

# Annexe: Code Python utilisé pour générer les visualisation

March 28, 2020

## 1 Statistiques en Suisse

Source: [https://opendata.swiss/en/dataset?keywords\\_en=protected-sites](https://opendata.swiss/en/dataset?keywords_en=protected-sites)

```
[1]: import matplotlib
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn-whitegrid')
```

### 1.1 Fréquence des séjours dans la nature

#### 1.1.1 Evolution de 2011 à 2015

```
[2]: title = 'Fréquence des séjours de la population dans la nature (en % de la_
    ↳ population résidente)'
evaluation = ['Chaque jour', 'Presque chaque\n jour', 'Au moins une\n fois par_
    ↳ semaine', 'Au moins une\n fois par mois', 'Moins d\'une fois\n par mois']
values_2011 = [20.7, 23, 39.2, 11.8, 5.2]
values_2015 = [19.3, 25.9, 38.4, 12.3, 4.1]
```

```
[3]: x = np.arange(len(evaluation)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots(figsize=(15,10))
rects1 = ax.bar(x - width/2, values_2011, width, label='En 2011', color='tab:
    ↳ blue', alpha=0.8)
rects2 = ax.bar(x + width/2, values_2015, width, label='En 2015', color='tab:
    ↳ cyan', alpha=0.8)

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('% de la population résidente', fontsize=16)
ax.set_title(title, fontsize=18)
ax.set_xticks(x)
ax.set_xticklabels(evaluation, fontsize=16)
ax.legend(fontsize=16)

def autolabel(rects):
```

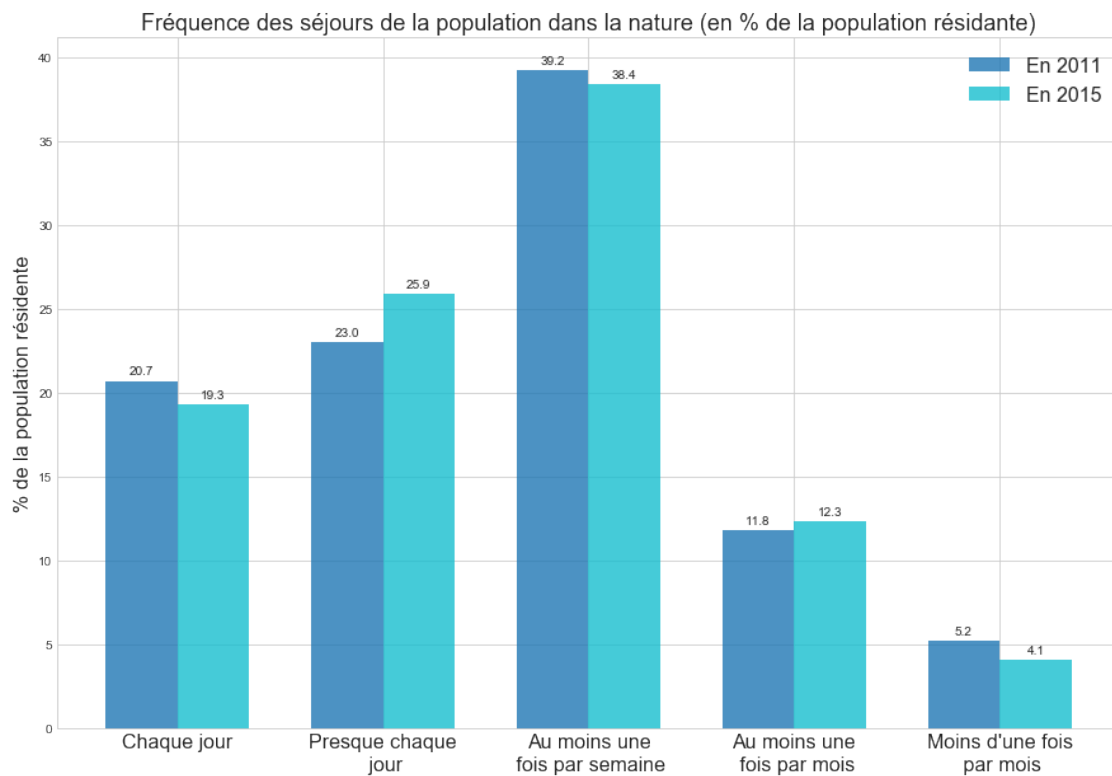
```

"""Attach a text label above each bar in *rects*, displaying its height."""
for rect in rects:
    height = rect.get_height()
    ax.annotate('{}' .format(height),
                xy=(rect.get_x() + rect.get_width() / 2, height),
                xytext=(0, 3), # 3 points vertical offset
                textcoords="offset points",
                ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

plt.savefig('frequence.png')
plt.show()

