

PYTHON FOR MACHINE LEARNING, DATA SCIENCE & DATA VISUALIZATION

Bài 7: *Trực quan hóa dữ liệu - Seaborn*

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-csdl/Python-for-Machine-Learning-Data-Science--Data-Visualization-Python-cho-may-hoc-Khoa-hoc-du-lieu-va-Truc-quan-hoa-du-lieu_191

Nội dung

1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Giới thiệu

- ❑ Với Analytics, cách tốt nhất để có được thông tin chi tiết là bằng cách trực quan hóa dữ liệu. Dữ liệu có thể được hình dung bằng cách biểu diễn như là các ô dễ hiểu, dễ khám phá và nắm bắt. Dữ liệu này giúp thu hút sự chú ý của các yếu tố chính.
- ❑ Để phân tích một tập hợp dữ liệu bằng Python, có thể sử dụng Matplotlib, một thư viện vẽ 2D được triển khai rộng rãi, hoặc Seaborn, là một thư viện trực quan bằng Python, được xây dựng trên Matplotlib.

Giới thiệu

❑ Đặc điểm của Seaborn

- Seaborn được xây dựng trên thư viện trực quan cốt lõi của Python là Matplotlib. Seaborn là một phần bổ sung, và không phải là một sự thay thế cho Matplotlib.
- Tuy nhiên, Seaborn có một số tính năng rất quan trọng:
 - Trực quan hóa dữ liệu đơn biến và hai biến
 - Phù hợp và hình dung các mô hình hồi quy tuyến tính
 - Lập kế hoạch dữ liệu chuỗi thời gian thống kê
 - Seaborn hoạt động tốt với cấu trúc dữ liệu NumPy và Pandas
 - Nó đi kèm với các theme để tạo kiểu Matplotlib đồ họa



Giới thiệu

❑ Seaborn Vs Matplotlib

- Matplotlib “cố gắng làm mọi việc dễ dàng hơn và làm cho việc khó khăn có thể giải quyết”. Seaborn hỗ trợ các phương pháp trực quan hóa dữ liệu phức tạp hơn nhưng vẫn cần matplotlib.
- Seaborn giúp giải quyết vấn đề lớn mà Matplotlib phải đối mặt, đó là:
 - Các tham số Matplotlib mặc định (Seaborn hoạt động với các tham số tùy chỉnh khác nhau)
 - Matplotlib làm việc với Dataframe không suôn sẻ, chỉ làm việc với các cột dữ liệu cụ thể trên Dataframe. (Seaborn làm việc với Dataframe và array chứa toàn bộ dữ liệu)

Nội dung

1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Vẽ biểu đồ với Seaborn

❑ Cài đặt seaborn

- Sử dụng: `pip install seaborn`

❑ Import thư viện

- `import pandas as pd`
- `from matplotlib import pyplot as plt`
- `import seaborn as sns`

Vẽ biểu đồ với Seaborn

❑ Tải dữ liệu

- Khi làm việc với Seaborn, ta có thể sử dụng một trong các bộ dữ liệu tích hợp mà chính thư viện cung cấp hoặc bạn có tải dữ liệu từ ngoài vào khung dữ liệu Pandas.

Vẽ biểu đồ với Seaborn

- Tải dữ liệu từ Built-in Seaborn Data Set

- Sử dụng `load_dataset()` function để tải dữ liệu từ Seaborn dataset.
- Sử dụng `get_dataset_names()` để xem tất cả các dataset có sẵn trong thư viện Seaborn

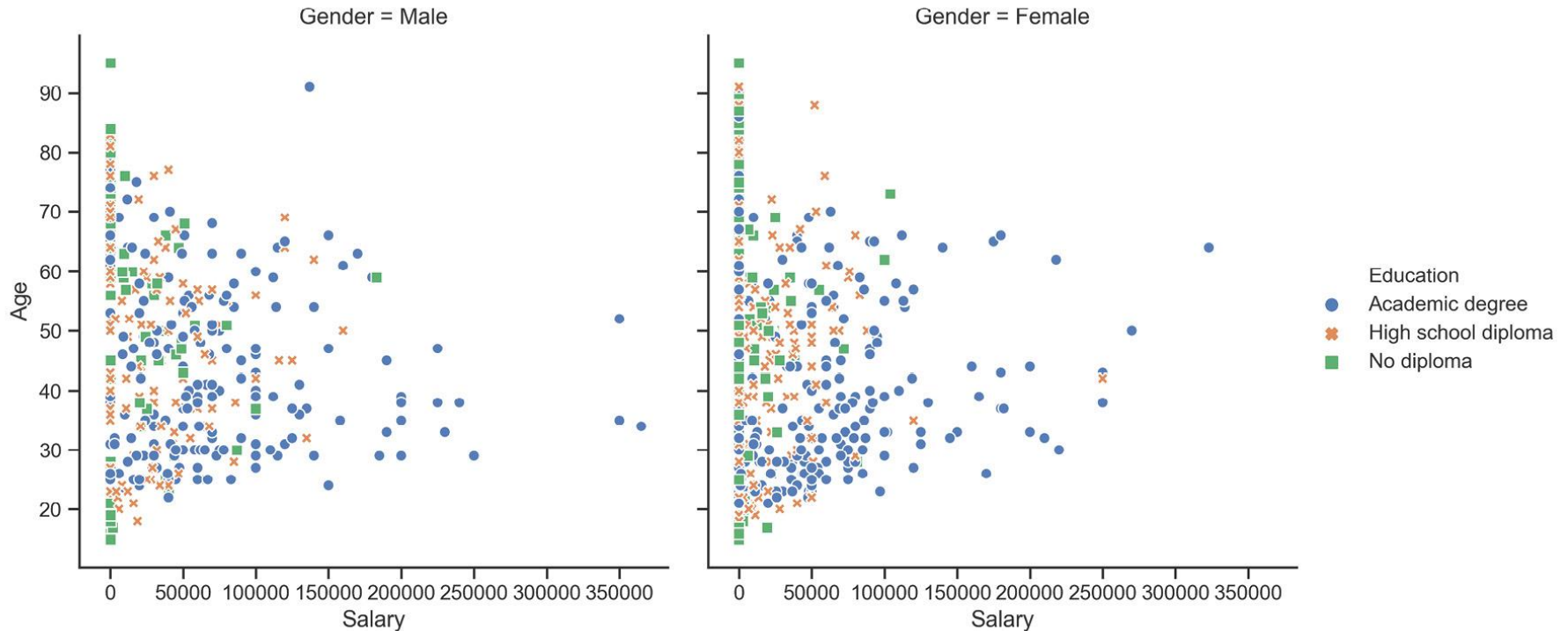
- Ví dụ:

```
# Ví dụ: tải dữ liệu iris  
iris = sb.load_dataset("iris")
```

```
# Ví dụ: xem các dataset:  
print(sb.get_dataset_names())
```

```
['anscombe', 'attention',  
'brain_networks', 'car_crashes',  
'diamonds', 'dots', 'exercise',  
'flights', 'fmri', 'gammas',  
'iris', 'mpg', 'planets', 'tips',  
'titanic']
```

Vẽ biểu đồ với Seaborn



```
sns.set(style="ticks")
data = pd.read_csv("data/salary.csv")
sns.relplot(x="Salary", y="Age", hue="Education",
            style="Education", col="Gender", data=data)
```

Vẽ biểu đồ với Seaborn

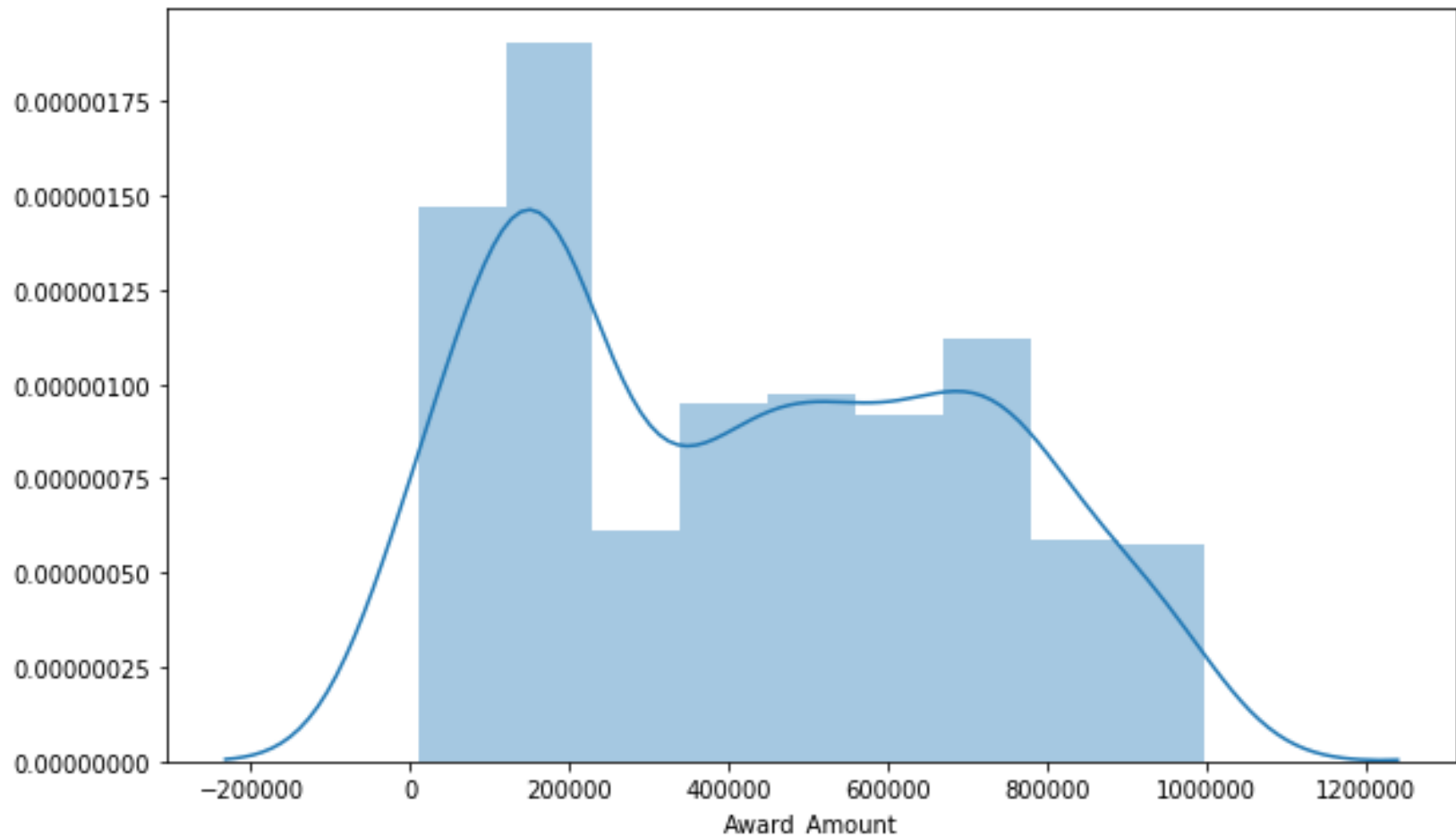
□ distplot

- Sử dụng `sb.distplot()`: vẽ một phân bố quan sát đơn biến
- Tương tự như histogram
- Mặc định, `distplot` tạo luôn một Gaussian Kernel Density Estimate (KDE)

Vẽ biểu đồ với Seaborn

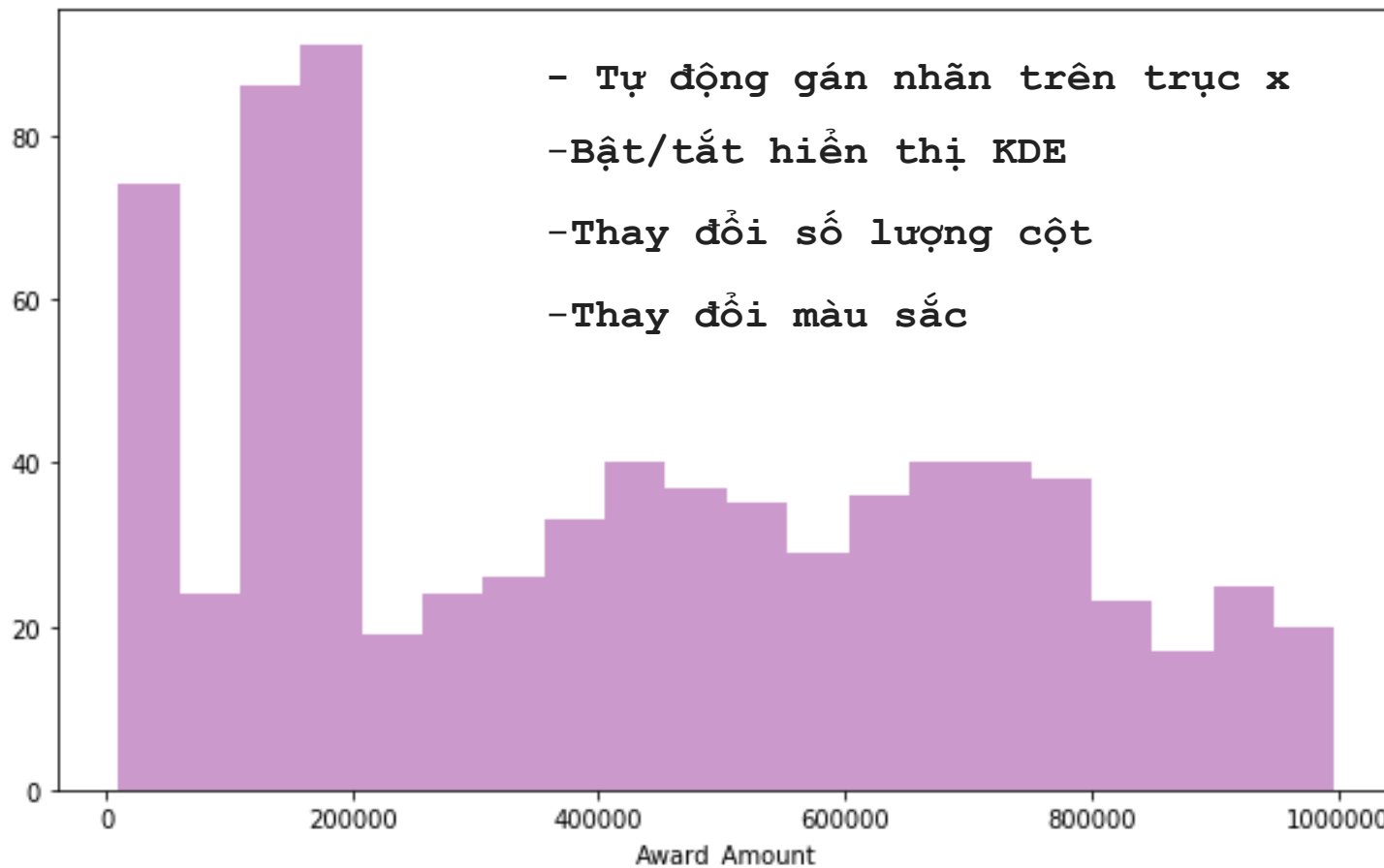
```
# Display a Seaborn distplot  
plt.figure(figsize=(10,6))  
sns.distplot(df['Award_Amount'])  
plt.show()
```

- Tự động gán nhãn trên trục x



Vẽ biểu đồ với Seaborn

```
# Create a distplot
plt.figure(figsize=(10,6))
sns.distplot(df['Award_Amount'], kde=False, bins=20, color='purple')
# Display a plot
plt.show()
```



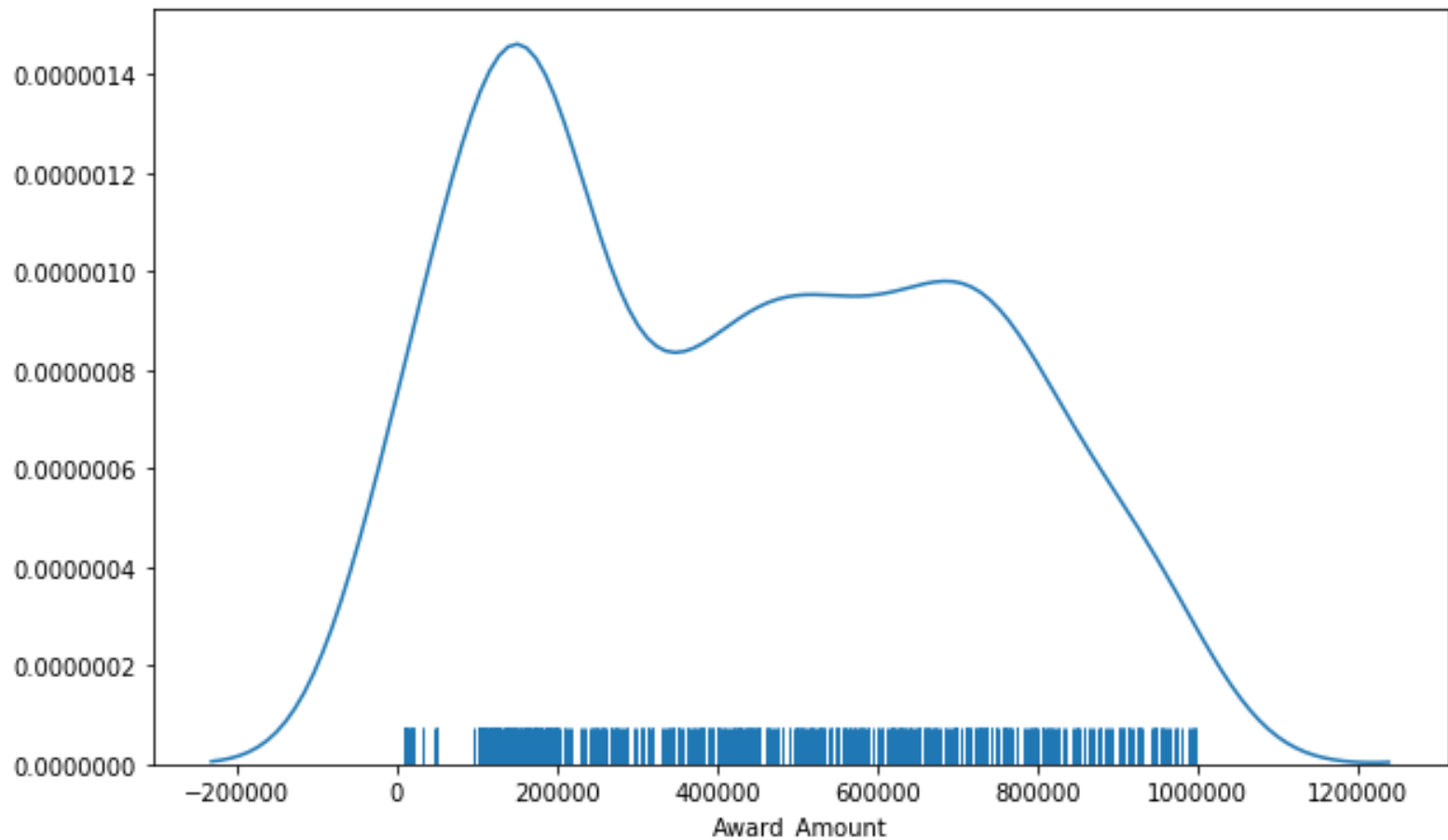
Vẽ biểu đồ với Seaborn

□ Distribution plot

- Một rug plot là một cách thay thế để hiển thị sự phân phối của dữ liệu.
- Một kde curve và một rug plot có thể kết hợp với nhau

Vẽ biểu đồ với Seaborn

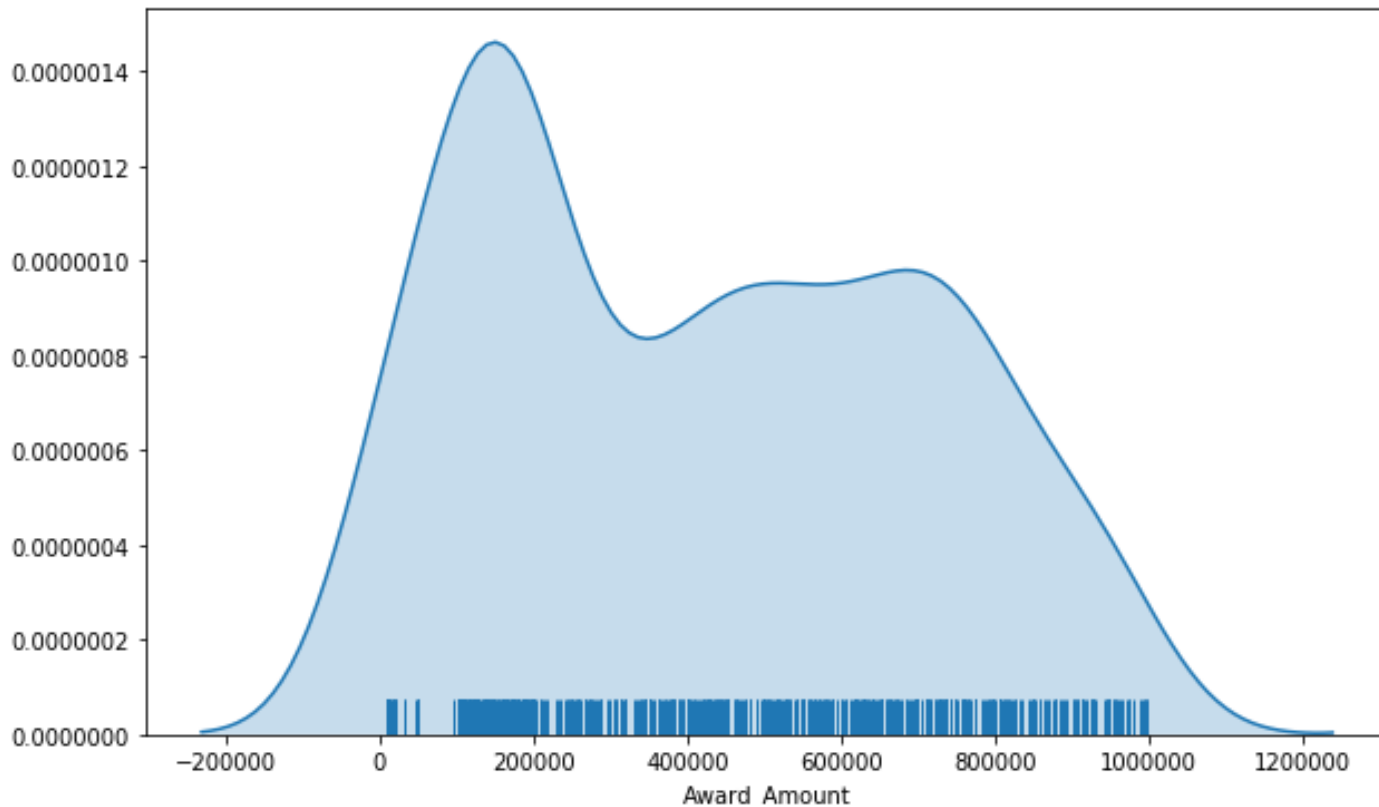
```
# Create a distplot of the Award Amount
plt.figure(figsize=(10,6))
sns.distplot(df['Award_Amount'], hist=False, rug=True)
# Plot the results
plt.show()
```



Vẽ biểu đồ với Seaborn

- Distplot có kdeplot và rugplot

```
# Create a distplot of the Award Amount
plt.figure(figsize=(10,6))
sns.distplot(df['Award_Amount'], hist=False, rug=True, kde_kws={'shade':'kde_kws'})
# Plot the results
plt.show()
```



Vẽ biểu đồ với Seaborn

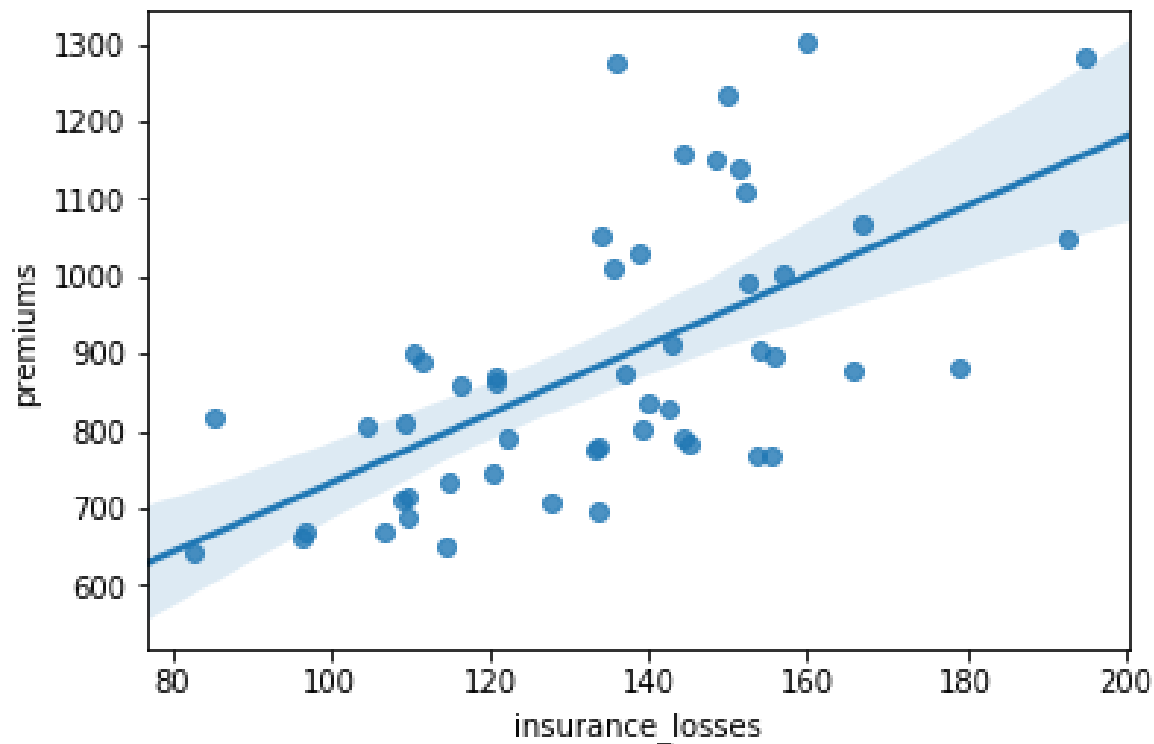
□ Regression plot

- regplot tạo ra một scatter plot với một regression line
- Cách sử dụng tương tự như distplot
- Các biến dữ liệu và x và y phải được xác định

Vẽ biểu đồ với Seaborn

```
sns.regplot(data=df,  
            x="insurance_losses",  
            y="premiums")
```

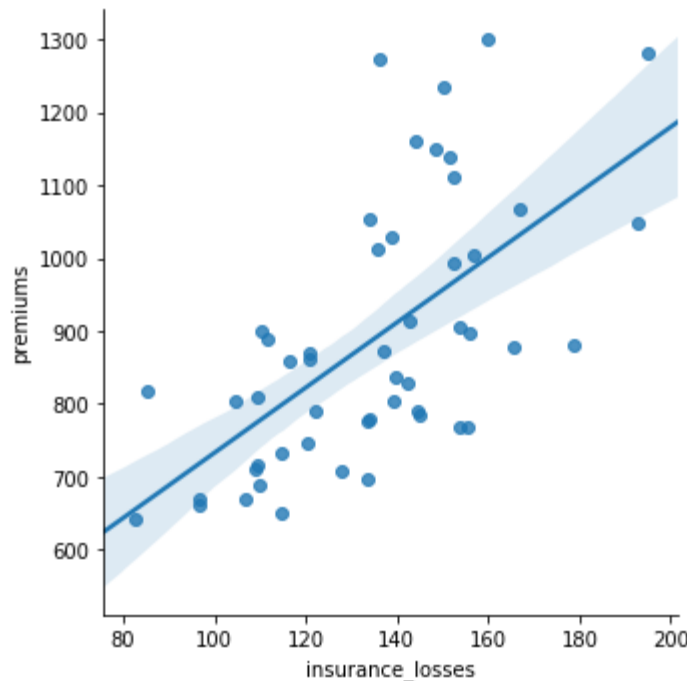
```
# Display the plot  
plt.show()
```



Vẽ biểu đồ với Seaborn

❑ Implot() tạo ra dựa trên regplot()

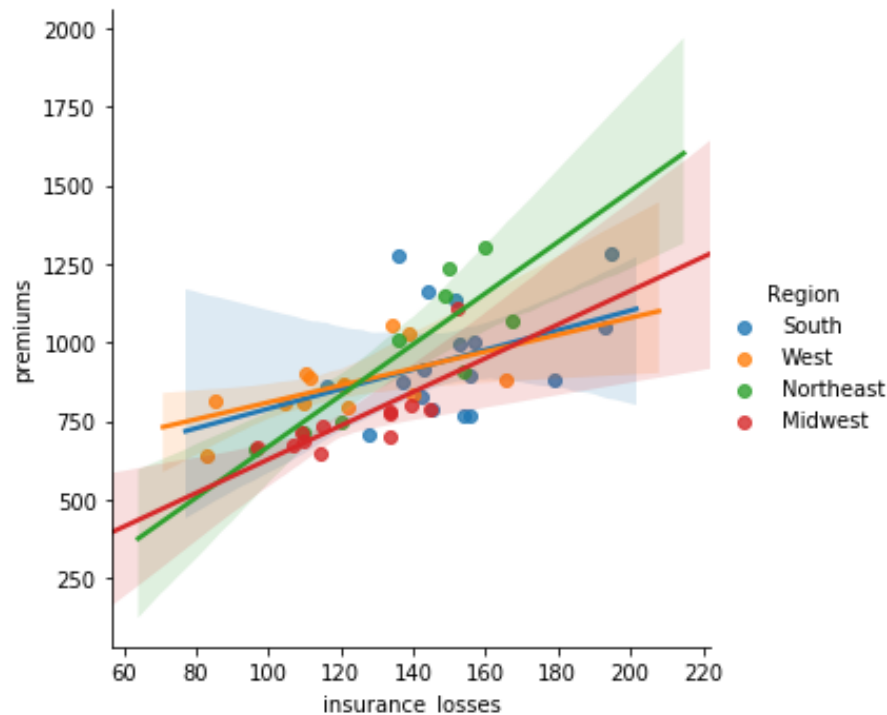
```
sns.lmplot(data=df, x="insurance_losses", y="premiums")  
# Display the plot  
plt.show()
```



Vẽ biểu đồ với Seaborn

❑ Implot() sắp xếp dữ liệu nhóm theo màu sắc (hue)

```
# Create a regression plot using hue
sns.lmplot(data=df, x="insurance_losses", y="premiums", hue="Region")
# Show the results
plt.show()
```

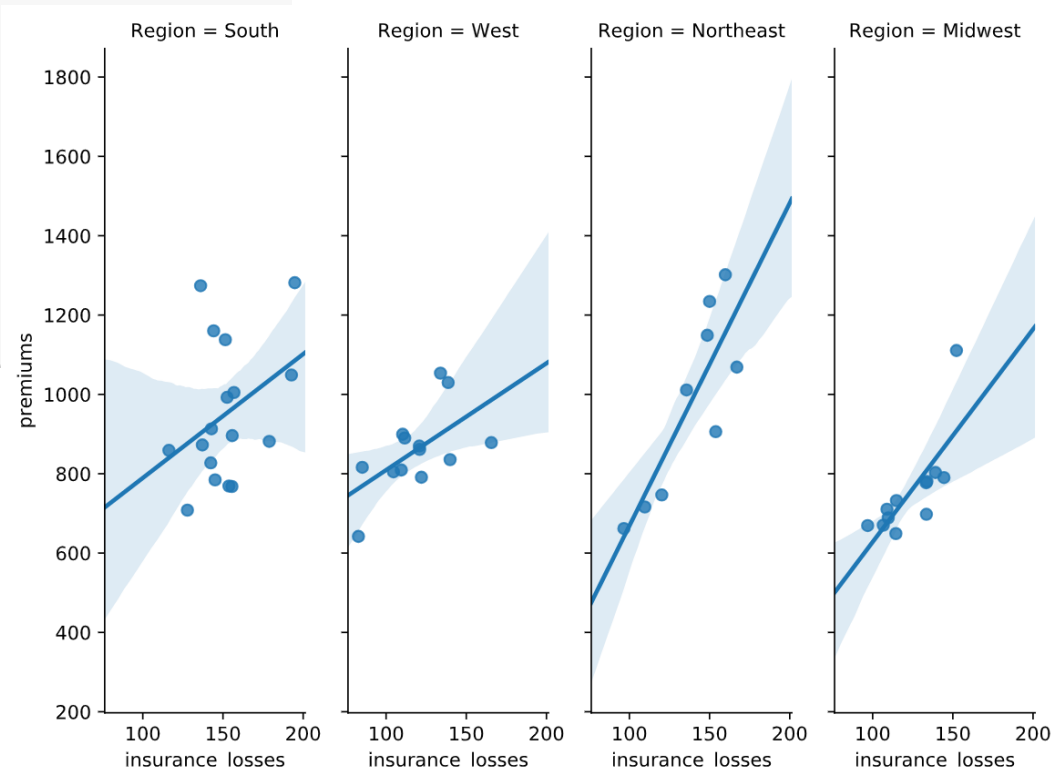


Vẽ biểu đồ với Seaborn

❑ Implot() sắp xếp dữ liệu nhóm theo cột (col)/ hoặc dòng (row)

```
# Create a regression plot with multiple rows
plt.figure(figsize=(10,10))
sns.lmplot(data=df,
           x="insurance_losses",
           y="premiums",
           col="Region")

# Show the plot
plt.show()
```



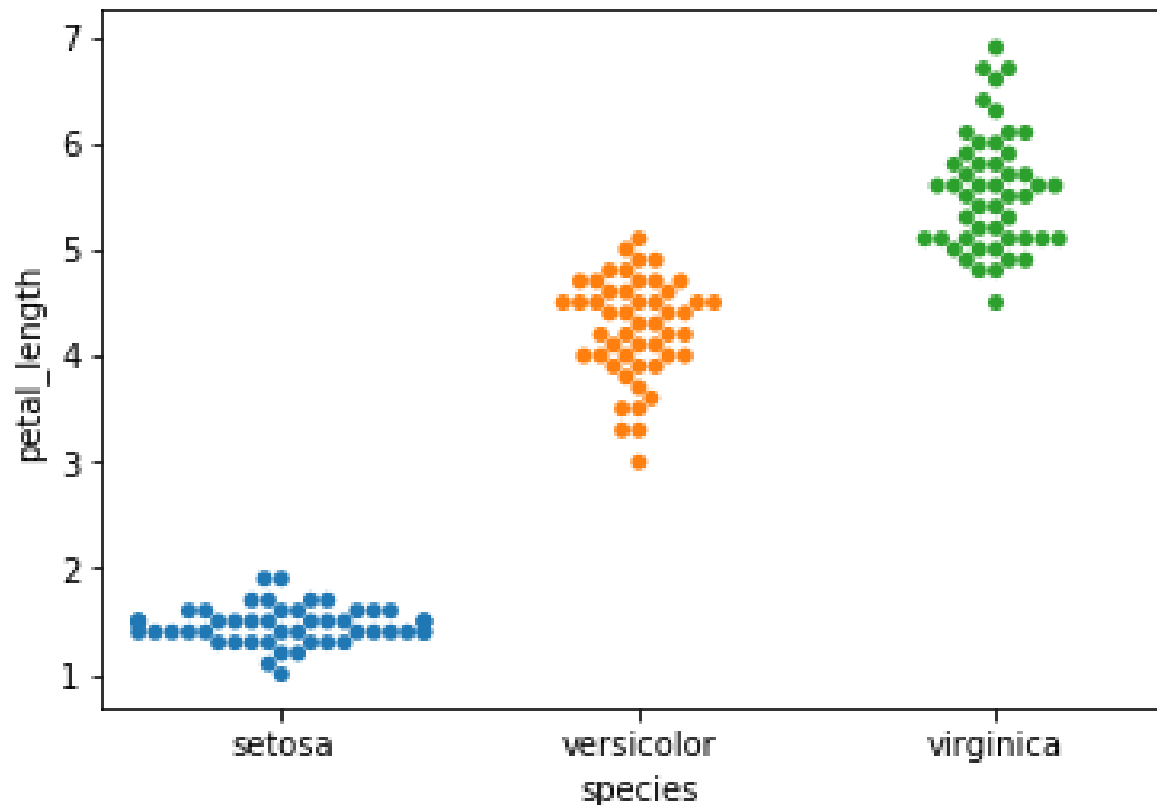
Vẽ biểu đồ với Seaborn

❑ Scatter plot

- Dạng 1: `sb.swarmplot()`: vẽ một scatterplot phân loại với các điểm không chồng chéo

Vẽ biểu đồ với Seaborn

```
# Construct iris plot
# x: dữ liệu hiển thị trên trục hoành, y: dữ liệu hiển thị trên trục tung, data: bộ dữ liệu
sb.swarmplot(x="species", y="petal_length", data=iris)
# Show plot
plt.show()
```



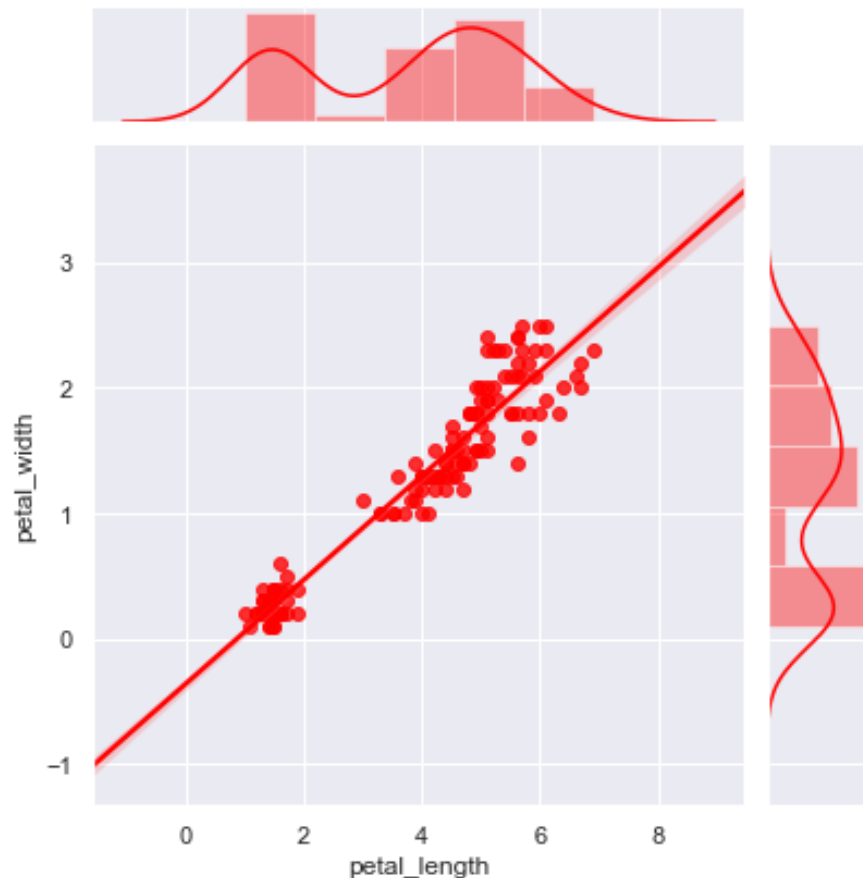
Vẽ biểu đồ với Seaborn

❑ Scatter plot

- Dạng 2: `sb.jointplot()`: vẽ một scatterplot với hai biến
 - Xem thông tin chi tiết của function: <https://seaborn.pydata.org/generated/seaborn.jointplot.html>

Vẽ biểu đồ với Seaborn

```
sb.set()  
# kind: 'scatter', 'reg', 'resid', 'kde', or 'hex'  
sb.jointplot(x = 'petal_length', y = 'petal_width', data = iris, kind = 'reg', color='red')  
plt.show()
```



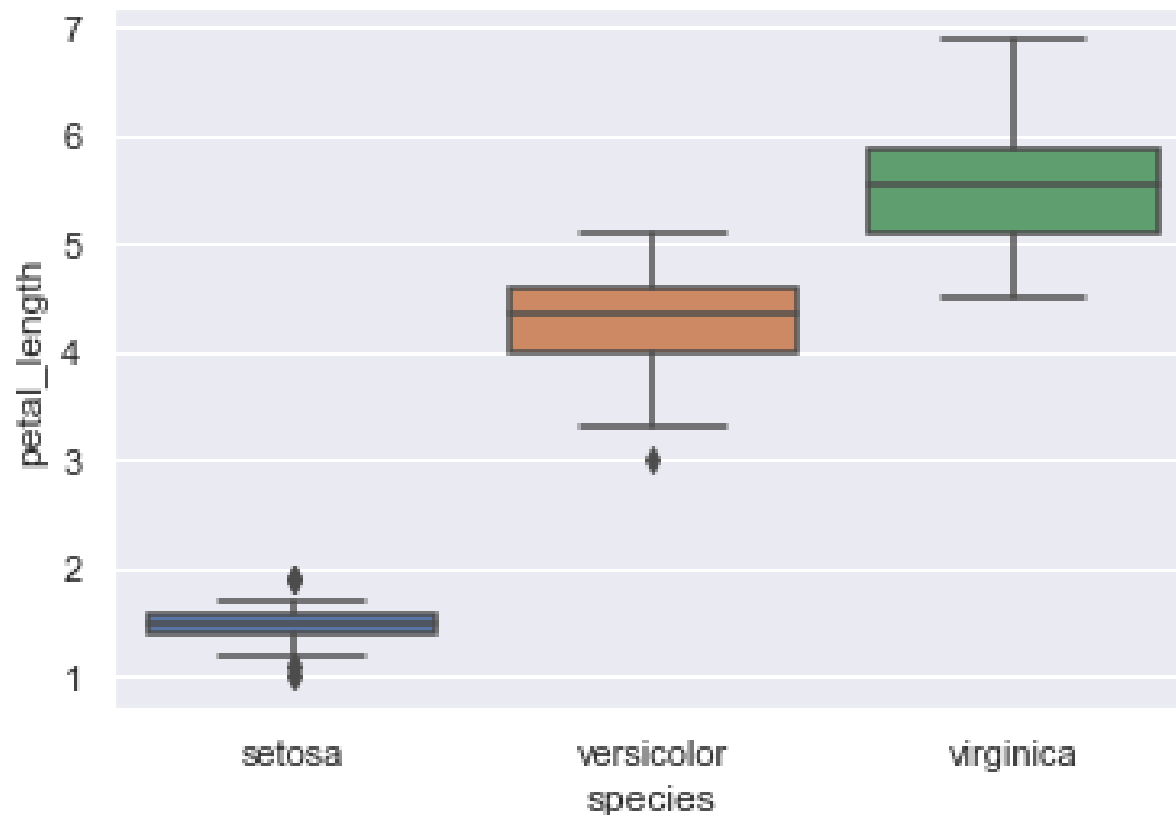
Vẽ biểu đồ với Seaborn

□ Boxplot

- Sử dụng `sb.boxplot()`: vẽ plot để hiển thị các phân phối liên quan đến category
- Xem thông tin chi tiết của function: <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

Vẽ biểu đồ với Seaborn

```
sb.boxplot(x = "species", y = "petal_length", data = iris)  
plt.show()
```



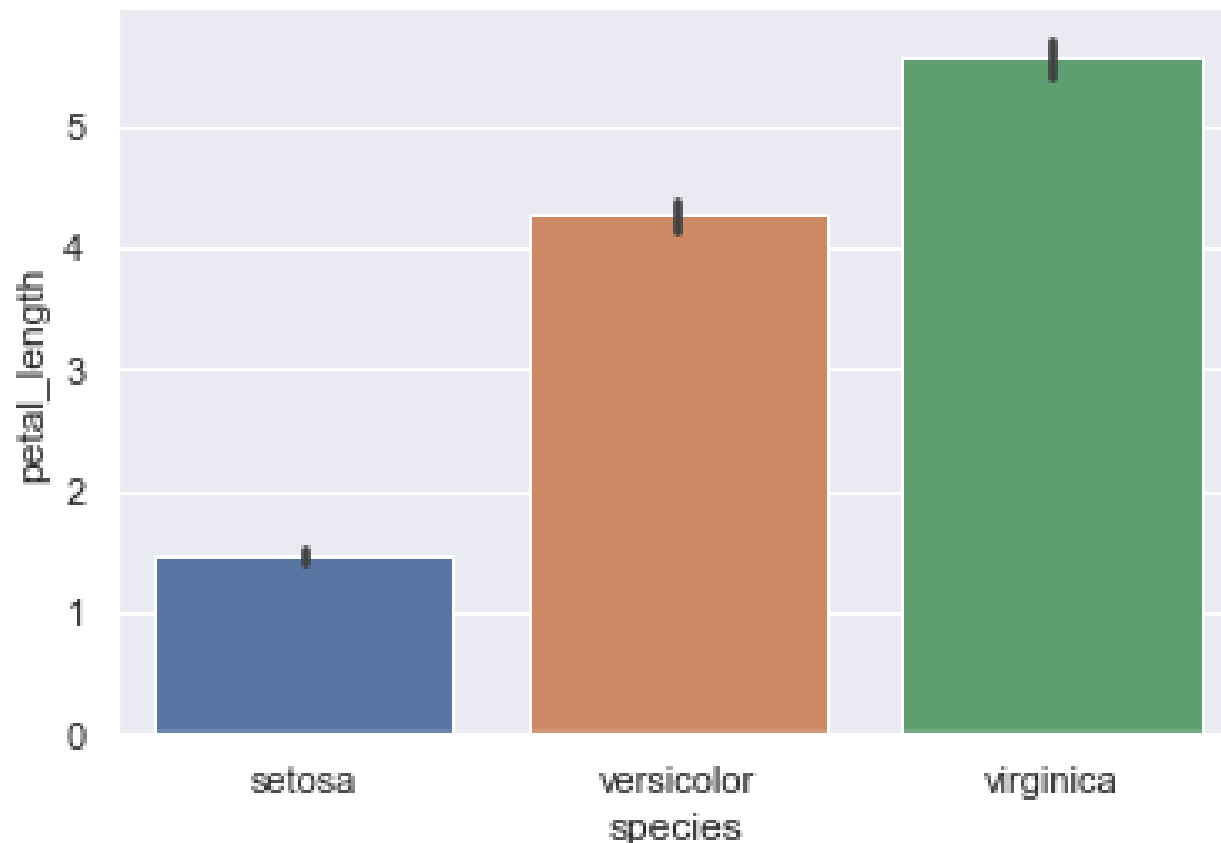
Vẽ biểu đồ với Seaborn

□ Barplot

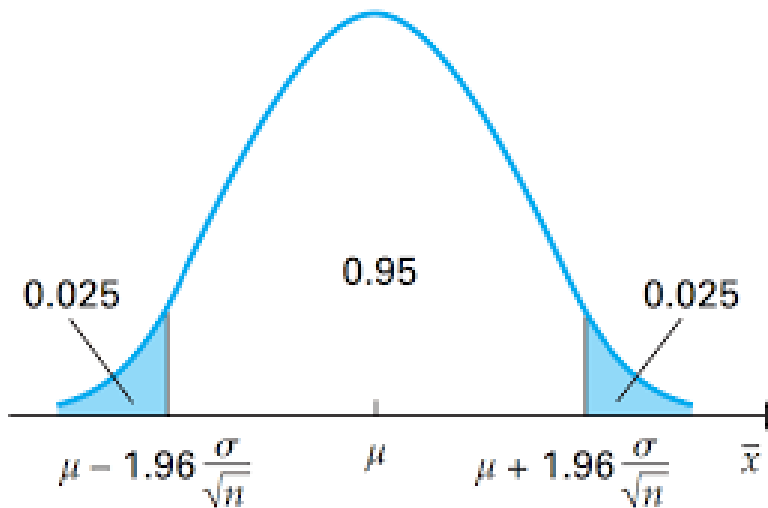
- Sử dụng `sb.barplot()`: vẽ plot để hiển thị dữ liệu dưới dạng khối hình chữ nhật
- Xem thông tin chi tiết của function: <https://seaborn.pydata.org/generated/seaborn.barplot.html>

Vẽ biểu đồ với Seaborn

```
sb.barplot(x = "species", y = "petal_length", data = iris)  
plt.show()
```



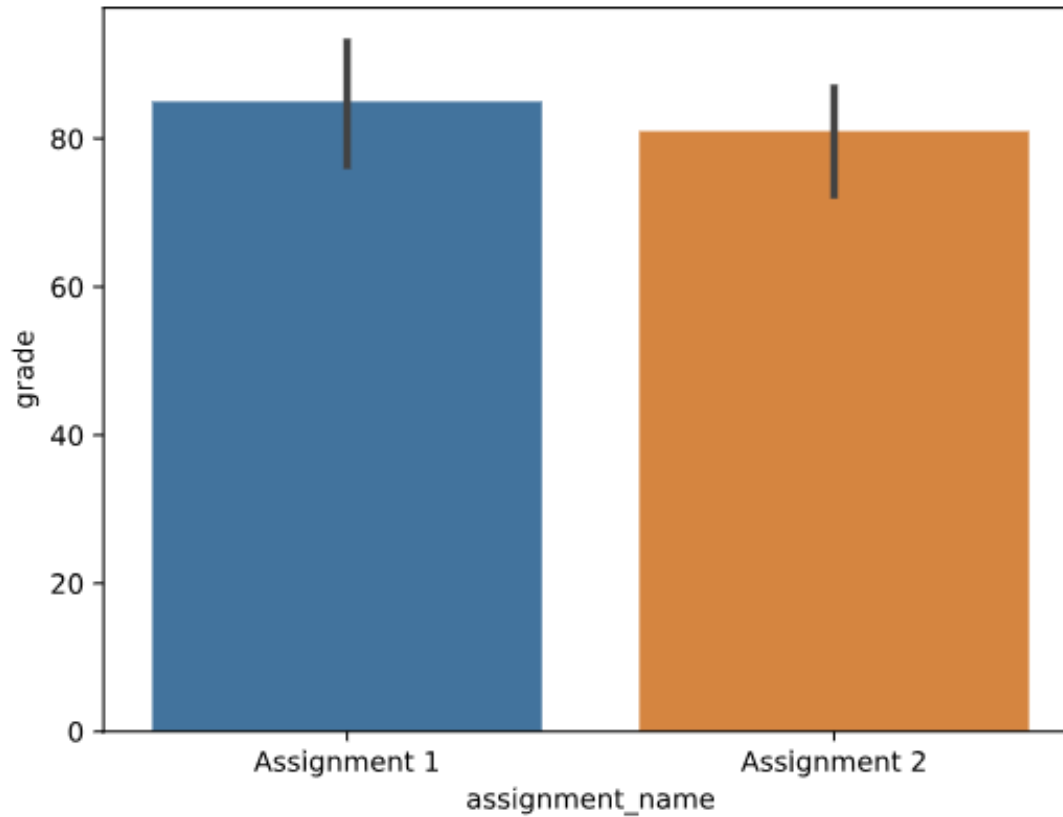
Khoảng tin cậy – Confidence Interval(CI)



Ví dụ: Đo chiều cao của 40 người chọn ngẫu nhiên được chiều cao TB là 175 cm. Biết độ lệch chuẩn của chiều cao là 20cm.

Với mức độ tin cậy 95%, hãy tính CI

ERROR BAR



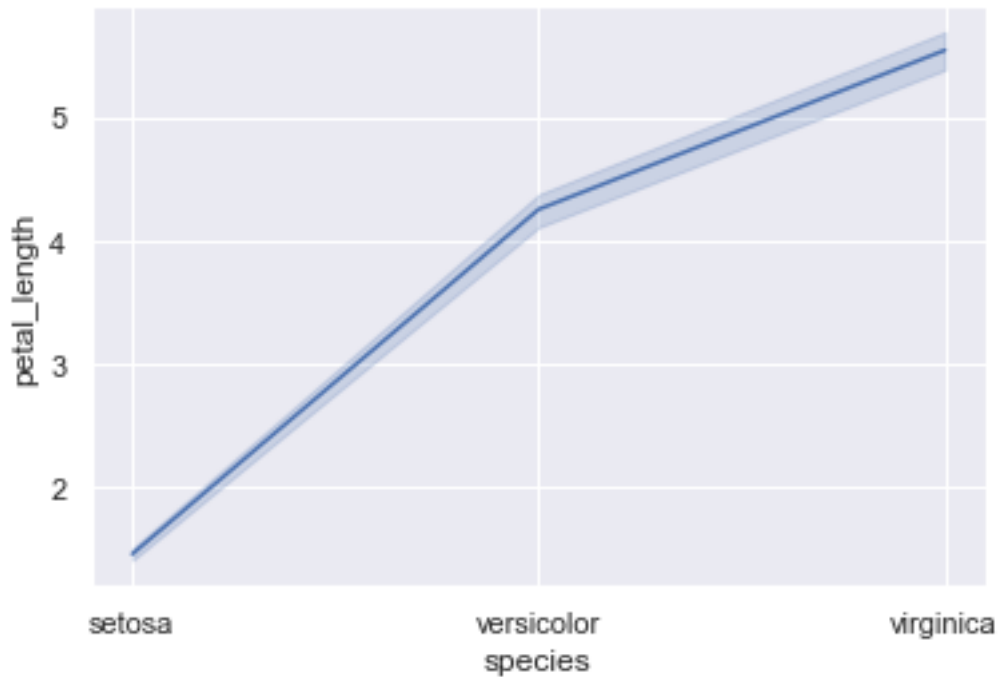
Vẽ biểu đồ với Seaborn

□ Lineplot

- Sử dụng `sb.lineplot()`: vẽ plot để hiển thị dữ liệu dưới dạng line
- Xem thông tin chi tiết của function: <https://seaborn.pydata.org/generated/seaborn.lineplot.html>

Vẽ biểu đồ với Seaborn

```
sb.lineplot(x = "species", y = "petal_length", data = iris)  
plt.show()
```



Nội dung

1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Seaborn styles

□ Sử dụng style

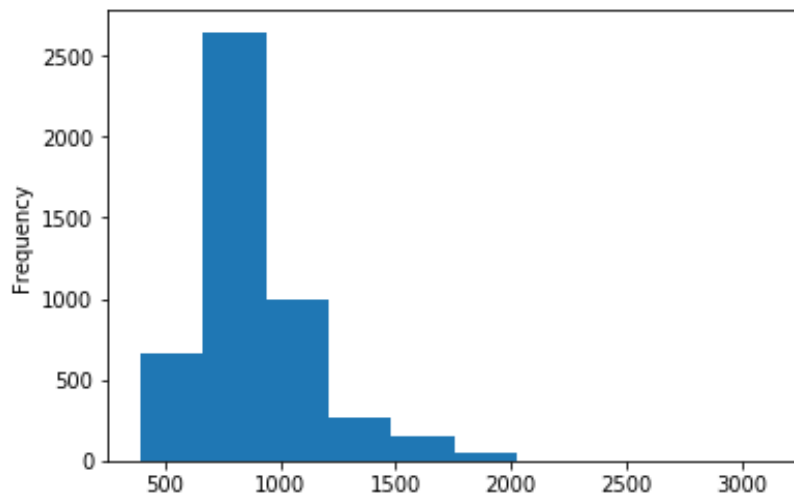
- Seaborn có các cấu hình mặc định theme có thể được áp dụng bằng `sns.set ()`
- Những style này cũng có thể ghi đè style cho các matplotlib plot và pandas plot

Seaborn styles

- Ví dụ

Pandas histogram

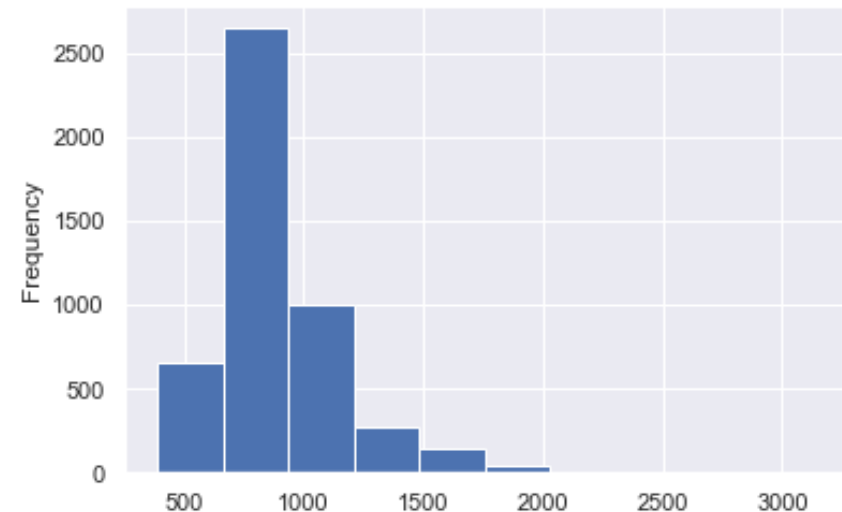
```
# Plot the pandas histogram
df['fmr_2'].plot.hist()
plt.show()
plt.clf()
```



Mặc định

```
# Set the default seaborn style
sns.set()
```

```
# Plot the pandas histogram again
df['fmr_2'].plot.hist()
plt.show()
plt.clf()
```

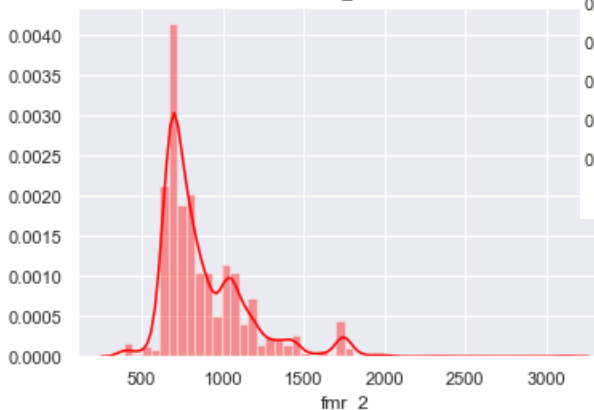
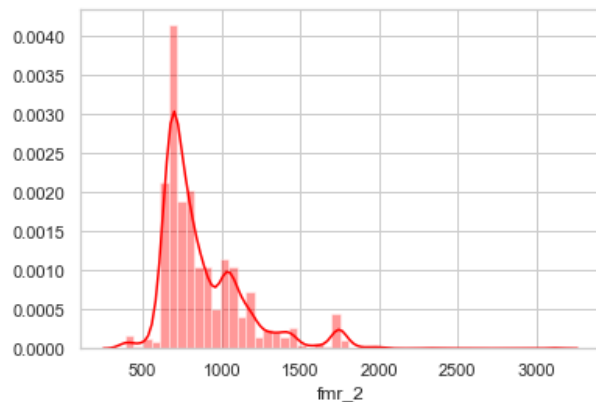
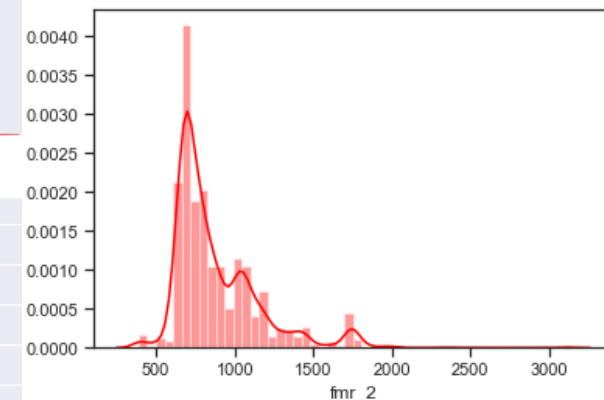
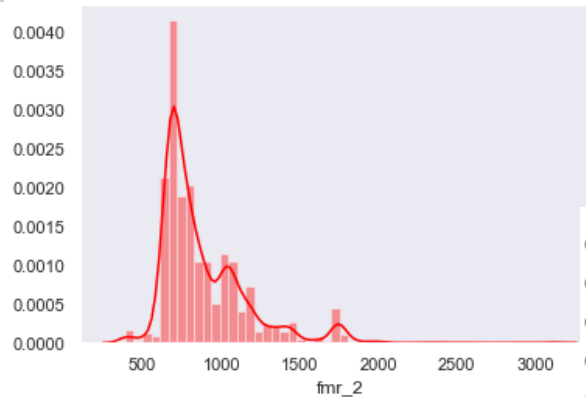
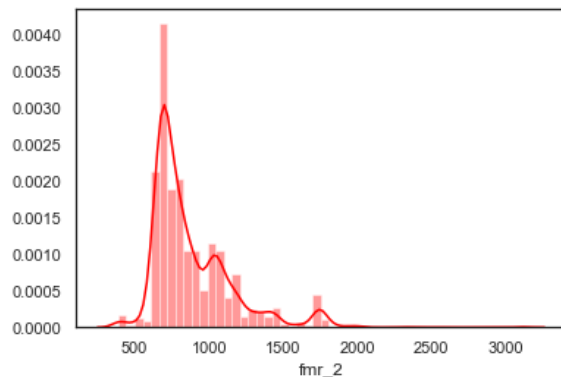


Seaborn style

Seaborn styles

□ Áp dụng theme với sns.set_style()

```
for style in ['white', 'dark', 'whitegrid', 'darkgrid', 'ticks']:
    sns.set_style(style)
    sns.distplot(df['fmr_2'], color='red')
    plt.show()
```

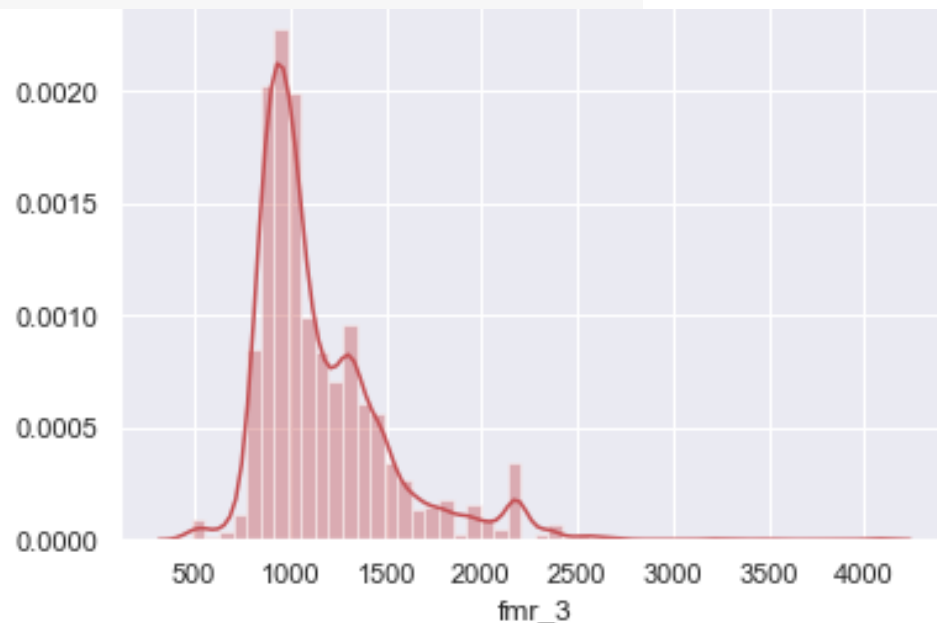


Seaborn styles

□ Color

- Seaborn hỗ trợ thiết lập màu cho biểu đồ sử dụng mã màu (color coder) của matplotlib

```
# Set style, enable color code, and create a red distplot  
sns.set(color_codes=True)  
sns.distplot(df['fmr_3'], color='r')  
# Show the plot  
plt.show()
```



Seaborn styles

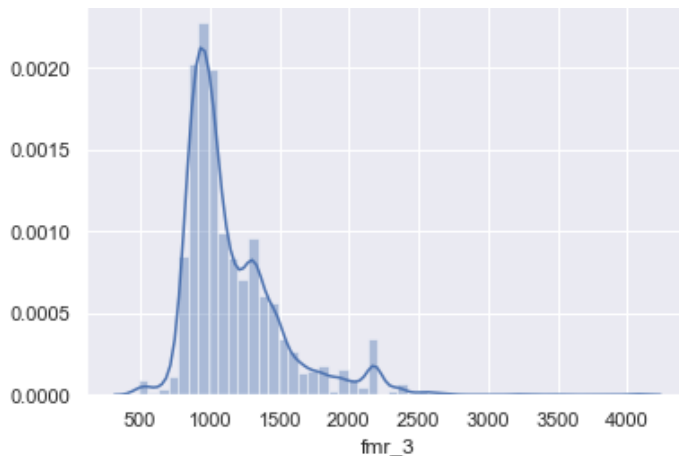
□ Color Palette

- Màu sắc đóng một vai trò rất quan trọng trong trực quan hóa.
- Khi được sử dụng hiệu quả, màu sắc có thể làm cho việc hiển thị có nhiều ý nghĩa hơn.
- Palette: Một bảng màu là một bề mặt phẳng mà trên đó người họa sĩ sắp xếp và pha trộn các màu với nhau.

Seaborn styles

- Seaborn sử dụng `set_palette()` để xác định bảng màu

```
for p in sns.palettes.SEABORN_PALETTES:  
    sns.set_palette(p)  
    sns.distplot(df['fmr_3'])  
    plt.show()  
    # Clear the plots  
    plt.clf()
```



Seaborn styles

- Sử dụng `sns.palettes`: để hiển thị palette
- Sử dụng `color_palette()` để cung cấp màu sắc cho biểu đồ, trả về palette hiện tại
`seaborn.color_palette(palette = None, n_colors = None, desat = None)`
 - `palette`: bảng màu
 - `n_colors`: số màu trong palette. `n_colors = None`: số màu sẽ là số mặc định dựa vào mẫu cụ thể. Mặc định `n_colors = 6`
 - `desat`: tỷ lệ bão hòa mỗi màu



Seaborn styles

Ví dụ:

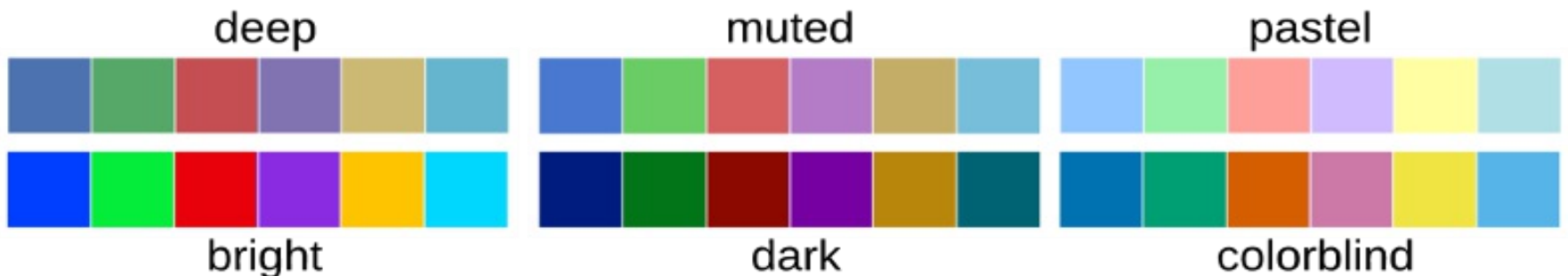
```
current_palette = sb.color_palette()
sb.palplot(current_palette) # được sử dụng để hiển
thị bảng màu theo chiều ngang
plt.show()
```



Seaborn styles

- Categorical palettes— Màu khác nhau nhưng chung style
 - Dùng khi dữ liệu rời rạc, không được sắp xếp

```
for p in sb.palettes.SEABORN_PALETTES:  
    sb.set_palette(p)  
    sb.palplot(sb.color_palette())  
    plt.show()
```



Seaborn styles

□ Tùy chỉnh bảng màu

- Circular Colors – Màu theo vòng
 - Dùng khi dữ liệu không được sắp xếp

```
# Circular colors  
sns.palplot(sns.color_palette("Paired", 12))  
plt.show()
```



Seaborn styles

- Sequential Colors - Màu tuần tự
 - Dùng để thể hiện sự phân bố dữ liệu từ các giá trị thấp đến các giá trị cao hơn trong một phạm vi. Thêm 's' cho màu trong tham số màu dùng để vẽ plot màu tuần tự.

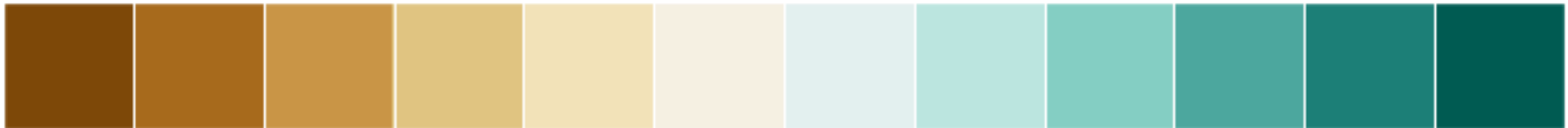
```
# Sequential colors  
sns.palplot(sns.color_palette("Reds", 12))  
plt.show()
```



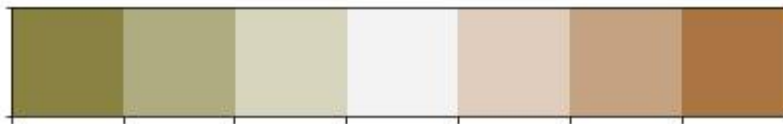
Seaborn styles

- Diverging colors – Màu phân kỳ
 - Dùng khi cả giá trị cao và giá trị thấp đều cần thiết

```
# Diverging colors  
sns.palplot(sns.color_palette("BrBG", 12))  
plt.show()
```

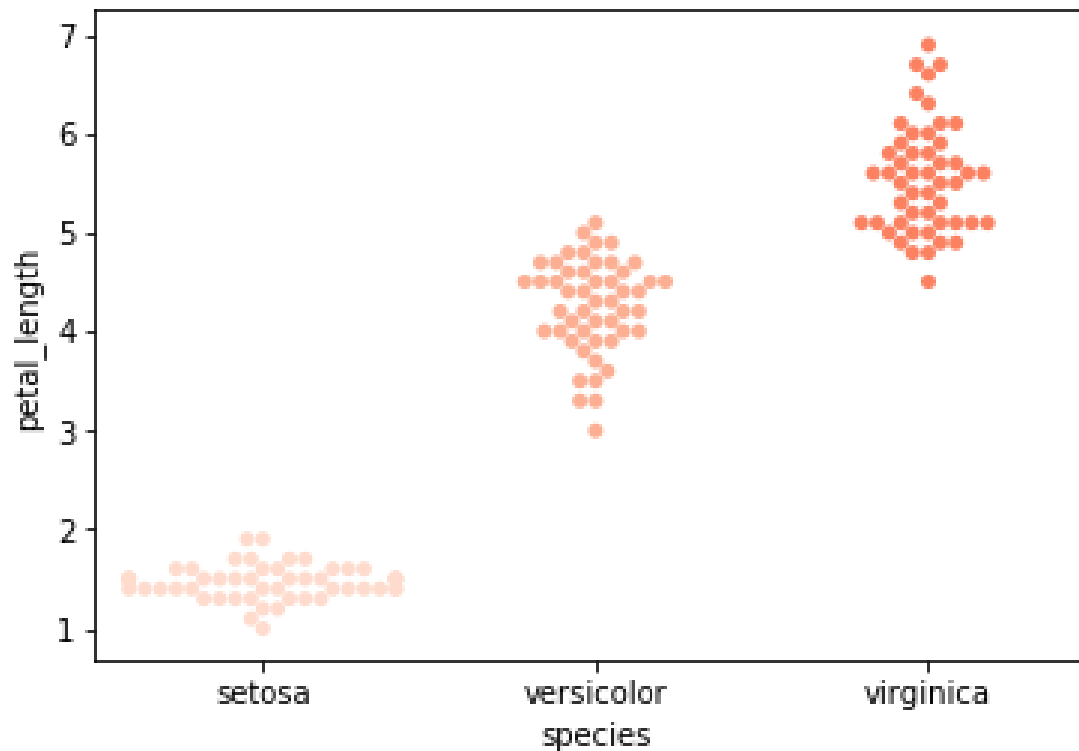


```
custom_palette5 = sns.diverging_palette(440, 40, n=7)  
sns.palplot(custom_palette5)
```



Seaborn styles

```
current_palette = sb.color_palette("Reds")  
sb.swarmplot(x="species", y="petal_length", data=iris, palette = current_palette)  
plt.show()
```

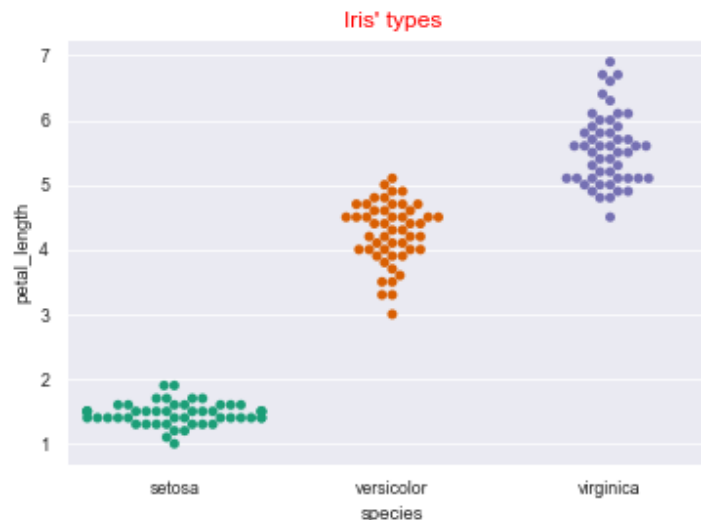


Seaborn styles

❑ Thêm tiêu đề cho biểu đồ

- Sử dụng `set_title("Tên tiêu đề")`

```
# Reset default params
sb.set()
# Set palette to Accent, Accent_r, Blues, Blues_r, BrBG, BrBG_r
sb.set_palette("Dark2")
plot = sb.swarmplot(x="species", y="petal_length", data=iris)
# set title for plot
plot.set_title("Iris' types", fontsize=12, color='red')
plt.show()
```



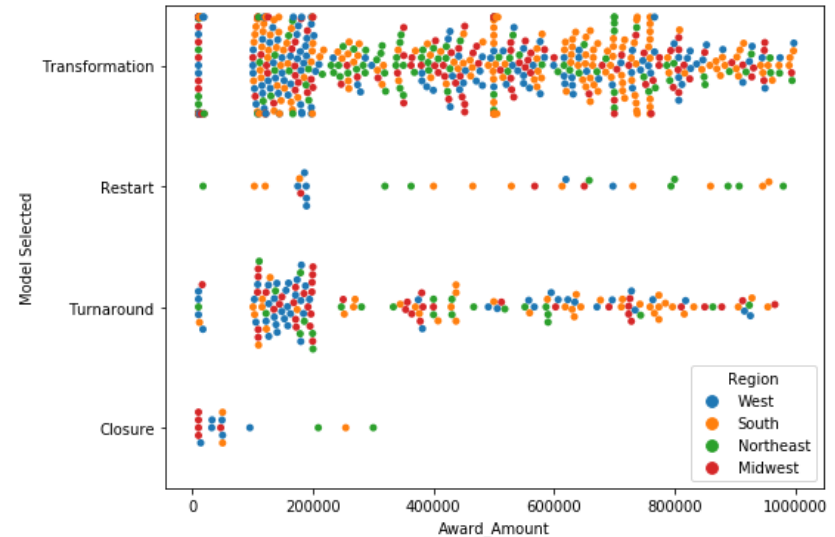
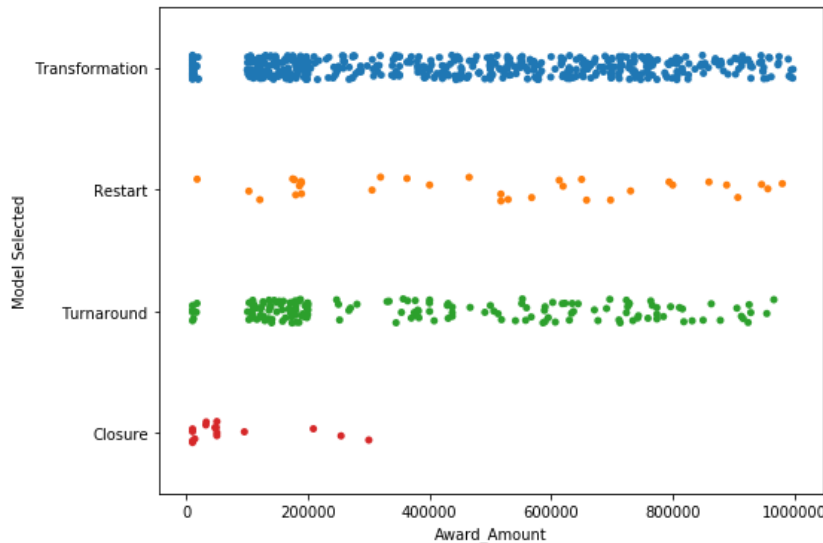
Nội dung

1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Các loại biểu đồ

□ Plot loại hiển thị từng quan sát: stripplot, swarmplot

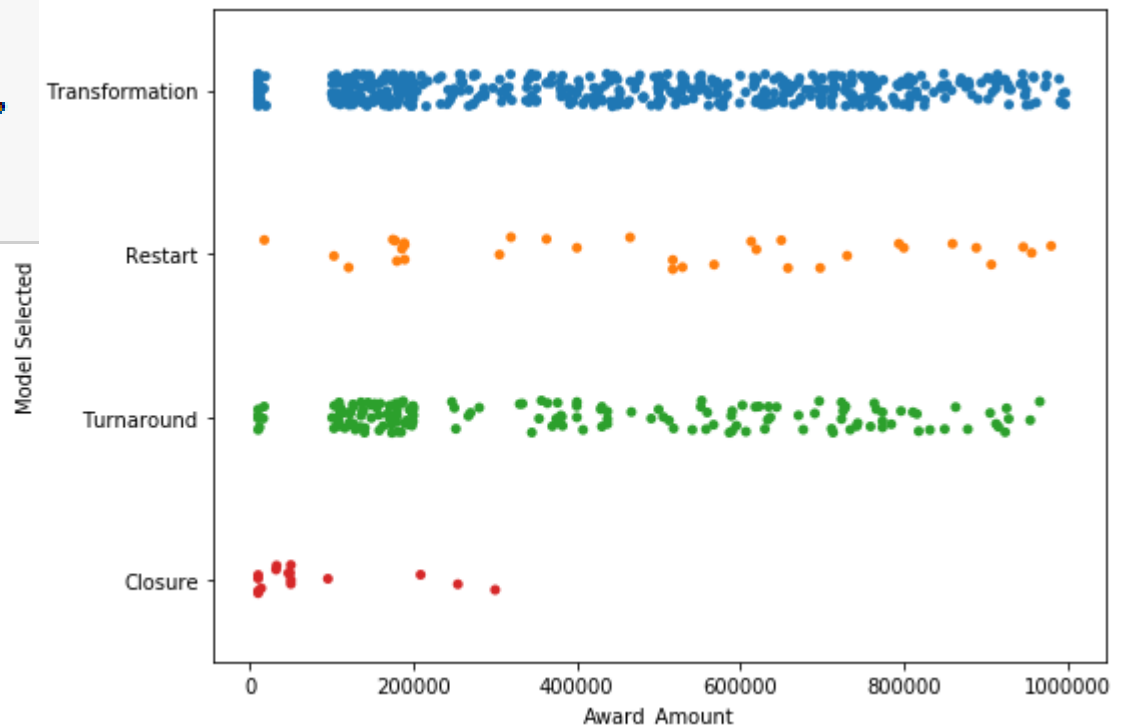
Categorical plot



Các loại biểu đồ

□ Stripplot

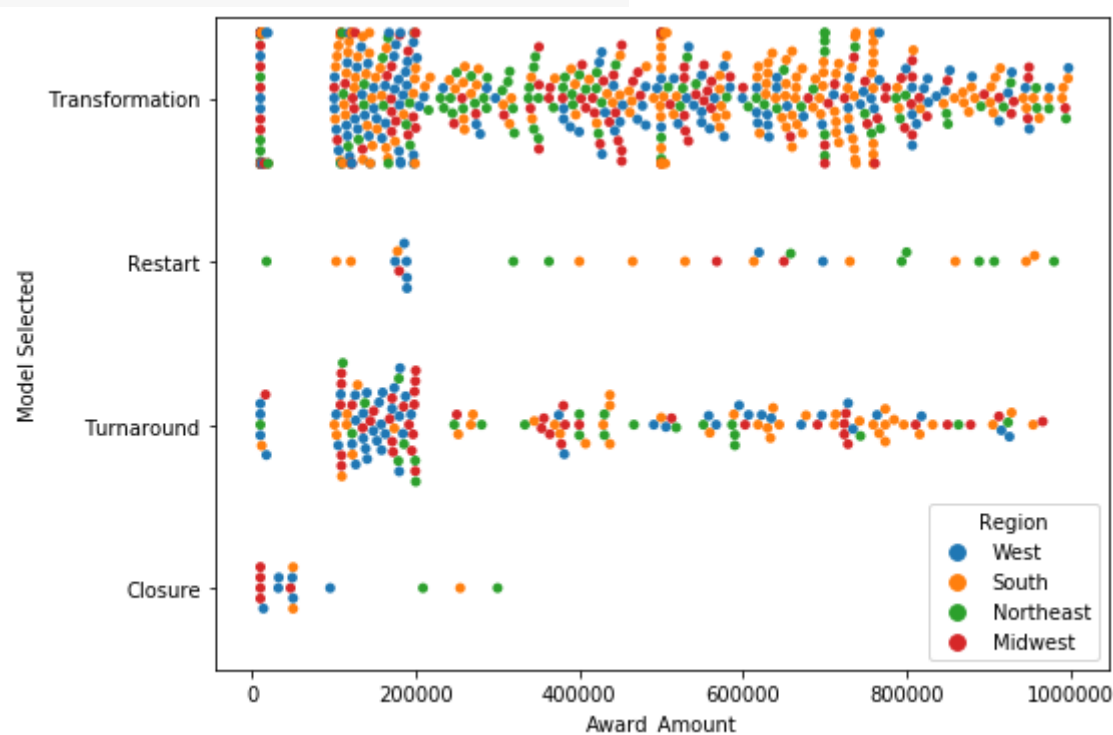
```
# Create the stripplot
plt.figure(figsize=(8,6))
sns.stripplot(data=df,
              x='Award_Amount',
              y='Model Selected',
              jitter=True)
plt.show()
```



Các loại biểu đồ

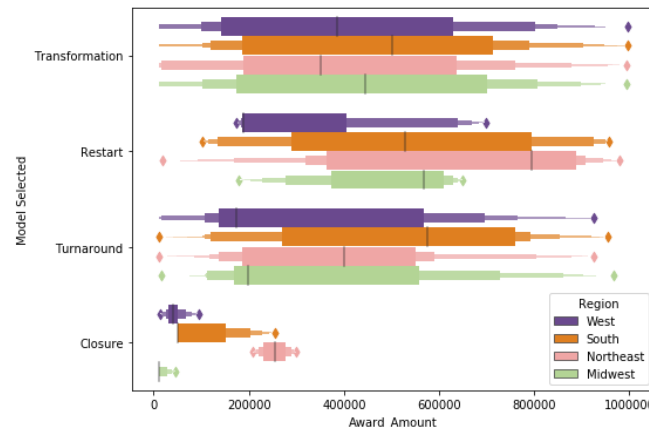
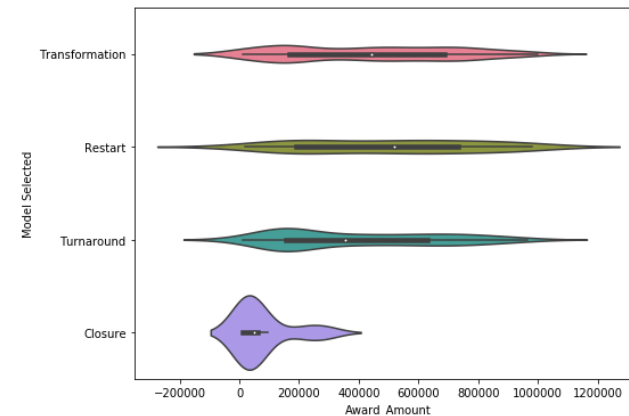
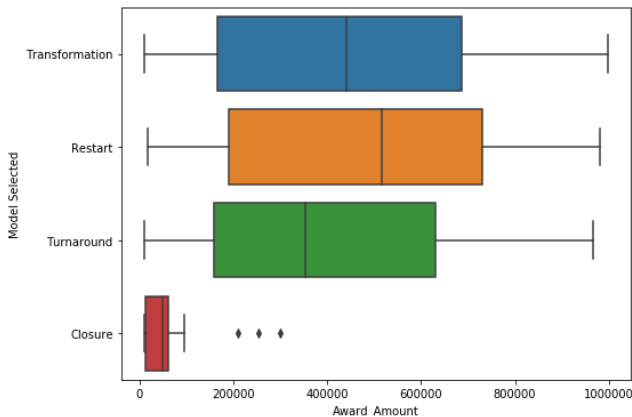
❑ Swarmplot (dữ liệu không chồng chéo lên nhau)

```
# Create and display a swarmplot with hue set to the Region
plt.figure(figsize=(8,6))
sns.swarmplot(data=df,
              x='Award_Amount',
              y='Model Selected',
              hue='Region')
plt.show()
```



Các loại biểu đồ

□ Plot loại hiển thị các đại diện trừu tượng: boxplot, violinplot, boxenplot

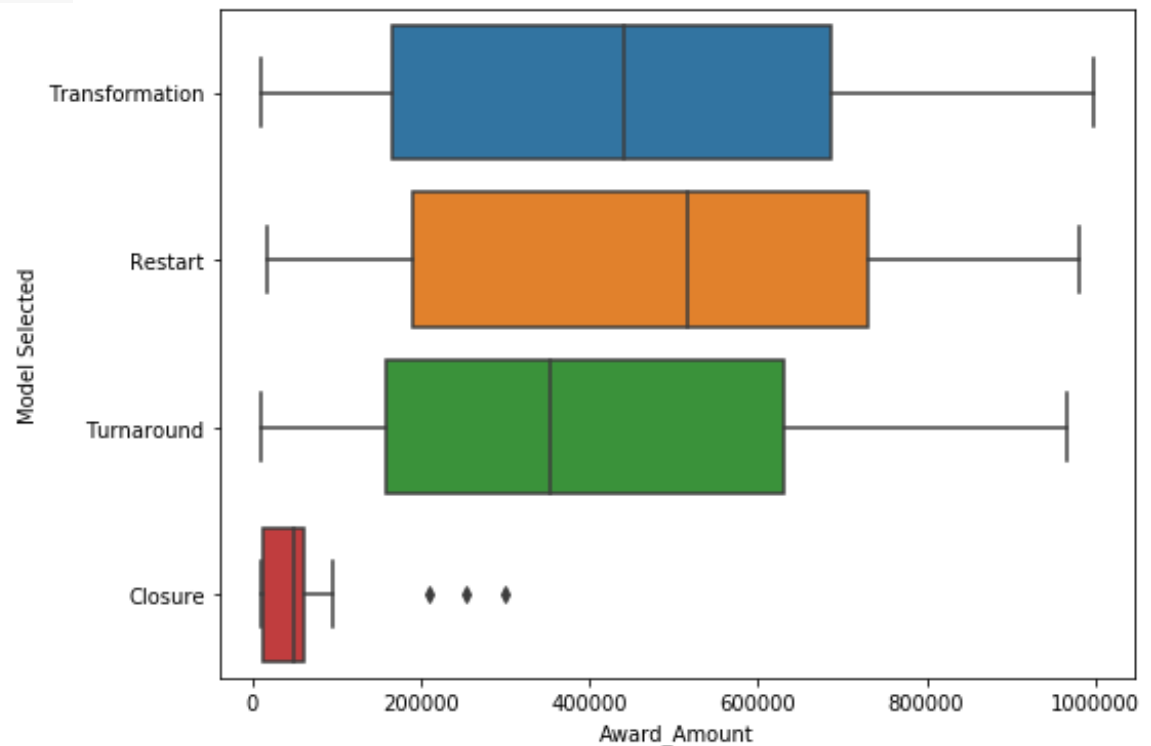


Categorical plot

Các loại biểu đồ

❑ boxplot

```
# Create a boxplot
plt.figure(figsize=(8,6))
sns.boxplot(data=df,
            x='Award_Amount',
            y='Model Selected',
            plt.show())
```

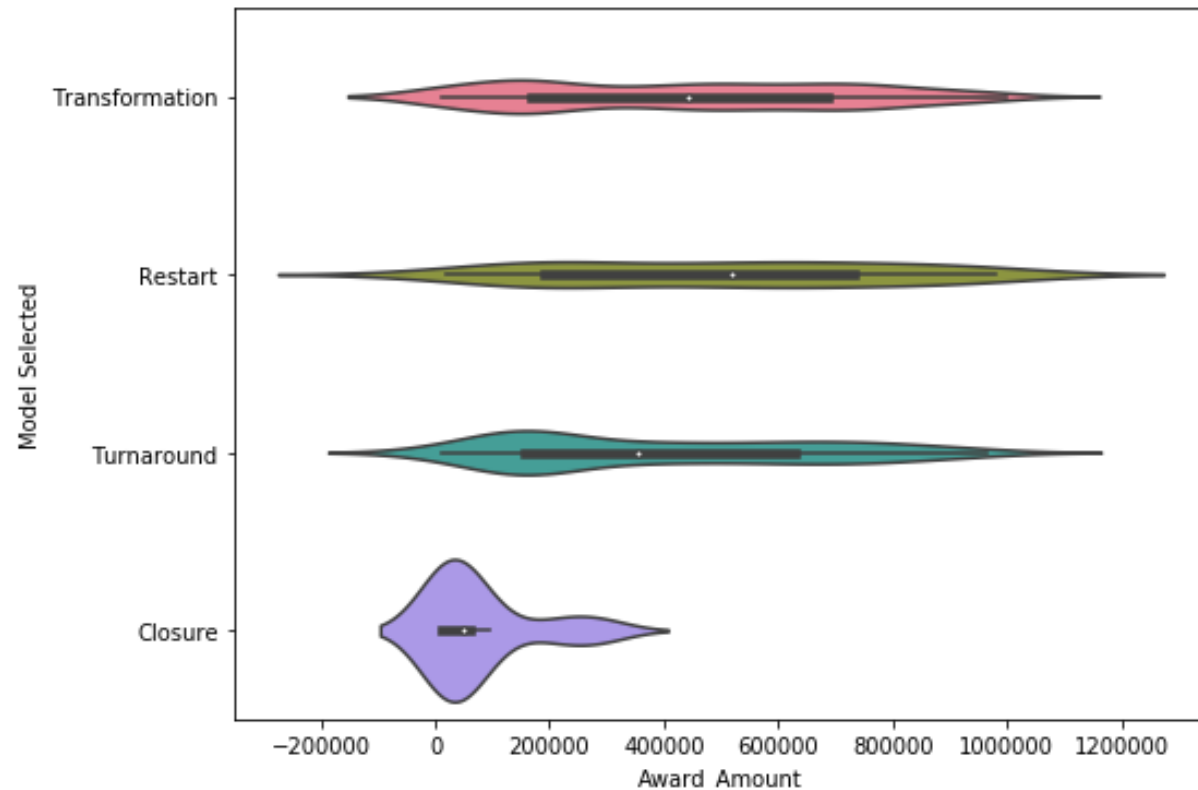


Các loại biểu đồ

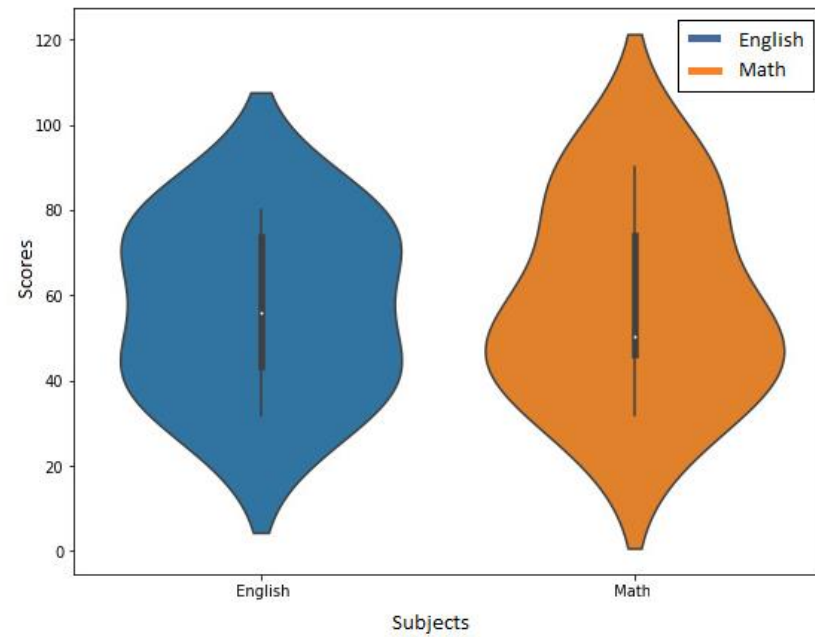
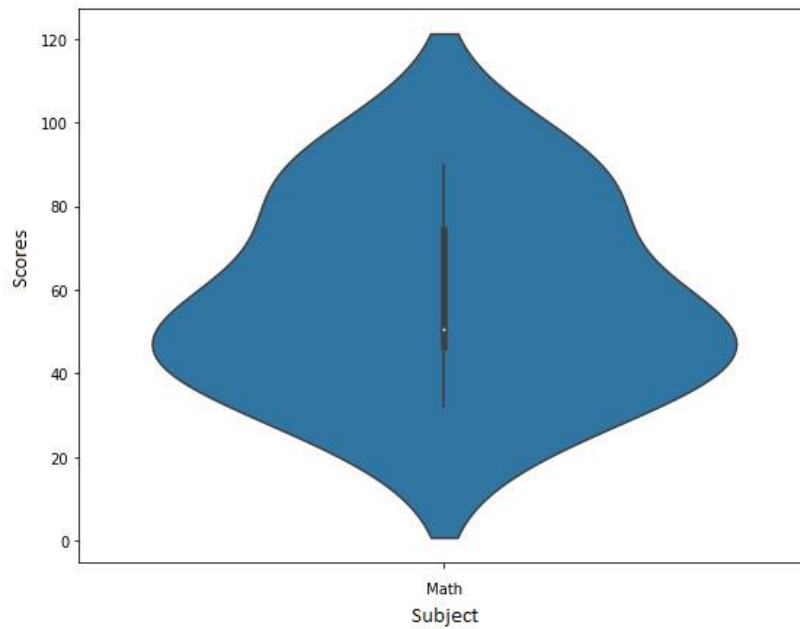
□ violinplot

```
# Create a violinplot with the husl palette
plt.figure(figsize=(8,6))
sns.violinplot(data=df,
               x='Award_Amount',
               y='Model Selected',
               palette='husl')

plt.show()
```



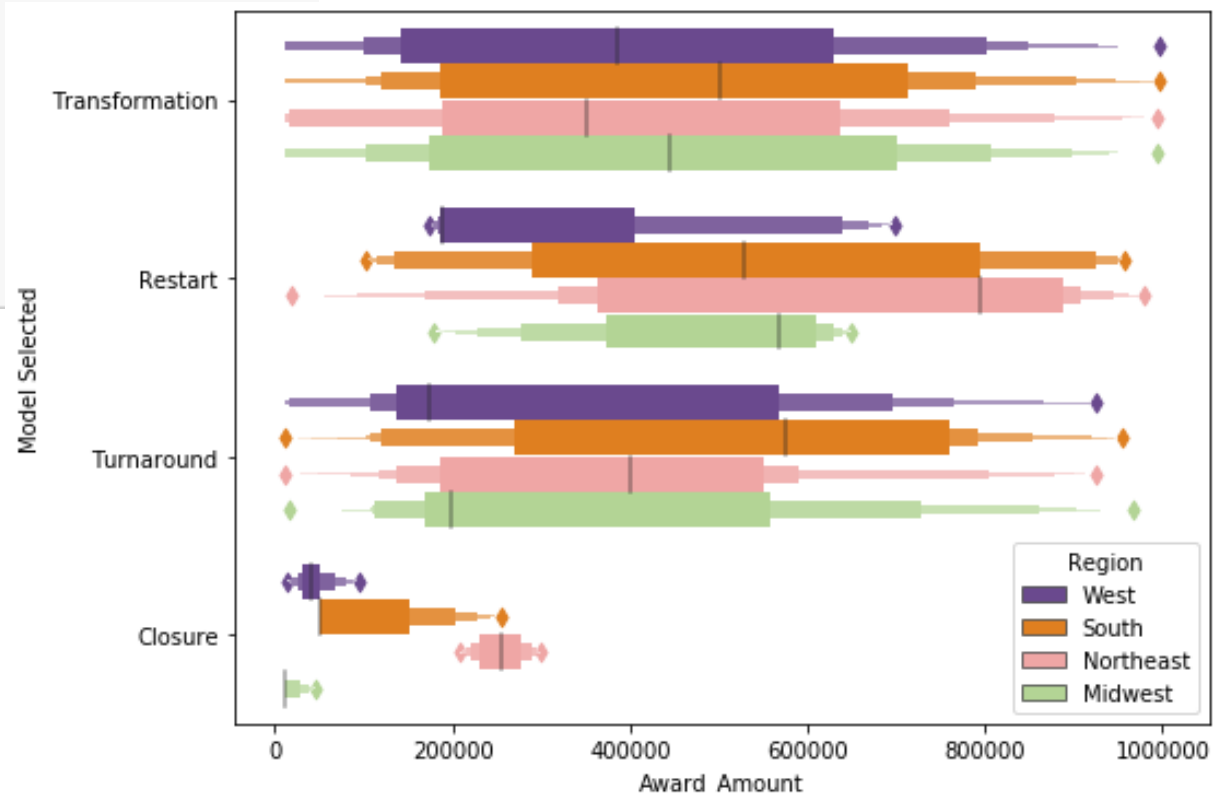
VIOLIN PLOT



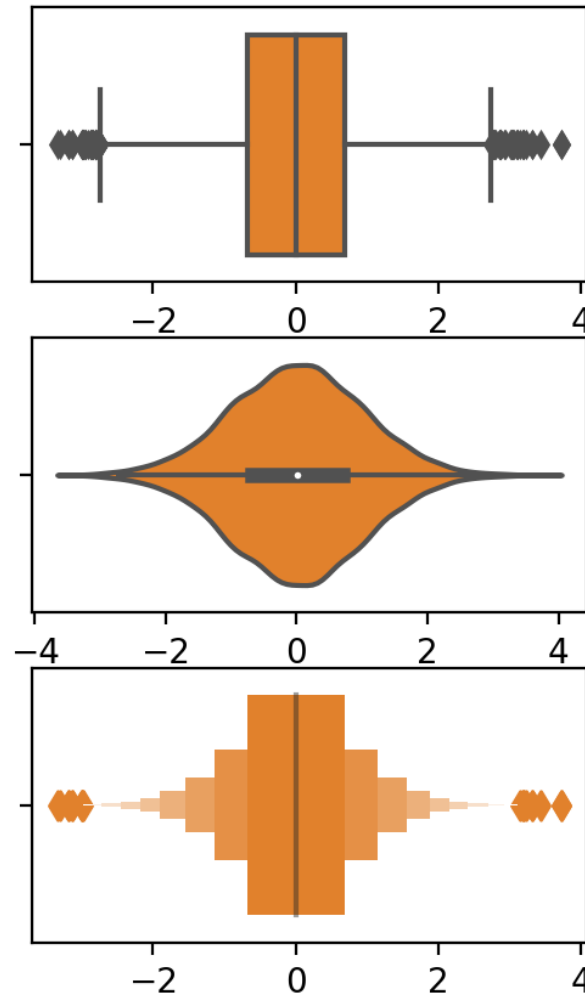
Các loại biểu đồ

boxenplot

```
# Create a boxenplot with the Paired palette  
# and the Region column as the hue  
plt.figure(figsize=(8,6))  
sns.boxenplot(data=df,  
              x='Award_Amount',  
              y='Model Selected',  
              palette='Paired_r',  
              hue='Region')  
  
plt.show()  
plt.clf()
```

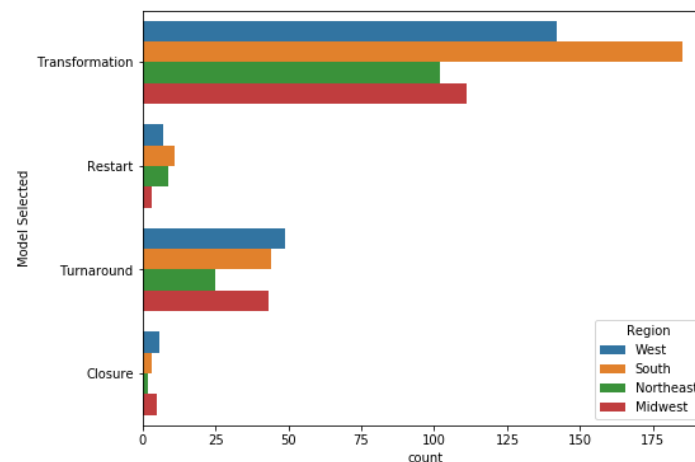
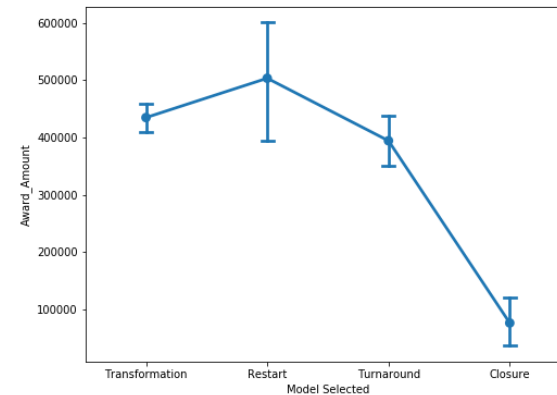
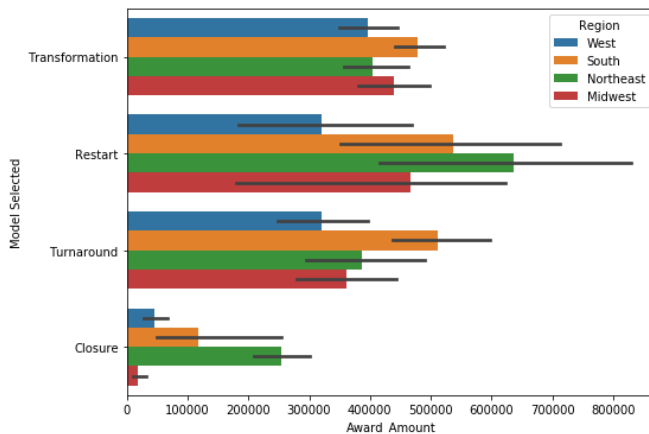


Các loại biểu đồ



Các loại biểu đồ

□ Plot loại hiển thị các ước tính thống kê: barplot, pointplot, countplot

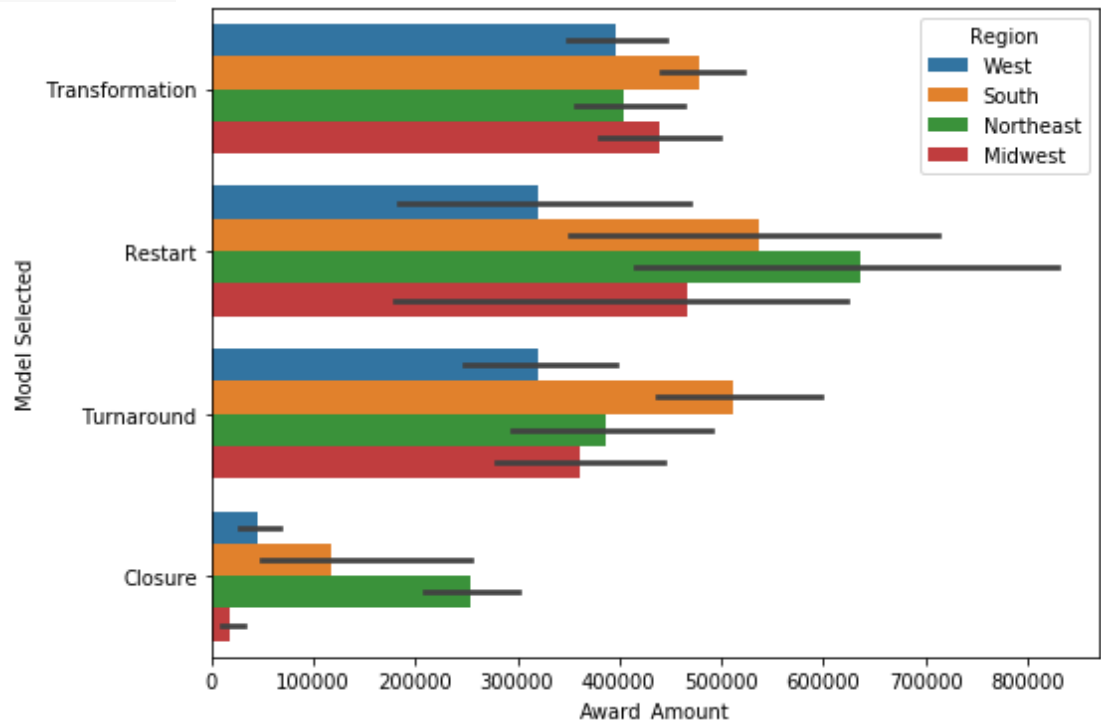


Categorical plot

Các loại biểu đồ

□ barplot

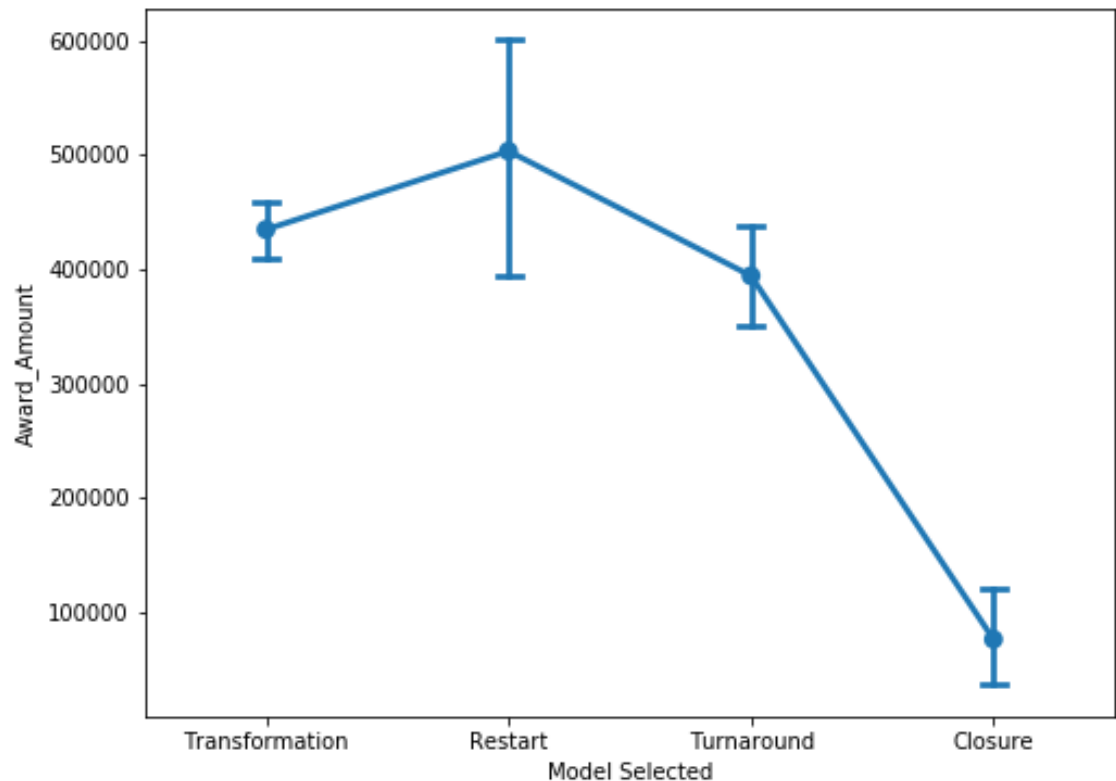
```
# Show a barplot with the number of models used  
# with each region a different color  
plt.figure(figsize=(8,6))  
sns.barplot(data=df,  
            x="Award_Amount",  
            y='Model Selected',  
            hue="Region")  
plt.show()
```



Các loại biểu đồ

□ pointplot

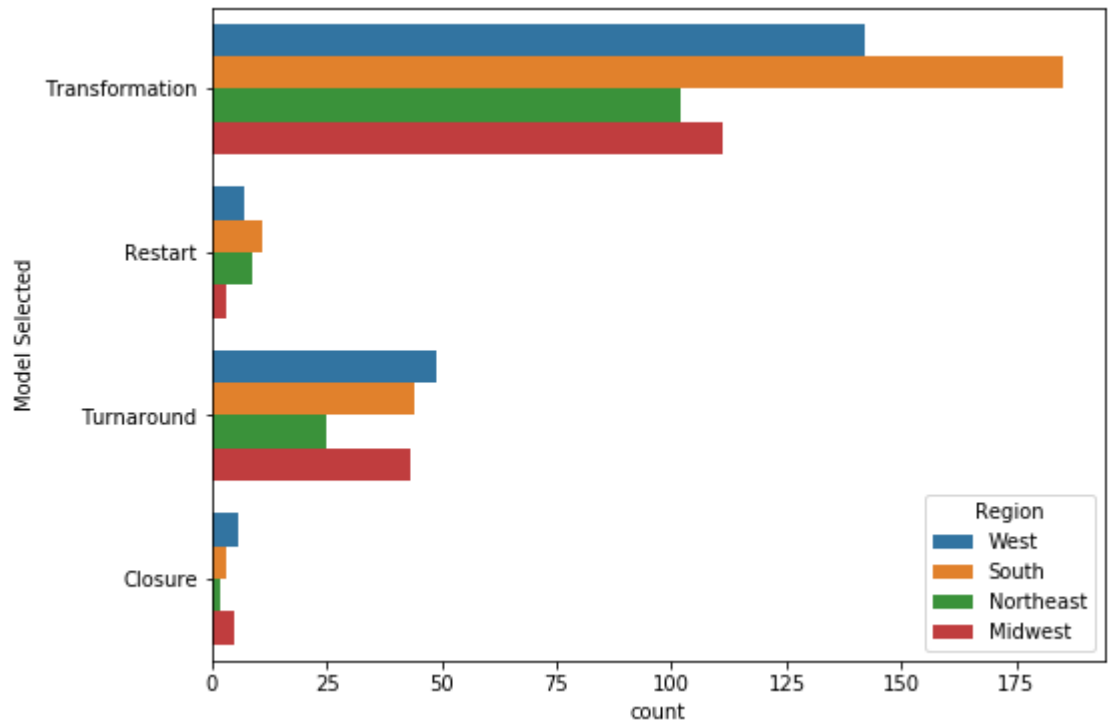
```
# Create a pointplot and include the capsize  
# in order to show bars on the confidence interval  
plt.figure(figsize=(8,6))  
sns.pointplot(data=df,  
              y='Award_Amount',  
              x='Model Selected',  
              capsize=.1)  
plt.show()
```



Các loại biểu đồ

□ countplot

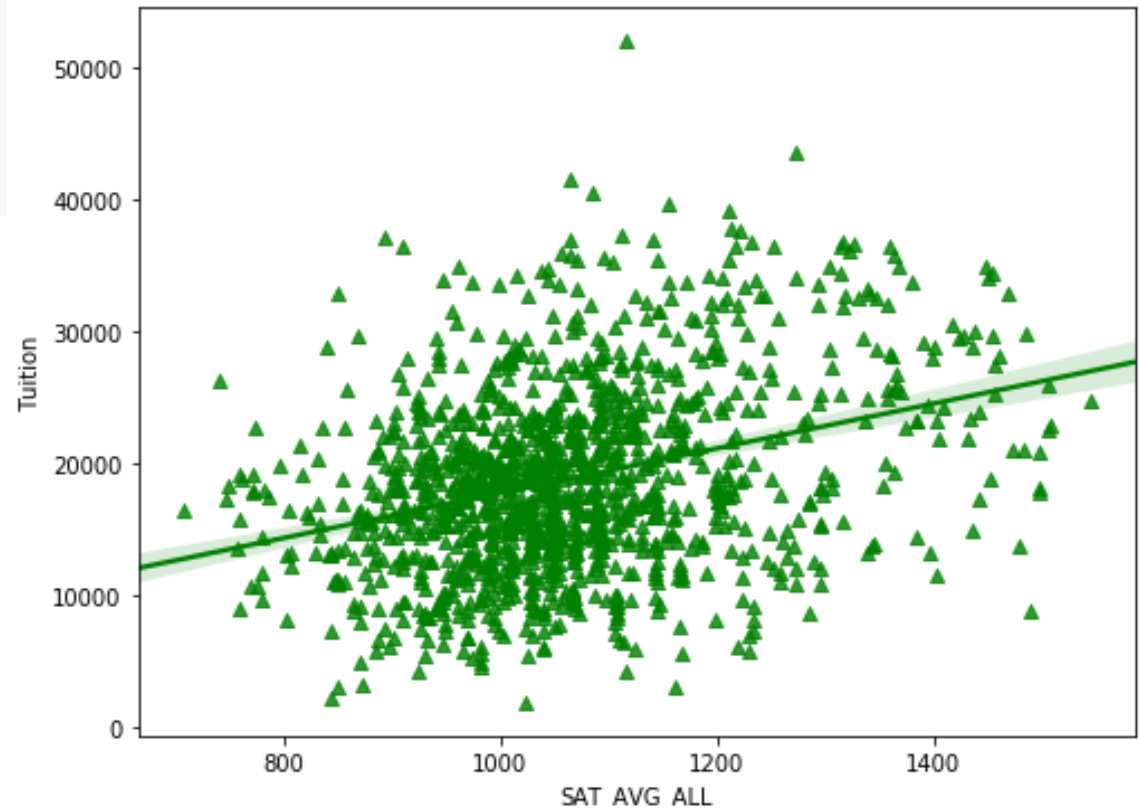
```
# Show a countplot with the number of models used  
# with each region a different color  
plt.figure(figsize=(8,6))  
sns.countplot(data=df,  
              y="Model Selected",  
              hue="Region")  
plt.show()
```



Các loại biểu đồ

□ Regression Plots - regplot

```
# Display a regression plot for Tuition
plt.figure(figsize=(8,6))
sns.regplot(data=df,
            y='Tuition',
            x="SAT_AVG_ALL",
            marker='^',
            color='g')
plt.show()
```

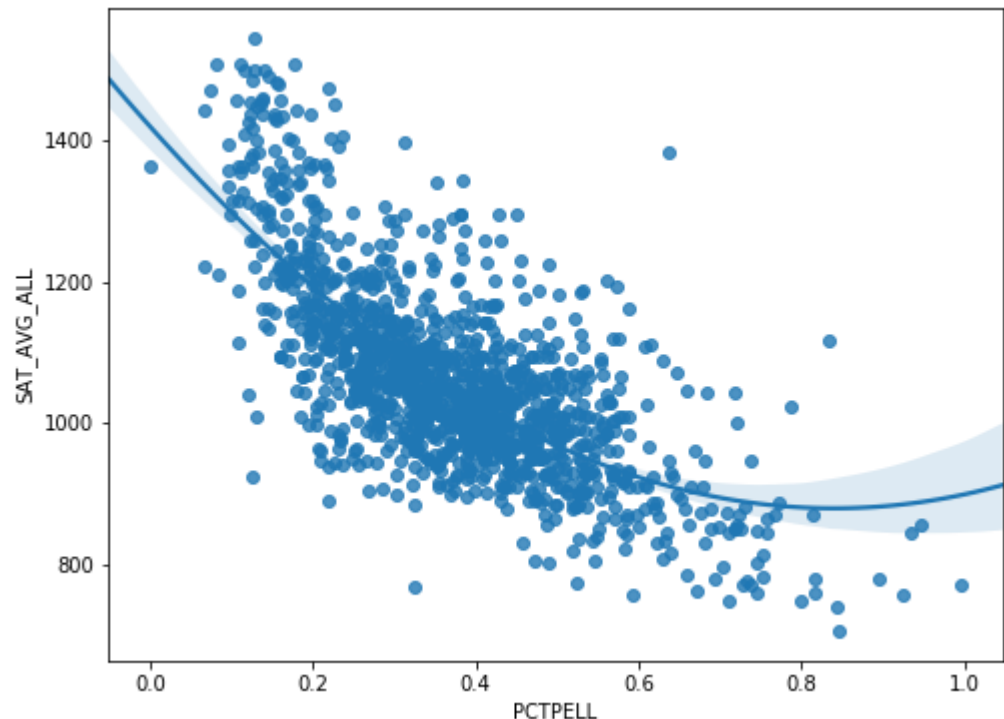


Các loại biểu đồ

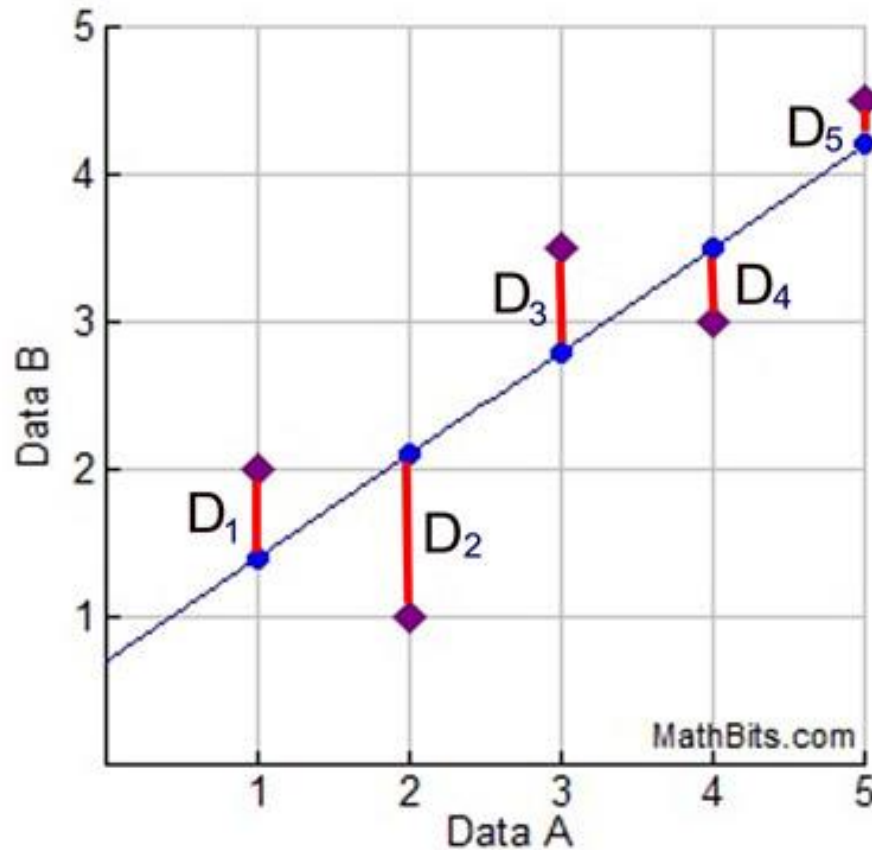
❑ Polynomial regression

- Seaborn hỗ trợ polynomial regression sử dụng tham số “order”

```
# polynomial regression  
plt.figure(figsize=(8,6))  
sns.regplot(data=df,  
            x='PCTPELL',  
            y='SAT_AVG_ALL',  
            order=2)  
plt.show()
```



Residual value



◆ Scatter Plot Points:

$\{(1,2), (2,1), (3,3\frac{1}{2}), (4,3), (5,4)\}$

● Regression Points

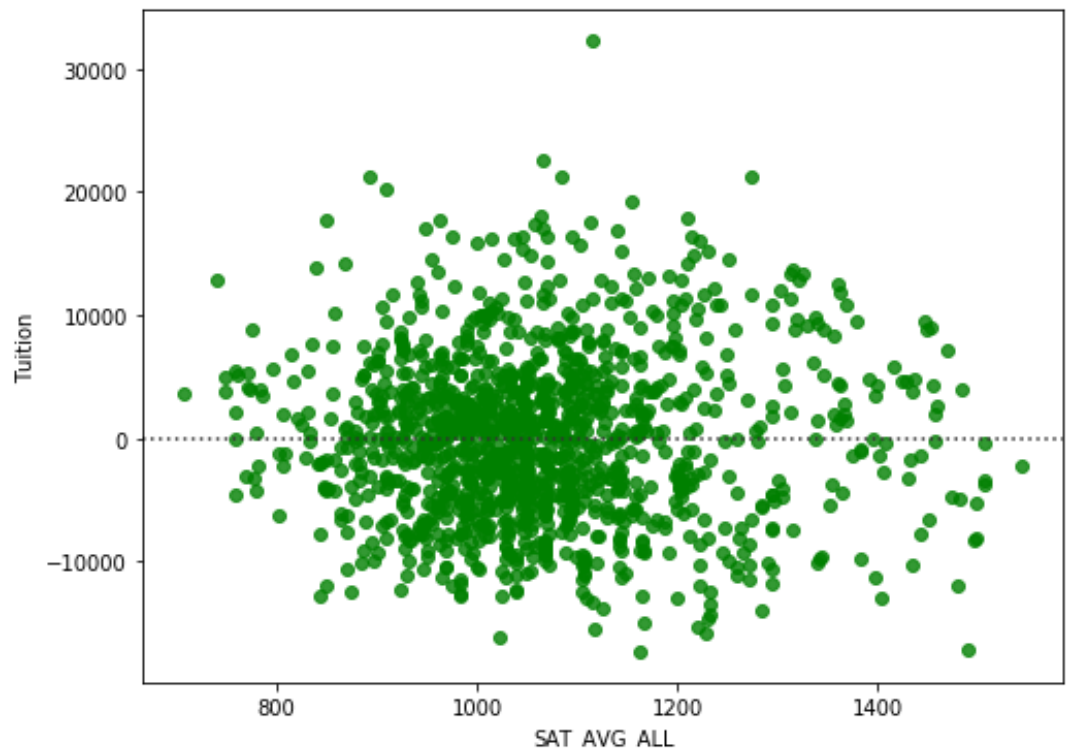
$\{(1,1.4), (2,2.1), (3,2.8), (4,3.5), (5,4.2)\}$

Các loại biểu đồ

❑Đánh giá hồi quy với residplot

- Hữu ích khi đánh giá sự phù hợp (fit) của model

```
# Display the residual plot  
plt.figure(figsize=(8,6))  
sns.residplot(data=df,  
              y='Tuition',  
              x="SAT_AVG_ALL",  
              color='g')  
plt.show()
```

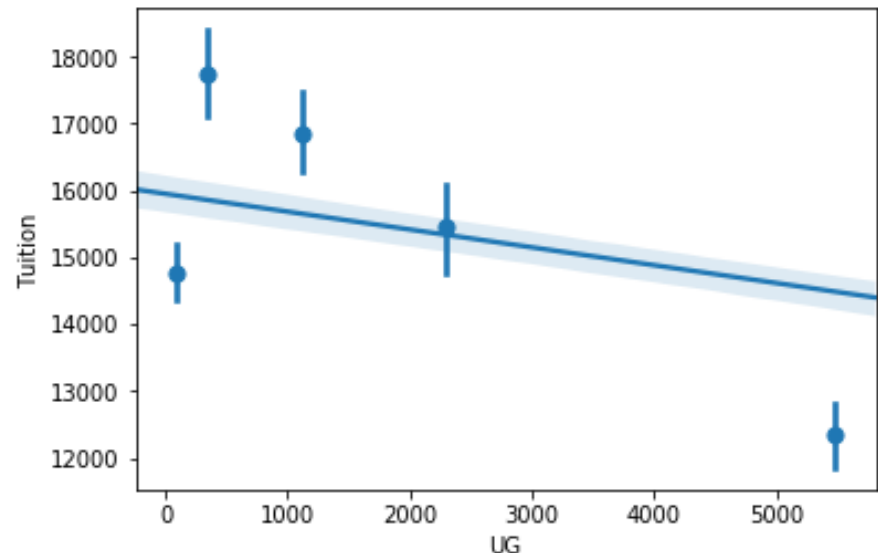


Các loại biểu đồ

□ Bining data

- `x_bins` có thể được sử dụng để chia dữ liệu thành các phần rời rạc (discrete bins)
- regression line vẫn phù hợp với tất cả dữ liệu

```
# Create a scatter plot  
# and bin the data into 5 bins  
sns.regplot(data=df,  
            y='Tuition',  
            x="UG",  
            x_bins=5)  
  
plt.show()
```

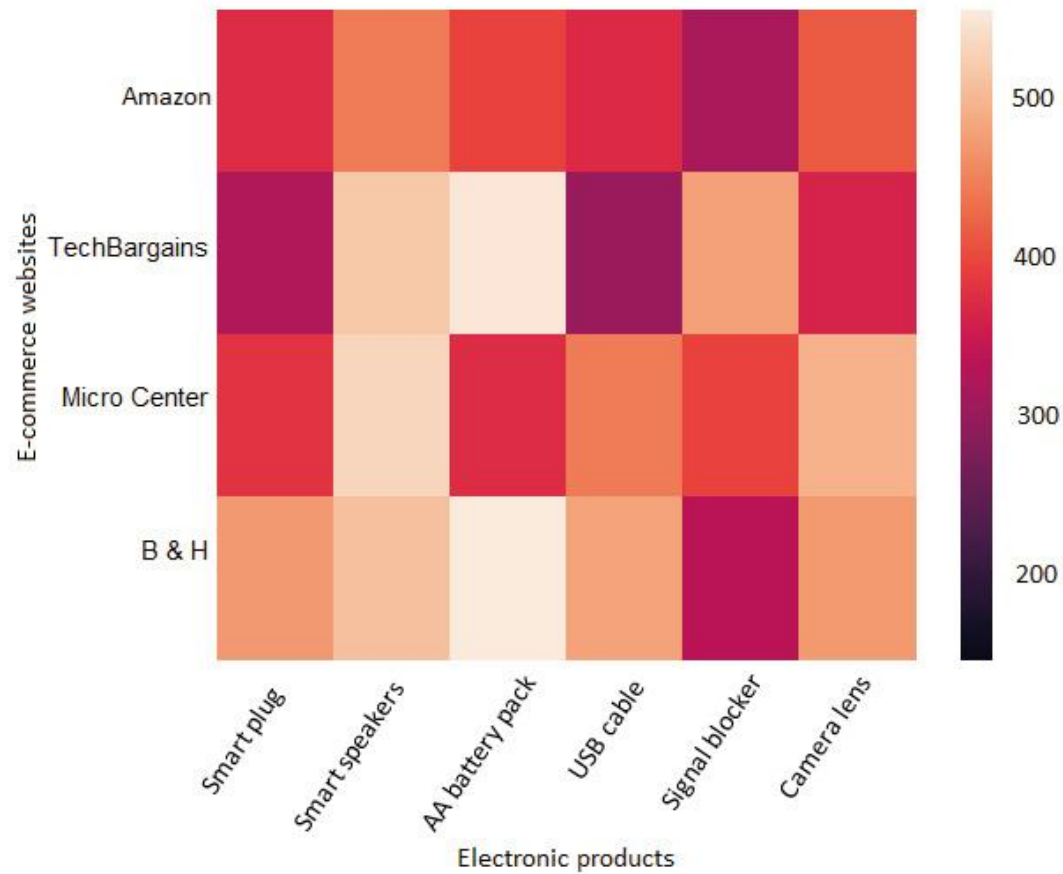


Các loại biểu đồ

❑ Matrix Plots

- Heatmap là một biểu đồ ma trận phổ biến được sử dụng để tóm tắt mối quan hệ giữa hai biến.
- `heatmap()` yêu cầu dữ liệu phải ở định dạng lưới.
- `Pandas.crosstab()` được sử dụng để tính toán dữ liệu

Heatmap



Các loại biểu đồ

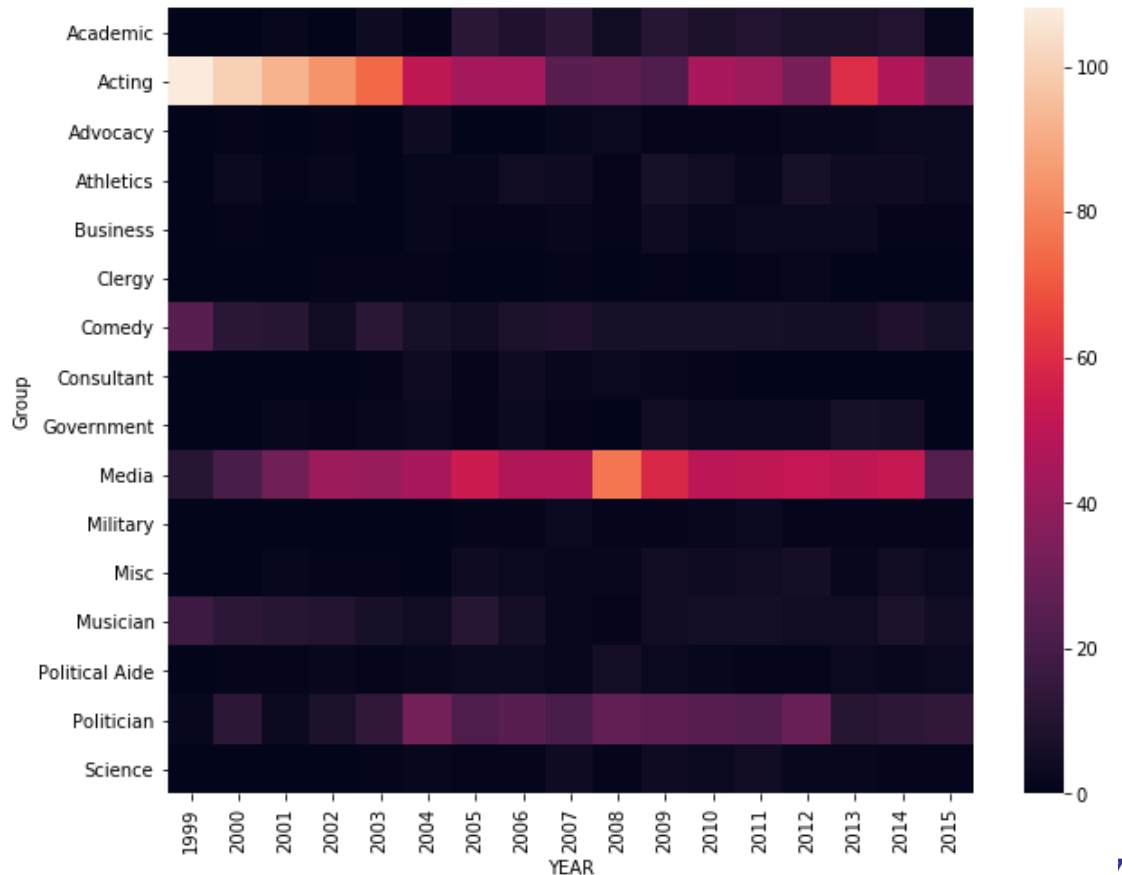
□ heatmap

```
# Create a crosstab table of the data
plt.figure(figsize=(10,10))
pd_crosstab = pd.crosstab(df["Group"], df["YEAR"])
print(pd_crosstab)

# Plot a heatmap of the table
sns.heatmap(pd_crosstab)

# Rotate tick marks for visibility
plt.yticks(rotation=0)
plt.xticks(rotation=90)

plt.show()
```



Các loại biểu đồ

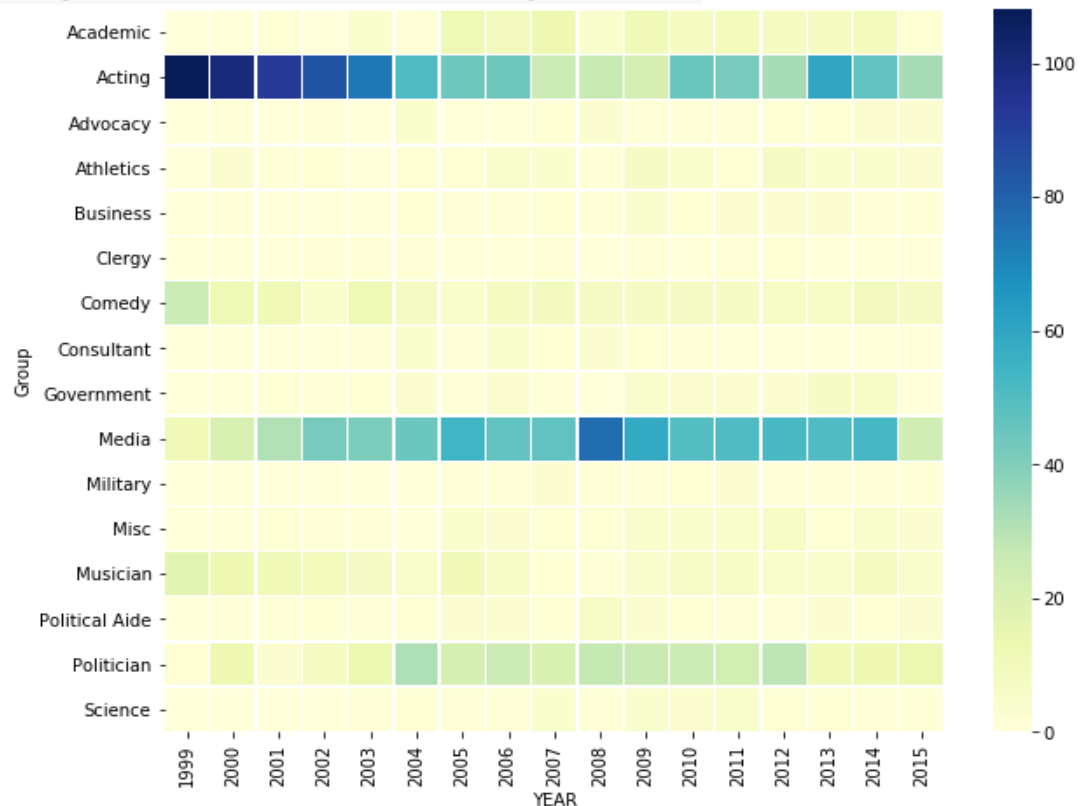
• Tùy chỉnh heatmap

```
# Create the crosstab DataFrame
pd_crosstab = pd.crosstab(df["Group"], df["YEAR"])

# Plot a heatmap of the table with no color bar and using the BuGn palette
sns.heatmap(pd_crosstab, cbar=True, cmap="YlGnBu", linewidths=0.3)

# Rotate tick marks for visibility
plt.yticks(rotation=0)
plt.xticks(rotation=90)

# Show the plot
plt.show()
```



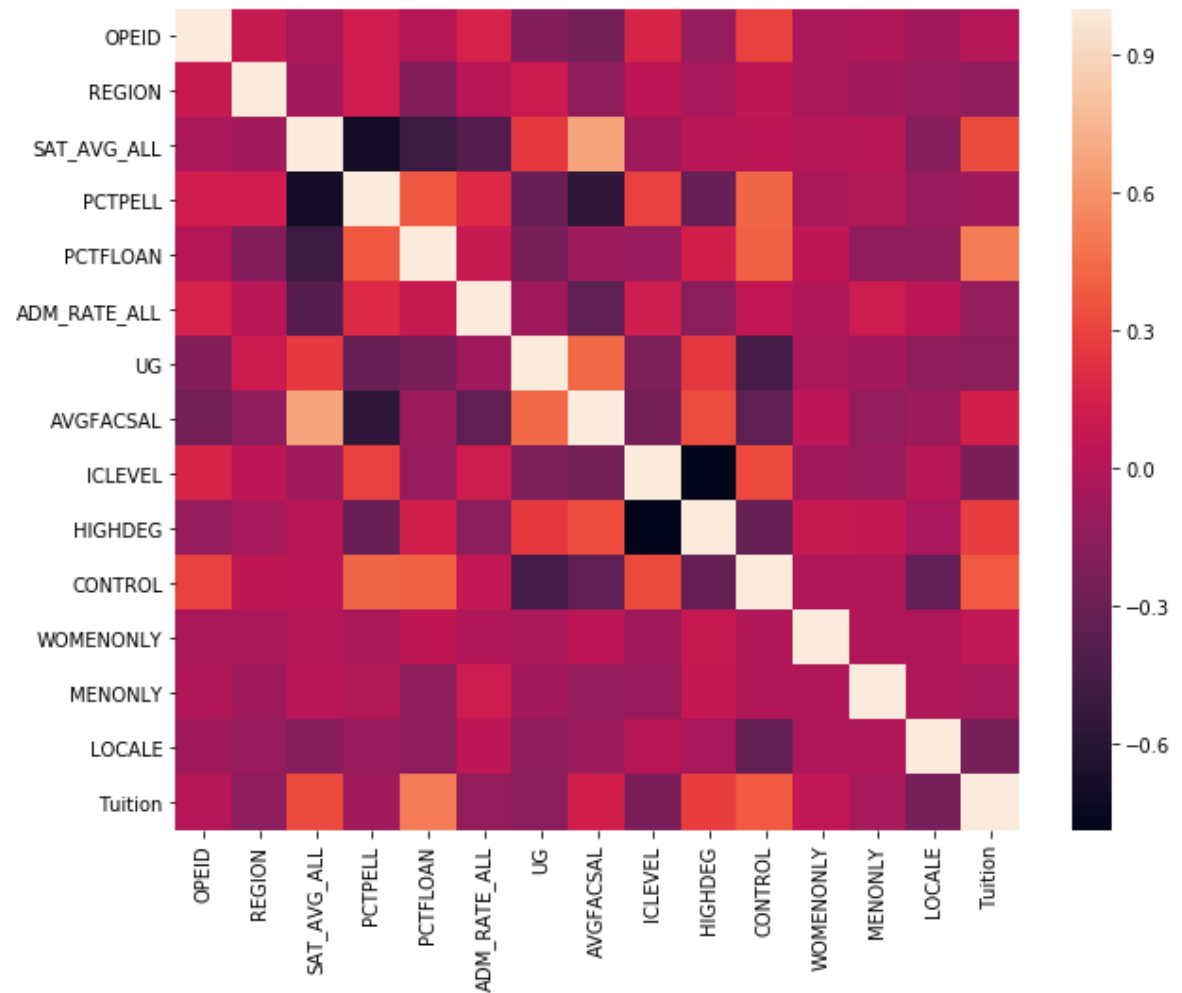
Các loại biểu đồ

❑ correlation matrix

- `pandas.corr()` tính toán tương quan giữa các cột trong một dataframe
- Output có thể được chuyển thành heatmap với seaborn

Các loại biểu đồ

```
plt.figure(figsize=(10,8))  
sns.heatmap(df.corr())  
plt.show()
```



Nội dung

1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Vẽ biểu đồ trên Data Aware Grid

- ❑ Grid plot của Seaborn yêu cầu dữ liệu dưới dạng "tidy format", nghĩa là một quan sát trên mỗi hàng dữ liệu.

Vẽ biểu đồ trên Data Aware Grid

AMONG ADULT MEN		Unnamed: 1	Adult Men	Age	Unnamed: 4	Unnamed: 5
0				18 - 34	35 - 64	65 and up
1	In general, how masculine or "manly" do you feel?					
2		Very masculine	37%	29%	42%	37%
3		Somewhat masculine	46%	47%	46%	47%
4		Not very masculine	11%	13%	9%	13%
5		Not at all masculine	5%	10%	2%	3%
6		No answer	1%	0%	1%	1%
7	How important is it to you that others see you as masculine?					
8		Very important	16%	18%	17%	13%
9		Somewhat important	37%	38%	37%	32%
10		Not too important	28%	18%	31%	37%
11		Not at all important	18%	26%	15%	18%
12		No answer	0%	0%	1%	0%

Untidy data

Vẽ biểu đồ trên Data Aware Grid

□ FacetGrid

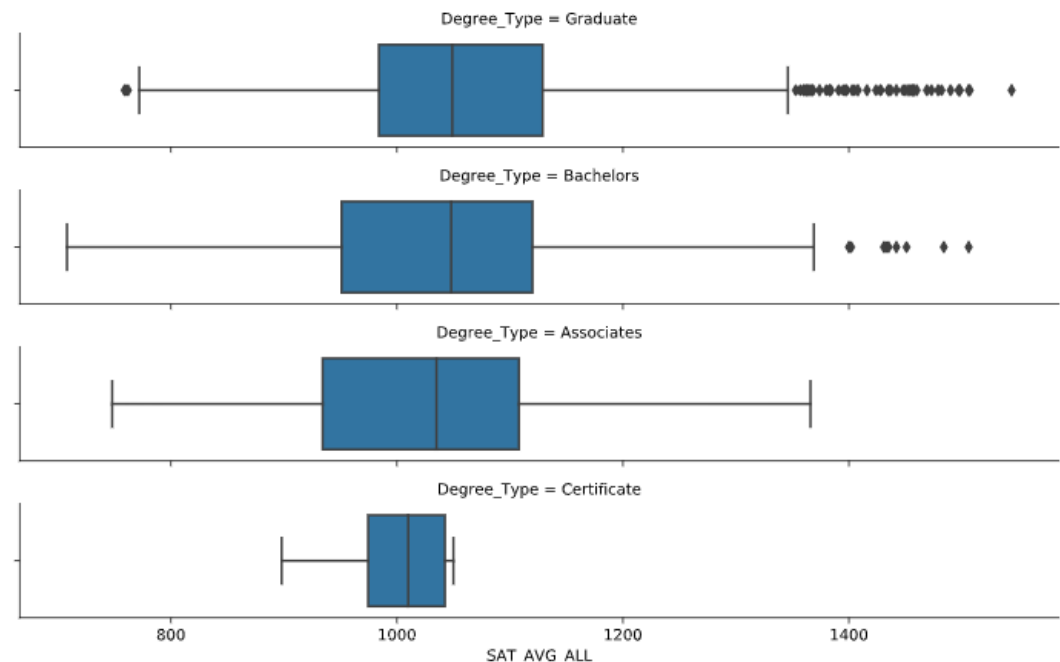
- Là nền tảng cho lưới nhận biết dữ liệu.
- Lưới nhận biết dữ liệu cho phép ta tạo một loạt các ô nhỏ hữu ích để hiểu các mối quan hệ dữ liệu phức tạp.
- Nó cho phép người dùng kiểm soát cách phân phối dữ liệu trên các cột, dòng và màu sắc
- Khi FacetGrid được tạo, loại plot cần phải được ánh xạ vào lưới

Vẽ biểu đồ trên Data Aware Grid

```
# Create FacetGrid with Degree_Type and specify the
order of the rows using row_order
g2 = sns.FacetGrid(df,
                    row="Degree_Type",
                    row_order=['Graduate', 'Bachelors',
                              'Associates', 'Certificate'])

# Map a boxplot of SAT_AVG_ALL onto the grid
g2.map(sns.boxplot, 'SAT_AVG_ALL')

# Show the plot
plt.show()
```

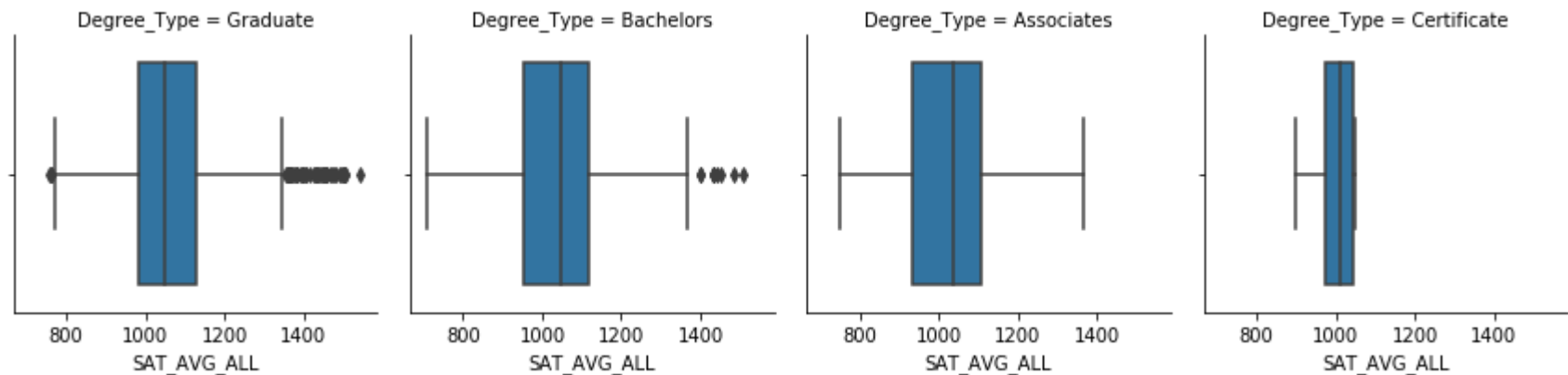


Vẽ biểu đồ trên Data Aware Grid

```
g2 = sns.FacetGrid(df,
                    col="Degree_Type",
                    col_order=['Graduate', 'Bachelors', 'Associates', 'Certificate'])

# Map a boxplot of SAT_AVG_ALL onto the grid
g2.map(sns.boxplot, 'SAT_AVG_ALL')

# Show the plot
plt.show()
plt.clf()
```



Vẽ biểu đồ trên Data Aware Grid

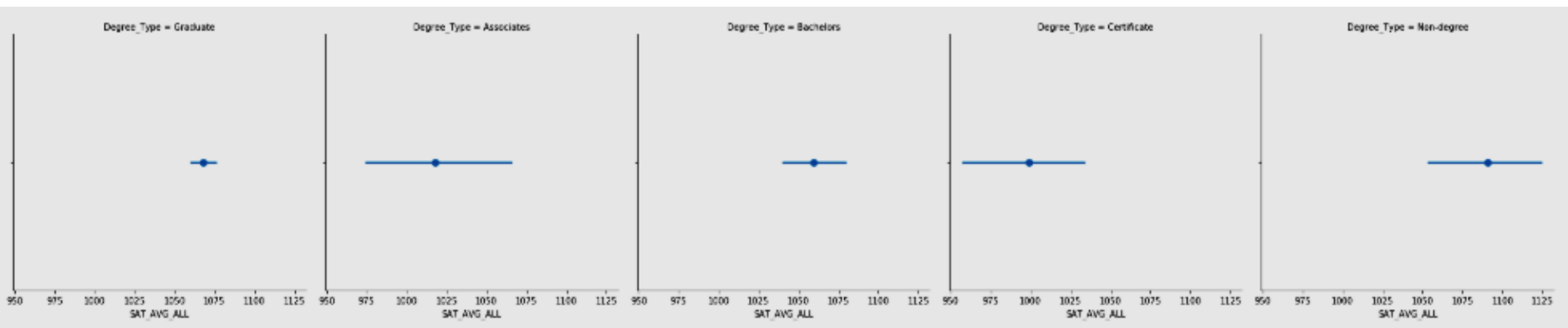
❑ catplot

- Được sử dụng cho dữ liệu phân loại (categorical data)
- Dễ dàng tạo các subplot với `col =` và `row =`

Vẽ biểu đồ trên Data Aware Grid

```
# Create a faceted pointplot of Average SAT_AVG_ALL scores faceted by Degree Type
sns.factorplot(data=df,
               x='SAT_AVG_ALL',
               kind='point',
               col='Degree_Type',
               col_order=['Graduate', 'Bachelors', 'Associates', 'Certificate'])

plt.show()
```

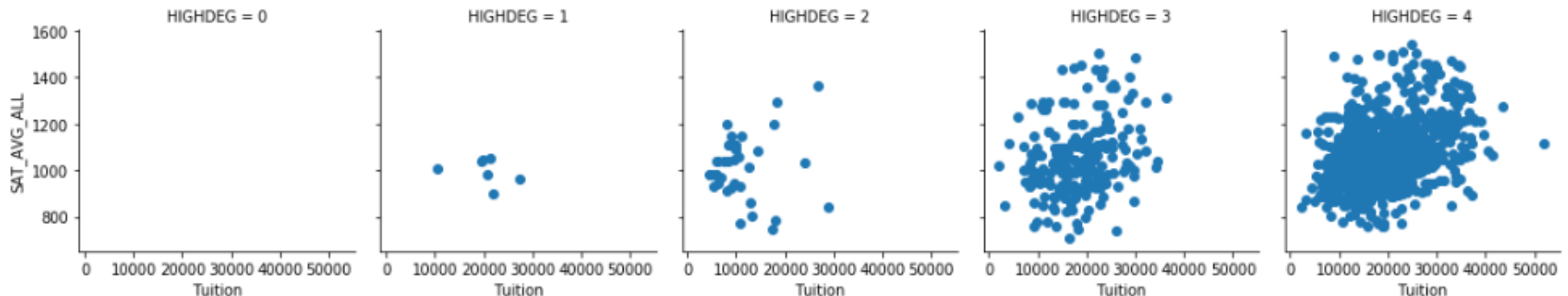


Vẽ biểu đồ trên Data Aware Grid

❑ FacetGrid cho regression

- FacetGrid() cũng có thể dùng cho scatter/regression plot

```
g = sns.FacetGrid(df, col="HIGHDEG")  
g.map(plt.scatter, 'Tuition', 'SAT_AVG_ALL')  
plt.show()
```



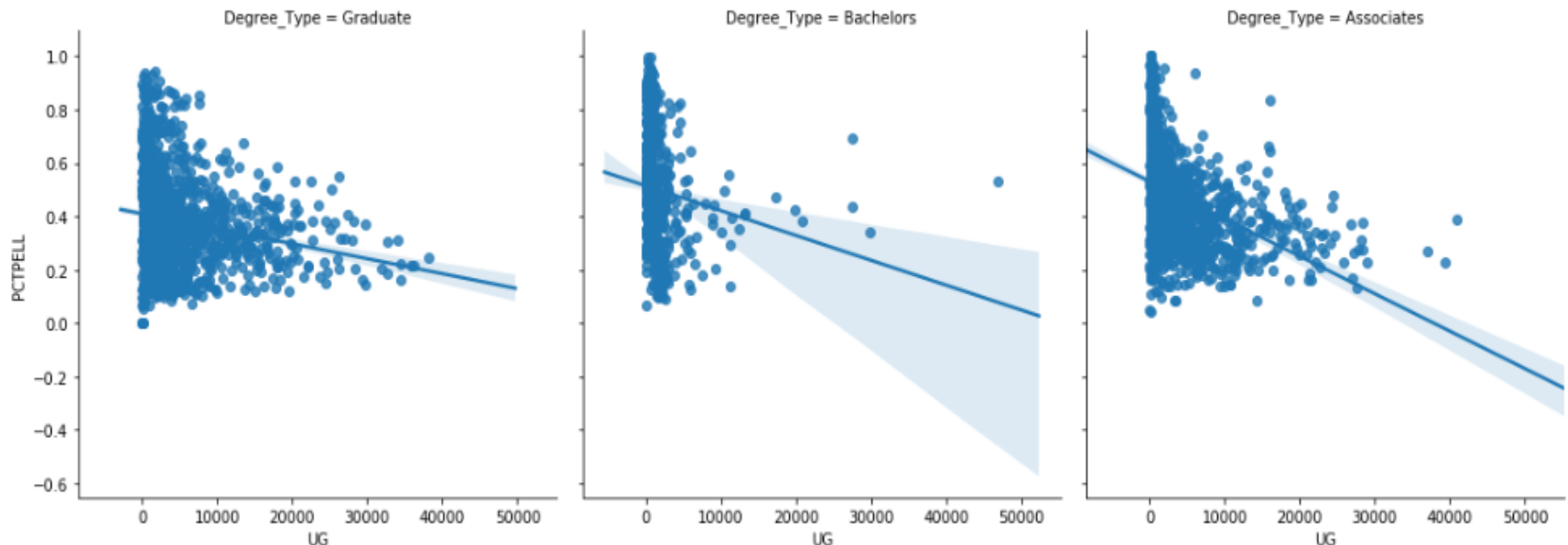
Vẽ biểu đồ trên Data Aware Grid

❑ Implot

- Được sử dụng để vẽ scatter plot với regression line trên các FacetGrid object.
- API tương tự như Facplot với sự khác biệt là mặc định Implot vẽ các regression lines.

