TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐỀ CƯƠNG

SỬ DỤNG NGÔN NGỮ PYTHON VÀO GIẢI QUYẾT CÁC BÀI TOÁN KINH TẾ

- **GVHD: PGS.TS Nguyễn Đình Thuân**
- **SVTH :**

  15520987 – Trần Văn Tùng

  15520708 – Lê Thị Đỗ Quyên

Tp.Hồ Chí Minh, Ngày 21 tháng 5 năm 2018

Mục Lục

I. Giới thiệu ngôn ngữ Python

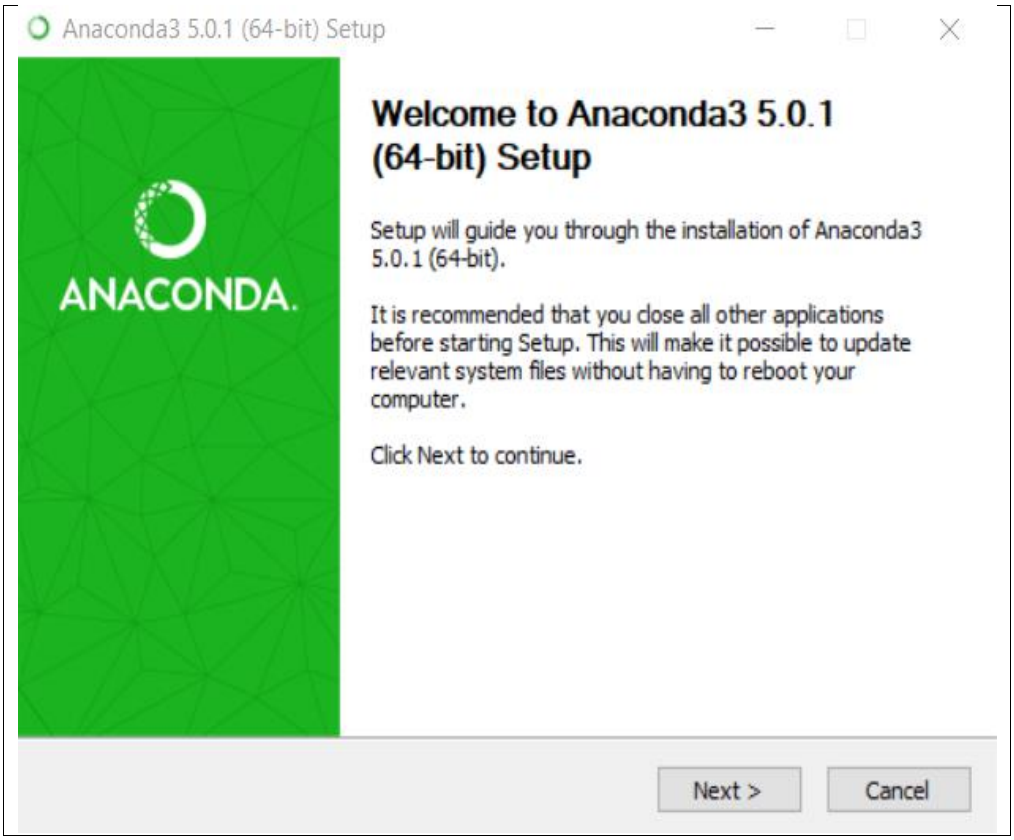
1. Giới thiệu:

- Python là một ngôn ngữ lập trình thông dịch do Guido van Rossum tạo ra năm 1990, nó được xem là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, như nhận định của chính Guido van Rossum trong một bài phỏng vấn ông. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động, do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics).
- Ưu điểm :
  - Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented).
  - Hỗ trợ module và hỗ trợ gói (package).
  - Xử lý lỗi bằng ngoại lệ (Exception).
  - Kiểu dữ liệu động ở mức cao.
  - Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
  - Đơn giản: cú pháp đơn giản giúp cho người lập trình dễ học và tìm hiểu
  - Python có tốc độ xử lý nhanh.
  - Tương tác: chế độ tương tác cho phép người lập trình thử nghiệm tương tác sửa lỗi của các đoạn mã.
  - Chất lượng: thư viện có tiêu chuẩn cao, Python có khối cơ sở dữ kiệu khá lớn, nhằm cung cấp giao diện cho các cơ sở dữ liệu thương mại lớn. Đồng thời Python được biên dịch và chạy trên tất cả nền tảng lớn hiện nay.
  - Mở rộng: Python cho phép người lập trình có thể thêm hoặc tùy chỉnh các công cụ nhằm tối đa hiệu quả có thể đạt được trong công việc.
  - GUI programming: giúp cho việc thực hiện ảnh minh họa di động một cách tự nhiên sống động.
- Nhược điểm:
  - Python không có các thuộc tính như: protected, private hay public, không có vòng lặp do .. while, switch .. case .
  - Python mặc dù nhanh hơn PHP, nhưng không nhanh so với C++ và Java.

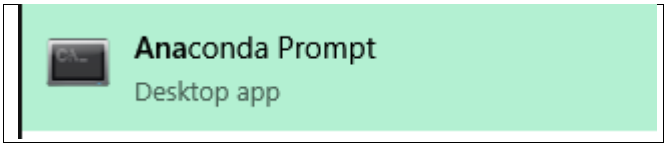
- Giới thiệu trình soạn thảo sử dụng cho báo cáo. Trong phạm vi phần báo cáo, chúng tôi sử dụng ngôn ngữ Python để tính hành các câu lênh, và trình soạn thảo Jupyter Notebook để hỗ trợ việc làm thực hiện các câu lệnh.

2. Cài đặt

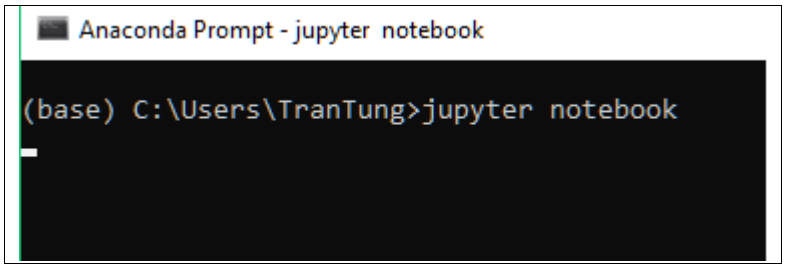
- Tiến hành cài đặt và sử dụng Python.
- Cài đặt Anacoda:



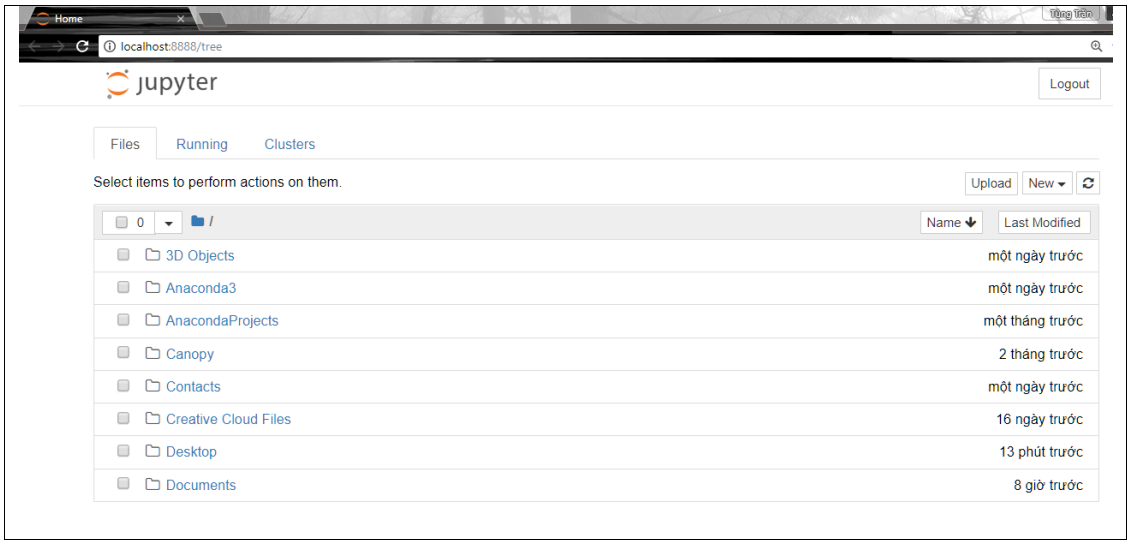
- Sau khi cài đặt thành công, tiến hành mở Anaconda Prompt:



- Tiến hành khởi động jupyter notebook:



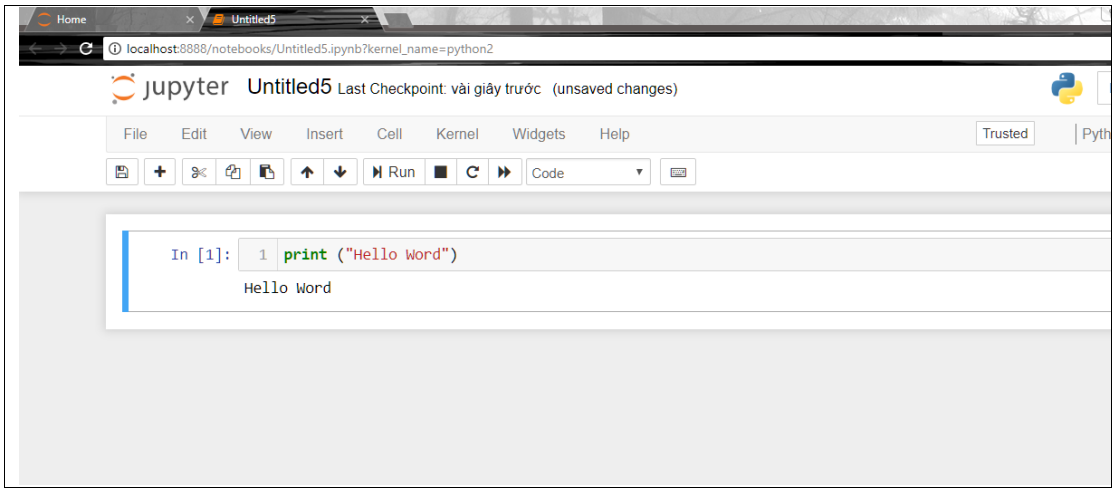
- Giao diện thao tác chính :



