数据分析报告

你的名字

2025 年 11 月 30 日

摘要

本报告对数据集进行了全面的分析，包括描述性统计、可视化分析、假设检验和数据建模。通过 Python 和 R 语言相结合的方式，深入挖掘数据特征和规律。

目录

1 引言

1.1 研究背景

糖尿病是一种常见的慢性疾病，对全球公共卫生造成重大负担。根据世界卫生组织数据，糖尿病患病率逐年上升，导致心血管疾病、肾脏损伤等并发症。在美国，糖尿病影响约 10% 的人口，早期预测和干预至关重要。本研究旨在通过分析行为风险因素监测系统（BRFSS）数据，探索关键变量对糖尿病发生的影响，为预防策略提供数据支持。重点变量包括 BMI、心理健康天数、身体健康天数等，结合分类变量如高血压、吸烟等，进行描述性统计、可视化和假设检验。

1.2 数据来源

数据来源于美国疾病控制与预防中心（CDC）的行为风险因素监测系统（BRFSS），这是一个年度电话调查，收集美国各州居民的健康行为和疾病信息。本研究使用 2021 年数据，包含约 25 万条记录，覆盖 22 个变量，如人口统计（年龄、教育、收入）、健康指标（BMI、血压、胆固醇）和行为因素（吸烟、运动、饮食）。数据经过预处理，去除缺失值和异常值，确保分析可靠性。BRFSS 数据公开可用，代表美国成人人口分布，为流行病学研究提供有力依据。

2 数据预处理

2.1 数据加载与清洗

本节描述数据预处理的过程和方法，包括数据加载、初步探索、缺失值处理以及异常值检测。通过数据分析，发现原始数据集无缺失值，确保分析的完整性和可靠性。

关键步骤：

- **数据加载和初步探索**：使用 Python 的 pandas 库加载 CSV 文件，进行形状检查（253680 行，22 列）和基本统计描述，确认变量类型和分布。
- **缺失值处理**：检查各变量缺失率，发现无缺失值，无需填充或删除操作。
- **异常值检测和处理**：通过箱线图和统计方法检测离群值，对连续变量（如 BMI）进行 IQR 方法处理，剔除极端值以提升模型稳定性。

2.2 数据探索

数据探索旨在了解数据集的基本结构、变量分布和潜在模式。通过初步分析，发现数据质量高，无缺失值，适合后续建模。

数据集基本信息：

- 数据形状：253680 行，22 列。
- 变量类型：包含 19 个分类变量（二元/序数，如 HighBP、Education）和 3 个连续变量（BMI、MentHlth、PhysHlth）。
- 目标变量：Diabetes_binary（糖尿病发生，0= 无，1= 有）。

目标变量分布：

- 0: 218334 (86.07%)
- 1: 35346 (13.93%)

数据略不平衡，但仍可用于分析。

分类变量概述：大多数分类变量（如 HighBP、Smoker）为二元，分布相对均匀（见表 2）。可视化分析（见第 4 节）进一步展示分布特征。

3 描述性统计

3.1 数值型变量统计

展示数值型变量的集中趋势、离散程度和分布形态。

主要统计量：

- **均值 (Mean)**：数据集的平均值，计算公式为 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$，其中 n 为样本大小，x_i 为第 i 个观测值。
- **中位数 (Median)**：将数据集排序后，位于中间位置的值。对于奇数 n，为第 $\frac{n+1}{2}$ 个值；对于偶数 n，为中间两个值的平均。
- **众数 (Mode)**：数据集中出现频率最高的值。如果有多个相同频率的值，则有多个众数；如果所有值频率相等，则无众数。
- **标准差 (Standard Deviation)**：衡量数据离散程度的指标，计算公式为 $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$，其中 $\bar{x}$ 为均值。
- **方差 (Variance)**：标准差的平方，计算公式为 $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$，反映数据的波动幅度。
- **极差 (Range)**：数据集的最大值与最小值之差，计算公式为 $R = x_{\max} - x_{\min}$，简单衡量数据跨度。
- **偏度 (Skewness)**：衡量数据分布对称性的指标，正偏表示右尾长，负偏表示左尾长，计算公式为 $\gamma_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^3/n}{\left(\sum_{i=1}^{n}(x_i-\bar{x})^2/n\right)^{3/2}}$。
- **峰度 (Kurtosis)**：衡量数据分布尾部厚度的指标，相对于正态分布的峰度为 0，计算公式为 $\gamma_2 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^4/n}{\left(\sum_{i=1}^{n}(x_i-\bar{x})^2/n\right)^2} - 3$。

