DATA ANALYSIS OF IMMIGRANTS RECEIVED SINCE 1980

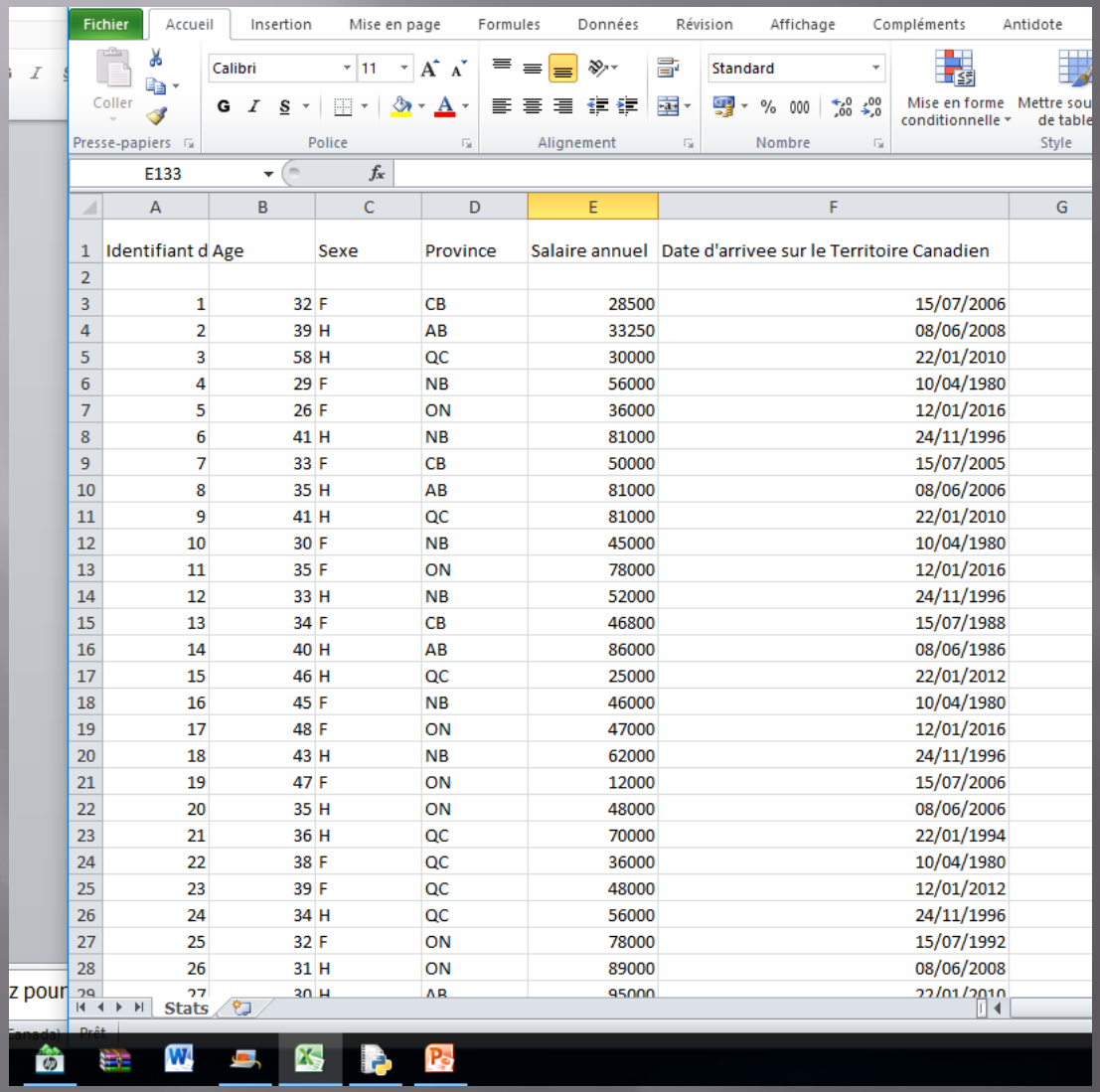
Analysis performed in Python Programming Language

By Joel Sandé

Warning :

- Numbers in this presentation are completely FALSE.
- I just wanted to highlight, from a practical case, the power of analysis of the Python language and its ability to generate results in visual form.
- I will leave a link to download the csv file from where the data are drawn. The code will be available upon request from sandejoel@yahoo.ca (Free if you are a student taking my course, or a government representative).

