# Major Paradoxes in Probability and Statistics

Kushagra Agrawal (A53264533), Simran Barnwal (A53299395)

# Contents



# List of Figures

## List of Tables



## Listings

# 1 Importance of the topic

Paradoxes in probability and statistics have arisen due to two major reasons. The first being pit-holes in human judgement and reasoning. Interestingly, popular paradoxes in probability and statistics were published in psychology journals and showed our widespread lack of understanding in the subject (Tversky and Kahneman, 1971 [TK71]). However, these are loosely termed as paradoxes because their results seem counter-intuitive. We have studied some of these like the Birthday paradox [Mis39] and the Monty Hall Problem [Sel75] in class. These are important to consolidate our understanding in the subject and highlight the pit-falls in our judgement. We will cover a few of these in this study.

The second category defines paradoxes that give different results based on different trains of established reasoning. These paradoxes like the Bertrand's paradox [Mar94] give conflicting answers that follow from different principles in statistics. The paradoxes in this category have led to deeper understanding of the subject and the development of fields like decision theory and the dominance principle and thus, the importance of studying their origins.

# 2 Paradoxes covered in this study

As part of the final project for ECE-225A, we aim to study and analyze four research papers that explain some of these popular paradoxes. We will be studying and explaining the possible applications and significance of the paradoxes in question. During the course of the project, we will be delving into the following topics:

1. **Bertrand's Paradox [Mar94]:** The formulation of the paradox is as follows: given a circle and an equilateral triangle inscribed in it, what is the probability that a random chord will be larger than the side of the triangle. There are at least three answers to the paradox, each depending on how we define a random chord. We will evaluate each and highlight how we need to be careful when evaluating domains with infinite possibilities.

2. **Necktie Paradox [Bro95]:** A popular paradox in decision theory, this problem delves into two people fighting over whose necktie is cheaper. The person who loses the bet needs to give their necktie to the other. Given a 50% chance of having the costlier necktie, each person argues that if they lose, they lose the value of their necktie but if they win they win a necktie of a "higher value". Thus, overall they are in a profit. The paradox here is that both persons can make the same argument. We will examine how to resolve the paradox, and prove the resolution via code.

3. **Allais' Paradox [All53]:** The Allais Paradox refers to a hypothetical choice in behavioral economics that contradicts expected utility theory. Upon conducting a series of experiments involving two choices with different payoffs, it is observed that participants choose situations with a higher payoff but lower probability, which directly contradicts expected utility theory. We will be examining the paradox in depth and highlight the intuition behind it.

4. **Simpson's Paradox [Sim51]:** This is a particularly interesting statistical phenomenon where a subset of data show positive, negative or no correlation between two variables but the trend reverses or disappears on examining the entire data as a whole. We will show some interesting visual examples of the same in the final study as well as the intuition behind this.

# 3 Bertrand's Paradox

## 3.1 Paradox Description

Let's consider an equilateral triangle inscribed within a circle. The question is to find the probability that a random chord of the circle is longer than the side of this equilateral triangle.

We know that this probability should be the same no matter what established method we choose to evaluate this probability.

However each of the following methods gives a different result as shown by the code as well. The helper methods of the code can be found in the Appendix.

### 3.1.1 Method 1: Two Random Endpoints

We choose two random points on the circle as the chord's endpoints.

1. Choose two angles uniformly, randomly and independently between 0 and $2\pi$
2. Find the coordinates corresponding to these two angles on a circle of unit radius
3. Find the length of the chord
4. Repeat this experiment several times and find the frequency that the chords were greater than the side of the inscribed equilateral triangle

```
def bertrand1():
    """1. Choose two points randomly on a circle
    Return the coordinates of the two points
    And the midpoint of the chord"""

    two_angles = np.random.random((test_cases,2)) * 2*np.pi
    chord_endpoints = np.array((r*np.cos(two_angles), r*np.sin(two_angles)))
    chord_endpoints = np.swapaxes(chord_endpoints, 0, 1)

    # The midpoints of the chords
    midpoints = np.mean(chord_endpoints, axis=2).T
    return chord_endpoints, midpoints
```

Listing 1: Bertrand's Paradox Method 1: Two Random Endpoints

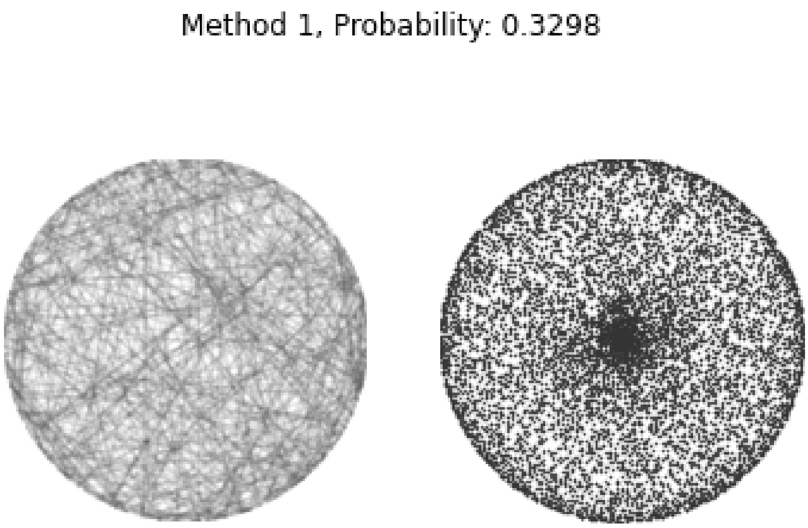


Figure 1: Bertrand's Paradox: Plots of chords, midpoints and the said probability for Method 1

### 3.1.2 Method 2: A random radial point as chord's midpoint

We choose a random circle inside the unit circle. A random point on this circle is the midpoint of the chord.

1. Choose a random radius uniformly, randomly and independently between 0 and 1. This is the inner circle.

2. Choose a point on the circumference of this circle (uniformly, randomly and independently). Let this be the midpoint of the chord.

3. Find the length of the chord

4. Repeat this experiment several times and find the frequency that the chords were greater than the side of the inscribed equilateral triangle

```
def bertrand2():
    """Choose a random radius<=r
    Then select a point randomly on this circle.
    Let this selected point be the midpoint of the chord"""

    radius_chosen = np.random.random(test_cases) * r
    angles = np.random.random(test_cases) * 2 * np.pi

    midpoints = np.array((radius_chosen * np.cos(angles),\
                          radius_chosen * np.sin(angles)))
    chords = get_chords_from_midpoints(midpoints)
    return chords, midpoints
```

Listing 2: Bertrand's Paradox Method 2: A random radial point as chord's midpoint

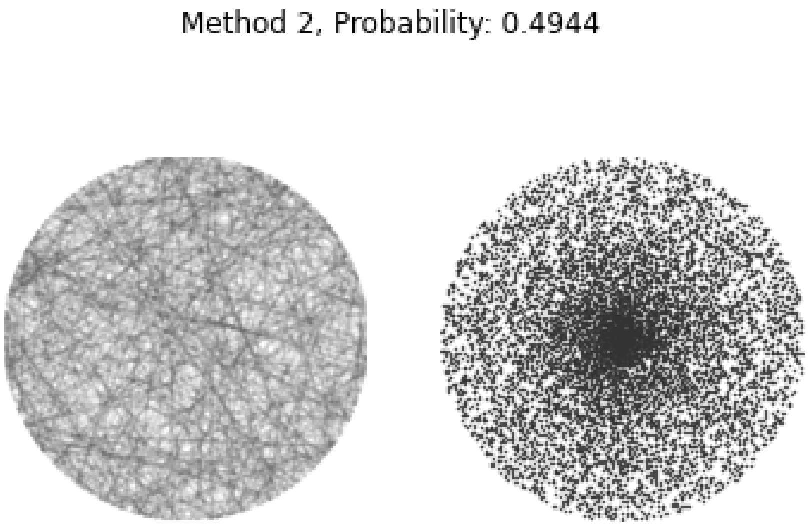


Figure 2: Bertrand's Paradox: Plots of chords, midpoints and the said probability for Method 2

### 3.1.3 Method 3: A random point inside the circle as chord's midpoint

We choose a random point anywhere inside the circle as the chord's midpoint. To choose a point uniformly, we first choose a radius and weigh the radius accordingly. More points lie on larger radii than the smaller radii.

1. Choose a random radius randomly and independently between 0 and 1. The larger radii should have higher probability of getting chosen. This is the inner circle.

2. Choose a point on the circumference of this circle (uniformly, randomly and independently). Let this be the midpoint of the chord.

3. Find the length of the chord

4. Repeat this experiment several times and find the frequency that the chords were greater than the side of the inscribed equilateral triangle

```python

def bertrand3():
    """
    In this method, we select a point randomly distributed in the circle
    and choose that as the midpoint of the chord

    To choose points that are uniformly distributed within the circle,
    we need to weigh the radius. A larger radius has more probability
    as there should be higher number of points farther out the centre.
    """

    radius_chosen = np.sqrt(np.random.random(test_cases)) * r
    angles = np.random.random(test_cases) * 2 * np.pi

    midpoints = np.array((radius_chosen * np.cos(angles),\
                          radius_chosen * np.sin(angles)))
    chords = get_chords_from_midpoints(midpoints)
    return chords, midpoints
```

Listing 3: Bertrand's Paradox Method 3: A random point inside the circle as chord's midpoint

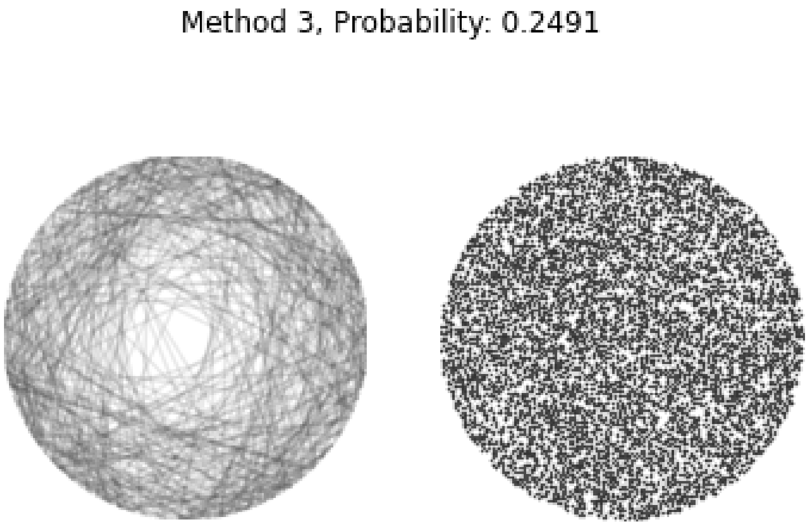


Figure 3: Bertrand's Paradox: Plots of chords, midpoints and the said probability for Method 3

### 3.2 Conclusions on Bertrand's Paradox

We see through Figures 1, 2 and 3 that each of these methods with different (correct) ways of choosing a chord, gave different probabilities. The paradox still stands unresolved [Sha07]. Thus, the "principle of indifference" [Sha07] may yield non-defined results when the possibilities are infinite.

## 4 Necktie Paradox

### 4.1 Paradox Description

In this paradox, there are two people fighting over whose necktie is cheaper. The person who loses the bet (i.e. the person with the costlier necktie) needs to give their necktie to the other person.

Each person can argue that 1. they have a 50 percent chance of winning 2. if they lose they lose the value of their necktie but if they win they win a necktie of a "higher value". Thus, overall they are in a profit. **The paradox here is that both persons can make the same argument.** However, both factions having an advantage in the bet isn't possible.

Table 1: Necktie Paradox: Possibilities Table

| Value of first person's necktie | Value of second person's necktie | First Person's Outcome | Second Person's Outcome |
|---|---|---|---|
| $x | $x | $0 | $0 |
| $x | $y | +$y | -$y |
| $y | $x | -$y | +$y |
| $y | $y | $0 | $0 |

### 4.2 Resolution of the paradox

To resolve the paradox, let's consider only two neckties that are of two kinds $x,$y. Let y be greater than x. Now, let's create a table of possibilities and what is lost or gained in each case

Let B1 be the random variable which specifies the outcome of the bet for Person 1. Given the outcome table above, we can calculate the expectation of B1.

$$E[B1] = 0.25 * \$0 + 0.25 * \$y - 0.25 * \$y + 0.25 * \$0 = \$0$$

Thus, overall, the bet doesn't favour any person in general and the bet stands resolved.

This can be verified by the code below.

```
import numpy as np
import matplotlib.pyplot as plt
def necktie_paradox(test_cases=10000):
    """Choose the costs of neckties of the two person's randomly.
    Assign the respective outcomes of the bet to each person.
    Repeat the experiment several times to find the expected outcome for both"""
    cost1_list, cost2_list = np.random.random(test_cases), np.random.random(test_cases)
    outcome1, outcome2 = 0, 0
    for cost1, cost2 in zip(cost1_list,cost2_list):
        if cost1>cost2:
            #person 2 wins, thus person 1 gives her tie to person 2
            outcome2+=cost1
            outcome1-=cost1
        elif cost2>cost1:
            #person 1 wins, thus person 2 gives her tie to person 1
            outcome1+=cost2
            outcome2-=cost2
        else:
            # both persons have ties of same costs
            continue
    # print("Expected Outcome for Person1: ",round(outcome1/test_cases,3))
    # print("Expected Outcome for Person2: ",round(outcome2/test_cases,3))
    return outcome1/test_cases, outcome2/test_cases

def plot_necktie():
    test_cases_list = [1000,3000,5000,7000,10000,30000,50000,100000,250000,500000]
    outcome1_list, outcome2_list = [], []
    for test_cases in test_cases_list:
        outcome1, outcome2 = necktie_paradox(test_cases)
        outcome1_list.append(outcome1)
        outcome2_list.append(outcome2)

    plt.plot(test_cases_list, outcome1_list, label="Expected Outcome Person1")
    plt.plot(test_cases_list, outcome2_list, label="Expected Outcome Person2")
    plt.grid(axis = 'both')
    plt.xlabel('Number of experiments')
    plt.ylabel('Expected Outcome of a Person')
    plt.legend()
```

```
    plt.show()
```

Listing 4: Necktie Paradox Code

The Figure 4 produced by the code above shows how the expected value of outcome reaches 0 for both the person's as the number of experiment's increasing thus, showing that no person has any advantage.

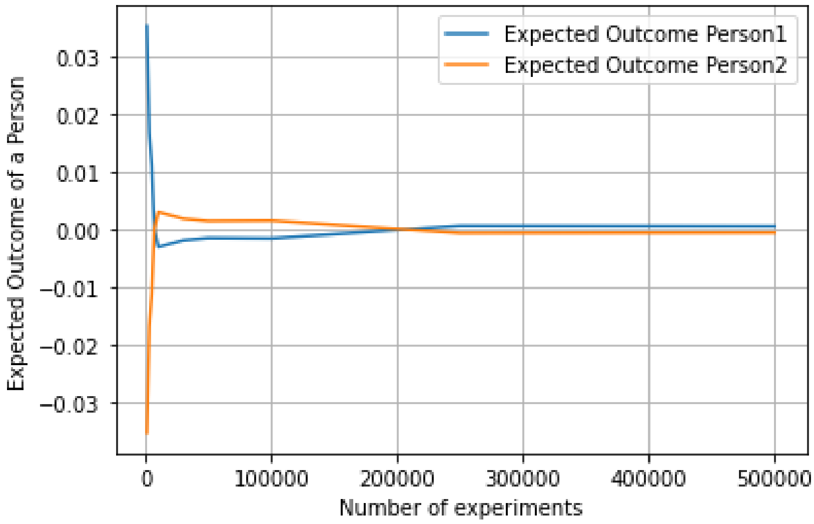


Figure 4: Necktie Paradox: Plots of expected outcomes for person1 and person2 vs number of experiments

### 4.3 Conclusions to Necktie Paradox

The necktie paradox can, thus, be resolved by carefully considering the value lost and gained in each scenario. As can be seen from Table 1 the probability of losing and winning is same (0.25 each) but the value of loss and gain are also the same, which neutralizes each other in the expectation. The plots also show that the net expected win of each person is zero.

## 5 Allais' Paradox

### 5.1 Paradox Description

The Allais' Paradox is a choice problem proposed by Maurice Allais [All53] primarily to show flaws in the Independence axiom (part of the expected utility theory) which is stated below:

**1 (Independence Axiom)** *Assume $X$, $Y$ and $Z$ are lotteries. Denote "$X$ is preferred to $Y$" as $X \succ Y$, and indifference between them by $X \sim Y$. The independence axiom states that if $X \succ Y$, then $pX + (1-p) \succ pY + (1-p)Z$ for all $Z$ and $p \in (0,1)$*

Therefore, according to the independence axiom, equal outcomes, added to each of the two choices should not affect the decision and would "cancel out". Allais argued that the axiom overlooks complementarities, the fact that our choices leading up to the decision are not independent of each other, one may affect the other. This makes the independence axiom a poor framework for rational action. We illustrate this intuition with the lottery example below.

### 5.2 Example of the paradox

Suppose we have two different experiments, each of which has two choices. We'll call the first set of experiments $A$ and $B$, and the second set $C$ and $D$. The payoffs and corresponding probabilities are given in Table 2.

Table 2: Allais Paradox: Winnings and probabilities

<table>
<tr><th colspan="2">A</th><th colspan="2">B</th><th colspan="2">C</th><th colspan="2">D</th></tr>
<tr><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td></tr>
<tr><td rowspan="3">$1 million</td><td rowspan="3">100%</td><td>$1 million</td><td>89%</td><td rowspan="2">Nothing</td><td rowspan="2">89%</td><td rowspan="2">Nothing</td><td rowspan="2">90%</td></tr>
<tr><td>Nothing</td><td>1%</td></tr>
<tr><td>$5 million</td><td>10%</td><td>$1 million</td><td>11%</td><td>$5 million</td><td>10%</td></tr>
</table>

Table 3: Allais Paradox: Re-writing payoffs

<table>
<tr><th colspan="2">A</th><th colspan="2">B</th><th colspan="2">C</th><th colspan="2">D</th></tr>
<tr><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td><td>Winnings</td><td>Probability</td></tr>
<tr><td>$1 million</td><td>89%</td><td>$1 million</td><td>89%</td><td>Nothing</td><td>89%</td><td>Nothing</td><td>89%</td></tr>
<tr><td rowspan="2">$1 million</td><td rowspan="2">11%</td><td>Nothing</td><td>1%</td><td rowspan="2">$1 million</td><td rowspan="2">11%</td><td>Nothing</td><td>1%</td></tr>
<tr><td>$5 million</td><td>10%</td><td>$5 million</td><td>10%</td></tr>
</table>

**The paradox:** Upon asking participants to choose between $A$ and $B$ and between $C$ and $D$, it was observed that most people preferred $A$ to $B$ and $D$ to $C$. However, this contradicts the decisions suggested by the independence axiom, which would prefer $C$ over $D$, provided we chose $A$ over $B$. This can be illustrated better by breaking down the probabilities as given in Table 3.

According to the independence axiom, we can disregard the 89% chance of winning a million dollars in $A$, $B$ and winning nothing in $C$, $D$, since it states that no additional information is provided by these factors. This renders experiments $B$ and $D$ to be the exact same, hence proving the source of the paradox. For the same set of experiments, we observed two different decisions which is not consistent with the expected utility theory.

**Intuitions behind participant's choices:** One possible reasoning behind the choices could be individual preference of certainty over a risky outcome. Since $A$ has a guaranteed payoff, a rational decision would be to choose that over $B$, even though $B$ has a higher expected value over $A$ (1M vs 1.3M). In the case of $C$ vs $D$, since both offer some uncertainty, one might choose $D$ over $C$, since it has a higher potential payoff, and a higher expected value (0.5M vs 0.11M).

Another possible rationale could be the magnitude of reward. While the independence axiom ignores that, it is possible that one chooses $A$ over $B$, since there is a 1% chance of winning nothing in $B$, which would be a significant loss even though $B$ offers a 10% chance of winning 5 times the reward that $A$ offers. This notion of loss is contingent on $A$ offering a guaranteed payout, which makes these choices dependent on each other, contradicting the independence axiom.

### 5.3 Conclusion of Allais Paradox

Allais paradox provides an interesting experiment that contradicts with the predictions made by the expected utility theory. It argues that a rational decision does not solely depend on expected utility, but rather should also take into account the variance of utility, i.e., a measure of risk. It further hints towards the idea that rational decisions may not be scale-free, and that this scale dependence should be factored in while making decisions.

## 6 Simpson's Paradox

### 6.1 Paradox Description

Simpson's paradox, quite simply, is observed when trends that appear in several groups of data disappear or reverse when the groups are combined. As one can probably guess, the primary cause of the paradox is an error in deciding whether data can be combined or should be looked at separately. Considering aggregated information such as percentages can often be misleading. Causal relations between data must be addressed when aggregating the data. The paradox is often observed in medical-science statistics and can be very problematic if not addressed properly.

The example below illustrates the paradox using a fictitious example. It highlights how the paradox can be resolved if we address causal relations and dependency between variables in the statistical modelling.

## 6.2 Example of Simpson's Paradox

We use a very simple example to illustrate Simpson's paradox. We generate data for number of hours of exercise per week versus the risk of developing heart disease for two sets of patients, below and above the age of 50. The below code snippet showcases our data generation process and plots the relationship between exercise and risk of heart disease.

```
'''
    Data Generation
'''
ages_list = np.random.randint(20, 50, SAMPLES)

hoursOfExercise = np.random.randint(1, 5, SAMPLES) + np.random.randn(SAMPLES)
prob = 12 + 0.5 * ages_list + -2.1 * hoursOfExercise + np.random.randn(SAMPLES) *
    2

youngFolks = pd.DataFrame({'age': ages_list, 'Hours Exercised': hoursOfExercise,
    'probability': prob })

old_ages_list = np.random.randint(50, high=85, size=SAMPLES)
old_hoursOfExercise = np.random.randint(3, high=8, size=SAMPLES) + np.random.
    randn(SAMPLES) * 0.5
old_prob = 40 + 0.32 * old_ages_list + -3.2 * old_hoursOfExercise + np.random.
    randn(SAMPLES)
oldFolks = pd.DataFrame({'age': old_ages_list, 'Hours Exercised':
    old_hoursOfExercise, 'probability': old_prob })
```

