Reppeto530Week4

January 5, 2024

1 Chapter 1

Brian Reppeto 530 Prof. Jim Week 4 HW

1.0.1 Exercise 3-1

[79]:

```
# Import and download the code from the github repo

from os.path import basename, exists


def download(url):
    filename = basename(url)
    if not exists(filename):
        from urllib.request import urlretrieve

        local, _ = urlretrieve(url, filename)
        print("Downloaded " + local)

download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/
  ↪2002FemResp.dct")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/
  ↪2002FemResp.dat.gz")
download("https://github.com/AllenDowney/ThinkStats2/raw/master/code/first.py")
```

[80]:

```
# import the .py files for calc

import thinkplot
import thinkstats2
import nsfg
import numpy as np
```

[81]:

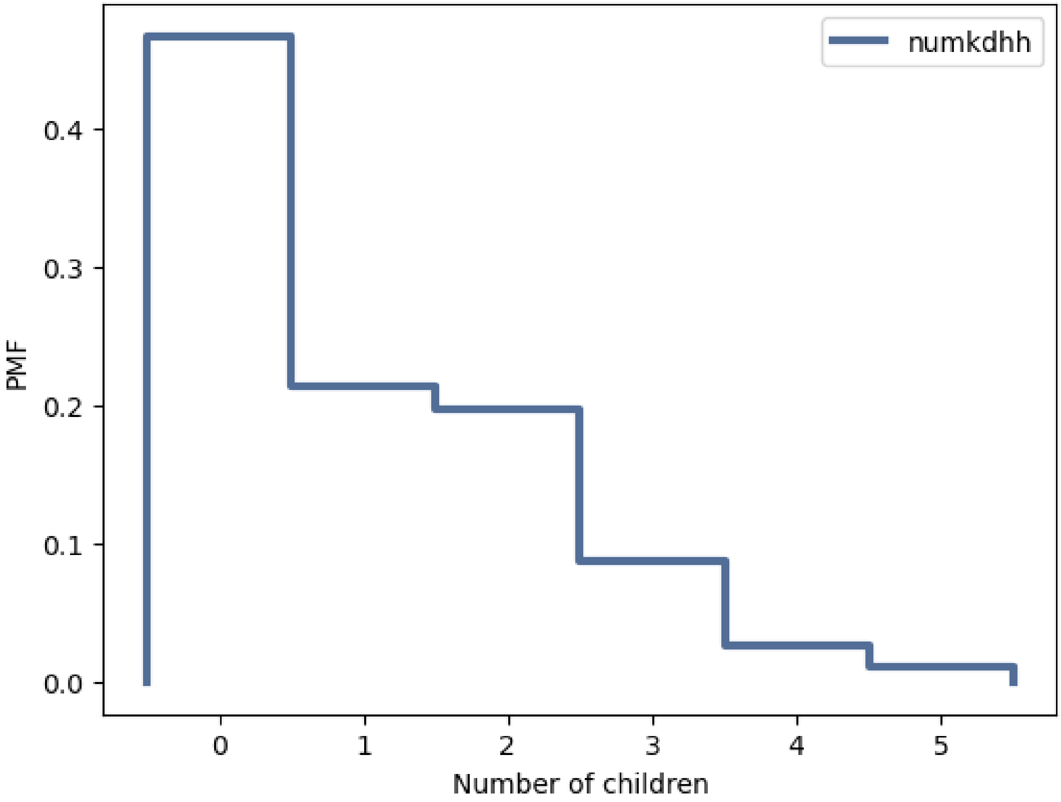
```
# Read the FemResp file

resp = nsfg.ReadFemResp()
```

[82]:

```
# using the thinkstats2.py file to calc the pmf using the FemResp file and␣
 ↪variable 'numkdhh'

pmf = thinkstats2.Pmf(resp.numkdhh, label='numkdhh')
```

[83]:

```
# using the thinkplot.py file the pmf by passing in the pmf_calc
"""Plots a Pmf or Hist as a line.

    Args:
      pmf: Hist or Pmf object
      options: keyword args passed to plt.plot
    """
"""Configures the plot.

    Pulls options out of the option dictionary and passes them to
    the corresponding plt functions.
    """

thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='Number of children', ylabel='PMF')
```



