Why we need statistics for marketing?

The study of statistics can roughly be categorized as one of two types: descriptive and inferential.

Descriptive statistics "describes" data. Think of this as summarizing what is directly observable.

e.g. What does the representative data point look like?

Other Examples?

Inferential statistics makes "inferences" about the population based on a sample from the population.

Example of population vs. sample?

Why is statistics needed to make inference from a sample?

Example of a marketing problem that statistics can address where we have a sample? Population?

e.g. Based on last week's sales, what is likely sales for the year?

This is the more important kind of statistics, and the one we're most interested in as marketers. We can't always sample every potential customer, for example, but we can get a pretty good idea of the population's behavior/thoughts based on a sample drawn from the population.

Descriptive Statistics

Measures of Central Tendency

What does this mean?

If you have a bunch of data, it's difficult to make sense of it all without some simplifying measures. The typical measures we use to describe a set of data are those that try to reduce N data points into 1 single representative data point.

We want to summarize the data to one point, something that is "representative" of the whole dataset

Example 1, let's say you measured the sales generated by a bunch of salespeople and you get the following dataset:

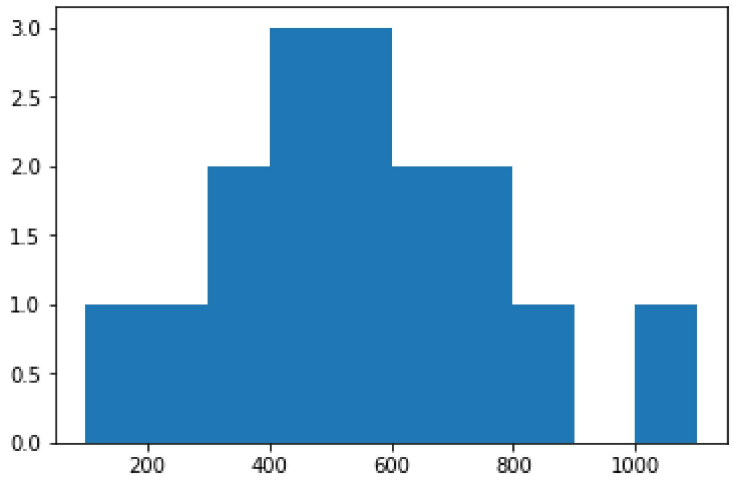
$$\{300, 430, 700, 370, 430, 600, 550, 800, 1100, 530, 700, 200, 100, 630, 430, 570\}$$

What does this data set look like?

In [1]:
```
import numpy as np, pandas as pd
% matplotlib inline
from matplotlib import pyplot as plt
data = pd.DataFrame([300,430,700,370,430,600,550,800,1100,530,700,200,100,630,430
data.columns = ['data']
from __future__ import division
data.to_csv('./example1.csv', index = False)
```

In [2]:
```
data = pd.read_csv('./example1.csv')
plt.hist(data.data)
```

Out[2]:
```
(array([ 1.,  1.,  2.,  3.,  3.,  2.,  2.,  1.,  0.,  1.]),
 array([  100.,   200.,   300.,   400.,   500.,   600.,   700.,   800.,
          900.,  1000.,  1100.]),
 <a list of 10 Patch objects>)
```



Looking at this dataset, what is a representative point? Why?

Intuitively, we have the idea of an "average" data point as being representative, it's basically the data point, μ, such that the differences for all points in the data set from μ sum up to 0.

Definition 1, mean

$$\mu = \frac{\sum_{k=1}^{k=N} x_k}{N}$$

$$\text{For some dataset } \{x_k\}_{k=1}^{N}$$

What's the mean of our dataset?

In [3]:
```
print 'The mean is '+ str(data.data.mean())
```

```
The mean is 527.5
```

Another representative point is to take the "middle" observation in a dataset.

Example: The middle of the dataset $\{1, 2, 3, 4, 5\}$ is 3.

What is this called?

Definition 2, median

The median, M, is defined as:

$$M = \begin{cases} x_{(n+1)/2}, & \text{if } n \text{ is odd} \\ (x_{\frac{n}{2}} + x_{\frac{n}{2}+1})/2, & \text{otherwise} \end{cases}$$

$$\text{For some ordered set } S = \{x_1, x_2, \ldots, x_n\} \text{ where } x_{k+1} \geq x_k \forall k \in \{1, \ldots, n-1\}$$

What happens when there's an even number of data points? What's the median of $\{1, 2, 3, 4, 5, 6\}$?

What is the median of our dataset in Example 1? Why is it different than the mean?

In [4]:
```
print 'The median is '+str(data.data.median())
```

```
The median is 540.0
```

For some datasets, it may make sense to look at the most commonly occurring value. This is typically most useful when the dataset has a relatively small number of possible values. This is called the mode.

Definition 3, mode

$$\text{Mode} = \arg\max_k \{\text{Size}\{x_j = x_k\}_{j=1}^N\} \; \forall k \in 1, \ldots, N$$

For example, the mode of the dataset of {1, 2, 2, 2, 3, 4} is 2.

What is the mode of our dataset? Does it make sense to use the mode for such data?

In [5]:
```
print 'The mode is '+str(data.data.mode())
```

```
The mode is 0    430
dtype: int64
```

Measures of dispersion

Is central tendency sufficient to describe a dataset?

For example, median, mean, and mode of the following 2 datasets are identical (0):

$$\{-1, 0, 0, 0, 1\}$$

$$\{-100, 0, 0, 0, 100\}$$

But they are obviously very different. How so?

The first set is less "spread out" than the second. This is the intuition for measures of dispersion.

The most basic measure of dispersion is range.

Definition 4, Range The range, R, of a dataset S is

$$R = \max S - \min S$$

What is the range of our dataset?

In [6]:
```
print 'The range is '+str(data.data.max()-data.data.min())
```

```
The range is 1000
```

The range measure only uses 2 data points, ignoring all the other data points.

This means that the range is susceptible to unrepresetnative outliers. We need a better measure that uses more of the data.

One solution to censor outliers is to use the "interquartile" range. This means, we look at only the 75th and 25th percentile values, and compute the difference between these values.

Example: Interquartile range of $\{1, 2, 3, 4, 5, 6, 7, 8\}$ is 5, i.e. $(7 - 2)$

While this eliminates outliers, it still does not take all values into account.

What we need is sort of a "representative" difference from a "representative" point of the dataset.

i.e. on "average" how far is a data point from an average data point.

Dispersion: Standard Deviation

The most mathematically "logical" measure is standard deviation.

With standard deviation, we take the square root of the "average" of the squared distance from the average of every entry in a dataset.

Definition 5, variance, standard deviaiton

Variance:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Standard Deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

This means we first calculate an average, then we subtract that average from each value in the dataset, square this difference, then add all of these squared differences up. We then divide by the number of these differences...finally, to rescale, we take the square root.

Sample Adjuistments (degree of freedom adjustment)

Basically, divide by $n-1$ instead of n

Definition 6, sample variance, standard deviation

Sample variance is:

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2$$

Sample standard Deviation:

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Notice from the formula, can standard deviation ever be negative?

Going back to our original dataset, what is the standard deviation?

In [7]:
```
print "The population standard deviation is " + str(np.std(data.data,ddof=0))
print "The sample standard deviation is " + str(np.std(data.data,ddof=1))
```

```
The population standard deviation is 233.81349405
The sample standard deviation is 241.481538287
```

More details on standard deviation

The most mathematically "logical" measure (you'll learn this if you take a calculus based stats class in grad school) of dispersion of the sort we describe above is standard deviation. With standard deviation, we take the square root of the "average" of the squared distance from the average of every entry in a dataset. This means we first calculate an average, then we subtract that average from each value in the dataset, square this difference, then add all of these squared differences up. We then divide by the number of these differences...finally, to rescale, we take the square root.