Listing 5: Simpson's Paradox: Data Generation

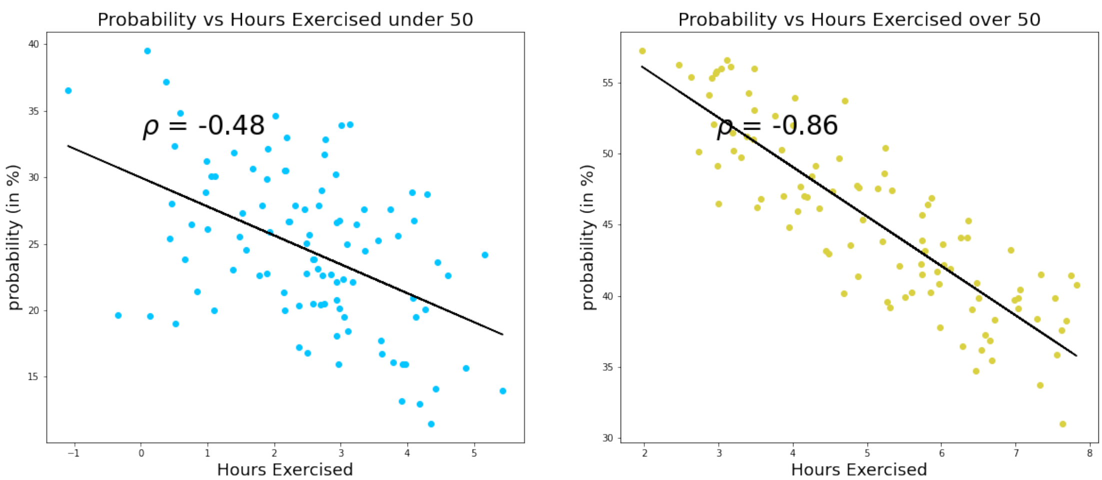


Figure 5: Simpson's Paradox: Plots between probability of heart disease vs hours of exercise for each age group

Figure 5 clearly illustrate the trend that increased hours of exercise implies a reduced risk of heart disease. However, if we combine the data and then plot the results, we observe the opposite trend.

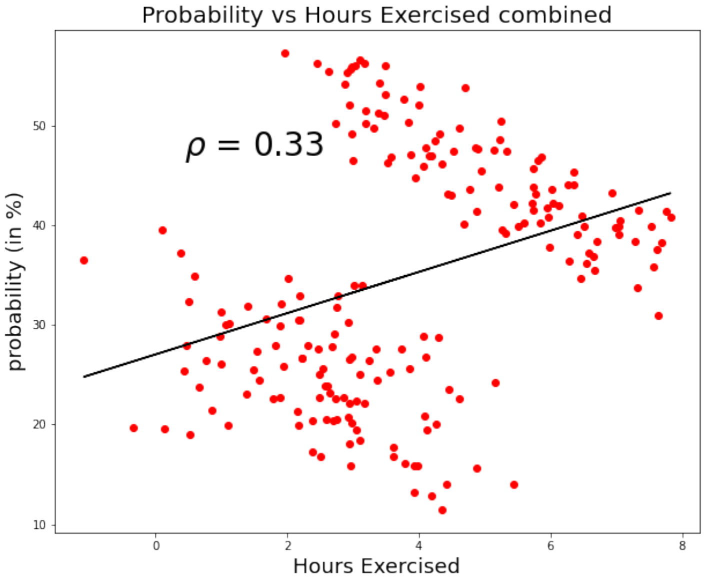


Figure 6: Simpson's Paradox: Plot between probability of heart disease vs hours of exercise for both the age groups

The above plot (Figure 6) showcases the trend having completely reversed from the first two plots. Does it imply that more hours of exercise actually increases the risk of heart disease? Not exactly. To resolve the paradox, we need to check the factors that influence the results. Here, we seem to have completely ignored the second factor affecting heart disease, age. If we plot the risk of heart disease vs age, we see the following trend:

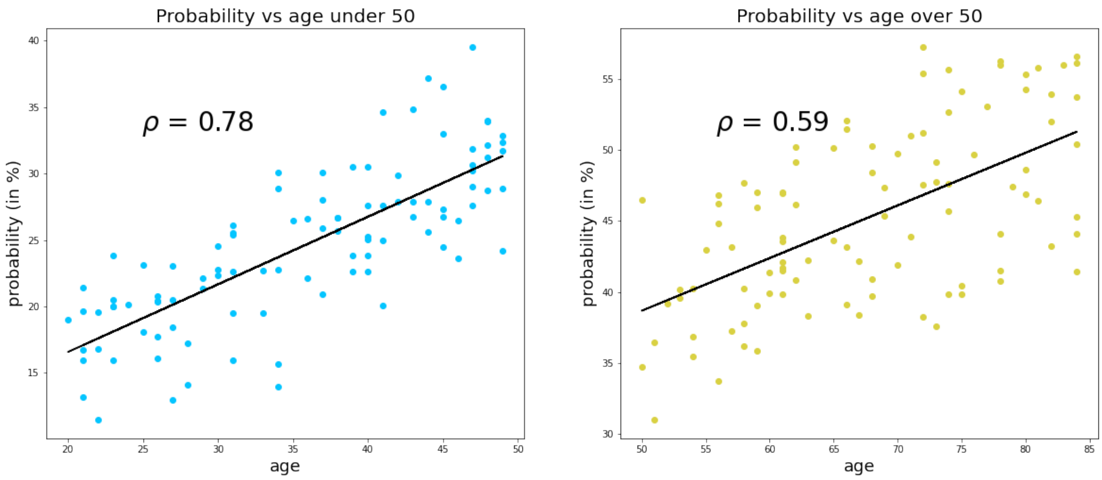


Figure 7: Simpson's Paradox: Plots between probability of heart disease vs age for both age groups

Clearly, Figure 7 shows that increasing age has a strong positive correlation with risk of heart disease. Therefore, it makes sense to keep age constant when evaluating the effect of hours spent exercising with risk of heart disease. This can be showcased below for patients aged 61.

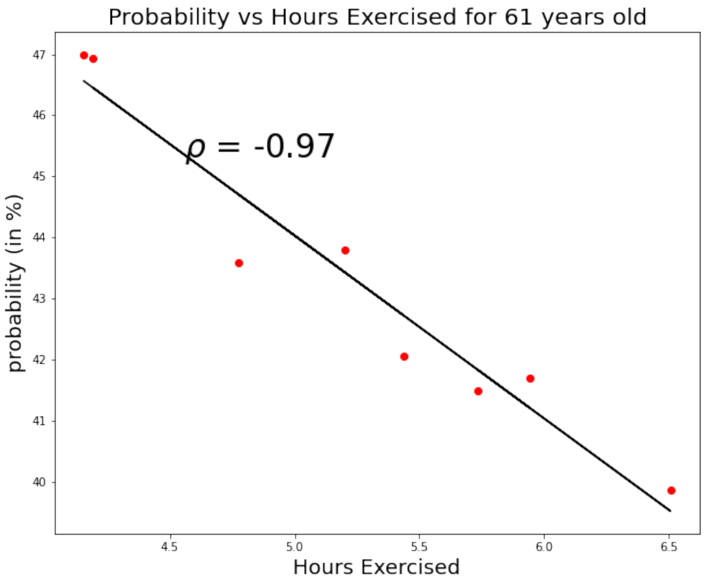


Figure 8: Simpson's Paradox: Plot between probability of heart disease vs hours of exercise for people aged 61

Now, in figure 8, we see a negative correlation between the two factors, which confirms our intuitive beliefs as well.

### 6.3 Resolution of Simpson's Paradox

From the above example, we can see that Simpson's paradox can occur when we do not account for causal relationships and dependent factors (confounding variables) in the data. The relevant query for the above example would be to assess individual risk, i.e., for a particular age, determine the hours one should spend exercising. It can be concluded that one must take into account the whole story the data is trying to tell and get a more complete picture by analyzing the relationships between factors to avoid Simpson's paradox.

## 7 Appendix

### 7.1 Helper Code for Bertrand's Paradox

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Circle
from matplotlib.lines import Line2D

#constants
GREY = (0.2,0.2,0.2)
chord_plots = 500
test_cases = 10000
r = 1
chord_length_limit = r * np.sqrt(3)

def get_chords_from_midpoints(midpoints):
    """Return the two endpoints of chords with the given midpoints
    """

    chords = np.zeros((test_cases, 2, 2))
    for i, (x_mid, y_mid) in enumerate(midpoints.T):
        m_line = -x_mid/y_mid
```

```
        c_line = y_mid + x_mid**2/y_mid
        A, B, C = m_line**2 + 1, 2*m_line*c_line, c_line**2 - r**2
        d = np.sqrt(B**2 - 4*A*C)
        x = np.array( ((-B + d), (-B - d))) / 2 / A
        y = m_line*x + c_line
        chords[i] = (x, y)
    return chords

bertrand_methods = {1: bertrand1, 2: bertrand2, 3: bertrand3}

def setup_axes():
    """Set up the two Axes with the circle and correct limits, aspect."""

    fig, axes = plt.subplots(nrows=1, ncols=2, subplot_kw={'aspect': 'equal'})
    for ax in axes:
        circle = Circle((0,0), r, facecolor='none')
        ax.add_artist(circle)
        ax.set_xlim((-r,r))
        ax.set_ylim((-r,r))
        ax.axis('off')
    return fig, axes

def plot_bertrand(method_number):
    # Plot the chords and their midpoints on for the selected method

    chords, midpoints = bertrand_methods[method_number]()

    # To keep track of chords longer than chord_length_limit
    success = [False] * test_cases

    fig, axes = setup_axes()
    for i, chord in enumerate(chords):
        x, y = chord
        if np.hypot(x[0]-x[1], y[0]-y[1]) > chord_length_limit:
            success[i] = True
        if i < chord_plots:
            line = Line2D(*chord, color=GREY, alpha=0.1)
            axes[0].add_line(line)
    axes[1].scatter(*midpoints, s=0.2, color=GREY)
    fig.suptitle('Method {}'.format(method_number))

    prob = np.sum(success)/test_cases
    print('Bertrand, method {} probability: {}'.format(method_number, prob))
    plt.savefig('bertrand{}.png'.format(method_number))
    plt.show()

plot_bertrand(1)
plot_bertrand(2)
plot_bertrand(3)
```

Listing 6: Helper Code: Bertrand's Paradox

## 7.2 Helper Code for Simpson's Paradox

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

SAMPLES = 100
np.random.seed(42)

def plot_scatter(dataframe, x_axis, y_axis, title, colour, ax):
```

```
    '''
        Helper function to generate required scatter plots and save to disk
    '''
    x, y = np.array(dataframe[x_axis]), np.array(dataframe[y_axis])
    intercept, slope = np.polynomial.polynomial.polyfit(x, y, 1)
    dataframe.plot(x = x_axis, y = y_axis, c = colour, style='o', legend=None, ax
   =ax)
    ax.plot(x, slope*x+intercept, '-', color='k')
    plt.title(title, fontsize=20)
    corr_coef = np.corrcoef(x, y)[0][1]
    ax = plt.gca()
    plt.xlabel(x_axis, fontsize=18)
    plt.ylabel(y_axis + ' (in %)', fontsize=18)
    plt.text(0.2, 0.75, r'$\rho$ = ' + f'{round(corr_coef, 2)}', fontsize = 28,
   color = 'k', transform=ax.transAxes)
```

Listing 7: Simpson's Paradox: Method for Scatter Plots

```
'''
    Plotting the combined hours exercised for both age groups
'''
plt.figure(figsize = (10, 8))
combined = pd.concat([youngFolks, oldFolks], axis = 0)
ax = plt.subplot(1, 1, 1)
plot_scatter(combined, 'Hours Exercised', 'probability', 'Probability vs Hours
    Exercised combined', 'r', ax)
```

Listing 8: Simpson's Paradox: Risk of heart disease vs Hours of exercise combined

```
'''
    Plotting age for people under 50 and over 50 years old
'''
plt.figure(figsize=(20,8))
ax = plt.subplot(1,2,1)
plot_scatter(youngFolks, 'age', 'probability', 'Probability vs age under 50', '
    #04c5ff', ax)

ax = plt.subplot(1,2,2)
plot_scatter(oldFolks, 'age', 'probability', 'Probability vs age over 50', '#
    d9d142', ax)
```

Listing 9: Simpson's Paradox: Risk of heart disease vs age

```
'''
    Resolving the paradox by using a stratified sample. Hours exercised for a
    particular age
'''
plt.figure(figsize = (10, 8))
combined = combined[combined['age'] == 61]
ax = plt.subplot(1, 1, 1)
plot_scatter(combined, 'Hours Exercised', 'probability', 'Probability vs Hours
    Exercised combined for 61 years old', 'r', ax)
```

Listing 10: Simpson's Paradox: Risk of heart disease vs Exercise for 61 year olds

# References


[Mis39] Richard Von Mises. "Über Aufteilungs- und Besetzungswahrscheinlichkeiten". In: *Revue de la faculté des sciences de l'Université d'Istanbul* 4 (1939), pp. 145–163.

[Sim51] E. H. Simpson. "The Interpretation of Interaction in Contingency Tables". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 13.2 (1951), pp. 238–241. ISSN: 00359246. URL: http://www.jstor.org/stable/2984065.

[All53] M. Allais. “Le Comportement de l’Homme Rationnel devant le Risque: Critique des Postulats et Axiomes de l’Ecole Americaine”. In: *Econometrica* 21.4 (1953), pp. 503–546. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1907921.

[TK71] Amos Tversky and Daniel Kahneman. “Belief in the law of small numbers”. In: *Psychological Bulletin* (1971), pp. 105–110.

[Sel75] Steve Selvin. “A problem in probability (letter to the editor)”. In: *The American Statistician* 29.1 (1975), pp. 67–71. URL: https://www.jstor.org/stable/2683689.

[Mar94] Louis Marinoff. “A Resolution of Bertrand’s Paradox”. In: *Philosophy of Science* 61.1 (1994), pp. 1–24. ISSN: 00318248, 1539767X. URL: http://www.jstor.org/stable/188286.

[Bro95] Aaron C. Brown. “Neckties, wallets, and money for nothing”. In: *Journal of Recreational Mathematics* 27.2 (1995), pp. 116–22.

[Sha07] Nicholas Shackel. “Bertrand’s Paradox and the Principle of Indifference*”. In: *Philosophy of Science* 74.2 (2007), pp. 150–175. ISSN: 00318248, 1539767X. URL: http://www.jstor.org/stable/10.1086/519028.