Stats.csv



	A	B	C	D	E	F	G
1	Identifiant d	Age	Sexe	Province	Salaire annuel	Date d'arrivee sur le Territoire Canadien	
2							
3	1	32	F	CB	28500	15/07/2006	
4	2	39	H	AB	33250	08/06/2008	
5	3	58	H	QC	30000	22/01/2010	
6	4	29	F	NB	56000	10/04/1980	
7	5	26	F	ON	36000	12/01/2016	
8	6	41	H	NB	81000	24/11/1996	
9	7	33	F	CB	50000	15/07/2005	
10	8	35	H	AB	81000	08/06/2006	
11	9	41	H	QC	81000	22/01/2010	
12	10	30	F	NB	45000	10/04/1980	
13	11	35	F	ON	78000	12/01/2016	
14	12	33	H	NB	52000	24/11/1996	
15	13	34	F	CB	46800	15/07/1988	
16	14	40	H	AB	86000	08/06/1986	
17	15	46	H	QC	25000	22/01/2012	
18	16	45	F	NB	46000	10/04/1980	
19	17	48	F	ON	47000	12/01/2016	
20	18	43	H	NB	62000	24/11/1996	
21	19	47	F	ON	12000	15/07/2006	
22	20	35	H	ON	48000	08/06/2006	
23	21	36	H	QC	70000	22/01/1994	
24	22	38	F	QC	36000	10/04/1980	
25	23	39	F	QC	48000	12/01/2012	
26	24	34	H	QC	56000	24/11/1996	
27	25	32	F	ON	78000	15/07/1992	
28	26	31	H	ON	89000	08/06/2008	

Here is the content of the csv file used to make these analyzes

Overview

- Introduction
- Analysis Methodology
- Small queries (Fonctions)
  - Number of records by province for a given yea.e
  - Number of women who have immigrated since 1980 in a given province, whose current annual salary is $ 45,000 or more
- Big-Queries
  - Statistics of Total Registration Numbers by Province from 1980 to 2016
  - Statistics for all provinces of the 2nd small request.
  - Personnalization
- Conclusion

Introduction

- Canada is a country that was built on cultural diversity since the 1980s.
- Many immigrants looking for a more stable geo-political environment immigrate to Canada.
- I gave myself the mandate to do a complete analysis of the recorded data of these migrants.
- I thank you for following me throughout my analysis . The source code is made in Python language. Let us start …

Analysis Methodology

- By habit, I like to make small queries (Functions) for warm-up: It is therefore by this that we will start.

- In general, when these queries are established, for the future, when one has to do with larger queries, simply go search them one by one or even combine them to facilitate the task during a large query.

- It becomes a game of boys, and the code is easier to maintain. Let's start …

Statistics of Total Registration Numbers by Province from 1980 to 2016

Small Queries

1) In preparation for a large query that spans all years of registration, we will create a small function

```
#=============================================================
#   Nombre_Enregistrements_Province_Annee(province, annee)
#=============================================================

def Nombre_Enregistrements_Province_Annee(province, annee):
    with open('Stats.csv','r') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=';')
        nombre = 0;
        for row in csv_reader:
            if row[3] == province and str(annee) in row[5]:
               nombre += 1

    csv_file.close()
    print nombre
    return nombre
```

Nombre_Enregistrements_Province_Annee('NB', 1996)

The answer for this query is 3

Small Queries

In preparation for a large query that deals with all years of registration, we will create a small function that stores in a table all records for a given province from the year 1980 to 2016

```
#=============================================================================================
#   Enregistrements_Province(province, annee)
#=============================================================================================

def Enregistrements_Province(province, annee):
    Enregistrements_pro = []

    for ann in annee:
        Enregistrements_pro.append(Nombre_Enregistrements_Province_Annee(province, ann))
    return Enregistrements_pro
```

Small Queries

```
#=============================================================================================================
#    Enregistrements_Province(province, annee)
#=============================================================================================================

def Enregistrements_Province_sexe_salaire(province, annee, sexe, salaire):
    Enregistrements_pro = []

    for ann in annee:
        Enregistrements_pro.append(Nombre_Enregistrements_sexe_Province_Annee(province, ann, sexe, salaire))
    return Enregistrements_pro
```

