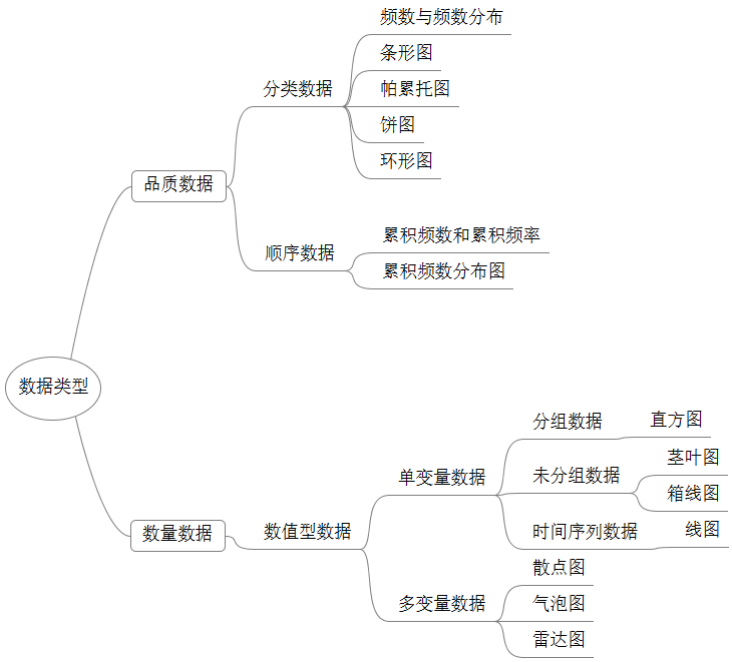
Task

- 数据预处理：数据审核、筛选、排序
- 品质数据的整理与展示：分类数据的整理与展示、顺序数据的整理与展示
- 数值型数据的整理与展示：数据分组、数据展示

概念思维导图



一、数据预处理

- **数据审核:** *检查数据中是否有错误*
  1. 原始数据，主要从完整性和准确性两方面去审核
  2. 完整性审核主要是检查应调查的单位或者个体是否有遗漏，所有的调查项目是否填写齐全等。
  3. 准确性审核主要是检查数据是否有错误，是否存在异常值，对于异常值视情况有所取舍。
  4. 对于二手数据，要考虑数据的时效性和适用性。
- **数据筛选：** *根据需要找出符合条件的某类数据*

```
import pandas as pd
df = pd.read_excel('data/1_table1.xlsx')
print(df)

#找出统计学成绩等于75的学生
df1 = df[df['统计学成绩']==75]
print(df1)
```

- **数据排序**：*寻找数据的基本特征*
    1. 概念： 数据排序是指按一定顺序将数据排列，以便研究者通过浏览数据发现一些明显的特征或趋势，找到解决问题的线索。除此之外，排序还有助于对数据进行检查纠错，以及为重新归类或分组等提供方便。
    2. 举例：美国的《财富》杂志每年都要在全世界范围内排出500强企业，通过这一信息，不仅可以了解自己企业所处的位置，清楚自己的差距，还可以从一个侧面了解到竞争对手的状况，有效制定企业的发展规划和战略目标。
    3. 对于分类数据，如果是字母型数据，排序则有升序、降序之分，但习惯上升序用得更多，因为升序与字母的自然排列相同。
    4. 对于数值型数据，排序只有两种，即递增和递减。
- 数据透视表： *对数据表中重要信息进行汇总和作图

```
import pandas as pd
import numpy as np
df = pd.read_excel('data/1_table2.xlsx')

print(df)

df1 = pd.pivot_table(df, index=['性别'])
print(df1)

df2 = pd.pivot_table(df, index=['性别', '买衣物首选因素'])
print(df2)

df3 = pd.pivot_table(df, index=['性别', '买衣物首选因素'], value=['平均月生活费（元）'])
print(d3)

df4 = pd.pivot_table(df, index=['性别', '买衣物首选因素'], value=['平均月生活费（元）'], aggfunc=[np.sum])
print(d4)

df5 = pd.pivot_table(df, index=['性别', '买衣物首选因素'], value=['平均月生活费（元）'], aggfunc=[np.mean, len])
print(df5)

df6 = pd.pivot_table(df, index=['性别', '买衣物首选因素'], columns=['家庭所在地区'],value=['平均月生活费（元）'])
print(df6)
```

二、品质数据的整理与展示

品质数据：包括分类数据和顺序数据，一般为非数字型数据。分类数据一般为无序数据，顺序数据一般为有序数据。

- ***分类数据的整理与图示***

分类数据本身就是对事物的一种分类

- **频数与频数分布**

```
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl

mpl.rcParams['font.sans.serif'] = ['FangSong']
mpl.rcParams['axes.unicode_minus'] = False

df = pd.read_excel('data/1_table3.xlsx')
print(df)

data = [(df.loc[:, x].value_counts()) for x in df.columns]
print(data)
```

- 条形图

```
fig = plt.figure()
fig.set(alpha=0.2)  # 设定图表颜色alpha参数
plt.subplot2grid((1, 2), (0, 0))  # 在一张大图里分列几个小图，位置是(0，0)
data1 = df['顾客性别'].value_counts(ascending=True)
data1.plot(kind='bar',
           title='顾客性别')
print(data1)
plt.xlabel("顾客性别")
plt.ylabel("频数")

plt.subplot2grid((1, 2), (0, 1))
data2 = df['饮料类型'].value_counts(ascending=True)
data2.plot(kind='bar',
           title='饮料类型')
plt.xlabel("顾客性别")
plt.ylabel("频数")
print(data2)

plt.show()
```

- 饼图

```

# 控制饼图为正圆
plt.axes(aspect='equal')
# plot方法对序列进行绘图
data2.plot(kind='pie',  # 选择图形类型
           autopct='%.1f%%',  # 饼图中添加数值标签
           radius=1,  # 设置饼图的半径
           startangle=180,  # 设置饼图的初始角度
           counterclock=False,  # 将饼图的顺序设置为顺时针方向
           title='不同类型饮料构成的饼图',  # 为饼图添加标题
```

```
            wedgeprops={'linewidth': 1.5, 'edgecolor': 'green'},  # 设置饼图内外边界的属性值
            textprops={'fontsize': 10, 'color': 'black'})  # 设置文本标签的属性值
# 显示图形
plt.show()
```

三、数值型数据的整理与展示

- 数据分组： 根据统计研究的需要，将原始数据按照某种标准分成不同的组别，分组后的数据称为分组数据，数据分组的主要目的是观察数据的分布特征。数据经分组后再计算出各组中数据出现的频数，就形成了一张频数分布表。数据分组的方

- 直方图

```
# -*- coding:utf-8 -*-
import pandas as pd
import matplotlib.pyplot as plt
import math
from itertools import groupby
import numpy as np
# 设置正常显示中文
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['FangSong']
mpl.rcParams['axes.unicode_minus'] = False

# 按照固定区间长度绘制频率分布直方图
# bins_interval 区间的长度
# margin        设定的左边和右边空留的大小
def probability_distribution(data, bins_interval=1, margin=1):
    bins = range(min(data)-1, max(data) + bins_interval, bins_interval)
    # print(len(bins))
    for i in range(0, len(bins)):
        print(bins[i])
    plt.xlim(min(data) - margin, max(data) + margin)
    plt.title("某电脑公司销售量分布的直方图")
    plt.xlabel('销售量（台）')
    plt.ylabel('频数（天）')
    # 频率分布density=True，频次分布density=False
    prob,left,rectangle = plt.hist(x=data, bins=bins, density=False,
histtype='bar', color=['r'])
    plt.show()
```

茎叶图

茎叶图是反映原始数据分布的图形。它由茎和叶两部分构成，其图形是由数字组成的。通过茎叶图，可以看出数据的分布形式及数据的离散状况，比如，分布是否对称，数据是否集中，是否有离群点等

绘制茎叶图的关键是设计好树茎。制作茎叶图时，首先把一个数字分成两份，通常以该组数据的高位数值作为树茎，而且叶上只保留该数值的最后一个数。例如，125分成12|5， 12 分成 1 | 2， 1.25分成 12 | 5 （单位：0.01）等；前部分是树茎，后部分是树叶。

树茎图类似于横置的直方图，与直方图相比，茎叶图既能给出数据的分布状况，又能给出每一个原始数值，既保留了原始数据的信息。而直方图虽然能很好的显示数据的分布，但是不能保留原始的值。

在应用方面，直方图通常适用于大批量数据，茎叶图通常适用于小批量数据

箱线图

根据一组数据的最大值，最小值，中位数，两个四分位数，这5个特征值绘制而成，

主要用于强调原始数据分布的特征或多组分布特征的比较

绘制方法：先找出一组数据的最大值，最小值，中位数，四分位数，然后，将两个四分位数画出箱子；将最大值和最小值与箱子连接，中位数在箱子中间

时间序列数据-线图

如果数值型数据是在不同时间取得的，即时间序列数据，则可以绘制线图。主要反映现象随时间变化的特征

多变量数据的图示：散点图和气泡图

- 散点图

  两个变量 直接的关系。一个变量放在横轴，一个变量放在纵轴

```
# -*- coding:utf-8 -*-
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
# 设置正常显示中文
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['FangSong']
mpl.rcParams['axes.unicode_minus'] = False

df = pd.read_excel('data/1_table5.xlsx')
#查看数据
print(df)

# 散点图
# 输入产量与温度数据
rainfall = df['降雨量'].values
production = df['产量'].values
colors = np.random.rand(len(rainfall))  # 颜色数组
plt.scatter(rainfall, production, s=200, c=colors)  # 画散点图，大小为 200
plt.xlabel('降雨量')  # 横坐标轴标题
plt.ylabel('产量')  # 纵坐标轴标题
plt.title('小麦产量与降雨量的散点图')
plt.show()
```

- 气泡图

  三个变量直接的关系。一个变量放在横轴，一个变量放在纵轴，另一个变量用气泡的大小表示

```
# 气泡图
tem = df['温度'].values
size = production
plt.scatter(tem, rainfall, s=production, c=colors, alpha=0.6)  # 画散点图,
alpha=0.6 表示不透明度为 0.6
plt.xlabel('温度')  # 横坐标轴标题
plt.ylabel('降雨量')  # 纵坐标轴标题
plt.title('小麦产量与降雨量和温度的气泡图(气泡大小表示产量)')
plt.show()
```