Sử dụng thư viện Matplotlib để vẽ một số loại biểu đồ cơ bản

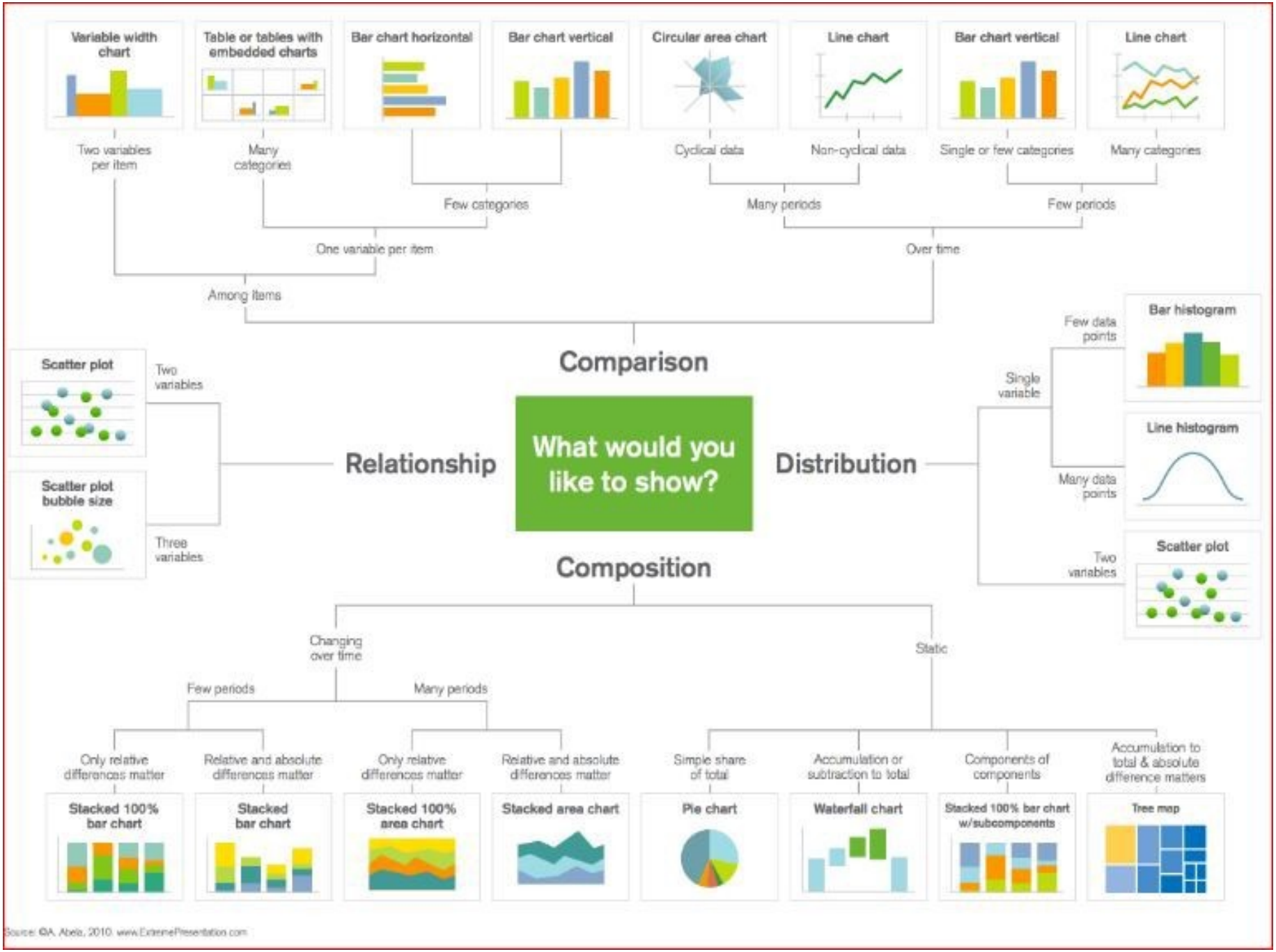
Nguyễn Thị Bích Thủy

Email: nbthuy2001@gmail.com

Tel: 0981 365 780

Nội dung

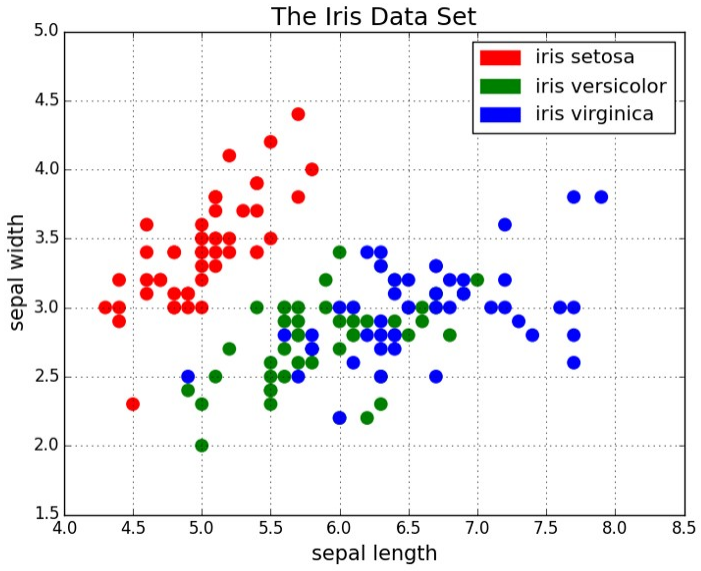
1. **Biểu đồ phân bố Scatter Plots**
2. **Biểu đồ đường Line Plots**
3. **Biểu đồ Histograms**
4. **Biểu đồ cột Bar Plots**
5. **Biểu đồ hộp Box Plots**

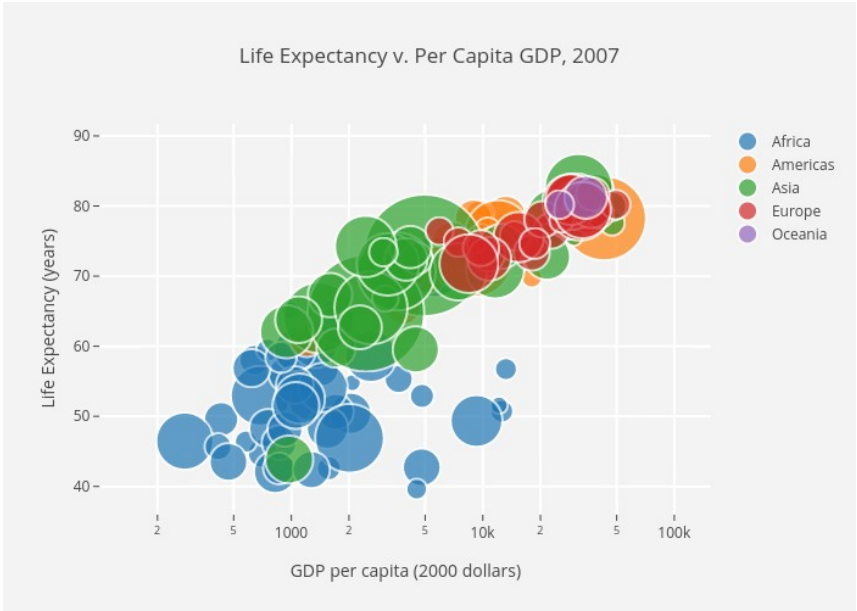


1.Biểu đồ phân bố Scatter plots

- Scatter plots được sử dụng trong các trường hợp:
  - Thể hiện mối quan hệ giữa 2 biến
  - Quan sát mối quan hệ giữa các nhóm dữ liệu bằng cách kết hợp các màu sắc
  - Trực quan hóa mối quan hệ giữa 3 biến sử dụng thêm một số các thông số, ví dụ như kích thước các điểm

Ví dụ


The Iris Data Set
iris setosa
iris versicolor
iris virginica
sepal width
sepal length
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5

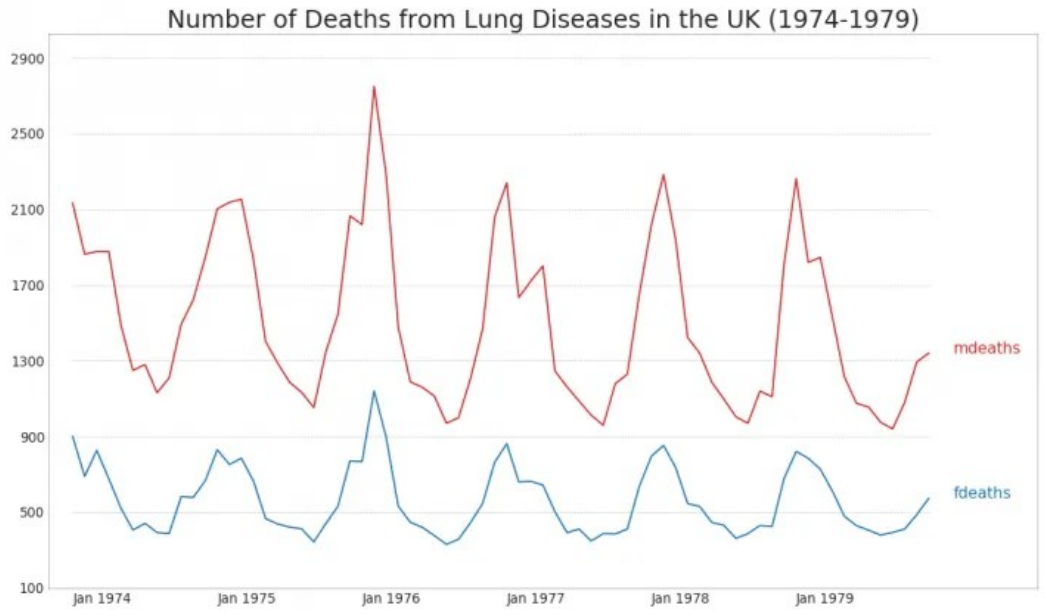

Life Expectancy v. Per Capita GDP, 2007
Life Expectancy (years)
GDP per capita (2000 dollars)
Africa
Americas
Asia
Europe
Oceania
90
80
70
60
50
40
2
5
1000
2
5
10k
2
5
100k

Code

```
# Import dataset

midwest =
pd.read_csv("https://raw.githubuse
rcontent.com/selva86/datasets/mast
er/midwest_filter.csv")


# Prepare Data

# Create as many colors as there
are unique midwest['category']

categories =
np.unique(midwest['category'])

colors =
[plt.cm.tab10(i/float(len(categori
es)-1)) for i in
range(len(categories))]


# Draw Plot for Each Category

plt.figure(figsize=(16, 10), dpi=
80, facecolor='w', edgecolor='k')

for i, category in
enumerate(categories):

    plt.scatter('area',
'poptotal',


data=midwest.loc[midwest.category=
=category, :],

                s=20, c=colors[i],
label=str(category))

# Decorations

plt.gca().set(xlim=(0.0, 0.1),
ylim=(0, 90000),

              xlabel='Area',
ylabel='Population')


plt.xticks(fontsize=12);
plt.yticks(fontsize=12)

plt.title("Scatterplot of Midwest
Area vs Population", fontsize=22)

plt.legend(fontsize=12)

plt.show()
```

2. Biểu đồ đường Line Plots

- Cho thấy sự phát triển theo thời gian của một/một số đại lượng



Code

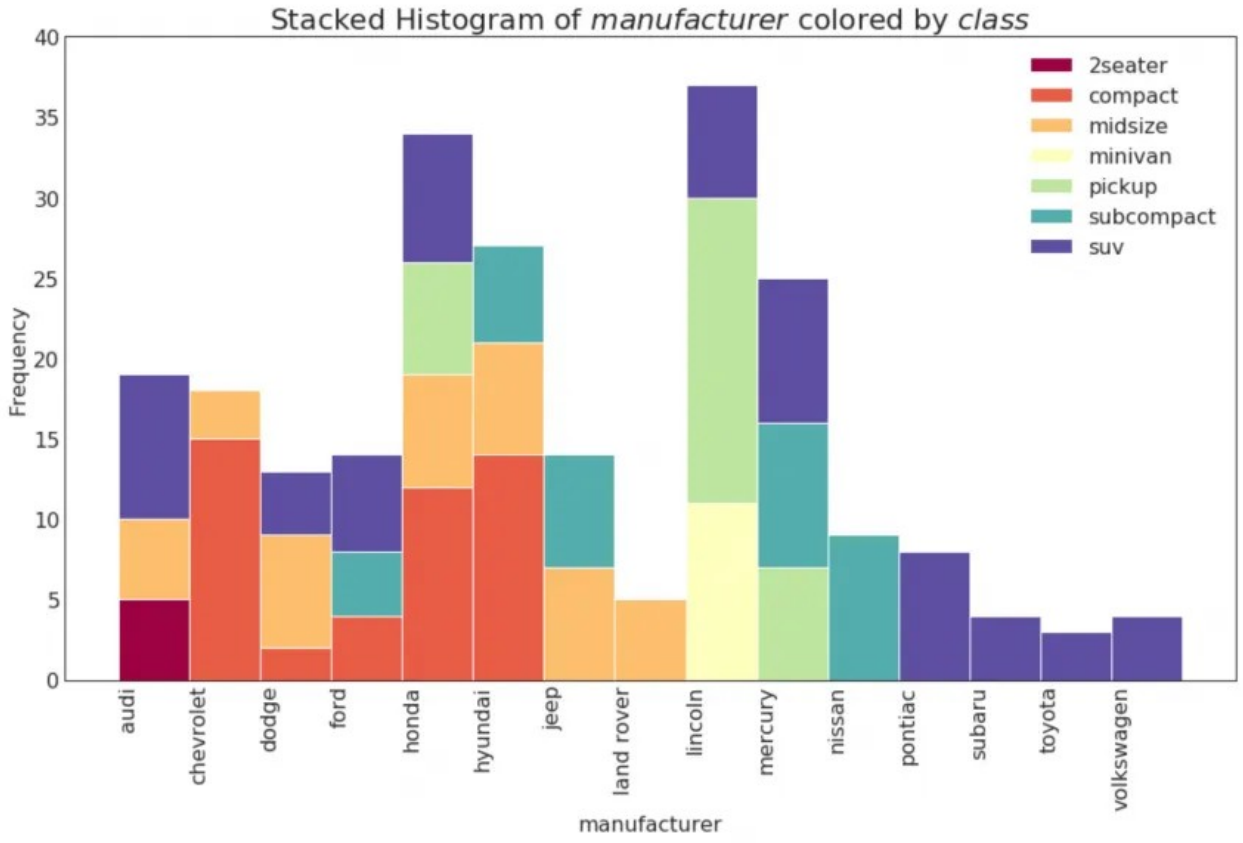
```
# Import Data
df =
  pd.read_csv('https://github.com/selva86/datasets/raw/master/mortality.csv')
# Define the upper limit, lower limit, interval of Y axis and colors
y_LL = 100
y_UL = int(df.iloc[:, 1:].max().max()*1.1)
y_interval = 400
mycolors = ['tab:red', 'tab:blue', 'tab:green', 'tab:orange']
# Draw Plot and Annotate
fig, ax = plt.subplots(1,1,figsize=(16, 9), dpi= 80)
columns = df.columns[1:]
for i, column in enumerate(columns):
    plt.plot(df.date.values, df[column].values, lw=1.5, color=mycolors[i])
    plt.text(df.shape[0]+1, df[column].values[-1], column, fontsize=14,
  color=mycolors[i])
```

Code (tiếp)

```
# Draw Tick lines
for y in range(y_LL, y_UL, y_interval):
    plt.hlines(y, xmin=0, xmax=71, colors='black', alpha=0.3, linestyles="--",
   lw=0.5)
# Decorations
plt.tick_params(axis="both", which="both", bottom=False, top=False,
                labelbottom=True, left=False, right=False, labelleft=True)
# Lighten borders
plt.gca().spines["top"].set_alpha(.3)
plt.gca().spines["bottom"].set_alpha(.3)
plt.gca().spines["right"].set_alpha(.3)
plt.gca().spines["left"].set_alpha(.3)
plt.title('Number of Deaths from Lung Diseases in the UK (1974-1979)', fontsize=22)
plt.yticks(range(y_LL, y_UL, y_interval), [str(y) for y in range(y_LL, y_UL,
   y_interval)], fontsize=12)
plt.xticks(range(0, df.shape[0], 12), df.date.values[::12],
   horizontalalignment='left', fontsize=12)
plt.ylim(y_LL, y_UL)
plt.xlim(-2, 80)
plt.show()
```

Histograms/Overlaid histograms

- Biểu diễn tần số của các biến



Code

```
# Import Data

df =
  pd.read_csv("https://github.com/sel
  va86/datasets/raw/master/mpg_ggplot
  2.csv")

# Prepare data

x_var = 'manufacturer’ groupby_var =
  'class'

df_agg = df.loc[:, [x_var,
  groupby_var]].groupby(groupby_var)

vals = [df[x_var].values.tolist() for
  i, df in df_agg]

# Draw

plt.figure(figsize=(16,9), dpi=
  80)colors =
  plt.cm.Spectral(i/float(len(vals)-
  1)) for i in range(len(vals))]

n, bins, patches = plt.hist(vals,
  df[x_var].unique().__len__(),
  stacked=True, density=False,
  color=colors[:len(vals)])
```

```
# Decoration

plt.legend({group:col for
  group, col in
  zip(np.unique(df[groupby_v
  ar]).tolist(),
  colors[:len(vals)])})

plt.title(f"Stacked
  Histogram of ${x_var}$
  colored by ${groupby_var}
  $", fontsize=22)

plt.xlabel(x_var)

plt.ylabel("Frequency")

plt.ylim(0, 40)

plt.xticks(ticks=bins,
  labels=np.unique(df[x_var]
  ).tolist(), rotation=90,
  horizontalalignment='left'
  )

plt.show()
```

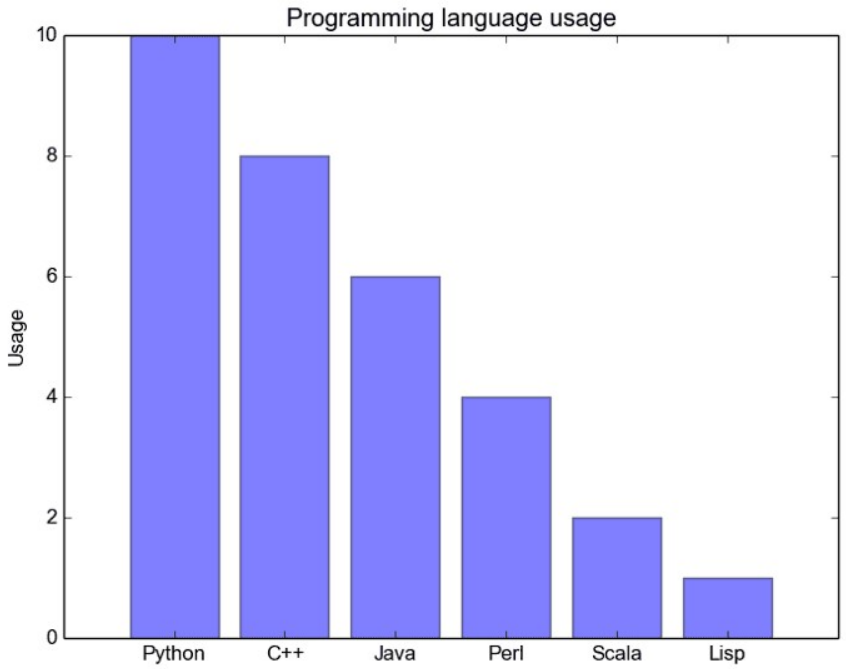
Biểu đồ cột Bar Plots

- Sử dụng hiệu quả khi có ít hơn 10 biến
- Khi quá nhiều mục: gây lộn xộn và khó hiểu.
- Dễ dàng nhận thấy sự khác biệt giữa các danh mục dựa trên kích thước của các cột
- Có thể phân chia thành 3 loại khác nhau: thông thường, nhóm và xếp chồng

Examples

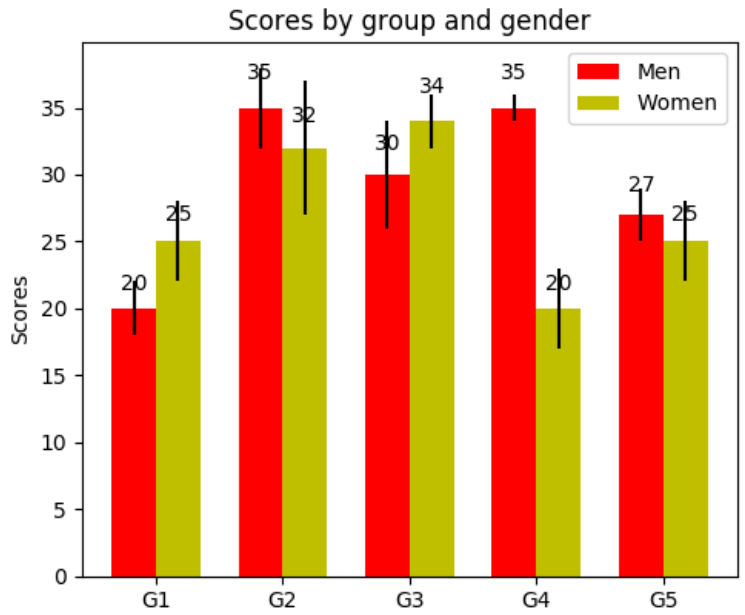
Regular bar plot

```
def barplot(x_data, y_data, error_data,
    x_label="", y_label="", title=""):

  _, ax = plt.subplots()

 # Draw bars, position them in the center of t
    tick mark on the x-axis ax.bar(x_data,
    y_data, color = '#539caf', align = 'center')

 # Draw error bars to show standard deviatio
    set ls to 'none' # to remove line between
    points

  ax.errorbar(x_data, y_data, yerr = error_dat
    color = '#297083', ls = 'none', lw = 2,
    capthick = 2) ax.set_ylabel(y_label)
    ax.set_xlabel(x_label)

  ax.set_title(title)
```



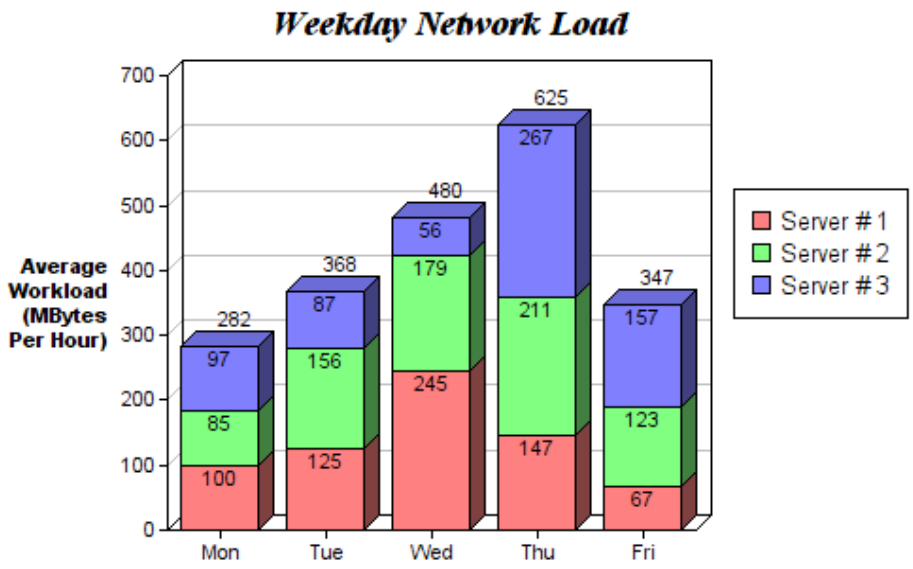
Biểu đồ cột nhóm

```
def groupedbarplot(x_data,  y_data_list, colors,
    y_data_names="", x_label="", y_label="", title=""):

  _, ax = plt.subplots()

 # Total width for all bars at one x location total_width = 0.8

# Width of each individual bar

  ind_width = total_width / len(y_data_list)

# This centers each cluster of bars about the x tick mark

  alteration = np.arange(-(total_width/2), total_width/2,
    ind_width)

# Draw bars, one category at a time for i in range(0,
    len(y_data_list)):

# Move the bar to the right on the x-axis so it doesn't #
    overlap with previously drawn ones

  ax.bar(x_data + alteration[i], y_data_list[i], color =
    colors[i], label = y_data_names[i], width = ind_width)
    ax.set_ylabel(y_label)

  ax.set_xlabel(x_label)

  ax.set_title(title)

  ax.legend(loc = 'upper right')
```



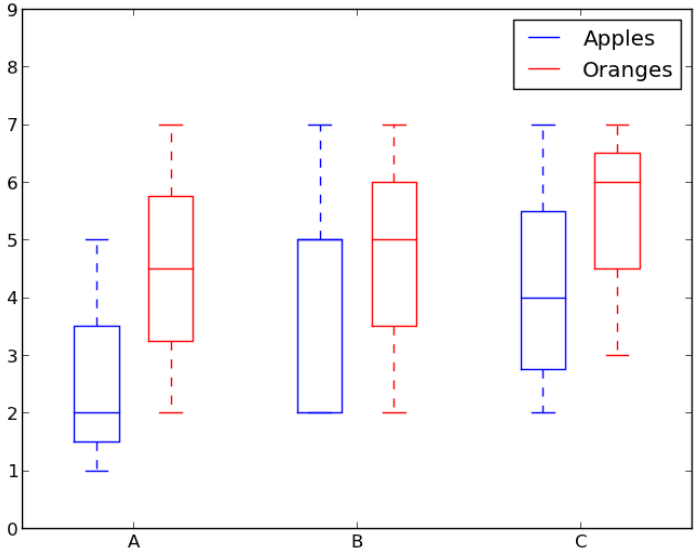
Biểu đồ cột chồng

```
def stackedbarplot(x_data, y_data_list, colors,
    y_data_names="", x_label="", y_label="", title=""):

  _, ax = plt.subplots()

# Draw bars, one category at a time

  for i in range(0, len(y_data_list)):

        if i == 0:

            ax.bar(x_data, y_data_list[i],      color =
    colors[i], align = 'center',      label =
    y_data_names[i])

        else:

# For each category after the first, the bottom of the #
    bar will be the top of the last category
    ax.bar(x_data, y_data_list[i], color = colors[i], bottom
    = y_data_list[i - 1], align = 'center', label =
    y_data_names[i]) ax.set_ylabel(y_label)

  ax.set_xlabel(x_label)

  ax.set_title(title)

  ax.legend(loc = 'upper right')
```



Biểu đồ hộp Box Plots

- Biểu diễn các thông tin về giá trị trung bình, min, max, 25%, 75% của các đại lượng



Code

```
def boxplot(x_data, y_data, base_color="#539caf", median_color="#297083", x_label="", y_label="", title=""):
    _, ax = plt.subplots()
# Draw boxplots, specifying desired style ax.boxplot(y_data # patch_artist must be True to control box fill ,
    patch_artist = True
# Properties of median line ,
    medianprops = {'color': median_color}
# Properties of box ,
    boxprops = {'color': base_color, 'facecolor': base_color}
# Properties of whiskers ,
    whiskerprops = {'color': base_color}
# Properties of whisker caps ,
    capprops = {'color': base_color})
# By default, the tick label starts at 1 and increments by 1 for # each box drawn. This sets the labels to the ones we want
    ax.set_xticklabels(x_data)
    ax.set_ylabel(y_label)
    ax.set_xlabel(x_label)
    ax.set_title(title)
```

Thank you!