Q8 stat130 week 01

September 11, 2024

[1]:

```
url = "https://raw.githubusercontent.com/datasets/commodity-prices/master/data/
 ↪commodity-prices.csv"
com = pd.read_csv(url)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 2
      1 url = "https://raw.githubusercontent.com/datasets/commodity-prices/
 ↪master/data/commodity-prices.csv"
----> 2 com = pd.read_csv(url)

NameError: name 'pd' is not defined
```

[2]:

```
import pandas as pd

url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanics.
 ↪csv"

df = pd.read_csv(url)
```

```
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
Cell In[2], line 5
      1 import pandas as pd
      3 url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/
 ↪titanics.csv"
----> 5 df = pd.read_csv(url)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:948,␣
 ↪in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, dtype, 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)
    935 kwds_defaults = _refine_defaults_read(
    936     dialect,
    937     delimiter,
   (…)
    944     dtype_backend=dtype_backend,
    945 )
    946 kwds.update(kwds_defaults)
--> 948 return _read(filepath_or_buffer, kwds)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:611,␣
 ↪in _read(filepath_or_buffer, kwds)
    608 _validate_names(kwds.get("names", None))
    610 # Create the parser.
--> 611 parser = TextFileReader(filepath_or_buffer, **kwds)
    613 if chunksize or iterator:
    614     return parser

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1448,
 ↪in TextFileReader.__init__(self, f, engine, **kwds)
   1445     self.options["has_index_names"] = kwds["has_index_names"]
   1447 self.handles: IOHandles | None = None
-> 1448 self._engine = self._make_engine(f, self.engine)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1705,
 ↪in TextFileReader._make_engine(self, f, engine)
   1703     if "b" not in mode:
   1704         mode += "b"
-> 1705 self.handles = get_handle(
   1706     f,
   1707     mode,
   1708     encoding=self.options.get("encoding", None),
   1709     compression=self.options.get("compression", None),
   1710     memory_map=self.options.get("memory_map", False),
   1711     is_text=is_text,
   1712     errors=self.options.get("encoding_errors", "strict"),
   1713     storage_options=self.options.get("storage_options", None),
   1714 )
   1715 assert self.handles is not None
   1716 f = self.handles.handle

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:718, in␣
 ↪get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,␣
 ↪errors, storage_options)
    715     codecs.lookup_error(errors)
    717 # open URLs
--> 718 ioargs = _get_filepath_or_buffer(
    719     path_or_buf,
    720     encoding=encoding,
    721     compression=compression,
    722     mode=mode,
    723     storage_options=storage_options,
    724 )
    726 handle = ioargs.filepath_or_buffer
    727 handles: list[BaseBuffer]

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:372, in _get_filepath_or_buffer(filepath_or_buffer, encoding, compression, mode, storage_options)
    370 # assuming storage_options is to be interpreted as headers
    371 req_info = urllib.request.Request(filepath_or_buffer, headers=storage_options)
--> 372 with urlopen(req_info) as req:
    373     content_encoding = req.headers.get("Content-Encoding", None)
    374     if content_encoding == "gzip":
    375         # Override compression based on Content-Encoding header

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:274, in urlopen(*args, **kwargs)
    268 """
    269 Lazy-import wrapper for stdlib urlopen, as that imports a big chunk of
    270 the stdlib.
    271 """
    272 import urllib.request
--> 274 return urllib.request.urlopen(*args, **kwargs)

File /opt/conda/lib/python3.11/urllib/request.py:216, in urlopen(url, data, timeout, cafile, capath, cadefault, context)
    214 else:
    215     opener = _opener
--> 216 return opener.open(url, data, timeout)

File /opt/conda/lib/python3.11/urllib/request.py:525, in OpenerDirector.open(self, fullurl, data, timeout)
    523 for processor in self.process_response.get(protocol, []):
    524     meth = getattr(processor, meth_name)
--> 525     response = meth(req, response)
    527 return response

File /opt/conda/lib/python3.11/urllib/request.py:634, in HTTPErrorProcessor.http_response(self, request, response)
    631 # According to RFC 2616, "2xx" code indicates that the client's
    632 # request was successfully received, understood, and accepted.
    633 if not (200 <= code < 300):
--> 634     response = self.parent.error(
    635         'http', request, response, code, msg, hdrs)
    637 return response
```

```
File /opt/conda/lib/python3.11/urllib/request.py:563, in OpenerDirector.
 ↪error(self, proto, *args)
    561 if http_err:
    562     args = (dict, 'default', 'http_error_default') + orig_args
--> 563     return self._call_chain(*args)

File /opt/conda/lib/python3.11/urllib/request.py:496, in OpenerDirector.
 ↪_call_chain(self, chain, kind, meth_name, *args)
    494 for handler in handlers:
    495     func = getattr(handler, meth_name)
--> 496     result = func(*args)
    497     if result is not None:
    498         return result

File /opt/conda/lib/python3.11/urllib/request.py:643, in HTTPDefaultErrorHandler.
 ↪http_error_default(self, req, fp, code, msg, hdrs)
    642 def http_error_default(self, req, fp, code, msg, hdrs):
--> 643     raise HTTPError(req.full_url, code, msg, hdrs, fp)

HTTPError: HTTP Error 404: Not Found
```

[3]:

```
import pandas as pd

url = "https://raw.githubusercontent.com/mwaskome/seaborn-data/master/titanic.
 ↪csv"

df = pd.read_csv(url)
```

```
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
Cell In[3], line 5
      1 import pandas as pd
      3 url = "https://raw.githubusercontent.com/mwaskome/seaborn-data/master/
 ↪titanic.csv"
----> 5 df = pd.read_csv(url)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:948,␣
 ↪in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, dtype, 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)
    935 kwds_defaults = _refine_defaults_read(
    936     dialect,
    937     delimiter,
   (…)
    944     dtype_backend=dtype_backend,
    945 )
    946 kwds.update(kwds_defaults)
--> 948 return _read(filepath_or_buffer, kwds)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:611, in _read(filepath_or_buffer, kwds)
    608 _validate_names(kwds.get("names", None))
    610 # Create the parser.
--> 611 parser = TextFileReader(filepath_or_buffer, **kwds)
    613 if chunksize or iterator:
    614     return parser

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1448, in TextFileReader.__init__(self, f, engine, **kwds)
   1445     self.options["has_index_names"] = kwds["has_index_names"]
   1447 self.handles: IOHandles | None = None
-> 1448 self._engine = self._make_engine(f, self.engine)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1705, in TextFileReader._make_engine(self, f, engine)
   1703     if "b" not in mode:
   1704         mode += "b"
-> 1705 self.handles = get_handle(
   1706     f,
   1707     mode,
   1708     encoding=self.options.get("encoding", None),
   1709     compression=self.options.get("compression", None),
   1710     memory_map=self.options.get("memory_map", False),
   1711     is_text=is_text,
   1712     errors=self.options.get("encoding_errors", "strict"),
   1713     storage_options=self.options.get("storage_options", None),
   1714 )
   1715 assert self.handles is not None
   1716 f = self.handles.handle

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:718, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    715     codecs.lookup_error(errors)
    717 # open URLs
--> 718 ioargs = _get_filepath_or_buffer(
    719     path_or_buf,
    720     encoding=encoding,
    721     compression=compression,
    722     mode=mode,
    723     storage_options=storage_options,
    724 )
    726 handle = ioargs.filepath_or_buffer
    727 handles: list[BaseBuffer]

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:372, in␣
 ↪_get_filepath_or_buffer(filepath_or_buffer, encoding, compression, mode,␣
 ↪storage_options)
    370 # assuming storage_options is to be interpreted as headers
    371 req_info = urllib.request.Request(filepath_or_buffer,␣
 ↪headers=storage_options)
--> 372 with urlopen(req_info) as req:
    373     content_encoding = req.headers.get("Content-Encoding", None)
    374     if content_encoding == "gzip":
    375         # Override compression based on Content-Encoding header

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:274, in␣
 ↪urlopen(*args, **kwargs)
    268 """
    269 Lazy-import wrapper for stdlib urlopen, as that imports a big chunk of
    270 the stdlib.
    271 """
    272 import urllib.request
--> 274 return urllib.request.urlopen(*args, **kwargs)

File /opt/conda/lib/python3.11/urllib/request.py:216, in urlopen(url, data,␣
 ↪timeout, cafile, capath, cadefault, context)
    214 else:
    215     opener = _opener
--> 216 return opener.open(url, data, timeout)

File /opt/conda/lib/python3.11/urllib/request.py:525, in OpenerDirector.
 ↪open(self, fullurl, data, timeout)
    523 for processor in self.process_response.get(protocol, []):
    524     meth = getattr(processor, meth_name)
--> 525     response = meth(req, response)
    527 return response

File /opt/conda/lib/python3.11/urllib/request.py:634, in HTTPErrorProcessor.
 ↪http_response(self, request, response)
    631 # According to RFC 2616, "2xx" code indicates that the client's
    632 # request was successfully received, understood, and accepted.
    633 if not (200 <= code < 300):
--> 634     response = self.parent.error(
    635         'http', request, response, code, msg, hdrs)
    637 return response

File /opt/conda/lib/python3.11/urllib/request.py:563, in OpenerDirector.
 ↪error(self, proto, *args)
```

```
    561 if http_err:
    562     args = (dict, 'default', 'http_error_default') + orig_args
--> 563     return self._call_chain(*args)

File /opt/conda/lib/python3.11/urllib/request.py:496, in OpenerDirector.
  ↪_call_chain(self, chain, kind, meth_name, *args)
    494 for handler in handlers:
    495     func = getattr(handler, meth_name)
--> 496     result = func(*args)
    497     if result is not None:
    498         return result

File /opt/conda/lib/python3.11/urllib/request.py:643, in HTTPDefaultErrorHandler.
  ↪http_error_default(self, req, fp, code, msg, hdrs)
    642 def http_error_default(self, req, fp, code, msg, hdrs):
--> 643     raise HTTPError(req.full_url, code, msg, hdrs, fp)

HTTPError: HTTP Error 404: Not Found
```

[4]:

```
import pandas as pd

url = "httpse://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.
 ↪csv"

df = pd.read_csv(url)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[4], line 5
      1 import pandas as pd
      3 url = "httpse://raw.githubusercontent.com/mwaskom/seaborn-data/master/
  ↪titanic.csv"
----> 5 df = pd.read_csv(url)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:948,␣
  ↪in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
  ↪usecols, dtype, 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)
    935 kwds_defaults = _refine_defaults_read(
    936     dialect,
    937     delimiter,
   (…)
    944     dtype_backend=dtype_backend,
    945 )
    946 kwds.update(kwds_defaults)
--> 948 return _read(filepath_or_buffer, kwds)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:611, in _read(filepath_or_buffer, kwds)
    608 _validate_names(kwds.get("names", None))
    610 # Create the parser.
--> 611 parser = TextFileReader(filepath_or_buffer, **kwds)
    613 if chunksize or iterator:
    614     return parser

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1448, in TextFileReader.__init__(self, f, engine, **kwds)
   1445     self.options["has_index_names"] = kwds["has_index_names"]
   1447 self.handles: IOHandles | None = None
-> 1448 self._engine = self._make_engine(f, self.engine)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1705, in TextFileReader._make_engine(self, f, engine)
   1703     if "b" not in mode:
   1704         mode += "b"
-> 1705 self.handles = get_handle(
   1706     f,
   1707     mode,
   1708     encoding=self.options.get("encoding", None),
   1709     compression=self.options.get("compression", None),
   1710     memory_map=self.options.get("memory_map", False),
   1711     is_text=is_text,
   1712     errors=self.options.get("encoding_errors", "strict"),
   1713     storage_options=self.options.get("storage_options", None),
   1714 )
   1715 assert self.handles is not None
   1716 f = self.handles.handle

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:718, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    715     codecs.lookup_error(errors)
    717 # open URLs
--> 718 ioargs = _get_filepath_or_buffer(
    719     path_or_buf,
    720     encoding=encoding,
    721     compression=compression,
    722     mode=mode,
    723     storage_options=storage_options,
    724 )
    726 handle = ioargs.filepath_or_buffer
    727 handles: list[BaseBuffer]

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:418, in _get_filepath_or_buffer(filepath_or_buffer, encoding, compression, mode, storage_options)
    415     pass
    417 try:
--> 418     file_obj = fsspec.open(
    419         filepath_or_buffer, mode=fsspec_mode, **(storage_options or {})
    420     ).open()
    421 # GH 34626 Reads from Public Buckets without Credentials needs anon=Tru
    422 except tuple(err_types_to_retry_with_anon):