[84]:

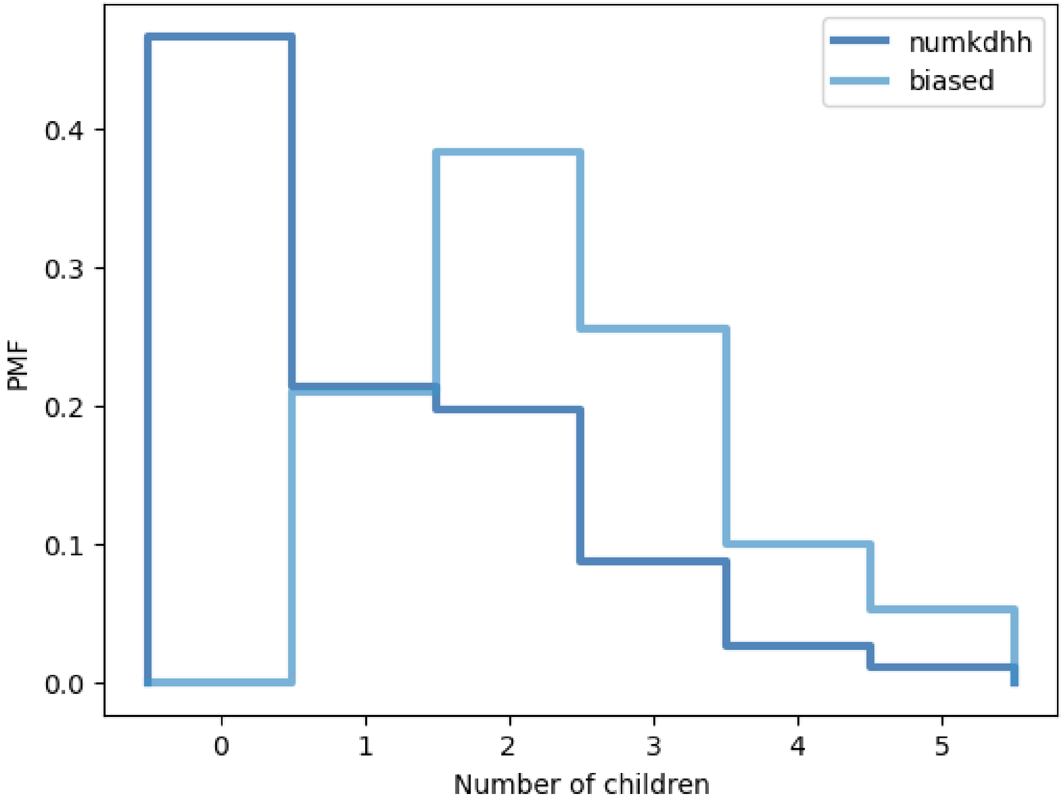
```
# Function for the biaspmf function to compute the bias PMF for surveying the students about classes

def BiasPmf(pmf, label):
    new_pmf = pmf.Copy(label=label) # copy of the PMF

    for x, p in pmf.Items():  # loop over items
        new_pmf.Mult(x, x)

    new_pmf.Normalize()  #normalize the new PMF
    return new_pmf
```

[85]:

```
# define the bias and pass the pmf as defined above

bias = BiasPmf(pmf, label="biased")
```

[86]:

```
# using the thinkplot.py file and the defined bias to plot the actual and biased distribution

thinkplot.PrePlot(2)
thinkplot.Pmfs([pmf, bias])
thinkplot.Config(xlabel="Number of children", ylabel="PMF")
```



[87]:

```python
# Calc the pmf mean

pmf.Mean()
```

[87]:

```
1.024205155043831
```

[88]:

```python
# Calc the bias mean

bias.Mean()
```

[88]:

```
2.403679100664282
```

Exercise 3-2

[89]:

```python
# function for the PmfMean: calculates the mean of a probability mass function (PMF).

def PmfMean(pmf):
    return sum(p * x for x, p in pmf.Items())
```

[90]:

```python
# Function for the PmfVar computes the variance of a PMF

def PmfVar(pmf, mu=None):
    if mu is None:
        mu = PmfMean(pmf)
    return sum(p * (x - mu) ** 2 for x, p in pmf.Items())
```

[92]:

```python
# PmfMean calc

PmfMean(pmf)
```

[92]:

```
1.024205155043831
```

[93]:

```python
# PmfVaR calc

PmfVar(pmf)
```

[93]:

```
1.4128643263531195
```

Exercise 4-1

I weighed 7.7 & was a first baby

[94]:

```python
#  Import the first.py file and the MakeFrames function

import first

live, firsts = first.MakeFrames()[:2]# Only unpack the first two values
```

[95]:

```
# Compute the distribution of birth weights for first and other babies

first_wgt = firsts.totalwgt_lb
first_wgt_dropna = first_wgt.dropna()
print('Firsts', len(first_wgt), len(first_wgt_dropna))
first_pmf = thinkstats2.Pmf(first_wgt_dropna, label='first')
```

```
Firsts 4413 4363
```

[96]:

```
# create the function for % rank

def PercentileRank(scores, your_score):
    count = 0
    for score in scores:
        if score <= your_score:
            count += 1

    percentile_rank = 100.0 * count / len(scores)
    return percentile_rank
```

[97]:

```
# Using the thinkstats2 create a Cumulative Distribution function

fst_cdf = thinkstats2.Cdf(firsts.totalwgt_lb)
```

[98]:

```
#  calc the first percentile
#  Only in the 66th percentile,  I will still call my mom.

fst_cdf.PercentileRank(7.7)
```

