NNSE 784
Advanced Analytics Methods

Instructor: F Doyle (CESTM L210)

MW 4:30 – 5:50, NFN 203

Slide Set #7
The Normal Distribution

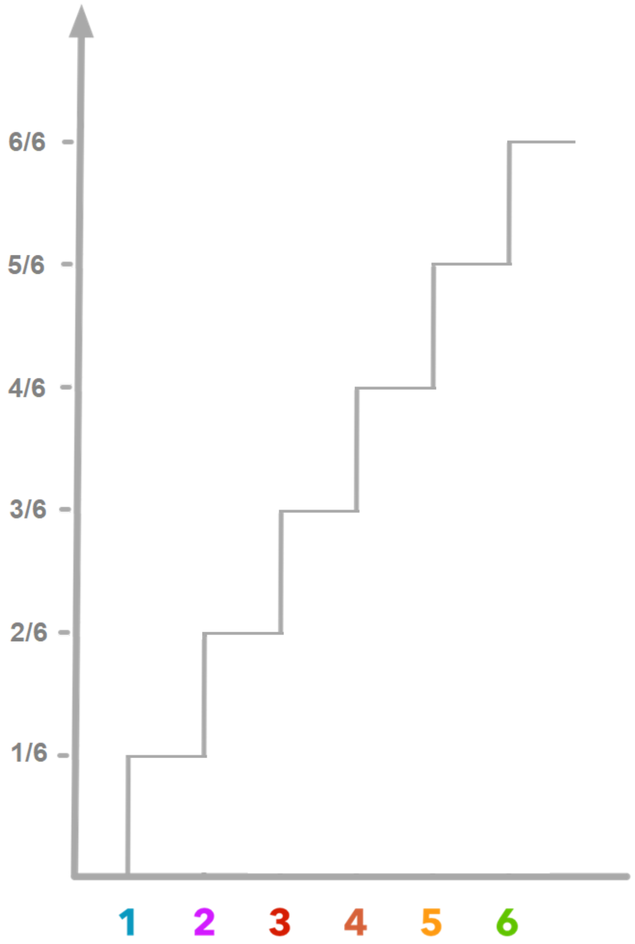
Outline for lecture

- Recap - cumulative probability (discrete and continuous)
- Introduce The Normal Distribution:
  - PDF
  - Characteristics
  - Z-score
- Example z-scores from tests with same mean and test score
- Calculating z-scores and associated probabilities with Python
- Checking sample data for normalcy

Recap - Cumulative Distribution Function

What is the probability of rolling a particular number or lower?

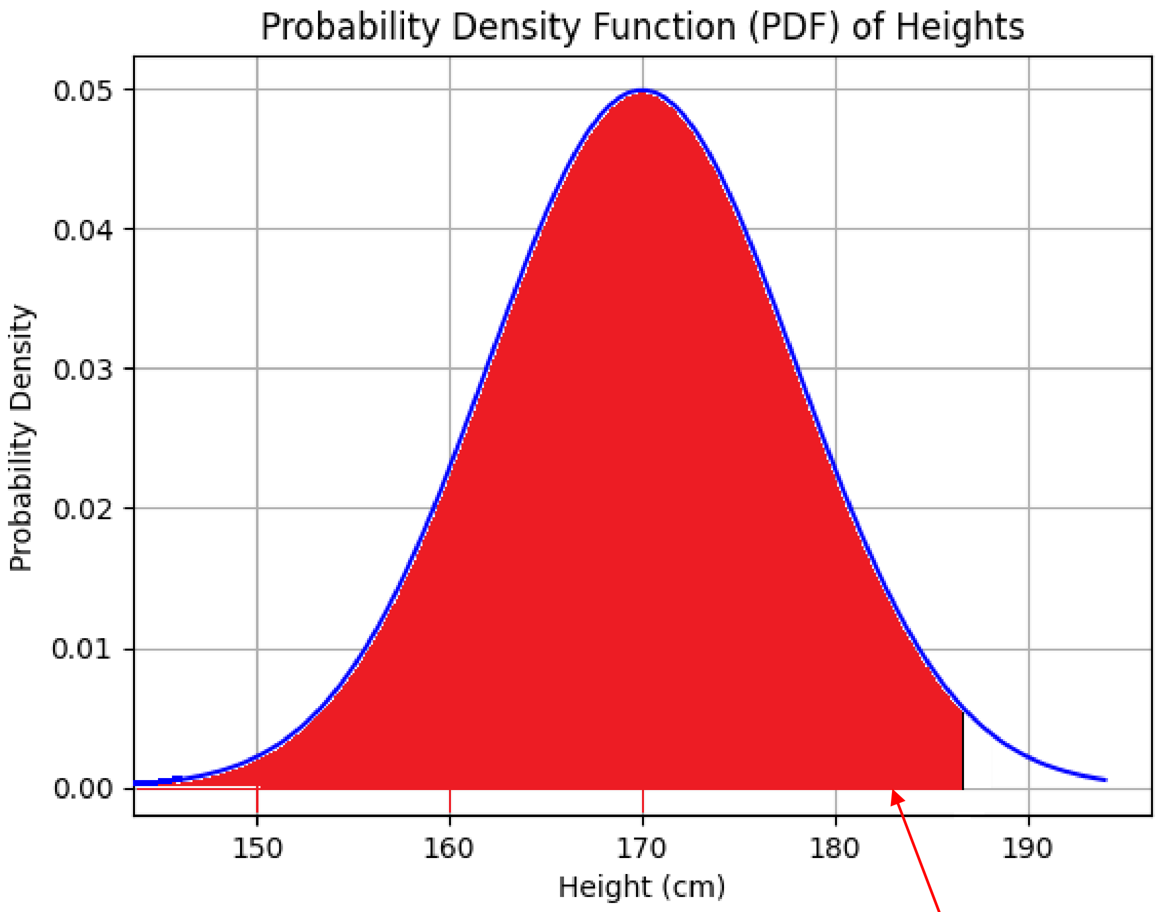
Probability on y axis

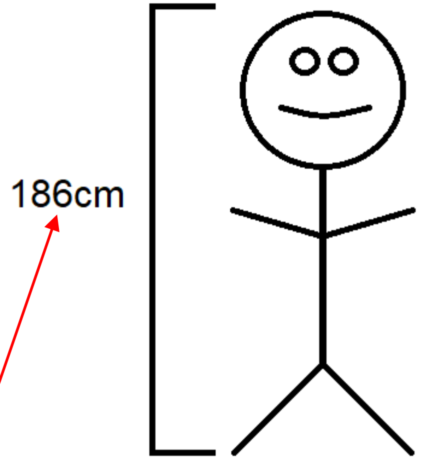


Remember each **individual** outcome had same probability:
$P(y=1) = P(y=2) = P(y=3) = P(y=4) = P(y=5) = P(y=6) = \frac{1}{6}$

Also…the probabilities of all possible outcomes must sum to 1

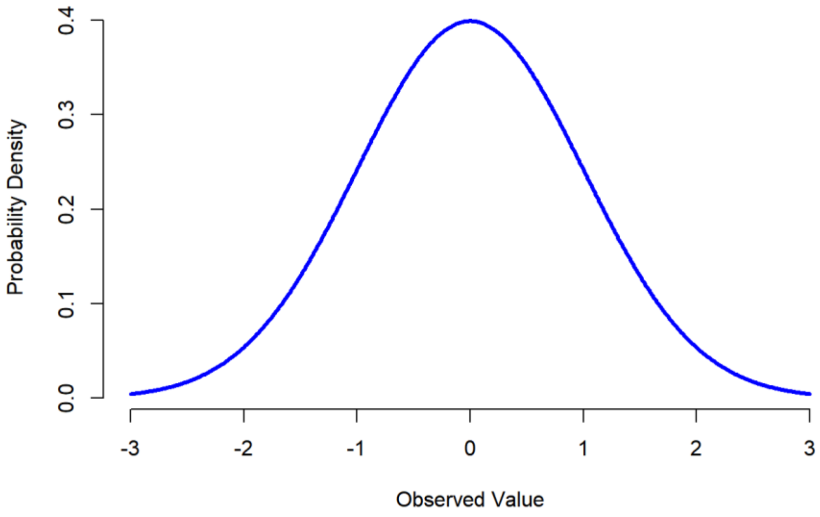
Recap - Probability Density Function





The cumulative probability of randomly picking someone 186cm or smaller

The Normal Distribution AKA Gaussian Distribution AKA "Bell Curve"



The Normal Distribution's Probability Density Function

Don't memorize!

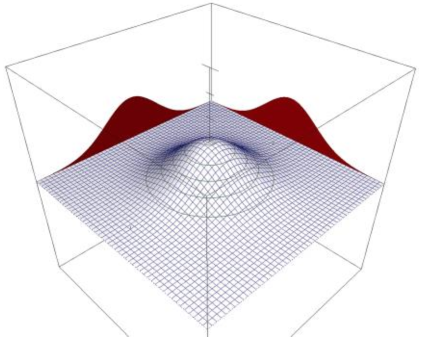
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Variables ("parameters":
μ = mean
σ = standard deviation

Constants:
$\pi = 3.14159$
e = 2.71828

Huh?

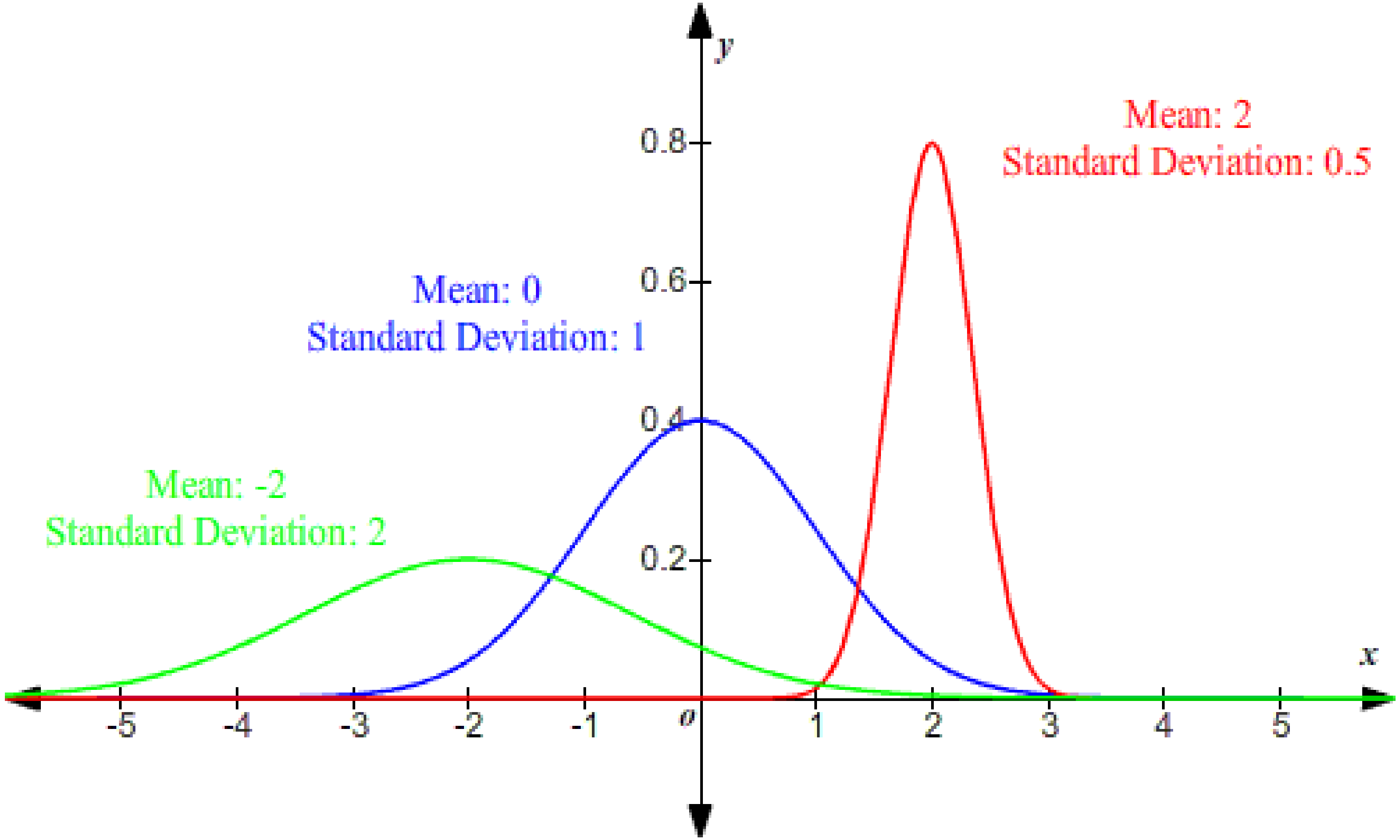
If interested, Google "why is pi in the normal distribution". Not at all required for this class, but interesting!



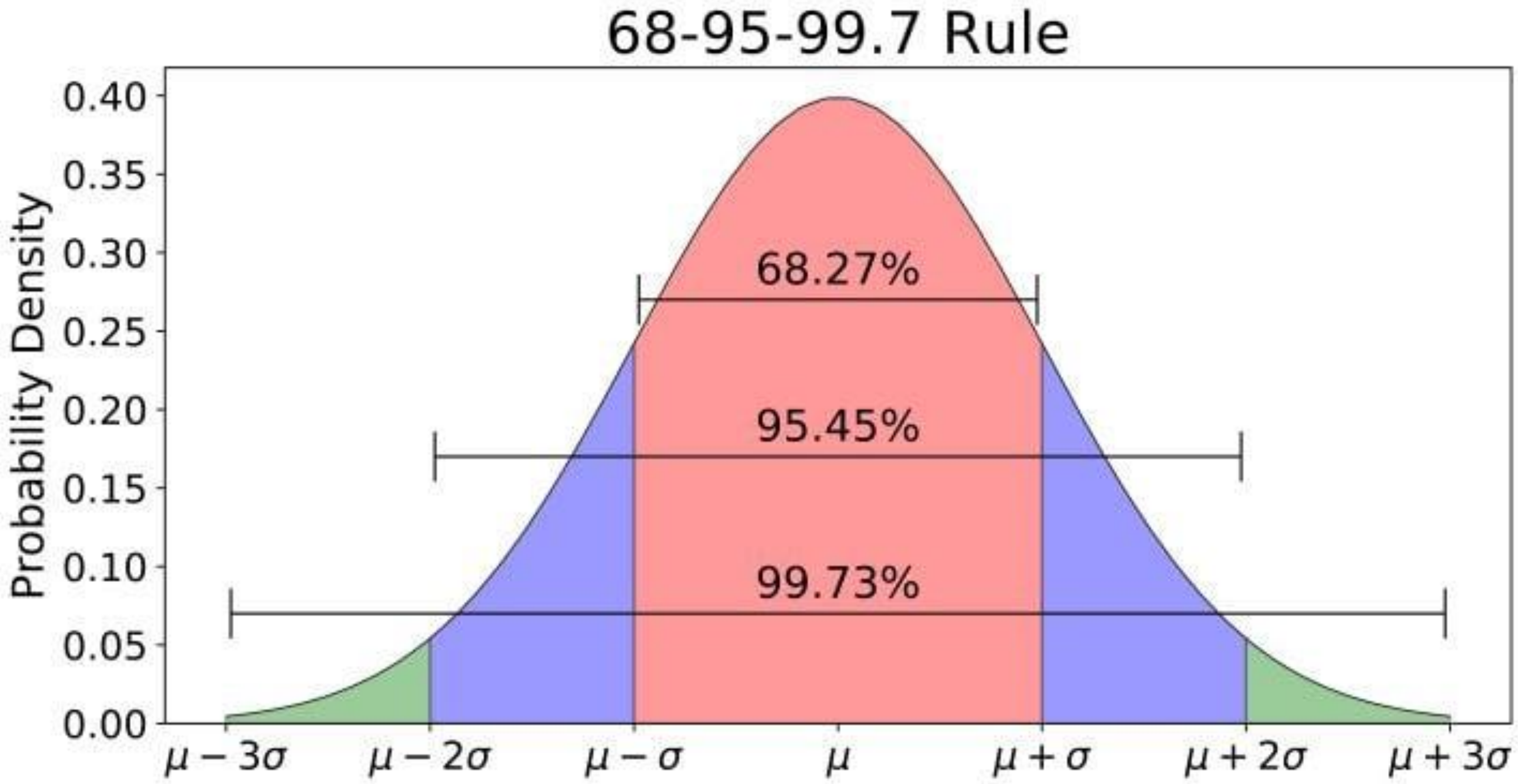
Characteristics of the Normal Distribution

- Symmetrical about its mean, μ (no skew)
- The mean, median and mode are all equal The area under the curve about the x-axis is one unit (it is a probability distribution)
- 68-95-99.7 rule:
  - 68% of the area under the curve is within +/- 1 standard deviation
  - 95% of the area under the curve is within +/- 2 standard deviation
  - 99.7% of the area under the curve is within +/- 3 standard deviation
- Completely determined by parameters μ and σ
  - Different values of μ shift the distribution left or right on the x axis
  - Different values of σ determine the spread of the distribution

Normal Distribution is Actually a Family of Countless Distributions with Differing μ's and σ's



Area Versus Standard Deviations from the Mean – The Empirical Rule / 68-95-99.7 Rule



No matter what μ and σ are, the area between μ+/- 1σ is about 68%; the area between μ+/- 2σ is about 95%; and the area between μ+/- 3σ is about 99.7%. Almost all values fall within 3 standard deviations.

The Z-score

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

- Where is the value x in relation to the rest of the population?
- The Z score is calculated in units of standard deviations, i.e.:
  - "How many standard deviations away from the mean is 'x'?"
- Allow you to compare values from different normal distributions

Z-score Example: Test Scores

Subject	Test score (x)	Class Mean score (μ)	Standard Deviation of Test Scores (σ)	Z score
Nanobiology	75	60		
Molecular Materials	75	60		

Z-score Example: Test Scores

Subject	Test score (x)	Class Mean score (μ)	Standard Deviation of Test Scores (σ)	Z score
Nanobiology	75	60	10	
Molecular Materials	75	60		

Z-score Example: Test Scores

Subject	Test score (x)	Class Mean score (μ)	Standard Deviation of Test Scores (σ)	Z score
Nanobiology	75	60	10	1.5
Molecular Materials	75	60		

Nanobio $Z = \frac{x - \mu}{\sigma}$ = (75 – 60) / 10 = 1.5

What's the probability of getting a score on the nanobio test of 75 or less, μ=60 and σ=10?

$$\therefore P(X \le 575) = \int_0^{75} \frac{1}{(10)\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{x-60}{10})^2} dx \longrightarrow \int_{-\infty}^{1.5} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}Z^2} dz$$

We don't need to do this math! Python will calculate for us.

Z-score Example: Test Scores

Subject	Test score (x)	Class Mean score (μ)	Standard Deviation of Test Scores (σ)	Z score
Nanobiology	75	60	10	1.5
Molecular Materials	75	60	5	

Z-score Example: Test Scores

Subject	Test score (x)	Class Mean score (μ)	Standard Deviation of Test Scores (σ)	Z score
Nanobiology	75	60	10	1.5
Molecular Materials	75	60	5	3

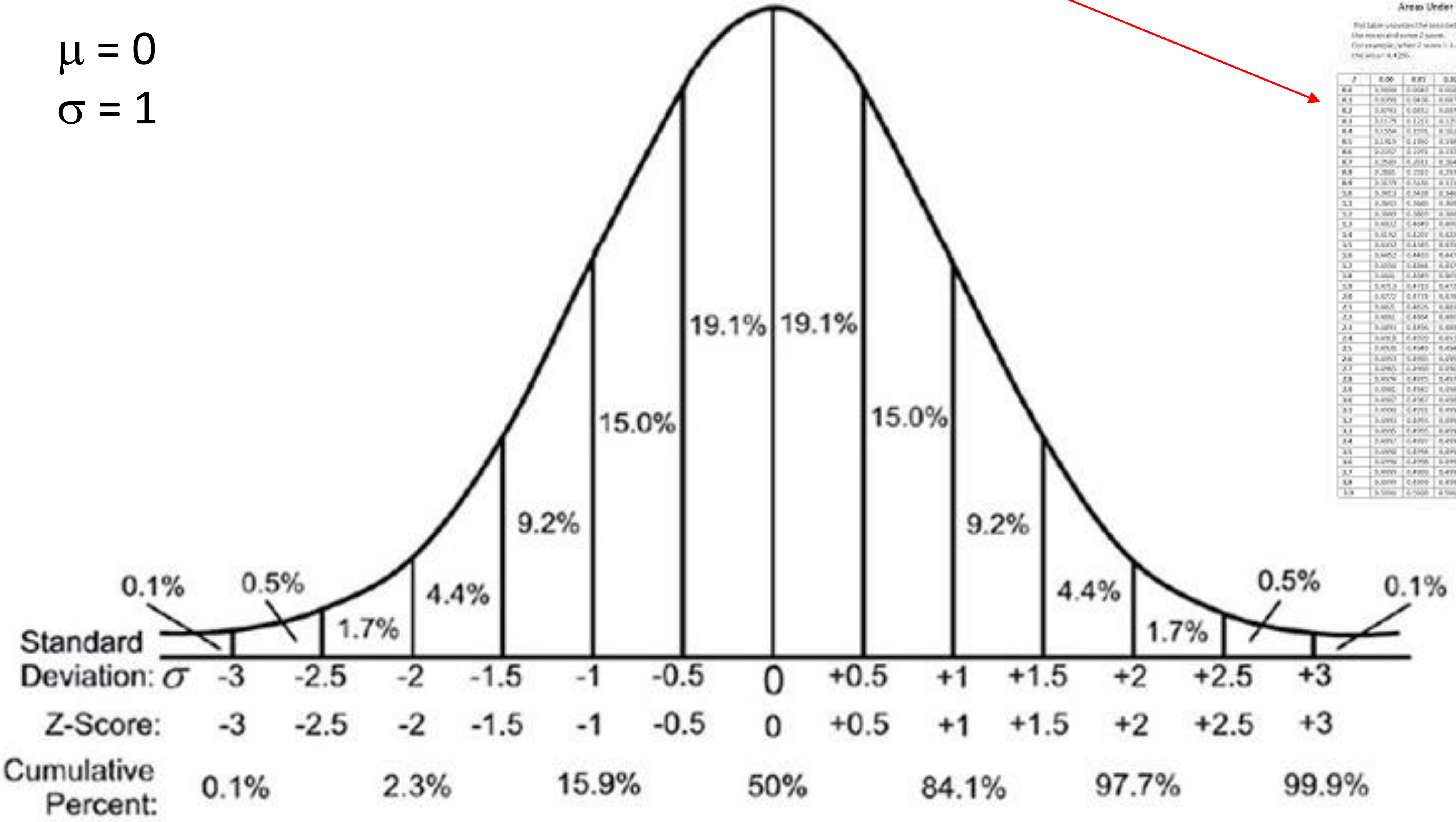
Nanobio $Z = \frac{x - \mu}{\sigma}$ = (75 – 60) / 10 = 1.5 | probability of scoring less 75 = .933

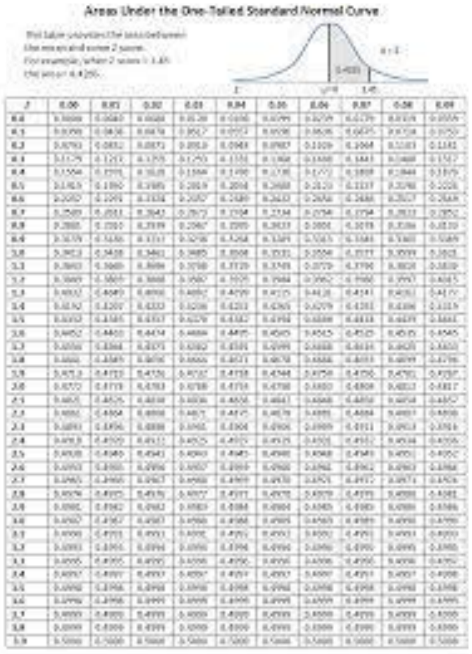
Molecular Materials $Z = \frac{x - \mu}{\sigma}$ = (75 – 60) / 5 = 3 | probability of scoring less 75 = .999

The Standard Normal Distribution

Precalculated probabilities were often looked up in tables in stats books

$\mu = 0$
$\sigma = 1$





Getting Z-score Probabilities with Python

```
from scipy.stats import norm

# Calculate the cumulative probability of a z-score less than 1.5
prob = norm.cdf(1.5)

# Print the p-value
print(prob)
```

```
0.9331927987311419
```

```
from scipy.stats import norm

# Calculate the cumulative probability of a z-score less than 3.0
prob = norm.cdf(3.0)

# Print the p-value
print(prob)
```

```
0.9986501019683699
```

Calculating Z-scores with Python

First load and confirm a dataframe (here we are loading the Pima diabetes dataset)…

```
import pandas as pd
filename = "C:\\Users\\doylef\\Desktop\\NNSE_784\\course_lectures\\data\\health\\pima-diabetes.data.csv
df = pd.read_csv(filename)
df.head()
```

	preg	gluc	bp	skin	insulin	bmi	dpf	age	outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Calculate Z-scores with scipy.stats.zscore()

```
from scipy.stats import zscore
gluc_z = zscore(df['gluc'])
type(gluc_z)
```

```
pandas.core.series.Series
```

```
z_df = pd.DataFrame()
z_df['gluc'] = df['gluc']
z_df['z_score'] = gluc_z
z_df.head()
```

	gluc	z_score
0	148	0.848324
1	85	-1.123396
2	183	1.943724
3	89	-0.998208
4	137	0.504055

Calculate the Cumulative Probability for Each Value

```
from scipy.stats import norm
#apply() allows us to use a method to calculate a value for each cell in the specified column
#we do this and assign the values to a new column "cdf" as we are calculating the cumulative
#probability for values lee than the one observed
z_df['cdf'] = z_df['z_score'].apply(norm.cdf)
z_df.head()
```

	gluc	z_score	cdf
0	148	0.848324	0.801871
1	85	-1.123396	0.130635
2	183	1.943724	0.974036
3	89	-0.998208	0.159089
4	137	0.504055	0.692889

```
# sort the dataframe by the z_score
df_sorted = z_df.sort_values( by ='z_score')
df_sorted.tail()
```

	gluc	z_score	cdf
228	197	2.381884	0.991388
408	197	2.381884	0.991388
8	197	2.381884	0.991388
561	198	2.413181	0.992093
661	199	2.444478	0.992747

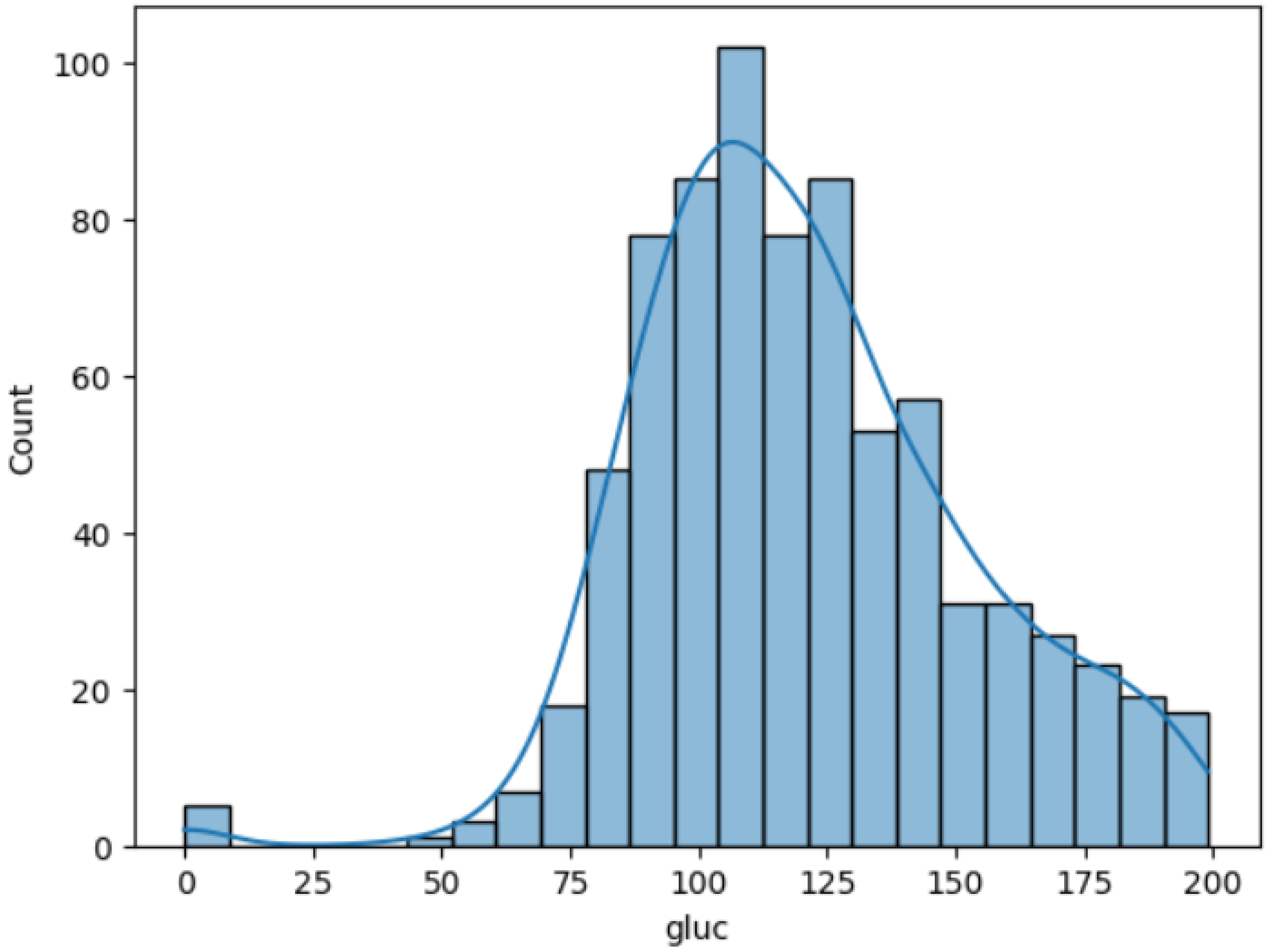
Are My Data Normally Distributed?

1. Look at the histogram! Does it appear bell shaped?
2. Compute descriptive summary measures—are mean, median, and mode similar?
3. Do 2/3 of observations lie within 1 std dev of the mean? Do 95% of observations lie within 2 std dev of the mean?
4. **Look at a normal probability plot—is it approximately linear?**
5. Run tests of normality (such as D'Agostino and Pearson's in scipy.stats.normaltest).

1.) Look at the histogram! Does it appear bell shaped?

```
import seaborn as sns
sns.histplot(df['gluc'], kde=True)
```

```
<AxesSubplot:xlabel='gluc', ylabel='Count'>
```



2) Compute descriptive summary measures—are mean, median, and mode similar?

```
df['gluc'].describe()
```

```
count    768.000000
mean     120.894531
std       31.972618
min        0.000000
25%       99.000000
50%      117.000000
75%      140.250000
max      199.000000
Name: gluc, dtype: float64
```

```
df['gluc'].mode()
```

```
0     99
1    100
dtype: int64
```

?

2) Compute descriptive summary measures—are mean, median, and mode similar?

```
df['gluc'].describe()
```

```
count    768.000000
mean     120.894531
std       31.972618
min        0.000000
25%       99.000000
50%      117.000000
75%      140.250000
max      199.000000
Name: gluc, dtype: float64
```

```
df['gluc'].mode()
```

```
0     99
1    100
dtype: int64
```

Mean and median are similar but mode is reasonably different (almost 2/3 of a standard deviation off).

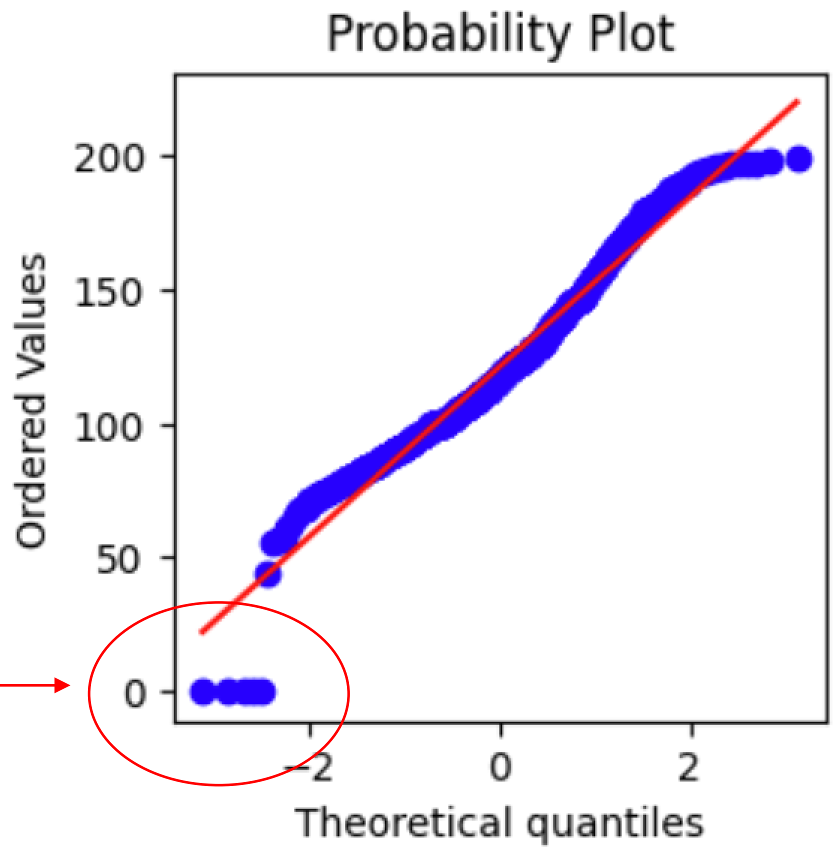
3) Check Percent of Values +/- 1 Standard Deviation

```
#first, for readability, set the two values in a list
stdev = df['gluc'].std()
range = [(df['gluc'].mean() - stdev),(df['gluc'].mean() + stdev)]
count = 0
for value in df['gluc']:
    if value > range[0] and value < range[1]:
        count = count + 1
print("Count within 1 standard deviation is {}.".format(count))
percentage = (count / df['gluc'].size) * 100
print("This is {}% of all values.".format(percentage))
```

```
Count within 1 standard deviation is 540.
This is 70.3125% of all values.
```

4) Look at a normal probability plot—is it approximately linear?

```
##qq plot to test normality
import matplotlib.pyplot as plt
import scipy
fig, ax = plt.subplots(figsize=(3, 3))
scipy.stats.probplot(df['gluc'], plot=ax)
```



Most obvious
break in linearity

5) Run a test of normality

```
scipy.stats.normaltest(df['gluc'])
```

```
NormaltestResult(statistic=12.385056622689767, pvalue=0.0020446506991363502)
```

In the case of this normaltest() method, the null hypothesis is that the sample values are from a normal distribution. Given the extremely low value of the p-val, we reject the null hypothesis (H0).