More concisely and mathematically

Definition 6, variance, standard deviation: Variance:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Standard Deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

However, if we are using a sample, our variance is,

Definition 7, sample variance, standard deviation:

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2$$

Sample standard Deviation:

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Going back to our original dataset, what is the standard deviation?

Notice from the formula, can standard deviation ever be negative?

There's a slight problem with the population formula. It assumes that the population mean is known for certain. However, there is always a difference between the population mean and the sample mean. So when we compute the squared differences using a sample mean, we are understating the "average" difference from the population mean. This sample bias of the mean must be corrected. It turns out, through some algebra, we can prove that the correct "correction" of this sample bias is to subtract 1 from N. The idea is that we have used the data points to compute a sample mean, so we are in essence using one of the data points "twice" (for the proof, see https://www.quora.com/On-the-sample-standard-deviation-why-do-we-subtract-N-by-1 (https://www.quora.com/On-the-sample-standard-deviation-why-do-we-subtract-N-by-1)). This is also known as the degree of freedom correction.

Sampling to learn about populations

Inferential statistics, is about how to make judgments about populations based on samples.

We will build up to statistical hypothesis testing starting from a very simple example, flipping a coin.

Assume we have a fair coin, with heads and tails. What's the probability it will land heads or tails?

What does probability mean?

Probability = 0? =1?

Simple enough. Let's flip a coin 100 times... Note that each flip is known as a "Bernoulli trial"

In [8]:
```
np.random.seed(111222)
p = .5
flips = 100
data = np.random.binomial(1, p, flips)
th = ['T','H']
print 'The 100 tosses: '+','.join([th[t] for t in data][:100])
```

```
The 100 tosses: H,H,T,T,T,H,T,T,H,H,T,H,T,H,T,H,T,T,H,H,T,T,T,T,T,T,T,H,T,T,
H,H,T,T,H,H,T,T,T,T,H,H,T,H,H,H,T,T,T,T,H,H,T,T,T,T,T,H,H,H,T,H,H,H,H,T,T,T,T,
T,H,H,H,H,H,T,T,T,H,T,T,H,H,H,T,H,T,H,T,H,H,T,H,H,T,T,T,T,T
```

How many heads did we expect to get?

... 50

How many did we actually get?

In [9]:
```
successes = (sum(data))
print 'We got ' +str(successes)+' heads out of the 100 coin flips'
```

```
We got 44 heads out of the 100 coin flips
```

Why didn't we get 50?

Discrete distributions:

At this point, we should introduce the concept of a distribution.

We want to answer questions like, what's the probability that in our sample, we will get exactly 50 heads? 49 heads? 70 heads?

A (discrete) probability distribution is just an assignment of probability to every outcome. This is called the **probability mass function (pmf)**.

We call this distribution discrete, because there are finitely countable number of outcomes in the event space.

For example, in our 100 tosses, there are either 0,1,2,3,..., or 100 heads.

It so happens that the probability of "success" of N Bernoulli trials follows a well known discrete distribution, the binomial distribution.

Binomial Distribution

The binomial distribution is relatively easily derived from the Bernoulli trials (proof here https://proofwiki.org/wiki/Binomial_Distribution_PMF (https://proofwiki.org/wiki/Binomial_Distribution_PMF)).

We will simply state the outcome here.

$$Pr(k \text{ successes}) = \binom{n}{k} p^k (1-p)^{k-1}$$

Where $\binom{n}{k}$ means **n choose k**, i.e. the binomial coefficient. This is the number of ways to have "k" successes from n trials regardless of order. Mathematically, this is computed as:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n \times (n-1) \times (n-2), \ldots, \times 1}{(n-k) \times (n-k-1), \ldots, \times 1 \times k \times k-1 \times k-2, \ldots, \times 1}$$

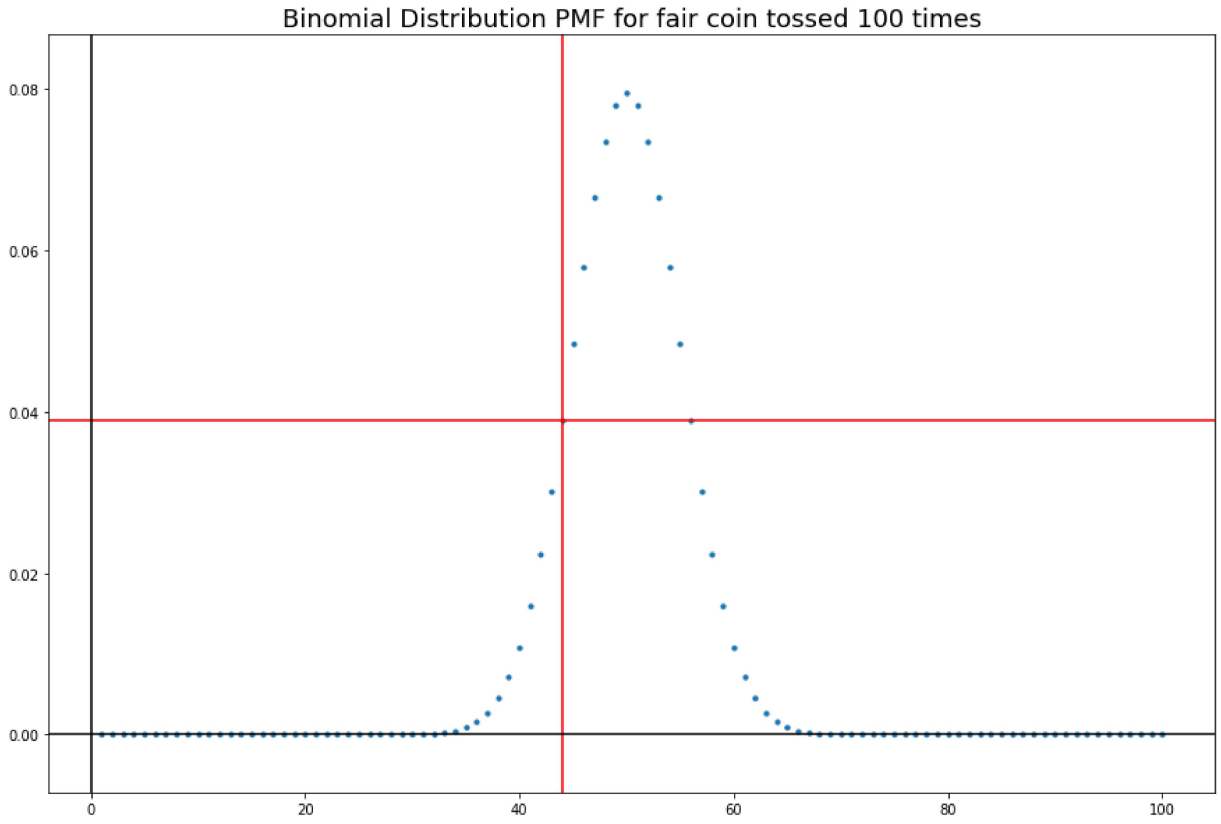
For example:

$$\binom{n}{0} = \frac{n!}{n!0!} = 1$$

Ok, so what does this distribution look like for our example? n=100, p =.5

In [10]:
```
import scipy as sp
from scipy import stats
eventspace = range(1,101)
binompmf = sp.stats.binom.pmf(eventspace, flips, p)
plt.figure(figsize = (15,10))
plt.scatter(eventspace, binompmf, s=10)
plt.axvline(x=successes, color='r')
plt.axhline(y=sp.stats.binom.pmf(successes, flips, p), color='r')
plt.axhline(y=0, color='black')
plt.axvline(x=0, color='black')
plt.title("Binomial Distribution PMF for fair coin tossed 100 times", fontsize =
print "The probability of getting our exact outcome of "+str(successes)+' is '+st
```

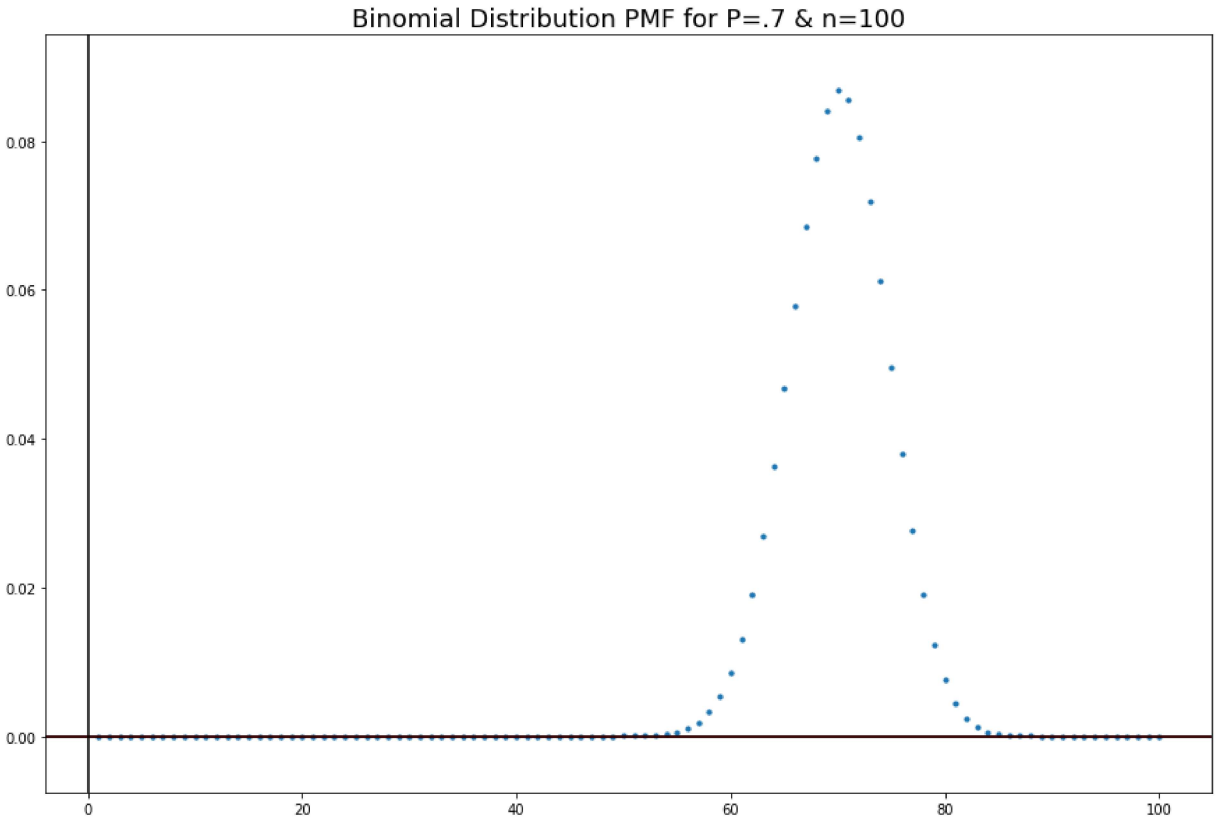
```
The probability of getting our exact outcome of 44 is 0.0389525597891
```



What about for $p = .7$?

In [11]:
```
p = .7
binompmf = sp.stats.binom.pmf(eventspace, flips, p)
plt.figure(figsize = (15,10))
plt.scatter(eventspace, binompmf, s=10)
plt.axhline(y=sp.stats.binom.pmf(successes, flips, p), color='r')
plt.axhline(y=0, color='black')
plt.axvline(x=0, color='black')
plt.title("Binomial Distribution PMF for P=.7 & n=100", fontsize = 18)
print "The probability of getting our exact outcome of "+str(successes)+' is '+st
```

```
The probability of getting our exact outcome of 44 is 3.95038655219e-08
```



Continuous distributions

As you may guess, distributions over **uncountable** event spaces are continuous distributions.

Example, uncountable things are like height. How many different heights are possible? We can't count them!

Interestingly, the probability that any 2 realizations from a continuous distribution is equal to each other is defined to be 0. **Any particular realization of a random variable drawn from a continuous distribution has a probability mass of 0**.

So instead of PMF we have a **probability density function (PDF)**.

PDF is defined by the derivative, or the instantaneous change, of the cumulative density function (CDF). **A CDF, in turn, is the function that maps the probability of x<= some X**

Normal distributions - a continuous distribution

Height is typically considered to be normally distributed. What's a normal distribution? It comes from the central limit theorem that we will introduce below. But let's first define it below.

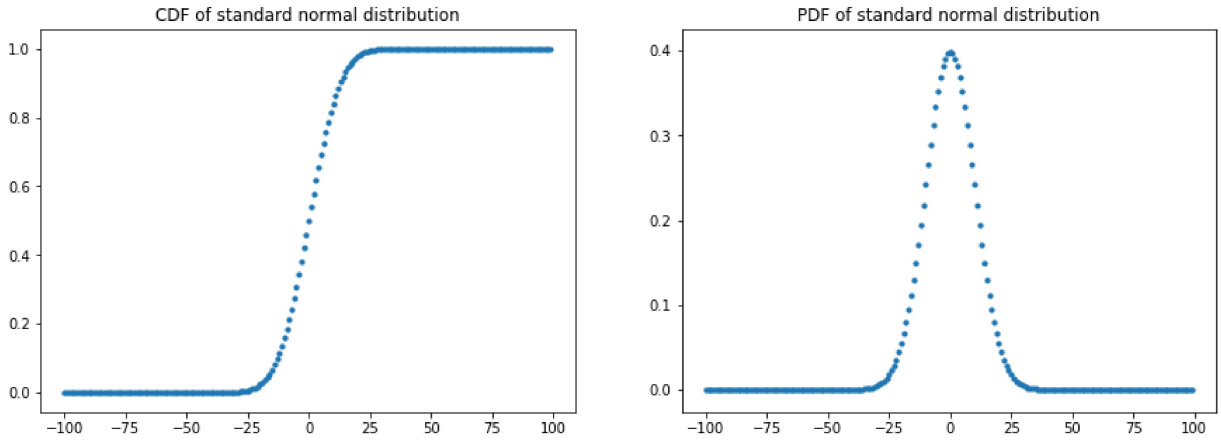
$$x \sim N(\mu, \sigma) \iff PDF(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

What does this look like? The familiar bell curve.

For example, let $\mu = 0, \sigma = 1$ as in the standard normal distribution. The probability density function looks like:

In [12]:
```
fig, ax = plt.subplots(1,2, figsize = (15,5))
rnormcdf = sp.stats.norm.cdf(np.array(range(-100,100))/10)
ax[0].scatter(range(-100,100), rnormcdf, s=10)
ax[0].set_title('CDF of standard normal distribution')
rnorm = sp.stats.norm.pdf(np.array(range(-100,100))/10)
ax[1].scatter(range(-100,100), rnorm, s=10)
ax[1].set_title('PDF of standard normal distribution')
```

Out[12]:
```
Text(0.5,1,u'PDF of standard normal distribution')
```



Okay, so what? (Central Limit Theorem)

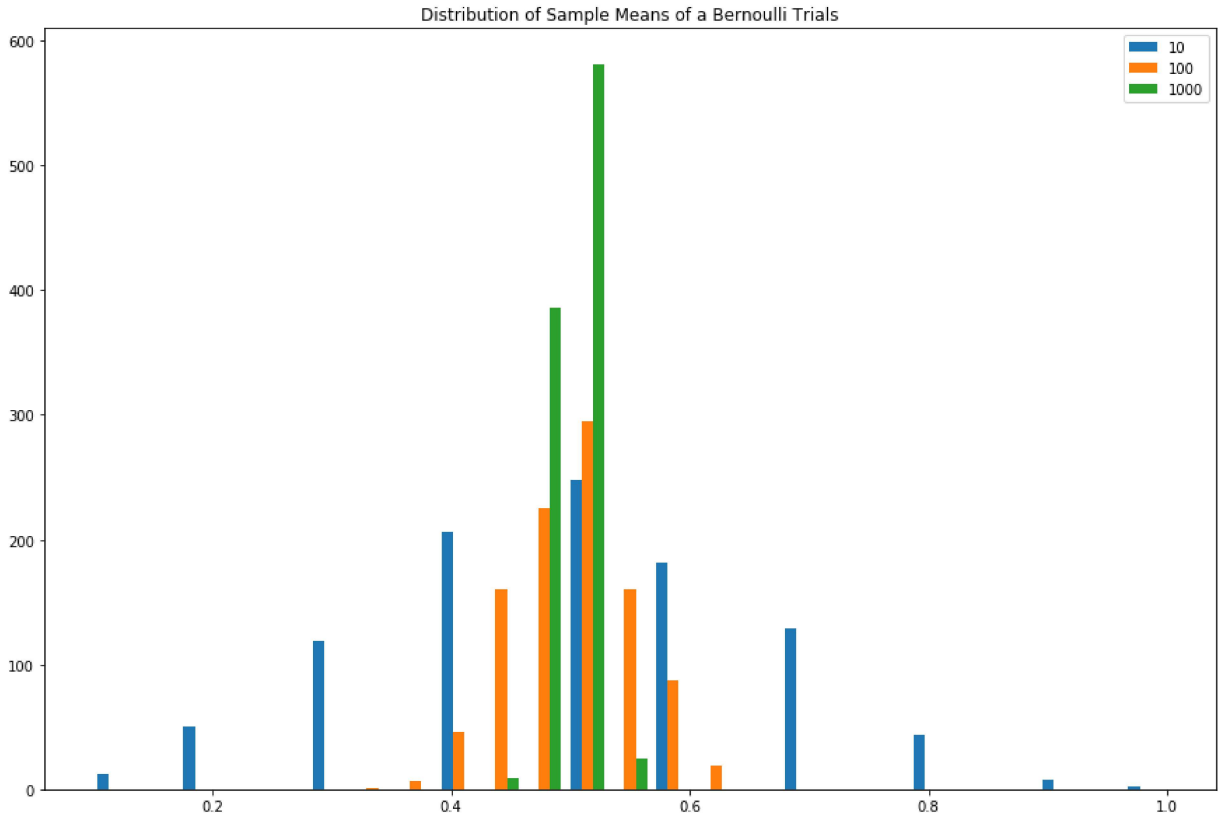
Okay, we learned something about the likelihood of getting our exact result using the binomial distribution. I thought we wanted to learn more about the population given a sample. This is a great place to introduce the central limit theorem.

Thought experiment. Suppose we tossed the coin 10/100/1000 times, then we repeated these bernoulli trials a lot. What does the distribution of the average sample success rate look like?

Let's do it!

In [13]:
```
S = list()
p = .5
fig = plt.figure(1, figsize = (15,10))
samplesizes = [10, 100, 1000]
for i,samp in enumerate(samplesizes):
    sampleofsamples = [np.mean(np.random.binomial(1, p, samp)) for r in range(0,1
    S.append(sampleofsamples)
common_params = dict(bins = 25, normed = False)
# common_params['histtype'] = 'step'
plt.hist(S,**common_params)
plt.legend(list((samplesizes)))
plt.title('Distribution of Sample Means of a Bernoulli Trials')
# plt.legend(samplesizes)
#     print 'Our sample of sample means have the following numbers'+','.join([str
```

Out[13]: Text(0.5,1,u'Distribution of Sample Means of a Bernoulli Trials')



In [14]:
```
print 'mean of 1000 samples of 1000 draws is ' +str(np.mean(S[-1]))
print 'standard deviation of 1000 samples of 1000 draws is '+str(np.std(S[-1]))
```

```
mean of 1000 samples of 1000 draws is 0.499745
standard deviation of 1000 samples of 1000 draws is 0.0162239321683
```

What's the point of this?

As we increase the number of draws to infinity, we end up with a normal distribution with smaller and smaller variance **NO MATTER WHAT DISTRIBUTION WE SAMPLE FROM!** ... it could be coin tosses, die rolls, measuring heights, or recorded sales.

Specifically, given a sequence of n draws $\{x_1, x_2, \ldots, x_n\}$ from any distribution with mean μ and standard deviation σ, as $n \to \infty$, the distribution of $\bar{x}_n = \frac{\sum_{i=1}^{n} x_i}{n}$ converges to $N(\mu, \sigma/\sqrt{(n)})$

For the Bernoulli distribution, the population mean is $p = .5$ and the population standard deviation is $\sqrt{(p(1-p))} = .5$. So with 1000 draws, the sample mean is distributed $N(.5, .5/\sqrt{1000})$

This is the power of the central limit theorem, it allows us to make asymptotic statistical inference about sample means.

Proof of means of binomial convergence to normal https://ocw.mit.edu/courses/nuclear-engineering/22-38-probability-and-its-applications-to-reliability-quality-control-and-risk-assessment-fall-2005/lecture-notes/sec4_binom_norm.pdf (https://ocw.mit.edu/courses/nuclear-engineering/22-38-probability-and-its-applications-to-reliability-quality-control-and-risk-assessment-fall-2005/lecture-notes/sec4_binom_norm.pdf)

Proof of theorem here https://en.wikipedia.org/wiki/Central_limit_theorem (https://en.wikipedia.org/wiki/Central_limit_theorem).

In [15]:
```
print 'mean and standard deviation from CLT: '+str(.5)+', '+str( np.sqrt(.25)/np.
print 'mean and standard deviation from 1000 samples: ' +str(np.mean(S[-1]))+', '
print 'Pretty close!'
```

```
mean and standard deviation from CLT: 0.5, 0.0158113883008
mean and standard deviation from 1000 samples: 0.499745, 0.0162239321683
Pretty close!
```

Now that we know sample means are distributed normally with increase in sample size

We can make inferences about the sample mean. Intuitively, CLT tells us that the larger the sample, the lower the standard deviation/variance of the distribution of the mean of the sample. The more we sample, the better our approximation of the population mean and population standard deviation using the sample mean and sample standard deviation.

This means that we can use the sample mean (best approximation for population mean) and sample standard deviation (best approximation for population standard deviation) to plug into the the CLT prediction of the distribution of the mean to construct a "confidence interval."

For example if we had 1000 samples of height and got a sample mean of 70 inches and a sample standard deviation of 6 inches, we can say the distribution of the mean of such a sample roughly follows $N(70, 6/\sqrt{1000}) = N(70, .19)$.

Since we know that this is a normal distribution, we can compute the 95% confidence interval. We just need to compute the inverse normal at p=.025 and p=.975.

In [16]:
```
bounds = sp.stats.norm.ppf([0.025, 0.975], loc=70, scale=6/np.sqrt(1000))
print 'Inverse of N(70,.19) evaluated at .025 and .975 are '+str(bounds[0])+', '+
```

```
Inverse of N(70,.19) evaluated at .025 and .975 are 69.6281229806, 70.371877019
4
```

In other words, based on our sample of 1000 heights, we can say with 95% confidence, that the average height of the population is between 69.63 and 70.37 inches

Note that the previous confidence interval is constructed using normal distribution because we have a large sample

What happens when we have smaller samples? Is the normal distribution still appropriate?

The CLT only says that the sample mean converges to a normal when sample sizes go to infinity. What about when sample sizes are not that big?

In this case, we need to use the t-distribution. There's a proof of the degrees of freedom correction that leads to the t-distribution on wikipedia. https://en.wikipedia.org/wiki/Student's_t-distribution (https://en.wikipedia.org/wiki/Student's_t-distribution)

Think of the t-distribution as a normal distribution corrected for its small sample size.

Let's say we got the sample mean and standard deviation as above (70, 6), but from a much smaller sample of 25. We should construct the confidence interval using a t distribution instead.

Rule of thumb: Use t when n<32

In [17]:
```
boundst = sp.stats.t.ppf([0.025, 0.975], df=24 , loc=70, scale=6/np.sqrt(25))
print 'Inverse of t(70,s=.19,df=24) evaluated at .025 and .975 are '+str(boundst[
```

```
Inverse of t(70,s=.19,df=24) evaluated at .025 and .975 are 67.523321726, 72.47
6678274
```

Another way to look at statistical inference besides confidence intervals is to look at hypothesis testing

Hypothesis testing comes from the following scientific principle.

In science, we from a testable hypothesis: i.e. a prediction.

1. We assume that the reality is not the prediction, and then find evidence to suggest that this is a bad assumption.
2. From a statistical point of view, we form a null hypothesis (assumed reality). And then we form an alternative hypthesis, which is everything that is not the null hypothesis.

For example, our null hypothesis could be that the average height is not greater than 67 inches.

The alternative hypothesis would be that the average height is greater than 67 inches.

Then, we compute based on our sample (let's use the larger 1000 subject sample), the probability that height is less than 67. We will reject the hypothesis if this probability is less than some threshold (typically .05).

In [18]:
```
print 'The probability that our sample comes from a distribution where our null i
print 'This is less than our threshold of .05, therefore we reject the null hypot
```

```
The probability that our sample comes from a distribution where our null is cor
rect is 1.29840351967e-56
This is less than our threshold of .05, therefore we reject the null hypothesis
to conclude that the average height is greater than 67 inches.
```

In [19]:
```
# Using the t distribution with a sample of 25
print 'The probability that our sample comes from a distribution where our null i
print 'This is less than our threshold of .05, therefore we reject the null hypot
```

```
The probability that our sample comes from a distribution where our null is cor
rect is 0.00982708755829
This is less than our threshold of .05, therefore we reject the null hypothesis
to conclude that the average height is greater than 67 inches.
```

Hypothesis testing with 2 sample means

Think about the following example:

We have 2 advertisements: 1 that makes an emotional appeal, another that makes a rational appeal. The advertisemnts are each displayed on 20 different days. The resulting sales are recorded.

Ad 1: mean sales = $338, standard deviation = $30

Ad 2: mean sales = $313, standard deviation = $50

Which ad was better?

Assumptions

We assume that the effect of each ad on each day is independently and identically distributed (IID), i.e. ad 1 sales follow one distribution, ad 2 sales follow another distribution.

This means, in expectation, the sales on each day with the same ad should be the same (even if the realized sales are different), and come from the same distribution.

If this is satisified, we can use the 2 sample independent t-test. This test basically answers the question: if a random day is drawn for ad 1, and a random day is drawn for ad 2, is the difference in sales statistically different from 0?

2 Independent Sample t-test of equal means

$$H_0 : \mu_1 = \mu_1 \text{ or alternatively: } \mu_1 - \mu_2 = 0$$
$$H_a : \mu_1 \neq \mu_1 \text{ or alternatively: } \mu_1 - \mu_2 \neq 0$$
$$T = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{s_1^2/N_1 + s_2^2/N_2}}$$

Where,
$\bar{X}_1$ and $\bar{X}_2$ are the means of samples 1 and 2,
s_1 and s_2 are sample standard deviations, or equivalently the sample variances are s_1^2 and s_2^2

The test statistic, T, is distributed via a t-distribution with d.f. v. Where,

$$v = \frac{(s_1^2/N_1 + s_2^2/N_2)^2}{(s_1^2/N_1)^2/(N_1 - 1) + (s_2^2/N_2)^2/(N_2 - 1)}$$

This is basically a weighting of each sample's standard deviations with the number of observations to compute a pooled degree of freedom.

We would reject H_0 when $|T| > t_{1-\alpha/2,v}$

$t_{1-\alpha/2,v}$ is the value corresponding to the $1 - \alpha$ percentile of a t-distribution of d.f. v. (You'd look this value up using canned routines/formulas in Excel/a stats package

1. What happens (to T) when $\bar{X}_1$ and $\bar{X}_2$ are close to eachother? (sample means are similar)
2. What happens when s_1 and s_2 are big? (sample standard deviations are big)
3. What happens when N_1 and N_2 are big? (sample sizes are big)

Back to the ad example

$$\bar{X_1} = 338$$
$$\bar{X_2} = 313$$
$$S_1 = 30$$
$$S_2 = 50$$

What is T for the example? (substute the appropriate levels.)

$$
\begin{aligned}
T &= \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{s_1^2/N_1 + s_2^2/N_2}} \\
&= \frac{338 - 313}{\sqrt{30^2/20 + 50^2/20}} \\
&= \frac{25}{\sqrt{900/20 + 2500/20}} \\
&= 1.917
\end{aligned}
$$

Next, compute the degree of freedom, v.

$$
\begin{aligned}
v &= \frac{(s_1^2/N_1 + s_2^2/N_2)^2}{(s_1^2/N_1)^2/(N_1 - 1) + (s_2^2/N_2)^2/(N_2 - 1)} \\
&= \frac{(900/20 + 2500/20)^2}{(900/20)^2/(19) + (2500/20)^2/(19)} \\
&= 31.11
\end{aligned}
$$

Are the 2 ads different in producing sales at the $\alpha = .05$ level? $t_{1-0.05/2,v} = 2.039$

No, we cannot reject the null hypothesis at the .05 level.

What about the .1 level? $t_{1-0.1/2,v} = 1.695$

Yes, we can reject the null hypothesis. Therefore, the 2 ads produce statistically different levels of sales at the $\alpha = 0.1$ level.

What is the cutoff point of α at which the 2 samples are statistically different?

CDF of t-distribution computed at $|T|$ with degree of freedom, v is .9678. This is the cumulative density at test value T. To find the critical α, we must subtract this from 1 (probability in the right tail), and multiply by 2 (probability of both tails). This is .0644.

We reject H_0 for any $\alpha > .0644$.

Relationship between variables

What are some examples of marketing questions involve relationship between variables?

How would you measure these relationships?

First, let's introduce the concept of covariance

Recall, variance was sort of the average distance from the mean of a sample.

Covariance is a generalization of variance. Instead of the average squared distance from a variable's mean, covariance looks at the average joint variation of 2 variables - how jointly different they are from their respective means.

i.e. When height is x units from the mean, how much is weight on average from its mean?

Think about this measure: (Person A's height - average height) X (Person A's weight - average weight)

If there's no relationship between height and weight, what's the average value of this measure?

What if there's a negative relationship? What if there's a positive relationship?

Formally, covariance is **Definition 7, covariance**:

$E[(x - \mu_x)(y - \mu_y)]$, where E is the expectation operator (expected value/ average). This is equivalent to

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$$

.

Note this definition of covariance uses population means. The equivalent sample covariance is:

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Example 2 ... let's say we have the following dataset on height and weight: [(60, 120), (75, 180), (66, 140), (73, 170), (69, 160)] Compute the covariance.

In [20]:
```
data = pd.DataFrame([(60, 120), (75, 180), (66, 140), (73, 170), (69, 160)])
data.columns = ['height', 'weight']
data.to_csv('./example2.csv', index = False)
```

In [21]:
```
data = pd.read_csv('./example2.csv')
print 'The covariance is ' + str(np.cov(data.height, data.weight)[0][1])
```

```
The covariance is 142.0
```

While covariance is good and all...

Doesn't it kind of depend how we are measuring the units? What if we measured height in feet instead of inches, doesn't it dramatically change covariance?

YES...

In [22]:
```
print 'When we convert height to feet, the covariance becomes ' + str(np.cov(data
```

```
When we convert height to feet, the covariance becomes 11.8333333333
```

So what do we do to "normalize" this metric?

We can scale the covariance by the standard deviation! Note this is the correlation... Formally, correlation between x and y is:

Definition 8, correlation

$$r_{x,y} = \frac{cov_{x,y}}{sd_x sd_y} = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n-1}\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

This is the linear correlation coefficient. (Note sample and population are the same since the normalization of 1/n or 1/(n-1) are both canceled out in the division.)

In [23]:
```
print 'The correlation is ' + str(np.corrcoef(data.height, data.weight)[0][1])
print 'When we convert height to feet, the covariance becomes ' + str(np.corrcoef
print 'They are identical! Importantly, correlations are bound between -1 and 1.'
```

```
The correlation is 0.992400546838
When we convert height to feet, the covariance becomes 0.992400546838
They are identical! Importantly, correlations are bound between -1 and 1.
```

Note 2 very important things about correlations

1. It is only a measure of the degree of association of any linear relationship. Nonlinear relationships are not necessarily accurately - or at all - captured.
2. It does not measure the "slope" of a linear relationship. i.e. Correlations tell you that height and weight are strongly related, but not how they are related in terms of average lbs per inch in height. We don't know if for every inch in height there's on average 1lb or 2lb or 3lb...etc. gain in weight. Any of these relationships can have the identical correlation. Think of correlation as the robustness of the linear relationship between 2 variables.

So, how do we capture the magnitude and direction of the relationship?

Regression

We first start with a simple regression model, one explanatory variable (independent variable) and one dependent variable. Let's say income is our explanatory variable and house size is our dependent variable.

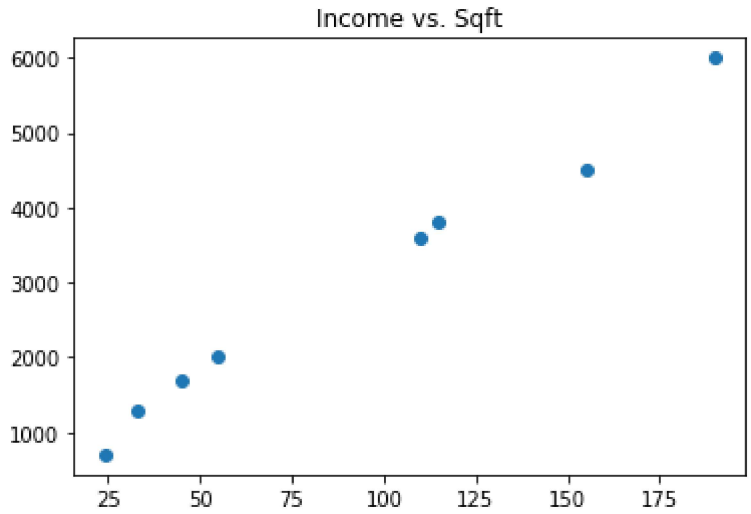
Let's say we have the following data of (income in 1000's of dollars, house size in sqft).

[(45, 1700), (55, 2000), (110, 3600), (115, 3800), (155, 4500), (190, 6000), (33, 1300), (24, 700)]

In [24]:
```
house = pd.DataFrame([(45, 1700), (55, 2000), (110, 3600), (115, 3800), (155, 450
house.columns = ['income', 'sqft']
house.to_csv('./example3.csv', index = False)
```

In [28]:
```
house = pd.read_csv('./example3.csv')
plt.scatter(house.income, house.sqft)
plt.title('Income vs. Sqft')
```

Out[28]:
```
Text(0.5,1,u'Income vs. Sqft')
```



It looks like we can roughly draw a line through these points.

But which line should we draw?

Linear regression tries to fit a line through these points that minimizes the distance from these points to the line in the "y" direction. In regression term, we want to minimize the residual sum of squares.

First, a generic function for a line can be written as (any line can be written as this):

$$y = \beta_0 + \beta_1 x$$

Where β_0 and β_1 are some parameters and x is a variable that can take on a range of values. The relationship between y and x are characterized by β_1. β_1 means that **for every unit of** x**,** y **changes by** β_1 **units**.

Linear regression basically poses the problem:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Deriving OLS estimator of univariate regression

Restating the above optimization problem:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Let's first solve for β_0. Recall, to find minima/maxima, we make sure the function is convex/concave in the desired parameter and set the first derivative w.r.t. that parameter equal to 0.

We begin by defining the regression error (aka residual). This is the difference between the regression line and the actual data point.

$$\epsilon = (y_i - (\beta_0 + \beta_1 x_i))$$

The loss function L can be defined as:

$$L(\cdot) = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Let's take the derivative of both sides wrt β_0, assuming we know everything else.

$$\frac{dL(\cdot)}{\beta_0} = \frac{d \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2}{d\beta_0}$$

$$\frac{dL(\cdot)}{\beta_0} = \frac{d \sum_{i=1}^{n} (y_i^2 - 2\beta_1 x_i y_i - 2\beta_0 y_i + \beta_1^2 x_i^2 + 2\beta_0\beta_1 x_i + \beta_0^2)}{d\beta_0}$$

$$\frac{dL(\cdot)}{\beta_0} = \sum_{i=1}^{n} (-2y_i + 2\beta_1 x_i + 2\beta_0)$$

We can set this derivative to equal 0 to find the minima

$$0 = \sum_{i=1}^{n} (-2y_i + 2\beta_1 x_i + 2\beta_0)$$

Now, let's solve for β_0

$$0 = \sum_{i=1}^{n} (-y_i + \beta_1 x_i) + \sum_{i=1}^{n} (\beta_0)$$

$$0 = \sum_{i=1}^{n} (-y_i + \beta_1 x_i) + n(\beta_0)$$

$$-n(\beta_0) = \sum_{i=1}^{n} (-y_i + \beta_1 x_i)$$

$$\beta_0 = \frac{\sum_{i=1}^{n} (y_i - \beta_1 x_i)}{n} = \frac{\sum_{i=1}^{n} (y_i) - \beta_1 \sum_{i=1}^{n} x_i}{n} = \bar{y} - \beta_1 \bar{x}$$

Great, we've now computed the estimator for β_0, assuming we know what β_1 - or more precisely, what the estimator for β_1 is. Let's repeat the derivative exercise for β_1

$$0 = \frac{d \sum_{i=1}^{n} (y_i^2 - 2\beta_1 x_i y_i - 2\beta_0 y_i + \beta_1^2 x_i^2 + 2\beta_0\beta_1 x_i + \beta_0^2)}{d\beta_1}$$

$$0 = \sum_{i=1}^{n} (-2x_i y_i - 2\beta_1 x_i^2 + 2\beta_0 x_i) = \sum_{i=1}^{n} (-x_i y_i + \beta_1 x_i^2 + \beta_0 x_i)$$

$$0 = \beta_0 \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} (x_i y_i) + \beta_1 \sum_{i=1}^{n} x_i^2$$

$$0 = (\bar{y} - \beta_1 \bar{x}) \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} (x_i y_i) + \beta_1 \sum_{i=1}^{n} x_i^2$$

$$0 = \bar{y} \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} (x_i y_i) + \beta_1 (\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i)$$

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i y_i) - \bar{y} \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

We've now solved for β_1, but we can rearrange the terms taking advantage of the fact that $\sum_{i=1}^{n}(x_i) = n\bar{x}$ and $\sum_{i=1}^{n}(y_i) = n\bar{y}$ to make a more intuitive expression.

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i y_i) - n\bar{y}\bar{x}}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

We then use the most common statistician's trick, add and subtract by the same thing...

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i y_i) - n\bar{y}\bar{x} - n\bar{y}\bar{x} + n\bar{y}\bar{x}}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

Now we can rewrite the product of the average terms as the sum of the average times the individual term, or as the sum of the product of the average terms.... i.e.
$n\bar{y}\bar{x} = \sum_{i=1}^{n}(x_i\bar{y}) = \sum_{i=1}^{n}(y_i\bar{x}) = \sum_{i=1}^{n}(\bar{y}\bar{x})$

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i y_i) - \sum_{i=1}^{n}(x_i\bar{y}) - \sum_{i=1}^{n}(y_i\bar{x}) + \sum_{i=1}^{n}(\bar{y}\bar{x})}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

$$\beta_1 = \frac{\sum_{i=1}^{n} x_i y_i - x_i\bar{y} - y_i\bar{x} + \bar{y}\bar{x}}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

We can factor the expression to get:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i}$$

Now, we can deal with the denominator in a similar way:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i^2) - n\bar{x}\bar{x} - n\bar{x}\bar{x} + n\bar{x}\bar{x}}$$

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} x_i^2 - \bar{x}x_i - x_i\bar{x} + \bar{x}\bar{x}}$$

Factorize to get:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

So, putting all together, we have

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

and

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Well, technically, these are not the true β's but rather estimates, or $\hat{\beta}$'s

So...

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

and

$$\hat{\beta_0} = \bar{y} - \hat{\beta_1} \bar{x}$$

Phew...

Let's review what we've accomplished. We said we wanted to fit the line $y = \beta_0 + \beta_1 x$ to a bunch of points $\{(x_i, y_i)\}$, $\forall i \in 1, 2, \ldots, n$. To do so, we want to minimize the squared errors, ϵ_i^2.

We do this by doing some calculus and algebra. **We end up with the expressions:**

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

and

$$\hat{\beta_0} = \bar{y} - \hat{\beta_1} \bar{x}$$

.

SO... in practice, we can figure out $\hat{\beta_1}$ first, and then substitute it into the equation for $\hat{\beta_0}$.

What does the equation for $\hat{\beta_1}$ actually say?

Notice, $\hat{\beta}_1 is just the covariance divided by the variance of the independent variable.

Now, let's look at the expression for $\hat{\beta_0}$. Notice that we have just the average of the dependent variable, y, minus the slope times the average of the independent variable, x.

In essence, we first computed the slope of the line that should fit, now we are moving the line up and down the y axis to find where the error at the average values of x and y is 0.

In [29]:
```
from statsmodels.regression.linear_model import OLS ## Let's use ordinary least s
house['constant'] = 1 # to do this we need a constant term
results = OLS(house.sqft, house[['income', 'constant']]).fit()
house['sqft_fit'] = np.dot(house[['income', 'constant']],results.params)
results.summary() # Look at the coef column
```

Out[29]:

OLS Regression Results

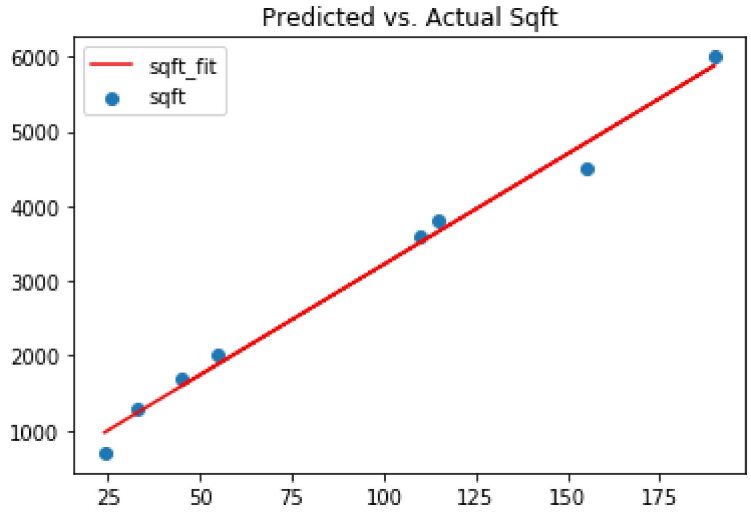
Dep. Variable:	sqft	**R-squared:**	0.989
Model:	OLS	**Adj. R-squared:**	0.987
Method:	Least Squares	**F-statistic:**	524.2
Date:	Wed, 24 Jan 2018	**Prob (F-statistic):**	4.55e-07
Time:	13:55:52	**Log-Likelihood:**	-52.929
No. Observations:	8	**AIC:**	109.9
Df Residuals:	6	**BIC:**	110.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
income	29.5817	1.292	22.895	0.000	26.420	32.743
constant	261.7598	138.683	1.887	0.108	-77.584	601.104

Omnibus:	3.702	**Durbin-Watson:**	2.170
Prob(Omnibus):	0.157	**Jarque-Bera (JB):**	1.860
Skew:	-1.147	**Prob(JB):**	0.395
Kurtosis:	2.438	**Cond. No.**	202.

In [30]:
```
plt.scatter(house.income, house.sqft)
plt.plot(house.income, house.sqft_fit, c='red')
plt.title('Predicted vs. Actual Sqft')
plt.legend(loc=2)
```

Out[30]: <matplotlib.legend.Legend at 0x11a887b8>



Some observations

- Look at the R^2 value in the regression output, this is a measure of fit. Coincidentally, this is the same value as the square of the correlation coefficient (see below).
- Note that the line of best fit can be used to predict a sqft for any arbitrary income size (though we don't know how good the out of sample predictions are).
- In the regression table, there is a column labeled "t." This is the t statistic associated with the estimated coefficient and the coefficient's standard error. The estimated coefficient is distributed via a t-distribution with standard deviation = standard error from the regression output.
- The t test is based on the hypothesis: Does including the variable x improve the regression fit? In the case of a simple regression model (with 1 variable), it means: does the variable improve our prediction of the dependent variable above and beyond just a simple average of the dependent variables.

In [31]:
```
np.square(house[['income', 'sqft']].corr())
```

Out[31]:

	income	sqft
income	1.000000	0.988683
sqft	0.988683	1.000000

In other words is the model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ better than the model $y_i = \beta_0 + \epsilon_i$? To do this, we need to compare the resiudals (ϵ's) of each model to eachother. We compute a standard error based on these residuals, the sample size, and the number of variables in each model. The formula

for computing the standard error is:

$$SE(\beta_1) = \sqrt{\frac{\sum_{i=1}^{n}(\epsilon_i^2)/(n-2)}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

t-statistic:

$$\frac{\hat{\beta}_i - v}{SE(\hat{\beta}_i)}$$

where v is some hypothesis you want to test... the t statistic used by the regression assumes the null hypothesis is $\hat{\beta} = v = 0$.

1. To interpret the t-stat, we can just look at the corresponding p-value. This is the probability that the null hypothesis ($\beta_i = 0, \forall i$).
2. A philosophical question. How do we specify a regression equation? We want the left hand side (IN this case sqft) to be "caused" by the right hand side variables (in this case income). Does it make sense in this situation? Seems like there should here and that income causes house size is more likely than house size causes income.

Building on the "causal" interpretation

What does it mean for X to cause Y?

Post hoc, ergo propter hoc. "After this, therefore because of this"

This is one type of causality known in economics as "Granger causality." This type of causality simply suggests because X happens before Y, it means that X causes Y.

This isn't true causality. We eat a lot of turkey every year one month before Christmas. But it does not mean that eating turkey causes Christmas...

Causality, therefore, is more nuanced than just timing of events. It must be accompanied by exclusion of other potential causes and also by theoretical rationale.

Here are some empirical observations:

1. Areas with more police have high crime rates . Policing causes crime?
2. The death rate in hospitals is really high. Hospitals are bad for health?
3. You watched a lot of TV last night and passed the test today. Watching TV causes better grades?

Expanding on the simple linear regression.

Obviously, the world is not as simple as one variable causing another. There's generally a confluence of variables at play. For example, your grade in a class does not just depend on effort, it could depend on how many other classes you're taking, whether or not you work, your family situation, your educational background, etc...

How do we take account of all of these things when trying to establish relationships between an outcome and its antecedents?

What if, instead of fitting the line of best fit $y = \beta_0 + \beta_1 x + \epsilon$, we fit a "hyperplane" of best fit: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m + \epsilon$?

Well, it turns out these two things are not that much different. Suppose your independent variabels data looks like:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ \vdots & \ddots & x_{2m} & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

Where you have n rows (observations) and m columns (variables). Your dependent variable looks like this:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

The equivalent matrix operation looks like:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ \vdots & \ddots & x_{2m} & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

In matrix notation, we would write this as:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \epsilon$$

We will skip the matrix derivation, but it's pretty much the same idea as the derivation above for the bivariate case.

We will end up with one formula for the vector β's, $\mathbf{b}$.

$$\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Where $'$ means Transpose and $^{-1}$ is the inverse. What $(\mathbf{X}'\mathbf{X})^{-1}$ does is it computes the matrix that makes would make $(\mathbf{X}'\mathbf{X})$ an identity matrix (diagonal elements are 1, 0 everywhere else). $(\mathbf{X}'\mathbf{X})$ itself is like the covariance matrix of independent variables without normalizing by sample size. $(\mathbf{X}'\mathbf{y})$ is like the covariance matrix of the dependent and the independent variables without normalizing by sample size.

In this conceptual way, the bivariate and multivariate cases are almost identical.

Example 4. Now suppose we also observe family size as a variable. The corresponding family size values are: [2, 2, 3, 3, 4, 3, 2, 3]

In [45]:
```
house['fam_size'] = [1, 2, 3, 2, 5, 6, 2, 1]
house.to_csv('./example4.csv', index = False)
house
```

Out[45]:

	income	sqft	constant	sqft_fit	fam_size	sqft_fit2
0	45	1700	1	1592.937914	1	1688.673925
1	55	2000	1	1888.755262	2	1885.614844
2	110	3600	1	3515.750679	3	3562.474043
3	115	3800	1	3663.659353	2	3858.839218
4	155	4500	1	4846.928747	5	4778.532703
5	190	6000	1	5882.289467	6	5797.650444
6	33	1300	1	1237.957096	2	1162.099240
7	24	700	1	971.721482	1	866.115584

In [46]:
```
house = pd.read_csv('./example4.csv')
results2 = OLS(house.sqft, house[['income','fam_size', 'constant']]).fit()
house['sqft_fit2'] = np.dot(house[['income','fam_size', 'constant']],results2.par
results2.summary() # Look at the coef column
```

Out[46]:

OLS Regression Results

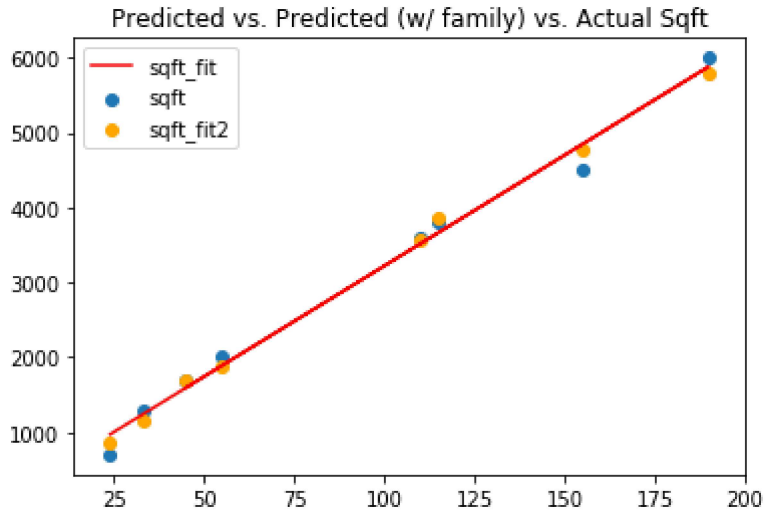
Dep. Variable:	sqft	**R-squared:**	0.990
Model:	OLS	**Adj. R-squared:**	0.986
Method:	Least Squares	**F-statistic:**	245.6
Date:	Wed, 24 Jan 2018	**Prob (F-statistic):**	1.02e-05
Time:	13:59:38	**Log-Likelihood:**	-52.465
No. Observations:	8	**AIC:**	110.9
Df Residuals:	5	**BIC:**	111.2
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
income	32.0474	3.414	9.386	0.000	23.271	40.824
fam_size	-89.2838	113.783	-0.785	0.468	-381.772	203.205
constant	283.2222	145.935	1.941	0.110	-91.916	658.360

Omnibus:	2.352	**Durbin-Watson:**	2.110
Prob(Omnibus):	0.309	**Jarque-Bera (JB):**	1.354
Skew:	-0.921	**Prob(JB):**	0.508
Kurtosis:	2.180	**Cond. No.**	210.

In [44]:
```
plt.scatter(house.income, house.sqft)
plt.plot(house.income, house.sqft_fit, c='red')
plt.scatter(house.income, house.sqft_fit2, c='orange')
plt.title('Predicted vs. Predicted (w/ family) vs. Actual Sqft')
plt.legend(loc=2)
```

Out[44]: <matplotlib.legend.Legend at 0x11dc1c50>



A note on factor variables

So far, our explanatory variables have been "continuous." That is, the magnitude of the value has meaning.

How do we explain the relationship between sales and choice of advertisement? There's no "continuous" measure of advertisement. Say we had 4 advertisements that are chosen at random to be displayed during each sales period. What is the effect of each ad on sales? We can't just run the regression:

$$Sales = \beta_0 + \beta_1 \times Ad + \epsilon$$

Ad #2 is not "more of an ad" than Ad #1

What we need to do is to create 4 variables: Ad_1, Ad_2, Ad_3, Ad_4. Where each variable is 0, when its corresponding ad is not displayed, and 1 when it is

This is called a dummy variable.

However, note that if we have to display one and only one ad, then we can drop any one of the Ad variables and still capture all the information in the data.

For example, if we drop Ad_1, $Ad_1 = 1$ is equivalent to $Ad_2 = Ad_3 = Ad_4$. Therefore, we can run the following regression:

$$Sales = \beta_0 + \beta_1 \times Ad_2 + \beta_2 \times Ad_3 + \beta_3 \times Ad_4 + \epsilon$$

In this specification, β_0 captures the effect of Ad_1, $\beta_1, \beta_2, \beta_3$ captures the relative effects of ads 1, 2, and 3 respectively. Relative = above (or below for negative) the impact of Ad 1.