

Provinces et Villes du Canada

Analyse de données faite par
Joel Sandé

Preface

- Les données issues de ses statistiques viennent de :

http://www.downloadexcelfiles.com/wo_en/download-list-cities-canada#.WpmQS-jOWM8

Pour plus de détails, visitez :

www.downloadexcelfiles.com

Plan de présentation

- Introduction
- Méthodologie d'Analyse
- Petite Requête
- Grosse Requête
- Conclusion

Préface

- Je veux par ce powerpoint, mettre en évidence, à partir d'un cas pratique, la puissance d'analyse du Langage Python et sa capacité à nous générer des résultats sous forme visuelle.
- Je laisserai un lien pour télécharger le fichier csv d'où les données sont tirées. Le code vous sera accessible sur demande à sandejoel@yahoo.ca (Gratuit si vous êtes un étudiant qui suit mon cours, ou un représentant du gouvernement).

Méthodologie d'analyse

- Par habitude, J'aime bien me faire de petites requêtes (Fonctions) d'échauffement : C'est donc par celle-ci que nous allons commencer.
- En générale, lorsque ces requêtes sont établies, pour la suite, lorsqu'on a à faire avec de plus grosses requêtes, il suffit d'aller les rechercher une à une ou même les combiner pour se faciliter la tâche lors d'une grosse requête.
- Ça devient un jeux d'enfant, et le code est plus facile à maintenir. Nous verrons à la fin un cas où cela n'est pas nécessaire.
- Et c'est parti ...

Les 10 Villes les plus peuplées du
Canada en 2011 avec leur
populations respectives.

Petite Requête

La population d'une ville pour une année donnée

```
def Population_Ville(ville, annee):  
    with open('List_of_cities_of_Canada.csv','r') as csv_file:  
        csv_reader = csv.reader(csv_file, delimiter=';')  
        for row in csv_reader:  
            if row[2] == ville and annee == 2011:  
                value = row[3].replace(" ", "")  
                nombre = int(value)  
            elif row[2] == ville and annee == 2006:  
                value = row[4].replace(" ", "")  
                nombre = int(value)  
        csv_file.close()  
    print nombre  
    return nombre
```

Population_Ville('Calgary', 2006)

La réponse est 988812

Petite Requête

La population d'une province pour une année donnée

```
#####  
#                               Definition of our functions  
#####  
  
def Population_Province(province, annee):  
    with open('List_of_cities_of_Canada.csv','r') as csv_file:  
        csv_reader = csv.reader(csv_file, delimiter=';')  
        nombre = 0  
        for row in csv_reader:  
            if row[1] == province and annee == 2011:  
                value = row[4].replace(" ", "")  
                nombre += int(value)  
            elif row[1] == province and annee == 2006:  
                value = row[5].replace(" ", "")  
                nombre += int(value)  
        csv_file.close()  
    print nombre  
    return nombre
```

Population_Province('Alberta', 2011)

La réponse est 2207686

Petite Requête

```
def Province_Datas(province):
    with open('List_of_cities_of_Canada.csv','r') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        population_2011 = 0
        population_2006 = 0
        change = 0
        n_change = 0
        land_area = 0
        population_density = 0
        n_PD = 0
        land_area = 0

        for row in csv_reader:
            if row[1] == province:

                value1 = row[3].replace(" ", "")
                population_2011 += int(value1)

                value2 = row[4].replace(" ", "")
                population_2006 += int(value2)

                value3 = row[5].replace(",",".")
                change += float(value3)
                n_change += 1

                value4 = row[6].replace(",",".")
                land_area += float(value4)

                value5 = row[7].replace(",",".")
                value5 = value5.replace(" ", "")
                population_density += float(value5)
                n_PD += 1

        change /= n_change
        population_density /= n_PD

    csv_file.close()
    print (' population 2011 = ' +str(population_2011)+ '\n population 2006
    return population_2011, population_2006, change, land_area, population_d
```

Recueillir toutes les
données d'une province

Province_Datas('Alberta')

population 2011 = 2473572
population 2006 = 2207686
change = 18.7166666667
land area = 2335.86
population density =
756.622222222

Petite requête

```
def City_Datas(ville):  
    with open('List_of_cities_of_Canada.csv','r') as csv_file:  
        csv_reader = csv.reader(csv_file, delimiter=',')  
        population_2011 = 0  
        population_2006 = 0  
        change = 0  
  
        land_area = 0  
        population_density = 0  
  
        land_area = 0  
  
        for row in csv_reader:  
            if row[2] == ville:  
  
                value1 = row[3].replace(" ", "")  
                population_2011 += int(value1)  
  
                value2 = row[4].replace(" ", "")  
                population_2006 = int(value2)  
  
                value3 = row[5].replace(",", ".")  
                change = float(value3)|  
  
                value4 = row[6].replace(",", ".")  
                value4 = value4.replace(" ", "")  
                land_area = float(value4)  
  
                value5 = row[7].replace(",", ".")  
                value5 = value5.replace(" ", "")  
                population_density = float(value5)  
  
        csv_file.close()  
        print (' population 2011 = ' +str(population_2011)+ '\n population 2006 = ' +str(population_2006)+ '\n change = ' +str(change))  
        return population_2011, population_2006, change, land_area, population_density
```

Recueillir toutes les
données d'une province

population 2011 = 883391
population 2006 = 812129
change = 8.8
land area = 2790.22
population density = 316.6

Petite Requête

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

def All_Provinces():
    provinces = []
    with open('List_of_cities_of_Canada.csv','r') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=';')
        for row in csv_reader:
            provinces.append(row[1])

    csv_file.close()
    dist_provinces = Distinct_List_of(provinces)

    print dist_provinces
    return dist_provinces

#=====
# Distinct_List_of : is a function that take a table, and return a shorter table containing
# distinct element. Elements repeted only once.
# A = [1, 2, 3, 3, 6]   B = Distinct_List_of(A)   B = [1, 2, 3, 6]
#=====

def Distinct_List_of(the_array):
    output = []
    for x in the_array:
        if x not in output:
            output.append(x)
    return output
```

Récupérer les noms de
toutes les provinces

Petite fonction qui me sert à faire une sélection distincte.

NB : Cete fonction n'est pas nécessaire dans la récupération du nom des villes car il n'y a pas de répétition

Petite Requête

Recupérer les noms de toutes les villes

```
def All_Cities():  
    cities = []  
    with open('List_of_cities_of_Canada.csv','r') as csv_file:  
        csv_reader = csv.reader(csv_file, delimiter=';')  
        for row in csv_reader :  
            if row[2] != 'Name':  
                cities.append(row[2])  
    csv_file.close()  
  
    #print cities  
    return cities
```

Grosses Requêtes

```
def Ordered_City_List():
    cities_2011 = All_Cities()
    pop_2011 = All_City_Population(2011)

    #ville, population_2011, population_2006, change, land_area, population_density = City_Datas(ville);

    nombre = input('Combien de villes voulez-vous inclure dans le rapport ? \n')

    ord_city_2011 = []
    ord_pop_2011 = []
    ord_change = []
    ord_land_area = []
    ord_population = []
    ord_density = []

    for x in range(0, nombre):
        mx_2011 = max(pop_2011)

        index_2011 = pop_2011.index(mx_2011)

        pop = pop_2011[index_2011]
        ord_pop_2011.append(pop)

        city_2011 = cities_2011[index_2011]

        print('city 2011 = ' + city_2011 + ' and pop = ' + str(mx_2011) + '\n')

        ord_city_2011.append(city_2011)

        pop_2011.remove(mx_2011)
        cities_2011.remove(city_2011)

    return ord_city_2011, ord_pop_2011, nombre
```

Les statistiques des 10 villes les plus peuplées du Canada:

Dans cette fonction, j'ai choisi de mettre en sorti, les noms des villes et leurs populations respectives. La grande difficulté de cette fonction se trouve dans la boucle FOR

Grosse Requête

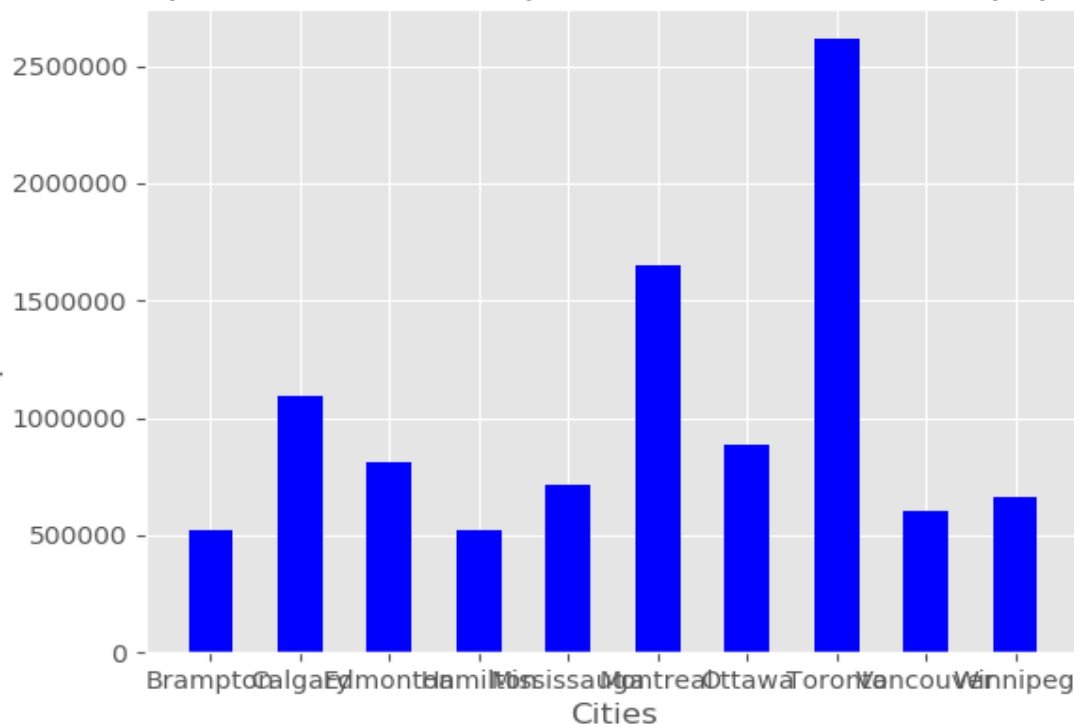
Les 10 villes les plus peuplées du Canada en 2011 avec leurs populations respectives

quelles villes voulez-vous inclure dans le rapport ?

```
Toronto and pop = 2615060  
Montreal and pop = 1649519  
Calgary and pop = 1096833  
Ottawa and pop = 883391  
Edmonton and pop = 812201  
Mississauga and pop = 713443  
Winnipeg and pop = 663617  
Vancouver and pop = 603502  
Brampton and pop = 523911  
Hamilton and pop = 519949
```

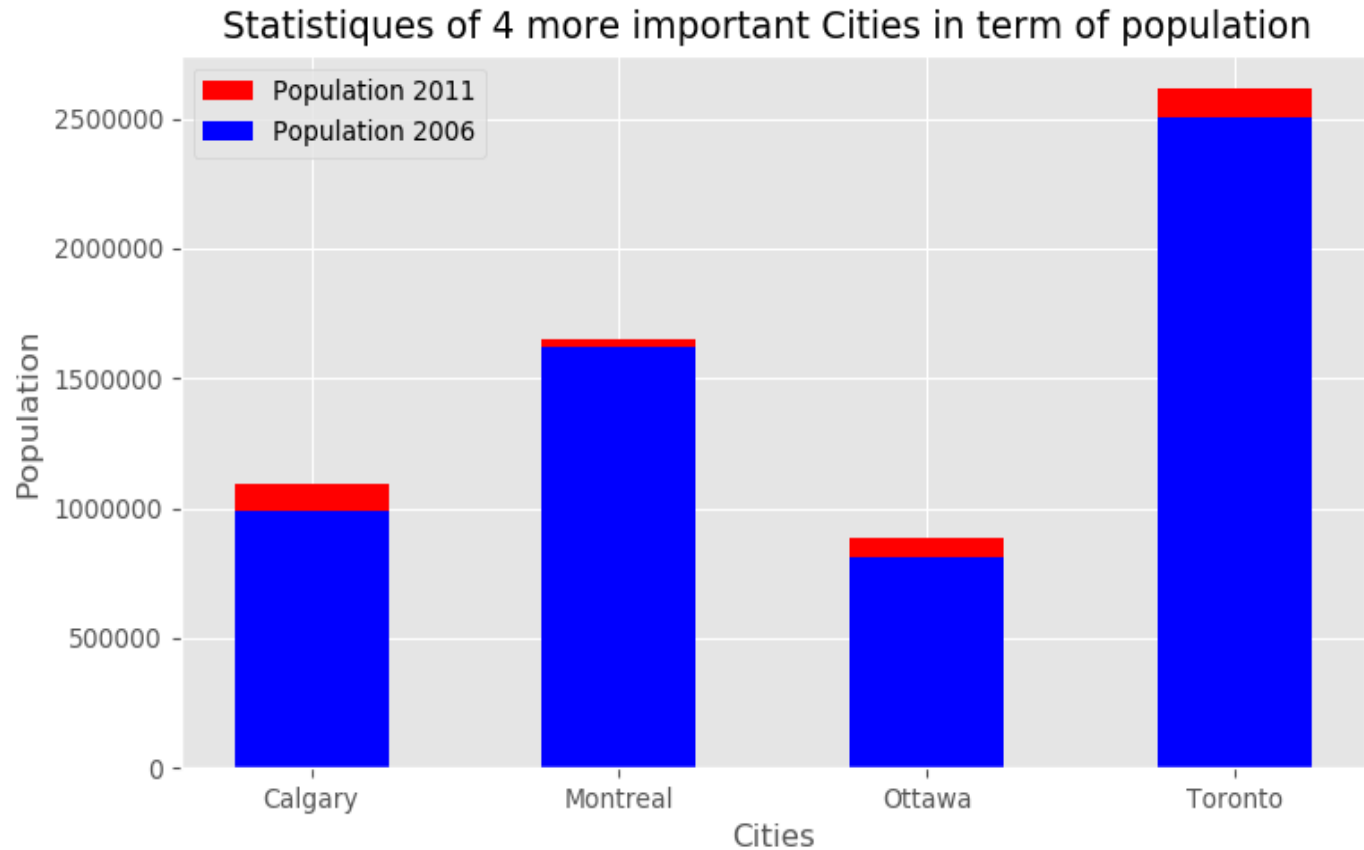
Figure 1

Statistiques de 10 villes importantes en termes de population



Grosse requête

Les statistiques des 4 villes les plus peuplées du Canada



Grosse Requête

villez voulez-vous inclure dans le rapport ?

Toronto and pop = 2615060

Montreal and pop = 1649519

Calgary and pop = 1096833

Ottawa and pop = 883391

Edmonton and pop = 812201

Mississauga and pop = 713443

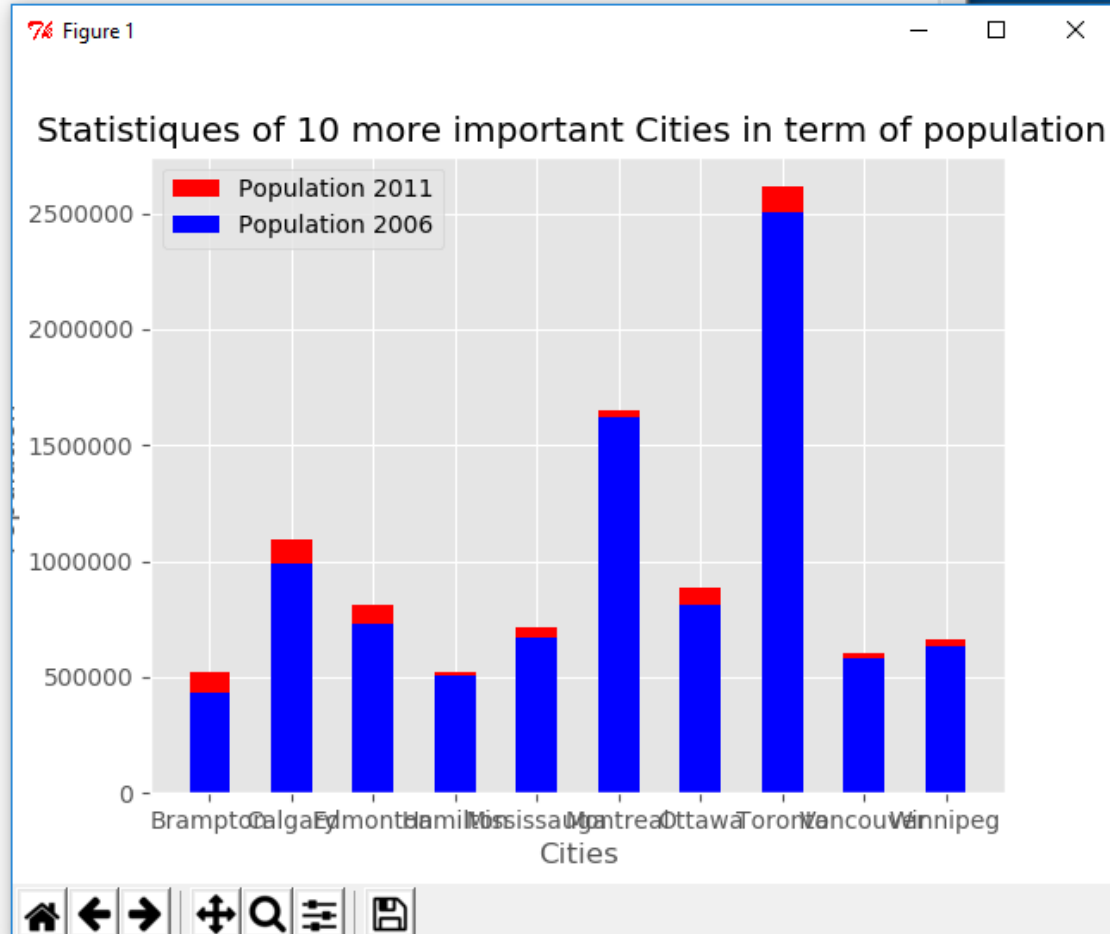
Winnipeg and pop = 663617

Vancouver and pop = 603502

Brampton and pop = 523911

Hamilton and pop = 519949

```
ville = Toronto
population 2011 = 2615060
population 2006 = 2503281
change = 4.5
land area = 630.21
population density = 4149.5
ville = Montreal
population 2011 = 1649519
population 2006 = 1620693
change = 1.8
land area = 365.13
population density = 4517.6
ville = Calgary
population 2011 = 1096833
population 2006 = 988812
change = 10.9
land area = 825.29
population density = 1329.0
```



Conclusion

- Les Populations continuent de croître et de croître, sans vouloir s'arrêter.