

#SASGF

VIRTUAL

SAS® GLOBAL FORUM 2021

AMERICAS | MAY 18 - 20

ASIA PACIFIC | MAY 19 - 20

EMEA | MAY 25 - 26

Hello Graphics World: Basic Charts in SAS[®] (ODS Graphics), R (ggplot2) & Python (Matplotlib)

Ted Conway, Chicago IL

Ted Conway is a long-time user of SAS software on a variety of platforms, including Linux, PC, mainframe, and AWS. Ted studied Computer Science at the University of Illinois (UIUC) and DePaul University. He lives in Chicago, IL.

#SASGF 2021

Hello Graphics World

Fundamental charts in SAS (ODS Graphics), R (ggplot2), and Python (Matplotlib)

Ted Conway, Chicago IL





AGENDA



HELLO GRAPHICS WORLD

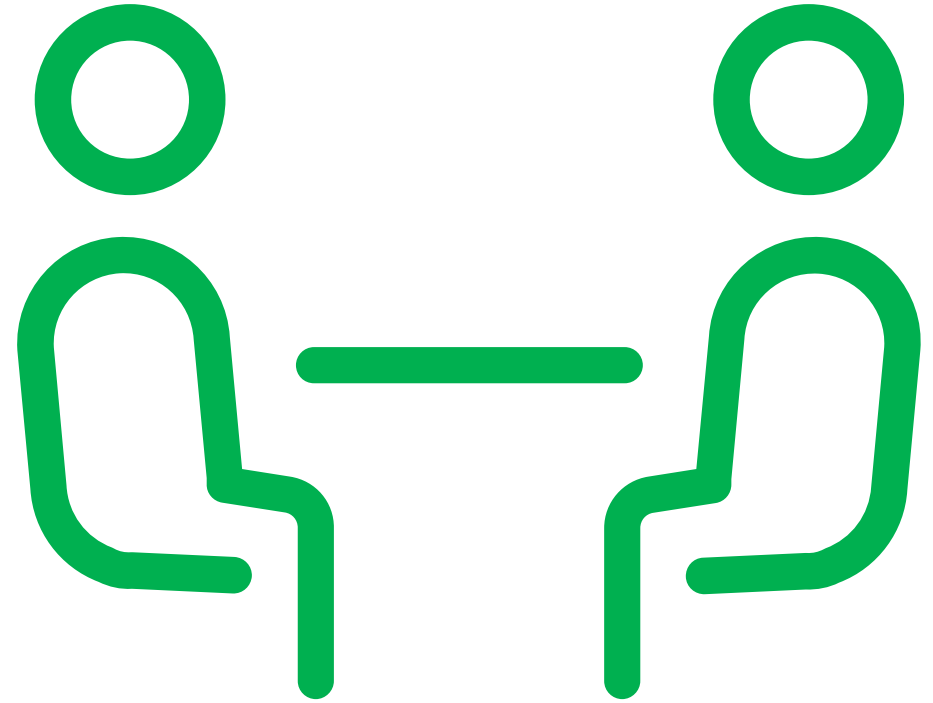
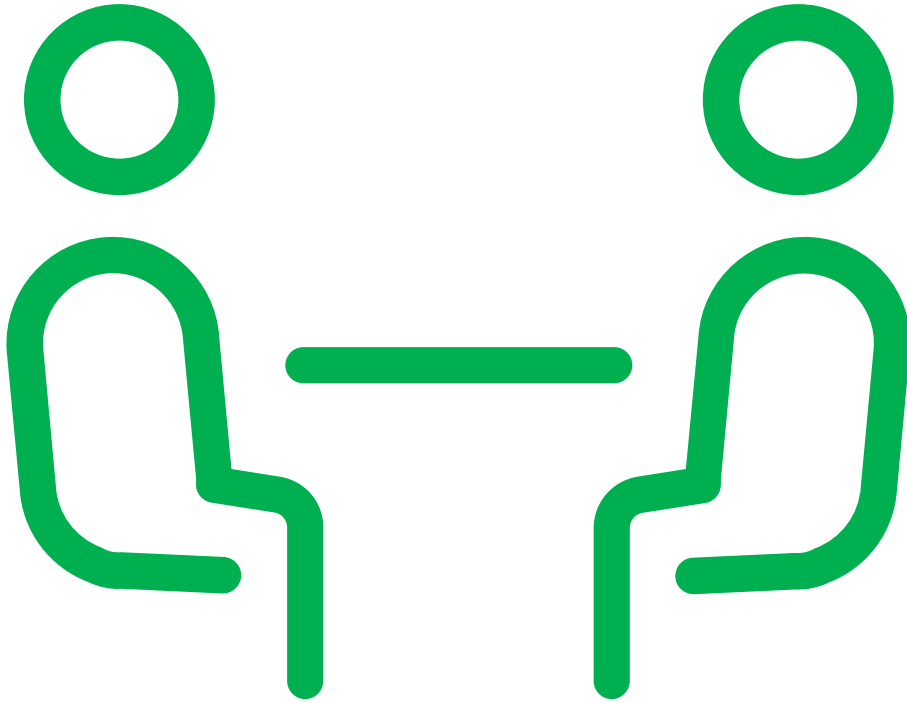
- ✓ Why Graphics Coding?
- ✓ SAS • R • Python Examples
 - Setup – Data and Libraries
 - Bar Chart, Bubble Plot, Histogram, Box Plot, Series/Line Plot
- ✓ Tradeoffs
- ✓ What Should I Use?

Ever use SAS ODS Graphics for your charts?

Graphics are easier with R & Python

What do you find easier in R & Python?

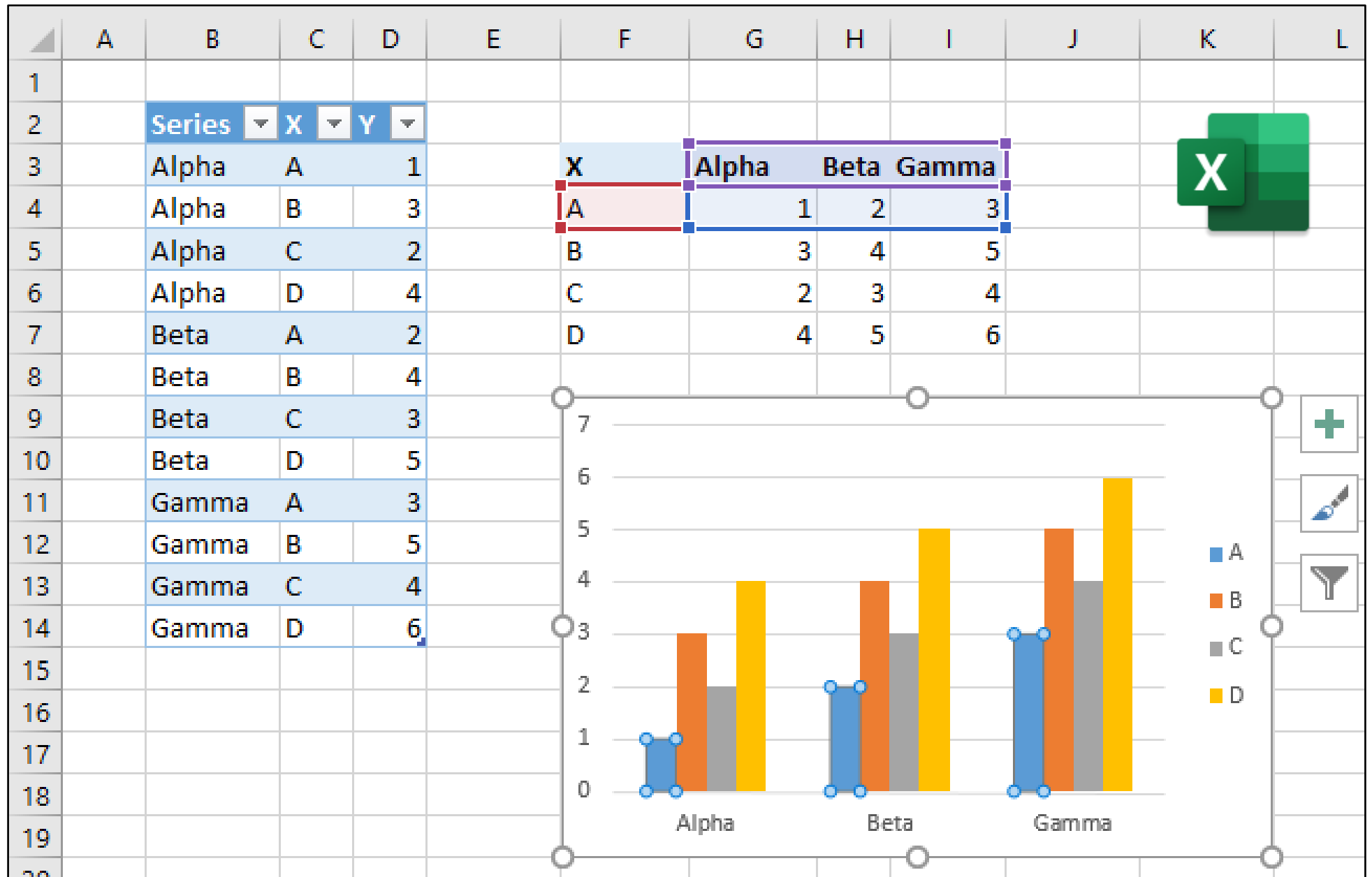
Uh, I actually use Excel to create charts



CHARTING WITH EXCEL

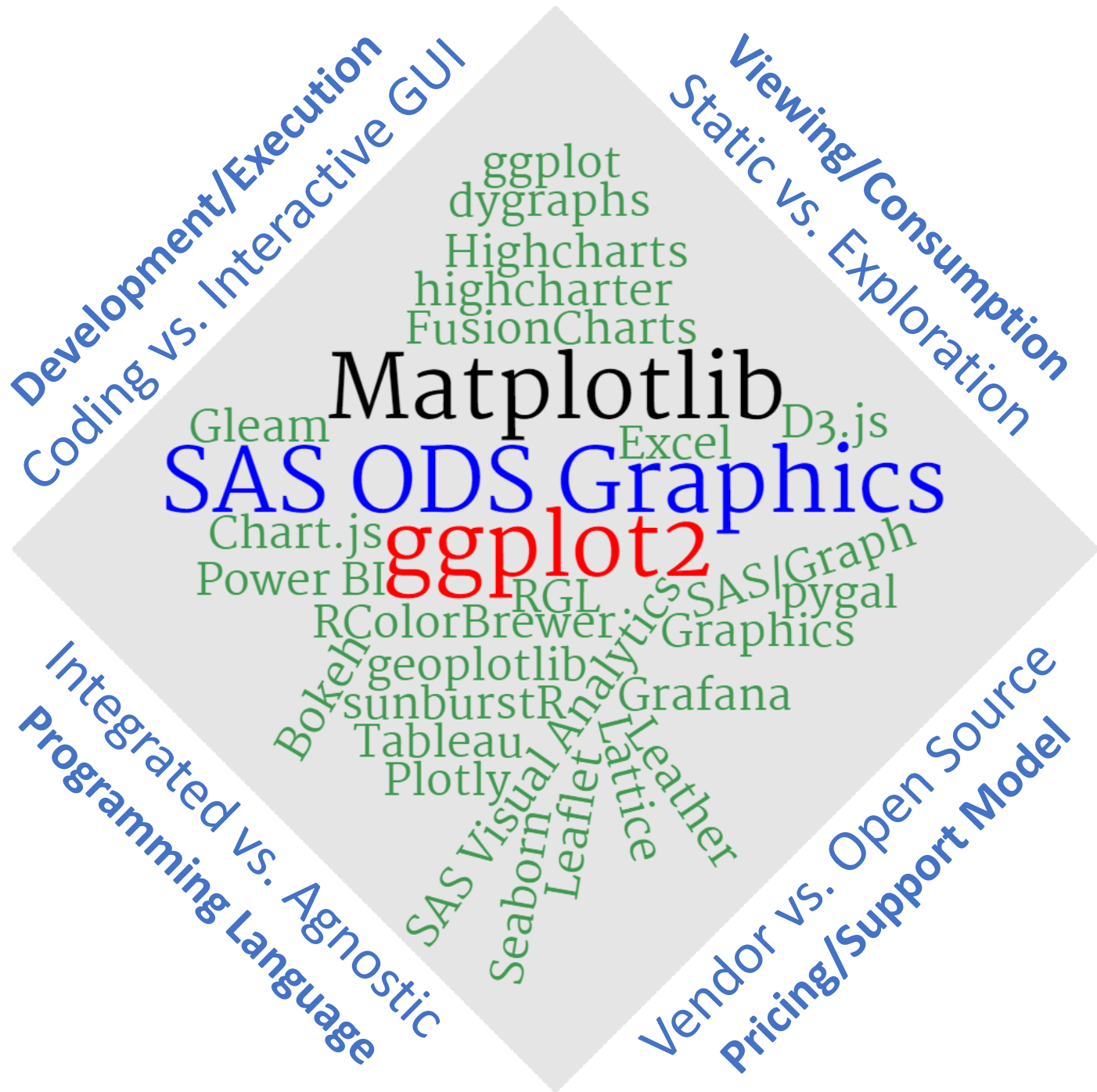
“Not all chart types are available in pivot charts...”

