Sampling distributions

In this notebook we will use the NHANES data to explore the sampling distributions of statistics. This is a somewhat more conceptual topic than we have encountered before when working with the NHANES data. Some of the illustrations covered below are not things that would typically be done in a data analysis, but rather are intended to clarify some important concepts about statistical methods. However even these methods have their uses in practice, for example the approaches demostrated below can be used to demonstrate how a statistic behaves in a "nonstandard" setting, or to assess how much data is needed to answer a question confidently.

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

In the cell below, we read the data from a local file:

In [2]:
```
da = pd.read_csv("nhanes_2015_2016.csv")
```

In [10]:
```
print(da)
```

```
       SEQN  ALQ101  ALQ110  ALQ130  SMQ020  RIAGENDR  RIDAGEYR  RIDRETH1  \
0     83732     1.0     NaN     1.0       1         1        62         3
1     83733     1.0     NaN     6.0       1         1        53         3
2     83734     1.0     NaN     NaN       1         1        78         3
3     83735     2.0     1.0     1.0       2         2        56         3
4     83736     2.0     1.0     1.0       2         2        42         4
5     83737     2.0     2.0     NaN       2         2        72         1
6     83741     1.0     NaN     8.0       1         1        22         4
7     83742     1.0     NaN     1.0       2         2        32         1
8     83743     NaN     NaN     NaN       2         1        18         5
9     83744     1.0     NaN     NaN       2         1        56         4
10    83747     1.0     NaN     1.0       1         1        46         3
11    83750     1.0     NaN     3.0       1         1        45         5
12    83752     1.0     NaN     2.0       1         2        30         2
13    83754     2.0     1.0     1.0       2         2        67         2
14    83755     1.0     NaN     3.0       2         1        67         4
15    83757     1.0     NaN     1.0       2         2        57         2
16    83759     2.0     2.0     NaN       2         2        19         1
17    83761     1.0     NaN     1.0       2         2        24         5
18    83762     NaN     NaN     NaN       1         2        27         4
19    83767     2.0     2.0     NaN       2         2        54         5
20    83769     1.0     NaN     2.0       2         1        49         5
21    83773     2.0     2.0     NaN       2         2        80         3
22    83775     2.0     1.0     NaN       1         2        69         2
23    83776     NaN     NaN     NaN       2         2        58         1
24    83777     2.0     2.0     NaN       2         1        56         5
25    83781     1.0     NaN     3.0       2         2        27         4
26    83784     1.0     NaN     4.0       1         1        22         2
27    83785     2.0     1.0     1.0       1         2        60         2
28    83786     NaN     NaN     NaN       2         1        51         4
29    83787     2.0     2.0     NaN       2         2        68         1
...     ...     ...     ...     ...     ...       ...       ...       ...
5705  93645     2.0     1.0     NaN       1         1        80         3
5706  93652     1.0     NaN     NaN       1         1        72         2
5707  93653     1.0     NaN     3.0       2         2        25         3
5708  93654     2.0     2.0     NaN       2         2        29         5
5709  93655     1.0     NaN     NaN       1         1        38         3
5710  93656     2.0     2.0     NaN       2         2        75         1
5711  93659     1.0     NaN     1.0       1         1        62         1
5712  93661     1.0     NaN     2.0       2         2        27         2
5713  93663     1.0     NaN     4.0       2         1        43         1
5714  93664     2.0     1.0     2.0       2         1        39         1
5715  93665     NaN     NaN     NaN       2         2        34         3
5716  93668     1.0     NaN     NaN       1         2        73         1
5717  93670     1.0     NaN     3.0       1         1        32         3
5718  93671     1.0     NaN     3.0       2         1        45         4
5719  93672     2.0     2.0     NaN       1         2        63         2
5720  93675     1.0     NaN     1.0       2         1        38         5
5721  93676     1.0     NaN     2.0       2         2        35         4
5722  93677     1.0     NaN     1.0       2         2        34         3
5723  93679     2.0     1.0     NaN       1         2        72         4
5724  93682     NaN     NaN     NaN       2         2        41         5
5725  93684     2.0     2.0     NaN       2         1        34         4
5726  93685     1.0     NaN     2.0       1         1        53         1
5727  93689     2.0     1.0     NaN       2         2        69         1
5728  93690     1.0     NaN     3.0       2         1        32         2
5729  93691     2.0     2.0     NaN       2         1        25         5
5730  93695     2.0     2.0     NaN       1         2        76         3
5731  93696     2.0     2.0     NaN       2         1        26         3
5732  93697     1.0     NaN     1.0       1         2        80         3
5733  93700     NaN     NaN     NaN       1         1        35         3
5734  93702     1.0     NaN     2.0       2         2        24         3

      DMDCITZN  DMDEDUC2  ...  BPXSY2  BPXDI2  BMXWT  BMXHT  BMXBMI  BMXLEG  \
0          1.0       5.0  ...   124.0    64.0   94.8  184.5    27.8    43.3
1          2.0       3.0  ...   140.0    88.0   90.4  171.4    30.8    38.0
2          1.0       3.0  ...   132.0    44.0   83.4  170.1    28.8    35.6
3          1.0       5.0  ...   134.0    68.0  109.8  160.9    42.4    38.5
4          1.0       4.0  ...   114.0    54.0   55.2  164.9    20.3    37.4
```

Sampling distribution of the mean

Sampling distributions describe how the value of a statistic computed from data varies when repeated samples of data are obtained. This can be explored mathematically, or by using a computer to simulate data repeatedly from a hypothetical population. When working with non-simulated data (i.e. from a study like NHANES), we usually do not have the ability to explicitly obtain an "independent copy" of the sample to actually "see" its sampling distribution. However we can "subsample" from a dataset to mimic what would happen if we were to sample repeatedly from the population that produced it. A subsample is a random sample drawn from a larger data set, containing only a fraction of its observations.

In the notebook cell below, we repeatedly subsample two disjoint subsets of size 100 from the NHANES data, calculate the mean systolic blood pressure within each of these two subsets, then calculate the difference between these two means. This difference reflects the "chance variation" that would have been observed if the NHANES project had only had the resources to sample 100 participants for their study. By sampling two subsets of size 100 and comparing the resulting calculated means, we can see how the findings of two researchers independently studying the same population might differ from each other by chance.

The subsampling process described above is then repeated 1000 times, so we can see how two samples of size 100 from the NHANES population tend to differ in terms of their mean systolic blood pressure.

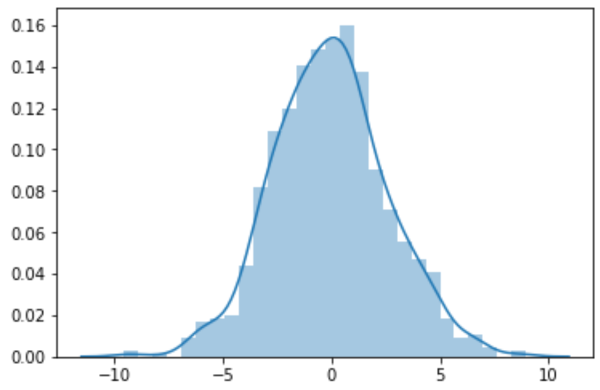
```
In [3]: m = 100 # Subsample size
        sbp_diff = [] # Storage for our subsample mean differences

        for i in range(1000):
            dx = da.sample(2*m)  # We need two subsamples of size m
            dx1 = dx.iloc[0:m, :]  # First subsample
            dx2 = dx.iloc[m:, :]  # Second subsample
            sbp_diff.append(dx1.BPXSY1.mean() - dx2.BPXSY1.mean())  # The difference o
        f mean BPXSY1 values
```

Next we look at the histogram of the 1000 mean differences generated above. We see that they typically fall between negative 5 and positive 5. This means that two researchers independently studying blood pressure in the same population may by chance obtain results that are up to around 5 units different, but are quite unlikely to obtain results that are by chance more than 10 units different.

```
In [4]: sns.distplot(sbp_diff)

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6482b3d7f0>
```



In the cell below, we look at some numerical statistics of the histogram plotted above.

```
In [5]: pd.Series(sbp_diff).describe()
```

```
Out[5]: count    1000.000000
        mean       -0.052593
        std         2.630826
        min        -9.559397
        25%        -1.847483
        50%        -0.058951
        75%         1.491945
        max         8.972518
        dtype: float64
```

The results shown above indicate that the mean systolic blood pressures calculated for two samples each with 100 people will typically differ by around 2.8 mm/Hg (the standard deviation), and will rarely differ by more than 5 mm/Hg.

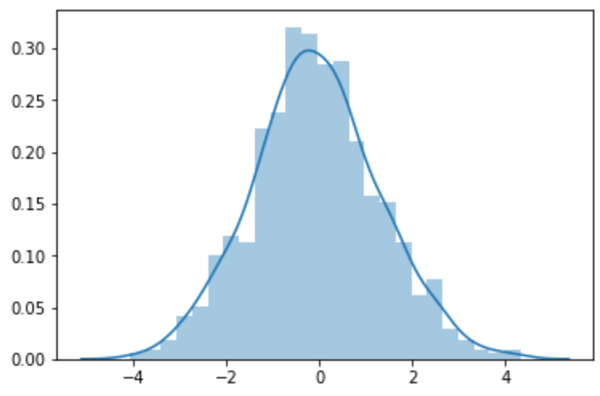
The sample size is a major determinant of the chance fluctuations in any statistic. Above we used samples of size 100, below we perform the same analysis using samples of size 400.

```
In [8]: m = 400  # Change the sample size, everything else below is unchanged from the
        cells above
        sbp_diff = []

        for i in range(1000):
            dx = da.sample(2*m)
            dx1 = dx.iloc[0:m, :]
            dx2 = dx.iloc[m:, :]
            sbp_diff.append(dx1.BPXSY1.mean() - dx2.BPXSY1.mean())

        sns.distplot(sbp_diff)
        pd.Series(sbp_diff).describe()
```

```
Out[8]: count    1000.000000
        mean       -0.067555
        std         1.363325
        min        -4.083418
        25%        -0.950829
        50%        -0.115126
        75%         0.795898
        max         4.337680
        dtype: float64
```



We see that with samples of size 400, the standard deviation is around 1.38, which is close to half of what it was when we used samples of size 100. The smaller standard deviation indicates that the chance flucatations in the mean systolic blood pressure are smaller when we have a larger sample size. This implies that we are able to estimate the population mean systolic blood pressure with more precision when we have samples of size 400 compared to when we have samples of size 100.

Importantly, increasing the sample size by a factor of 4 (from 100 to 400) led to a reduction of the standard deviation by a factor of 2. This scaling behavior is very common in statistics -- increasing the sample size by a factor of K leads to a reduction in the standard deviation by a factor of sqrt(K). Thus, for a factor of 4 increase in the sample size we see a factor of 2 reduction in standard deviation. Similarly, for a factor of 9 increase in the sample size we would typically see a factor of 3 reduction in the standard deviation.

Sampling distribution of the correlation coefficient

As discussed in the lectures, many statistics that are more complex than the sample mean behave similarly to the mean in terms of their sampling behavior. Below we modify the subsampling analysis from above to consider the Pearson correlation coefficient between systolic and diastolic blood pressure. Note that the standard deviation still drops by approximately a factor of 2 when the sample size increases by a factor of four (from 100 to 400).

This short Python program uses nested `for` loops. The outer loop manages the sample size, and the inner loop obtains 1000 subsamples at a given sample size, calculates correlation coefficients for two subsamples, and records their difference.

In [7]:
```
for m in 100, 400:  # m is the subsample size
    sbp_diff = [] # calculate correlation coefficients from independent sample
s of size m
    for i in range(1000):
        dx = da.sample(2*m)
        dx1 = dx.iloc[0:m, :]
        dx2 = dx.iloc[m:, :]
        r1 = np.corrcoef(dx1.loc[:, ["BPXSY1", "BPXDI1"]].dropna().T)
        r2 = np.corrcoef(dx2.loc[:, ["BPXSY1", "BPXDI1"]].dropna().T)
        sbp_diff.append(r1 - r2)
    print("m=%d" % m, np.std(sbp_diff), np.sqrt(2 / m))
```

```
m=100 0.133980647532 0.141421356237
m=400 0.0629321887874 0.0707106781187
```

The simulation above shows that when the subsample size increases from 100 to 400 (a factor of 4), the standard deviation of the difference between two correlation coefficients decreases by roughly a factor of 2. The mathematical expression sqrt(2 / m) is an approximation to this standard deviation that can be computed without access to any data.

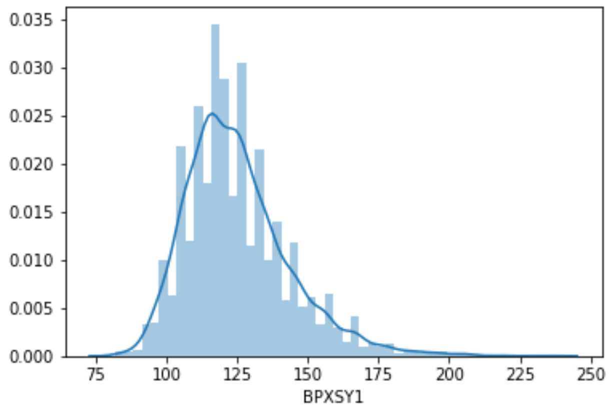
The shape of sampling distributions

Above we focused on the magnitude of the difference between a statistic calculated on two independent samples from a population. Here we focus instead on the shape of the distribution of statistics calculated on subsamples. As discussed in the lectures, the central limit theorem implies that many (but not all) statistics have approximately normal sampling distributions, even if the underlying data are not close to being normally distributed.

We will illustrate this phenomenon using the systolic blood pressure data from the NHANES study. First we use a histogram to look at the distribution of individual systolic blood pressure values. Note that it is somewhat right-skewed.

```
In [8]: sns.distplot(da.BPXSY1.dropna())
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f14ac8e2390>
```

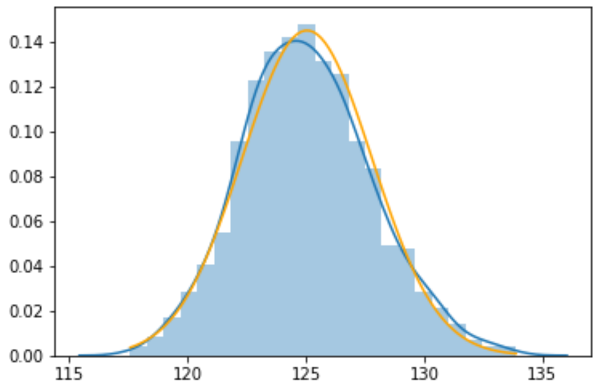


Next we calculate 1000 sample means from 1000 subsamples of size 50 and inspect their distribution.

```
In [9]: m = 50
        sbp_mean = []
        for i in range(1000):
            dx = da.sample(m)
            sbp_mean.append(dx.BPXSY1.dropna().mean())
        sns.distplot(sbp_mean)

        # The lines below plot the density of a normal approximation to the data gener
        ated above
        x = np.linspace(np.min(sbp_mean), np.max(sbp_mean), 100)
        from scipy.stats.distributions import norm
        v = dx.BPXSY1.dropna()
        y = norm.pdf(x, np.mean(sbp_mean), np.std(sbp_mean))
        plt.plot(x, y, color='orange')
```

```
Out[9]: [<matplotlib.lines.Line2D at 0x7f14ac6fa518>]
```



The plots above show that while the distribution of individual systolic blood pressure measures is somewhat skewed to the right, the distribution of means of size 50 is approximately symmetric. The distribution of means is also approximately normal, as shown by the orange curve, which is the best-fitting normal approximation to the data.