MALMÖ UNIVERSITY

MA660E, Lab Report

Sirajulhaq Wahaj

Part Two: Statistics and inference

Heart Disease Dataset

This dataset contains information about patients and various attributes related to heart disease, collected from Cleveland Clinic and made available on Kaggle. It includes both qualitative and quantitative variables, which are ideal for performing analyses such as descriptive statistics, confidence intervals, hypothesis testing, correlation analysis, and multiple linear regression.

Source: Kaggle - Heart Disease Data

Variables

Quantitative Variables

- **id**: Unique identifier for each patient
- **age**: Age of the patient in years
- **trestbps**: Resting blood pressure in mm Hg
- **chol**: Serum cholesterol level in mg/dl
- **thalch**: Maximum heart rate achieved
- **oldpeak**: ST depression induced by exercise relative to rest
- **ca**: Number of major vessels (0-3) colored by fluoroscopy
- **num**: Diagnosis of heart disease (angiographic disease status), where `0` indicates no disease and `1-4` indicates presence of disease

Qualitative Variables

- **sex**: Sex of the patient, either `Male` or `Female`
- **dataset**: Source of the data, e.g., Cleveland

- **cp**: Chest pain type, with categories `typical angina`, `asymptomatic`, `non-anginal`, or `atypical angina`
- **fbs**: Fasting blood sugar > 120 mg/dl, represented as `TRUE` if true and `FALSE` otherwise
- **restecg**: Resting electrocardiographic results, either `normal` or `lv hypertrophy` (left ventricular hypertrophy)
- **exang**: Exercise-induced angina, with `TRUE` if present and `FALSE` otherwise
- **slope**: Slope of the peak exercise ST segment, categorized as `upsloping`, `flat`, or `downsloping`
- **thal**: Type of thalassemia, with values `normal`, `fixed defect`, or `reversable defect`

Results Part.

Question 1. Descriptive Statistics

Perform descriptive statistics analysis for at least two qualitative and two quantitative variables.

Solution

Quantitative Descriptive Statistics:

- The average age of participants is about 53.5 years, with ages ranging from 28 to 77 years.
- For blood pressure, the average is 132, but it can range from 0 to 200.
- Cholesterol levels have an average of 200, with values ranging from 0 to 603.
- The thalium stress test results average around 138, with values between 60 and 202.
- The oldpeak (which measures depression induced by exercise) averages at 0.85, but it can range from -2.6 to 6.2.

Qualitative Descriptive Statistics:

- Sex: The majority of participants are male (726), with fewer females (194).
- Chest pain type: Most participants are classified as having asymptomatic chest pain (496), followed by non-anginal pain (204), atypical angina (174), and typical angina (46).
- Dataset origin: Most cases come from Cleveland (304), followed by Hungary (293), VA Long Beach (200), and Switzerland (123).

Question 2. Confidence Intervals

Calculate the confidence interval for one quantitative variable and the confidence interval for the difference between two groups.

Solution:

Confidence Interval for Mean Age (95.0%): 52.901 to 54.121

Confidence Interval for Difference in Cholesterol Levels (Male - Female) (95.0%): -66.383 to -37.290

Question 3. T-test or ANOVA

Conduct a T-test to check if there is a significant difference between two groups, or Perform an ANOVA to see if all groups have the same mean for a characteristic.

Solution:

ANOVA Results for Cholesterol Levels across Chest Pain Types:

F-statistic: 7.5912

P-value: 0.0001

Result: Significant differences in cholesterol levels across chest pain types ($p < 0.05$).

Question 4. Non-Parametric Test

Conduct a non-parametric test for the same variable as in Exercise 3 and compare the conclusions with ANOVA results.

Solution:

Kruskal-Wallis Test:

Statistic: 12.772943982536457, p-value: 0.005154264553910447

Conclusion: There is a statistically significant difference in cholesterol levels among the chest pain types.

---

5. Correlation Analysis

Identify the strongest correlations and any statistically insignificant relationships.

