

Data Visualization in R with **ggplot2**

The Structure of ggplot2 (Part 3)

Cédric Scherer

Physalia Courses | November 9-13 2020

Photo by Richard Strozyński

Part 3

Multipanel Plots

ggplot2: Build a data MASTERpiece



Illustration by Allison Horst (github.com/allisonhorst/stats-illustrations)

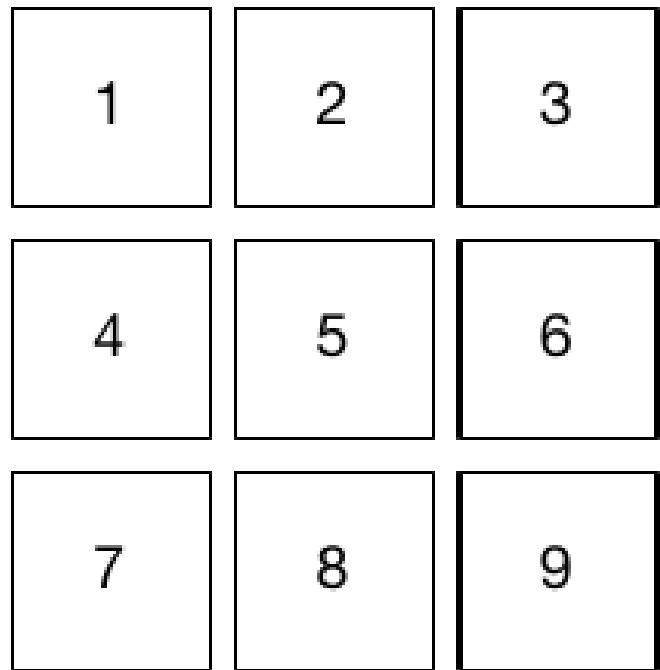
The Structure of `ggplot2`

Layer	Function	Explanation
Data	<code>ggplot(data)</code>	The raw data that you want to visualise.
Aesthetics	<code>aes()</code>	Aesthetic mappings of the geometric and statistical objects.
Layers	<code>geom_*</code> () and <code>stat_*</code> ()	The geometric shapes and statistical summaries representing the data.
Scales	<code>scale_*</code> ()	Maps between the data and the aesthetic dimensions.
Coordinate System	<code>coord_*</code> ()	Maps data into the plane of the data rectangle.
Facets	<code>facet_*</code> ()	The arrangement of the data into a grid of plots.
Visual Themes	<code>theme()</code> and <code>theme_*</code> ()	The overall visual defaults of a plot.
Annotations	<code>annotate()</code>	Add additional labels, geometries or images to a plot.

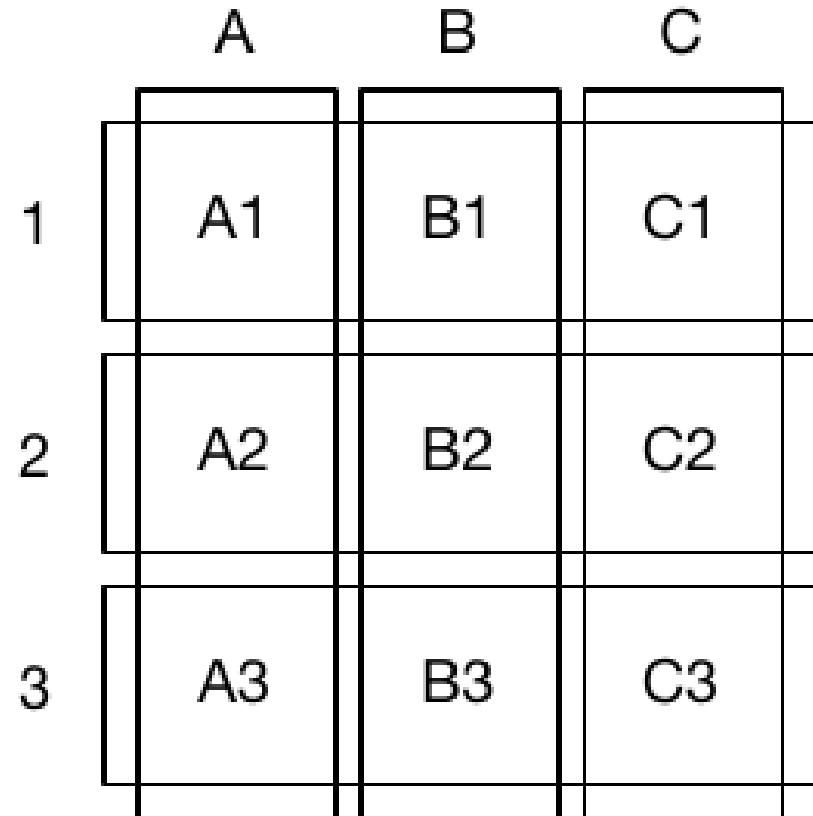
Facets

facet_*()

facet_*()



facet_wrap



facet_grid

Adapted from "ggplot2: Elegant Graphics for Data Analysis" by Hadley Wickham

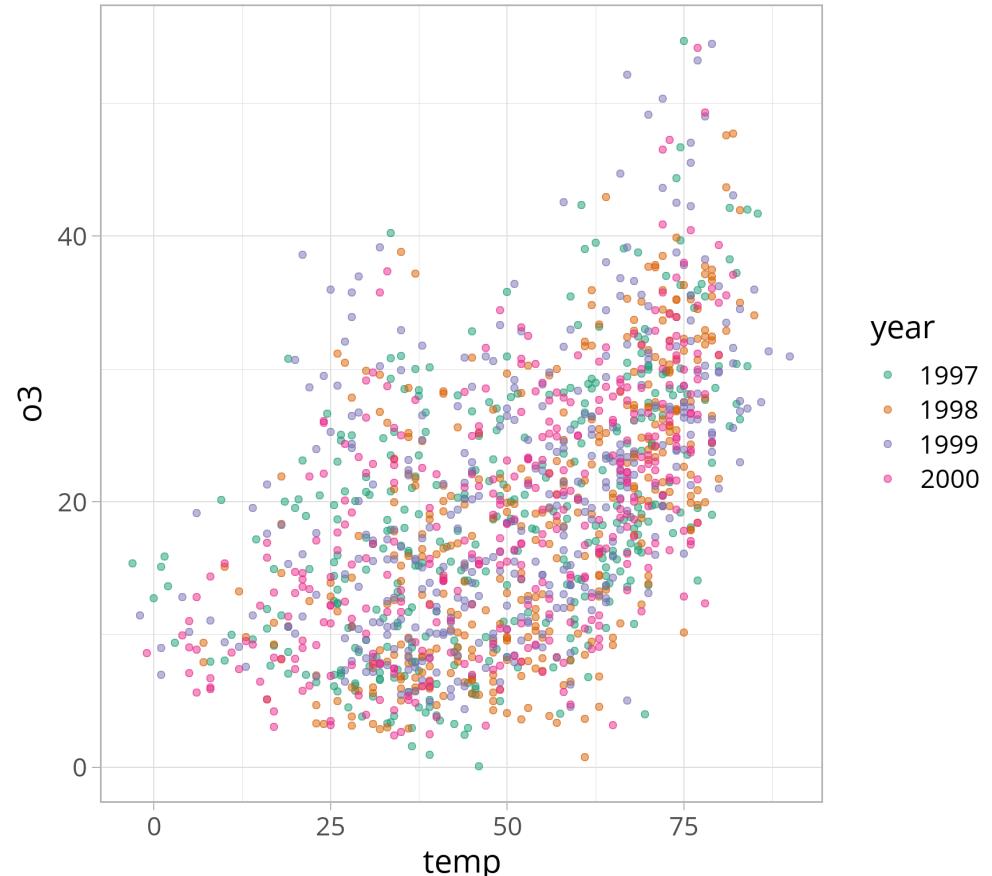
Facets

facet_wrap()

facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

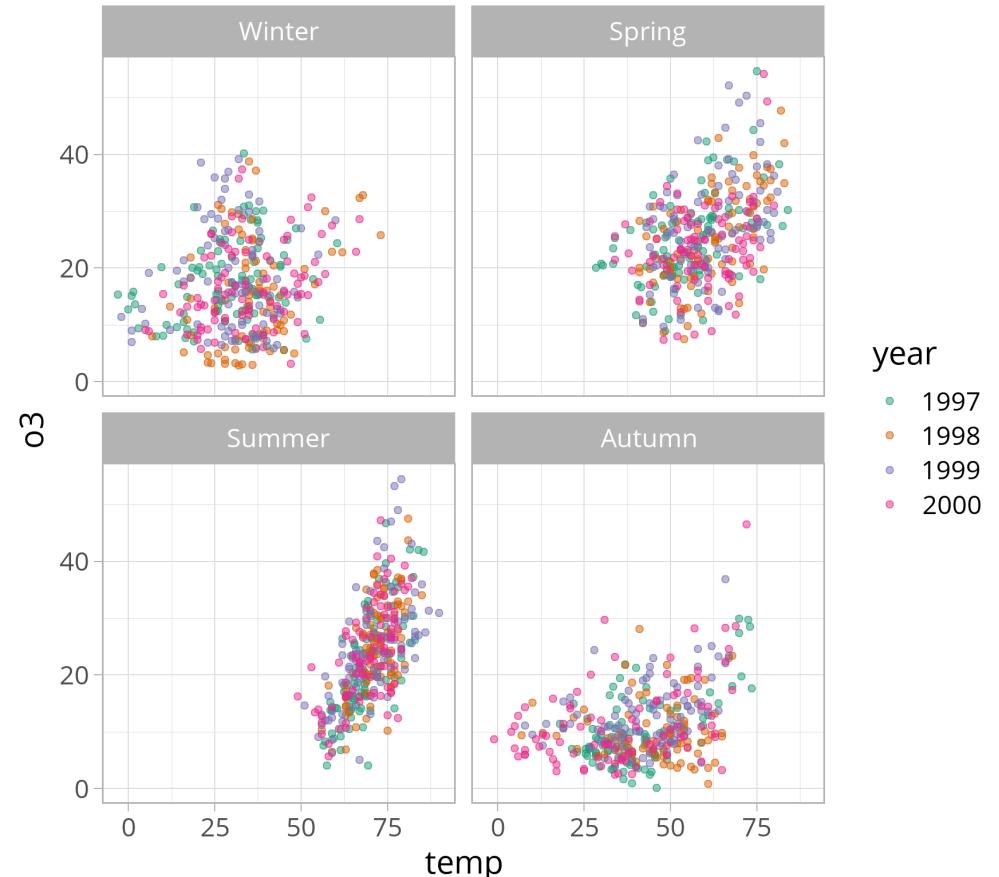
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(  
    aes(color = year),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Dark2"  
  )
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

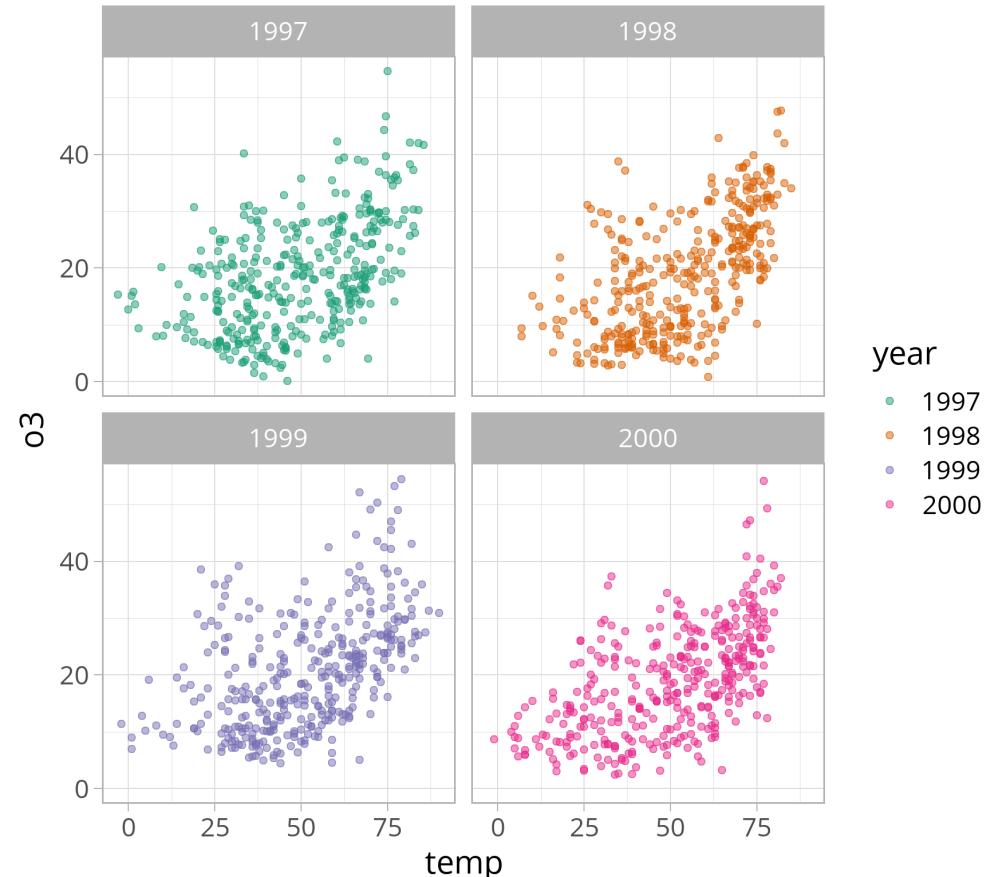
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(  
    aes(color = year),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  facet_wrap(~ season)
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

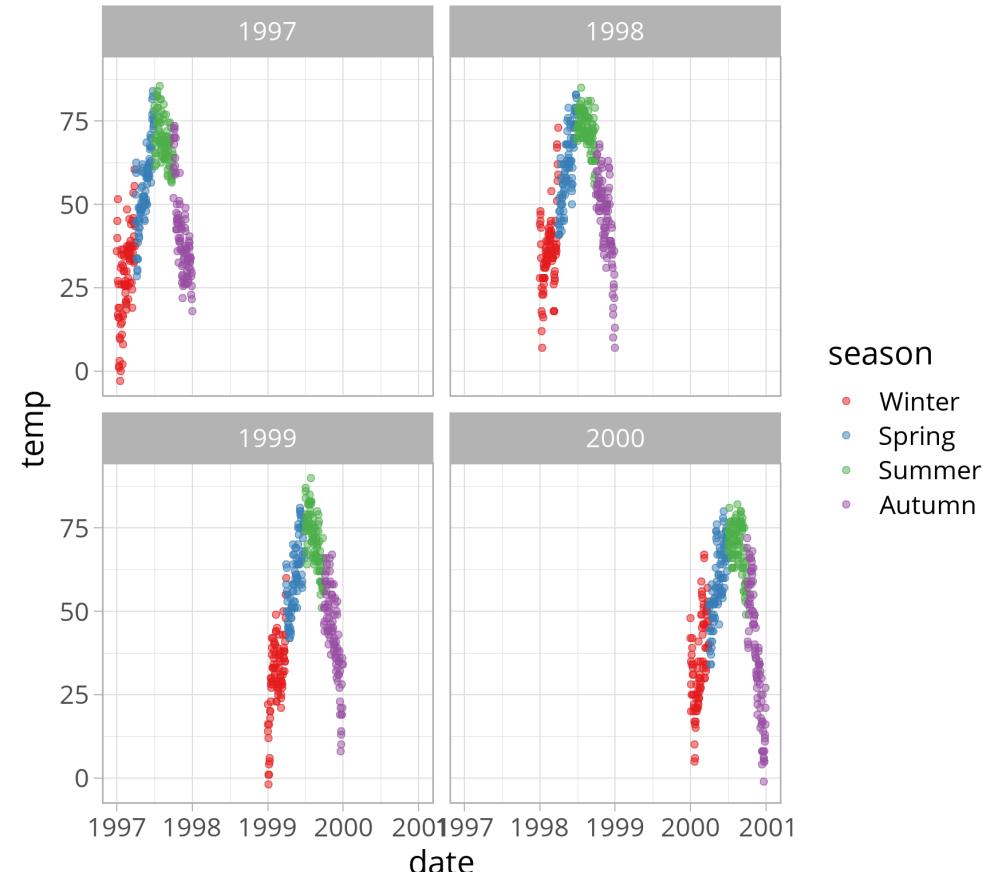
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(  
    aes(color = year),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  facet_wrap(~ year)
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

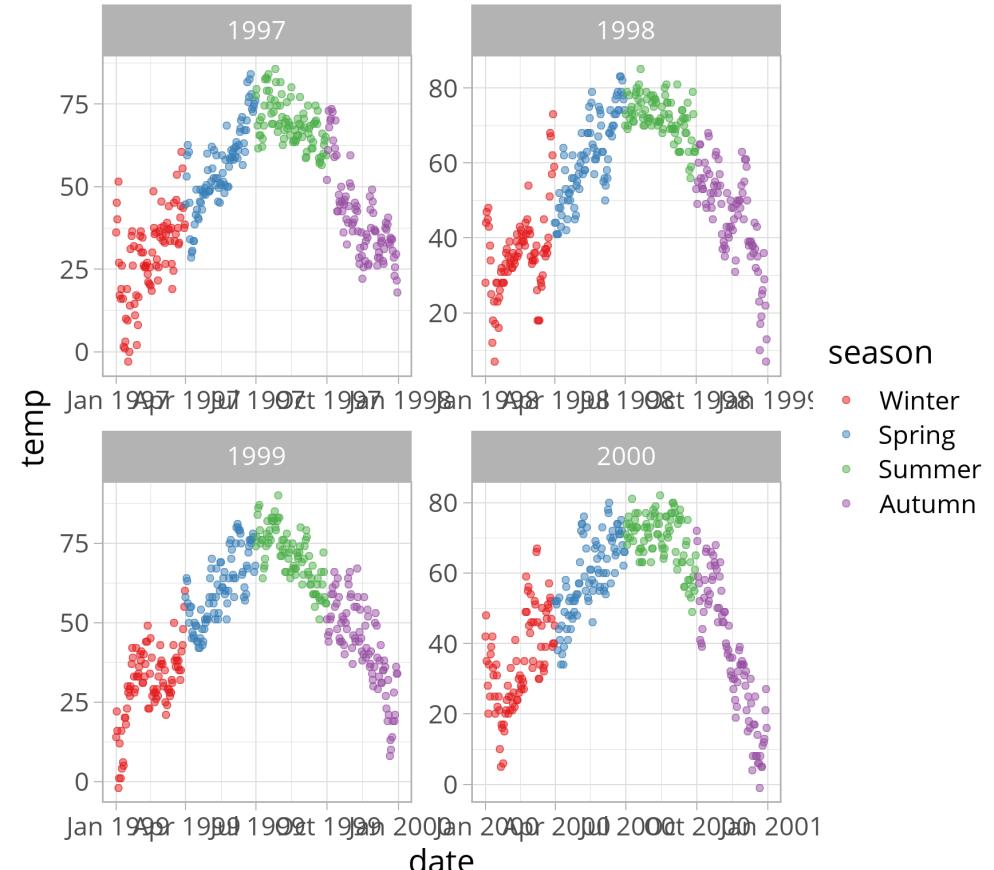
```
ggplot(chic, aes(date, temp)) +  
  geom_point(  
    aes(color = season),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Set1"  
  ) +  
  facet_wrap(~ year)
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

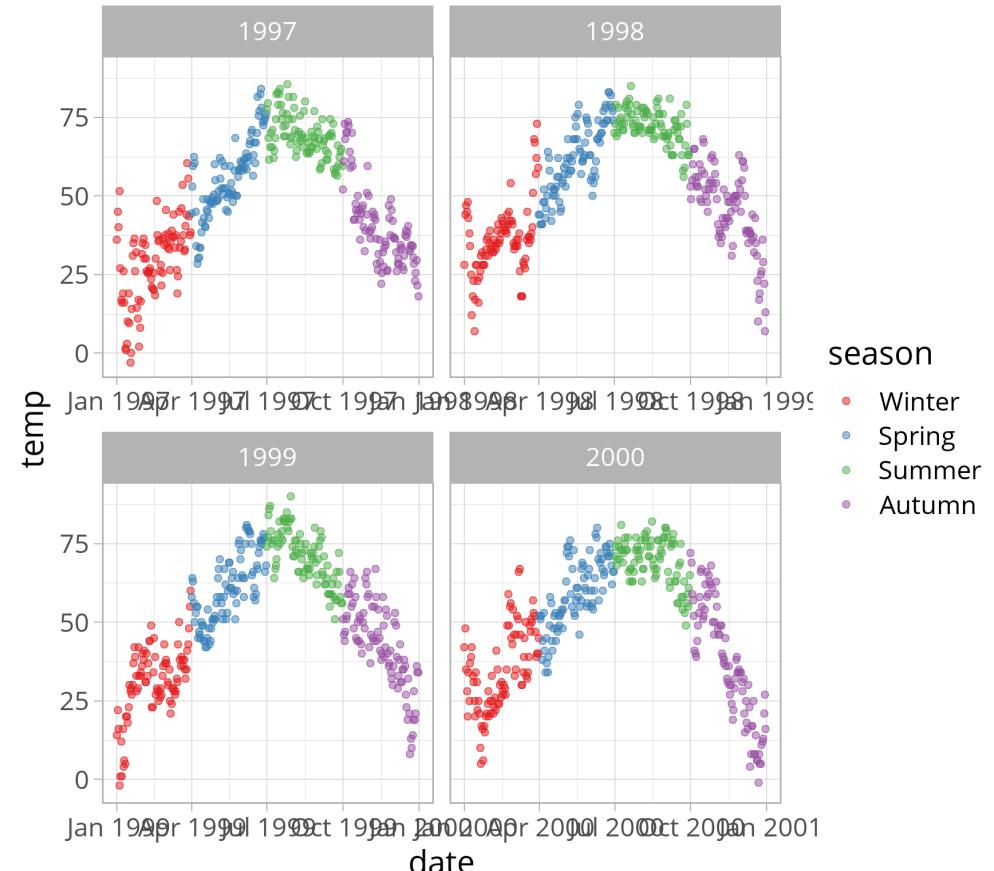
```
ggplot(chic, aes(date, temp)) +  
  geom_point(  
    aes(color = season),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Set1"  
  ) +  
  facet_wrap(  
    ~ year,  
    scales = "free"  
  )
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

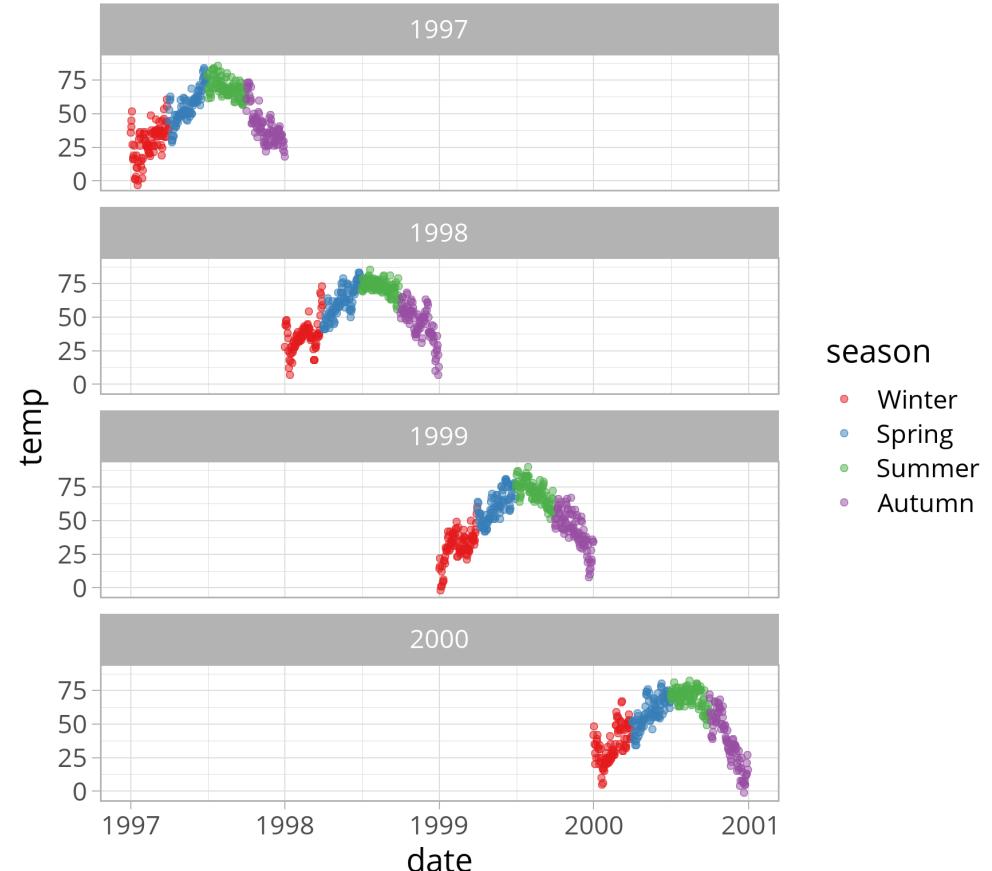
```
ggplot(chic, aes(date, temp)) +  
  geom_point(  
    aes(color = season),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Set1"  
  ) +  
  facet_wrap(  
    ~ year,  
    scales = "free_x"  
  )
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

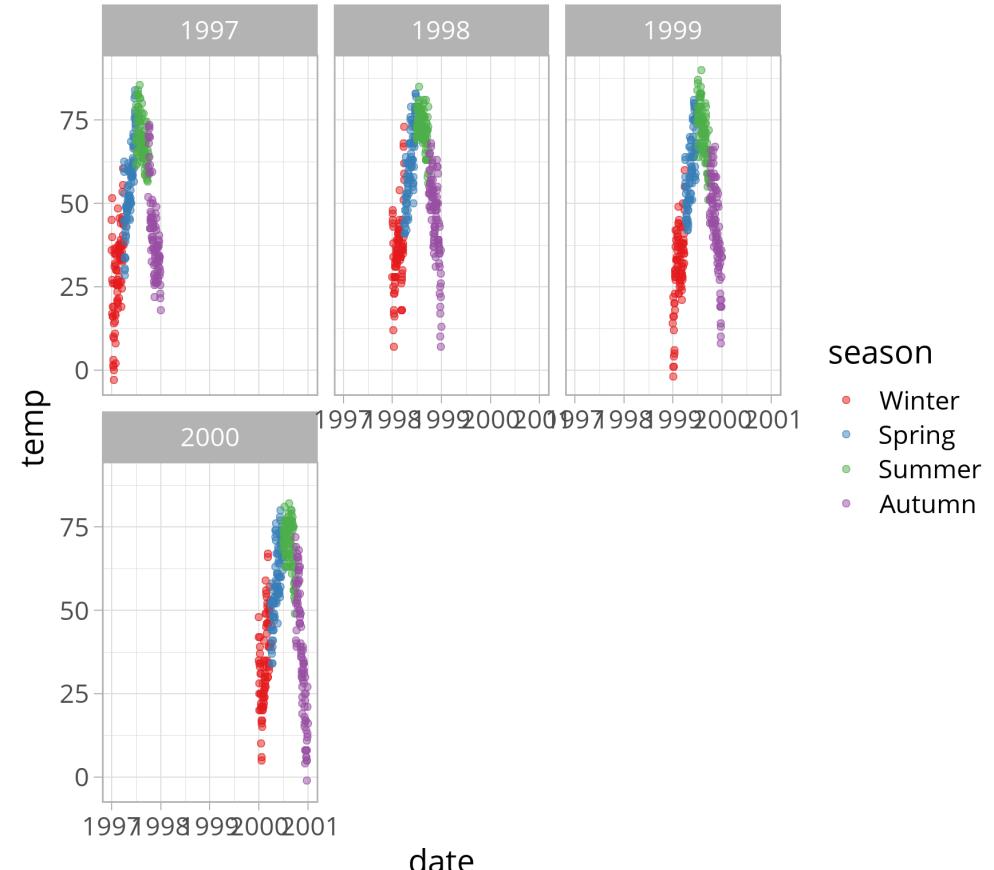
```
ggplot(chic, aes(date, temp)) +  
  geom_point(  
    aes(color = season),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Set1"  
  ) +  
  facet_wrap(  
    ~ year,  
    nrow = 4  
  )
```



facet_wrap()

`facet_wrap()` splits the data into small multiples based on **one grouping variable**:

```
ggplot(chic, aes(date, temp)) +  
  geom_point(  
    aes(color = season),  
    alpha = .5  
  ) +  
  scale_color_brewer(  
    palette = "Set1"  
  ) +  
  facet_wrap(  
    ~ year,  
    ncol = 3  
  )
```



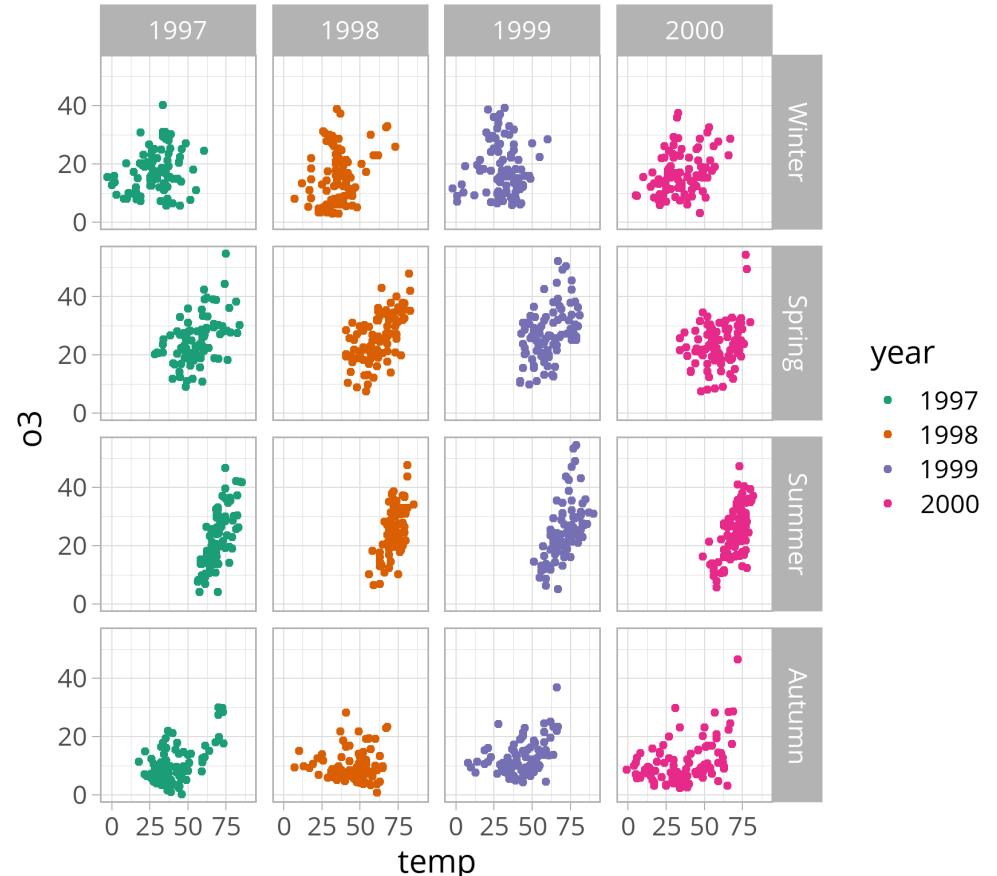
Facets

facet_grid()

facet_grid()

`facet_grid()` spans a grid of each combination of **two grouping variables**:

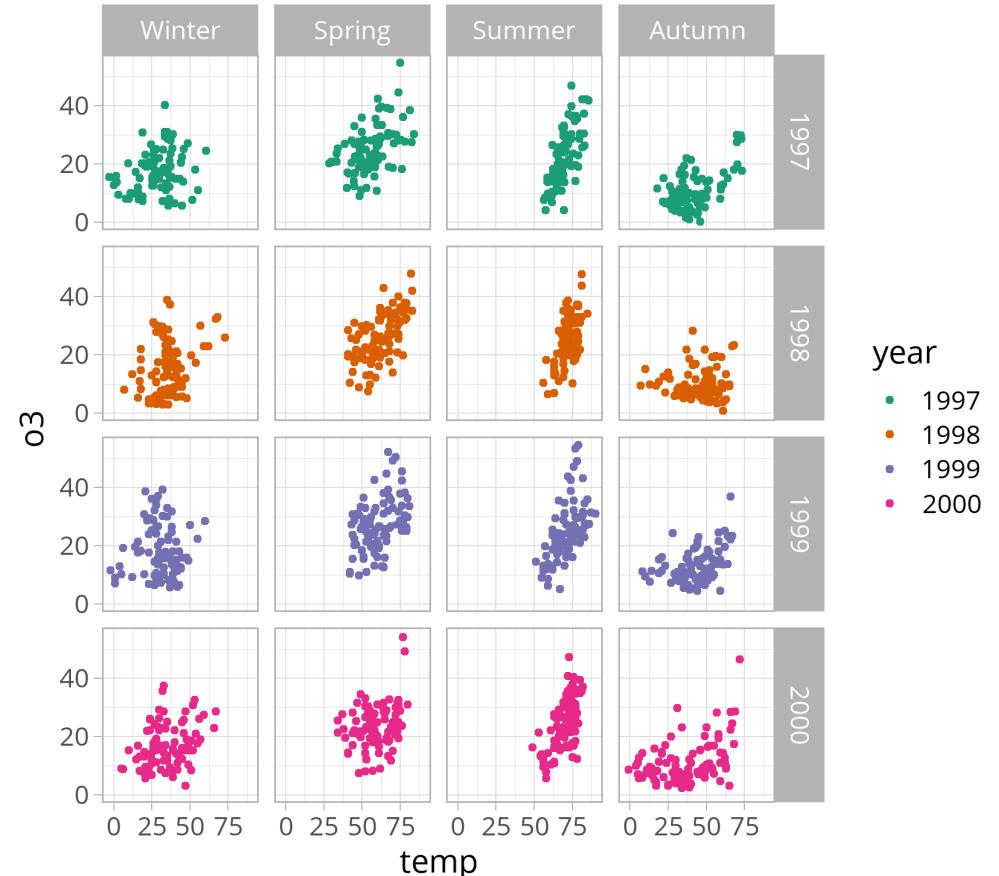
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(aes(color = year)) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  facet_grid(season ~ year)
```



facet_grid()

`facet_grid()` spans a grid of each combination of **two grouping variables**:

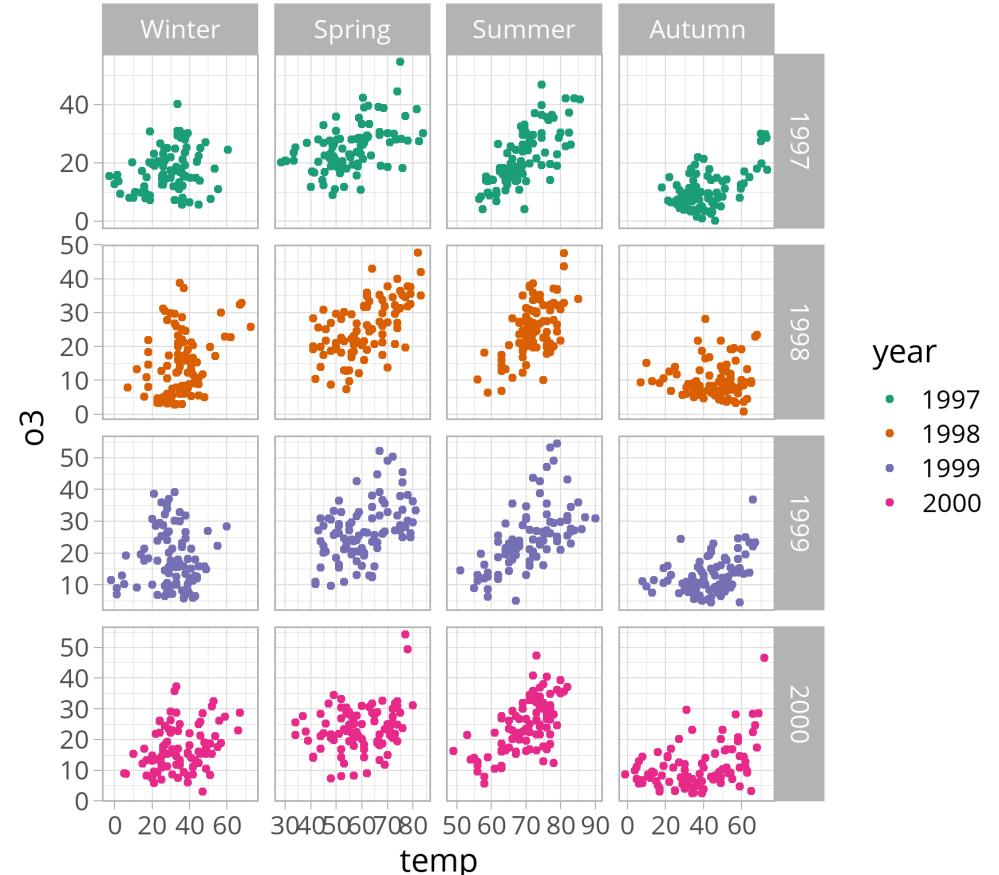
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(aes(color = year)) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  facet_grid(year ~ season)
```



facet_grid()

`facet_grid()` spans a grid of each combination of **two grouping variables**:

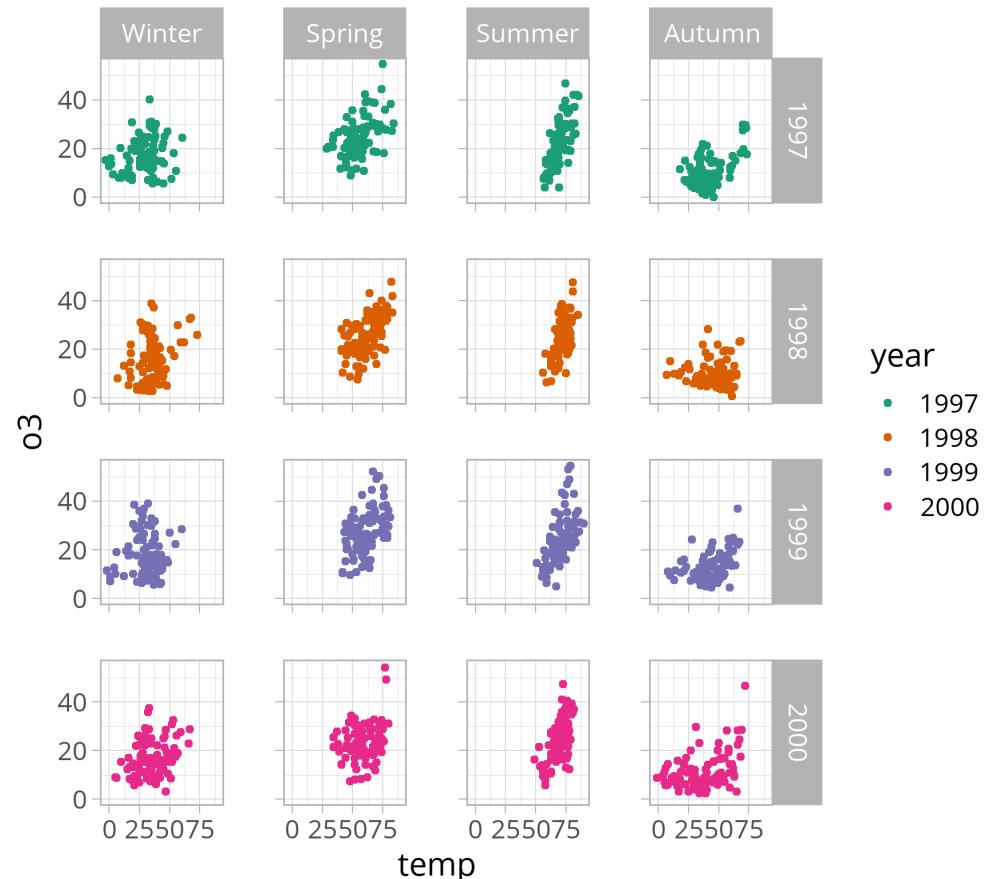
```
ggplot(chic, aes(temp, o3)) +  
  geom_point(aes(color = year)) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  facet_grid(  
    year ~ season,  
    scales = "free"  
  )
```



facet_rep_grid()

`facet_rep_grid()` from the `lemon` package allows you to draw axis lines and ticks for all facets:

```
ggplot(chic, aes(temp, o3)) +  
  geom_point(aes(color = year)) +  
  scale_color_brewer(  
    palette = "Dark2"  
  ) +  
  lemon::facet_rep_grid(  
    year ~ season  
)
```





The **patchwork** Package

The Composer of Plots

patchwork

Combine + arrange
your ggplots!

PLAN:
 $(P1+P2)/P3$

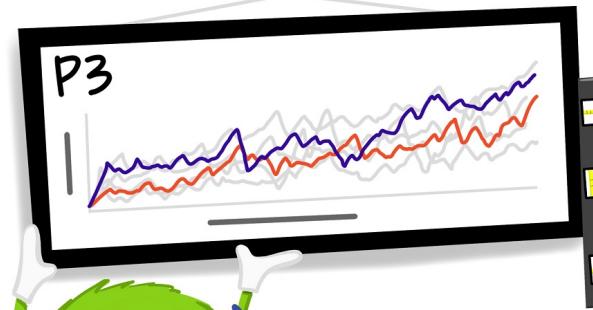
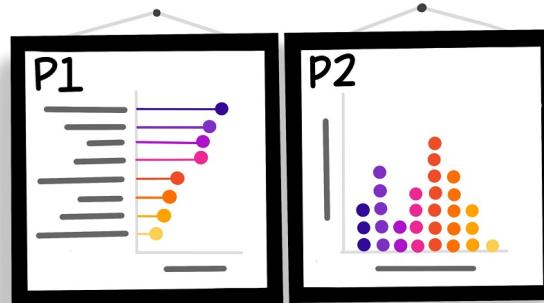
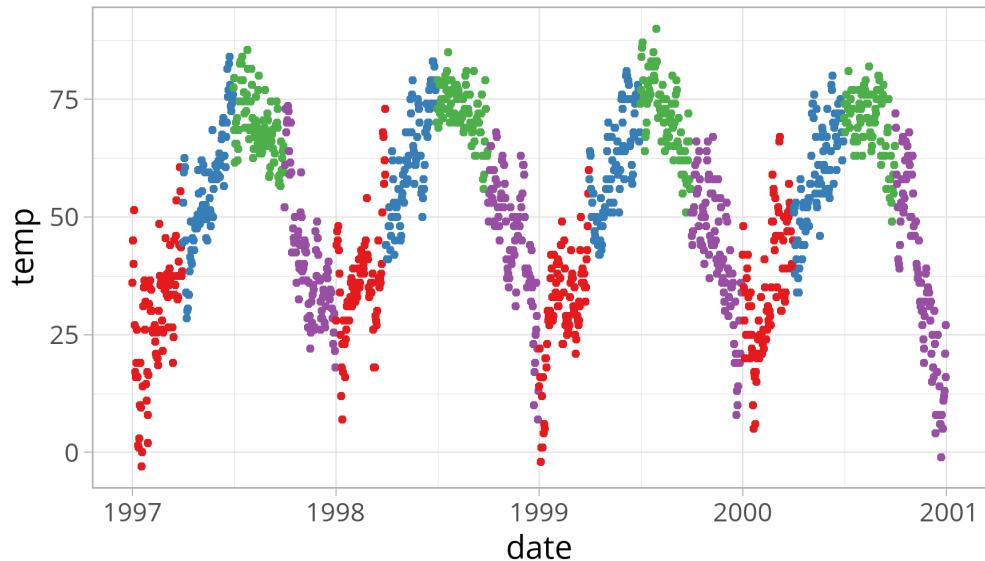


Illustration by Allison Horst (github.com/allisonhorst/stats-illustrations)

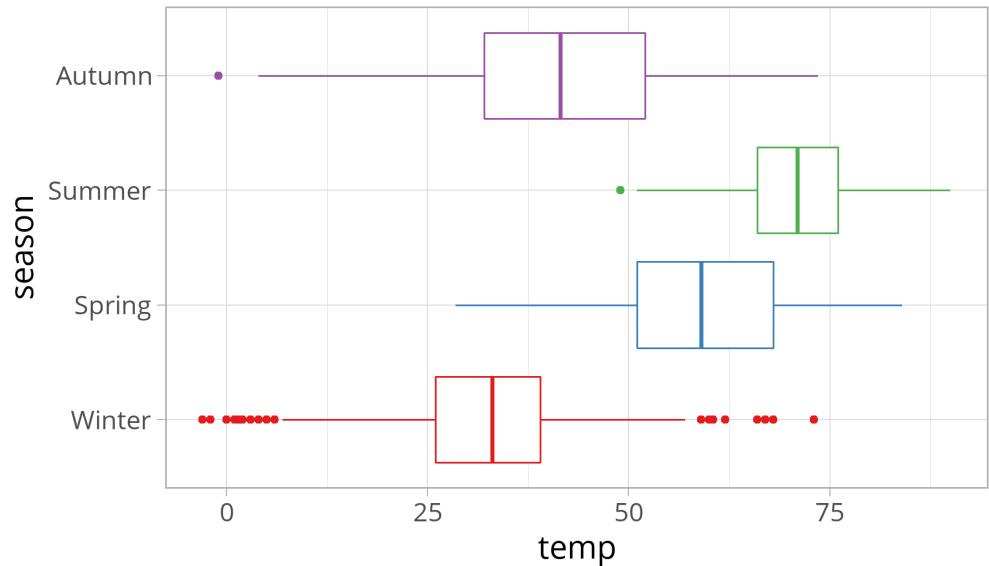
The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

```
(time <- ggplot(chic, aes(date, temp)) +  
  geom_point(aes(color = season)) +  
  scale_color_brewer(palette = "Set1",  
                     guide = "none"))
```



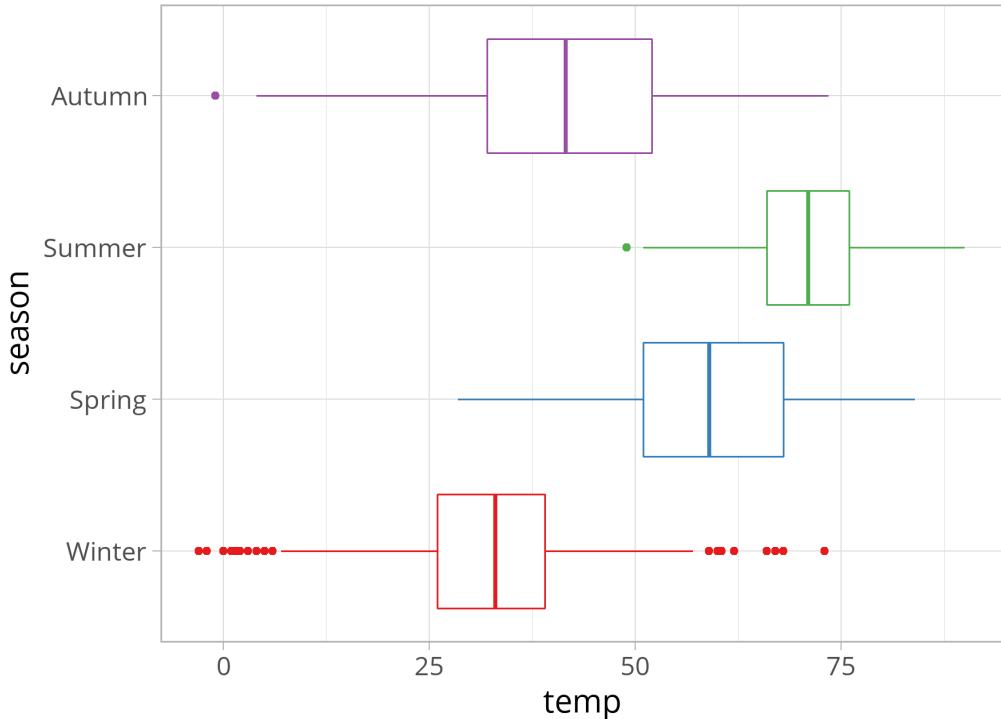
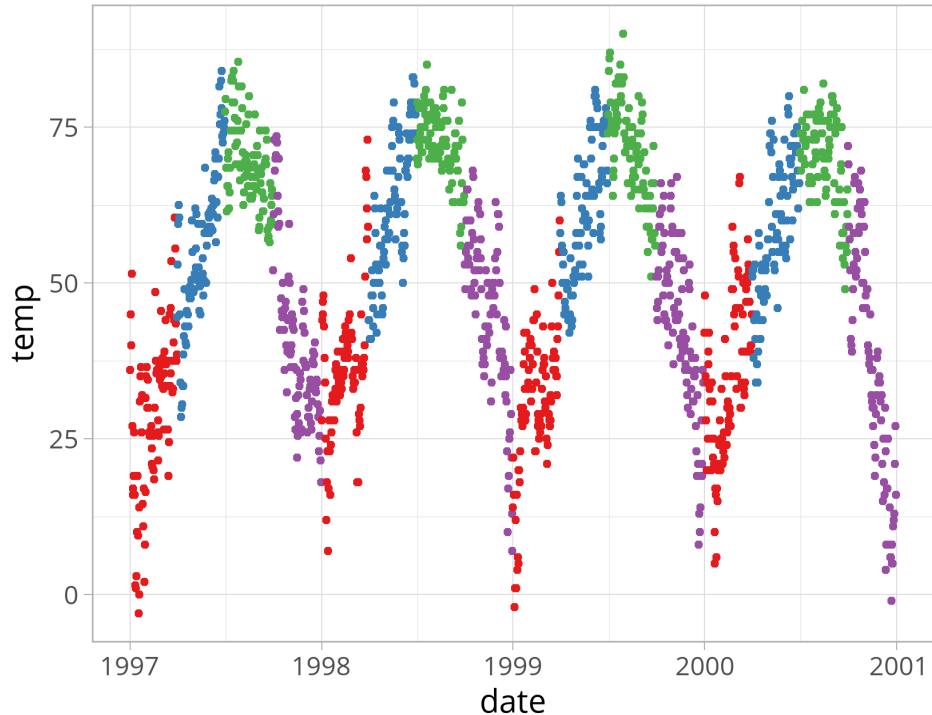
```
(box <- ggplot(chic, aes(temp, season)) +  
  geom_boxplot(aes(color = season)) +  
  scale_color_brewer(palette = "Set1",  
                     guide = "none"))
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

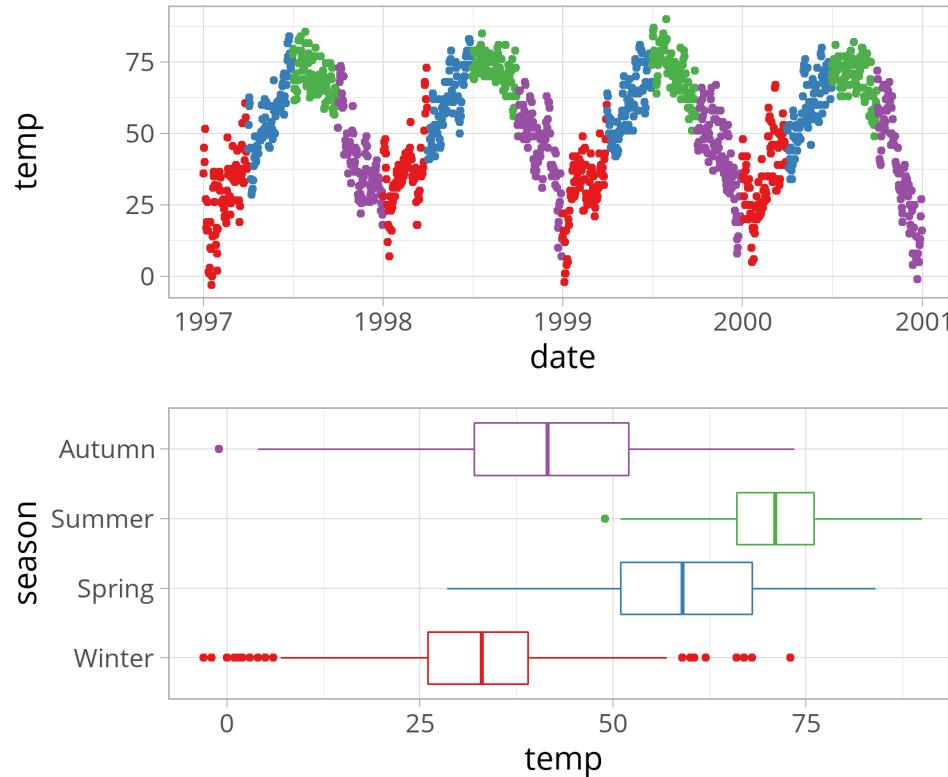
```
library(patchwork)  
time + box
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

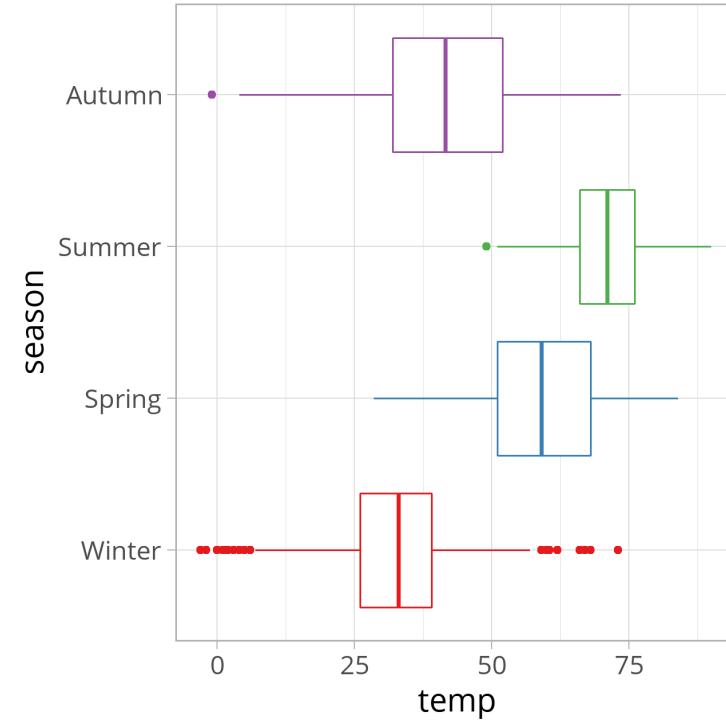
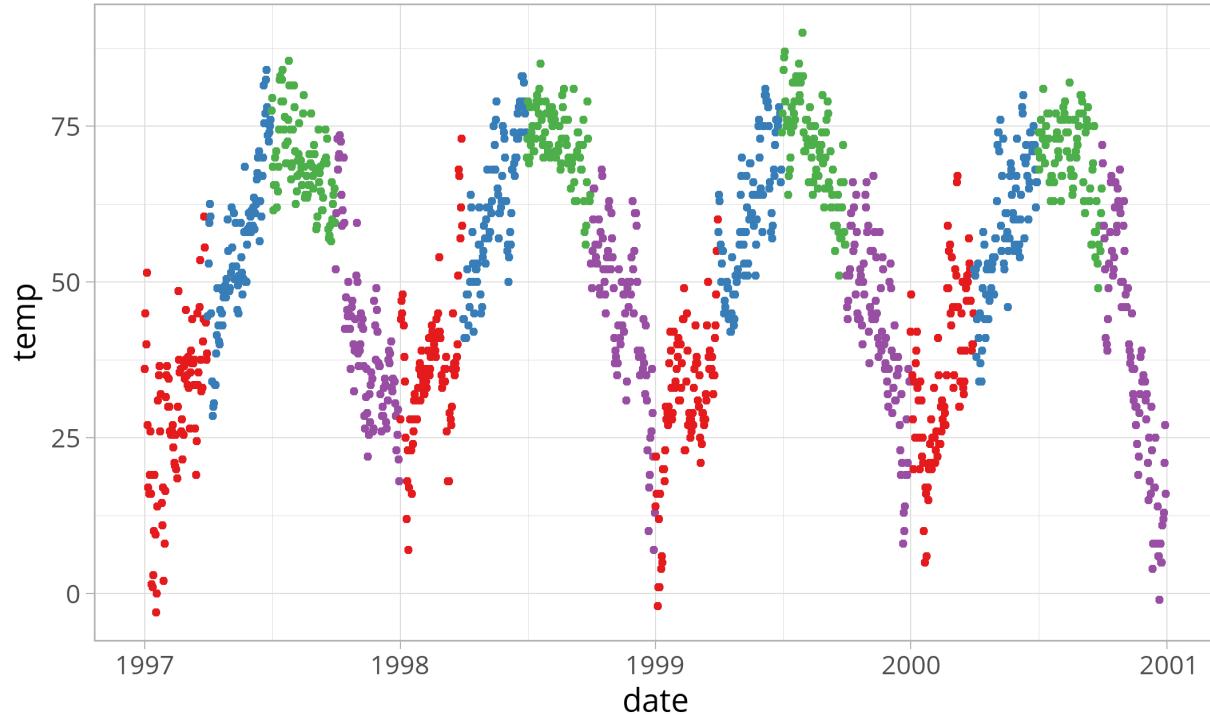
time / box



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

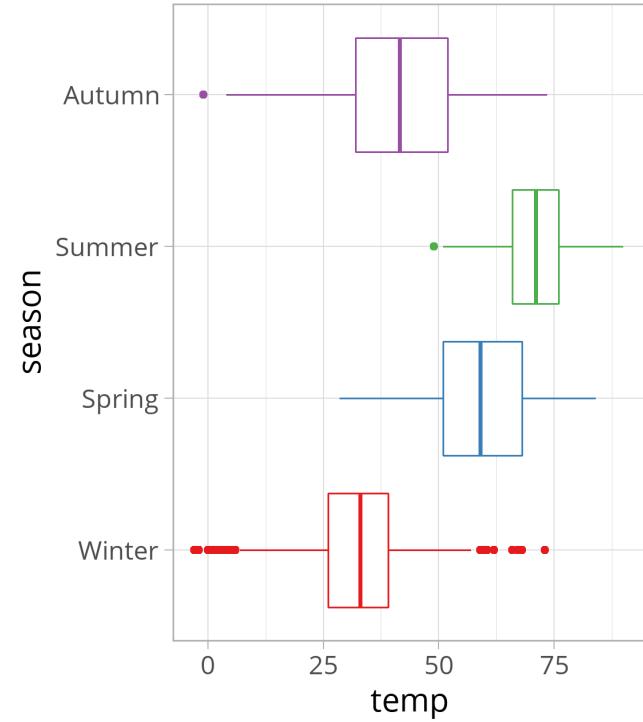
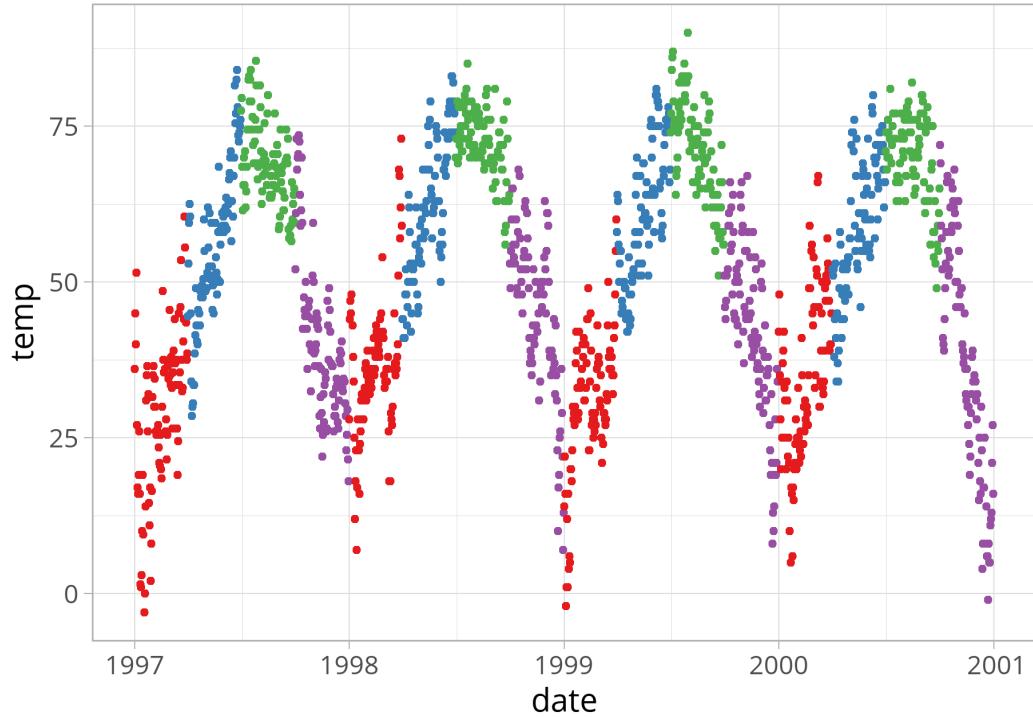
```
time + box + plot_layout(widths = c(2, 1))
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

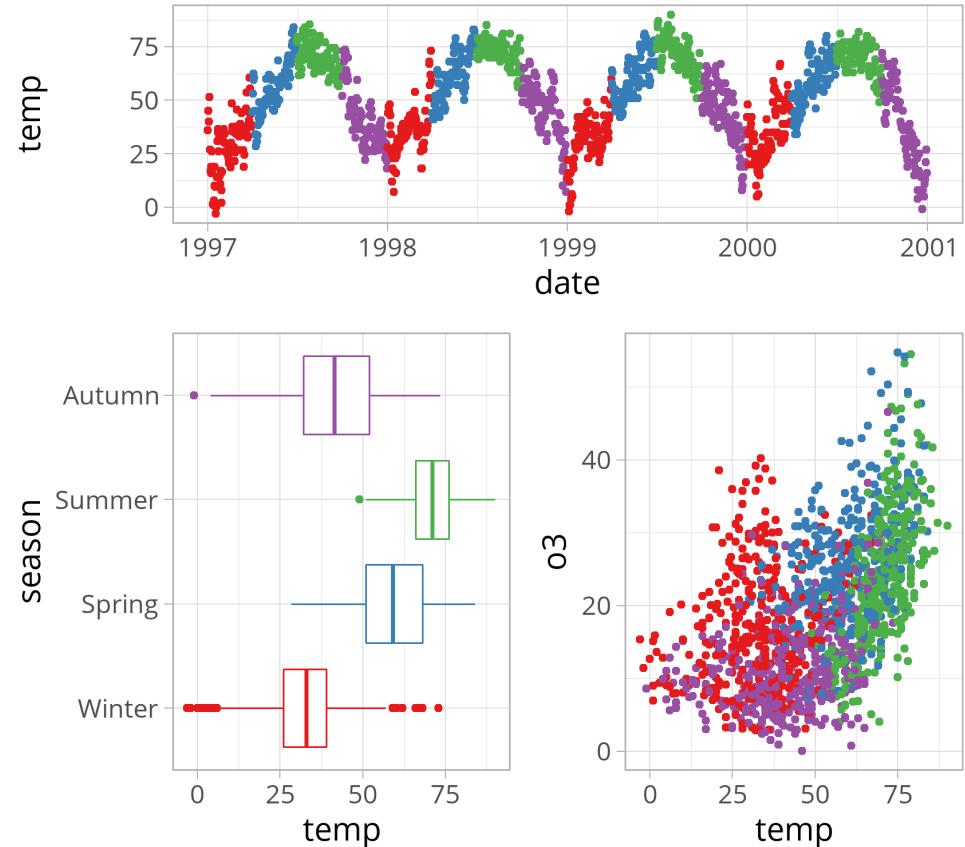
```
time + plot_spacer() + box + plot_layout(widths = c(2, .5, 1))
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

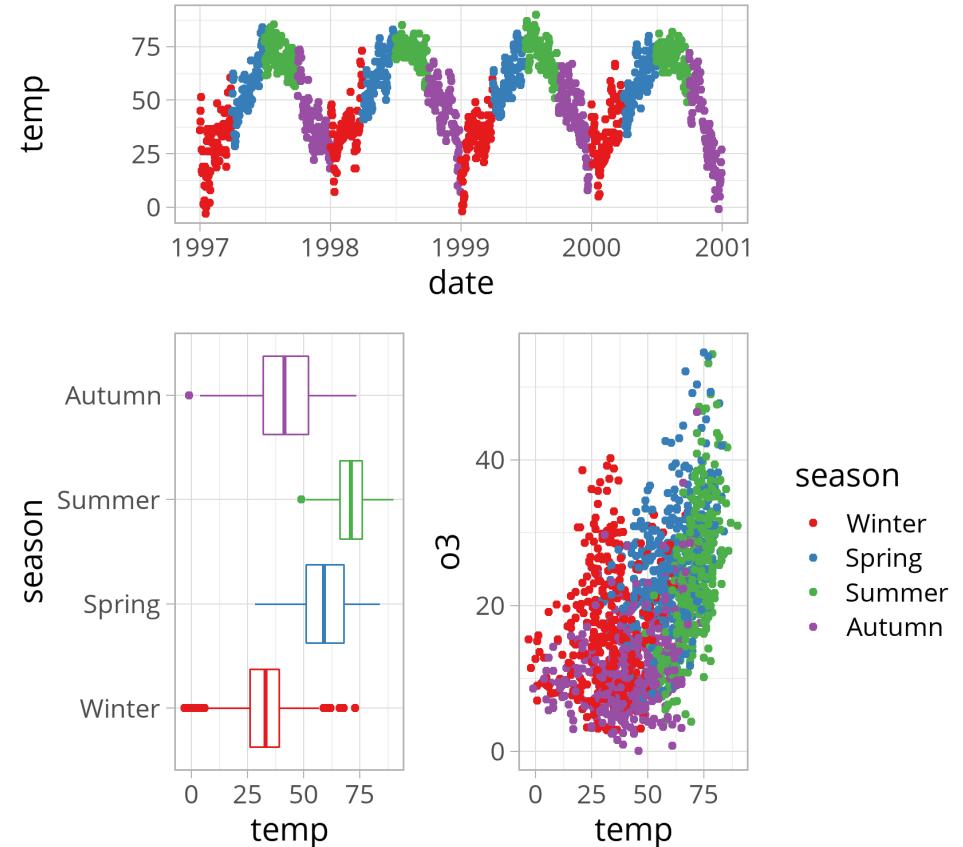
```
scatter <-  
  ggplot(  
    chic,  
    aes(temp, o3)  
  ) +  
  geom_point(aes(color = season)) +  
  scale_color_brewer(  
    palette = "Set1",  
    guide = "none"  
  )  
  
time / (box + scatter) +  
  plot_layout(heights = c(1, 2))
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

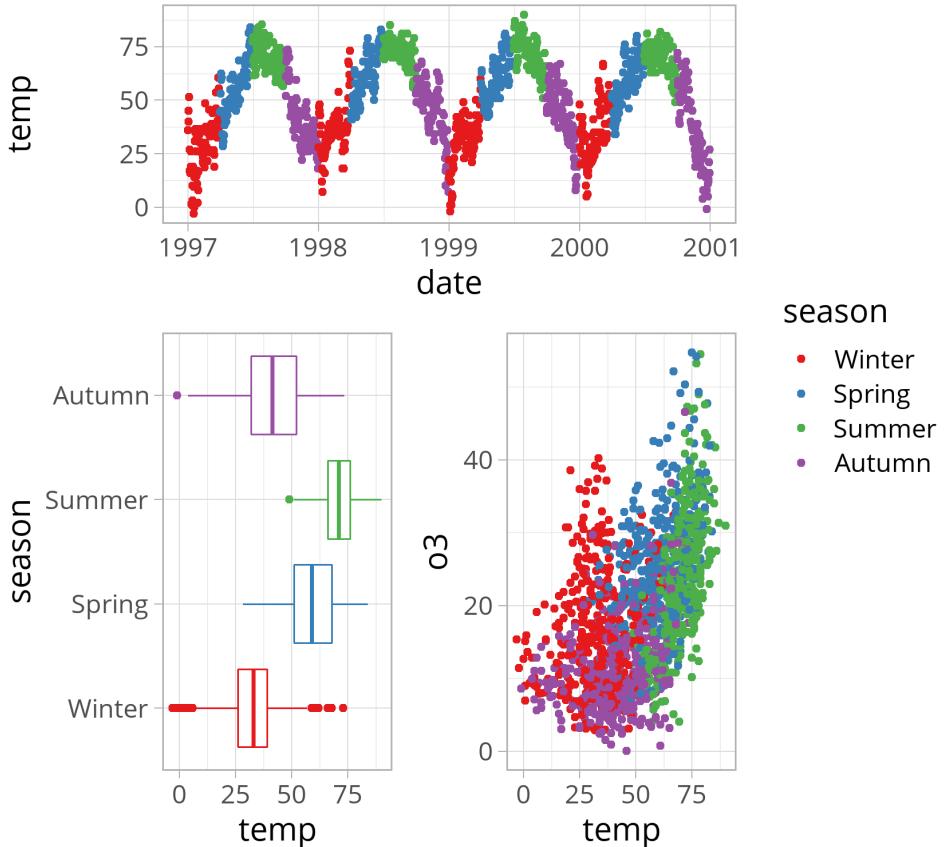
```
scatter <-  
  ggplot(  
    chic,  
    aes(temp, o3)  
  ) +  
  geom_point(aes(color = season)) +  
  scale_color_brewer(  
    palette = "Set1"#,  
    #guide = "none"  
  )  
  
time / (box + scatter) +  
  plot_layout(heights = c(1, 2))
```



The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

```
time / (box + scatter) +
  plot_layout(
    heights = c(1, 2),
    guides = "collect"
  )
```

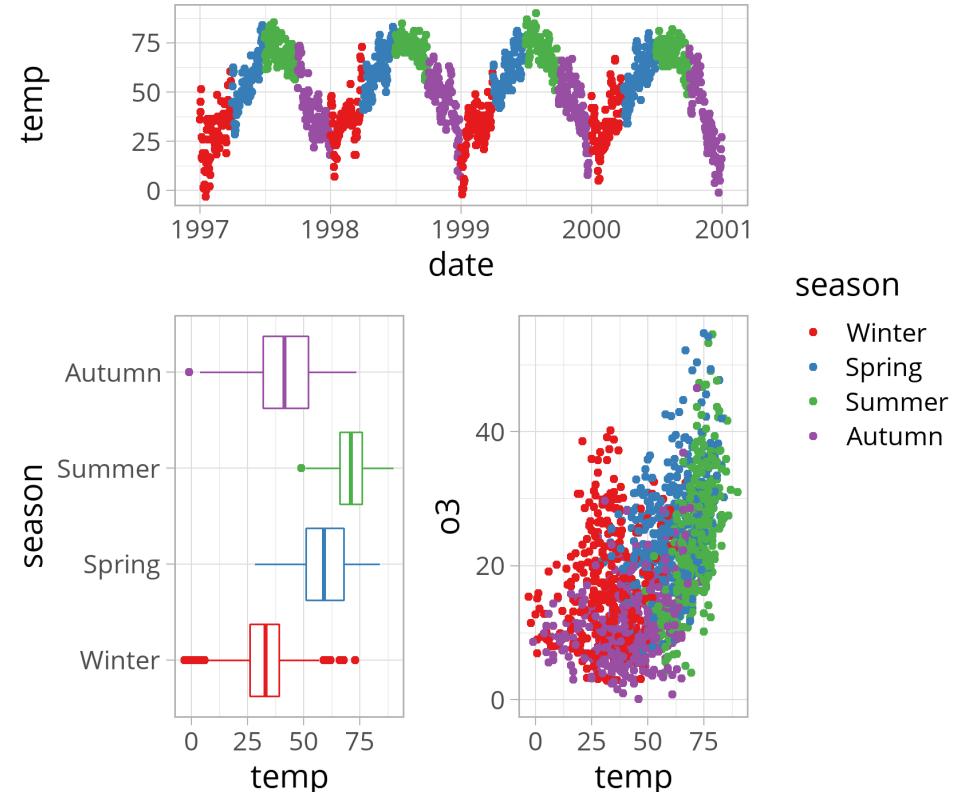


The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

```
time / (box + scatter) +
  plot_layout(
    heights = c(1, 2),
    guides = "collect"
  ) +
  plot_annotation(
    title = "Temperature & Ozone in Chicago, IL"
  )
```

Temperature & Ozone in Chicago, IL



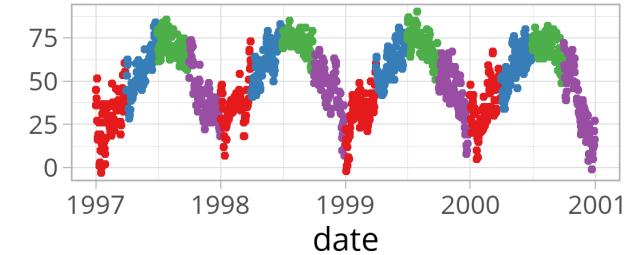
The patchwork package

Build up your multipanel plot sequentially using **The Composer of Plots**:

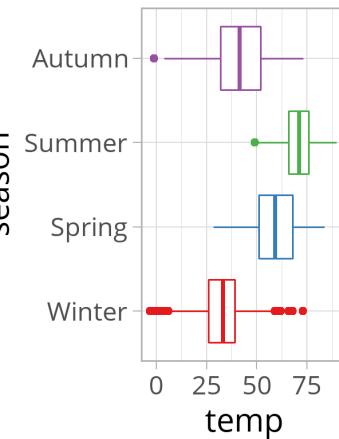
```
time / (box + scatter) +
  plot_layout(
    heights = c(1, 2),
    guides = "collect"
  ) +
  plot_annotation(
    title = "Temperature & Ozone in Chicago, IL",
    tag_levels = "A"
  )
```

Temperature & Ozone in Chicago, IL

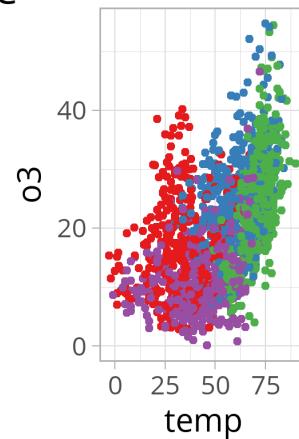
A



B

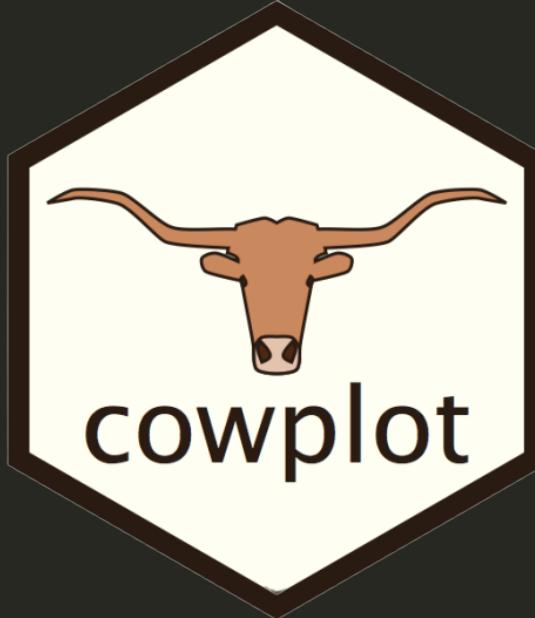


C



season

- Winter
- Spring
- Summer
- Autumn



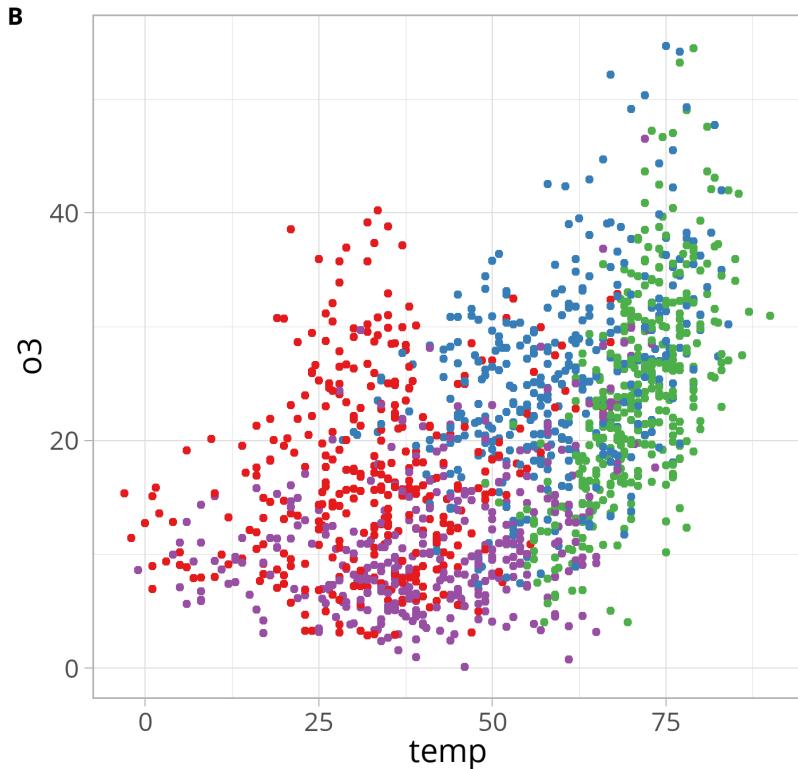
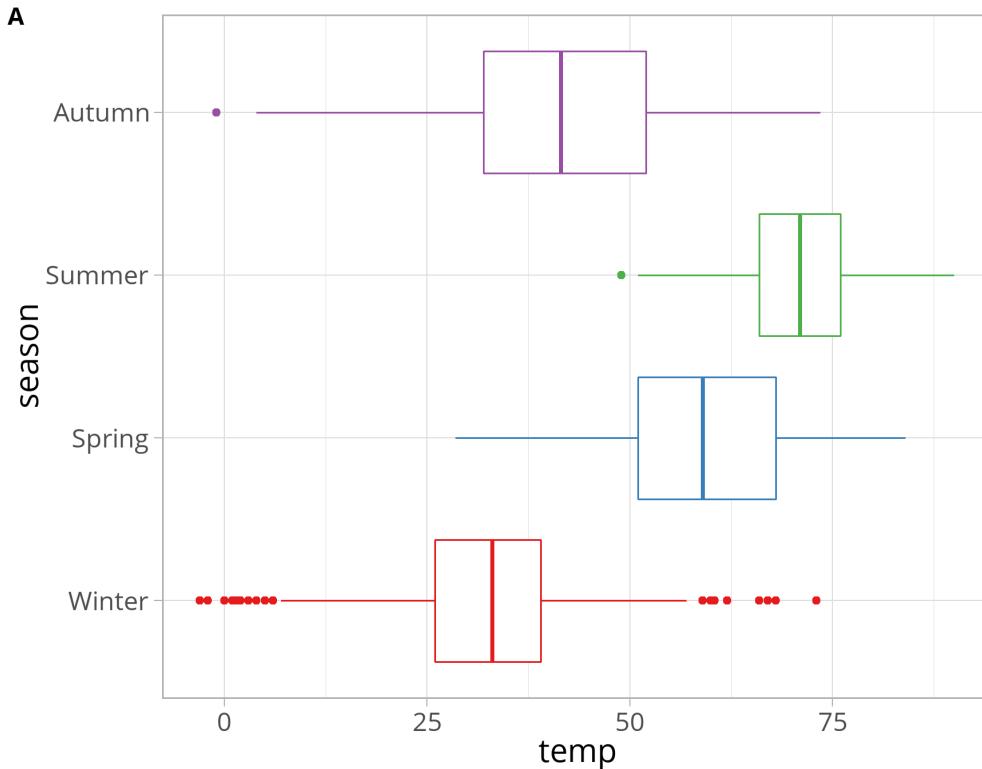
The **cowplot** Package

Streamlined Plot Annotations

The `cowplot` Package

The `cowplot` package provides the function `plot_grid()` to arrange plots into a grid and label them:

```
library(cowplot)
plot_grid(box, scatter, labels = c('A', 'B'), label_size = 12)
```



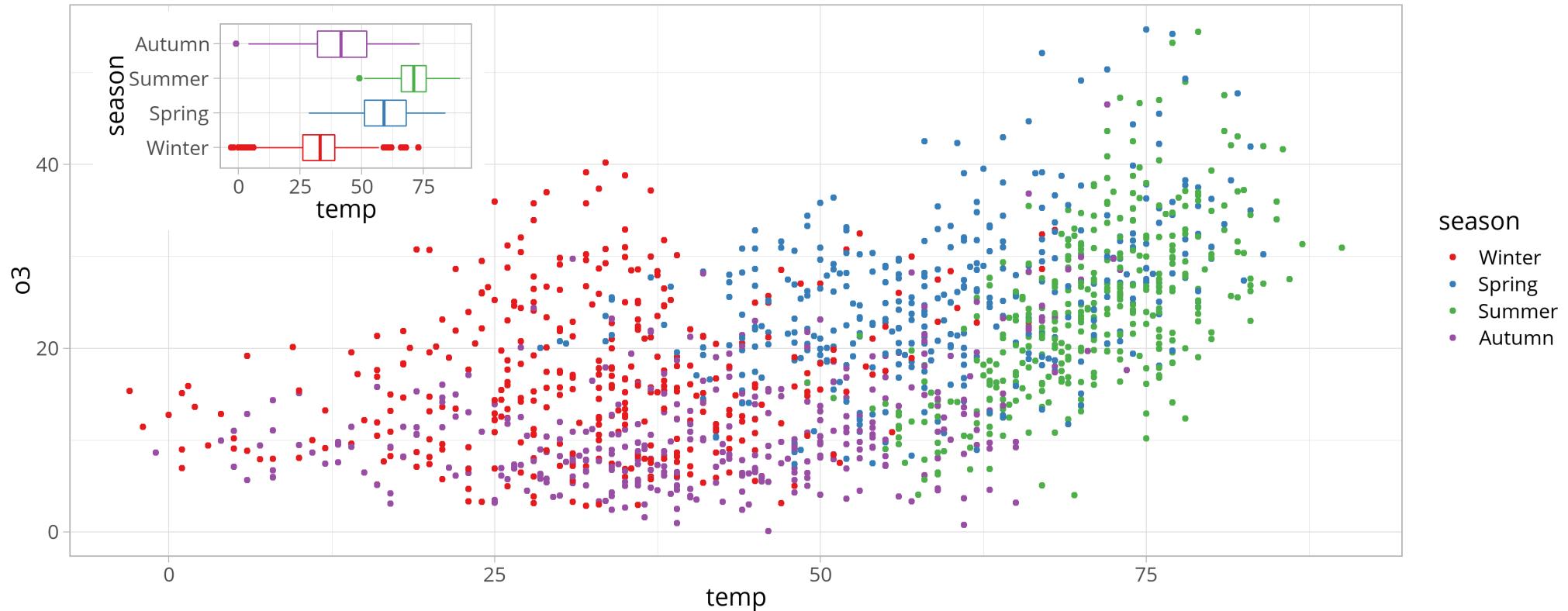
season

- Winter
- Spring
- Summer
- Autumn

The `cowplot` Package

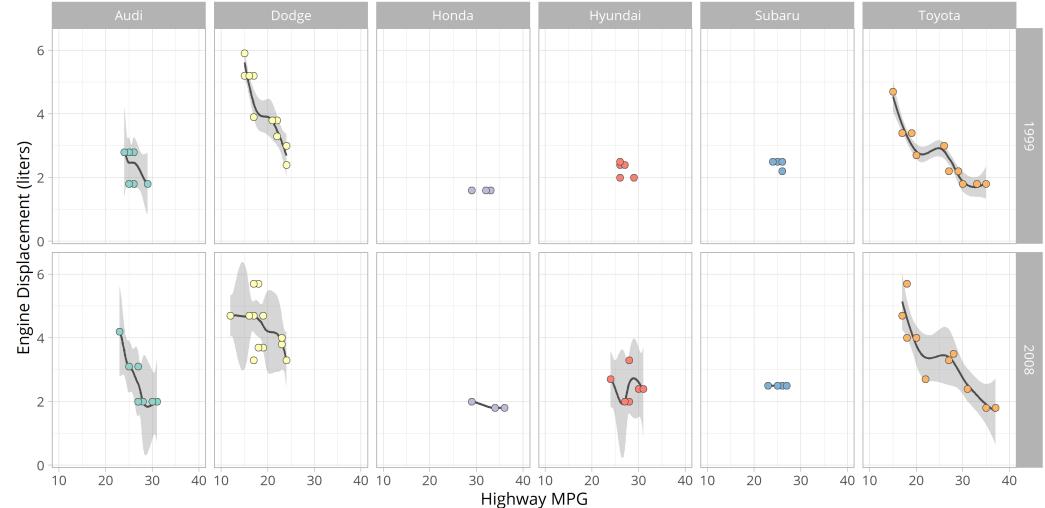
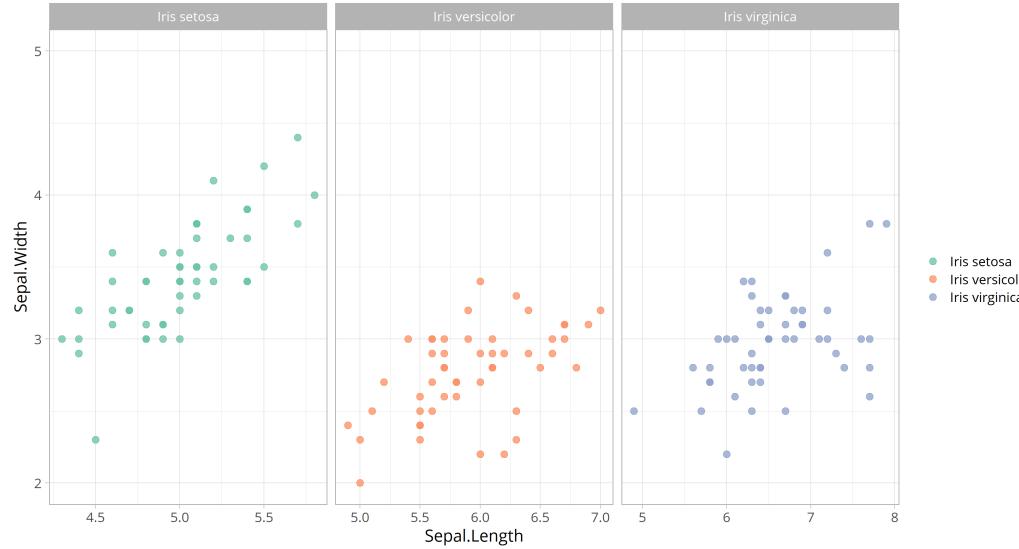
The `cowplot` package provides the function `plot_grid()` to arrange plots into a grid and label them:

```
ggdraw(scatter) + draw_plot(box, .06, .62, .25, .35)
```



Your Turn!

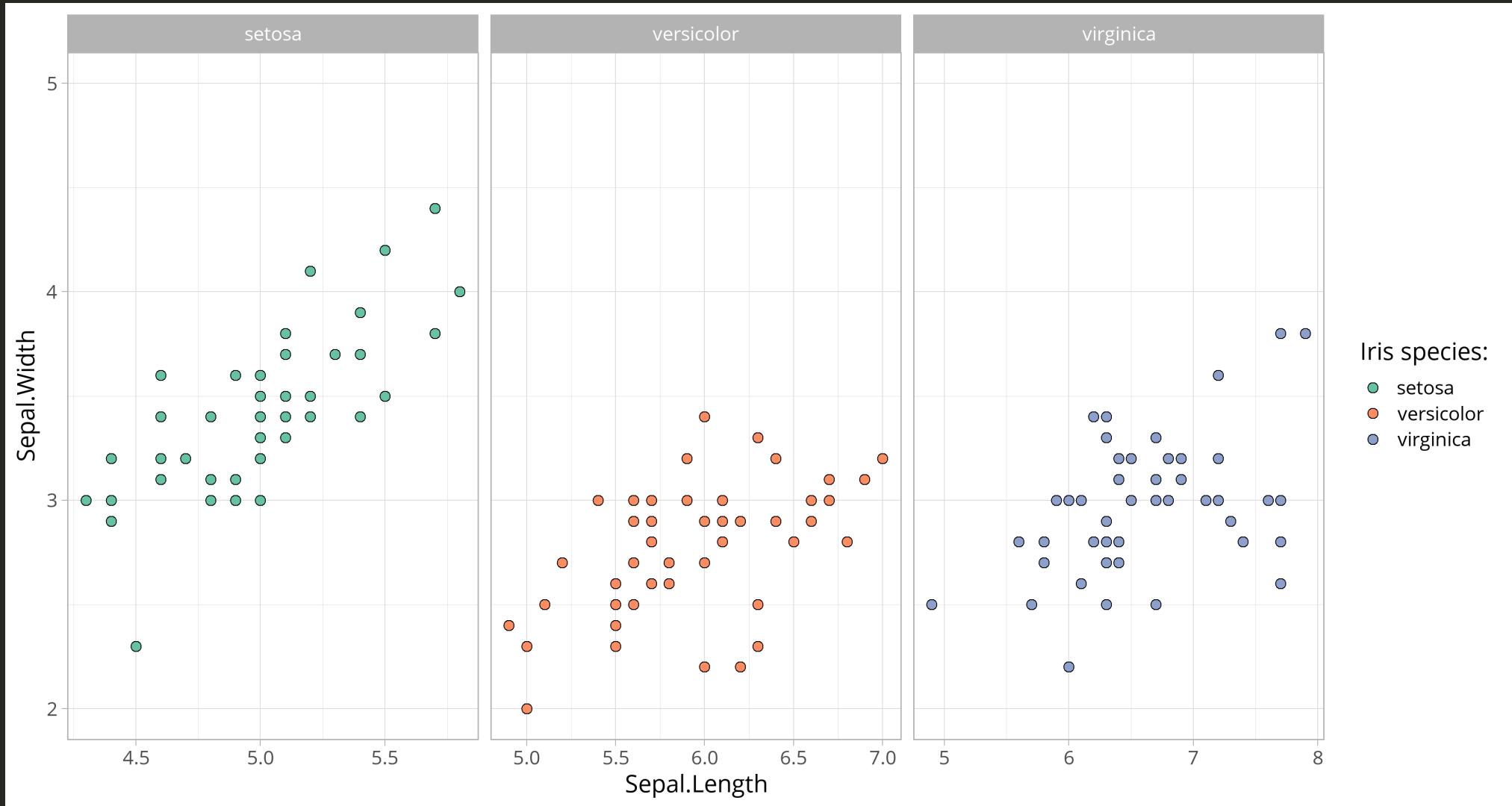
- Create the following two facets:



Data: EPA, available on www.fueleconomy.gov.
Note: Only manufacturers ending with a vowel are shown.

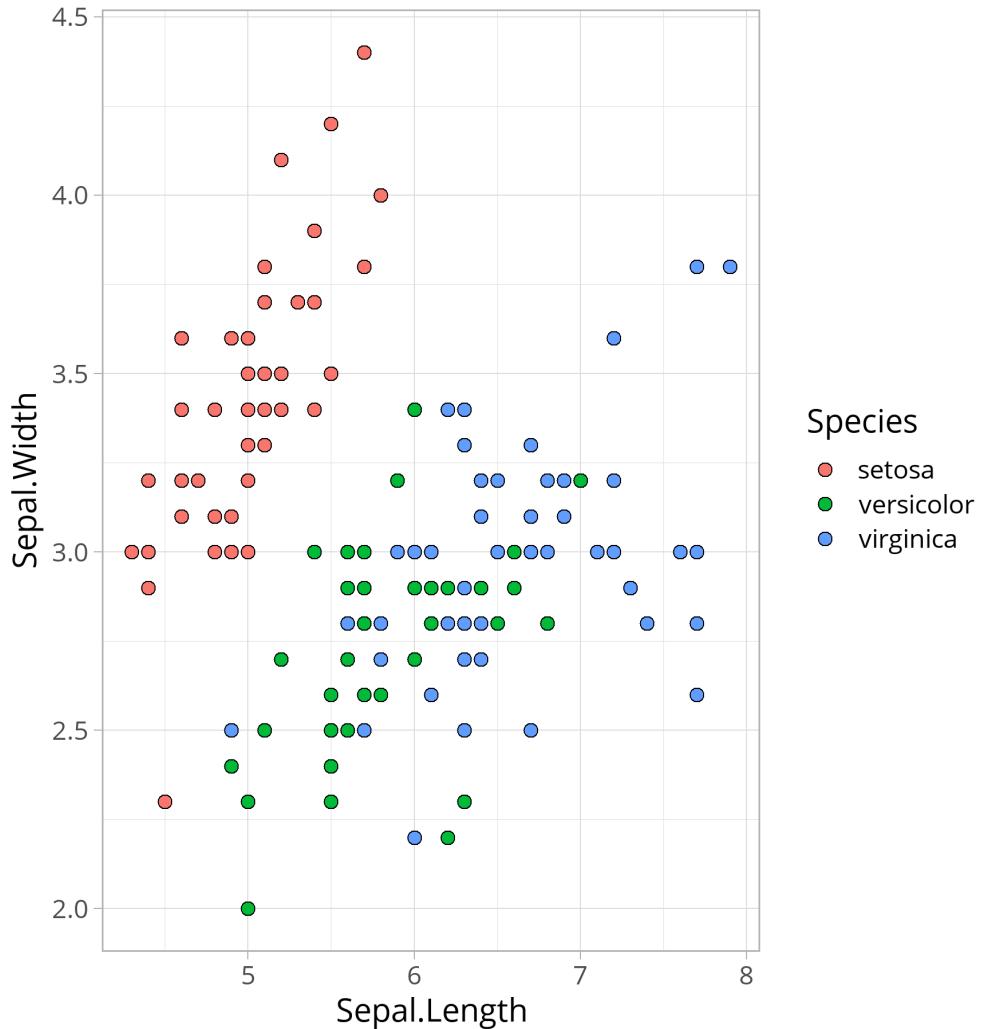
- Create three plots with legends for shape and fill. Combine them with patchwork or cowplot.

Your Turn!



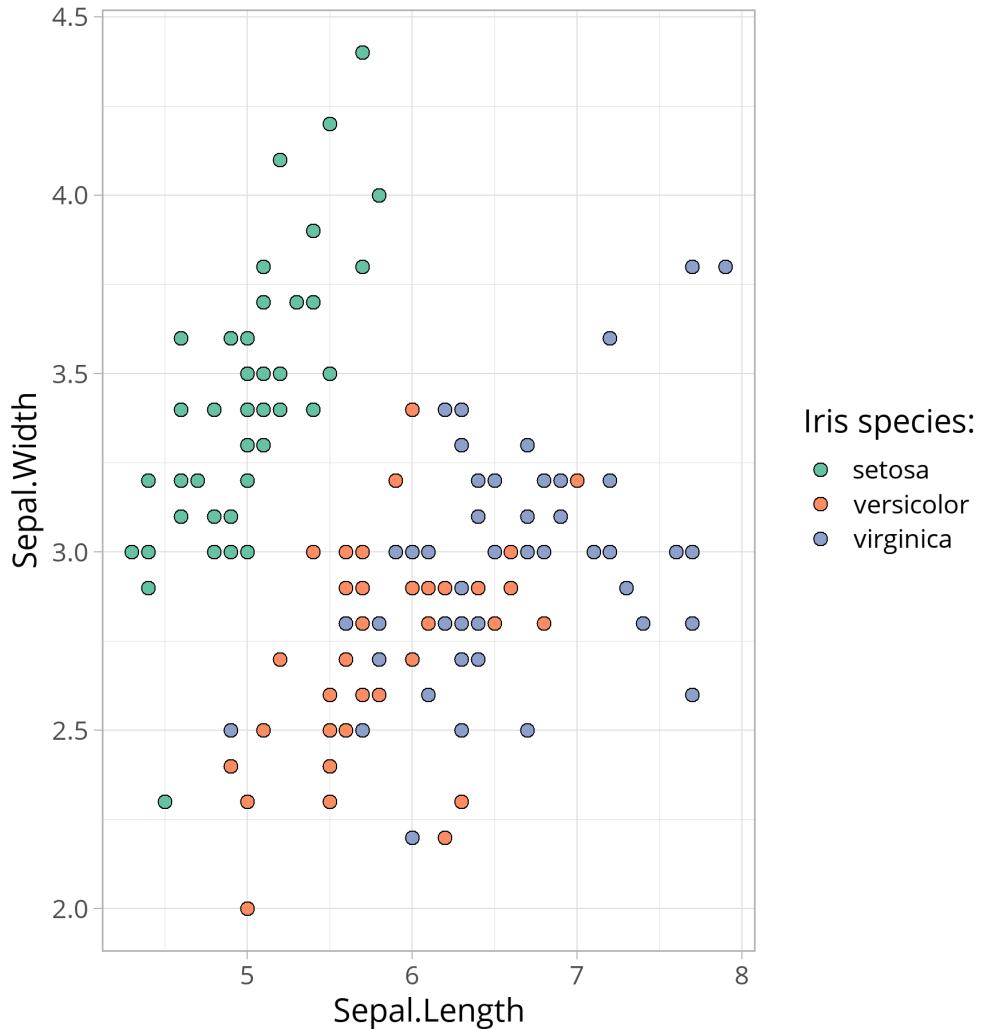
Your Turn!

```
ggplot(  
  iris,  
  aes(  
    Sepal.Length,  
    Sepal.Width  
) +  
  geom_point(  
    aes(fill = Species),  
    shape = 21,  
    color = "black",  
    size = 3  
)
```



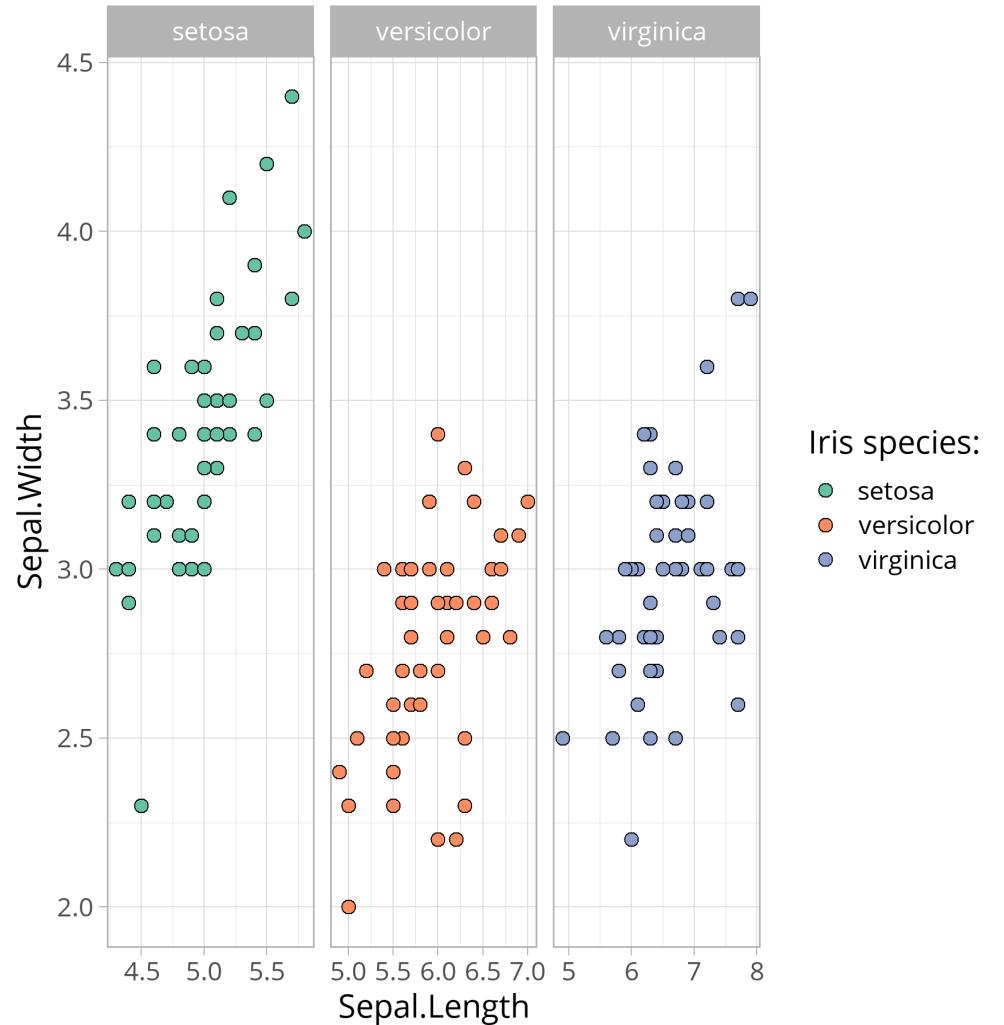
Your Turn!

```
(g <- ggplot(  
  iris,  
  aes(  
    Sepal.Length,  
    Sepal.Width  
  )) +  
  geom_point(  
    aes(fill = Species),  
    shape = 21,  
    color = "black",  
    size = 3  
  ) +  
  scale_fill_brewer(  
    palette = "Set2",  
    name = "Iris species:"  
  )  
)
```



