Untitled

January 17, 2023

[14]:

```python
import seaborn as sns
import matplotlib.pyplot as pl
%matplotlib inline
```

[6]:

```python
df=sns.load_dataset("iris")
print(df)
df.species.unique()
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
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
..            …            …             …            …        …
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

[150 rows x 5 columns]
```

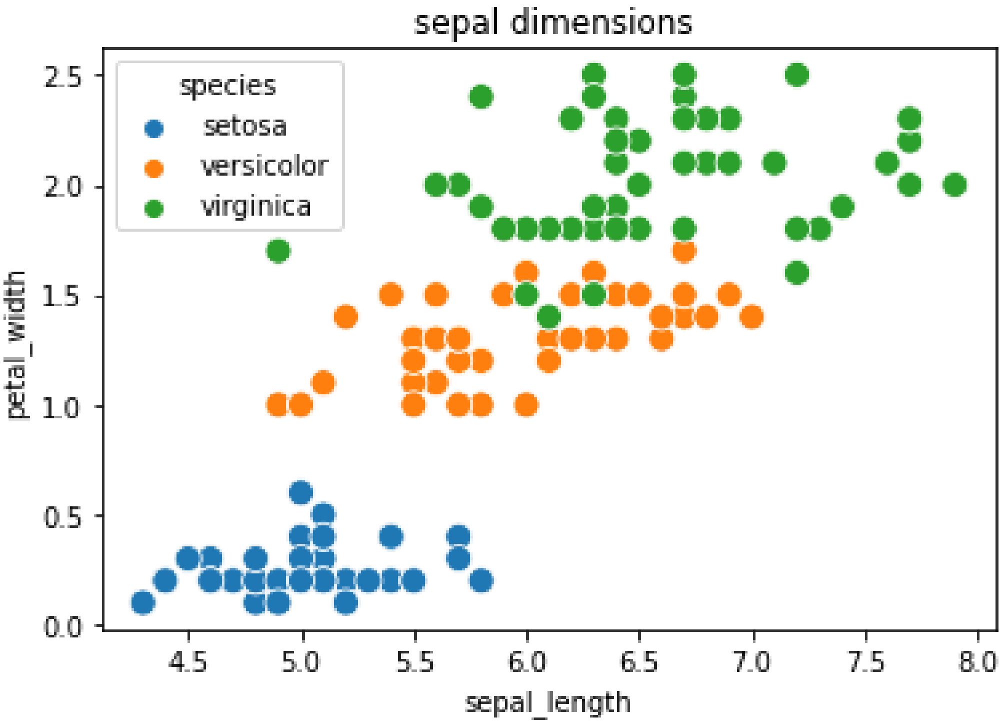
[6]:

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

[18]:

```python
pl.title(" sepal dimensions")
sns.scatterplot(df.sepal_length,df.petal_width,
hue=df.species,
s=100)
pl.figure(figsize=[12,8]);
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



```
<Figure size 864x576 with 0 Axes>
```

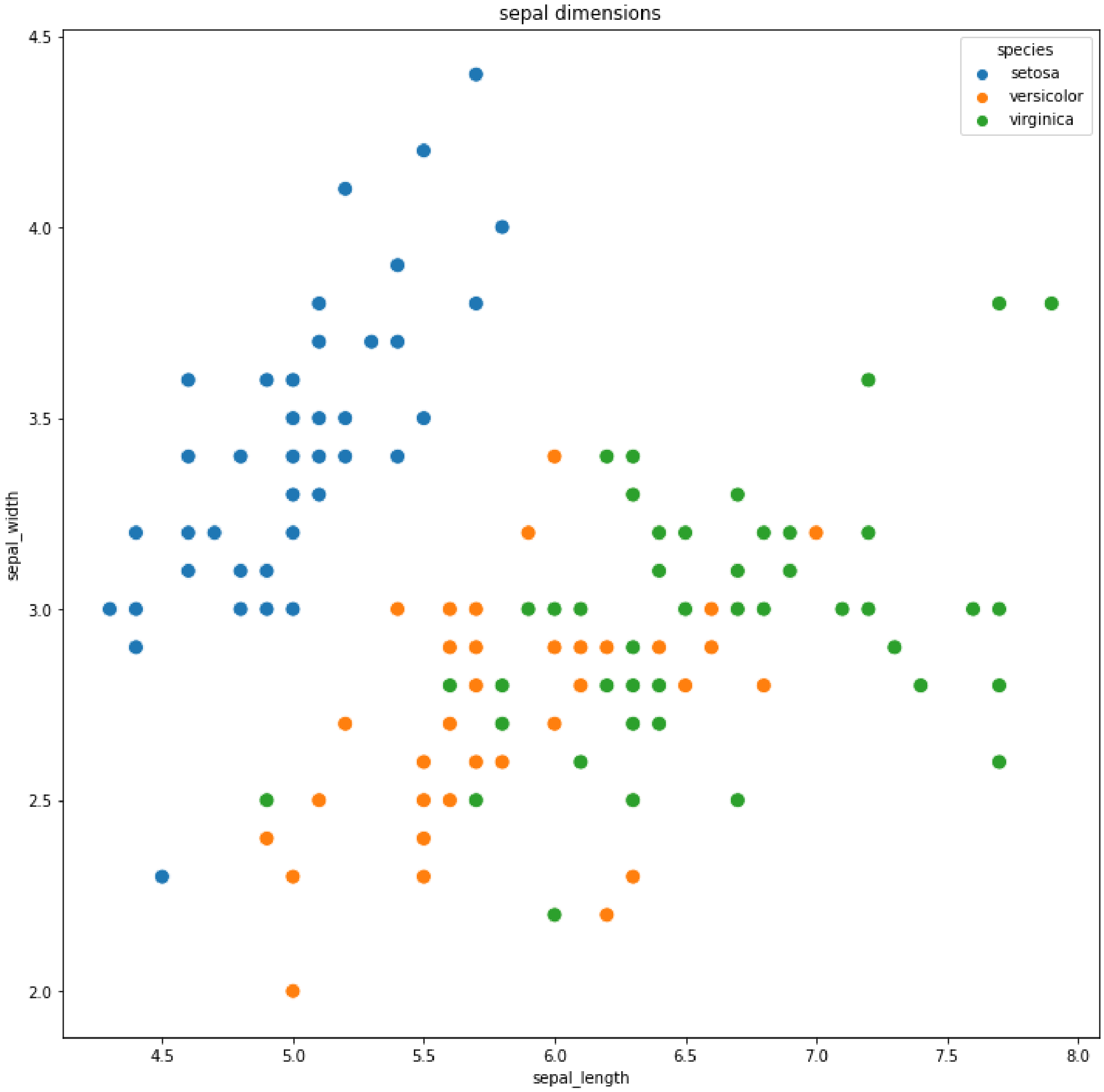
[22]:

```
pl.figure(figsize=[12,12])
pl.title("sepal dimensions")
sns.scatterplot("sepal_length","sepal_width",hue="species",s=100,data=df)
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

[22]:

```
<AxesSubplot:title={'center':'sepal dimensions'}, xlabel='sepal_length',
ylabel='sepal_width'>
```



[23]:
```
df.sepal_width
```

[23]:
```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      …
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
```

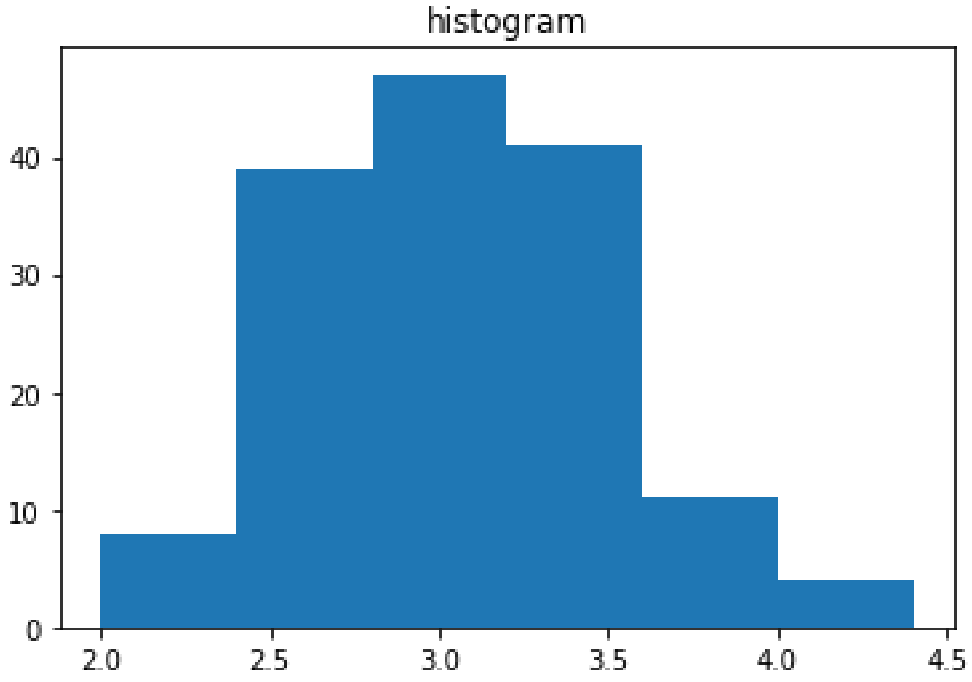
```
Name: sepal_width, Length: 150, dtype: float64
```

[24]:
```
df.sepal_width.describe()
```

[24]:
```
count    150.000000
mean       3.057333
std        0.435866
min        2.000000
25%        2.800000
50%        3.000000
75%        3.300000
max        4.400000
Name: sepal_width, dtype: float64
```

[29]:
```
pl.title("histogram")
pl.hist(df.sepal_width,bins=6)
```

[29]:
```
(array([ 8., 39., 47., 41., 11.,  4.]),
 array([2. , 2.4, 2.8, 3.2, 3.6, 4. , 4.4]),
 <BarContainer object of 6 artists>)
```

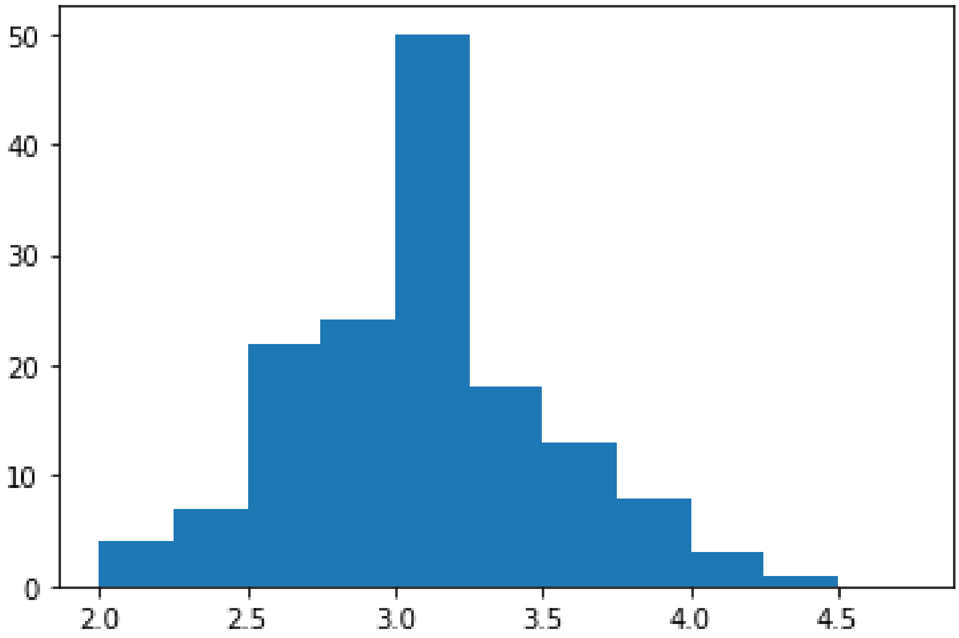


[30]:
```
import numpy as np
np.arange(2,5,0.25)
```

```
[30]: array([2.  , 2.25, 2.5 , 2.75, 3.  , 3.25, 3.5 , 3.75, 4.  , 4.25, 4.5 ,
             4.75])
```

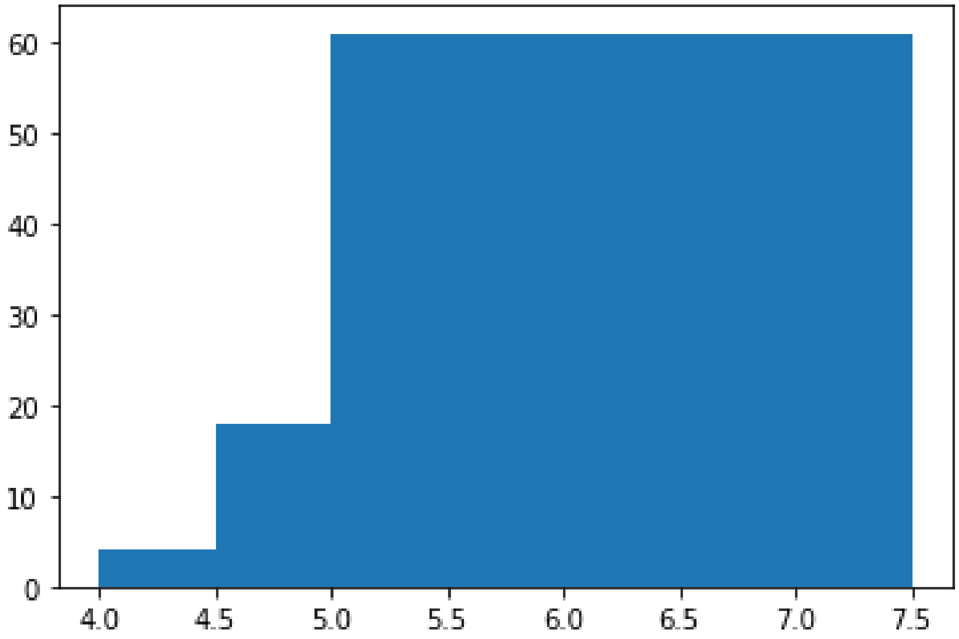
```
[31]: pl.hist(df.sepal_width,bins=np.arange(2,5,0.25))
```

```
[31]: (array([ 4.,  7., 22., 24., 50., 18., 13.,  8.,  3.,  1.,  0.]),
       array([2.  , 2.25, 2.5 , 2.75, 3.  , 3.25, 3.5 , 3.75, 4.  , 4.25, 4.5 ,
              4.75]),
       <BarContainer object of 11 artists>)
```



```
[35]: pl.hist(df.sepal_length,bins=(4,4.5,5,6,7.5))
```

```
[35]: (array([ 4., 18., 61., 61.]),
       array([4. , 4.5, 5. , 6. , 7.5]),
       <BarContainer object of 4 artists>)
```



[38]:

```
county=(
"mombasa",
"kwale",
"Kilifi",
"Tanariver",
"Lamu",
"Taita/Taveta",
"Garissa",
"Wajir",
"Mandera",
"Marsarbit",
"Isiolo",
"Meru",
"Tharaka Nithi",
"Embu",
"Kitui",
"Machakos",
"Makueni",
"Nyandarua",
"Nyeri",
"Kirinyaga",
"Muranga",
"Kiambu",
"Turkana",
"West Pokot",
"Samburu",
"Trans_Nzoia",
"Uasin Gishu",
"Elgeyo Marakwet",
"Nandi",
"Baringo",
"Laikipia",
"Nakuru",
"Narok",
"Kajiado",
"Kericho",
"Bomet",
"Kakamega",
"Vihiga",
"Bungoma",
"Busia",
"Siaya",
"Kisumu",
"Homa bay",
"Migori",
"Kisii",
"Nyamira",
"Nairobi city")
print(county)
```

```
('mombasa', 'kwale', 'Kilifi', 'Tanariver', 'Lamu', 'Taita/Taveta', 'Garissa',
'Wajir', 'Mandera', 'Marsarbit', 'Isiolo', 'Meru', 'Tharaka Nithi', 'Embu',
'Kitui', 'Machakos', 'Makueni', 'Nyandarua', 'Nyeri', 'Kirinyaga', 'Muranga',
'Kiambu', 'Turkana', 'West Pokot', 'Samburu', 'Trans_Nzoia', 'Uasin Gishu',
'Elgeyo Marakwet', 'Nandi', 'Baringo', 'Laikipia', 'Nakuru', 'Narok', 'Kajiado',
'Kericho', 'Bomet', 'Kakamega', 'Vihiga', 'Bungoma', 'Busia', 'Siaya', 'Kisumu',
'Homa bay', 'Migori', 'Kisii', 'Nyamira', 'Nairobi city')
```

[40]:

```
male=(
610257,
425121,
704089,
158550,
76103,
173337,
458975,
415374,
434976,
243548,
139510,
767698,
193764,
304208,
549003,
710707,
489691,
315022,
374228,
302011,
523940,
1187146,
478087,
307013,
156774,
489107,
580269,
227317,
441259,
336322,
259440,
1077272,
579042,
557098,
450741,
434287,
987133,
283678,
812146,
426252,
471669,
560942,
539560,
536187,
605784,
290907,
2192452)
print(male)
```

```
(610257, 425121, 704089, 158550, 76103, 173337, 458975, 415374, 434976, 243548,
139510, 767698, 193764, 304208, 549003, 710707, 489691, 315022, 374228, 302011,
523940, 1187146, 478087, 307013, 156774, 489107, 580269, 227317, 441259, 336322,
259440, 1077272, 579042, 557098, 450741, 434287, 987133, 283678, 812146, 426252,
471669, 560942, 539560, 536187, 605784, 290907, 2192452)
```

[42]:

```
intersex=[
30,
18,
25,
2,
4,
7,
34,
49,
37,
18,
9,
41,
7,
24,
33,
34,
20,
20,
31,
31,
31,
135,
21,
15,
7,
28,
28,
12,
22,
13,
18,
95,
26,
38,
28,
23,
40,
12,
35,
28,
18,
23,
23,
35,
38,
13,
245]
```

[43]:
```
print(intersex)
```

```
[30, 18, 25, 2, 4, 7, 34, 49, 37, 18, 9, 41, 7, 24, 33, 34, 20, 20, 31, 31, 31,
135, 21, 15, 7, 28, 28, 12, 22, 13, 18, 95, 26, 38, 28, 23, 40, 12, 35, 28, 18,
23, 23, 35, 38, 13, 245]
```

[44]:

```
female=[
598046,
441681,
749673,
157391,
67813,
167327,
382344,
365840,
432444,
216219,
128483,
777975,
199406,
304367,
587151,
711191,
497942,
323247,
384845,
308369,
532669,
1230454,
448868,
314213,
153546,
501206,
582889,
227151,
444430,
330428,
259102,
1084835,
578805,
560704,
451008,
441379,
970406,
306323,
858389,
467401,
521496,
594609,
592367,
580214,
661038,
314656,
2204376]
print(female)
```

```
[598046, 441681, 749673, 157391, 67813, 167327, 382344, 365840, 432444, 216219,
128483, 777975, 199406, 304367, 587151, 711191, 497942, 323247, 384845, 308369,
532669, 1230454, 448868, 314213, 153546, 501206, 582889, 227151, 444430, 330428,
259102, 1084835, 578805, 560704, 451008, 441379, 970406, 306323, 858389, 467401,
521496, 594609, 592367, 580214, 661038, 314656, 2204376]
```

[45]:

```
total=[
1208333,
866820,
1453787,
315943,
143920,
340671,
841353,
781263,
867457,
459785,
268002,
1545714,
393177,
608599,
1136187,
1421932,
987653,
638289,
759104,
610411,
1056640,
2417735,
926976,
621241,
310327,
990341,
1163186,
454480,
885711,
666763,
518560,
2162202,
1157873,
1117840,
901777,
875689,
1957579,
590013,
1670570,
893681,
993183,
1155574,
1131950,
1116436,
1266860,
605576,
4397073]
print(female)
```

```
[598046, 441681, 749673, 157391, 67813, 167327, 382344, 365840, 432444, 216219,
128483, 777975, 199406, 304367, 587151, 711191, 497942, 323247, 384845, 308369,
532669, 1230454, 448868, 314213, 153546, 501206, 582889, 227151, 444430, 330428,
259102, 1084835, 578805, 560704, 451008, 441379, 970406, 306323, 858389, 467401,
521496, 594609, 592367, 580214, 661038, 314656, 2204376]
```

```
[ ]:
```