Experiment 4

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv('https://gist.githubusercontent.com/netj/8836201/raw/6f9306ad21398ea43c
```

In [4]:
```
df.head(100)
```

Out[4]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
95	5.7	3.0	4.2	1.2	Versicolor
96	5.7	2.9	4.2	1.3	Versicolor
97	6.2	2.9	4.3	1.3	Versicolor
98	5.1	2.5	3.0	1.1	Versicolor
99	5.7	2.8	4.1	1.3	Versicolor

100 rows × 5 columns

In [5]:
```
df.tail()
```

Out[5]:

	sepal.length	sepal.width	petal.length	petal.width	variety
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

In [6]:
```
df.shape
```

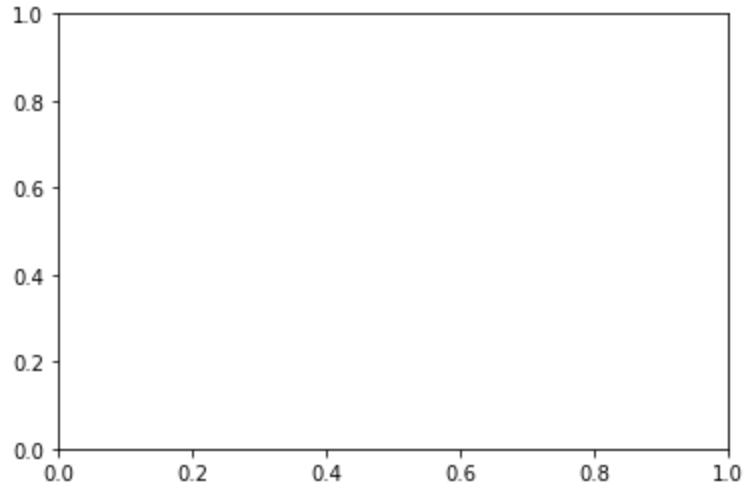
Out[6]:
```
(150, 5)
```

In [7]:
```
setosa = df[df['variety']=='Setosa']
versicolor = df[df['variety']=='Versicolor']
virginica = df[df['variety']=='Virginica']
```

In [8]:
```
plt.plot(setosa['sepal.length'],0)
```

```
---------------------------------------------------------------------------
```

```
ValueError                                Traceback (most recent call last)
Input In [8], in <cell line: 1>()
----> 1 plt.plot(setosa['sepal.length'],0)

File ~/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packages/matplotlib/pyplot.py:2728, in plot(scalex, scaley, data, *args, **kwargs)
   2726 @_copy_docstring_and_deprecators(Axes.plot)
   2727 def plot(*args, scalex=True, scaley=True, data=None, **kwargs):
-> 2728     return gca().plot(
   2729         *args, scalex=scalex, scaley=scaley,
   2730         **({"data": data} if data is not None else {}), **kwargs)

File ~/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packages/matplotlib/axes/_axes.py:1662, in Axes.plot(self, scalex, scaley, data, *args, **kwargs)
   1419 """
   1420 Plot y versus x as lines and/or markers.
   1421
   (...)
   1659 (``'green'``) or hex strings (``'#008000'``).
   1660 """
   1661 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1662 lines = [*self._get_lines(*args, data=data, **kwargs)]
   1663 for line in lines:
   1664     self.add_line(line)

File ~/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packages/matplotlib/axes/_base.py:311, in _process_plot_var_args.__call__(self, data, *args, **kwargs)
    309     this += args[0],
    310     args = args[1:]
--> 311 yield from self._plot_args(
    312     this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey)

File ~/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packages/matplotlib/axes/_base.py:504, in _process_plot_var_args._plot_args(self, tup, kwargs, return_kwargs, ambiguous_fmt_datakey)
    501     self.axes.yaxis.update_units(y)
    503 if x.shape[0] != y.shape[0]:
--> 504     raise ValueError(f"x and y must have same first dimension, but "
    505                      f"have shapes {x.shape} and {y.shape}")
    506 if x.ndim > 2 or y.ndim > 2:
    507     raise ValueError(f"x and y can be no greater than 2D, but have "
    508                      f"shapes {x.shape} and {y.shape}")

ValueError: x and y must have same first dimension, but have shapes (50,) and (1,)
```

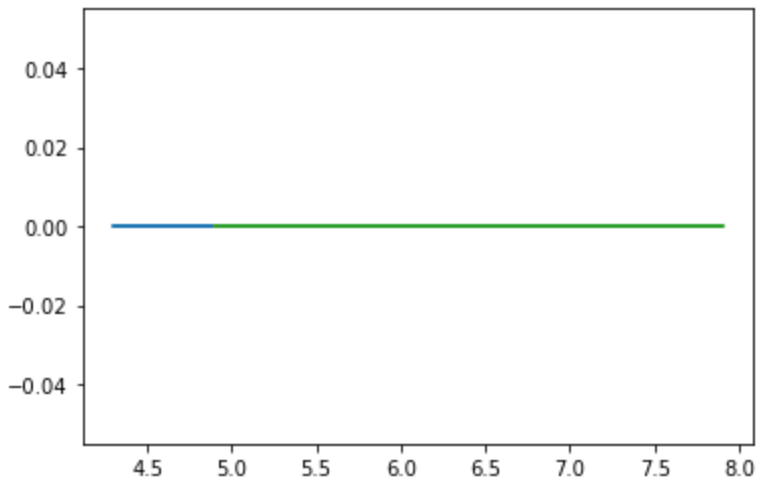


In [ ]:
```
plt.plot(setosa['sepal.length'],np.zeros_like(setosa['sepal.length']))
```

In [9]:
```
plt.plot(setosa['sepal.length'],np.zeros_like(setosa['sepal.length']))
plt.plot(versicolor['sepal.length'],np.zeros_like(versicolor['sepal.length']))
```

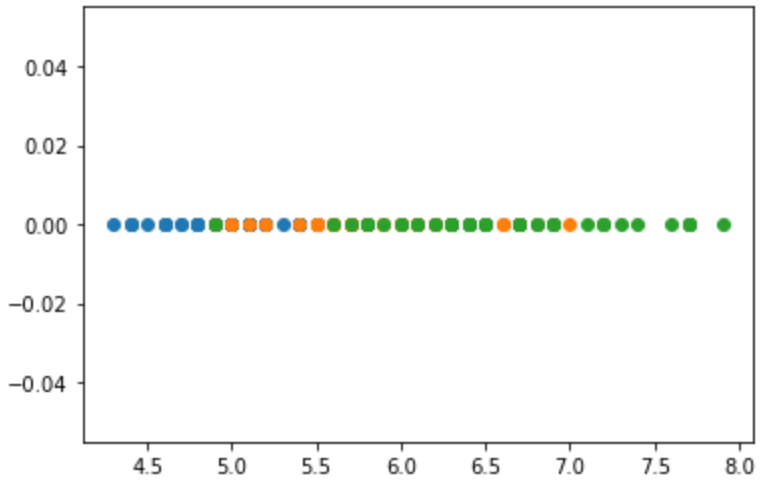
```
plt.plot(virginica['sepal.length'],np.zeros_like(virginica['sepal.length']))
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x7ff71065bf10>]
```



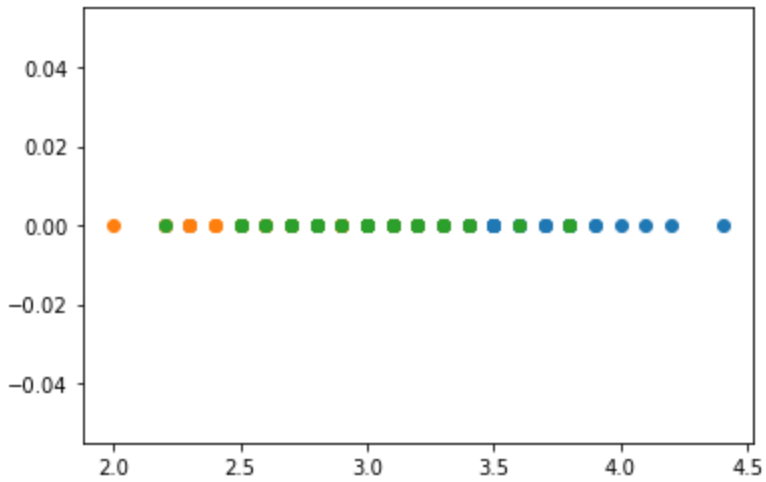
In [10]:
```
plt.plot(setosa['sepal.length'],np.zeros_like(setosa['sepal.length']), 'o')
plt.plot(versicolor['sepal.length'],np.zeros_like(versicolor['sepal.length']),'o')
plt.plot(virginica['sepal.length'],np.zeros_like(virginica['sepal.length']),'o')
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x7ff6e23f2be0>]
```



In [11]:
```
plt.plot(setosa['sepal.width'],np.zeros_like(setosa['sepal.width']), 'o')
plt.plot(versicolor['sepal.width'],np.zeros_like(versicolor['sepal.width']),'o')
plt.plot(virginica['sepal.width'],np.zeros_like(virginica['sepal.width']),'o')
```

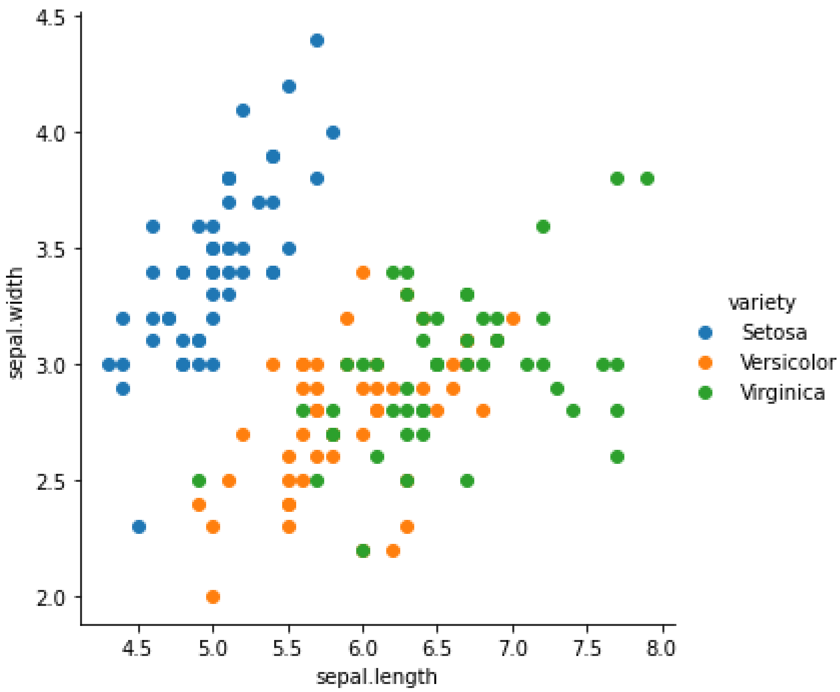
Out[11]:
```
[<matplotlib.lines.Line2D at 0x7ff7045d2190>]
```



In [12]:
```
sns.FacetGrid(df, hue = 'variety', height=5).map(plt.scatter, "sepal.length","sepal.widt
```

```
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
```

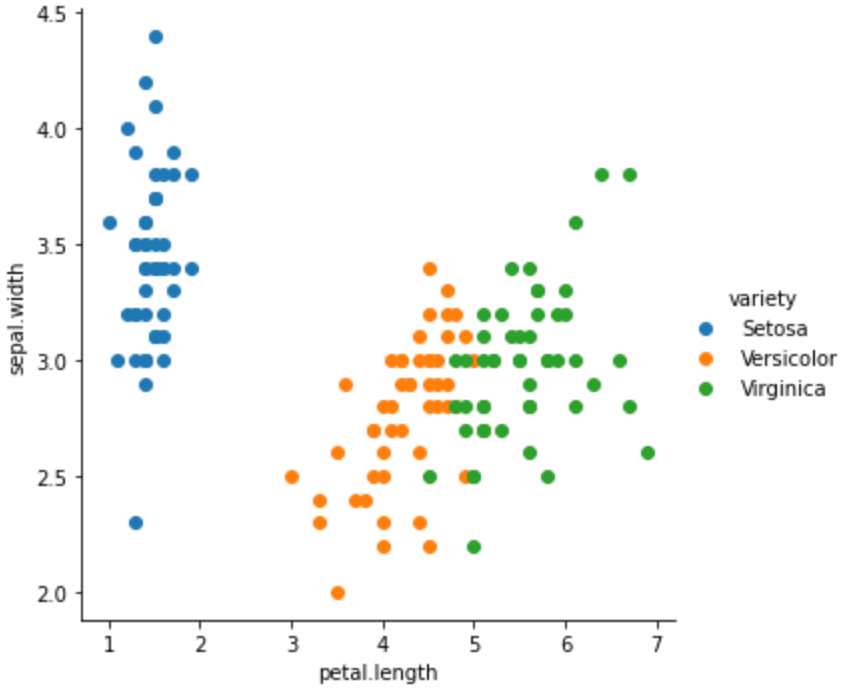
Out[12]: <seaborn.axisgrid.FacetGrid at 0x7ff704606100>



In [13]:
```
sns.FacetGrid(df, hue = 'variety', height=5).map(plt.scatter, "petal.length","sepal.widt
```

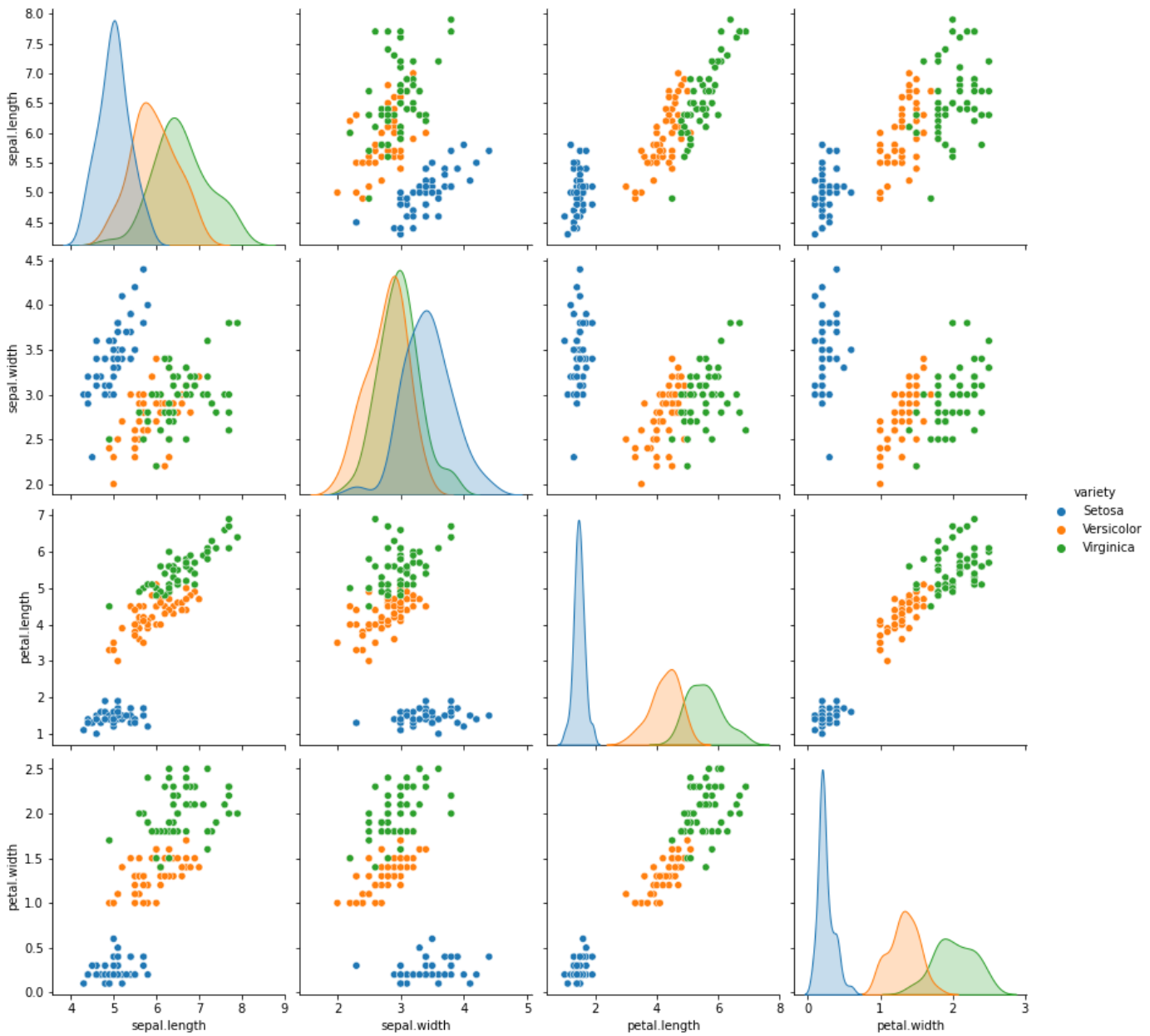
```
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
/Users/chiragchan/opt/anaconda3/envs/data_visualisation_python/lib/python3.9/site-packag
es/seaborn/axisgrid.py:745: FutureWarning: iteritems is deprecated and will be removed i
n a future version. Use .items instead.
  plot_args = [v for k, v in plot_data.iteritems()]
```

Out[13]: <seaborn.axisgrid.FacetGrid at 0x7ff704d7f3d0>



In [14]:
```
sns.pairplot(df, hue='variety', height=3)
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x7ff6f17f7c70>
```


sepal.length
sepal.width
petal.length
petal.width
variety
Setosa
Versicolor
Virginica

In [ ]:

In [ ]: