

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import scipy.stats as st
```

## Settembre 2018

### Esercizio 0

1)  $X \sim \text{Bern}(p)$  con  $p \in (0, 1)$

2)  $\text{Var}(X) = p(1-p)$  quindi  $\text{dvstd} = \sqrt{p(1-p)}$

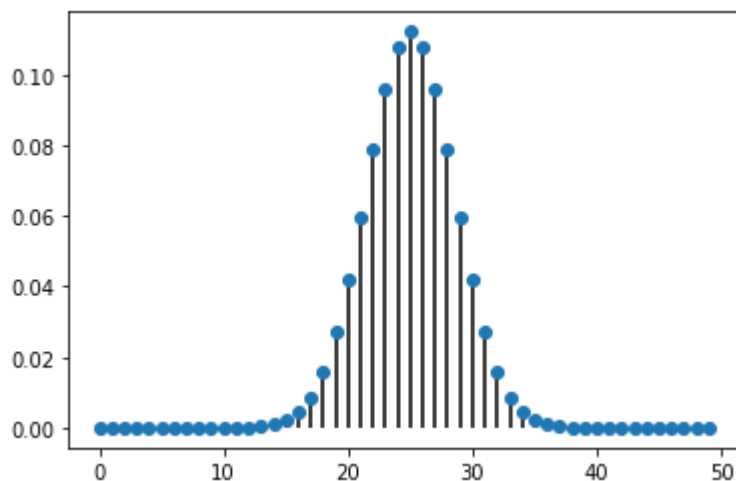
3)  $k \in \mathbb{N}$ ,  $Y \sim \text{Bin}(k, p)$

$$P(X = n) = \binom{k}{n} p^n (1-p)^{k-n}$$

4)  $k = 50$ ,  $p = 0.5$

In [3]:

```
Y=st.binom(50,0.5)
x=np.arange(0,50)
plt.vlines(x,0,Y.pmf(x))
plt.plot(x,Y.pmf(x),'o')
plt.show()
```



## 5) Definizione Stimatore

$$6) E(X) = p, T_n = \bar{X}$$

$$E(T_n) = E\left(\sum \frac{X_i}{n}\right) = \frac{1}{n}E\left(\sum X_i\right) = \frac{1}{n} \sum E(X_i) = \frac{n}{n}E(X) = p$$

$$7) n \geq 1$$

$$\text{Dimostrare che } P(-\epsilon < T_n - p < \epsilon) \approx 2\Phi\left(\epsilon \frac{\sqrt{n}}{\sigma}\right) - 1$$

Standardizzo:  $P(|T_n - p| < \epsilon)$  lo divido per la deviazione standard di  $T_n$  cioè  $\sqrt{\frac{1}{n}\sigma^2}$  e ottengo

$$P\left(|Z| < \frac{\epsilon\sqrt{n}}{\sigma}\right)$$

Applico il teorema del limite centrale:

$$P\left(|Z| < \frac{\epsilon\sqrt{n}}{\sigma}\right) \approx \Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - \Phi\left(-\frac{\epsilon\sqrt{n}}{\sigma}\right) = \Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - (1 - \Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right)) = 2\Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - 1$$

## Esercizio 1

In [4]:

```
fin = pd.read_csv("finanziamenti.csv", delimiter=";", decimal=",")
fin.columns
```

Out[4]:

```
Index(['id', 'TemaPrioritario', 'FONTE', 'CodiceCategoria', 'CATEGORIA',
      'UNITA', 'FinProvincia', 'FinRegione', 'TotSpese'],
      dtype='object')
```

In [5]:

```
#1
fin['CodiceCategoria']
print("Valore nominale in quanto indica un codice")
```

Valore nominale in quanto indica un codice

In [6]:

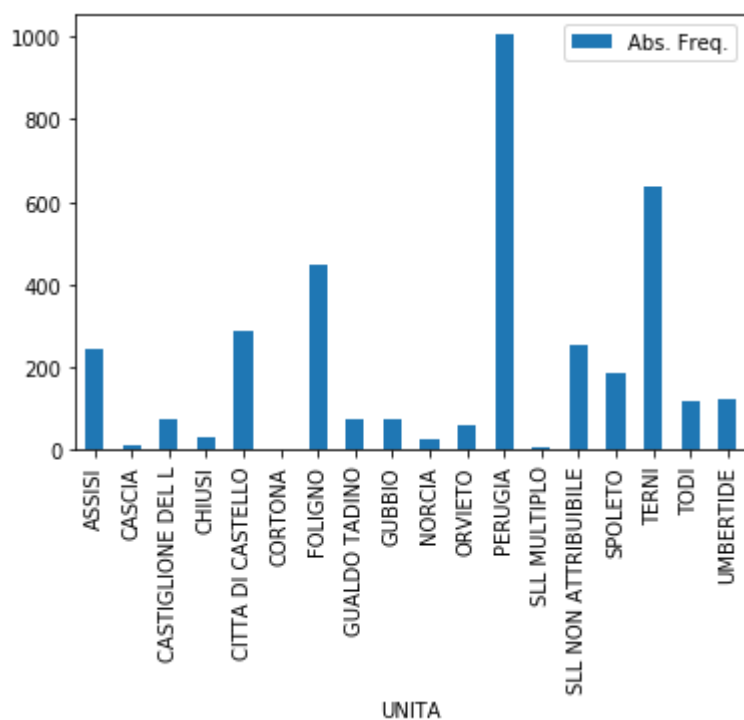
```
#2
finu = pd.crosstab(index=fin['UNITA'],columns=['Abs. Freq.'],colnames=[''])
finu
```

Out[6]:

	Abs. Freq.
UNITA	
ASSISI	243
CASCIA	13
CASTIGLIONE DEL L	75
CHIUSI	30
CITTA DI CASTELLO	288
CORTONA	1
FOLIGNO	449
GUALDO TADINO	75
GUBBIO	76
NORCIA	28
ORVIETO	60
PERUGIA	1005
SLL MULTIPLO	5
SLL NON ATTRIBUIBILE	255
SPOLETO	186
TERNI	638
TODI	117
UMBERTIDE	124

In [10]:

```
#3
finu.plot.bar()
plt.show()
```



In [25]:

```
#4 vedi foglio
```

In [27]:

```
#5
progetti_a=fin[fin['FinProvincia']<(fin['FinRegione'])]
progetti_b=fin[fin['FinProvincia']>=(fin['FinRegione'])]
```

In [34]:

```
print("A: {}".format(len(progetti_a)))
print("B: {}".format(len(progetti_b)))
```

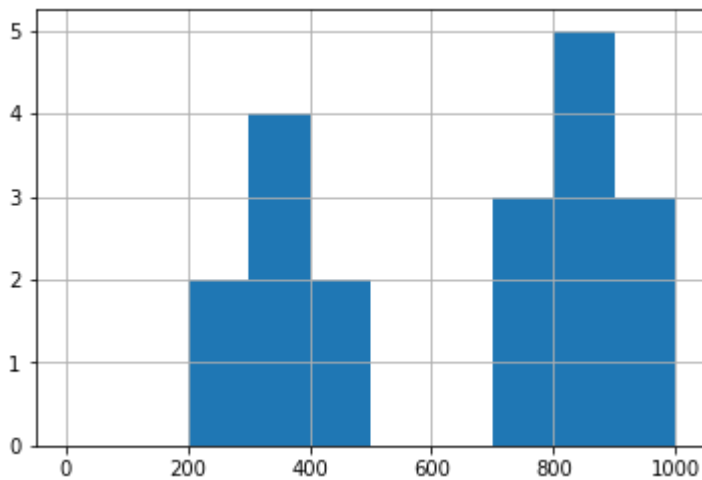
A: 368  
B: 3284

In [36]:

```
#6.1
mask = progetti_a['FinProvincia'] >= 200
mask1 = progetti_a['FinProvincia'] < 1000
selezione_progetti_a = progetti_a[mask & mask1]
```

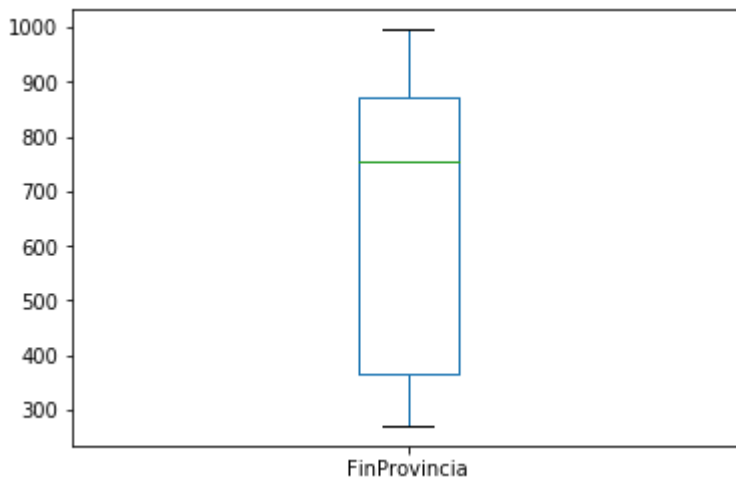
In [39]:

```
#6.2
bins=np.arange(0,1001,100)
selezione_progetti_a['FinProvincia'].hist(bins = bins)
plt.show()
```



In [40]:

```
#6.3
selezione_progetti_a['FinProvincia'].plot.box()
plt.show()
```



In [41]:

```
#6.4
print("Tra i due grafici ritengo che l'istogramma sia più informativo in quanto, nonost  
ante nel box plot possiamo trarre un sacco di informazioni non rileviamo la più importa  
nte : il carattere è suddiviso in due gruppi che sembrerebbero seguire una distribuzion  
e normale 'BIMODALE'")
```

Tra i due grafici ritengo che l'istogramma sia più informativo in quanto, nonostante nel box plot possiamo trarre un sacco di informazioni non rileviamo la più importante : il carattere è suddiviso in due gruppi che sembrerebbero seguire una distribuzione normale 'BIMODALE'

In [42]:

```
#6.5
selezione_progetti_a['FinProvincia'].mean()
```

Out[42]:

636.9052631578948

In [43]:

```
selezione_progetti_a['FinProvincia'].std()
```

Out[43]:

264.80233322588253

In [47]:

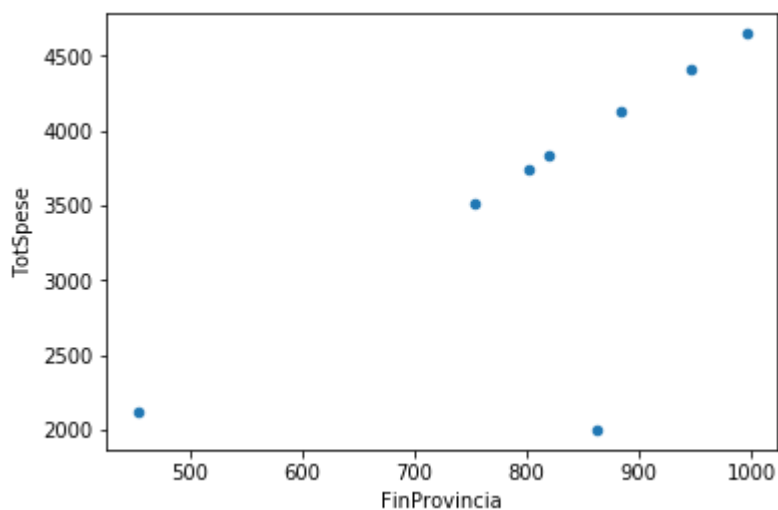
```
#6.6
print("Nessuno vedi grafico")
```

Nessuno vedi grafico

In [52]:

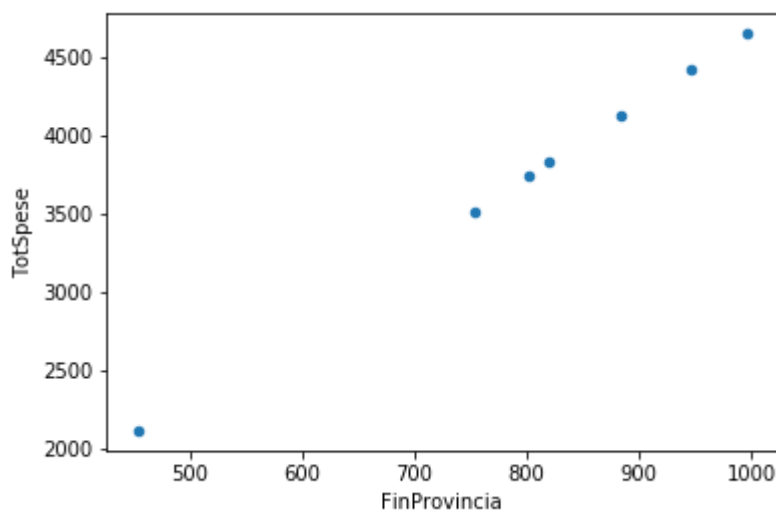
```
#6.7
print(selezione_progetti_a['FinProvincia'].corr(selezione_progetti_a['TotSpese']))
selezione_progetti_a.plot.scatter('FinProvincia', 'TotSpese')
plt.show()
```

0.6964011723762348



In [53]:

```
#6.8
selezione_progetti_a_noutliers=selezione_progetti_a[selezione_progetti_a['TotSpese']>2000]
selezione_progetti_a_noutliers.plot.scatter('FinProvincia','TotSpese')
plt.show()
```



## Esercizio 2

In [54]:

```
#1
len(fin)-len(fin.dropna(axis=0,subset=['TotSpese']))
```

Out[54]:

1134

In [56]:

```
#2
Z=st.norm()
dev=fin['TotSpese'].std()
(Z.ppf(1.95/2)*(len(fin))**0.5)/dev,(Z.ppf(1.95/2)*dev)/len(fin)**0.5
```

Out[56]:

(0.0005678277621445123, 6765.183171365392)

In [57]:

```
#3
```

In [58]:

```
#4  
print("Normale")
```

Normale

In [59]:

```
#5
```

In [60]:

```
#6
```

In [61]:

```
#7
```

In [62]:

```
#8
```