“...XY scatter charts and bubble charts are not”

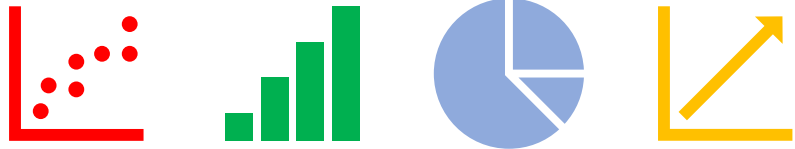


Data Visualization Tools

Used with R • Python • SAS




Core Plotting Libraries



matplotlib

python™



 **sas**
THE POWER TO KNOW.
SAS® 9.4 ODS Graphics:
Procedures Guide, Sixth
Edition

 **sas**



Why Create Charts With Coding?

- ✓ Easy-to-use, flexible, rich feature set for data visualization; produce a wealth of single and multiple-panel chart types
- ✓ Facilitates automated chart creation; promotes reproducible, repeatable results
- ✓ Scales to produce dozens, hundreds, even thousands of precisely-sized & formatted charts with consistent “look and feel”
- ✓ Integrated with programming language – make data chart-ready in no time!
- ✓ Enable viewers to quickly understand data

```
>>> print("Hello World!")
```

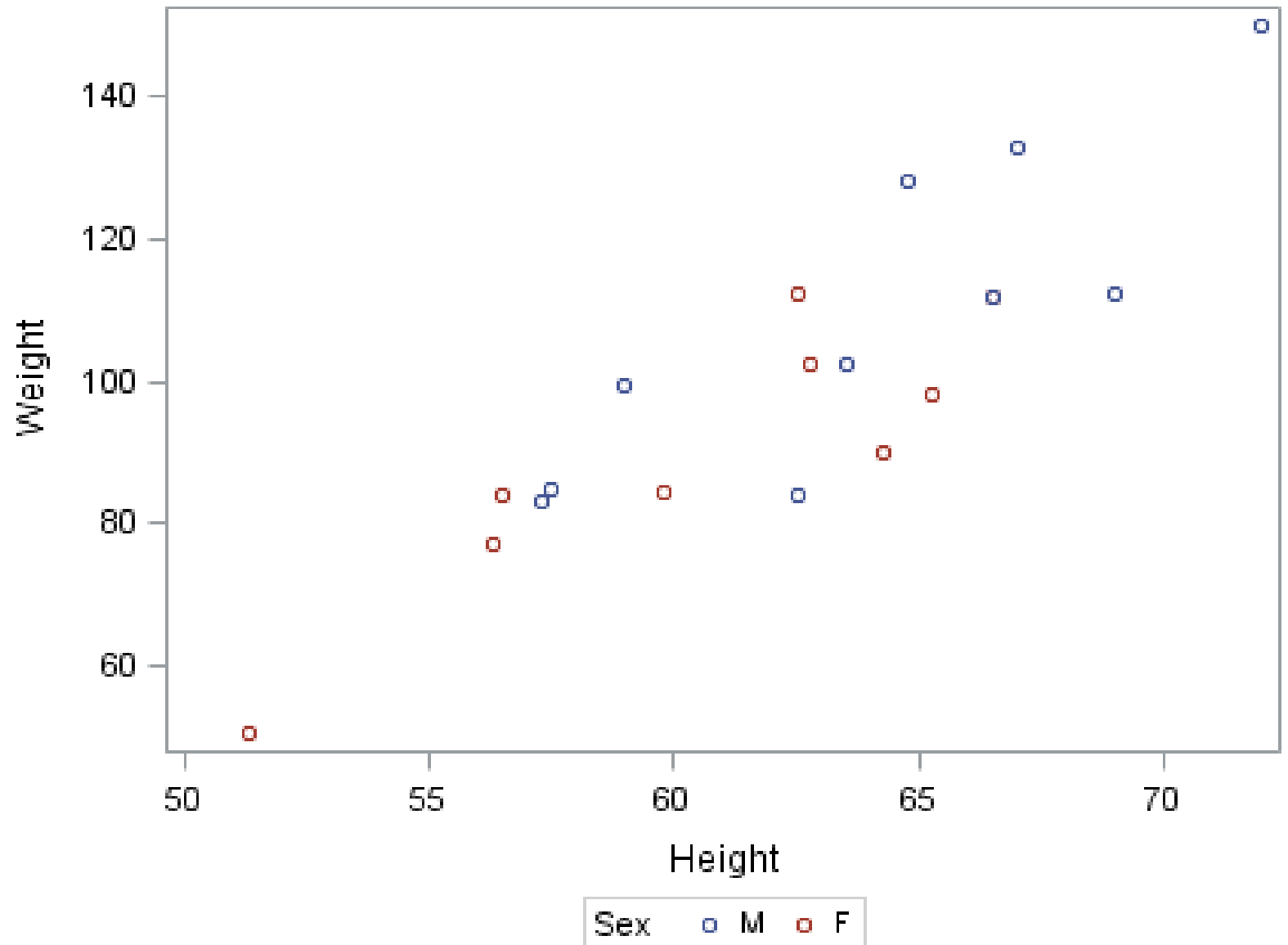
```
Hello World!
```

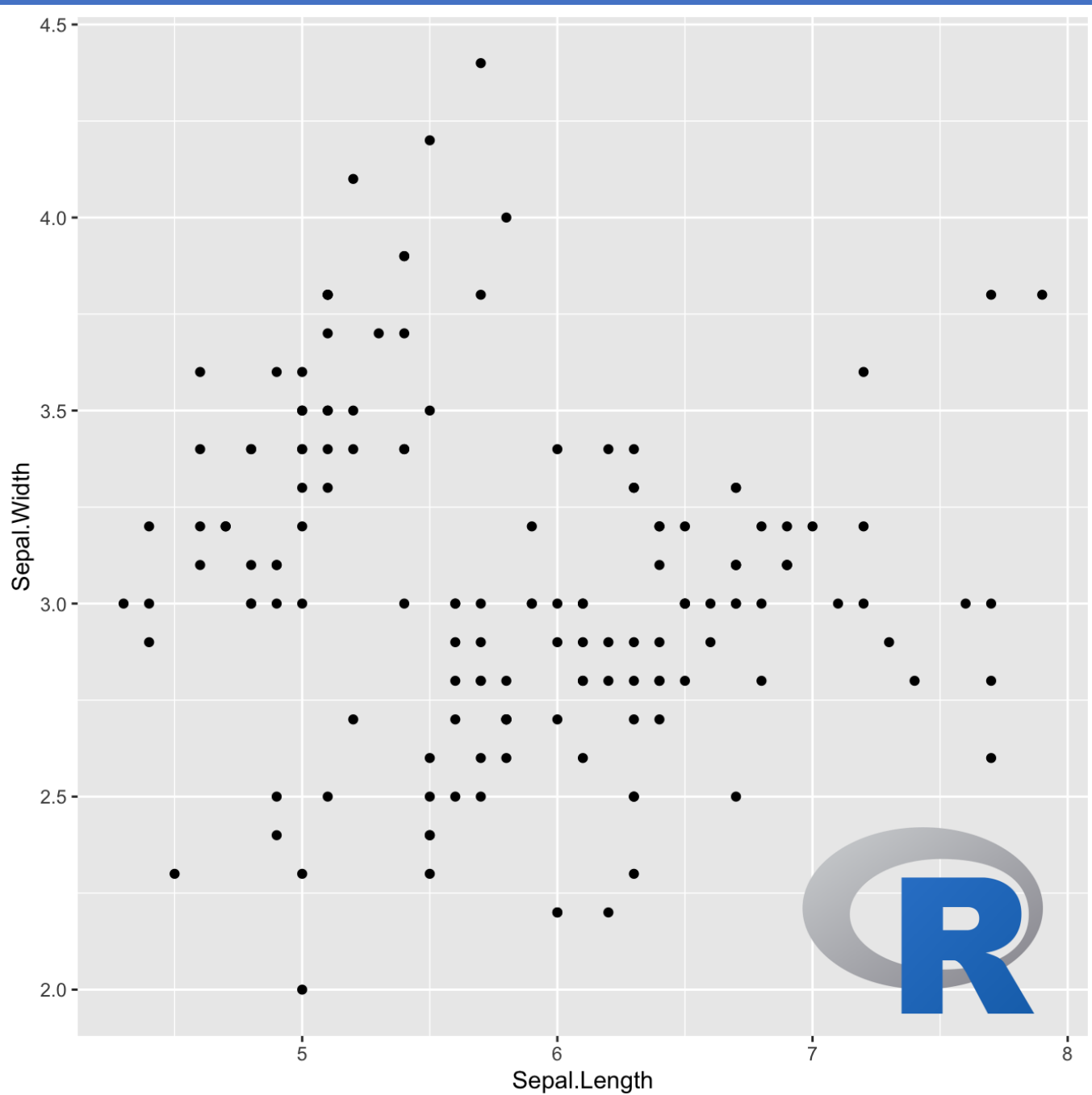
```
>>>
```

```
proc sgplot
data=sashelp.class;
scatter x=height
y=weight /
group=sex;
```

NOTES:

- STATEMENTS + KEYWORDS +
OPTIONS-ORIENTED SYNTAX





```
library(ggplot2)
ggplot(iris,
  aes(x=Sepal.Length,
    y=Sepal.Width)) +
  geom_point()
```

NOTES:

- LIBRARY DEPENDENCY
- FUNCTIONAL SYNTAX
- AES=AESTHETIC PROPERTIES (VISUAL)
- GEOM=GEOMETRIC OBJECT



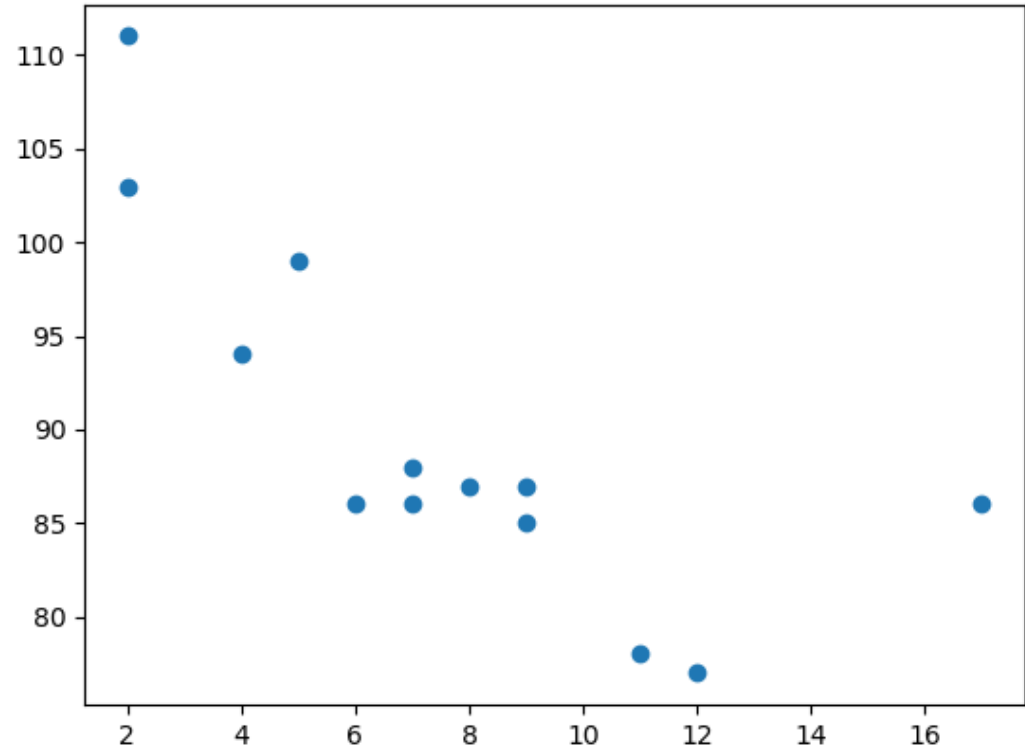
```
import matplotlib.pyplot as plt
```

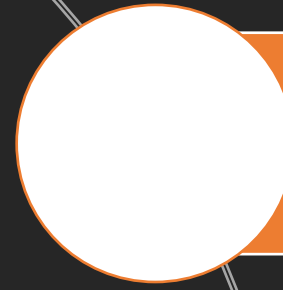
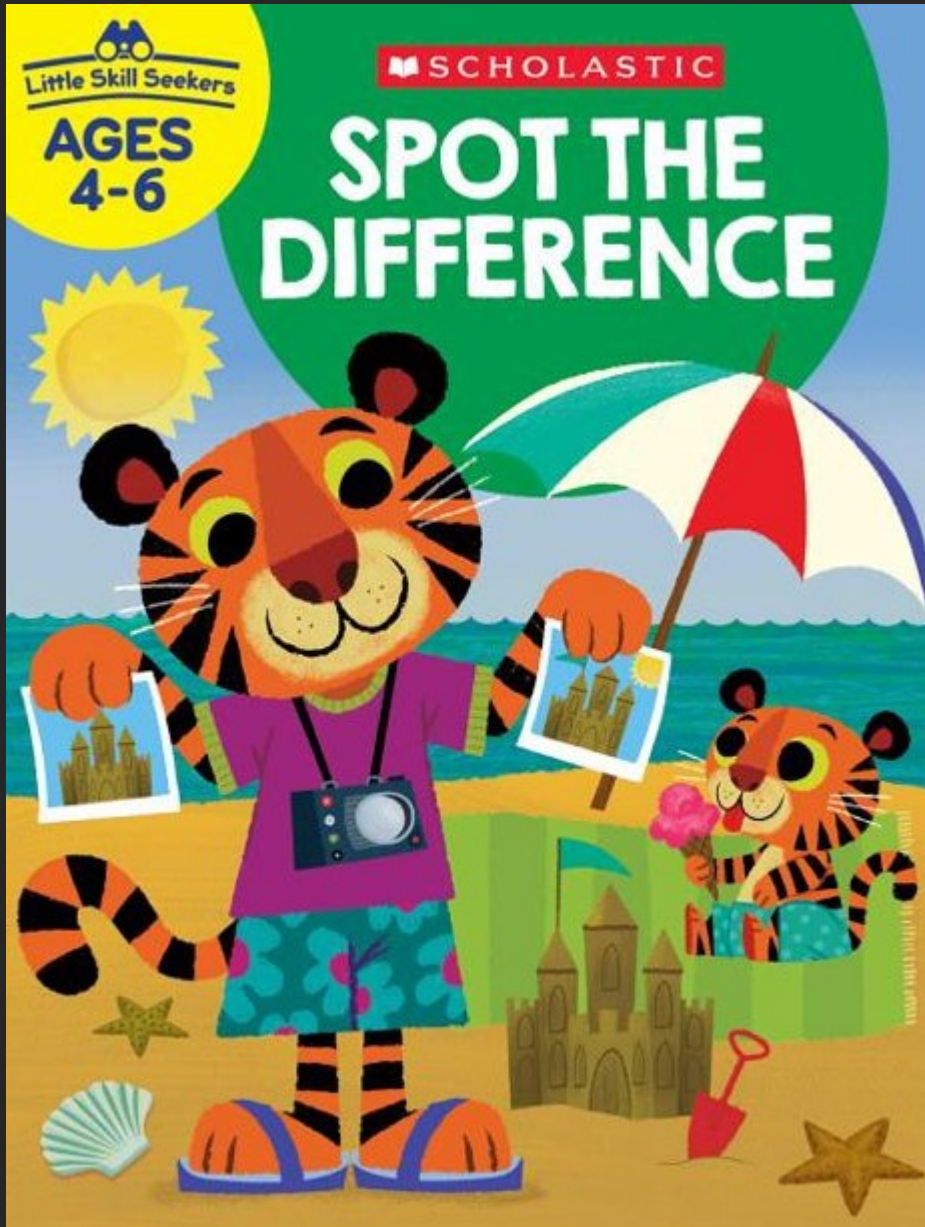
```
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]  
y = [99,86,87,88,111,86,103,87,94,  
78,77,85,86]
```

```
plt.scatter(x, y)  
plt.show()
```

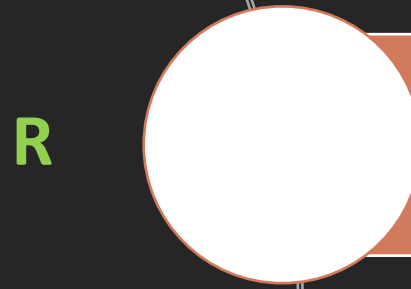
NOTES:

- LIBRARY DEPENDENCY
- OBJECT-ORIENTED AND FUNCTIONAL SYNTAX

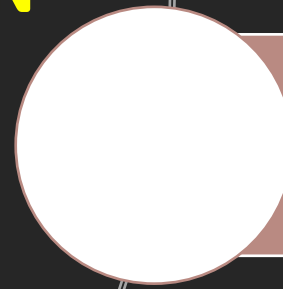




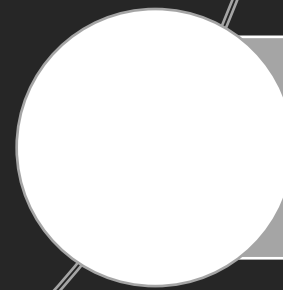
Library requirements



Sources of data



Pre-charting data prep



Graphics grammar/syntax

PYTHON

SAS

R

Note: R/Python examples drawn from [Matplotlib vs. ggplot2: Which to Choose for 2020 and Beyond?](#), by Dario Radečić

Sashelp.cars --- 2004 Car Data

The CONTENTS Procedure

Variables in Creation Order				
#	Variable	Type	Len	Format Label
1	Make	Char	13	
2	Model	Char	40	
3	Type	Char	8	
4	Origin	Char	6	
5	DriveTrain	Char	5	
6	MSRP	Num	8	DOLLAR8.
7	Invoice	Num	8	DOLLAR8.
8	EngineSize	Num	8	Engine Size (L)
9	Cylinders	Num	8	
10	Horsepower	Num	8	
11	MPG_City	Num	8	MPG (City)
12	MPG_Highway	Num	8	MPG (Highway)
13	Weight	Num	8	Weight (LBS)
14	Wheelbase	Num	8	Wheelbase (IN)
15	Length	Num	8	Length (IN)

Sashelp.air --- Airline Data (Monthly: Jan49-Dec60)

The CONTENTS Procedure

Variables in Creation Order				
#	Variable	Type	Len	Format Label
1	DATE	Num	8	MONYY.
2	AIR	Num	8	international airline travel (thousands)

The First Five Observations Out of 144

DATE	AIR
JAN49	112
FEB49	118
MAR49	132
APR49	129
MAY49	121

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
Acura	MDX	SUV	Asia	All	\$36,945	\$33,337	3.5	6	265	17	23	4451	106	189
Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761	2.0	4	200	24	31	2778	101	172
Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	2.4	4	200	22	29	3230	105	183
Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6	270	20	28	3575	108	186
Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6	225	18	24	3880	115	197

```
ods graphics / height=7in with=12in antialias noborder;  
data air; set sashelp.air; format date yymmddd10.;  
proc export data=air dbms=csv outfile=".../air.csv" replace;  
proc export data=sashelp.cars dbms=csv outfile=".../cars.csv" replace;
```

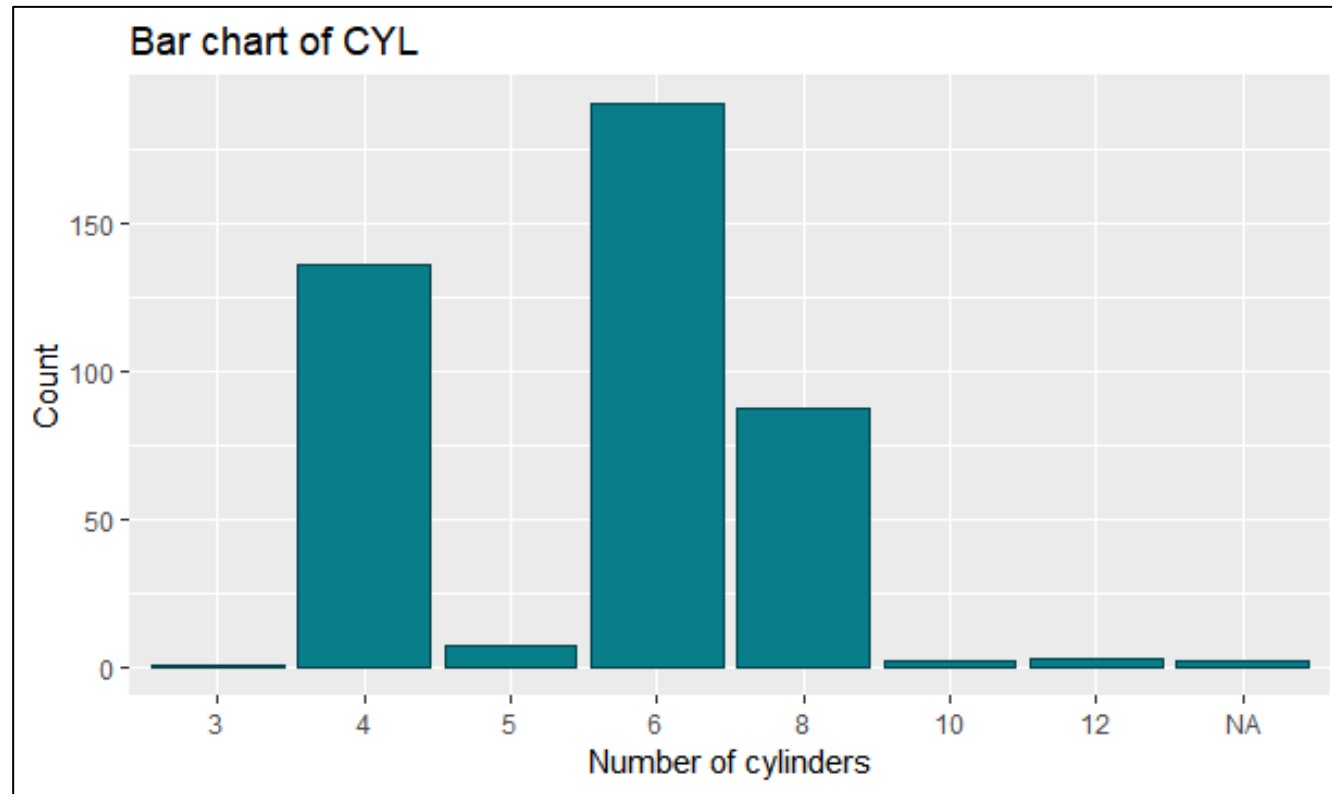


```
library(ggplot2)  
df <- read.csv('.../SASUniversityEdition/myfolders/cars.csv')  
df <- read.csv('.../SASUniversityEdition/myfolders/air.csv')
```

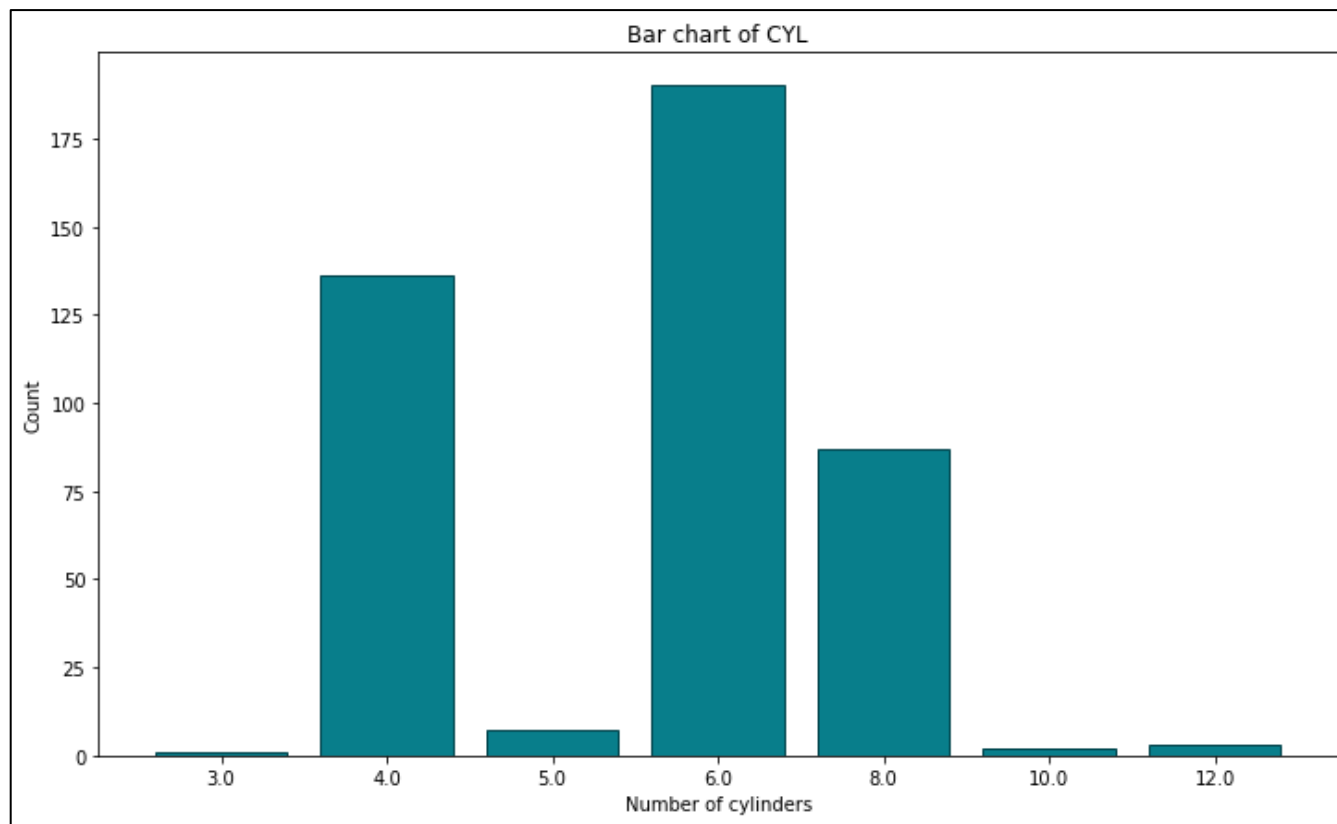


```
import pandas as pd  
import matplotlib.pyplot as plt  
import matplotlib.dates as mdates  
df=pd.read_csv('.../SASUniversityEdition/myfolders/cars.csv')  
df=pd.read_csv('.../SASUniversityEdition/myfolders/air.csv')
```



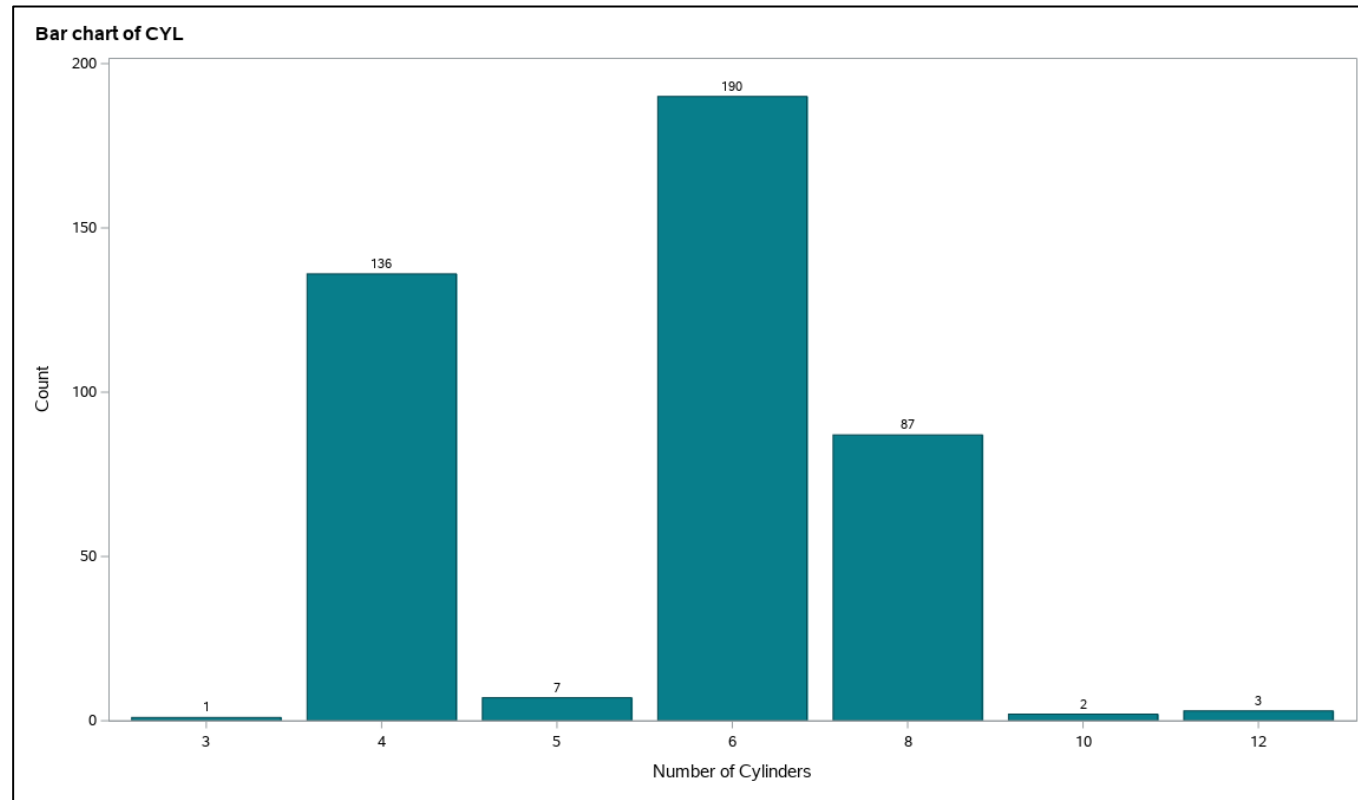


```
df$Cylinders <- factor(df$Cylinders)
ggplot(df, aes(x=Cylinders)) +
  geom_bar(fill='#087E8B', color='#02454d') +
  ggtitle('Bar chart of CYL') +
  xlab('Number of cylinders') + ylab('Count')
```

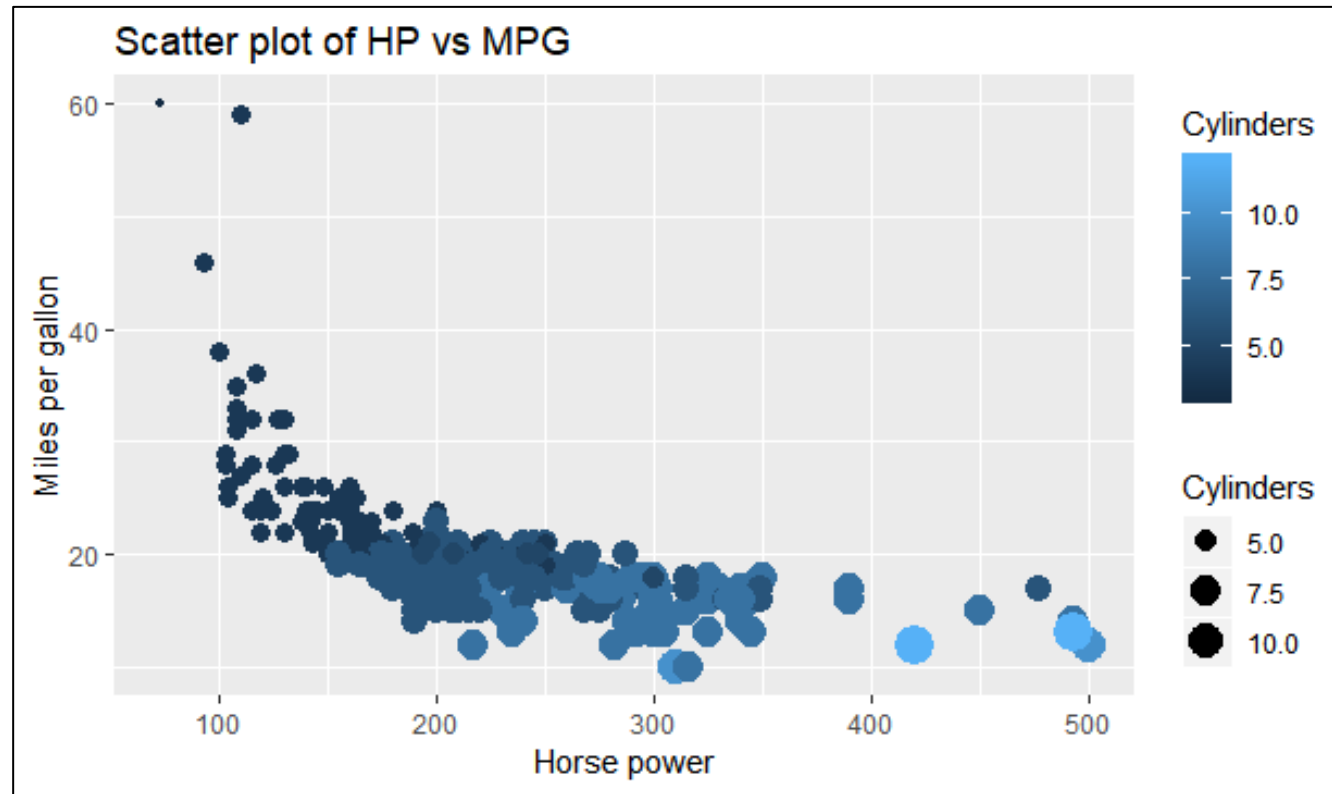


```
bar_counts =  
df['Cylinders'].value_counts().sort_index()  
bar_x =  
df['Cylinders'].value_counts().sort_index().index  
bar_height =  
df['Cylinders'].value_counts().sort_index().values  
plt.figure(figsize=(12, 7))
```

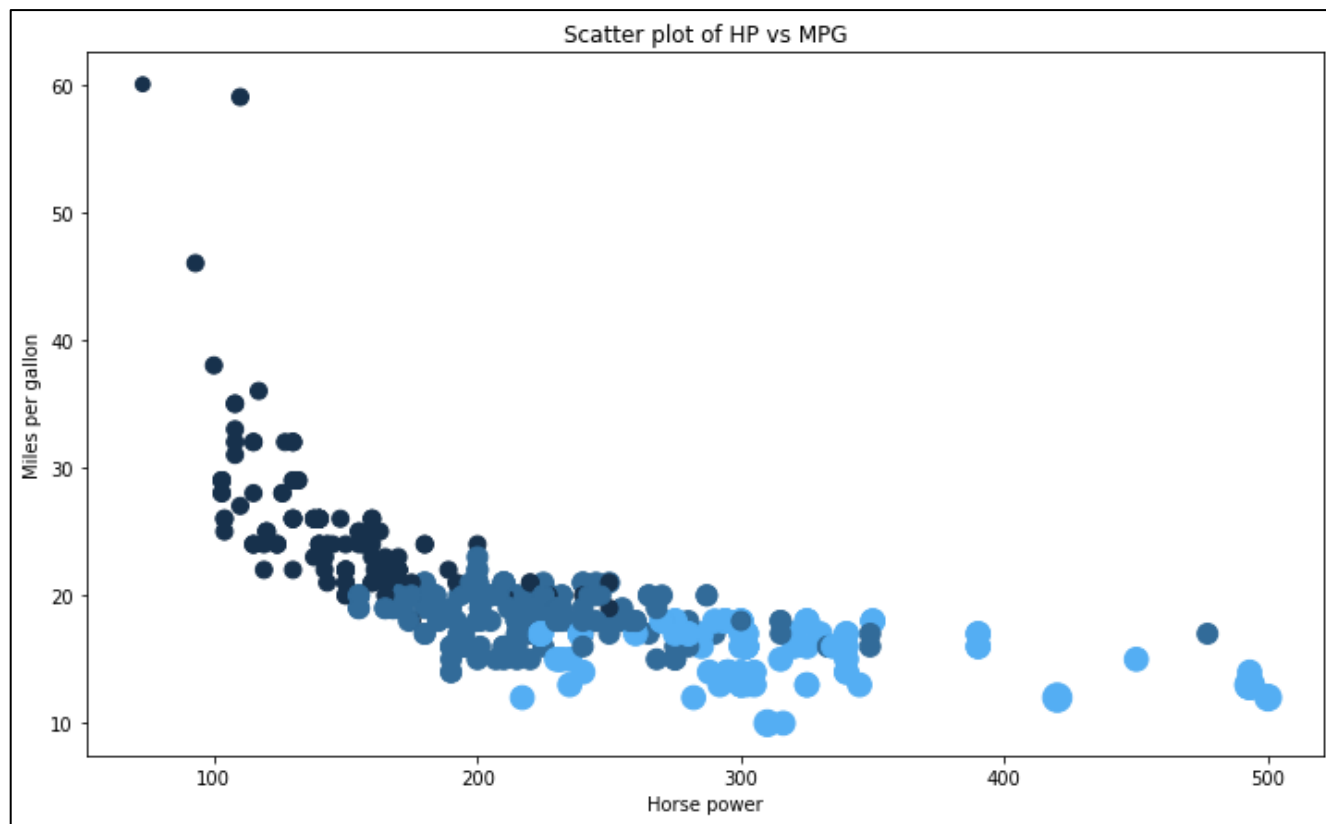
```
plt.bar(x=bar_x.astype(str), height=bar_height,  
color='#087E8B', ec='#02454d')  
plt.title('Bar chart of CYL')  
plt.xlabel('Number of cylinders')  
plt.ylabel('Count');
```



```
proc sgplot data=sashelp.cars;  
title justify=left 'Bar chart of CYL';  
vbar cylinders / fillattrs=(color=cX087E8B) outlineattrs=(color=cx02454d) datalabel;  
label cylinders='Number of Cylinders'; yaxis label='Count';
```

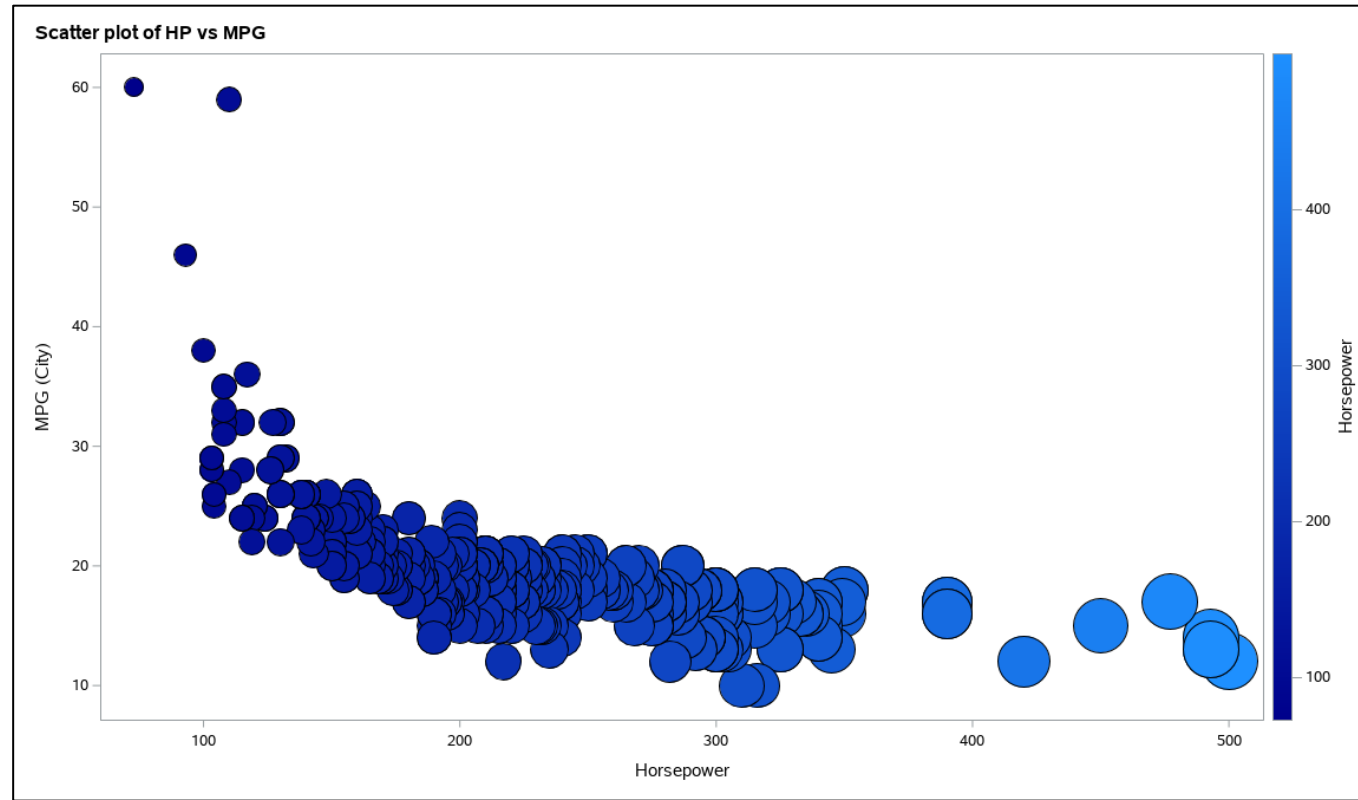


```
ggplot(df, aes(x=Horsepower, y=MPG_City)) +  
  geom_point(aes(size=Cylinders, color=Cylinders)) +  
  ggtitle('Scatter plot of HP vs MPG') +  
  xlab('Horse power') + ylab('Miles per gallon')
```

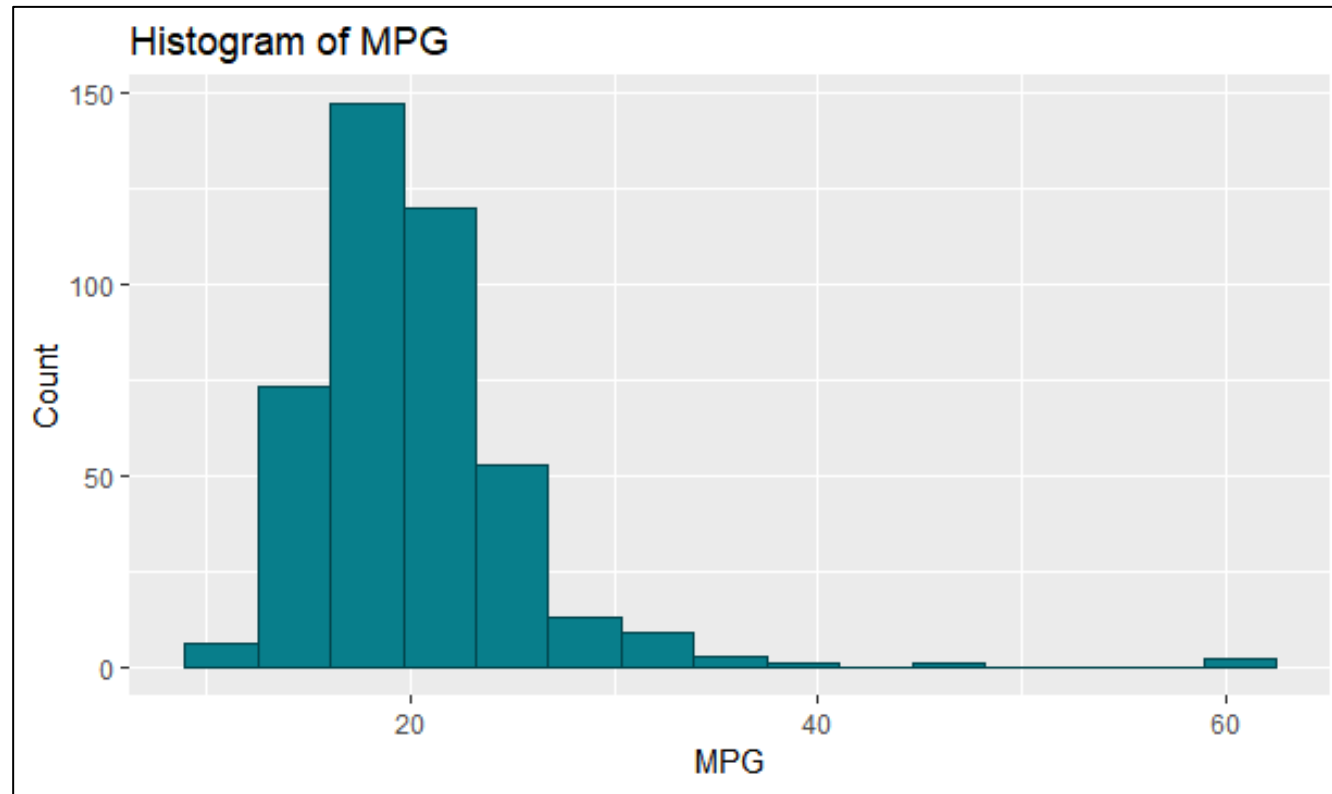


```
colors = []
for val in df['Cylinders']:
    if val <= 4: colors.append('#17314c')
    elif val <= 6: colors.append('#326b99')
    else: colors.append('#54aef3')
plt.figure(figsize=(12, 7))
```

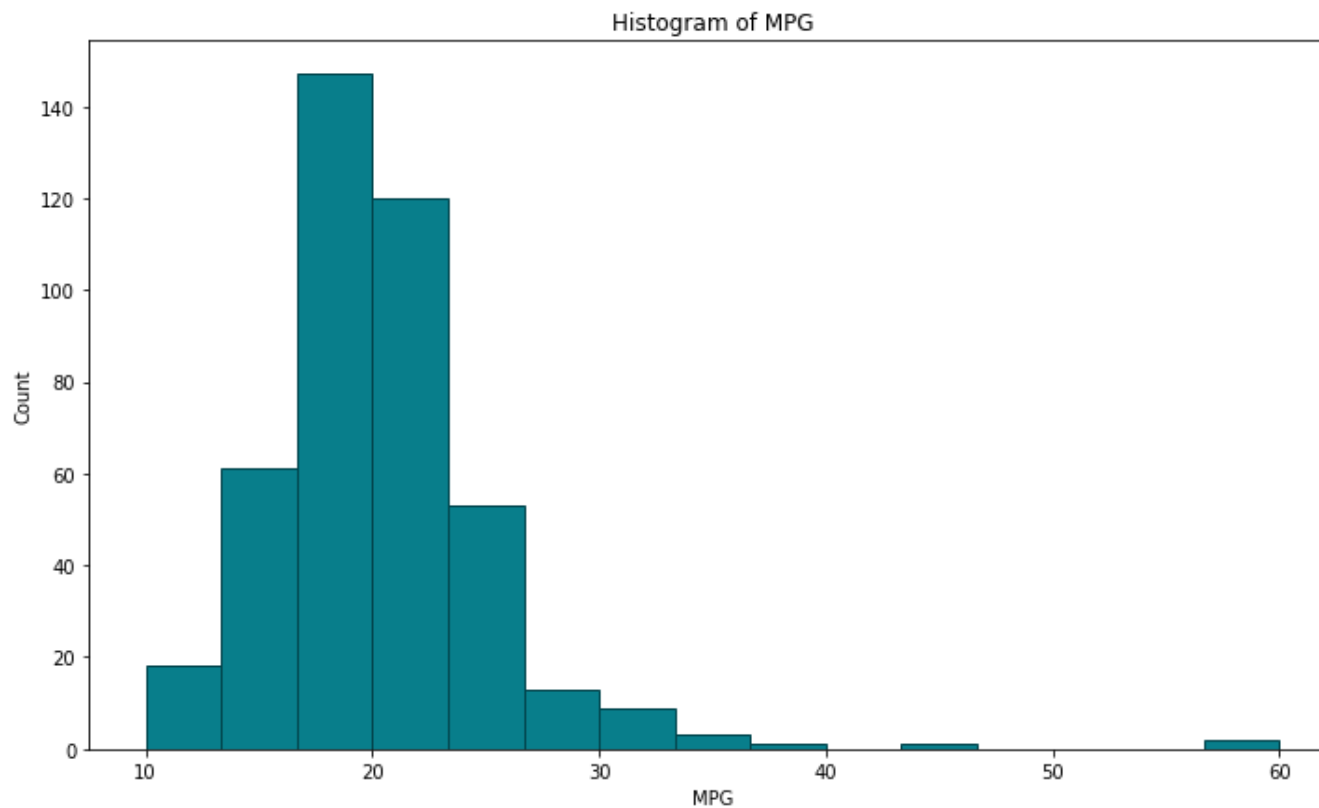
```
plt.scatter(x=df['Horsepower'],
            y=df['MPG_City'], s=df['Cylinders'] * 20,
            c=colors)
plt.title('Scatter plot of HP vs MPG')
plt.xlabel('Horse power')
plt.ylabel('Miles per gallon');
```



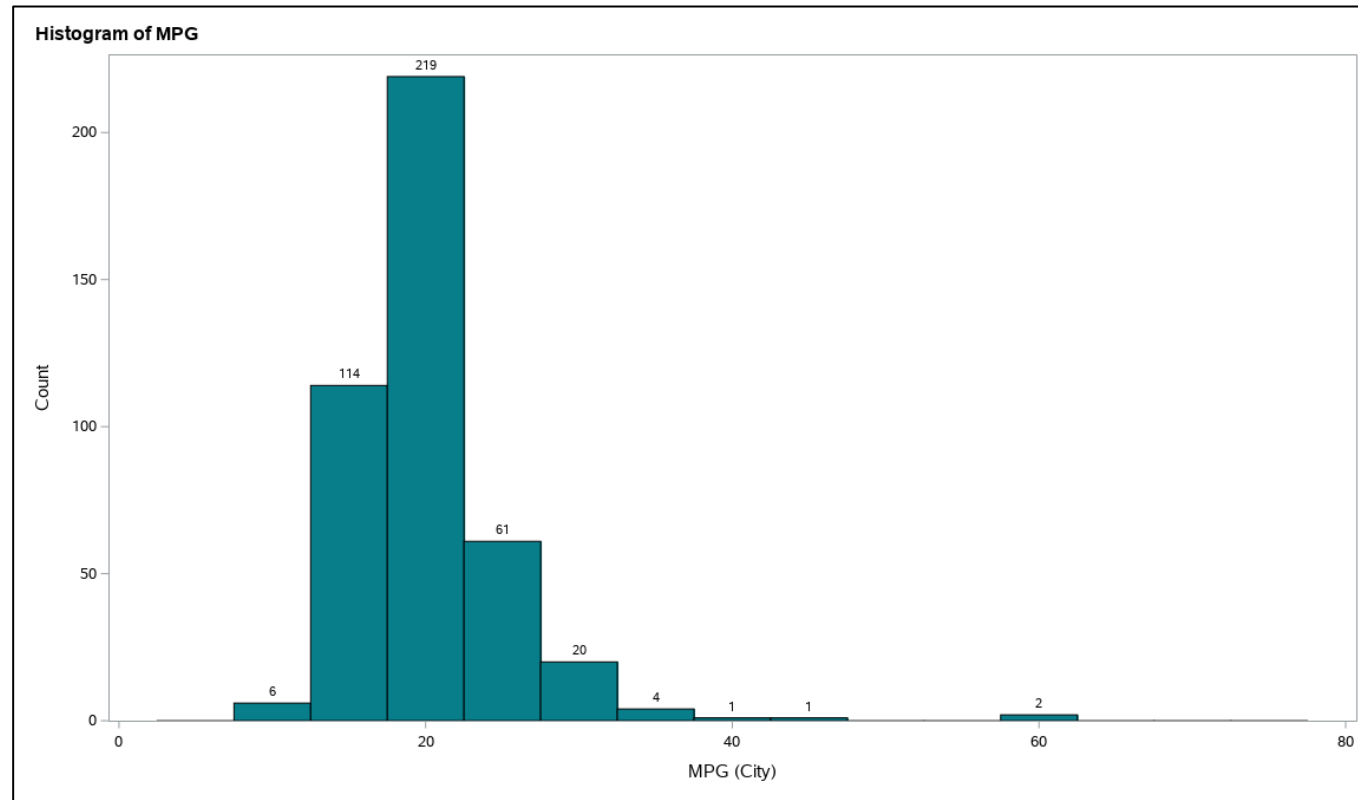
```
proc sgplot data=sashelp.cars;  
title justify=left 'Scatter plot of HP vs MPG';  
bubble x=horsepower y=mpg_city size=horsepower /  
colorresponse=horsepower colormodel=(darkblue dodgerblue);
```



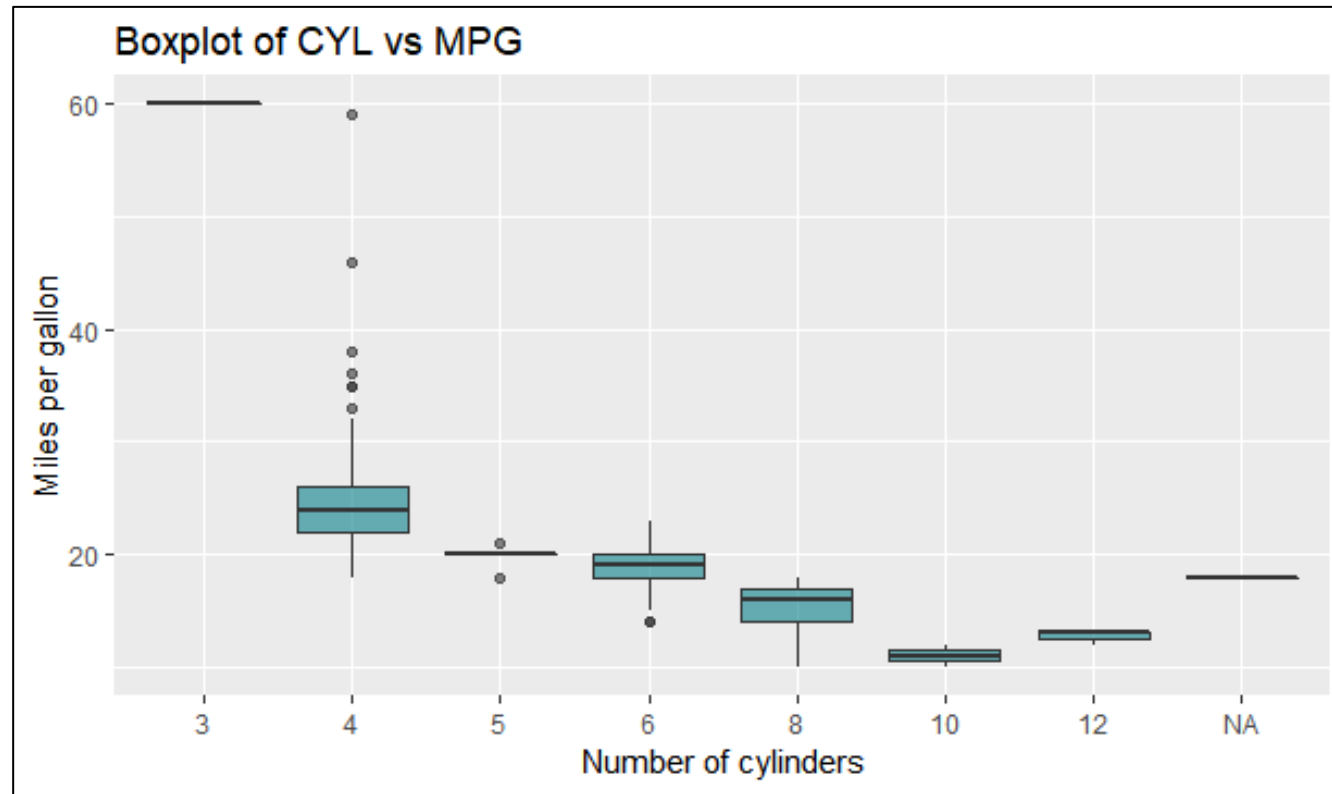
```
ggplot(df, aes(x=MPG_City)) +  
  geom_histogram(bins=15, fill='#087E8B', color='#02454d') +  
  ggtitle('Histogram of MPG') + xlab('MPG') + ylab('Count')
```



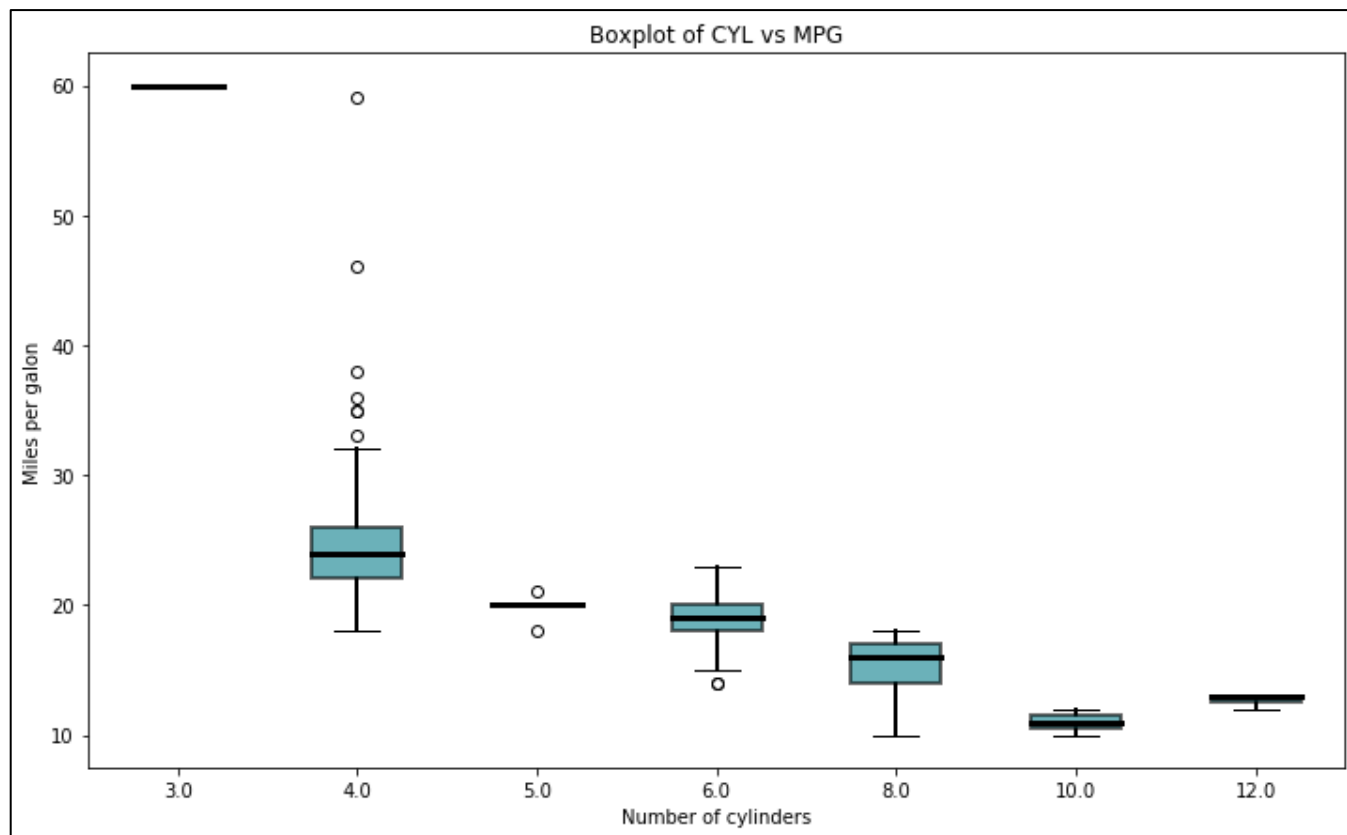
```
plt.figure(figsize=(12, 7))
plt.hist(df['MPG_City'], bins=15,
color='#087E8B', ec='#02454d')
plt.title('Histogram of MPG')
plt.xlabel('MPG')
plt.ylabel('Count');
```

```
proc sgplot data=sashelp.cars;  
title justify=left 'Histogram of MPG';  
histogram mpg_city / nbins=15 scale=count fillattrs=(color=cX087E8B) datalabel=count;
```

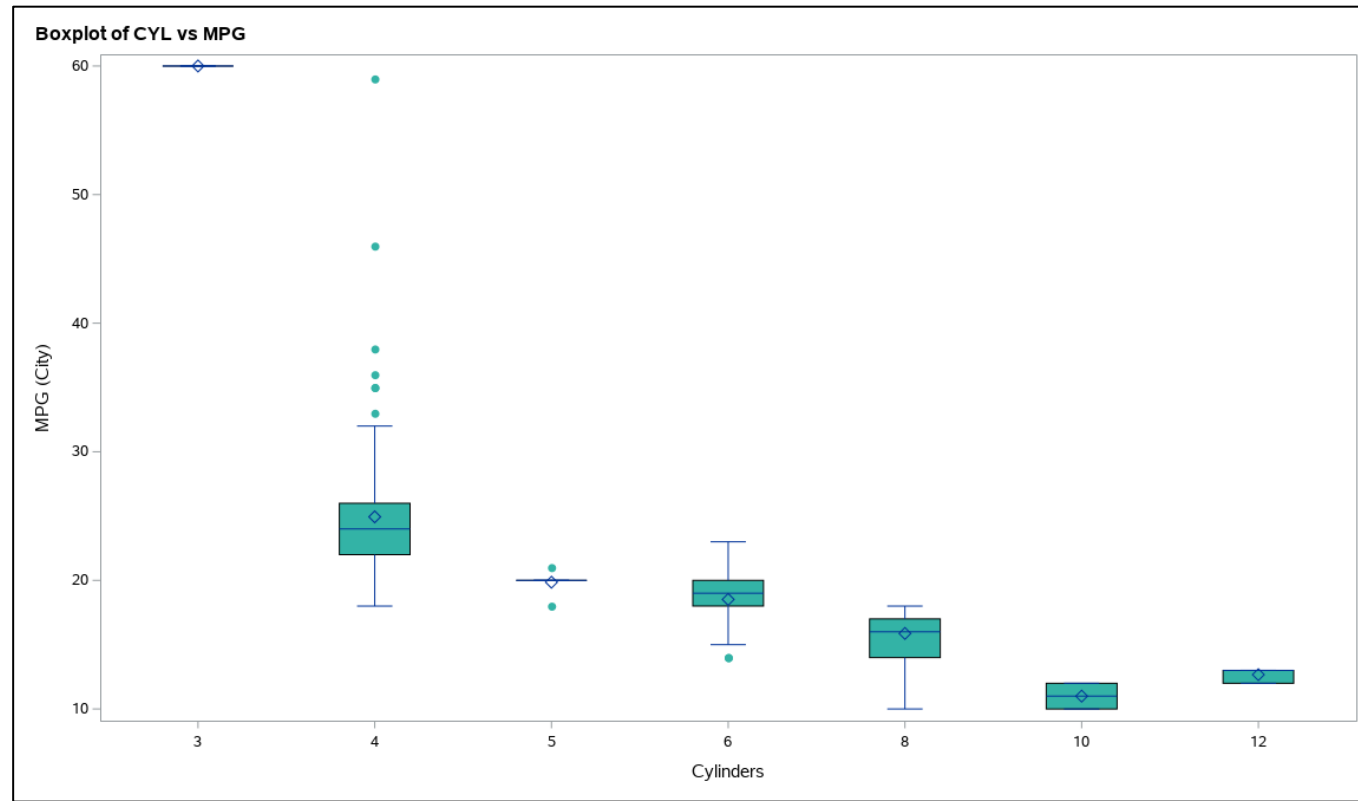


```
ggplot(df, aes(x=as.factor(Cylinders), y=MPG_City)) +  
  geom_boxplot(fill='#087E8B', alpha=0.6) +  
  ggtitle('Boxplot of CYL vs MPG') +  
  xlab('Number of cylinders') + ylab('Miles per gallon')
```

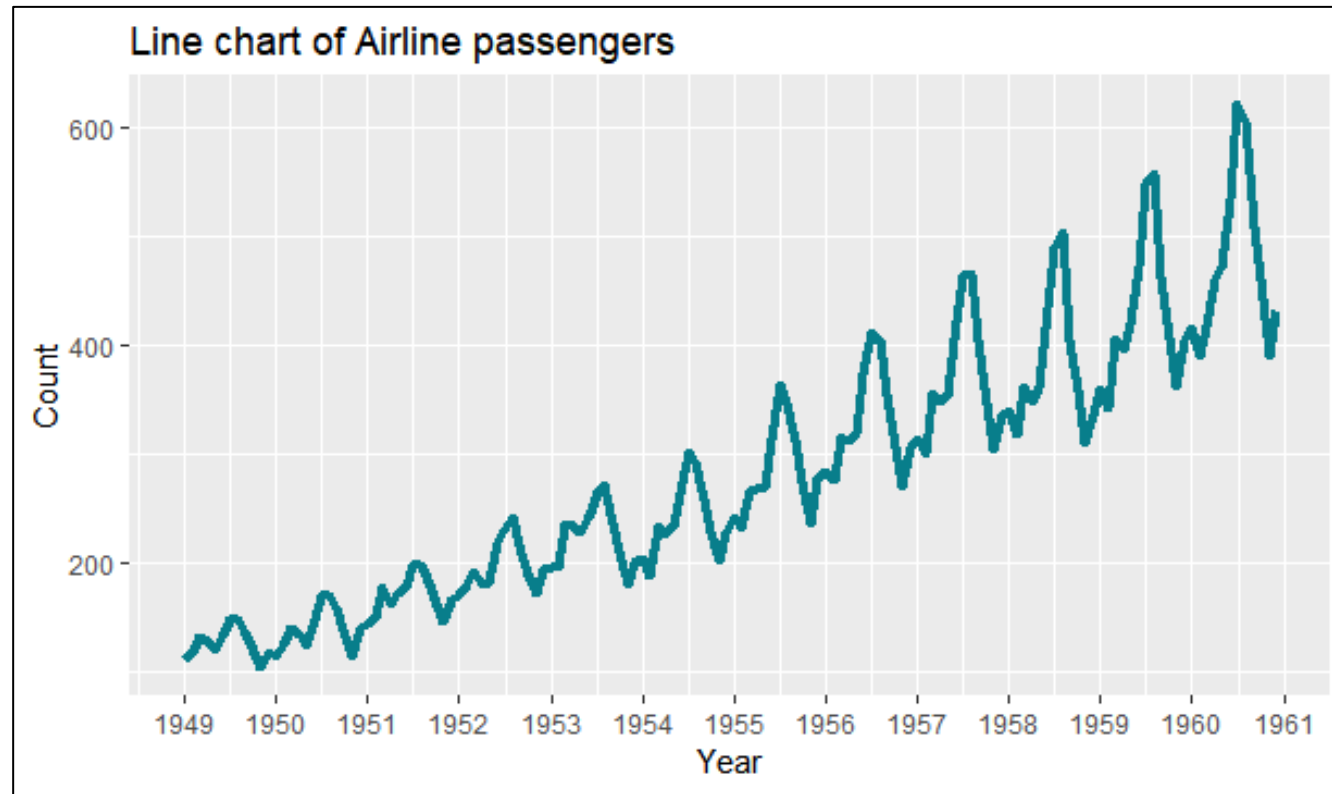


```
dfCylMPG = []
unqCyl=df[df.Cylinders.notnull()]['Cylinders'].unique().tolist()
unqCyl.sort()
for x in unqCyl:
    dfCylMPG.append(df[df['Cylinders'] == x]['MPG_City'].tolist())
fig = plt.figure(1, figsize=(12, 7))
ax = fig.add_subplot(111)
bp = ax.boxplot(dfCylMPG, patch_artist=True)
for box in bp['boxes']:
```

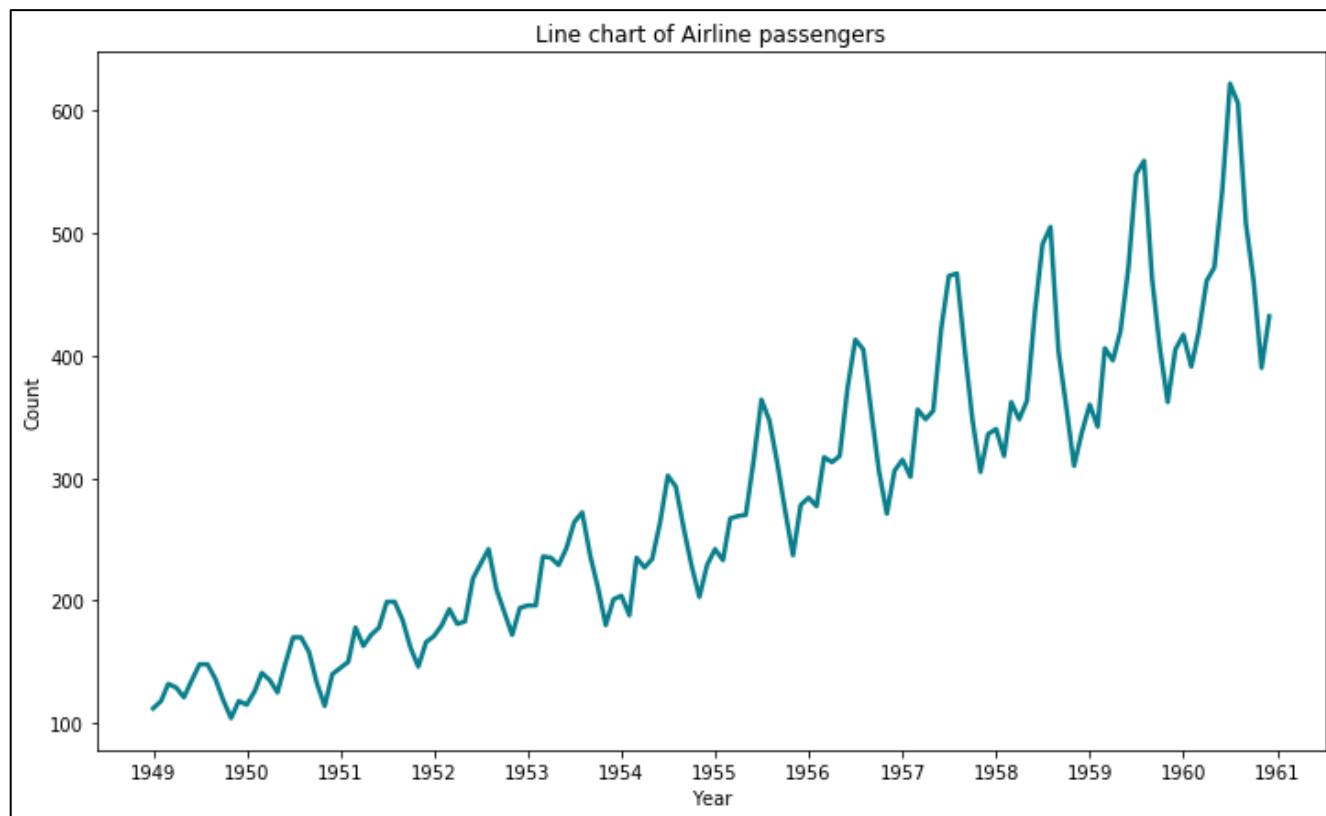
```
    box.set(facecolor='#087E8B', alpha=0.6, linewidth=2)
for whisker in bp['whiskers']:
    whisker.set(linewidth=2)
for median in bp['medians']:
    median.set(color='black', linewidth=3)
ax.set_title('Boxplot of CYL vs MPG')
ax.set_xlabel('Number of cylinders')
ax.set_ylabel('Miles per gallon')
ax.set_xticklabels(unqCyl)
```



```
proc sgplot data=sashelp.cars;  
title justify=left 'Boxplot of CYL vs MPG';  
vbox mpg_city / category=cylinders fillattrs=(color=cX33b3a6)  
outlierattrs=(color=cX33b3a6 symbol=circlefilled);
```

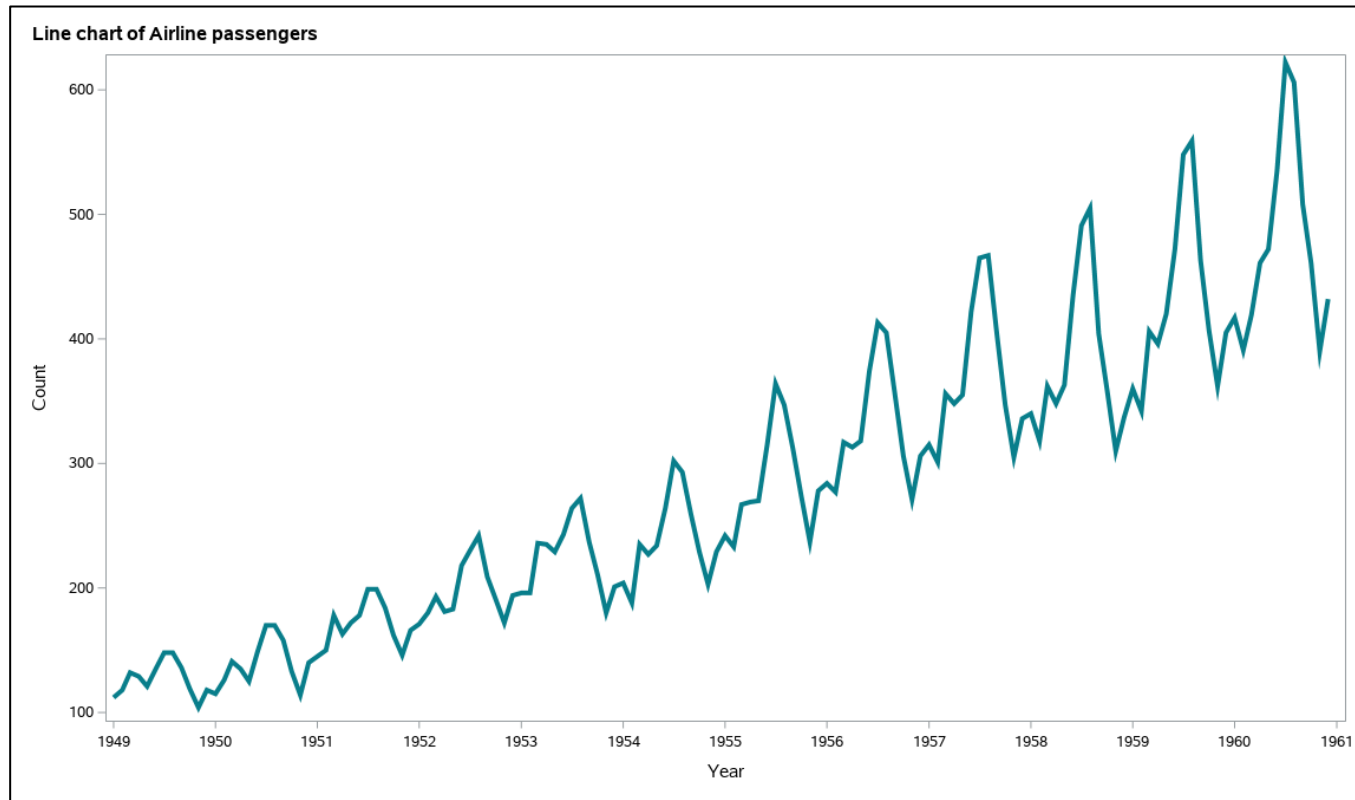


```
df$MONTH <- as.Date(df$DATE)
ggplot(df, aes(x=MONTH, y=AIR)) +
  geom_line(size=1.5, color='#087E8B') +
  scale_x_date(date_breaks='1 year', date_labels='%Y') +
  ggtitle('Line chart of Airline passengers') +
  xlab('Year') + ylab('Count')
```



```
df['DATE'] = df['DATE'].apply(lambda x:
pd.to_datetime(x))
fig = plt.figure(1, figsize=(12, 7))
ax = fig.add_subplot(111)
line = ax.plot(df['DATE'], df['AIR'], lw=2.5,
color='#087E8B')
formatter = mdates.DateFormatter('%Y')
```

```
ax.xaxis.set_major_formatter(formatter)
locator = mdates.YearLocator()
ax.xaxis.set_major_locator(locator)
ax.set_title('Line chart of Airline
passengers')
ax.set_xlabel('Year')
ax.set_ylabel('Count');
```



```
proc sgplot data=sashelp.air;  
title justify=left 'Line chart of Airline passengers';  
series x=date y=air / lineattrs=(thickness=3pt color=cX087E8B);  
label date='Year' air='Count';
```



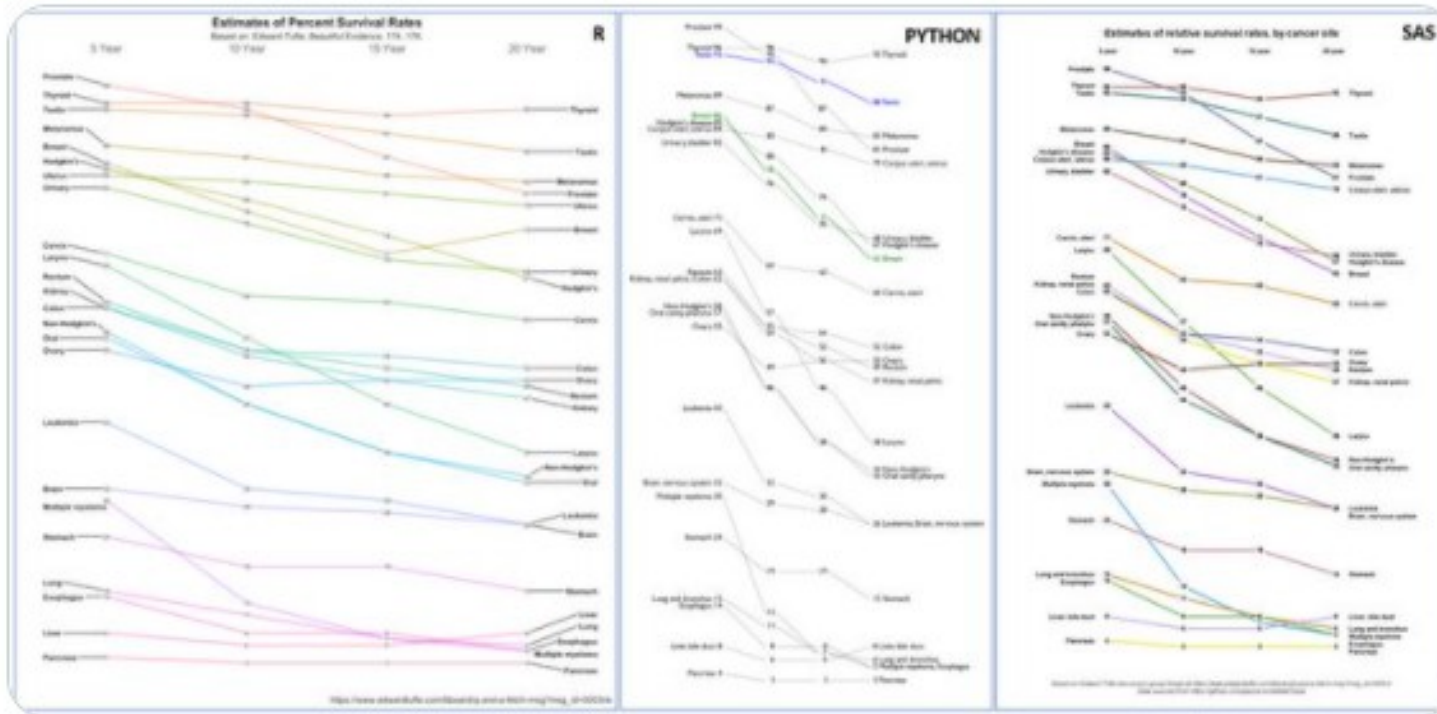
Tweet



SGF Observer
@vivasasvegas



"Hello, Slopegraph!" A #SAS ODS Graphics version of #R and #Python takes on @EdwardTufte's cancer survival slopegraph. communities.sas.com/t5/SAS-GRAPH-a...





Home

Explore

Notifications

Messages

Bookmarks

Lists

Profile

More

Tweet



SGF Observer @vivasasvegas

Tweet



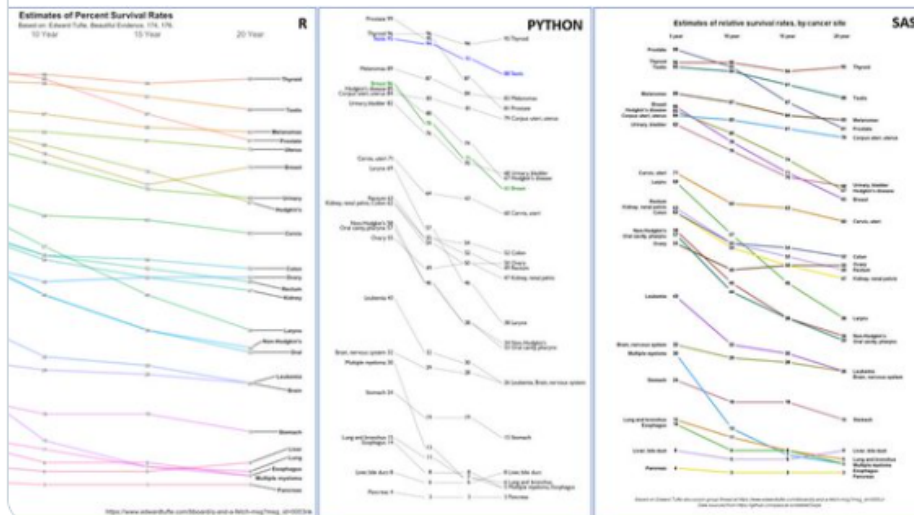
Edward Tufte @EdwardTufte

Looking good, getting close to lunatic time-sink detailing of my 2004 version by hand in Adobe Illustrator. Still a ways to go in #Rstats et al. for publication-quality graphics (e.g. Efron + Hastie, Computer Age Statistical Inference, replacing legends with direct labels.



SGF Observer @vivasasvegas · May 29, 2018

"Hello, Slopegraph!" A #SAS ODS Graphics version of #R and #Python takes on @EdwardTufte's cancer survival slopegraph. communities.sas.com/t5/SAS-GRAPH-a...



Search Twitter

Relevant people



Edward Tufte @EdwardTufte Statistician,visualizer,artist. Founded Graphics Press, Farms, ET Modern Gallery.



SGF Observer @vivasasvegas (Tweetin' On) The Dock of Mandalay Bay. SGF 2011 affiliated with SAS). Opinions own. Capiche?

What's happening

US elections · 4 hours ago Trump is reportedly considering appointing Sidney Powell as special counsel, according to The New York Times, Wall Street Journal and CNN

#HennessyFightNight Join Hennessy for the Canelo vs S Never Stop. Never Settle. Promoted by Hennessy US

NBA · This afternoon

Messages



Time, Effort,
Cost



Presentation
Quality and
'Wow' Factor



Complexity
and
Maintenance



Interactivity
(Exploration
vs. Readily
Available
Information)



Target
Audience
and Medium

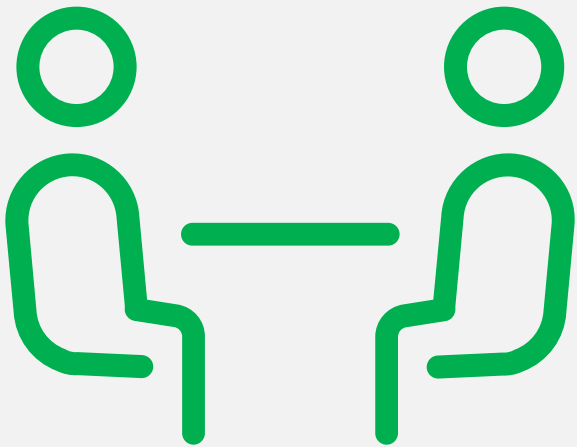


FACTORS TO BALANCE

IN CONCLUSION

What should I use?

It depends, but...



Q. What should you use for graphics?

A. It depends, of course. But with SAS ODS Graphics and Graph Template Language (GTL), there's often **no need to leave the friendly confines of Base SAS** for your charting needs!

EASY IS EASY • HARD IS POSSIBLE • ELIMINATES COSTLY CONTEXT SWITCHING • APPEALING, TERSE SYNTAX • QUALITY DOCUMENTATION • BUILT FOR DATA SCIENCE • NO DEPENDENCY WORRIES • FULL SAS LANGUAGE INTEGRATION • SUPPORT COMMUNITIES • EASY TO LEARN, WALKTHRU, MAINTAIN & ENHANCE



Thank you!

Ted Conway
ted.j.conway@gmail.com
@vivasasvegas