

```
[3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object']):
    plt.figure(figsize=(8, 6))
    sns.countplot(x=column, data=df)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45) # Rotate x-axis labels if needed
    plt.show()

# 2. Identify Outliers

# Box plots to visualize outliers
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=df[column])
    plt.title(f'Boxplot of {column}')
    plt.xlabel(column)
    plt.show()

# 3. Check for Correlations

# Correlation matrix (heatmap)
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Pair plots (for multiple pairwise relationships)
sns.pairplot(df)
plt.show()
```

```
FileNotFoundError                         Traceback (most recent call last)
Cell In[3], line 6
      3 import seaborn as sns
      4 # Load your dataset (replace 'your_data.csv' with the actual filename)
----> 5 df = pd.read_csv('your_data.csv')
      6 # 1. Visualize Variable Distributions
      7
      8 # Histograms for numerical variables
      9
     10 # Bar plots for categorical variables
     11 for column in df.select_dtypes(include=['float64', 'int64']):
```

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)

```
1013     kwds_defaults = _refine_defaults_read(
1014         dialect,
1015         delimiter,
1016         ...
1022         dtype_backend=dtype_backend,
1023     )
1024     kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
617     _validate_names(kwds.get("names", None))
618     # Create the parser.
--> 620     parser = TextFileReader(filepath_or_buffer, **kwds)
621     if chunksize or iterator:
622         return parser
```

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
1617 self.options["has_index_names"] = kwds["has_index_names"]
1619 self.handles = IOHandles(_None = None)
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFileReader._make_engine(self, f, engine)
1878 if "b" not in mode:
1879 mode += "b"
-> 1880 self.handles = get_handle(
1881 f,
1882 mode,
1883 encoding=self.options.get("encoding", None),
1884 compression=self.options.get("compression", None),
1885 memory_map=self.options.get("memory_map", False),
1886 is_text=is_text,
1887 errors=self.options.get("encoding_errors", "strict"),
1888 storage_options=self.options.get("storage_options", None),
1889)
1890 assert self.handles is not None
1891 f = self.handles.handle

File ~/anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)

```

868 elif isinstance(handle, str):
869     # Check whether the filename is to be opened in binary mode.
870     # Binary mode does not support 'encoding' and 'newline'.
871     if ioargs.encoding and "b" not in ioargs.mode:
872         # Encoding
--> 873         handle = open(
874             handle,
875             ioargs.mode,
876             encoding=ioargs.encoding,
877             errors=errors,
878             newline="",
879         )
880     else:
881         # Binary mode
882         handle = open(handle, ioargs.mode)

FileNotFoundException: [Errno 2] No such file or directory: 'your_data.csv'

[ ]:

[5]: import pandas as pd

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

# Read the CSV file into a DataFrame
df = pd.read_csv("data.csv")

# Display the first 5 rows
print(df.head().to_markdown(index=False, numalign="left", stralign="left"))

# Print the column names and their data types
print(df.info())

-----
FileNotFoundException          Traceback (most recent call last)
Cell In[5], line 7
    4 pd.set_option('display.max_columns', None)
    5 # Read the CSV file into a DataFrame
--> 7 df = pd.read_csv("data.csv")
    8 # Display the first 5 rows
    9 print(df.head().to_markdown(index=False, numalign="left", stralign="left"))

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    1013 kwds_defaults = _refine_defaults_read(
    1014     dialect,
    1015     delimiter,
    (...),
    1022     dtype_backend=dtype_backend,
    1023 )
    1024 kwds.update(kwds_defaults)
--> 1026 return _read(filepath_or_buffer, kwds)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
   617 _validate_names(kwds.get("names", None))
   619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
   622 if chunksize or iterator:
   623     return parser

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
   1617     self.options["has_index_names"] = kwds["has_index_names"]
   1619 self.handles: IOHandles | None = None
--> 1620 self._engine = self._make_engine(f, self.engine)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFileReader._make_engine(self, f, engine)
   1878     if "b" not in mode:
   1879         mode += "b"
--> 1880 self.handles = get_handle(
   1881     f,
   1882     mode,
   1883     encoding=self.options.get("encoding", None),
   1884     compression=self.options.get("compression", None),
   1885     memory_map=self.options.get("memory_map", False),
   1886     is_text=is_text,
   1887     errors=self.options.get("encoding_errors", "strict"),
   1888     storage_options=self.options.get("storage_options", None),
   1889 )
   1890 assert self.handles is not None
   1891 f = self.handles.handle

File ~/anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
   868 elif isinstance(handle, str):
   869     # Check whether the filename is to be opened in binary mode.
   870     # Binary mode does not support 'encoding' and 'newline'.
   871     if ioargs.encoding and "b" not in ioargs.mode:
   872         # Encoding
--> 873         handle = open(
   874             handle,
   875             ioargs.mode,
   876             encoding=ioargs.encoding,
   877             errors=errors,
   878             newline="",
   879         )
   880     else:
   881         # Binary mode
   882         handle = open(handle, ioargs.mode)

FileNotFoundException: [Errno 2] No such file or directory: 'data.csv'

[7]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))

```

```

sns.histplot(df[column], kde=True)
plt.title(f'Distribution of {column}')
plt.xlabel(column)
plt.ylabel('Frequency')
plt.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object']):
    plt.figure(figsize=(8, 6))
    sns.countplot(x=column, data=df)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45) # Rotate x-axis labels if needed
    plt.show()

# 2. Identify Outliers

# Box plots to visualize outliers
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=df[column])
    plt.title(f'Boxplot of {column}')
    plt.xlabel(column)
    plt.show()

# 3. Check for Correlations

# Correlation matrix (heatmap)
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Pair plots (for multiple pairwise relationships)
sns.pairplot(df)
plt.show()

```

```

-----
FileNotFoundError                         Traceback (most recent call last)
Cell In[7], line 6
      3 import seaborn as sns
      5 # Load your dataset (replace 'your_data.csv' with the actual filename)
----> 6 df = pd.read_csv('your_data.csv')
      8 # Visualize Variable Distributions
      9
     10 # Histograms for numerical variables
     11 for column in df.select_dtypes(include=['float64', 'int64']):
     12
     13     # Your code here to visualize each numerical variable

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
     1013 kwds_defaults = _refine_defaults_read(
     1014     dialect,
     1015     delimiter,
     (...),
     1022     dtype_backend=dtype_backend,
     1023 )
     1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
   617     _validate_names(kwds.get("names", None))
   619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
   622 if chunksize or iterator:
   623     return parser

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
  1617     self.options["has_index_names"] = kwds["has_index_names"]
  1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFileReader._make_engine(self, f, engine)
  1878     if "b" not in mode:
  1879         mode += "b"
-> 1880 self.handles = get_handle(
  1881     f,
  1882     mode,
  1883     encoding=self.options.get("encoding", None),
  1884     compression=self.options.get("compression", None),
  1885     memory_map=self.options.get("memory_map", False),
  1886     is_text=is_text,
  1887     errors=self.options.get("encoding_errors", "strict"),
  1888     storage_options=self.options.get("storage_options", None),
  1889 )
  1890 assert self.handles is not None
  1891 f = self.handles.handle

File ~/anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
  868     elif isinstance(handle, str):
  869         # Check whether the filename is to be opened in binary mode.
  870         # Binary mode does not support 'encoding' and 'newline'.
  871         if ioargs.encoding and "b" not in ioargs.mode:
  872             # Encoding
--> 873             handle = open(
  874                 handle,
  875                 ioargs.mode,
  876                 encoding=ioargs.encoding,
  877                 errors=errors,
  878                 newline="",
  879             )
  880     else:
  881         # Binary mode
  882         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'your_data.csv'

```

```
[9]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object', 'category']):
    plt.figure(figsize=(8, 6))
    sns.countplot(x=column, data=df)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45) # Rotate x-axis labels if needed
    plt.show()

# 2. Identify Outliers

# Box plots to visualize outliers
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=df[column])
    plt.title(f'Boxplot of {column}')
    plt.xlabel(column)
    plt.show()

# 3. Check for Correlations

# Correlation matrix (heatmap)
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Pair plots (for multiple pairwise relationships)
sns.pairplot(df)
plt.show()

```

```

-----
FileNotFoundError                         Traceback (most recent call last)
Cell In[9], line 6
      3 import seaborn as sns
      5 # Load your dataset (replace 'your_data.csv' with the actual filename)
-> 6 df = pd.read_csv('your_data.csv')
     8 # 1. Visualize Variable Distributions
     9
10 # Histograms for numerical variables
11 for column in df.select_dtypes(include=['float64', 'int64']):
12
13     kwds_defaults = _refine_defaults_read(
14         dialect,
15         delimiter,
16         ...)
17     dtype_backend=dtype_backend,
18 )
19 kwds.update(kwds_defaults)
-> 20 return _read(filepath_or_buffer, kwds)

File ~/anaconda3/lib/site-packages/pandas/io/parsers/readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    1013 kwds_defaults = _refine_defaults_read(
    1014     dialect,
    1015     delimiter,
    1016     ...
    1017     dtype_backend=dtype_backend,
    1018 )
    1019 kwds.update(kwds_defaults)
-> 1020 return _read(filepath_or_buffer, kwds)

File ~/anaconda3/lib/site-packages/pandas/io/parsers.py:620, in _read(filepath_or_buffer, kwds)
   617 _validate_names(kwds.get("names", None))
   618 # Create the parser.
-> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
   621 if chunksize or iterator:
   622     return parser

File ~/anaconda3/lib/site-packages/pandas/io/parsers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
   1617 self.options["has_index_names"] = kwds["has_index_names"]
   1618 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~/anaconda3/lib/site-packages/pandas/io/parsers/readers.py:1880, in TextFileReader._make_engine(self, f, engine)
   1878     if "b" not in mode:
   1879         mode += "b"
-> 1880 self.handles = get_handle(
   1881     f,
   1882     mode,
   1883     encoding=self.options.get("encoding", None),
   1884     compression=self.options.get("compression", None),
   1885     memory_map=self.options.get("memory_map", False),
   1886     is_text=is_text,
   1887     errors=self.options.get("encoding_errors", "strict"),
   1888     storage_options=self.options.get("storage_options", None),
   1889 )
1890 assert self.handles is not None
1891 f = self.handles.handle

File ~/anaconda3/lib/site-packages/pandas/io/common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
   868 elif isinstance(handle, str):
   869     # Check whether the filename is to be opened in binary mode.
   870     # Binary mode does not support 'encoding' and 'newline'.
-> 871     if ioargs.encoding and "b" not in ioargs.mode:
   872         # Encoding
   873         handle = open(
   874             handle,
   875             ioargs.mode,
   876             encoding=ioargs.encoding,
   877             errors=errors,
   878             newline="",
   879         )

```

```

880     else:
881         # Binary mode
882         handle = open(handle, ioargs.mode)
883
884 FileNotFoundError: [Errno 2] No such file or directory: 'your_data.csv'
[ ]:

