2021-2022学年第2学期

**《Python程序设计》课程项目报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 生1 | 学号：2020302121048 | 姓名：曾舸舵 | 成绩： |
| 学 生2 | 学号：2020302011082 | 姓名：胡锐兵 | 成绩： |
| 学 生3 | 学号：2020302011088 | 姓名：王翔 | 成绩： |
| 学 生4 | 学号： | 姓名： | 成绩： |
| 学 生5 | 学号： | 姓名： | 成绩： |
| 指导教师 |  | | |
| 完成日期 | 2022年 月 日 | | |

## **1 选题**

* IMDB 电影评分数据
  + 统计参演电影类型最多的3位演员
  + 统计收益最高的三类电影
* 个人年收入调查数据
  + 受教育程度和收入之间的关系
* 全球新冠肺炎COVID-19 数据集
  + 全球疫情趋势变化
  + 全球各国确诊人数前30名

* + 全球各国疫情情况一览并统计确诊人数最多、最少的30个国家

* 安居客租房信息数据集
  + 考察安居客数据中，租金与房屋面积、地区的关系
* 商店会员购物数据
  + 不同年龄性别人群的消费能力
  + 用现金买水果的男士中哪个年龄段的最多

## **2 需求分析**

需求如选题所言。考虑到探索性数据分析没有固定模式而是依照数据进行处理，所以需求和设计都需要根据具体情境进行设计。

## **3 设计**

设计见代码。考虑到探索性数据分析没有固定模式而是依照数据进行处理，所以需求和设计都需要根据具体情境进行设计。

## **4 关键代码**

用到的包：

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from pylab import mpl

import re

from pyecharts.charts import Map,Page

from pyecharts import options as opts

### IMDB 电影评分数据

#### 统计参演电影类型最多的3位演员

此选题由曾舸舵、王翔采用不同实现方式处理。

曾舸舵部分：将数据读取为data\_imdb，并设置电影名称为index。

1. 观察数据后发现，Genre和Actors两列的数据是字符串形式，且分隔符为逗号和部分空格。将这两列的数据都处理成列表形式。

data\_imdb.loc[:, "Genre"] = data\_imdb["Genre"].str.split(r",\s\*")

data\_imdb.loc[:, "Actors"] = data\_imdb["Actors"].str.split(r",\s\*")

2. 建立演员与所演电影种类的表df， 并将初始值设为False

df = pd.DataFrame(index=actors, columns=genres)

df = df.fillna(False)

3. 逐行读取数据，将相应演员和类型所对应数据改为True

for item, row in data\_imdb[["Actors", "Genre"]].iterrows():

    for actor in row[0]:

        for genre in row[1]:

            df.loc[actor, genre] = True

4. 进行可视化，统计df中每个演员所演电影的类型数，显示前20个。

plt.figure(figsize=(18,14))

a = pd.DataFrame(df.sum(axis=1).nlargest(20), columns=["number"])

sns.barplot(data=a, y=a.index, x="number")

5. 经过观察，有1名演员第一，5名演员并列第二，显示这6名演员名称

top\_actors = [actor for actor in a.nlargest(6, "number").index]

top\_actors

['Brad Pitt', 'Hugh Jackman', 'Chloë Grace Moretz', 'Amy Adams', 'Scarlett Johansson', 'Johnny Depp']

王翔部分：将数据读取到df中，创建一个名为pairs的

1.将数据读取到df中，创建一个名为pairs的字典

df = pd.read\_csv('imdb\_movies.csv')

pairs = {}

2.将数据中的演员信息提取出来，存放在actors,并在对actors中的每个元素actor作为键，与该演员参演的电影类型作为值配对。

for i in range(len(df)):

actors = re.split(r' \*, \*', df.at[i, 'Actors'])

for actor in actors:

if pairs.get(actor, None) != None:

pairs[actor] |= set(re.split(r' \*, \*', df.at[i, 'Genre']))

else:

pairs[actor] = set(re.split(r' \*, \*', df.at[i, 'Genre']))

3.先将pairs按照actor的顺序排序，再将pairs中配对的actor与电影类型提取出来分别放入index与data中，并用以创建df1.

pairs = sorted(list(pairs.items()), key=lambda item: item[0])

index = [item[0] for item in pairs]

data = [item[1] for item in pairs]

df1 = pd.DataFrame({'演员': index, '类型': data})

4.df1中添加一列“数量”列（数量为该演员参演的电影类型数量），并按照电影类型数量大小降序排序，根据观察，发现“数量”第一第二人数共有六人。则再将“数量”第一、第二的演员的对应信息放入ans中,ans中的演员就是所求的参演电影类型最多的6个人。

df1['数量'] = df1['类型'].apply(lambda x: len(x))

df1.sort\_values('数量', ascending=False)

ans = df1[df1['数量']>= df1['数量'].max()-1].sort\_values('数量', ascending=False)

5.将字体设为仿宋，解决文字空白问题，将ans的数据用matplot可视化

mpl.rcParams['font.sans-serif']=['FangSong']

mpl.rcParams['axes.unicode\_minus']=False

plt.figure()

ans.plot(x='演员',y='数量',kind='bar')

plt.show()

6.显示参演电影类型最多的六个人名字

for i in ans['演员']:

print(i)

Brad Pitt

Amy Adams

Chloë Grace

Moretz

Hugh Jackman

Johnny Depp

Scarlett Johansson

#### 2. 统计收益最高的三类电影

此项目由胡锐兵处理

读取数据“imdb\_movies.csv”为movie

1.逐行读取数据，读取出每个电影类别及其收益，并将其装入列表res

res = []

for type in list(set(','.join(movie['Genre']).split(','))):

tmp = movie[movie['Genre'].str.contains(type)]

res.append([type,np.sum(tmp['Revenue(Millions)'])])

1. 依据收益将res进行排序

res.sort(key=lambda x: (x[1],x[0]),reverse=True)

words = [r[0] for r in res]

counts = [r[1] for r in res]

1. 为防止中文乱码，使用plt.rcParams

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

1. 利用matplotlib进行数据可视化，显示出收益最高的前三类电影的柱状图

plt.bar(words[:3],counts[:3])

plt.xlabel('电影类型')

plt.ylabel('总收益')

plt.title('收益最高的三类电影')

plt.show()

### 个人年收入调查数据

#### 受教育程度和收入之间的关系

此项目由曾舸舵和胡锐兵分别用不同方式处理。

曾舸舵部分：将数据读取为data\_income

1. 按照受教育年龄，将受教育程度按照类别进行排序并Category化，并将原始数据的受教育程度由object转变为Category。

cat\_degrees = data\_income[["edu", "edu\_num"]].sort\_values(['edu\_num'])["edu"].unique()

from pandas.api.types import CategoricalDtype

cat\_degrees\_order = CategoricalDtype(

    cat\_degrees,

    ordered=True

)

data\_income["edu"] = data\_income["edu"].astype(cat\_degrees\_order)