Vẽ biểu đồ trên Data Aware Grid

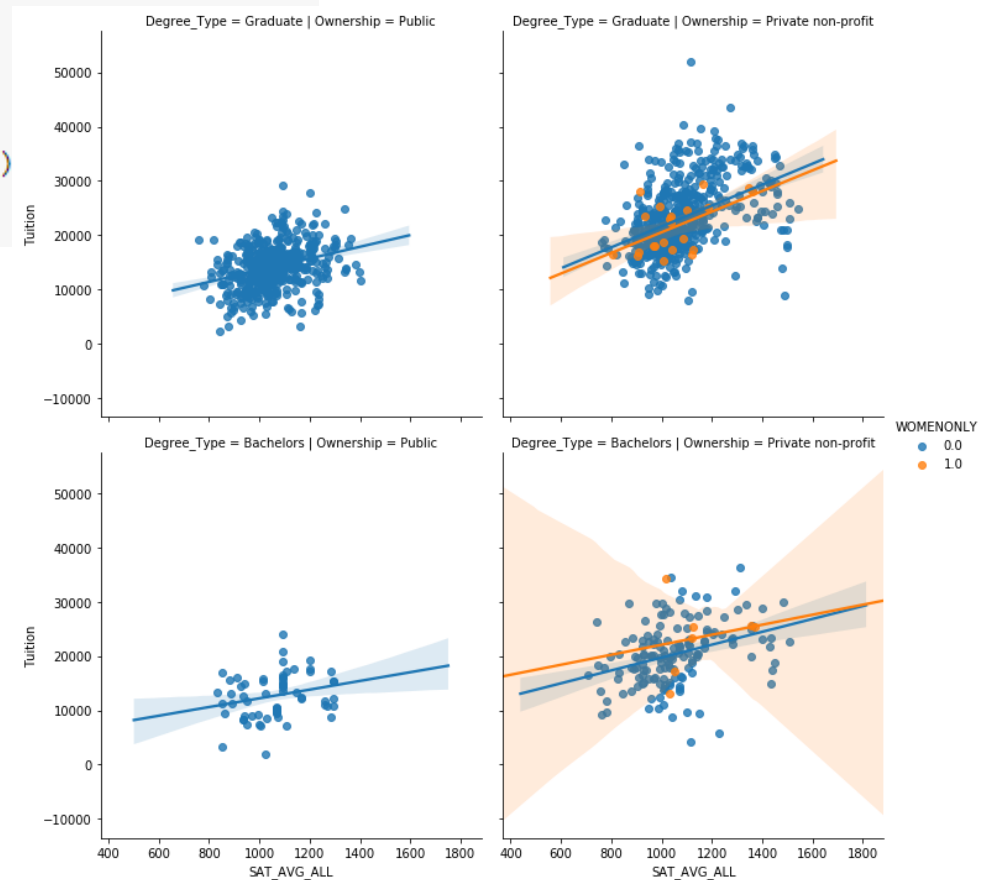
```
# Create the plot as an lmplo  
sns.lmplot(data=df,  
           x='UG',  
           y='PCTPELL',  
           col="Degree_Type",  
           col_order=['Graduate', 'Bachelors', 'Associates'])  
  
plt.show()
```



Vẽ biểu đồ trên Data Aware Grid

```
# Create an lmplot that has a column for Ownership, a row for Degree_Type
# and hue based on the WOMENONLY column and columns defined by inst_order
sns.lmplot(data=df,
          x='SAT_AVG_ALL',
          y='Tuition',
          col="Ownership",
          row='Degree_Type',
          row_order=['Graduate', 'Bachelors'],
          hue='WOMENONLY',
          col_order=['Public', 'Private non-profit'])

plt.show()
```



Vẽ biểu đồ trên Data Aware Grid

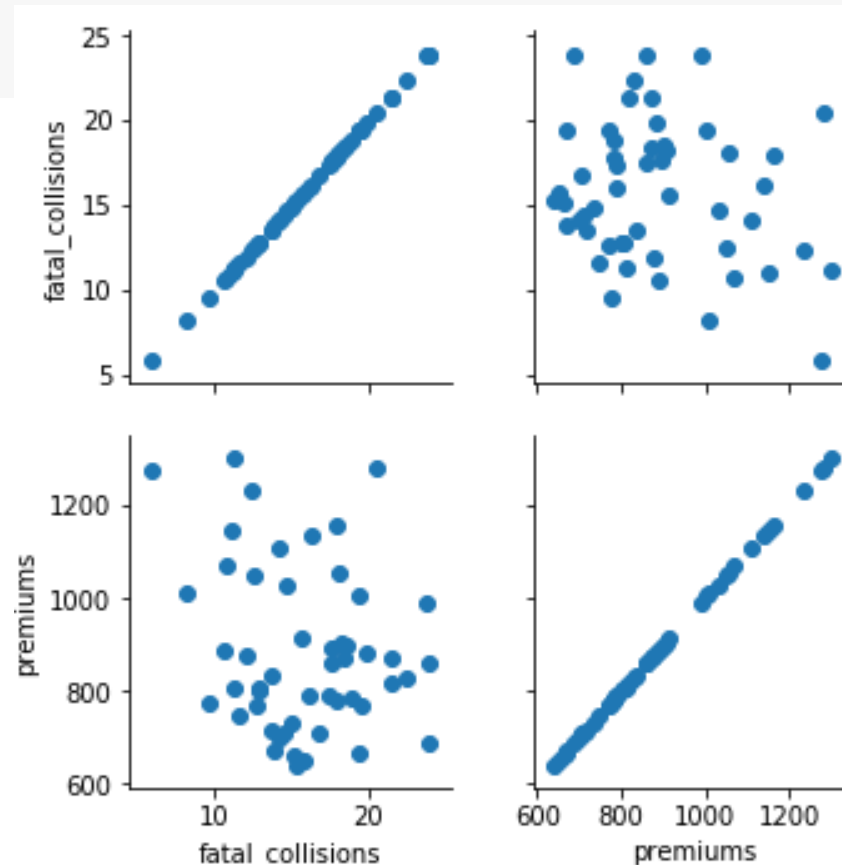
❑ PairGrid

- Khi khám phá một tập dữ liệu, một trong những nhiệm vụ cần làm đầu tiên là khám phá mối quan hệ giữa các cặp biến (Pairwise relationships).
- Seaborn hỗ trợ phân tích pair-wise bằng cách sử dụng PairGrid.
- Vẽ một mạng lưới các ô con bằng cách sử dụng cùng một loại biểu đồ để trực quan hóa dữ liệu.

Vẽ biểu đồ trên Data Aware Grid

• Tạo PairGrid

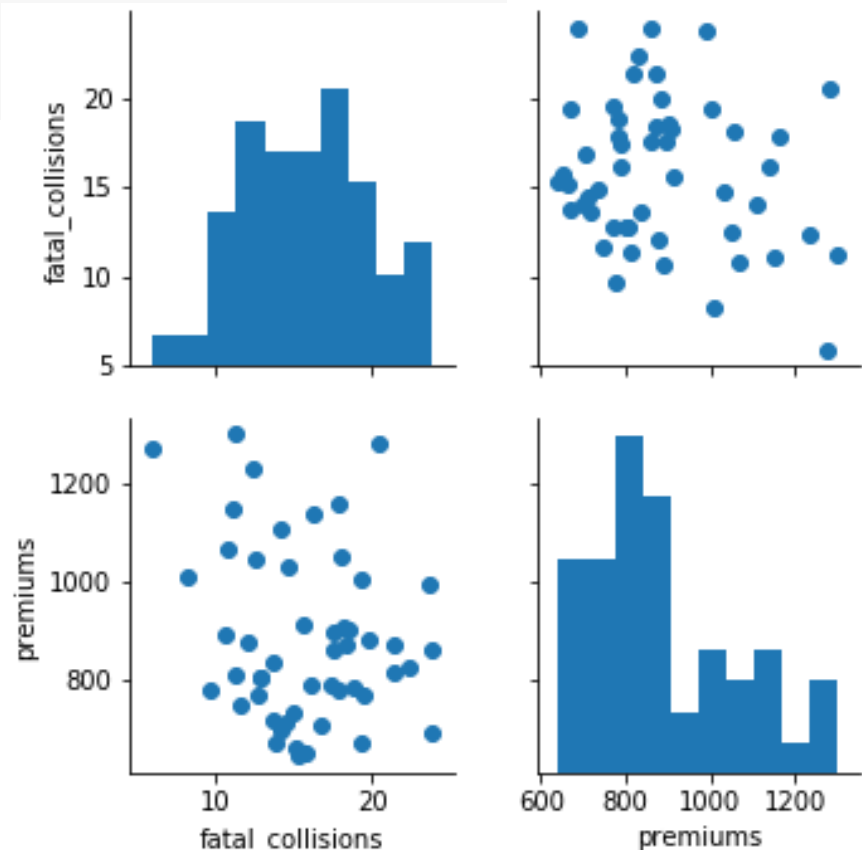
```
# Create a PairGrid with a scatter plot for fatal_collisions and premiums  
g = sns.PairGrid(df, vars=["fatal_collisions", "premiums"])  
g2 = g.map(plt.scatter)  
plt.show()
```



Vẽ biểu đồ trên Data Aware Grid

- Tùy chỉnh đường chéo trên PairGrid

```
# Create the same pairgrid but map a histogram on the diag
g = sns.PairGrid(df, vars=["fatal_collisions", "premiums"])
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter)
plt.show()
```



Vẽ biểu đồ trên Data Aware Grid

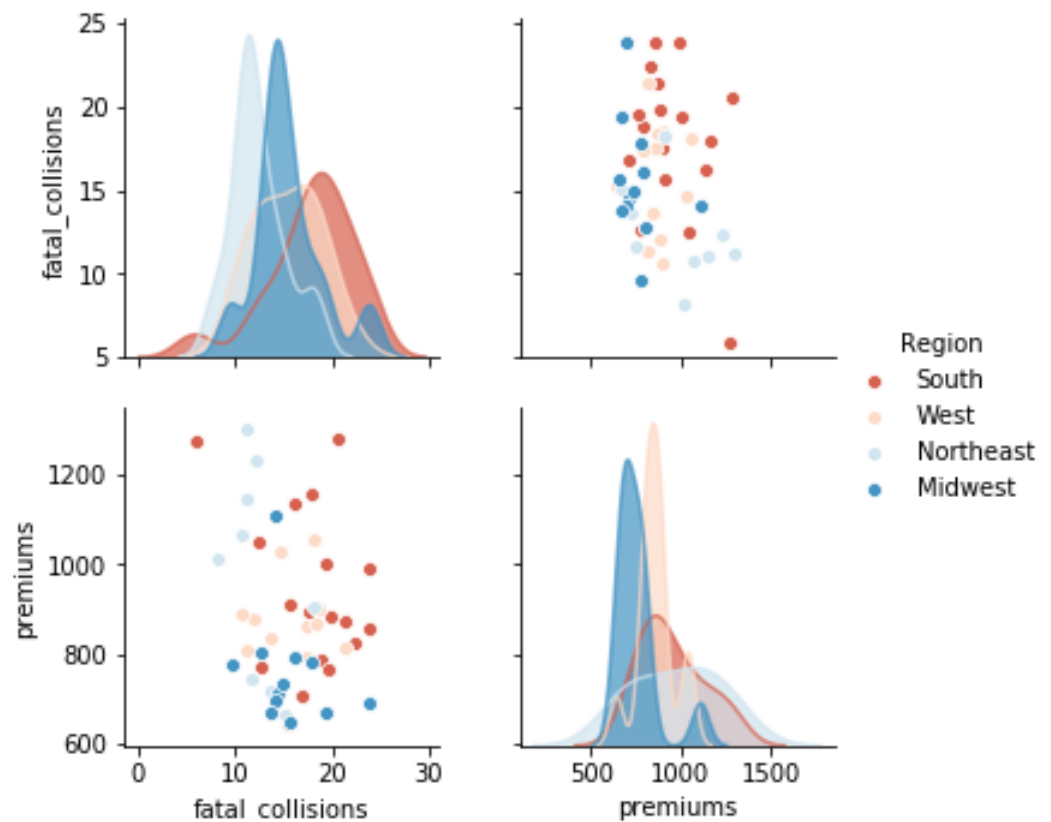
□ PairPlot

- pairplot() là một cách thuận tiện hơn để khám phá mối quan hệ giữa các cặp biến.
- Là một shortcut cho PairGrid

Vẽ biểu đồ trên Data Aware Grid

● PairPlot

```
# Plot a pairplot and use a different color palette and color code by Region
sns.pairplot(data=df,
             vars=["fatal_collisions", "premiums"],
             kind='scatter',
             hue='Region',
             palette='RdBu',
             diag_kws={'alpha':.7})
plt.show()
```



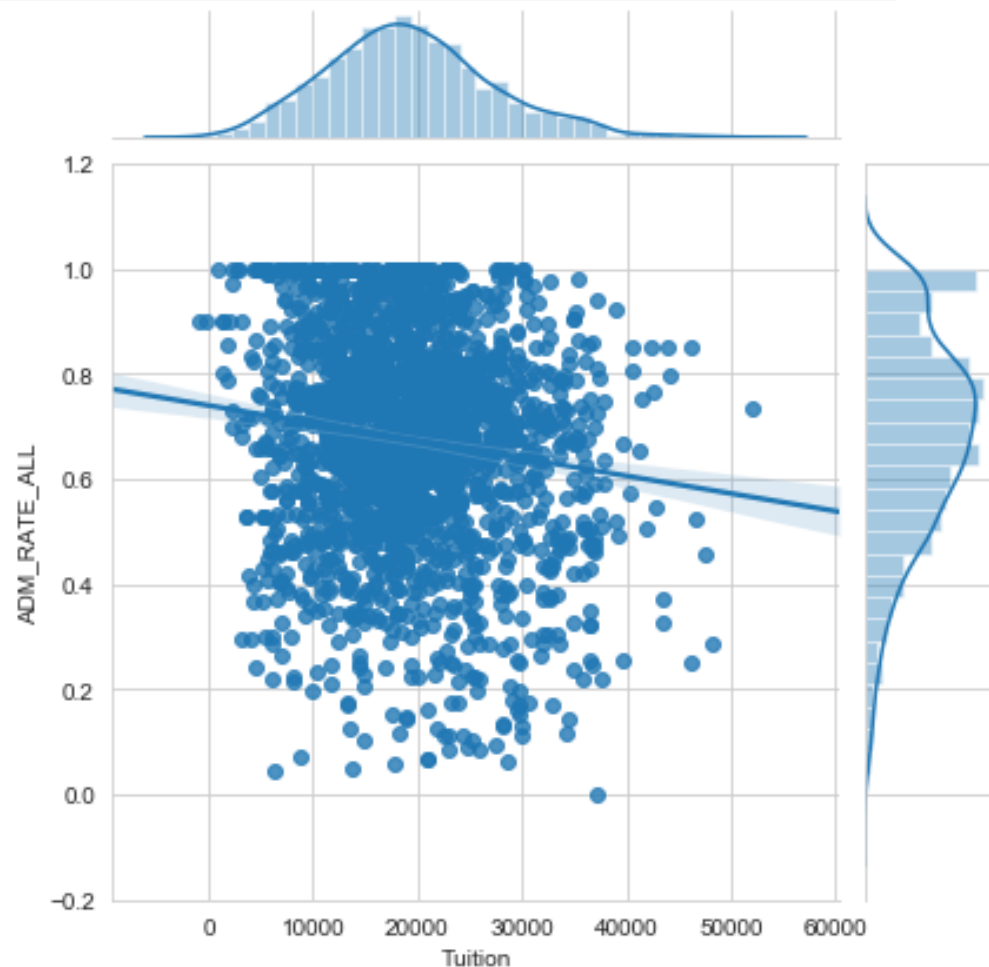
Vẽ biểu đồ trên Data Aware Grid

❑ JointGrid

- JointGrid của Seaborn kết hợp các biểu đồ đơn biến (univariate plot) như histogram, rug plot, kde plot với các biểu đồ nhị phân (bivariate plot) như scatter, regression plot. Seaborn cung cấp các chức năng thuận tiện để kết hợp nhiều biểu đồ với nhau.

Vẽ biểu đồ trên Data Aware Grid

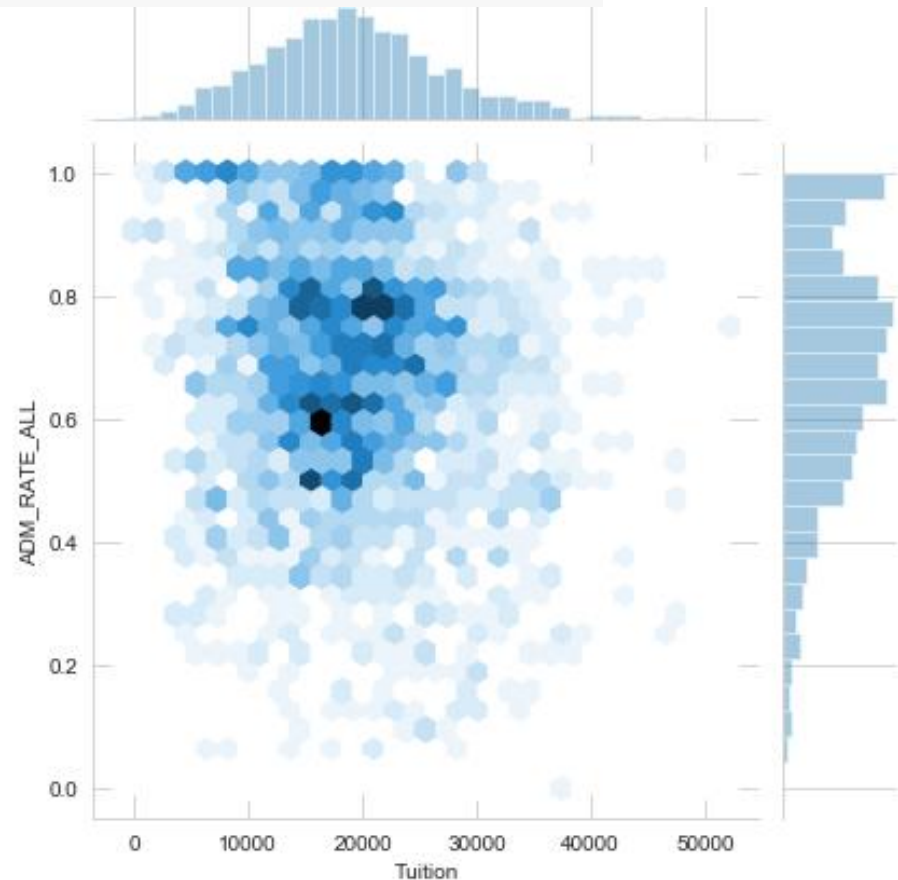
```
g = sns.JointGrid(data=df, x="Tuition", y="ADM_RATE_ALL")  
g.plot(sns.regplot, sns.distplot)  
plt.show()  
plt.clf()
```



Vẽ biểu đồ trên Data Aware Grid

□ Joinplot

```
# Joinplot  
sns.jointplot(data=df, x="Tuition", y="ADM_RATE_ALL", kind='hex')  
plt.show()
```



Nội dung

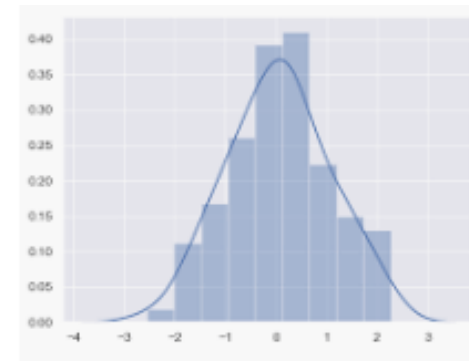
1. Giới thiệu
2. Vẽ biểu đồ với Seaborn
3. Seaborn styles
4. Các loại biểu đồ
5. Vẽ biểu đồ trên Data Aware Grid
6. Tổng kết

Tổng kết

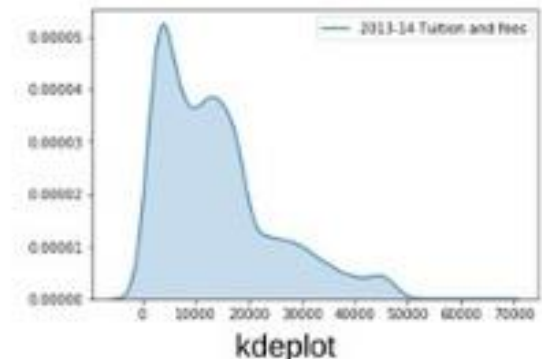
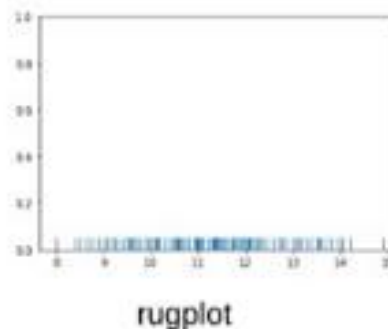
❑ Lựa chọn biểu đồ

- Phân tích phân phối đơn biến (Univariate Distribution Analysis)

- Tốt nhất nên dùng `distplot()`



- Phương án thay thế: `rugplot()` hoặc `kdeplot()`

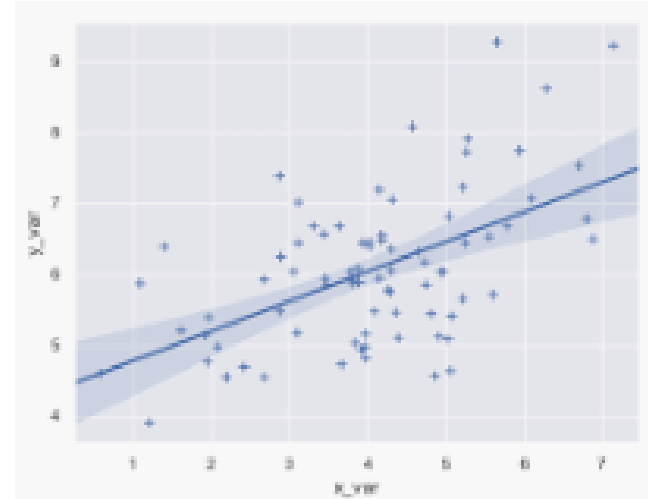
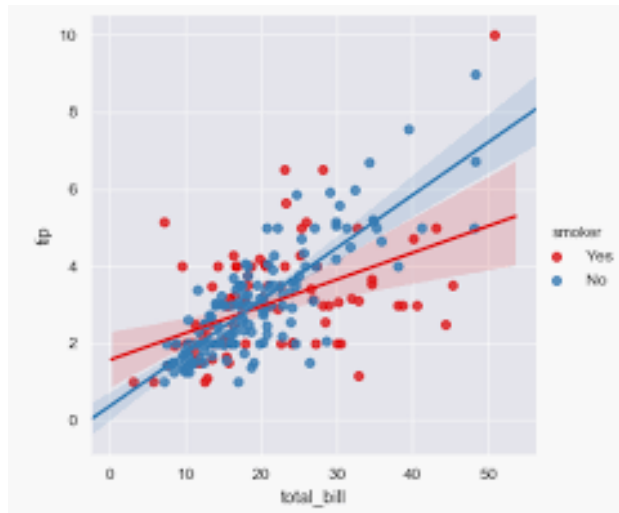


- *Note: `plt.hist()` của Matplotlib*

Tổng kết

❑ Lựa chọn biểu đồ

- Phân tích hồi quy (Regression Analysis)
 - Dùng `Implot()`, `regplot()`



- *Note: `plt.scatter()` của Matplotlib*

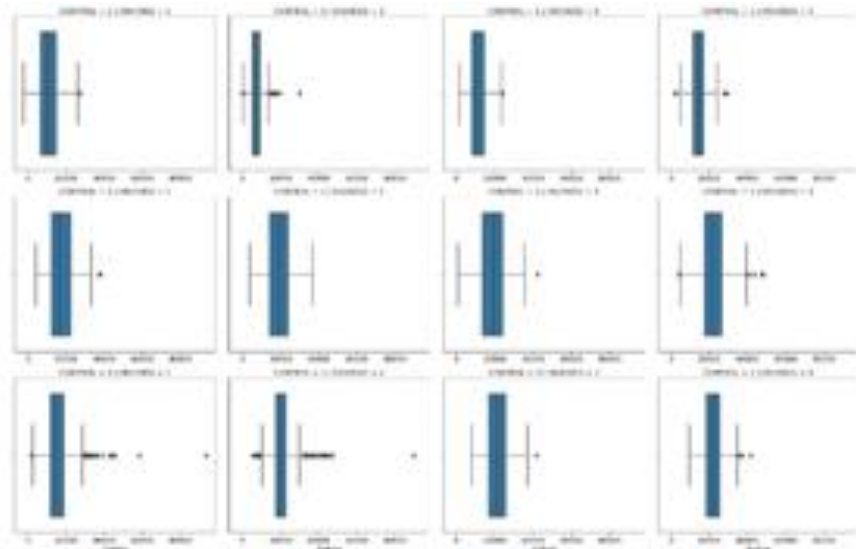
Tổng kết

□ Lựa chọn biểu đồ

- Biểu đồ phân loại

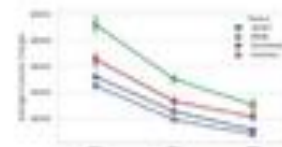
- `factorplot()`
- `barplot()`, `pointplot()`, `countplot()`
- `boxplot()`, `violinplot()`, `boxenplot()`
- `stripplot()`, `swarmplot()`

Tổng kết

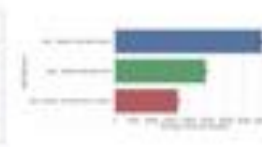


factorplot

FacetGrid



pointplot



barplot



countplot



boxplot



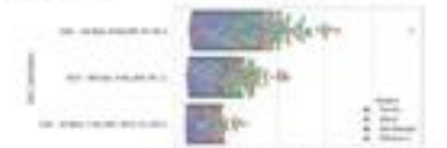
violinplot



lvplot



stripplot



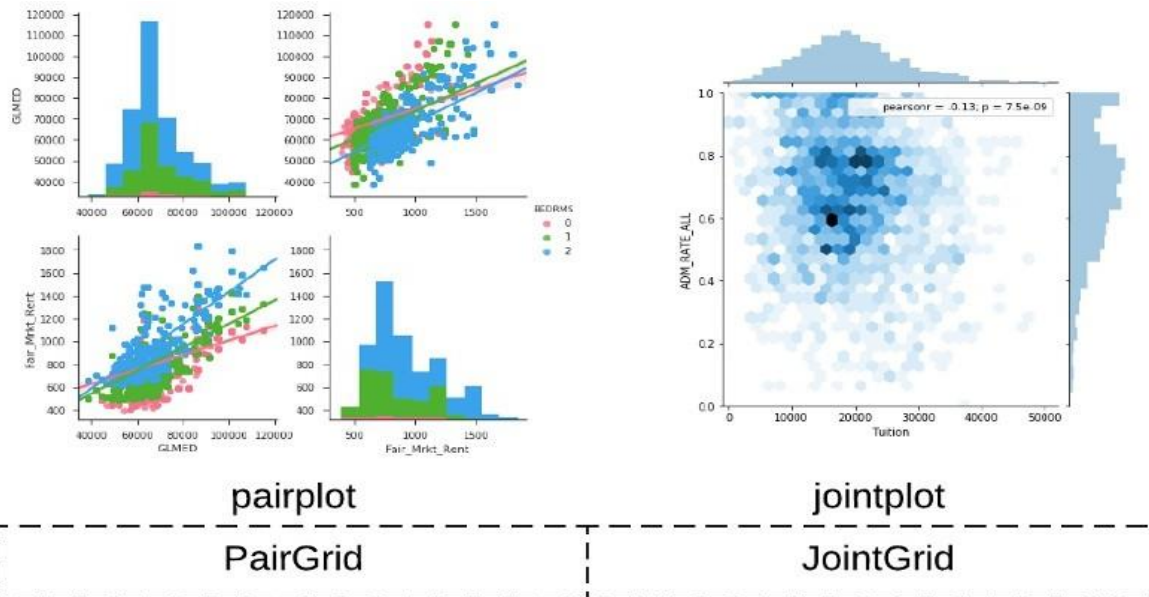
swarmplot

Tổng kết

□ Lựa chọn biểu đồ

- Pairplot, jointplot

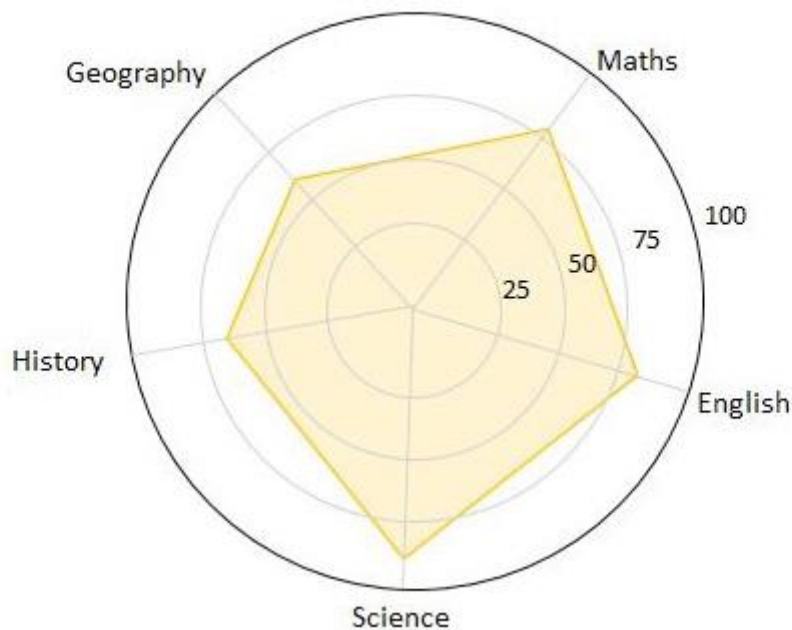
- Thực hiện phân tích hồi quy với Implot()
- Phân tích phân phối với distplot()





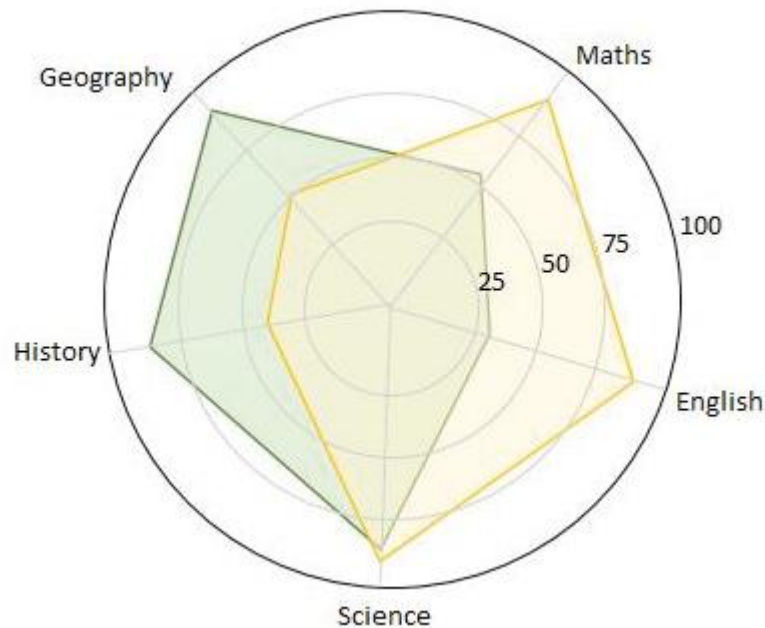
RADAR CHART

The following diagram shows a radar chart for a single variable. This chart displays data about a student scoring marks in different subjects:

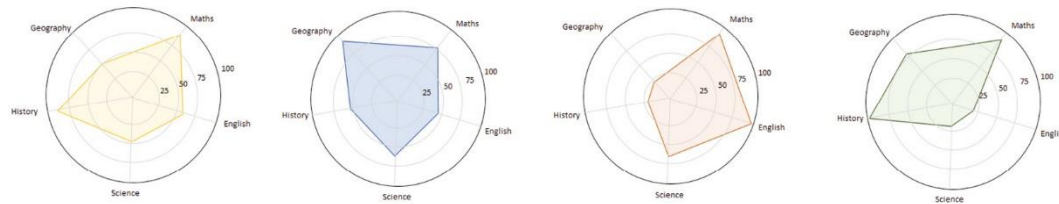


RADAR CHART

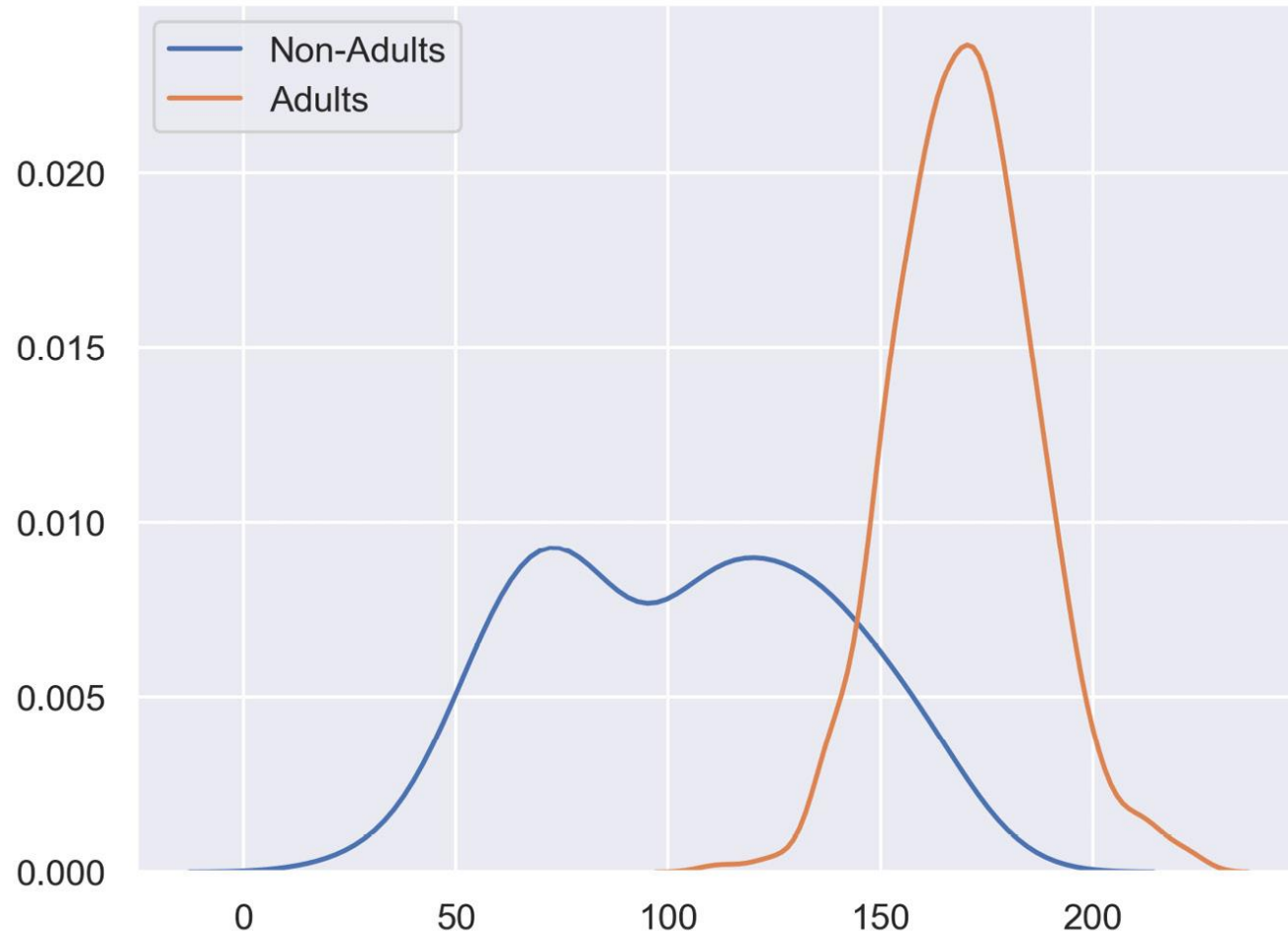
The following diagram shows a radar chart for two variables/groups. Here, the chart explains the marks that were scored by two students in different subjects:



The following diagram shows a radar chart for multiple variables/groups. Each chart displays data about a student's performance in different subjects:



DENSITY PLOT



Grid Type	Grid Function	Axes Functions	Variable Type
FacetGrid	factorplot	stripplot, swarmplot, boxplot, violinplot, lmplot, pointplot, barplot, countplot	Categorical
FacetGrid	lmplot	regplot	Continuous
PairGrid	pairplot	regplot, distplot, kdeplot	Continuous
JointGrid	jointplot	regplot, kdeplot, residplot	Continuous
ClusterGrid	clustermap	heatmap	Continuous

FacetGrid vs. AxesSubplot objects

Object Type	Plot Types	Characteristics
FacetGrid	<code>relplot()</code> , <code>catplot()</code>	Can create subplots
AxesSubplot	<code>scatterplot()</code> , <code>countplot()</code> , etc.	Only creates a single plot

Understanding Plotting

- Know whether the plotting method requires one or two variables
- Line, bar, and scatter plots require two variables
- Bar plots require:
 - X-coordinates to locate the bar
 - Another variable for the height of the bar
- Boxplots, histograms, and KDEs use only a single variable

point. The same holds true for bar plots, which requires some x coordinates to

Packt>

Introduction

- When beginning visualization:
 - Focus only on univariate plots
- Bar charts – univariate plots for categorical data
- Histograms, boxplots, or KDEs – univariate plots for continuous data

univariate plots tend to be bar charts for categorical data that I usually

Packt>

Show transcript

Plotting Flights Per Week

- Use time series plot with dates on x axis
- `to_datetime` function has nifty trick that identifies column names
- If you have DataFrame with year, month, and day:
 - Pass this DataFrame to `to_datetime` function
 - It returns a sequence of Timestamps

x-axis. Unfortunately, we don't have pandas timestamps in any of the columns.

Doing | x Google | x Section | x Section | x operat | x Bài 07: | x Activity | x activity | x Demo | x Demo | x matplotlib | x + -

subscription.packtpub.com/video/big_data_and_business_intelligence/9781789340495/61404/61407/doing-multivariate-analysis-with-seaborn-grids

BACK

Data Visualization Recipes in Python

Video

Theodore Petros

Data Visualization Recipes in Python

Generate visually stunning and explanatory plots with Python

Contents Bookmarks (0)

Visualization with Matplotlib and Pandas

Visualization with Seaborn and Pandas

Stacking Area Charts to Discover Emerging Trends

Understanding the Differences Between Seaborn and Matplotlib

Doing Multivariate Analysis with Seaborn Grids

Uncovering Simpson's Paradox in the Diamonds Dataset...

Grids

Hierarchy Between Functions (Continued)

- Seaborn Axes functions – called independently to produce single plot
- Grid functions:
 - Uses Axes functions to build grid
 - Final objects returned are of grid type
- Advanced use cases necessitate direct use of Grid types
- Mostly, calling underlying Grid functions is needed

▶ 🔊 -3:50 1x

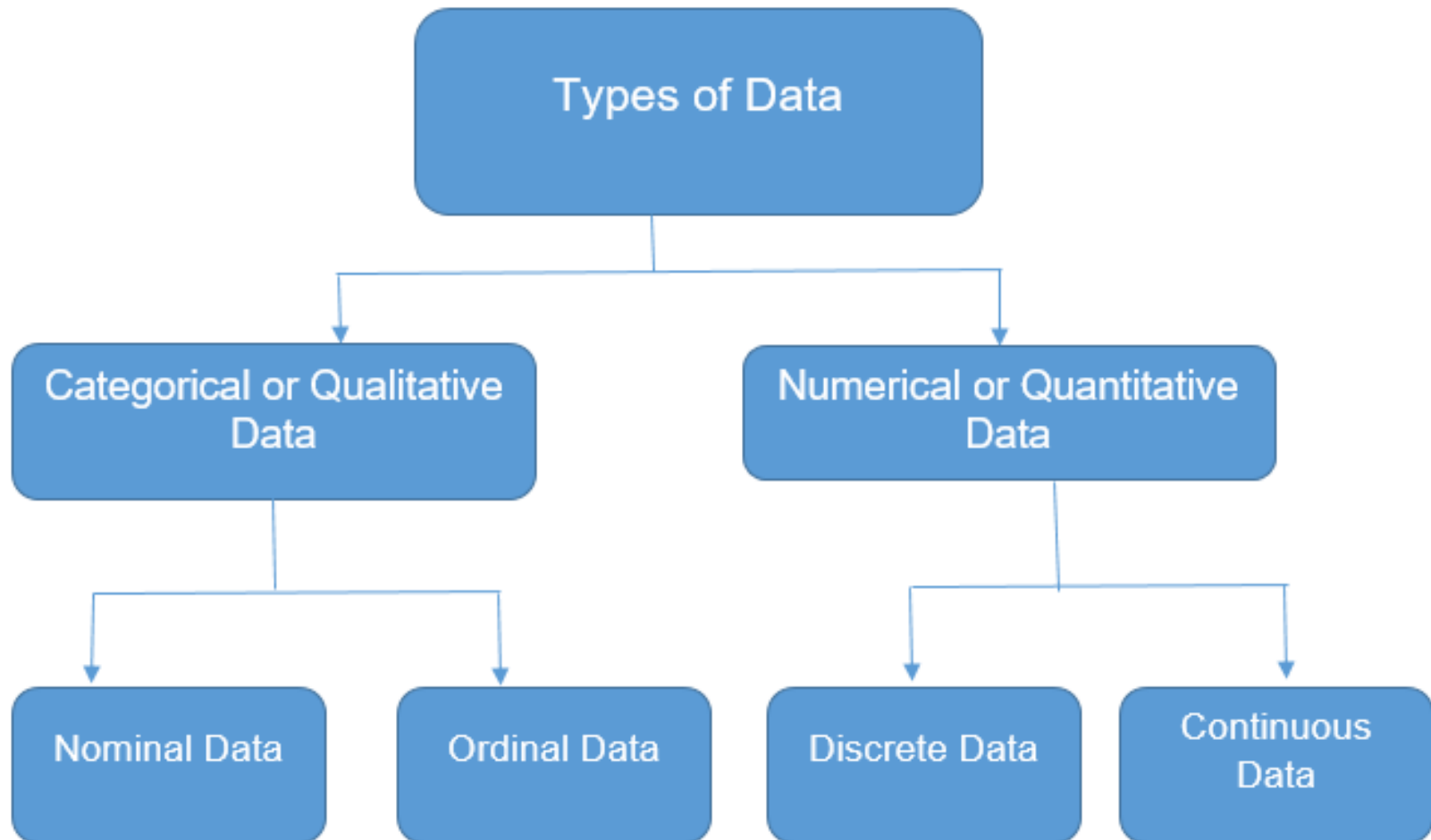
[Show transcript](#)

To understand seaborn further, it is helpful to be aware of the hierarchy between the functions that return multiple Axes as a seaborn Grid and those that return single Axes. Let's perform multivariate analysis with seaborn grids.

Create a simple regression plot with a seaborn Axes function

Windows taskbar: 6:42 AM 4/13/2020

Loại dữ liệu



Titanic data

1. PassengerId: Unique Id of a passenger
2. Survived: If the passenger survived(0-No, 1-Yes)
3. Pclass: Passenger Class (1 = 1st, 2 = 2nd, 3 = 3rd)
4. Name: Name of the passenger
5. Sex: Male/Female
6. Age: Passenger age in years
7. SibSp: No of siblings/spouses aboard
8. Parch: No of parents/children aboard
9. Ticket: Ticket Number
10. Fare: Passenger Fare
11. Cabin: Cabin number
12. Embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)



Iris Versicolor

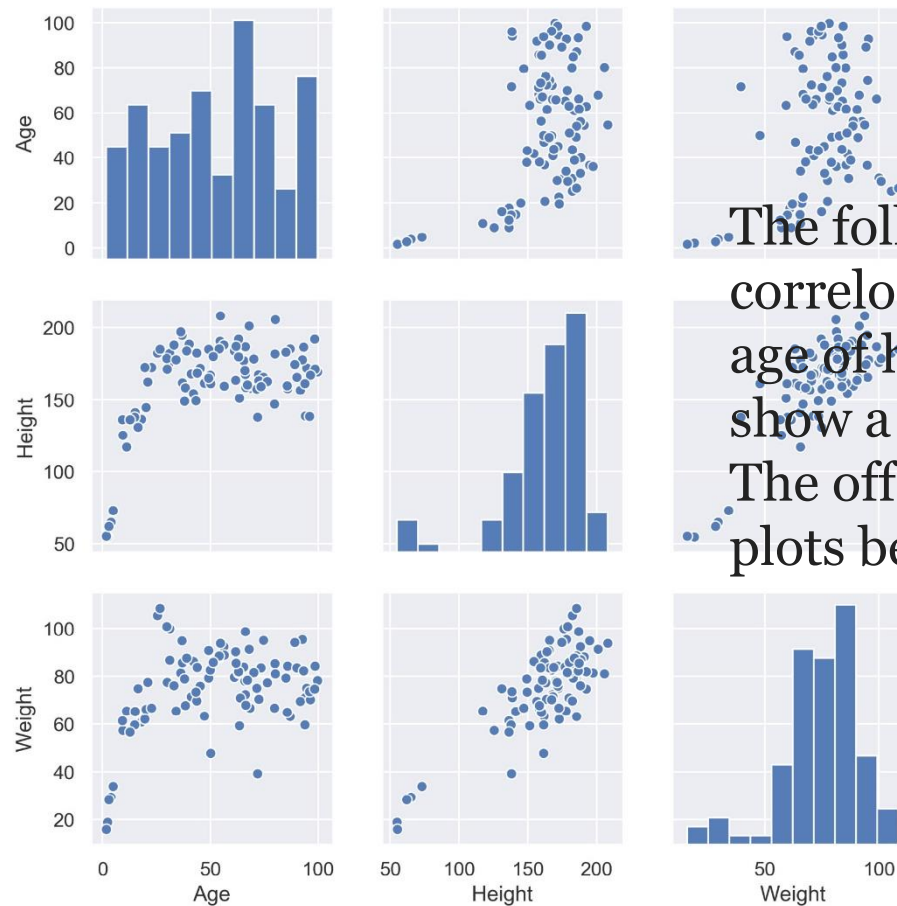


Iris Setosa

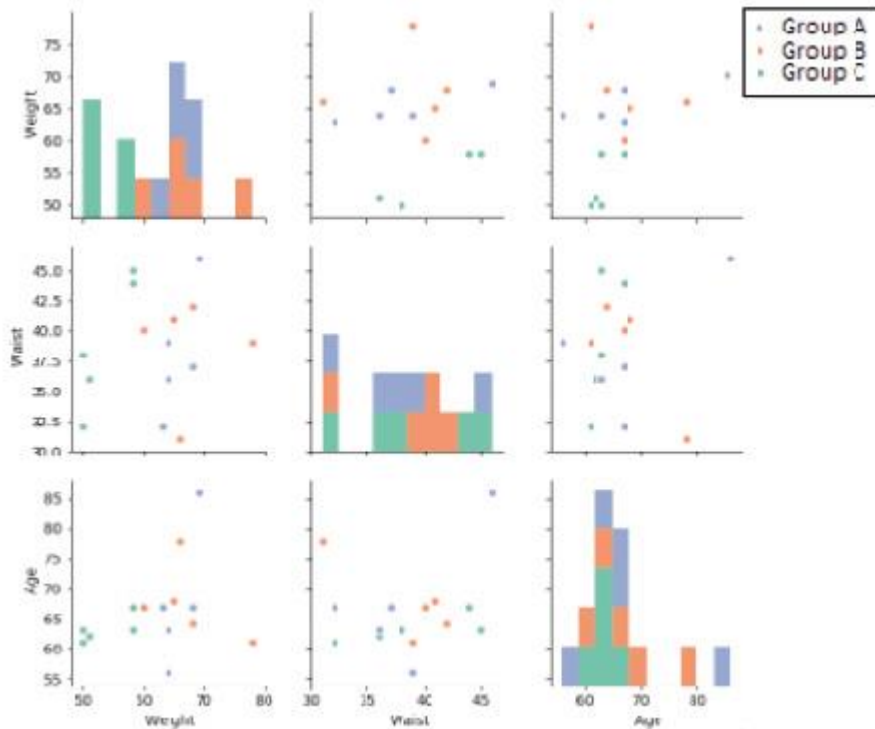


Iris Virginica

❑ <https://www.kaggle.com/kralmachine/seaborn-tutorial-for-beginners/notebook>







The following diagram shows a correlogram for height, weight, and age of humans. The diagonal plots show a histogram for each variable. The off-diagonal elements show scatter plots between variable pairs:




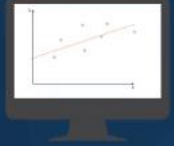


The following diagram shows the correlogram with data samples separated by color into different groups:

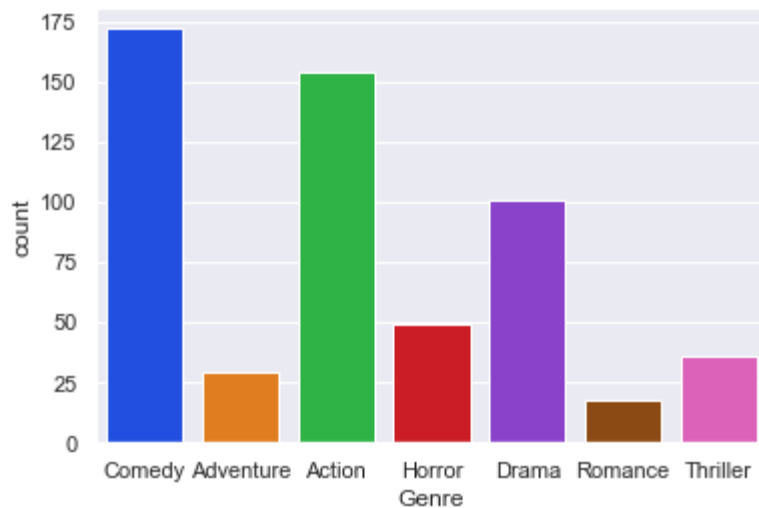
CORRELATION

- 1 Relationship 
- 2 Movement together 
- 3 $\rho(x,y) = \rho(y,x)$ 
- 4 Single point 

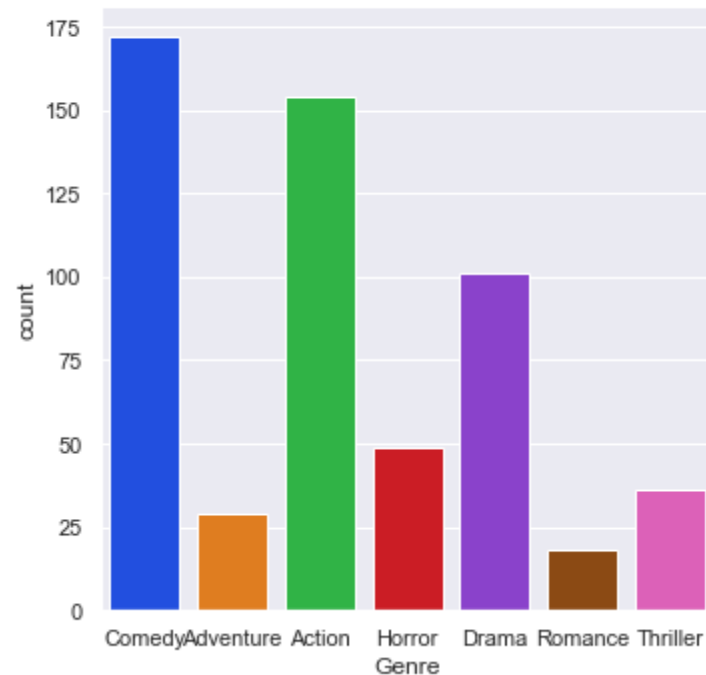
REGRESSION

- One variable affects the other 
- cause and effect 
- One way 
- Line 

```
sns.countplot(x='Genre',data=movie)  
plt.show()
```



```
: sns.catplot(x='Genre',data=movie, kind="count")  
plt.show()
```



Đặt tiêu đề cho FacetGrid

Adding a title to AxesSubplot

FacetGrid

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")
```

```
g.fig.suptitle("New Title",  
               y=1.03)
```

AxesSubplot

```
g = sns.boxplot(x="Region",  
                y="Birthrate",  
                data=gdp_data)
```

```
g.set_title("New Title",  
            y=1.03)
```

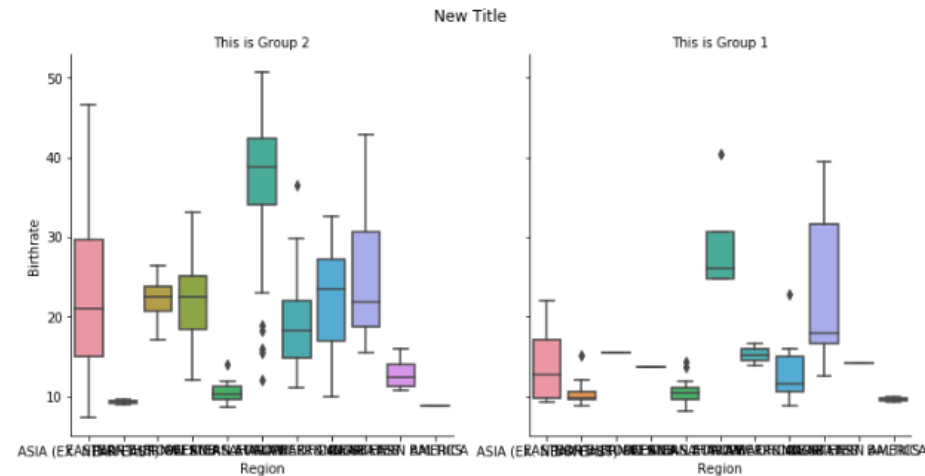
Đặt tiêu đề cho FacetGrid

Titles for subplots

```
g = sns.catplot(x="Region",
                y="Birthrate",
                data=gdp_data,
                kind="box",
                col="Group")

g.fig.suptitle("New Title",
               y=1.03)

g.set_titles("This is {col_name}")
```




```
g = sns.catplot(x="Region",
                y="Birthrate",
                data=gdp_data,
                kind="box")

g.set(xlabel="New X Label",
      ylabel="New Y Label")

plt.show()
```