```

```

[12]: import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns

      # Load your dataset
      data = pd.read_csv("your_dataset.csv") # Replace with your actual file path

      # 1. Data Overview and Summary Statistics
      print(data.head()) # Display the first few rows
      print(data.describe()) # Summary statistics (mean, std, min, max, quartiles)

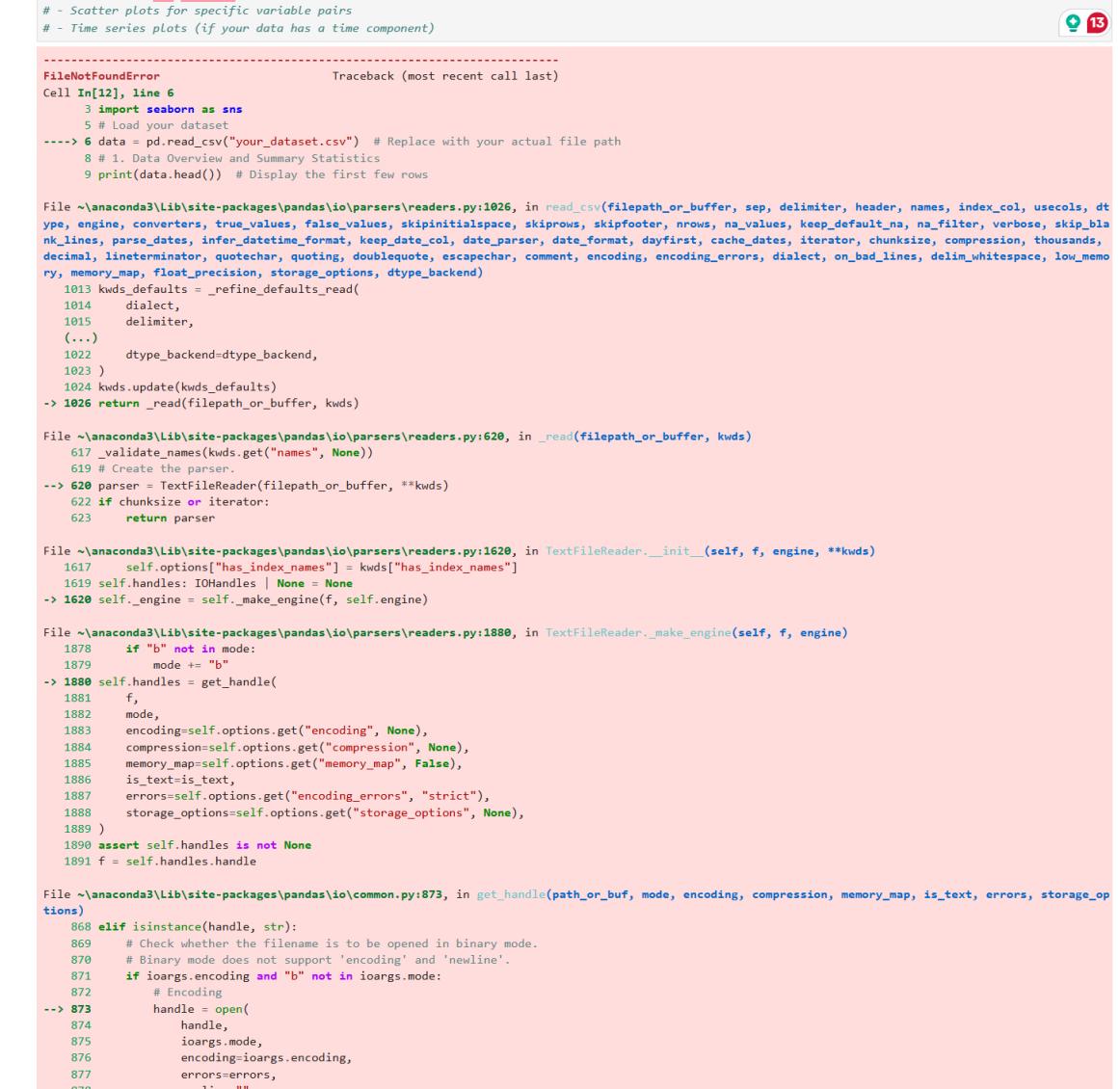
      # 2. Data Visualization
      # 2.1 Histograms for variable distributions
      plt.figure(figsize=(10, 6))
      for column in data.columns:
          plt.subplot(2, 3, len(plt.gcf().axes) + 1) # Dynamic subplot creation
          plt.hist(data[column], bins=20, alpha=0.7)
          plt.title(f'Histogram of {column}')
          plt.xlabel(column)
          plt.ylabel('Frequency')
      plt.tight_layout()
      plt.show()

      # 2.2 Boxplots for outlier detection
      plt.figure(figsize=(10, 6))
      for column in data.columns:
          plt.subplot(2, 3, len(plt.gcf().axes) + 1)
          sns.boxplot(x=data[column])
          plt.title(f'Boxplot of {column}')
      plt.tight_layout()
      plt.show()

      # 3. Correlation Analysis
      correlation_matrix = data.corr()
      plt.figure(figsize=(8, 6))
      sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
      plt.title('Correlation Matrix')
      plt.show()

      # Additional EDA techniques (if needed)
      # - Pair plots (sns.pairplot) for visualizing relationships between multiple variables
      # - Scatter plots for specific variable pairs
      # - Time series plots (if your data has a time component)

```



```

FileNotFoundError                                     Traceback (most recent call last)
Cell In[12], line 6
      3 import seaborn as sns
      5 # Load your dataset
--> 6 data = pd.read_csv("your_dataset.csv") # Replace with your actual file path
      8 # 1. Data Overview and Summary Statistics
      9 print(data.head()) # Display the first few rows

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    1013 kwds_defaults = _refine_defaults(read(
    1014     dialect,
    1015     delimiter,
    (...),
    1022     dtype_backend=dtype_backend,
    1023 )
    1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
   617 _validate_names(kwds.get("names", None))
   619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
   622 if chunksize or iterator:
   623     return parser

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
  1617     self.options["has_index_names"] = kwds["has_index_names"]
  1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFileReader._make_engine(self, f, engine)
  1878     if "b" not in mode:
  1879         mode += "b"
-> 1880 self.handles = get_handle(
  1881     f,
  1882     mode,
  1883     encoding=self.options.get("encoding", None),
  1884     compression=self.options.get("compression", None),
  1885     memory_map=self.options.get("memory_map", False),
  1886     is_text=is_text,
  1887     errors=self.options.get("encoding_errors", "strict"),
  1888     storage_options=self.options.get("storage_options", None),
  1889 )
  1890 assert self.handles is not None
  1891 f = self.handles.handle

File ~/anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
  868 elif isinstance(handle, str):
  869     # Check whether the filename is to be opened in binary mode.
  870     # Binary mode does not support 'encoding' and 'newline'.
  871     if ioargs.encoding and "b" not in ioargs.mode:
  872         # Encoding
-> 873         handle = open(
  874             handle,
  875             ioargs.mode,
  876             encoding=ioargs.encoding,
  877             errors=errors,
  878             buffering=1024

```

```

676         newline= ,
677     )
680 else:
681     # Binary mode
682     handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'your_dataset.csv'

[14]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_dataset.csv' with your actual file)
df = pd.read_csv('your_dataset.csv')

# 1. Visualize variable distributions

# Histograms for numerical variables
for col in df.select_dtypes(include=['float64', 'int64']).columns:
    plt.figure(figsize=(8, 6))
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()

# Box plots for numerical variables to identify outliers
for col in df.select_dtypes(include=['float64', 'int64']).columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(data=df, x=col)
    plt.title(f'Box Plot of {col}')
    plt.show()

# Bar plots for categorical variables
for col in df.select_dtypes(include=['object']).columns:
    plt.figure(figsize=(8, 6))
    sns.countplot(data=df, x=col)
    plt.title(f'Distribution of {col}')
    plt.xticks(rotation=45)
    plt.show()

# 2. Check for correlations between variables

# Correlation matrix (for numerical variables)
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Scatter plots for pairs of variables (for numerical variables)
for col1 in df.select_dtypes(include=['float64', 'int64']).columns:
    for col2 in df.select_dtypes(include=['float64', 'int64']).columns:
        if col1 != col2:
            plt.figure(figsize=(8, 6))
            sns.scatterplot(data=df, x=col1, y=col2)
            plt.title(f'Scatter Plot of {col1} vs. {col2}')
            plt.show()

```

```

-----
FileNotFoundError                         Traceback (most recent call last)
Cell In[14], line 6
      3 import seaborn as sns
      5 # Load your dataset (replace 'your_dataset.csv' with your actual file)
----> 6 df = pd.read_csv('your_dataset.csv')
      8 # 1. Visualize variable distributions
      9
10 # Histograms for numerical variables
11 for col in df.select_dtypes(include=['float64', 'int64']).columns:
12     plt.figure(figsize=(8, 6))
13     sns.histplot(df[col], kde=True)
14     plt.title(f'Distribution of {col}')
15     plt.show()

File ~/anaconda3/Lib/site-packages/pandas/io/parsers/readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dt
6 encoding=ioargs.encoding,
7 errors=errors,
8 newline='',
9 )
10 else:
11     # Binary mode
12     handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'your_dataset.csv'

```

```

[16]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Bar plots for categorical variables
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object']):
    plt.figure(figsize=(8, 6))
    sns.countplot(data=df, x=column)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

```

```

plt.figure(figsize=(8, 6))
sns.countplot(x=column, data=df)
plt.title(f'Distribution of {column}')
plt.xlabel(column)
plt.ylabel('Count')
plt.xticks(rotation=45) # Rotate x-axis labels if needed
plt.show()

# 2. Identify Outliers

# Box plots to visualize outliers
for column in df.select_dtypes(include=['float64', 'int64']):
    10 # Histograms for numerical variables
    11 for column in df.select_dtypes(include=['float64', 'int64']):

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    1013 kwds_defaults = _refine_defaults_read(
    1014     dialect,
    1015     delimiter,
    (...),
    1022     dtype_backend=dtype_backend,
    1023 )
    1024 kwds.update(kwds_defaults)
--> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
    1617 self.options["has_index_names"] = kwds["has_index_names"]
    1619 self.handles: IOHandles | None = None
--> 1620 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1880, in TextFileReader._make_engine(self, f, engine)
    869     # Check whether the filename is to be opened in binary mode.
    870     # Binary mode does not support 'encoding' and 'newline'.
    871     if ioargs.encoding and "b" not in ioargs.mode:
    872         # Encoding
--> 873         handle = open(
    874             handle,
    875             ioargs.mode,
    876             encoding=ioargs.encoding,
    877             errors=errors,
    878             newline="",
    879         )
    880     else:
    881         # Binary mode
    882         handle = open(handle, ioargs.mode)

FileNotFoundException: [Errno 2] No such file or directory: 'your_data.csv'

[18]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    import seaborn as sns

# Load your dataset (replace 'your_data.csv' with the actual filename)
df = pd.read_csv('your_data.csv')

# 1. Visualize Variable Distributions

# Histograms for numerical variables
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object']):
    plt.figure(figsize=(8, 6))
    sns.countplot(x=column, data=df)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45) # Rotate x-axis labels if needed
    plt.show()

# 2. Identify Outliers
.

-----
FileNotFoundError                         Traceback (most recent call last)
Cell In[18], line 6
      3 import seaborn as sns
      5 # Load your dataset (replace 'your_data.csv' with the actual filename)
--> 6 df = pd.read_csv('your_data.csv')
      8 # 1. Visualize Variable Distributions
      9
     10 # Histograms for numerical variables
     11 for column in df.select_dtypes(include=['float64', 'int64']):
```

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)

10 # Histograms for numerical variables

11 for column in df.select_dtypes(include=['float64', 'int64']):

```

    ry, memory_map, float_precision, storage_options, dtype_backend)
1013     kwds_defaults = _refine_defaults_read(
1014         dialect,
1015         delimiter,
1016         (...),
1022         dtype_backend=dtype_backend,
1023     )
1024     kwds.update(kwds_defaults)
--> 1026 return _read(filepath_or_buffer, kwds)

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in _read(filepath_or_buffer, kwds)
   617     _validate_names(kwds.get("names", None))
   619 # Create the parser.
--> 620     pit = plt.FacetGrid(df, **kwds)
   621     pit.ylabel('Frequency')
   622     pit.show()

# Bar plots for categorical variables
for column in df.select_dtypes(include=['object']):
    plt.figure(figsize=(8, 6))
    sns.countplot(x=column, data=df)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45) # Rotate x-axis labels if needed
    plt.show()

# 2. Identify Outliers

# Box plots to visualize outliers
for column in df.select_dtypes(include=['float64', 'int64']):
    plt.figure(figsize=(8, 6))
    sns.boxplot(x=df[column])
    plt.title(f'Boxplot of {column}')
    plt.xlabel(column)
    plt.show()

# 3. Check for Correlations

# Correlation matrix (heatmap)
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
plt.title('Correlation Matrix')
plt.show()

# Pair plots (for multiple pairwise relationships)
sns.pairplot(df)
plt.show()

-----
FileNotFoundError                         Traceback (most recent call last)
Cell In[18], line 6
      3 import seaborn as sns
      5 # Load your dataset (replace 'your_data.csv' with the actual filename)
--> 6 df = pd.read_csv('your_data.csv')
      8 # 1. Visualize Variable Distributions
      9
10 # Histograms for numerical variables
11 for column in df.select_dtypes(include=['float64', 'int64']):

```

```

File ~/anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtypes, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
1013     kwds_defaults = _refine_defaults_read(
1014         dialect,
1015         delimiter,
1016         (...),
1022         dtype_backend=dtype_backend,
1023     )
1024     kwds.update(kwds_defaults)
--> 1026 handle = self._handle(handle_defaults)
1027
1028 elif isinstance(handle, str):
1029     # Check whether the filename is to be opened in binary mode.
1030     # Binary mode does not support 'encoding' and 'newline'.
1031     if ioargs.encoding and "b" not in ioargs.mode:
1032         # Encoding
--> 1033         handle = open(
1034             handle,
1035             ioargs.mode,
1036             encoding=ioargs.encoding,
1037             errors=errors,
1038             newline="",
1039         )
1040     else:
1041         # Binary mode
1042         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'your_data.csv'

```

[]:



```

1887     errors=self.options.get("encoding_errors", "strict"),
1888     storage_options=self.options.get("storage_options", None),
1889 )
1890 assert self.handles is not None
1891 f = self.handles.handle

File ~/anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
   868 elif isinstance(handle, str):
   869     # Check whether the filename is to be opened in binary mode.
   870     # Binary mode does not support 'encoding' and 'newline'.
   871     if ioargs.encoding and "b" not in ioargs.mode:
   872         # Encoding
--> 873         handle = open(
   874             handle,
   875             ioargs.mode,

```

```
876         encoding=ioargs.encoding,
877         errors=errors,
878         newline="",
879     )
880     else:
881         # Binary mode
882         handle = open(handle, ioargs.mode)

FileNotFoundException: [Errno 2] No such file or directory: 'your_data.csv'
```

[]:



```
file ~\anaconda3\lib\site-packages\pandas\io\common.py:873, in get_handle(path_or_buf, mode, encoding, compression, memory_map, as_text, errors, storage_options)
868 elif isinstance(handle, str):
869     # Check whether the filename is to be opened in binary mode.
870     # Binary mode does not support 'encoding' and 'newline'.
871     if ioargs.encoding and "b" not in ioargs.mode:
872         # Encoding
--> 873     handle = open(
874         handle,
875         ioargs.mode,
876         encoding=ioargs.encoding,
877         errors=errors,
878         newline="",
879     )
880 else:
881     # Binary mode
882     handle = open(handle, ioargs.mode)

FileNotFoundException: [Errno 2] No such file or directory: 'your_data.csv'
```

[]:

