* 什么是词云

词云又叫文字云，是对文本数据中出现频率较高的“关键词”在视觉上的突出呈现，形成关键词的渲染形成类似云一样的彩色图片，从而一眼就可以领略文本数据的主要表达意思。

* 准备工作：

python开发环境、wordcloud、jieba、matplotlib、numpy 、PIL 等库文件安装好。

* pip 安装方法：
  1. pip install xxx
  2. 使用idea 直接安装

**wordcloud生成词云的原理简介**   
wordcloud生成词云的原理其实并不复杂，大体分成5步（具体可自行查看源码）：

1.wordcloud制作词云时，首先要对对文本数据进行分词，使用**process\_text（）**方法，这一步的主要任务是去除停用词   
2.第二步是计算每个词在文本中出现的频率，生成一个哈希表。词频用于确定一个词的重要性   
3.根据词频的数值按比例生成一个图片的布局，类**IntegralOccupancyMap** 是该词云的算法所在，是词云的数据可视化方式的核心。生成词的颜色、位置、方向等   
4.最后将词按对应的词频在词云布局图上生成图片，核心方法是**generate\_from\_frequencies**,不论是**generate（）**还是**generate\_from\_text（）**都最终用到**generate\_from\_frequencies**   
完成词云上各词的着色,默认是随机着色   
5.词语的各种增强功能大都可以通过wordcloud的构造函数实现，里面提供了22个参数，还可以自行扩展。

#coding:utf-8 import matplotlib.pyplot as plt from wordcloud import WordCloud,ImageColorGenerator,STOPWORDS import jieba import numpy as np from PIL import Image #读入背景图片 abel\_mask = np.array(Image.open("filepath")) #读取要生成词云的文件 text\_from\_file\_with\_apath = open('filepath').read() #通过jieba分词进行分词并通过空格分隔 wordlist\_after\_jieba = jieba.cut(text\_from\_file\_with\_apath, cut\_all = True) wl\_space\_split = " ".join(wordlist\_after\_jieba) #my\_wordcloud = WordCloud().generate(wl\_space\_split) 默认构造函数 my\_wordcloud = WordCloud( background\_color='white', # 设置背景颜色 mask = abel\_mask, # 设置背景图片 max\_words = 200, # 设置最大现实的字数 stopwords = STOPWORDS, # 设置停用词 font\_path = C:/Users/Windows/fonts/simkai.ttf',# 设置字体格式，如不设置显示不了中文 max\_font\_size = 50, # 设置字体最大值 random\_state = 30, # 设置有多少种随机生成状态，即有多少种配色方案 scale=.5 ).generate(wl\_space\_split) # 根据图片生成词云颜色 image\_colors = ImageColorGenerator(abel\_mask) #my\_wordcloud.recolor(color\_func=image\_colors) # 以下代码显示图片 plt.imshow(my\_wordcloud) plt.axis("off") plt.show()

* 实例   
  **生成李克强总理2017年政府工作报告词云**   
  1.采用默认构造函数WordCloud().generate()生成   
    
  2.WordCloud()中设置一些参数设置mask 参数添加如下背景图案：   
     
  根据上面的背景图案，最终生成的词云时这样的：   
  