```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Load Titanic dataset from seaborn
df = sns.load_dataset('titanic')

df.info()
df.describe(include='all')
df.head()
print(df.isnull().sum())

df['sex'].value_counts().plot(kind='bar')
plt.title('Gender Distribution')
plt.show()

df['age'].hist(bins=30)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

sns.pairplot(df, hue='survived')
plt.show()

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

sns.barplot(x='sex', y='survived', data=df)
plt.title('Survival Rate by Gender')
plt.show()

sns.boxplot(x='survived', y='age', data=df)
plt.title('Age Distribution by Survival')
plt.show()

sns.boxplot(x=df['fare'])
plt.title('Fare Distribution with Outliers')
plt.show()
```

Out[4]:

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:1344, in
AbstractHTTPHandler.do_open(self, http_class, req, **http_conn_args)
   1343 try:
-> 1344     h.request(req.get_method(), req.selector, req.data, headers,
   1345               encode_chunked=req.has_header('Transfer-encoding'))
   1346 except OSError as err: # timeout error
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1338, in
HTTPConnection.request(self, method, url, body, headers, encode_chunked)
   1337 """Send a complete request to the server."""
-> 1338 self._send_request(method, url, body, headers, encode_chunked)
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1384, in
HTTPConnection._send_request(self, method, url, body, headers, encode_chunked)
```

```
   1383     body = _encode(body, 'body')
-> 1384 self.endheaders(body, encode_chunked=encode_chunked)
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1333, in
HTTPConnection.endheaders(self, message_body, encode_chunked)
   1332     raise CannotSendHeader()
-> 1333 self._send_output(message_body, encode_chunked=encode_chunked)
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1093, in
HTTPConnection._send_output(self, message_body, encode_chunked)
   1092 del self._buffer[:]
-> 1093 self.send(msg)
   1095 if message_body is not None:
   1096
   1097     # create a consistent interface to message_body
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1037, in
HTTPConnection.send(self, data)
   1036 if self.auto_open:
-> 1037     self.connect()
   1038 else:
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1472, in
HTTPSConnection.connect(self)
   1470 "Connect to a host on a given (SSL) port."
-> 1472 super().connect()
   1474 if self._tunnel_host:
File /ext/venvs/cocalc/lib/python3.12/http/client.py:1003, in
HTTPConnection.connect(self)
   1002 sys.audit("http.client.connect", self, self.host, self.port)
-> 1003 self.sock = self._create_connection(
   1004     (self.host,self.port), self.timeout, self.source_address)
   1005 # Might fail in OSs that don't implement TCP_NODELAY
File /ext/venvs/cocalc/lib/python3.12/socket.py:865, in
create_connection(address, timeout, source_address, all_errors)
    864 if not all_errors:
--> 865     raise exceptions[0]
    866 raise ExceptionGroup("create_connection failed", exceptions)
File /ext/venvs/cocalc/lib/python3.12/socket.py:850, in
create_connection(address, timeout, source_address, all_errors)
    849     sock.bind(source_address)
--> 850 sock.connect(sa)
    851 # Break explicitly a reference cycle
OSError: [Errno 99] Cannot assign requested address

During handling of the above exception, another exception occurred:
URLError                                  Traceback (most recent call last)
Cell In[4], line 8
      5 get_ipython().run_line_magic('matplotlib', 'inline')
      7 # Load Titanic dataset from seaborn
----> 8 df = sns.load_dataset('titanic')
     10 df.info()
     11 df.describe(include='all')
File /ext/venvs/cocalc/lib/python3.12/site-packages/seaborn/utils.py:572, in
load_dataset(name, cache, data_home, **kws)
    570 cache_path = os.path.join(get_data_home(data_home),
os.path.basename(url))
    571 if not os.path.exists(cache_path):
--> 572     if name not in get_dataset_names():
    573         raise ValueError(f"'{name}' is not one of the example
datasets.")
    574     urlretrieve(url, cache_path)
File /ext/venvs/cocalc/lib/python3.12/site-packages/seaborn/utils.py:499, in
get_dataset_names()
    493 def get_dataset_names():
```

```
    494     """Report available example datasets, useful for reporting issues.
    495
    496     Requires an internet connection.
    497
    498     """
--> 499     with urlopen(DATASET_NAMES_URL) as resp:
    500         txt = resp.read()
    502     dataset_names = [name.strip() for name in txt.decode().split("\n")]
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:215, in urlopen(url,
data, timeout, cafile, capath, cadefault, context)
    213 else:
    214     opener = _opener
--> 215 return opener.open(url, data, timeout)
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:515, in
OpenerDirector.open(self, fullurl, data, timeout)
    512     req = meth(req)
    514 sys.audit('urllib.Request', req.full_url, req.data, req.headers,
req.get_method())
--> 515 response = self._open(req, data)
    517 # post-process response
    518 meth_name = protocol+"_response"
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:532, in
OpenerDirector._open(self, req, data)
    529     return result
    531 protocol = req.type
--> 532 result = self._call_chain(self.handle_open, protocol, protocol +
    533                           '_open', req)
    534 if result:
    535     return result
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:492, in
OpenerDirector._call_chain(self, chain, kind, meth_name, *args)
    490 for handler in handlers:
    491     func = getattr(handler, meth_name)
--> 492     result = func(*args)
    493     if result is not None:
    494         return result
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:1392, in
HTTPSHandler.https_open(self, req)
   1391 def https_open(self, req):
-> 1392     return self.do_open(http.client.HTTPSConnection, req,
   1393                         context=self._context)
File /ext/venvs/cocalc/lib/python3.12/urllib/request.py:1347, in
AbstractHTTPHandler.do_open(self, http_class, req, **http_conn_args)
   1344         h.request(req.get_method(), req.selector, req.data, headers,
   1345                   encode_chunked=req.has_header('Transfer-encoding'))
   1346     except OSError as err: # timeout error
-> 1347         raise URLError(err)
   1348     r = h.getresponse()
   1349 except:
URLError: <urlopen error [Errno 99] Cannot assign requested address>
```