

# Projet groupe Élections présidentielles

September 3, 2019

## 1 Projet elections presidentielles

- Vous travaillez pour un organisme de presse.
- Vous devez fournir ttes les analyses et les viz correspondantes aux résultats du 1er tour et
- Surtout des cartes géo interactives (stylé et sympa)
- Soyez curieux et inventifs
- N.B : le notebook est un support (pr vs aider) et non le projet final

Rq : pourquoi ne pas utiliser Flask pr les viz sur Elections Presidentielles

```
In [1]: #attention installer plotly 3.1 pas la derniere version
        #conda install -c plotly plotly=3.10.0
```

```
In [2]: import matplotlib
        import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np
        from pandas.plotting import scatter_matrix
        #pour faire la carte interactive
        import folium
        from folium.plugins import FastMarkerCluster
        import geopandas as gpd

        from branca.colormap import LinearColormap
        #corrélations
        import seaborn as sns
        sns.set()
        #ouvrir dans le navigateur
        import webbrowser
```

```
In [3]: matplotlib.__version__
```

```
Out[3]: '3.0.3'
```

```
In [4]: %matplotlib inline
```

```
In [5]: plt.style.use('ggplot')
```

```
In [6]: all_sheet_in_odict_df = pd.read_excel('source/resultat-elections-2012.xls', sheet_name=
```

```
In [7]: all_sheet_in_odict_df.keys()
```

```
Out[7]: odict_keys(['France entière T1T2', 'Régions T1', 'Régions T2', 'Départements T1', 'Dép
```

```
In [8]: all_sheet_in_odict_df['Circo leg T1'].head()
```

```
Out[8]:
```

	Code du département	Libellé du département	Code de la circonscription	\
0	1	AIN	1	
1	1	AIN	2	
2	1	AIN	3	
3	1	AIN	4	
4	1	AIN	5	

	Libellé de la circonscription	Inscrits	Abstentions	% Abs/Ins	Votants	\
0	1ère circonscription	79016	12872	16.29	66144	
1	2ème circonscription	86823	13126	15.12	73697	
2	3ème circonscription	69832	13522	19.36	56310	
3	4ème circonscription	84491	13160	15.58	71331	
4	5ème circonscription	73646	13316	18.08	60330	

	% Tot/Ins	Blancs et nuls	...	Prénom.8	Voix.8	% Voix/Ins.8	\
0	83.71	1392	...	Nicolas	1465	1.85	
1	84.88	1380	...	Nicolas	1584	1.82	
2	80.64	1030	...	Nicolas	1177	1.69	
3	84.42	1484	...	Nicolas	1636	1.94	
4	81.92	1167	...	Nicolas	1346	1.83	

	% Voix/Exp.8	Sexe.9	Nom.9	Prénom.9	Voix.9	% Voix/Ins.9	% Voix/Exp.9
0	2.26	M	HOLLANDE	François	15675	19.84	24.21
1	2.19	M	HOLLANDE	François	16126	18.57	22.30
2	2.13	M	HOLLANDE	François	12682	18.16	22.94
3	2.34	M	HOLLANDE	François	15084	17.85	21.60
4	2.28	M	HOLLANDE	François	13529	18.37	22.87

```
[5 rows x 75 columns]
```

```
In [9]: tour1 = pd.read_excel('source/resultat-elections-2012.xls', sheet_name = 'Départements
```

```
In [10]: tour1['Code du département'].replace({'ZA': '971', 'ZB': '972', 'ZC': '973', 'ZD': '974', 'ZM
```

```
In [11]: tour2 = pd.read_excel('source/resultat-elections-2012.xls', sheet_name = 'Départements
```

```
In [12]: tour2['Code du département'].replace({'ZA': '971', 'ZB': '972', 'ZC': '973', 'ZD': '974', 'ZM
```

```

In [13]: tour1_2 = tour1.merge(tour2, on = 'Code du département', suffixes = ('T1', 'T2'))

In [14]: tour1_2["rHollandeT1"] = tour1_2['Voix.9'] / (tour1_2["VotantsT1"] - tour1_2["Blancs et nulsT1"])
tour1_2["rSarkozyT1"] = tour1_2['Voix.2'] / (tour1_2["VotantsT1"] - tour1_2["Blancs et nulsT1"])
tour1_2["rNulT1"] = tour1_2["Blancs et nulsT1"] / tour1_2["VotantsT1"]
tour1_2["rHollandeT2"] = tour1_2["VoixT2"] / (tour1_2["VotantsT2"] - tour1_2["Blancs et nulsT2"])
tour1_2["rSarkozyT2"] = tour1_2['Voix.1T2'] / (tour1_2["VotantsT2"] - tour1_2["Blancs et nulsT2"])
tour1_2["rNulT2"] = tour1_2["Blancs et nulsT2"] / tour1_2["VotantsT2"]
data = tour1_2[["Code du département", "Libellé du départementT1",
                "VotantsT1", "rHollandeT1", "rSarkozyT1", "rNulT1",
                "VotantsT2", "rHollandeT2", "rSarkozyT2", "rNulT2"]]

In [15]: tour1_2.to_excel("source/temp/tour1_2.xlsx")
tour1.to_excel("source/temp/tour1.xlsx")
tour2.to_excel("source/temp/tour2.xlsx")
data.to_excel("source/temp/data.xlsx")

In [16]: dix=data.iloc[0:10]
vingt=data.iloc[10:20]
trente=data.iloc[20:30]
quarante=data.iloc[30:40]
cinquante=data.iloc[40:50]
soixante=data.iloc[50:60]
soixante_dix=data.iloc[60:70]
quatre_vingt=data.iloc[70:80]
quatre_vingt_dix=data.iloc[80:90]
cent=data.iloc[90:100]
cent_sept=data.iloc[100:107]

In [17]: dix.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], label=("Hollande", "Sarkozy"),
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/dix')

vingt.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=(8, 4),
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/vingt')

trente.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=(8, 4),
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/trente')

```

```

quarante.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/quarante')

cinquante.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/cinquante')

soixante.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/soixante')

soixante_dix.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figs:
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/soixante_dix')

quatre_vingt.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figs:
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/quatre_vingt')

quatre_vingt_dix.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], :
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/quatre_vingt_dix')

cent.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=(8,4
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')

```

```

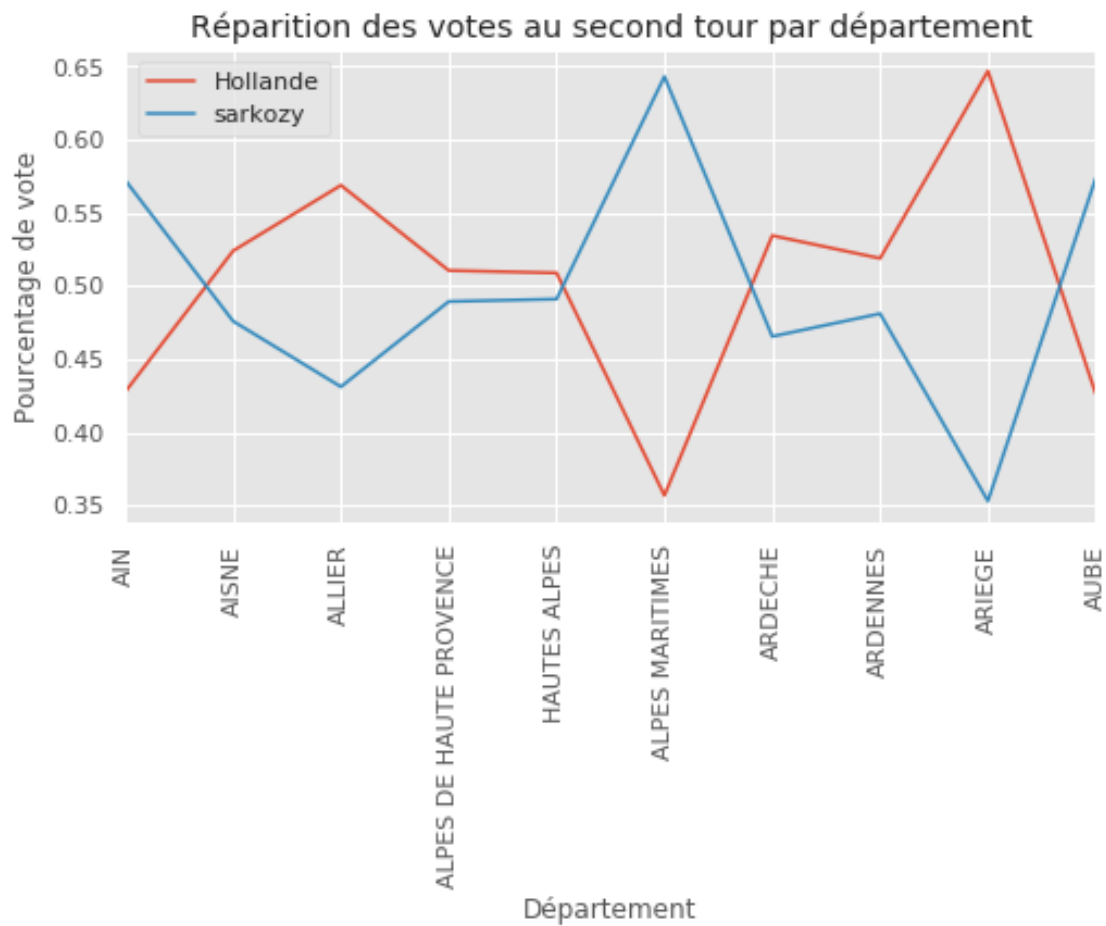
plt.savefig('templates/cent')

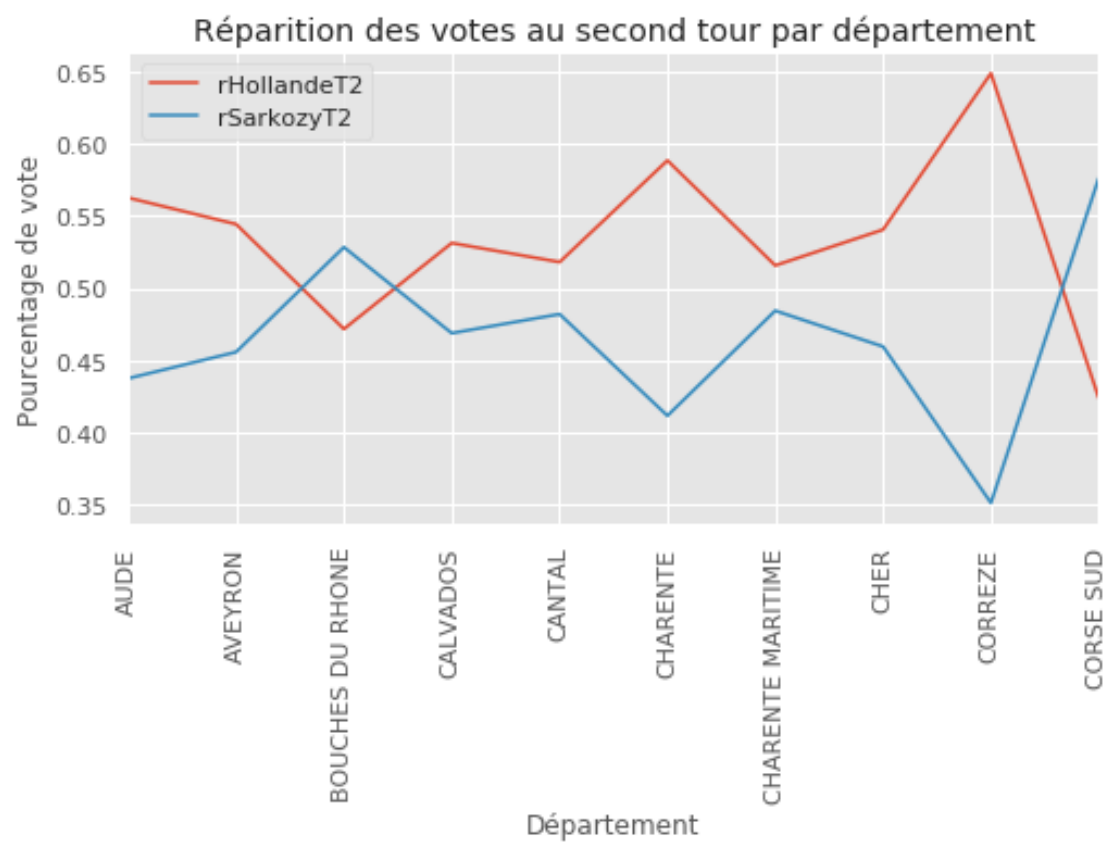
cent_sept.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"], figsize=
plt.xticks(rotation='vertical')
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
plt.savefig('templates/cent_sept')

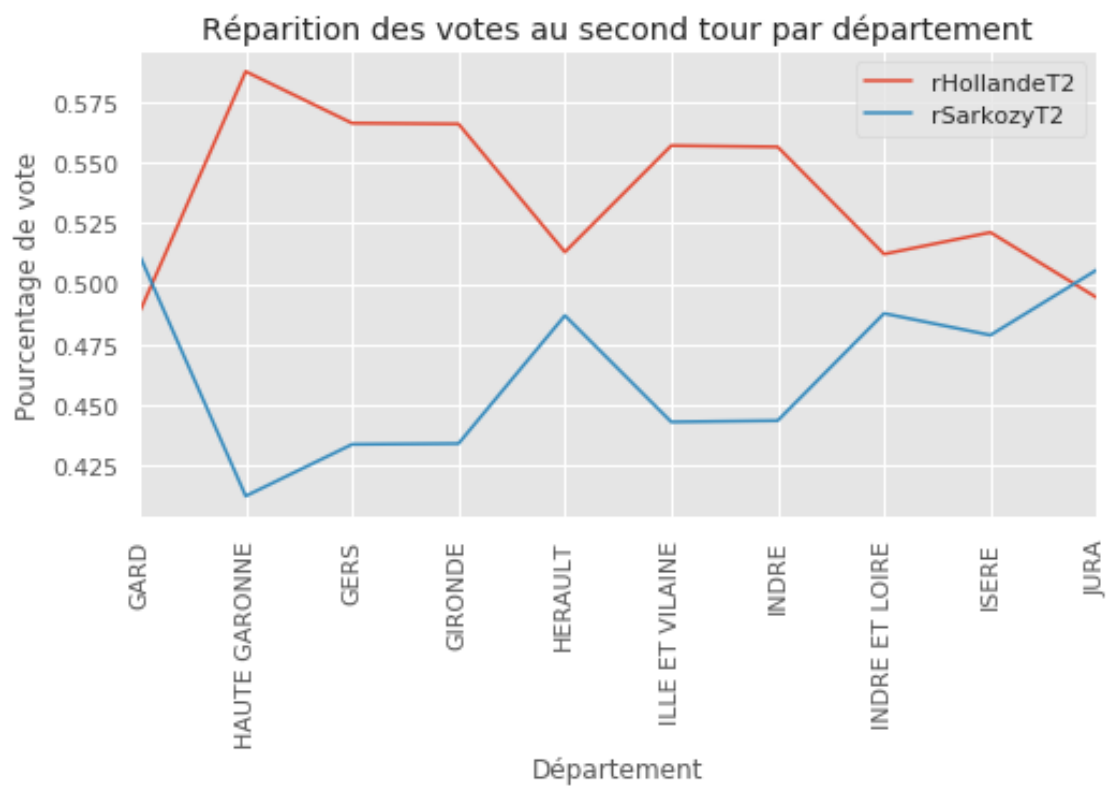
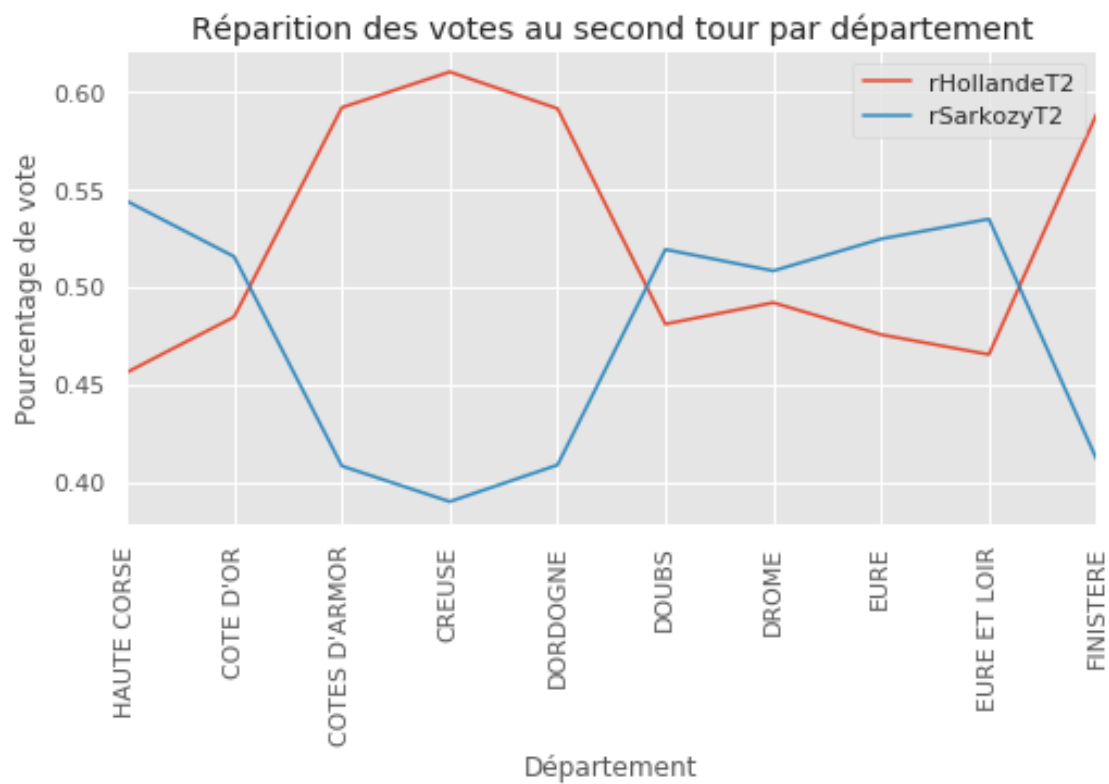
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')

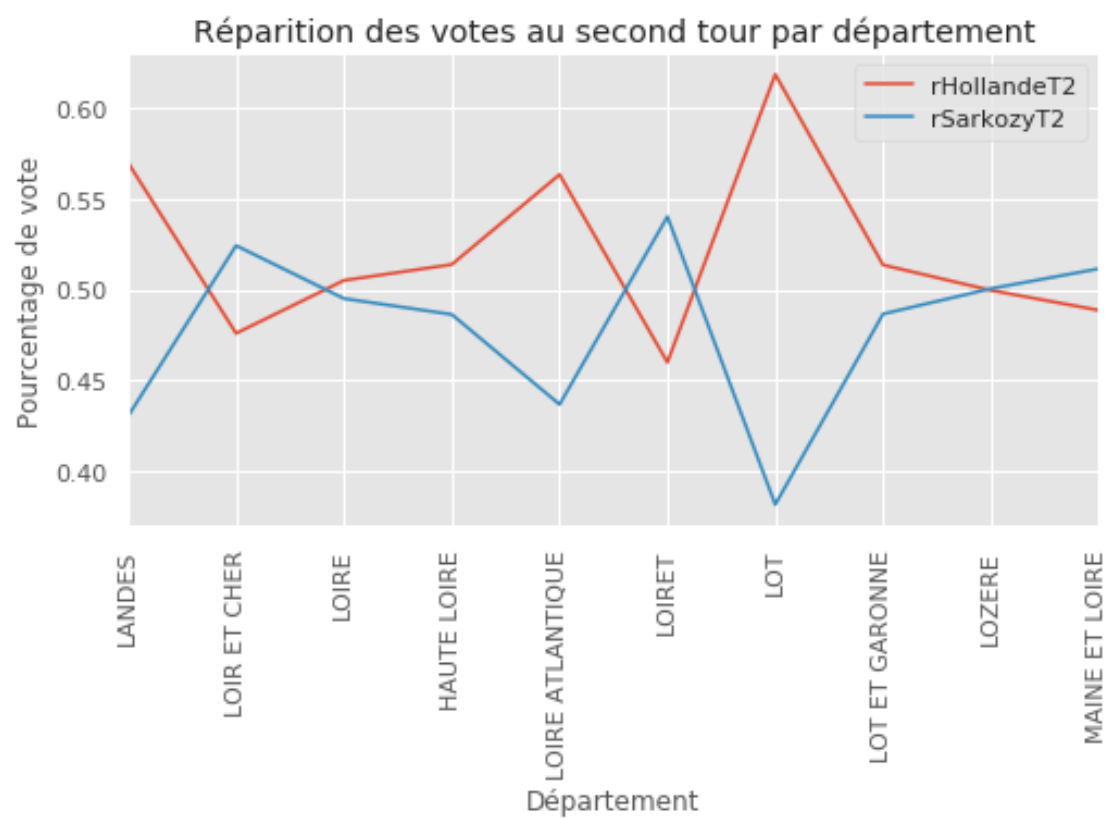
print("")

```

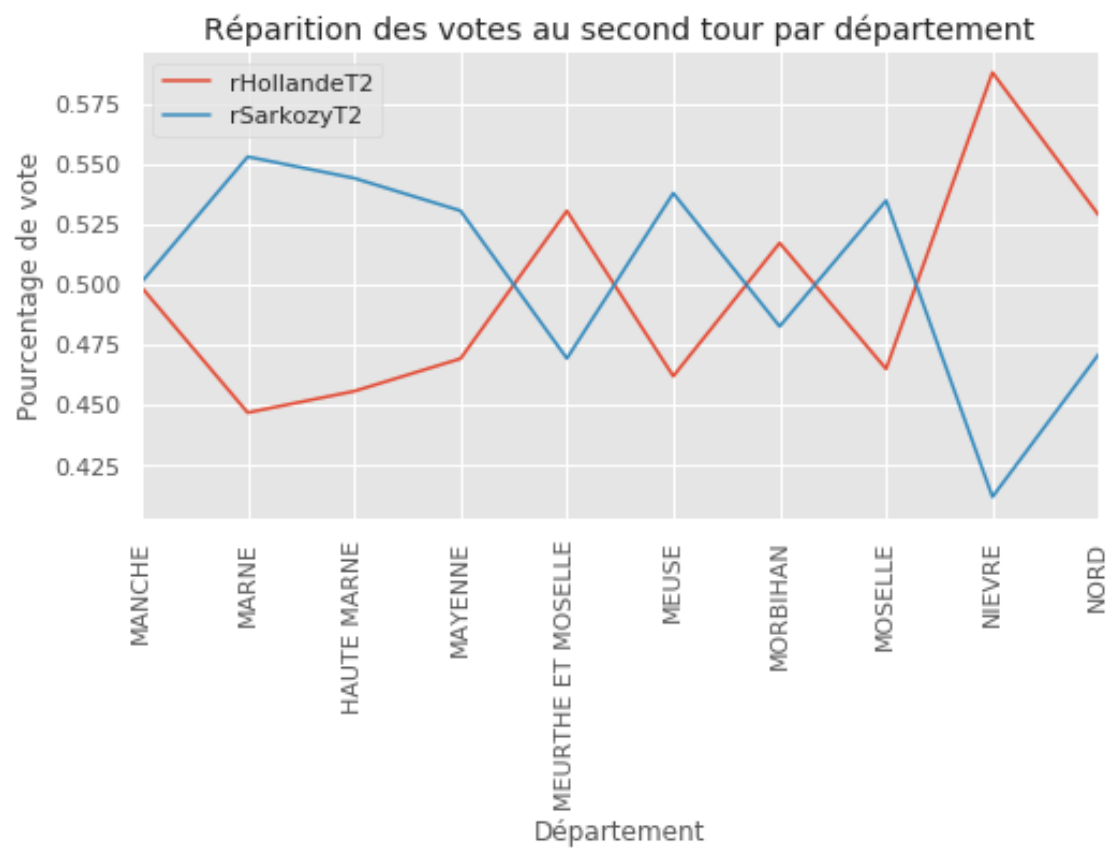


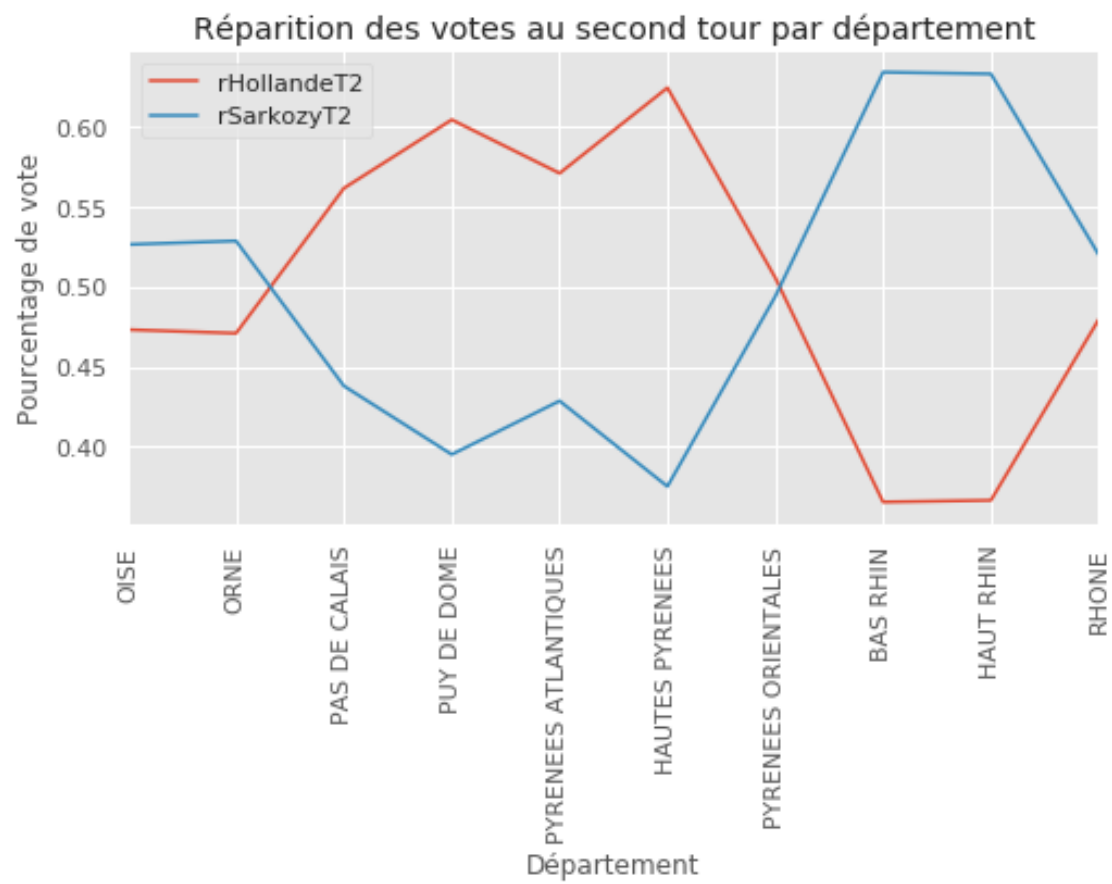


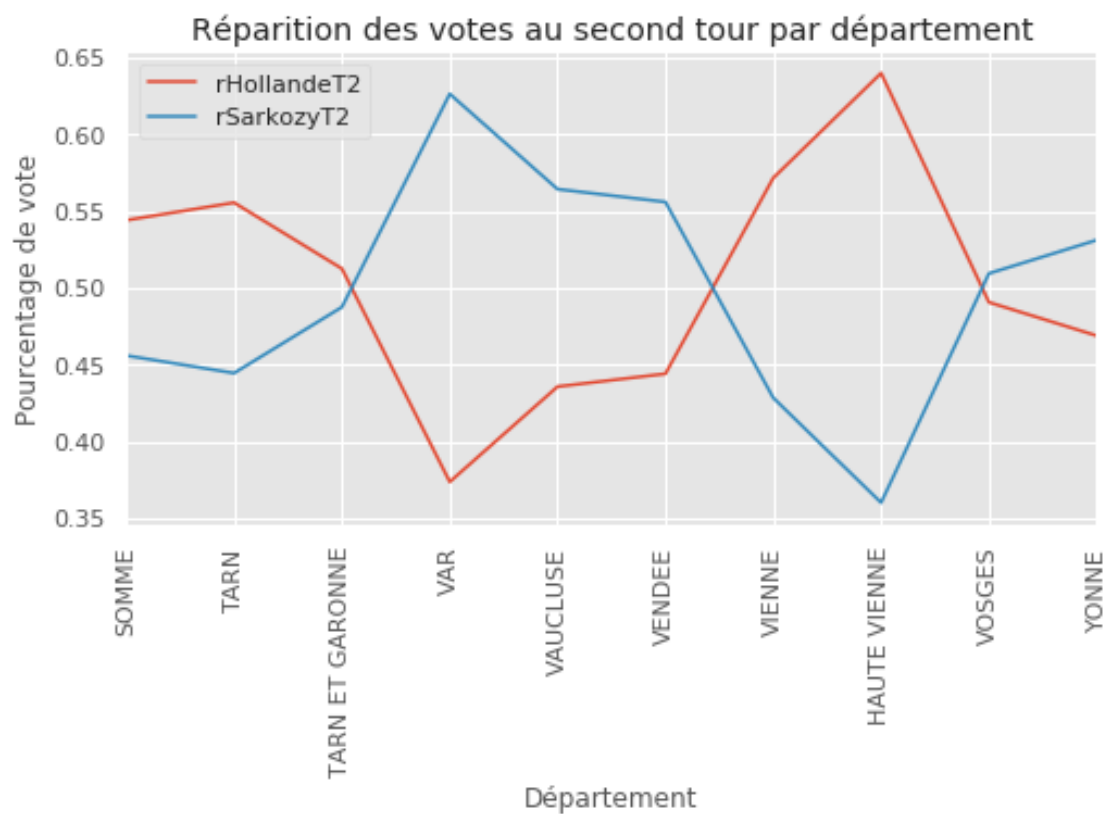
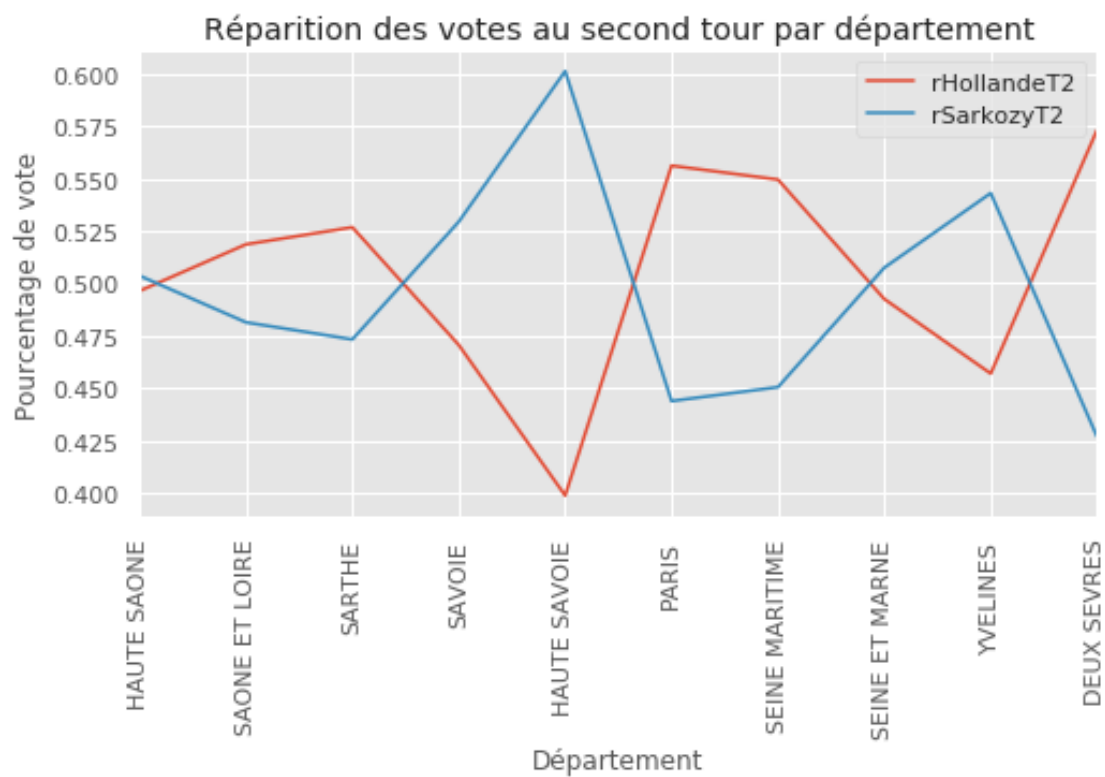