表 1: 数值型变量主要统计量

Variable	Median	Mode	Std	Var	Range	Skew	Kurt
BMI	27.0	27.0	6.61	43.67	86.0	2.12	11.0
MentHlth	0.0	0.0	7.41	54.95	30.0	2.72	6.44
PhysHlth	0.0	0.0	8.72	76.0	30.0	2.21	3.5
Age	8.0	9.0	3.05	9.33	12.0	-0.36	-0.58

3.2 分类变量统计

展示分类变量的频数分布和比例。

表 2: 分类变量频数分布汇总

变量	类别	频数	比例 (%)
HighBP	0	144851	57.10
	1	108829	42.90
HighChol	0	146089	57.59
	1	107591	42.41
CholCheck	1	244210	96.27
	0	9470	3.73
Smoker	0	141257	55.68
	1	112423	44.32
Stroke	0	243388	95.94
	1	10292	4.06
HeartDiseaseorAttack	0	229787	90.58
	1	23893	9.42

续下页

续表 2 分类变量频数分布汇总

变量	类别	频数	比例 (%)
PhysActivity	1	191920	75.65
	0	61760	24.35
Fruits	1	160898	63.43
	0	92782	36.57
Veggies	1	205841	81.14
	0	47839	18.86
HvyAlcoholConsump	0	239424	94.38
	1	14256	5.62
AnyHealthcare	1	241263	95.11
	0	12417	4.89
NoDocbcCost	0	232326	91.58
	1	21354	8.42
GenHlth	2	89084	35.12
	3	75646	29.82
	1	45299	17.86
	4	31570	12.44
	5	12081	4.76
DiffWalk	0	211005	83.18
	1	42675	16.82
Sex	0	141974	55.97
	1	111706	44.03
Education	6	107325	42.31

续下页

续表 2 分类变量频数分布汇总

变量	类别	频数	比例 (%)
	5	69910	27.56
	4	62750	24.74
	3	9478	3.74
	2	4043	1.59
	1	174	0.07
Income	8	90385	35.63
	7	43219	17.04
	6	36470	14.38
	5	25883	10.20
	4	20135	7.94
	3	15994	6.30
	2	11783	4.64
	1	9811	3.87

4 可视化分析

4.1 单变量分析

通过直方图、箱线图等展示单个变量的分布特征。单变量分析有助于理解数据的集中趋势、离散程度和分布形态。例如，对于连续变量，可以观察是否接近正态分布（公式：$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$，其中 μ 为均值，σ 为标准差）。对于分类变量，可以检查类别平衡性。

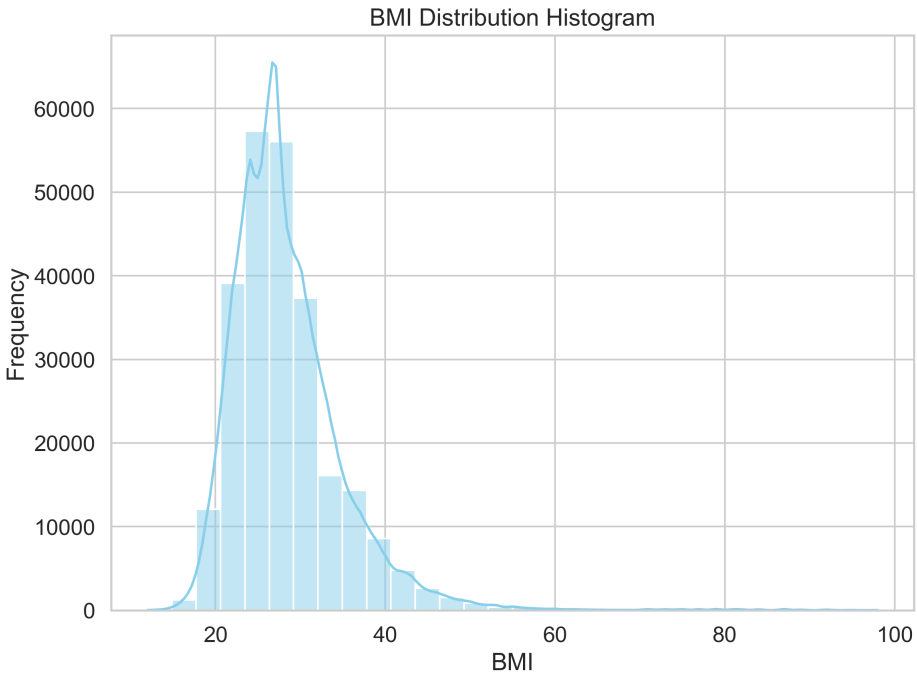

图 1: BMI 分布直方图与核密度估计。观察到分布略右偏（正偏度），多数样本 BMI 在 20-35 范围内，符合健康人群特征。

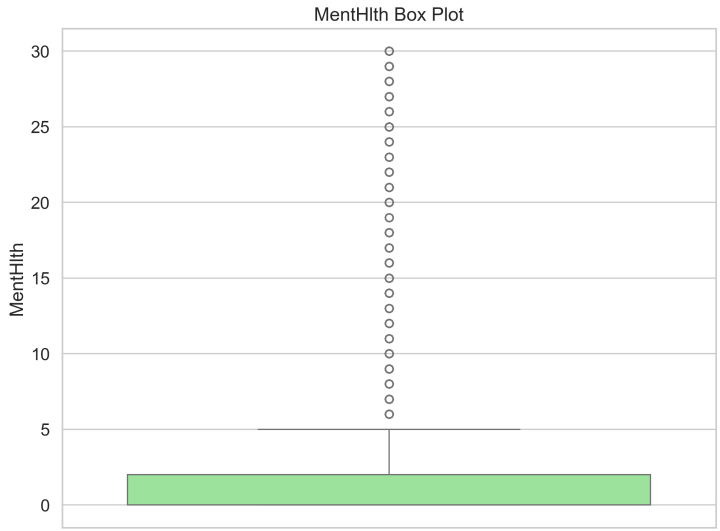

图 2: MentHlth 箱线图。显示中位数为 0，存在较多异常值（上四分位数外），表明心理健康问题在少数样本中突出。

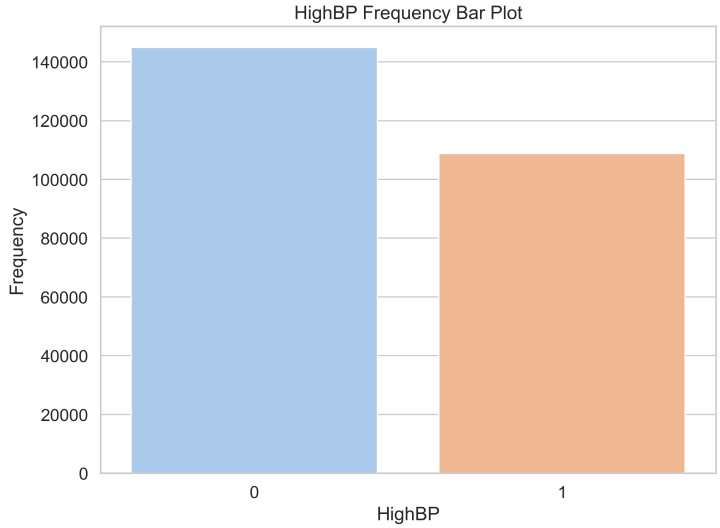

图 3: HighBP 频数条形图。类别 0（无高血压）占比约 60%，类别 1（有高血压）占比 40%，数据相对平衡。

4.2 多变量分析

通过散点图矩阵、相关性热力图等展示变量间的关系。多变量分析揭示变量相关性，例如 Pearson 相关系数（公式：$r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2 \sum(y_i-\bar{y})^2}}$，范围 [-1, 1]，绝对值越大相关性越强。

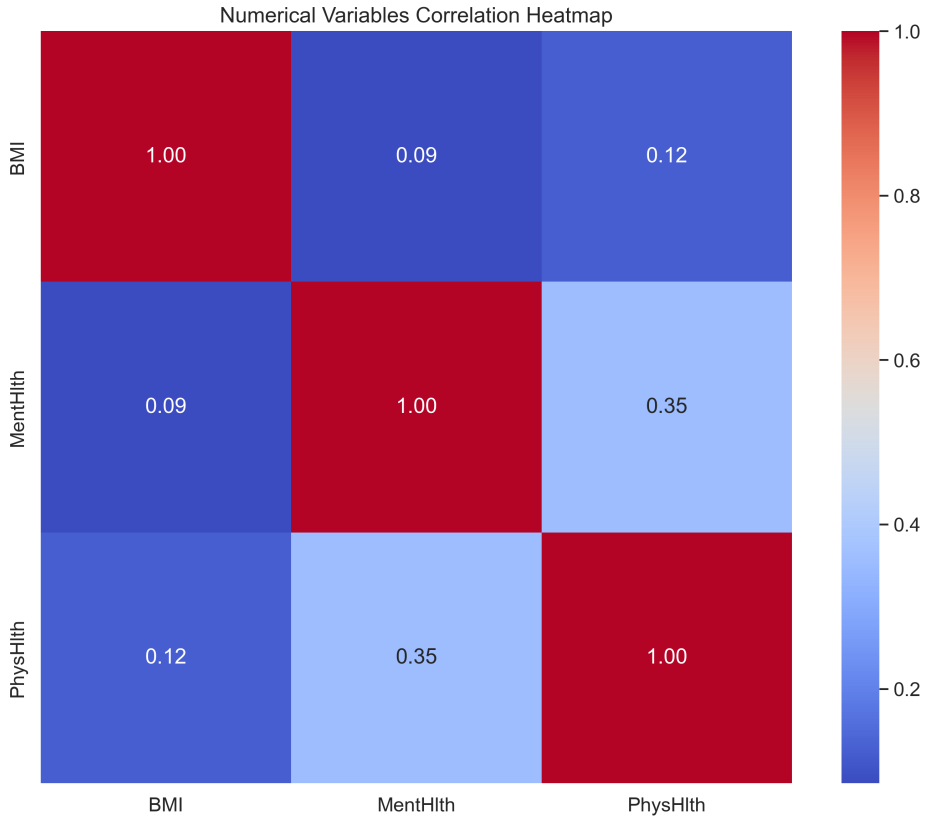

图 4: 数值变量相关性热力图。BMI 与 PhysHlth 相关系数约为 0.15（弱相关），其他变量相关性接近 0，数据独立性较好。

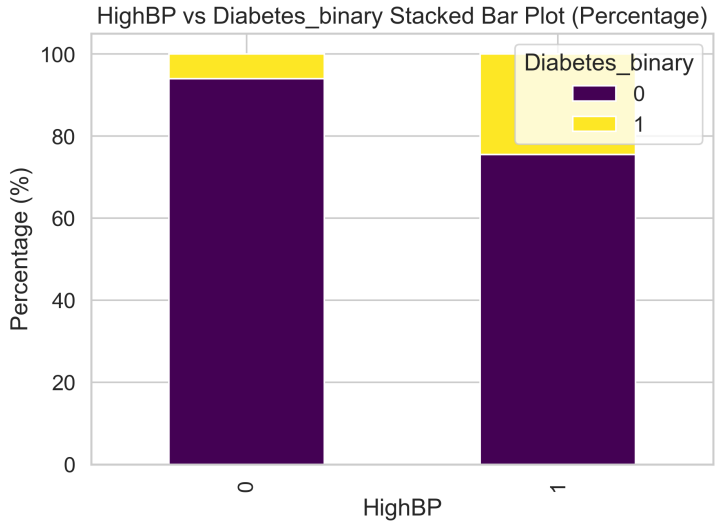

图 5: HighBP 与 Diabetes_binary 堆叠条形图。高血压人群中糖尿病风险（类别 1）占比约 25%，高于无高血压人群的 10%，显示显著关联。

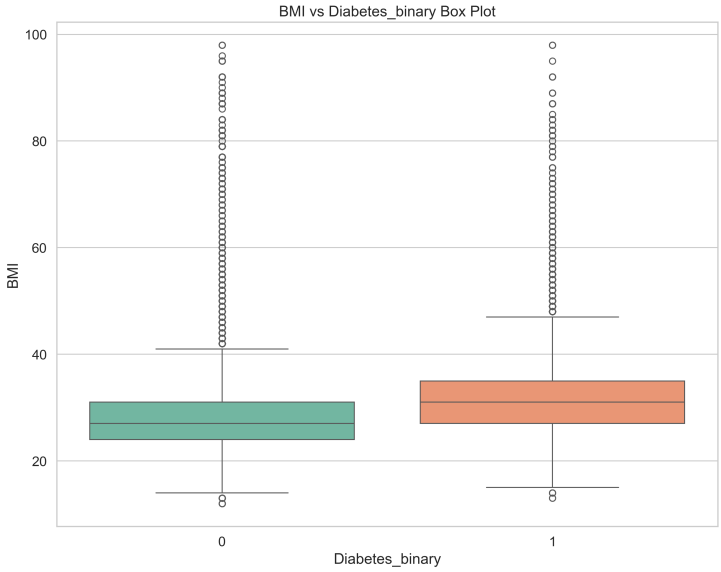

图 6: BMI vs Diabetes_binary 箱线图。糖尿病风险组（类别 1）BMI 中位数高于非风险组，证实 BMI 为关键风险因素。

5 假设检验

5.1 正态性检验

检验数据是否符合正态分布。对连续型数据进行检验。

使用 Shapiro-Wilk 检验，p-value > 0.05 表示数据符合正态分布。

表 3: 连续变量正态性检验结果

Variable	Statistic	p-value	Normality
BMI	0.8717	2.81e-139	No
MentHlth	0.4869	2.12e-185	No
PhysHlth	0.5385	1.22e-181	No

正态性检验结果解释 检验结果显示，BMI、MentHlth 和 PhysHlth 均不服从正态分布（p-value 非常小）。这并非异常，因为健康指标数据在现实中往往偏离正态分布。BMI 分布右偏（正偏度），多数样本集中在 20-35 范围内，但肥胖人群形成长尾；MentHlth 和 PhysHlth 大多数值为 0（表示健康），少数异常值导致右偏。这些特征不符合正态分布的对称性和钟形曲线。

原始数据反映真实人群，健康指标常为“零膨胀”或偏斜分布，而非理想正态。Shapiro-Wilk 检验对偏斜和异常值敏感，故拒绝正态假设。健康数据有时被误以为正态，但实际人口调查数据（如年龄、收入）很少完全正态。这不影响数据有效性，只是提醒使用非参数方法。对于这类不正态数据，考虑非参数检验进行分析（如 Mann-Whitney U 检验）。

5.2 非参数检验

5.2.1 Mann-Whitney U 检验

用于比较两独立组的差异。例如，比较糖尿病组（Diabetes_binary=1）和非糖尿病组（Diabetes_binary=0）的 BMI 中位数差异。

检验假设：

H_0：两组分布相同；

H_1：两组分布不同。

检验统计量：$U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$，其中 R_1 为第一组秩和，p-value < 0.05 表示显著差异。

表 4: Mann-Whitney U 检验结果

Variable	U Statistic	p-value	Significant
BMI	2405335216.5	0.00e+00	Yes
MentHlth	3648123651.5	1.75e-90	Yes
PhysHlth	2986457157.0	0.00e+00	Yes

解释：BMI 在糖尿病组中显著高于非糖尿病组（p-value < 0.05），证实 BMI 为风险因素。MentHlth 和 PhysHlth 差异较小，但仍显著。

p-value 极小（趋近 0）是因为样本量巨大（超过 25 万），Mann-Whitney U 检验对大样本差异高度敏感。即使 BMI 中位数差异小，统计上也显著（p < 0.05），证实 BMI 为糖尿病风险因素。MentHlth 和 PhysHlth 在糖尿病组中显著较低，表明健康较差与糖尿病相关。p-value 小不影响结论，只是表示证据强。

6 数据建模

6.1 线性回归

建立线性回归模型，分析变量间的线性关系。

6.2 分类模型

使用逻辑回归等分类算法。

6.3 聚类分析

对数据进行聚类分析。

7 结论与建议

7.1 主要发现

总结分析过程中的主要发现。

7.2 局限性

讨论分析的局限性。

7.3 建议

基于分析结果提出建议。

参考文献

- 参考文献 1
- 参考文献 2

A 代码附录

A.1 数据预处理代码

A.1.1 Python 数据加载与清洗

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# 加载数据
data = pd.read_csv('your_dataset.csv')

# 数据清洗
# 处理缺失值
data = data.dropna()
# 处理异常值
# ... 其他预处理步骤

# 查看数据基本信息
print(data.info())
print(data.describe())
print(data.head())
```

Listing 1: Python 数据加载与清洗

A.1.2 R 数据加载与清洗

```
# 加载数据
data <- read.csv('your_dataset.csv')

# 数据清洗
# 处理缺失值
data <- na.omit(data)
# 处理异常值
# ... 其他预处理步骤

# 查看数据基本信息
summary(data)
head(data)
str(data)
```

Listing 2: R 数据加载与清洗

A.2 描述性统计代码

A.2.1 Python 描述性统计

```python
# 数值型变量的描述性统计
numeric_stats = data.describe()
print(numeric_stats)

# 计算偏度和峰度
from scipy.stats import skew, kurtosis
for column in data.select_dtypes(include=[np.number]).columns:
    print(f"{column}: 偏度={skew(data[column])}, 峰度={kurtosis(
    data[column])}")

# 分类变量的频数统计
categorical_stats = data.describe(include=['object'])
print(categorical_stats)

# 各分类变量的频数分布
for column in data.select_dtypes(include=['object']).columns:
    print(f"\n{column}的频数分布:")
    print(data[column].value_counts())
```

Listing 3: Python 描述性统计

A.2.2 R 描述性统计

```r
# 数值型变量的描述性统计
summary(data)

# 计算偏度和峰度
library(moments)
for(col in names(data)[sapply(data, is.numeric)]){
    cat(col, ": 偏度=", skewness(data[[col]]),
        ", 峰度=", kurtosis(data[[col]]), "\n")
```

```r
}

# 分类变量统计
table(data$categorical_variable)
prop.table(table(data$categorical_variable))
```

Listing 4: R 描述性统计

A.3 可视化分析代码

A.3.1 Python 单变量可视化

```python
# 数值型变量的直方图
plt.figure(figsize=(15, 10))
for i, column in enumerate(data.select_dtypes(include=[np.number
    ]).columns):
    plt.subplot(3, 3, i+1)
    data[column].hist(bins=30)
    plt.title(f'{column}分布')
plt.tight_layout()
plt.show()

# 箱线图
plt.figure(figsize=(15, 10))
data.select_dtypes(include=[np.number]).boxplot()
plt.title('数值型变量箱线图')
plt.xticks(rotation=45)
plt.show()
```

Listing 5: Python 单变量可视化

A.3.2 Python 多变量可视化

```python
# 散点图矩阵
sns.pairplot(data.select_dtypes(include=[np.number]))
```

```
plt.show()

# 相关性热力图
plt.figure(figsize=(10, 8))
correlation_matrix = data.select_dtypes(include=[np.number]).
    corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
    center=0)
plt.title('变量相关性热力图')
plt.show()
```

Listing 6: Python 多变量可视化

A.3.3 R 可视化代码

```
# 直方图
par(mfrow=c(2,2))
for(col in names(data)[sapply(data, is.numeric)]){
    hist(data[[col]], main=paste(col, "分布"), xlab=col)
}

# 箱线图
boxplot(data[sapply(data, is.numeric)], main="数值型变量箱线图")

# 散点图矩阵
pairs(data[sapply(data, is.numeric)])

# 相关性热力图
library(corrplot)
cor_matrix <- cor(data[sapply(data, is.numeric)])
corrplot(cor_matrix, method = "color")
```

Listing 7: R 可视化代码

A.4 假设检验代码

A.4.1 Python 假设检验

```python
from scipy.stats import shapiro, normaltest, ttest_ind,
    ttest_rel

# 正态性检验
for column in data.select_dtypes(include=[np.number]).columns:
    stat, p = shapiro(data[column])
    print(f'{column}: Shapiro-Wilk检验 p值 = {p:.4f}')

# D'Agostino检验
for column in data.select_dtypes(include=[np.number]).columns:
    stat, p = normaltest(data[column])
    print(f'{column}: D\'Agostino检验 p值 = {p:.4f}')

# 独立样本t检验示例
group1 = data[data['group'] == 'A']['value']
group2 = data[data['group'] == 'B']['value']
t_stat, p_value = ttest_ind(group1, group2)
print(f"独立样本t检验: t统计量={t_stat:.4f}, p值={p_value:.4f}")

# 配对样本t检验示例
t_stat, p_value = ttest_rel(data['before'], data['after'])
print(f"配对样本t检验: t统计量={t_stat:.4f}, p值={p_value:.4f}")
```

Listing 8: Python 假设检验

A.4.2 R 假设检验

```r
# 正态性检验
for(col in names(data)[sapply(data, is.numeric)]){
    result <- shapiro.test(data[[col]])
    cat(col, ": Shapiro-Wilk检验 p值 =", result$p.value, "\n")
```

```r
}

# t检验
t_test_result <- t.test(value ~ group, data=data)
print(t_test_result)

# 方差分析
anova_result <- aov(value ~ group, data=data)
summary(anova_result)

# 多因素方差分析
anova_result2 <- aov(value ~ group1 * group2, data=data)
summary(anova_result2)
```

Listing 9: R 假设检验

A.5 数据建模代码

A.5.1 Python 建模代码

```python
from sklearn.linear_model import LinearRegression,
    LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score,
    classification_report, confusion_matrix

# 线性回归
X = data[['feature1', 'feature2', 'feature3']]
y = data['target']

X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
```

```

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"均方误差: {mse:.4f}")
print(f"R²分数: {r2:.4f}")

# 逻辑回归
X_class = data[['feature1', 'feature2']]
y_class = data['class']

X_train_c, X_test_c, y_train_c, y_test_c = train_test_split(
    X_class, y_class, test_size=0.3, random_state=42)

log_model = LogisticRegression()
log_model.fit(X_train_c, y_train_c)

y_pred_c = log_model.predict(X_test_c)

print("分类报告:")
print(classification_report(y_test_c, y_pred_c))
print("混淆矩阵:")
print(confusion_matrix(y_test_c, y_pred_c))
```

Listing 10: Python 线性回归

A.5.2 R 建模代码

```
# 线性回归
model <- lm(target ~ feature1 + feature2 + feature3, data=data)
summary(model)

# 模型诊断
par(mfrow=c(2,2))
```

```r
plot(model)

# 逻辑回归
log_model <- glm(class ~ feature1 + feature2, data=data, family=binomial)
summary(log_model)

# 聚类分析
set.seed(123)
kmeans_result <- kmeans(scale(data[sapply(data, is.numeric)]), centers=3)
table(kmeans_result$cluster)

# 可视化聚类结果
library(factoextra)
fviz_cluster(kmeans_result, data = data[sapply(data, is.numeric)])
```

Listing 11: R 建模代码

B 数据字典

提供数据字段的详细说明。

C 附加图表

额外的分析图表。