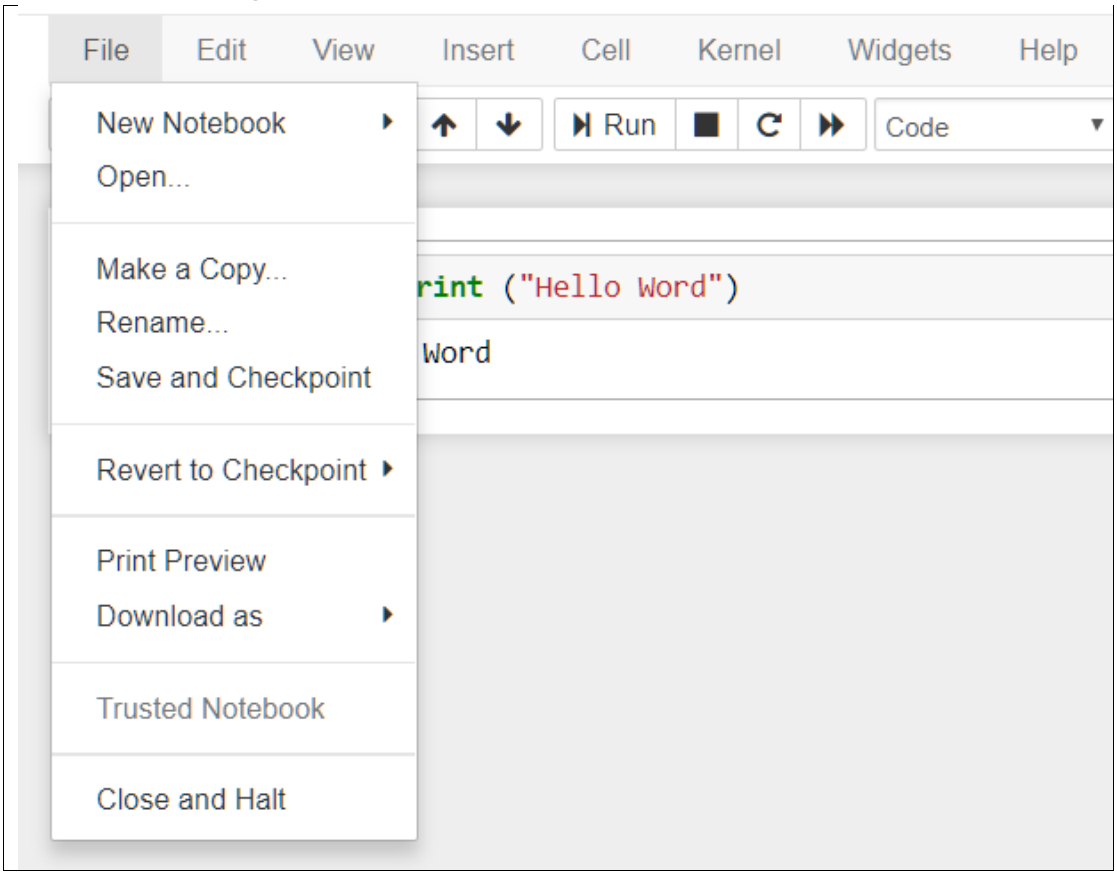
3. Sử dụng

- Tạo một file python mới:

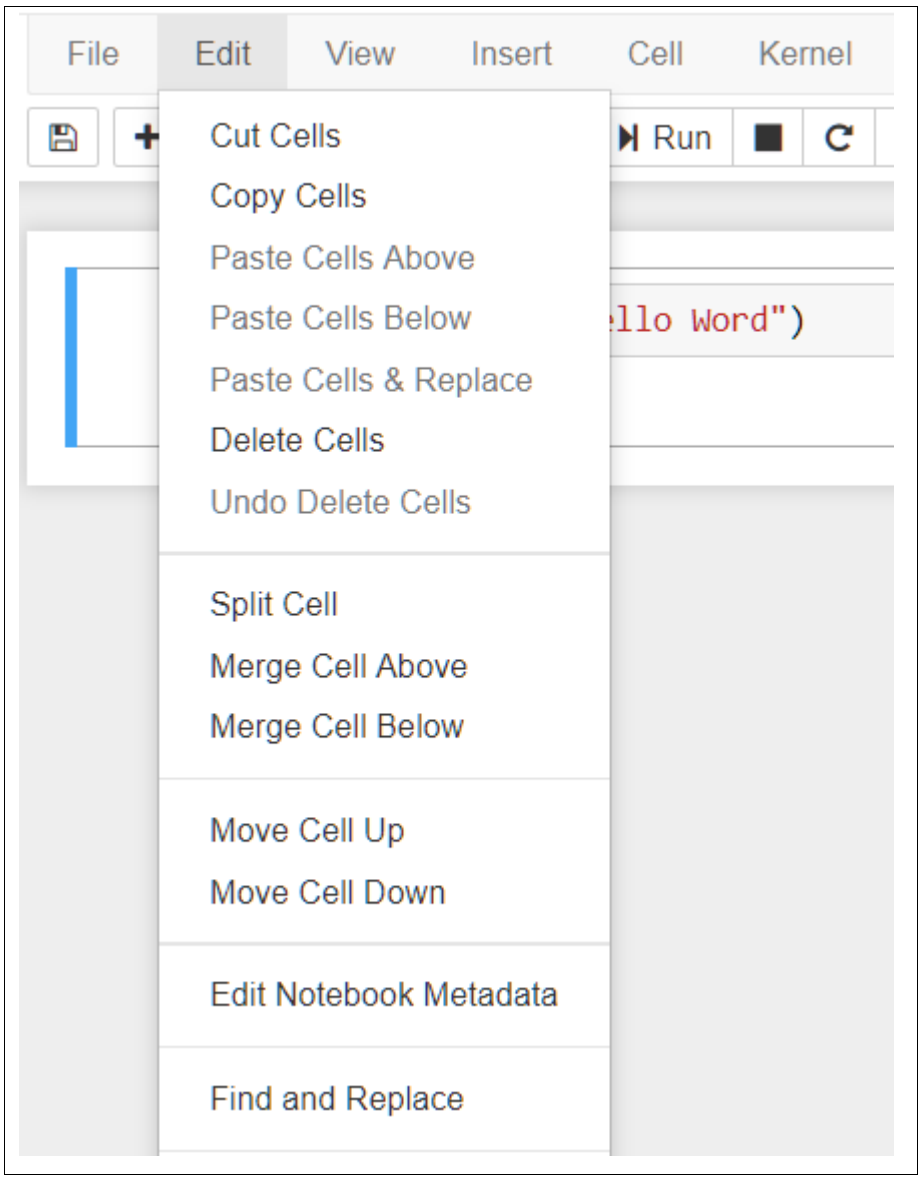
- Các thao tác:



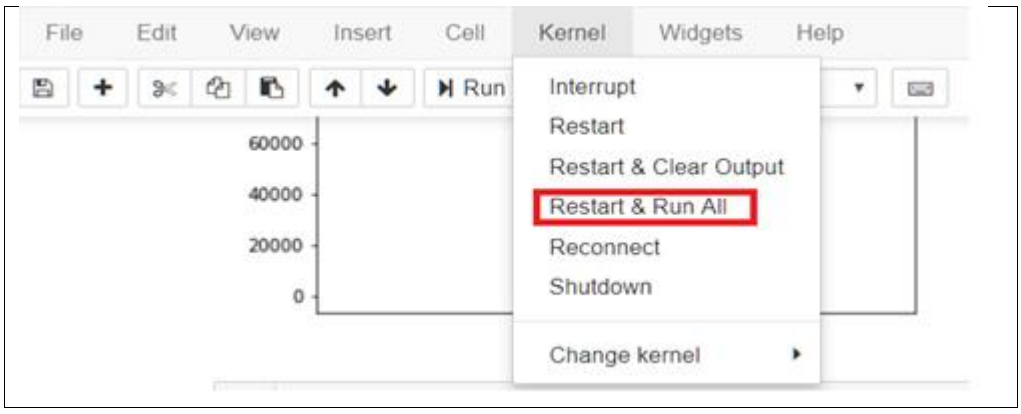
  - Chức năng File:



  - Các chức năng Edit:



- Thực hiện thao tác trên để chạy toàn bộ đoạn code:



- Và còn nhiều chức năng khác.

II. Biện Pháp Thống Kê Mô Tả Và Hiển Thị Dữ Liệu

1. Dữ liệu dân số Việt Nam 1990 - 2018

- Dữ liệu: Data_DanSoVietNam.xlsx
- Code : Descriptive Statistical Measures_DanSoVietNam.ipynb
- Phát biểu bài toán: dữ liệu đưa vào cung cấp số liệu về dân số của Việt Nam trong khoảng thời gian 1990 - 2017. Qua đó chúng tôi thực hiện các phép cơ bản để đưa ra nhận xét về dân số Việt Nam. Tiến hành tính các giá trị: Count, Min, Max, Mean, Median, Mode, Quantile, Range, Mode, Variance, Standard Deviation, Coefficient of Deviation, Skewness, Kurtosis (đơn vị Người).
- Đầu tiên tiến hành import các thư viện cần thiết:

```
import warnings; warnings.simplefilter("ignore")

%matplotlib inline
import matplotlib.pyplot as pl
import seaborn as sns
import numpy as np
import pandas as pd
import scipy
from scipy import stats
import matplotlib.pyplot as pl
```

- Đọc file dữ liệu excel :

```
cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
DanSo = pd.read_excel(r"Data_DanSoVietNam.xlsx", sheetname = 0)
```

- Các giá trị:

```
1 DanSo["Population"].max()
```

```
95554478
```

```
1 DanSo["Population"].min()
```

```
68209605
```

```
1 DanSo["Population"].mean()
```

```
83214790.86206897
```

```
1 DanSo["Population"].median()
```

```
83527678.0
```

```
1 DanSo["Population"].quantile()
```

```
83527678.0
```

```
1 DanSo["Population"].max() - DanSo["Population"].min()
```

```
27344873
```

```
1 DanSo["Population"].var()
```

```
64591519909616.05
```

```
1 DanSo["Population"].std()
```

```
8036884.963069712
```

```
1 np.cov(DanSo["Population"])
```

```
array(6.45915199e+13)
```

```
from scipy.stats import kurtosis, skew
kurtosis(DanSo["Population"])
```

```
-0.991800508992922
```

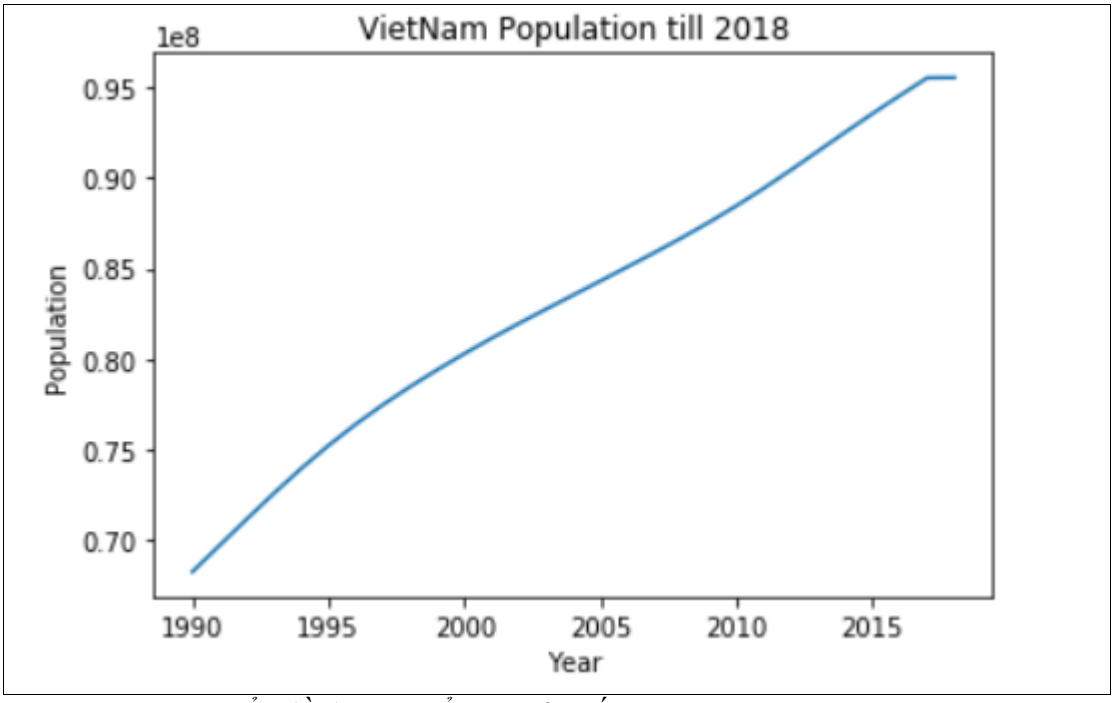
```
skew(DanSo["Population"])
```

```
-0.15692564409609483
```

- Giải thích số liệu:
  - Max: cho biết giá trị lớn nhất của các quan sát.
  - Min: cho biết giá trị nhỏ nhất của các quan sát.
  - Median: cho biết giá trị trung vị của các quan sát.
  - Mean: cho biết giá trị trung bình của các quan sát.
  - Var (phương sai): dùng để đo lường mức độ phân tán của tập các giá trị quan sát
  - Std (độ lệch chuẩn): đo độ phân tán dữ liệu xung quanh giá trị trung bình của nó.
  - Skewness: cho biết dạng phân phối của các giá trị quan sát, với skew < 0: các giá trị quan sát sẽ tập trung chủ yếu vào các giá trị nhỏ nhất.
  - Kurtosis: đánh giá dỉnh của đừng cong quan sát với dạng phân phối chuẩn, với kur < 0 đường cong có dạng hẹp hơn hay tương đối bằng phẳng.

- Minh họa dữ liệu bằng biểu đồ:
  - Biểu đồ đường:

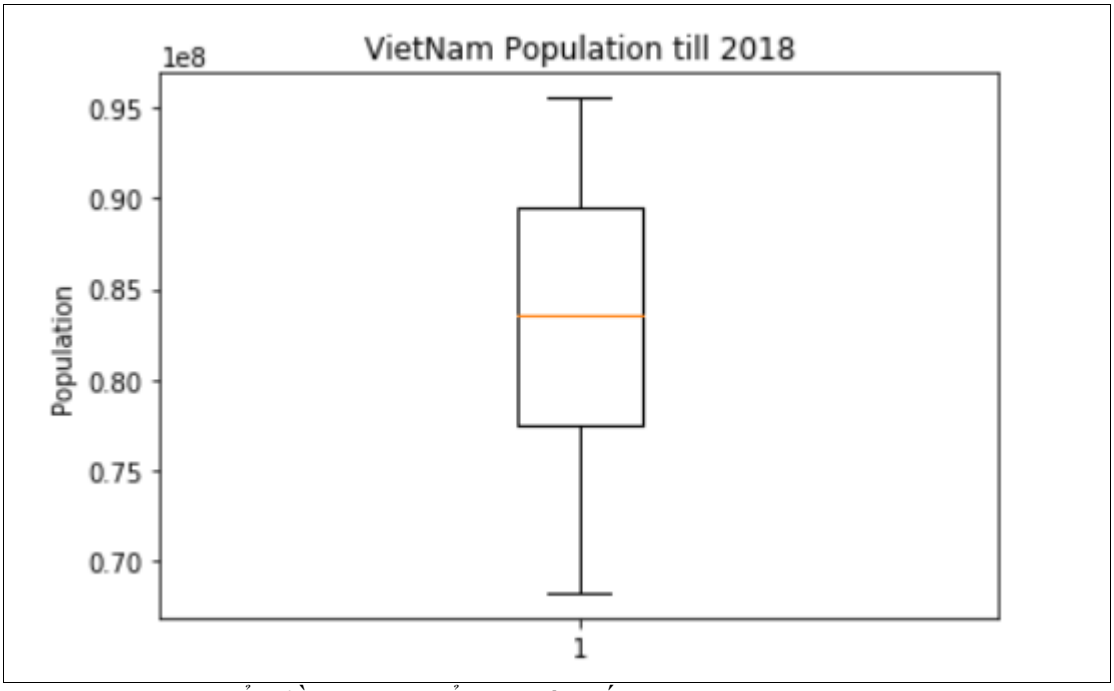
```
x = DanSo['Year']
y = DanSo['Population']
pl.xlabel('Year')
pl.ylabel('Population')
pl.title('VietNam Population till 2018')
pl.plot(x, y)
pl.show()
```



Biểu đồ đường thể hiện dân số Việt Nam 1990 – 2018

- Biểu đồ Boxplot

```
1  pl.boxplot(y)
2  pl.ylabel('Population')
3  pl.title('VietNam Population till 2018')
```



Biểu đồ boxplot thể hiện dân số Việt Nam 1990 – 2018

- Nhận xét: qua các phép tính cơ bản ta thấy dân số Việt Nam tăng nhanh trong giai đoạn 1990 - 2018, và là nước có dân số cao so với diện tích trong khu vực và thế giới.

2. Tỷ Lệ Dân Thành Thị 2010 - 2016

- Dữ liệu: Data_TiLeDanThanhThi.xlsx

- Code: Descriptive Statistical Measures_TyLeDanThanhThi.ipynb

- Phát biểu bài toán: dữ liệu đưa vào cung cấp số liệu về tỷ lệ dân thành thị của Việt Nam trong khoảng thời gian 2010 - 2016 . Qua đó chúng tôi thực hiện các phép cơ bản để đưa ra nhận xét về dân số Việt Nam. Tiến hành tính các giá trị: Count, Min, Max, Mean, Median, Mode, Quantile, Range, Mode, Variance, Standard Deviation, Coefficient of Deviation, Skewness, Kurtosis (đơn vị Người).

- Đầu tiên tiến hành import các thư viện cần thiết:

```
import warnings; warnings.simplefilter("ignore")

%matplotlib inline
import matplotlib.pyplot as pl
import seaborn as sns
import numpy as np
import pandas as pd
import scipy
from scipy import stats
import matplotlib.pyplot as pl
```

- Đọc file dữ liệu excel :

```
cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
ThanhThi = pd.read_excel(r"Data_TiLeDanThanhThi.xlsx", sheetname = 0)
```

- Các giá trị:

```
1 ThanhThi["Ti le dan thanh thi(%)"].max()
```

```
34.6
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].min()
```

```
29.9
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].mean()
```

```
32.472857142857144
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].median()
```

```
32.45
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].quantile()
```

```
32.45
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].max() - ThanhThi["Ti le dan thanh thi(%)"].min()
```

```
4.700000000000003
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].var()
```

```
3.066157142857142
```

```
1 ThanhThi["Ti le dan thanh thi(%)"].std()
```

```
1.75104458619909
```

```
1 np.cov(ThanhThi["Ti le dan thanh thi(%)"])
```

```
array(3.06615714)
```

```
from scipy.stats import kurtosis, skew
kurtosis(ThanhThi["Ti le dan thanh thi(%)"])
```

```
-1.1820840325408153
```

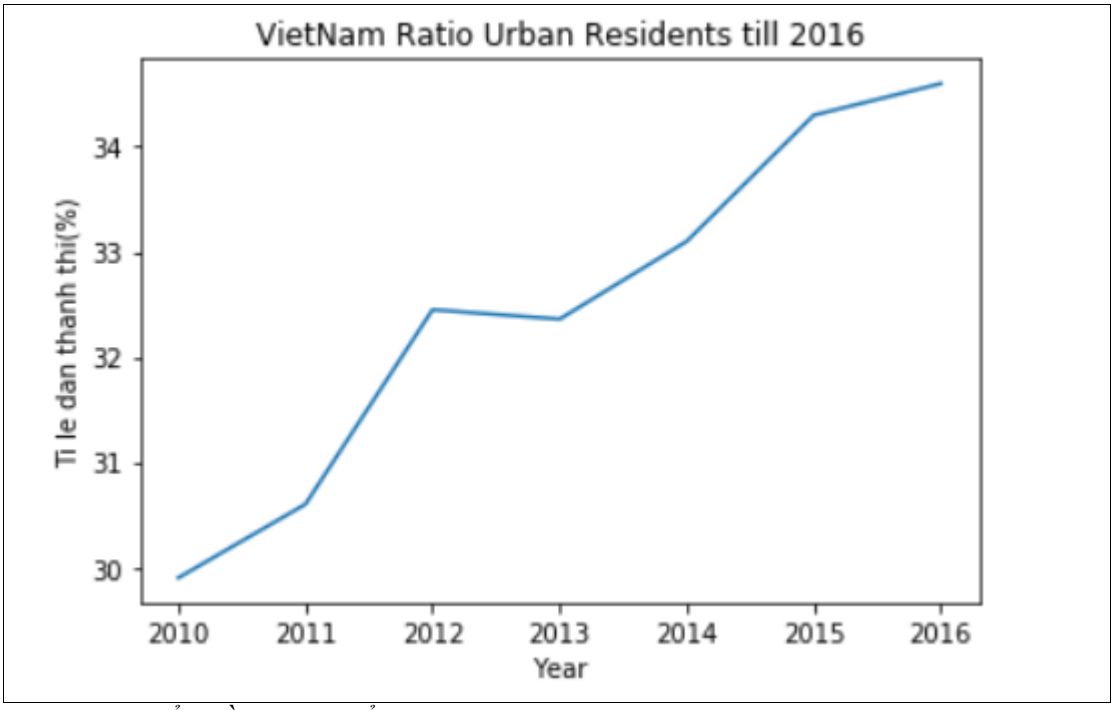
```
skew(ThanhThi["Ti le dan thanh thi(%)"])
```

```
-0.25585533031912483
```

- Giải thích số liệu:
  - Max: cho biết giá trị lớn nhất của các quan sát.
  - Min: cho biết giá trị nhỏ nhất của các quan sát.
  - Median: cho biết giá trị trung vị của các quan sát.
  - Mean: cho biết giá trị trung bình của các quan sát.
  - Var (phương sai): dùng để đo lường mức độ phân tán của tập các giá trị quan sát
  - Std (độ lệch chuẩn): đo độ phân tán dữ liệu xung quanh giá trị trung bình của nó.
  - Skewness: cho biết dạng phân phối của các giá trị quan sát, với skew < 0: các giá trị quan sát sẽ tập trung chủ yếu vào các giá trị nhỏ nhất.
  - Kurtosis: đánh giá dỉnh của đừng cong quan sát với dạng phân phối chuẩn, với kur < 0 đường cong có dạng hẹp hơn hay tương đối bằng phẳng.

- Minh họa dữ liệu bằng biểu đồ:
  - Biểu đồ đường:

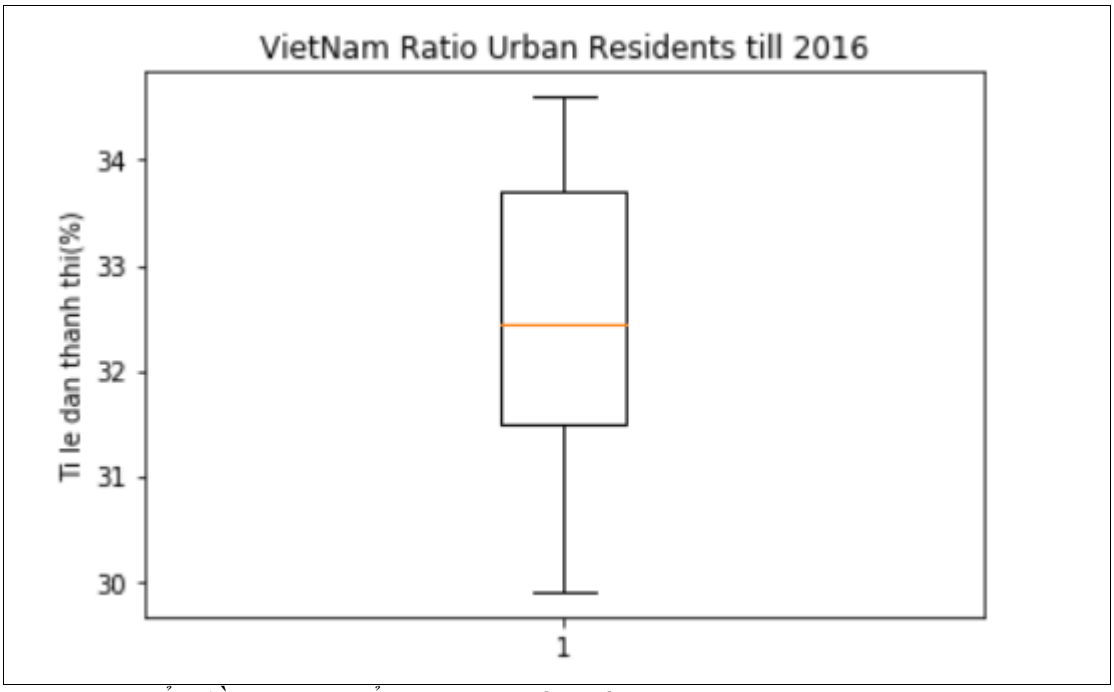
```
x = ThanhThi['Year']
y = ThanhThi['Ti le dan thanh thi(%)']
pl.xlabel('Year')
pl.ylabel('Ti le dan thanh thi(%)')
pl.title('VietNam Ratio Urban Residents till 2016')
pl.plot(x, y)
pl.show()
```



Biểu đồ đường thể hiện Tỷ lệ dân thành thị Việt Nam 2010 - 2016

- Biểu đồ Boxplot

```
1  pl.boxplot(y)
2  pl.ylabel('Ti le dan thanh thi(%)')
3  pl.title('VietNam Ratio Urban Residents till 2016')
```



Biểu đồ boxplot thể hiện Tỷ lệ dân thành thị Việt Nam 2010 – 2016

- Nhận xét: Thông qua các số liệu đã tính toán ở trên, và dữ liệu đã được hiển thị ở trên, ta thấy tỷ lệ dân thành thị ở Việt Nam còn thấp. Tuy nhiên đã tăng mạnh trong những năm qua, đặt biệt tăng vọt từ năm 2013.

III. Suy Diễn Thống Kê

- Suy diễn thống kê là phương pháp dùng để thống kê dữ liệu, kiểm định dữ liệu có đúng với giả thuyết hay không. Suy diễn thống kê gồm có 4 loại: So sánh trung bình của một tổng thể với một giá trị cụ thể (One – Sample Hypoothesis Test), (), Phân tích phương sai (Anova), kiểm định Chi -Square xét sự độc lập của hai nhóm tổng thể.

1. Kiểm Định Trung Bình Một Mẫu

a) Dữ liệu GDP Việt Nam

- Dữ liệu: Data_GDPVietNam.csv
- Code : One_Sample T-Test_GDPVietNam.ipynb
- Mô tả dữ liệu: dữ liệu cung cấp thu nhập bình quân đầu người trên năm (GDP) của Việt Nam từ 1994 - 2017 (đơn vị USD).
- Phát biểu bài toán, chúng tôi đưa giả giả thuyết như sau:
  - H0 : "Giá trị trung bình GDP của Việt Nam trong khoảng 1994 – 2017 bằng 2000(USD)"
  - H1: "Giá trị trung bình GDP của Việt Nam trong khoảng 1994 – 2017 khác 2000 (USD)"

- Đầu tiên import thư viện vào đọc giữ liệu:

```
1 import pandas as pd
2 from scipy import stats
```

- Đọc dữ liệu:

```
1 cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
1 GDP = pd.read_csv('Data_GDPVietNam.csv')
2 GDP.head()
```

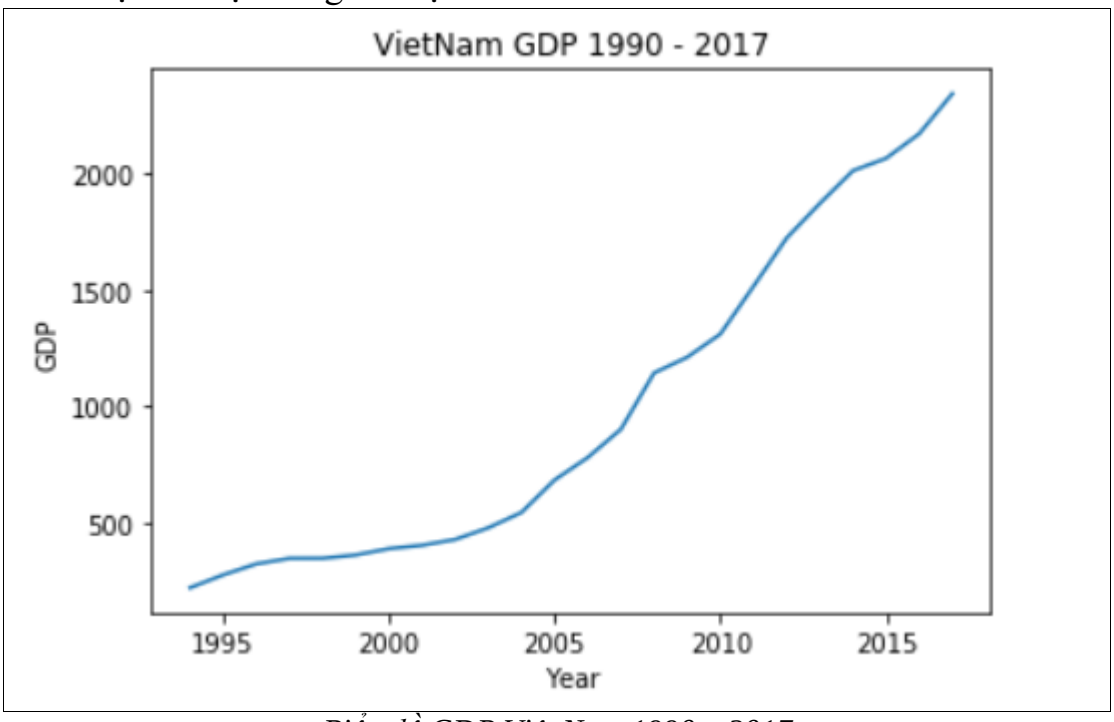
- Tiếp theo ta tính one-sample test:

```
one_sample_data = GDP['GDP(USD)']
df=one_sample_data.count()-1
one_sample = stats.ttest_1samp(one_sample_data,2000)
print one_sample
```

- Kết quả:

```
Ttest_1sampResult(statistic=-6.946022249267605, pvalue=4.426861272749156e-07)
```

- Kết luận: pvalue: 4.42e-06 < 0.05 , nên ta từ chối giả thuyết H0 . Từ đó ta kết luận thu nhập bình quân người trên năm của Việt Nam 1990 -2017 khác 2000(USD)
- Minh họa dữ liệu bằng đồ thị:



Biểu đồ GDP Việt Nam 1990 – 2017

- Nhận xét : dựa vào đồ thị ta thấy thu nhập bình quân của người Việt Nam còn khá thấp so với mặt bằng chung, vì Việt Nam là nước đang phát triển, nhưng GDP chưa cao. Thu nhập bình quân tăng mạnh thừ năm 2005 đến nay, cho thấy Việt Nam đang trên đà phát triển.

b) Dữ liệu Huy Chương Vàng Việt Nam

- Dữ liệu: Data_HuyChuongVang.csv
- Code : One_Sample T-Test_HuyChuongVangVietNam.ipynb

- Mô tả dữ liệu: dữ liệu cung cấp số huy chương vàng cấp thế giới mà Việt Nam đã đạt được trong 2002 – 2015.
- Phát biểu bài toán, chúng tôi đưa giả giả thuyết như sau:
    - H0 : "Giá trị trung bình số huy chương vàng Việt Nam đạt được trong khoảng 2002– 2015 bằng 40"
    - H1: "Giá trị trung bình số huy chương vàng Việt Nam đạt được trong khoảng 2002– 2015 khác 40"

- Đầu tiên import thư viện vào đọc giữ liệu:

```
1 import pandas as pd
2 from scipy import stats
```

- Đọc dữ liệu:

```
1 cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
1 HCV = pd.read_csv('Data_HuyChuongVang.csv')
2 HCV.head()
```

- Tiếp theo ta tính one-sample test:

```
1 one_sample_data = HCV['HuyChuongVang']
2 df=one_sample_data.count()-1
3 one_sample = stats.ttest_1samp(one_sample_data,40)
4 print one_sample
```

- Kết quả:

```
Ttest_1sampResult(statistic=-2.3008290751356757, pvalue=0.03859941073154601)
```

- Kết luận: pvalue: 0.03 < 0.05 , nên ta từ chối giả thuyết H0 . Từ đó ta kết luận giá trị trung bình số huy chương vàng cấp thế giới mà Việt Nam đạt được từ 2002 - 2015 khác 40 huy chương vàng.
- Minh họa dữ liệu bằng đồ thị:

Biểu đồ số huy chương vàng Thế giới Việt Năm 2002- 2015

- Nhận xét : dựa vào đồ thị số huy chương vàng thế giới đạt được của Việt Nam biến động. Và giá trị trung bình số huy chương thế giới của Việt Nam còn thấp.

2. Kiểm Định Trung Bình Hai Mẫu

a) Dữ Liệu Du Lịch Việt Nam

- Được thực hiện trên tập dữ liệu : Data_DuLich_TwoSample.xlsx
- Code: Two_Sample_Final-DuLich.ipynb
- Giả thuyết:
    - H0: Giá trị trung bình của số lượt khách quốc tế từ năm 2015 đến tháng 4/2018 đến tham quan Việt Nam bằng đường hàng không và đường biển **không có** sự chênh lệch nhau nhiều.
    - H1: Giá trị trung bình của số lượt khách quốc tế đến tham quan Việt Nam bằng đường hàng không và đường biển **có** sự chênh lệch nhau nhiều..
- Đầu tiên ta thêm thư viện và đọc file dữ liệu:

```
1  import pandas as pd
```

```
1  cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
1  data = pd.read_excel(r"Data_DuLich_TwoSample.xlsx", sheetname=0)
```

```
C:\Users\TranTung\Anaconda3\envs\py27\lib\site-packages\pandas\util\_d
is deprecated, use `sheet_name` instead
  return func(*args, **kwargs)
```

```
1  data.head()
```

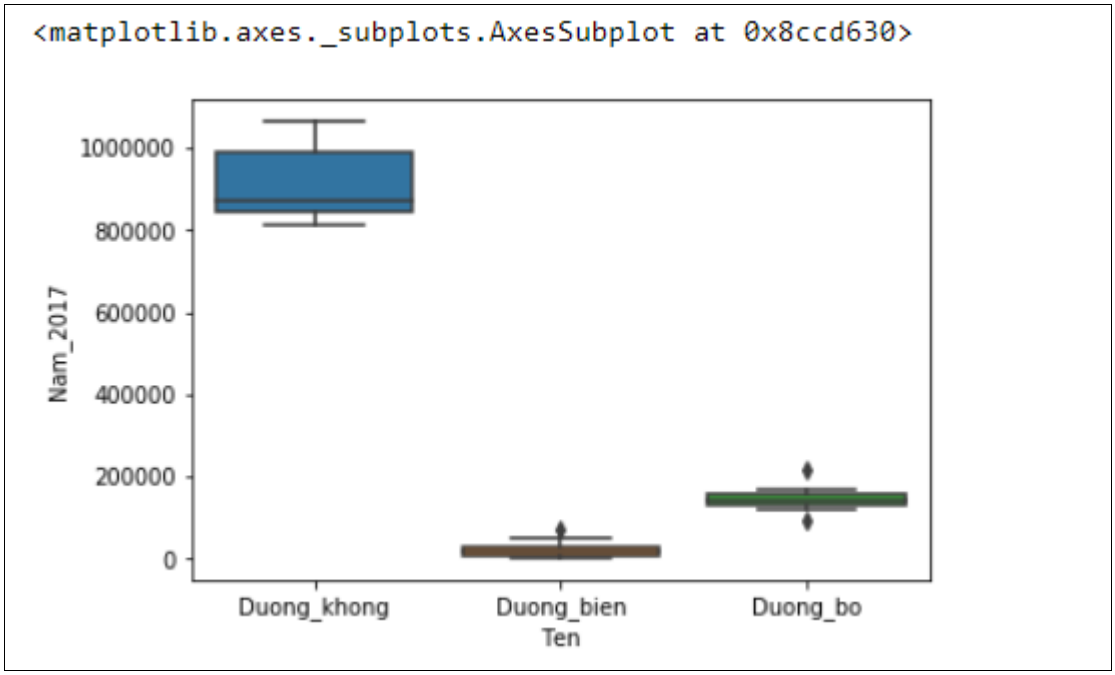
	Ten	Nam_2015	Nam_2016	Nam_2017	Nam_2018
0	Duong_khong	561883	659394	834975	1150969.0
1	Duong_khong	610834	667321	984013	1145961.0
2	Duong_khong	538554	659846	812594	1068785.0
3	Duong_khong	534507	661484	879864	1068792.0
4	Duong_khong	458758	643894	847525	NaN

- Tính two-sample:

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5  import scipy.stats as stats
6
7
8  DuongKhong_Sat = data[data['Ten'] == 'Duong_khong']['Nam_2017'].dropna()
9  DuongBo_Sat = data[data['Ten'] == 'Duong_bo']['Nam_2017'].dropna()
10 data_gender = data.groupby(['Ten'])
11 data_gender.boxplot(column=['Nam_2017'])
12
13 # paired 2 sample t-test
14 stats.ttest_ind(DuongKhong_Sat,DuongBo_Sat, equal_var=False)
15
```

- Kết quả:

```
Ttest_indResult(statistic=28.583367300867373, pvalue=1.2435305170186
```



- Kết luận: p-value: 1.243530517018681e-13 < 0.05 , nên từ chối chấp nhận giả thuyết H0. Giá trị trung bình của số lượt khách quốc tế đến tham quan Việt Nam bằng đường hàng không và đường biển có sự chênh lệch nhau nhiều.

3. Phân tích phương sai (Analysis of Variance)

a) Tỷ Lệ Hộ Nghèo theo khu vực

- Dữ liệu: Data_TyLeHoNgheo_Anova.csv
- Code: Anova_TyLeHoNgheo.ipynb
- Mô tả dữ liệu: dữ liệu cung cấp tỷ lệ hộ nghèo của 4 khu vực : Bắc trung bộ và duyên hải miền trung, tây nguyên, đông nam bộ, đồng bằng sông cửu long trong giai đoạn 1998 – 2016. Tỷ lệ hộ nghèo được tính theo thu nhập bình quân 1 người 1 tháng của hộ gia đình theo chuẩn nghèo của Chính phủ giai đoạn 2011-2016 được cập nhật theo chỉ số giá tiêu dùng như sau:2010: 400 nghìn đồng đối với khu vực nông thôn và 500 nghìn đồng đối với khu vực thành thị.; 2012: 530 nghìn đồng đối với khu vực nông thôn và 660 nghìn đồng đối với khu vực thành thị.;2013: 570 nghìn đồng đối với khu vực nông thôn và 710 nghìn đồng đối với khu vực thành thị.; 2014: 605 nghìn đồng đối với khu vực nông thôn và 750 nghìn đồng đối với khu vực thành thị.; 2015: 615 nghìn đồng đối với khu vực nông thôn và 760 nghìn đồng đối với khu vực thành thị.; 2016: 630 nghìn đồng đối với khu vực nông thôn và 780 nghìn đồng đối với khu vực thành thị.
- Phát biểu bài toán: tiến hành tính toán và xem xét có sự khác biệt về tỷ lệ hộ nghèo giữa khác khu vực, và đưa ra sự khác biệt như thế nào của mỗi nhóm. Đưa ra giả thuyết:

- H0 : Không có sự khác biệt về tỷ lệ hộ nghèo trên từng nhóm khu vực.
- H1: Có sự khác biệt về về tỷ lệ hộ nghèo trên từng nhóm khu vực.

- Tiến hành import các thư viện và thực hiện các câu lệnh theo hình dưới đây:

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

- Đọc dữ liệu:

```
cd C:\Users\TranTung\Desktop\DuLieu
```

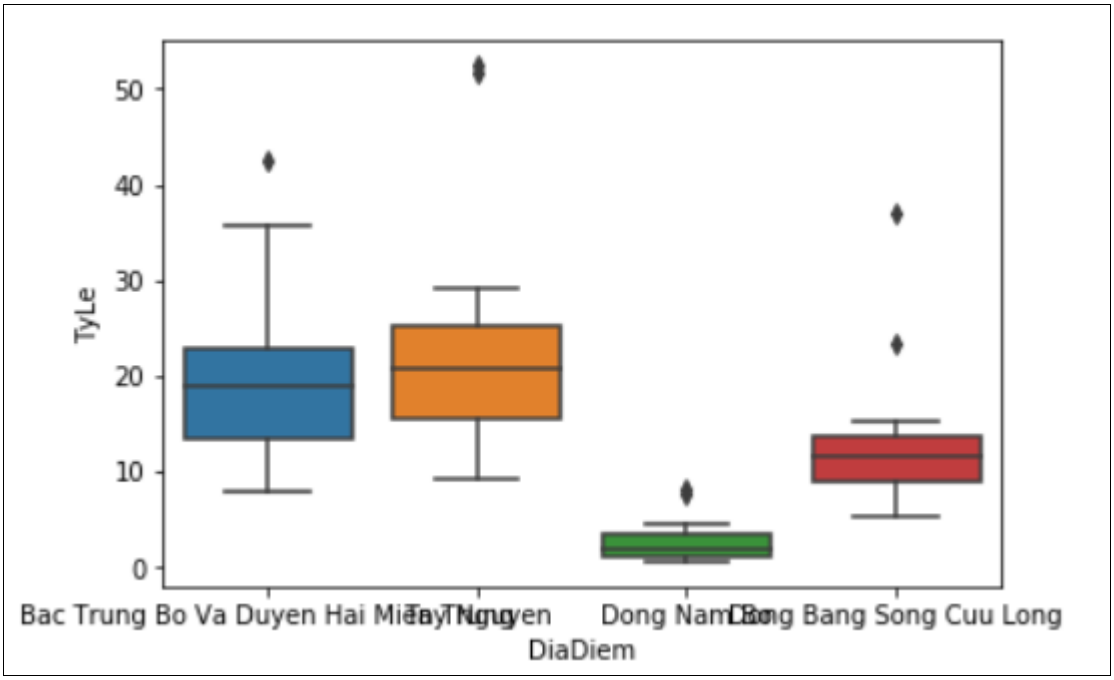
```
C:\Users\TranTung\Desktop\DuLieu
```

```
HoNgheo = pd.read_csv('Data_TyLeHoNgheo_Anova.csv')
```

- Minh họa dữ liệu:

```
import matplotlib.pyplot as pl
import seaborn as sns
```

```
ax = sns.boxplot(x = 'DiaDiem', y= 'TyLe', data = HoNgheo)
```



- Phân tích phương sai Anova:

```
cw_lm=ols('TyLe ~ DiaDiem ', data=HoNgheo).fit()
print(sm.stats.anova_lm(cw_lm, typ=2))
```

- Kết quả:

	sum_sq	df	F	PR(>F)
DiaDiem	3108.810000	3.0	10.641367	0.000022
Residual	4284.776667	44.0	NaN	NaN

- Kết luận : ta thấy pValue = 0.000022 < 0.05 (mức ý nghĩa), nên ta bác bỏ giả thuyết H0. Suy ra, có sự khác biệt về tỷ lệ hộ nghèo giữa các nhóm vùng với nhau.

- Tiến hành kiểm tra sự khác biệt của mỗi nhóm với nhau:

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison
```

```
mc = MultiComparison(HoNgheo['TyLe'], HoNgheo['DiaDiem'])
result = mc.tukeyhsd()

print(result)
print(mc.groupsunique)
```

- Kết quả:

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================================================================
              group1                         group2          meandiff  lower    upper  reject
---------------------------------------------------------------------------------------------
Bac Trung Bo Va Duyen Hai Mien Trung Dong Bang Song Cuu Long   -6.7   -17.4573  4.0573 False
Bac Trung Bo Va Duyen Hai Mien Trung       Dong Nam Bo        -17.4   -28.1573 -6.6427  True
Bac Trung Bo Va Duyen Hai Mien Trung        Tay Nguyen         3.8    -6.9573  14.5573 False
      Dong Bang Song Cuu Long              Dong Nam Bo        -10.7   -21.4573  0.0573 False
      Dong Bang Song Cuu Long               Tay Nguyen         10.5   -0.2573  21.2573 False
            Dong Nam Bo                     Tay Nguyen         21.2   10.4427  31.9573  True
---------------------------------------------------------------------------------------------
['Bac Trung Bo Va Duyen Hai Mien Trung' 'Dong Bang Song Cuu Long'
 'Dong Nam Bo' 'Tay Nguyen']
```

- Kết luận :
  - Ta thấy quan hệ giữa nhóm “Bac Trung Bo Va Duyen Hai Mien Trung ” với nhóm “Dong Nam Bo” có khoảng tin cậy lower và upper

đề nhỏ hơn 0, nên ta có thể kết luận sự khác biệt của hai nhóm này có ý nghĩa thống kê.

- Tương tự như "Dong Nam Bo " và "Tay Nguyen" có khoảng tin cậy lower và upper đều lớn hơn 0 nên sự khác biệt của hai nhóm này có ý nghĩa thống kê.
- Còn các nhóm còn lại không có khoảng tin cậy lower và upper đều nhỏ hơn 0, hoặc đều lớn hơn 0 nên không thể kết luận chúng khác biệt nhau.

b) Thu Nhập Bình Quân Theo Ngành Kinh Tế

- Dữ liệu: Data_ThuNhapTheoNganh_Anovaa.csv
- Code: Anova_ThuNhapTheoNganh.ipynb
- Mô tả dữ liệu: dữ liệu cung cấp thu nhập bình quân của lao động làm công ăn lương trong khu vực nhà nước theo 4 khu vực : thông tin và truyền thông, giáo dục và đào tạo, y tế, vui chơi giải trí 2005 - 2015. Đưa ra giả thuyết:
  - H0 : Không có sự khác biệt về thu nhập bình quân của lao động làm công ăn lươn của 4 ngành.
  - H1: Có sự khác biệt về thu nhập bình quân của lao động làm công ăn lươn của 4 ngành.
- Tiến hành import các thư viện và thực hiện các câu lệnh theo hình dưới đây:

```
1 import pandas as pd
2 import statsmodels.api as sm
3 from statsmodels.formula.api import ols
```

- Đọc dữ liệu:

```
1 cd C:\Users\TranTung\Desktop\DuLieu
```

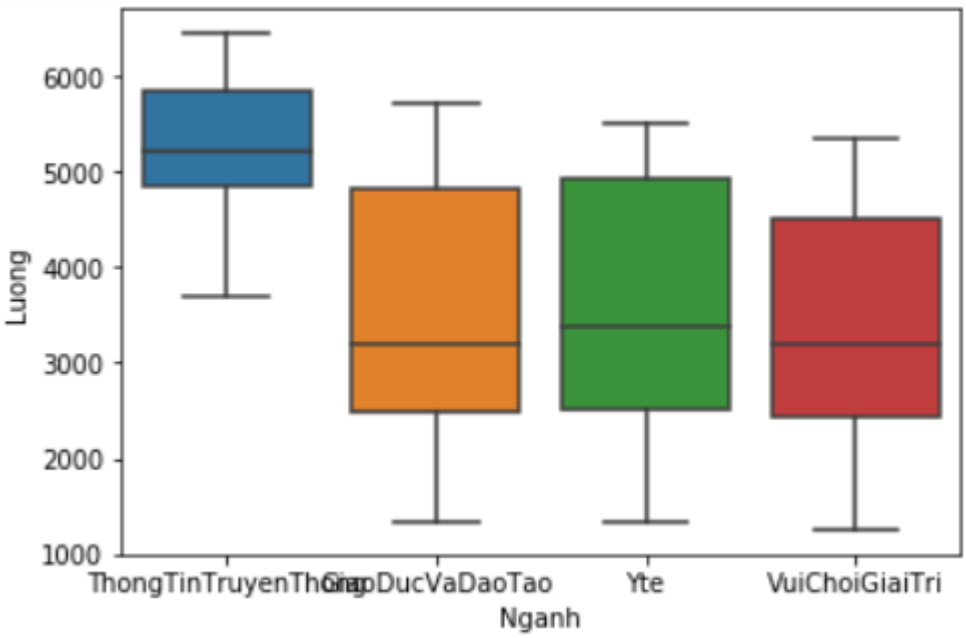
```
C:\Users\TranTung\Desktop\DuLieu
```

```
1 ThuNhap = pd.read_csv('Data_ThuNhapTheoNganh_Anovaa.csv')
```

- Minh họa dữ liệu:

```
import matplotlib.pyplot as pl
import seaborn as sns
```

```
ax = sns.boxplot(x = 'Nganh', y= 'Luong', data = ThuNhap)
```



- Phân tích phương sai Anova:

```
cw_lm=ols('Luong ~ Nganh ', data=ThuNhap).fit()
print(sm.stats.anova_lm(cw_lm, typ=2))
```

- Kết quả:

	sum_sq	df	F	PR(>F)
Nganh	2.279763e+07	3.0	4.290251	0.010937
Residual	6.376587e+07	36.0	NaN	NaN

- Kết luận : ta thấy pValue = 0.01 < 0.05 (mức ý nghĩa), nên ta bác bỏ giả thuyết H0. Suy ra, có sự khác biệt về bình quân thu nhập của các lao động làm công ăn lương nhà nước theo 4 ngành ở trên.

- Tiến hành kiểm tra sự khác biệt của mỗi nhóm với nhau:

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison
```

```
mc = MultiComparison(ThuNhap['Luong'], ThuNhap['Nganh'])
result = mc.tukeyhsd()

print(result)
print(mc.groupsunique)
```

- Kết quả:

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================================================
       group1              group2       meandiff   lower      upper   reject
-----------------------------------------------------------------------------
  GiaoDucVaDaoTao   ThongTinTruyenThong 1709.83   106.8236  3312.8364  True
  GiaoDucVaDaoTao      VuiChoiGiaiTri   -134.25  -1737.2564 1468.7564 False
  GiaoDucVaDaoTao           Yte          55.15   -1547.8564 1658.1564 False
ThongTinTruyenThong    VuiChoiGiaiTri   -1844.08 -3447.0864 -241.0736  True
ThongTinTruyenThong         Yte         -1654.68 -3257.6864  -51.6736  True
   VuiChoiGiaiTri           Yte          189.4   -1413.6064 1792.4064 False
-----------------------------------------------------------------------------
['GiaoDucVaDaoTao' 'ThongTinTruyenThong' 'VuiChoiGiaiTri' 'Yte']
```

- Kết luận :
  - Ta thấy quan hệ giữa nhóm "GiaoDucVaDaoTao" với nhóm "ThongTinTruyenThong" có khoảng tin cậy lower và upper đều lớn hơn 0, nên ta có thể kết luận sự khác biệt của hai nhóm này có ý nghĩa thống kê.
  - Tương tự như "ThongTinTruyenThong" và "VuiChoiGiaiTri" có khoảng tin cậy lower và upper đều nhỏ hơn 0 nên sự khác biệt của hai nhóm này có ý nghĩa thống kê.
  - Tương tự như "ThongTinTruyenThong" và "Yte" có khoảng tin cậy lower và upper đều nhỏ hơn 0 nên sự khác biệt của hai nhóm này có ý nghĩa thống kê.
  - Còn các nhóm còn lại không có khoảng tin cậy lower và upper đều nhỏ hơn 0, hoặc đều lớn hơn 0 nên không thể kết luận chúng khác biệt nhau.

4. Kiểm Định Chi -Square

a) Bình Quân Thu Nhập Theo Địa Phương và Nguồn Thu

- Dữ Liệu: Data_ThuNhapThangNam2016_ChiSquare.csv

- Code: Chi-Quare_ThuNhapThang.ipynb
- Mô tả dữ liệu: dữ liệu cung cấp hai nhóm tổng thể mà ta cần xét hai nhóm có độc lập với nhau hay không. Trong dữ liệu này có hai nhóm đó là địa phương và nguồn thu.
- Phát biểu bài toán: xét sự độc lập của hai nhóm địa phương và nguồn thu. Xét xem chúng độc lập hay phụ thuộc nhau. Ta có giả thuyết:
    - H0 : Hai nhóm địa phương và nguồn thu độc lập với nhau.
    - H1: Hai nhóm địa phương và nguồn thu phụ thuộc với nhau.
- Chúng ta tiến hành thêm thư viện :

```
1 import pandas as pd
2 import numpy as np
```

- Chuyển dữ liệu từ file excel dưới dạng ma trận trong Python, việc này được thực hiện thủ công:

	Thu từ tiền lương, tiền công	Thu từ nông, lâm nghiệp, thủy sản	Thu phi nông, lâm nghiệp, thủy sản
Hà Nội	3120	302	1060
Đà Nẵng	2382	88	1301
TP. Hồ Chí Minh	3249	61	1521

- Thực hiện Chi Square:

```
1 a = np.array([[3120, 302, 1060], [2382, 88, 1301],[3249, 61, 1521]])
2
3 from scipy.stats import chi2_contingency
4 chi2_contingency(a)
```

- Kết quả:

```
(323.8983406038088,
 7.558248082526123e-69,
 4L,
 array([[2997.70574748,  154.49266279, 1329.80158973],
        [2522.16608071,  129.98479058, 1118.84912871],
        [3231.12817181,  166.52254662, 1433.34928157]]))
```

- Kết luận : giá trị Chi-Square : 323.89, pvalue = 7.55e-69 < 0.05 (mức ý nghĩa). Cho nên ta bác bỏ giả thuyết H0. Và điều đó cũng cho biết rằng giá trị nguồn thu (thu nhập) của người dân mỗi vùng khác nhau sẽ phụ thuộc theo từng vùng mà họ sinh sống.

b) Tỷ Lệ Thất Nghiệp Ở Thành Thị và Nông Thôn Theo Vùng

- Dữ Liệu: Data_TyLeThatNghiepTTNT_ChiSquare.csv
- Code: Chi-Quare_TyLeThatNghiepTheoVung.ipynb
- Mô tả dữ liệu: dữ liệu cung cấp hai nhóm tổng thể mà ta cần xét hai nhóm có độc lập với nhau hay không. Trong dữ liệu này có hai nhóm đó là Vùng và Loại Đô Thị.
- Phát biểu bài toán: xét sự độc lập của hai nhóm vùng và loại đô thị. Xét xem chúng độc lập hay phụ thuộc nhau. Ta có giả thuyết:
  - H0 : Hai nhóm vùng và loại đô thị độc lập với nhau.
  - H1: Hai nhóm vùng và loại đô thị phụ thuộc với nhau.
- Chúng ta tiến hành thêm thư viện :

```
1  import pandas as pd
2  import numpy as np
```

- Chuyển dữ liệu từ file excel dưới dạng ma trận trong Python, việc này được thực hiện thủ công:

	Thành thị	Nông thôn
Đồng bằng sông Hồng	3.42	1.94
Trung du và miền núi phía Bắc	3.11	0.72
Bắc Trung Bộ và duyên hải miền Trung	4.51	2.05
Tây Nguyên	2.27	0.57
Đông Nam Bộ	3.05	2.17
Đồng bằng sông Cửu Long	3.22	2.63

- Thực hiện Chi Square:

```
1  a = np.array([[3.42, 1.94], [3.11, 0.72],[4.51, 2.05]
2                ,[2.27, 0.57],[3.05, 2.17],[3.22, 2.63]])
3
4  from scipy.stats import chi2_contingency
5  chi2_contingency(a)
```

- Kết quả:

```
(1.1201239066801532, 0.952306993137951, 5L, array([[3.53839514, 1.82160486],
       [2.52836817, 1.30163183],
       [4.33057316, 2.22942684],
       [1.87482131, 0.96517869],
       [3.44597438, 1.77402562],
       [3.86186784, 1.98813216]]))
```

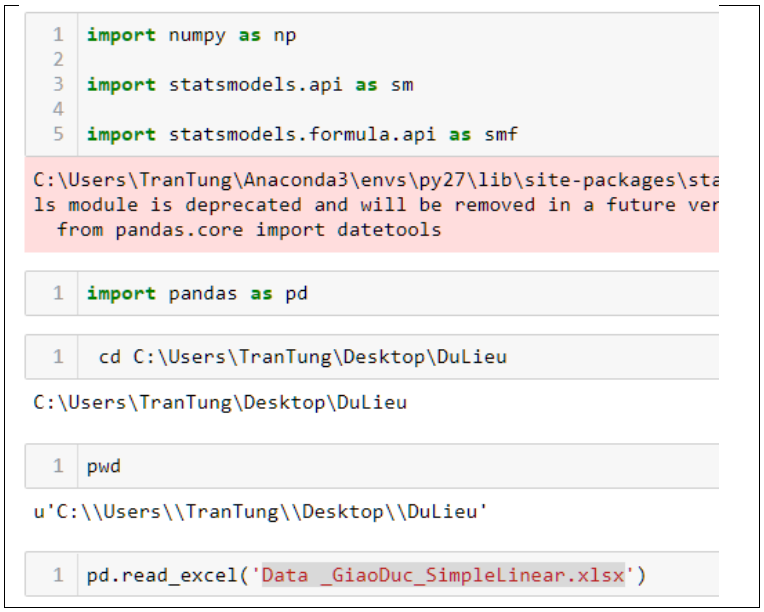
- Kết luận : giá trị Chi-Square : 1.12, pvalue = 0.95 > 0.05 (mức ý nghĩa). Cho nên ta chấp nhận giả thuyết H0. Và điều đó cũng cho biết rằng tỷ lệ thất nghiệp theo loại đô thị sẽ độc lập với từng vùng miền.

IV. Phân Tích Hồi Quy Tuyến Tính

1. Hồi Quy Tuyến Tính Đơn

a) Dữ Liệu Giáo Dục Việt Nam

- Dữ liệu: Data _GiaoDuc_SimpleLinear.xlsx
- Code: Simple Linear Regression _ GiaoDuc.ipynb
- Giả thuyết:
  - H0 = "Phương trình tìm được không có ý nghĩa, trình độ giảng dạy của giáo viên thường không quyết định đầu ra/số lượng tốt nghiệp của sinh viên."
  - H1 = "Phương trình tìm được có ý nghĩa, trình độ giảng dạy của giáo viên thường quyết định đầu ra/số lượng tốt nghiệp của sinh viên."
- Tiến hành thêm thư viện vào đọc file dữ liệu :



```
import numpy as np

import statsmodels.api as sm

import statsmodels.formula.api as smf
```

```
C:\Users\TranTung\Anaconda3\envs\py27\lib\site-packages\sta
ls module is deprecated and will be removed in a future ver
  from pandas.core import datetools
```

```
import pandas as pd
```

```
 cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
pwd
```

```
u'C:\\Users\\TranTung\\Desktop\\DuLieu'
```

```
pd.read_excel('Data _GiaoDuc_SimpleLinear.xlsx')
```

```
1  pd.read_excel('Data _GiaoDuc_SimpleLinear.xlsx')
```

	Tên	TrenDaiHoc(NghinNguoi)	SoSinhVienTotNghiep(NghinNguoi)
0	2005	23.86	195.0
1	2006	24.33	216.5
2	2007	26.59	215.2
3	2008	30.28	208.7
4	2009	31.37	223.9
5	2010	38.30	278.3
6	2011	45.51	334.5
7	2012	48.56	357.2
8	2013	54.89	350.6
9	2014	59.98	377.9

- Tiến hành các phép tính tính hồi quy tuyến tính, và tìm ra phương trình có nghĩa:

```
1  import statsmodels.api as sm # import statsmodels
2
3   ## X usually means our input variables (or independent variables)
4  X = x["SoSinhVienTotNghiep(NghinNguoi)"]
5  ## Y usually means our output/dependent variable
6  y = x["TrenDaiHoc(NghinNguoi)"]
7  X = sm.add_constant(X) ## let's add an intercept (beta_0) to our mode
8
9  # Note the difference in argument order
10 model = sm.OLS(y, X).fit() ## sm.OLS(output, input)
11 predictions = model.predict(X)
12
13 # Print out the statistics
14 model.summary()
```

- Kết quả:

OLS Regression Results

Dep. Variable:	TrenDaiHoc(NghinNguoi)	**R-squared:**	0.946
Model:	OLS	**Adj. R-squared:**	0.939
Method:	Least Squares	**F-statistic:**	138.9
Date:	Fri, 06 Jul 2018	**Prob (F-statistic):**	2.46e-06
Time:	14:01:45	**Log-Likelihood:**	-24.874
No. Observations:	10	**AIC:**	53.75
Df Residuals:	8	**BIC:**	54.35
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-10.4141	4.265	-2.442	0.040	-20.250	-0.578
SoSinhVienTotNghiep(NghinNguoi)	0.1769	0.015	11.785	0.000	0.142	0.211

Omnibus:	2.660	**Durbin-Watson:**	1.360
Prob(Omnibus):	0.264	**Jarque-Bera (JB):**	0.923
Skew:	-0.041	**Prob(JB):**	0.630
Kurtosis:	1.514	**Cond. No.**	1.18e+03

- Kết luận:
  - Phương trình tìm được là: Y = -10.4141+ 0.1769 b1
  - Ta thấy p-value: 2.46e-06< 0.05 = > Từ chối H0
  - Kết luận: Phương trình tìm được và trình độ giảng dạy của giáo viên thường quyết định đầu ra/số lượng tốt nghiệp của sinh viên.

b) Dữ liệu Diện Tích Sản Xuất Nông Nghiệp

- Tập dữ liệu: Data_DLSX_SimpleLinear.csv
- Code: Linear Regression_DLSX.ipynb
- Giả thuyết:
  - H0 = "Phương trình tìm được không có ý nghĩa, và Tổng sản lượng lúa thu được không phụ thuộc vào Tổng diện tích đất trồng."
  - H1 = "Phương trình tìm được có ý nghĩa, Tổng sản lượng lúa thu được không phụ thuộc vào Tổng diện tích đất trồng."
- Tiến hành thêm thư viện vào đọc file dữ liệu :

```
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
import pandas as pd
```

```
cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
pwd
```

```
u'C:\\Users\\TranTung\\Desktop\\DuLieu'
```

```
pd.read_csv('Data_DLSX_SimpleLinear.csv')
```

	Nam	Tong Dien Tich (NghinHa)	Dien Tich Lua (Nghin ha)	Dien Tich Ng (Nghn ha)	Tong San Luong(Nghin tan)	San Luong Lua (Nghin Tan)	San Luong Ngo (Nghin Tan)
0	1990	6476.9	6042.8	431.8	19897.7	19225.1	671.0
1	1991	6752.7	6302.8	447.6	20295.8	19621.9	672.0
2	1992	6956.3	6475.3	478.0	22342.8	21590.4	747.9
3	1993	7058.3	6559.4	496.5	23720.5	22836.5	882.2
4	1994	7135.7	6598.6	534.6	24673.7	23528.2	1143.9
5	1995	7324.3	6765.6	556.8	26142.5	24963.7	1177.2

- Tiến hành các phép tính tính hồi quy tuyến tính, và tìm ra phương trình có nghĩa:

```
import statsmodels.api as sm # import statsmodels

X = x["Tong Dien Tich (NghinHa)"] ## X usually means our input variables (or independent variables)
y = x["Tong San Luong(Nghin tan)"] ## Y usually means our output/dependent variable
X = sm.add_constant(X) ## let's add an intercept (beta_0) to our model

# Note the difference in argument order
model = sm.OLS(y, X).fit() ## sm.OLS(output, input)
predictions = model.predict(X)

# Print out the statistics
model.summary()
```

-Kết Quả:

OLS Regression Results

Dep. Variable:	Tong San Luong(Nghin tan)	**R-squared:**	0.927
Model:	OLS	**Adj. R-squared:**	0.924
Method:	Least Squares	**F-statistic:**	317.2
Date:	Mon, 21 May 2018	**Prob (F-statistic):**	1.02e-15
Time:	00:09:09	**Log-Likelihood:**	-250.59
No. Observations:	27	**AIC:**	505.2
Df Residuals:	25	**BIC:**	507.8
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-6.676e+04	5823.293	-11.464	0.000	-7.88e+04	-5.48e+04
Tong Dien Tich (NghinHa)	12.6977	0.713	17.810	0.000	11.229	14.166

Omnibus:	3.072	**Durbin-Watson:**	0.208
Prob(Omnibus):	0.215	**Jarque-Bera (JB):**	2.490
Skew:	-0.735	**Prob(JB):**	0.288
Kurtosis:	2.766	**Cond. No.**	9.16e+04

- Kết luận:
    - Phương trình tìm được là: Y = -6.676e+04 + 12.6977 b1
    - Ta thấy p-value: 1.02e-15< 0.05 = > Từ chối H0
    - Kết luận: Phương trình tìm được và Tổng sản lượng lúa thu được phụ thuộc vào Tổng diện tích đất trồng

2. Hồi Quy Tuyến Tính Bội

b) Dữ Liệu Thu Nhập

- Dữ liệu: Data_DienTich_MultipleLinear.xlsx
- Code: Multiple Linear Regression-DienTichNha. .ipynb
- Giả thuyết:
    - H0 = "Phương trình tìm được không có ý nghĩa, và **Diện tích nhà ở** không phụ thuộc vào **Thu nhập bình quân đầu người,năng suất lao động, giới tính"**
    - H1 = "Phương trình tìm được có ý nghĩa, và **Diện tích nhà ở** phụ thuộc vào **Thu nhập bình quân đầu người,năng suất lao động, giới tính"**
- Thêm thư viện và đọc file dữ liệu:

```
import numpy as np

import statsmodels.api as sm

import statsmodels.formula.api as smf
import pandas as pd
```

- Đọc dữ liệu:

```
cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
pd.read_excel('Data_DienTich_MultipleLinear.xlsx',sheetname=0)
```

```
1  pd.read_excel('Data_DienTich_MultipleLinear.xlsx',sheetname=0)
```

```
C:\Users\TranTung\Anaconda3\envs\py27\lib\site-packages\pandas\util\_decorators.py:118: FutureWarning: Th
is deprecated, use `sheet_name` instead
  return func(*args, **kwargs)
```

	Tên	DienTich(m2/Nguoi)	NangSuatLaoDong(NghinVND/Nguoi/thang)	ThuNhapBinhQuanDauNguoi(TrieuVND/Nguoi)	Nam	Nu
0	2005	0.0	0	21.40	69.1	73.1
1	2006	16.9	1058	23.35	69.5	73.5
2	2007	0.0	0	25.30	69.8	73.7
3	2008	18.7	1605	32.00	72.5	76.4
4	2009	0.0	0	34.70	70.2	75.6
5	2010	20.7	2130	44.00	70.3	75.7
6	2011	0.0	0	55.20	70.4	75.8
7	2012	21.5	2989	63.10	70.4	75.8
8	2013	0.0	0	68.70	70.5	75.8
9	2014	24.0	3968	74.70	70.6	76.0

- Tiến hành tính Multiple Linear Regression

```
1  import statsmodels.api as sm # import statsmodels
2  ## Y usually means our output/dependent variable
3  y = x["DienTich(m2/Nguoi)"]
4  ## X usually means our input variables (or independent variables)
5  X = x[["NangSuatLaoDong(NghinVND/Nguoi/thang)","ThuNhapBinhQuanDauNguoi(TrieuVND/Nguoi)","Nam","Nu"] ]
6  X = sm.add_constant(X) ## let's add an intercept (beta_0) to our model
7
8  # Note the difference in argument order
9  model = sm.OLS(y, X).fit() ## sm.OLS(output, input)
10 predictions = model.predict(X)
11 # Print out the statistics
12 print model.summary()
```

- Kết quả:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     DienTich(m2/Nguoi)   R-squared:                       0.931
Model:                            OLS   Adj. R-squared:                  0.897
Method:                 Least Squares   F-statistic:                     27.09
Date:                Fri, 06 Jul 2018   Prob (F-statistic):           0.000106
Time:                        13:38:56   Log-Likelihood:                -31.827
No. Observations:                  13   AIC:                             73.65
Df Residuals:                       8   BIC:                             76.48
Df Model:                           4
Covariance Type:            nonrobust
```

```
                                        coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
const                                -94.9766    102.474     -0.927      0.381    -331.283     141.330
NangSuatLaoDong(NghinVND/Nguoi/thang)    0.0070      0.001      9.120      0.000       0.005       0.009
ThuNhapBinhQuanDauNguoi(TrieuVND/Nguoi)   -0.0675      0.069     -0.985      0.354      -0.225       0.091
Nam                                   3.3333      2.074      1.607      0.147      -1.449       8.115
Nu                                   -1.7891      1.914     -0.935      0.377      -6.202       2.624
==============================================================================
Omnibus:                        1.835   Durbin-Watson:                   2.981
Prob(Omnibus):                  0.399   Jarque-Bera (JB):                0.953
Skew:                           0.657   Prob(JB):                        0.621
Kurtosis:                       2.821   Cond. No.                     2.07e+05
```

- **Kết luận:**
  - Phương trình tìm được là: Y = -94.9766 + 0.0070*b1 – 0.0675*b2 +3.3333* b3 – 1.7891*b4
  - Ta thấy p-value: 0.000106 < 0.05 = > Từ chối H0
  - Kết luận: Phương trình tìm được có ý nghĩa, và **Diện tích nhà ở** phụ thuộc vào **Thu nhập bình quân đầu người,năng suất lao động, giới tính**

V. Kỹ Thuật Dự Báo

1. Dự báo với mô hình ARIMA

a) Chỉ số đô la

- Trong phần này sẽ tiến hành trên tập dữ liệu đơn giản : Data_ChiSoDoLa.csv
- Code: Forecast_Arima_ChiSoDoLa.ipynb
- Đầu tiên sẽ tiến hành import các thư viện cần thiết:

```
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
import matplotlib.pylab as plt
from matplotlib import pyplot
```

- Đọc file dữ liệu:

```
1 cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
1 import pandas as pd
2 pd.read_csv('Data_ChiSoDoLa.csv')
```

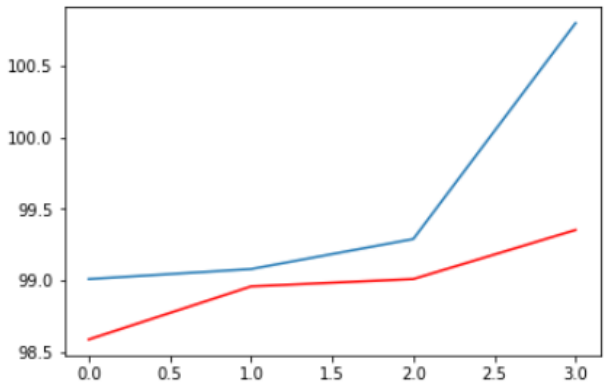
- Dựa vào giá trị lịch sử trong tập dữ liệu, ta khởi tạo mô hình.Xét dự báo cho giai đoạn giá mua vàng trong tương lai là tháng 1 của năm 2018

```
history = [x for x in train.astype(float)]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(3,1,0))
    model_fit = model.fit(disp=0)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))

error = mean_squared_error(test, predictions)
print('Test MSE: %.3f' % error)
print(model_fit.summary())
# plot
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
residuals = pd.DataFrame(model_fit.resid)
print(residuals.describe())
```

-Và chúng ta sẽ được kết quả dự báo tháng 1 của năm 2018

```
predicted=98.587910, expected=99.010000
predicted=98.958626, expected=99.080000
predicted=99.010156, expected=99.290000
predicted=99.353424, expected=100.800000
Test MSE: 0.591
                             ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                   10
Model:                 ARIMA(3, 1, 0)   Log Likelihood                1.086
Method:                       css-mle   S.D. of innovations              0.208
Date:                Tue, 10 Jul 2018   AIC                              7.828
Time:                        19:13:16   BIC                              9.341
Sample:                             1   HQIC                             6.168

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1272      0.146     -0.871      0.417      -0.413       0.159
ar.L1.D.y      0.6794      0.383      1.774      0.126      -0.071       1.430
ar.L2.D.y      0.1153      0.458      0.252      0.809      -0.782       1.012
ar.L3.D.y     -0.2955      0.399     -0.741      0.487      -1.077       0.486
```

```
                                    Roots
================================================================================
                Real          Imaginary           Modulus         Frequency
--------------------------------------------------------------------------------
AR.1          -1.8469           -0.0000j            1.8469           -0.5000
AR.2           1.1186           -0.7624j            1.3537           -0.0952
AR.3           1.1186           +0.7624j            1.3537            0.0952
--------------------------------------------------------------------------------
```



-Từ kết quả ta thấy chỉ số đô la trong tháng 1 của năm 2018 sẽ tăng trưởng với tốc độ chậm lại so với năm 2017 và chỉ số đô la sẽ giảm đi so với năm 2017

b) Dữ Liệu Việt Nam

- Trong phần này sẽ tiến hành trên tập dữ liệu đơn giản : Data_GiaVangSJC.csv
- Code: Forecast_Arima_GiaVangSJC.ipynb
- Đầu tiên sẽ tiến hành import các thư viện cần thiết:

```
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
import matplotlib.pylab as plt
from matplotlib import pyplot
```

- Đọc file dữ liệu: (*GiaVangSJC*= pd.read_csv("/path")):

```
cd C:\Users\TranTung\Desktop\DuLieu
```

```
C:\Users\TranTung\Desktop\DuLieu
```

```
import pandas as pd
pd.read_csv('Data_GiaVangSJC.csv')
```

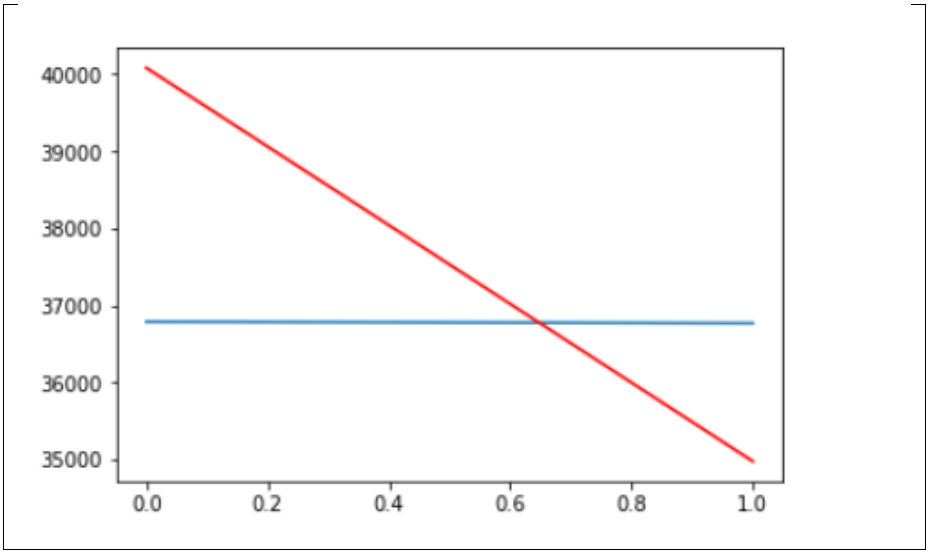
	Thang	GiaMua(NgayCuoiThang)(NghinVND/Chi)	GiaBan(NgayCuoiThang)(NghinVND/Chi)
0	7/1/2018	55400	55480
1	7/2/2018	43350	46240
2	7/3/2018	44750	45640
3	7/4/2018	44790	44300
4	7/5/2018	36640	36790
5	7/6/2018	36570	36770

- Dựa vào giá trị lịch sử trong tập dữ liệu, ta khởi tạo mô hình.Xét dự báo cho giai đoạn giá mua vàng trong tương lai là ngày 7/7/2018 của năm 2018

```
history = [x for x in train.astype(float)]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(1,1,0))
    model_fit = model.fit(disp=0)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))

error = mean_squared_error(test, predictions)
print('Test MSE: %.3f' % error)
print(model_fit.summary())
# plot
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
residuals = pd.DataFrame(model_fit.resid)
print(residuals.describe())
```

-Bây giờ mô hình của chúng ta được khởi tạo, chúng tôi có thể đánh giá nó. Và chúng ta sẽ được kết quả dự báo của ngày 7/7/2018

```
predicted=40083.298539, expected=36790.000000
predicted=34977.667823, expected=36770.000000
Test MSE: 7029134.951
                             ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                    4
Model:                 ARIMA(1, 1, 0)   Log Likelihood                -38.283
Method:                       css-mle   S.D. of innovations           3309.008
Date:                Tue, 10 Jul 2018   AIC                             82.565
Time:                        09:08:27   BIC                             80.724
Sample:                             1   HQIC                            78.525

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3861.1409   1334.646     -2.893      0.102   -6476.998   -1245.284
ar.L1.D.y     -0.5615      0.569     -0.987      0.428      -1.677       0.554
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -1.7810           +0.0000j            1.7810            0.5000
-----------------------------------------------------------------------------
```



- Từ kết quả ta thấy giá vàng trong ngày 7/7 của năm 2018 sẽ giảm với tốc độ chậm lại so với năm 2017 và giá vàng sẽ giảm đi so với ngày 6/7/2018.

VI. Tài Liệu Tham Khảo

- Giáo trình môn Phân Tích Dữ Liệu Kinh Doanh.
- Thư viện hỗ trợ trong Python : https://docs.python.org/3/index.html