Petites requêtes

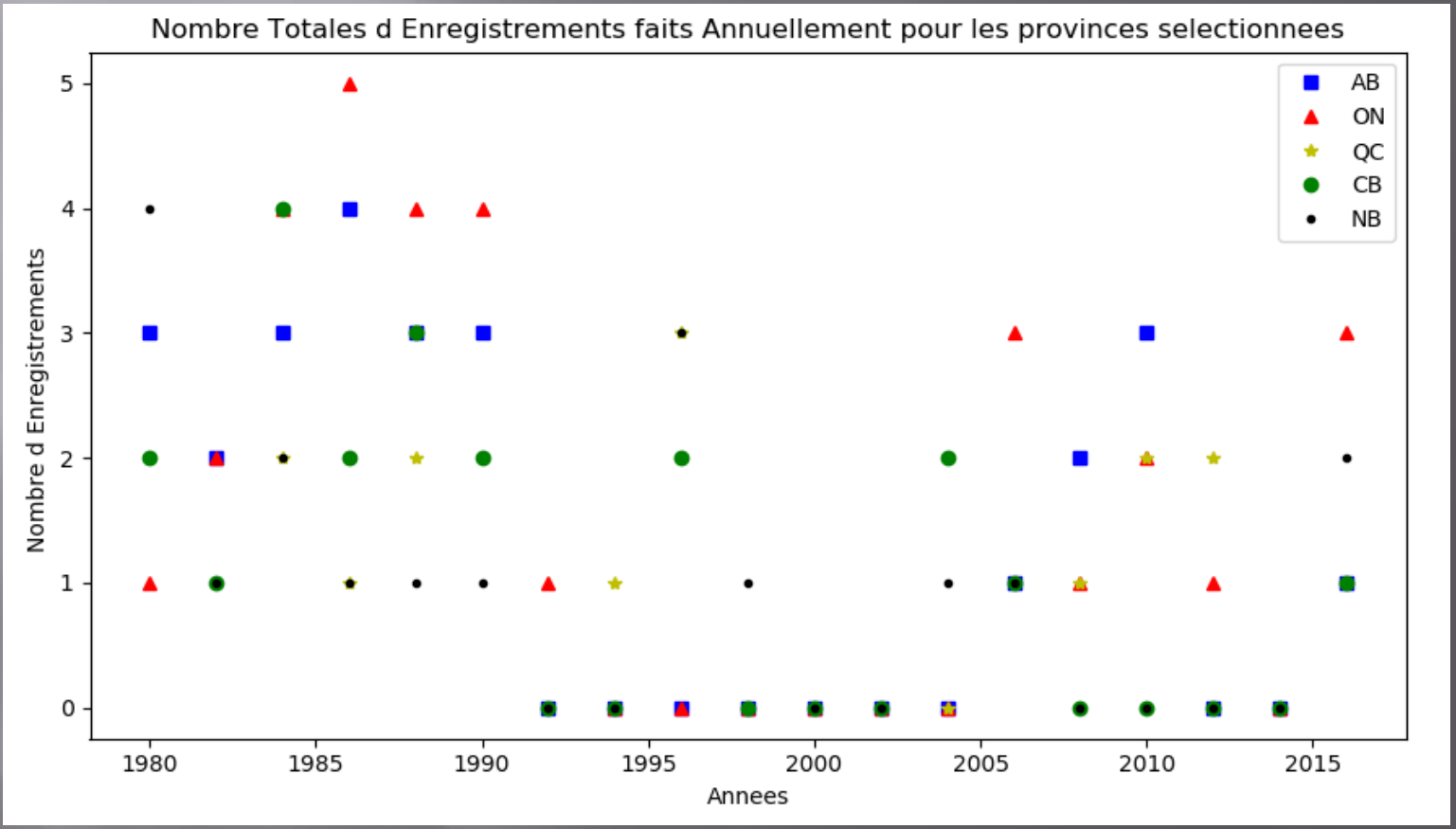
The second part of this big request is to identify the provinces.

```
#=============================================================================================================================
#   Toutes_Les_Enregistrements()
#=============================================================================================================================

def Toutes_Les_Enregistrements():
    AB = []
    ON = []
    QC = []
    CB = []
    NB = []

    provinces = ['AB', 'ON', 'QC', 'CB', 'NB']
    annees = [1980, 1982, 1984, 1986, 1988, 1990, 1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014, 2016]

    for prov in provinces:
        if prov == 'AB':
           AB = Enregistrements_Province('AB', annees)
        elif prov == 'ON':
           ON = Enregistrements_Province('ON', annees)
        elif prov == 'QC':
           QC = Enregistrements_Province('QC', annees)
        elif prov == 'CB':
           CB = Enregistrements_Province('CB', annees)
        elif prov == 'NB':
           NB = Enregistrements_Province('NB', annees)

    return AB, ON, QC, CB, NB
```

Big Query

```
#=============================================================================================================================================
#   Statistiques_des_Enregistrements()
#=============================================================================================================================================

def Statistiques_des_Enregistrements():
    AB, ON, QC, CB, NB = Toutes_Les_Enregistrements()
    annees = [1980, 1982, 1984, 1986, 1988, 1990, 1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014, 2016]

    #print('annees = '+str(len(annees)))
    #print('AB = '+str(len(AB)))
    #print('ON = '+str(len(ON)))
    #print('QC = '+str(len(QC)))
    #print('CB = '+str(len(CB)))
    #print('NB = '+str(len(NB)))

    plt.plot(annees, AB, 'bs', label='AB')
    plt.plot(annees, ON, 'r^', label='ON')
    plt.plot(annees, QC, 'y*', label='QC')
    plt.plot(annees, CB, 'go', label='CB')
    plt.plot(annees, NB, 'k.', label='NB')

    #-----------------------------------
    plt.title('Nombre Totales d Enregistrements faits Annuellement pour les provinces selectionnees')
    plt.ylabel('Nombre d Enregistrements ')
    plt.xlabel('Annees')

    plt.legend()
    plt.show()
```

Visualization



Statistics of Number of women who have immigrated since 1980 and whose current annual salary is >= 45,000