File /opt/conda/lib/python3.11/site-packages/fsspec/core.py:459, in open(urlpath, mode, compression, encoding, errors, protocol, newline, **kwargs)
    399 def open(
    400     urlpath,
    401     mode="rb",
   (…)
    407     **kwargs,
    408 ):
    409     """Given a path or paths, return one ``OpenFile`` object.
    410 
    411     Parameters
   (…)
    457       https://filesystem-spec.readthedocs.io/en/latest/api.html#other-known-implementations
    458     """
--> 459     out = open_files(
    460         urlpath=[urlpath],
    461         mode=mode,
    462         compression=compression,
    463         encoding=encoding,
    464         errors=errors,
    465         protocol=protocol,
    466         newline=newline,
    467         expand=False,
    468         **kwargs,
    469     )
    470     if not out:
    471         raise FileNotFoundError(urlpath)

File /opt/conda/lib/python3.11/site-packages/fsspec/core.py:283, in open_files(urlpath, mode, compression, encoding, errors, name_function, num, protocol, newline, auto_mkdir, expand, **kwargs)
    204 def open_files(
    205     urlpath,
    206     mode="rb",
   (…)
    216     **kwargs,
    217 ):
    218     """Given a path or paths, return a list of ``OpenFile`` objects.
    219 
    220     For writing, a str path must contain the "*" character, which will␣
  ↪be filled
   (…)
    281       https://filesystem-spec.readthedocs.io/en/latest/api.
  ↪html#other-known-implementations
    282     """
--> 283     fs, fs_token, paths = get_fs_token_paths(
    284         urlpath,
    285         mode,
    286         num=num,
    287         name_function=name_function,
    288         storage_options=kwargs,
    289         protocol=protocol,
    290         expand=expand,
    291     )
    292     if fs.protocol == "file":
    293         fs.auto_mkdir = auto_mkdir

File /opt/conda/lib/python3.11/site-packages/fsspec/core.py:623, in␣
  ↪get_fs_token_paths(urlpath, mode, num, name_function, storage_options,␣
  ↪protocol, expand)
    621 if protocol:
    622     storage_options["protocol"] = protocol
--> 623 chain = _un_chain(urlpath0, storage_options or {})
    624 inkwargs = {}
    625 # Reverse iterate the chain, creating a nested target_* structure

File /opt/conda/lib/python3.11/site-packages/fsspec/core.py:332, in␣
  ↪_un_chain(path, kwargs)
    330 for bit in reversed(bits):
    331     protocol = kwargs.pop("protocol", None) or split_protocol(bit)[0] o␣
  ↪"file"
--> 332     cls = get_filesystem_class(protocol)
    333     extra_kwargs = cls._get_kwargs_from_urls(bit)
    334     kws = kwargs.pop(protocol, {})

File /opt/conda/lib/python3.11/site-packages/fsspec/registry.py:233, in␣
  ↪get_filesystem_class(protocol)
    231 if protocol not in registry:
    232     if protocol not in known_implementations:
--> 233         raise ValueError(f"Protocol not known: {protocol}")
    234     bit = known_implementations[protocol]
    235     try:
```

```
ValueError: Protocol not known: httpse
```

