

数据科学导论大作业

Student name: 张晨阳 171180524

Course: 数据科学导论

Due date: 2019 年 11 月 24 日

1. 数据来源

本报告所选取数据来源于Kaggle数据集,¹该数据集包含美国各县的基本社会情况（种族结构，人口，经济等）及其在2012和2016年美国总统大选的投票数据。本次实践对美国各县的社会情况进行了一定的分析，并以此对总统大选结果进行分析与预测。

作业压缩包中所含文件说明

说明文档.pdf	本文件
.py文件	源代码
.shx .shp .dbf文件	绘制地图所需的地理信息文件

2. 预处理

预处理代码位于文件pretreatment.py，定义了函数pretreatment()进行预处理。具体步骤如下。

2.1. 读取文件。采用pd.read_csv()读取文件

```
raw_data = pd.read_csv('../votes.csv')
```

2.2. 特征选择。原始数据的特征十分丰富，共有82列，每列特征具体含义可参见文件county_facts_dictionary.csv。为了简化分析，选取了其中具有代表性一部分的数据，这一选取也是基于美国的社会文化情况做出的，例如因为种族和性别在美国社会中是很重要的议题，所以选取了大量种族结构和性别结构的特征。每列特征具体含义如下：

¹<https://www.kaggle.com/joelwilson/2012-2016-presidential-elections>

state_fips	州编号	votes_dem_2012	奥巴马得票数
state_abbr	州名缩写	votes_gop_2012	罗姆尼得票数
county_name	县名	age65plus	年龄65岁以上人口比例
population2014	2014年该县总人口	SEX255214	女性人口比例
votes_dem_2016	希拉里得票数	White	白人人口比例
votes_gop_2016	特朗普得票数	Black	黑人人口比例
Clinton	希拉里得票率	Hispanic	西班牙裔人口比例
Trump	特朗普得票率	RHI425214	亚裔人口比例
Edu_batchelors	具有学士学位及以上人口比例	Income	人均收入（年）
Poverty	贫困人口比例		

在这里可能存在着冗余的数据，比如得票数和得票率，但是因为后续的分析中有时需要用到得票数，有时需要用到得票率，且仅有两人的得票数算出的得票率与真实情况会存在一定偏差（由于还有其他政党及独立候选人的存在），所以两列数据都保留了。另外得票率在某种程度上也反映了该地区对这名候选人的支持率，下文统一用得票率进行称呼。

之后对一些列名进行了重命名使其意义更清晰，将百分数转为小数与其他数据统一格式，并以county_name作为新的索引。

```
columns=['state_fips','county_fips','state_abbr','county_name','
          population2014','votes_dem_2016','
          votes_gop_2016','Clinton','Trump',
          'votes_dem_2012','votes_gop_2012','age65plus','SEX255214','White',
          'Black','Hispanic','RHI425214','Edu_batchelors','Income','Poverty']
data = raw_data[columns]
data = data.rename(columns={'SEX255214':'Female'})
data[['Female','Asian','age65plus','Edu_batchelors']] /= 100
data = data.set_index('county_name')
```

2.3. 检查数据缺失. 因无数据缺失，故不用进行数据缺失处理

```
print(data.isnull().sum())
#data = data.dropna()
```

2.4. 建立新列. 根据投票结果可以确定谁在该县赢得选举，新增两列result_2016,result_2012表示选举结果，1代表民主党候选人获胜（希拉里，奥巴马），0代表共和党候选人获胜（特朗普，罗姆尼）

```
result_2016 =[1 if x > y else 0 for x, y in np.array(data[['  
                votes_dem_2016', 'votes_gop_2016'  
                ]])]
result_2012 =[1 if x > y else 0 for x, y in np.array(data[['  
                votes_dem_2012', 'votes_gop_2012'  
                ]])]
data.loc[:, 'result_2016'] = result_2016
data.loc[:, 'result_2012'] = result_2012
```

2.5. 排序. 同一个州内按照人口从多到少将县进行排序

```
data = data.sort_values(by = ['state_fips', 'population2014'],  
                        ascending = [True, False])
```

3. EDA 探索性数据分析

该部分代码位于文件eda.py

3.1. 查看数据基本信息. :

```
print('Basic information of the datasets after pretreatment:')
print(data.info())
print(data.describe())
```

```

Basic information of the datasets after pretreatment:
<class 'pandas.core.frame.DataFrame'>
Index: 3112 entries, Jefferson County to Niobrara County
Data columns (total 20 columns):
state_fips      3112 non-null int64
state_abbr      3112 non-null object
population2014  3112 non-null int64
votes_dem_2016  3112 non-null int64
votes_gop_2016  3112 non-null int64
Clinton         3112 non-null float64
Trump           3112 non-null float64
votes_dem_2012  3112 non-null int64
votes_gop_2012  3112 non-null int64
age65plus       3112 non-null float64
Female          3112 non-null float64
White           3112 non-null float64
Black           3112 non-null float64
Hispanic        3112 non-null float64
Asian           3112 non-null float64
Edu_batchelors  3112 non-null float64
Income          3112 non-null int64
Poverty         3112 non-null float64
result_2016     3112 non-null int64
result_2012     3112 non-null int64
dtypes: float64(10), int64(9), object(1)
memory usage: 510.6+ KB

```

图 1: 数据集基本信息-1

```
sns.heatmap(data[heatmap_col].corr(), annot = True)
```

	state_fips	population2014	votes_dem_2016	votes_gop_2016	Clinton	Trump
count	3112.000000	3.112000e+03	3.112000e+03	3112.000000	3112.000000	3112.000000
mean	30.548522	1.022237e+05	2.001874e+04	19600.109576	0.317070	0.636152
std	14.965305	3.276072e+05	7.190185e+04	40362.196846	0.153578	0.156499
min	1.000000	8.600000e+01	4.000000e+00	57.000000	0.031447	0.041221
25%	19.000000	1.115775e+04	1.166000e+03	3206.000000	0.204759	0.549478
50%	29.000000	2.596100e+04	3.153000e+03	7164.500000	0.284739	0.667431
75%	46.000000	6.822550e+04	9.599750e+03	17427.250000	0.399610	0.751471
max	56.000000	1.011670e+07	1.893770e+06	620285.000000	0.928466	0.952727

图 2: 数据集基本信息-2

3.2. 查看数据相关性. 通过seaborn库的热力图可以比较直观的表现数据间的相关关系

```

heatmap_col = ['Clinton', 'Trump', 'population2014', 'age65plus', '
               Female', 'White', 'Black',
               'Hispanic', 'Edu_batchelors', 'Income', 'Poverty']
plt.figure(figsize = (13, 10), dpi = 100)
sns.heatmap(data[heatmap_col].corr(), annot = True)
plt.title('correlation heatmap')

```

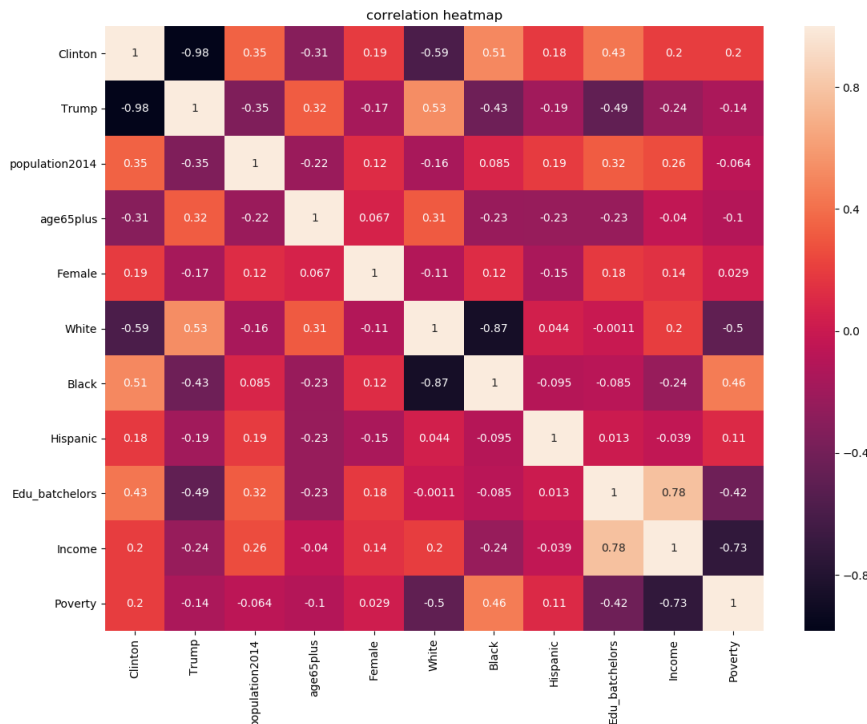


图 3: 相关性热力图

从图中可以看出：

1. 部分属性之间存在着较强的相关关系，比如希拉里得票率和特朗普得票率之间负相关关系（这是必然的），还有人均收入与贫困率之间的负相关关系，这一情况也是符合常理的。
2. 与种族有关的属性中，白人比例，黑人比例与得票率之间也有着比较强的相关关系，虽然0.5的相关系数在统计中不能说是很强，但考虑到选举受多种社会因素影响的，仅仅一个种族结构的特征有0.5的相关系数已经是比较显著了，这也印证了种族在美国政治中的重要性。
3. 女性与希拉里得票率之间的相关关系仅有0.19，比预期要低很多，一方面女性传统上倾向于支持民主党候选人，另一方面希拉里本身也是女性，理应更受女性欢迎，因此这一结果令人出乎意料。
4. 人口数量，受高等教育程度（学士学位拥有者）与希拉里得票率之间有着弱相关关系，这也是比较符合常理的。（大城市，高学历者倾向于支持民主党）

从heatmap中找出相关系数较大的几个特征，通过pairplot作多变量图进一步观察数据间的关系：

```
pairplot_col = ['Clinton', 'Trump', 'population2014',
               'age65plus', 'White', 'Black', 'Edu_batchelors']
sns.pairplot(data[pairplot_col], diag_kind = 'kde',
```

```
plot_kws = {'alpha': 0.2})
```

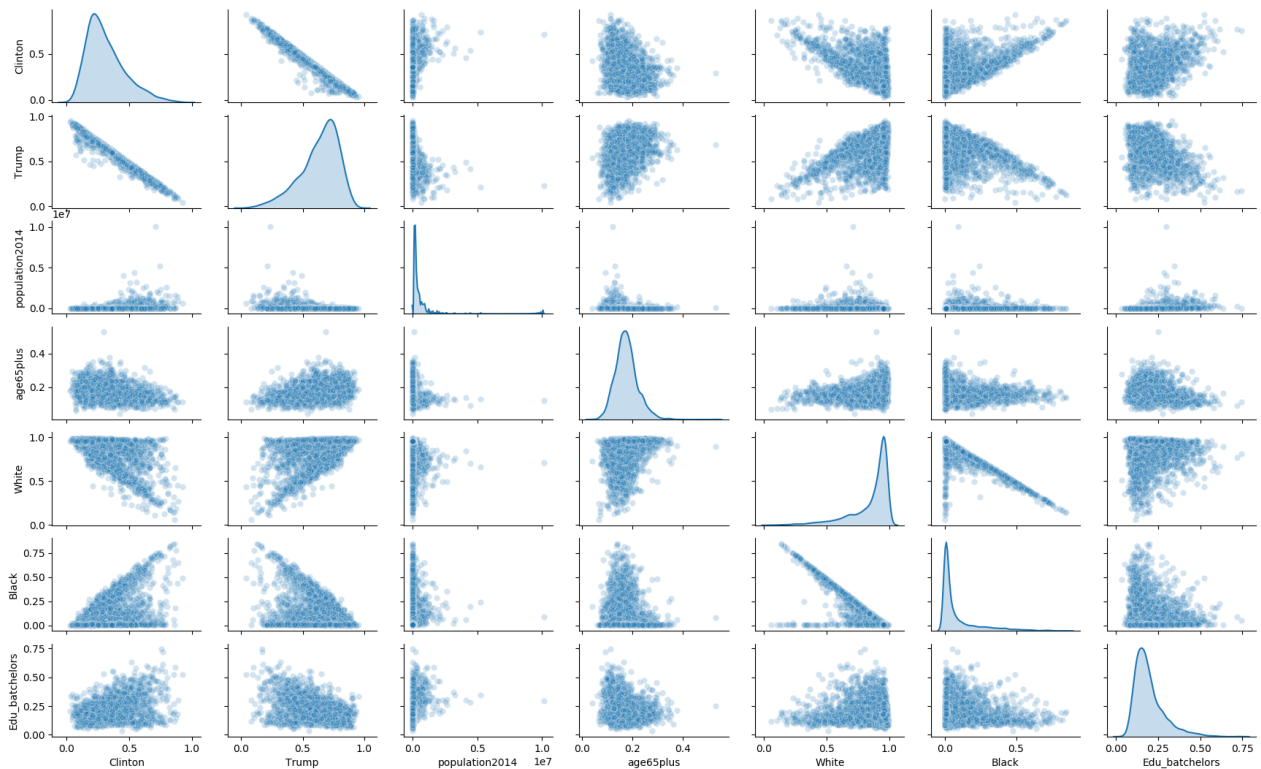


图 4: 多变量图

从图中我们发现，或许是数据量太大的原因，许多特征间并没有很清晰的线性关系，在一些图中散点图是一团一团的，在一些图散点图呈现一种三角形状，即在 $x = 0$ 和 $y = 0$ 附近聚集了大量离群值，除此之外呈现着线性关系。不过如果我们将数据细分到每个州，还是可以发现线性关系的，之后会进行展示。

3.3. 观察特征分布情况. 这里选用了箱型图观察美国各县的部分人群结构特征，选用箱型图的原因是因为可以比较直观的看出数据分散情况，分布是否不均衡。

```
plt.subplot(241)
data[['Female']].boxplot()
plt.subplot(242)
data[['White']].boxplot()
plt.subplot(243)
data[['Black']].boxplot()
plt.subplot(244)
data[['Hispanic']].boxplot()
plt.subplot(245)
data[['age65plus']].boxplot()
plt.subplot(246)
data[['Edu_batchelors']].boxplot()
```

```
plt.subplot(247)
data[['Income']].boxplot()
plt.subplot(248)
data[['Poverty']].boxplot()
```

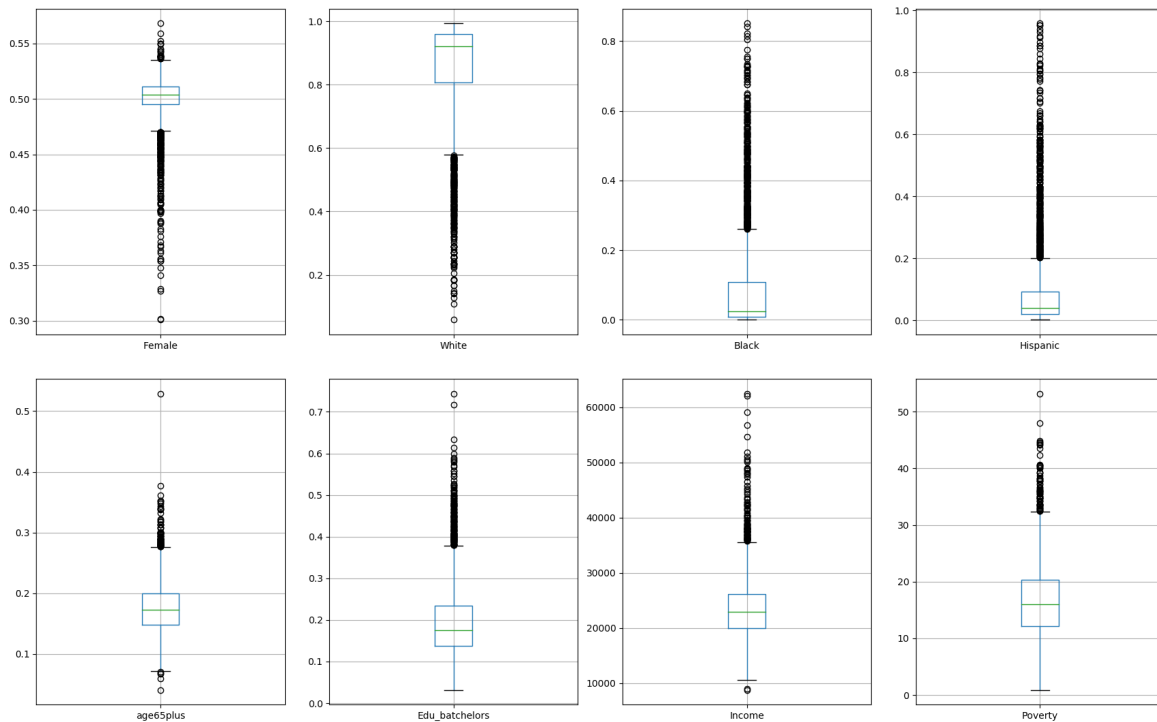


图 5: 人群结构特征箱型图

从图中我们看出，以上特征均呈现分布不均衡的现象，有着大量离群值

1. 女性比例的箱型图还是比较正常的，大部分数据集集中在0.5左右，两侧各有部分离群值。
2. 白人属于多数族裔，在绝大部分县市都占大多数，黑人、西班牙裔在绝大多数县市占比较少，但在另一侧存在大量离群值，这也符合美国的种族结构。
3. 从年龄分布可以看出美国存在一定的老龄化危机（国际上认为65岁以上人口占比超过7%就存在老龄化问题）。
4. 美国同样存在一定程度的贫富差距

3.4. 封装打印统计特性的函数。 这里定义了函数`print_result`,可以打印出数据指定属性的各个统计特性，例如平均值，方差，中位数等

```
def print_result(data, attribute):
    for index in attribute:
        print('Some statistical properties of %s:' % index)
```



```

print('=====')
print('mean:', data[[index]].mean()[0])
print('median:', data[[index]].median()[0])
print('first quartile:', data[[index]].quantile(q = 0.25)[0])
print('third quartile:', data[[index]].quantile(q = 0.75)[0])
print('mode', data[[index]].mode().values[0, 0])
print('varying range:', data[[index]].min()[0], '~', data[[index]].max()[0])

print('max-min:', data[[index]].max()[0]-data[[index]].min()[0])
print('standard deviation:', data[[index]].std()[0])

```

以人口为例，打印出人口数量的各项统计描述结果。

```
print_result(data, ['population2014'])
```

```

Some statistical properties of population2014:
=====
mean: 102223.72589974293
median: 25961.0
first quartile: 11157.75
third quartile: 68225.5
mode 1396
varying range: 86 ~ 10116705
max-min: 10116619
standard deviation: 327607.2325530627
=====

```

图 6: 人口数量描述性统计结果

由此我们看出，美国人口的分布是非常的不均衡，第三四分位数接近7万人，而最大值却为1000万人，通过箱型图我们会有更加直观的感受。

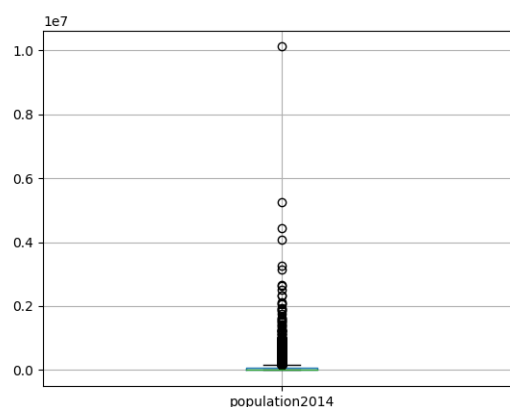
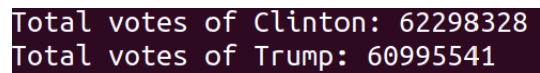


图 7: 人口数量箱型图

箱型图几乎都被压扁了，这说明在绝大多数县市，人口数量都是很稀少的，大部分人口集中在大城市，这一结论可以用来解释下一节发生的现象。

3.5. 查看希拉里和特朗普获得总票数。 对于选举来说，大家一定比较关心两位候选人最终得票数如何，以下结果会给出令人意想不到的事实。



```
Total votes of Clinton: 62298328
Total votes of Trump: 60995541
```

图 8: 得票结果

希拉里的总票数比特朗普多！ 可以为什么现在的美国总统是特朗普呢，这是美国的选举制度导致的。美国的总统选举并不是直接普选，而是根据各州的人口和经济状况等分配给每个州一定票数，称为“选举人票”，赢得该州普选（即在该州得票最多）的候选人获得该州的全部选举人票，最后获得选举人票数最多的候选人即为新一任美国总统。赢了总票数却输了选举人票这一情况在美国发生过很多次，上一次发生在2000年（戈尔vs.小布什）。对于发生这一情况的原因，我做出的合理推测是，因为希拉里在一些州大幅度领先特朗普（称为深蓝州，例如加州，纽约州），但是无论她领先多少，她在该州获得的选举人票数是不变的，而在一些州特朗普以微弱优势赢了希拉里（称为摇摆州，例如：密歇根州，威斯康辛州），最终特朗普赢得了多数选举人票。

3.6. 绘制选举结果地图。 因为数据集中的信息是基于地理位置的，因此通过地图可以比较直观地看出选举结果与地理位置的关系。

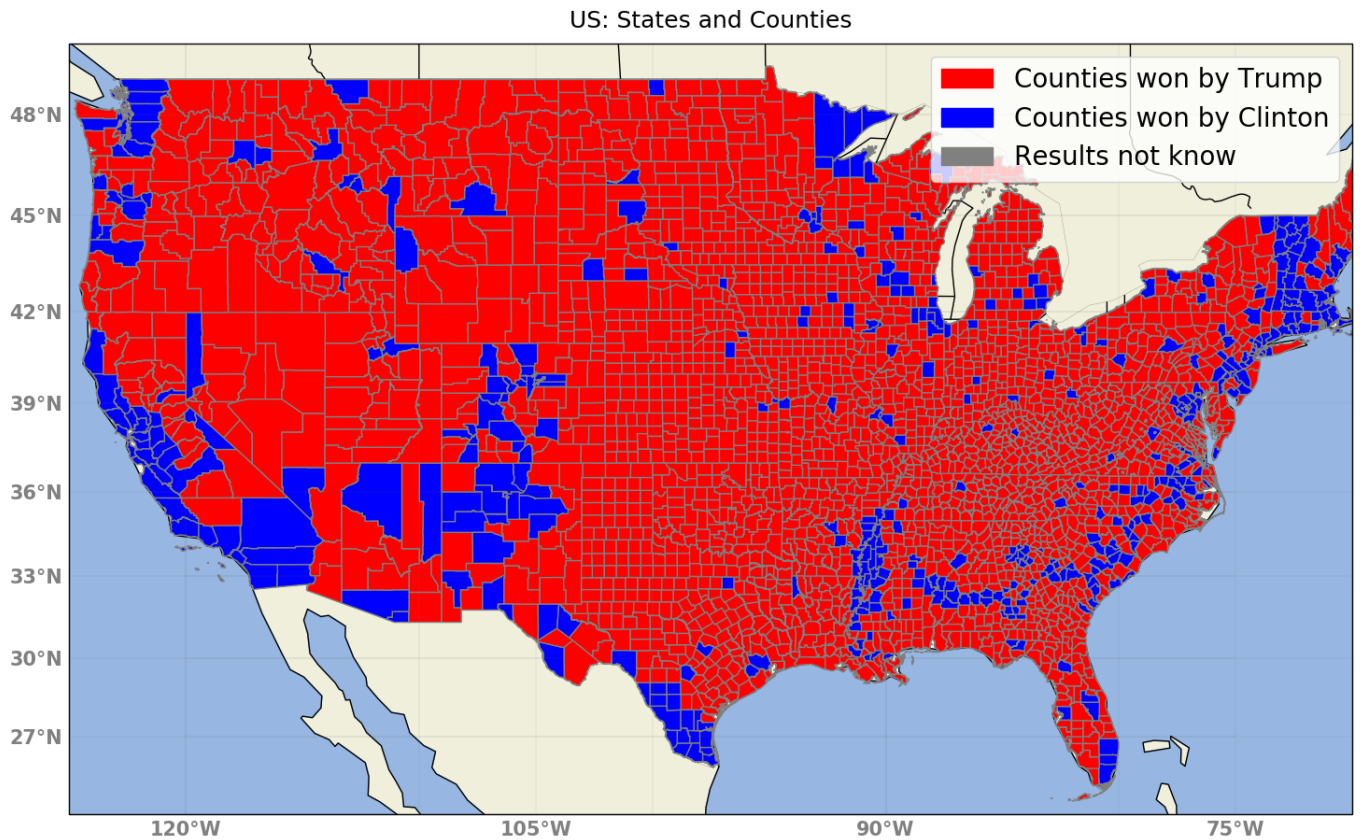


图 9: 选举结果地图

从图中我们可以很明显地看出以下几个特征：（1）红色区域远远多于蓝色区域，也就是特朗普在绝大多数县都获得了多数票数。（2）蓝色区域主要集中在沿海等经济发达地区，尤其西南部的加利福尼亚以及东北沿海（纽约州等）。

由此可以得出两个初步结论：

1. 美国人口分布很不均衡，有相当大部分人口集中在人口发达地区，而广大的内陆地区人口较少（与前几节中得到的结论相契合）
2. 经济越发达的地区越倾向于支持希拉里，广大农业地区倾向于支持特朗普

3.7. 绘制人口分布地图。 图中颜色越深，表示人口越多。

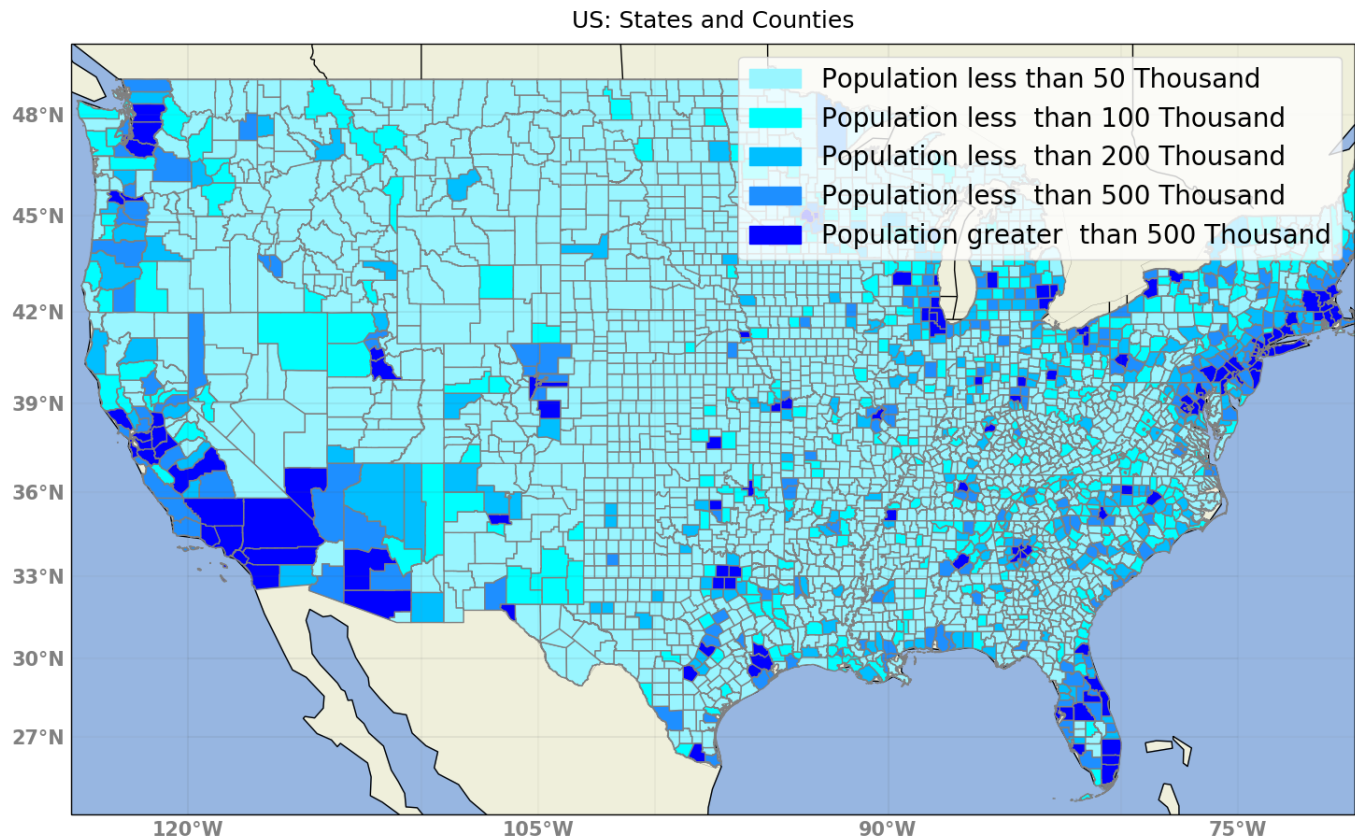


图 10: 人口分布地图

将此图与选举地图相对比，我们发现两者具有较好的重叠性，即人口越多的地方相应着希拉里获胜的地方，这也进一步印证了我们的结论。同样我们也可以绘制其他特征的分布地图，由于篇幅限制，这里就不一一列出了。但是我们从图中仅能得出一些较为直观的感受，还需要通过具体数据进行进一步验证。

3.8. 根据前述探索和已有经验，我们认为大城市（人口数量多），黑人，受过高等教育者更加支持希拉里，而白人比较不支持希拉里。

通过探索各属性显著的前100个县中希拉里获胜的比例做出进一步结论

```
print(data.sort_values(by = 'population2014', ascending = False).
      head(100).result_2016.sum()/100)

print(data.sort_values(by = 'population2014',
      ascending = False).head(100).population2014.sum()/data.
      population2014.sum())

print(data.sort_values(by = 'population2014', ascending = False).
      tail(100).result_2016.sum()/100)

print(data.sort_values(by = 'Female', ascending = False).head(100).
      result_2016.sum()/100)
```

```
print(data.sort_values(by = 'Black', ascending = False).head(100).
      result_2016.sum()/100)

print(data.sort_values(by = 'White', ascending = False).head(100).
      result_2016.sum()/100)

print(data.sort_values(by = 'Edu_batchelors', ascending = False).
      head(100).result_2016.sum()/100)
```

```
print(data.sort_values(by = 'population2014',
                      ascending = False).head(300).population2014.sum()/data.
                      population2014.sum())

county_population300 = data.sort_values(by = 'population2014',
                                       ascending = False).head(300)

Clinton_bigwin = [1 if x - y >= 0.3 else 0 for x,y in np.array(
                  county_population300[['Clinton',
                                       'Trump']])]

Trump_bigwin = [1 if y - x >= 0.3 else 0 for x,y in np.array(
                county_population300[['Clinton',
                                     'Trump']])]

print('Among the 300 most populous cities:')
print('Num of counties Clinton won:', county_population300.
      result_2016.sum())

print('Num of counties Clinton won 30%:',sum(Clinton_bigwin))
```

```
人数最多的100个县中希拉里获胜的比例:
0.88
人数最多的100个县占美国总人口的比例:
0.42620777015331957
人数最少的100个县中希拉里获胜的比例:
0.06
女性比例最多的100个县中希拉里获胜的比例:
0.58
黑人比例最多的100个县中希拉里获胜的比例:
0.95
白人比例最多的100个县中希拉里获胜的比例:
0.02
受教育程度最高的100个县中希拉里获胜的比例:
0.78
```

(a)

```
人口最多的300个县占总人口比例:
0.6558039132594001
Among the 300 most populous cities:
Num of counties Clinton won: 184
Num of counties Clinton won 30%: 59
Num of counties Trump won 30%: 18
```

(b)

希拉里在人口最多的100个县（占总人口43%）中赢了88%, 300个县（占总人口66%）中赢了60%多，其中有59个县大比例（30%）领先特朗普，而特朗普仅有18个县大幅领先希拉里。这也印证了希拉里在大城市获得了压倒性支持并导致总票数高于特朗普。

3.9. 细分数据. 我们希望从州一级的数据中得到一些更清晰的结论。首先，州一级的数据应该更具有相关性，因为美国各州的经济、文化、政治等方面有着较大差异，若将全

美国的数据放在一起由于多种因素的互相影响，我们不能得到比较清晰的结果，正如我们从pairplot图中观察到的。我们从总数据中选出三类数据，摇摆州数据，深红州数据，深蓝州数据。

```
swing_state = ['MI', 'OH', 'PA', 'WI', 'FL']
blue_state = ['CA', 'NY', 'MA']
red_state = ['AL', 'TX', 'TN']
data_swing = data[data['state_abbr'].isin(swing_state)]
data_red = data[data['state_abbr'].isin(red_state)]
data_blue = data[data['state_abbr'].isin(blue_state)]
```

3.10. 相关关系. 通过探索发现，在深蓝州和摇摆州，受高等教育程度与希拉里得票率之间有着比较清晰的线性关系。这一发现可以在后续用来进行线性回归任务。

```
plt.scatter(data_swing['Edu_batchelors'], data_swing['Clinton'], color=[1,0,1], alpha=0.5)
plt.scatter(data_red['Edu_batchelors'], data_red['Clinton'], color='r', alpha=0.5)
plt.scatter(data_blue['Edu_batchelors'], data_blue['Clinton'], color=[0,0,1], alpha=0.5)
plt.legend(['swing', 'red', 'blue'], fontsize = 15)
plt.xlabel('Batchelors ppercent')
plt.ylabel('Vote rate of Clinton')
```

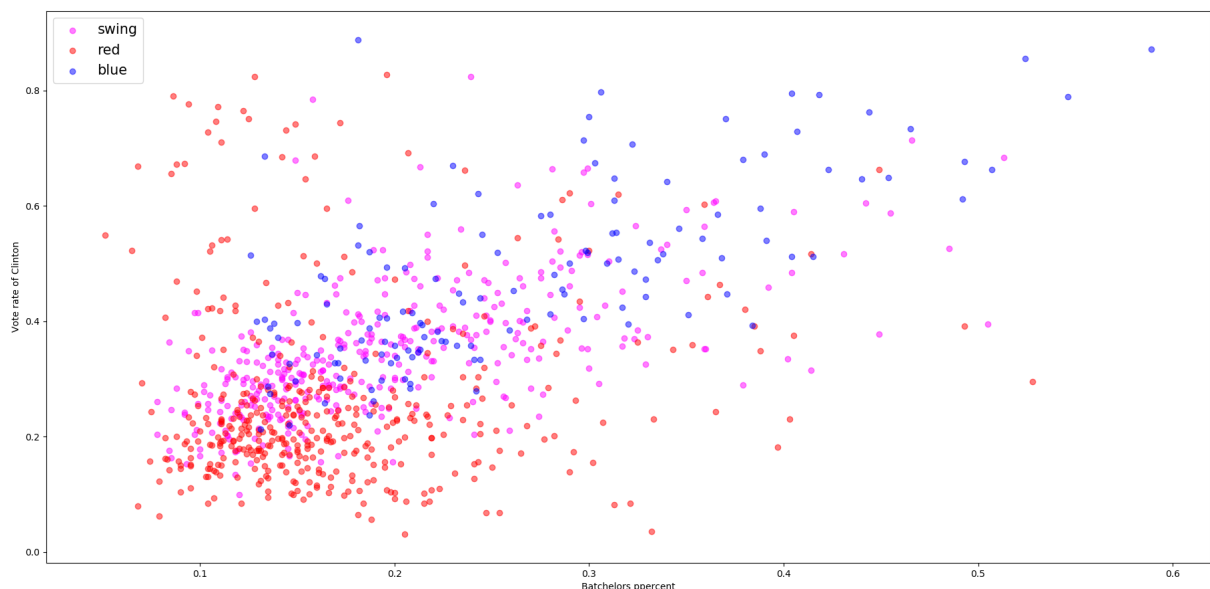


图 11: 受高度教育程度与希拉里得票率散点图

深红州的数据呈现一团团的，有可能pairplot中的大量离群值是由于深红州数据造成的（猜测）。

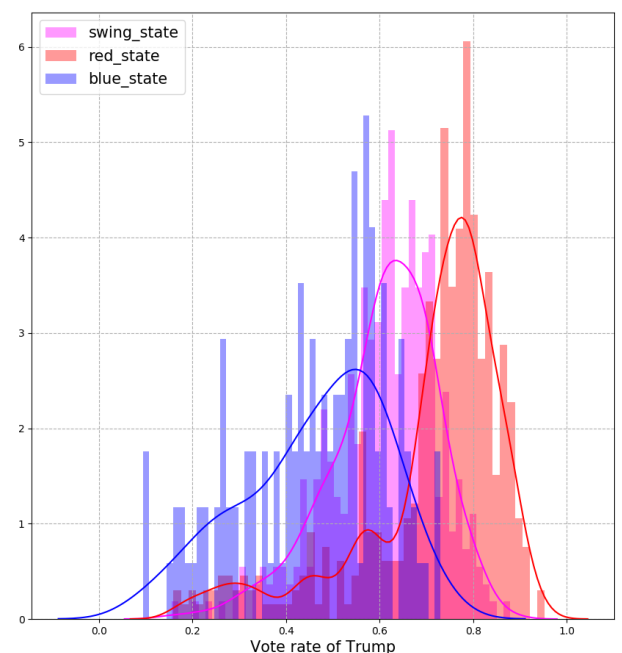
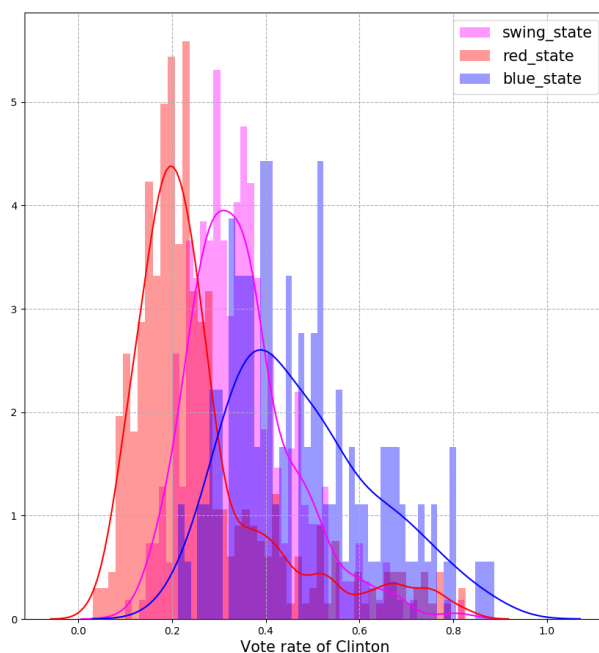
3.11. 得票率直方图. :

```
plt.subplot(121)
sns.distplot(data_swing['Clinton'], bins = 50, kde_kws = {'color': [1,0,1]}, color = [1,0,1])
sns.distplot(data_red['Clinton'], bins = 50, kde_kws={"color": "r"}, color = 'r')
sns.distplot(data_blue['Clinton'], bins = 50, kde_kws={"color": [0,0,1]}, color = [0,0,1])
plt.legend(['swing_state', 'red_state', 'blue_state'], fontsize = 15)

plt.grid(linestyle = '--')
plt.xlabel('Vote rate of Clinton', fontsize = 15)

plt.subplot(122)
sns.distplot(data_swing['Trump'], bins = 50, kde_kws = {'color': [1,0,1]}, color = [1,0,1])
sns.distplot(data_red['Trump'], bins = 50, kde_kws={"color": "r"}, color = 'r')
sns.distplot(data_blue['Trump'], bins = 50, kde_kws={"color": [0,0,1]}, color = [0,0,1])
plt.legend(['swing_state', 'red_state', 'blue_state'], fontsize = 15)

plt.grid(linestyle = '--')
plt.xlabel('Vote rate of Trump', fontsize = 15)
```



从图中看出:

1. 深蓝州更加支持希拉里，深红州更加支持特朗普，摇摆州位于中间位置（符合先验知识）
2. 深红州均值较大，方差较小，深红州人民团结一致支持特朗普？而深蓝州均值较小，方差较大，蓝州人民支持希拉里的程度似乎没有红州人民支持特朗普一样。（这也反映了希拉里败选的一个原因，因为很多民主党选民讨厌希拉里，所以他们宁愿不投票）
3. 特朗普在摇摆州更受支持，极大的可能正是因为摇摆州人民的支持使得特朗普当选

4. 统计推断

在进行了EDA之后，选取其中一些数据进行统计推断。统计推断主要包括参数估计与假设检验，两者都是根据样本信息对总体的数量特征进行推断，但是角度不同。参数估计是以样本估计总体参数的真值，假设检验是以样本检验对总体参数的先验假设是否成立。

4.1. 参数估计。 该部分代码位于文件par_estimate.py中。

4.1.1. 点估计。 选取深红州，深蓝州和摇摆州希拉里得票率数据做均值的点估计。

```
pnt_estimate_red = []
pnt_estimate_blue = []
pnt_estimate_swing = []
for i in range(1000):
    sample = np.random.choice(a = data_red['Trump'], size = 100)
    pnt_estimate_red.append(sample.mean())
for i in range(1000):
    sample = np.random.choice(a = data_blue['Trump'], size = 100)
    pnt_estimate_blue.append(sample.mean())
for i in range(1000):
    sample = np.random.choice(a = data_swing['Trump'], size = 100)
    pnt_estimate_swing.append(sample.mean())
sns.distplot(pd.DataFrame(pnt_estimate_red), kde_kws={"color": 'r'},
              color = 'r')
sns.distplot(pd.DataFrame(pnt_estimate_blue), kde_kws={"color": [0,0,1]},
              color = [0,0,1])
sns.distplot(pd.DataFrame(pnt_estimate_swing), kde_kws = {'color': [1,0,1]},
              color = [1,0,1])
plt.legend(['red_state', 'blue_state', 'swing_state'], fontsize = 15)
plt.grid(linestyle = '--')
plt.xlabel('Vote rate of Trump')
```

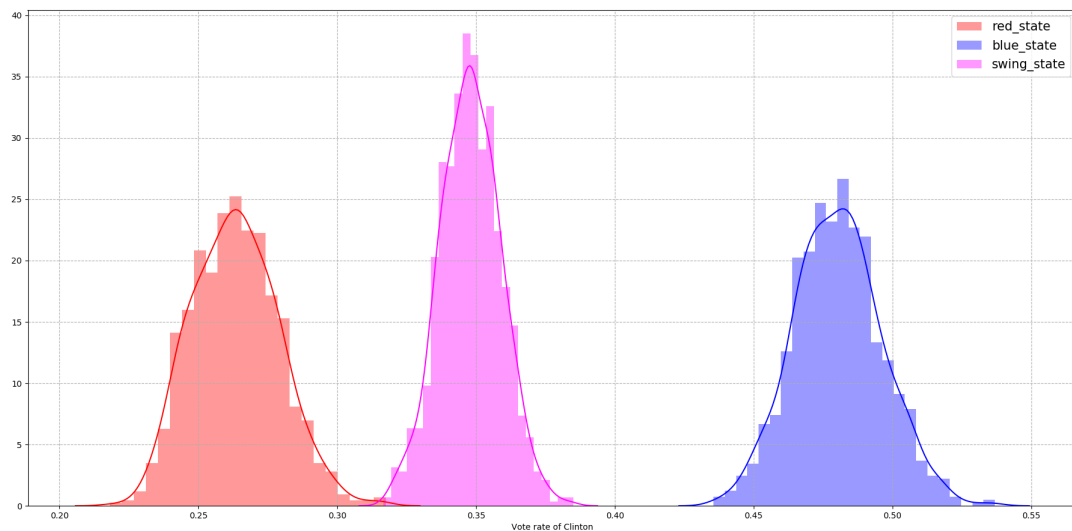



图 12: 样本均值的分布 (希拉里)

```
Mean point estimation in red states 0.26333657652895476
Mean point estimation in blue states 0.4803123414170364
Mean point estimation in swing states 0.347990945268237
```

图 13: 点估计结果 (希拉里)

即使在深蓝州希拉里得票率的均值的点估计值也未能超过0.5，这是一件很奇怪的事情。这不禁令人想看一下特朗普的数据。

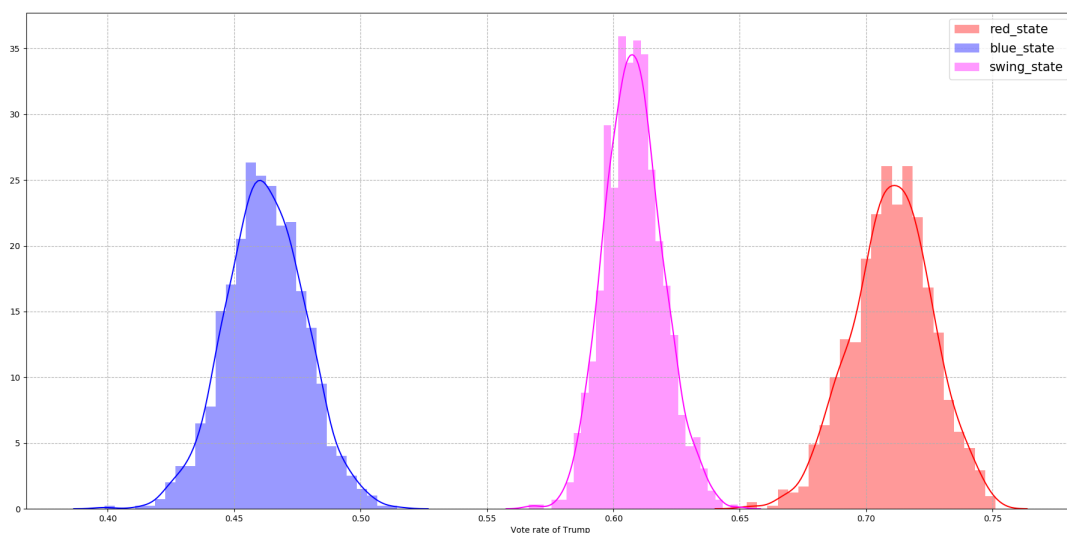


图 14: 样本均值的分布 (特朗普)

```
Mean point estimation in red states 0.7100982496979561
Mean point estimation in blue states 0.46231484067991174
Mean point estimation in swing states 0.6081129131983296
```

图 15: 点估计结果 (特朗普)

即使在蓝州特朗普也有0.46的得票率均值估计，与希拉里相差不大，而在摇摆州和红州大幅度领先希拉里，看来特朗普真有种众望所归的味道。

4.1.2. 区间估计。 获得深红州、深蓝州和摇摆州希拉里得票率均值的置信水平为95%的置信区间。

```
def MakeConfidenceInterval(sample_size, data, alpha):
    sample = np.random.choice(a = data, size = sample_size)
    sigma = sample.std()/np.sqrt(sample_size)
    return stats.t.interval(alpha = alpha, df = sample_size - 1,
                            loc = sample.mean(), scale = sigma)
print('Confidence interval with confidence level of 95% in red state:',
      MakeConfidenceInterval(100, data_red['Clinton'], 0.95))
print('Confidence interval with confidence level of 95% in blue state:',
      MakeConfidenceInterval(100, data_blue['Clinton'], 0.95))
print('Confidence interval with confidence level of 95% in swing state:',
      MakeConfidenceInterval(100, data_swing['Clinton'], 0.95))
```

```
Confidence interval in red state: (0.23756825974296666, 0.3041750898451179)
Confidence interval in blue state: (0.443734231588439, 0.5068633498972412)
Confidence interval in swing state: (0.3286679784124668, 0.3702145432031882)
```

图 16: 区间估计结果

4.2. 假设检验。 该部分代码位于文件hypo_test.py中。

4.2.1. t检验。 在EDA中我们发现在深蓝州和深红州希拉里的得票率分布具有较大差异，因此我们有理由判断希拉里在深蓝州得票率的均值 μ_b 不等于深红州得票率的均值 μ_r 。现在我们做出假设：

$$H_0 : \mu_b = \mu_r$$

$$H_1 : \mu_b \neq \mu_r$$

现根据样本进行t检验：

```
sample_red = np.random.choice(a = data_red['Clinton'], size = 100)
sample_blue = np.random.choice(a = data_blue['Clinton'], size = 100)

alpha = 0.05
t_statistic, p_value = stats.ttest_rel(a = sample_red, b = sample_blue)
print('t = ', t_statistic)
print('p = ', p_value)
if p_value <= alpha:
    print('Refuse H0')
```

```
else:
    print('Accept H1')
```

```
t = -8.590701713166169
p = 1.281400720680821e-13
Refuse H0
```

图 17: t检验结果

根据结果，我们拒绝 H_0 ，即我们认为在深蓝州和深红州希拉里得票率不相等。

4.2.2. F检验. 在EDA中可以看出希拉里在深红州和摇摆州得票率曲线比较相似，现通过F检验判断两者方差是否相等。

$$H_0 : \sigma_{swing} = \sigma_{red}$$

$$H_1 : \sigma_{swing} \neq \sigma_{red}$$

```
F = data_swing['Clinton'].var()/data_red['Clinton'].var()
df1 = len(data_swing) - 1
df2 = len(data_red) - 1
p_value_2 = 1 - 2*abs(0.5 - stats.f.cdf(F, df1, df2))
print('p = ', p_value_2)
if p_value <= alpha:
    print('Refuse H0')
else:
    print('Accept H1')
```

```
p = 2.4141810772704275e-10
Refuse H0
```

图 18: F检验结果

由此可见两样本方差并不相等

5. 回归与分类

5.1. 线性回归. 对摇摆州和深蓝州地区的获得学士学位人口比例和希拉里得票率进行线性回归。

5.2. 朴素贝叶斯分类. 接下来几节将通过三种机器学习分类方法来对大选结果进行预测，并进行分类性能比较。

5.2.1. 定义函数`classification_result()`. 该函数参数为训练集和测试集以及分类器，功能是完成模型拟合并输出测试集的分类结果。

5.3. 决策树分类.

5.4. KNN分类.

5.5. 分类性能比较.

6. 总结