Marginal Workers
Employment

INTRODUCTION

- **In industrialized countries the principal way in which adults and their families make ends meet economically is through employment in the formal labor force.**
- **In the United States, earnings in the form of wages and salaries account for 71 percent of household income, on average.**
- **Even among individuals in poor families, half (49 percent) of all household income comes from earnings—their single most important source of income (Current Population Survey 1998).**

MARGINAL EMPLOYMENT AS UNDEREMPLOYMENT

- **Marginal employment per se is not as yet aconventional concept in sociology and thus there are no standard empirical definitions of this term.**
- **As noted, we conceptualize it as consisting of both the degree of attachment to the labor force and the quality of jobs.**
- **Drawing on the Labor Utilization Framework of Phillip Hauser (1974), Clifford Clogg (1979) developed a measure of underemployment that consists of the following categories.**

WORKES

- **Workers are those who produce or transform goods or provide services, for their own consumption and for that of others.**
- **Money payment for work does not always accompany its performance, although this is general in advanced economies.**
- **Two problems are highlighted in analyzing workers as a social**
- **class: (1) the sources of the present position of workers in advanced economies and.**
- **Class:(2) The consequences for workers will be analyzed in terms of their new structural position in society as consumers and their resulting social outlook, including their class consciousness.**

JOB DISPLACEMENT

- **Even a stable work history is no guarantee against future employment hardship.**
- **An important determinant of labor-force instability is job displacement.**

- **To get a sense of the magnitude of the problem, consider that between 1993 and 1995, 15 percent of U.S. workers experienced job displacement (Kletzer 1998).**
- **The underlying causes of displacement are not well documented, but are likely rooted in technological change, rising foreign competition, and declining productivity within industries (Kletzer 1998)**
- **Once displaced, there also is inequality in the prospects of finding new work after displacement. Here it helps to be white, male, and better educated.**

CONTINGENT AND TEMPORARY WORKERS

- **Also contributing to volatility of labor-force attachment is the emergence of contingent labor and temporary workers. In the 1970s, international competition had begun to challenge U.S.**
- **firms began to restructure in order to maintain their competitive edge. By the mid-1980s, terms such as "mergers," "acquisitions," "deindustrialization," and "downsizing" had become commonplace (Harrison and Bluestone 1988).**
- **Instead an increasing proportion of the American workforce is now engaged in contingent work**

Understanding Marginal Probability with Python

- **An essential concept of mathematics, marginal probability, will be studied in this article.**
- **Implementing it using Python and its various tools is something that we will learn**
- **Talking about probability in science, business and medicine are places in which it is fundamental It supports our understanding of uncertainty in a variety of domains.**

Marginal Probability

Introduction

- **An understanding of complicated systems and making reasonable choices is the fundamental idea of marginal probability, which is the basic idea of probability theory.**
- **Insights into the probabilities of specific events are offered, along with the results of other factors.**

Visualizing Marginal Probabilities

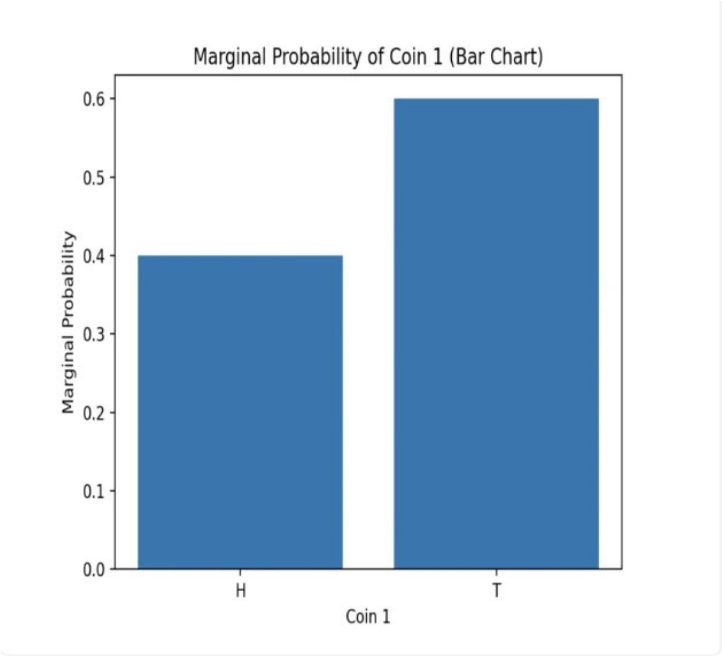
- **As discussed previously, we will make our results visual so that we can understand them better.**
- **We will be trying various plots, as shown below. Since we are not performing any calculations but rather.**
- **using Matplotlib to visualize the findings from the previous parts, we will use the same values as in the previous section.**

Bar

```
import matplotlib.pyplot as plt
plt.bar(marginal_prob_coin1['Coin1'], marginal_prob_coin1['Joint_Probability'])
```

```
plt.xlabel('Coin 1')
plt.ylabel('Marginal Probability')
plt.title('Marginal Probability of Coin 1 (Bar Chart)')
plt.show()


plt.bar(marginal_prob_coin2['Coin2'], marginal_prob_coin2['Joint_Probability'])
plt.xlabel('Coin 2')
plt.ylabel('Marginal Probability')
plt.title('Marginal Probability of Coin 2 (Bar Chart)')
plt.show()
```

- **To build the bar chart, we use the bar function on line 2.**
- **The x-values (Coin 1 outcomes) for the bars are specified by the first argument, marginal_prob_coin1[‘Coin1’]. The second option,**
- **Then the same code is used to plot another bar for the second coin.**
- **You can see the bar plot that we just created in the image below.**



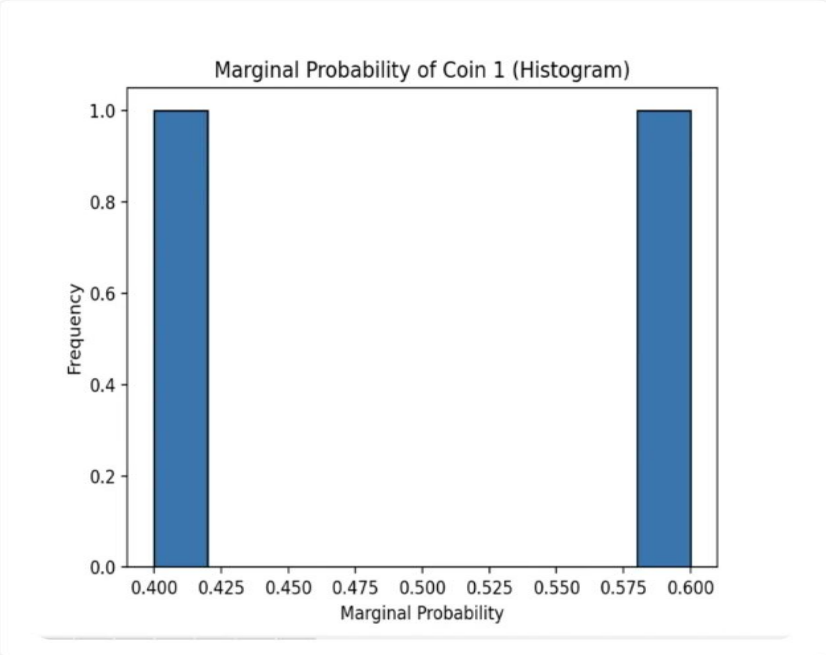
Marginal Probability Bar Plot

Histogram

```
plt.hist(marginal_prob_coin1['Joint_Probability'], bins=10, edgecolor='black')
plt.xlabel('Marginal Probability')
plt.ylabel('Frequency')
plt.title('Marginal Probability of Coin 1 (Histogram)')
plt.show()

plt.hist(marginal_prob_coin2['Joint_Probability'], bins=10, edgecolor='black')
```

```
plt.xlabel('Marginal Probability')
plt.ylabel('Frequency')
plt.title('Marginal Probability of Coin 2 (Histogram)')
plt.show()
```

- **The marginal probabilities of Coin 1 are represented by a histogram in the code for the histogram visualization.**
- **The hist function is used after the necessary libraries have been imported.**
- **Using the marginal probabilities as input data, it defines the number of bins to segment the data**
- **The color of bin edges is controlled by the edge color parameter.**
- **The histogram is displayed when plt.show() is used.**
- **Providing a visually interesting look at the probability distribution and behavior, this histogram displays the frequency distribution of Coin 1's marginal probability.**

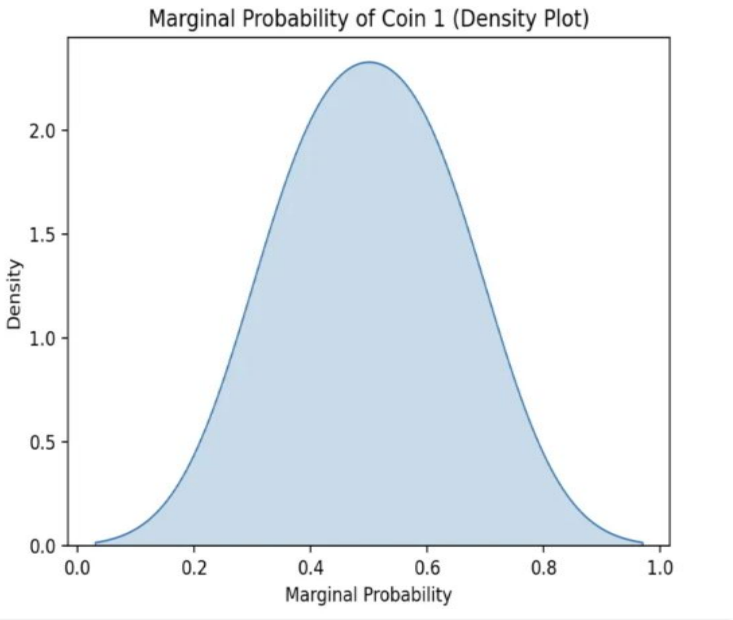


Density Plot

```
import seaborn as sns
sns.kdeplot(marginal_prob_coin1['Joint_Probability'], shade=True)
plt.xlabel('Marginal Probability')
plt.ylabel('Density')
plt.title('Marginal Probability of Coin 1 (Density Plot)')
plt.show()

sns.kdeplot(marginal_prob_coin2['Joint_Probability'], shade=True)
plt.xlabel('Marginal Probability')
plt.ylabel('Density')
```

```
plt.title('Marginal Probability of Coin 2 (Density Plot)')
plt.show()
```

- **The density plot is used in the code for the density plot visualization to show the marginal probabilities of Coin 1.**
- **The kdeplot function from the Seaborn library is used after the required libraries have been imported.**
- **It shades the region under the density curve using the marginal probabilities as input data.**
- **The axes are labeled and given titles via the xlabel, ylabel, and title functions.**
- **By visualizing the probability density distribution of Coin 1's marginal probabilities in a plot,**
- **we are able to understand its pattern of distribution and learn more about how it behaves within the joint distribution.**



Real-World Applications

- **Numerous uses of marginal probabilities can be found in daily life.**
- **The planning of interventions and disease prediction in epidemiology are supported by them.**
- **To assess risk and make investment decisions, finance experts use them.**
- **It is advantageous for genetics to predict genetic features and illnesses.**
- **To interpret attitudes and trends in survey analysis, they are used.**
- **To provide a correct forecast of weather, marginal probabilities are used.**
- **In machine learning to model uncertainty, they are important.**
- **In understanding complex systems, risk assessment, facilitating better decision-making.**
- **And all these sectors, marginal probabilities give an important idea to separate the effects of particular variables**

Scatter Plot with Marginal Histograms in Python with Seaborn

- **Scatter Plot with Marginal Histograms is basically a joint distribution plot with the marginal distributions of the two variables.**
- **In data visualization, we often plot the joint behavior of two random variables (bi-variate distribution) or any number of random variables.**
- **But if data is too large, overlapping can be an issue. Hence, to distinguish between variables it is useful to have the probability distribution of each variable on the side along with the joint plot.**
- **This individual probability distribution of a random variable is referred to as its marginal probability distribution.**
- **In seaborn, this is facilitated with jointplot().**
- **It represents the bi-variate distribution using scatterplot() and the marginal distributions using histplot().**

Approach

- **Import seaborn library**
- **Load dataset of your choice**
- **Use jointplot() on variables of your dataset**

Example 1:

```
# importing and creating alias for seaborn

import seaborn as sns


# loading tips dataset

tips = sns.load_dataset("tips")


# plotting scatterplot with histograms for features total bill and tip.

sns.jointplot(data=tips, x="
```

Example 2:

```
import seaborn as sns


tips = sns.load_dataset("tips")


# here "*" is used as a marker for scatterplot
```

```
sns.jointplot(data=tips, x="
```

Example3:

```
import seaborn as sns


tips = sns.load_dataset("tips")


sns.jointplot(data=tips,
```

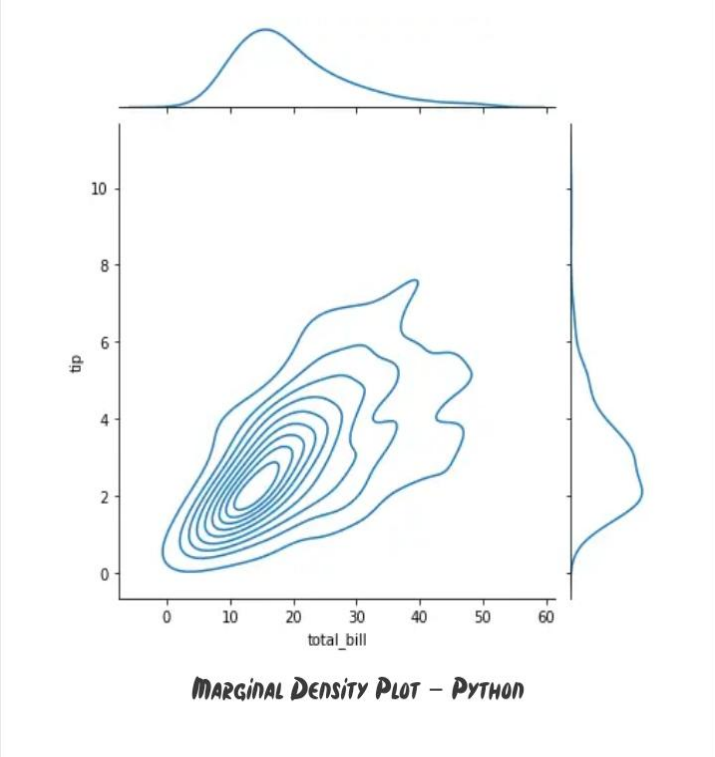
Example 4:

Code for getting marginal effects from logistic regression using python.

```
ex logit marginal effects.py
# *---------------------------------------------------------------
# | PROGRAM NAME: ex logit marginal effects.py
# | DATE: 3/4/21
# | CREATED BY: MATT BOGARD
# | PROJECT FILE:
# *---------------------------------------------------------------
# | PURPOSE: example code for getting marginal effects from logistic regression
# *---------------------------------------------------------------

import pandas as pd
import numpy as np
from matplotlib import style
from matplotlib import pyplot as plt
import statsmodels.formula.api as smf
import statsmodels.api as sm

# simulate some data
df = pd.DataFrame(columns=['x','D', 'p','y'])
np.random.seed(123)
df['ID'] = np.random.uniform(1, 10000, 10000)
df['D'] = np.random.binomial(1, .5, 10000)
np.random.seed(123)
df['x'] = np.random.normal(30,1,10000)
xb = -9 + 3.5*df['D'] + .2*df['x'] #-9 + 3.5*gender + 0.2*age
df['p'] = 1/(1 + np.exp(-xb))
np.random.seed(123)
```

```
df['y'] = np.random.binomial(1, df['p'], 10000)
df.head()
df.describe()


style.use("fivethirtyeight")

#---------------------------------
# for a baseline, run linear probabiity model (LPM)
#---------------------------------

smf.ols('y ~ D + x', data=df ).fit().summary()

#---------------------------------------
# logistic regression
#---------------------------------------

model = smf.logit(formula='y ~ D + x', data=df).fit()
model.summary()

print(model.get_margeff(at ='overall').summary()) # get marginal effects
print(model.get_margeff(at ='mean').summary()) # get marginal effects

# mean marginal effects from logistic regressin are close to LPM model

# syntax
#get_margeff(at='overall', method='dydx', atexog=None, dummy=False, count=False)

# 'at' Options are:

# ‘overall’, The average of the marginal effects at each observation.

# ‘mean’, The marginal effects at the mean of each regressor.

# ‘median’, The marginal effects at the median of each regressor.

# ‘zero’, The marginal effects at zero for each regressor.

# ‘all’, The marginal effects at each observation. If at is all only margeff will be available
from the returned object.

#----------------------------------------
# logistic regression  with glm
#----------------------------------------
```

Marginal Density Plot Python Visualization Output



- **Let's see an example to customize color for marginal density plot in python using seaborn library.**
- **We will need seaborn package to show marginal density plot. Install package using below command.**

Conclusion

- **I hope you found above article on marginal density plot in python.**
- **program using seaborn package informative and educational.**
- **Use sns.jointplot() function of seaborn module to create marginal density plot in python program.**
- **color, ratio, space to customize color, ratio and no space or huge space in marginal density plot.**