Assignment 2 Ben Gowaski

In [72]:
```
import pandas as pd
fileURL = "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
df = pd.read_csv(fileURL, names=["Sepal Length", "Sepal Width","Petal Length", "Petal Width", "Name"], I
```

1.1 Summary Statistics

Print the first 5 elements of your DataFrame using the command head(). How many features are there and what are their types (e.g., numeric, nominal)?

Compute and display summary statistics for each feature available in the dataset. These must include the minimum value, maximum value, mean, range, standard deviation, variance, count, and 25:50:75% percentiles.

In [73]:
```
df.head()
```

Out[73]:

	Sepal Length	Sepal Width	Petal Length	Petal Width	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [74]:
```
features = len(df.columns)
features
```

Out[74]: 5

In [75]:
```
types = df.dtypes
types
```

Out[75]:
```
Sepal Length    float64
Sepal Width     float64
Petal Length    float64
Petal Width     float64
Name             object
dtype: object
```

In [76]:
```
sum_stats = df.describe()
variance = df.var()
sum_stats
```

Out[76]:

	Sepal Length	Sepal Width	Petal Length	Petal Width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [77]:
```
variance
```

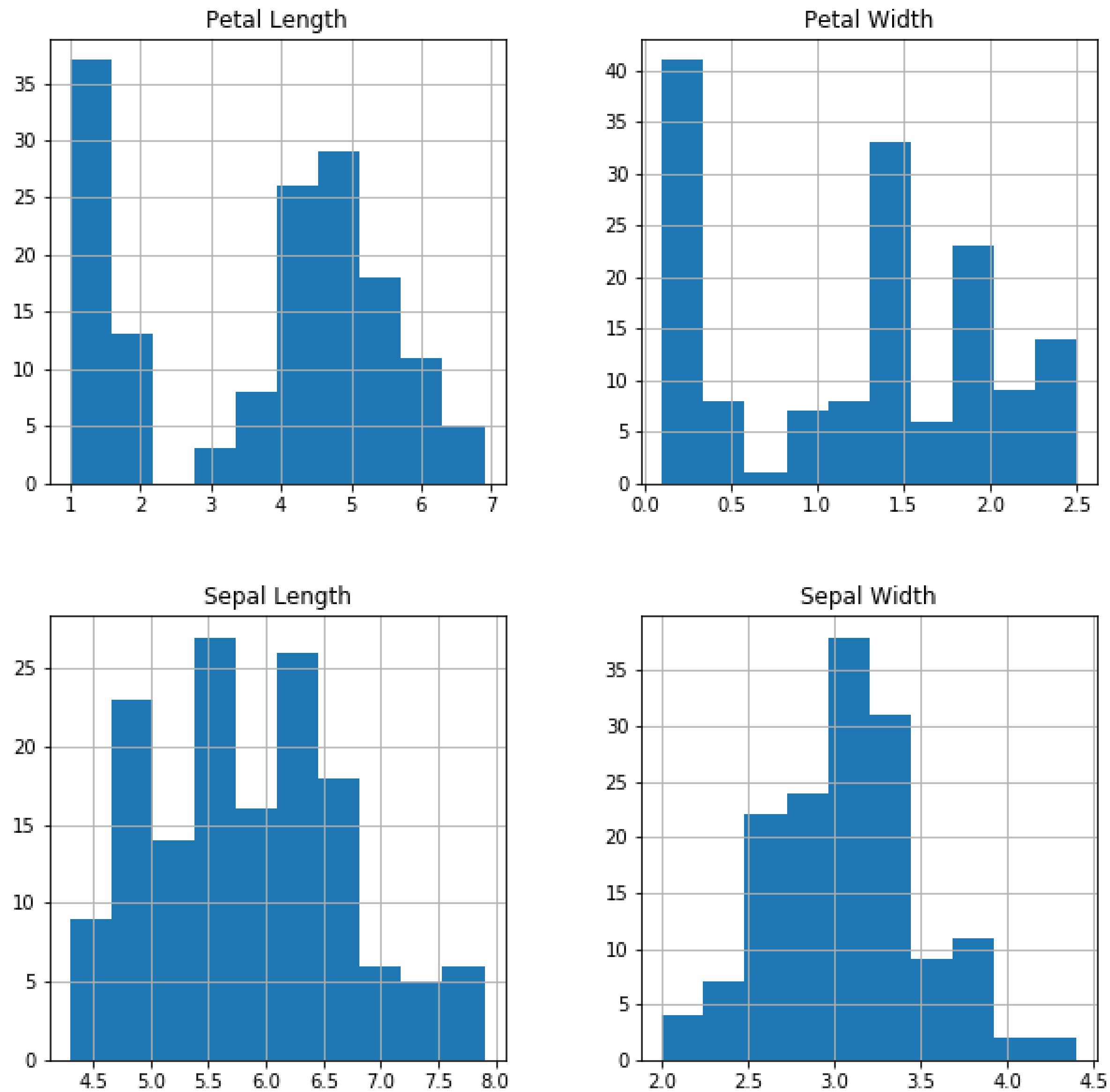
Out[77]:
```
Sepal Length    0.685694
Sepal Width     0.188004
Petal Length    3.113179
Petal Width     0.582414
dtype: float64
```

1.2 Data Visualization

Histograms: To illustrate the feature distributions, create a histogram for each feature in the dataset. You may plot each histogram individually or combine them all into a single plot. When generating histograms for this assignment, use the default number of bins. Recall that a histogram provides a graphical representation of the distribution of the data.

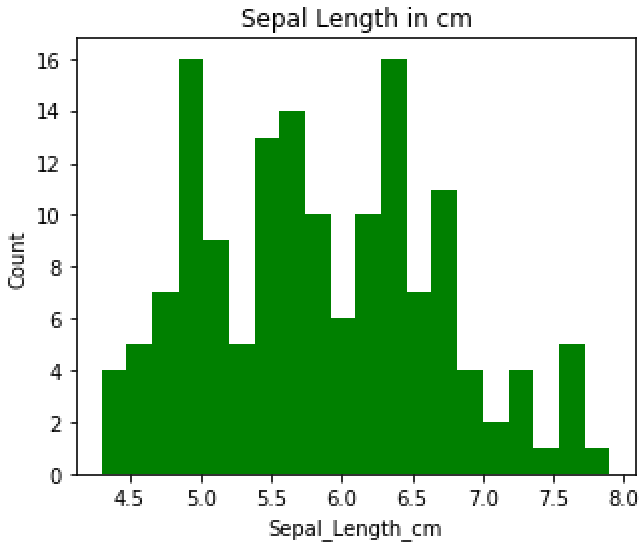
Box Plots: To further assess the data, create a boxplot for each feature in the dataset. All of the boxplots will be combined into a single plot. Recall that a boxplot provides a graphical representation of the location and variation of the data through their quartiles; they are especially useful for comparing distributions and identifying outliers.

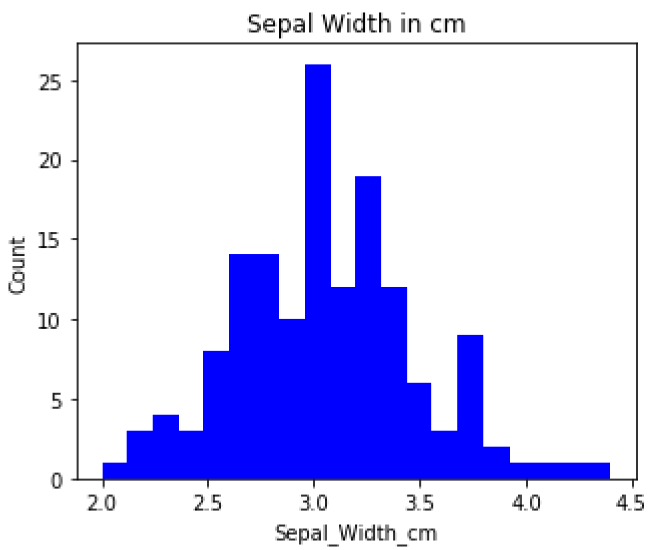
In [78]:
```
feature_hist = df.hist(figsize=(10,10))
```

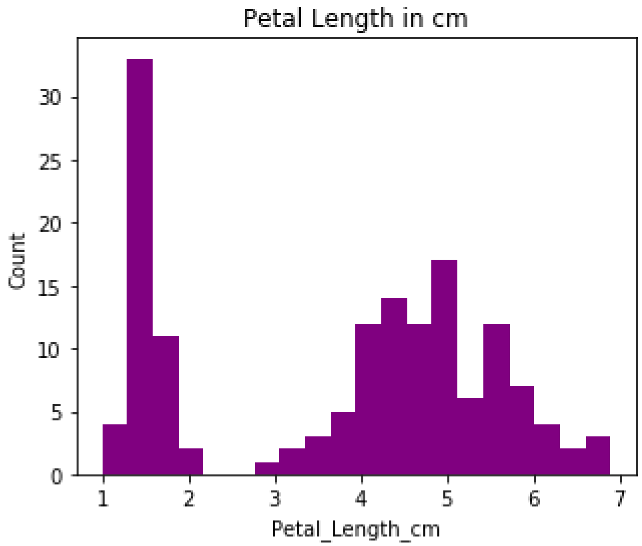


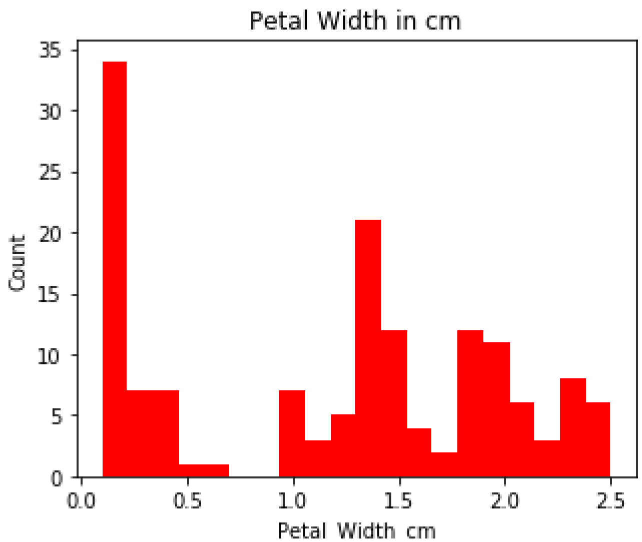
In [101]:
```
#Try using matplotlib to get some prettier histograms!
import matplotlib.pyplot as plt

#Sepal Length
plt.figure(figsize = (5, 4))
x = df["Sepal Length"]
plt.hist(x, bins = 20, color = "green")
plt.title("Sepal Length in cm")
plt.xlabel("Sepal_Length_cm")
plt.ylabel("Count")
#Sepal Width
plt.figure(figsize = (5, 4))
x = df["Sepal Width"]
plt.hist(x, bins = 20, color = "Blue")
plt.title("Sepal Width in cm")
plt.xlabel("Sepal_Width_cm")
plt.ylabel("Count")
#Petal Length
plt.figure(figsize = (5, 4))
x = df["Petal Length"]
plt.hist(x, bins = 20, color = "purple")
plt.title("Petal Length in cm")
plt.xlabel("Petal_Length_cm")
plt.ylabel("Count")
#Petal Width
plt.figure(figsize = (5, 4))
x = df["Petal Width"]
plt.hist(x, bins = 20, color = "red")
plt.title("Petal Width in cm")
plt.xlabel("Petal_Width_cm")
plt.ylabel("Count")
```

Out[101]:
```
Text(0, 0.5, 'Count')
```

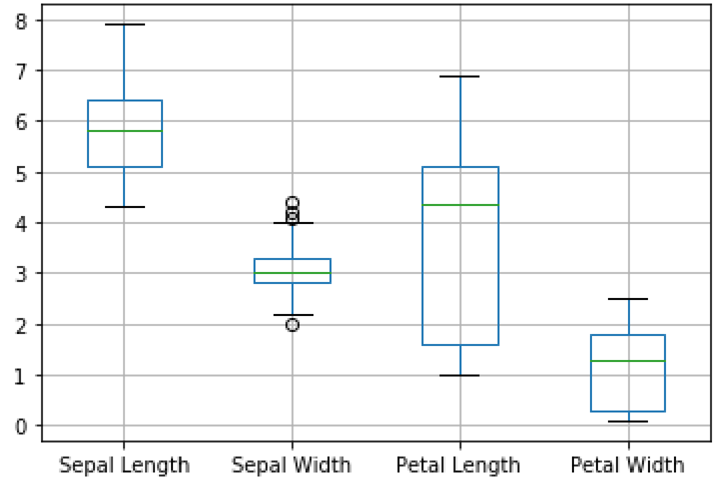

Sepal Length in cm
Count
Sepal_Length_cm
0
2
4
6
8
10
12
14
16
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


Sepal Width in cm
Count
Sepal_Width_cm
0
5
10
15
20
25
2.0
2.5
3.0
3.5
4.0
4.5


Petal Length in cm
Count
Petal_Length_cm
0
5
10
15
20
25
30
1
2
3
4
5
6
7


Petal Width in cm
Count
Petal_Width_cm
0
5
10
15
20
25
30
35
0.0
0.5
1.0
1.5
2.0
2.5

In [79]:
```
feature_box = df.boxplot()
```



2 Pen-Based Handwritten Digits Dataset [35 points]

Repeat the same process described in Part 1, but this time load THIS DATASET(http://archive.ics.uci.edu/ml/machinelearning-databases/pendigits/pendigits.tra) (http://archive.ics.uci.edu/ml/machinelearning-databases/pendigits/pendigits.tra)). Note that the Digits Dataset is much larger than the Iris dataset, both with respect to the number of instances and the number of features.

A description of this dataset can be found here(http://archive.ics.uci.edu/ml/datasets/PenBased+Recognition+of+Handwritten+Digits) (http://archive.ics.uci.edu/ml/datasets/PenBased+Recognition+of+Handwritten+Digits)).

In [80]:
```
pendigits = "http://archive.ics.uci.edu/ml/machine-learning-databases/pendigits/pendigits.tra"
df1 = pd.read_csv(pendigits, header=None)
```

In [81]:
```
df1.head()
```

Out[81]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	47	100	27	81	57	37	26	0	0	23	56	53	100	90	40	98	8
1	0	89	27	100	42	75	29	45	15	15	37	0	69	2	100	6	2
2	0	57	31	68	72	90	100	100	76	75	50	51	28	25	16	0	1
3	0	100	7	92	5	68	19	45	86	34	100	45	74	23	67	0	4
4	0	67	49	83	100	100	81	80	60	60	40	40	33	20	47	0	1

In [82]:
```
cols = len(df1.columns)
cols
```

Out[82]: 17

In [83]:
```
dtypes = df1.dtypes
dtypes
```

Out[83]:
```
0      int64
1      int64
2      int64
3      int64
4      int64
5      int64
6      int64
7      int64
8      int64
9      int64
10     int64
11     int64
12     int64
13     int64
14     int64
15     int64
16     int64
dtype: object
```

In [84]:
```
df1_stats = df1.describe()
df1_var = df1.var()
df1_stats
```

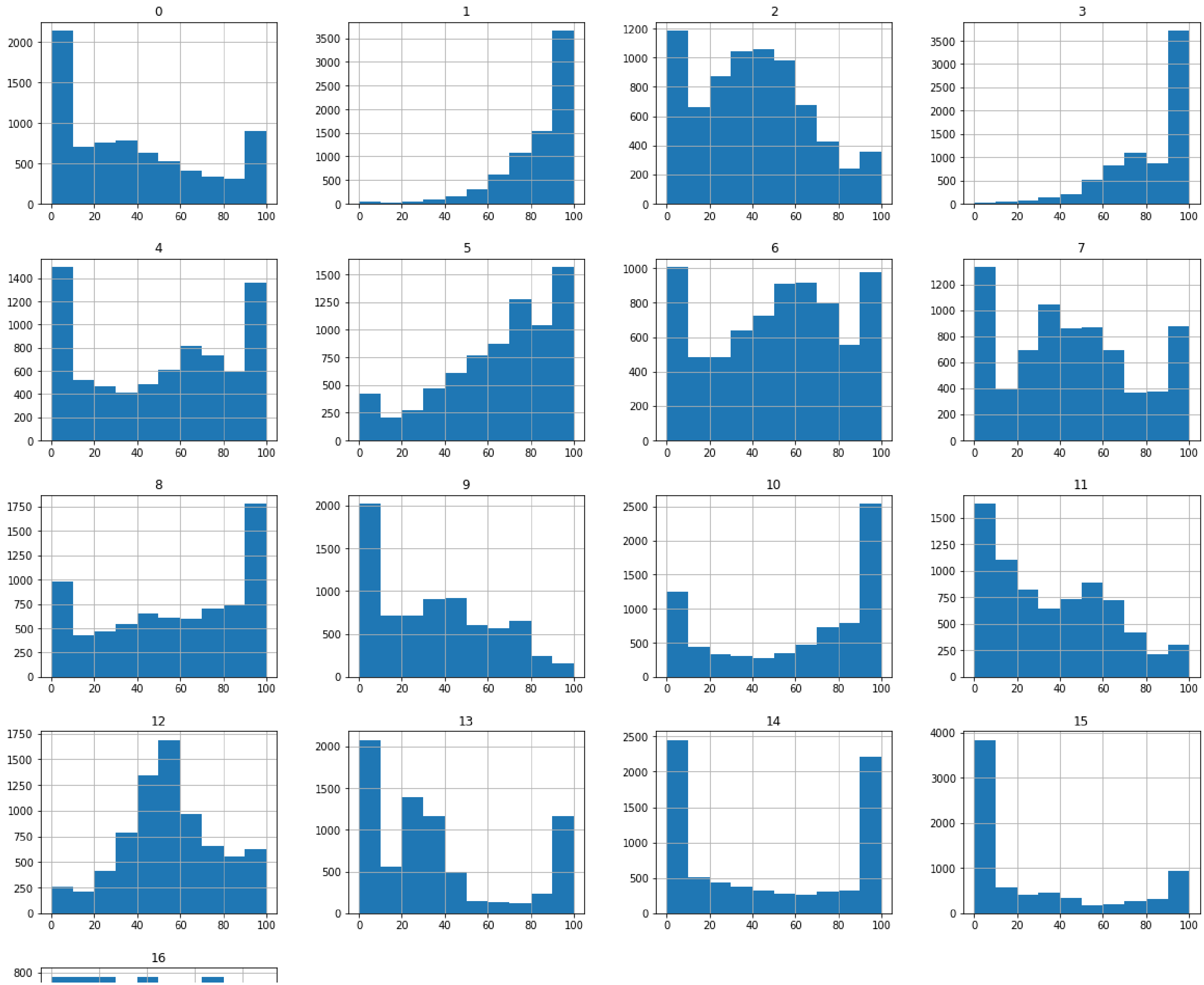
Out[84]:

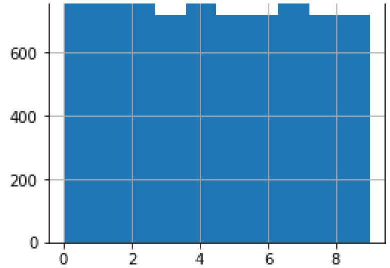
	0	1	2	3	4	5	6	7	8	
count	7494.000000	7494.000000	7494.000000	7494.000000	7494.000000	7494.000000	7494.000000	7494.000000	7494.000000	7494.00000
mean	37.384307	84.679343	40.005604	82.889512	50.878303	65.044436	51.471844	44.599680	57.129971	34.06912
std	33.322024	16.848420	26.256025	19.638582	34.927201	27.377341	30.680075	30.659478	33.680340	27.45998
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	5.000000	76.000000	20.000000	70.000000	17.000000	48.000000	28.000000	22.000000	30.000000	7.00000
50%	31.000000	89.000000	39.000000	89.000000	56.000000	71.000000	54.000000	42.000000	60.000000	33.00000
75%	61.000000	100.000000	58.000000	100.000000	81.000000	86.000000	75.000000	65.000000	88.000000	55.00000
max	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.00000

In [85]:
```
df1_var
```

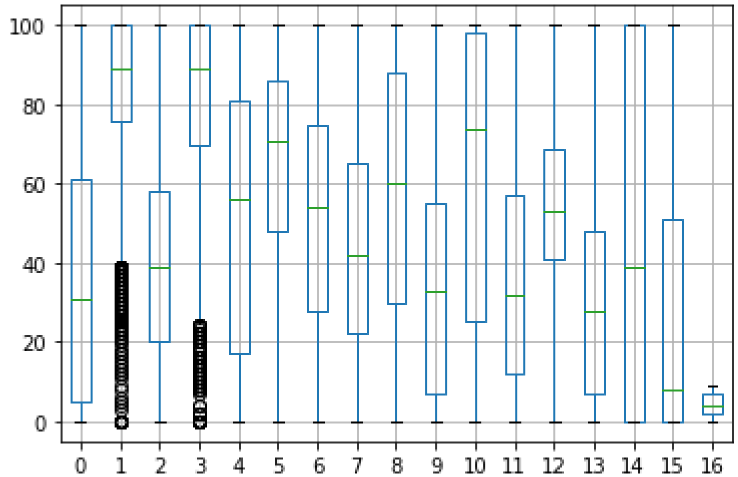
Out[85]:
```
0      1110.357293
1       283.869273
2       689.378856
3       385.673898
4      1219.909384
5       749.518778
6       941.266990
7       940.003609
8      1134.365329
9       754.051007
10     1378.693497
11      756.020974
12      510.750095
13     1103.808295
14     1724.889891
15     1282.434467
16        8.277018
dtype: float64
```

In [86]:
```
df1_hist = df1.hist(figsize=(20,20))
```





In [87]:
```
df1_box = df1.boxplot()
```



3 Conceptual Questions [30 points]

Answer the following questions about the analysis you just performed. Include the answers to this questions as text content (using markdown or text cells on Jupyter notebook) in the same notebook file used for parts 1 and 2.

3.1 Consider the histograms you generated for the Iris dataset. How do the shapes of the histograms for petal length and petal width differ from those for sepal length and sepal width? Now consider just the petal length histogram. Is there a particular value of petal length (which 2 ranges from 1.0 to 6.9) where the distribution of petal lengths (as illustrated by the histogram) could be best segmented into two parts?

Answer: The petal length and width histograms are noticeably disconnected between 2 and 3 and 0.5 and 1 centimeters respectively. The sepal length and width histograms have a more uniform continuous distribution across all values. As mentioned, the petal length is disconnected between 2 and 3 centimeters, so it could be segmented into to parts ranging from 0 to 2.5 and 2.6 to 6.9.

3.2 Now consider the boxplots you generated for the Iris dataset. There should be four boxplots, one for each feature. Based upon these boxplots, is there a pair of features that appear to have significantly different medians? Recall that the degree of overlap between variabilities is an important initial indicator of the likelihood that differences in means or medians are meaningful. Also, based solely upon the box plots, which feature appears to explain the greatest amount of the data?

Answer: Sepal length and petal width have significantly different medians with ~7.5 and ~1.33 centimeters respectively. Petal length has the largest range out of the 4 box plots which can explain the greatest amount of data.

3.3 Lastly, consider the boxplots you generated for the Digits dataset. Do you observe any outliers? If so, for what features? Now consider the corresponding histograms. What sort of distribution do the second and forth features display? With that in mind, explain the outliers, or lack thereof, in terms of what you observe from the histograms

Answer: There are a lot of outliers for features 1 and 3. Histograms 1 and 3 (second and fourth) display a left skewed distribution (https://blog.minitab.com/blog/3-things-a-histogram-can-tell-you) (https://blog.minitab.com/blog/3-things-a-histogram-can-tell-you)). For both of these histograms, the data starts to become left-skewed past 40 and 20 respectively which is also where we see the outliers fall beneath on the box plots.

```
In [ ]:
```