R ggplot2 Notes

Learning R, statistics and ggplot2

Alberto Valdez

October 13, 2022

Contents

1 MPG Dataset

Learning R with Emacs.[1] Trying to follow Google's R style guide. [2]

[1] `https://orgmode.org/worg/org-contrib/babel/languages/ob-doc-R.html`

[2] `https://web.stanford.edu/class/cs109l/unrestricted/resources/google-style.html`

```
library(ggplot2)
library(tidyverse)
```

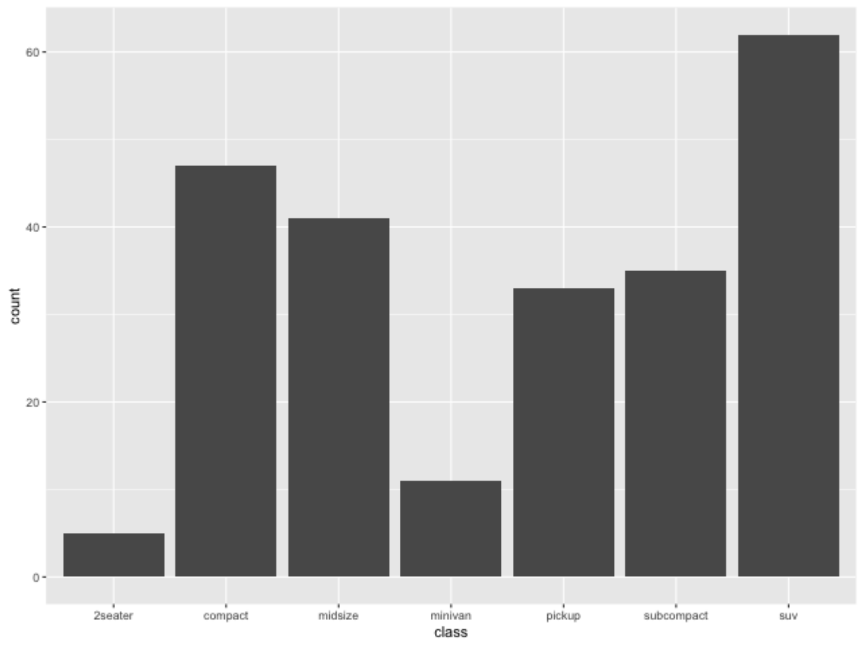
The mpg dataset contains fuel economy data from the EPA for vehicles manufactured between 1999 and 2008. The mpg dataset is built into R and is used throughout R documentation due to its availability, diversity of variables, and overall cleanliness of data. For our purposes, we'll use the mpg data to demonstrate how to implement each of our ggplot visualizations.

```
head(mpg)
```

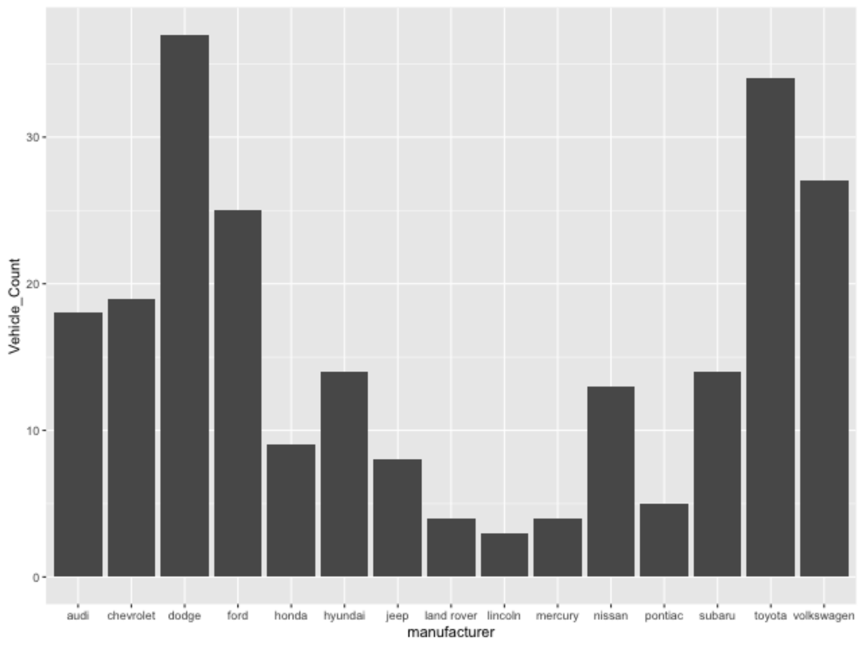
manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact

1.1 Plotting bars.

```
plt <- ggplot(mpg, aes(x=class))
plt + geom_bar()
```



```
mpg_summary <- mpg %>%
  group_by(manufacturer) %>%
  summarize(Vehicle_Count=n(), .groups = 'keep') #create summary table
plt <- ggplot(
  mpg_summary,
  aes(x=manufacturer,y=Vehicle_Count)) #import dataset into ggplot2
plt + geom_col()
```



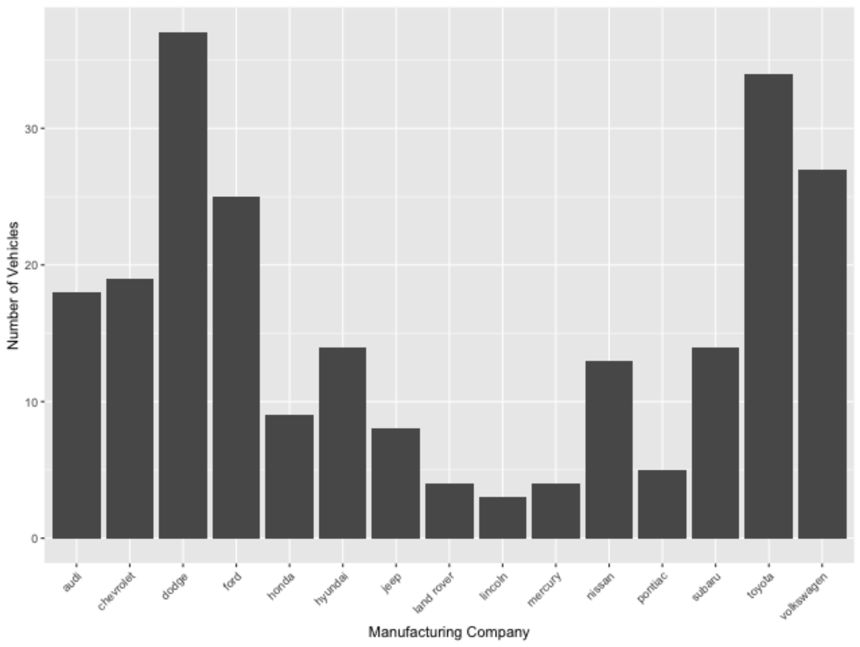
More info at the ggplot2 docs[3].

1.2 Formatting output

Adding labels and themes.

```
plt + geom_col() +
  xlab("Manufacturing Company") +
  ylab("Number of Vehicles") +
  theme(axis.text.x=element_text(angle=45, hjust=1))
```

[3]https://ggplot2.tidyverse.org/reference/index.html



2 MPG Summary

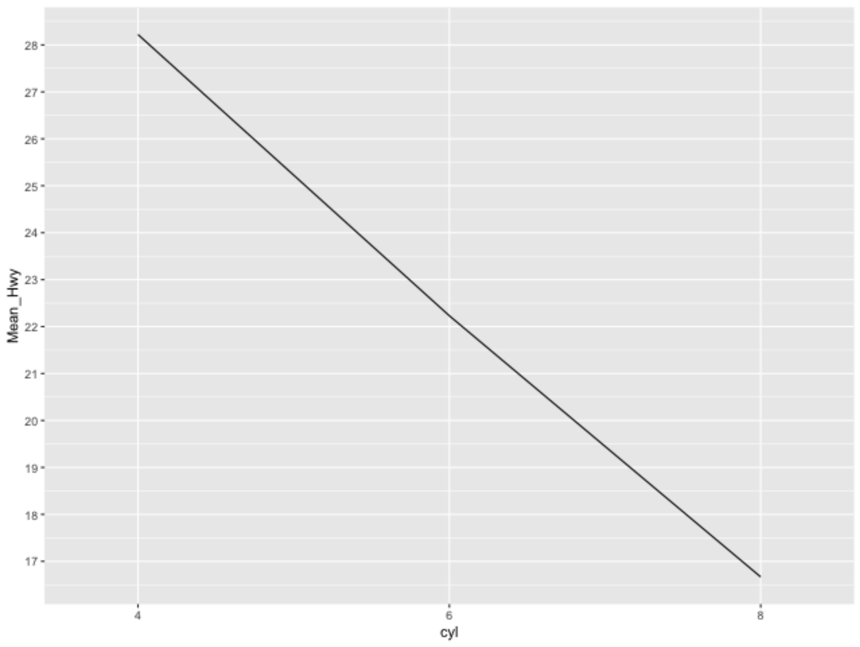
2.1 Summary Table

```
mpg_summary <-
  subset(mpg, manufacturer=="toyota") %>%
  group_by(cyl) %>%
  summarize(Mean_Hwy=mean(hwy), .groups="keep")
```

cyl	Mean_{Hwy}
4	28.2222222222222
6	22.2307692307692
8	16.6666666666667

Import dataset into ggplot and plot the data and adjust the axis.

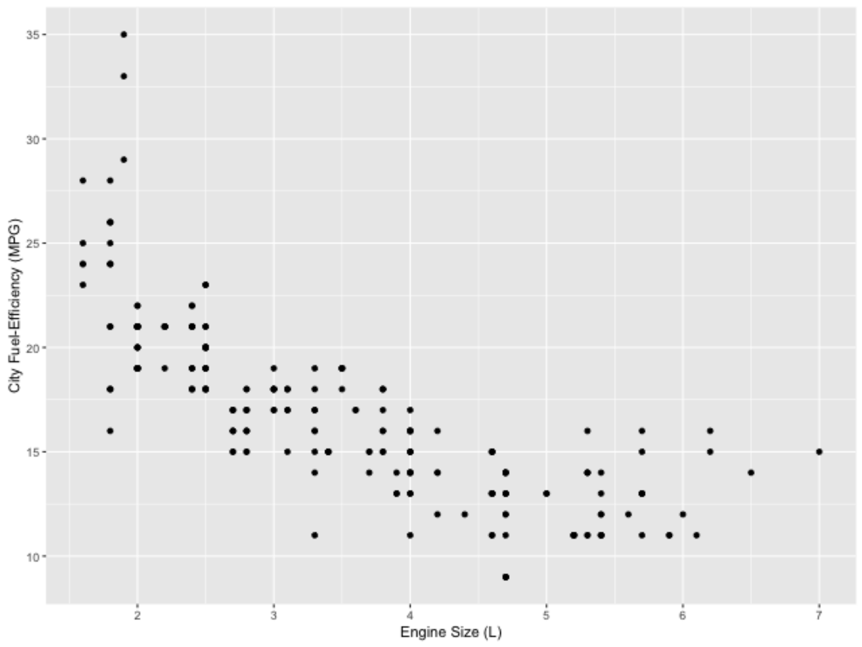
```
plt <- ggplot(mpg_summary, aes(x=cyl, y=Mean_Hwy))
plt + geom_line() +
  scale_x_discrete(limits=c(4, 6, 8)) +
  scale_y_continuous(breaks = c(15:30))
```



2.2 Mpg dataset

Import into ggplot and plot data with formatting.

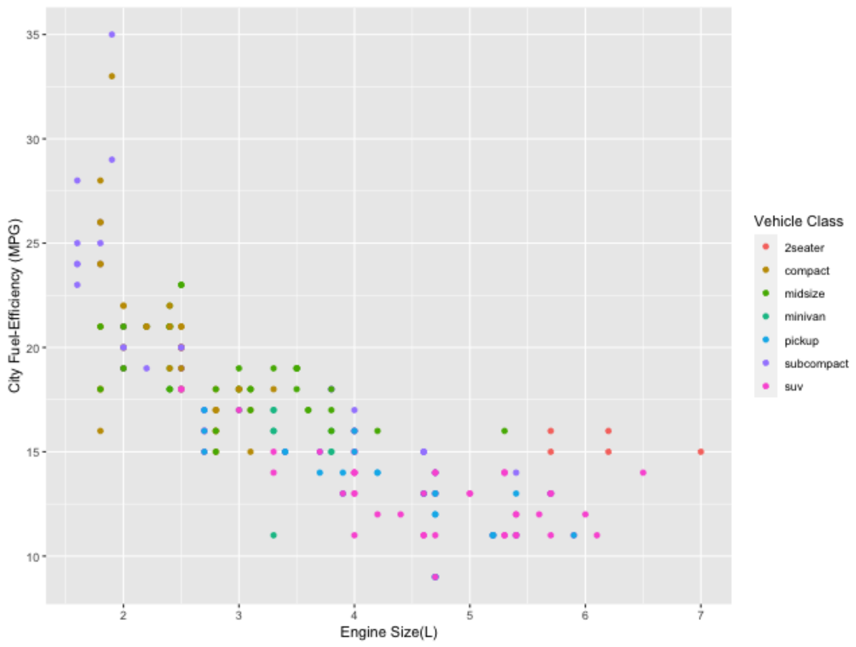
```
plt <- ggplot(mpg, aes(x=displ, y=cty))
plt + geom_point() +
  xlab("Engine Size (L)") +
  ylab("City Fuel-Efficiency (MPG)")
```



Aesthetic changes.

- alpha changes the transparency of each data point
- color changes the color of each data point
- shape changes the shape of each data point
- size changes the size of each data point

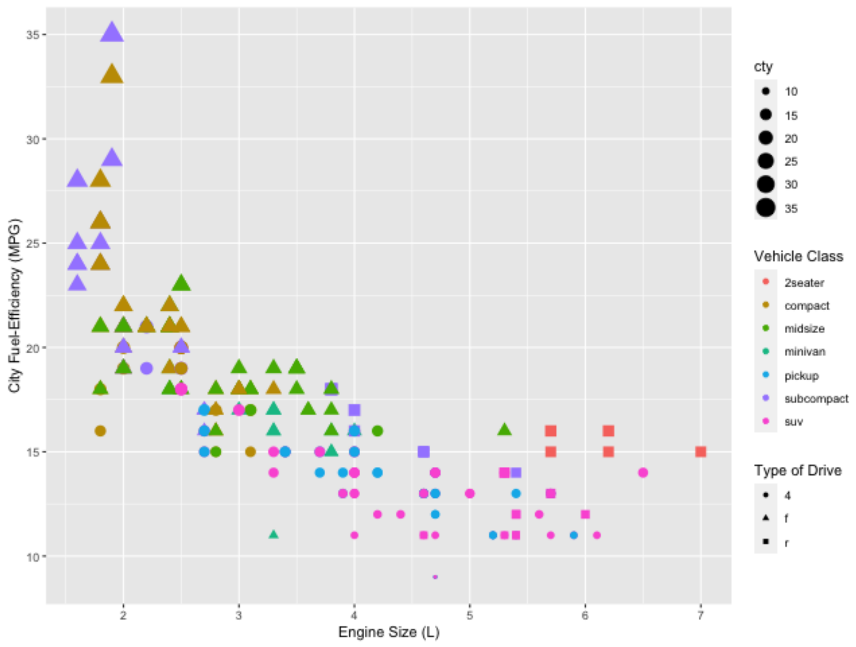
```
plt <- ggplot(mpg, aes(x=displ, y=cty, color=class))
plt + geom_point() +
  labs(
    x="Engine Size(L)",
    y="City Fuel-Efficiency (MPG)",
    color="Vehicle Class"
)
```



Different shapes.[4]

```
plt <- ggplot(
  mpg,
  aes(x=displ, y=cty, color=class, shape=drv, size=cty)
)
plt + geom_point() +
  labs(
    x="Engine Size (L)",
    y="City Fuel-Efficiency (MPG)",
    color="Vehicle Class",
    shape="Type of Drive"
)
```

[4]`https://ggplot2.tidyverse.org/reference/geom_point.html#aesthetics`



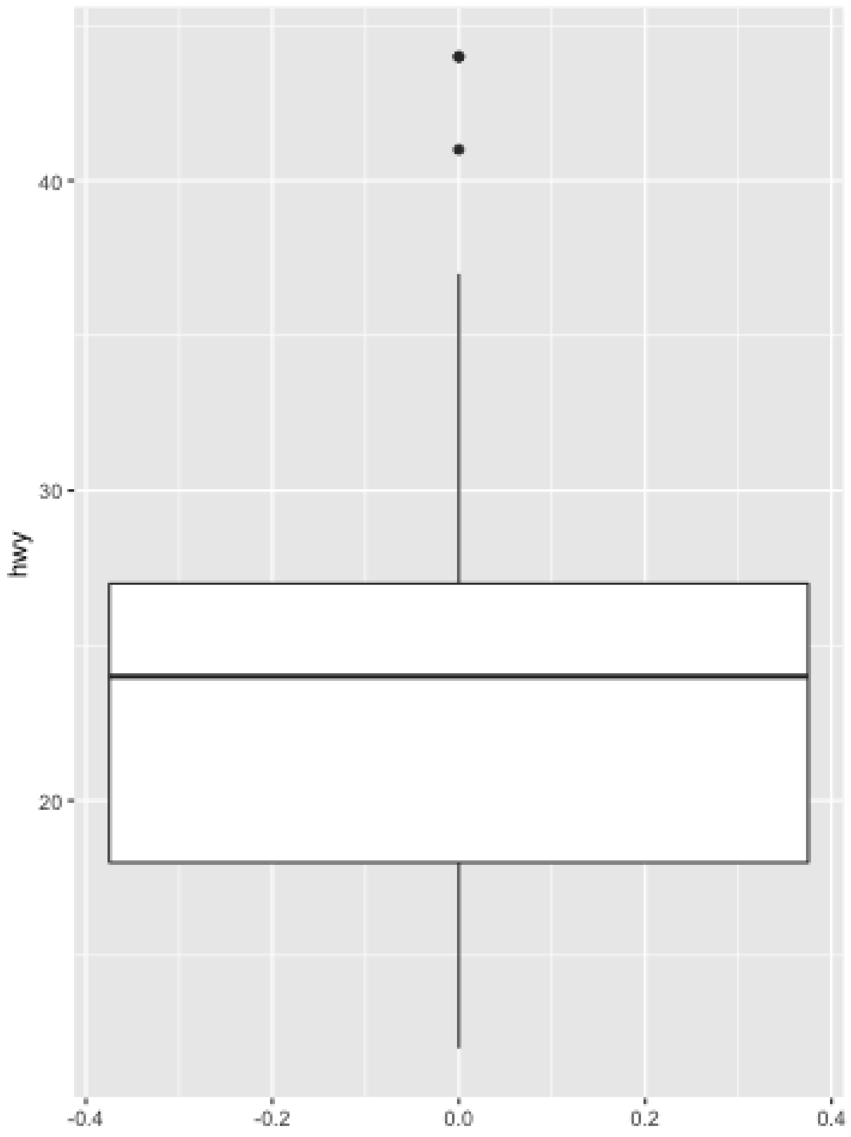
Although there is no technical limit to the number of variables we can add to a ggplot figure, there are diminishing returns. A good rule of thumb is to limit the number of variables displayed in a single figure to a maximum of 3 or 4.

3 Boxplot

Unlike the previous ggplot objects, $\text{geom}_{\text{boxplot}}$()expects a numeric vector assigned to the y-value[5].

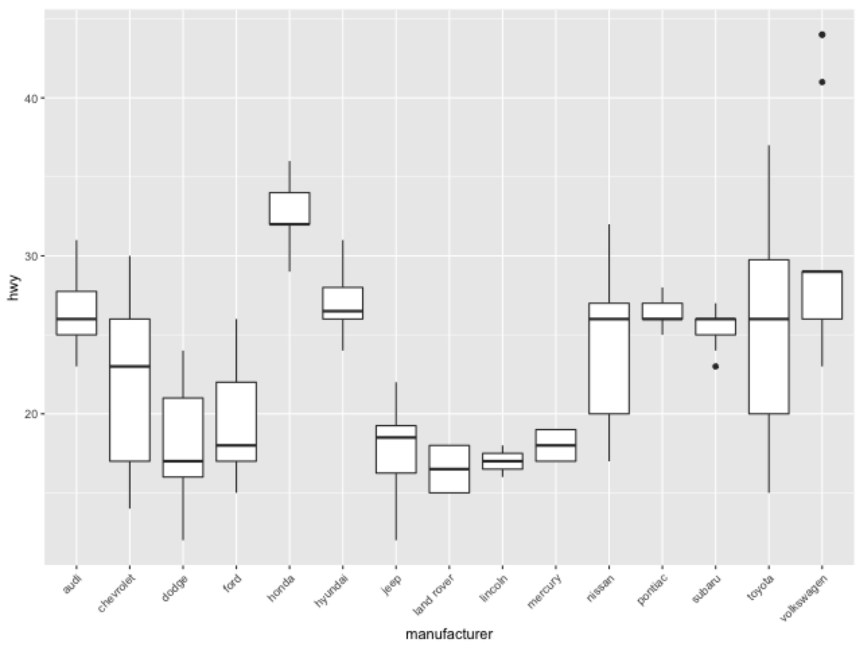
```
plt <- ggplot(mpg, aes(y=hwy))
plt + geom_boxplot()
```

[5]`https://ggplot2.tidyverse.org/reference/geom_boxplot.html#aesthetics`



Creating multiple boxes.

```
plt <- ggplot(mpg, aes(x=manufacturer, y=hwy))
plt +
  geom_boxplot() +
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



4 Heatmaps

Heatmap plots help visualize the relationship between one continuous numerical variable and two other variables (categorical or numerical).

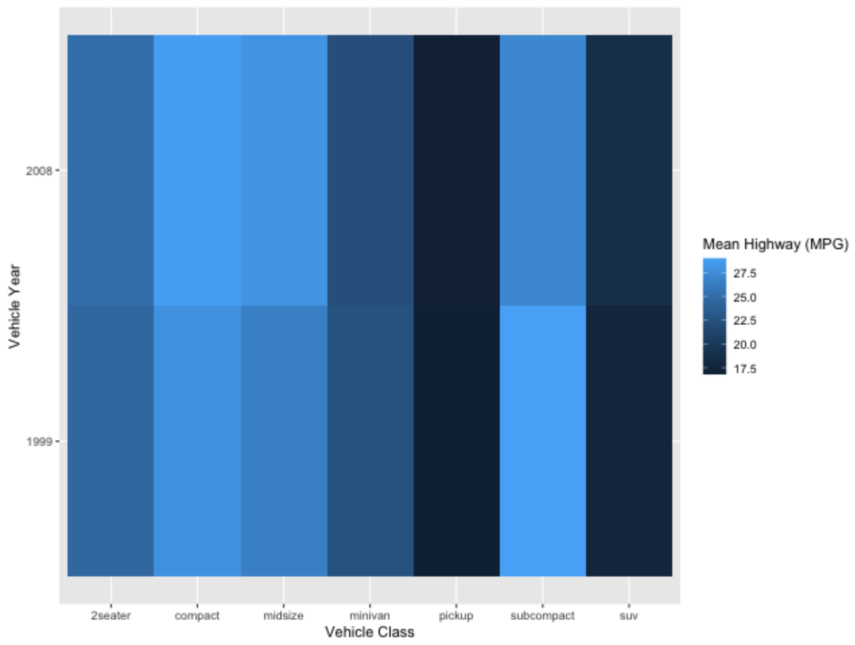
4.1 Class and Year Summary

```
mpg_summary <- mpg %>%
  group_by(class, year) %>%
  summarize(Mean_Hwy=mean(hwy), .groups='keep')
```

class	year	Mean_{Hwy}
2seater	1999	24.5
2seater	2008	25
compact	1999	27.92
compact	2008	28.7272727272727
midsize	1999	26.5
midsize	2008	28.047619047619
minivan	1999	22.5
minivan	2008	22.2
pickup	1999	16.8125
pickup	2008	16.9411764705882
subcompact	1999	29
subcompact	2008	27.125
suv	1999	17.551724137931
suv	2008	18.6363636363636

Plotting heatmap.

```
plt <- ggplot(
  mpg_summary,
  aes(x=class, y=factor(year), fill=Mean_Hwy))
plt + geom_tile() +
  labs(
    x="Vehicle Class",
    y="Vehicle Year",
    fill="Mean Highway (MPG)")
```



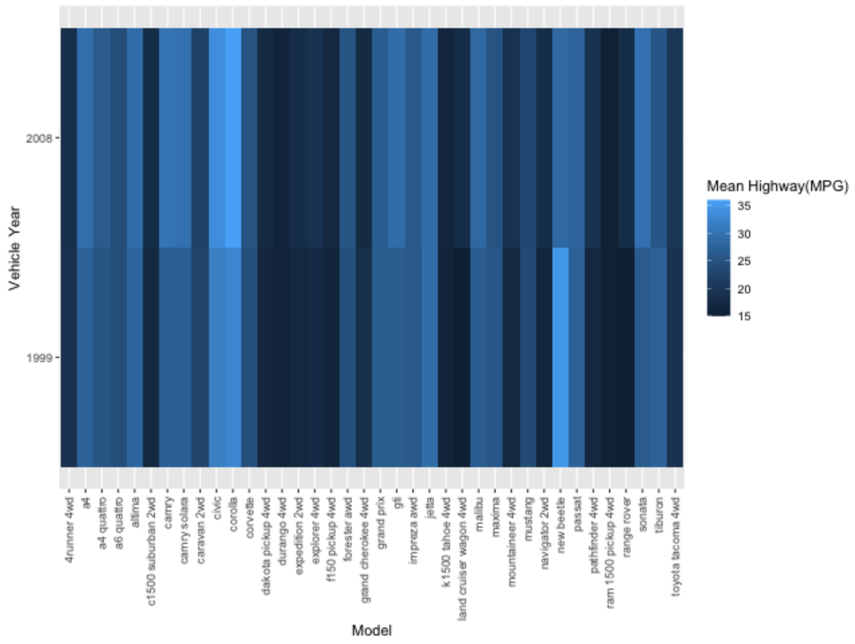
4.2 Model and Year Summary

```
mpg_summary <- mpg %>%
  group_by(model, year) %>%
  summarize(Mean_Hwy=mean(hwy), .groups='keep')
mpg_summary %>% head
```

model	year	Mean_{Hwy}
4runner 4wd	1999	19
4runner 4wd	2008	18.5
a4	1999	27.5
a4	2008	29.3333333333333
a4 quattro	1999	25.25
a4 quattro	2008	26.25

Adding labels to heatmap.

```
plt <- ggplot(
  mpg_summary,
  aes(
    x=model,
    y=factor(year),
    fill=Mean_Hwy)
)
plt + geom_tile() +
  labs(
    x="Model",
    y="Vehicle Year",
    fill="Mean Highway(MPG)"
) +
  theme(
    axis.text.x = element_text(
      angle=90,
      hjust=1,
      vjust=0.5
    )
)
```



We can always refer to the ggplot cheatsheet[6].

[6]`https://github.com/rstudio/cheatsheets/blob/main/data-visualization.pdf`

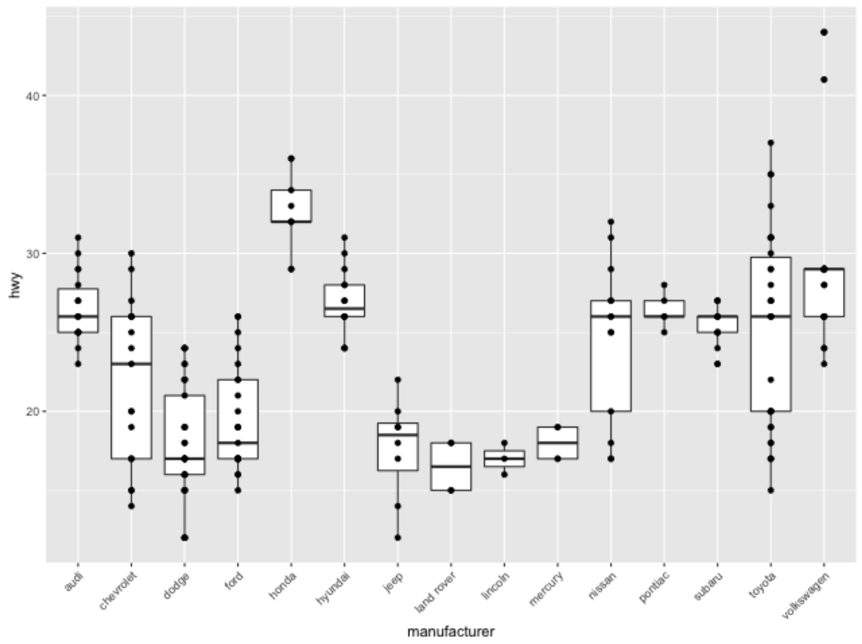
5 Layered Plots

There are two types of plot layers:

1. Layering additional plots that use the same variables and input data as the original plot
2. Layering of additional plots that use different but complementary data to the original plot

```
plt <- ggplot(mpg, aes(x=manufacturer,y=hwy))
```

```
plt + geom_boxplot() +
  theme(axis.text.x=element_text(angle=45,hjust=1)) +
  geom_point()
```



By layering our data points on top of our boxplot, we can see the general distribution of values within each box as well as the number of data points.

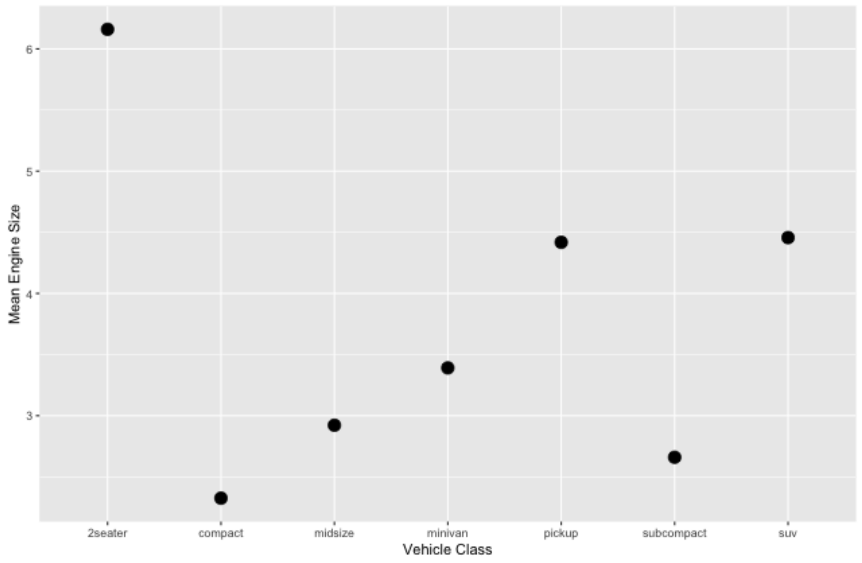
5.1 Summary of Class

```
mpg_summary <- mpg %>%
  group_by(class) %>%
  summarize(Mean_Engine=mean(displ), .groups='keep')
```

class	$\text{Mean}_{\text{Engine}}$
2seater	6.16
compact	2.32553191489362
midsize	2.9219512195122
minivan	3.39090909090909
pickup	4.41818181818182
subcompact	2.66
suv	4.45645161290323

Plotting scatter plot.

```
plt <-
  ggplot(mpg_summary, aes(x=class,y=Mean_Engine))
plt +
  geom_point(size=4) +
  labs(x="Vehicle Class",y="Mean Engine Size")
```



5.2 Plotting Error Bars

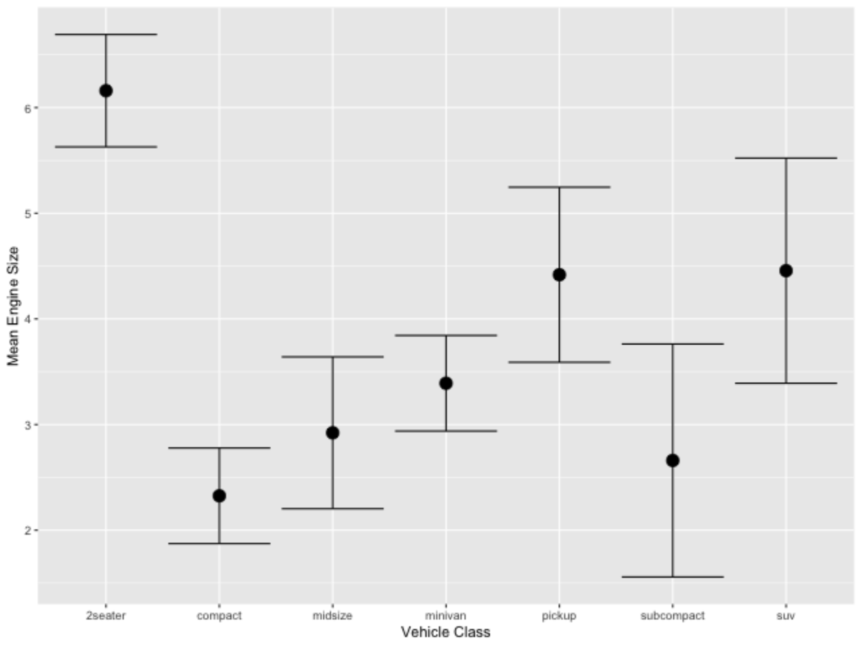
Summary of Mean and Standard Deviation of Vehicle Class.

```
mpg_summary <-
  mpg %>%
  group_by(class) %>%
  summarize(Mean_Engine=mean(displ), SD_Engine=sd(displ), .groups='keep')
```

class	$\text{Mean}_{\text{Engine}}$	$\text{SD}_{\text{Engine}}$
2seater	6.16	0.531977443130815
compact	2.32553191489362	0.452273524927782
midsize	2.9219512195122	0.71850963637308
minivan	3.39090909090909	0.452668853478004
pickup	4.41818181818182	0.828573527762679
subcompact	2.66	1.10245714869372
suv	4.45645161290323	1.06580547047714

Plotting.

```
plt <-
  ggplot(mpg_summary, aes(x=class,y=Mean_Engine))
plt +
  geom_point(size=4) +
  labs(x="Vehicle Class", y="Mean Engine Size") +
  geom_errorbar(aes(ymin=Mean_Engine-SD_Engine, ymax=Mean_Engine+SD_Engine))
```



5.3 Faceting

Often when our data is in a long format, we want to avoid visualizing all data within a single plot. Rather, we want to plot all our measurements but keep each level (or category) of our grouping variable separate.

Using gather for converting to long format.[7]

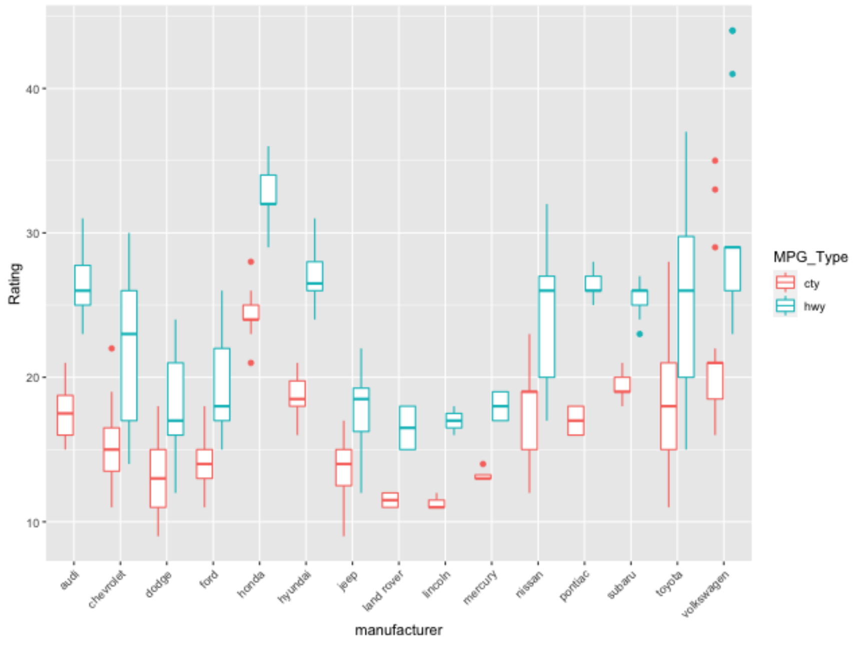
```
mpg_long <-
  mpg %>% gather(key="MPG_Type", value="Rating", c(cty,hwy))
head(mpg_long)
```

manufacturer	model	displ	year	cyl	trans	drv	fl	class	MPG_{Type}	Rating
audi	a4	1.8	1999	4	auto(l5)	f	p	compact	cty	18
audi	a4	1.8	1999	4	manual(m5)	f	p	compact	cty	21
audi	a4	2	2008	4	manual(m6)	f	p	compact	cty	20
audi	a4	2	2008	4	auto(av)	f	p	compact	cty	21
audi	a4	2.8	1999	6	auto(l5)	f	p	compact	cty	16
audi	a4	2.8	1999	6	manual(m5)	f	p	compact	cty	18

Plotting many boxplots and coloring by MG type.

[7] `https://tidyr.tidyverse.org/reference/gather.html`

```
plt <-
  ggplot(mpg_long, aes(x=manufacturer,y=Rating,color=MPG_Type))
plt +
  geom_boxplot() +
  theme(axis.text.x=element_text(angle=45,hjust=1))
```



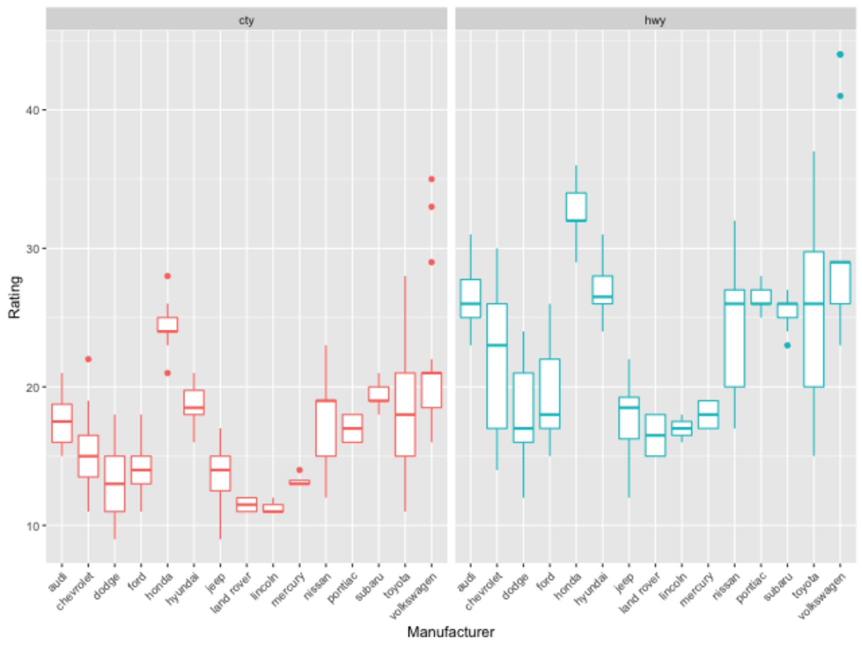
One solution would be to facet the different types of fuel efficiency within the visualization using the facet$_{\text{wrap}}$() function[8].

5.4 Facet Wrap

By faceting our boxplots by fuel-efficiency type, it's easier to make comparisons across manufacturers.

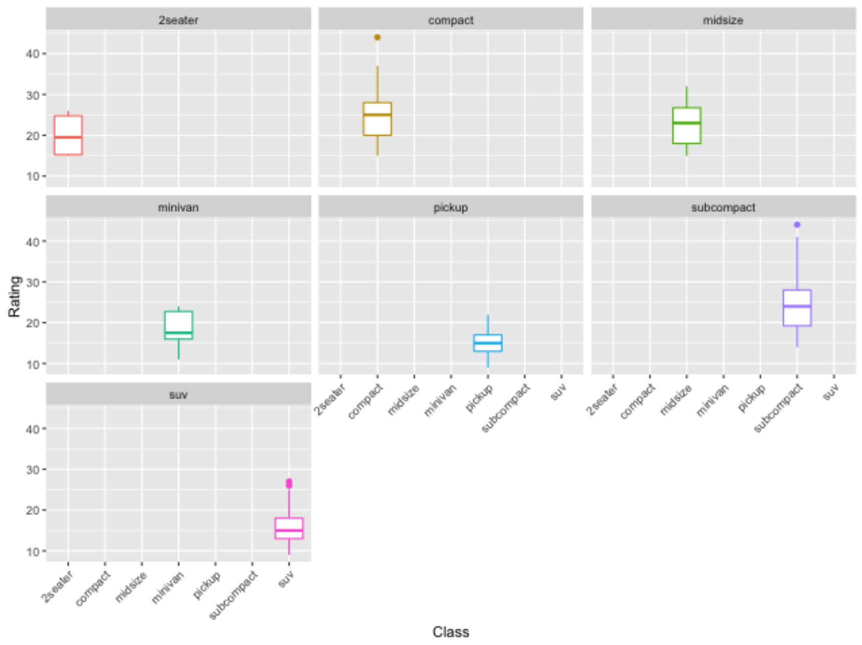
```
plt <-
  ggplot(mpg_long,aes(x=manufacturer,y=Rating,color=MPG_Type))
plt +
  geom_boxplot() +
  facet_wrap(vars(MPG_Type)) +
  theme(axis.text.x=element_text(angle=45,hjust=1), legend.position = "none") +
  xlab("Manufacturer")
```

[8]https://ggplot2-book.org/facet.html



Using multiple variables for facet wrap can lead to too many different figures.

```
plt <-
  ggplot(mpg_long, aes(x=class, y=Rating, color=class))
plt +
  geom_boxplot() +
  facet_wrap(vars(class)) +
  theme(
    axis.text.x=element_text(angle=45,hjust=1),
    legend.position = "none") +
  xlab("Class")
```


2seater
compact
midsize
minivan
pickup
subcompact
suv
Rating
10
20
30
40
Class
2seater
compact
midsize
minivan
pickup
subcompact
suv