

# 数据可视化

### 消费总额占比分析



### 人均消费占比分析



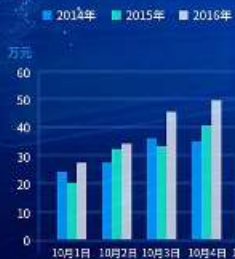
### 性别消费占比分析



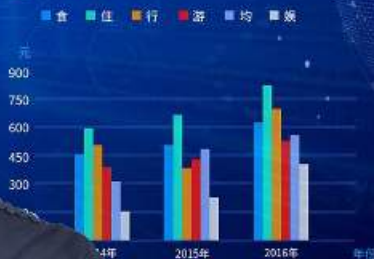
### 区域消费占比分析



### 历年十一黄金周景区收入



### 十一黄金周人均消费对比



# 目录

## Contents

---

第一章

绪 论

第二章

Matplotlib的基本使用

第三章

Matplotlib的进阶使用

第四章

...

第五章

...

# 第十一章 金融数据可视化 分析案例

1

引言

2

数据准备

3

宏观金融数据可视化

4

股票数据可视化

5

汇率数据可视化

# 1. 引言





<https://tushare.pro/>

Tushare是一个免费、开源的python财经数据接口包，该接口能够为金融分析人员提供快速、整洁、和多样的便于分析的数据，其中包含宏观金融、股票和汇率等数据。

**2.**

## **数据准备**



### ◇ Linux

打开 Terminal 窗口，输入以下内容：

```
$ sudo apt-get install python3-pandas
```

```
$ sudo apt-get install python3-tushare
```

```
$ sudo apt-get install python3-akshare
```

### ◇ MacOS

打开 Terminal 窗口，输入以下内容：

```
$ pip3 install pandas
```

```
$ pip3 install tushare
```

```
$ pip3 install akshare
```

### ◇ Windows

打开 Anaconda Prompt 窗口，输入以下内容：

```
pip install pandas
```

```
pip install tushare
```

```
pip install akshare
```



```

Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 (from matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/8a/bb/488841f56197b13700afd5658fc279a2025a39e22449b7cf29864669b15d/pyparsing-2.4.7-py2.py3-none-any.whl (67kB)
    |#####| 71kB 68kB/s
Collecting pillow>=6.2.0 (from matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/9c/f0/00f71c1a52859f8f1b82ed6bc2bf5890321511b642c01242d38df02bb5d0/Pillow-7.2.0-cp38-cp38-win32.whl (1.8MB)
    |#####| 1.8MB 145kB/s
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/83/c0/b370ca17fd314f188d98194c0217054416401418df4928ca844dd5b4bb4e/kiwisolver-1.2.0-cp38-none-win32.whl (43kB)
    |#####| 51kB 3.2MB/s
Collecting numpy>=1.15 (from matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/aa/10/92a98ab6a17b5e780d7104864eb3b125bf932368d6b9cb07dcef4c05665a/numpy-1.19.1-cp38-cp38-win32.whl (10.9MB)
    |#####| 10.9MB 1.6MB/s
Collecting python-dateutil>=2.1 (from matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl (227kB)
    |#####| 235kB 726kB/s
Collecting six (from cyclur>=0.10->matplotlib)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: certifi, six, cyclur, pyparsing, pillow, kiwisolver, numpy, python-dateutil, matplotlib
Successfully installed certifi-2020.6.20 cyclur-0.10.0 kiwisolver-1.2.0 matplotlib-3.3.1 numpy-1.19.1 pillow-7.2.0 pyparsing-2.4.7 python-dateutil-2.8.1 six-1.15.0

```

```
In [14]: ▶ import pandas as pd
import tushare as ts
```



# 3. 宏观金融数据可视化



首先需要了解和查看数据，方便后续绘制图形

下面我们来绘制全国居民消费价格指数图表

```
import pandas as pd # 导入模块 pandas, 便于数据预处理
import tushare as ts
ts.set_token('你的token') # https://tushare.pro/ 注册获取 token, 并通过积分获取权限
pro= ts.pro_api()
CPI = pro.cn_cpi () # 获取数据
CPI.head() # 查看数据前 5 行
```



	month	nt_val	nt_yoy	nt_mom	nt_accu	town_val	town_yoy	town_mom	town_accu	cnt_val	cnt_yoy	cnt_mom	cnt_accu
0	202110	101.5	1.5	0.7	100.7	101.6	1.6	0.7	100.8	101.2	1.2	0.7	100.5
1	202109	100.7	0.7	0.0	100.6	100.8	0.8	0.0	100.7	100.2	0.2	0.1	100.4
2	202108	100.8	0.8	0.1	100.6	101.0	1.0	0.1	100.7	100.3	0.3	0.2	100.4
3	202107	101.0	1.0	0.3	100.6	101.2	1.2	0.3	100.6	100.4	0.4	0.2	100.4
4	202106	101.1	1.1	-0.4	100.5	101.2	1.2	-0.4	100.6	100.7	0.7	-0.5	100.4



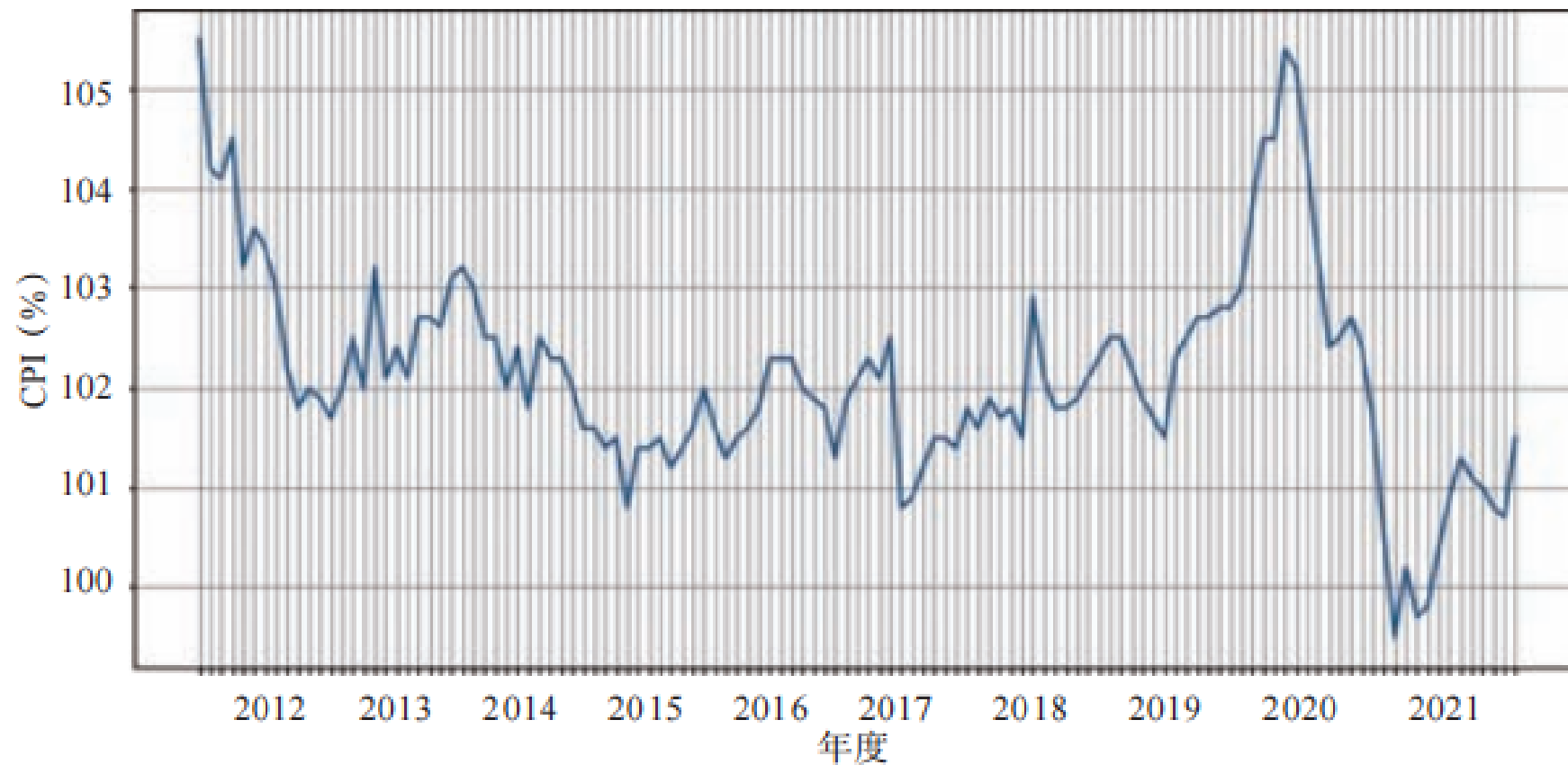
在查看了数据后，我们就可以用matplotlib绘制可视化图形了。



```
# 导入 matplotlib.pyplot 模块绘图
import matplotlib.pyplot as plt
plt.rcParams['font.family']='SimHei' #'SimHei' 为黑体
plt.rcParams['axes.unicode_minus'] = False # 显示负号
x = CPI['month'][0:120,][::-1] #2012 年至 2021 年数据 , 并翻转数据
y = CPI['nt_val'][0:120,][::-1]
xlabels = ['' for i in range(120)]
for i in range(10):
    xlabels[i*12+5]=str(i+2012) ## 只保留年份作为 x 轴标签
plt.figure(figsize=(15,7)) # 设置画布大小
plt.title("2012 年 ~2021 年 CPI 月度变化折线图 CPI 月度图表 ",fontsize=20) # 设置标题
plt.plot(x,y,linewidth=2) # 绘制折线图
plt.grid(True) # 显示网格
plt.xlabel('年度',size=25) # 设置 x 轴
plt.ylabel("CPI(%)",size=25) # 设置 y 轴
plt.xticks(x, xlabels, size=20) # 设置 x 轴刻度及字体大小
plt.yticks(size=20) # 设置 y 轴刻度及字体大小
plt.show() # 显示图
```



2012年至2021年CPI月度变化折线图 CPI月度图表





下面我们来绘制国内生产总值图表

```
import pandas as pd # 导入模块 pandas, 便于数据预处理
GDP = pro.cn_gdp(start_q='2011Q1', end_q='2020Q4') # 查看 2011 年至 2020 年季度数据
print(GDP.head())
```



	quarter	gdp	gdp_yoy	pi	pi_yoy	si	si_yoy	ti	ti_yoy
0	2020Q4	1015986.2	2.3	77754.1	3.0	384255.3	2.6	553976.8	2.1
1	2020Q3	722786.4	0.7	48122.5	2.3	274266.7	0.9	400397.1	0.4
2	2020Q2	456614.4	-1.6	26053.0	0.9	172759.0	-1.9	257802.4	-1.6
3	2020Q1	206504.3	-6.8	10186.2	-3.2	73638.0	-9.6	122680.1	-5.2
4	2019Q4	990865.1	6.1	70466.7	3.1	386165.3	5.7	534233.1	6.9

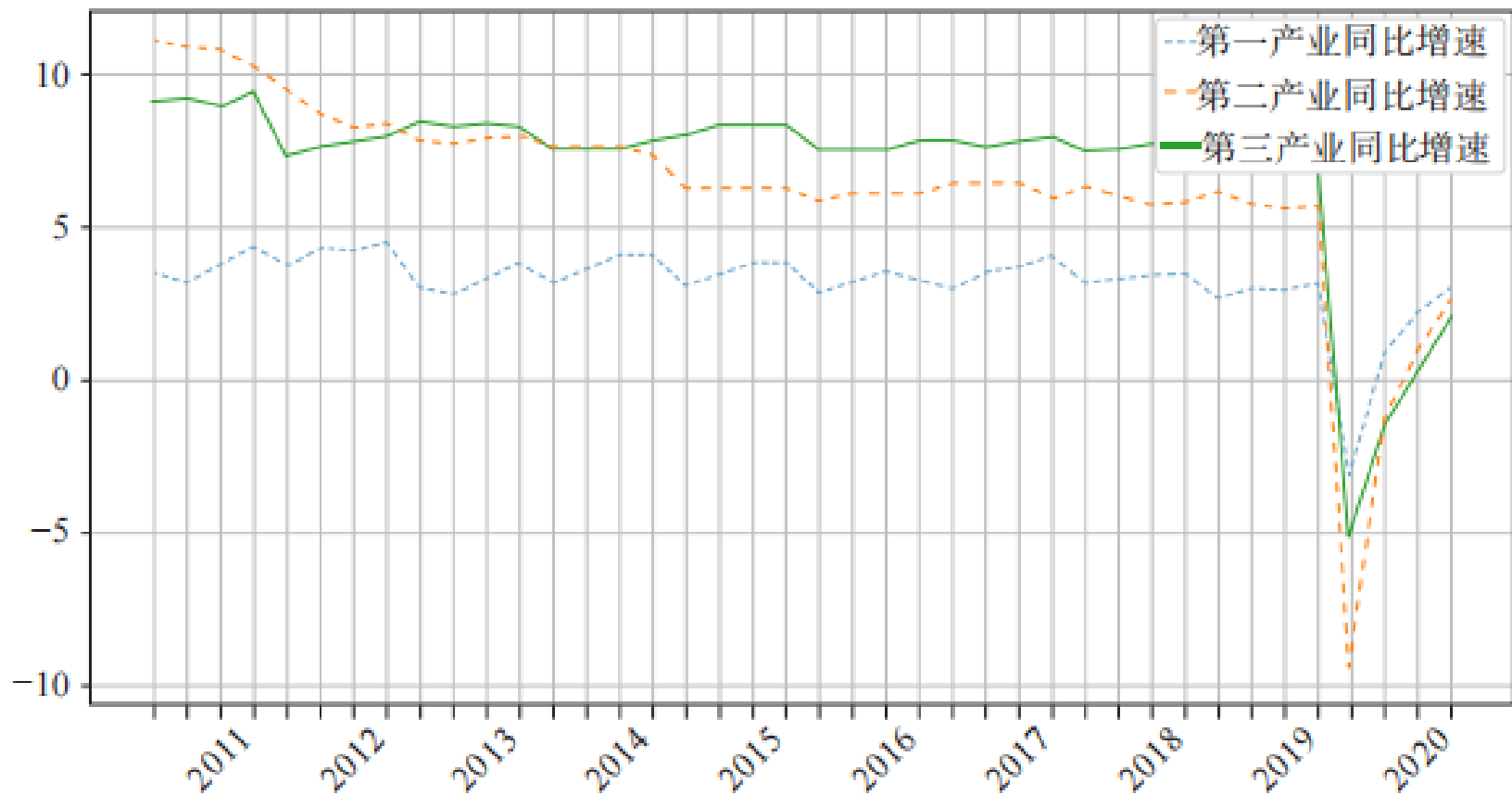




```
import matplotlib.pyplot as plt # 导入模块 matplotlib, 便于绘制图片
x=GDP['quarter'][:, :-1]
#2011~2020 年 GDP 构成数据
y1 = list(GDP['pi_yoy'][:, :-1]) # 第一产业同比增速 (%)
y2 = list(GDP['si_yoy'][:, :-1]) # 第二产业同比增速 (%)
y3 = list(GDP['ti_yoy'][:, :-1]) # 第三产业同比增速 (%)
plt.figure(figsize=(12, 6))
p1 = plt.plot(x, y1, label = '第一产业同比增速', linestyle='--') # 设置绘制风格
p2 = plt.plot(x, y2, label = '第二产业同比增速', linestyle='-.')
p3 = plt.plot(x, y3, label = '第三产业同比增速', linewidth=3)
plt.title('2011~2020 年 GDP 构成同比增速变化折线图', fontsize=20)
xlabels = ['' for i in range(len(GDP))]
for i in range(10):
    xlabels[i*4+2]=str(i+2011) ## 只保留年份作为 x 轴标签
plt.xticks(x, xlabels, rotation=45, fontsize=15)
plt.xticks(rotation=45, fontsize=15)
plt.yticks(fontsize=15)
plt.grid(True) # 显示网格
plt.legend() # 显示图例
plt.show() # 显示图片
```



2011 ~ 2020 年 GDP 构成同比增速变化折线图





下面我们来绘制货币供应量图表

```
import pandas as pd # 导入模块 pandas, 便于数据预处理
money = pro.cn_m(start_m='199201', end_m='202012')
print (money.head())
```



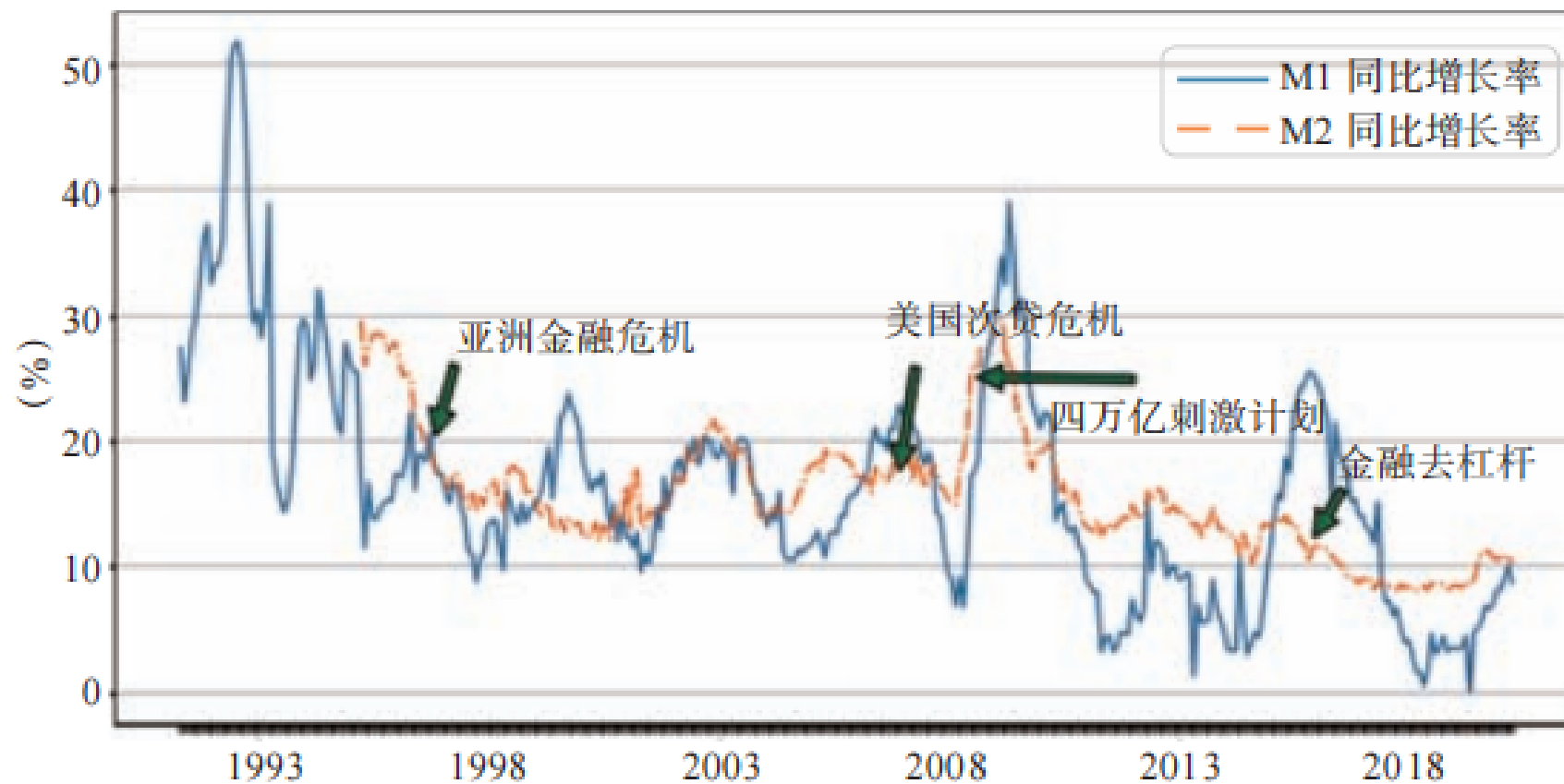
	month	m0	m0_yoy	m0_mom	m1	m1_yoy	m1_mom	m2
0	202012	84300.0	9.2	3.31	625600.0	8.6	1.13	2186800.0
1	202011	81600.0	10.3	0.74	618600.0	10.0	1.54	2172000.0
2	202010	81000.0	10.4	-1.70	609200.0	9.1	1.15	2149700.0
3	202009	82400.0	11.1	3.00	602300.0	8.1	0.17	2164100.0
4	202008	80000.0	9.4	0.17	601300.0	8.0	1.71	2136800.0



```
import matplotlib.pyplot as plt # 导入模块 matplotlib, 便于绘制图片
x=money['month'][:, :-1]
y1 = list(money['m1_yoy'])[:, :-1] # M1 同比增长率
y2 = list(money['m2_yoy'])[:, :-1] # M2 同比增长率
plt.figure(figsize=(12, 6))
p1 = plt.plot(x, y1, label = 'M1 同比增长率', linestyle='-')
p2 = plt.plot(x, y2, label = 'M2 同比增长率', linestyle='-')
plt.annotate(' 亚洲金融危机 ', size=13, # 设置显示文字
            xy=('199707', 20), # 设置箭头显示位置
            xytext=('199801', 27), # 设置文字显示位置
            arrowprops=dict(facecolor='green', shrink=0.05)) # 设置颜色
plt.annotate(' 美国次贷危机 ', size=13, xy=('200708', 17),
            xytext=('200801', 27), arrowprops=dict(facecolor='green', shrink=0.05))
plt.annotate(' 四万亿刺激计划 ', size=13, xy=('200902', 25),
            xytext=('201301', 23), arrowprops=dict(facecolor='green', shrink=0.05))
plt.annotate(' 金融去杠杆 ', size=13, xy=('201608', 12),
            xytext=('201705', 17), arrowprops=dict(facecolor='green', shrink=0.05))
plt.title("M1 和 M2 变化折线图 ", fontsize=20)
plt.legend()
plt.grid(axis='y') # 显示网格
xlabels = ['' for i in range(len(money['month']))]
for i in range(6):
    xlabels[i*60+12]=str(i*5+1993) ## 只保留几个年份作为 x 轴标签 plt.xticks
    (x, xlabels, size = 20)
plt.xlabel(' 年度 ', size=20) # 设置 x 轴标签及字体大小
plt.ylabel("(%)", size=20) # 设置 y 轴标签及字体大小
plt.show()
```



M1 和 M2 变化折线图





也可以直接计算 GDP 年度增长和货币供应量关系  
(用 M1 和 M2 分别除以 GDP 得到的数据) 进行  
绘图

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/147035053015006102>