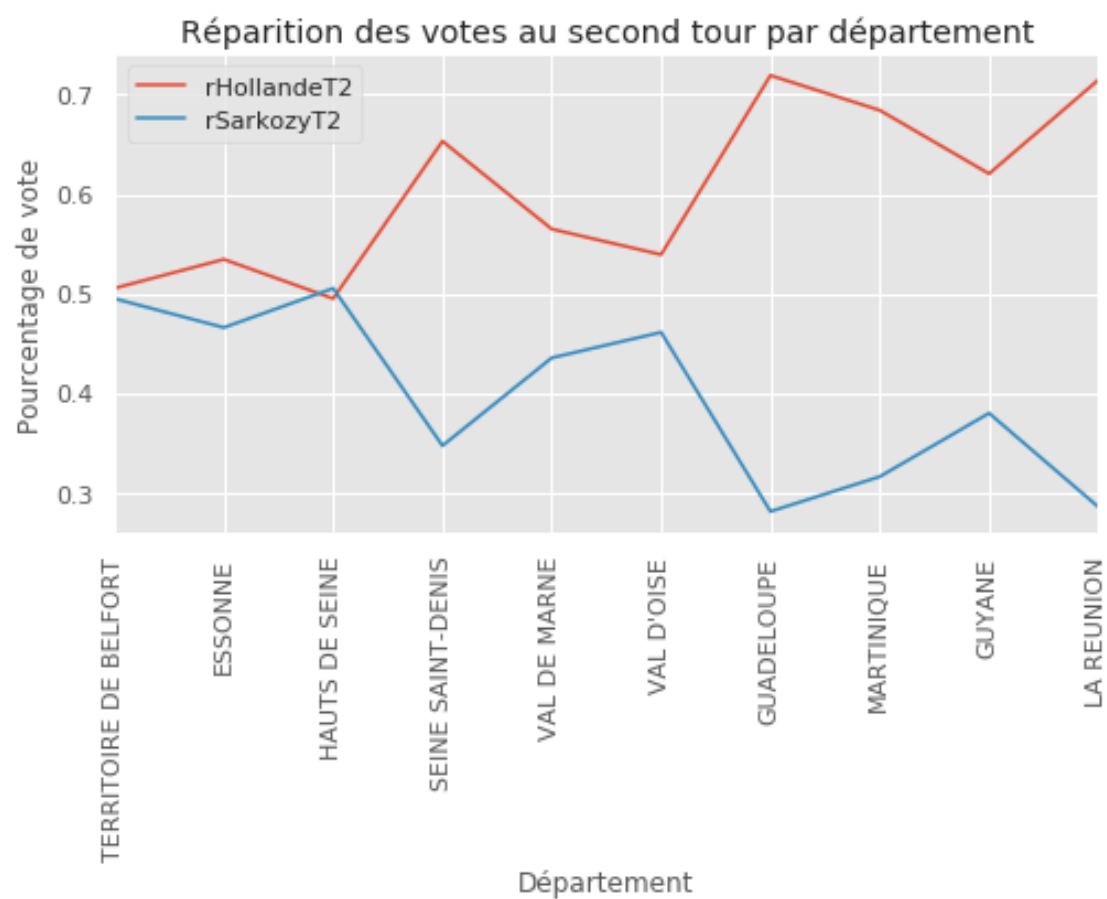


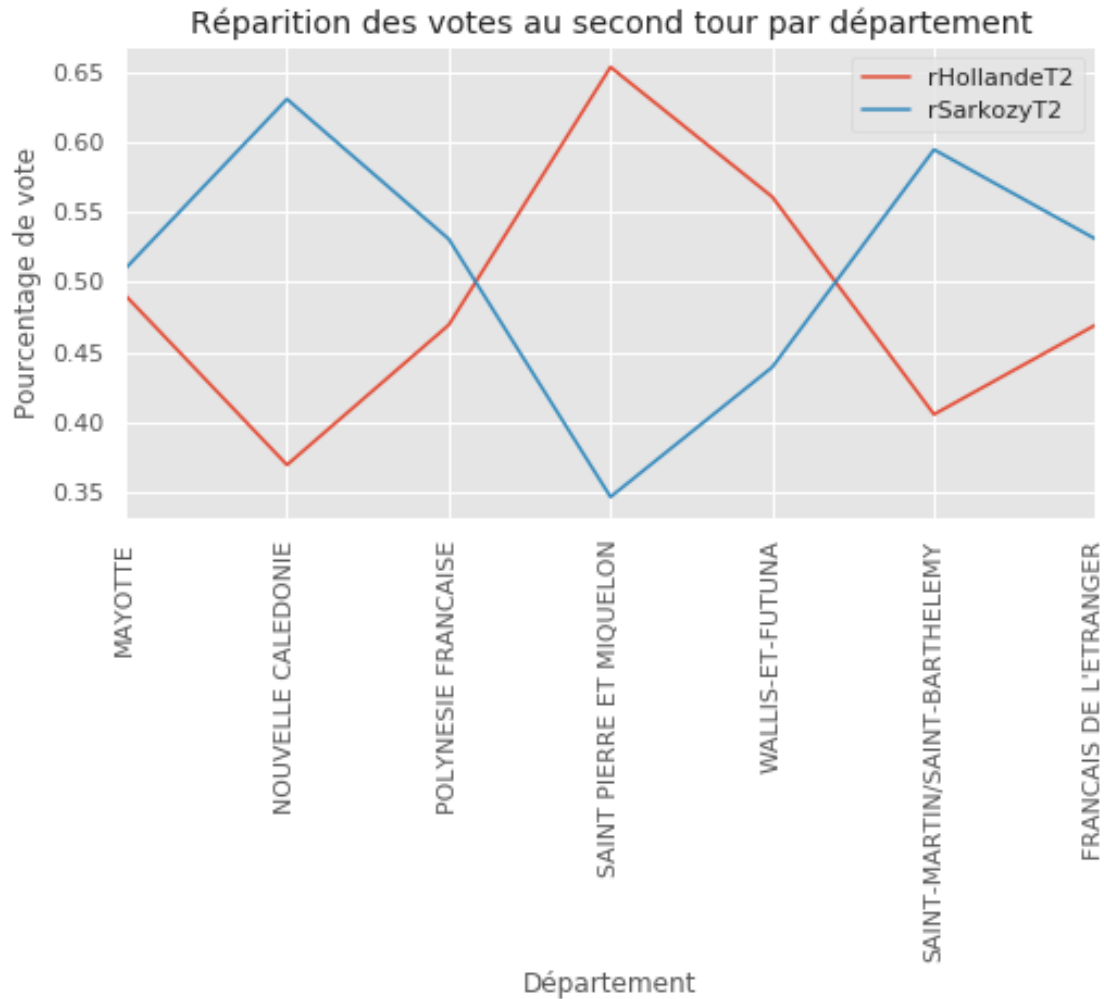




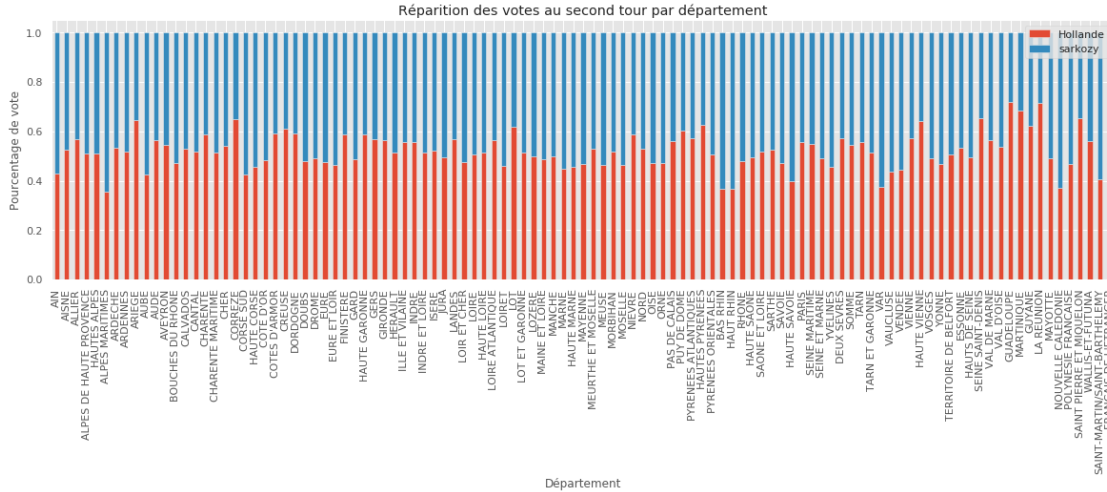






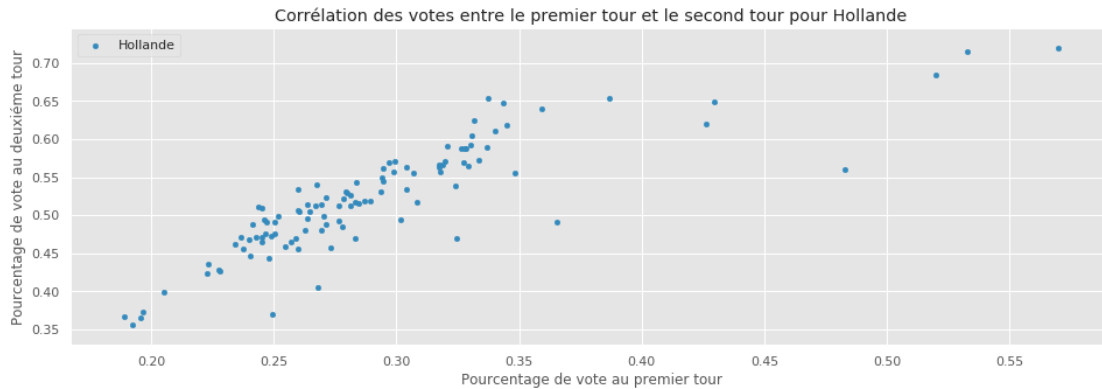


```
In [18]: data.plot(x="Libellé du départementT1", y=["rHollandeT2", "rSarkozyT2"],
                  label=("Hollande", "sarkozy"), figsize=(20,5), kind="bar", stacked=True)
plt.title('Répartition des votes au second tour par département')
plt.xlabel('Département')
plt.ylabel('Pourcentage de vote')
print("")
plt.savefig('templates/Répartition des votes au second tour par département')
```



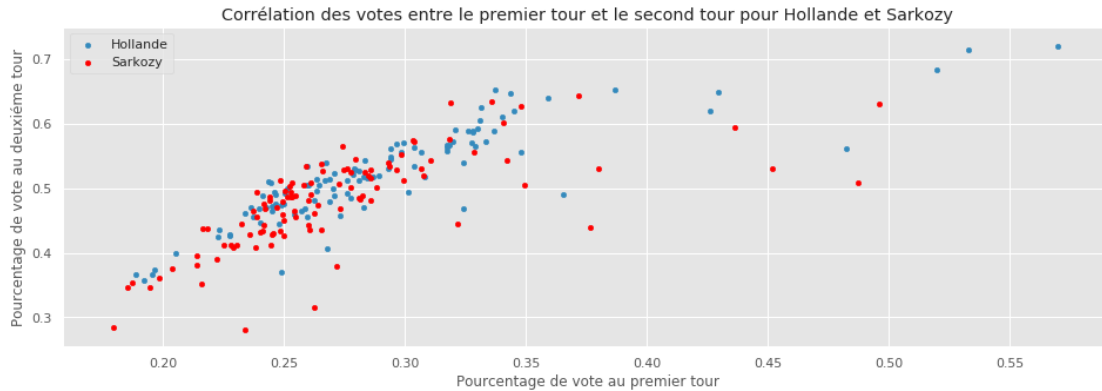
In [19]: # Graphe de corrélation (nuage de point = scatter plot)

```
data.plot(x="rHollandeT1", y="rHollandeT2", figsize=(16,5),
          kind="scatter", label="Hollande", title="correlation")
plt.title('Corrélation des votes entre le premier tour et le second tour pour Hollande')
plt.xlabel('Pourcentage de vote au premier tour')
plt.ylabel('Pourcentage de vote au deuxième tour')
plt.savefig('templates/Corrélation des votes entre le premier tour et le second tour p
```



In [20]: # Superposition de graphes ; 2 scatter plot avec des couleurs diff et une légende

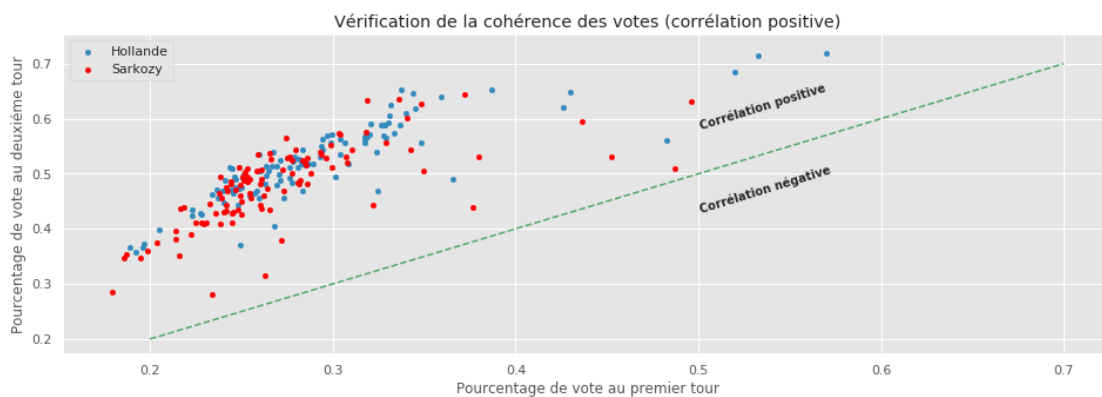
```
ax=data.plot(x="rHollandeT1", y="rHollandeT2", figsize=(16,5),
             kind="scatter", label="Hollande", title="Corrélation des votes entre le
data.plot(x="rSarkozyT1", y="rSarkozyT2", kind="scatter", label="Sarkozy", ax=ax, c="
plt.xlabel('Pourcentage de vote au premier tour')
plt.ylabel('Pourcentage de vote au deuxième tour')
plt.savefig('templates/Corrélation des votes entre le premier tour et le second tour p
```



In [21]: # Ajouter une ligne sur un graphe

```
ax=data.plot(x="rHollandeT1", y="rHollandeT2", figsize=(16,5),
             kind="scatter", label="Hollande", title="Vérification de la cohérence des votes",
             data.plot(x="rSarkozyT1", y="rSarkozyT2", kind="scatter", label="Sarkozy", ax=ax, c="red"),
             plt.xlabel('Pourcentage de vote au premier tour')
             plt.ylabel('Pourcentage de vote au deuxième tour')

ax.plot([0.2,0.7], [0.2,0.7], "g--")
ax.text(0.5, 0.65, "Corrélation positive", weight="bold", rotation="17")
ax.text(0.5, 0.5, "Corrélation négative", weight="bold", rotation="17")
plt.savefig('templates/Vérification de la cohérence des votes (corrélacion positive)')
```



In [22]: # plusieurs graphes sur la même figure

```
fig, axes = plt.subplots(1, 2, figsize=(16,5), sharey=True)

data.plot(x="rHollandeT1", y="rHollandeT2", kind="scatter",
          label="Hollande", ax=axes[0])
```

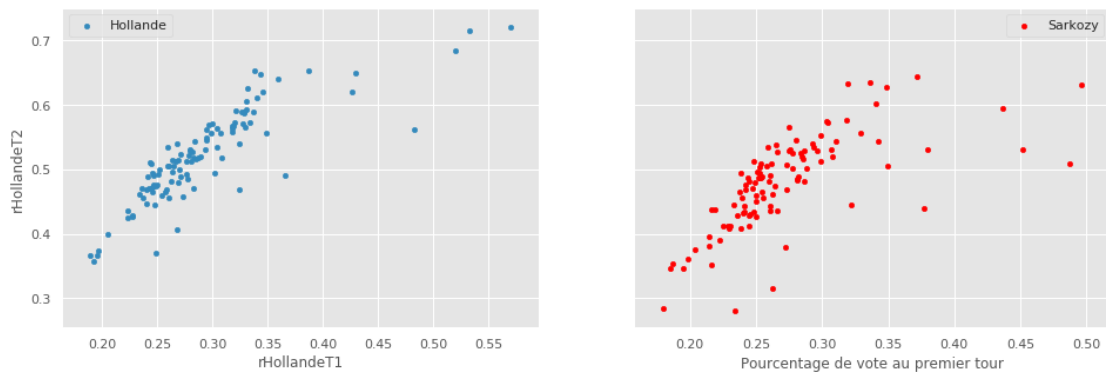
```

plt.xlabel('Pourcentage de vote au premier tour')
plt.ylabel('Pourcentage de vote au deuxième tour')

data.plot(x="rSarkozyT1", y="rSarkozyT2", kind="scatter",
          label="Sarkozy", ax=axes[1], c="red")
plt.xlabel('Pourcentage de vote au premier tour')
plt.ylabel('Pourcentage de vote au deuxième tour')

print('')
plt.savefig('templates/Corrélation des votes entre le premier tour et le second tour')

```

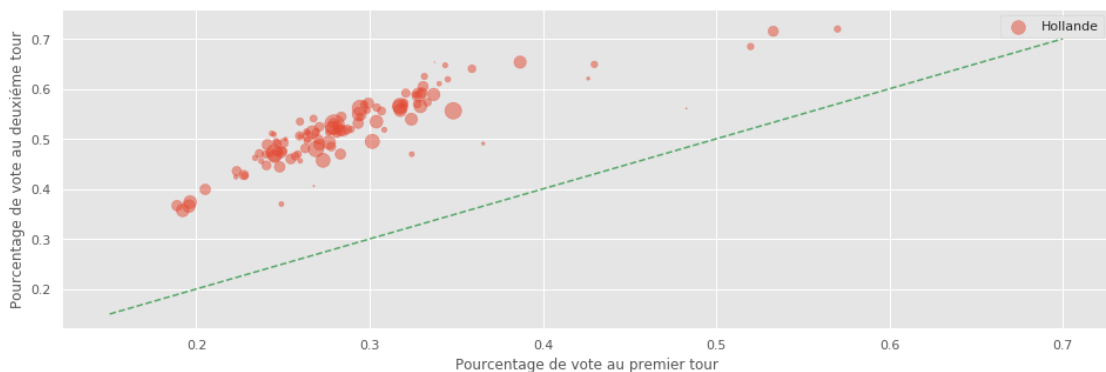


```

In [23]: fig, axes = plt.subplots(1, 1, figsize=(16,5))
c = axes.scatter(x=data["rHollandeT1"],
                 y=data["rHollandeT2"],
                 s=data["VotantsT1"]/5000, alpha=0.5)
plt.xlabel('Pourcentage de vote au premier tour')
plt.ylabel('Pourcentage de vote au deuxième tour')

axes.plot([0.15,0.7], [0.15,0.7], "g--")
axes.legend( (c,), ("Hollande",) );

```





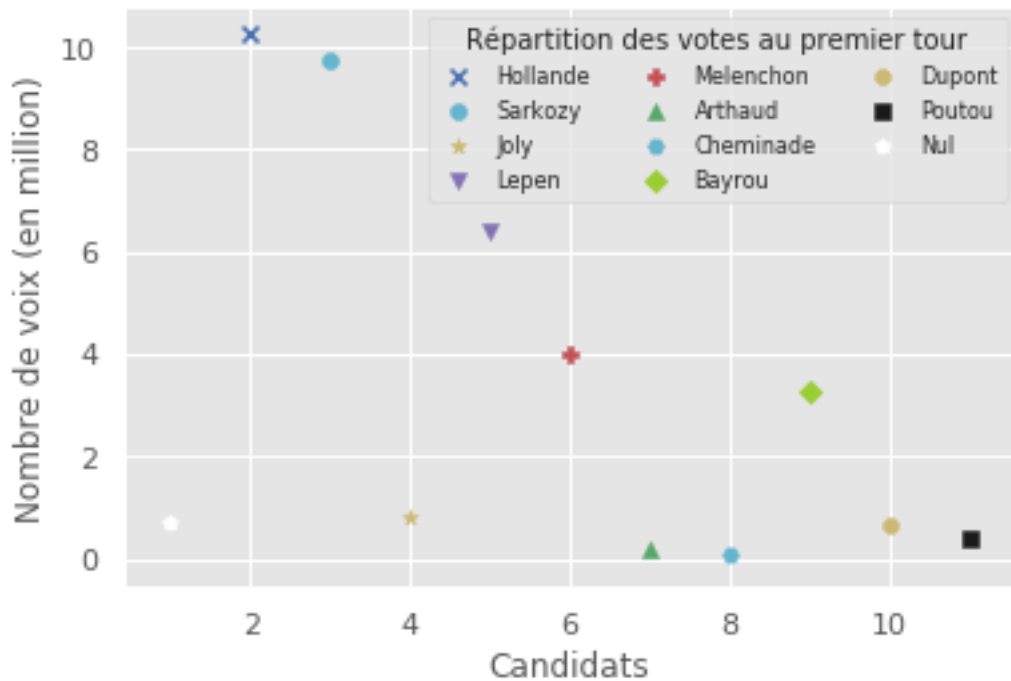
```

In [24]: Joly=sum(tour1_2["VoixT1"])/1000000
Lepen=sum(tour1_2["Voix.1T1"])/1000000
Melenchon=sum(tour1_2["Voix.3"])/1000000
Arthaud=sum(tour1_2["Voix.5"])/1000000
Cheminade=sum(tour1_2["Voix.6"])/1000000
Bayrou=sum(tour1_2["Voix.7"])/1000000
Dupont=sum(tour1_2["Voix.8"])/1000000
Poutou=sum(tour1_2["Voix.4"])/1000000
HollandeT1=sum(tour1_2["Voix.9"])/1000000
SarkozyT1=sum(tour1_2["Voix.2"])/1000000
NulT1=sum(tour1_2["Blancs et nulsT1"])/1000000

In [25]: colors = ['b', 'c', 'y', 'm', 'r', 'g', 'c', 'yellowgreen', 'y', 'k', 'w']

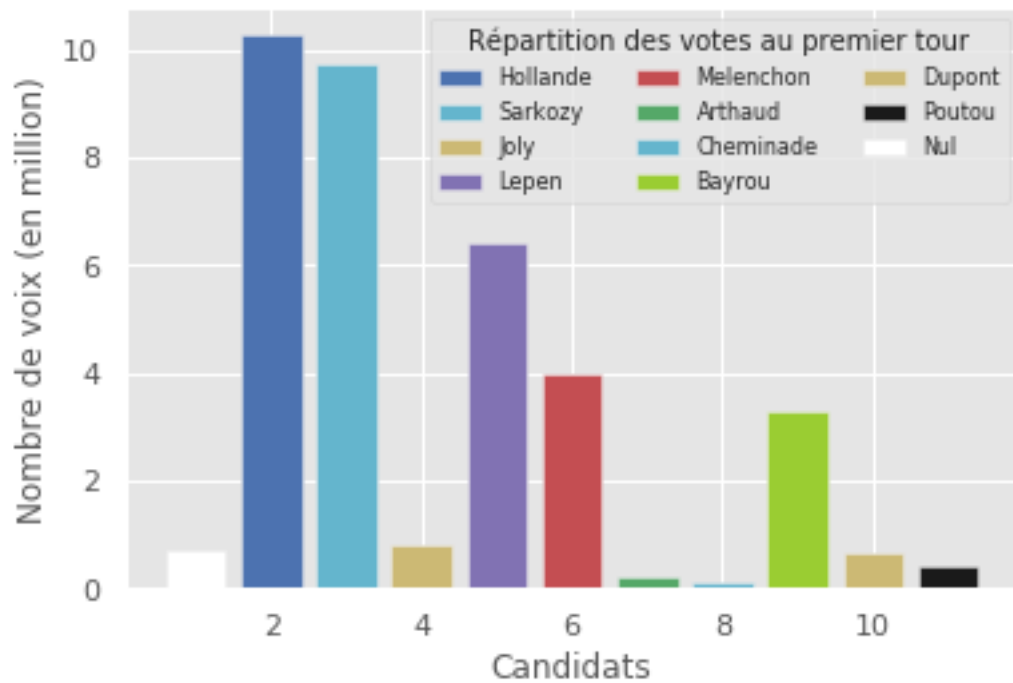
HT1 = plt.scatter(2, HollandeT1, marker='x', color=colors[0])
ST1 = plt.scatter(3, SarkozyT1, marker='o', color=colors[1])
JT1 = plt.scatter(4, Joly, marker='*', color=colors[2])
LT1 = plt.scatter(5, Lepen, marker='v', color=colors[3])
MT1 = plt.scatter(6, Melenchon, marker='P', color=colors[4])
AT1 = plt.scatter(7, Arthaud, marker='^', color=colors[5])
CT1 = plt.scatter(8, Cheminade, marker='H', color=colors[6])
BT1 = plt.scatter(9, Bayrou, marker='D', color=colors[7])
DT1 = plt.scatter(10, Dupont, marker='8', color=colors[8])
PT1 = plt.scatter(11, Poutou, marker='s', color=colors[9])
NT1 = plt.scatter(1, NulT1, marker='p', color=colors[10])
plt.legend((HT1, ST1, JT1, LT1, MT1, AT1, CT1, BT1, DT1, PT1, NT1),
           ('Hollande', 'Sarkozy', 'Joly', 'Lepen', 'Melenchon', 'Arthaud', 'Cheminade', 'B
           scatterpoints=1,
           loc='upper right',
           ncol=3,
           fontsize=8, title="Répartition des votes au premier tour")
plt.xlabel('Candidats')
plt.ylabel('Nombre de voix (en million)')
print("")
plt.savefig('templates/Répartition des votes au premier tour_scatter');

```



```
In [26]: colors = ['b', 'c', 'y', 'm', 'r', 'g', 'c', 'yellowgreen', 'y', 'k', 'w']
```

```
HT1 = plt.bar(2, HollandeT1, color=colors[0])
ST1 = plt.bar(3, SarkozyT1, color=colors[1])
JT1 = plt.bar(4, Joly, color=colors[2])
LT1 = plt.bar(5, Lepen, color=colors[3])
MT1 = plt.bar(6, Melenchon, color=colors[4])
AT1 = plt.bar(7, Arthaud, color=colors[5])
CT1 = plt.bar(8, Cheminade, color=colors[6])
BT1 = plt.bar(9, Bayrou, color=colors[7])
DT1 = plt.bar(10, Dupont, color=colors[8])
PT1 = plt.bar(11, Poutou, color=colors[9])
NT1 = plt.bar(1, NulT1, color=colors[10])
plt.legend((HT1, ST1, JT1, LT1, MT1, AT1, CT1, BT1, DT1, PT1, NT1),
           ('Hollande', 'Sarkozy', 'Joly', 'Lepen', 'Melenchon', 'Arthaud', 'Cheminade', 'Bayrou', 'Dupont', 'Poutou', 'Nul'),
           scatterpoints=1,
           loc='upper right',
           ncol=3,
           fontsize=8, title="Répartition des votes au premier tour")
plt.xlabel('Candidats')
plt.ylabel('Nombre de voix (en million)')
print("")
plt.savefig('templates/Répartition des votes au premier tour_plot');
```

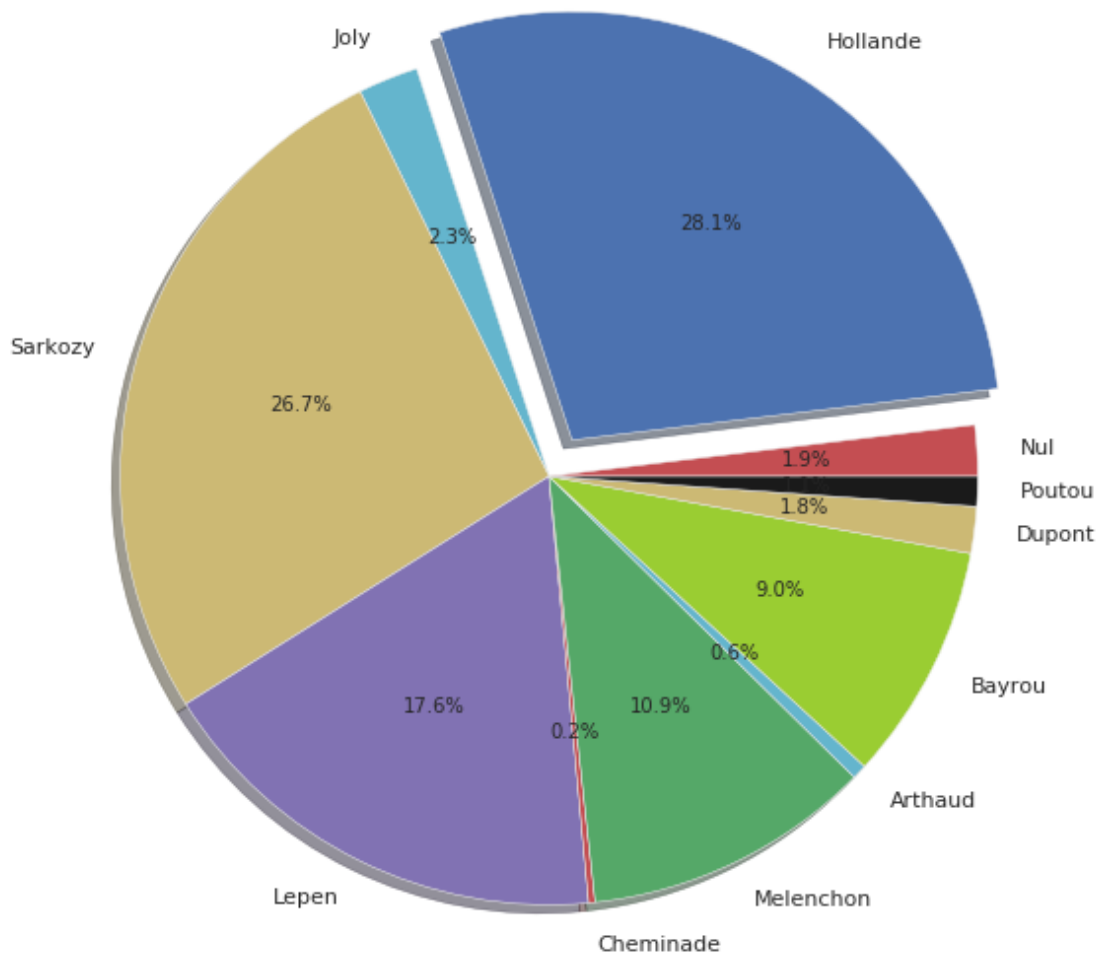


```
In [27]: colors = ['r','b', 'c', 'y', 'm', 'r', 'g', 'c', 'yellowgreen', 'y', 'k']
slices=[NulT1,HollandeT1,Joly,SarkozyT1,Lepen,Cheminade,Melenchon,Arthaud,Bayrou,Dupont,Poutou]
labels=['Nul','Hollande','Joly', 'Sarkozy','Lepen','Cheminade','Melenchon','Arthaud',

plt.pie (slices,labels=labels,colors=colors, shadow = True, autopct='%1.1f%%',explode=0)

plt.title("Répartition des votes au premier tour", fontsize=20)
plt.gcf().set_size_inches(10,10)
print("")
plt.savefig('templates/Répartition des votes au premier tour_pie');
```

## Répartition des votes au premier tour



```
In [28]: plt.boxplot([NulT1,HollandeT1,Joly,SarkozyT1,Lepen,Cheminade,Melenchon,Arthaud,BayrouT1])
plt.title("Répartition des votes au premier tour en million", fontsize=20)
plt.ylabel('Nbre de voix', fontsize=12)

print("Voix des candidats au premier tour")
df = pd.DataFrame({'En millions': [NulT1,HollandeT1,Joly,SarkozyT1,Lepen,Cheminade,Melenchon,Arthaud,BayrouT1]})
print(df.describe())
plt.savefig('templates/Répartition des votes au premier tour_boxplot')
```

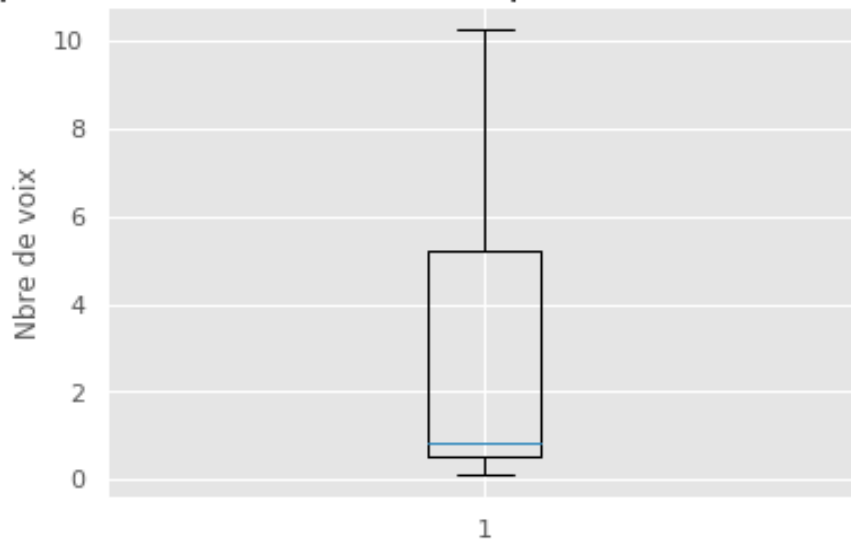
Voix des candidats au premier tour  
En millions

```

count    11.000000
mean      3.325854
std       3.855358
min       0.089545
25%       0.527533
50%       0.828345
75%       5.203124
max       10.272705

```

## Répartition des votes au premier tour en million



```

In [29]: chomage = pd.read_excel('source/chomage.xls',sheet_name = 'Département', index_col = 1)
chomage.rename(columns={'T2_2012':'Taux_chomage_T2_2012'}, inplace=True)

chomage_2012=chomage[['Code du département','Libellé', 'Taux_chomage_T2_2012' ]]

departement_elections_tour1=pd.read_excel('source/resultat-elections-2012.xls',sheet_name='Département', index_col=1)
departement_elections_tour2=pd.read_excel('source/resultat-elections-2012.xls',sheet_name='Département', index_col=1)
chomage_tour1 = departement_elections_tour1.merge(chomage_2012, left_on = 'Code du département', right_on = 'Code du département')
chomage_tour1.to_excel("source/temp/chomage_tour1.xlsx")
chomage_tou1_mini=chomage_tour1[['% Abs/Ins','% Tot/Ins', '% Exp/Ins', '% Exp/Vot', '% Tot/Vot']]
chomage_tou1_mini.to_excel("source/temp/chomage_tou1_mini.xlsx")
chomage_tour2 = departement_elections_tour2.merge(chomage_2012, left_on = 'Code du département', right_on = 'Code du département')
chomage_tour2.to_excel("source/temp/chomage_tour2.xlsx")

In [30]: print("Correlations entre votes et chômage au premier tour:")
sns.heatmap(chomage_tou1_mini.corr(method='pearson'),annot=True,cmap='RdYlGn',linewidth=1)
fig=plt.gcf()

```

```

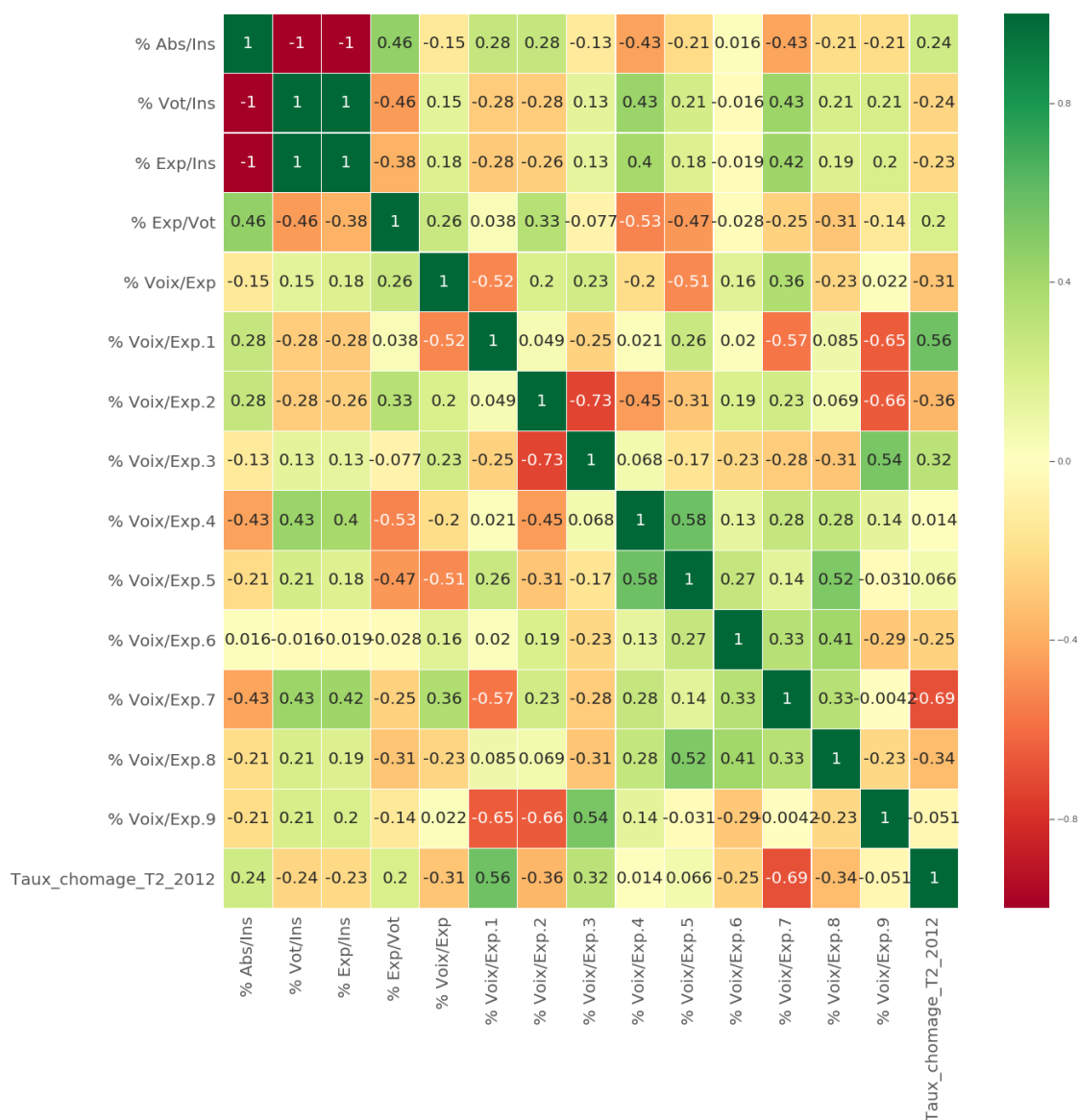
fig.set_size_inches(20,20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
chaine = "\nOn observe une forte corrélation entre le taux de chômage et \nle nombre de
print(chaine)
plt.savefig('templates/Correlations entre votes et chômage au premier tour')

```

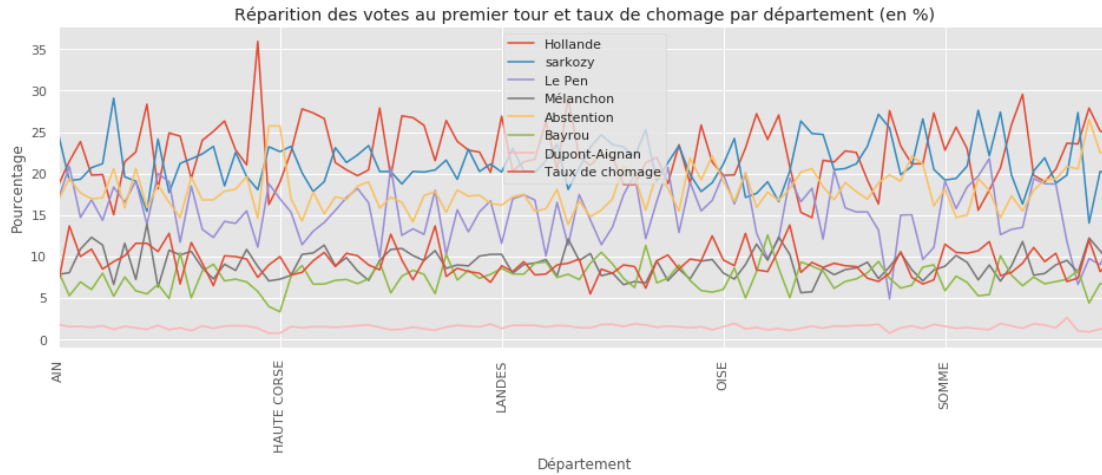
Correlations entre votes et chômage au premier tour:

On observe une forte corrélation entre le taux de chômage et le nombre de personnes ayant voté pour Le Pen, puis un peu moins pour Mélanchon et enfin pour l'abstention.

On remarque aussi que la corrélation est très négative entre le taux de chômage et Bayrou et deux fois moins avec Sarkozy et Dupont-Aignan au premier

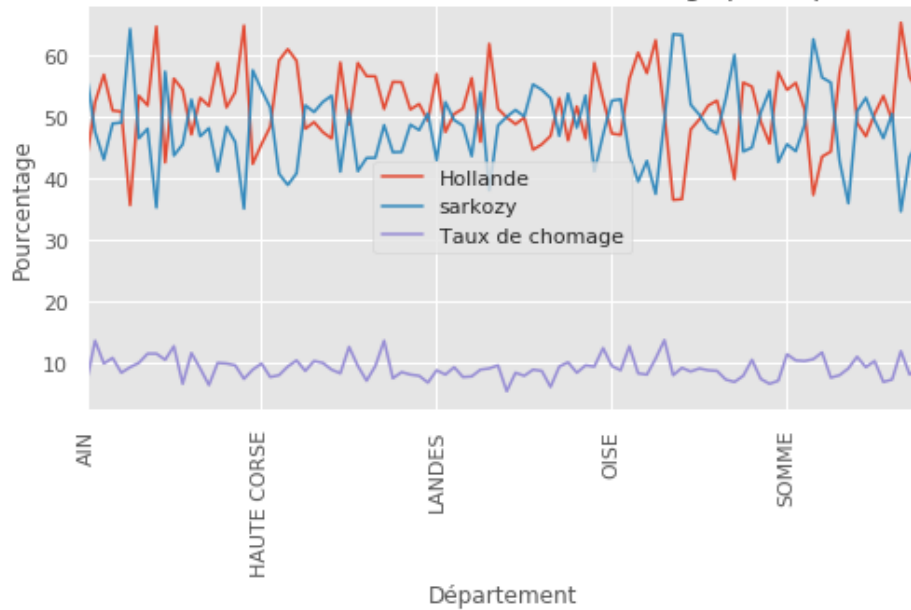


```
In [31]: chomage_tour1.plot(x="Libellé du département", y=["% Voix/Ins.9", "% Voix/Ins.2", "% Voix/Ins.1"],
```



```
In [32]: chomage_tour2.plot(x="Libellé du département", y=["% Voix/Exp", "% Voix/Exp.1", "Taux_ch
```

Répartition des votes au second tour et taux de chômage par département (en %)



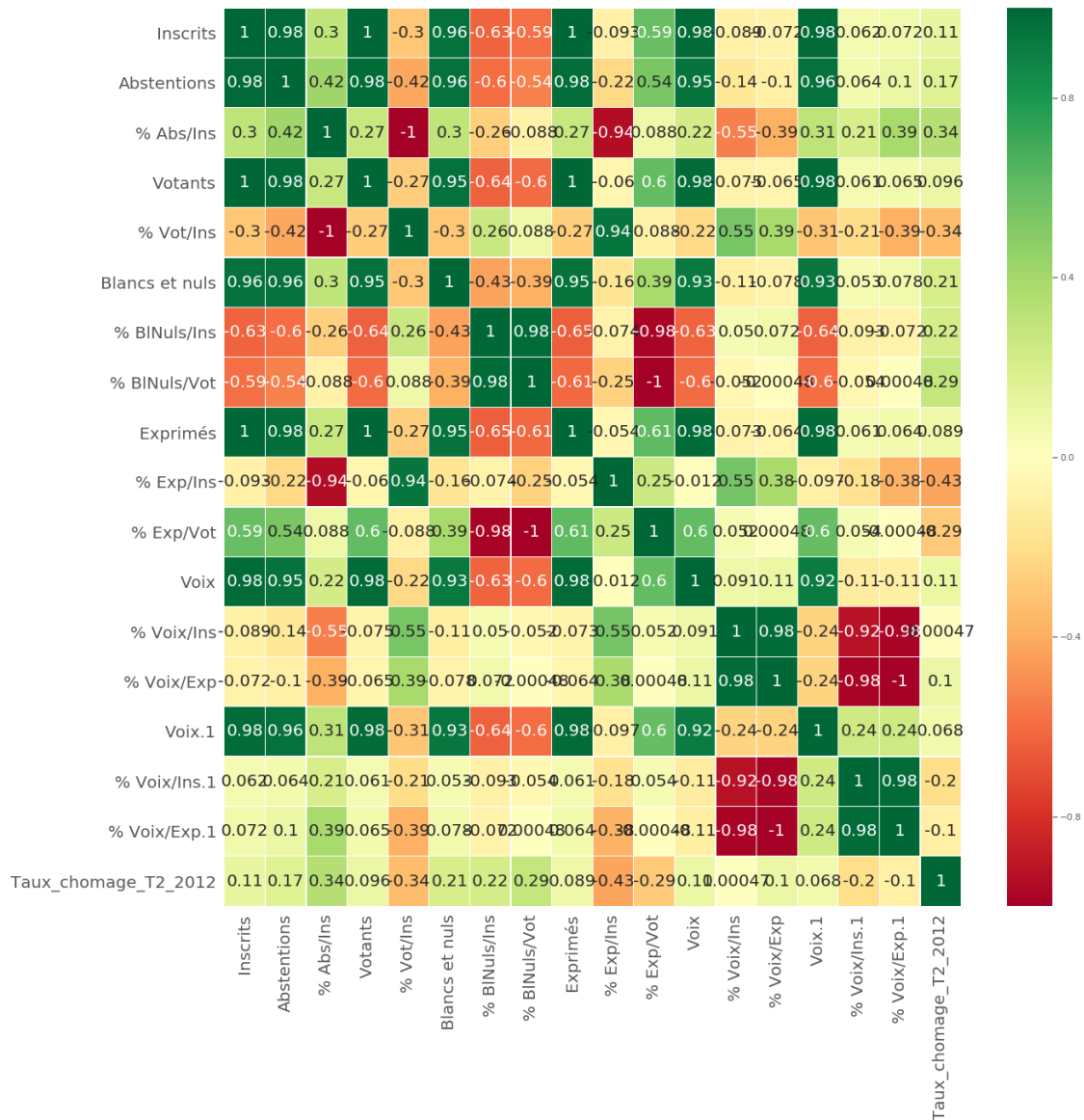
```
In [33]: print("Correlations entre votes et chômage au deuxième tour:")
sns.heatmap(chomage_tour2.corr(method='pearson'),annot=True,cmap='RdYlGn',linewidths=0)
fig=plt.gcf()
fig.set_size_inches(20,20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
chaine = "\nOn observe une légère corrélation entre le taux de chômage et \n\nl'abstention."
print(chaine)
plt.savefig('templates/Correlations entre votes et chômage deuxieme tour')
```

Correlations entre votes et chômage au deuxième tour:

On observe une légère corrélation entre le taux de chômage et l'abstention ainsi que les votes nuls.

On remarque aussi que la corrélation est deux fois plus forte entre le taux de chômage et Hollande qu'entre le taux de chômage et Sarkozy





```
In [34]: print("Tableau des corrélations entre votes et chomages")
          data.corr(method='pearson')
```

Tableau des corrélations entre votes et chomages

```
Out [34]:
```

	VotantsT1	rHollandeT1	rSarkozyT1	rNulT1	VotantsT2	\
VotantsT1	1.000000	-0.116703	-0.026674	-0.317862	0.999308	
rHollandeT1	-0.116703	1.000000	-0.267538	0.600723	-0.099533	
rSarkozyT1	-0.026674	-0.267538	1.000000	-0.036376	-0.019299	
rNulT1	-0.317862	0.600723	-0.036376	1.000000	-0.306548	
VotantsT2	0.999308	-0.099533	-0.019299	-0.306548	1.000000	

rHollandeT2	-0.084958	0.843918	-0.680546	0.421257	-0.076931
rSarkozyT2	0.084958	-0.843918	0.680546	-0.421257	0.076931
rNulT2	-0.304039	-0.188659	-0.427603	0.231829	-0.317031

	rHollandeT2	rSarkozyT2	rNulT2
VotantsT1	-0.084958	0.084958	-0.304039
rHollandeT1	0.843918	-0.843918	-0.188659
rSarkozyT1	-0.680546	0.680546	-0.427603
rNulT1	0.421257	-0.421257	0.231829
VotantsT2	-0.076931	0.076931	-0.317031
rHollandeT2	1.000000	-1.000000	0.073186
rSarkozyT2	-1.000000	1.000000	-0.073186
rNulT2	0.073186	-0.073186	1.000000

```
In [35]: tour2['% Voix/Exp'].isna().sum()
```

```
Out[35]: 1
```

```
In [36]: tour2['% Voix/Exp'].fillna(method='ffill', inplace=True)
```

```
In [37]: tour2['% Voix/Exp'].isna().sum()
```

```
Out[37]: 0
```

```
In [38]: tour2.to_excel("source/temp/tour2.xlsx")
```

```
In [39]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/d
state_data = pd.read_excel('source/temp/tour2.xlsx')
state_data = state_data.to_csv('source/temp/tour2.csv', encoding='utf-8')
state_data = pd.read_csv('source/temp/tour2.csv')
map1 = folium.Map(location=[48.866667, 2.333333],
                    zoom_start = 5)
```

```
folium.Choropleth(
    geo_data=state_geo,
    name='choropleth',
    data=state_data,
    columns=['Code du département', '% Voix/Exp'],
    key_on='feature.properties.code',
    fill_color='BuPu', nan_fill_color='white',
    fill_opacity=0.6, nan_fill_opacity=None,
    line_color='black', line_weight=1, line_opacity=1,
    legend_name='Pourcentage de vote pour Hollande au deuxième tour par département'
).add_to(map1)
```

```
folium.LayerControl().add_to(map1)
print("Pourcentage de vote pour Hollande au deuxième tour par département")
map1
```

Pourcentage de vote pour Hollande au deuxième tour par département

```
Out[39]: <folium.folium.Map at 0x7f85457c5978>
```

```
In [40]: map1.save('templates/map1.html')
```

```
In [41]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/r
state_data = pd.read_excel('source/temp/tour2.xlsx')
state_data = state_data.to_csv('source/temp/tour2.csv', encoding='utf-8')
state_data = pd.read_csv('source/temp/tour2.csv')
map1_1 = folium.Map(location=[48.866667, 2.333333],
                    zoom_start = 5)

folium.Choropleth(
    geo_data=state_geo,
    name='choropleth',
    data=state_data,
    columns=['Code du département', '% Voix/Exp'],
    key_on='feature.properties.code',
    fill_color='OrRd', nan_fill_color='white',
    fill_opacity=0.6, nan_fill_opacity=None,
    line_color='black', line_weight=1, line_opacity=1,
    legend_name='Pourcentage de vote pour Hollande, au deuxième tour, par région'
).add_to(map1_1)

folium.LayerControl().add_to(map1_1)
print("Pourcentage de vote pour Hollande, au deuxième tour, par région")
map1_1
```

Pourcentage de vote pour Hollande, au deuxième tour, par région

```
Out[41]: <folium.folium.Map at 0x7f85483e1f98>
```

```
In [42]: map1_1.save('templates/map1_1.html')
```

```
In [ ]:
```

```
In [43]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/d
state_data = pd.read_excel('source/temp/tour2.xlsx')
state_data = state_data.to_csv('source/temp/tour2.csv', encoding='utf-8')
state_data = pd.read_csv('source/temp/tour2.csv')
map1_2 = folium.Map(location=[48.866667, 2.333333],
                    zoom_start = 5)

folium.Choropleth(
    geo_data=state_geo,
    name='choropleth',
```

```

        data=state_data,
        columns=['Code du département', '% Voix/Exp.1'],
        key_on='feature.properties.code',
        fill_color='BuPu', nan_fill_color='white',
        fill_opacity=0.6, nan_fill_opacity=None,
        line_color='black', line_weight=1, line_opacity=1,
        legend_name='Pourcentage de vote pour Sarkozy au deuxième tour par département'
    ).add_to(map1_2)

    folium.LayerControl().add_to(map1_2)
    print("Pourcentage de vote pour Sarkozy au deuxième tour par département")
    map1_2

```

Pourcentage de vote pour Sarkozy au deuxième tour par département

Out[43]: <folium.folium.Map at 0x7f8544d743c8>

In [44]: map1\_2.save('templates/map1\_2.html')

```

In [45]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/r
state_data = pd.read_excel('source/temp/tour2.xlsx')
state_data = state_data.to_csv('source/temp/tour2.csv', encoding='utf-8')
state_data = pd.read_csv('source/temp/tour2.csv')
map1_3 = folium.Map(location=[48.866667, 2.333333],
                    zoom_start = 5)

```

```

    folium.Choropleth(
        geo_data=state_geo,
        name='choropleth',
        data=state_data,
        columns=['Code du département', '% Voix/Exp.1'],
        key_on='feature.properties.code',
        fill_color='GnBu', nan_fill_color='white',
        fill_opacity=0.6, nan_fill_opacity=None,
        line_color='black', line_weight=1, line_opacity=1,
        legend_name='Pourcentage de vote pour Sarkozy au deuxième tour par région'
    ).add_to(map1_3)

    folium.LayerControl().add_to(map1_3)
    print("Pourcentage de vote pour Sarkozy au deuxième tour par région")
    map1_3

```

Pourcentage de vote pour Sarkozy au deuxième tour par région

Out[45]: <folium.folium.Map at 0x7f8544e1dac8>

In [46]: map1\_3.save('templates/map1\_3.html')

```
In [ ]:
```

```
In [47]: tour2['diff']=tour2['Voix']-tour2['Voix.1']
```

```
In [48]: def win(diff):  
    if diff > 0:  
        res = 1  
    else:  
        res = 0  
    return res  
tour2['Winner'] = list(map(win, tour2['diff']))
```

```
In [49]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/d  
#state_data = pd.read_excel('source/temp/tour2.xlsx')  
state_data = tour2.to_csv('source/temp/tour2.csv', encoding='utf-8')  
state_data = pd.read_csv('source/temp/tour2.csv')  
map2 = folium.Map(location=[48.866667, 2.333333],  
                    zoom_start = 5)  
  
folium.Choropleth(  
    geo_data=state_geo,  
    name='choropleth',  
    data=state_data,  
    columns=['Code du département', 'Winner'],  
    key_on='feature.properties.code',  
    fill_color='BuPu', nan_fill_color='purple',  
    fill_opacity=0.6, nan_fill_opacity=None,  
    line_color='black', line_weight=1, line_opacity=1,  
    legend_name='Votes pour Sarkozy (Blanc) ou Hollande (violet) au deuxieme tour par  
) .add_to(map2)  
  
folium.LayerControl().add_to(map2)  
print("Votes pour Sarkozy (Blanc) ou Hollande (violet) au deuxieme tour par département  
map2
```

Votes pour Sarkozy (Blanc) ou Hollande (violet) au deuxieme tour par département

```
Out[49]: <folium.folium.Map at 0x7f8544eba0b8>
```

```
In [50]: map2.save('templates/map2.html')
```

```
In [51]: state_geo = f'https://raw.githubusercontent.com/gregoireddavid/france-geojson/master/r  
#state_data = pd.read_excel('source/temp/tour2.xlsx')  
state_data = tour2.to_csv('source/temp/tour2.csv', encoding='utf-8')  
state_data = pd.read_csv('source/temp/tour2.csv')  
map3 = folium.Map(location=[48.866667, 2.333333],  
                    zoom_start = 5)
```

```

folium.Choropleth(
    geo_data=state_geo,
    name='choropleth',
    data=state_data,
    columns=['Code du département', 'Winner'],
    key_on='feature.properties.code',
    fill_color='YlGn', nan_fill_color='black',
    fill_opacity=0.6, nan_fill_opacity=None,
    line_color='black', line_weight=1, line_opacity=1,
    legend_name='Votes pour Sarkozy (Blanc) ou Hollande (vert) au deuxieme tour par r
).add_to(map3)

folium.LayerControl().add_to(map3)
print("Votes pour Sarkozy (Blanc) ou Hollande (vert) au deuxieme tour par région")
map3

```

Votes pour Sarkozy (Blanc) ou Hollande (vert) au deuxieme tour par région

Out[51]: <folium.folium.Map at 0x7f8544e16630>

In [52]: map3.save('templates/map3.html')

In [53]: webbrowser.open('templates/map1.html')  
webbrowser.open('templates/map1\_1.html')  
webbrowser.open('templates/map1\_2.html')  
webbrowser.open('templates/map1\_3.html')  
webbrowser.open('templates/map2.html')  
webbrowser.open('templates/map3.html')

Out[53]: True

Créer PDF

A executer sur le terminal sudo apt-get install texlive-xetex sudo apt-get install pandoc  
pip install nbconvert # OR conda install nbconvert EXECUTER CETTE LIGNE SUR LE TERMI-  
NAL jupyter nbconvert --to pdf --TemplateExporter.exclude\_input=False "Projet groupe élections  
présidentielles.ipynb"

In [ ]: