Python Graphics

April 22, 2020

Administrative Notes

Lab 9 - make sure you can copy the files hail.py and fib.py on gl.umbc.edu:

```
cp /afs/umbc.edu/users/a/r/arsenaul/pub/hail.py hail.py

cp /afs/umbc.edu/users/a/r/arsenaul/pub/fib.py  fib.py
```

Homework 9 due next Monday

Project 2 due next Monday

More about Homework 6

Part 3 asks you to reverse the characters in a string

- But I warned you that you can't directly manipulate the elements of the string, because strings are immutable
  - s = 'UMBC Retrievers'

You might want to try:

```
temp = s[0]
s[0] = s[-1]
s[-1] = temp
```

But you can't do that!!

So you have to split the string into a list - e.g.,
l = ['U','M', 'B', 'C', …]
So you can swap list elements because lists are mutable.
How? It doesn't seem like you can use s.split() (Why not?)

Think about slicing off one character at a time.

- Start with an empty list
- Take one character of the string, and append it to the list

Downloading and installing Python modules

Generally, use pip (Package Installer for Python)

In a Windows command window/Mac terminal/Linux command line: type

pip install <name of package>

Today we're going to talk about plotnine, so:

pip install plotnine

Graphics

“Graphics” encompasses a great number of things. Animation, like Pixar movies. Plots and charts. Drawing simple figures.

Fortunately, Python provides tools for doing whatever you want.

- “Turtle” graphics for drawing simple figures
  - import turtle
- “Plotnine” or “ggplot” for charts and plots to visualize and analyze data
  - That’s what we’re going to cover today
  - You might have to install plotnine on your computer - remember
    - pip install plotnine

Turtle graphics

The idea is that you create a new, blank window. You command a “turtle” by telling it to move a certain number of pixels in a certain direction, then turn a certain number of degrees, then move a certain number of pixels, etc.

You can draw simple shapes this way

You can “fill” the shapes with color if you actually draw a closed shape

You can change the color and thickness of the line

You can skip drawing some parts of the figure by lifting the turtle up from the screen; putting the turtle down when you’re ready to draw

Turtle graphics - some examples

gnuplot

A generic plotting library that can be called by Python routines

http://www.gnuplot.info/

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

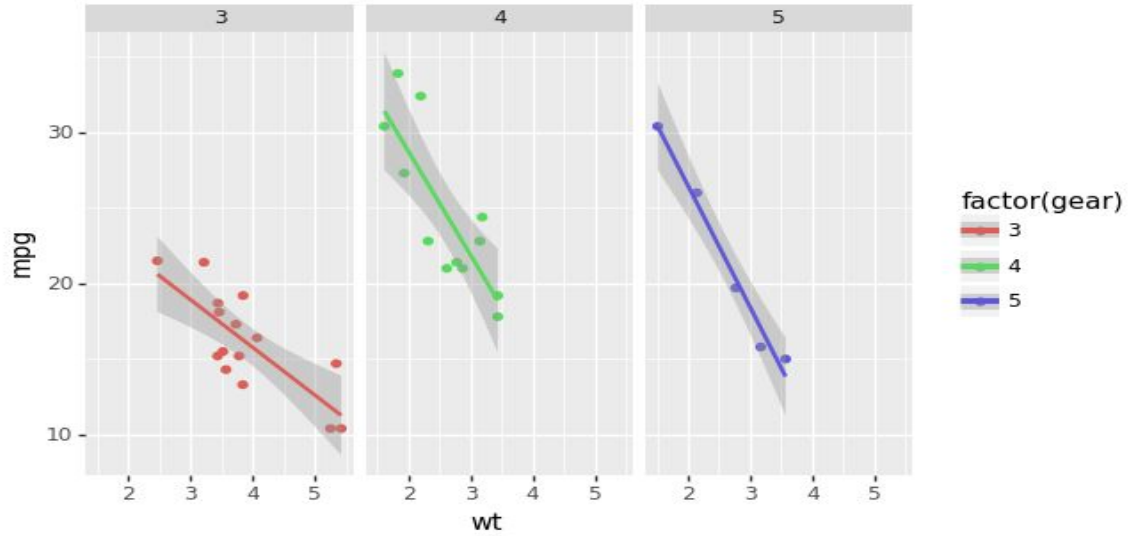
A “grammar of graphics”

- 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

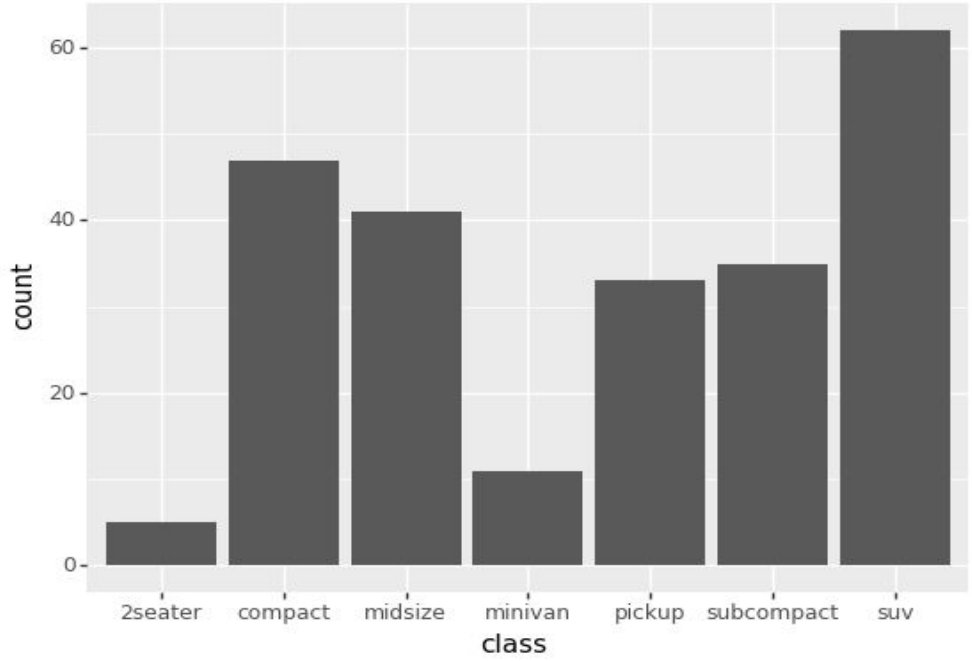
```
from plotnine import ggplot, geom_point, aes, stat_smooth, facet_wrap
from plotnine.data import mtcars

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



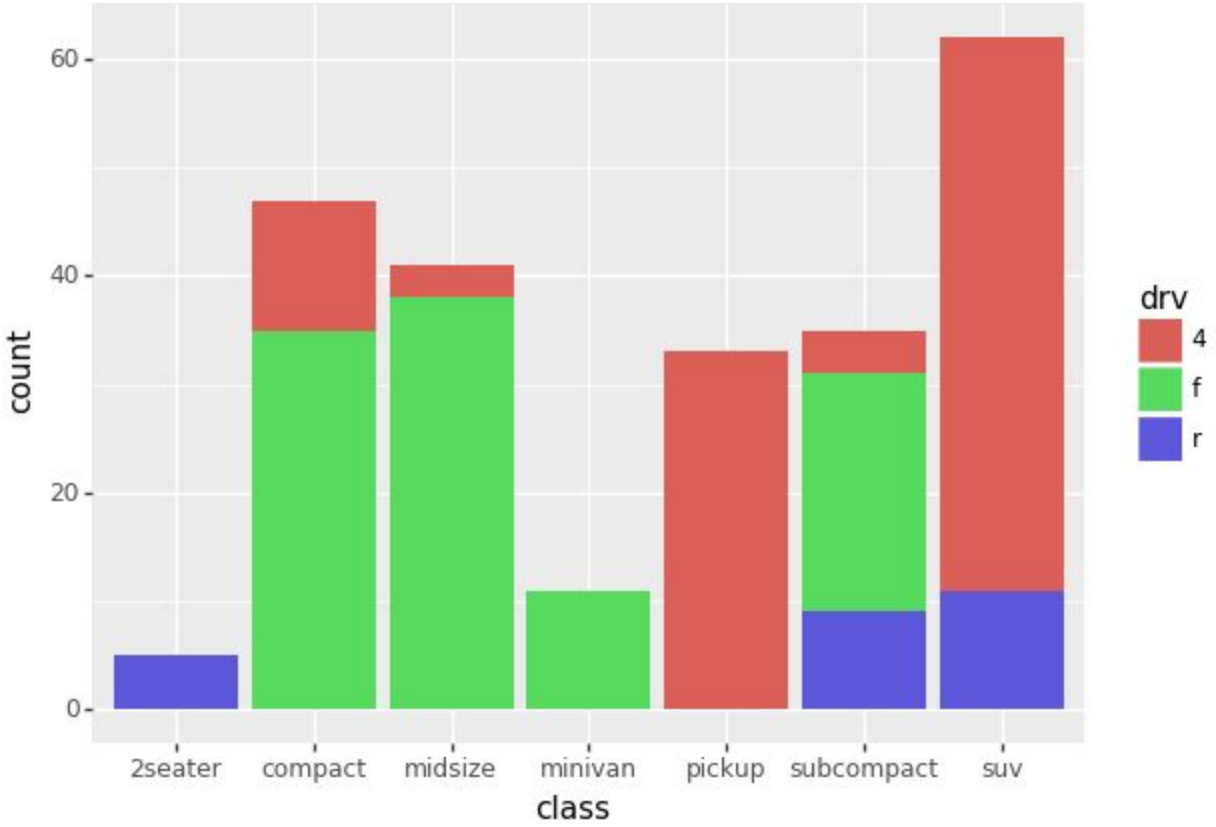
[2]:

	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

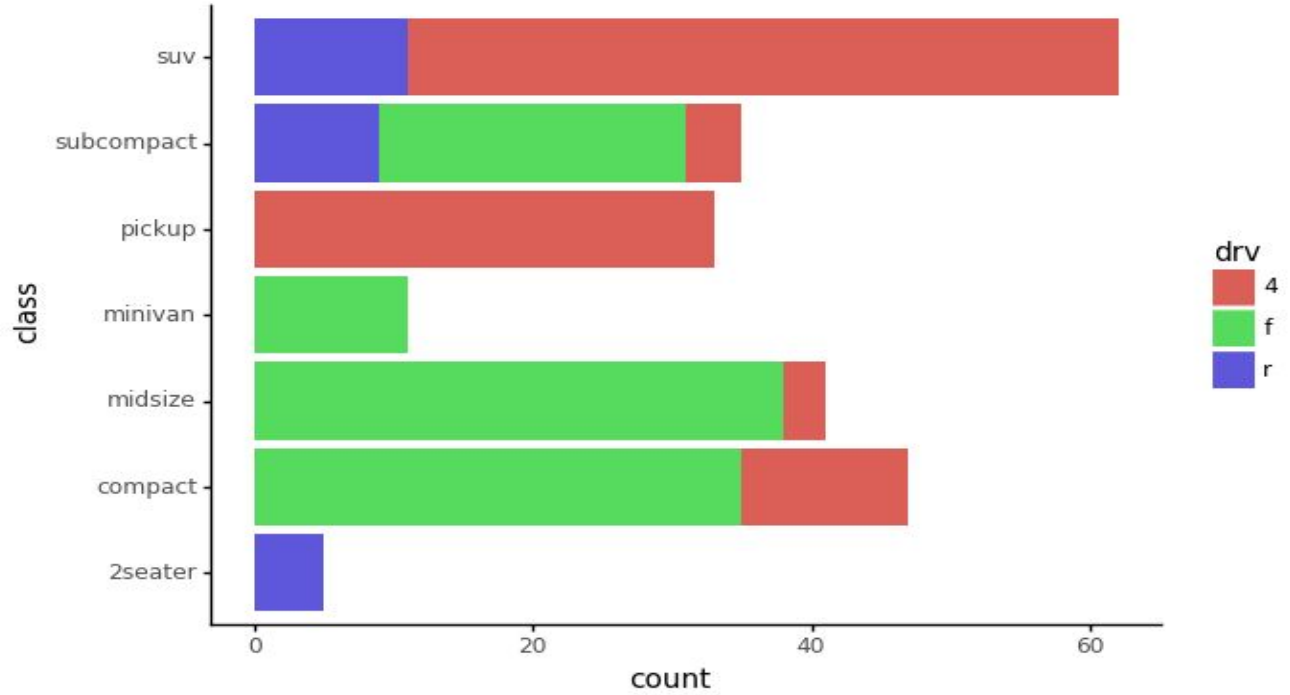
[3]:
```
ggplot(mpg) + geom_bar(aes(x='class'))
```



```
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