1. Data collection and storage:

- Create a folder named "data" in the main project directory.
- Inside the "data" folder, create another folder named "raw_data".
- Store the original dataset file "table.csv" in the "raw_data" folder.

```
import os
import pandas as pd

data_folder = 'data'
raw_folder = os.path.join(data_folder, 'raw')

if not os.path.exists(raw_folder):
    os.makedirs(raw_folder)

raw_data_file = os.path.join(raw_folder, 'data.csv')
raw_data = pd.read_csv(raw_data_file)
```

2. Data cleaning and preprocessing:

- Create a folder named "src" in the main project directory.
- Inside the "src" folder, create a file named "data_cleaning.py".
- Load the raw data from the "raw_data" folder using the pandas library.
- Clean the data by removing any missing or invalid values, and convert the columns to the appropriate data types.
- Save the cleaned data in a new file named "clean_data.csv" in a folder named "clean_data" inside the "data" folder.

```
import os
import pandas as pd
```

```
data_folder = 'data'
processed_folder = os.path.join(data_folder, 'processed')

if not os.path.exists(processed_folder):
    os.makedirs(processed_folder)

cleaned_data_file = os.path.join(processed_folder, 'cleaned_data.csv')

# Load the data
raw_data_file = os.path.join(data_folder, 'raw', 'data.csv')
raw_data = pd.read_csv(raw_data_file)

# Clean the data
cleaned_data = raw_data.dropna() # remove missing values

# Save the cleaned data
cleaned_data.to_csv(cleaned_data_file, index=False)
```

3. Data analysis and visualization:

- Create a folder named "notebooks" in the main project directory.
- Inside the "notebooks" folder, create a Jupyter notebook named "data_analysis.ipynb".
- Load the cleaned data from the "clean_data" folder using the pandas library.
- Perform data analysis, such as calculating descriptive statistics, correlation coefficients, and regression models.
- Create visualizations, such as scatter plots, histograms, and box plots, to better understand the relationships between the variables.
- Save the final results and visualizations in a folder named "results" inside the "data" folder.

```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

notebooks_folder = 'notebooks'
results_folder = os.path.join(notebooks_folder, 'results')

if not os.path.exists(results_folder):
    os.makedirs(results_folder)

# Load the cleaned data
cleaned_data_file = os.path.join('data', 'processed', 'cleaned_data.csv')
cleaned_data = pd.read_csv(cleaned_data_file)

# Create visualizations
```

```
sns.scatterplot(data=cleaned_data, x='study_time', y='final_grade')
plt.savefig(os.path.join(results_folder
```

```
project/
|
├── data/
|   ├── raw_data/
|   |   └── table.csv
|   ├── clean_data/
|   |   └── clean_data.csv
|   └── results/
|       ├── descriptive_stats.txt
|       ├── correlation_matrix.png
|       ├── regression_model_summary.txt
|       ├── scatter_plot.png
|       ├── histogram.png
|       └── box_plot.png
|
├── src/
|   └── data_cleaning.py
|
└── notebooks/
    └── data_analysis.ipynb
```

Task 1:

```
import pandas as pd
import os
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

# Set up folder structure
data_folder = 'data'
raw_data_folder = os.path.join(data_folder, 'raw_data')
cleaned_data_folder = os.path.join(data_folder, 'cleaned_data')
analysis_folder = os.path.join(data_folder, 'analysis')
exploratory_folder = os.path.join(analysis_folder, 'exploratory')
hypotheses_folder = os.path.join(analysis_folder, 'hypotheses')

# Create folders if they don't exist
os.makedirs(raw_data_folder, exist_ok=True)
os.makedirs(cleaned_data_folder, exist_ok=True)
os.makedirs(exploratory_folder, exist_ok=True)
os.makedirs(hypotheses_folder, exist_ok=True)

# Load raw data
raw_data_path = os.path.join(raw_data_folder, 'my_data.csv')
raw_data = pd.read_csv(raw_data_path)

# Data cleaning and preprocessing
cleaned_data = raw_data.dropna() # remove missing values
```

```
# Save cleaned data
cleaned_data_path = os.path.join(cleaned_data_folder, 'my_data_cleaned.csv')
cleaned_data.to_csv(cleaned_data_path)

# Create visualizations
def create_visualization(data, x_col, y_col, kind, save_path):
    sns.set_theme()
    plot = sns.catplot(data=data, x=x_col, y=y_col, kind=kind)
    plot.savefig(save_path)

create_visualization(cleaned_data, 'study_time', 'final_grade', 'scatter',
os.path.join(exploratory_folder, 'my_data_visualizations', 'scatterplot.png'))
create_visualization(cleaned_data, 'final_grade', None, 'hist',
os.path.join(exploratory_folder, 'my_data_visualizations', 'histogram.png'))
create_visualization(cleaned_data, 'gender', 'final_grade', 'bar',
os.path.join(exploratory_folder, 'my_data_visualizations', 'barplot.png'))
create_visualization(cleaned_data, None, 'final_grade', 'box',
os.path.join(exploratory_folder, 'my_data_visualizations', 'boxplot.png'))

# Modeling and inference
# Fit linear regression model
model = LinearRegression()
model.fit(cleaned_data[['study_time']], cleaned_data['final_grade'])

# Make predictions
predictions = model.predict(cleaned_data[['study_time']])
```

```
results = pd.DataFrame({'actual': cleaned_data['final_grade'], 'predicted': predictions})


# Save hypothesis results
hypothesis_results_path = os.path.join(hypotheses_folder, 'my_hypothesis_results.csv')
with open(hypothesis_results_path, 'w') as f:
    results.to_csv(f)
```