Theodore Koby-Hercsky
DSC530-T302

# 4.2 Exercises

## Exercises

**Exercise 3-1:** Something like the class size paradox appears if you survey children and ask how many children are in their family. Families with many children are more likely to appear in your sample, and families with no children have no chance to be in the sample.

Use the NSFG respondent variable `numkdhh` to construct the actual distribution for the number of children under 18 in the respondents' households.

Now compute the biased distribution we would see if we surveyed the children and asked them how many children under 18 (including themselves) are in their household.

Plot the actual and biased distributions, and compute their means.

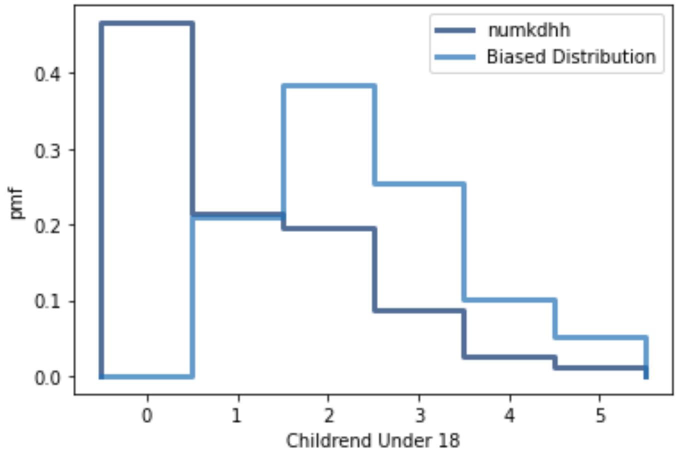
In [37]:
```
resp = nsfg.ReadFemResp()
```

In [38]:
```
# Set the PMF equal to Pmf to resp.numkdhh as the exercise requested
# Then also label it numkdhh.

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

In [50]:
```
# Next I will use thinkplot again to first plot the Pmfs
# Which will include both the pmf and Biased_distribution and
# Then I will configure them to see children under 18 and pmf

thinkplot.Pmfs([pmf, Biased_distribution])
thinkplot.Config(xlabel='Childrend Under 18', ylabel='pmf')
```




In [60]:
```
# Next I wil comput the mean of our pmf
pmf.Mean()
```

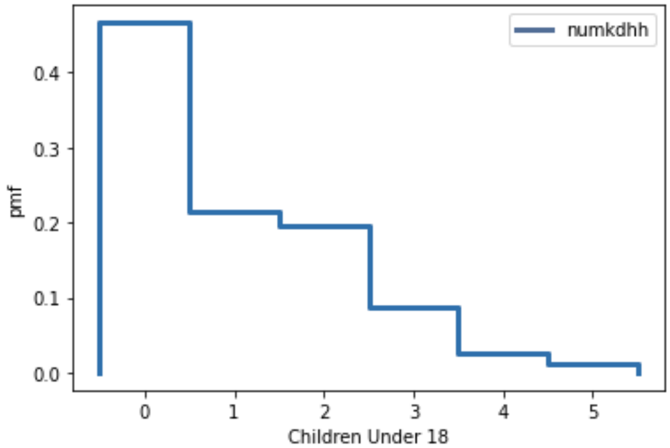
Out[60]: 1.024205155043831

In [59]:
```
# Next I wil comput the mean of our Biased distribution
Biased_distribution.Mean()
```

Out[59]: 2.403679100664282

In [46]:
```
# Next I will use thinkplot to configure the plot for the pmf
# That will show the number of children under 18 in the respondents
# household.

thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='Children Under 18', ylabel='pmf')
thinkplot.Pmf(pmf)
```

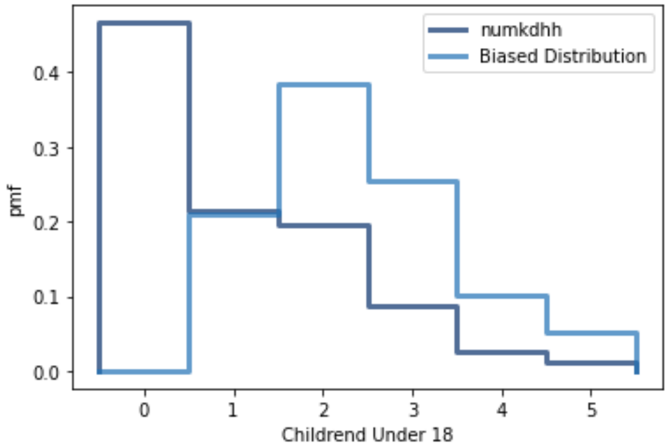



In [47]:
```
# Next I will be computing the biased distribution by setting the
# Biased children equal to the BiasPmf and labeling it biased distributi

Biased_distribution = BiasPmf(pmf, label='Biased Distribution')
```

In [50]:
```
# Next I will use thinkplot again to first plot the Pmfs
# Which will include both the pmf and Biased_distribution and
# Then I will configure them to see children under 18 and pmf

thinkplot.Pmfs([pmf, Biased_distribution])
thinkplot.Config(xlabel='Childrend Under 18', ylabel='pmf')
```




In [60]:
```
# Next I wil comput the mean of our pmf
pmf.Mean()
```

Out[60]: 1.024205155043831

In [59]:
```
# Next I wil comput the mean of our Biased distribution
Biased_distribution.Mean()
```
Out[59]: 2.403679100664282

```
As seen in the plot we see that the biased distribution is a little
less than the numkdhh as the main spike exceeds 0.4 while Biased
Distribution is just under 0.4.
```

**Exercise 3-2:** Write functions called PmfMean and PmfVar that take a Pmf objecy and compute the mean and variance. To Test these methods, check that they are consistent with the methods Mean and Var provided by Pmf

In [62]:
```
# Next I wrote a function that calculates the mean for pmf and
# Set it equal to PmfMean
PmfMean = pmf.Mean()
PmfMean
```
Out[62]: 1.024205155043831

In [64]:
```
# Next I wrote a function that calculates the variance for pmf and
# Set it equal to PmfVar

PmfVar = pmf.Var()
PmfVar
```
Out[64]: 1.4128643263531195

```
As seen above we see that the Mean is 1.024 and the variance for our
pmf is 1.41 as seen above.
```

# Exercises

**Exercise 4-1:** How much did you weigh at birth? If you don't know, call your mother or someone else who knows. Using the NSFG data (all live births), compute the distribution of birth weights and use it to find your percentile rank. If you were a first baby, find your percentile rank in the distribution for first babies. Otherwise use the distribution for others. If you are in the 90th percentile or higher, call your mother back and apologize.

In [29]:
```
# I was the first baby my mom had and I weighed 6.3 pounds
# Being that I was the first baby I will find what my percentile rank
# Is compared to the first_cdf
first_cdf.PercentileRank(6.3)
```
Out[29]: 20.05500802200321

```
As seen above I was only in the 20th percentile meaning I was only
bigger and or the same weight as 20% of the other first babies.
```

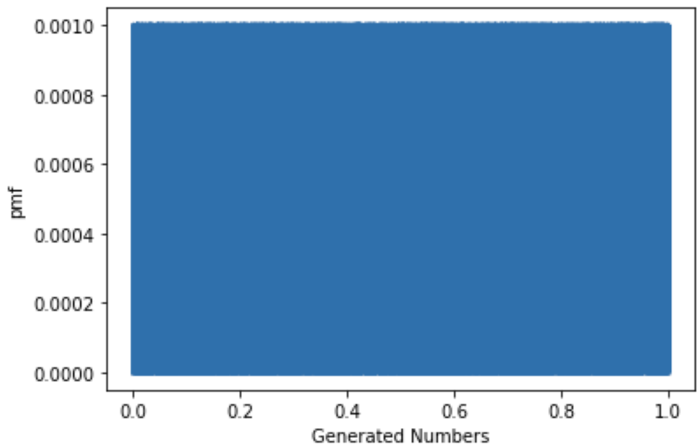
**Exercise 4-2:** The numbers generated by `numpy.random.random` are supposed to be uniform between 0 and 1; that is, every value in the range should have the same probability.

Generate 1000 numbers from `numpy.random.random` and plot their PMF. What goes wrong?

Now plot the CDF. Is the distribution uniform?

In [30]:
```
# First I will set one_thousand equal to np random 1000 numbers
one_thousand = np.random.random(1000)
```

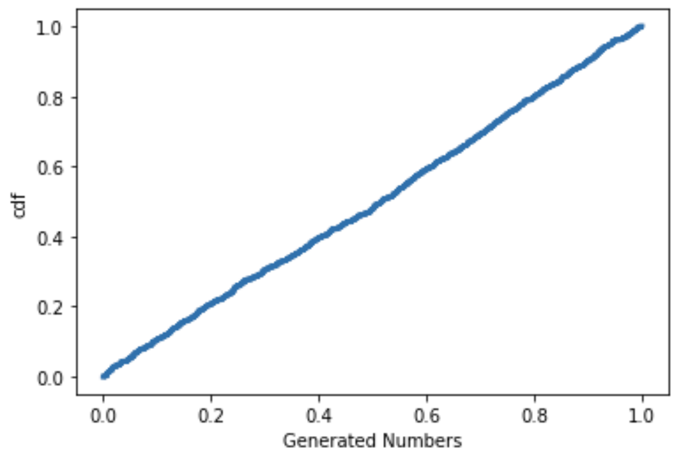
In [34]:
```
# Next I will plot their PMF by setting pmf equal to one_thousand
pmf = thinkstats2.Pmf(one_thousand)
thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='Generated Numbers', ylabel='pmf')
thinkplot.Pmf(pmf)
```



As seen above we seen that the plot just shows a blue box for all the numbers generated.

In [33]:
```
# Next I will plot the CDF and see if that changes the results
# First I will set the cdf equal to the one_thousand random numbers.
cdf=thinkstats2.Cdf(one_thousand)
thinkplot.Cdf(cdf)
thinkplot.Config(xlabel='Generated Numbers', ylabel='cdf')
thinkplot.Cdf(cdf)
```

Out[33]:
```
{'xscale': 'linear', 'yscale': 'linear'}
```




As seen above we see that by using cdf instead of pmf the plot we generate creates a upward movement in cdf as the amount of generated numbers goes up.