Small Query

1) Statistics of Number of women who have immigrated since 1980 and whose current annual salary is?> = 45,000

```
#=============================================================================
#   Nombre_Enregistrements_Sexe_Province_Annee(province, annee, sexe, salaire)
#=============================================================================

def Nombre_Enregistrements_sexe_Province_Annee(province, annee, sexe, salaire):
    with open('Stats.csv','r') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=';')
        nombre = 0;

        for row in csv_reader:
            date = row[5]
            ann = date[6:10]
            print ('ann = ' +str(ann)+ '\n')
            if row[3] == province and row[2] == sexe and row[4] >= salaire and annee >= int(ann):
               nombre += 1

    csv_file.close()
    print nombre
    return nombre
```

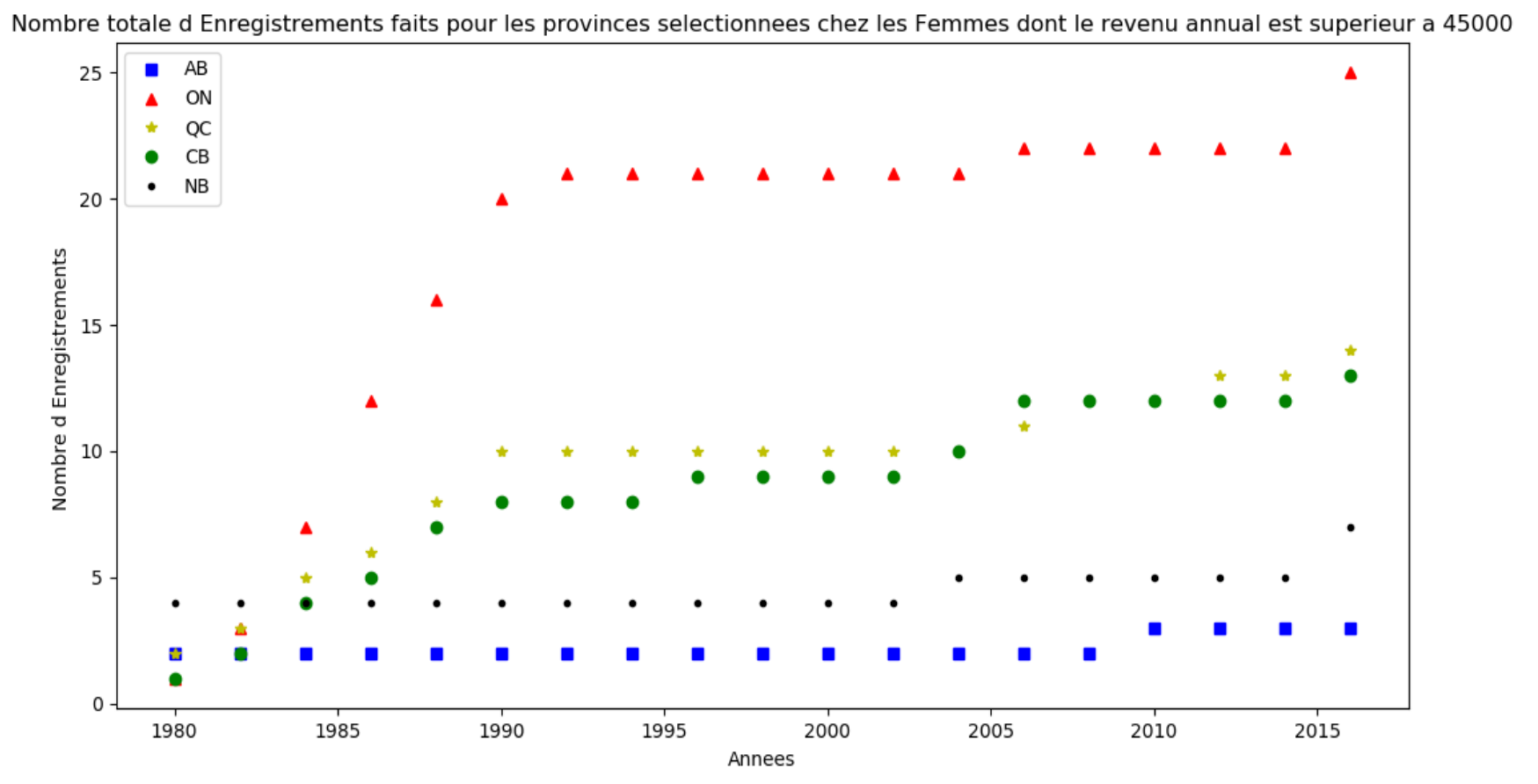
Nombre_Enregistrements_sexe_Province_Annee('QC', 1980, 'F', 45000)

The answer for this query is 14

Small Query

```
#=============================================================================================================================
#    Statistiques_des_Enregistrements()
#=============================================================================================================================

def Toutes_Les_Enregistrements_sexe_salaire():
    AB = []
    ON = []
    QC = []
    CB = []
    NB = []

    provinces = ['AB', 'ON', 'QC', 'CB', 'NB']
    annees = [1980, 1982, 1984, 1986, 1988, 1990, 1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014, 2016]

    for prov in provinces:
        if prov == 'AB':
           AB = Enregistrements_Province_sexe_salaire('AB', annees, 'F', 45000)
        elif prov == 'ON':
           ON = Enregistrements_Province_sexe_salaire('ON', annees, 'F', 45000)
        elif prov == 'QC':
           QC = Enregistrements_Province_sexe_salaire('QC', annees, 'F', 45000)
        elif prov == 'CB':
           CB = Enregistrements_Province_sexe_salaire('CB', annees, 'F', 45000)
        elif prov == 'NB':
           NB = Enregistrements_Province_sexe_salaire('NB', annees, 'F', 45000)

    return AB, ON, QC, CB, NB
```

Visualization



Note that there are more immigrants coming to ONTARIO than in other provinces

Personnalisation

We can even customize the 2nd request:

- Vary the years of the request
- Decide whether it's either men's or women's statistics or even both
- Choose the desired salary
- …

Small Query

1) Vary the years of the request

```
#=============================================================================
#
#=============================================================================

def Annees():
    annees = []
    annee_debut = input('a partir de quelle annee voulex -vous ces statistiques ? ')
    annee_fin = input('jusqu a quelle annee voulez-vous ces statistiques ? ')
    annee_fin = annee_fin+1;

    for x in range(annee_debut, annee_fin):
        annees.append(x)
    print annees
```

Small Query

2) Produce statistics of men and women,
3) Input the starting salary of your choice

```
#=============================================================================================
#    Statistiques_des_Enregistrements()
#=============================================================================================

def Enregistrements_Personnalised_sexe_salaire():
    AB_F = []
    AB_H = []
    ON_F = []
    ON_H = []
    QC_F = []
    QC_H = []
    CB_F = []
    QC_H = []
    NB_F = []
    NB_H = []

    provinces = ['AB', 'ON', 'QC', 'CB', 'NB']
    annees = Annees() #[2004, 2006, 2008, 2010, 2012]
    salaire = input('Vous voulez des statistiques superieur a quel salaire ? ')

    for prov in provinces:
        if prov == 'AB':
           AB_F = Enregistrements_Province_sexe_salaire('AB', annees, 'F', salaire)
           AB_H = Enregistrements_Province_sexe_salaire('AB', annees, 'H', salaire)
        elif prov == 'ON':
           ON_F = Enregistrements_Province_sexe_salaire('ON', annees, 'F', salaire)
           ON_H = Enregistrements_Province_sexe_salaire('ON', annees, 'H', salaire)
        elif prov == 'QC':
           QC_F = Enregistrements_Province_sexe_salaire('QC', annees, 'F', salaire)
           QC_H = Enregistrements_Province_sexe_salaire('QC', annees, 'H', salaire)
        elif prov == 'CB':
           CB_F = Enregistrements_Province_sexe_salaire('CB', annees, 'F', salaire)
           CB_H = Enregistrements_Province_sexe_salaire('CB', annees, 'H', salaire)
        elif prov == 'NB':
           NB_F = Enregistrements_Province_sexe_salaire('NB', annees, 'F', salaire)
           NB_H = Enregistrements_Province_sexe_salaire('NB', annees, 'H', salaire)

    return AB_F, AB_H, ON_F, ON_H, QC_F, QC_H, CB_F, CB_H, NB_F, NB_H, annees, salaire
```

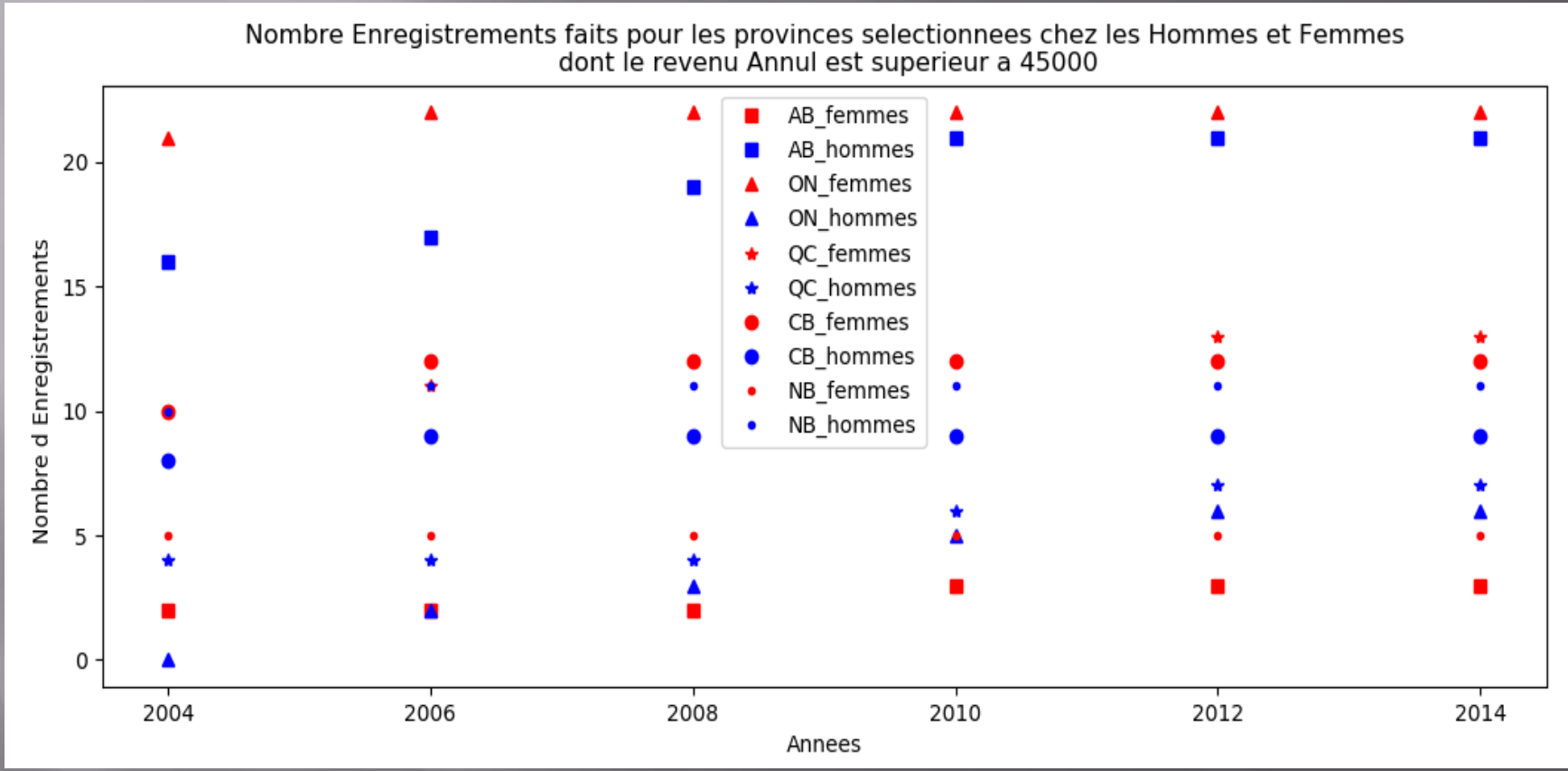
Big Query

```
#=============================================================================================================================================
#    Statistiques_des_Enregistrements_sexe_salaire()
#=============================================================================================================================================

def Statistiques_des_Enregistrements_Personnalise_sexe_salaire():
    AB_F, AB_H, ON_F, ON_H, QC_F, QC_H, CB_F, CB_H, NB_F, NB_H, annees, salaire = Enregistrements_Personnalised_sexe_salaire()

    plt.plot(annees, AB_F, 'rs', label='AB_femmes')
    plt.plot(annees, AB_H, 'bs', label='AB_hommes')

    plt.plot(annees, ON_F, 'r^', label='ON_femmes')
    plt.plot(annees, ON_H, 'b^', label='ON_hommes')

    plt.plot(annees, QC_F, 'r*', label='QC_femmes')
    plt.plot(annees, QC_H, 'b*', label='QC_hommes')

    plt.plot(annees, CB_F, 'ro', label='CB_femmes')
    plt.plot(annees, CB_H, 'bo', label='CB_hommes')

    plt.plot(annees, NB_F, 'r.', label='NB_femmes')
    plt.plot(annees, NB_H, 'b.', label='NB_hommes')

    #-------------------------------------------------
    plt.title('Nombre Enregistrements faits pour les provinces selectionnees chez les Hommes et Femmes \n' +
              'dont le revenu Annul est superieur a ' +str(salaire))
    plt.ylabel('Nombre d Enregistrements ')
    plt.xlabel('Annees')

    plt.legend()
    plt.show()
```

Execution

```
 RESTART: C:\Users\Admin\Desktop\Cegep Victoriaville\SITE_WEB_COURS\Informatique
\Data Science\Assurances_Emploi_DataScience\Stats_Demandes.py
a partir de quelle annee voulez -vous ces statistiques ? 2004
jusqu a quelle annee voulez-vous ces statistiques ? 2014
2004
2006
2008
2010
2012
2014
Vous voulez des statistiques superieur a quel salaire ? 45000
```

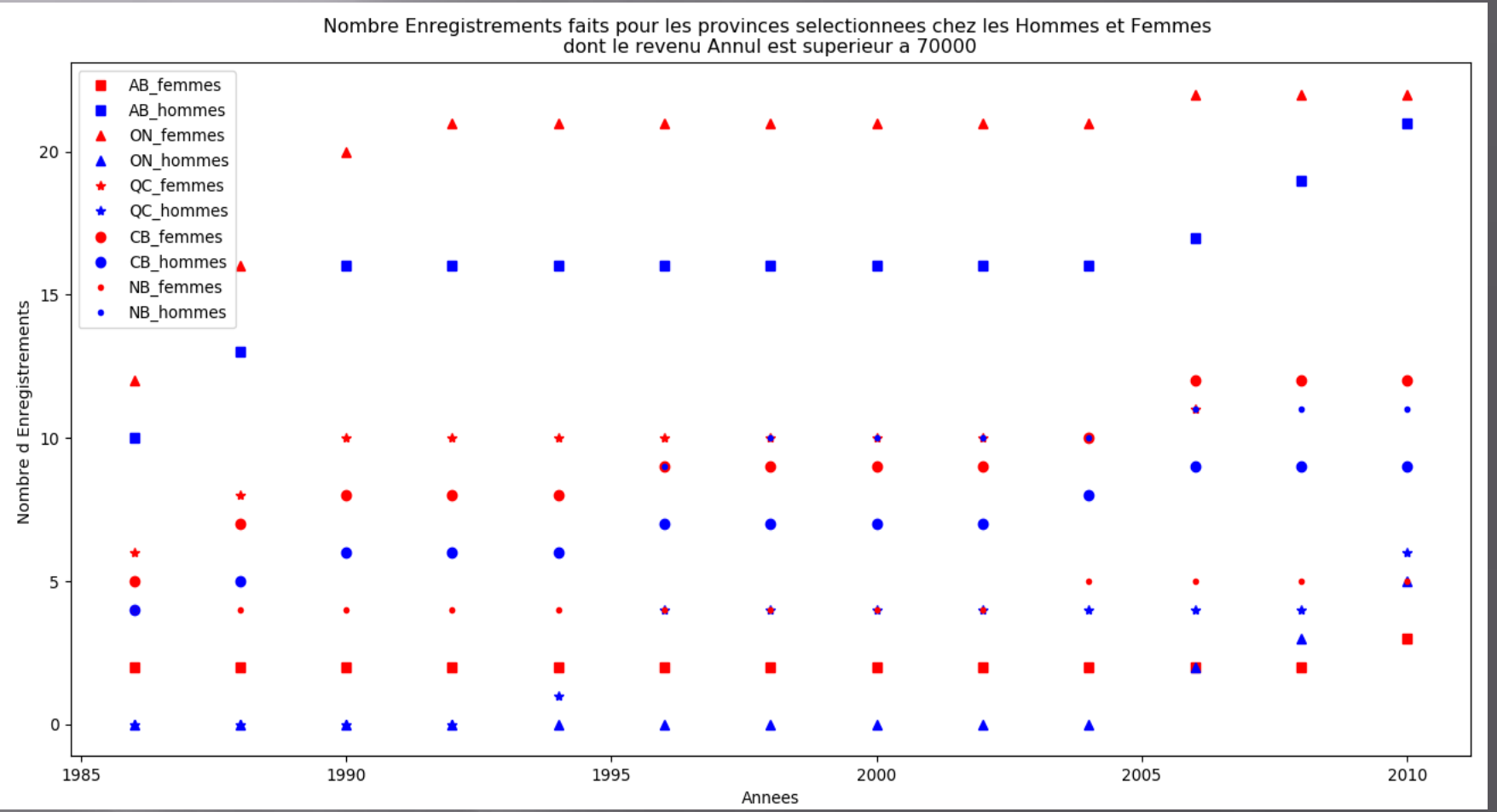
Visualization



Execution

```
 RESTART: C:\Users\Admin\Desktop\Cegep Victoriaville\SITE_WEB_COURS\Informatique
\Data Science\Assurances_Emploi_DataScience\Stats_Demandes.py
a partir de quelle annee voulez -vous ces statistiques ? 1986
jusqu a quelle annee voulez-vous ces statistiques ? 2010
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
Vous voulez des statistiques superieur a quel salaire ? 70000
```

Visualization



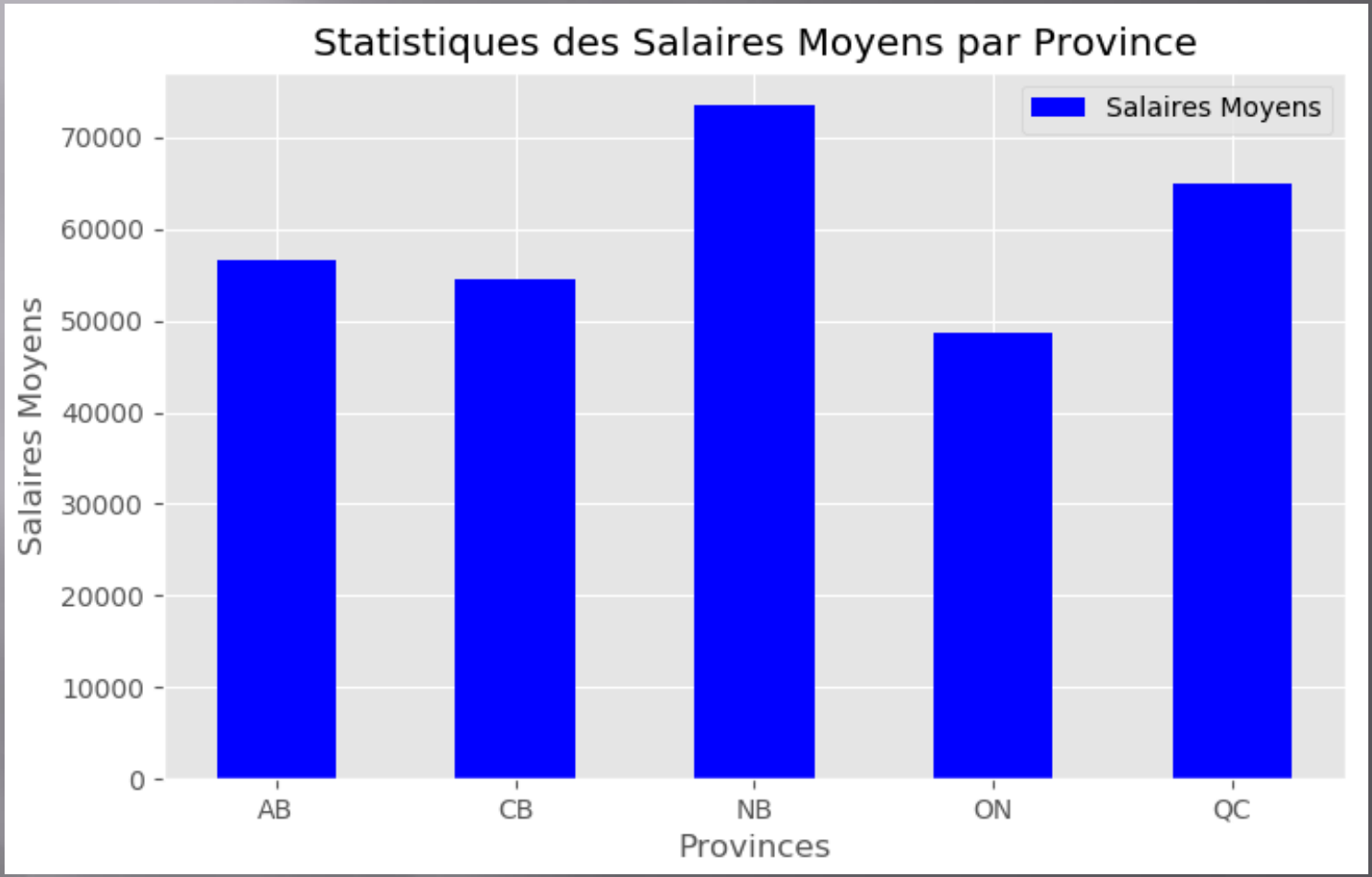
Small additional request

- It is not always required to do sub-queries.
- We can do courses with 2 nested loops.
- This is the case if we want to see the average salary by province.

```
def Enregistrements_Province_salaire():
    provinces = ['AB','ON','QC','CB','NB']
    salaires = []
    salaire_AB = 0
    n_AB = 0
    salaire_ON = 0
    n_ON = 0
    salaire_QC = 0
    n_QC = 0
    salaire_CB = 0
    n_CB = 0
    salaire_NB = 0
    n_NB = 0

    with open('Stats.csv','r') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=';')
        nombre = 0;
        for x in range(0, 4):
            for row in csv_reader:
                if row[3] == 'AB' and row[4] != '':
                    salaire_AB = salaire_AB + int(row[4])
                    n_AB += 1
                elif row[3] == 'ON' and row[4] != '':
                    salaire_ON = salaire_ON + int(row[4])
                    n_ON += 1
                elif row[3] == 'QC' and row[4] != '':
                    salaire_QC = salaire_QC + int(row[4])
                    n_QC += 1
                elif row[3] == 'CB' and row[4] != '':
                    salaire_CB = salaire_CB + int(row[4])
                    n_CB += 1
                elif row[3] == 'NB' and row[4] != '':
                    salaire_NB = salaire_NB + int(row[4])
                    n_NB += 1

        salaire_AB = salaire_AB / n_AB
        salaire_ON = salaire_ON / n_ON
        salaire_QC = salaire_QC / n_QC
        salaire_CB = salaire_CB / n_CB
        salaire_NB = salaire_NB / n_NB

        salaires.append(salaire_AB)
        salaires.append(salaire_ON)
        salaires.append(salaire_QC)
        salaires.append(salaire_CB)
        salaires.append(salaire_NB)

    return salaires
```

```
#=================================================================
#
#=================================================================

def Statistiques_des_Salaires_Totaux_Par_Provinces():
    plt.style.use('ggplot')
    provinces = ['AB','ON','QC','CB','NB']
    salaires = Enregistrements_Province_salaire()

    plt.bar(provinces, salaires, label='Salaires Moyens', color='b', align='center', width=0.5)
    #-----------------------------------------------------------
    plt.title('Statistiques des Salaires Moyens par Province')
    plt.ylabel('Salaires Moyens')
    plt.xlabel('Provinces')

    plt.legend()
    plt.show()
```

Visualisation



Results show that : NB > QC > AB > CB > ON

Thank you for following me in this adventure of DataSciences

Contact me at sandejoel@yahoo.ca
for questions