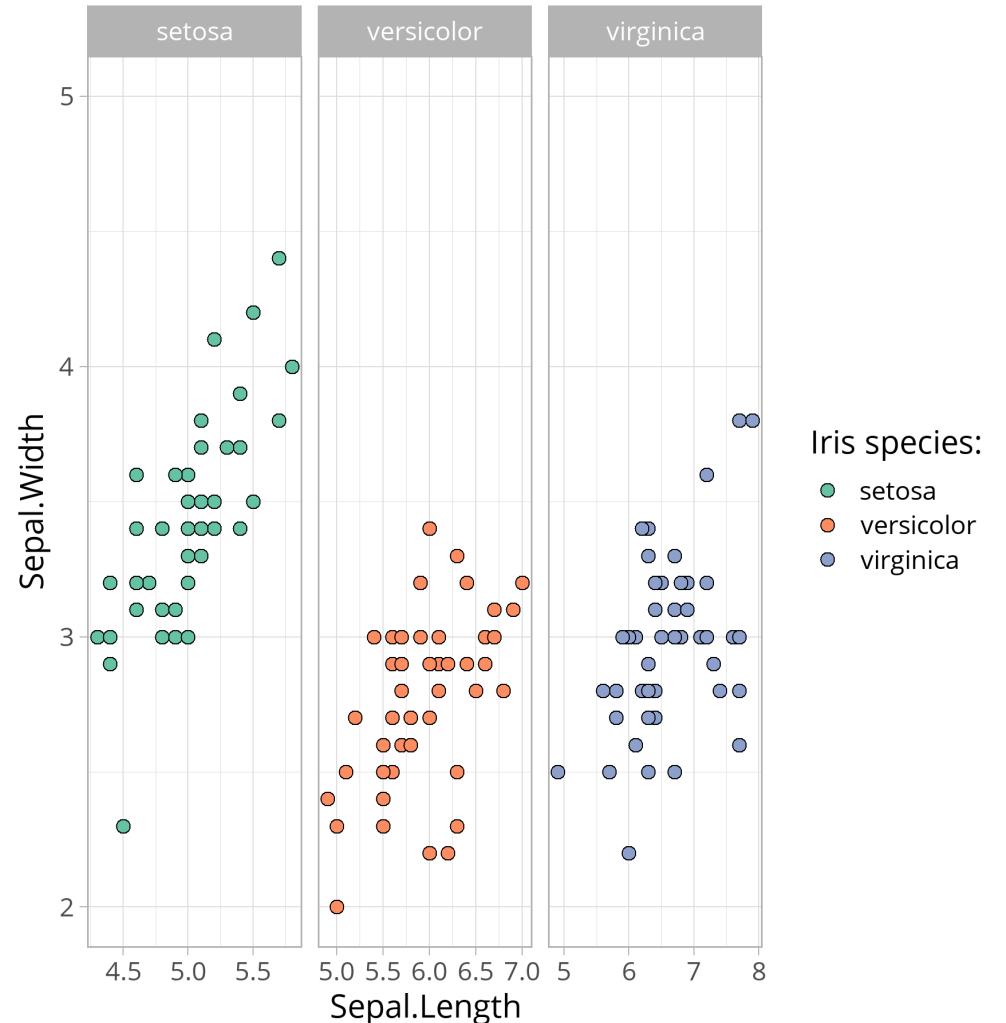
Your Turn!

```
g +
  facet_wrap(
    ~ Species,
    scales = "free_x"
  )
```

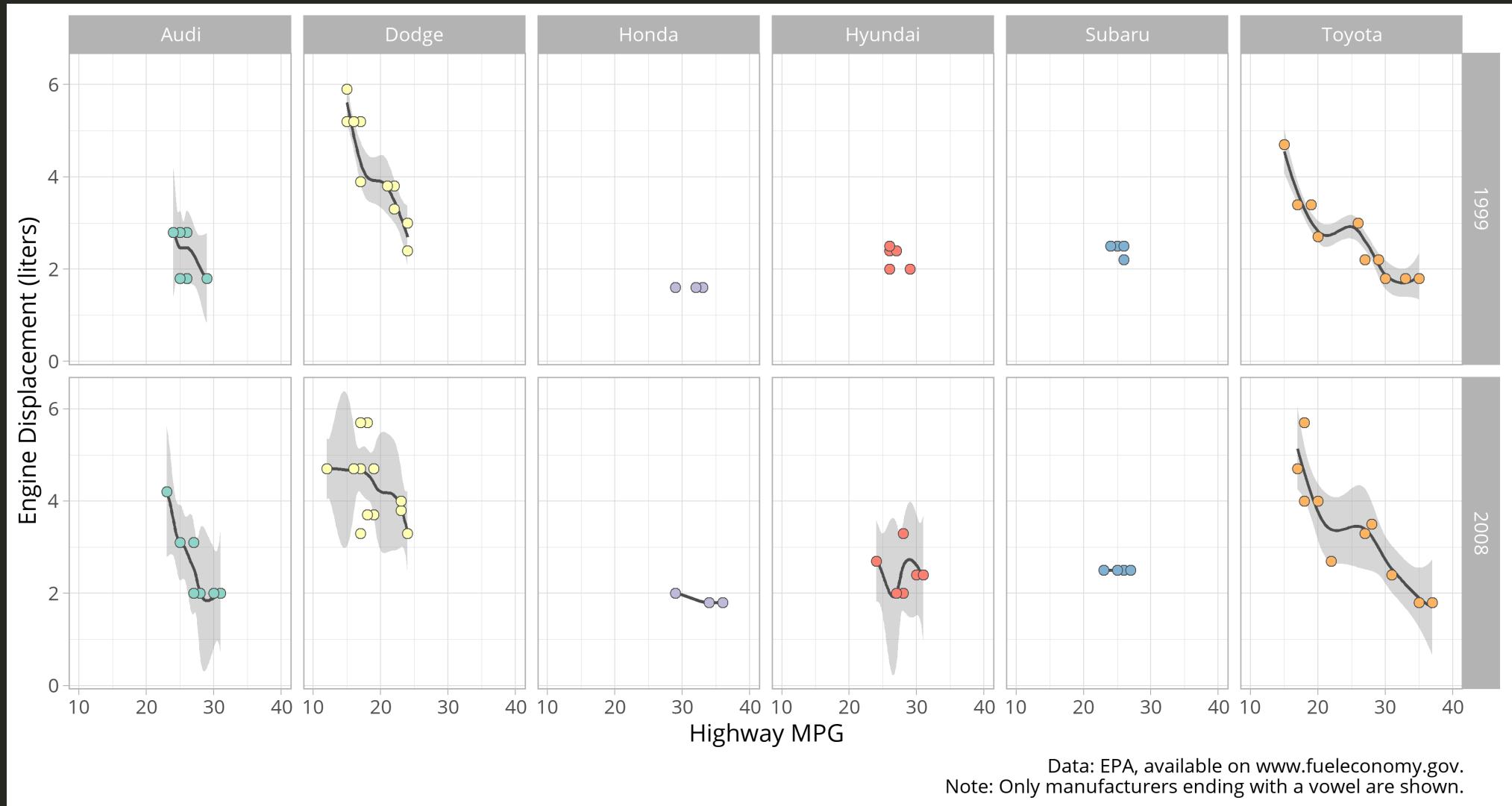


Your Turn!

```
g +
  facet_wrap(
    ~ Species,
    scales = "free_x"
  ) +
  scale_y_continuous(
    breaks = seq(2, 5, by = 1),
    limits = c(NA, 5)
  )
```



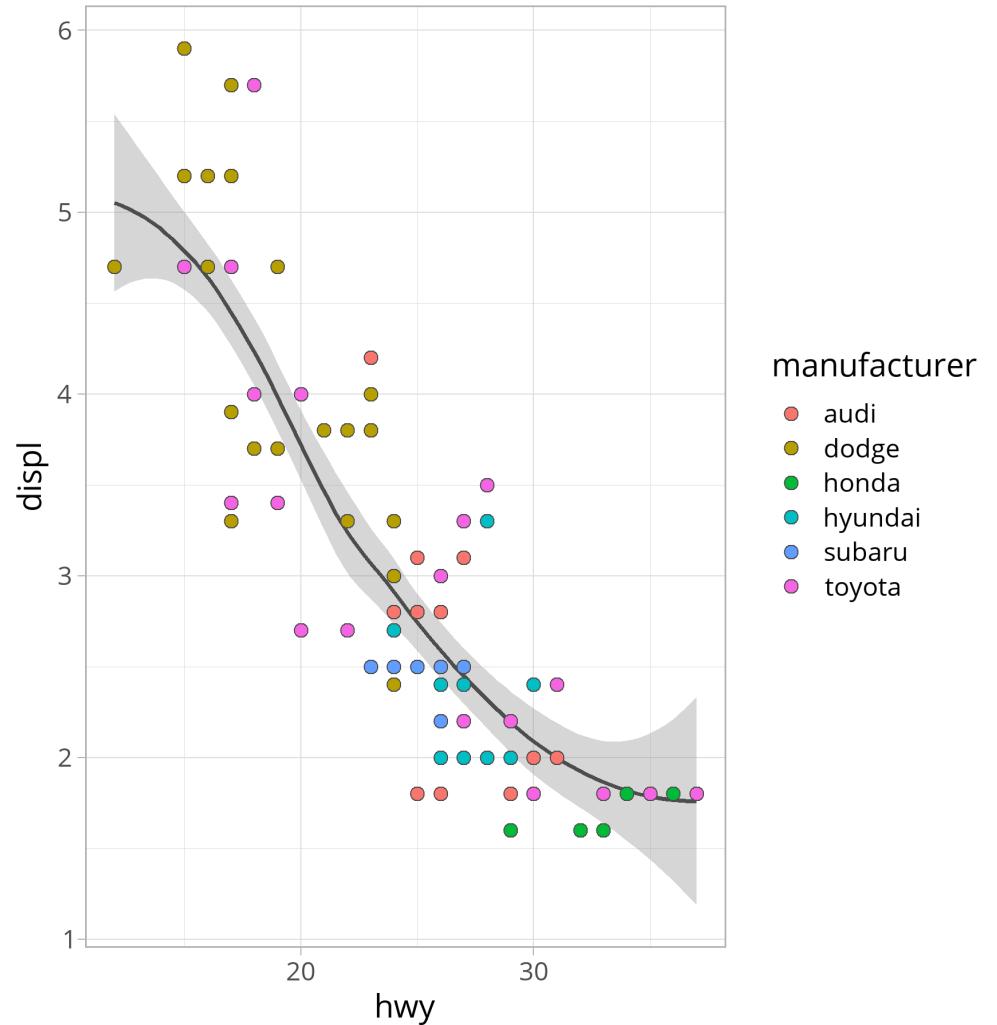
Your Turn!



Your Turn!

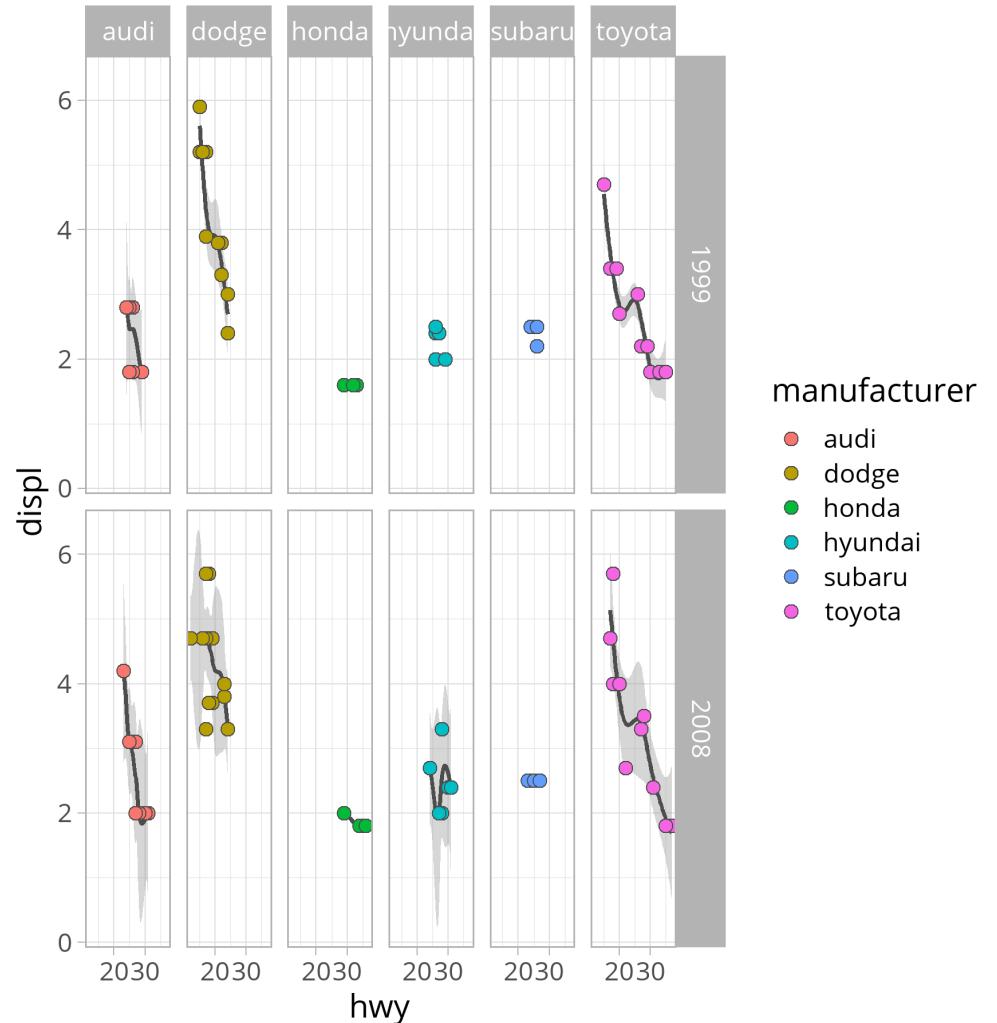
```
mpg2 <- mpg %>%
  filter(
    str_detect(manufacturer, "[aeiou]$"))
)

ggplot(mpg2, aes(hwy, displ)) +
  geom_smooth(color = "grey30") +
  geom_point(
    aes(fill = manufacturer),
    shape = 21,
    color = "grey30",
    size = 3
)
```



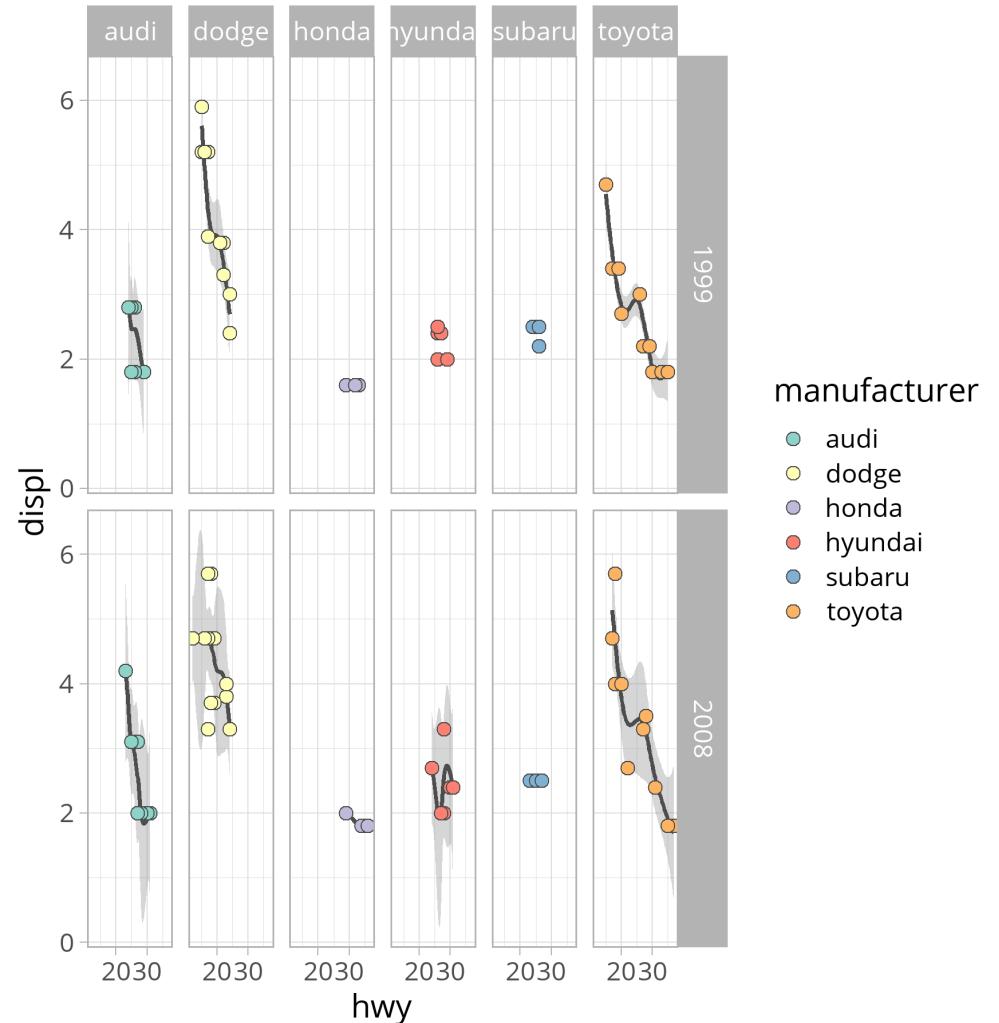
Your Turn!

```
mpg2 <- mpg %>%
  filter(
    str_detect(manufacturer, "[aeiou]$"))
ggplot(mpg2, aes(hwy, displ)) +
  geom_smooth(color = "grey30") +
  geom_point(
    aes(fill = manufacturer),
    shape = 21,
    color = "grey30",
    size = 3
  ) +
  facet_grid(year ~ manufacturer)
```



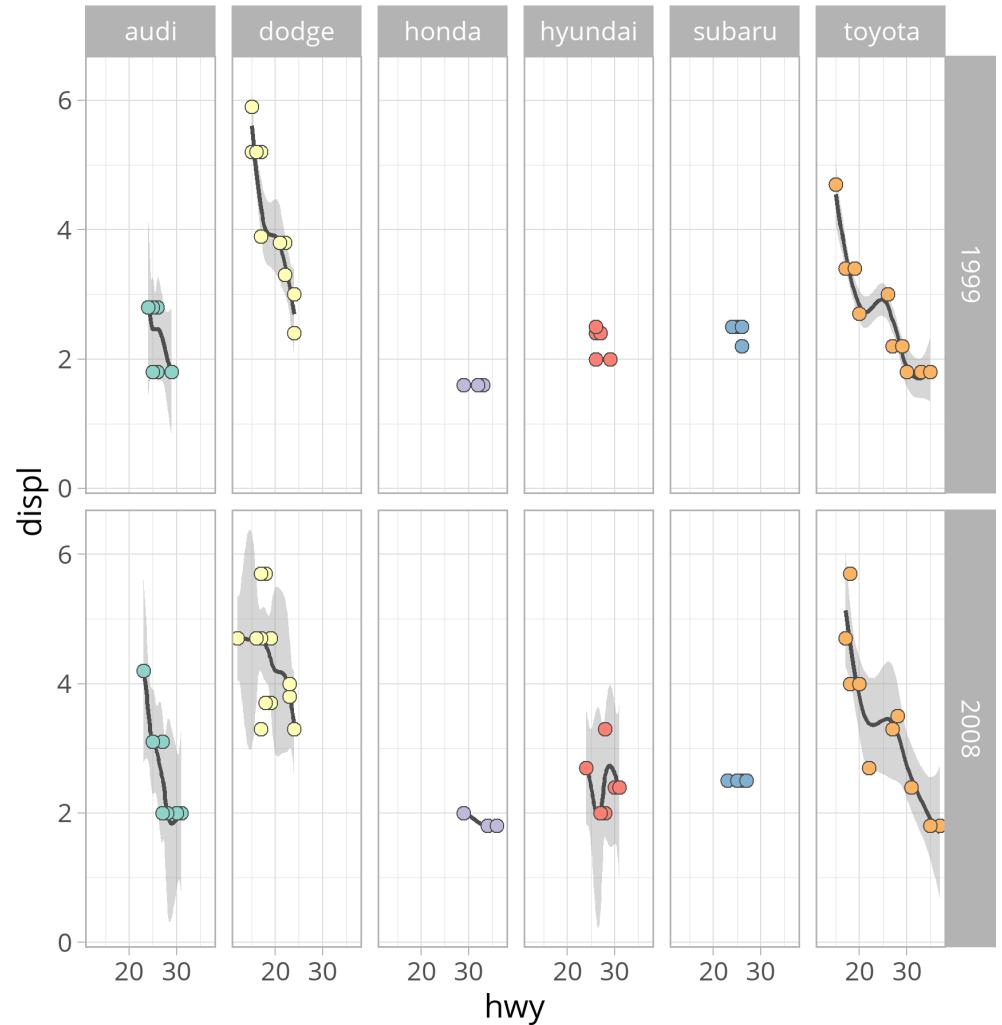
Your Turn!

```
ggplot(mpg2, aes(hwy, displ)) +  
  geom_smooth(color = "grey30") +  
  geom_point(  
    aes(fill = manufacturer),  
    shape = 21,  
    color = "grey30",  
    size = 3  
  ) +  
  facet_grid(year ~ manufacturer) +  
  scale_fill_brewer(  
    palette = "Set3"  
  )
```



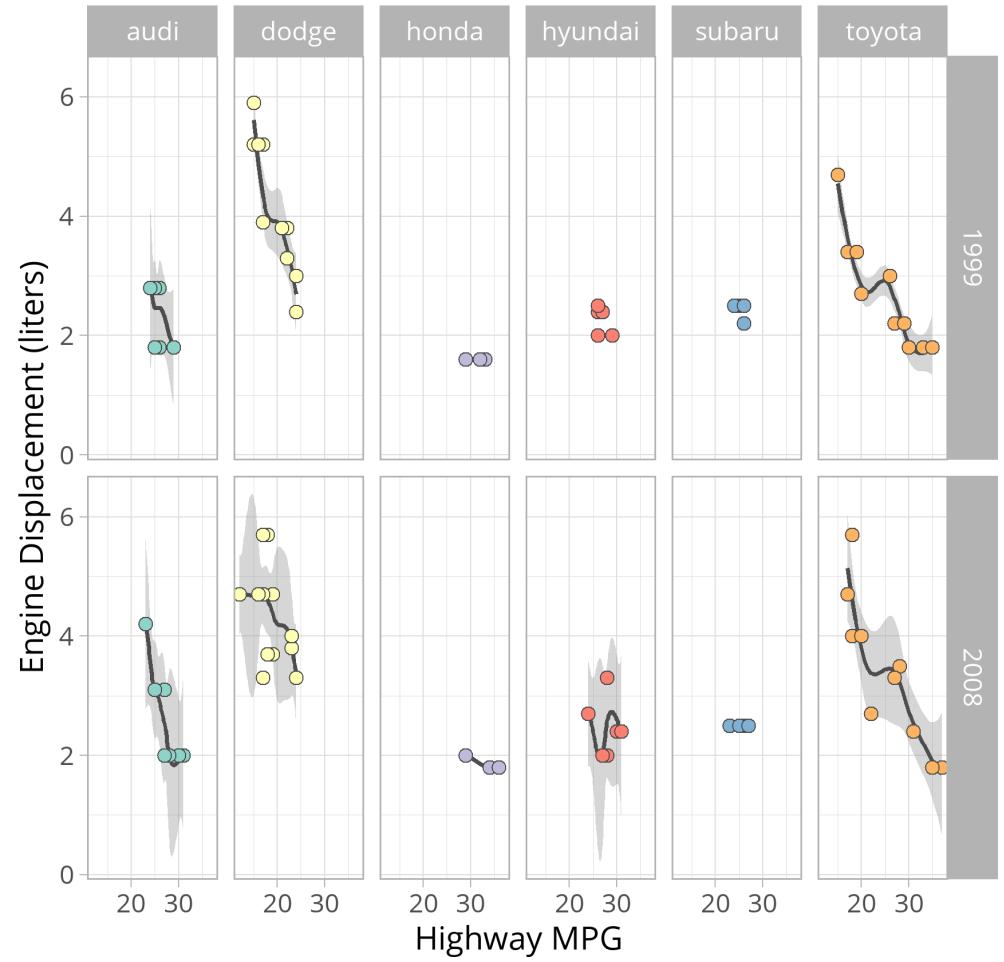
Your Turn!

```
ggplot(mpg2, aes(hwy, displ)) +  
  geom_smooth(color = "grey30") +  
  geom_point(  
    aes(fill = manufacturer),  
    shape = 21,  
    color = "grey30",  
    size = 3  
  ) +  
  facet_grid(year ~ manufacturer) +  
  scale_fill_brewer(  
    palette = "Set3",  
    guide = "none"  
)
```



Your Turn!

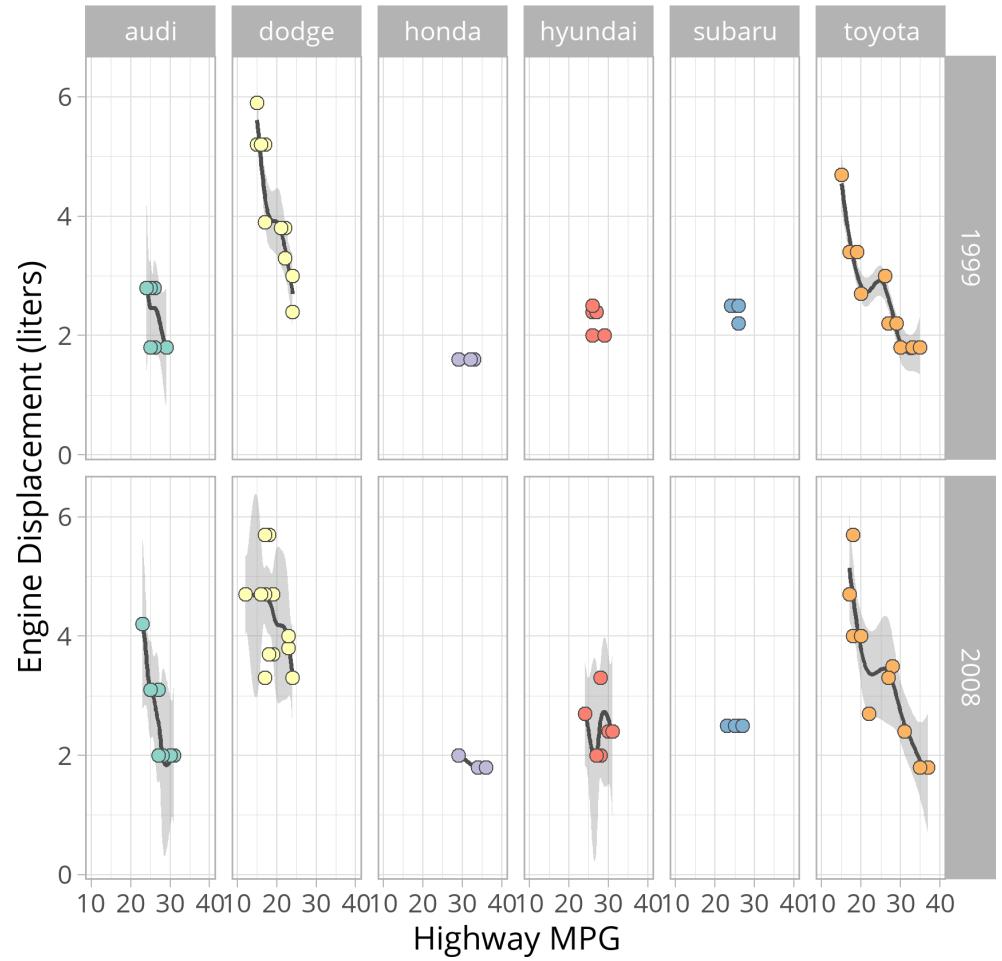
```
ggplot(mpg2, aes(hwy, displ)) +  
  geom_smooth(color = "grey30") +  
  geom_point(  
    aes(fill = manufacturer),  
    shape = 21,  
    color = "grey30",  
    size = 3  
  ) +  
  facet_grid(year ~ manufacturer) +  
  scale_fill_brewer(  
    palette = "Set3",  
    guide = "none"  
  ) +  
  labs(  
    x = "Highway MPG",  
    y = "Engine Displacement (liters)",  
    caption = "Data: EPA, available on www.fueleconomy.gov"  
  )
```



Data: EPA, available on www.fueleconomy.gov.
Note: Only manufacturers ending with a vowel are shown.

Your Turn!

```
ggplot(mpg2, aes(hwy, displ)) +  
  geom_smooth(color = "grey30") +  
  geom_point(  
    aes(fill = manufacturer),  
    shape = 21,  
    color = "grey30",  
    size = 3  
  ) +  
  facet_grid(year ~ manufacturer) +  
  scale_x_continuous(  
    breaks = seq(10, 40, by = 10),  
    limits = c(10, 40)  
  ) +  
  scale_fill_brewer(  
    palette = "Set3",  
    guide = "none"  
  ) +  
  labs(  
    x = "Highway MPG",  
    y = "Engine Displacement (liters)",  
    caption = "Data: EPA, available on www.fueleconomy.gov"  
  )
```

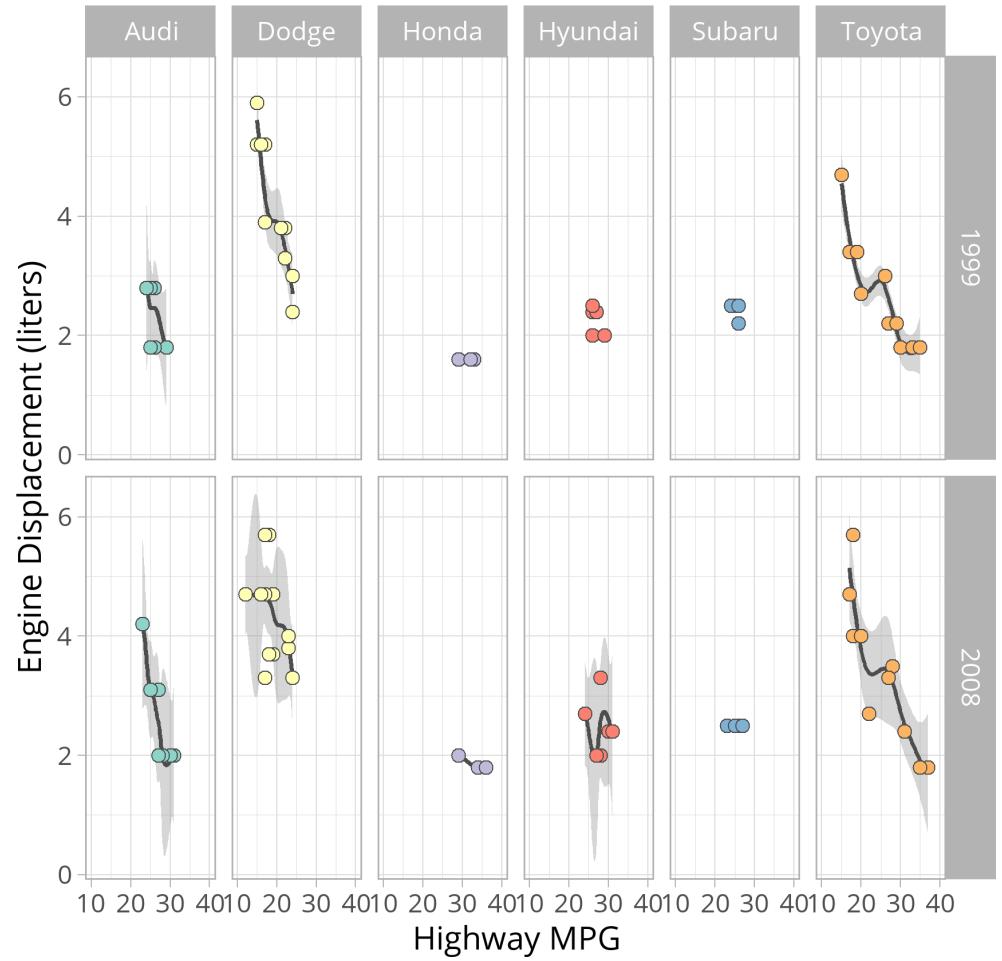


Data: EPA, available on www.fueleconomy.gov.
Note: Only manufacturers ending with a vowel are shown.

Your Turn!

```
mpg3 <- mpg %>%
  filter(
    str_detect(manufacturer, "[aeiou]$"))
  ) %>%
  mutate(manufacturer =
    str_to_title(manufacturer)
  )

ggplot(mpg3, aes(hwy, displ)) +
  geom_smooth(color = "grey30") +
  geom_point(
    aes(fill = manufacturer),
    shape = 21,
    color = "grey30",
    size = 3
  ) +
  facet_grid(year ~ manufacturer) +
  scale_x_continuous(
    breaks = seq(10, 40, by = 10),
    limits = c(10, 40)
  ) +
  scale_fill_brewer(
    palette = "Set3",
```



Data: EPA, available on www.fueleconomy.gov.
Note: Only manufacturers ending with a vowel are shown.