Statistical Distributions - Introduction

Introduction

In this section, you'll learn about different probability distributions!

It's all Stats!

You've already seen the value of descriptive statistics when doing exploratory data analysis. In this section, we're going to dive deeper into a range of statistical concepts. We're going to start by looking at discrete and continuous distributions and how you can use stem and leaf plots for visualizing distributions.

We're then going to look at a range of techniques for representing distributions - the Probability Mass Function, the Cumulative Distribution Function, and the Probability Density Function.

We're then going to dig a little deeper into the Normal/Gaussian distribution and the Standard Normal Distribution, and we'll introduce the use cases for z-tables and p-values for describing statistical significance. We'll discuss the "one-sample z test", the most basic type of hypothesis test, before introducing the concepts of skewness and kurtosis that can be used to quantify how "un-normal" a given distribution is.

In the Appendix to this Module, we'll introduce some additional distribution functions, like the uniform, Poisson, and exponential distributions, and use them to solve practical problems.

Summary

In this section, we're going to take a deeper dive into a range of foundational statistical concepts that we'll need as we start to dig into machine learning later in the course.

Statistical Distributions and Their Use Cases

Introduction

As a data scientist, you'll often have to work with statistical distributions. This includes selecting which distribution is most representative of a given set of data. A typical use case includes A/B testing, where understanding the process that generated the data is important. You can think of distributions in relation to statistical analysis as to data structures to computer programming.

There are an enormous amount of distributions out there, but you'll see about a handful of distributions that can represent the vast majority of situations you'll come across. In the upcoming series of lessons, you'll look at ways to analyze common distributions you will encounter most frequently.

Objectives

You will be able to:

- Define statistical distributions
- Differentiate between discrete and continuous distributions
- List the common distributions and their use cases

What is a Statistical Distribution?

A statistical distribution is a representation of the frequencies of potential events or the percentage of time each event occurs.

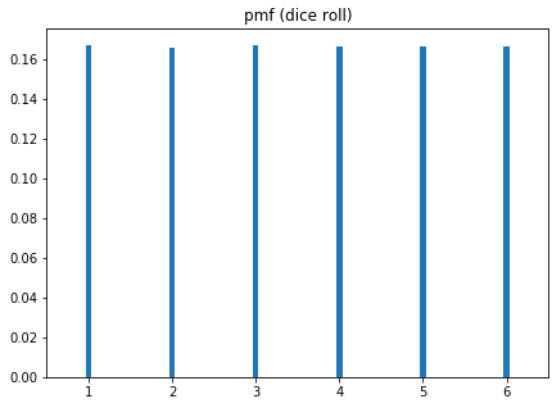
This may feel pretty vague which is why we'll use two examples to clarify this concept.

Rolling a Dice Distribution

Let's think back about our example rolling a dice. You know that when rolling dice once, you will obtain a number between 1 and 6, with each outcome to be as likely, as denoted in this table:

outcome	1	2	3	4	5	6
probability	1/6	1/6	1/6	1/6	1/6	1/6

You can also represent this graphically as follows:

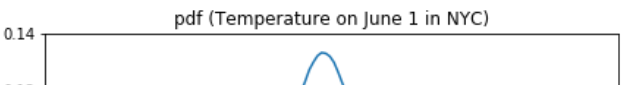


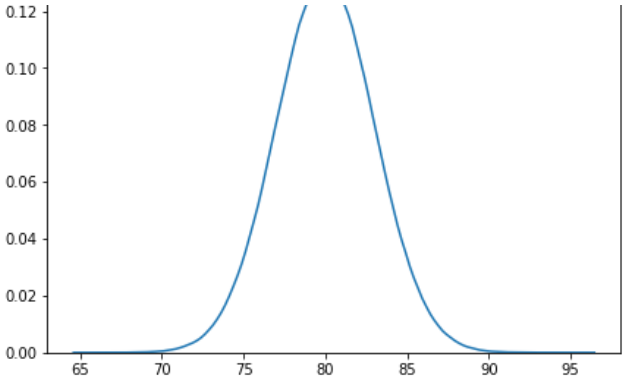
Note how, with a fair die, the chance of throwing each number is *exactly* 1/6 (or 0.1666). The number of outcomes is finite and the outcome is a set of values. In this case, you are dealing with a **discrete distribution**.

Weather Distribution

Let's look at another situation. Imagine we want to think of the distribution of the temperature in New York on June 1st. Thinking about this, you could say that the temperature would generally range between 65 and 95 Degrees (more extreme values would be exceptional), with the average around 80 Degrees Fahrenheit.

A potential distribution looks like this:





Note that instead of bars, which we had for the dice example, we have *continuous* lines here. Our distribution is a **continuous distribution** because temperature is a continuous value (we can have a temperature of 80 degrees, of 80.5 degrees, of 80.0034 degrees, etc.).

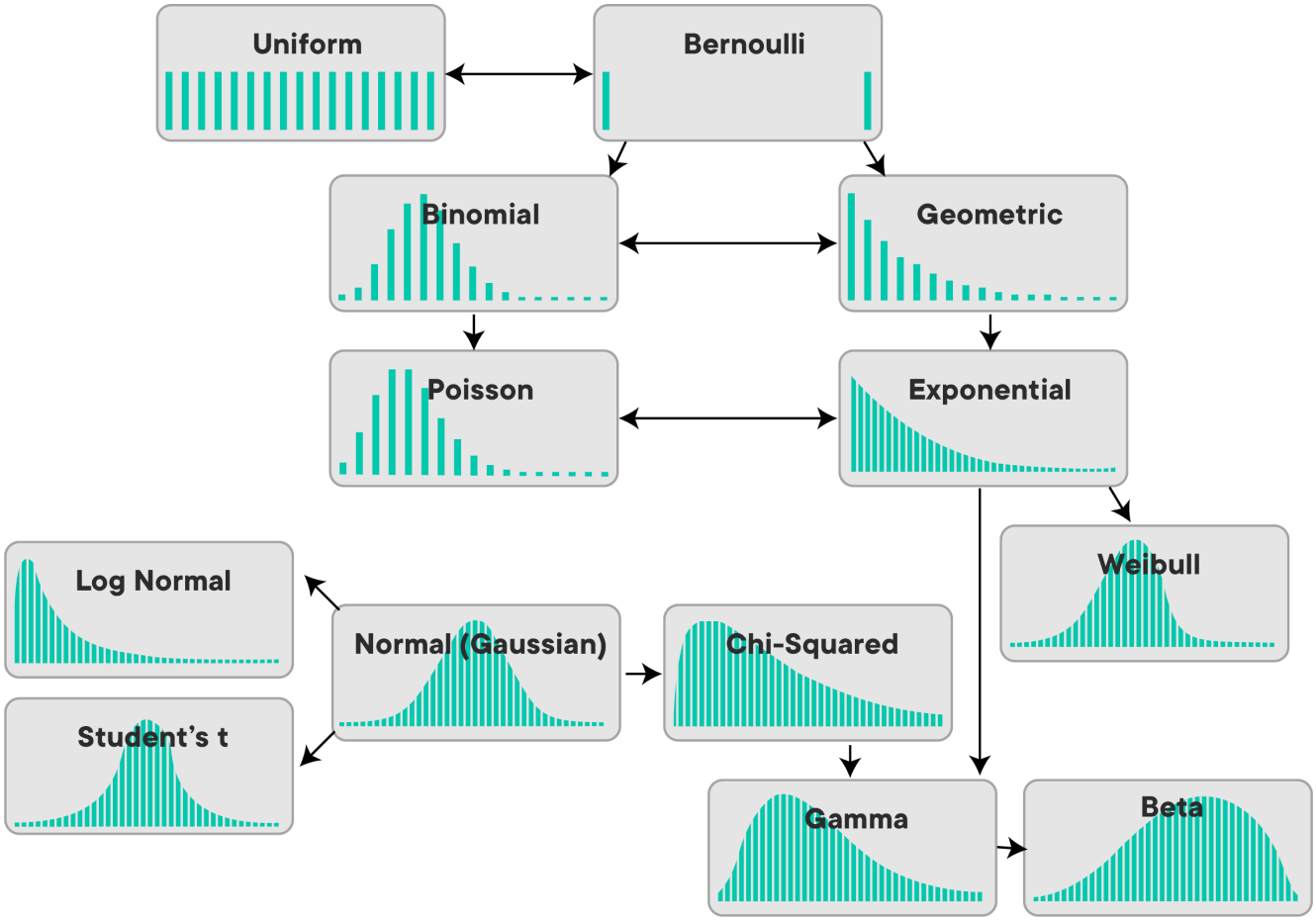
Discrete vs Continuous Distributions

When dealing with **discrete** data you use a **Probability Mass Function (PMF)** (as in our dice example). When dealing with **continuous** data, you use a **Probability Density Function (PDF)** (see our weather example).

Based on the variation of their attributes, data distributions can take many shapes and forms. In the next few lessons, you'll learn how to describe data distributions. Very often, distributions are described using their statistical mean (or **expected value**) and variance of the data, but this is not always the case. You'll see more on this in the next few lessons.

Common Distributions

In this image, you can see the general shapes of some common distributions. The horizontal axis in each chart represents the set of possible numeric outcomes. The vertical axis describes the probability of the respective outcomes.



You'll get a more in-depth overview of some important distributions in the next few lessons, but to give you an initial idea of some applications, we'll give you a quick overview below. Let's quickly talk about some common distributions and their use cases below.

Examples of Discrete Distributions

The Bernoulli Distribution

The Bernoulli distribution represents the probability of success for a certain experiment (the outcome being "success or not", so there are two possible outcomes). A coin toss is a classic example of a Bernoulli experiment with a probability of success 0.5 or 50%, but a Bernoulli experiment can have any probability of success between 0 and 1.

The Poisson Distribution

The Poisson distribution represents the probability of n events in a given time period when the overall rate of occurrence is constant. A typical example is pieces of mail. If your overall mail received is constant, the number of items received on a single day (or month) follows a Poisson distribution. Other examples might include visitors to a website, or customers arriving at a store, or clients waiting to be served in a queue.

The Uniform Distribution

The uniform distribution occurs when all possible outcomes are equally likely. The dice example shown before follows a uniform distribution with equal probabilities for throwing values from 1 to 6. The dice example follows a discrete uniform distribution, but continuous uniform distributions exist as well.

Examples of Continuous Distributions

The Normal or Gaussian distribution

A normal distribution is the single most important distribution, you'll basically come across it very often. The normal distribution follows a bell shape and is a foundational distribution for many models and theories in statistics and data science. A normal distribution turns up very often when dealing with real-world data including heights, weights of different people, errors in some measurement or grades on a test. Our temperature example above follows a normal distribution as well!

Summary

In this lesson, you learned about the concept of (discrete and continuous) statistical distributions, as well as some common ones. You'll learn more about distributions and their properties in the next few lessons!

The Probability Mass Function (PMF)

Introduction

In this lesson, you'll look at a way to represent discrete distributions - the probability mass function (PMF), which maps from each value to its probability. You'll explore probability density functions (PDFs) for continuous data later!

Objectives

You will be able to:

- Describe how probability is represented in the probability mass function
- Visualize the PMF and describe its relationship with histograms

What is a Probability Mass Function (PMF)?

A probability mass function (PMF), sometimes referred to as a frequency function, is a function that associates probabilities with discrete random variables. You already learned about this in the context of coin flips and dice rolls. The **discrete** part in discrete distributions means that there is a **known number of possible outcomes**.

Based on your experience of rolling a dice, you can develop a PMF showing the probabilities of each possible value between 1 and 6 occurring.

More formally:

> The Probability Mass Function (PMF) maps a probability (P) of observing an outcome x of our discrete random variable X in a way that this function takes the form $f(x) = P(X = x)$.

X being a discrete random variable, we can say that the range R_X is a countable set of all possible values of X. They can be represented as a set as follows:

$R_x = \{x_1, x_2, x_3, \ldots\}$

where $x_1, x_2, x_3, \ldots$ are the possible values of x.

Say we are interested in quantifying the probability that X is equal to some given quantity x_3. That is, we want to know $P(x_3)$. For example, in the case of our dice, we might be interested in the probability of getting a 3, which, in this case, would be $P(3) = \frac{1}{6}$

Think of the event A, such that $A = \{X = x_k\}$ is defined as the set of outcomes s in the sample space S for which the corresponding value of X is equal to x_k. This can be written as:

$$A = \{s \in S \mid X(s) = x_k\}$$

(Remember that $s \in S$ is mathematical notation for "s belongs to S" or "s is in S").

PMF Intuition

Let's work through a brief example calculating the probability mass function for a discrete random variable!

You have previously seen that a **probability** is a number in the range [0,1] that is calculated as the *frequency expressed as a fraction of the sample size.* This means that, in order to convert any random variable's frequency into a probability, we need to perform the following steps:

- Get the frequency of every possible value in the dataset
- Divide the frequency of each value by the total number of values (length of dataset)
- Get the probability for each value

Let's show this using a simple toy example:

In [1]:

```
# Count the frequency of values in a given dataset
import collections
x = [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 5, 5]
```

```
counter = collections.Counter(x)
print(counter)

print(len(x))
```

```
Counter({1: 4, 2: 4, 3: 2, 5: 2, 4: 1})
13
```

You'll notice that this returned a dictionary, with keys being the possible outcomes, and values of these keys set to the frequency of items. You can calculate the PMF using step 2 above.

Note: You can read more about the `collections` *library here.*

In [2]:

```
# Convert frequency to probability - divide each frequency value by total number of values
pmf = []
for key,val in counter.items():
    pmf.append(round(val/len(x), 2))

print(counter.keys(), pmf)
```

```
dict_keys([1, 2, 3, 4, 5]) [0.31, 0.31, 0.15, 0.08, 0.15]
```

You notice that the PMF is normalized so the total probability is 1.

In [3]:

```
import numpy as np

np.array(pmf).sum()
```

Out[3]:

```
1.0
```

Visualizing a PMF

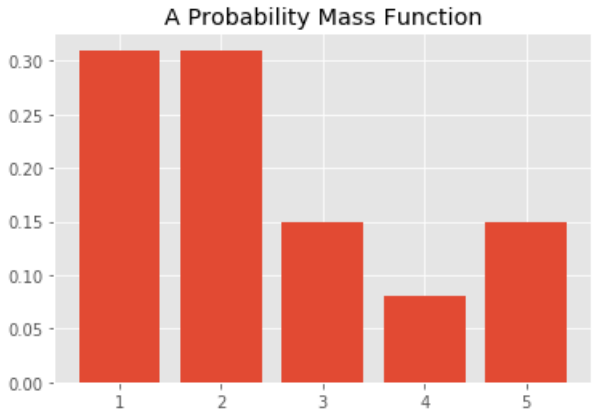
You can inspect the probability mass function of a discrete variable by visualizing the distribution using `matplotlib`. You can use a simple bar graph to show the probability mass function using the probabilities calculated above.

Here's the code:

In [4]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')

plt.bar(counter.keys(), pmf);
plt.title("A Probability Mass Function");
```

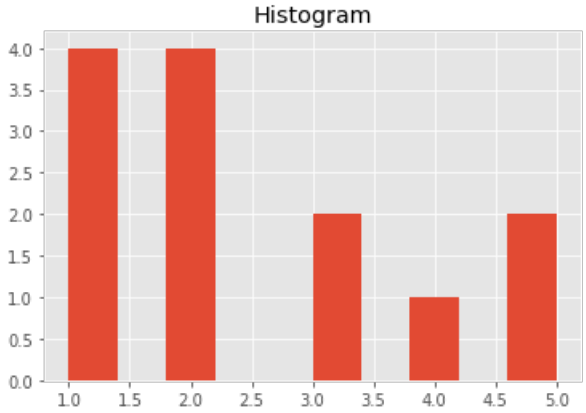


This looks pretty familiar. It's essentially a normalized histogram! You can use `plt.hist(x)` to obtain the histogram.

In [5]:

```
plt.hist(x);
```

```
plt.title('Histogram');
```



If you look carefully, there is only a difference in the y-axis: the histogram shows the frequency count of each value in a dataset, whereas the bar plot here shows probabilities.

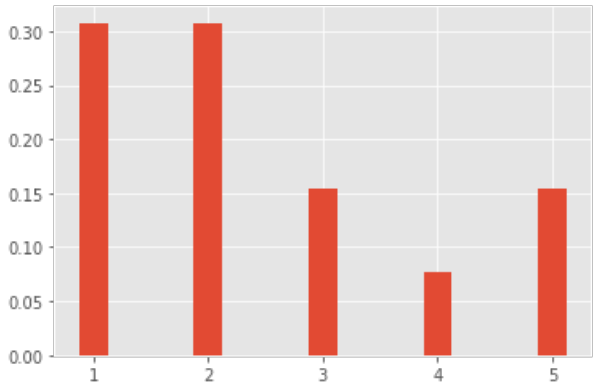
You can alter your histogram to show probabilities instead of frequency counts using the `density = True` argument.

While we're at it, let's rescale our x-axis a little bit better in our histogram. You can also change the width of your vertical bars using the argument `rwidth`.

In [6]:

```
xtick_locations = np.arange(1.5, 7.5, 1)  # x=5, 15, 25, ...
xtick_labels = ['1', '2', '3', '4', '5']
bins = range(1, 7, 1)
plt.xticks(xtick_locations, xtick_labels)

plt.hist(x, bins=bins,  rwidth=0.25, density=True);
```



Expected Value and Variance

When talking about distributions, there will generally be two descriptive quantities you're interested in: the **expected value** and the **Variance**. For discrete distributions, the expected value of your discrete random value X is given by:

$$E(X) = \mu = \sum_i p(x_i)x_i$$

The variance is given by:

$$E((X-\mu)^2) = \sigma^2 = \sum_i p(x_i)(x_i-\mu)^2$$

The table below pust these formulas into practice using our example to get a better understanding!

outcome	1	2	3	4	5	$\sum$
probability	0.31	0.31	0.15	0.08	0.15	1
$p(x_i)x_i$	0.31	0.62	0.45	0.32	0.75	2.45
$(x_i-\mu)^2$	$(-1.45)^2 = 2.1025$	$(-0.45)^2 = 0.2025$	$(0.55)^2 = 0.3025$	$(1.55)^2 = 2.4025$	$(2.55)^2 = 6.5025$	

outcome	x_i	$p(x_i)$	$x_i - \mu$	$(x_i - \mu)^2$	$p(x_i)(x_i - \mu)^2$
1	1	0.65	-1.45	2.1025	1.366625
2	2	0.25	-0.45	0.2025	0.050625
3	3	0.05	0.55	0.3025	0.015125
4	4	0.03	1.55	2.4025	0.072075
5	5	0.02	2.55	6.5025	0.130050
$\sum$					1.634500

As you can see from the far right column, the expected value is equal to 2.45 and the variance is equal to 1.927475. Even though for this example these values may not be super informative, you'll learn how these two descriptive quantities are often important parameters in many distributions to come!

NOTE: In some literature, the PMF is also called the **probability distribution**. The phrase distribution function is usually reserved exclusively for the cumulative distribution function CDF.

Summary

In this lesson, you learned more about the probability mass function and how to get a list of probabilities for each possible value in a discrete random variable by looking at their frequencies. You also learned about the concept of expected value and variance for discrete distributions. Moving on, you'll learn about probability density functions for continuous variables.

The Probability Density Function (PDF)

Introduction

So far, you learned about discrete random variables and how to calculate or visualize their distribution functions. In this lesson, you'll learn about continuous variables and probability density function (PDF) as a way to model the probability of occurrence for these types of variables.

Objectives

You will be able to:

- Calculate the PDF from a given dataset containing real-valued random variables
- Differentiate between probability and probability density
- Explain why it is not possible to calculate the probability for a specific point with continuous variables
- Estimate probabilities for continuous variables by using interpolation

Continuous Variables

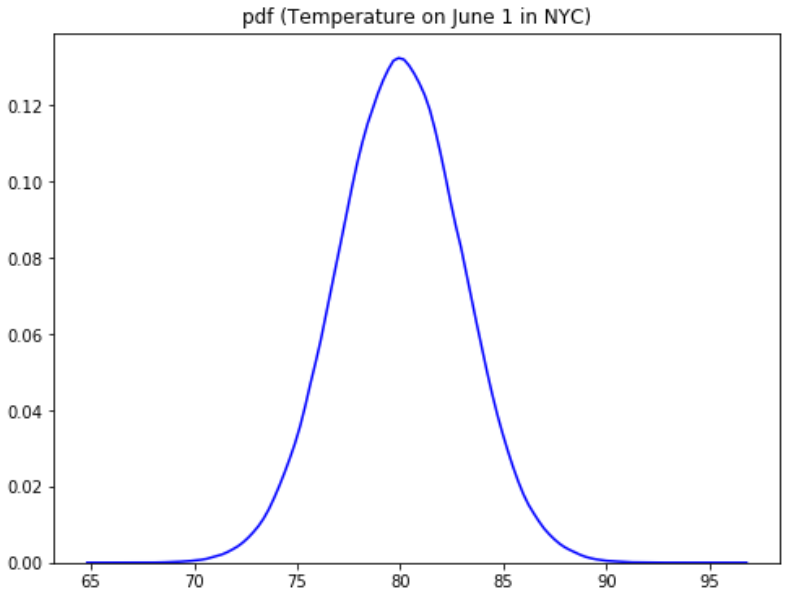
Continuous variables can take on any real value. For example, let's say that the height of a person is 6.1 feet. Is this person *actually* exactly 6.1 feet? How precise is this measurement? In reality, the height could be 6.11 feet, 6.09 feet or 6.1000033 feet.

Ultimately, we can not identify the exact value of a continuous variable. Instead, we can approximate it to a given accuracy. These mathematical nuances lead to some interesting properties when investigating the associated probabilities of these values.

Typical examples of continuous variables are height, weight, temperature, and other variables that can take on *any* real value.

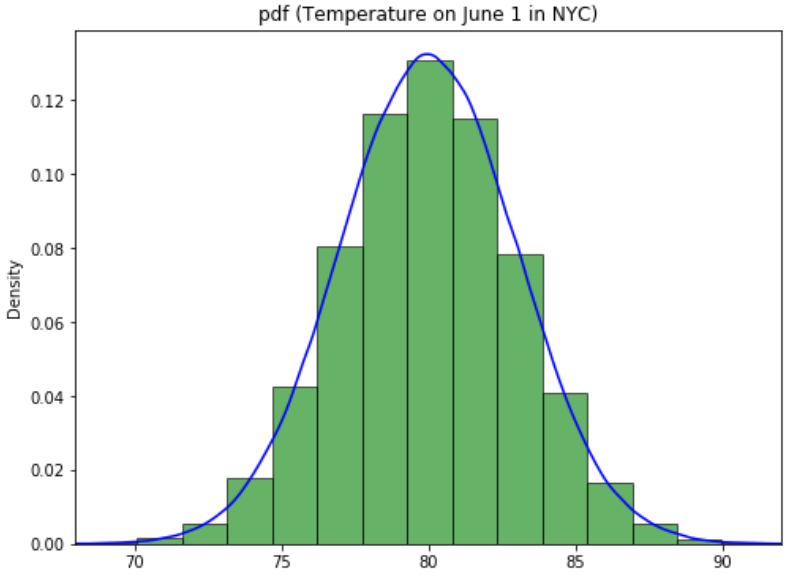
Probability versus Probability Density

Continuous variables can take on an infinite number of variables. Let's have another look at the temperature in NYC on June 1st. Here, we can say our random variable X represents possible temperatures on June 1st in NYC.

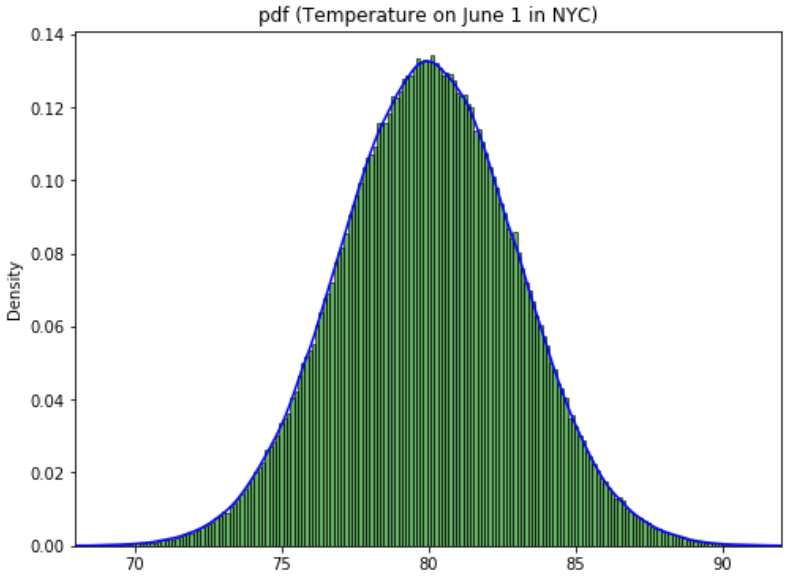


Looking at the plot above, you can see that most of the values lie around the mean (around 80 degrees Fahrenheit). As you move away from the mean in both directions, the blue line goes down creating two tails at both ends. The blue line above shows a **Probability Density Function**, as compared to probability functions we saw when looking at the PMFs.

A Probability Density Function (PDF) helps identify the regions in the distribution where observations are more likely to occur, in other words, where the observation occurrence is **more dense**. It is actually not hard to think about the analogy with a probability mass function. Imagine that we would put out temperatures in "bins" of about 1.5 degrees Fahrenheit. This is what your function would look like:



The histogram you see here may remind you very much of the PMF you saw before! Remember that when dealing with PMFs, you calculated the **mass** for each class, or the number of occurrences. This is exactly what we did creating this histogram. The idea is that, when you "bin" your continuous variables, and you gradually increase the number of bins used, you get a function-like outline of your bins which looks exactly like the PDF:

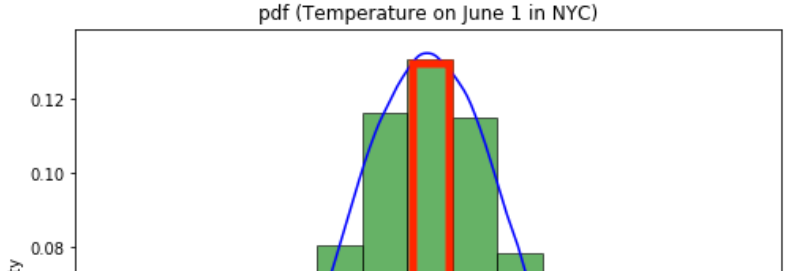


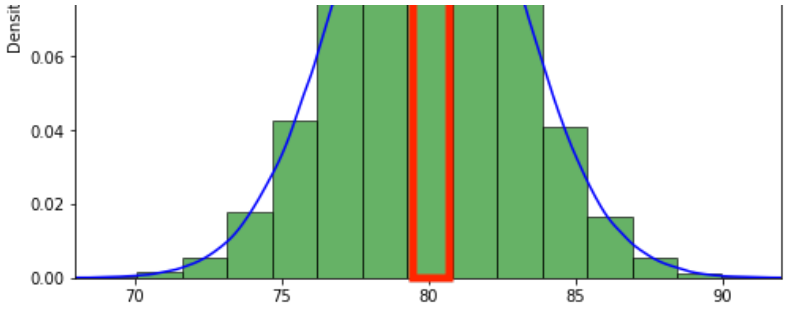
For the distribution of NYC temperatures, the mean region has a high probability density as compared to tails. This means that more extreme temperatures (roughly <70 or >90) are way less likely on a late spring day like June 1. The function shape here is an example of a **Normal Distribution**, which you'll learn more about later.

> The probability density function (also called a probability distribution function) shows all possible values for temperature, which in theory has an infinite amount of possibilities.

Interpreting the PDF

Let's look at this plot again and the y-axis:





Looking at the histogram, and based on the middle bin, you can make the following statement:

About 13% of the time you'll observe a temperature between 79.3 and 80.8 Degrees Fahrenheit.

This is a typical probability mass function statement, where one bar or bin is associated with a fixed probability. The reason why we were able to do this is because we *binned* the temperatures.

When interpreting probability density functions, you need to be **very** careful about determining probabilities for certain events, especially when it comes to exact numbers of events occurring. For example, think about this: if we have a very high tech thermometer, what is the probability that the temperature in NYC on June 1 is *exactly* 80 Degrees?

$$P(\text{Temp} = 80)$$

Looking at the graph, you may think that this probability is around 0.13, **but this is not the case**. The idea of continuous variables and PDFs is that the probability of any given arbitrary number is always 0, simply because there is an infinite number of possibilities we can check (what is $P(\text{Temp} = 80.3)$? $P(\text{Temp} = 80.0001)$?$P(\text{Temp} = 80.00000895)$?) So, the probability of the temperature being exactly 80 Degrees is 0. When using a PDF, the only way of finding a probability associated with a certain temperature here is when using an *interval* of ranges, so something like:

$$P(79.9 < \text{Temp} < 80.1)$$

You'll see more on this later!

Visualizing Probability Density Functions

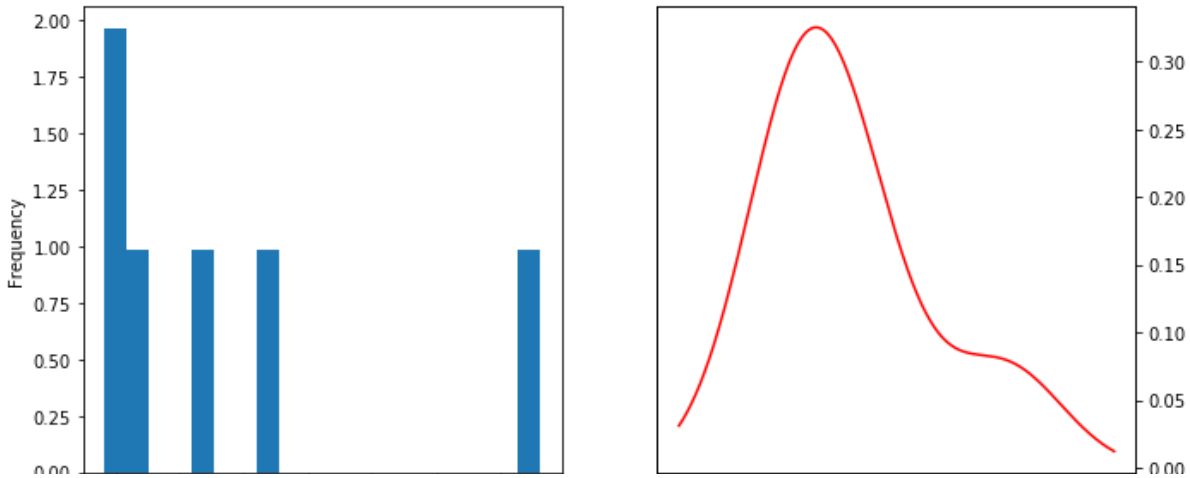
PDFs can be visualized using histograms and density plots. You've had quite a bit of practice on histograms. Now, you'll learn how to plot a density plot for a distribution in Python.

Density Estimation and Plotting

As you've seen before, a density plot is a "smoothed" version of a histogram estimated from the observations. To estimate a density function from given continuous data, you can use parametric or non-parametric methods.

Parametric methods use parameters like mean and standard deviation of given data and attempt to work out the **shape** of the distribution that the data belongs to. These may implement maximum likelihood methods to fit a distribution to the given data. You'll learn more about this later.

Kernel density estimation or KDE is a common non-parametric estimation technique to plot a curve (the kernel) at every individual data point. These curves are then added to plot a smooth density estimation. The kernel most often used is a Gaussian (which produces a bell curve at each data point). Other kernels can be used in special cases when the underlying distribution is not normal.



In the image above, the histogram (left) and kernel density estimate (right) are constructed using the same data.

Expected Value and Variance

Next, let's explore expected value and variance for PDFs. Recall the following two formulas for PMFs:

$$E(X) = \mu = \sum_i p(x_i)x_i$$

$$E((X-\mu)^2) = \sigma^2 = \sum_i p(x_i)(x_i-\mu)^2$$

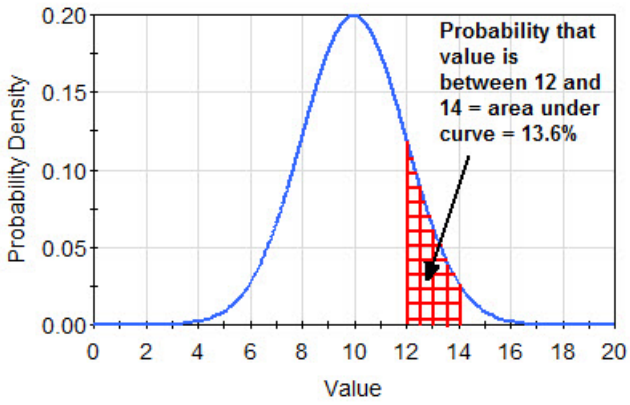
We want to translate this to something we can use in this context of probability density functions. Note how $P(X = x_i) = p(x_i)$ is part of this formula, and how we mentioned before that for PDFs, the probability of our random variable taking exactly a specified value is 0. We need to change our formulas slightly, and this can be done by using **integrals**:

$$E(X) = \mu = \int_{-\infty}^{+\infty} p(x)x dx$$

$$E((X-\mu)^2) = \sigma^2 = \int_{-\infty}^{+\infty} p(x)(x-\mu)^2 dx$$

Recall how the integral basically is a measure to calculate the area under a certain curve. Thinking of it this way, the transition from a histogram to a curve makes much more sense as well. Let's look at the plot below for an illustration. If you want to get the probability to obtain exactly 9, you would get a 1-dimensional line down which isn't really an "area". For this reason, $P(X = 9) = 0$.

If you want to determine the probability of observing a value within a specific range though, you can look at the area under the curve to obtain this probability as highlighted in red below:



The formal mathematical representation for calculating an area under the curve is given by:

$$P(a \leq x \leq b) = \int_a^b f(x)dx \geq 0$$

> To obtain a probability of observing a value in an interval $[a, b]$, you can use an integral (which gives you the area under the curve) from a to b using your PDF $f(x)$

Don't worry too much about the adapted formula for PDFs. The main takeaway here is that you simply can't use the same summation expression because $P(X = x_i) = 0$ for any x_i.

Seaborn

At this stage, it's useful to have another look at the visualization library **Seaborn**, which can do wonders for statistical visualizations.

We'll quickly introduce density plot creation using Seaborn here. You'll learn about other Seaborn plots at a later stage!

Let's import the Seaborn library first.

In [1]:

```
import seaborn as sns
```

The function that we are interested in right now is the `seaborn.distplot()` function which can help visualize a distribution in a number of statistical ways including histograms, density plots, and area plots with a lot of coloring and customization features.

```
seaborn.distplot(a, bins=None, hist=True, kde=True, rug=False, fit=None, hist_kws=None, kde_kws=
None, rug_kws=None, fit_kws=None, color=None, vertical=False, norm_hist=False, axlabel=None, lab
el=None, ax=None)
```

Here is the official documentation if you want to learn more!

Let's Look at Some Data

We'll look at another weight-height dataset, this time containing 10,000 observations about the heights and weights of individuals, grouped by gender. Let's load the dataset first.

In [2]:

```
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import pandas as pd

data = pd.read_csv('weight-height.csv')

print(data.head())
data.describe()
```

```
  Gender     Height      Weight
0   Male  73.847017  241.893563
1   Male  68.781904  162.310473
2   Male  74.110105  212.740856
3   Male  71.730978  220.042470
4   Male  69.881796  206.349801
```

Out[2]:

	Height	Weight
count	10000.000000	10000.000000
mean	66.367560	161.440357
std	3.847528	32.108439
min	54.263133	64.700127
25%	63.505620	135.818051
50%	66.318070	161.212928
75%	69.174262	187.169525
max	78.998742	269.989699

Let's plot the density plot for data in the `Height` column using seaborn `distplot()`. We'll be plotting:

- a Box and Whiskers plot
- a histogram
- a non-parametric Kernel Density Estimation plot
- Parametric distribution fit plot

... all in one single go.

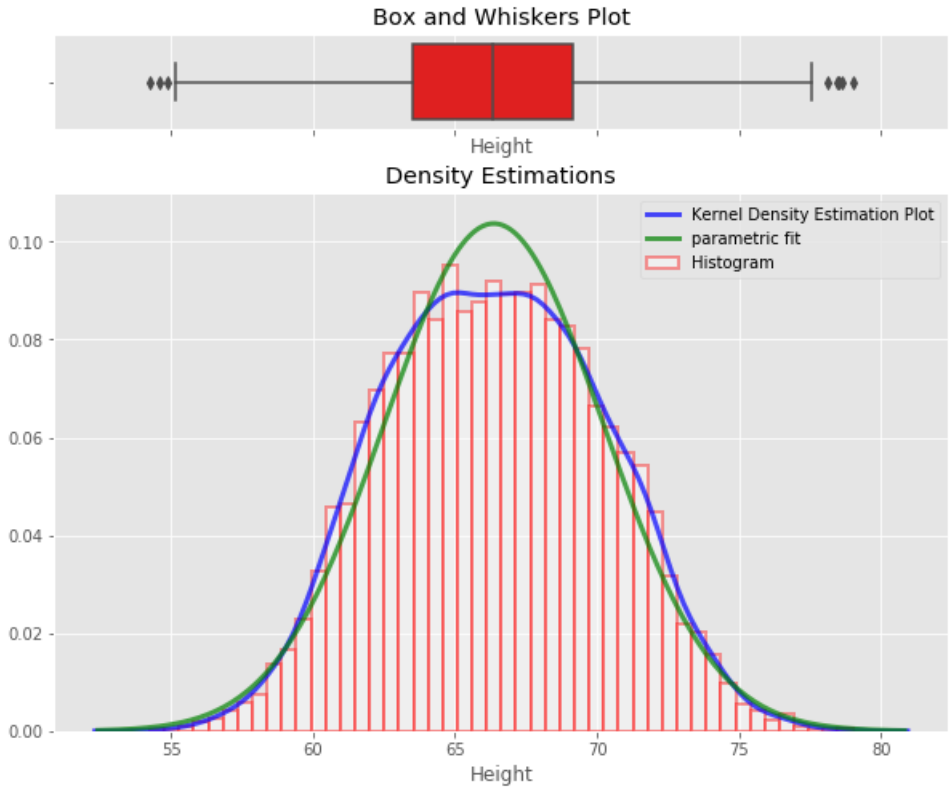
In [3]:

```
import scipy.stats as stats
# Create two vertical subplots sharing 15% and 85% of plot space
# sharex allows sharing of axes i.e. building multiple plots on same axes
fig, (ax, ax2) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85)}, figsize = (10,
8) )

sns.distplot(data.Height,
             hist=True, hist_kws={
                                  "linewidth": 2,
                                  "edgecolor" :'red',
                                  "alpha": 0.4,
                                  "color":  "w",
```

```
                                "label": "Histogram",
                                },
             kde=True, kde_kws = {'linewidth': 3,
                                'color': "blue",
                                "alpha": 0.7,
                                'label':'Kernel Density Estimation Plot'
                                },
             fit= stats.norm, fit_kws = {'color' : 'green',
                                         'label' : 'parametric fit',
                                         "alpha": 0.7,
                                          'linewidth':3},
             ax=ax2)
ax2.set_title('Density Estimations')

sns.boxplot(x=data.Height, ax = ax,color = 'red')
ax.set_title('Box and Whiskers Plot')
ax2.set(ylim=(0, .08))
plt.ylim(0,0.11)
plt.legend();
```



You can see how you can easily visualize multiple statistical aspects of a distribution.

You can further customize these plots as you wish. You can have a look at official Seaborn documentation for their color pellets, styles, labeling options, etc.

Interpolation

Another (rather naive - yet effective) way is to estimate a density function using interpolation between the peaks of the histogram. We can use the `np.histogram` function to calculate histogram values and interpolate between these values to plot a density curve.

In [4]:

```
import numpy as np
n, bins = np.histogram(data.Height, 20, density=1)
n , bins
```

Out[4]:

```
(array([0.00040428, 0.00145539, 0.00533644, 0.01228997, 0.02603534,
        0.04883648, 0.07034393, 0.08376588, 0.09031514, 0.08724265,
        0.08958744, 0.08562554, 0.07204189, 0.05829652, 0.0412361 ,
        0.02061805, 0.00970261, 0.00347677, 0.00153625, 0.00040428]),
 array([54.26313333, 55.49991378, 56.73669423, 57.97347468, 59.21025513,
        60.44703558, 61.68381603, 62.92059648, 64.15737693, 65.39415738,
```

```
60.44703558, 61.68381603, 62.92059648, 64.15737693, 65.39415738,
66.63093784, 67.86771829, 69.10449874, 70.34127919, 71.57805964,
72.81484009, 74.05162054, 75.28840099, 76.52518144, 77.7619619 ,
78.99874235]))
```

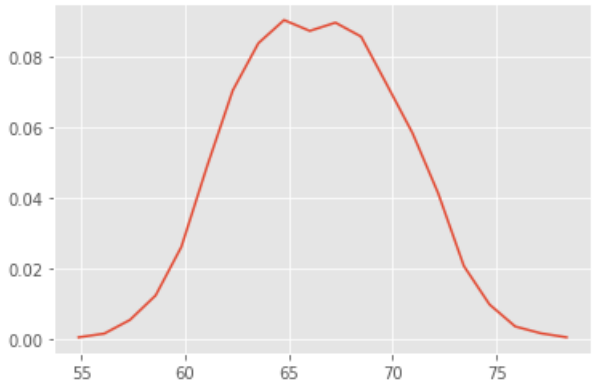
Here `n` represents the values of the histogram and `bins` are the bin positions along the x-axis. We can interpolate between these values to calculate the points for the density curve as you can see below.

In [5]:

```
# Initialize numpy arrays according to number of bins with zeros to store interpolated values
pdfx = np.zeros(n.size)
pdfy = np.zeros(n.size)

# Interpolate through histogram bins
# identify middle point between two neighbouring bins, in terms of x and y coords
for k in range(n.size):
    pdfx[k] = 0.5*(bins[k]+bins[k+1])
    pdfy[k] = n[k]

# plot the calculated curve
plt.plot(pdfx, pdfy);
```

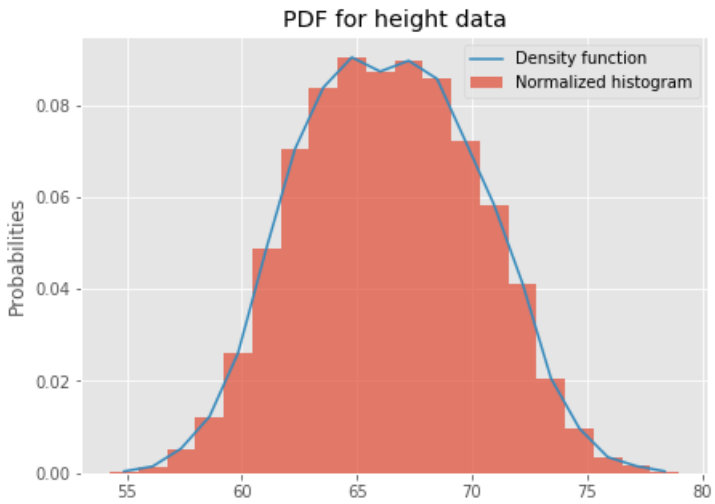


That looks reasonable! This plot reflects our density function. You can plot it on top of the normalized histogram now and get a complete picture of underlying data.

In [6]:

```
plt.figure(figsize=(7,5))
data.Height.plot.hist(bins = 20, normed=True, label = 'Normalized histogram', alpha = 0.7)
# plot the calculated curve
plt.plot(pdfx, pdfy, label = 'Density function')
plt.ylabel ('Probabilities')
plt.legend()
plt.title ('PDF for height data')
plt.show()
```

```
C:\Anaconda3\envs\learn-env\lib\site-packages\pandas\plotting\_matplotlib\hist.py:62: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  n, bins, patches = ax.hist(y, bins=bins, bottom=bottom, **kwds)
```



This looks pretty good! In the next lab, you'll practice your knowledge!

NOTE: Be careful when using this naive interpolation method! The results depend very much on the number of bins used when creating your histogram.

Summary

In this lesson, you learned about the probability density function and identified the difference between point probabilities (for PMFs) and PMFs for continuous variables. One important takeaway is that the probability of a specific value for a continuous variable is zero! You can use integrals to get probabilities for a range of values when using PDFs. The idea of taking ranges of values will become more important when looking at Cumulative Density Functions, but let's practice our PDF knowledge first!

The Cumulative Distribution Function

Introduction

The PMF function that we saw before works great for inspecting discrete random variables and calculating their expected values. However, we did see that when moving towards continuous random variables, obtaining probabilities for observing a specific outcome is not possible (or, simply put, the probabilities were 0). We also noted that when working with PDFs, you can't really read the y-axis and have to be careful with interpretation. In this lesson, you'll learn about the cumulative distribution function (CDF) and how it is useful to overcome these issues.

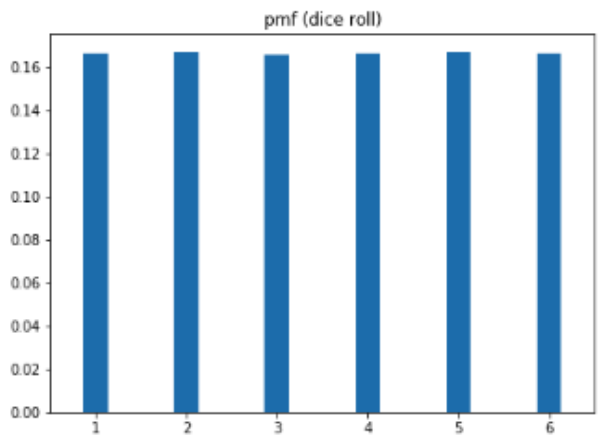
Objectives

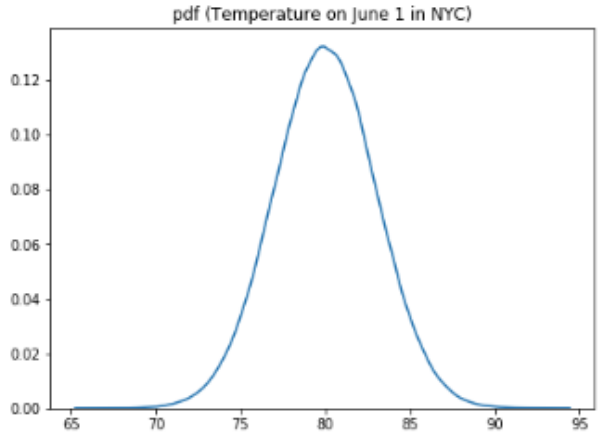
You will be able to:

- Differentiate between a PMF, PDF, and a CDF in terms of cumulative probabilities
- Calculate CDF in Python for a given discrete variable with limited set of possible values
- Visualize and inspect a given CDF in order to make assumptions about the underlying data

Limitations of PMFs and PDFs

To illustrate the use of Cumulative Distribution Functions, let's have another look at the PMF and PDF of our dice and temperature example:

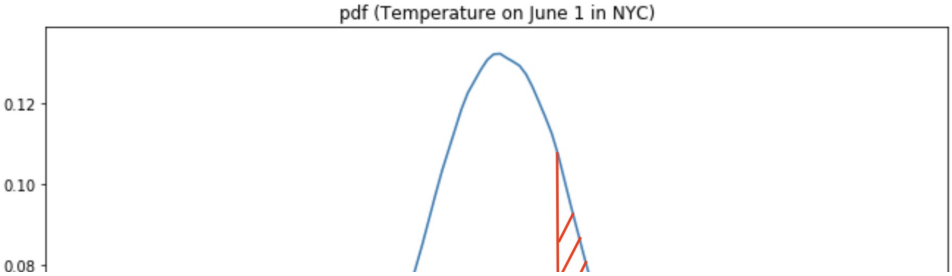


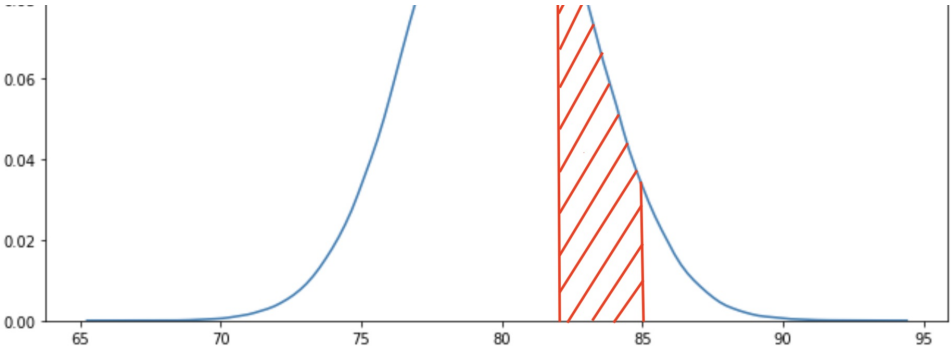


Recall how we could easily read probabilities from the dice PMF plot ("The probability of throwing a 4 is 16.66%"), but it is much harder to interpret the temperature PDF. What is the probability that the temperature is exactly 80 degrees? We learned in the previous lesson that all these so-called "point probabilities" are 0, so the bottom line is that it is very hard to "read" any interesting information from a PDF. The PDF is mainly there to get a sense of the data density, but you cannot readily read the y-axis to get to probabilities.

We did see last that when you want to calculate probabilities, you need to take integrals and look at ranges of values of your continuous random variables. For example, you can ask yourself the question: "What is the probability the temperature in NYC is between 82 and 85 degrees on June 1?" The answer is the surface of the red shaded area!

From the last lesson, you learned that you can use the integral to get this "area under the curve" value by taking the integral as follows:





$$P(82 \leq X \leq 85) = \int_{82}^{85} f_x(x)dx \geq 0$$

This is the rationale that is being used when working with Cumulative Density Functions, which will be introduced next.

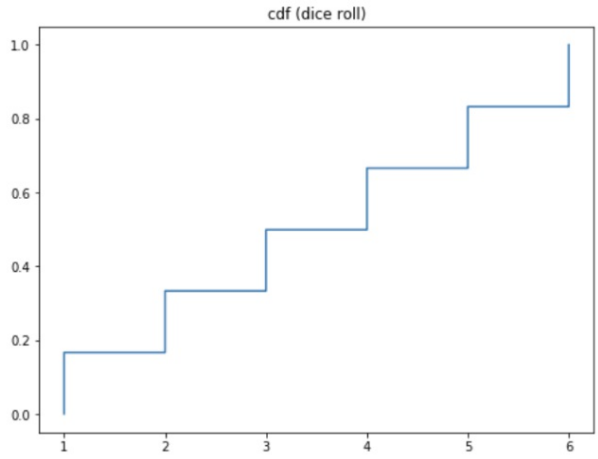
How does a Cumulative Density Function (CDF) work?

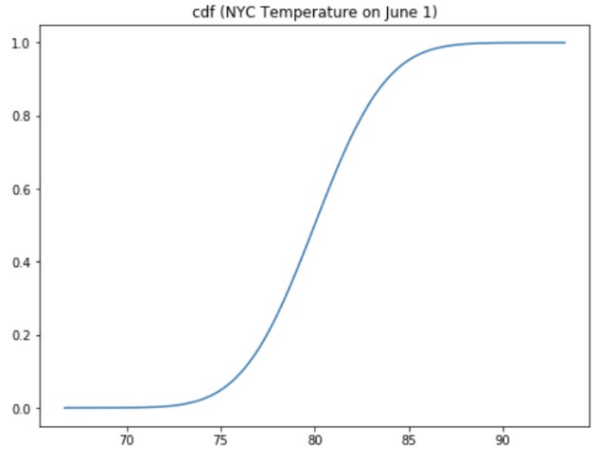
The CDF is a function of x just like a PMF or a PDF, where x is any value that can possibly appear in a given distribution. To calculate the $CDF(x)$ for any value of x, we compute the proportion of values in the distribution less than or equal to x as follows:

$$F(x) = P(X \leq x)$$

> The Cumulative Distribution Function, CDF, gives the probability that the variable X is less than or equal to a certain possible value x.

The cumulative distribution functions for a dice roll and the weather in NYC are plotted below.





This is also what "cumulative" means - you're simply adding up probabilities.

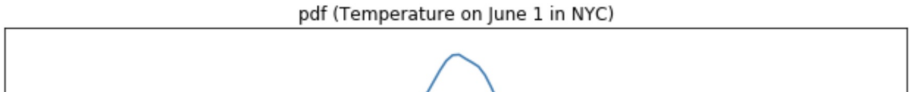
You'll notice that in general, CDFs are smooth curves for continuous random variables, where they are "step functions" when looking at discrete random variables. Looking at these curves, we can answer questions by looking at the y-axis.

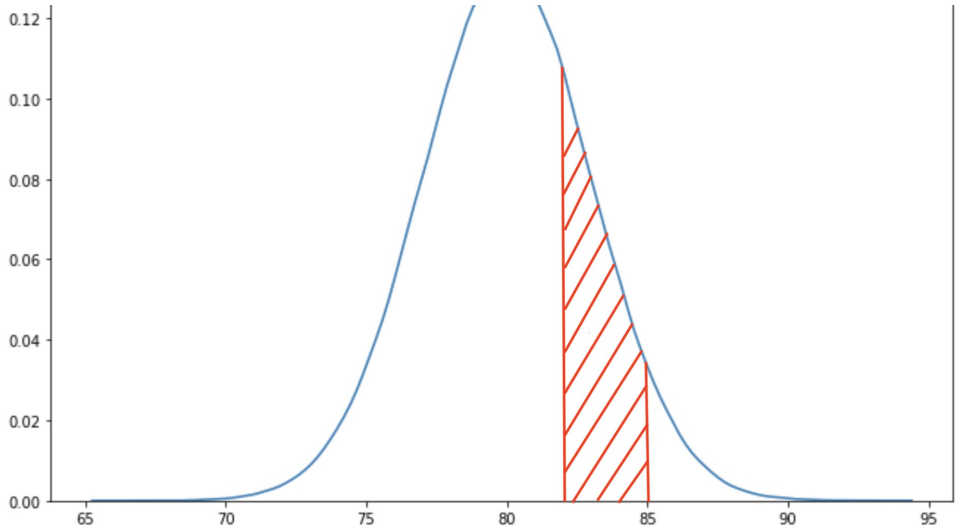
What is the probability that you throw a value $\leq$ 4 when throwing a dice? 0.6667 or 66.67. For this discrete example it is pretty straightforward, as this is the probability of throwing a 1 OR 2 OR 3 OR 4, so $0.1666 * 4$.

What is the probability that the temperature in NYC is $\leq$ 79? Looking at the associated y-value when looking at an x-value of 79, this probability is around 40% or 0.4.

Calculating more probabilities using the CDF

Let's go back to our weather example introduced before. An additional advantage of CDFs is that you can use them to easily calculate things like:





The idea is that

$$P(82 \leq X \leq 85) = P(X \leq 85) - P(X \leq 82) = F_X(85) - F_X(82)$$

This means that you can look at the y-value of your cumulative density function to get the answer to this question.

$$F_X(85) - F_X(82) \approx 0.95 - 0.6 = 0.35$$

Summary

In this lesson, we looked at a CDF as a so-called "percentile probability function" of discrete or continuous random variables. You learned how to calculate and visualize a CDF and how to use them to calculate certain probabilities.

Bernoulli and Binomial Distribution

Introduction

Now that you learned about probability mass functions, probability density functions, and cumulative density functions (PMFs, PDFs, and CDFs, respectively), let's dive into the world of distributions!

In this section, you'll learn about two foundational probability distributions that are extremely useful and have an endless amount of applications: the Bernoulli distribution and the Binomial distribution. You'll notice that these distributions formalize a lot of the theory you learned in the probability theory section!

Objectives

You will be able to:

- Describe the components of a Bernoulli distribution
- Describe how a Binomial Distribution is related to a Bernoulli Distribution
- Use `numpy` to randomly generate Binomial and Bernoulli trials
- Use `matplotlib` to show the output of generated Binomial and Bernoulli trials

The Bernoulli or Binary distribution

In the previous sections, we discussed several probability theory situations regarding throwing a dice or flipping a coin. The Bernoulli distribution is a discrete distribution that formalizes the idea of a coin flip.

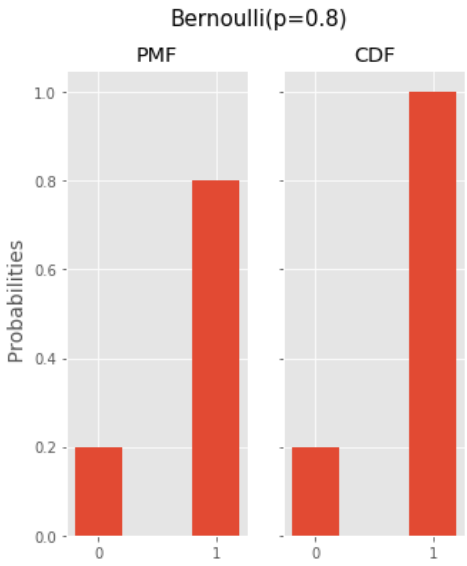
The Bernoulli experiment is a simple experiment in which there is a binary outcome: 0-1, success-failure, heads-tails, etc.

If we were to model a coin flip with a Bernoulli distribution, we could say that 0 means heads, and 1 tails. With a fair coin, obtaining either value when performing a coin toss would have an assigned probability of 0.5. The Bernoulli experiment can also describe events with different probability structures. For example, let's say that the chance of scoring a penalty goal is 80%. Where Y is the penalty outcome:

$Y = Bernoulli(p)$ and $p = P(Y = 1) = 0.8$.

The distribution is defined by 1 parameter, the parameter p, describing the chance of "success".

Let's look at the PMF and the CDF when $p = 0.8$.



As you can see these functions look pretty straightforward when you plot them.

Now, what is the mean and the variance of these functions? Recall that

$E(X) = \mu = \sum_i p(x_i)x_i = 0.2 * 0 + 0.8 * 1 = 0.8$

$E((X-\mu)^2) = \sigma^2 = \sum_i p(x_i)(x_i-\mu)^2 = 0.2 * (-0.8)^2 + 0.8 * (0.2)^2 = (0.8 * 0.2) * (0.2 + 0.8) = 0.16$

A general rule for the Bernoulli distribution is that: $E(X) = p$ and $\sigma^2 = p * (1 - p)$.

Note how the Bernoulli distribution describes a single coin flip, a single penalty shot, etc. What if we repeat this process multiple times and are interested in the probability of obtaining a certain number of 1s/successes/tails? This process is described by the **binomial distribution**.

The Binomial distribution

The binomial distribution describes the process of performing n *independent* Bernoulli trials. So what does it mean that the trials are independent?

When we say that events are **independent, this means that an event is not affected by previous events**.

Applying this to our penalty goal example, this means that the assumption is that, when a soccer player misses a penalty and then tries again, the fact that he missed it the previous time does not affect his chances of making it now: the probability is still 80% ($p = 0.8$)!

As we have a repeated Bernoulli experiment, the binomial distribution has two parameters: p (the success probability) and n (the number of times the experiment is repeated). We say that random variable Y follows a Binomial distribution: $Y = bin(n, p)$.

Now we're interested in finding an expression that gives us the probability to find each possible amount of successes k between 0 and n. Going back to our goal penalties example, imagine we still have $p = 0.8$, but have 3 consecutive penalty shots. What is:

- $P(Y = 0)$ (or the soccer player doesn't score a single time)?
- $P(Y = 1)$ (or the soccer player scores exactly once)?
- $P(Y = 2)$ (or the soccer player scores exactly twice)?
- $P(Y = 3)$ (or the soccer player scores exactly three times)?

Calculating $P(Y = 0)$ is pretty easy, it's simply $0.2 * 0.2 * 0.2$, so 0.008.

Getting to $P(Y = 1)$ is a little bit more complicated. It's essentially $0.8 * 0.2 * 0.2 + 0.2 * 0.8 * 0.2 + 0.2 * 0.2 * 0.8$, so it's the probability of scoring during the first shot, or the second shot, or the third shot. So essentially, its $3 * (0.8 * 0.2 * 0.2)$. You can see how combinatorics play a role here! The general formula is given as follows:

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

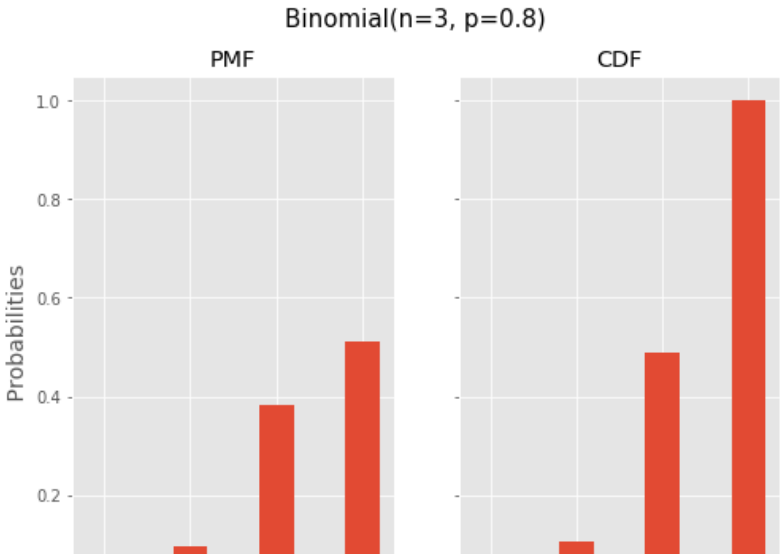
Where k denotes the number of successes. Applying this on $P(Y = 2)$ we get

$$P(Y = 2) = \binom{3}{2} 0.8^2 (1 - 0.8)^{(3-2)} = \frac{3!}{2!1!} * 0.8^2 * 0.2^1 = 3 * 0.8^2 * 0.2$$

Repeating this for the other discrete values between 0 and 3, you get:

- $P(Y = 0) = 0.008$
- $P(Y = 1) = 0.096$
- $P(Y = 2) = 0.384$
- $P(Y = 3) = 0.512$

Note how they sum to one, which is exactly what's expected! Let's look at their PMF and CDF.



Just like before, let's look at the mean and variance:

$E(X) = \mu = \sum_i p(x_i)x_i = 0.008 * 0 + 0.096 * 1 + 0.384 * 2 + 0.512 * 3 = 2.4$

$E((X-\mu)^2) = \sigma^2 = \sum_i p(x_i)(x_i-\mu)^2 = 0.008 * (-2.4)^2 + 0.096 * (-1.4)^2 + 0.384 * (-0.4)^2 + 0.512 * 0.6^2 = 0.48$

Very similarly to Bernoulli, a general rule for the Binomial distribution is that: $E(X) = n * p$ and $\sigma^2 = n * p * (1-p)$. You simply multiply your results with the number of trials n!

Use NumPy to randomly generate Binomial and Bernoulli trials.

In the first part, you learned about Bernoulli and Binomial using their formulas to obtain the probability distributions. You can also perform random sampling. What you're basically doing then is selecting a sample from a statistical population in a way that every possible sample has a predetermined probability of being selected.

Applied to our example, imagine that 1 represents scoring a penalty goal and 0 represents missing. If $p = 0.8$, if you take 100 penalty shots in a row, you'd expect to score about 80 goals. The reason why it's not exactly 80 is because uncertainty plays a role, and this is exactly what you can model using the NumPy random library! Below, we're using `np.random.binomial` to generate how many successful penalties we have when shooting 100 penalties.

In [1]:

```
import numpy as np
np.random.seed(123) # set a seed to get the same results
np.random.binomial(100, 0.8)
```

Out[1]:

```
78
```

Now, let's try this again:

In [2]:

```
np.random.binomial(100, 0.8)
```

Out[2]:

```
82
```

And again:

In [3]:

```
np.random.binomial(100, 0.8)
```

Out[3]:

```
83
```

You can see how this number changes slightly every time and fluctuates around 80. If you'd repeat this many times, and then divide the final result by the number of times you've repeated this, you could expect that the amount of successes will converge to 80. The for loop below does this 500 times.

In [4]:

```
iteration = []
for loop in range(500):
    iteration.append(np.random.binomial(100, 0.8))
    np_it = np.array(iteration)
```

In [5]:

```
sum(np_it)/500
```

Out[5]:

```
80.052
```

Now we'll use `np.random.binomial` to illustrate our findings regarding penalties above. Let's keep track of how many times we

observe 0 goals, 1 goal, 2 goals, and 3 goals and find the probabilities through simulation. Now, let's repeat our experiment 10000 times.

In [6]:

```
n = 10000
iteration = []
for loop in range(n):
    iteration.append(np.random.binomial(3, 0.8))
    np_it = np.array(iteration)
```

`np_it` stores the total penalty goal outcomes (0 to 3) for each of the 10000 iterations. Now using `np.unique()` with the optional argument `return_counts`, you get the levels of the k as in your Binomial formula along with how often they occurred when running 10000 trials.

In [7]:

```
values, counts = np.unique(np_it, return_counts=True)
print(values)
print(counts)
```

```
[0 1 2 3]
[  58  929 3946 5067]
```

Visualize these results

Now, let's use these results and visualize them in terms of fractions. You'll see that these fractions are approximations of the values as calculated in the Binomial distribution formula.

In [8]:

```
import matplotlib.pyplot as plt
plt.bar(values, counts/10000, align='center', alpha=0.9)
plt.xticks(values)
plt.ylabel('Fraction')
plt.title('Number of penalty goals')
plt.show()
```

```
<Figure size 640x480 with 1 Axes>
```

Let's now look at the values and compare them with the theoretical result. Recall that the theoretical result was: $\{0.008, 0.096, 0.384, 0.512\}$.

In [9]:

```
counts/10000
```

Out[9]:

```
array([0.0058, 0.0929, 0.3946, 0.5067])
```

This seems pretty close to our theoretical result! Try using many more trials (50,000 or 100,000) and see how the sampling result changes!

Summary

In this lecture, you learned about the Bernoulli and Binomial distributions, you learned how to use the formula for the Binomial distribution, and how to simulate Binomial trials to get to approximations of the Binomial distribution probabilities.

The Normal Distribution

Introduction

For data scientists and machine learning professionals, the normal (also referred to as Gaussian) distribution stands out as one of the most commonly used distribution models. This lesson provides an introduction to the normal distribution, its characteristics, and its significance in data science.

Objectives

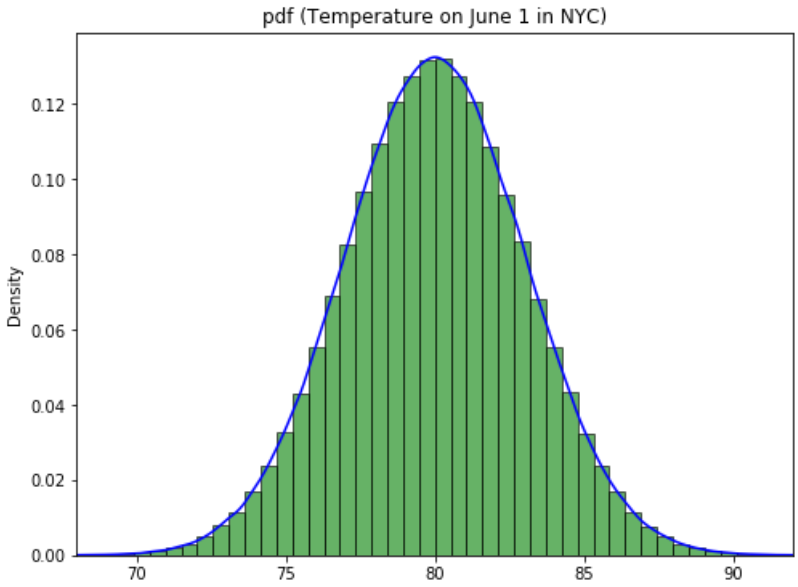
You will be able to:

- List the unique characteristics of a normal distribution
- Identify real world instances of things that follow a Gaussian distribution
- Use `numpy` to generate a random normal distribution

The Normal Distribution

The normal distribution is the most important and most widely used distribution in statistics and data science. It is also called the "bell curve," due to its bell shape, or the "Gaussian curve" after the German mathematician Karl Friedrich Gauss.

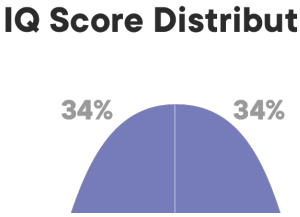
Recall our NYC weather distribution. This is a classic example of a normal distribution. The idea is that there is sort of an expectation around what the temperature will be on June 1 (80 degrees Fahrenheit) and that temperatures much lower or much higher are less likely the further they move away from this expected temperature. This type of behavior is present in many phenomena, as you'll see later.

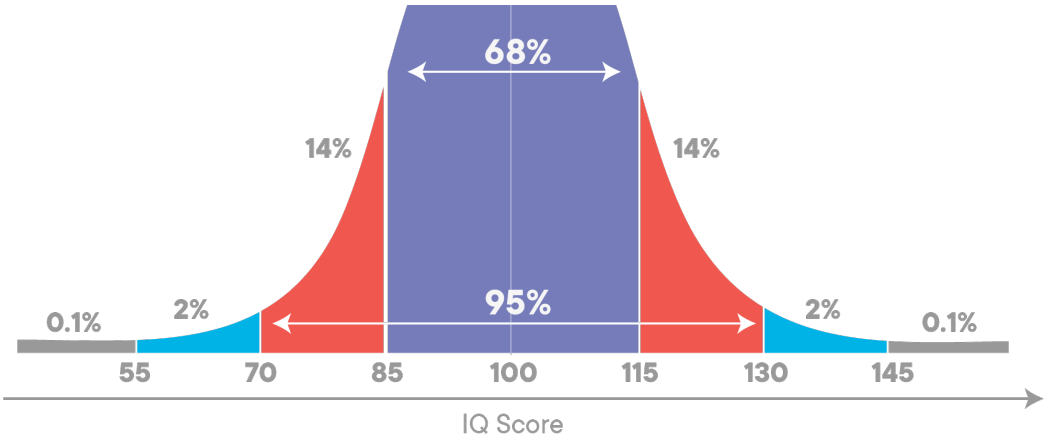


The normal distribution is **a continuous distribution**. In practice though, you'll see many discrete distributions that follow a bell curve shape:

- The observed values are actually discrete. For example, human IQ follows a normal distribution, but IQ is only specified up to the unit digit level, e.g. an IQ of 90, 91, or 92.
- The values in our distribution are actually continuous (e.g. our temperature example) but recorded up to a certain constant because there is (obviously) no "exact" thermometer that measures temperature up to an infinite amount of digits.

Even though the IQ level is not actually recorded as a continuous variable, you'll see that the distribution is generally represented as a smooth curve!





The Probability Density Function

The probability density function equation for the normal distribution is given by the following expression:

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

Here,

- μ is the mean
- σ is the standard deviation
- $\pi \approx 3.14159$
- $e \approx 2.71828$

Don't worry if your head is spinning right now. Don't worry about the formula, what you really need to remember is that:

- A normal distribution has 2 key parameters, μ and σ, which define the mean and the spread of the distribution, respectively.
- If you apply our formulas of expected values and variance seen in the PDF lesson before, where $X \sim N(x)$:
  - $E(X) = \int_{-\infty}^{+\infty} p(x)x dx = \mu$
  - $E((X-\mu)^2) = \int_{-\infty}^{+\infty} p(x)(x-\mu)^2 dx = \sigma^2$

  where μ and σ are as specified in the formula of $N(x)$

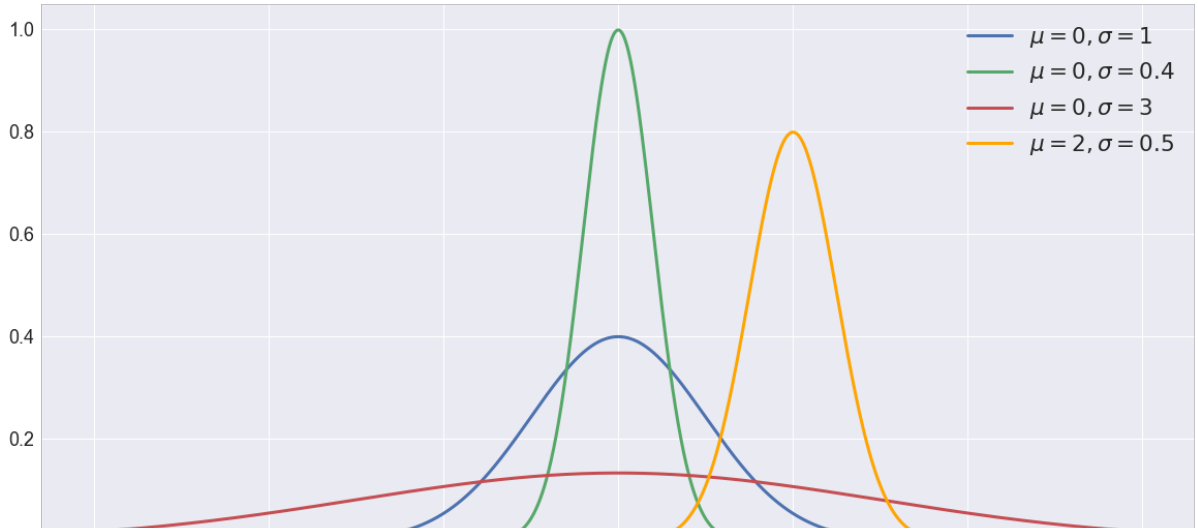
Mean and Standard Deviation

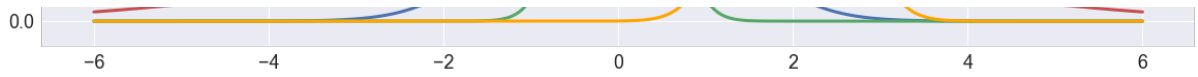
Here is a first simple definition for normal distributions:

> **The Normal Distribution is symmetrical and its mean, median and mode are equal.**

A normal distribution is **centered around its mean**, so the distribution is not skewed (you'll have a chance to learn more about skewness later). This doesn't mean that normal distributions cannot appear in different shapes and forms. How exactly the distribution behaves depends on the 2 key parameters, as specified before: the **mean** and the **standard deviation**.

The following figure shows four normal distributions:



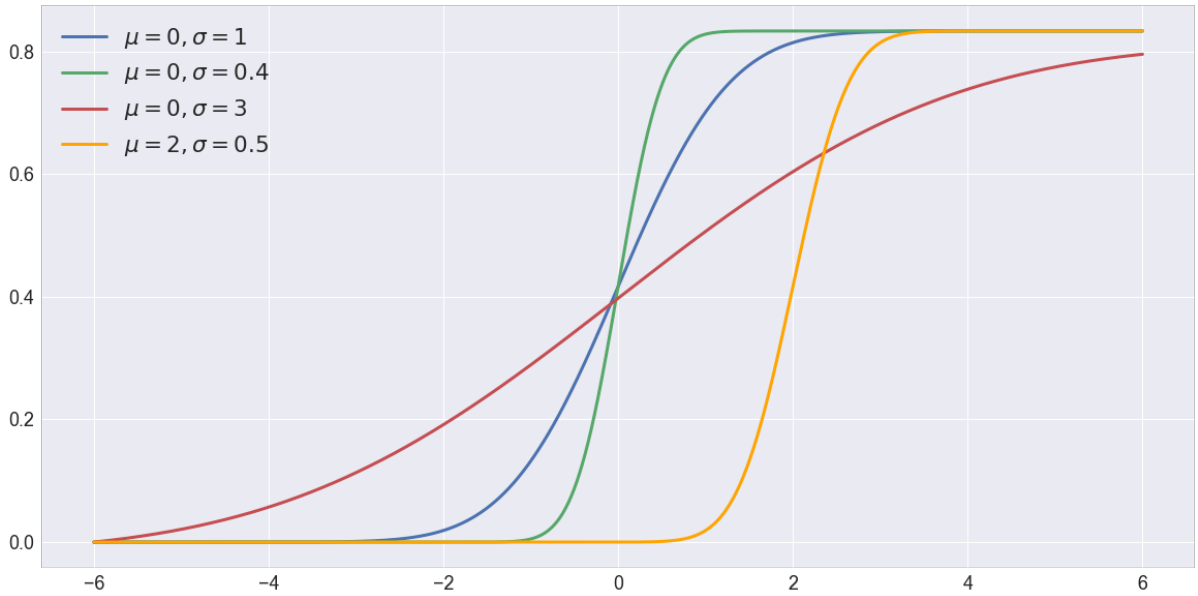


- The green distribution has a mean of 0 and a standard deviation of 0.4
- The distribution in blue has a mean of 0 and a standard deviation of 1.
- The distribution in red has a mean of 0 and a high spread with standard deviation of 3.
- The orange distribution has a mean of 2 and a standard deviation of 0.5.

All of these distributions have the following properties in common:

- They are symmetric around the mean,
- They have relatively higher densities of values at the center of the distribution and relatively lower density in the tails

The CDFs of these distributions are shown below:

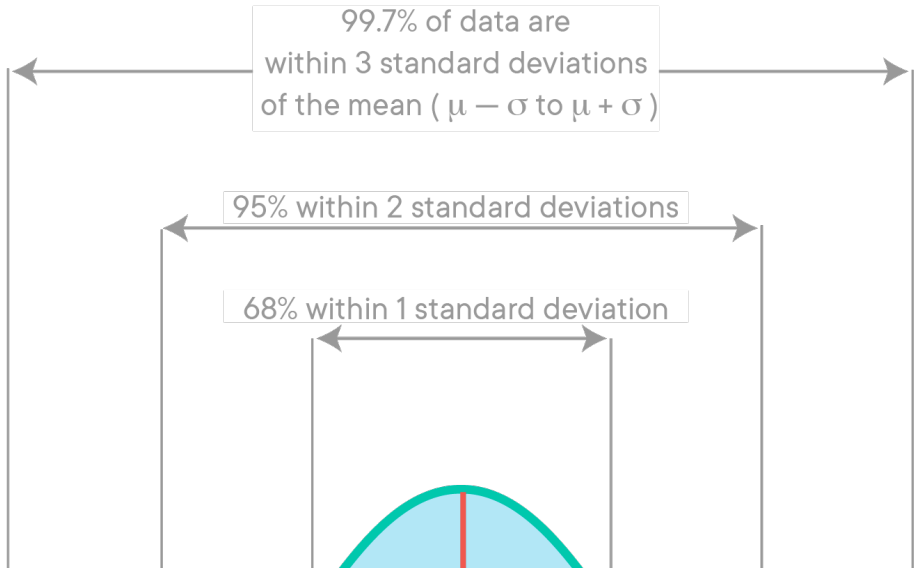


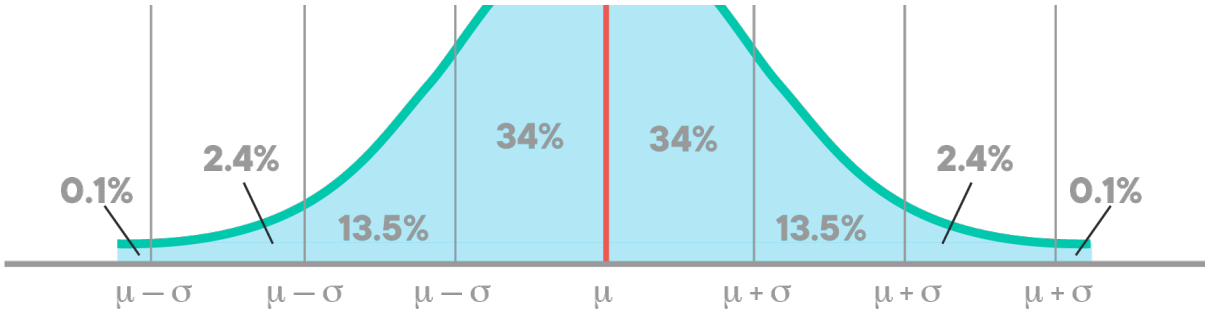
Some More Characteristics of the Normal Distribution

Let's summarize the key characteristics of the normal distribution below:

- Normal distributions are symmetric around their mean
- The mean, median, and mode of a normal distribution are equal
- The area under the bell curve is equal to 1.0
- Normal distributions are denser in the center and less dense in the tails
- Normal distributions are defined by two parameters, the mean (μ) and the standard deviation (σ).
- Around 68% of the area of a normal distribution is within *one standard deviation* of the mean ($(\mu - \sigma)$ to $(\mu + \sigma)$)
- Approximately 95% of the area of a normal distribution is within two standard deviations of the mean ($(\mu - 2\sigma)$ to $(\mu + 2\sigma)$).

Let's look at the image below to get a better sense of the two last statements. In this image, the spread is differentiated between levels of deviation.





This forms a 68-95-99.7 rule, i.e., 68% values of a normal distribution are within 1 standard deviation of the mean, 95% within 2 standard deviations and 99.7 % within 3 standard deviations. This rule is also called the empirical rule. Normally distributed data is considered ideal for analysis due to this simplicity of description. Values in the extreme of tails (more than 3 standard deviations) can be considered "interesting events" as their probability of occurrence is very low (1 occurrence in about ~300!). In other cases, you'll consider them as outliers due to noise or error of measurement. It all depends on your analysis question.

Keeping this in mind, have another look at the IQ distribution and identify "extreme events" in terms of IQ!

Why So Popular?

In this section, you'll learn about some reasons why normal distributions are so popular among data scientists:

Ubiquitous in Natural Phenomena

An amazingly vast number of natural processes naturally follow the normal distribution. A simple normal distribution gives the best model approximation for natural processes like weight, height, blood pressure, etc. Errors committed during some measurements are also found to be normally distributed so they can be modeled and isolated with ease. The income, expenditure and other social attributes of masses are often normally distributed as well.

Central Limit Theorem

The Central Limit Theorem states:

> When you add **a large number** of independent random variables, irrespective of the original distribution of these variables, **their sum tends towards a normal distribution**.

The theorem provides a reason why many natural phenomena follow a normal distribution.

The key takeaway from the central limit theorem is that it allows different distributions to be processed as a normal distribution, even when they do not fulfill the normality requirements shown above. We'll discuss this further when we talk about hypothesis testing.

Here is an interesting youtube video highlighting this phenomenon for now. We will consider this in detail later.

Simplified Computation

When undergoing transformations, a number of distributions tend to change their nature and may result in a totally new distribution. With normal distributions, we can add random variables, take their product or apply any other advanced transformations (like Fourier transformations or Convolutions) - the resulting distribution will always be normal.

For every normal model approximation, there may exist a complex multi-parameter distribution that gives a better approximation than the normal distribution. Even then, a normal distribution is often the preferred distribution to use because it makes the math a lot simpler!

Normal Distributions in Python

In Python, the NumPy module provides a ton of methods to generate and inspect random variables.

You can generate a random normal distribution by providing its parameters μ and σ (mean and sd) to `np.random.norm()`, along with n (number of values to be generated for the normal distribution).

In [1]:

```
import numpy as np
import seaborn as sns

mu, sigma = 0.5, 0.1
```

```
n = 1000
s = np.random.normal(mu, sigma, n)
sns.distplot(s);
```

The density function of a normal distribution can also be plotted using a matplotlib *line plot* and using the formula given above. You'll try to do this in the next lab.

Summary

This lesson provides an introduction to normal distributions, the most common distribution family in the field of statistics and data analysis. You learned about the key characteristics of normal distributions, their density function based on mean and standard deviations, and briefly discussed the reasons behind their ubiquitous nature.

The Standard Normal Distribution

Introduction

In this lesson, we will introduce a special case of normal distributions: "The Standard Normal Distribution".

Objectives

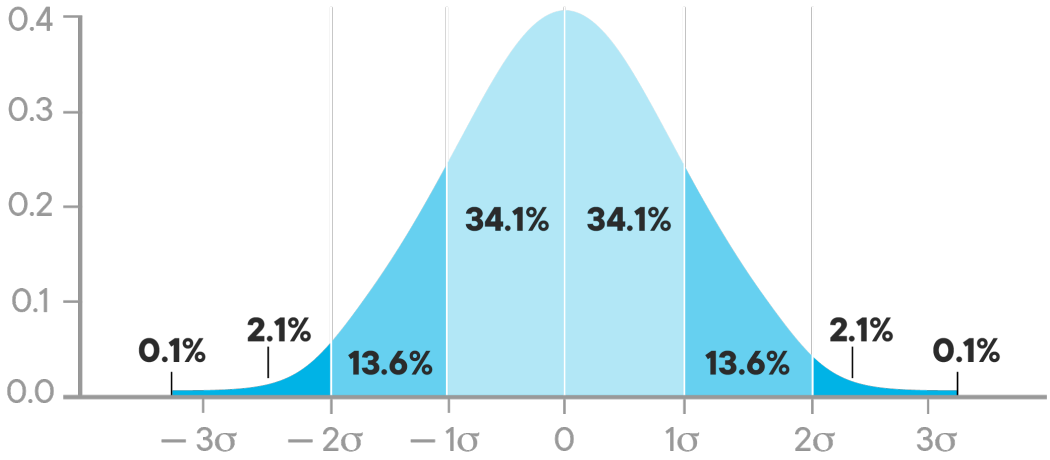
You will be able to:

- Compare and contrast the normal and standard normal distribution
- Calculate and interpret the z-score (standard score) for an observation from normally distributed data
- Convert between a normal and a standard normal distribution

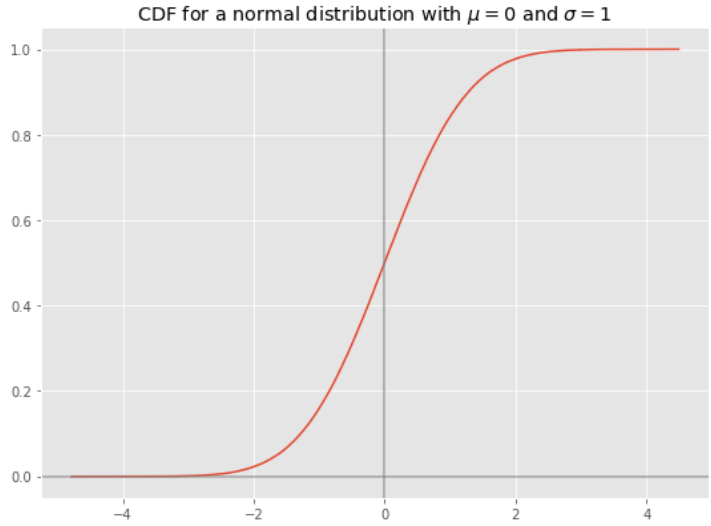
What is a Standard Normal Distribution?

Previously, you learned about the normal (or Gaussian) distribution, which is characterized by a bell shape curve. We also identified the mean and standard deviation as the defining parameters of this distribution. As mentioned before, normal distributions do not necessarily have the same means and standard deviations.

The standard normal distribution, however, is a **special case** of the normal distribution. The Standard Normal Distribution is a normal distribution with a mean of 0 and a standard deviation of 1.



Plotting a continuous cumulative distribution function for the standard normal distribution, the CDF would look like this:



Thinking back to the standard deviation rule for normal distributions:

- 68% of the area lies in the interval of 1 standard deviation from the mean, or mathematically speaking, 68% is in the interval $[\mu - \sigma, \mu + \sigma]$
- 95% of the area lies in the interval of 2 standard deviations from the mean, or mathematically speaking, 95% is in the interval

- 95% of the area lies in the interval of 2 standard deviations from the mean, or mathematically speaking, 95% is in the interval $[(\mu - 2\sigma), (\mu + 2\sigma)]$
- 99% of the area lies in the interval of 3 standard deviations from the mean, or mathematically speaking, 99% is in the interval $[(\mu - 3\sigma), (\mu + 3\sigma)]$

With a $\mu = 0$ and $\sigma = 1$, this means that for the standard normal distribution:

- 68% of the area lies between -1 and 1.
- 95% of the area lies between -2 and 2.
- 99% of the area lies between -3 and 3.

This simplicity makes a standard normal distribution very desirable to work with. The exciting news is that you can very easily **transform** any normal distribution to a standard normal distribution!

Standard Score (z-score)

The standard score (more commonly referred to as a z-score) is a very useful statistic because it allows us to:

1. Calculate the probability of a certain score occurring within a given normal distribution and
2. Compare two scores that are from different normal distributions.

Any normal distribution can be converted to a standard normal distribution and vice versa using this equation:

$$z = \frac{x - \mu}{\sigma}$$

Here, x is an observation from the original normal distribution, μ is the mean and σ is the standard deviation of the original normal distribution.

The standard normal distribution is sometimes called the z-distribution. A z-score always reflects the number of standard deviations above or below the mean.

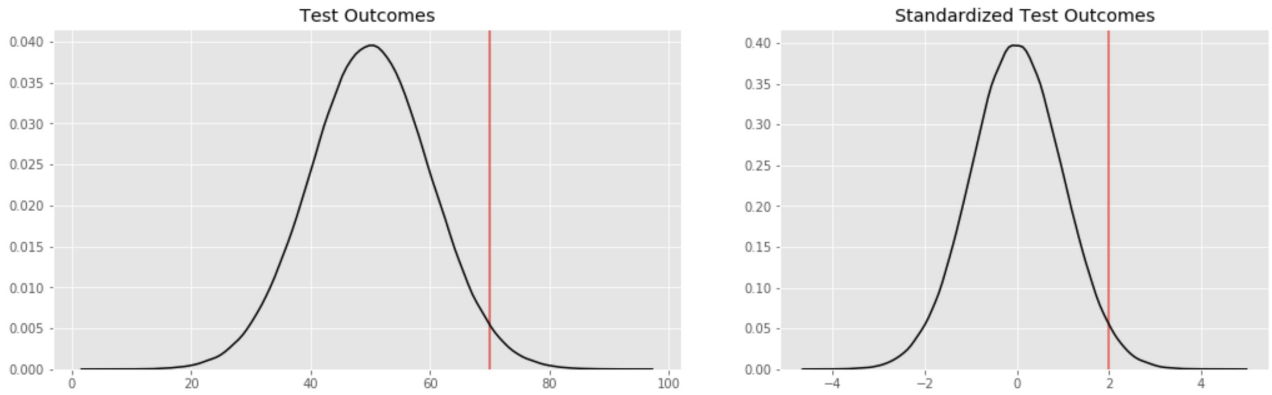
An example

Imagine some test results follow a normal distribution with a mean score of 50 and a standard deviation of 10. One of the students scored a 70 on the test. Using this information and z-scores makes it easy to tell how she performed in terms of standard deviations from the mean. Converting a test score of 70 to a z-score, an x of 70 would be, in this case:

$$z = \frac{70 - 50}{10} = 2$$

By transforming the test result of 70 to a z-score of 2, we now know that the student's original score was 2 standard deviations above the mean score. Note that the z distribution will only be a normal distribution if the original distribution of x was normal.

In summary, calculating the z-score gives us quick and easy access to understanding how **extreme** a certain result is. Looking at the original distribution ($\mu = 50$, $\sigma = 10$) and the standard normal distribution ($\mu = 0$, $\sigma = 1$) while highlighting $x = 70$ and $z = 2$ gives the following result:



Visually, the idea is that the area under the curve, left and right from the vertical red line, are identical in the left plot and the right plot!

Thinking along these lines, you can also convert a z-score back to an original score x by using the same formula as:

$$x = \mu + z\sigma$$

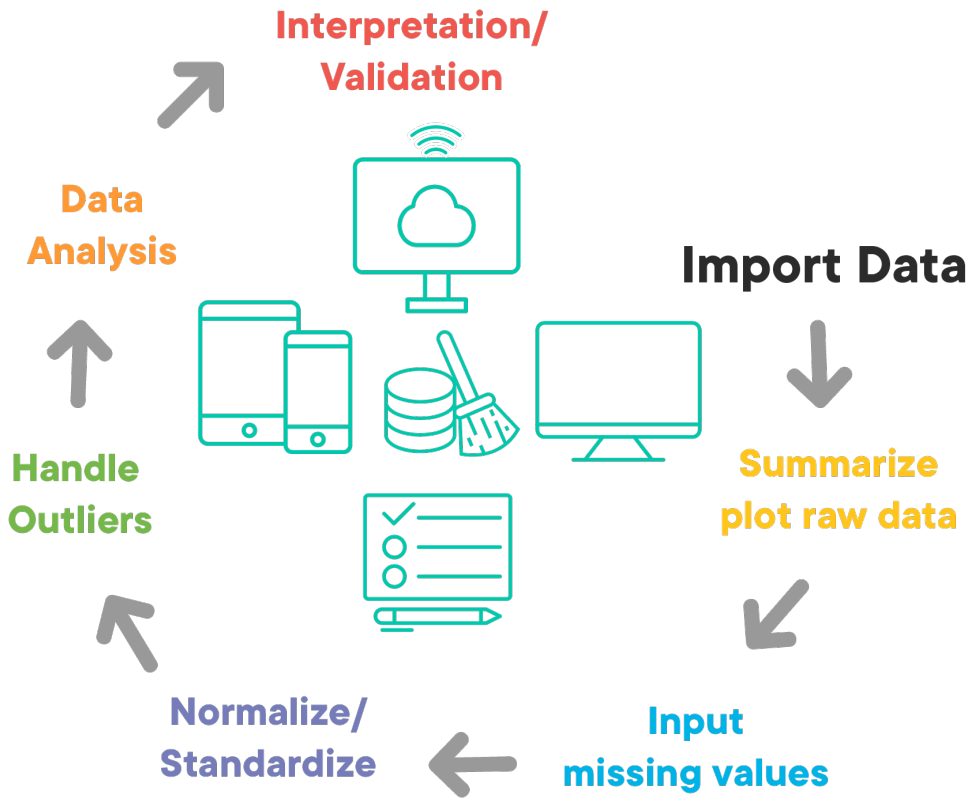
For the above example, this would work out as:

$x = 50 + 2 * 10 = 70$

Data Standardization

Data standardization is a common data preprocessing skill, which is used to compare a number of observations belonging to different normal distributions which may have distinct means and standard deviations.

Standardization applies a z-score calculation, as shown above, on each element of the distribution. The output of this process is a **z-distribution** or a **standard normal distribution**.



Let's look at a quick example. First, we'll randomly generate two normal distributions with different means and standard deviations. Let's generate 1000 observations for each. Next, we'll use `seaborn` to plot the results.

In [1]:

```
import numpy as np
import seaborn as sns
mean1, sd1 = 5, 3 # dist 1
mean2, sd2 = 10, 2 # dist 2
d1 = np.random.normal(mean1, sd1, 1000)
d2 = np.random.normal(mean2, sd2, 1000)
sns.distplot(d1);
sns.distplot(d2);
```

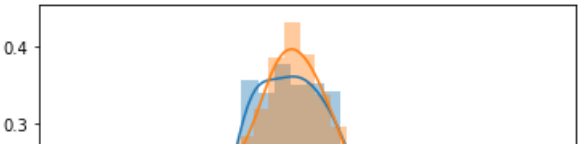
You can see that these distributions differ from each other and are not directly comparable.

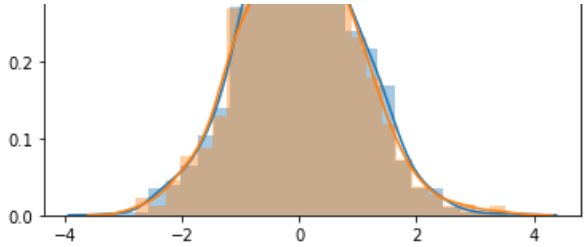
For a number of machine learning algorithms and data visualization techniques, it is important that the effect of the scale of the data is removed before you start thinking about building your model. Standardization allows for this by converting the distributions into a z-distribution, bringing them to a common scale (with $\mu = 0$, $\sigma = 1$). Let's standardize the above distributions and look at the effect.

In [2]:

```
# Stardardizing and visualizing distributions

sns.distplot([(x - d1.mean())/d1.std() for x in d1]);
sns.distplot([(x - d2.mean())/d2.std() for x in d2]);
```





You see that both distributions are directly comparable on a common standard scale. As mentioned earlier, this trick will come in handy with analytics experiments while training machine learning algorithms.

Level up (Optional)

Convert standard distributions back to the original normal distributions using the formula given above. Visualize them to see your original distributions.

Summary

In this lesson, you learned about a special case of the normal distribution called the standard normal distribution. You also learned how to convert any normal distribution to a standard normal distribution using the z-score. You'll continue working on this in the following labs.

Statistical Testing with z-score and p-value

Introduction

In this lesson, you'll learn how the z-score is important when performing statistical tests. This lesson is meant to be a segue from standard normal distributions into testing, but you'll learn about testing in more detail later!

Objectives

You will be able to:

- Compare and contrast a population and a sample
- Explain what is meant by a "representative" sample
- Use the z-table and `scipy` methods to acquire the p value for a given z-score
- Define what null and alternative hypotheses mean and when they are used
- Define the significance threshold and its relation to p-value

Statistical significance

Statistical significance is one of those terms that is often used when someone claims that some data collection and analysis proves a **certain point** (or hypothesis). The terminology around statistical significance is often not well understood, however, it is a simple idea that can be understood fairly easily.

Statistical significance is based on a few concepts: samples and populations, hypothesis testing, the normal distribution, and p-values. In this lesson, we'll either repeat or introduce all of the foundational concepts associated with statistical significance.

First, let's look at how to differentiate between samples and populations.

Population vs sample

The first step of every statistical analysis you will perform is the population vs. sample check or to determine whether the data you are dealing with is either a **population** or a **sample**.

> A **population** is the collection of **all the items of interest in a study**. The numbers you obtain when using a population are called **parameters**.
>
> A **sample** is a **subset of the population**. The numbers you obtain when working with a sample are called **statistics**.

Example

Imagine we want to conduct a survey of the job prospects of the students studying at Flatiron School.

What is the population?

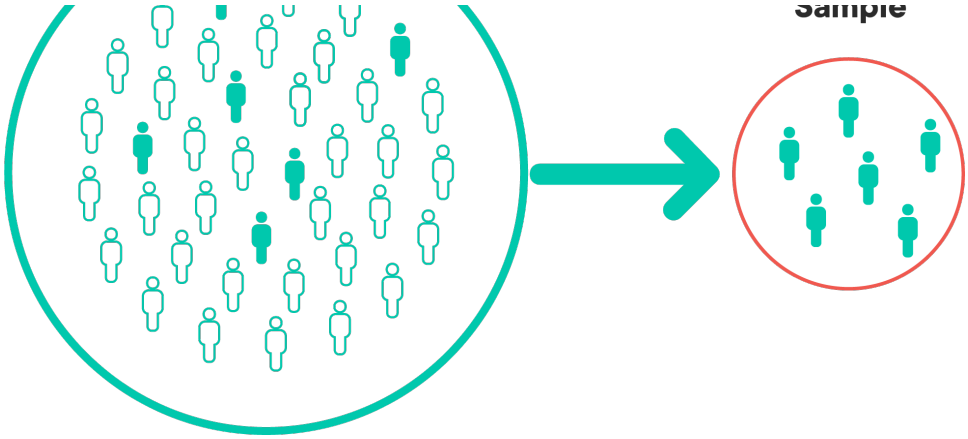
You can go ahead and contact all the students studying at Flatiron School campuses around the world in our physical campuses. But that would not be the whole population of Flatiron students. A lot of students take Flatiron's courses online. Including those students from all over the world would make a COMPLETE student population at Flatiron. Also, by the time you finish contacting all these students, you would realize that a lot of new students have enrolled in the courses since you last surveyed.

So you can see that populations are hard to describe and inspect in real life.

What is a sample?

A sample is much easier to describe and inspect. Conducting a survey on a sample is less time-consuming and less costly too. Time and resources are the main reasons we prefer drawing samples over working with entire populations.

As we first wanted to do, we can just go to the New York campus. We can visit during the lunch hour in the canteen because we know it will be full of people. We can then interview 50 of them. This would be called a student sample.



Is the sample "representative" ?

So what are the chances these 50 students can provide us answers that are a true representation of the whole student population of Flatiron School globally? You guessed it, the chances are pretty low. The sample is *neither* **random** nor **representative** of the whole population.

> A random sample is collected when each member of the sample is chosen from the population strictly by chance. In order for the sample to be random, each member should be equally likely to be chosen.

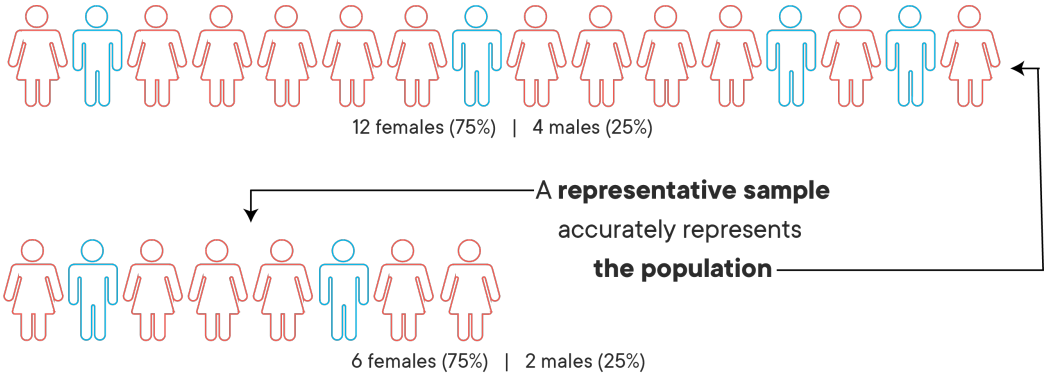
It is clear that these two requirements were violated by picking 50 students from the NYC campus.

How can you make it a representative sample?

> A representative sample is a subset of the population that accurately reflects the members of the entire population

Our sample represented the NYC campus students who have lunch at the canteen. If we ran a survey about job prospects of Flatiron students who eat in the NYC campus canteen, we would have done well.

By now, you must be wondering how to draw a sample that is both random and representative. Well, the safest way would be to get access to the student database with **all students around the world** and contact individuals in a random manner.



We said populations are hard to define and observe. Then, we saw that sampling is difficult. But samples have two big advantages:

1. Once well-understood, it is not that hard to recognize if a sample is a representative one
2. Statistical tests are designed to work with incomplete data, so making a small sampling error is not always a problem

Now that we understand using samples vs. populations, we can move on with statistical testing.

Statistical Testing

There are a huge number of statistical tests and you will always try to select the one that best fits your research design. This link provides a quick introduction to a number of testing criteria. We'll cover some of these in-depth later on.

Right now, let's talk about one of the most basic types of statistical testing techniques based on the z-score, the one-sample z-test.

The One-sample z-test

The one-sample z-test is used when you want to know if your sample comes from a particular population.

For instance, when collecting data from successive cohorts of students taking the Data Science Bootcamp, you may want to know if

this particular sample of students is similar to or different from Flatiron students in general.

> **The one-sample z-test is used only for tests related to the sample mean.**

When running a one-sample z-test, you test whether the average of the sample suggests that the students come from a certain population with a known mean or whether it may come from a different population.

You already know what a z-score is and how to standardize a dataset into a z-distribution. By itself, the z-score doesn't provide a lot of information to conclude a question significantly (except for saying how many standard deviations some observation is from a mean). The real value from a z-test comes from comparing it against a **z-table**.

The z-table

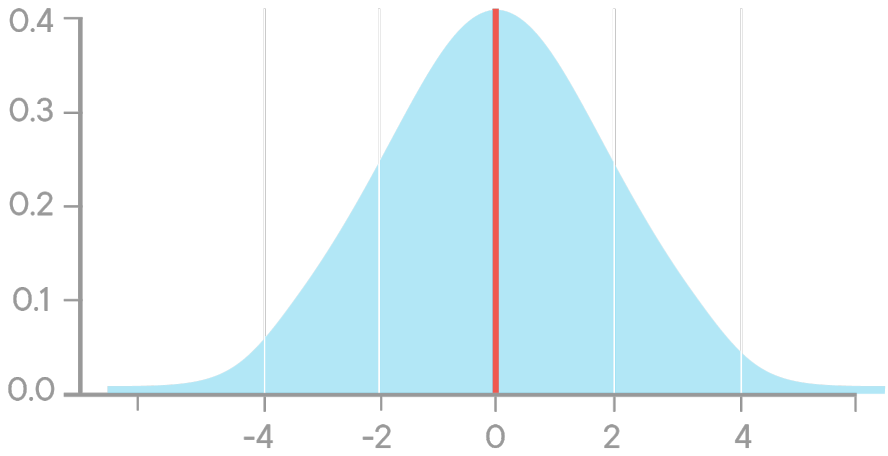
A z-table contains cumulative probabilities of a standard normal distribution up until a given z-score value. By transforming normal distributions with various means and standard deviations, you can use this z-table for any value that follows a normal distribution. The z-table is short for the "Standard Normal z-table".

> **The area under the whole of a normal distribution curve is 1, or 100 percent. The z-table helps by telling us what percentage is under the curve up to any particular point.**

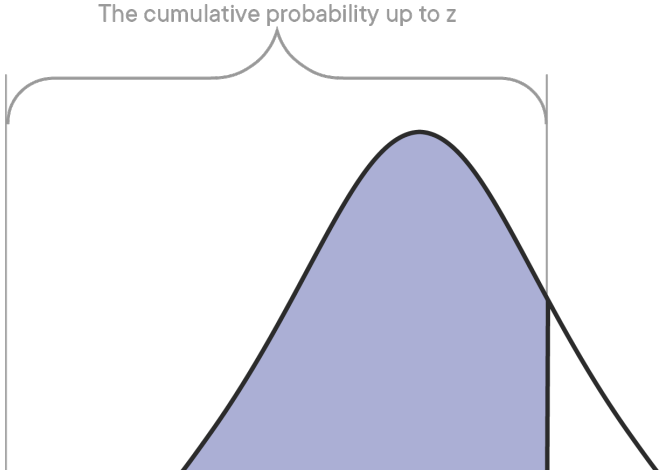
Here is a link to an online version of z-table. This lesson's GitHub repository also contains a pdf version of this table, `z-table.pdf`.

- **The rows** of the table contain z-values in the form $x.x$ along the left margins of the table, specifying the ones and tenths.
- **The columns** fine-tune these values to hundredths, allowing us to look up the probability of being below any standardized value z of the form $x.xx$.

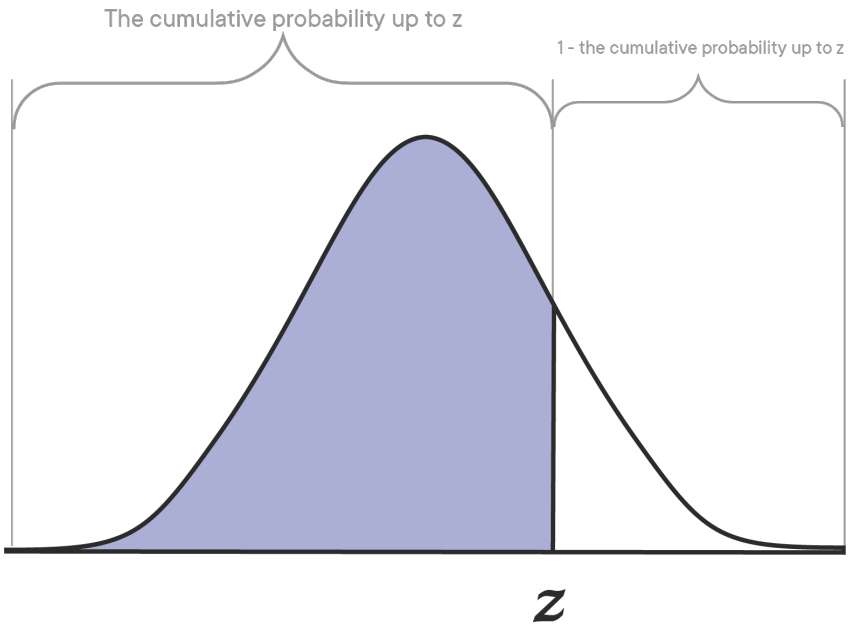
You know that a cumulative probability is the sum of the probabilities of all values up until a given point. An easy example is the mean. The mean is the exact middle of the normal distribution, so we know that the sum of all probabilities of getting values from the left side up until the mean is 0.5. Also, the sum of probabilities from the mean to right tail would also sum up to 0.5. The mean is denoted by the red line in the image below.



When using the idea of cumulative probability in the context of the standard normal distribution, we look at the cumulative distribution until a point z.



As the sum of all probabilities is equal to 1 or 100%, you can use the z-table to calculate probabilities on both sides of the z-score under the standard normal distribution.



Using z-scores, you can answer questions like "how far is a value from the mean" and "how likely is a value this far from the mean to be from the same group of observations?"

Example

> What is the probability of a z-score being less than or equal to 1.5?

To find the answer using the z-table, you have to go and look where the row for 1.5 intersects with the column for 0.00; this value is 0.9332. The z-table shows only “less than” probabilities so it gives you exactly what you need for this question. The probability of a z-score being less than or equal to 1.5 is 0.9332.

> What is the probability of a z-score being greater or equal to 1.34?

Use the z-table to find where the row for 1.3 intersects with the column for 0.04, which is 0.9099. Because the z-table gives you only "less than" probabilities, subtract P(Z < 1.34) from 1 (remember that the total probability is 1.00 or 100%). So, **1 - 0.9099 = 0.0901**.

Here is a short video on how to use a z-table

The z-table alternative in python

When programming in Python, SciPy provides a handful of features so you won't have to go out consulting z-tables for automated analysis. For normal distributions, probabilities **up to the z-score** can be calculated with `.cdf` -method as shown below:

In [1]:

```
# Z-table in Python
import scipy.stats as stats

# Probabilities up to z-score of 1.5
print(stats.norm.cdf(1.5))

# Probabilities greater than z-score of 1.34
print (1-stats.norm.cdf(1.34))
```

```
0.9331927987311419
0.09012267246445238
```

What Are Hypotheses ?

A very important aspect of statistical testing is setting up a hypothesis to be tested through data analysis.

The hypothesis is a data scientist's initial understanding of an observation prior to the testing. This hypothesis is known as the **Alternative Hypothesis** (written as H_a).

The opposite to the Alternative Hypothesis is known as a **Null Hypothesis** (written as H_0).

The table below shows three sets of Null and Alternative Hypotheses. Each makes a statement about how the mean μ is related to some hypothesized value M.

Set	H_0	H_a	Tails
1	$\mu = M$	$\mu \neq M$	2
2	$\mu \geq M$	$\mu < M$	1
3	$\mu \leq M$	$\mu > M$	1

Here the tails represent if we are testing both sides of the distribution or only one side.

As an example, an analyst may want to check how effective a new drug is by setting an Alternative Hypothesis that the new drug reduces blood pressure by 10%.

In this case, the Null Hypothesis states that the new drug has no effect on the patients. When performing hypothesis testing, you generally will try to reject the null hypothesis to obtain what we call "Statistically Significant Results".

P-value

You will now learn about the **p-value** as a statistical summary of the compatibility between the observed data and what you would expect to see in a population assuming the statistical model is correct. The concepts of p-value and level of significance are vital components of hypothesis testing and methods like regression. However, they can be a little tricky to understand. We'll try to explain the concept in an easy, logical way.

> In hypothesis testing you set a null hypothesis, then draw a sample, and test your null hypothesis based on that sample.

For example, imagine your null hypothesis H_0 is that the population mean μ is $\mu = 10$. Upon drawing a sample, you get a mean of 12. **With the p-value, you are going to obtain a probability that, given a null hypothesis of $\mu = 10$, you would observe a sample mean of 12.**

> If your p-value is low, you will reject your null hypothesis. You will basically say that **based on current evidence and testing, the null hypothesis is not true.**
>
> If your p-value is high, you will fail to reject your null hypothesis. You will fail to reject the null hypothesis, that is, you will say that **based on current evidence and testing, the null hypothesis cannot be rejected.**

You'll see that the phrase "accepting a null hypothesis" is not used. This is because conclusions of hypothesis tests will state that "we reject H_0 in favor of H_a" or that "we cannot reject H_0 in favor of H_a", which is less definitive and leaves room for errors while testing. You reject or fail to reject a null hypothesis based on the evidence you have.

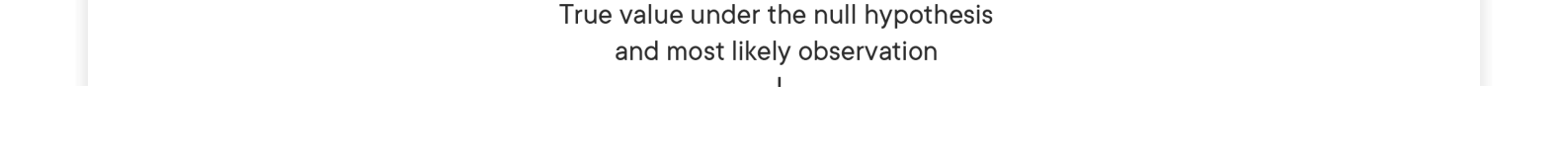
> It is important to understand what you have **assumed** and what you have **observed**

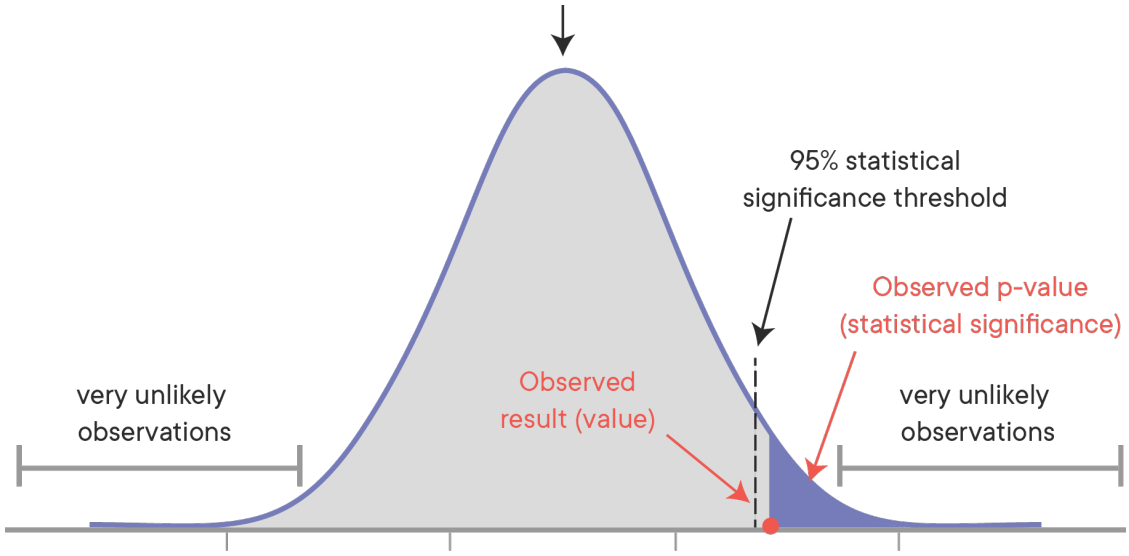
You assumed your population mean is 10, without actually observing that. You have observed a sample mean of 12 after testing your sample.

You then verify whether the sample mean you obtained is consistent with the population mean you have assumed. In other words, what are the chances of getting the result (sample mean) if the assumption is actually true (population mean). What is the probability that the sample mean is 12, assuming that the population mean is 10?

This chance or probability is called a p-value.

If your p-value is low, you say that that the result is **significant**, in the sense that you conclude that the sample mean is **significantly different** from the population mean.





What is the Significance Threshold (alpha, α)?

You noticed that we talked about "high" and "low" p-values, but that is pretty vague. What number is high and what number is low?

This is where the significance level, also denoted as alpha or α comes in. α is the threshold value that defines whether a p-value is low or high. You can define your alpha level yourself, but you'll see that an alpha level of $\alpha = 0.05$ is most commonly used. You'll see $\alpha = 0.1$ and $\alpha = 0.01$ appear frequently as well.

What level of alpha to use depends on your situation. Choosing a lower alpha leads to a test that is more strict, so you will be less likely to be able to reject your null-hypothesis (which is generally what you want). Choosing a higher alpha or significance level leads to a higher probability of rejecting the null-hypothesis. The downside of using a higher alpha level, however, is that you run a higher risk of falsely concluding that there is a difference between your null-hypothesis and your observed results when there actually isn't any.

This may all seem a little vague for now. You'll get a better understanding when we dig deeper later on.

Summary

In this lesson, you learned about the basics of hypothesis testing and went through the concepts and terminologies that are used while performing statistical tests. Now, let's move on to perform a one-sample z-test to validate a hypothesis that the sample drawn belongs to the same population.

One-Sample z-Test

Introduction

A one-sample z-test is the most basic type of hypothesis test. It is performed when the population means and standard deviation are known. This makes the analysis very simple. The main takeaway from this lesson and the next lab is to have an idea around the process of hypothesis testing and understanding test statistics and p-values.

Objectives:

You will be able to:

- Explain use cases for a 1-sample z-test
- Set up null and alternative hypotheses
- Use the z-table and scipy methods to acquire the p value for a given z-score
- Calculate and interpret p-value for significance of results

One-Sample z-test

The one-sample z-test is best suited for situations where you want to investigate whether a given "sample" comes from a particular "population".

The best way to explain how one-sample z-tests work is by using an example.

Let's set up a problem scenario (known as a research question or analytical question) and apply a one-sample z-test, while explaining all the steps required to call our results "statistically significant".

The Analytical Question

A data scientist wants to examine if there is an effect on IQ scores when using tutors. To analyze this, she conducts IQ tests on a sample of 40 students and wants to compare her students' IQ to the general population IQ. The way an IQ score is structured, we know that a standardized IQ test has a mean of 100 and a standard deviation of 16. When she tests her group of students, however, she gets an average IQ of 103. Based on this finding, does tutoring make a difference?

Step 1: State Your Hypotheses

The Alternative Hypothesis (H_a)

The alternative hypothesis always reflects the idea or theory that needs to be tested. For this problem, you want to test if tutoring has resulted in a significant increase in student IQ. So, you would write it down as:

> The sample mean is **significantly** bigger than the population mean

Again, significance is key here. If we denote the sample mean as $\bar{x}$, and population mean as mu (μ), you can write the alternative hypothesis as:

$$H_a: \mu < \bar{x}$$

The alternative hypothesis here is that the population mean μ is less than the sample mean $\bar{x}$. In other situations, you could check for both possibilities of μ being smaller OR bigger than by checking $\mu \neq \bar{x}$.

Maybe the tutoring results in a lower IQ... Who knows!

For now, you'll just check for a **significant increase**, for now, to keep the process simple.

The Null Hypothesis (H_0)

For a one-sample z-test, you define your null hypothesis as there being **no significant difference** between the specified sample and population means. In other words, under the null hypothesis, you assume that any observed (generally small) difference between the sample and population means may be present due to sampling or experimental error. Considering this, for this problem, you can define a null hypothesis (H_0) as:

> There is **no significant difference** between the sample mean and population mean

Remember the emphasis is on a *significant* difference, rather than just any difference as a natural result of taking samples.

Denoting the sample mean as $\bar{x}$, and the population mean as μ, you can write the null hypothesis as:

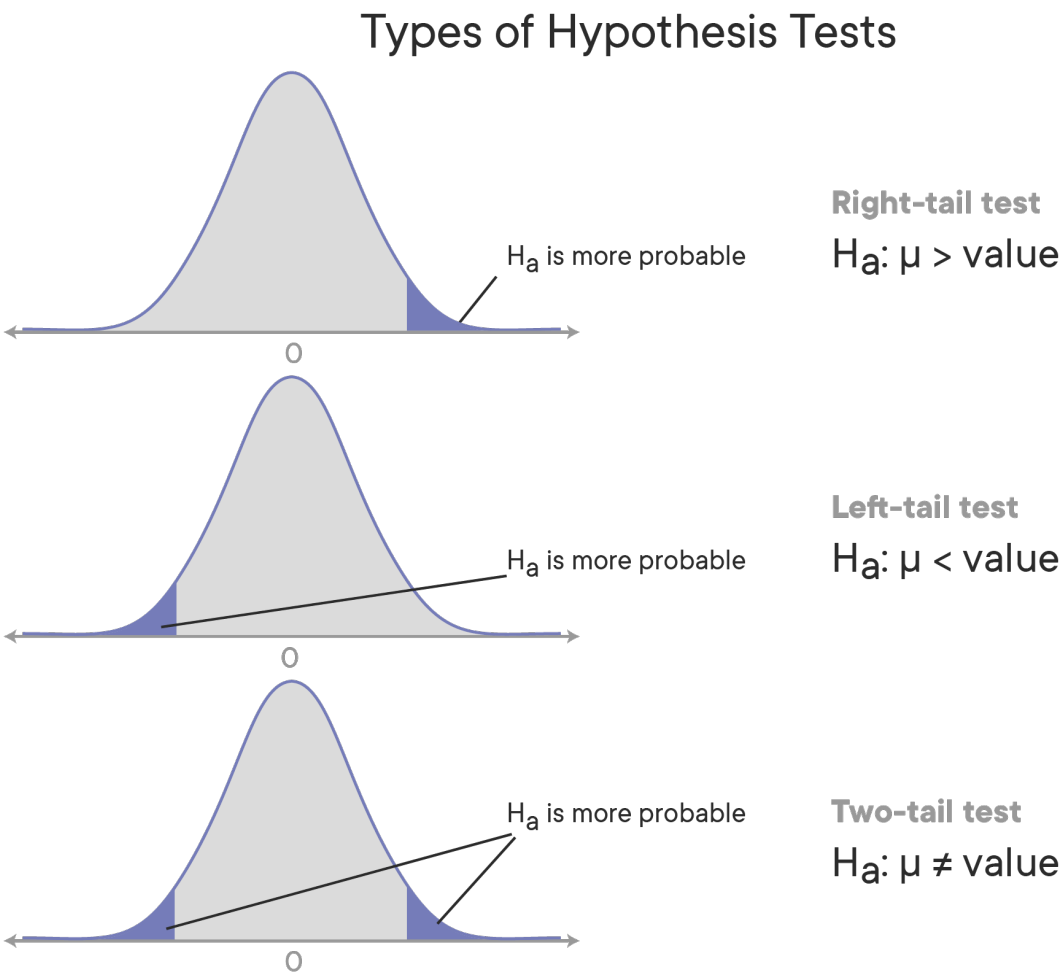
$$H_0: \mu \geq \bar{x}$$

Step 2: Specify a Significance Level (alpha)

Now that your hypotheses are in place, you have to decide on your significance level alpha (α) as a cut-off value to define whether you can reject your null hypothesis or not.

As discussed previously, often, α is set to 0.05, which also has as a side-effect that there is a 5 percent chance that you will reject the null hypothesis when it is true.

Later, you'll see that using α, you'll formulate your test result as: "with a confidence level of 95%, we can state that...". For a z-distribution, this can be shown as below:



If you test both sides of the distribution ($\mu \neq \bar{x}$, when μ can either be smaller OR bigger), you need to perform a two-tail test to see if tutoring results in lower OR higher IQs.

Each purple region would be calculated as $\frac{\alpha}{2}$. When testing a single side (as in the example) i.e. just higher OR just lower, you can use a one-tail test as shown in the first and second images. The α value we use is 0.05 or 5%.

Step 3: Calculate the test statistic

For z-tests, a z-statistic is used as our test statistic. You'll see other test statistics suitable for other tests later. A one-sample z-statistic is calculated as:

$$\text{z-statistic} = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}$$

This formula slightly differs from the standard score formula. It includes the square root of n to reflect that we are dealing with the sample variance here.

Now, all you need to do is use this formula given your sample mean $\bar{x}$, the population standard deviation σ, and the number of items in the sample (n). μ_0 is the mean you're testing the hypothesis for, or the "hypothesized mean".

Let's use Python to calculate this.

In [1]:

```
import scipy.stats as stats
from math import sqrt
x_bar = 103 # sample mean
n = 40 # number of students
sigma = 16 # sd of population
mu = 100 # Population mean

z = (x_bar - mu)/(sigma/sqrt(n))
z
```

Out[1]:

```
1.1858541225631423
```

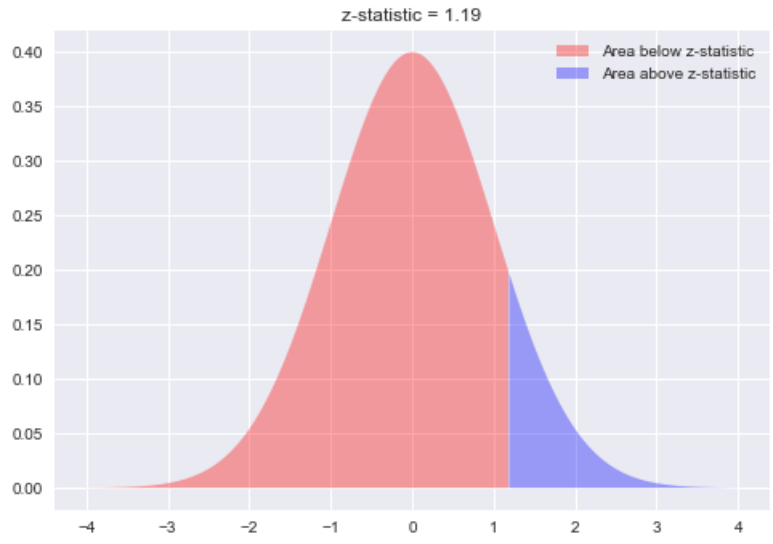
Let's try to plot this z-value on a standard normal distribution to see what it means.

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

plt.style.use('seaborn')
plt.fill_between(x=np.arange(-4,1.19,0.01),
                 y1= stats.norm.pdf(np.arange(-4,1.19,0.01)) ,
                 facecolor='red',
                 alpha=0.35,
                 label= 'Area below z-statistic'
                 )

plt.fill_between(x=np.arange(1.19,4,0.01),
                 y1= stats.norm.pdf(np.arange(1.19,4,0.01)) ,
                 facecolor='blue',
                 alpha=0.35,
                 label= 'Area above z-statistic')
plt.legend()
plt.title ('z-statistic = 1.19');
```



Step 4: Calculate the p-value

Remember that z-values in a standard normal distribution represent standard deviations. Just like before, you need to look up the related probability value in a z-table, or use `scipy.stats` to calculate it directly. In SciPy, the cumulative probability up to the z-value can be calculated as:

In [3]:

```
stats.norm.cdf(z)
```

Out[3]:

```
0.8821600432854813
```

The percent area under the curve from to a z-score of 1.19 is 88.2% (using the z-table or SciPy calculations), this means that the average intelligence of the tutored set of students is bigger than 88.2% of the population. But with alpha specified as 0.05, we wanted it to be greater than 95% to prove the hypothesis to be significant.

Mathematically, you want to get the p-value, and this can be done by subtracting the z-value from 1, since the sum of probabilities is always 1.

In [4]:

```
pval = 1 - stats.norm.cdf(z)
pval
```

Out[4]:

```
0.11783995671451875
```

Step 5: Interpret p-value

Our p-value (0.12) is larger than the alpha of 0.05. So what does that mean? Can you not conclude that tutoring leads to an IQ increase?

Well, you still can't really say that for sure. **What we can say is that there is not enough evidence to reject the null hypothesis with the given sample, given an alpha of 0.05.** There are ways to scale experiments up and collect more data or apply sampling techniques to be sure about the real impact.

And even when the sample data helps to reject the null hypothesis, you still cannot be 100% sure of the outcome. What you can say, however, is given the evidence, the results show a significant increase in the IQ as a result of tutoring, instead of saying "tutoring improves IQ".

Summary

In this lesson, you learned to run a one-sample z-test to compare sample and population where the population mean and standard deviation are known. This is the most basic test in statistics, but in the real world, the true population means and standard deviations are rarely identifiable and you need to work with samples. That's where more advanced tests come in to play, which you will learn about later.

Skewness and Kurtosis

Introduction

We have previously identified a normal distribution to be symmetrical in shape. But when you're dealing with real-world data you'll often come across asymmetric distributions as well. In this lesson, you'll learn how to measure asymmetry (or skewness) in a distribution. Additionally, you'll learn about kurtosis. Kurtosis defines whether a distribution is truly "normal" or whether it may have so-called "fatter" or "thinner" tails than you would observe when data are normally distributed.

Objectives

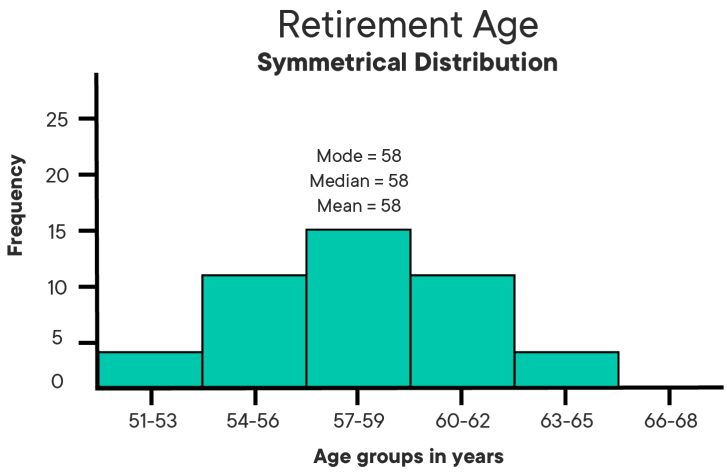
You will be able to:

- Define skewness and kurtosis and their relationship to symmetric distributions

Symmetric Distributions

A distribution is symmetric if the relative frequency or probability of certain values are equal at equal distances from the point of symmetry. The point of symmetry for normal distributions is the mean (and at the same time median and mode!)

Have a look at following histogram:



This distribution meets all of the conditions of being symmetrical.

The most common symmetric distribution is the normal distribution, however, there are a number of other distributions that are symmetric. Here is a good article that looks into all sorts of symmetrical distributions. We'll focus on normal distributions (by far the most common group) here, and see how these can lose symmetry!

Skewness

Skewness is the degree of distortion or deviation from the symmetrical normal distribution. Skewness can be seen as a measure to calculate the lack of symmetry in the data distribution.

Skewness helps you identify extreme values in one of the tails. Symmetrical distributions have a skewness of 0.

Distributions can be **positively** or **negatively** skewed.

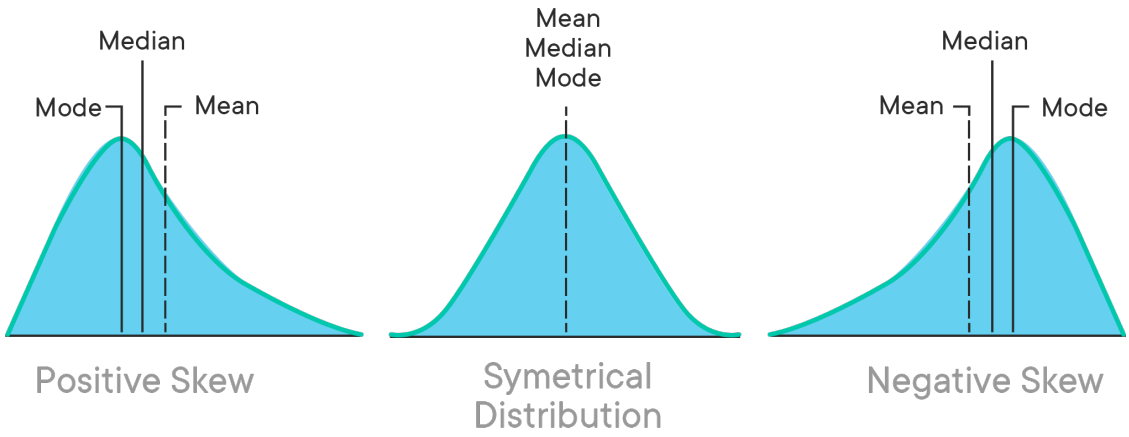
Positive Skewness

A distribution is **positively skewed** when the tail on the right side of the distribution is longer (also often called "fatter"). When there is positive skewness, the mean and median are bigger than the mode.

Negative Skewness

Distributions are **negatively skewed** when the tail on the left side of the distribution is longer or fatter than the tail on the right side. When there is negative skewness, the mean and median are smaller than the mode.

This behavior is shown in the images below:



Skewness can have implications for data analysis and the usage of certain models. The "normality assumption" seen before does not hold when data is skewed. When data is skewed, you'll need to transform the data first.

Measuring Skewness

For univariate data $Y_1, Y_2, \ldots, Y_n$ the formula for skewness is:

$$\frac{\sum_{i=1}^{n}(Y_i - Y)^3}{\frac{n}{s^3}}$$

where Y is the mean, s is the standard deviation, and n is the number of data points. This formula for skewness is referred to as the **Fisher-Pearson coefficient of skewness**. There are also other ways to calculate skewness, yet this one is the one that is used most commonly.

Using this formula, when is data skewed?

The rule of thumb seems to be:

- A skewness between -0.5 and 0.5 means that the data are pretty symmetrical
- A skewness between -1 and -0.5 (negatively skewed) or between 0.5 and 1 (positively skewed) means that the data are moderately skewed.
- A skewness smaller than -1 (negatively skewed) or bigger than 1 (positively skewed) means that the data are highly skewed.

Example

Imagine you have house values ranging from 200,000 USD to 1,500,000 USD with an average of 800,000 USD.

If the peak of the distribution is left of the average value, the house prices are positively skewed. This means that more than half of the houses were sold for less than the average value 800,000 USD, and that there are a limited number of houses that were sold for a *much* higher value than 800,000 USD, leading to a long tail in the higher price ranges.

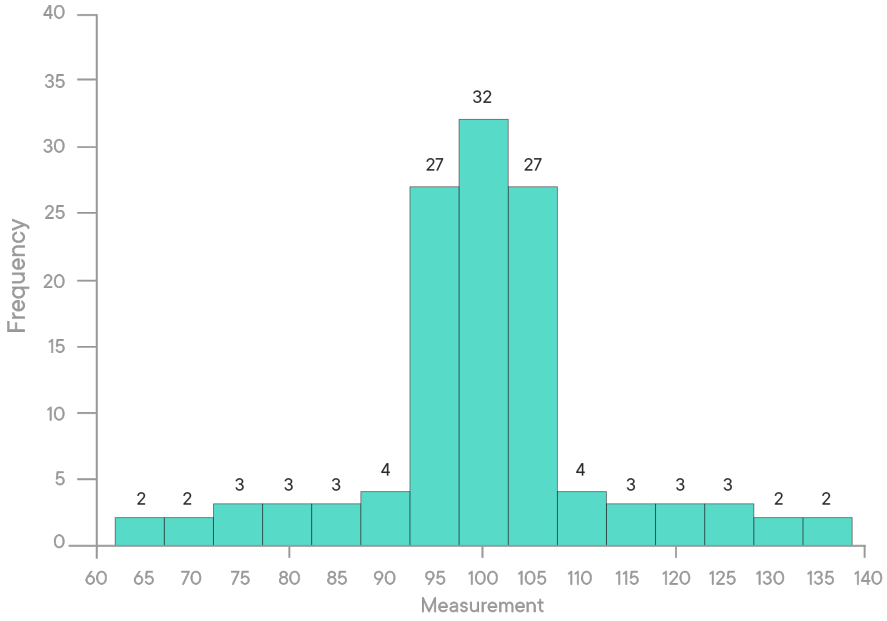
If the peak of the distributed data is on the right-hand side of the average value, this means there is negative skewness, meaning that more than half of the houses were sold for more than the average value of 800,000 USD. Additionally, this means that there is a long tail in the lower price ranges.

Kurtosis

Kurtosis deals with the lengths of tails in the distribution.

> **Where skewness talks about extreme values in one tail versus the other, kurtosis aims at identifying extreme values in both tails at the same time!**

You can think of Kurtosis as a **measure of outliers** present in the distribution.



The distribution denoted in the image above has relatively more observations around the mean, then a steep decline and longer tails compared to the normal distribution.

Measuring Kurtosis

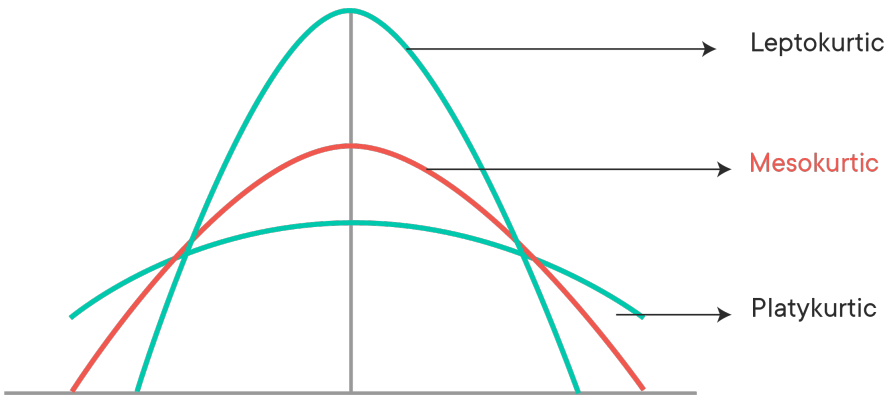
For univariate data $Y_1, Y_2, \ldots, Y_n$ the formula for kurtosis is:

$$\frac{\sum_{i=1}^{n}(Y_i - Y)^4}{\frac{n}{s^4}}$$

If there is a high kurtosis, then you may want to investigate why there are so many outliers. The presence of outliers could be indications of errors on the one hand, but they could also be some interesting observations that may need to be explored further. For banking transactions, for example, an outlier may signify fraudulent activity. How we deal with outliers mainly depends on the domain.

Low kurtosis in a data set is an indication that data has light tails or lacks outliers. If we get low kurtosis, then also we need to investigate and trim the dataset of unwanted results.

How much kurtosis is bad kurtosis?



Mesokurtic ($kurtosis \approx 3$)

A mesokurtic distribution has kurtosis statistics that lie close to the ones of a normal distribution. Mesokurtic distributions have a kurtosis of around 3. According to this definition, the standard normal distribution has a kurtosis of 3.

Platykurtic ($kurtosis < 3$):

When a distribution is platykurtic, the distribution is shorter and tails are thinner than the normal distribution. The peak is lower and broader than Mesokurtic, which means that the tails are light and that there are fewer outliers than in a normal distribution.

Leptokurtic ($kurtosis > 3$)

When you have a leptokurtic distribution, you have a distribution with longer and fatter tails. The peak is higher and sharper than the peak of a normal distribution, which means that data have heavy tails and that there are more outliers.

Outliers stretch your horizontal axis of the distribution, which means that the majority of the data appear in a narrower vertical range. This is why the leptokurtic distribution looks "skinny".

Summary

In this lesson, you learned about skewness and kurtosis. In the next lab, you'll learn how to measure skewness and kurtosis in Python.