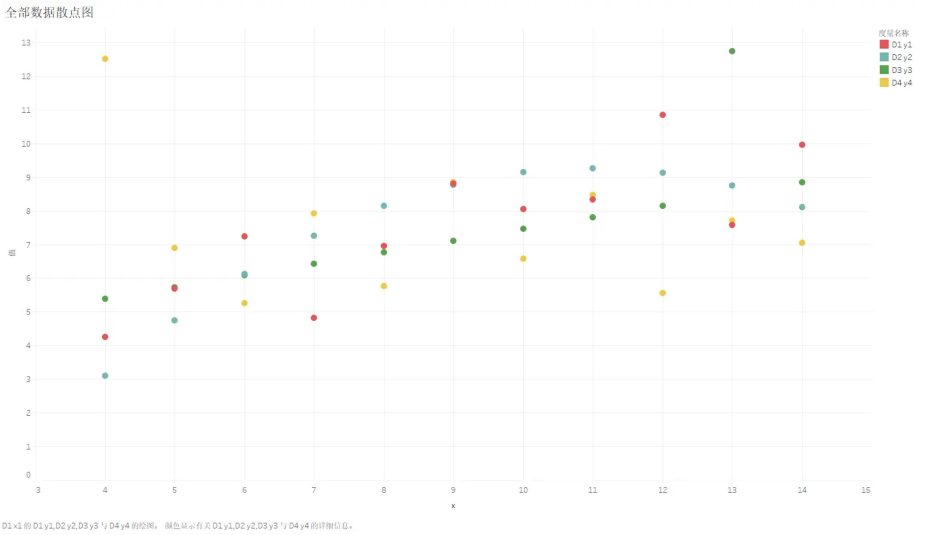
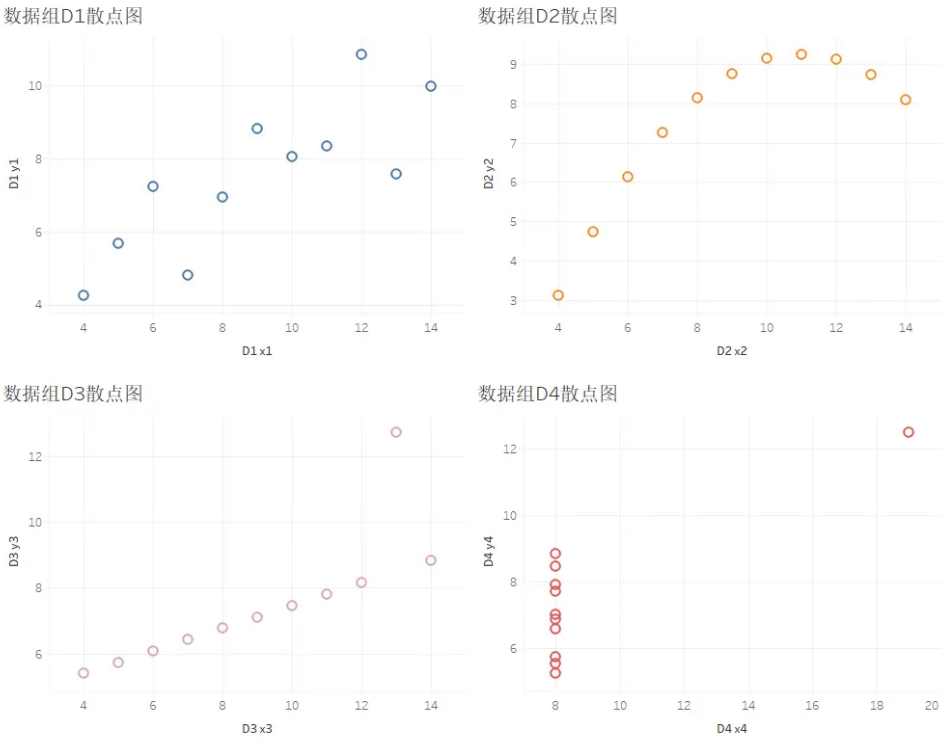
20214064–李恺阳–第1次作业

任务一：可视化数据集，分析说明数据特征

- 对Anscombe's quartet数据集进行可视化
  - 可视化工具：Tableau
  - 可视化结果：
    - 分别对每组数据进行可视化



    - 将四组数据在一个图表中可视化



- 分析说明这四组数据的分布特征
  - 分别计算四组数据的均值、方差、相关系数
    - 计算公式
      - $\mu = \frac{\sum_{x=1}^{n} y_x}{n}$

- 方差： $s^2 = \frac{\sum_{i=1}^{n}(y_i - \mu)^2}{n}$
- 相关系数： $\gamma = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum_{x=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$

- 使用python计算各组数据的相关指标

```python
import numpy as np
import pandas as pd
# 读取Excel文件并去掉前三行
df = pd.read_excel("Anscombe's quartet.xlsx", skiprows=[0, 1])
data = np.array(df)
# 遍历每组数据，计算均值、方差和相关系数
for i in range(4):
    mean_x = data[:,i*2].mean()
    print(f"\nStatistics for Group {i + 1}:\n")
    print(f"Mean_X: {mean_x:.2f}")
    mean_y = data[:,i*2+1].mean()
    print(f"Mean_Y: {mean_y:.2f}")
    var_x = np.var(data[:,i*2], ddof=0)
    print(f"Variance_X: {var_x:.2f}")
    var_y = np.var(data[:,i*2+1])
    print(f"Variance_Y: {var_y:.2f}")
    corr = np.corrcoef(data[:,i*2], data[:,i*2+1])[0, 1]
    print(f"Correlation: {corr:.2f}")

```

- 计算结果
  - 数据组D1
    -
```
Statistics for Group 1:

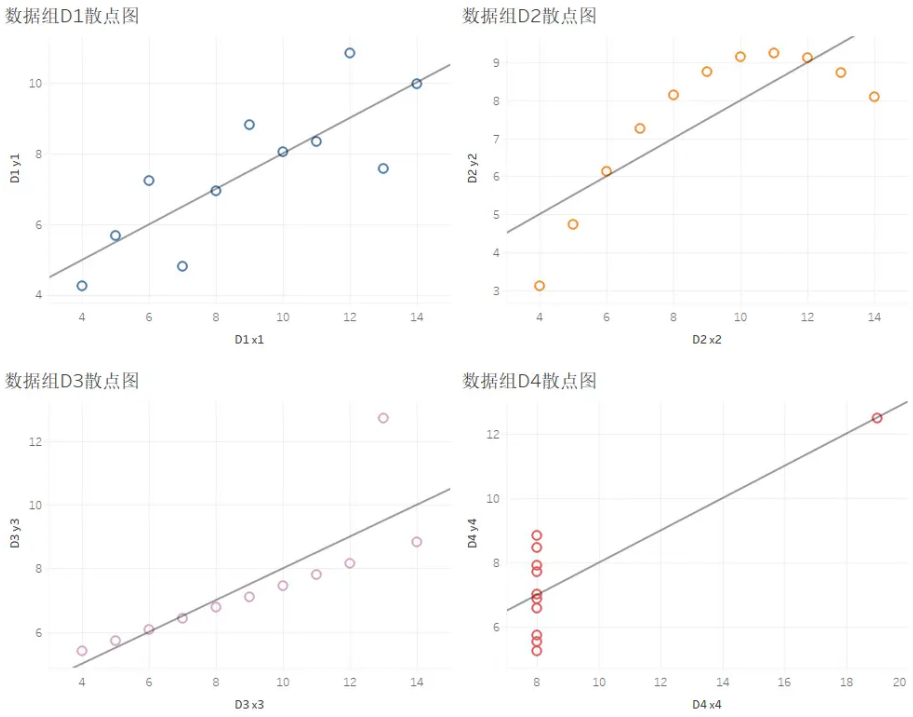
Mean_X: 9.00
Mean_Y: 7.50
Variance_X: 10.00
Variance_Y: 3.75
Correlation: 0.82
```
  - 数据组D2
    -
```
Statistics for Group 2:

Mean_X: 9.00
Mean_Y: 7.50
Variance_X: 10.00
Variance_Y: 3.75
Correlation: 0.82
```
  - 数据组D3
    -
```
Statistics for Group 3:

Mean_X: 9.00
Mean_Y: 7.50
Variance_X: 10.00
Variance_Y: 3.75
Correlation: 0.82
```

- 数据组D4
  - ```
    Statistics for Group 4:

    Mean_X: 9.00
    Mean_Y: 7.50
    Variance_X: 10.00
    Variance_Y: 3.75
    Correlation: 0.82
    ```
- 对四组数据均进行线性拟合



- 由上图可知，拟合结果极为相似

- 结果分析
  - 从各项指标来看，四组数据的均值、方差、相关指数均相等，线性拟合极为相似，仅从数据上来看，四组数据反映出的情况极为相似。但从实际绘图上来看，第一三组都接近线性分布，第三组更加精确，而第二组数据更为接近二次分布，第四组数据与线性分布相差甚远。从上述分析可知，仅看数据的均值、方差、相关指数等指标会导致实验结果并不可靠，而数据可视化的操作让实验结果的可靠性大大提升。因此，数据可视化对于数据分析挖掘方面是极为重要的。

任务二：用脚本语言或编程语言，计算四组数据的最小二乘法回归线方程

- 使用python编写最小二乘法回归线方程
  - 最小二乘法思想：
    - 最小二乘法将最小化误差平方之和来作为目标，从而找到最优模型，这个模型可以拟合观察数据
    - 最小二乘法步骤：
      - 残差： $e_i = y_i - (\omega \times x_i + b)$
      - $Min \sum_{i}^{n} e_i^2$
      - 得到回归线方程 $y = \omega \times x + b$
    ```

- Python 是一种解释型、面向对象、动态类型的编程语言
- 代码：

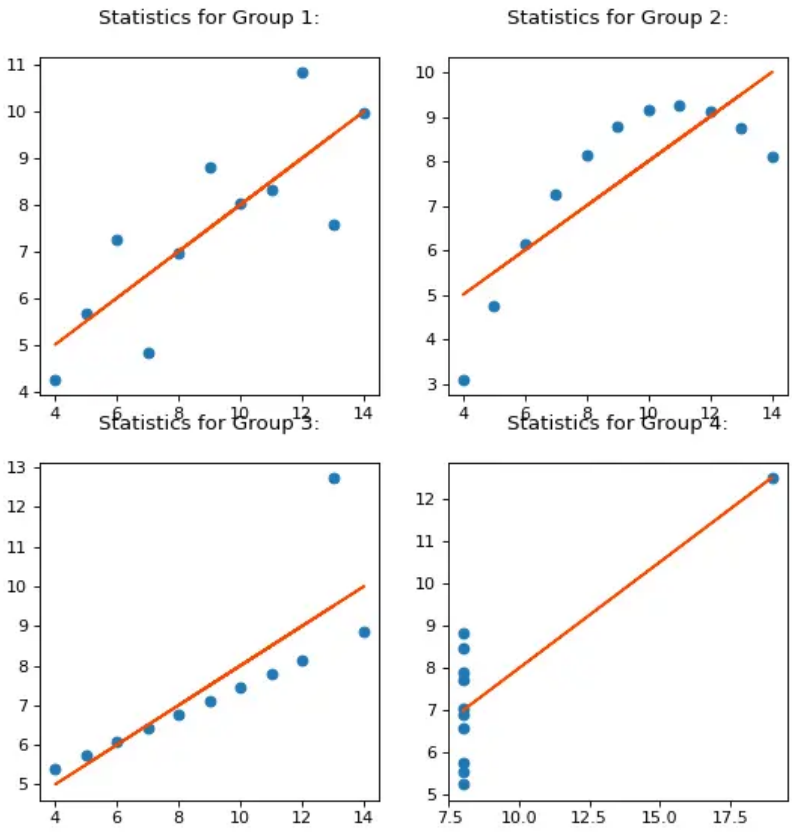
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("Anscombe's quartet.xlsx", skiprows=[0, 1])
data = np.array(df)
plt.figure(figsize=(8,8), dpi=80)
for j in range(4):
    data_x = data[:,j*2]
    data_y = data[:,j*2+1]
    m = len(data_y)
    x_bar = np.mean(data_x)
    sum_yx = 0
    sum_x2 = 0
    sum_delta = 0
    for i in range(m):
        x = data_x[i]
        y = data_y[i]
        sum_yx += y * (x - x_bar)
        sum_x2 += x ** 2
    # 根据公式计算w
    w = sum_yx / (sum_x2 - m * (x_bar ** 2))
    for i in range(m):
        x = data_x[i]
        y = data_y[i]
        sum_delta += (y - w * x)
    b = sum_delta / m
    print(f"Group {j + 1}: y = {w:.2f} * x +{b:.2f}")
    pred_y = w * data_x + b
    ax1 = plt.subplot(2,2,j+1)
    ax1.scatter(data_x, data_y)
    ax1.plot(data_x, pred_y, c='orangered', label='line')
    ax1.set_title(f"\nStatistics for Group {j + 1}:\n")
plt.show()

```

- 运行结果：
  - 四组数据的回归线方程：

```
Group 1: y = 0.50 * x +3.00
Group 2: y = 0.50 * x +3.00
Group 3: y = 0.50 * x +3.00
Group 4: y = 0.50 * x +3.00
```

- 回归线与散点图



- 结果分析：
  - 从上述结果来看，四组数据所得到的回归线方程是一样的，但是四组数据的真实分布趋势确实完全不同的。因此，数据可视化是数据分析过程中保障实验可靠性的重要操作之一。