Python Graphics

CMSC 201 Section 60
November 25, 2024

Administrative Notes

Project 3 is due December 9

- No extensions except for a true emergency
- If you don’t have Jupyter Notebook working on your laptop by now, you need to rectify that immediately

The final is Monday, December 16, 3:30 - 5:30 in this room.

- If that is a problem for you, let me know

REMINDER: THERE IS NO CLASS THIS WEDNESDAY BECAUSE IT’S THE EVENING BEFORE THANKSGIVING

A general note first, and then some specifics that relate to Project 3

ggplot

The “ggplot” idea started in the R programming language by a guy named Hadley Wickham

- It was why a lot of data analytics people used R instead of Python
- So python had to come up with its own version of ggplot
    - Plotnine is the best implementation

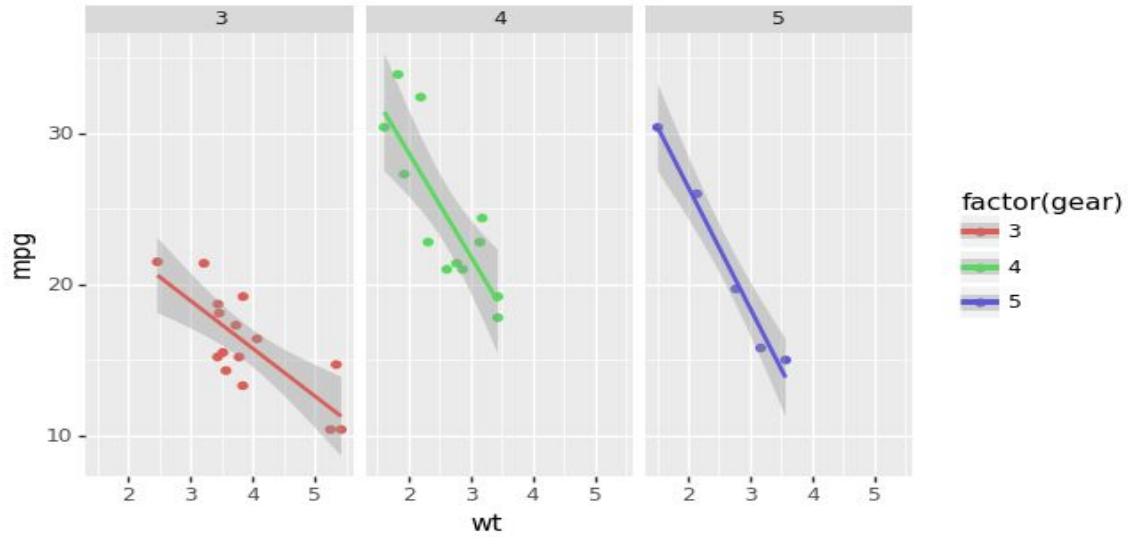
A “grammar of graphics”

- First, you have a basic plot object - ‘ggplot’
- Then add an ‘aesthetic’ - aes
  - Labels for the X and Y axes
  - Colors to use for points
- Then add a ‘geometry’ - geom
  - Points
  - Bar charts
  - Stacked parts
  - Histograms…
- Then transform the data using statistical methods
  -

Plotnine examples

Example

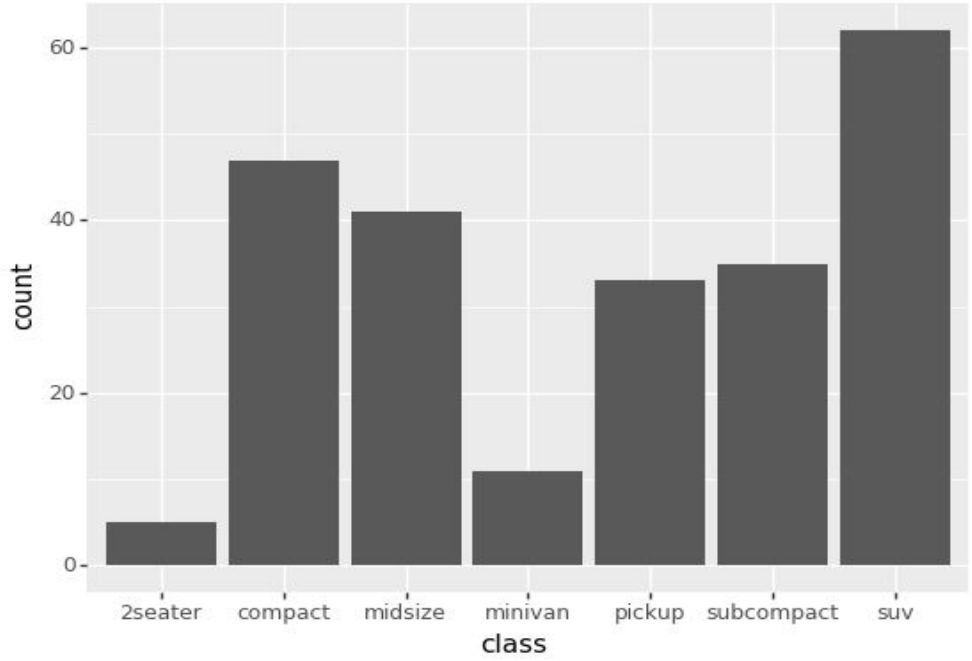
```
from plotnine import ggplot, geom_point, aes, stat_smooth, facet_wrap
from plotnine.data import mtcars

(ggplot(mtcars, aes('wt', 'mpg', color='factor(gear)'))
 + geom_point()
 + stat_smooth(method='lm')
 + facet_wrap('~gear'))
```



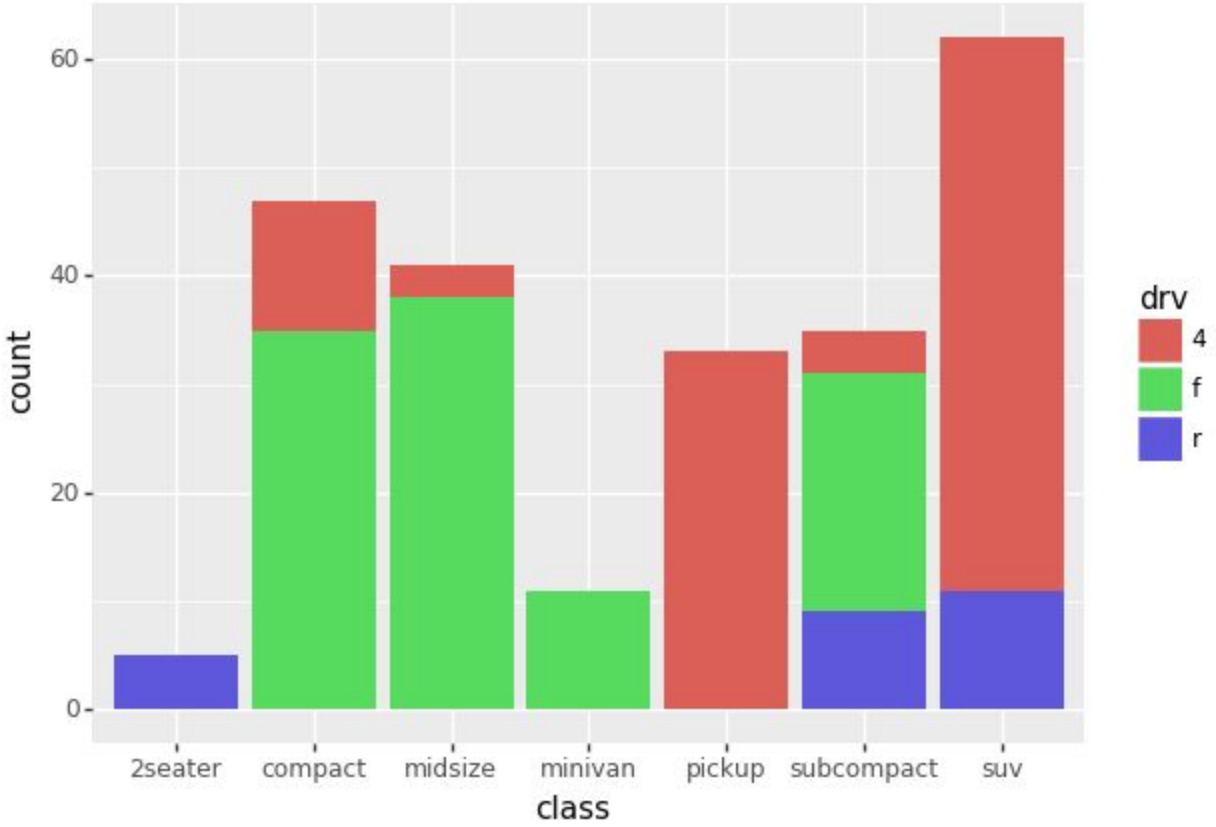
[2]:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact

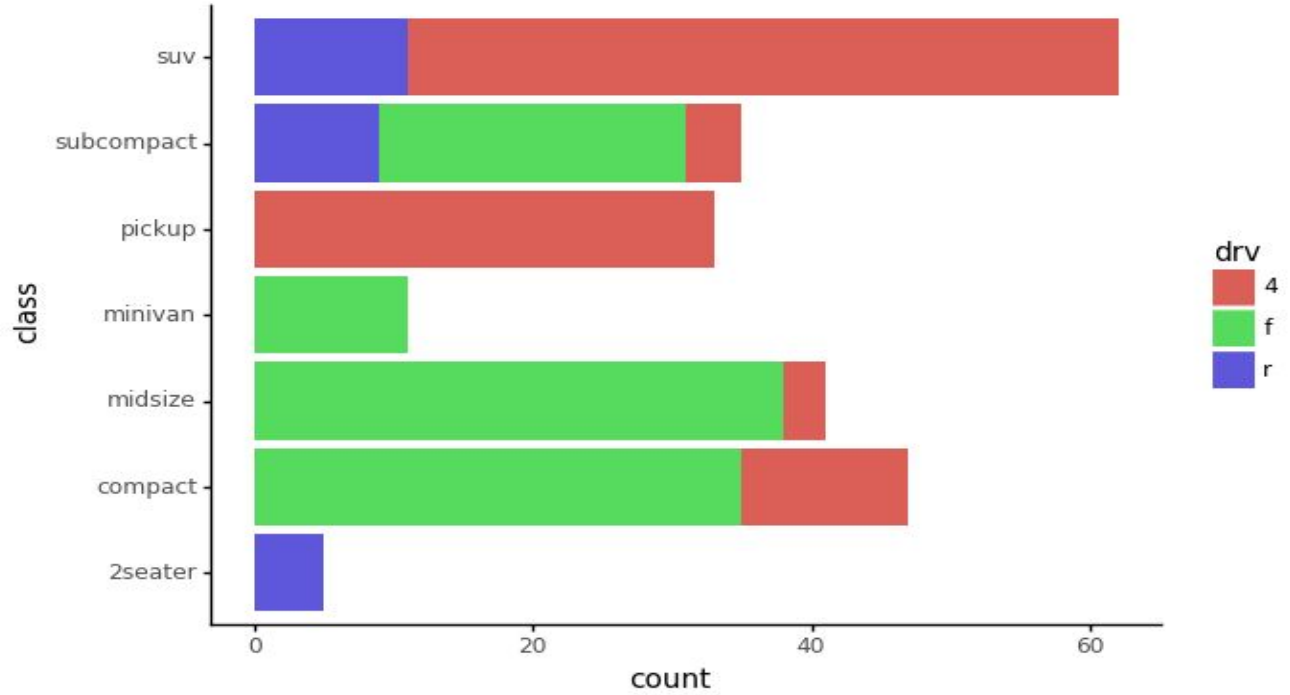
[3]:
```
ggplot(mpg) + geom_bar(aes(x='class'))
```



```
ggplot(mpg) + geom_bar(aes(x='class', fill='drv'))
```



```
(
    ggplot(mpg)
    + geom_bar(aes(x='class', fill='drv'))
    + coord_flip()
    + theme_classic()
)
```



```
water_color = '#a3ccff'
wall_color = 'white'
road_color = 'brown'

# Create label text by merging the territory name and
# the claimant to the territory
def fmt_labels(names, claimants):
    labels = []
    for name, claimant in zip(names, claimants):
        if name:
            labels.append('{} ({})'.format(name, claimant))
        else:
            labels.append('({})'.format(claimant))
    return labels

(ggplot()
 + geom_map(westeros, fill=None)
 + geom_map(islands, fill=None)
 + geom_map(political, aes(fill='ClaimedBy'), color=None, show_legend=False)
 + geom_map(wall, draw='LineString', color=wall_color, size=2)
 + geom_map(lakes, fill=water_color, color=None)
 + geom_map(rivers, aes(size='size'), draw='LineString', color=water_color, show_legend=False)
 + geom_map(roads, aes(size='size'), draw='LineString', color=road_color, alpha=0.5, show_legend=False)
 + geom_map(cities, draw='Point', size=1)
 + geom_text(
     political,
     aes('geometry.centroid.x', 'geometry.centroid.y', label='fmt_labels(name, ClaimedBy)'),
     size=8,
     fontweight='bold'
 )
 + geom_text(
     cities,
     aes('geometry.centroid.x', 'geometry.centroid.y', label='name'),
     size=8,
     ha='left',
     nudge_x=.20
 )
 + labs(title="The Political Territories of Westeros")
 + scale_fill_brewer(type='qual', palette=8)
 + scale_x_continuous(expand=(0, 0, 0, 1))
 + scale_y_continuous(expand=(0, 1, 0, 0))
 + scale_size_continuous(range=(0.4, 1))
 + theme_void()
 + theme(figure_size=(8, 12), panel_background=element_rect(fill=water_color))
)
```

One for the chemists and chemical engineers...

https://plotnine.readthedocs.io/en/latest/generated/plotnine.geoms.geom_tile.html#periodic-table-of-elements

Now some specifics related to Project 3

Code

The code we are going through now is on github under “Code_samples.”

It’s the file called “illustrate_graphics.py”