the three stages of a reproducible workflow are:

Data cleaning and preprocessing: This stage involves preparing the data for analysis by checking for missing values, outliers, and inconsistencies. This stage aims to make the data ready for analysis by ensuring it is accurate, consistent, and complete. In this stage, you can do the following work:

Check for missing values and remove them or impute them.

Check for outliers and deal with them accordingly.

Check for inconsistencies in the data, such as spelling errors, and correct them.

Rename columns and select relevant features.

Convert data types if needed.

The folder structure for this stage can be organized as follows:

```
data/
├── raw_data/
│   ├── data_file_1.csv
│   └── data_file_2.csv
└── cleaned_data/
    ├── data_file_1_cleaned.csv
    └── data_file_2_cleaned.csv
```

Exploratory data analysis: This stage involves exploring the data to gain insights and identify patterns. This stage aims to understand the data better and generate hypotheses for further analysis. In this stage, you can do the following work:

Compute summary statistics.

Create visualizations.

Identify patterns and correlations.

Check for assumptions.

Generate hypotheses for further analysis.

The folder structure for this stage can be organized as follows:

```
analysis/
├── exploratory/
│   ├── data_file_1_summary_stats.csv
│   ├── data_file_2_summary_stats.csv
│   ├── data_file_1_visualizations/
│   └── data_file_2_visualizations/
└── hypotheses/
    ├── hypothesis_1.csv
    └── hypothesis_2.csv
```

Modeling and inference: This stage involves building models and testing hypotheses. This stage aims to draw conclusions from the data and make predictions. In this stage, you can do the following work:

Select appropriate models.

Test hypotheses.

Validate models.

Make predictions.

The folder structure for this stage can be organized as follows:

```
analysis/
├── modeling/
│   ├── model_1.py
│   └── model_2.py
```

```
└── inference/
    ├── results_1.csv
    └── results_2.csv
```

Task 1: Reproducible workflow

```
import pandas as pd
import os

# 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 = 'analysis'
exploratory_folder = os.path.join(analysis_folder, 'exploratory')
hypotheses_folder = os.path.join(analysis_folder, 'hypotheses')

if not os.path.exists(raw_data_folder):
    os.makedirs(raw_data_folder)
if not os.path.exists(cleaned_data_folder):
    os.makedirs(cleaned_data_folder)
if not os.path.exists(exploratory_folder):
    os.makedirs(exploratory_folder)
if not os.path.exists(hypotheses_folder):
    os.makedirs(hypotheses_folder)
```

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

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

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

# Exploratory data analysis
summary_stats = cleaned_data.describe() # compute summary statistics
summary_stats.to_csv(os.path.join(exploratory_folder,
'my_data_summary_stats.csv'))

import matplotlib.pyplot as plt
import seaborn as sns

# Create visualizations
sns.scatterplot(data=cleaned_data, x='study_time', y='final_grade')
plt.savefig(os.path.join(exploratory_folder, 'my_data_visualizations',
'scatterplot.png'))

sns.histplot(data=cleaned_data, x='final_grade')
```

```
plt.savefig(os.path.join(exploratory_folder, 'my_data_visualizations',
'histogram.png'))


sns.barplot(data=cleaned_data, x='gender', y='final_grade')

plt.savefig(os.path.join(exploratory_folder, 'my_data_visualizations',
'barplot.png'))


sns.boxplot(data=cleaned_data, y='final_grade')

plt.savefig(os.path.join(exploratory_folder, 'my_data_visualizations',
'boxplot.png'))


# Modeling and inference

from sklearn.linear_model import LinearRegression


# 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})

results.to_csv(os.path.join(hypotheses_folder, 'my_hypothesis_results.csv'))
```