```



### 1.1.2 Groupé par classe d'âge

```

[4]: #selon age
title = 'Fréquence des séjours de la population dans la nature en 2015 groupé_
      ↳ par classe d\'âge (de 15 à 74 ans)'

```

```

evaluation = ['Chaque jour', 'Presque chaque \njour', 'Au moins une \nfois par_\nsemaine', 'Au moins une \nfois par mois', 'Moins d\'une \nfois par mois']
values_15_24 = [12.2, 27.6, 37.3, 17.8, 5.1]
values_25_34 = [11, 25.2, 45, 15.8, 3]
values_35_44 = [19.4, 25.2, 39.5, 12.6, 3.3]
values_45_54 = [19.1, 24.2, 40.4, 11.4, 4.9]
values_55_64 = [25.9, 22.3, 39.3, 7.9, 4.7]
values_65_74 = [29.1, 33.2, 25.5, 8.7, 3.6]

```

```

[5]: fig, ((ax1, ax2), (ax3, ax4), (ax5, ax6)) = plt.subplots(3, 2, figsize=(30,30))

ax1.bar(evaluation, values_15_24, color='tab:blue', alpha=0.8)
ax1.set_ylim(top=47)
ax1.set_xticklabels(evaluation, fontsize=16)
ax1.set_title('15 - 24 ans', fontsize=18)

ax2.bar(evaluation, values_25_34, color='darkslategray', alpha=0.8)
ax2.set_ylim(top=47)
ax2.set_xticklabels(evaluation, fontsize=16)
ax2.set_title('25 - 34 ans', fontsize=18)

ax3.bar(evaluation, values_35_44, color='steelblue', alpha=0.8)
ax3.set_ylim(top=47)
ax3.set_xticklabels(evaluation, fontsize=16)
ax3.set_title('35 - 44 ans', fontsize=18)

ax4.bar(evaluation, values_45_54, color='cadetblue', alpha=0.8)
ax4.set_title('45 - 54 ans', fontsize=18)
ax4.set_xticklabels(evaluation, fontsize=16)
ax4.set_ylim(top=47)

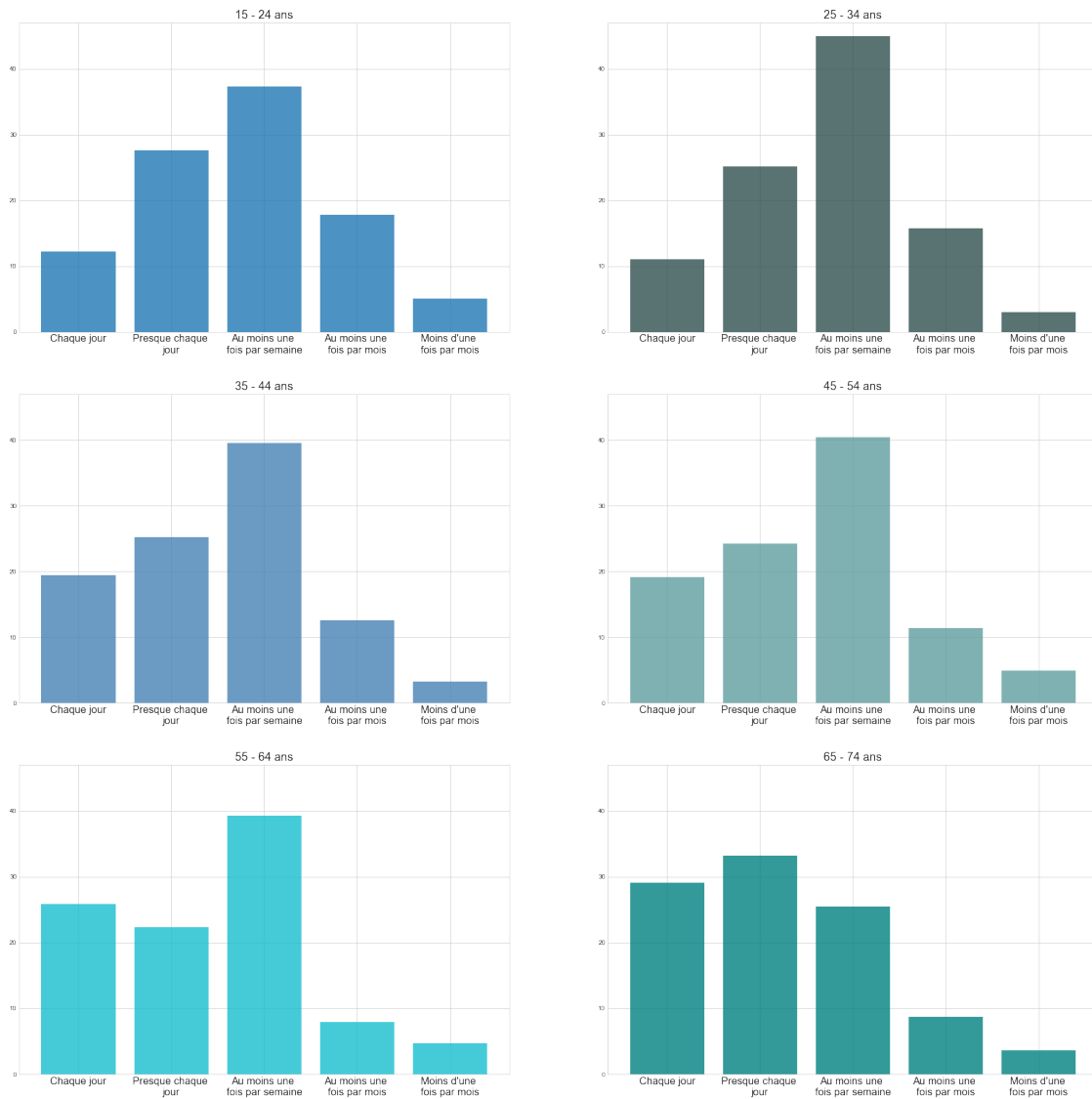
ax5.bar(evaluation, values_55_64, color='tab:cyan', alpha=0.8)
ax5.set_title('55 - 64 ans', fontsize=18)
ax5.set_xticklabels(evaluation, fontsize=16)
ax5.set_ylim(top=47)

ax6.bar(evaluation, values_65_74, color='teal', alpha=0.8)
ax6.set_title('65 - 74 ans', fontsize=18)
ax6.set_xticklabels(evaluation, fontsize=16)
ax6.set_ylim(top=47)

fig.suptitle(title, fontsize=24)
plt.savefig('age.png')
plt.show()

```

Fréquence des séjours de la population dans la nature en 2015 groupé par classe d'âge (de 15 à 74 ans)



### 1.1.3 Groupé selon urbain/rural

```
[6]: #selon ville compagne
title = 'Fréquence des séjours de la population dans la nature en 2015 groupé_
      ↳selon urbains et ruraux'
evaluation = ['Chaque jour','Presque chaque\n jour','Au moins une\n fois par_
      ↳semaine','Au moins une\n fois par mois','Moins d\'une fois\n par mois']
values_urbain = [16.1,25.4,40.1,13.6,4.8]
values_ruraux = [28.3,27.4,33.5,8.7,2.2]
```

```

[7]: x = np.arange(len(evaluation)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots(figsize=(15,10))
rects1 = ax.bar(x - width/2, values_urbain, width, label='Population urbaine',
    →color='tab:blue',alpha=0.8)
rects2 = ax.bar(x + width/2, values_ruraux, width, label='Population_
    →rurale',color='tab:cyan',alpha=0.8)

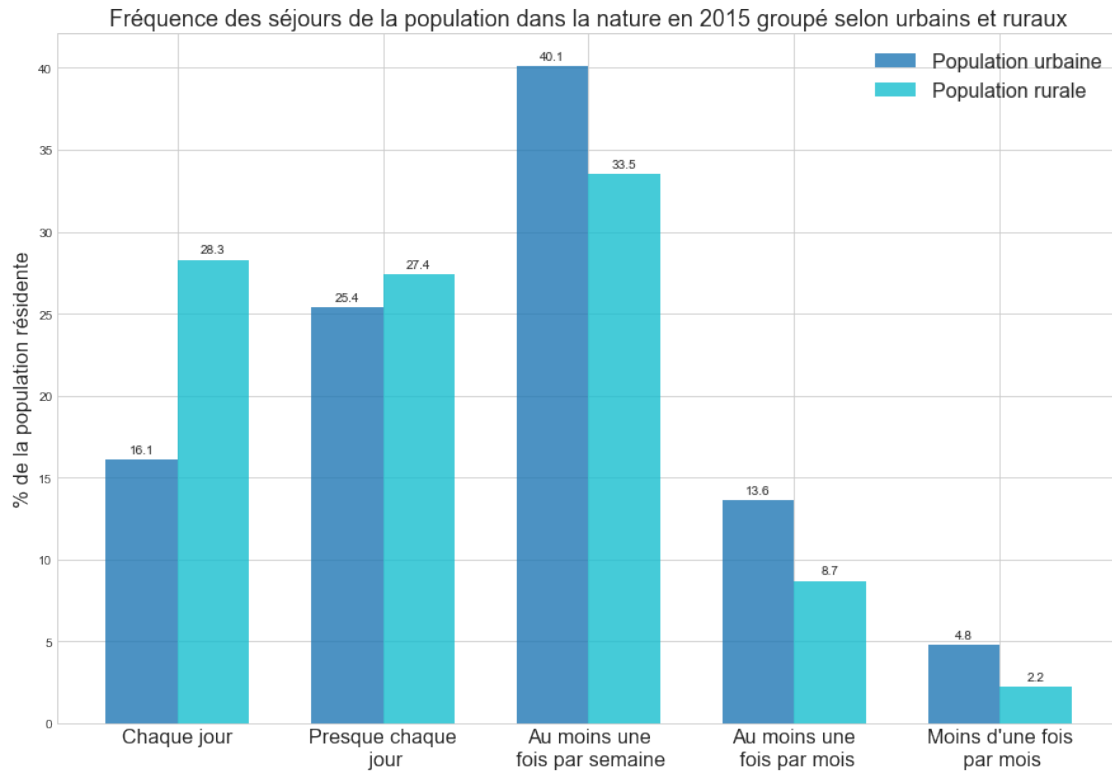
# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('% de la population résidente',fontsize=16)
ax.set_title(title, fontsize=18)
ax.set_xticks(x)
ax.set_xticklabels(evaluation,fontsize=16)
ax.legend(fontsize=16)

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}' .format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

plt.savefig('urbain.png')
plt.show()

```

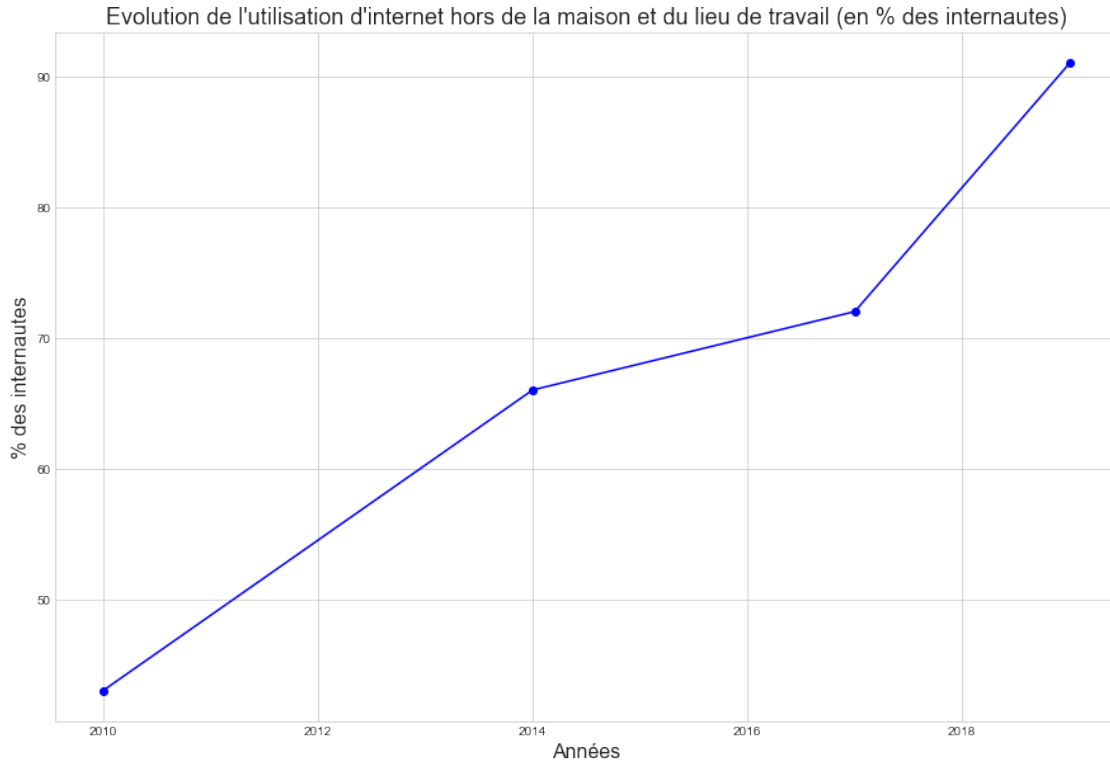


## 1.2 Evolution de l'utilisation d'internet

```
[8]: title = 'Evolution de l\'utilisation d\'internet hors de la maison et du lieu de
→travail (en % des internautes)'
values = [43,66,72,91]
years = [2010,2014,2017,2019]
```

```
[9]: plt.figure(figsize=(15,10))
# Plot continuous green line with circle markers
plt.plot(years,values, 'bo-')

# Plot axes labels and show the plot
plt.title(title, fontsize=18)
plt.xlabel('Années',fontsize=16)
plt.ylabel('% des internautes',fontsize=16)
plt.savefig('internet.png')
plt.show()
```



### 1.3 Appréciation de la qualité de l'environnement

```
[10]: title = 'Appréciation de la qualité de l\'environnement (en % de la population,
→résidante)'
evaluation = ['Très bonne', 'Assez bonne', 'Plutôt mauvaise', 'Très mauvaise']

domicile_2011 = [39.8, 55.1, 4.6, 0.4]
suisse_2011 = [23.1, 68.6, 7.8, 0.5]
monde_2011 = [2.7, 20.8, 65.3, 11.3]
```

```
domicile_2015 = [37.8, 56.7, 5.2, 0.3 ]
suisse_2015 = [25.3, 67.1, 7.1, 0.5]
monde_2015 = [2.7, 17.2, 66.8, 13.2 ]
```

```
[11]: x = np.arange(len(evaluation)) # the label locations
width = 0.25 # the width of the bars

fig, ax = plt.subplots(figsize=(15,10))
rects1 = ax.bar(x - width, domicile_2015, width, label='Au domicile', color='tab:
→blue', alpha=0.8)
rects2 = ax.bar(x, suisse_2015, width, label='En Suisse', color='tab:
→green', alpha=0.8)
```

```

rects3 = ax.bar(x + width, monde_2015, width, label='Dans le monde', color='tab:
→red',alpha=0.8)

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('% de la population résidente',fontsize=16)
ax.set_title(title, fontsize=18)
ax.set_xticks(x)
ax.set_xticklabels(evaluation,fontsize=16)
ax.legend(fontsize=16)

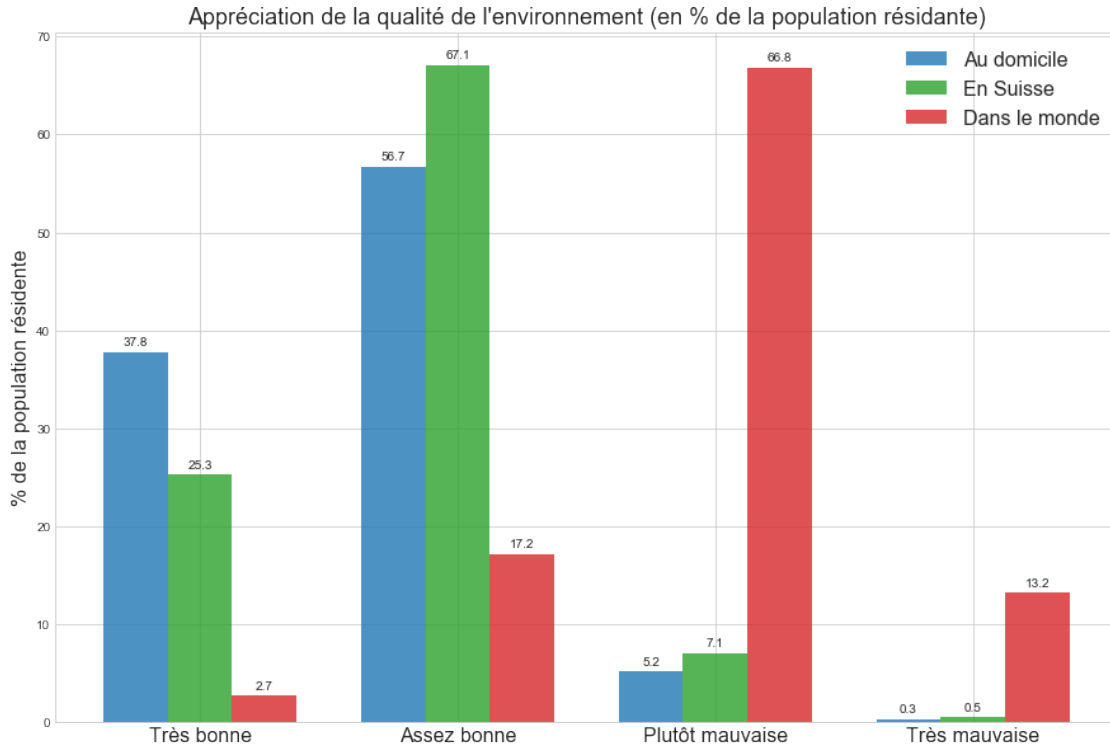
def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)
autolabel(rects3)

plt.savefig('qualité.png')
plt.show()

```





## 1.4 Satisfaction avec le paysage

```
[12]: title = 'Satisfaction avec le paysage (en % de la population résidente)'
evaluation = ['très satisfait', 'plutôt satisfait', 'plutôt insatisfait', 'pas
→satisfait du tout']
values_2011 = [55.6, 38.1, 5.1, 1.2]
values_2015 = [48.8, 44.0, 6.2, 1.0]
```

```
[13]: x = np.arange(len(evaluation)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots(figsize=(15,10))
rects1 = ax.bar(x - width/2, values_2011, width, label='En 2011', color='tab:
→blue', alpha=0.8)
rects2 = ax.bar(x + width/2, values_2015, width, label='En 2015', color='tab:
→cyan', alpha=0.8)

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('% de la population résidente', fontsize=14)
ax.set_title(title, fontsize=18)
ax.set_xticks(x)
ax.set_xticklabels(evaluation, fontsize=16)
```

```

ax.legend(fontsize=16)

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}' .format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

plt.savefig('paysage.png')
plt.show()

```