[5]:
```python
import pandas as pd

url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.
 ↪csve"

df = pd.read_csv(url)
```

```
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
Cell In[5], line 5
      1 import pandas as pd
      3 url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/
 ↪titanic.csve"
----> 5 df = pd.read_csv(url)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:948,␣
 ↪in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, dtype, 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)
    935 kwds_defaults = _refine_defaults_read(
    936     dialect,
    937     delimiter,
   (…)
    944     dtype_backend=dtype_backend,
    945 )
    946 kwds.update(kwds_defaults)
--> 948 return _read(filepath_or_buffer, kwds)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:611,␣
 ↪in _read(filepath_or_buffer, kwds)
    608 _validate_names(kwds.get("names", None))
    610 # Create the parser.
--> 611 parser = TextFileReader(filepath_or_buffer, **kwds)
    613 if chunksize or iterator:
    614     return parser

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1448,
 ↪in TextFileReader.__init__(self, f, engine, **kwds)
   1445     self.options["has_index_names"] = kwds["has_index_names"]
   1447 self.handles: IOHandles | None = None
```

```
-> 1448 self._engine = self._make_engine(f, self.engine)

File /opt/conda/lib/python3.11/site-packages/pandas/io/parsers/readers.py:1705,␣
 ↪in TextFileReader._make_engine(self, f, engine)
   1703     if "b" not in mode:
   1704         mode += "b"
-> 1705 self.handles = get_handle(
   1706     f,
   1707     mode,
   1708     encoding=self.options.get("encoding", None),
   1709     compression=self.options.get("compression", None),
   1710     memory_map=self.options.get("memory_map", False),
   1711     is_text=is_text,
   1712     errors=self.options.get("encoding_errors", "strict"),
   1713     storage_options=self.options.get("storage_options", None),
   1714 )
   1715 assert self.handles is not None
   1716 f = self.handles.handle

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:718, in␣
 ↪get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,␣
 ↪errors, storage_options)
    715     codecs.lookup_error(errors)
    717 # open URLs
--> 718 ioargs = _get_filepath_or_buffer(
    719     path_or_buf,
    720     encoding=encoding,
    721     compression=compression,
    722     mode=mode,
    723     storage_options=storage_options,
    724 )
    726 handle = ioargs.filepath_or_buffer
    727 handles: list[BaseBuffer]

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:372, in␣
 ↪_get_filepath_or_buffer(filepath_or_buffer, encoding, compression, mode,␣
 ↪storage_options)
    370 # assuming storage_options is to be interpreted as headers
    371 req_info = urllib.request.Request(filepath_or_buffer,␣
 ↪headers=storage_options)
--> 372 with urlopen(req_info) as req:
    373     content_encoding = req.headers.get("Content-Encoding", None)
    374     if content_encoding == "gzip":
    375         # Override compression based on Content-Encoding header

File /opt/conda/lib/python3.11/site-packages/pandas/io/common.py:274, in␣
 ↪urlopen(*args, **kwargs)
    268 """
    269 Lazy-import wrapper for stdlib urlopen, as that imports a big chunk of
    270 the stdlib.
    271 """
    272 import urllib.request
--> 274 return urllib.request.urlopen(*args, **kwargs)

File /opt/conda/lib/python3.11/urllib/request.py:216, in urlopen(url, data,␣
 ↪timeout, cafile, capath, cadefault, context)
    214 else:
    215     opener = _opener
--> 216 return opener.open(url, data, timeout)

File /opt/conda/lib/python3.11/urllib/request.py:525, in OpenerDirector.
 ↪open(self, fullurl, data, timeout)
    523 for processor in self.process_response.get(protocol, []):
    524     meth = getattr(processor, meth_name)
--> 525     response = meth(req, response)
    527 return response

File /opt/conda/lib/python3.11/urllib/request.py:634, in HTTPErrorProcessor.
 ↪http_response(self, request, response)
    631 # According to RFC 2616, "2xx" code indicates that the client's
    632 # request was successfully received, understood, and accepted.
    633 if not (200 <= code < 300):
--> 634     response = self.parent.error(
    635         'http', request, response, code, msg, hdrs)
    637 return response

File /opt/conda/lib/python3.11/urllib/request.py:563, in OpenerDirector.
 ↪error(self, proto, *args)
    561 if http_err:
    562     args = (dict, 'default', 'http_error_default') + orig_args
--> 563     return self._call_chain(*args)

File /opt/conda/lib/python3.11/urllib/request.py:496, in OpenerDirector.
 ↪_call_chain(self, chain, kind, meth_name, *args)
    494 for handler in handlers:
    495     func = getattr(handler, meth_name)
--> 496     result = func(*args)
    497     if result is not None:
    498         return result

File /opt/conda/lib/python3.11/urllib/request.py:643, in HTTPDefaultErrorHandler.
 ↪http_error_default(self, req, fp, code, msg, hdrs)
    642 def http_error_default(self, req, fp, code, msg, hdrs):
--> 643     raise HTTPError(req.full_url, code, msg, hdrs, fp)
```

```
HTTPError: HTTP Error 404: Not Found
```

[6]:

```
import pandas as pd

url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.
 ↪csv"

df = pd.read_csv(url)
```

[7]:

```
DF.groupby("col1")["col2"].describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 DF.groupby("col1")["col2"].describe()

NameError: name 'DF' is not defined
```

[8]:

```
pd.read_csv(url
```

```
  Cell In[8], line 2

    ^
SyntaxError: incomplete input
```

[9]:

```
df.groupby("col1")["col2"].describile()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[9], line 1
----> 1 df.groupby("col1")["col2"].describile()

File /opt/conda/lib/python3.11/site-packages/pandas/core/frame.py:8869, in␣
 ↪DataFrame.groupby(self, by, axis, level, as_index, sort, group_keys, observed ␣
 ↪dropna)
   8866 if level is None and by is None:
   8867     raise TypeError("You have to supply one of 'by' and 'level'")
-> 8869 return DataFrameGroupBy(
   8870     obj=self,
   8871     keys=by,
   8872     axis=axis,
   8873     level=level,
   8874     as_index=as_index,
   8875     sort=sort,
   8876     group_keys=group_keys,
   8877     observed=observed,
   8878     dropna=dropna,
   8879 )

File /opt/conda/lib/python3.11/site-packages/pandas/core/groupby/groupby.py:
  ↪1278, in GroupBy.__init__(self, obj, keys, axis, level, grouper, exclusions,␣
  ↪selection, as_index, sort, group_keys, observed, dropna)
   1275 self.dropna = dropna
   1277 if grouper is None:
-> 1278     grouper, exclusions, obj = get_grouper(
   1279         obj,
   1280         keys,
   1281         axis=axis,
   1282         level=level,
   1283         sort=sort,
   1284         observed=False if observed is lib.no_default else observed,
   1285         dropna=self.dropna,
   1286     )
   1288 if observed is lib.no_default:
   1289     if any(ping._passed_categorical for ping in grouper.groupings):

File /opt/conda/lib/python3.11/site-packages/pandas/core/groupby/grouper.py:
  ↪1009, in get_grouper(obj, key, axis, level, sort, observed, validate, dropna)
   1007         in_axis, level, gpr = False, gpr, None
   1008     else:
-> 1009         raise KeyError(gpr)
   1010 elif isinstance(gpr, Grouper) and gpr.key is not None:
   1011     # Add key to exclusions
   1012     exclusions.add(gpr.key)

KeyError: 'col1'
```

```
[10]: df.groupby("sibsp")["age"].describe()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[10], line 1
----> 1 df.groupby("sibsp")["age"].describe()

File /opt/conda/lib/python3.11/site-packages/pandas/core/groupby/groupby.py:
  ↪1312, in GroupBy.__getattr__(self, attr)
   1309 if attr in self.obj:
   1310     return self[attr]
-> 1312 raise AttributeError(
   1313     f"'{type(self).__name__}' object has no attribute '{attr}'"
   1314 )

AttributeError: 'SeriesGroupBy' object has no attribute 'describe'
```

[11]:
```
df.groupby("sibsp")["age"].describe()
```

[11]:
```
       count       mean        std   min    25%   50%    75%   max
sibsp
0      471.0  31.397558  13.647767  0.42  22.00  29.0  39.00  80.0
1      183.0  30.089727  14.645033  0.67  20.00  30.0  39.00  70.0
2       25.0  22.620000  14.679230  0.75  16.00  23.0  28.00  53.0
3       12.0  13.916667  11.317391  2.00   3.75   9.5  23.25  33.0
4       18.0   7.055556   4.880601  1.00   3.25   6.5   9.00  17.0
5        5.0  10.200000   5.805170  1.00   9.00  11.0  14.00  16.0
8        0.0        NaN        NaN   NaN    NaN   NaN    NaN   NaN
```

[12]:
```
df.groupby(sex)["age"].describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 df.groupby(sex)["age"].describe()

NameError: name 'sex' is not defined
```

[13]:
```
df.groupby("sex")[age].describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[13], line 1
----> 1 df.groupby("sex")[age].describe()

NameError: name 'age' is not defined
```

[14]:
```
df.groupby("Sex")["age"].describe()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[14], line 1
----> 1 df.groupby("Sex")["age"].describe()

File /opt/conda/lib/python3.11/site-packages/pandas/core/frame.py:8869, in␣
  ↪DataFrame.groupby(self, by, axis, level, as_index, sort, group_keys, observed␣
  ↪dropna)
   8866 if level is None and by is None:
```

```
   8867     raise TypeError("You have to supply one of 'by' and 'level'")
-> 8869 return DataFrameGroupBy(
   8870     obj=self,
   8871     keys=by,
   8872     axis=axis,
   8873     level=level,
   8874     as_index=as_index,
   8875     sort=sort,
   8876     group_keys=group_keys,
   8877     observed=observed,
   8878     dropna=dropna,
   8879 )

File /opt/conda/lib/python3.11/site-packages/pandas/core/groupby/groupby.py:
  ↪1278, in GroupBy.__init__(self, obj, keys, axis, level, grouper, exclusions,␣
  ↪selection, as_index, sort, group_keys, observed, dropna)
   1275 self.dropna = dropna
   1277 if grouper is None:
-> 1278     grouper, exclusions, obj = get_grouper(
   1279         obj,
   1280         keys,
   1281         axis=axis,
   1282         level=level,
   1283         sort=sort,
   1284         observed=False if observed is lib.no_default else observed,
   1285         dropna=self.dropna,
   1286     )
   1288 if observed is lib.no_default:
   1289     if any(ping._passed_categorical for ping in grouper.groupings):

File /opt/conda/lib/python3.11/site-packages/pandas/core/groupby/grouper.py:
  ↪1009, in get_grouper(obj, key, axis, level, sort, observed, validate, dropna)
   1007         in_axis, level, gpr = False, gpr, None
   1008     else:
-> 1009         raise KeyError(gpr)
   1010 elif isinstance(gpr, Grouper) and gpr.key is not None:
   1011     # Add key to exclusions
   1012     exclusions.add(gpr.key)

KeyError: 'Sex'
```

[15]:

```
df.groupby("sex")["age"].describe()
```

[15]:

```
        count       mean        std   min   25%   50%   75%   max
sex
female  261.0  27.915709  14.110146  0.75  18.0  27.0  37.0  63.0
male    453.0  30.726645  14.678201  0.42  21.0  29.0  39.0  80.0
```

[ ]:

```
fig.show(renderer="png")
```

[ ]:

```
```