Solution:

```
Correlation Matrix:
                age  trestbps      chol    thalch   oldpeak
age       1.000000  0.230784 -0.086010 -0.349715  0.233550
trestbps  0.230784  1.000000  0.089484 -0.104747  0.161217
chol     -0.086010  0.089484  1.000000  0.226047  0.047454
thalch   -0.349715 -0.104747  0.226047  1.000000 -0.149401
oldpeak   0.233550  0.161217  0.047454 -0.149401  1.000000

Strongest Correlations (|correlation| > 0.5):

Statistically Insignificant Relationships (p > 0.05):
chol and oldpeak: p-value = 0.1504
oldpeak and chol: p-value = 0.1504
```

Question: 6. Multiple Linear Regression

Perform a multiple linear regression analysis.

In this multiple regression analysis, the goal is to understand how the dependent variable, num, is influenced by several independent variables: age, trestbps, thalch, oldpeak, sex, cp, fbs, restecg, exang, and chol. By examining these predictors together, we aim to see how well they explain changes in num and identify which factors have the most significant impact.

Solution:

The multiple regression analysis explains 40.4% of the variability in the dependent variable, num, as indicated by the R-squared value. Significant predictors include age, oldpeak, sex, cp, restecg, exang, and chol, which have a strong relationship with num. Variables like trestbps and fbs showed less impact on the outcome. The model was trained and tested using a 70-30 split, and evaluation metrics such as R-squared and Mean Squared Error were used to assess its performance. While the model performed reasonably well, the high condition number suggests potential multicollinearity among some predictors, warranting further investigation to ensure reliable results.

```
R-squared: 0.382249792102602
Mean Squared Error: 0.8061845802675568
Coefficients: [ 1.47960179e-02 -2.11397611e-04 -5.45987756e-03  3.20096038e-01
  3.90277773e-01  2.32907223e-01  2.77445628e-02  1.30664271e-01
  1.55654860e-01 -1.84402977e-03]
Intercept: 0.08562560509436445
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    num   R-squared:                       0.404
Model:                            OLS   Adj. R-squared:                  0.396
Method:                 Least Squares   F-statistic:                     49.57
Date:                Mon, 09 Dec 2024   Prob (F-statistic):           1.47e-75
Time:                        15:36:28   Log-Likelihood:                -937.76
No. Observations:                 741   AIC:                             1898.
Df Residuals:                     730   BIC:                             1948.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0234      0.415     -0.056      0.955      -0.837       0.791
age            0.0137      0.004      3.528      0.000       0.006       0.021
trestbps       0.0012      0.002      0.633      0.527      -0.002       0.005
thalch        -0.0052      0.001     -3.461      0.001      -0.008      -0.002
oldpeak        0.3184      0.034      9.246      0.000       0.251       0.386
sex            0.3327      0.079      4.209      0.000       0.178       0.488
cp             0.2272      0.038      5.913      0.000       0.152       0.303
fbs            0.0893      0.099      0.904      0.366      -0.105       0.283
restecg        0.1321      0.039      3.417      0.001       0.056       0.208
exang          0.1960      0.080      2.437      0.015       0.038       0.354
chol          -0.0019      0.000     -5.794      0.000      -0.002      -0.001
==============================================================================
Omnibus:                       81.790   Durbin-Watson:                   1.917
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              109.623
Skew:                           0.846   Prob(JB):                     1.57e-24
Kurtosis:                       3.831   Cond. No.                     3.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.91e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Code Part

In [164…

```python
import pandas as pd
from scipy.stats import kruskal
from scipy.stats import pearsonr
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np
from scipy import stats
```

Data Cleaning

In [166…

```python
# silient downcasting and warnnings
pd.set_option('future.no_silent_downcasting', True)

data_set = pd.read_csv('heart_disease_uci.csv')
null_values = data_set.isnull().sum()

# Columns with null values
quantitative_columns = ['trestbps', 'chol', 'thalch', 'oldpeak', 'ca', 'age', 'r
qualitative_columns = ['sex', 'cp', 'restecg', 'fbs', 'exang', 'slope', 'thal']
```

```python
# 1. Quantitative Columns: Fill missing values with the median
data_cleaned = data_set.copy()
for col in quantitative_columns:
    if data_set[col].isnull().sum() > 0:
        median_value = data_set[col].median()
        data_cleaned[col] = data_set[col].fillna(median_value)

# 2. Qualitative Columns: Fill missing values with the mode
for col in qualitative_columns:
    if data_set[col].isnull().sum() > 0:
        mode_value = data_set[col].mode()[0]
        data_cleaned[col] = data_set[col].fillna(mode_value).infer_objects()
# Convert sex column to numeric
data_cleaned['sex'] = data_cleaned['sex'].map({'Female': 0, 'Male': 1})
data_cleaned['cp'] = data_cleaned['cp'].map({'typical angina': 0, 'atypical angi
data_cleaned['fbs'] = data_cleaned['fbs'].map({False: 0, True: 1})
data_cleaned['exang'] = data_cleaned['exang'].map({False: 0, True: 1})
data_cleaned['restecg'] = data_cleaned['restecg'].map({'normal': 0, 'abnormal':
```

In [156…

```python
null_values = data_cleaned.isnull().sum()

columns_with_null = null_values[null_values > 0]
if len(columns_with_null) > 0:
    print("Columns with null values:")
else:
    print("No columns with null values.")

for column, null_count in columns_with_null.items():
    print(f"{column}: {null_count} null values")
```

```
Columns with null values:
restecg: 179 null values
```

Part Two: 1. Descriptive Statistics

Perform descriptive statistics analysis for at least two qualitative and two quantitative variables.

In [157…

```python
# Columns to analyze
quantitative_columns = ['age', 'trestbps', 'chol', 'thalch', 'oldpeak']
qualitative_columns = ['sex', 'cp', 'dataset']

# Generate descriptive statistics for quantitative variables
quantitative_stats = data_cleaned[quantitative_columns].describe()

# Generate frequency counts for qualitative variables
qualitative_stats = {col: data_cleaned[col].value_counts() for col in qualitativ

# Output results to console in a structured format

# Quantitative Descriptive Statistics
print("\n==================== Quantitative Descriptive Statistics ==============
print(f"\nDescriptive statistics for quantitative variables ({', '.join(quantita
print("\nThis includes measures such as the mean, standard deviation, min, 25th
print(quantitative_stats)

# Qualitative Descriptive Statistics
```

```
print("\n==================== Qualitative Descriptive Statistics ===============
print(f"\nFrequency counts for qualitative variables ({', '.join(qualitative_col
for col, stats in qualitative_stats.items():
    print(f"\nFor the variable '{col}', the distribution is as follows:")
    print(stats)
```

```
==================== Quantitative Descriptive Statistics ====================

Descriptive statistics for quantitative variables (age, trestbps, chol, thalch, o
ldpeak):

This includes measures such as the mean, standard deviation, min, 25th percentile
(Q1), median (50th percentile), 75th percentile (Q3), and max for each of these c
olumns.
              age    trestbps        chol      thalch     oldpeak
count  920.000000  920.000000  920.000000  920.000000  920.000000
mean    53.510870  131.995652  199.908696  137.692391    0.853261
std      9.424685   18.451300  109.040171   25.145235    1.058049
min     28.000000    0.000000    0.000000   60.000000   -2.600000
25%     47.000000  120.000000  177.750000  120.000000    0.000000
50%     54.000000  130.000000  223.000000  140.000000    0.500000
75%     60.000000  140.000000  267.000000  156.000000    1.500000
max     77.000000  200.000000  603.000000  202.000000    6.200000

==================== Qualitative Descriptive Statistics ====================

Frequency counts for qualitative variables (sex, cp, dataset):

For the variable 'sex', the distribution is as follows:
sex
1    726
0    194
Name: count, dtype: int64

For the variable 'cp', the distribution is as follows:
cp
3    496
2    204
1    174
0     46
Name: count, dtype: int64

For the variable 'dataset', the distribution is as follows:
dataset
Cleveland        304
Hungary          293
VA Long Beach    200
Switzerland      123
Name: count, dtype: int64
```

Part Two: 2. Confidence Intervals

Calculate the confidence interval for one quantitative variable and the confidence interval for the difference between two groups.

In [158…

```
import numpy as np
from scipy import stats
```

```
# Set confidence level
confidence_level = 0.95

# Confidence Interval for Mean Age
def calculate_age_confidence_interval(data):
    age_mean = data['age'].mean()
    age_std = data['age'].std()
    age_n = data['age'].count()
    age_se = age_std / np.sqrt(age_n)
    return stats.t.interval(confidence_level, df=age_n-1, loc=age_mean, scale=ag

# Confidence Interval for Difference in Cholesterol Levels
def calculate_cholesterol_difference_confidence_interval(data):
    chol_male = data[data['sex'] == 'Male']['chol']
    chol_female = data[data['sex'] == 'Female']['chol']

    chol_male_mean = chol_male.mean()
    chol_female_mean = chol_female.mean()
    chol_male_std = chol_male.std()
    chol_female_std = chol_female.std()

    n_male = chol_male.count()
    n_female = chol_female.count()

    se_diff = np.sqrt((chol_male_std**2 / n_male) + (chol_female_std**2 / n_fema
    mean_diff = chol_male_mean - chol_female_mean
    df_diff = min(n_male, n_female) - 1

    return stats.t.interval(confidence_level, df=df_diff, loc=mean_diff, scale=s

# Calculate and print results
age_ci = calculate_age_confidence_interval(data_cleaned)
print(f"Confidence Interval for Mean Age ({confidence_level*100}%): {age_ci[0]:.

chol_ci = calculate_cholesterol_difference_confidence_interval(data_cleaned)
print(f"Confidence Interval for Difference in Cholesterol Levels (Male - Female)
```

```
Confidence Interval for Mean Age (95.0%): 52.901 to 54.121
Confidence Interval for Difference in Cholesterol Levels (Male - Female) (95.0%):
nan to nan
```

Part Two: 3. T-test or ANOVA

Conduct a T-test to check if there is a significant difference between two groups, or Perform an ANOVA to see if all groups have the same mean for a characteristic.

In [159…

```
# Separate cholesterol levels by chest pain type (cp)
cp_groups = []
for cp in data_cleaned['cp'].unique():
    # Filter cholesterol values for each unique chest pain type without dropping
    chol_values = data_cleaned[data_cleaned['cp'] == cp]['chol']
    cp_groups.append(chol_values)

#cp_groups = [data_cleaned[data_cleaned['cp'] == cp]['chol'] for cp in data_clea

# Perform one-way ANOVA
f_stat, p_value = stats.f_oneway(*cp_groups)
```

```
# Output the result
print("ANOVA Results for Cholesterol Levels across Chest Pain Types:")
print(f"F-statistic: {f_stat:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpretation
if p_value < 0.05:
    print("Result: Significant differences in cholesterol levels across chest pa
else:
    print("Result: No significant differences in cholesterol levels across chest
```

```
ANOVA Results for Cholesterol Levels across Chest Pain Types:
F-statistic: 7.5912
P-value: 0.0001
Result: Significant differences in cholesterol levels across chest pain types (p
< 0.05).
```

Part Two: 4. Non-Parametric Test

Conduct a non-parametric test for the same variable as in Exercise 3 and compare the conclusions with ANOVA results.

In [160…

```
# Conduct the Kruskal-Wallis test
kruskal_stat, kruskal_p_value = kruskal(*cp_groups)

# Output the result
print("Kruskal-Wallis Test:")
print(f"Statistic: {kruskal_stat}, p-value: {kruskal_p_value}")

# Interpretation based on p-value
if kruskal_p_value < 0.05:
    print("Conclusion: There is a statistically significant difference in choles
else:
    print("Conclusion: No statistically significant difference in cholesterol le
```

```
Kruskal-Wallis Test:
Statistic: 12.772943982536457, p-value: 0.005154264553910447
Conclusion: There is a statistically significant difference in cholesterol levels
among the chest pain types.
```

Part Two: 5. Correlation Analysis

Identify the strongest correlations and any statistically insignificant relationships.

In [161…

```
quantitative_columns = ['age', 'trestbps', 'chol', 'thalch', 'oldpeak']
correlation_matrix = data_cleaned[quantitative_columns].corr()

print("Correlation Matrix:")
print(correlation_matrix)

#find the strongest correlations like |correlation| > 0.5

strong_correlations = []
for col1 in quantitative_columns:
    for col2 in quantitative_columns:
        if col1 != col2:
```

```
                correlation = correlation_matrix.loc[col1, col2]
                if abs(correlation) > 0.5:
                    strong_correlations.append((col1, col2, correlation))

print("\nStrongest Correlations (|correlation| > 0.5):")
for col1, col2, corr in strong_correlations:
    print(f"{col1} and {col2}: correlation = {corr:.2f}")


#Check statistically insignificant relationships (p > 0.05)
insignificant_correlations = []
for col1 in quantitative_columns:
    for col2 in quantitative_columns:
        if col1 != col2:
            corr, p_value = pearsonr(data_cleaned[col1].dropna(), data_cleaned[c
            if p_value > 0.05:
                insignificant_correlations.append((col1, col2, p_value))

print("\nStatistically Insignificant Relationships (p > 0.05):")
for col1, col2, p_value in insignificant_correlations:
    print(f"{col1} and {col2}: p-value = {p_value:.4f}")
```

```
Correlation Matrix:
               age  trestbps      chol    thalch   oldpeak
age       1.000000  0.230784 -0.086010 -0.349715  0.233550
trestbps  0.230784  1.000000  0.089484 -0.104747  0.161217
chol     -0.086010  0.089484  1.000000  0.226047  0.047454
thalch   -0.349715 -0.104747  0.226047  1.000000 -0.149401
oldpeak   0.233550  0.161217  0.047454 -0.149401  1.000000

Strongest Correlations (|correlation| > 0.5):
Statistically Insignificant Relationships (p > 0.05):
chol and oldpeak: p-value = 0.1504
oldpeak and chol: p-value = 0.1504
```

Part Two: 6. Multiple Linear Regression

Perform a multiple linear regression analysis.

In this multiple regression analysis, the goal is to understand how the dependent variable, num, is influenced by several independent variables: age, trestbps, thalch, oldpeak, sex, cp, fbs, restecg, exang, and chol. By examining these predictors together, we aim to see how well they explain changes in num and identify which factors have the most significant impact.

In [162…

```
# Define the target and predictor variables
#['age', 'sex', 'cp', 'trestbps', 'fbs', 'restecg', 'thalch', 'exang']
X = data_cleaned[['age', 'trestbps', 'thalch', 'oldpeak', 'sex', 'cp', 'fbs', 'r
y = data_cleaned['num']  # Target variable

# Drop any rows with missing values in X or y
X = X.dropna()
y = y.loc[X.index]  # Keep y aligned with the non-null X

# Add a constant to X to account for the intercept
X = sm.add_constant(X)
```

```
# Fit the model
model = sm.OLS(y, X).fit()

# Output the summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    num   R-squared:                       0.404
Model:                            OLS   Adj. R-squared:                  0.396
Method:                 Least Squares   F-statistic:                     49.57
Date:                Mon, 09 Dec 2024   Prob (F-statistic):           1.47e-75
Time:                        18:45:43   Log-Likelihood:                -937.76
No. Observations:                 741   AIC:                             1898.
Df Residuals:                     730   BIC:                             1948.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0234      0.415     -0.056      0.955      -0.837       0.791
age            0.0137      0.004      3.528      0.000       0.006       0.021
trestbps       0.0012      0.002      0.633      0.527      -0.002       0.005
thalch        -0.0052      0.001     -3.461      0.001      -0.008      -0.002
oldpeak        0.3184      0.034      9.246      0.000       0.251       0.386
sex            0.3327      0.079      4.209      0.000       0.178       0.488
cp             0.2272      0.038      5.913      0.000       0.152       0.303
fbs            0.0893      0.099      0.904      0.366      -0.105       0.283
restecg        0.1321      0.039      3.417      0.001       0.056       0.208
exang          0.1960      0.080      2.437      0.015       0.038       0.354
chol          -0.0019      0.000     -5.794      0.000      -0.002      -0.001
==============================================================================
Omnibus:                       81.790   Durbin-Watson:                   1.917
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              109.623
Skew:                           0.846   Prob(JB):                     1.57e-24
Kurtosis:                       3.831   Cond. No.                     3.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.91e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [168…

```
data_cleaned = data_cleaned.dropna(subset=['age', 'trestbps', 'thalch', 'oldpeak
X = data_cleaned[['age', 'trestbps', 'thalch', 'oldpeak', 'sex', 'cp', 'fbs', 'r
y = data_cleaned['num']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_

# Initialize and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
print("R-squared:", r2_score(y_test, y_pred))
```

```
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

```
R-squared: 0.382249792102602
Mean Squared Error: 0.8061845802675568
Coefficients: [ 1.47960179e-02 -2.11397611e-04 -5.45987756e-03  3.20096038e-01
  3.90277773e-01  2.32907223e-01  2.77445628e-02  1.30664271e-01
  1.55654860e-01 -1.84402977e-03]
Intercept: 0.08562560509436445
```