

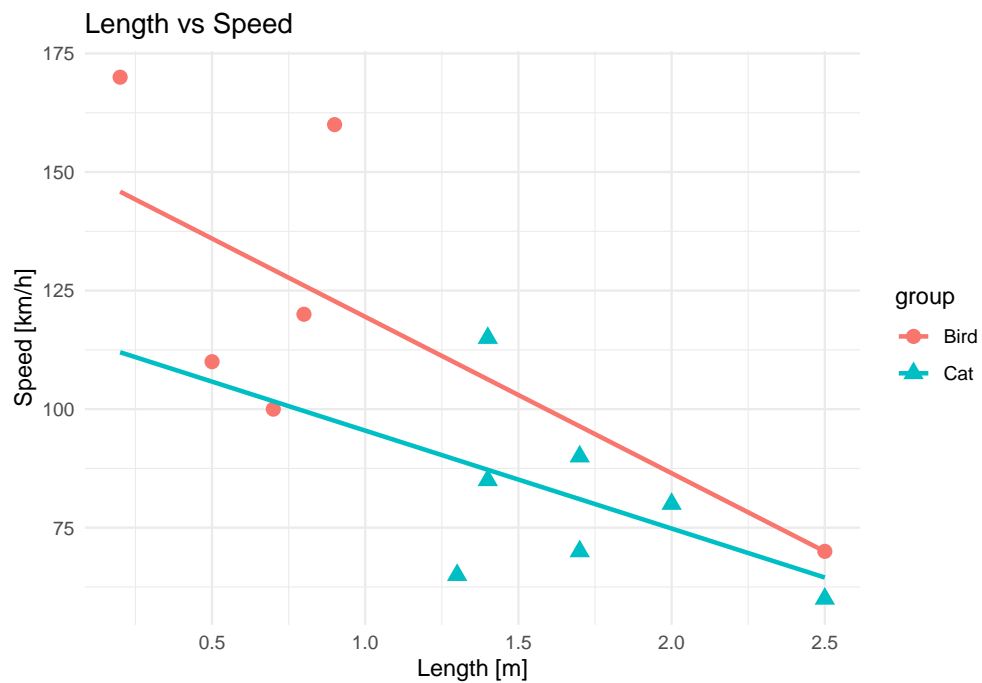
# ggplot2

Jędrzej Rybczyński

7/28/2021

1. Dla zbioru danych `cats_birds` przedstaw zależność pomiędzy zmiennymi `length` oraz `speed` na wykresie rozrzutu. Dodaj trend liniowy do wykresu. Obie grupy zwierząt powinny mieć własną prostą regresji oraz typ i kolor punktów.

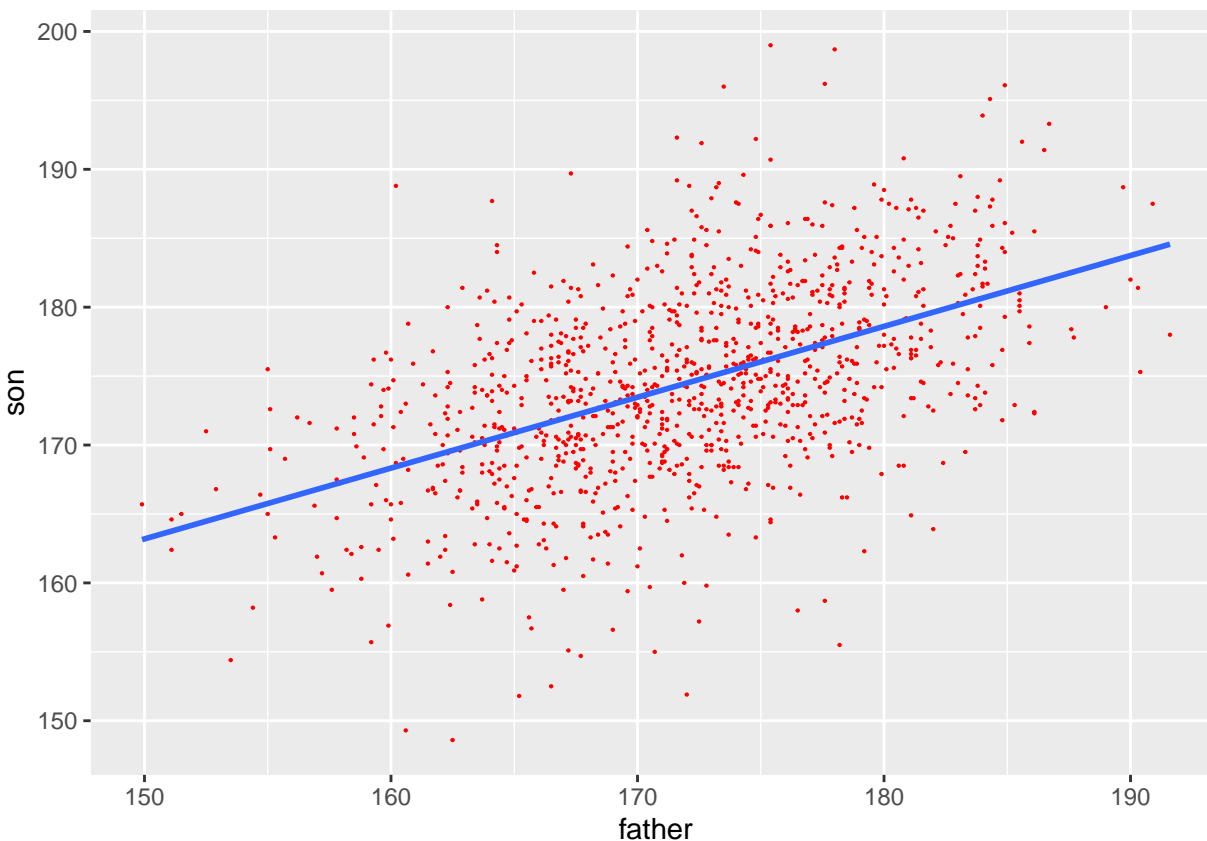
```
ggplot(cats_birds, aes(x = length,
                        y = speed,
                        shape = group,
                        color = group)) +
  geom_point(size = 3) +
  geom_smooth(method = 'lm',
              se = FALSE,
              fullrange = TRUE) +
  ggtitle('Length vs Speed') +
  xlab('Length [m]') +
  ylab('Speed [km/h]') +
  theme_minimal()
```



2. W zbiorze danych pearson mamy informacje o wysokości ojców i synów. Przedstaw zależność pomiędzy tymi cechami na wykresie rozrzutu. Dodaj trend liniowy bez przedziału ufności.

```
pearson %>%  
  ggplot(aes(x = father, y = son)) +  
  geom_point(size = 0.1, col = 'red') +  
  geom_smooth(se = FALSE, method = 'lm')
```

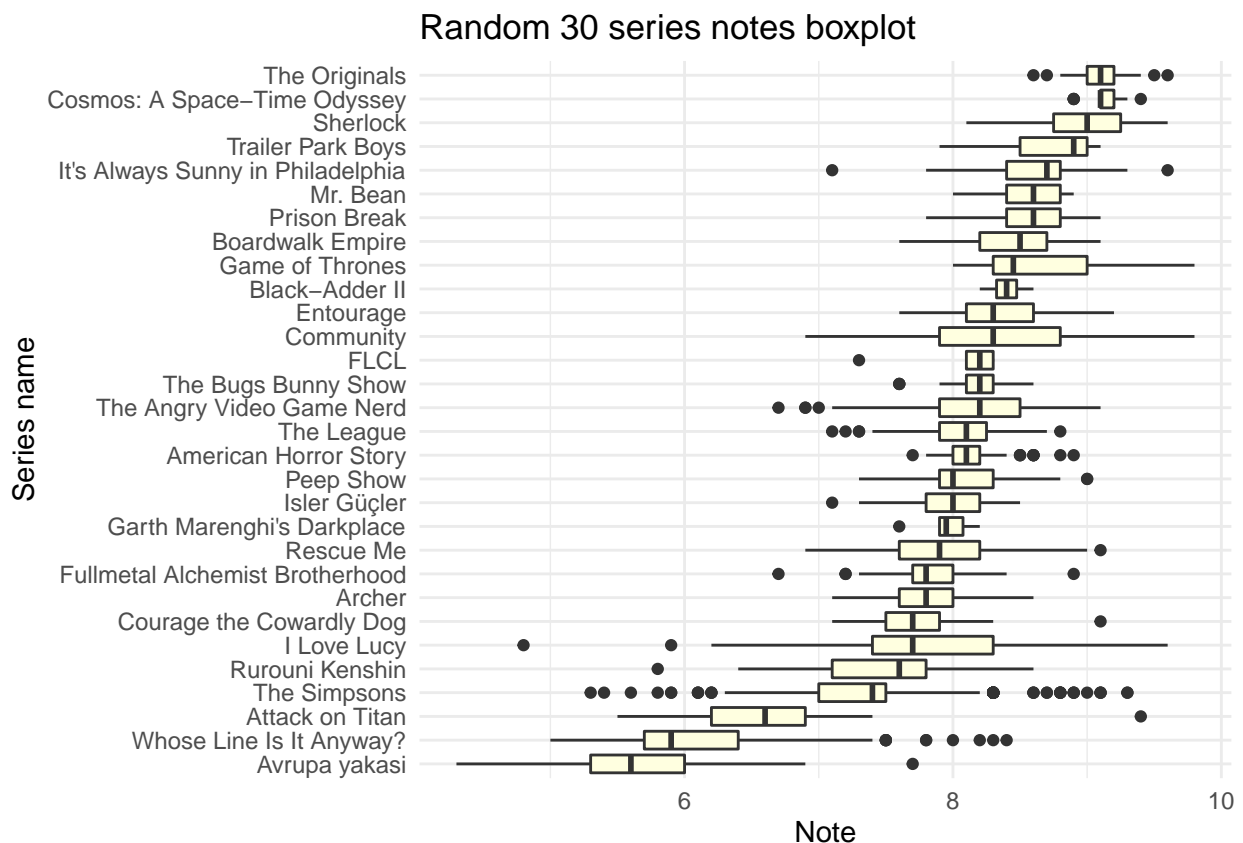
```
## `geom_smooth()` using formula 'y ~ x'
```



3. W zbiorze danych seriesIMDB mamy informacje o odcinkach około 200 seriali. Stwórz wykres, w którym na osi Y będzie nazwa serialu, a na osi X wykres pudełkowy ocen odcinków danego serialu. Popraw wykres tak, aby serie były posortowane według median ocen.

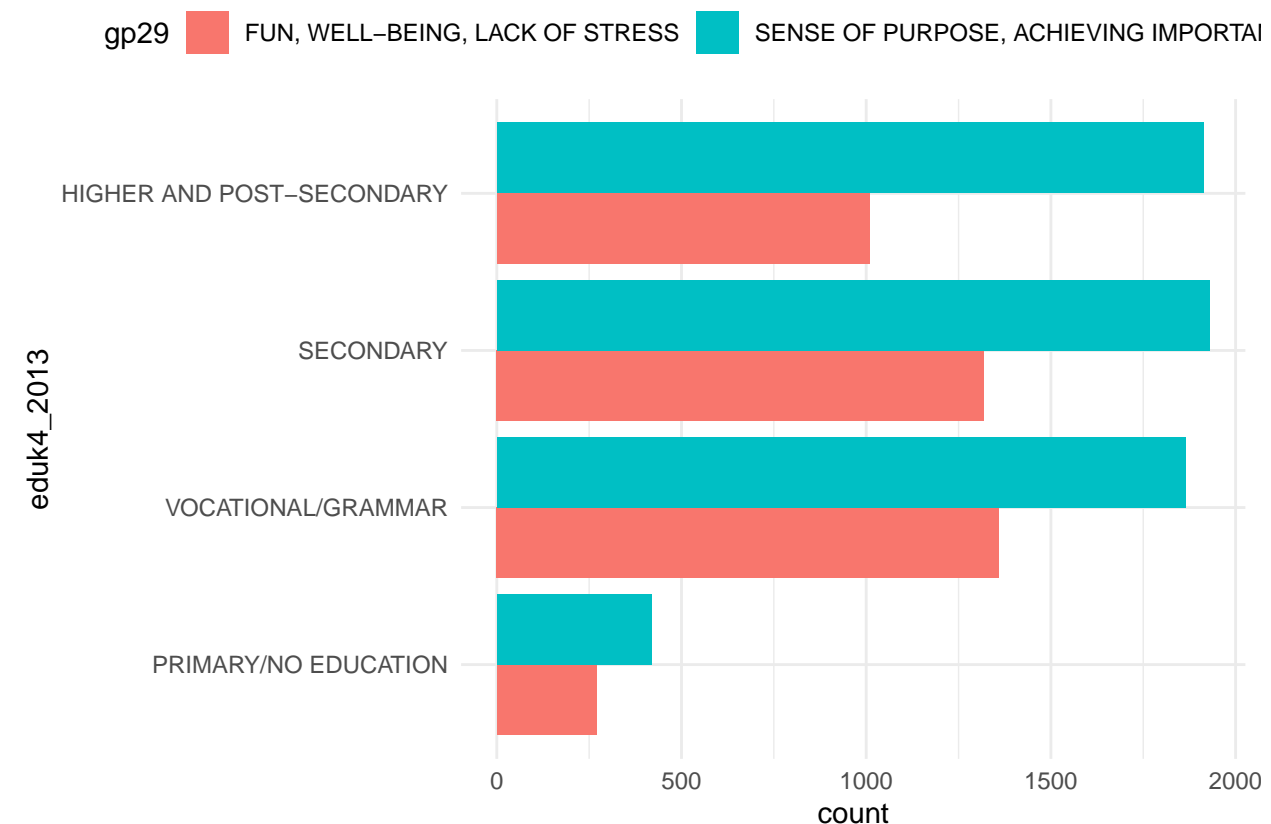
```
n <- 30
set.seed(100)
series.sample.names <- sample(unique(seriesIMDB$series), n)
seriesIMDB %>%
  filter(series %in% series.sample.names) -> df

df %>%
  mutate(series = reorder(series, note, median)) %>% # !!!
  ggplot(aes(x = series, y = note)) + # albo ggplot(df, ...)
  geom_boxplot(fill = 'lightyellow') +
  coord_flip() +
  ggtitle('Random 30 series notes boxplot') +
  xlab('Series name') +
  ylab('Note') +
  theme_minimal()
```



4. W zbiorze danych diagnosis mamy zmienną eduk4\_2013 z poziomami edukacji i zmienną gp29 z odpowiedziami na pytanie co jest najważniejsze w życiu. Stwórz wykres słupkowy. Każdy słupek odpowiada jednemu poziomowi edukacji. Każdy słupek powinien być podzielony na części wypełnione różnymi kolorami w zależności od zmiennej gp29.

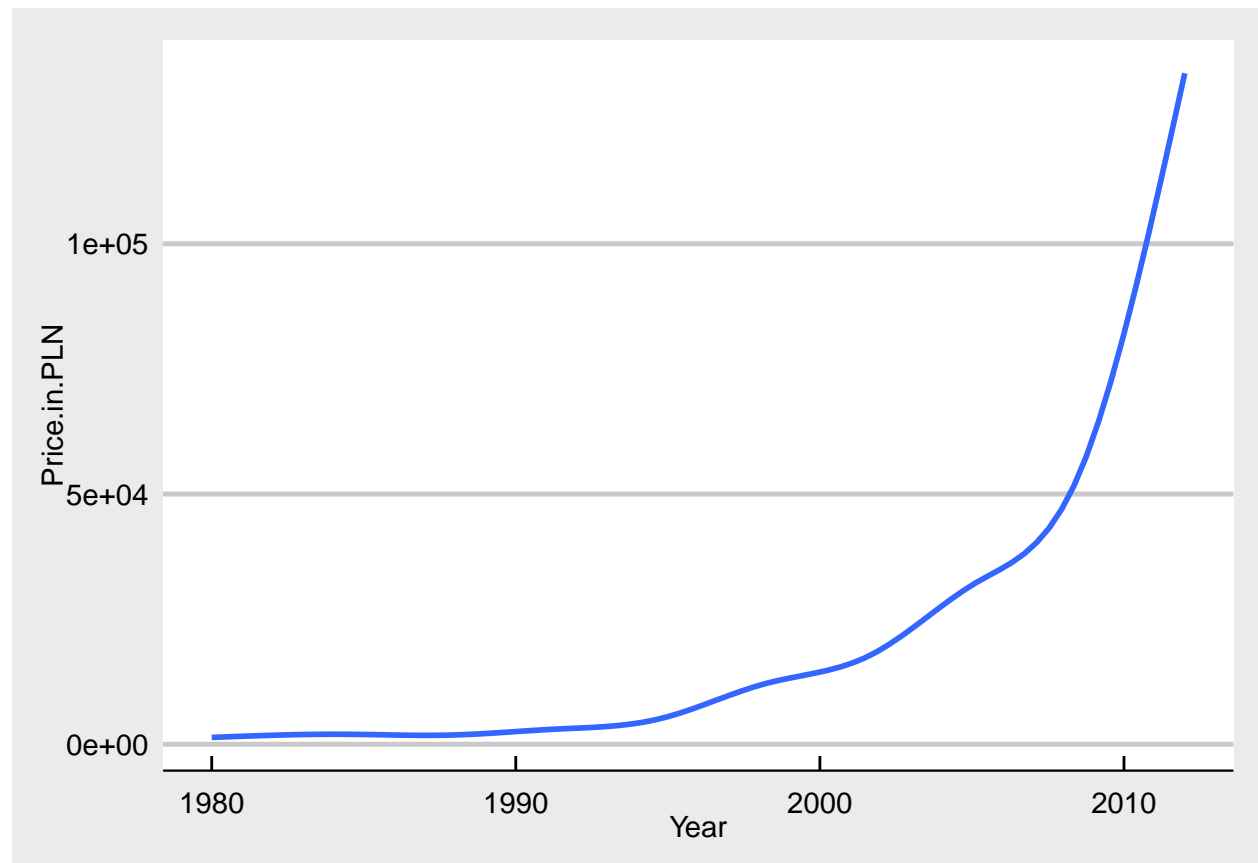
```
diagnosis %>%
  na.omit() %>%
  as_tibble() %>%
  select(eduk4_2013, gp29) %>%
  ggplot(aes(x = eduk4_2013, fill = gp29)) +
  geom_bar(position = 'dodge') +
  theme_minimal() +
  theme(legend.position = 'top') +
  coord_flip()
```



5. Ze zbioru `auta2012` wybierz jedynie auta Volkswagen Passat. Użyj funkcji `geom_smooth()`, aby przedstawić zależność pomiędzy rokiem produkcji i ceną. Spróbuj zastosować różne szablony wykresu.

```
auta2012 %>%  
  filter(Model == 'Passat', Brand == 'Volkswagen') %>%  
  as_tibble() %>%  
  ggplot(aes(x = Year, y = Price.in.PLN)) +  
  geom_smooth(se = FALSE) +  
  theme_economist_white()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

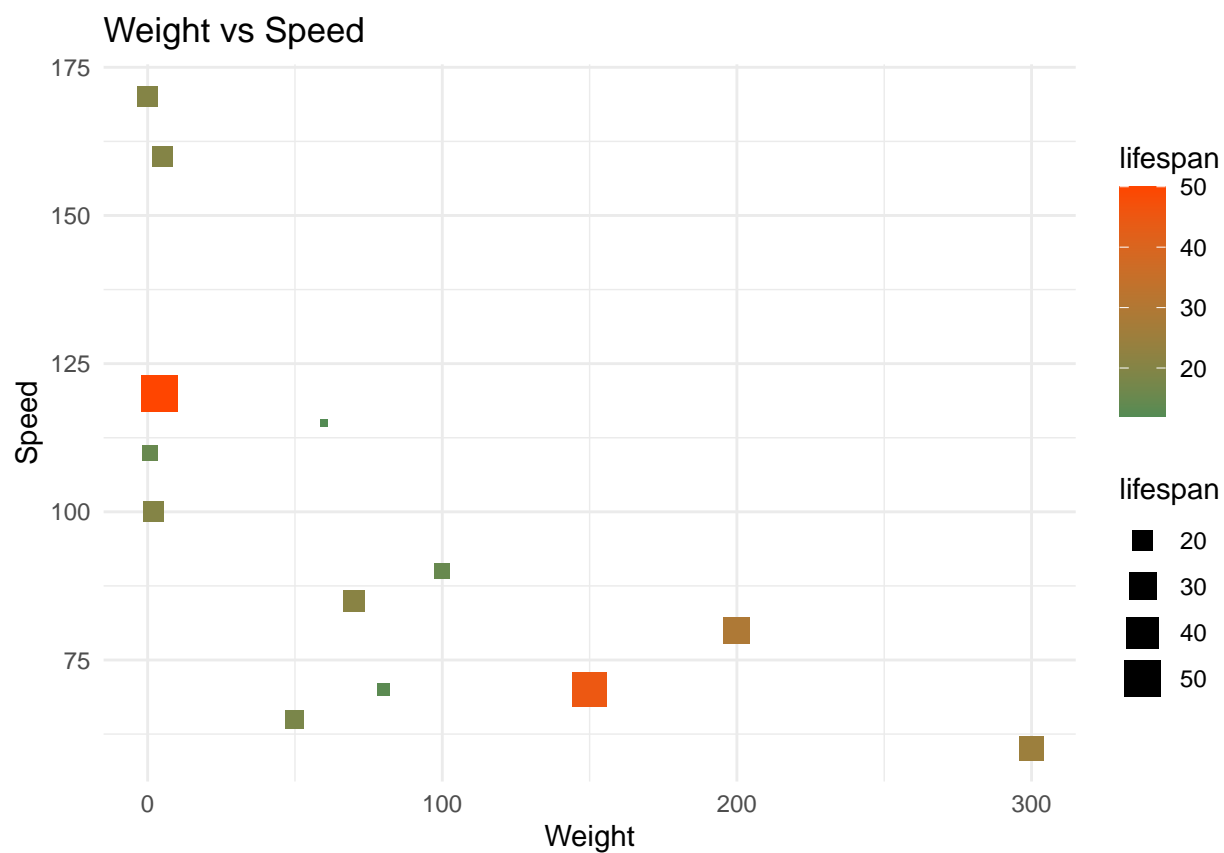


## 6. Zmień poniższy wykres:

```
ggplot(cats_birds, aes(x = weight, y = speed, size = lifespan, color = lifespan)) +  
  geom_point()
```

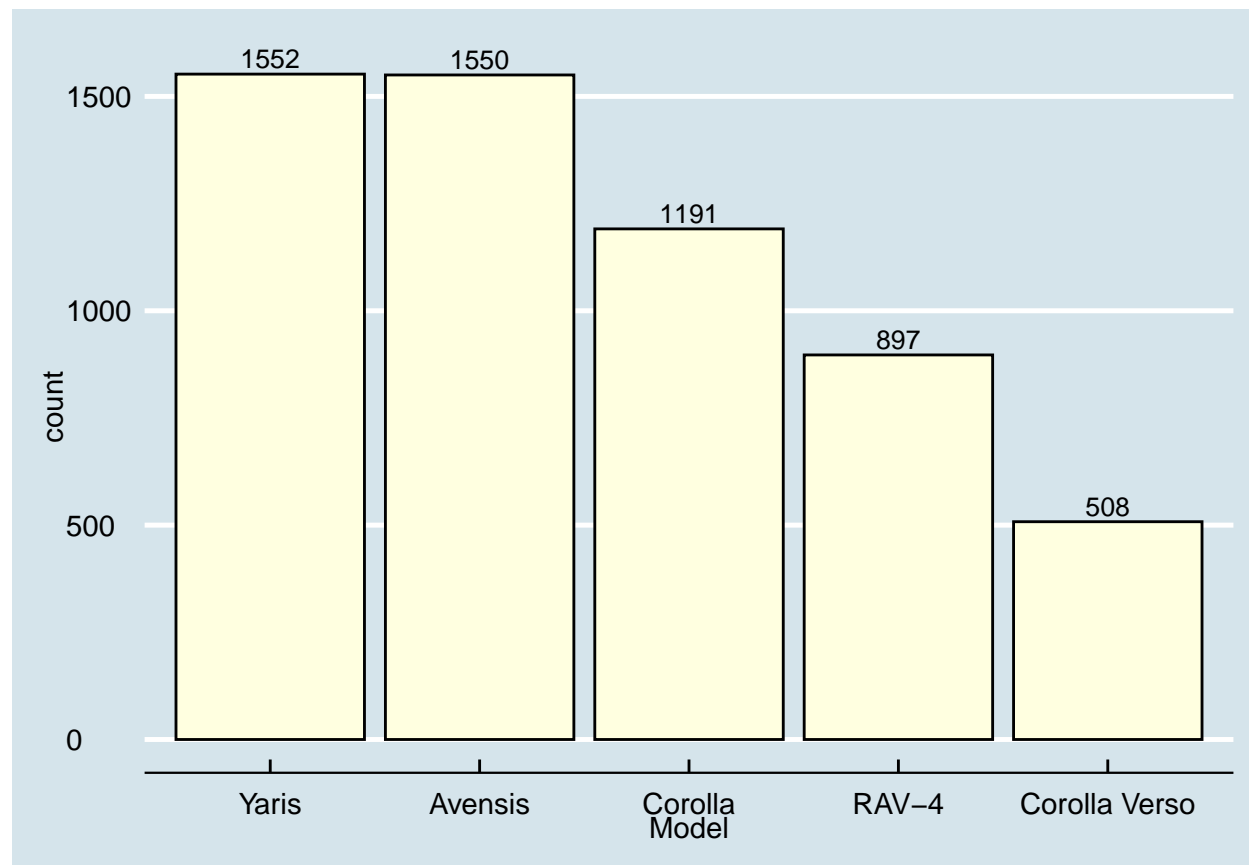
Zastąp skalę kolorów na gradient od koloru zielonego do czerwonego i zastąp kropki kwadratami. Dodaj tytuł i etykiety osi.

```
ggplot(cats_birds,  
  aes(x = weight,  
    y = speed,  
    size = lifespan,  
    color = lifespan)) +  
  geom_point(shape = 15) +  
  scale_color_gradient(low = 'palegreen4', high = 'orangered') +  
  labs(x = 'Weight', y = 'Speed', title = 'Weight vs Speed') +  
  theme_minimal()
```



7. Ze zbioru danych `auta2012` wybierz wszystkie samochody marki Toyota. Na wykresie słupkowym zwizualizuj liczbę pięciu najpopularniejszych modeli w kolejności malejących liczebności. Nie zapomnij o legendzie.

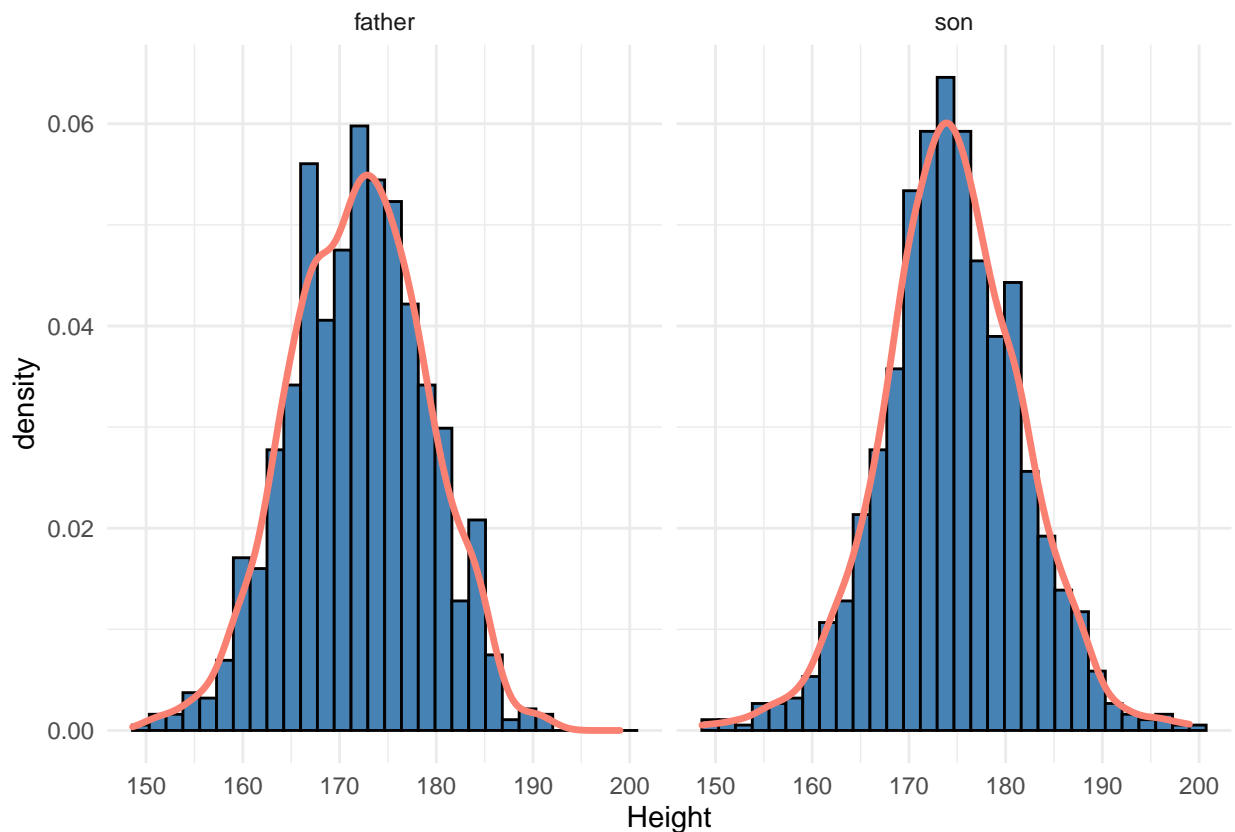
```
auta2012 %>%  
  filter(Brand == 'Toyota') %>%  
  as_tibble() -> df  
  
df %>%  
  count(Model) %>%  
  arrange(-n) %>%  
  slice(1:5) %>%  
  pull(Model) %>%  
  as.character() -> popular.models  
  
df %>%  
  filter(Model %in% popular.models) %>%  
  mutate(Model = factor(Model, levels = popular.models, ordered = TRUE)) %>%  
  ggplot(aes(x = Model)) +  
  geom_bar(fill = 'lightyellow', col = 'black') +  
  stat_count(aes(y = ..count.., label = ..count..), geom="text", vjust = -0.3, size = 3.5) +  
  theme_economist()
```



8. Dla wzrostu synów i ojców ze zbioru danych pearson narysuj na jednym wykresie histogram wraz z nieparametryczną oceną gęstości dla każdej cechy. Linie pomiędzy słupkami histogramu pokoloruj na białe. Oba wykresy powinny być obok siebie na jednym diagramie.

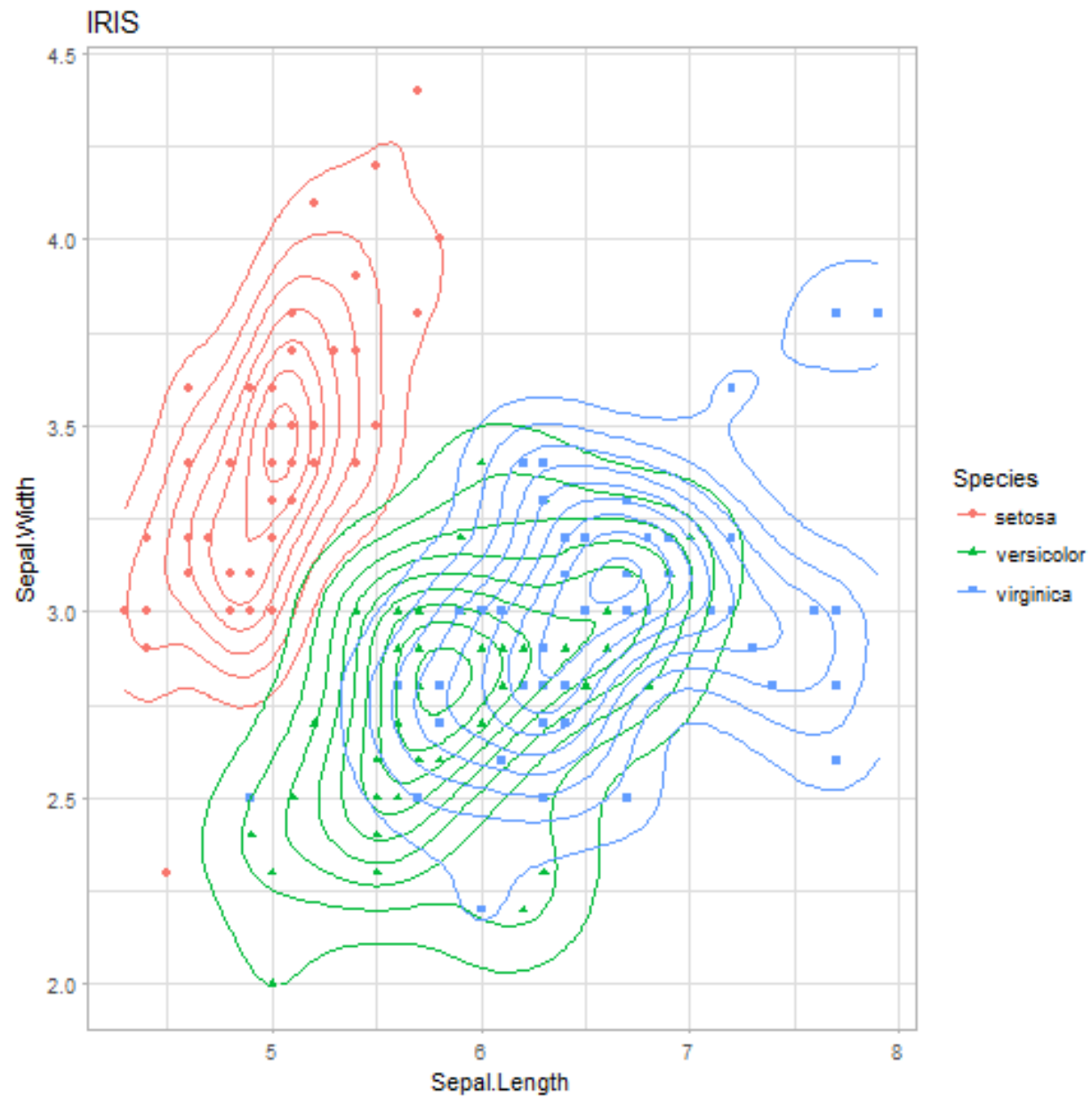
```
tidyr::pivot_longer(pearson,
  cols = everything(),
  names_to = 'F_S',
  values_to = 'Height') %>%
  ggplot(aes(x = Height)) +
  geom_histogram(aes(y= ..density..), col = 'black', fill='steelblue') +
  geom_density(col = 'salmon', size = 1.2) +
  facet_wrap('F_S', ncol = 2) +
  theme_minimal()
```

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

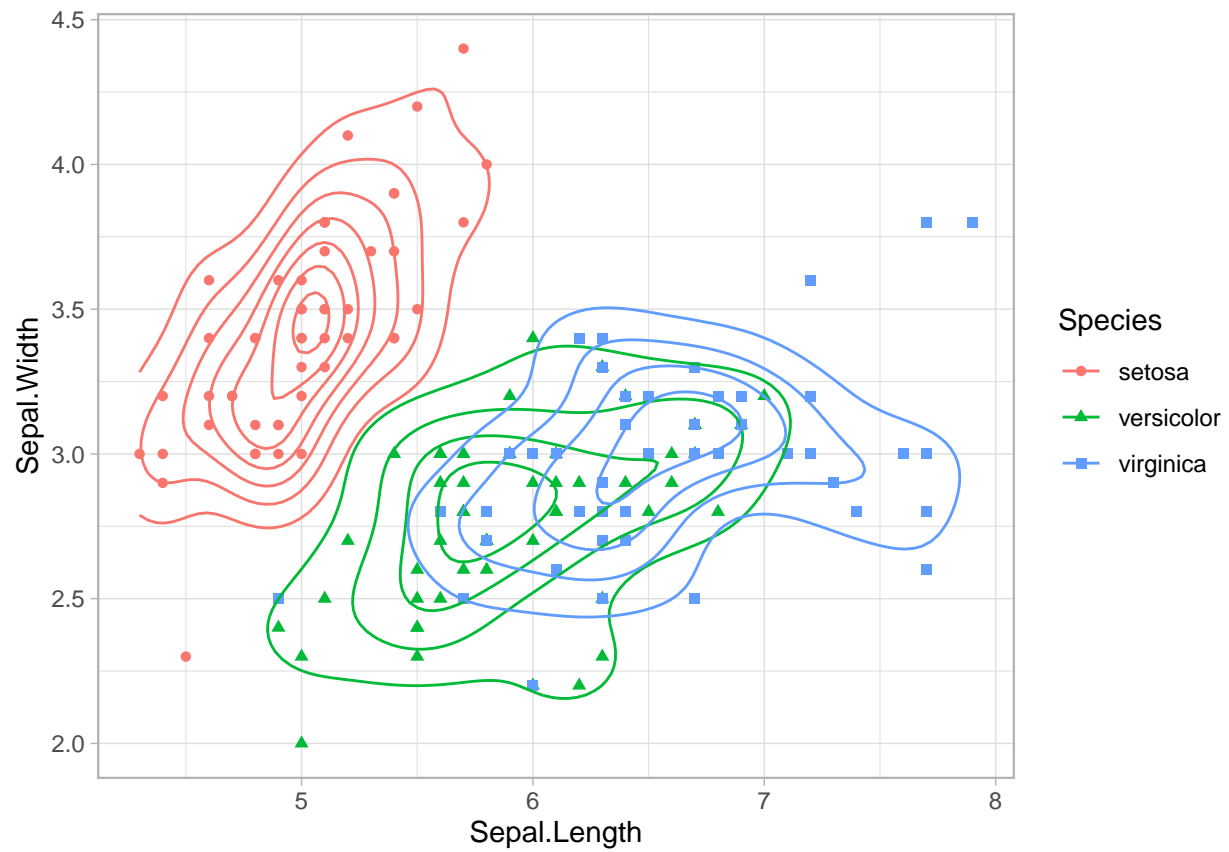




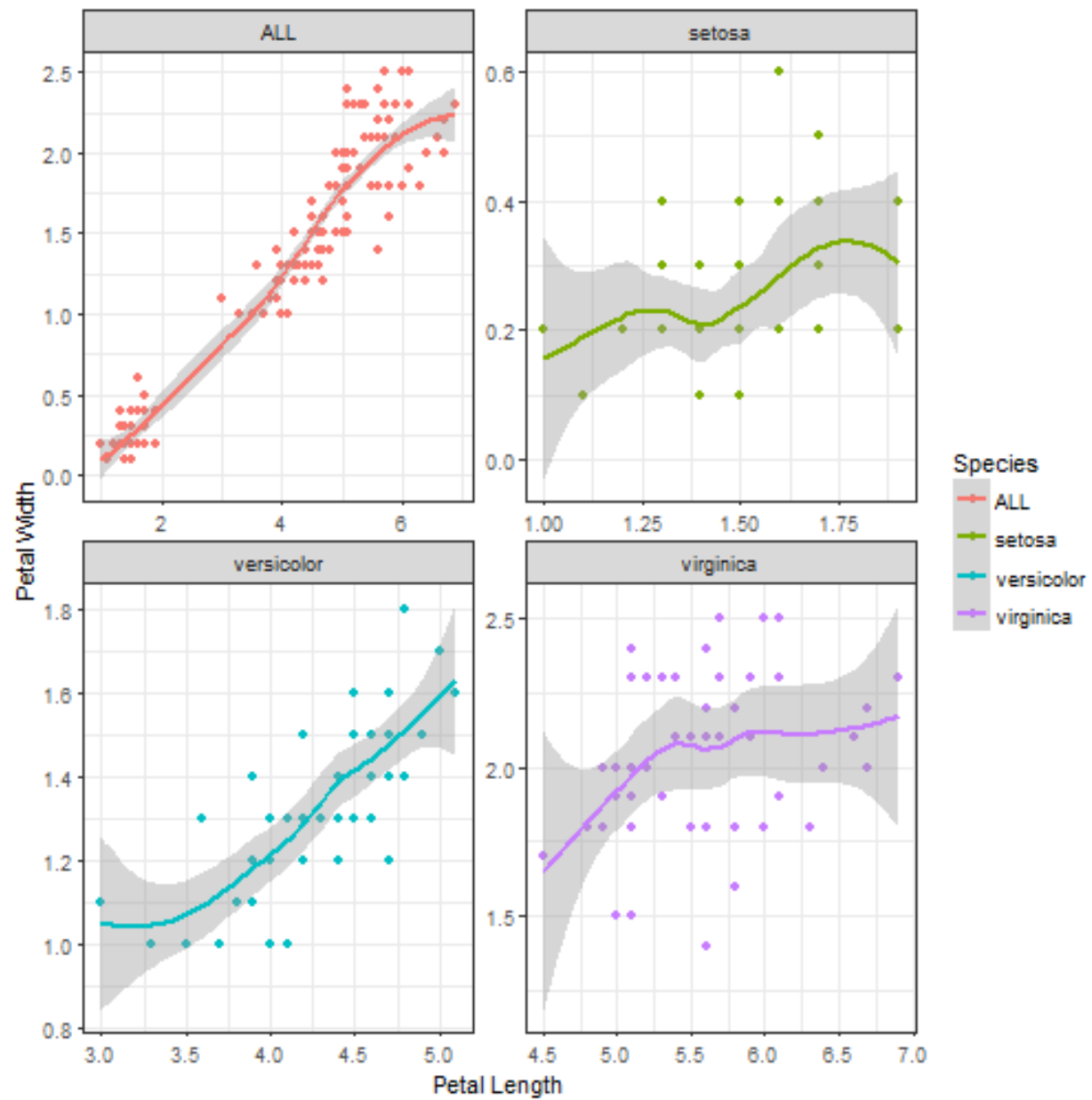
9. Dla zbioru danych iris przygotuj obrazek jak na Rysunku 1. Rozważ użycie funkcji `geom_density2d()` oraz `theme_light()`.



```
iris %>%
  ggplot(aes(x = Sepal.Length,
             y = Sepal.Width,
             col = Species,
             shape = Species)) +
  geom_point() +
  geom_density2d() +
  theme_light()
```

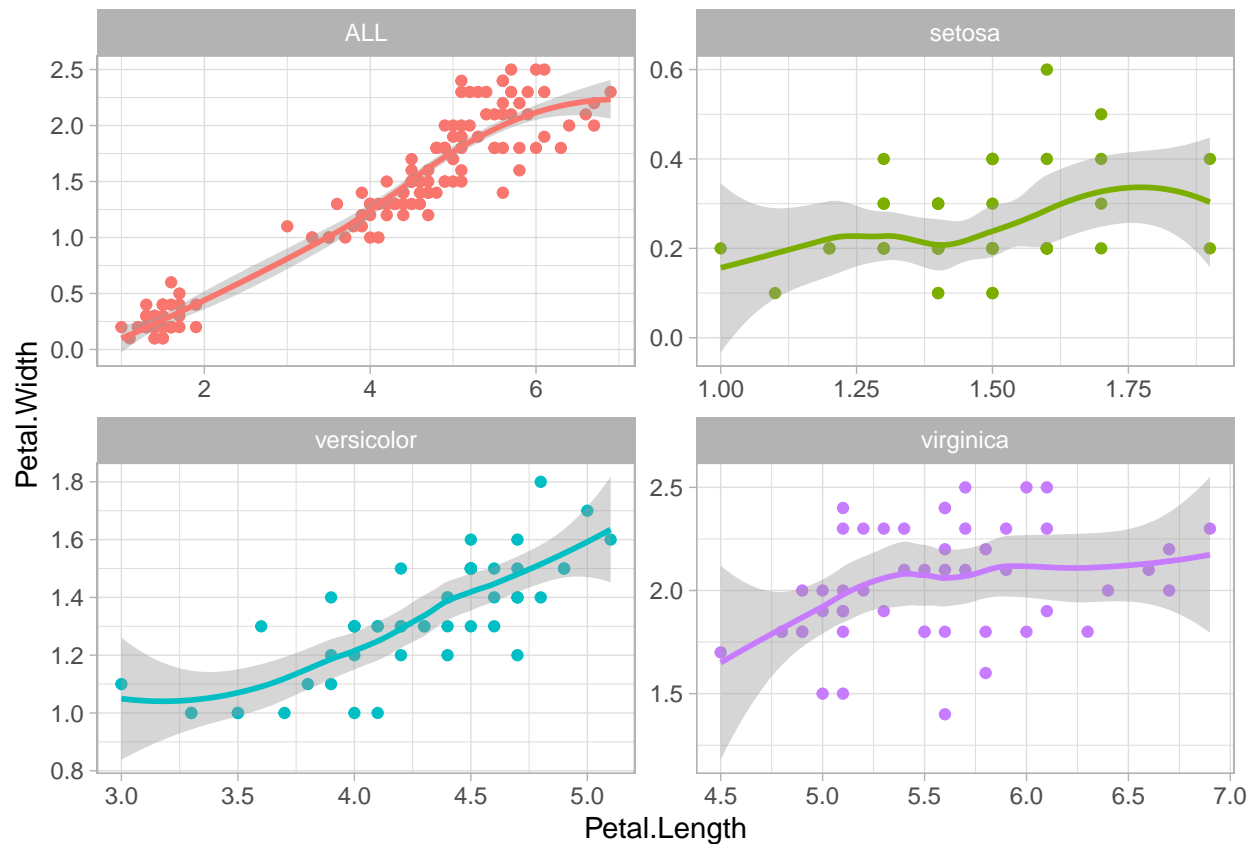


10. Dla zbioru danych iris przygotuj obrazek jak na Rysunku 2.

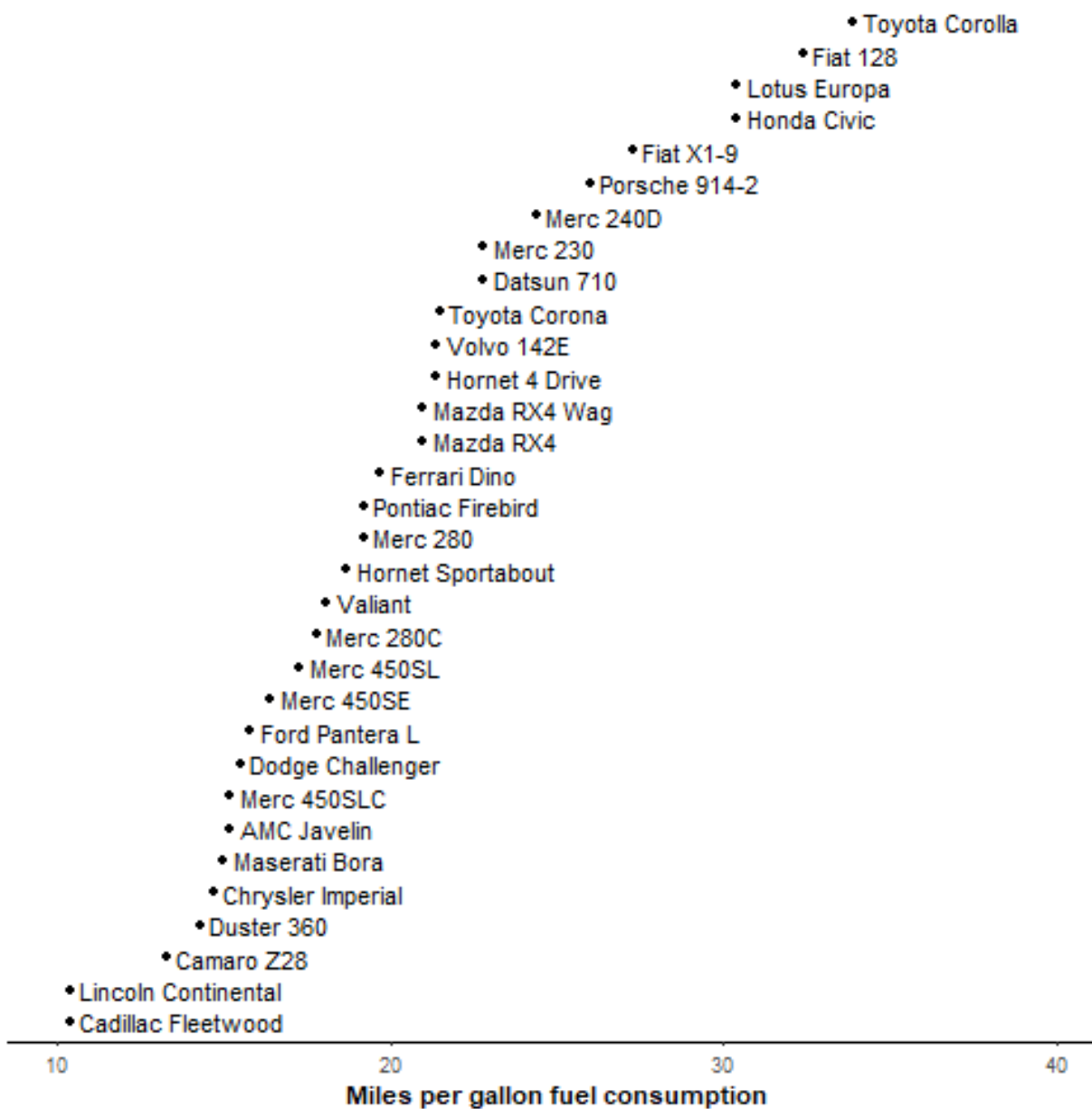


```
iris %>%
  as_tibble() %>%
  mutate(Species = 'ALL') %>%
  bind_rows(iris) %>% # dokleja wszystkie oryginalne wiersze, duplikuje razy 2
  ggplot(aes(x = Petal.Length, y = Petal.Width, col = Species)) +
  geom_point() +
  geom_smooth() +
  theme_light() +
  guides(color = FALSE) +
  facet_wrap('Species', ncol = 2, nrow = 2, scales = 'free')
```

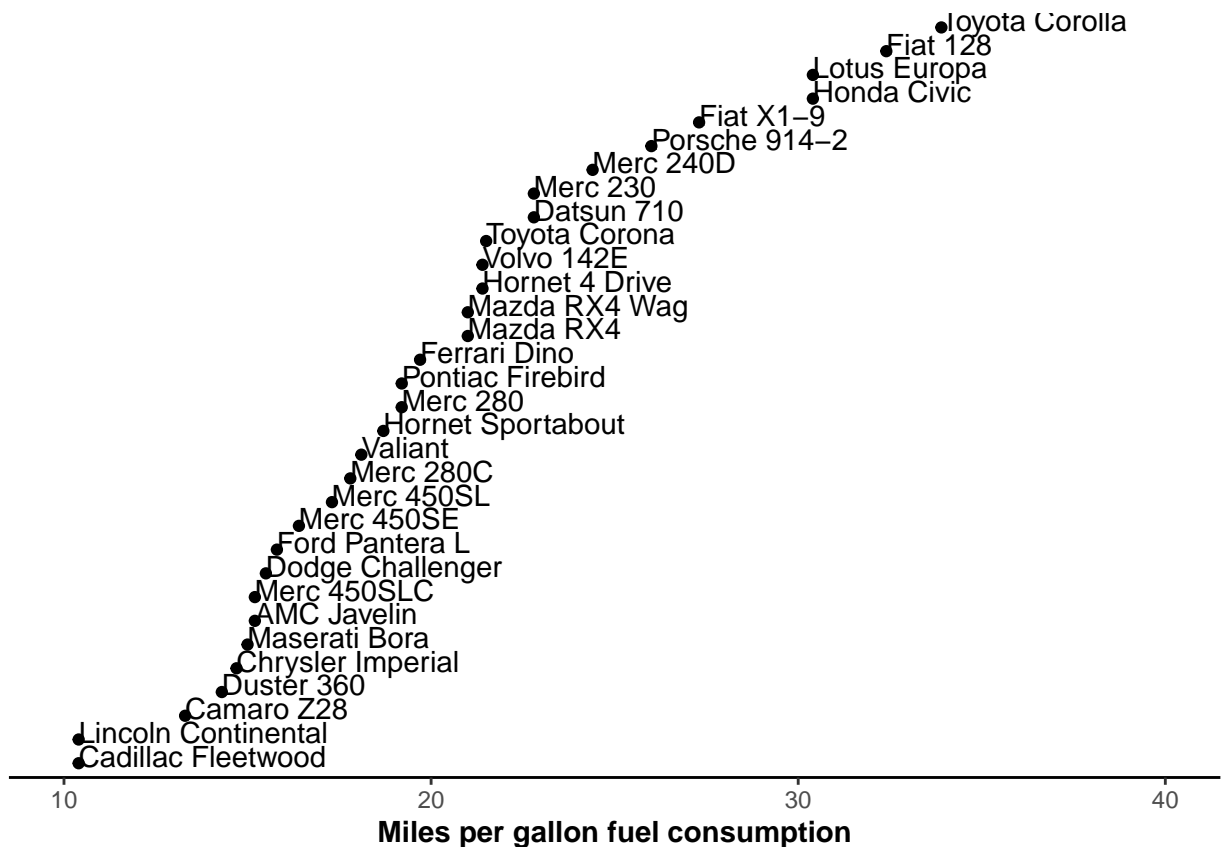
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



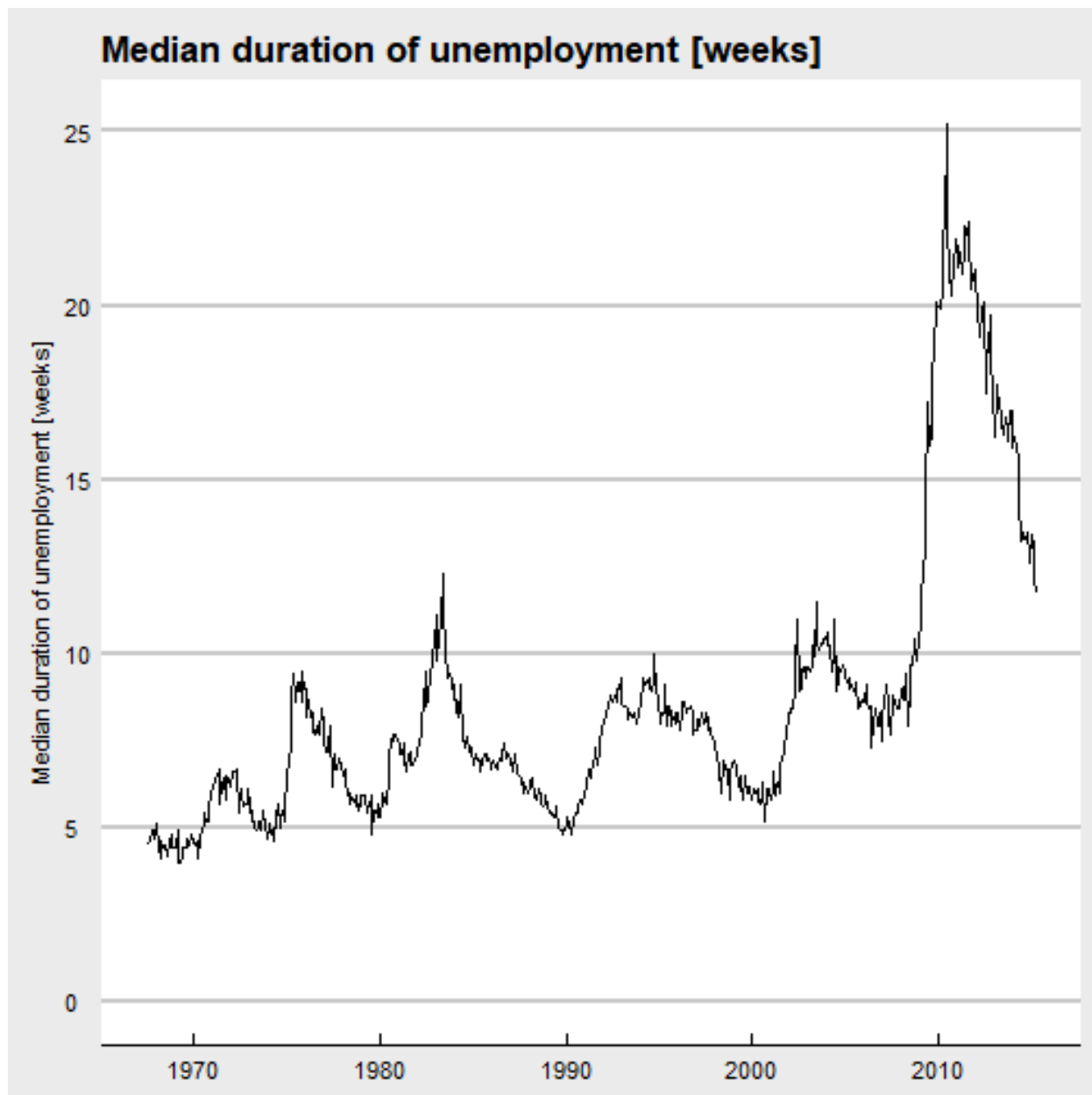
11. Dla zbioru danych mtcars przygotuj obrazek jak na Rysunku 3.



```
mtcars %>%
  as_tibble(rownames = 'Car') %>%
  mutate(Car = reorder(Car, mpg, sum)) %>% # moze byc bez sum
  ggplot(aes(x = mpg, y = Car, label = Car)) +
  geom_point() +
  geom_text(nudge_y = .3, hjust = 'left') +
  xlab('Miles per gallon fuel consumption') +
  ylab('') +
  xlim(10, 40) +
  theme_classic() +
  theme(axis.title.x = element_text(face = 'bold'),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.line.y = element_blank())
```



12. Dla zbioru danych `economics` z pakietu `ggplot2` przygotuj obrazek jak na Rysunku 4. Wykorzystaj szablon `theme_economist_white()` z pakietu `ggthemes`.



```
economics %>%
  ggplot(aes(x = date, y = uempmed)) +
  geom_line() +
  ylab('Median duration of unemployment (weeks)') +
  xlab('') +
  ggtitle('Median duration of unemployment (weeks)') +
  theme_economist_white()
```

