In [3]:

```
import pandas as pd
odf = pd.read_csv('MIS581data.csv')
odf.head()
```

Out[3]:

	State	ID	Hourly	Annual
0	MA	9924	26.50	55114.22660
1	MA	19303	26.07	54865.94911
2	MA	8909	26.01	54741.82433
3	MA	10675	25.88	54489.89727
4	MA	14963	25.74	53938.51356

In [5]:

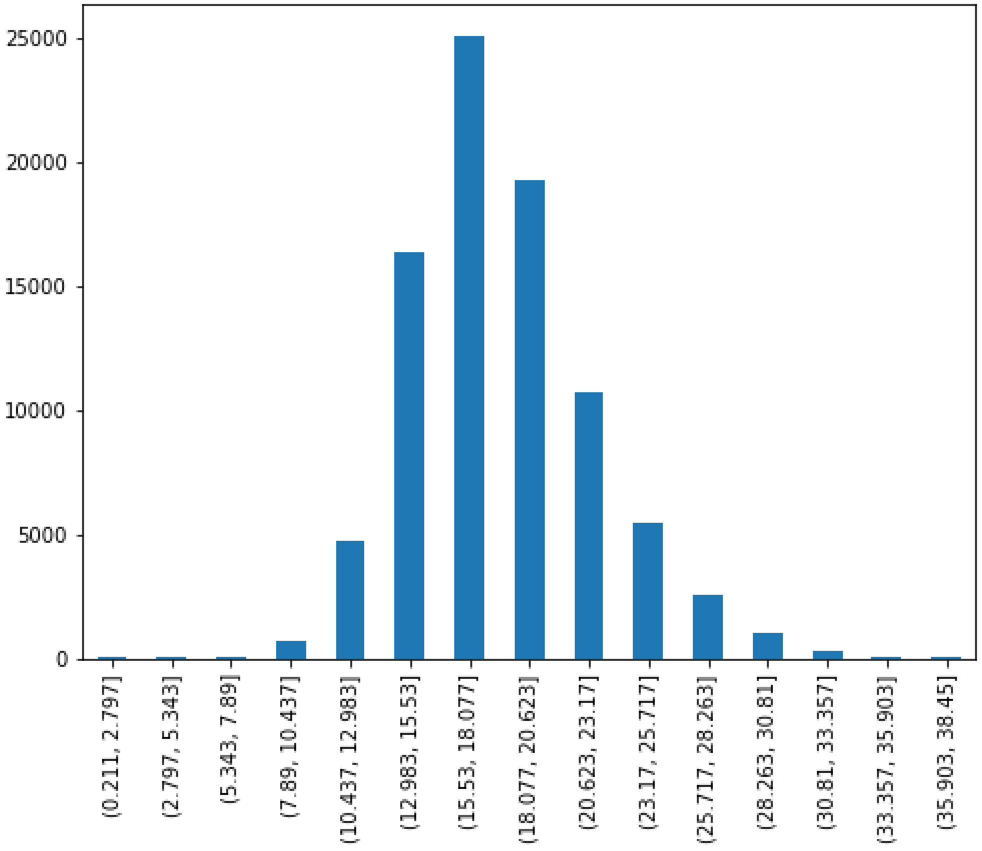
```
odf.State.value_counts()
```

Out[5]:

```
MA    46150
WA    40010
Name: State, dtype: int64
```

In [58]:
```
# Plotting
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(8,6))
# Bar chart of hourly wages
odf.Hourly.value_counts(bins=15, sort=False).plot(kind='bar')
```

Out[58]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21b1056a070>
```

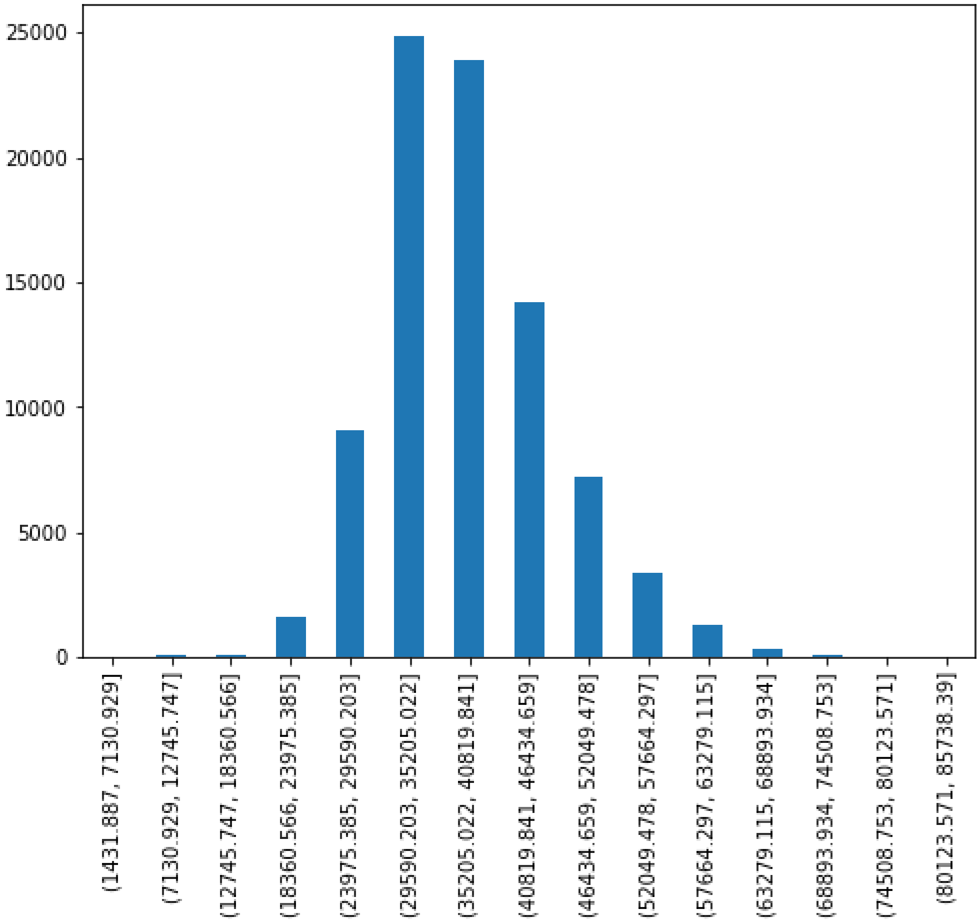


In [60]:

```python
# Bar chart of Annual wages
plt.figure(figsize=(8,6))
odf.Annual.value_counts(bins=15, sort=False).plot(kind='bar')
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21b0e96d070>
```



In [54]:

```
import math
import statistics
import numpy as np
import scipy.stats
from statistics import variance
from statistics import stdev

# MA---
print("Massachusetts Statistics:")
#Average hourly wage
print("MA Hourly Mean:      ", end="")
print(odf[odf['State']=='MA'].Hourly.mean())
# Median hourly wage
print("MA Hourly Median:    ", end="")
print(odf[odf['State'] == "MA"].Hourly.median())
# Range
Maximum = max(odf[odf['State'] == "MA"].Hourly)
Minimum = min(odf[odf['State'] == "MA"].Hourly)
Range = Maximum-Minimum
print("MA Maximum = {}, Minimum = {} and Range = {}".format(Maximum, Minimum,
Range))
# Variance
print("MA Hourly Variance:  ", end="")
print(variance(odf[odf['State']=='MA'].Hourly))
# Standard Deviation
print("MA Hourly Standard Deviation:  ", end="")
print(stdev(odf[odf['State']=='MA'].Hourly))
#Average Annual wage
print("MA Annual Mean:      ", end="")
print(odf[odf['State']=='MA'].Annual.mean())
# Median Annual wage
print("MA Annual Median:    ", end="")
print(odf[odf['State'] == "MA"].Annual.median())
# Range
Maximum = max(odf[odf['State'] == "MA"].Annual)
Minimum = min(odf[odf['State'] == "MA"].Annual)
Range = Maximum-Minimum
print("MA Maximum = {}, Minimum = {} and Range = {}".format(Maximum, Minimum,
Range))
# Variance
print("MA Annual Variance:  ", end="")
print(variance(odf[odf['State']=='MA'].Annual))
# Standard Deviation
print("MA Annual Standard Deviation:  ", end="")
print(stdev(odf[odf['State']=='MA'].Annual))
print("  ")

#WA---
print("Washington Statistics:")
print("WA Hourly Mean:      ", end="")
print (odf[odf['State']=='WA'].Hourly.mean())
print("WA Hourly Median:    ", end="")
print(odf[odf['State'] == "WA"].Hourly.median())
Maximum = max(odf[odf['State'] == "WA"].Hourly)
Minimum = min(odf[odf['State'] == "WA"].Hourly)
Range = Maximum-Minimum
```

```
print("WA Maximum = {}, Minimum = {} and Range = {}".format(Maximum, Minimum,
Range))
print("WA Hourly Variance:  ", end="")
print(variance(odf[odf['State']=='WA'].Hourly))
print("WA Hourly Standard Deviation:  ", end="")
print(stdev(odf[odf['State']=='WA'].Hourly))
print("WA Annual Mean:      ", end="")
print (odf[odf['State']=='WA'].Annual.mean())
print("WA Annual Median:    ", end="")
print(odf[odf['State'] == "WA"].Annual.median())
Maximum = max(odf[odf['State'] == "WA"].Annual)
Minimum = min(odf[odf['State'] == "WA"].Annual)
Range = Maximum-Minimum
print("WA Maximum = {}, Minimum = {} and Range = {}".format(Maximum, Minimum,
Range))
print("WA Annual Variance:  ", end="")
print(variance(odf[odf['State']=='WA'].Annual))
print("WA Annual Standard Deviation:  ", end="")
print(stdev(odf[odf['State']=='WA'].Annual))
```

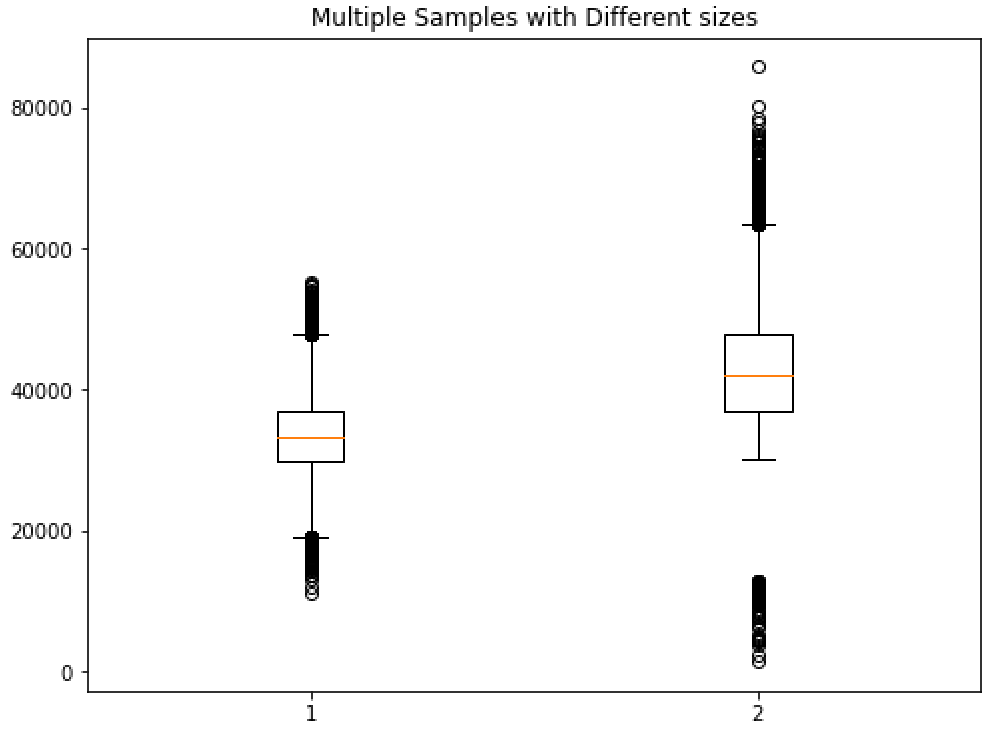
```
Massachusetts Statistics:
MA Hourly Mean:      16.057868905742154
MA Hourly Median:    16.05
MA Maximum = 26.5, Minimum = 6.23 and Range = 20.27
MA Hourly Variance:  6.678781618385658
MA Hourly Standard Deviation:  2.584333882915607
MA Annual Mean:      33430.99164900714
MA Annual Median:    33407.99479
MA Maximum = 55114.2266, Minimum = 11124.90397 and Range = 43989.32263
MA Annual Variance:  28566836.726873863
MA Annual Standard Deviation:  5344.795293261835

Washington Statistics:
WA Hourly Mean:      20.595051737065866
WA Hourly Median:    20.21
WA Maximum = 38.45, Minimum = 0.25 and Range = 38.2
WA Hourly Variance:  13.863571454995233
WA Hourly Standard Deviation:  3.723381722976471
WA Annual Mean:      42876.802418895444
WA Annual Median:    42027.78
WA Maximum = 85738.39, Minimum = 1516.11 and Range = 84222.28
WA Annual Variance:  60154657.02502826
WA Annual Standard Deviation:  7755.943335599369
```

In [62]:

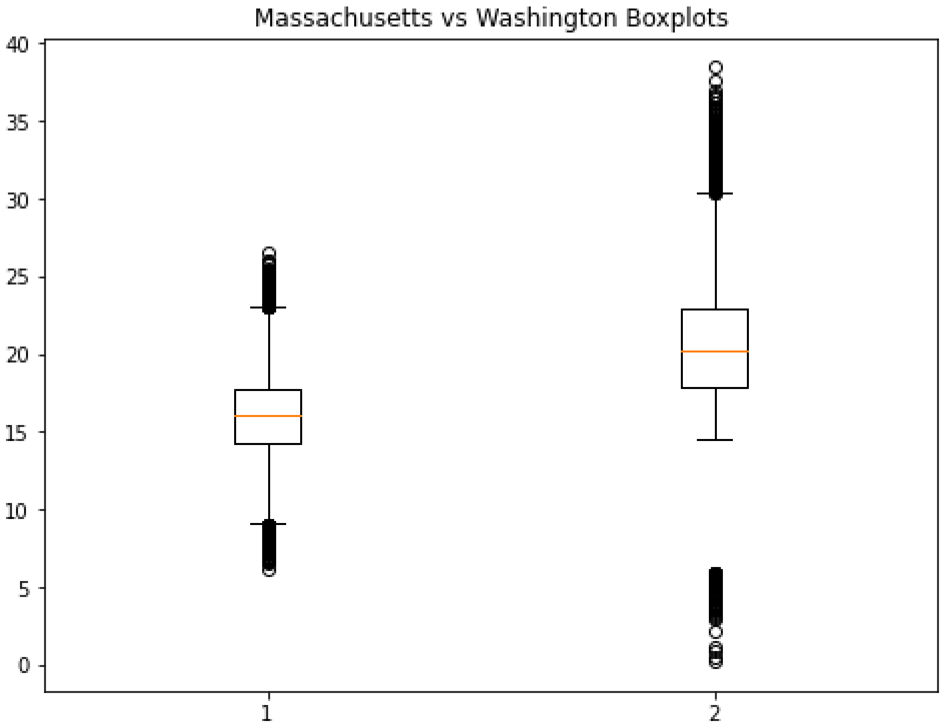
```
data = [odf[odf['State'] == "MA"].Annual, odf[odf['State'] == "WA"].Annual]
fig, ax = plt.subplots(figsize=(8,6))
ax.set_title('Multiple Samples with Different sizes')
ax.boxplot(data)

plt.show()
```



In [64]:
```
data = [odf[odf['State'] == "MA"].Hourly, odf[odf['State'] == "WA"].Hourly]
fig, ax = plt.subplots(figsize=(8,6))
ax.set_title('Massachusetts vs Washington Boxplots')
ax.boxplot(data)

plt.show()
```

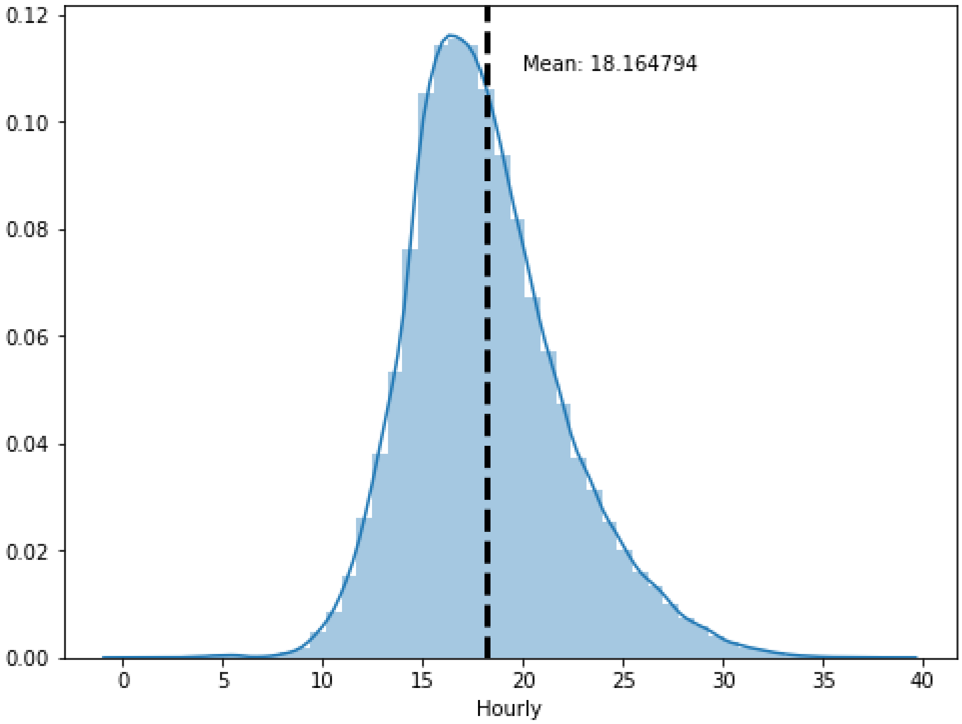


In [55]:

```
import seaborn as sns
sns.set
def plot_distribution(inp):
    plt.figure(figsize=(8,6))
    ax = sns.distplot(inp)
    plt.axvline(np.mean(inp), color='k', linestyle='dashed', linewidth=3)
    _,max_ = plt.ylim()
    plt.text(
        inp.mean() + inp.mean()/10,
        max_ - max_ / 10,
        "Mean: {:2f}".format(inp.mean())
    )
    return plt.figure
plot_distribution(odf.Hourly)
```

Out[55]:

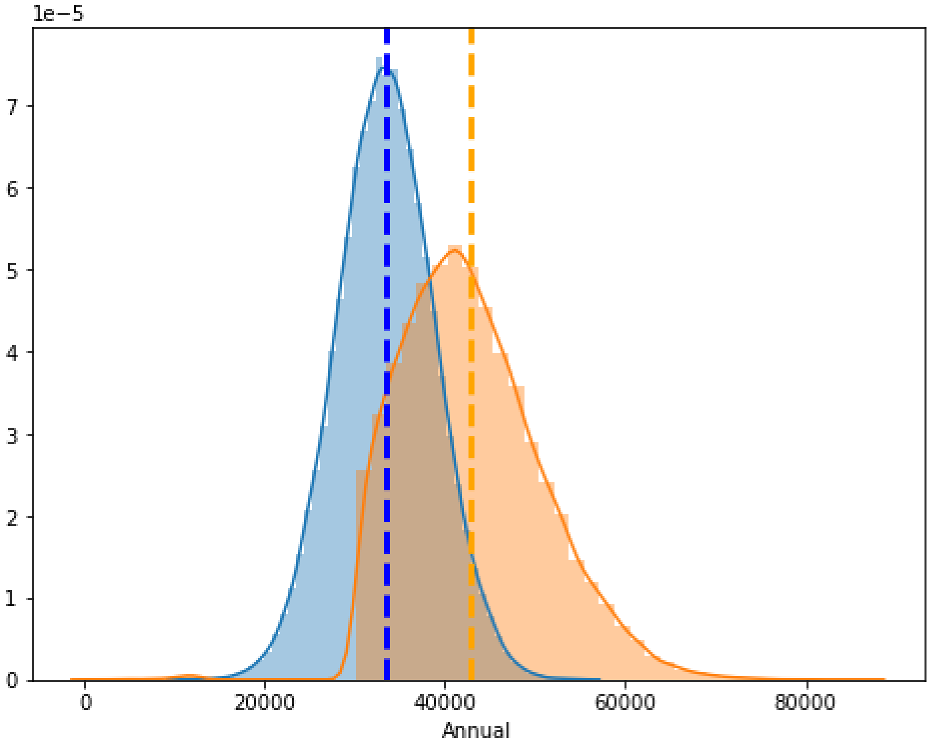
```
<function matplotlib.pyplot.figure(num=None, figsize=None, dpi=None, facecolo
r=None, edgecolor=None, frameon=True, FigureClass=<class 'matplotlib.figure.F
igure'>, clear=False, **kwargs)>
```



In [56]:
```
def plot_2_dist(arr1, arr2):
    plt.figure(figsize=(8,6))
    ax1 = sns.distplot(arr1)
    ax2 = sns.distplot(arr2)
    plt.axvline(np.mean(arr1), color='b', linestyle='dashed', linewidth=3)
    plt.axvline(np.mean(arr2), color='orange', linestyle='dashed', linewidth=3
)
    return plt.figure

plot_2_dist(odf[odf['State']=='MA'].Annual, odf[odf['State']=='WA'].Annual)
```

Out[56]:
```
<function matplotlib.pyplot.figure(num=None, figsize=None, dpi=None, facecolo
r=None, edgecolor=None, frameon=True, FigureClass=<class 'matplotlib.figure.F
igure'>, clear=False, **kwargs)>
```



In [49]:

```
from scipy.stats import f_oneway
from scipy.stats import ttest_ind
def compare_2_samples(arr_1, arr_2, alpha, sample_size):
    stat, p = ttest_ind(arr_1, arr_2)
    print('Statistics=%.3f, p=%.3f' % (stat, p))
    if p > alpha:
        print('Same distributions (fail to reject H0)')
    else:
        print('Different distributions (reject H0 in favor of H1)')

sample_size = 30000
arr1_sampled = np.random.choice(odf[odf['State']=='MA'].Annual, sample_size)
arr2_sampled = np.random.choice(odf[odf['State']=='WA'].Annual, sample_size)
compare_2_samples(arr1_sampled, arr2_sampled, 0.05, sample_size)
```

```
Statistics=-175.283, p=0.000
Different distributions (reject H0 in favor of H1)
```

In [ ]: