Replicating Del Negro Et Al

Ed Herbst

March 13, 2025

Outline

The first thing we have to do is download the SPF probability forecasts.

```
import pandas as p
spf_url = '/home/eherbst/Downloads/SPFmicrodata(2).xlsx' #'h
spf = p.read_excel(spf_url, sheet_name='PRPGDP', na_values=
spf['DATE'] = spf.apply(lambda x: p.Period(f'{int(x.YEAR)}-
spf = spf.drop(['INDUSTRY','YEAR','QUARTER'], axis=1).set_in
spf.head()
```

```
class IVDRandomVariable:
    """
    Class for an interval valued discrete random variable.
    """
    def __init__(self, intervals, probabilities):
        """
        Initializes the interval valued discrete random vari

        Args:
            intervals (list): A list of tuples representing
            probabilities (list): A list of probabilities as
```

```
        """
        self.intervals = intervals
        self.probabilities = probabilities

    def pmf(self):
        """
        Computes the probability mass function (PMF) of the

        Returns:
            A list of tuples representing the points and pro
        """
        pmf_points = []
        for i in range(len(self.intervals)):
            a, b = self.intervals[i]

            # probability of the interval
            p = self.probabilities[i]

            # points in the interval
```

```
        points = range(a, b+1)

        # add each point to the PMF with corresponding p
        for x in points:
            pmf_points.append((x, p))

    # sort the PMF by x value
    pmf_points.sort()

    # combine probabilities for each x value
    pmf = []
    last_x, last_p = pmf_points[0]
    for x, p in pmf_points[1:]:
        if x == last_x:
            last_p += p
        else:
            pmf.append((last_x, last_p))
            last_x, last_p = x, p
    pmf.append((last_x, last_p))
```

```
        return pmf

    def plot_pmf(self):
        """
        Plots the probability mass function (PMF) of the int
        """
        pmf = self.pmf()
        x = [p[0] for p in pmf]
        y = [p[1] for p in pmf]
        plt.bar(x, y, width=0.8)
        plt.xlabel('x')
        plt.ylabel('P(X=x)')
        plt.title('Probability Mass Function')
        plt.show()

import matplotlib.pyplot as plt
IVDRandomVariable([(1, 2), (3, 4)], [0.5, 0.5]).plot_pmf()
```