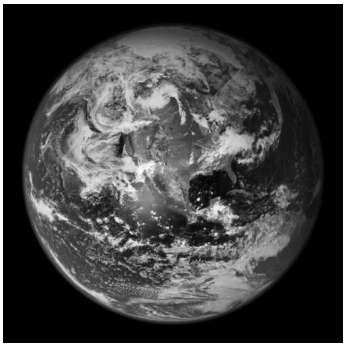
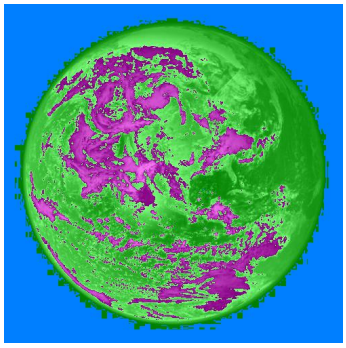
可视化作业 5

张言健 16300200020

1 设计灰度向彩色（伪彩）变换的算法、实现代码并用图片进行测试。

```
image = io.imread("./image/earth_grey.jpg")

pseudoimage = np.zeros((height, width, 3), dtype='float64')

for i in range(height):
    for j in range(width):
        pseudoimage[i][j][0] = 255.0-255.0*image[i][j]
        pseudoimage[i][j][1] = 127.0-255.0*image[i][j]
        pseudoimage[i][j][2] = 0.0-255.0*image[i][j]


pseudoimage = dtype(pseudoimage)

io.imsave("./image/earth_pseudo.jpg", pseudoimage)
plt.show()
```

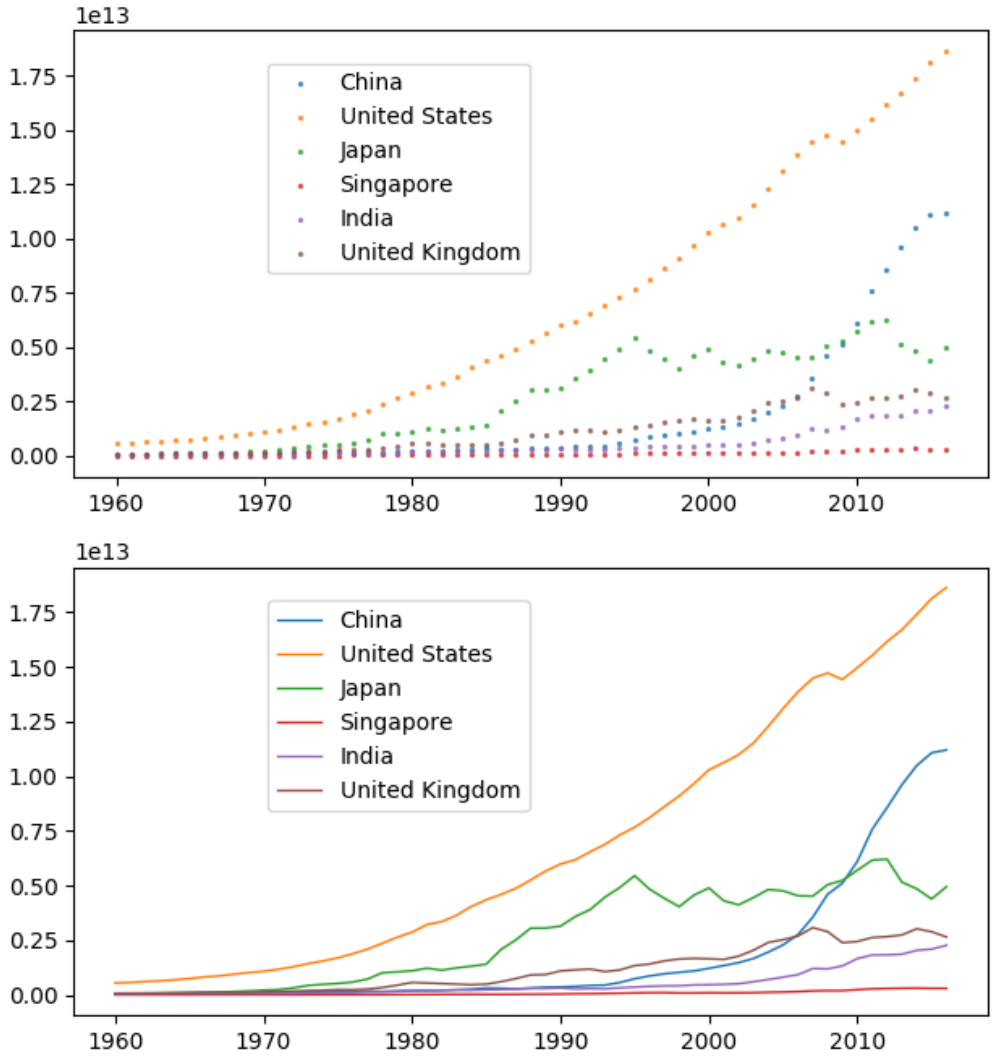
原彩	灰度图	伪彩
		

2 请使用世界各国GDP总量数据（从http://www.gapminder.org网站下载；或从elearning下载），(1) 用折线、散点做一个完整可视化图，显示世界各国20年的GDP数值；(2) 使用地图做图，显示世界各国GDP在20年来的动态变化。建议选择显示至少超过5个国家的数据。

```
df = pd.read_excel("GDP-fromworldbank.xls", skiprows=3, index_col=0)
countries = ['China', "United States", "Japan",
             "Singapore", "India", "United Kingdom"]
country_data = [df.loc[country][3:-1] for country in countries]
years = [y for y in range(1960, 2017)]
fig = plt.figure()
ax1 = plt.subplot(211)
ax2 = plt.subplot(212)
for i, country in enumerate(countries):
    x = years
    y = [t for t in country_data[i]]
    ax1.scatter(x, y, s=2, alpha=0.7,   label=country)
    ax2.plot(x, y, linewidth=1.0,  label=country)
```

```
    ax1.legend(loc='upper left', bbox_to_anchor=(0.2, 0.95))
    ax2.legend(loc='upper left', bbox_to_anchor=(0.2, 0.95))
```

Result

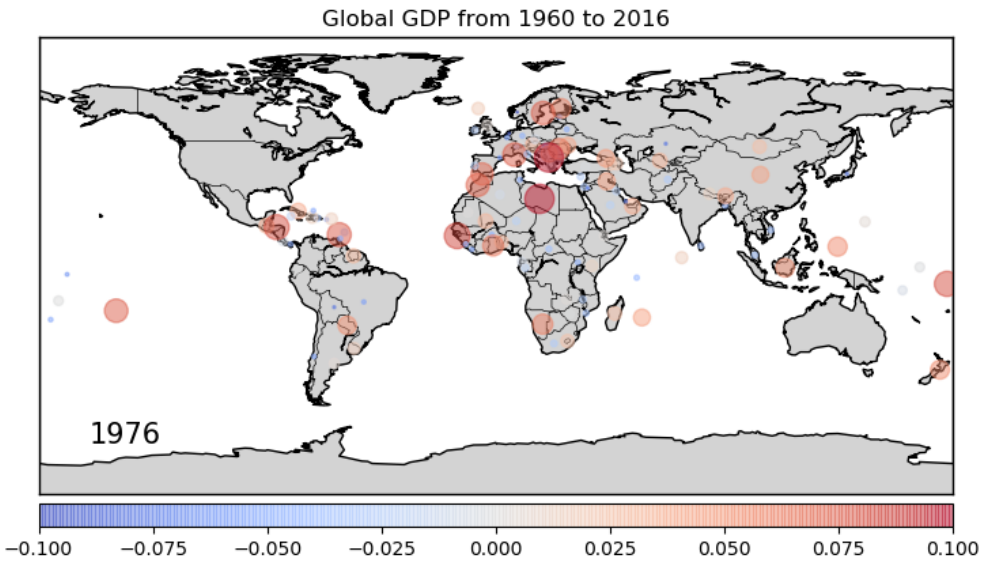


我们从[country2location]下载的文件中获取了国家对应的经纬度信息

```
with open("countries_latitude_longitude.json", 'r') as load_f:
    latlon_list = json.load(load_f)

# Map country code to latitude and longitude
name2lat = {}
name2long = {}
for item in latlon_list:
    name2lat[item["name"]] = item["latitude"]
    name2long[item["name"]] = item["longitude"]

df_loc = pd.DataFrame(columns=['lat', 'long'], data=[
                      [name2lat[name.strip()], name2long[name.strip()]] if name
in name2lat and name in name2long else [None, None] for name in df["Country
Name"]])
```

```
df = pd.concat([df, df_loc])


# Calculate GDP for each countries and year
MAX_GDP = np.max(df[years].fillna(0.00001))
MIN_GDP = np.min(df[years].fillna(0.00001))
current_gdp = df[years[0]].fillna(0.00001)
log_gdp = np.log10(current_gdp)
rate = (log_gdp - np.min(log_gdp)) / \
    (np.max(log_gdp) - np.min(log_gdp))

cmap = plt.get_cmap('coolwarm')
year_legend = plt.text(-160, -70, str(years[0]), fontsize=15)
x, y = map(df['long'], df['lat'])
scat = map.scatter(x, y, (1+rate)**8, marker='o',
                   alpha=0.5, zorder=10,
                   cmap=cmap, c=rate)
cbar = map.colorbar(scat, location='bottom')
ani = animation.FuncAnimation(fig, update, interval=200, init_func=init,
                              frames=2017-1960, blit=True)

def update(latency):
    print(years[latency])
    current_gdp = df[years[latency]]
    log_gdp = np.log10(current_gdp)
    rate = (log_gdp - np.min(log_gdp)) /  (np.max(log_gdp) - np.min(log_gdp))
    scat.set_sizes((1+rate)**8)
    year_legend.set_text(years[latency])
    scat.set_color(cmap(rate))
    # scat.set_color(log_gdp)
    return scat, year_legend,


def init():
    return scat, year_legend,
```

Result:



3 请使用地震数据（从http://www.r-project.org下载叫quakes数据；或从elearning下载），使用地图可视化的方法对数据进行可视化，展现地震的地点。

```
quakes = pd.read_csv("quakes.csv")

fig = plt.figure(figsize=(10, 10))


map = Basemap(projection='merc', resolution='l', area_thresh=1200.0,
              lat_0=0, lon_0=130,
              llcrnrlon=min(quakes.long), llcrnrlat=min(quakes.lat),
              urcrnrlon=max(quakes.long), urcrnrlat=max(quakes.lat))

plt.title("Earthquakes Location")

map.drawcoastlines()
map.drawcountries()

map.drawmapboundary(fill_color='powderblue')
```

```
map.fillcontinents(color='#ddaa66', lake_color='aqua')
max_depth = max(quakes.depth)
for i, item in quakes.iterrows():
    x, y = map(item.long, item.lat)
    # 点越大，表示地震强度越大
    makersize = item.mag ** 3
    # 颜色越深，表示震源越深
    colors = np.array([0.2, 0.0, item.depth/(max_depth+100)])
    map.scatter(x, y, makersize, color=colors, marker='o',
                alpha=0.5)

plt.show()
```

Result