1:2668] 6 4 3 4 4 6 7 7 5 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   age = col_double(),
##   ..   male = col_character(),
##   ..   eduyears = col_double(),
##   ..   nzeuro = col_double(),
##   ..   rightwing = col_double(),
##   ..   neg_ath = col_character(),
##   ..   neg_bd = col_character(),
##   ..   neg_ch = col_character(),
##   ..   neg_hd = col_character(),
##   ..   neg_jw = col_character(),
##   ..   neg_ms = col_character(),
##   ..   thr_ath = col_double(),
##   ..   thr_bd = col_character(),
##   ..   thr_ch = col_character(),
##   ..   thr_hd = col_character(),
##   ..   thr_jw = col_character(),
##   ..   rural = col_character(),
##   ..   thr_ms = col_character(),
##   ..   wave = col_double(),
##   ..   religiosity = col_double()
##   .. )
```

```r
#Re-code variables and save to new dataset
ip <- issp %>%
  mutate(
    id = factor(id),
    thr_ath = as.factor(thr_ath),
    thr_bd = as.factor(thr_bd),
    thr_ch = as.factor(thr_ch),
    thr_hd = as.factor(thr_hd),
    thr_jw = as.factor(thr_jw),
    thr_ms = as.factor(thr_ms),
```

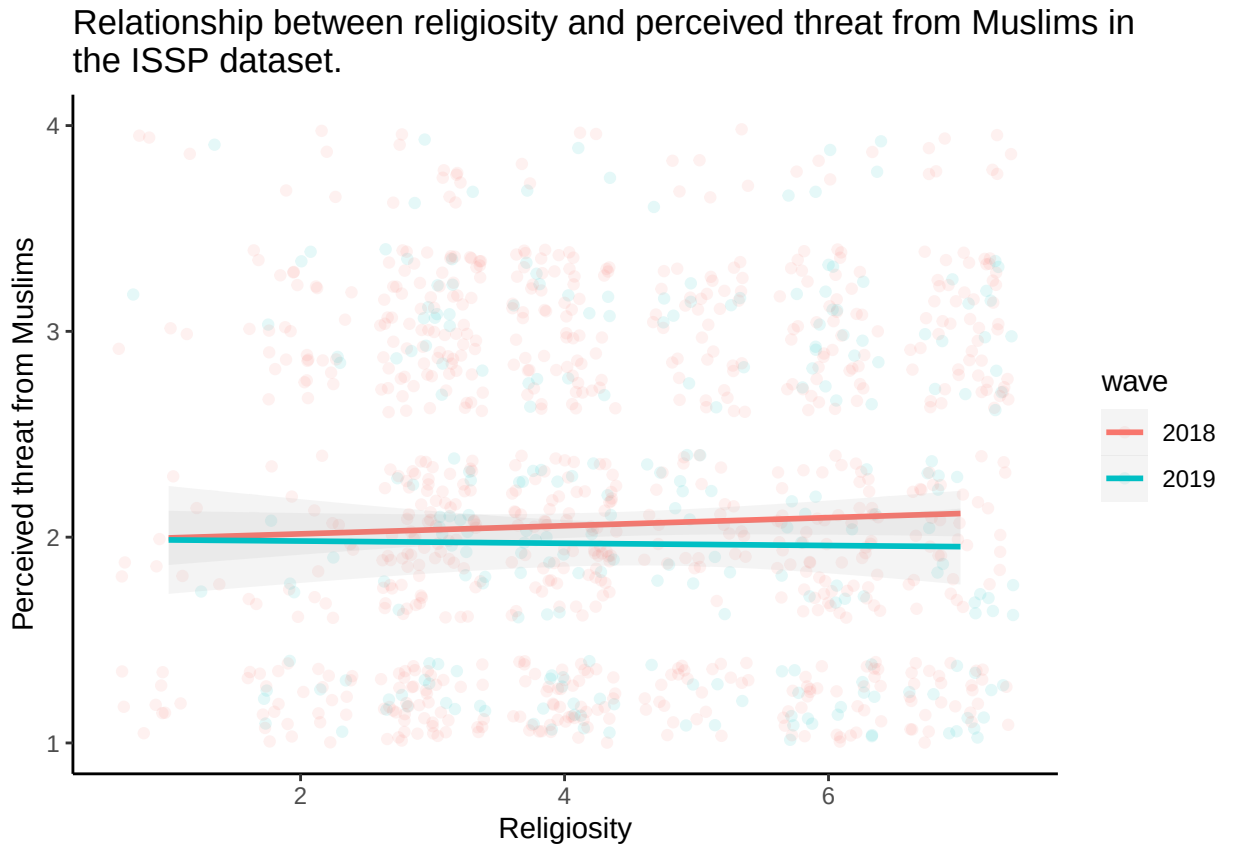
```
    neg_ath = as.factor(neg_ath),
    neg_bd = as.factor(neg_bd),
    neg_ch = as.factor(neg_ch),
    neg_hd  = as.factor(neg_hd),
    neg_jw = as.factor(neg_jw),
    neg_ms = as.factor(neg_ms),
    wave  = as.factor(wave),
    nzeuro = as.factor(nzeuro),
    eduyears = as.numeric(eduyears),
    male = as.factor(male),
    age = as.numeric(age),
    rightwing = as.numeric(rightwing),
    rural = as.factor(rural),
    religiosity = as.numeric(religiosity)
  )

#Store plot of ip dataset using jitter to distinguish frequency of Likert responses
g4 <- ggplot(data = ip, aes(y = as.numeric(thr_ms), x = religiosity, colour = wave))  +  geom_jitter(al
  geom_smooth(method = lm, fullrange = FALSE, alpha = 0.1) +
  labs(title = str_wrap("Relationship between religiosity and perceived threat from Muslims in the ISSP
  xlab("Religiosity") +
  ylab("Perceived threat from Muslims") +
  scale_y_continuous(limits = c(0,4))
g4
```



Question 10

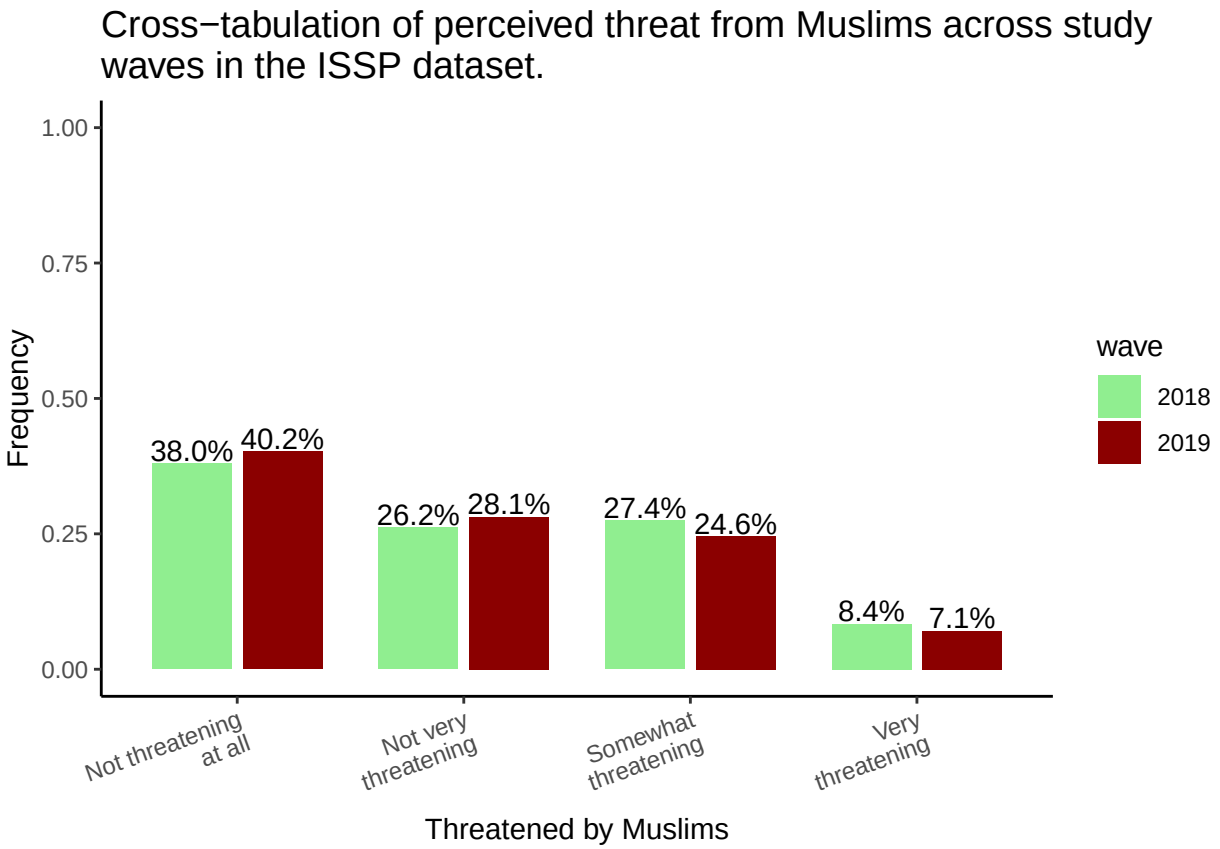
We can adjust the limits of the y axis using the `scale_y_continuous()` function.

```
#Adjust y-axis limits
g4 + scale_y_continuous(limits = c(1,4))
```



Question 11

```
#Plot cross-tabulation of ip dataset
plot_xtab(
    ip$thr_ms,
    ip$wave,
    show.total = FALSE,
    show.n = FALSE,
    geom.colors = c("lightgreen", "darkred")
    ) +
  labs(title = str_wrap("Cross-tabulation of perceived threat from Muslims across study waves in the ISS
  xlab("Threatened by Muslims") +
  ylab("Frequency") +
  scale_y_continuous(limits=c(0,1)) +
  theme(plot.title = element_text(size=14), axis.text.x = element_text(angle = 20, hjust = 1)
)
```



The original graph has been amended above as follows:

- Added title.
- Separated axis label code onto different lines.
- Removed hashtag from `scale_y_continuous()` and `theme()` functions to display as code, not comment.
- Resized y axis to reflect that scale of cross-tab plot cannot exceed 1.
- Combined plot title and axis text into a single `theme()` function.
- Resized title to match other plots.
- Separated final end parenthesis onto different line.