[98]:

```
66.10130644052258
```

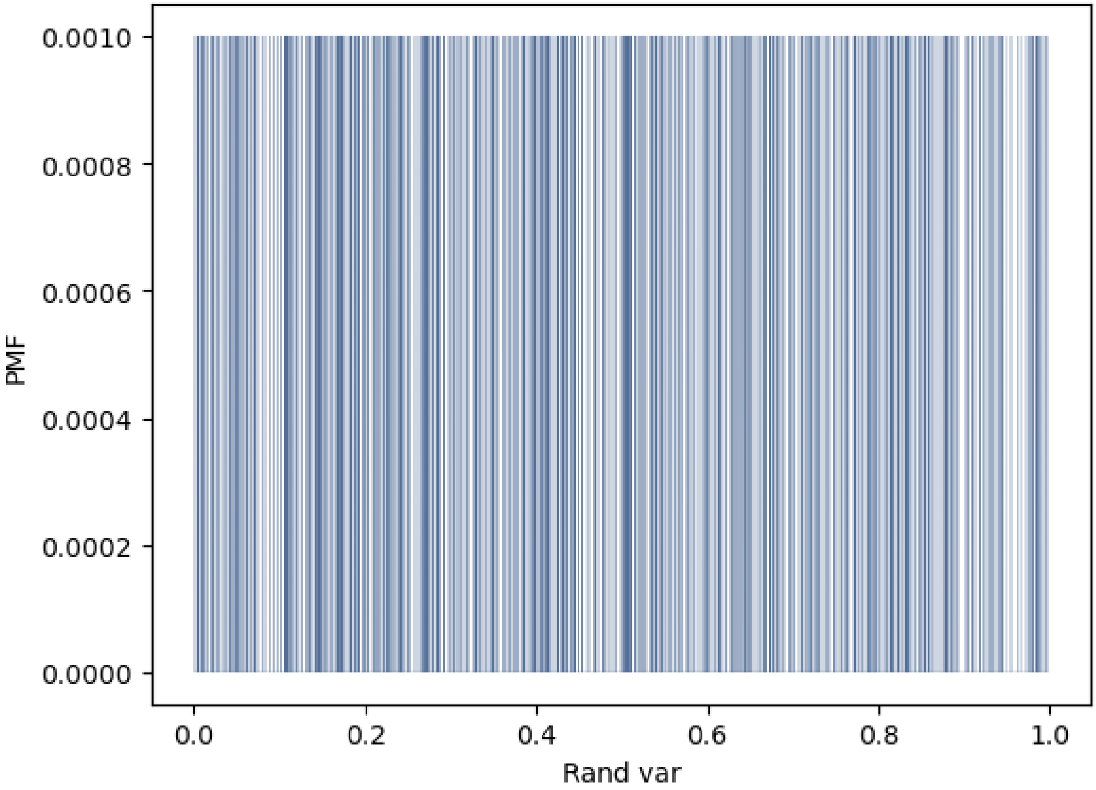
Exercise 4-2

[99]:

```
# generate an array with 100 random numbers using the numpy lib and the random␣
 ↪module

r = np.random.random(1000)
```

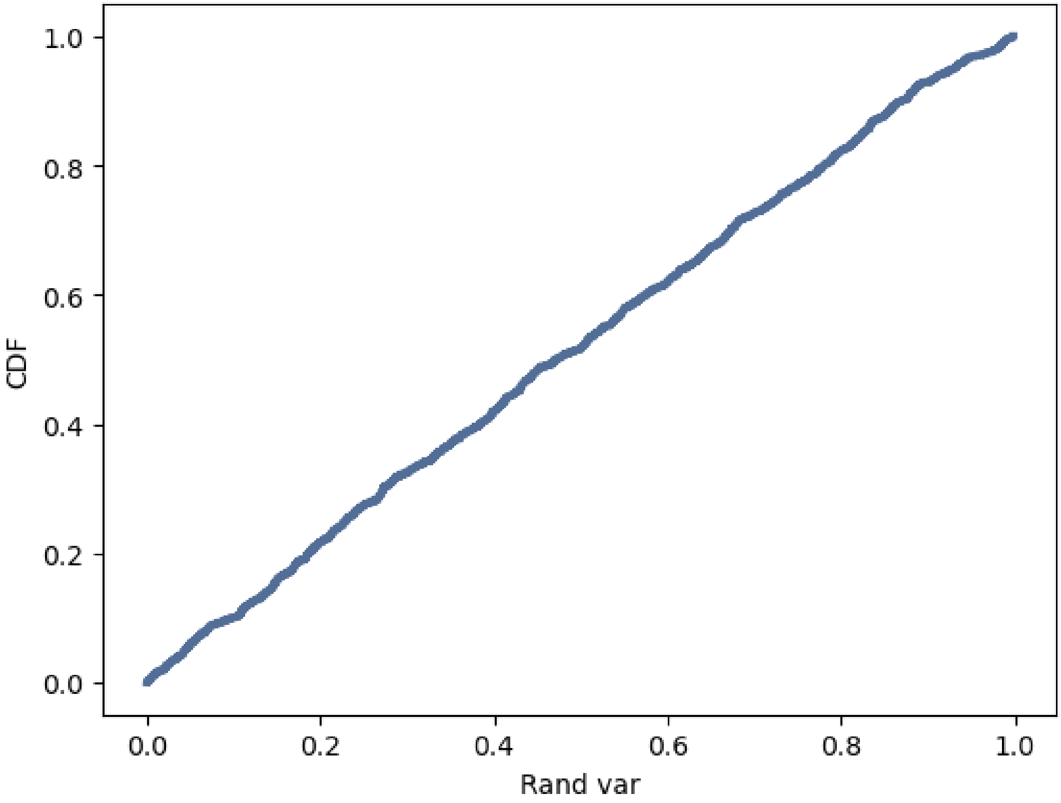
[100]:

```
# create a PMF for the array of random numbers r and plot the PMF

pmf = thinkstats2.Pmf(r)
thinkplot.Pmf(pmf, linewidth=0.1)
thinkplot.Config(xlabel='Rand var', ylabel='PMF')
```



[101]:

```
# create a CDF for the array of random numbers r and plot the CDF

cdf = thinkstats2.Cdf(r)
thinkplot.Cdf(cdf)
thinkplot.Config(xlabel='Rand var', ylabel='CDF')
```


1.0
0.8
0.6
0.4
0.2
0.0
CDF
0.0
0.2
0.4
0.6
0.8
1.0
Rand var

[ ]: