In [1]:
```
#import packages
import pandas as pd
import seaborn as sns
```

Running this chunk lets you have multiple outputs from a single chunk; run it first!

from IPython.core.interactiveshell import InteractiveShell InteractiveShell.ast_node_interactivity = "all"

In [2]:
```
flights = sns.load_dataset('flights')
flights.to_csv("flights.csv")
```

In [3]:
```
flights= pd.read_csv("flights.csv", index_col = 0)
```

In [4]:
```
flights.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 144 entries, 0 to 143
Data columns (total 7 columns):
Unnamed: 0.1          144 non-null int64
Unnamed: 0.1.1        144 non-null int64
Unnamed: 0.1.1.1      144 non-null int64
Unnamed: 0.1.1.1.1    144 non-null int64
year                  144 non-null int64
month                 144 non-null object
passengers            144 non-null int64
dtypes: int64(6), object(1)
memory usage: 9.0+ KB
```

This data set has 3 columns and 144 observations. Each row in this dataset represents an observation.

In [5]:
```
#view the first rows
flights.head()
```

Out[5]:

	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	year	month	passengers
0	0	0	0	0	1949	January	112
1	1	1	1	1	1949	February	118
2	2	2	2	2	1949	March	132
3	3	3	3	3	1949	April	129
4	4	4	4	4	1949	May	121

In [6]:
```
#perform EDA

flights.describe()
```

Out[6]:

	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	year	passengers
count	144.000000	144.000000	144.000000	144.000000	144.000000	144.000000
mean	71.500000	71.500000	71.500000	71.500000	1954.500000	280.298611
std	41.713307	41.713307	41.713307	41.713307	3.464102	119.966317
min	0.000000	0.000000	0.000000	0.000000	1949.000000	104.000000
25%	35.750000	35.750000	35.750000	35.750000	1951.750000	180.000000
50%	71.500000	71.500000	71.500000	71.500000	1954.500000	265.500000
75%	107.250000	107.250000	107.250000	107.250000	1957.250000	360.500000
max	143.000000	143.000000	143.000000	143.000000	1960.000000	622.000000

In [7]:
```
flights['month'].value_counts()
```

Out[7]:
```
July          12
October       12
February      12
June          12
November      12
March         12
May           12
September     12
April         12
December      12
January       12
August        12
Name: month, dtype: int64
```

In [8]:
```
#filter
(flights.filter(['passengers', 'month'])
#group by sex
.groupby(['month'])
#compute mean, standard deviation, and counts
.agg(['mean','std','count']))
```

Out[8]:

	passengers		
	mean	std	count
month			
April	267.083333	107.374839	12
August	351.083333	155.783333	12
December	261.833333	103.093808	12
February	235.000000	89.619397	12
January	241.750000	101.032960	12
July	351.333333	156.827255	12
June	311.666667	134.219856	12
March	270.166667	100.559194	12
May	271.833333	114.739890	12
November	232.833333	95.185783	12
October	266.583333	110.744964	12
September	302.416667	123.954140	12

In [ ]: