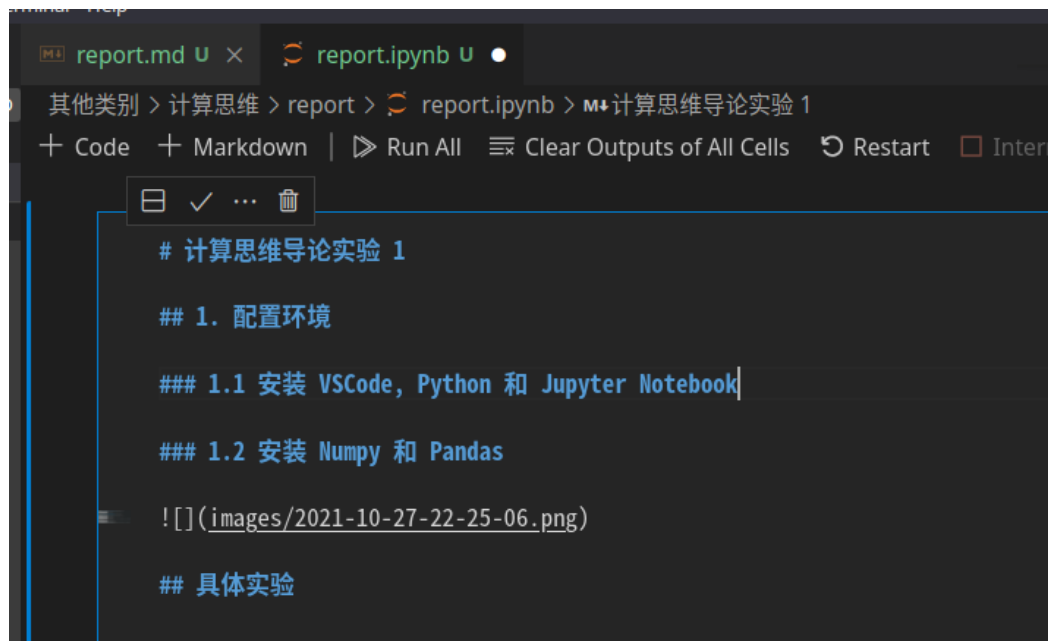


# 计算思维导论实验 1

201300035 方盛俊

## 1. 配置环境

### 1.1 安装 VSCode, Python 和 Jupyter Notebook



```
# 计算思维导论实验 1

## 1. 配置环境

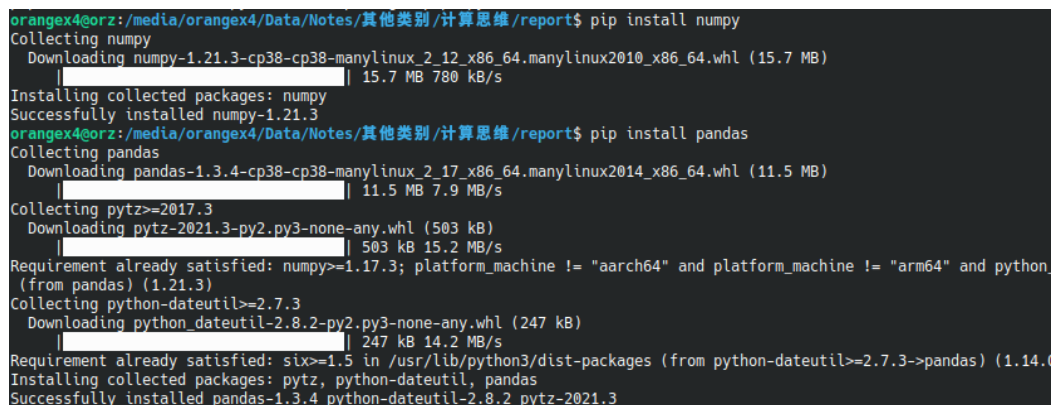
### 1.1 安装 VSCode, Python 和 Jupyter Notebook

### 1.2 安装 Numpy 和 Pandas

!(images/2021-10-27-22-25-06.png)

## 具体实验
```

### 1.2 安装 Numpy 和 Pandas



```
orangex4@orx:/media/orangex4/Data/Notes/其他类别/计算思维/report$ pip install numpy
Collecting numpy
  Downloading numpy-1.21.3-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
    | 15.7 MB 780 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.21.3
orangex4@orx:/media/orangex4/Data/Notes/其他类别/计算思维/report$ pip install pandas
Collecting pandas
  Downloading pandas-1.3.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
    | 11.5 MB 7.9 MB/s
Collecting pytz>=2017.3
  Downloading pytz-2021.3-py2.py3-none-any.whl (503 kB)
    | 503 kB 15.2 MB/s
Requirement already satisfied: numpy>=1.17.3; platform_machine != "aarch64" and platform_machine != "arm64" and python
  (from pandas) (1.21.3)
Collecting python-dateutil>=2.7.3
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
    | 247 kB 14.2 MB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7.3->pandas) (1.14.0)
Installing collected packages: pytz, python-dateutil, pandas
Successfully installed pandas-1.3.4 python-dateutil-2.8.2 pytz-2021.3
```

## 2. 学习 Pandas

```
In [ ]: import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```
In [ ]:
```

```
df = pd.read_csv('telecom_churn.csv')
df.head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	No	Yes	25	265.1	110	45.07	
1	OH	107	415	No	Yes	26	161.6	123	27.47	
2	NJ	137	415	No	No	0	243.4	114	41.38	
3	OH	84	408	Yes	No	0	299.4	71	50.90	
4	OK	75	415	Yes	No	0	166.7	113	28.34	

```
In [ ]: df.shape
```

```
Out[ ]: (3333, 20)
```

```
In [ ]: df.columns
```

```
Out[ ]: Index(['State', 'Account length', 'Area code', 'International plan',
              'Voice mail plan', 'Number vmail messages', 'Total day minutes',
              'Total day calls', 'Total day charge', 'Total eve minutes',
              'Total eve calls', 'Total eve charge', 'Total night minutes',
              'Total night calls', 'Total night charge', 'Total intl minutes',
              'Total intl calls', 'Total intl charge', 'Customer service calls',
              'Churn'],
              dtype='object')
```

```
In [ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   State                                3333 non-null   object
1   Account length                       3333 non-null   int64
2   Area code                           3333 non-null   int64
3   International plan                   3333 non-null   object
4   Voice mail plan                      3333 non-null   object
5   Number vmail messages                3333 non-null   int64
6   Total day minutes                    3333 non-null   float64
7   Total day calls                      3333 non-null   int64
8   Total day charge                     3333 non-null   float64
9   Total eve minutes                    3333 non-null   float64
10  Total eve calls                      3333 non-null   int64
11  Total eve charge                     3333 non-null   float64
12  Total night minutes                  3333 non-null   float64
13  Total night calls                    3333 non-null   int64
14  Total night charge                   3333 non-null   float64
15  Total intl minutes                   3333 non-null   float64
16  Total intl calls                     3333 non-null   int64
17  Total intl charge                     3333 non-null   float64
18  Customer service calls               3333 non-null   int64
19  Churn                               3333 non-null   bool
```

dtypes: bool(1), float64(8), int64(8), object(3)  
memory usage: 498.1+ KB

```
In [ ]: df['Churn'] = df['Churn'].astype('int64')
```

```
In [ ]: df.describe()
```

```
Out[ ]:
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000

```
In [ ]: df.describe(include=['object', 'bool'])
```

```
Out[ ]:
```

	State	International plan	Voice mail plan
count	3333	3333	3333
unique	51	2	2
top	WV	No	No
freq	106	3010	2411

```
In [ ]: df['Churn'].value_counts()
```

```
Out[ ]: 0    2850  
1     483  
Name: Churn, dtype: int64
```

```
In [ ]: df['Churn'].value_counts(normalize=True)
```

```
Out[ ]: 0    0.855086  
1    0.144914  
Name: Churn, dtype: float64
```

```
In [ ]: df.sort_values(by='Total day charge', ascending=False).head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge
365	CO	154	415	No	No	0	350.8	75	59.64

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge
985	NY	64	415	Yes	No	0	346.8	55	58.96
2594	OH	115	510	Yes	No	0	345.3	81	58.70
156	OH	83	415	No	No	0	337.4	120	57.36
605	MO	112	415	No	No	0	335.5	77	57.04

◀ ▶

In [ ]: `df.sort_values(by=['Churn', 'Total day charge'], ascending=[True, False]).head()`

Out[ ]:

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge
688	MN	13	510	No	Yes	21	315.6	105	53.65
2259	NC	210	415	No	Yes	31	313.8	87	53.35
534	LA	67	510	No	No	0	310.4	97	52.77
575	SD	114	415	No	Yes	36	309.9	90	52.68
2858	AL	141	510	No	Yes	28	308.0	123	52.36

◀ ▶

In [ ]: `df['Churn'].mean()`

Out[ ]: 0.14491449144914492

In [ ]: `df[df['Churn'] == 1].mean()`

Out[ ]:

```
Account length      102.664596
Area code           437.817805
Number vmail messages    5.115942
Total day minutes     206.914079
Total day calls       101.335404
Total day charge       35.175921
Total eve minutes     212.410145
Total eve calls       100.561077
Total eve charge       18.054969
Total night minutes   205.231677
Total night calls     100.399586
Total night charge     9.235528
Total intl minutes    10.700000
Total intl calls       4.163561
Total intl charge      2.889545
Customer service calls 2.229814
Churn                 1.000000
dtype: float64
```

In [ ]: `df[df['Churn'] == 1]['Total day minutes'].mean()`

Out[ ]: 206.91407867494823

```
In [ ]: df[(df['Churn'] == 0) & (df['International plan'] == 'No')]['Total intl minutes']
```

```
Out[ ]: 18.9
```

```
In [ ]: df.loc[0:5, 'State': 'Area code']
```

```
Out[ ]:
```

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415
3	OH	84	408
4	OK	75	415
5	AL	118	510

```
In [ ]: df.iloc[0:5, 0:3]
```

```
Out[ ]:
```

	State	Account length	Area code
0	KS	128	415
1	OH	107	415
2	NJ	137	415
3	OH	84	408
4	OK	75	415

```
In [ ]: df[-1:]
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge
3332	TN	74	415	No	Yes	25	234.4	113	39.85

```
In [ ]: df.apply(np.max)
```

```
Out[ ]:
```

State	WY
Account length	243
Area code	510
International plan	Yes
Voice mail plan	Yes
Number vmail messages	51
Total day minutes	350.8
Total day calls	165
Total day charge	59.64
Total eve minutes	363.7
Total eve calls	170
Total eve charge	30.91
Total night minutes	395.0
Total night calls	175

```
Total night charge      17.77
Total intl minutes      20.0
Total intl calls         20
Total intl charge        5.4
Customer service calls   9
Churn                     1
dtype: object
```

```
In [ ]: df[df['State'].apply(lambda state: state[0] == 'W')].head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
9	WV	141	415	Yes	Yes	37	258.6	84	43.96	
26	WY	57	408	No	Yes	39	213.0	115	36.21	
44	WI	64	510	No	No	0	154.0	67	26.18	
49	WY	97	415	No	Yes	24	133.2	135	22.64	
54	WY	87	415	No	No	0	151.0	83	25.67	

```
In [ ]: d = {'No': False, 'Yes': True}
df['International plan'] = df['International plan'].map(d)
df.head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	False	Yes	25	265.1	110	45.07	
1	OH	107	415	False	Yes	26	161.6	123	27.47	
2	NJ	137	415	False	No	0	243.4	114	41.38	
3	OH	84	408	True	No	0	299.4	71	50.90	
4	OK	75	415	True	No	0	166.7	113	28.34	

```
In [ ]: df = df.replace({'Voice mail plan': d})
df.head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	False	True	25	265.1	110	45.07	
1	OH	107	415	False	True	26	161.6	123	27.47	
2	NJ	137	415	False	False	0	243.4	114	41.38	
3	OH	84	408	True	False	0	299.4	71	50.90	
4	OK	75	415	True	False	0	166.7	113	28.34	

```
In [ ]:
```

```
columns_to_show = ['Total day minutes', 'Total eve minutes', 'Total night minutes']
df.groupby(['Churn'])[columns_to_show].describe(percentiles=[])
```

Out[ ]:

Total day minutes									
	count	mean	std	min	50%	max	count	mean	std
Churn									
0	2850.0	175.175754	50.181655	0.0	177.2	315.6	2850.0	199.043298	50.292175
1	483.0	206.914079	68.997792	0.0	217.6	350.8	483.0	212.410145	51.728910

In [ ]:

```
columns_to_show = ['Total day minutes', 'Total eve minutes', 'Total night minutes']
df.groupby(['Churn'])[columns_to_show].agg([np.mean, np.std, np.min, np.max])
```

Out[ ]:

Total day minutes					Total eve minutes				
	mean	std	amin	amax	mean	std	amin	amax	
Churn									
0	175.175754	50.181655	0.0	315.6	199.043298	50.292175	0.0	361.8	200.0
1	206.914079	68.997792	0.0	350.8	212.410145	51.728910	70.9	363.7	205.0

In [ ]:

```
df.pivot_table(['Total day calls', 'Total eve calls', 'Total night calls'], ['Area code'])
```

Out[ ]:

	Total day calls	Total eve calls	Total night calls
Area code			
408	100.496420	99.788783	99.039379
415	100.576435	100.503927	100.398187
510	100.097619	99.671429	100.601190

In [ ]:

```
pd.crosstab(df['Churn'], df['International plan'])
```

Out[ ]:

International plan	False	True
Churn		
0	2664	186
1	346	137

In [ ]:

```
pd.crosstab(df['Churn'], df['Voice mail plan'], normalize=True)
```

Out[ ]:

Voice mail plan	False	True
Churn		
0	0.602460	0.252625
1	0.120912	0.024002

```
In [ ]: total_calls = df['Total day calls'] + df['Total eve calls'] + df['Total night cal
df.insert(loc=len(df.columns), column='Total calls', value=total_calls)
df.head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	False	True	25	265.1	110	45.07	
1	OH	107	415	False	True	26	161.6	123	27.47	
2	NJ	137	415	False	False	0	243.4	114	41.38	
3	OH	84	408	True	False	0	299.4	71	50.90	
4	OK	75	415	True	False	0	166.7	113	28.34	

5 rows × 21 columns

```
In [ ]: df['Total charge'] = df['Total day charge'] + df['Total eve charge'] + df['Total
df.head()
```

```
Out[ ]:
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	False	True	25	265.1	110	45.07	
1	OH	107	415	False	True	26	161.6	123	27.47	
2	NJ	137	415	False	False	0	243.4	114	41.38	
3	OH	84	408	True	False	0	299.4	71	50.90	
4	OK	75	415	True	False	0	166.7	113	28.34	

5 rows × 22 columns

```
In [ ]: df.drop(['Total charge', 'Total calls'], axis=1, inplace=True)
df.drop([1, 2]).head()
```

```
Out[ ]:
```

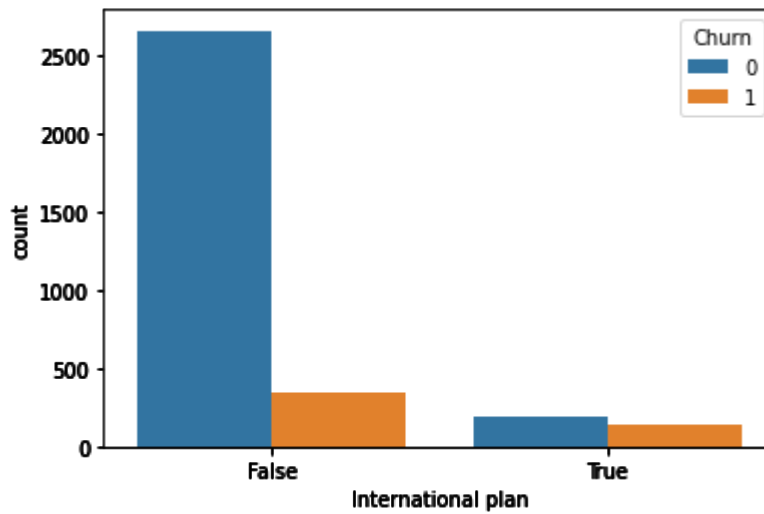
	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	mi
0	KS	128	415	False	True	25	265.1	110	45.07	
3	OH	84	408	True	False	0	299.4	71	50.90	
4	OK	75	415	True	False	0	166.7	113	28.34	
5	AL	118	510	True	False	0	223.4	98	37.98	
6	MA	121	510	False	True	24	218.2	88	37.09	

```
In [ ]: import matplotlib.pyplot as plt
import seaborn as sns
```



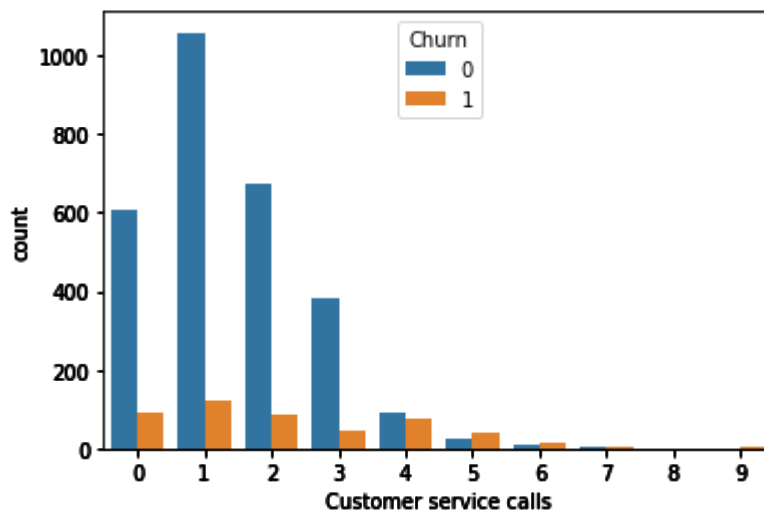
```
sns.countplot(x='International plan', hue='Churn', data=df)
```

Out[ ]: <AxesSubplot:xlabel='International plan', ylabel='count'>



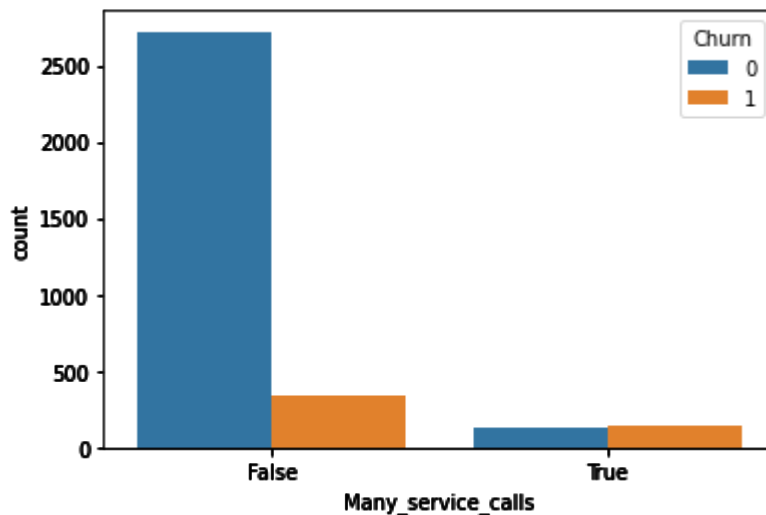
```
In [ ]: pd.crosstab(df['Churn'], df['Customer service calls'], margins=True)  
sns.countplot(x='Customer service calls', hue='Churn', data=df)
```

Out[ ]: <AxesSubplot:xlabel='Customer service calls', ylabel='count'>



```
In [ ]: df['Many_service_calls'] = (df['Customer service calls'] > 3)  
pd.crosstab(df['Many_service_calls'], df['Churn'], margins=True)  
sns.countplot(x='Many_service_calls', hue='Churn', data=df)
```

Out[ ]: <AxesSubplot:xlabel='Many\_service\_calls', ylabel='count'>



```
In [ ]: pd.crosstab(df['Many_service_calls'] & df['International plan'], df['Churn'])
```

```
Out[ ]: Churn    0    1
row_0
False  2841  464
True    9    19
```

### 3. 对 test.csv 数据集进行实验

```
In [ ]: df = pd.read_csv('test.csv')
df.sort_values(by=['attempts', 'name'])
```

```
Out[ ]:   index  attempts   name  qualify  score
0      0         1  Anastasia    yes    12.5
9      9         1   Jonas    yes    19.0
7      7         1   Laura    no     NaN
6      6         1  Matthew    yes    14.5
8      8         2   Kevin    no     8.0
4      4         2   Emily    no     9.0
2      2         2  Katherine    yes    16.5
1      1         3   Dima    no     9.0
3      3         3   James    no     NaN
5      5         3  Michael    yes    20.0
```

### 4. 总结与收获

Python 是一个对数据处理来说非常优异的语言, 可以很方便地读取, 变换, 处理, 并可可视化地展示数据. 从中, 我们可以应用我们的计算思维, 从数据中挖掘出种种的可能性.