



数据分析与可视化大作业

作业进行中

已用45天2小时

开始

2020/10/19 12:00

截止

2020/12/08 12:00

剩余4天21小时 - 10%

现在是 14:52:58

保存

添加封面

添加附件

删除

未评分

对环境问题和气温变化的分析

对2013到2020的AQI值和污染物分析，对1800年到2013年温度变化分析，以及对2013年到2020年的AQI值和污染物分析。

- ChinaTemperatures.csv

删除
- GlobalLandTemperaturesByCountry.csv

删除
- 城市.XLSX

删除
- analysis of temperature.py

删除
- 全球温度地图.html

删除
- 平均温度最高的十个国家.html

删除
- 平均温度最低的五个国家.html

删除
- 中国与全球年温度折线图.html

删除
- 中国与全球预测温度图.html

删除
- 全球温度可视化汇总.html

删除

一. 问题的引出

近年来，环境问题变得日渐严峻，一些关于环境指标新的名词出现在人们的视野中：“全球变暖”、“AQI指数”、“大气污染物”，这些词语都是当下人们对于环境恶劣变化开始关注的衡量指标。

在线编程

吐槽

记得从2015年2月28日,柴静推出其自费拍摄的雾霾深度调查《穹顶之下》开始,人们对于环境变化才开始关注起来,当人们终于开始抬头望向天空,他们才发现,原来的晴空万里已经被浓烟所笼罩,原来星星遍布的夜晚已经被霓虹灯隐藏的难以寻觅。

现代化的工业不仅仅带来了AQI值的普遍升高,也带来了全球变暖的严峻问题。本可视化数据着重于对于环境污染指标的分析 and 全球变暖的预测。

二. 数据来源

本数据从两个方面获取:

(1) 引用的 'GlobalLandTemperaturesByCountry.csv' 文件从kaggle数据网站上获取, 其中包含1743年到2013年各国历年温度情况, ChinaTemperatures.csv为其中节选, 便于后续对中国情况的单独分析。

(2) 引用的 '城市.XLSX' 文件从<https://www.aqistudy.cn/historydata/>上获取, 截止2020年12月, 上面可以获取2013-12到2020-12的各个城市污染指标数据。

三. 数据分析可视化

1.预处理

(1) 所用库的导入

```
from pyecharts.charts import *
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import random
import pandas as pd
from matplotlib.pyplot import *
import pyecharts.options as opts
import matplotlib.pyplot as plt
import xlrd
```

(2) 颜色函数定义

为了更好的绘制各个图线, 定义颜色函数, 进行不同线段颜色随机生成构建图表

```
# 随机获取颜色
def randomcolor(kind):
    colors = []
    for i in range(kind):
```



首页



任务



在线编程

吐槽

课件

问答

成绩

设置

```

for i in range(14):
    colArr = ['1', '2', '3', '4', '5', '6', '7', '8', '9', 'A']
    color = ""
    for i in range(6):
        color += colArr[random.randint(0, 14)]
    colors.append("#" + color)
return colors

```

(3) 读取各个数据

```

all_country = pd.read_csv("GlobalLandTemperaturesByCountry.csv")
China_date = pd.read_csv('ChinaTemperatures.csv')
name = list(pd.read_excel('城市.XLSX')['城市'].drop_duplicates())
data = xlrd.open_workbook('城市.XLSX')

```

2.城市污染分析

(1) 数据处理

```

table = data.sheets()[0]
dic1 = {k: [] for k in name}
for i in range(1, table.nrows):
    x = table.row_values(i)
    dic1[x[0]].append((x[1], x[2], x[5], x[6], x[7], x[8], x[9], x[10]))
lisx = list(range(0, 85))
lisy = []
lisdate = []

```

(2) 绘制六大城市AQI污染值数对比

```

for i in range(1, len(lisdate)):
    lisdate.append('{}'.format(i))
color_series = randomcolor(8)
color_number = 0
for j in name:
    for i in lisx:
        lisy.append(dic1[j][i][1])
        plt.plot(lisx, lisy, color=color_series[color_number], label=j)
        lisy = [] # 清空为下次作图
        color_number = color_number + 1 # 变换随机颜色

plt.xticks(list(range(0, 85, 12)), lisdate, rotation=50)
plt.title("AQI污染指数", fontproperties='simHei')
plt.xlabel("年份", fontproperties='simHei')
plt.ylabel("AQI值", fontproperties='simHei')
plt.grid(alpha=0.4)
plt.legend(prop="simHei", loc='upper right')
plt.savefig('./AQI污染值数城市对比.png')

```

在线
编程
吐槽

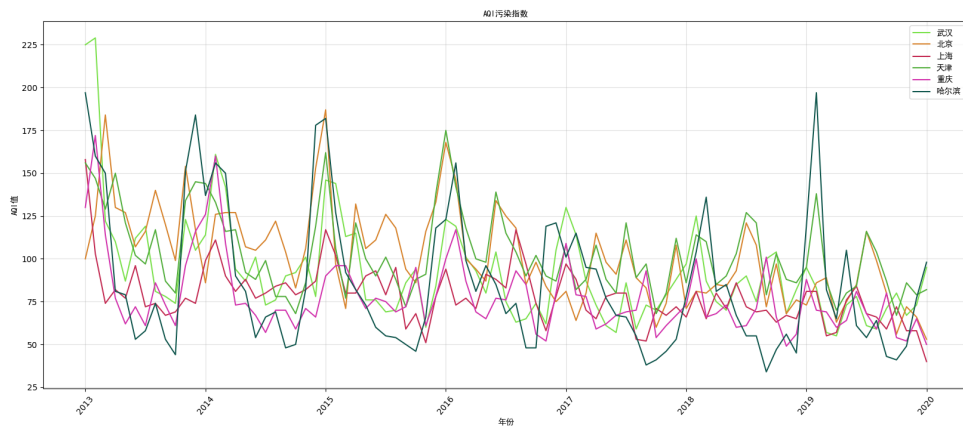


```

print('AQI污染值数城市对比已生成')
# plt.show()
plt.close()

```

生成图片如下:



分析：由图中发现，在六个中国主要城市之中，各个城市AQI大于50的时间几乎占据七年中90%以上，在100以下的良好天气也几乎只占40%。但是在2017年开始，AQI在125以上天数明显减少，当时人们对于环境问题开始逐渐关注起来，除了哈尔滨在2019年有异常反弹外，其余五个城市基本有较好改善。

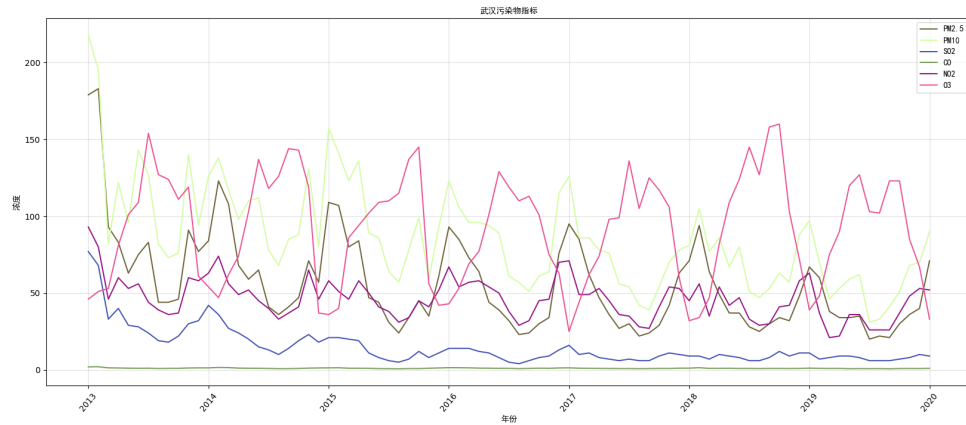
(3) 绘制武汉污染物指标

```

color_number = 0
name_label = ['PM2.5', 'PM10', 'SO2', 'CO', 'NO2', 'O3']
plt.figure(figsize=(20,8),dpi=100)
for j in range(2, 8):
    for i in lisx:
        lisy.append(dic1["武汉"][i][j])
    plt.plot(lisx, lisy, color=color_series[color_number], label=name_label[j])
    lisy = [] # 清空为下次作图
    color_number = color_number + 1 # 变换随机颜色
plt.xticks(list(range(0, 85, 12)), lisdate, rotation=50)
plt.title("武汉污染物指标", fontproperties='simHei')
plt.xlabel("年份", fontproperties='simHei')
plt.ylabel("浓度", fontproperties='simHei')
plt.grid(alpha=0.4)
plt.legend(prop="simHei", loc='upper right')
plt.savefig('./武汉污染物指标.png')
print('武汉污染物指标已生成')
# plt.show()
plt.close()

```





分析：由图中发现，在七年之间，CO一直维持较小的值没有太多波动，SO2的情况明显改善，由2013的75到2020趋于平稳的10左右，其余指标在每年之中波动，除PM10在波动中逐年递减，其余没有较大变化。

3.全球温度分析

(1) 数据处理

```
cou_tem = all_country.groupby("Country")['AverageTemperature'].mean
cou_tem = cou_tem.dropna() # 删除缺省
country = list(cou_tem.index) # 将国家转化列表
temperature = list(cou_tem.values) # 将温度转化列表
zip_country_temperature1 = zip(country, temperature) # 打包成元组
zip_country_temperature2 = zip(country, temperature)
top_ten_country = sorted(zip_country_temperature1, key=lambda tup: tup[1])
low_ten_country = sorted(zip_country_temperature2, key=lambda tup: tup[1])
all_country['dt_year'] = pd.to_datetime(all_country['dt']).dt.year
all_country['dt_month'] = pd.to_datetime(all_country['dt']).dt.month
China_date['dt_year'] = pd.to_datetime(China_date['dt']).dt.year
year_tem = all_country.groupby(['dt_year'])['AverageTemperature'].mean
year_tem = year_tem.dropna() # 删除缺省
year_tem_China = China_date.groupby(['dt_year'])['AverageTemperature'].mean
year_tem_China = year_tem_China.dropna() # 删除缺省
year = list(year_tem.index) # 将年份转化列表
```

(2) 绘制温度地图

```
map = Map()
map.add("国家", [list(i) for i in zip(country, temperature)], ↑)
map.set_series_opts(label_opts=opts.LabelOpts(is_show=False))
map.set_global_opts(
    title_opts=opts.TitleOpts(title="全球温度图表"),
```

在线编程

吐槽

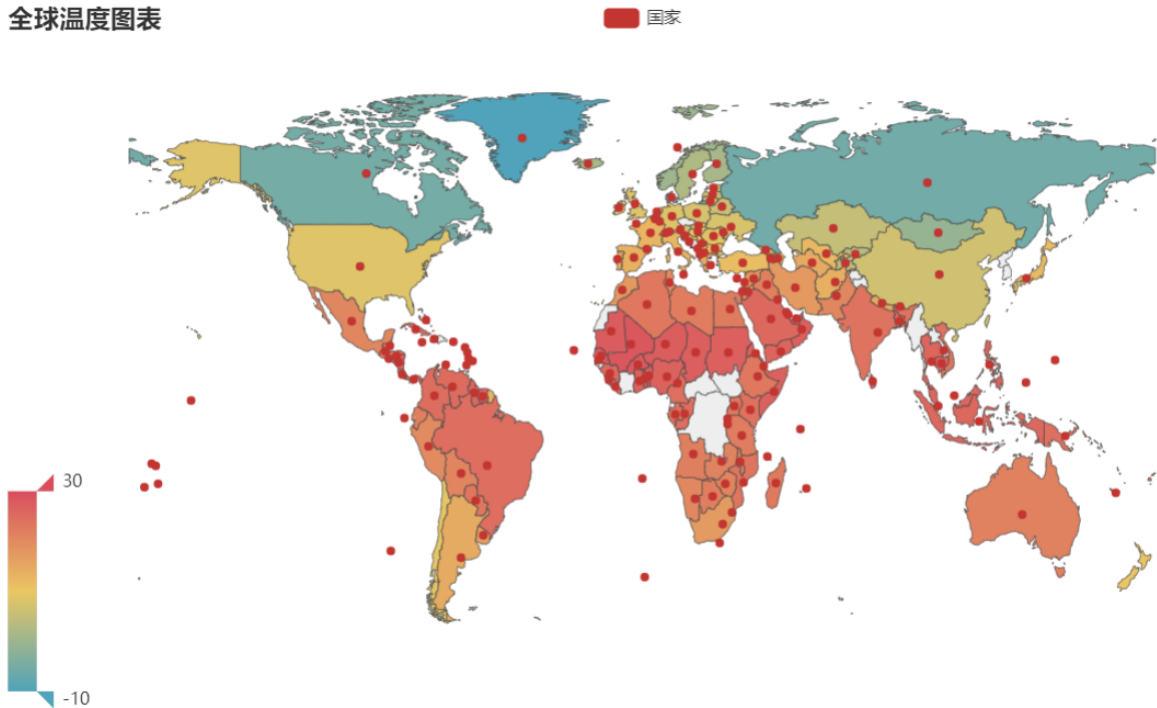
```

visualmap_opts=opts.VisualMapOpts(max_=30, min_=-10) # 最高温
)
# map.render('全球温度地图.html')
print("全球温度地图已生成！！")

```

绘制出来为html格式无法上传图片，请查看附件中‘全球温度地图.html’，下图为网页截图

全球温度图表



分析：在本图片中，我们可以发现，除了缺失数据国家以外，各个国家的温度的确较为符合其经纬度的特征温度。

(2) 绘制最高温度前十位国家的平均温度情况（条形图）

```

for i in range(10):
    x.append(
        optsBarItem(
            name=country[i],
            value=round(temperature[i], 2),
            itemstyle_opts=opts.ItemStyleOpts(color=color_series[
        )
    )
    # 绘制柱形图
bar = Bar()
bar.add_xaxis(country)
bar.add_yaxis(
    series_name='温度',
    y_axis=x,
    is_selected=True,
    label_opts=opts.LabelOpts(is_show=True))
bar.set_series_opts(label_opts=opts.LabelOpts(is_show=True, posit

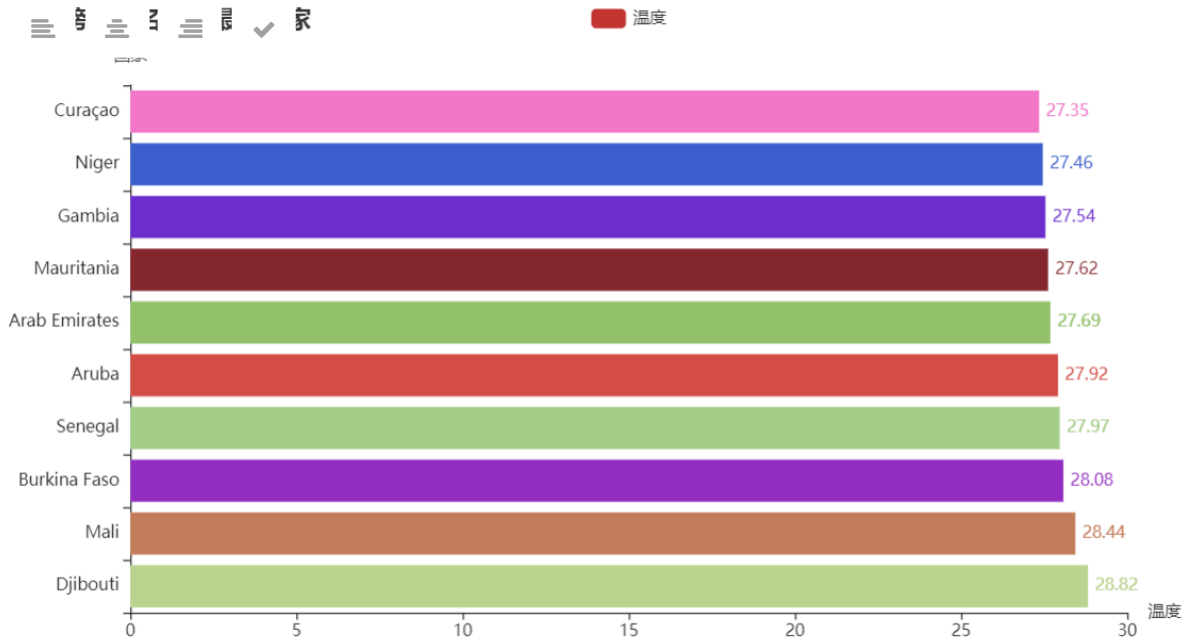
```

在线编程
吐槽



```
bar.set_global_opts(
    title_opts=opts.TitleOpts(title="排行榜前十名温度最高国家"),
    tooltip_opts=opts.TooltipOpts(
```

绘制出来为html格式无法上传图片，请查看附件中‘平均温度最高的十个国家.html’，下图为网页截图



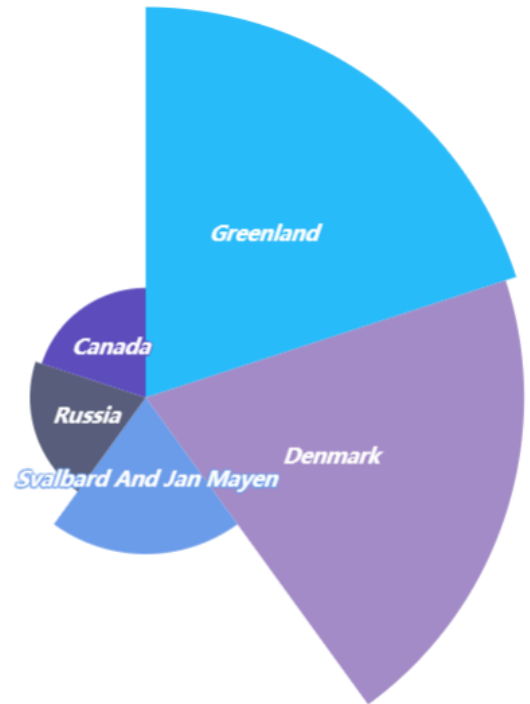
分析：图中取出温度最高的十个国家，在分析全球变暖问题时候，温度较高的城市需要重点关注其变化，也需要着重从这几个城市之中找到并分析引起变暖的成因，不能只归结到他们地理位置的温度本身就较高这一原因

(3) 绘制温度最低五个国家（玫瑰图）

```
country = [x[0] for x in low_ten_country] # 后五个国家
temperature = [x[1] for x in low_ten_country] # 后五个国家对应温度
# print(temperature)
color_series = randomcolor(len(country))
pie = Pie()
pie.add("", [list(i) for i in zip(country, temperature)], radius=[
    , rosetype='area'])
pie.set_global_opts(title_opts=opts.TitleOpts(title='温度最低五个国家'),
    legend_opts=opts.LegendOpts(is_show=False))
pie.set_series_opts(label_opts=opts.LabelOpts(is_show=True, position='inner', font_style='italic', font_family='Microsoft YaHei'))
pie.set_colors(color_series)
pie.render('平均温度最低的五个国家.html')
print('平均温度最高的五个国家玫瑰图已生成！！')
```

在线编程
吐槽

绘制出来为html格式无法上传图片，请查看附件中‘平均温度最低的五个国家.html’，下图为网页截图



分析：同理，在分析全球变暖问题时候，最低温度的几个城市也需要着重关注，因为全球变暖将导致这些冰川城市的冰川融化，分析全球变暖问题时候考虑这些国家的温度变化也显得十分重要

(4) 绘制中国与全球年温度折线图

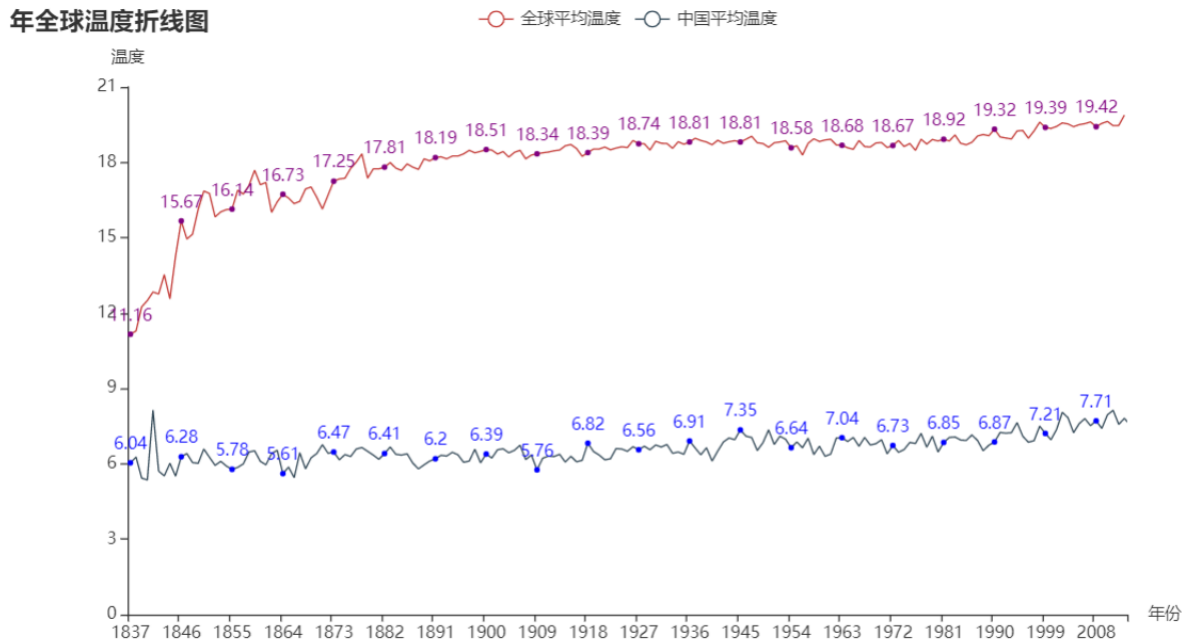
```
is_selected=True,
label_opts=opts.LabelOpts(is_show=True))
line.add_yaxis(
    series_name='中国平均温度',
    y_axis=y,
    is_selected=True,
    label_opts=opts.LabelOpts(is_show=True))
line.set_series_opts(label_opts=opts.LabelOpts(is_show=True))
line.set_global_opts(
    title_opts=opts.TitleOpts(title="年全球温度折线图"),
    tooltip_opts=opts.TooltipOpts(
        is_show=True,
        trigger="axis",
        axis_pointer_type="shadow"),

    xaxis_opts=opts.AxisOpts(name='年份'),
    yaxis_opts=opts.AxisOpts(name='温度'),
)
```




```
# line.render("中国与全球年温度折线图.html")
```

绘制出来为html格式无法上传图片，请查看附件中‘中国与全球年温度折线图.html’，下图为网页截图



分析：图中显示了中国和全球温度变化的1837到2013年数据，可以明显观察到全球变暖的大趋势，虽然中国在全球变暖问题上趋势变换较为缓慢，但并不是松懈的理由，更要分析为何波动较大，以及趋向于8°如何缓解的问题思考

(5) 绘制中国与全球预测温度图

```

        )
    )
    for i in range(0, 22):
        x1.append(
            opts.LineItem(
                name=year_set_China[i],
                value=round(predict_temp_China_list[i], 2),
                itemstyle_opts=opts.ItemStyleOpts(color='blue')
            )
        )
    for i in range(0, 22):
        x_dimensional.append(
            opts.LineItem(
                name=year_set[i],
                value=round(predict_temp_two_list[i], 2),
                itemstyle_opts=opts.ItemStyleOpts(color='green')
            )
        )
    line1 = Line()

```

在线编程
吐槽

```
line1.add xaxis(vear set)
```

绘制出来为html格式无法上传图片，请查看附件中‘中国与全球预测温度图.html’，下图为网页截图



分析：本图根据数据只有到2013为止的较为便捷利用的数据为回归样本，以线性回归方程以及多项式回归分别对2014到2035年的全球温度以及中国温度进行分析，并将预测样本中2014年到2020年与现在可获取数据进行对比可以发现，在预测模拟中，线性回归方程的全球温度预测模拟较为优秀，拟合程度高。也为人们敲响了警钟，如果按这个趋势的全球升温，不久人们将面临地球母亲的报复了，呼吁人们关注环境问题，关注全球变暖的问题。文明出行，为自己生活的家园做出自己的贡献。

(5) 将上述表格统一到一个网页中

```
page = Page()
page.add(map)
page.add(pie)
page.add(bar)
page.add(line)
page.add(line1)

page.render('全球温度可视化汇总.html')
print("全球温度可视化汇总已生成！！")
```

绘制出来为html格式无法上传图片，请查看附件中‘全球温度可视化汇总.html’，下图为网页截图