1. 按照受教育程度分类，并统计相同受教育程度下，收入的占比

df\_edu\_icm = data\_income.groupby(["edu"])['income'].value\_counts(normalize=True).unstack() #顺便标准化

df\_edu\_icm

1. 利用热图进行可视化

plt.figure(figsize=(18,14))

# sns.set(font\_scale = 2)

sns.heatmap(df\_edu\_icm, annot=True, cmap="YlGnBu")

胡锐兵部分：将数据读取为dat

1. 将dat里的income小于等于50k的改为0，将大于50k的改为1

dat['income'] = dat['income'].map({'<=50k':0,'>50k':1}]

1. 逐行读取数据，获得每个‘edu\_num’及其对应的‘income'指标，并放在res列表中

res = []

for edutype in dat['edu\_num'].unique():

tmp=dat[dat[edu\_num']==edutype]

res.append([edutype,np.mean(tmp['income'])])

1. 利用matplotlib进行数据可视化,绘出受教育程度和收入关系的柱状图

plt.bar([i[0] for i in res],[i[1] for i in res])

plt.xlabel('受教育程度')

plt.ylabel('平均收入')

plt.title('受教育程度和收入的关系')

plt.show()

由此可发现，受教育程度越高，收入越高。

### 全球新冠肺炎COVID-19 数据集

#### 全球疫情趋势变化

该项目由胡锐兵完成：

读取全球确诊数据为confirmed，全球死亡数据为deaths，全球康复数据为recovered

1. 将confirmed、deaths、recovered的每日总人数求出来,并每隔七天取一次数据点

plt.figure(figsize=(15,6))

res = []

for col in confirmed.columns[4:]:

res.append([col,np.sum(confirmed[col])])

plt.scatter(['-'.join(res[i][0].split('/')[:2]) for i in range(len(res)) if i%7==0],[res[i][1] for i in range(len(res))if i%7==0],label='confirmed')

res=[]

for col in deaths.columns[4:]:

res.append([col,np.sum(deaths[col])])

plt.scatter(['-'.join(res[i][0].split('/')[:2]) for i in range(len(res)) if i%7==0],[res[i][1] for i in range(len(res))if i%7==0],label='deaths')

res=[]

for col in recovered.columns[4:]:

res.append([col,np.sum(recovered[col])])

plt.scatter(['-'.join(res[i][0].split('/')[:2]) for i in range(len(res)) if i%7==0],[res[i][1] for i in range(len(res))if i%7==0],label='recovered')

2.数据可视化绘图

plt.xlabel('日期')

plt.ylabel('人数')

plt.title('全球疫情趋势变化')

plt.legend()

plt.show()

#### 全球各国确诊人数前30名

该选题由胡锐兵完成：

1.定义绘制玫瑰图函数

def rosetype\_pie(country,confirmed,size,colors):

plt.rcParams['font.sans-serif'] = ['SimHei'] # 中文显示

num= len(size) # 柱子的数量

width = 2 \* np.pi / num # 每个柱子的宽度

rad = np.cumsum([width] \* num) # 每个柱子的角度

plt.figure(figsize=(8, 8),dpi=500,) # 创建画布

ax = plt.subplot(projection='polar')

ax.set\_ylim(-1, np.ceil(max(size) + 1)) # 中间空白,-1为空白半径大小，可自行调整

ax.set\_theta\_zero\_location('N',-5.0) # 设置极坐标的起点方向 W,N,E,S, -5.0为偏离数值，可自行调整

ax.set\_theta\_direction(1) # 1为逆时针，-1为顺时针

ax.grid(False) # 不显示极轴

ax.spines['polar'].set\_visible(False) # 不显示极坐标最外的圆形

ax.set\_yticks([]) # 不显示坐标间隔

ax.set\_thetagrids([]) # 不显示极轴坐标

ax.bar(rad, size, width=width, color=colors, alpha=1) # 画图

ax.bar(rad, 1, width=width, color='white', alpha=0.15) # 中间添加白色色彩使图案变浅

ax.bar(rad, 3, width=width, color='white', alpha=0.1) # 中间添加白色色彩使图案变浅

ax.bar(rad, 5, width=width, color='white', alpha=0.05) # 中间添加白色色彩使图案变浅

ax.bar(rad, 7, width=width, color='white', alpha=0.03) # 中间添加白色色彩使图案变浅

# 设置text

for i in np.arange(num):

if i < 8:

ax.text(rad[i], # 角度

size[i]-0.2, # 长度

country[i]+'\n'+str(confirmed[i])+'例', # 文本

rotation=rad[i] \* 180 / np.pi -5, # 文字角度

rotation\_mode='anchor',

# alpha=0.8,#透明度

fontstyle='normal',# 设置字体类型，可选参数[ ‘normal’ | ‘italic’ | ‘oblique’ ]，italic斜体，oblique倾斜

fontweight='black', # 设置字体粗细，可选参数 [‘light’, ‘normal’, ‘medium’, ‘semibold’, ‘bold’, ‘heavy’, ‘black’]

color='white', # 设置字体颜色

size=size[i]/2.2, # 设置字体大小

ha="center", # 'left','right','center'

va="top", # 'top', 'bottom', 'center', 'baseline', 'center\_baseline'

)

elif i < 15:

ax.text(rad[i]+0.02,

size[i]-0.7,

country[i] + '\n' + str(confirmed[i]) + '例',

fontstyle='normal',

fontweight='black',

color='white',

size=size[i] / 1.6,

ha="center",

)

else:

ax.text(rad[i],

size[i]+0.1,

str(confirmed[i]) + '例 ' + country[i],

rotation=rad[i] \* 180 / np.pi + 85,

rotation\_mode='anchor',

fontstyle='normal',

fontweight='black',

color='black',

size=4,

ha="left",

va="bottom",

)

plt.show()

2.求出2020年10月11日确诊数前30的国家：读取全球确诊数据为confirmed

confirmed = pd.read\_csv('time\_series\_covid19\_confirmed\_global.csv') # 利用pandas读取数据

confirmed=confirmed.groupby('Country/Region')['10/11/20'].sum().reset\_index()

confirmed.sort\_values(by='10/11/20',ascending=False,inplace=True)

file = confirmed.head(30)

3.将参数country、confirmed、size、colors设置好并填入函数，生成2020年10月11日确诊人数前30的国家的玫瑰图（由于代码长度关系，此处省略size和colors的设置代码）

country = file['Country/Region'].tolist()

confirmed = file['10/11/20'].tolist()

rosetype\_pie(country, confirmed, size, colors)

#### 全球各国疫情情况一览并统计确诊人数最多、最少的30个国家

该项目由王翔完成：

1.提取2020年10月11日全球各国新冠疫情确诊人数并降序排序，并提取出确诊人数最多、最少的30个国家的信息，分别储存在most、least里。

pd.set\_option('display.unicode.ambiguous\_as\_wide', True)

pd.set\_option('display.unicode.east\_asian\_width', True)

df = pd.read\_csv('time\_series\_covid19\_confirmed\_global.csv')

df=df.groupby('Country/Region')['10/11/20'].sum().reset\_index()

df.sort\_values(by='10/11/20',ascending=False,inplace=True)

df['Country/Region']=df['Country/Region'].str.replace('US','United States')

most=df.head(30)

least=df.tail(30)

2.提取国家名称与确诊人数，首先绘制全球各国新冠疫情一览图

data1=list(df['Country/Region'])

data2=list(df['10/11/20'])

map = Map( init\_opts=opts.InitOpts(width="1000px", height="600px", bg\_color="#ADD8E6",

page\_title="2020年10月11日全球疫情确诊人数",theme="white"))

map.add("确诊人数",[list(z) for z in zip(data1,data2)],is\_map\_symbol\_show=False,

maptype="world",label\_opts=opts.LabelOpts(is\_show=False),

itemstyle\_opts=opts.ItemStyleOpts(color="rgb(49,60,72)"))

map.set\_global\_opts(title\_opts = opts.TitleOpts(title='10/11/20 全球疫情确诊人数'),

legend\_opts=opts.LegendOpts(is\_show=False),

visualmap\_opts = opts.VisualMapOpts(max\_=100000))

map.render('world\_map.html')

3.提取确诊人数最多30个国家名称与确诊人数，绘制图像

data1=list(most['Country/Region'])

data2=list(most['10/11/20'])

map = Map( init\_opts=opts.InitOpts(width="1000px", height="600px", bg\_color="#ADD8E6",

page\_title="2020年10月11日全球疫情确诊人数",theme="white"))

map.add("确诊人数",[list(z) for z in zip(data1,data2)],is\_map\_symbol\_show=False,

maptype="world",label\_opts=opts.LabelOpts(is\_show=False),

itemstyle\_opts=opts.ItemStyleOpts(color="rgb(49,60,72)"))

map.set\_global\_opts(title\_opts = opts.TitleOpts(title='10/11/20 全球疫情确诊人数最多30个国家'),

legend\_opts=opts.LegendOpts(is\_show=False),

visualmap\_opts = opts.VisualMapOpts(max\_=100000))

map.render('worldmost\_map.html')

data1

['United States', 'India', 'Brazil', 'Russia', 'Colombia', 'Argentina', 'Spain', 'Peru', 'Mexico', 'France', 'South Africa', 'United Kingdom', 'Iran', 'Chile', 'Iraq', 'Bangladesh', 'Italy', 'Philippines', 'Saudi Arabia', 'Turkey', 'Indonesia', 'Germany', 'Pakistan', 'Israel', 'Ukraine', 'Canada', 'Netherlands', 'Belgium', 'Romania', 'Morocco']

4.提取确诊人数最少30个国家名称与确诊人数，绘制图像

data1=list(least['Country/Region'])

data2=list(least['10/11/20'])

map = Map( init\_opts=opts.InitOpts(width="1000px", height="600px", bg\_color="#ADD8E6",

page\_title="2020年10月11日全球疫情确诊人数",theme="white"))

map.add("确诊人数",[list(z) for z in zip(data1,data2)],is\_map\_symbol\_show=False,

maptype="world",label\_opts=opts.LabelOpts(is\_show=False),

itemstyle\_opts=opts.ItemStyleOpts(color="rgb(49,60,72)"))

map.set\_global\_opts(title\_opts = opts.TitleOpts(title='10/11/20 全球疫情确诊人数最少30个国家'),

legend\_opts=opts.LegendOpts(is\_show=False),

visualmap\_opts = opts.VisualMapOpts(max\_=100000))

map.render('worldleast\_map.html')

data1

['Sao Tome and Principe', 'San Marino', 'Diamond Princess', 'Papua New Guinea', 'Taiwan\*', 'Burundi', 'Tanzania', 'Comoros', 'Eritrea', 'Mauritius', 'Mongolia', 'Bhutan', 'Cambodia', 'Monaco', 'Barbados', 'Seychelles', 'Brunei', 'Liechtenstein', 'Antigua and Barbuda', 'Saint Vincent and the Grenadines', 'Dominica', 'Fiji', 'Saint Lucia', 'Timor-Leste', 'Grenada', 'Laos', 'Saint Kitts and Nevis', 'Holy See', 'Western Sahara', 'MS Zaandam']

### 安居客租房信息数据集

#### 考察安居客数据中，租金与房屋面积、地区的关系

此选题由曾舸舵处理。将原始数据读取为data\_ajk。

1. 将"房屋面积"一列由字符串转化为以平方米为单位的int型数据，将"地区"一列数据去掉"安居客"字样。

data\_ajk["房屋面积"] = data\_ajk["房屋面积"].map(lambda string: int(string[:-2]))

data\_ajk["地区"] = data\_ajk["地区"].map(lambda string: string[:-3])

1. 考察过房屋面积的最大值和最小值后，将房屋面积数据分段和分类,以50平方米为间隔，记为area。

area = pd.cut(data\_ajk["房屋面积"], bins=range(0, 2050, 50))

1. 透视数据，查看不同地区，不同房屋面积的租金的最大值，同时将将缺失值填为0

pd.pivot\_table(data\_ajk, "租金", area,  ["地区"], aggfunc=np.max).fillna(0)

1. 用热图进行可视化

plt.figure(figsize=(18,14))

sns.set(font\_scale=1.2, font="Simhei")

sns.heatmap(tmp, annot=True, cmap="OrRd")

### 5. 商店会员购物数据

#### 不同年龄性别人群的消费能力

此选题由曾舸舵负责。将数据读取为data\_shopping。

以5岁为间隔划分年龄段, 并且透视数据，查看不同性别，不同年龄人群的平均消费额

age = pd.cut(data\_shopping["年龄"], bins=range(15, 55, 5))

data\_stat = data\_shopping.pivot\_table("消费额", ["性别", age]).unstack()

data\_stat

用柱状图进行可视化

sns.set(context='notebook', font='SimHei', font\_scale=2)

plt.figure(figsize=(18,14))

sns.barplot(x=age, y="消费额", hue="性别", data=data\_shopping)

#### 用现金买水果的男士中哪个年龄段的最多

此选题由王翔负责。将数据读取到df中。

1. 以“付款方式”、“性别”、“物品”为判断标准，筛选出用现金购买水果的男士的年龄信息，放入列表age中。

age=[]

for i in range(len(df)):

if df.at[i,'付款方式']=='CASH':

if df.at[i,'性别']=='M':

if df.at[i,'物品']=='水果':

age.append(df.at[i,'年龄']

2.找到age中的最大值最小值

print(max(age))

print(min(age))

50 16

3.根据观察，决定以10岁为一个年龄段划分，将age中的数据在[10,20)的改为10，[20,30)的改为20，[30,40)改为30，[40,50)的改为40，[50,60)的改为50，并创建一个列表y

for i in range(len(age)):

if age[i]>=10 and age[i]<20:

age[i]=10

if age[i]>=20 and age[i]<30:

age[i]=20

if age[i]>=30 and age[i]<40:

age[i]=30

if age[i]>=40 and age[i]<50:

age[i]=40

if age[i]>=50 and age[i]<60:

age[i]=50

y=[]

4.分别将age中10、20、30、40、50的数量放入y中，并找到y中最大的数，输出用现金买水果的男士最多的年龄段

for i in range(5):

y.append(age.count(10\*i+10))

max=y[0]

t=1

for i in range(5):

if y[i]>=max:

max=y[i]

t=i+1

print(f"用现金买水果的男士最多的年龄段为：{t\*10}到{t\*10+10}")

5.创建x列表来存储x轴，设字体为仿宋，处理文字空白，设置标题为用现金买水果的男士年龄段分布图，x轴命名为年龄段，y轴命名为人数，作图

x=['10-20','20-30','30-40','40-50','50-60']

mpl.rcParams['font.sans-serif']=['FangSong']

mpl.rcParams['axes.unicode\_minus']=False

plt.bar(x=x, height=y, label='Age',alpha=0.8)

plt.title('用现金买水果的男士年龄段分布图')

plt.xlabel("年龄段")

plt.ylabel("人数")

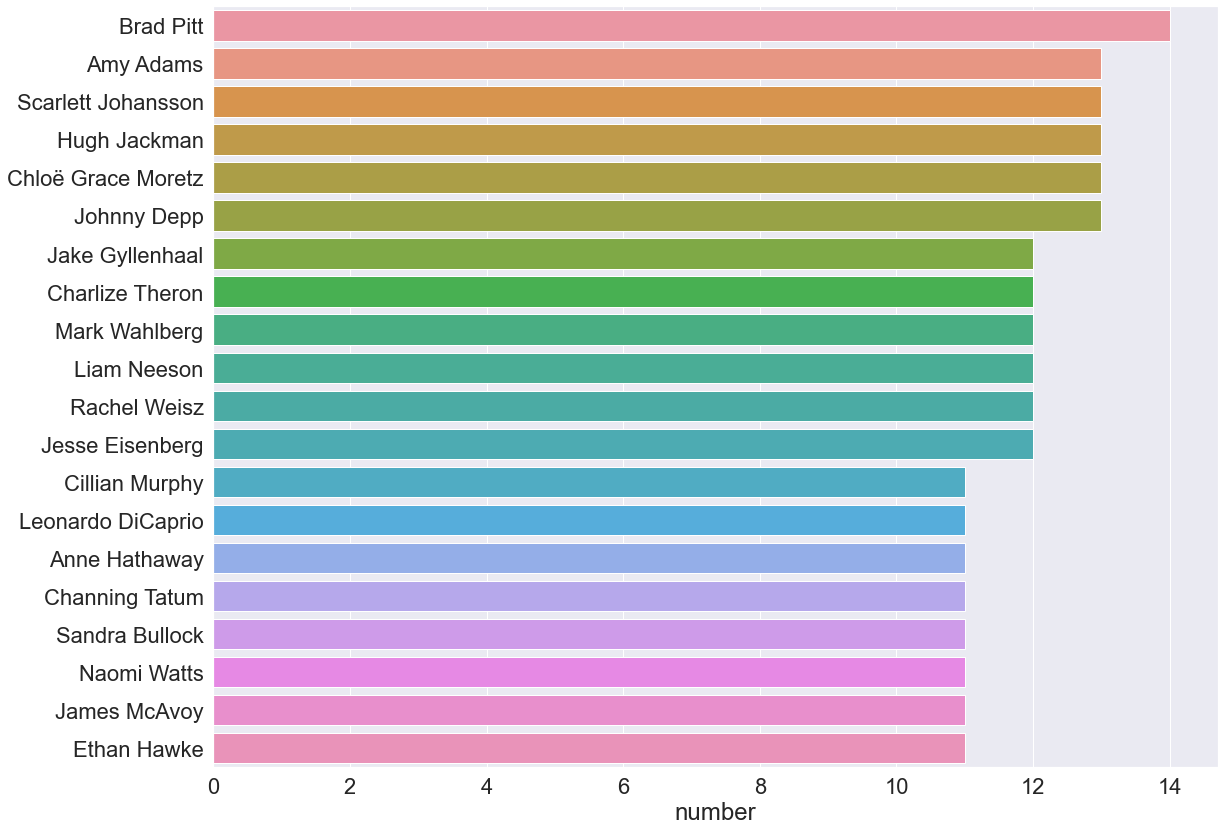
plt.show()

## **5 运行效果**

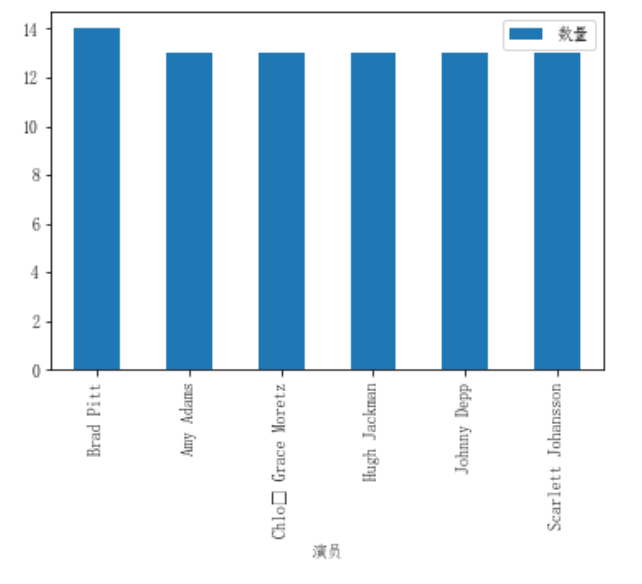
### IMDB 电影评分数据

#### 1. 统计参演电影类型最多的3位演员

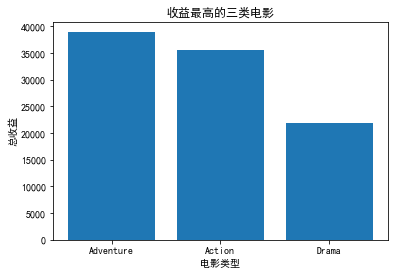
曾舸舵：



王翔：



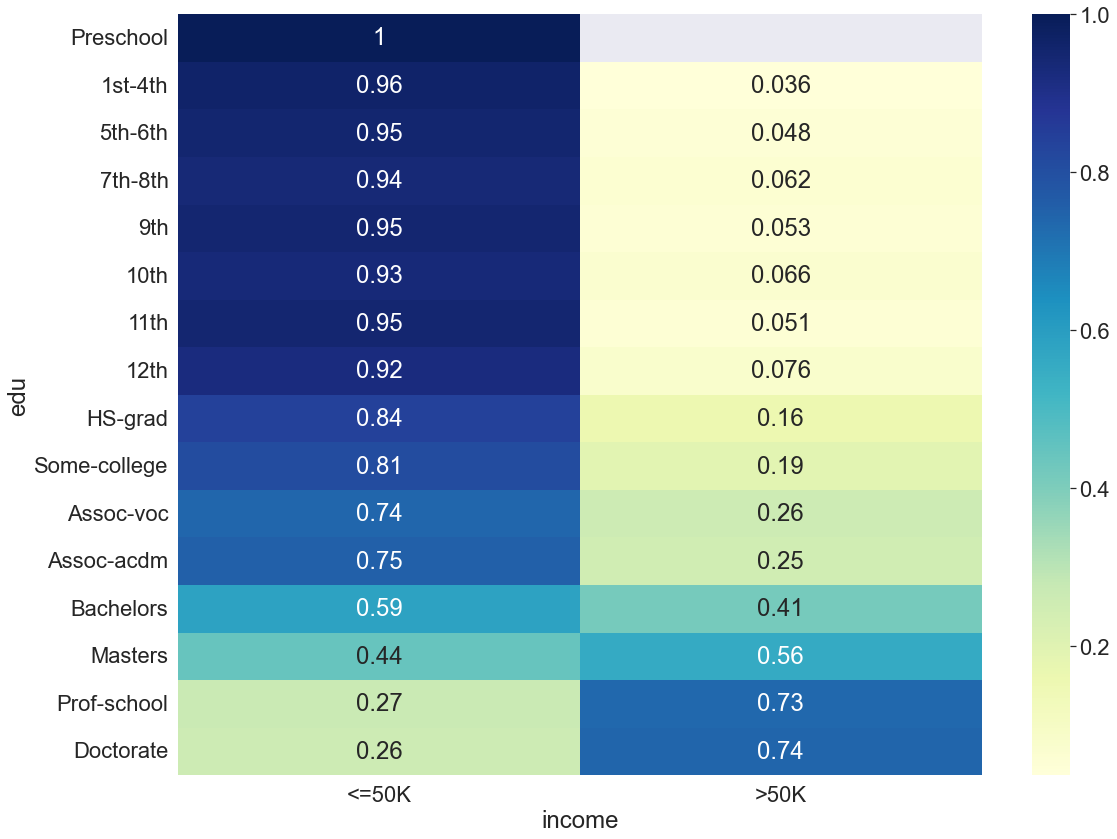
#### 2. 统计收益最高的三类电影



### 个人年收入调查数据

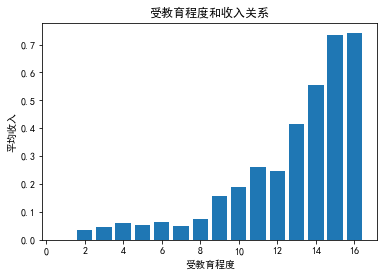
#### 受教育程度和收入之间的关系

曾舸舵：



由此可见， 在硕士学位以下，收入<=50k居多。在硕士学位以上，收入>50k居多。所以受教育程度越高，越高收入就越高

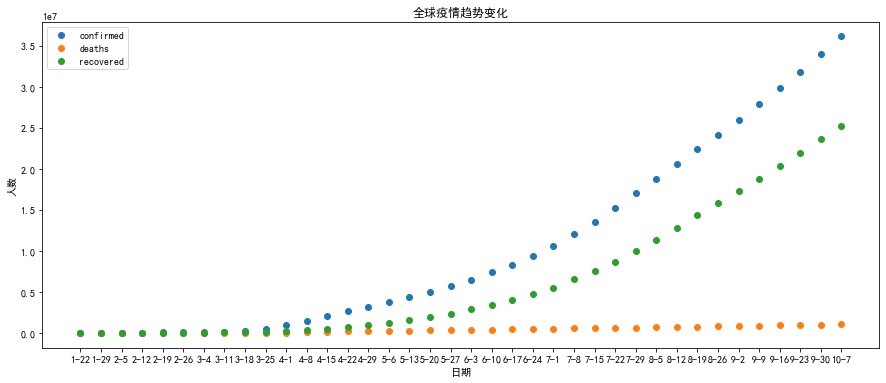
胡锐兵：



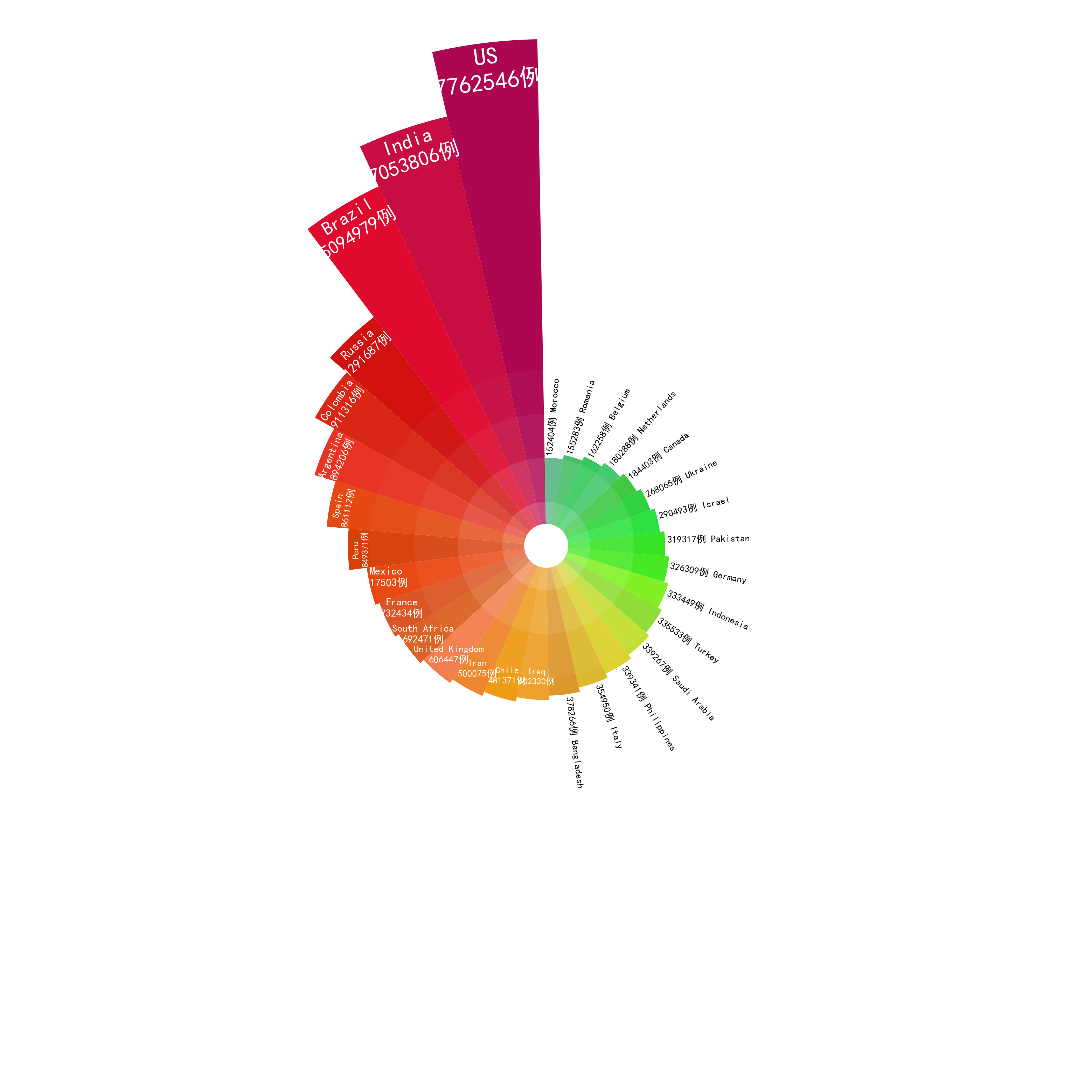
由图可知，大体上受教育程度越高，平均收入越高。

### 全球新冠肺炎COVID-19 数据集

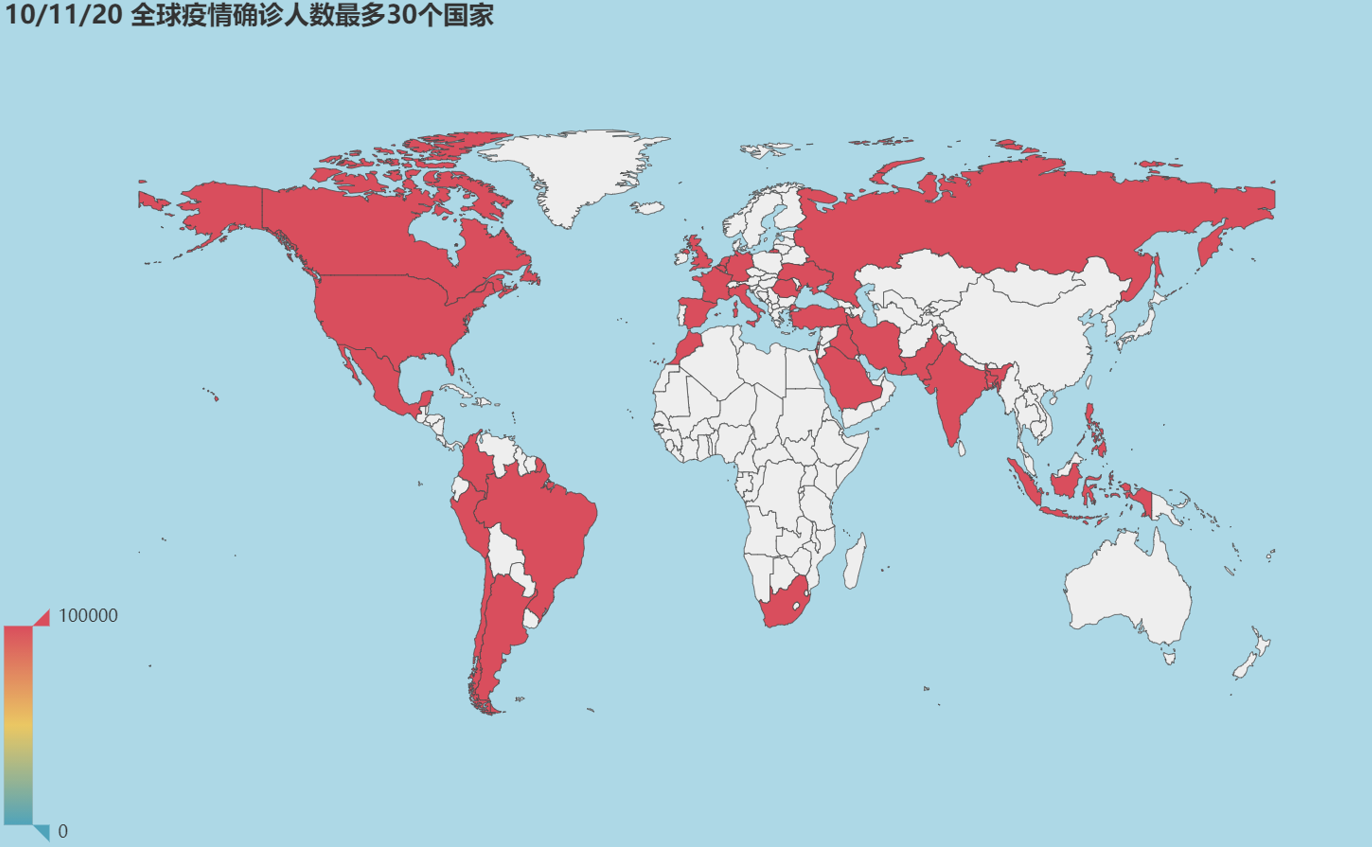
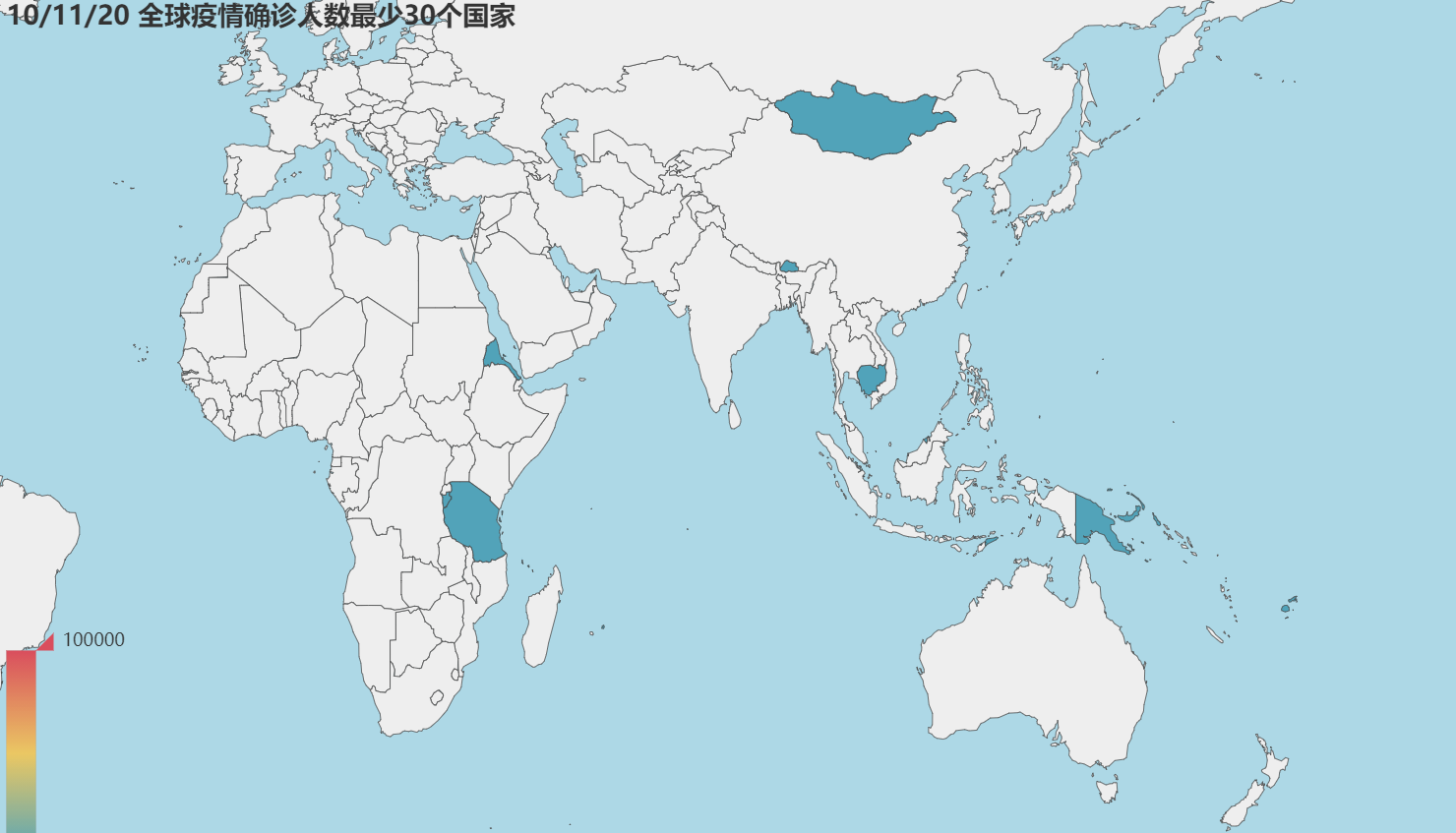
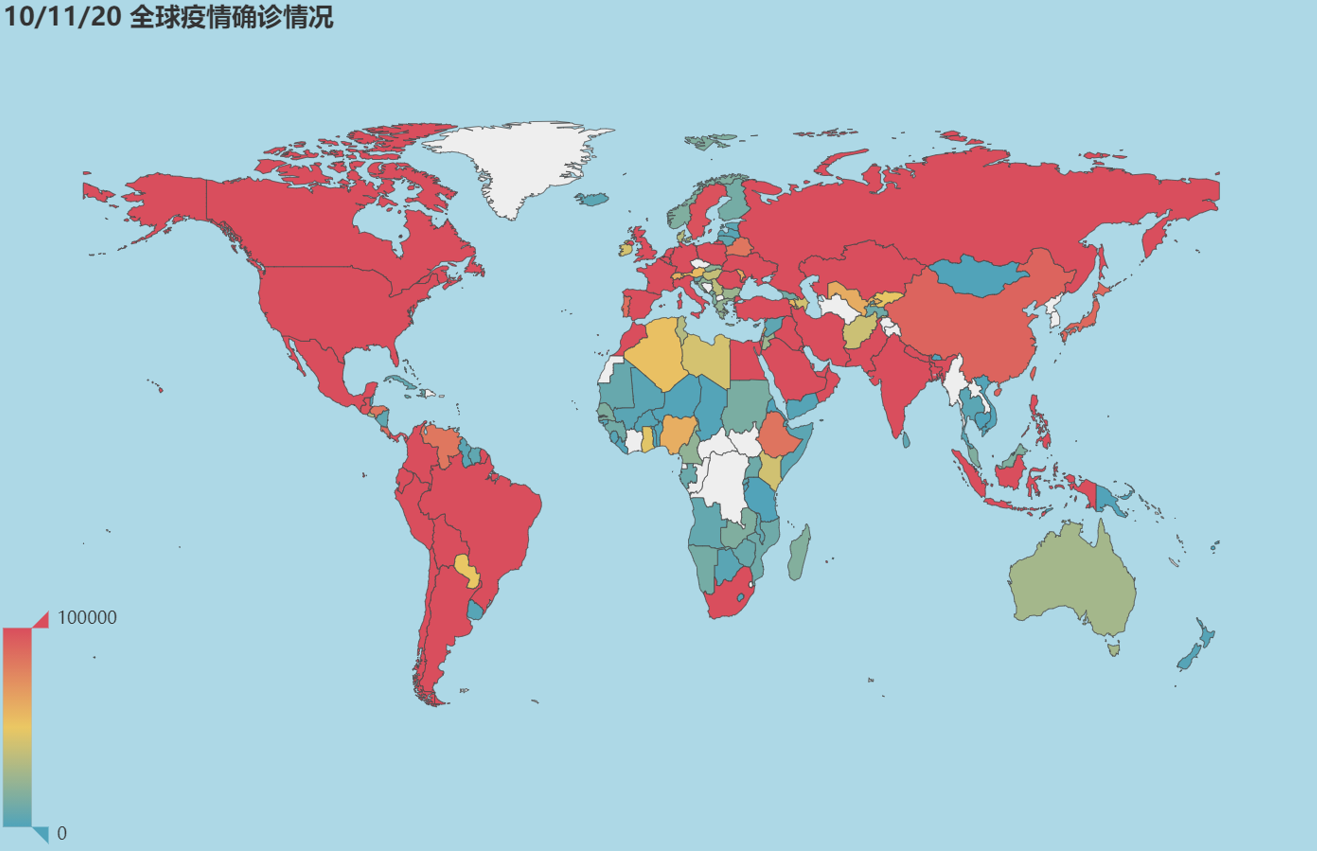
#### 全球疫情趋势变化



#### 全球各国确诊人数前30名

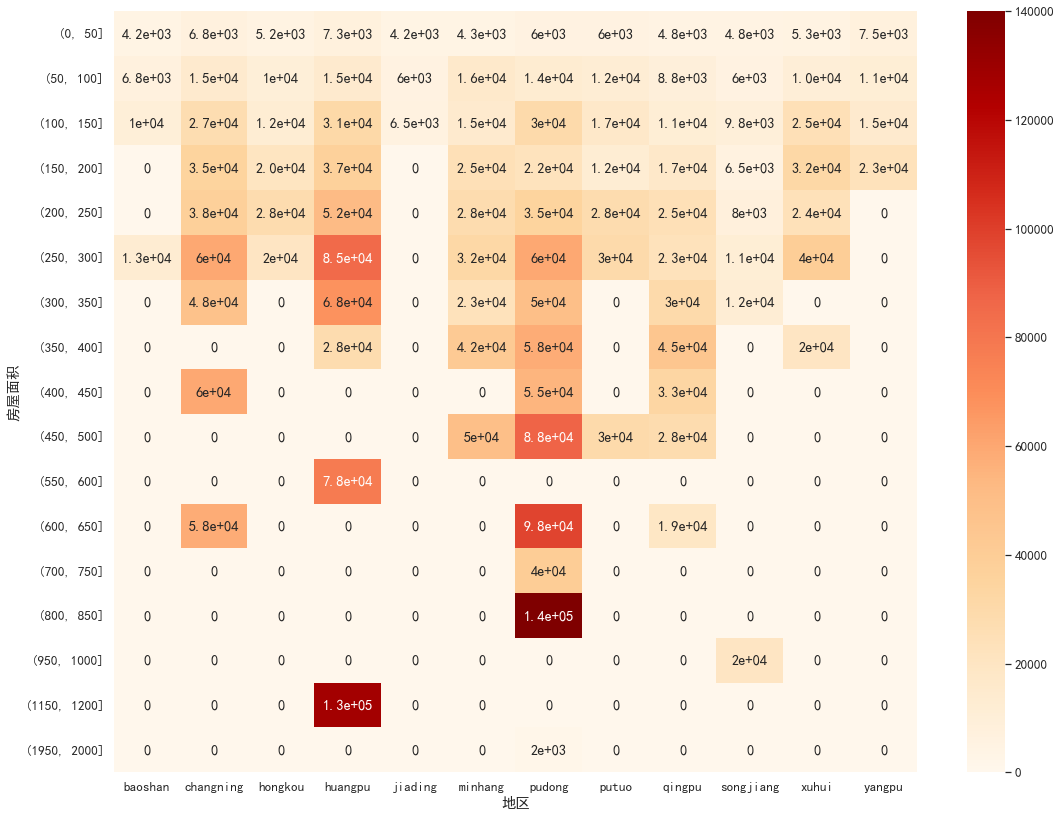


#### 全球各国疫情情况一览并统计确诊人数最多、最少的30个国家



### 4. 安居客租房信息数据集

#### 考察安居客数据中，租金与房屋面积、地区的关系

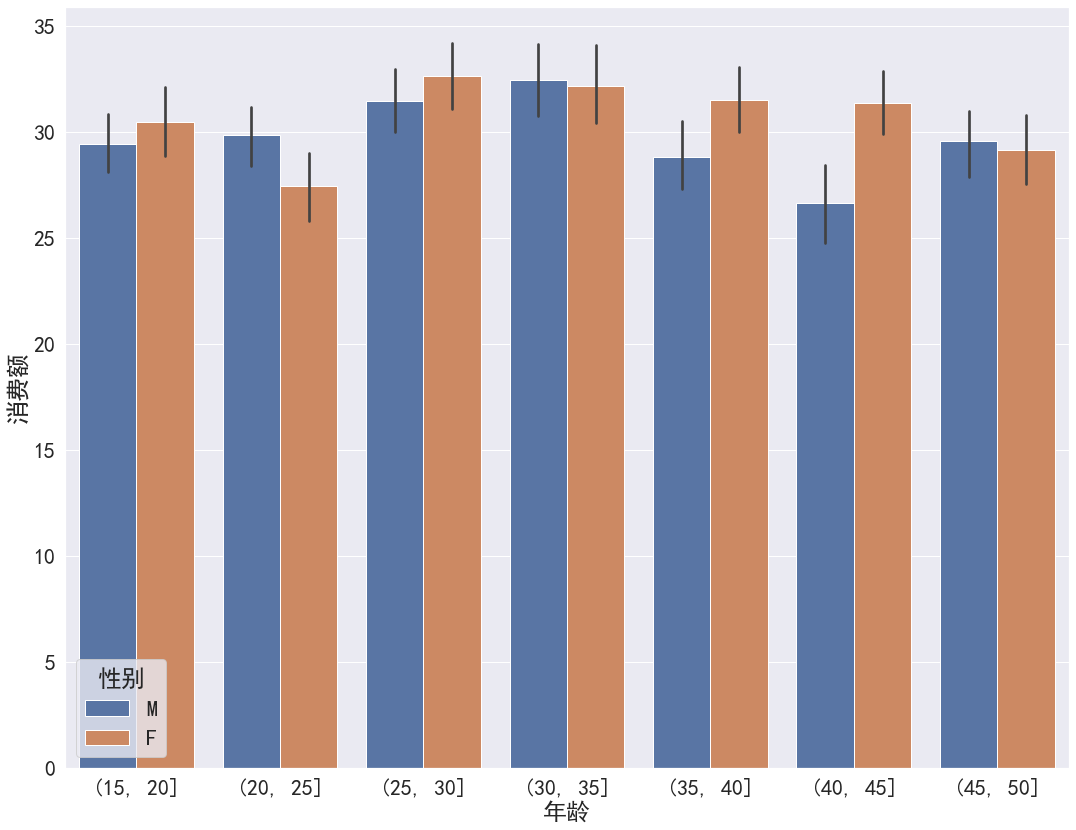


可以看出，同等房屋面积下，pudong（浦东）地区租金更贵，而浦东地区有房屋面积介于(1950, 2000]处疑似存在数据填写失误。

整体来说，相同地区下，房屋面积越大，租金越贵

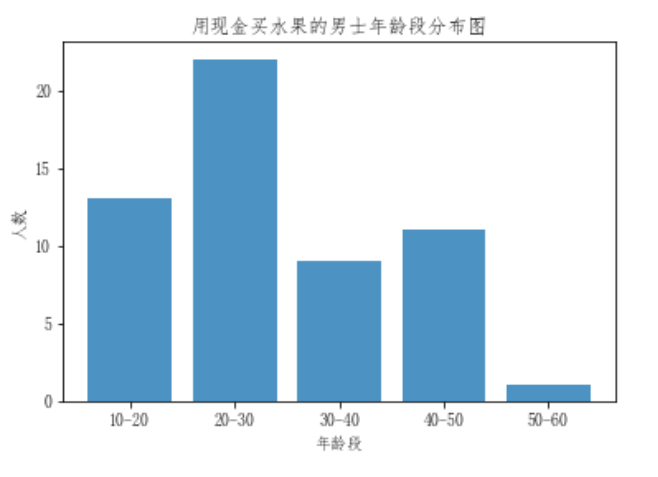
### 5. 商店会员购物数据

#### 不同年龄性别人群的消费能力



由此可见各个年龄段不同性别的消费水平

#### 用现金买水果的男士年龄段分布图：



由此可见20-30年龄段的男士用现金买水果最多。

## **6 总结**

完成了选题内容。在探索性数据分析中加深了对数据分析所需包的理解与应用。不足在于仍然不是十分熟练。

## **7 评语**

|  |  |
| --- | --- |
| 教师  评语 |  |