# Final - MyrnaAcuna

March 3, 2019

## 1 Examples and Exercises from Think Stats, 2nd Edition

http://thinkstats2.com



```
In [1]: from __future__ import print_function, division

        %matplotlib inline

        import numpy as np

        import nsfg
        import first
        import thinkstats2
        import thinkplot
        import os
        import pandas
        import statistics
        df = pandas.read_csv('data.csv')

In [2]: from statistics import mode

In [3]: import seaborn as sns

In [4]: #data cleanup
        df.drop(columns=['from_station_id','from_station_name','usertype','to_station_id','to_s

Out[4]:          trip_id  year  month  week  day  hour  gender            starttime  \
        0        2355134  2014      6    27    0    23    Male  2014-06-30 23:57:00
        1        2355133  2014      6    27    0    23    Male  2014-06-30 23:56:00
        2        2355130  2014      6    27    0    23    Male  2014-06-30 23:33:00
        3        2355129  2014      6    27    0    23  Female  2014-06-30 23:26:00
        4        2355128  2014      6    27    0    23  Female  2014-06-30 23:16:00
        5        2355127  2014      6    27    0    23    Male  2014-06-30 23:11:00
        6        2355126  2014      6    27    0    23    Male  2014-06-30 23:08:00
        7        2355125  2014      6    27    0    23    Male  2014-06-30 23:07:00
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | 2355123 | 2014 | 6 | 27 | 0 | 23 | Male | 2014-06-30 23:06:00 |
| 9 | 2355116 | 2014 | 6 | 27 | 0 | 22 | Male | 2014-06-30 22:46:00 |
| 10 | 2355111 | 2014 | 6 | 27 | 0 | 22 | Male | 2014-06-30 22:22:00 |
| 11 | 2355110 | 2014 | 6 | 27 | 0 | 22 | Female | 2014-06-30 22:17:00 |
| 12 | 2355106 | 2014 | 6 | 27 | 0 | 22 | Male | 2014-06-30 22:14:00 |
| 13 | 2355100 | 2014 | 6 | 27 | 0 | 22 | Female | 2014-06-30 22:04:00 |
| 14 | 2355098 | 2014 | 6 | 27 | 0 | 22 | Female | 2014-06-30 22:04:00 |
| 15 | 2355097 | 2014 | 6 | 27 | 0 | 22 | Male | 2014-06-30 22:03:00 |
| 16 | 2355096 | 2014 | 6 | 27 | 0 | 22 | Male | 2014-06-30 22:02:00 |
| 17 | 2355095 | 2014 | 6 | 27 | 0 | 22 | Female | 2014-06-30 22:01:00 |
| 18 | 2355092 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:55:00 |
| 19 | 2355091 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:55:00 |
| 20 | 2355081 | 2014 | 6 | 27 | 0 | 21 | Female | 2014-06-30 21:46:00 |
| 21 | 2355079 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:45:00 |
| 22 | 2355076 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:45:00 |
| 23 | 2355075 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:44:00 |
| 24 | 2355073 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:44:00 |
| 25 | 2355072 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:44:00 |
| 26 | 2355069 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:44:00 |
| 27 | 2355067 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:44:00 |
| 28 | 2355065 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:43:00 |
| 29 | 2355064 | 2014 | 6 | 27 | 0 | 21 | Male | 2014-06-30 21:43:00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 9495205 | 16734099 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:10:00 |
| 9495206 | 16734098 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:09:00 |
| 9495207 | 16734097 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:09:00 |
| 9495208 | 16734096 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:09:00 |
| 9495209 | 16734095 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:09:00 |
| 9495210 | 16734094 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:08:00 |
| 9495211 | 16734093 | 2017 | 10 | 39 | 6 | 0 | Female | 2017-10-01 00:08:00 |
| 9495212 | 16734092 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:08:00 |
| 9495213 | 16734091 | 2017 | 10 | 39 | 6 | 0 | Female | 2017-10-01 00:08:00 |
| 9495214 | 16734090 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:07:00 |
| 9495215 | 16734089 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:07:00 |
| 9495216 | 16734088 | 2017 | 10 | 39 | 6 | 0 | Female | 2017-10-01 00:07:00 |
| 9495217 | 16734087 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:07:00 |
| 9495218 | 16734086 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:07:00 |
| 9495219 | 16734084 | 2017 | 10 | 39 | 6 | 0 | Female | 2017-10-01 00:06:00 |
| 9495220 | 16734083 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:06:00 |
| 9495221 | 16734082 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:06:00 |
| 9495222 | 16734081 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:05:00 |
| 9495223 | 16734080 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:04:00 |
| 9495224 | 16734078 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:04:00 |
| 9495225 | 16734077 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:04:00 |
| 9495226 | 16734076 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:04:00 |
| 9495227 | 16734075 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:04:00 |
| 9495228 | 16734074 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:03:00 |
| 9495229 | 16734073 | 2017 | 10 | 39 | 6 | 0 | Male | 2017-10-01 00:02:00 |

```
9495230  16734072  2017     10    39    6      0  Female  2017-10-01 00:01:00
9495231  16734071  2017     10    39    6      0    Male  2017-10-01 00:01:00
9495232  16734070  2017     10    39    6      0    Male  2017-10-01 00:01:00
9495233  16734067  2017     10    39    6      0  Female  2017-10-01 00:00:00
9495234  16734066  2017     10    39    6      0  Female  2017-10-01 00:00:00

                    stoptime  tripduration  temperature  latitude_start  \
0        2014-07-01 00:07:00     10.066667         68.0       41.939365
1        2014-07-01 00:00:00      4.383333         68.0       41.864580
2        2014-06-30 23:35:00      2.100000         68.0       41.921687
3        2014-07-01 00:24:00     58.016667         68.0       41.877749
4        2014-06-30 23:26:00     10.633333         68.0       41.872187
5        2014-06-30 23:17:00      5.600000         68.0       41.933341
6        2014-06-30 23:13:00      5.066667         68.0       41.882091
7        2014-06-30 23:16:00      8.750000         68.0       41.891738
8        2014-06-30 23:09:00      2.783333         68.0       41.961626
9        2014-06-30 22:51:00      4.833333         68.0       41.874337
10       2014-06-30 22:27:00      5.466667         68.0       41.888243
11       2014-06-30 22:22:00      5.483333         68.0       41.809443
12       2014-06-30 22:17:00      2.716667         68.0       41.875933
13       2014-06-30 22:08:00      3.766667         68.0       41.903448
14       2014-06-30 22:24:00     20.483333         68.0       41.933341
15       2014-06-30 22:10:00      6.916667         68.0       41.881487
16       2014-06-30 22:13:00     11.233333         68.0       41.896802
17       2014-06-30 22:29:00     27.500000         68.0       41.891738
18       2014-06-30 21:59:00      4.150000         70.0       41.890749
19       2014-06-30 22:01:00      5.633333         70.0       41.912202
20       2014-06-30 21:54:00      8.116667         70.0       41.897660
21       2014-06-30 21:56:00     10.850000         70.0       41.896910
22       2014-06-30 21:48:00      2.816667         70.0       41.899643
23       2014-06-30 22:07:00     22.333333         70.0       41.926277
24       2014-06-30 22:02:00     18.116667         70.0       41.891860
25       2014-06-30 21:53:00      9.200000         70.0       41.897660
26       2014-06-30 21:51:00      7.050000         70.0       41.881469
27       2014-06-30 21:46:00      2.100000         70.0       41.954245
28       2014-06-30 21:50:00      6.400000         70.0       41.906724
29       2014-06-30 22:00:00     17.633333         70.0       41.909592
...                      ...           ...          ...             ...
9495205  2017-10-01 00:17:00      7.066667         53.1       41.882091
9495206  2017-10-01 00:21:00     11.383333         53.1       41.878287
9495207  2017-10-01 00:14:00      4.933333         53.1       41.853605
9495208  2017-10-01 00:21:00     12.500000         53.1       41.939743
9495209  2017-10-01 00:15:00      6.183333         53.1       41.882242
9495210  2017-10-01 00:16:00      8.016667         53.1       41.874754
9495211  2017-10-01 00:16:00      8.316667         53.1       41.874754
9495212  2017-10-01 00:27:00     19.150000         53.1       41.935733
9495213  2017-10-01 00:27:00     19.533333         53.1       41.935733
9495214  2017-10-01 00:13:00      5.583333         53.1       41.925563
```

| | | | | |
|---|---|---|---|---|
| 9495215 | 2017-10-01 00:13:00 | 5.833333 | 53.1 | 41.925563 |
| 9495216 | 2017-10-01 00:28:00 | 21.516667 | 53.1 | 41.945636 |
| 9495217 | 2017-10-01 00:28:00 | 21.516667 | 53.1 | 41.945636 |
| 9495218 | 2017-10-01 00:29:00 | 22.083333 | 53.1 | 41.889187 |
| 9495219 | 2017-10-01 00:12:00 | 5.550000 | 53.1 | 41.851375 |
| 9495220 | 2017-10-01 00:21:00 | 15.166667 | 53.1 | 41.851375 |
| 9495221 | 2017-10-01 00:09:00 | 3.716667 | 53.1 | 41.940106 |
| 9495222 | 2017-10-01 00:21:00 | 16.700000 | 53.1 | 41.943739 |
| 9495223 | 2017-10-01 00:12:00 | 7.383333 | 53.1 | 41.906724 |
| 9495224 | 2017-10-01 00:13:00 | 9.183333 | 53.1 | 41.886835 |
| 9495225 | 2017-10-01 00:11:00 | 6.983333 | 53.1 | 41.977997 |
| 9495226 | 2017-10-01 00:17:00 | 12.866667 | 53.1 | 41.966555 |
| 9495227 | 2017-10-01 00:10:00 | 6.000000 | 53.1 | 41.943739 |
| 9495228 | 2017-10-01 00:37:00 | 33.366667 | 53.1 | 41.885483 |
| 9495229 | 2017-10-01 00:12:00 | 9.750000 | 53.1 | 41.961004 |
| 9495230 | 2017-10-01 00:12:00 | 11.066667 | 53.1 | 41.932620 |
| 9495231 | 2017-10-01 00:12:00 | 11.033333 | 53.1 | 41.932620 |
| 9495232 | 2017-10-01 00:15:00 | 13.950000 | 53.1 | 41.912133 |
| 9495233 | 2017-10-01 00:06:00 | 6.016667 | 53.1 | 41.918084 |
| 9495234 | 2017-10-01 00:12:00 | 12.350000 | 53.1 | 41.853780 |

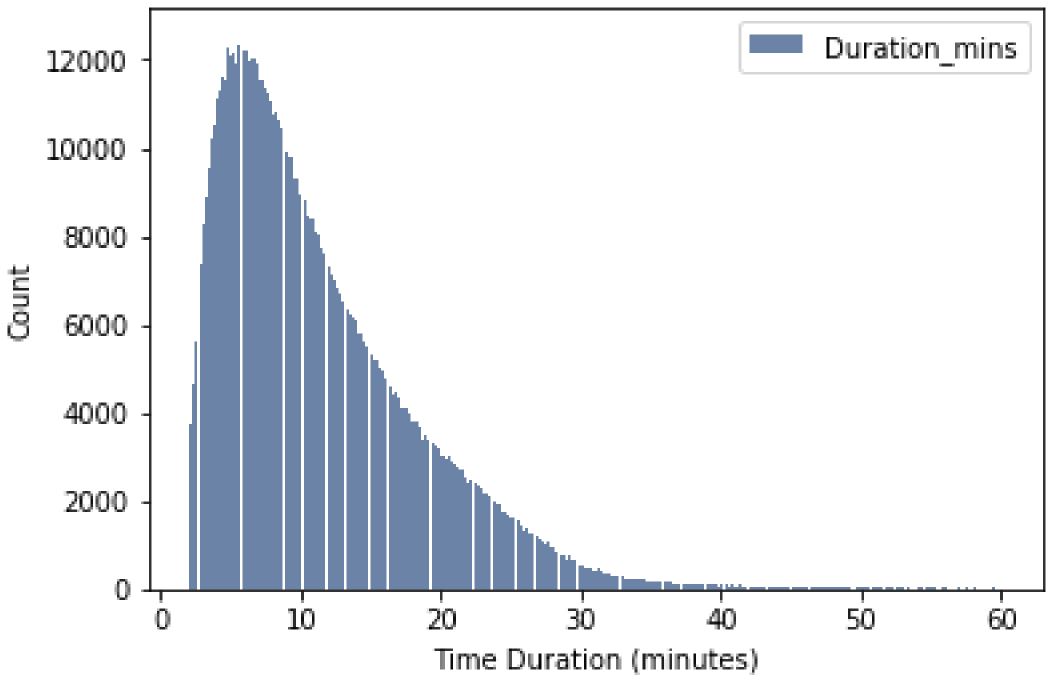
| | longitude_start | latitude_end | longitude_end |
|---|---|---|---|
| 0 | -87.668385 | 41.945512 | -87.645980 |
| 1 | -87.646930 | 41.869482 | -87.655486 |
| 2 | -87.653714 | 41.919936 | -87.648830 |
| 3 | -87.649633 | 41.887155 | -87.627750 |
| 4 | -87.661501 | 41.877749 | -87.649633 |
| 5 | -87.648747 | 41.932595 | -87.665939 |
| 6 | -87.639833 | 41.884730 | -87.627734 |
| 7 | -87.626937 | 41.891860 | -87.620620 |
| 8 | -87.674101 | 41.963250 | -87.679258 |
| 9 | -87.639566 | 41.883380 | -87.641170 |
| 10 | -87.636390 | 41.878948 | -87.639750 |
| 11 | -87.591875 | 41.799568 | -87.594747 |
| 12 | -87.630585 | 41.881320 | -87.629521 |
| 13 | -87.669313 | 41.895966 | -87.667747 |
| 14 | -87.648747 | 41.895966 | -87.667747 |
| 15 | -87.654752 | 41.883380 | -87.641170 |
| 16 | -87.635638 | 41.881469 | -87.635177 |
| 17 | -87.626937 | 41.909396 | -87.677692 |
| 18 | -87.632060 | 41.884576 | -87.631890 |
| 19 | -87.634664 | 41.904509 | -87.640500 |
| 20 | -87.623510 | 41.899007 | -87.629928 |
| 21 | -87.621743 | 41.893843 | -87.641851 |
| 22 | -87.667700 | 41.899714 | -87.677234 |
| 23 | -87.630834 | 41.978353 | -87.659753 |
| 24 | -87.620620 | 41.910579 | -87.638618 |
| 25 | -87.623510 | 41.886024 | -87.624117 |

```
26             -87.635177     41.891733     -87.648727
27             -87.654406     41.954383     -87.648043
28             -87.634830     41.896544     -87.630931
29             -87.653497     41.943340     -87.670970
...                   ...           ...            ...
9495205        -87.639833     41.894666     -87.638437
9495206        -87.643909     41.899368     -87.648480
9495207        -87.631886     41.857950     -87.640826
9495208        -87.658865     41.932684     -87.636250
9495209        -87.641066     41.877726     -87.654787
9495210        -87.649807     41.869417     -87.660996
9495211        -87.649807     41.865054     -87.656959
9495212        -87.663576     41.904613     -87.640552
9495213        -87.663576     41.904613     -87.640552
9495214        -87.658404     41.921822     -87.644140
9495215        -87.658404     41.921822     -87.644140
9495216        -87.727737     41.945636     -87.727737
9495217        -87.727737     41.945636     -87.727737
9495218        -87.627754     41.905664     -87.638517
9495219        -87.618835     41.857813     -87.624550
9495220        -87.618835     41.857813     -87.624550
9495221        -87.645451     41.933666     -87.648959
9495222        -87.664020     41.921822     -87.644140
9495223        -87.634830     41.910578     -87.649422
9495224        -87.622320     41.870769     -87.625734
9495225        -87.668047     41.969090     -87.674237
9495226        -87.688487     41.973815     -87.659660
9495227        -87.664020     41.946176     -87.673308
9495228        -87.652305     41.907655     -87.672552
9495229        -87.649603     41.945529     -87.646439
9495230        -87.642385     41.925563     -87.658404
9495231        -87.642385     41.925563     -87.658404
9495232        -87.634656     41.939743     -87.658865
9495233        -87.643749     41.912133     -87.634656
9495234        -87.646650     41.857556     -87.661535

[9495235 rows x 15 columns]
```

```
In [5]: #for faster output
        df2 = df[['tripduration','trip_id']]
```

Here's the histogram of birth weights:

```
In [6]: hist = thinkstats2.Hist(df2.tripduration, label='Duration_mins')
        thinkplot.Hist(hist)
        thinkplot.Config(xlabel='Time Duration (minutes)', ylabel='Count')
```

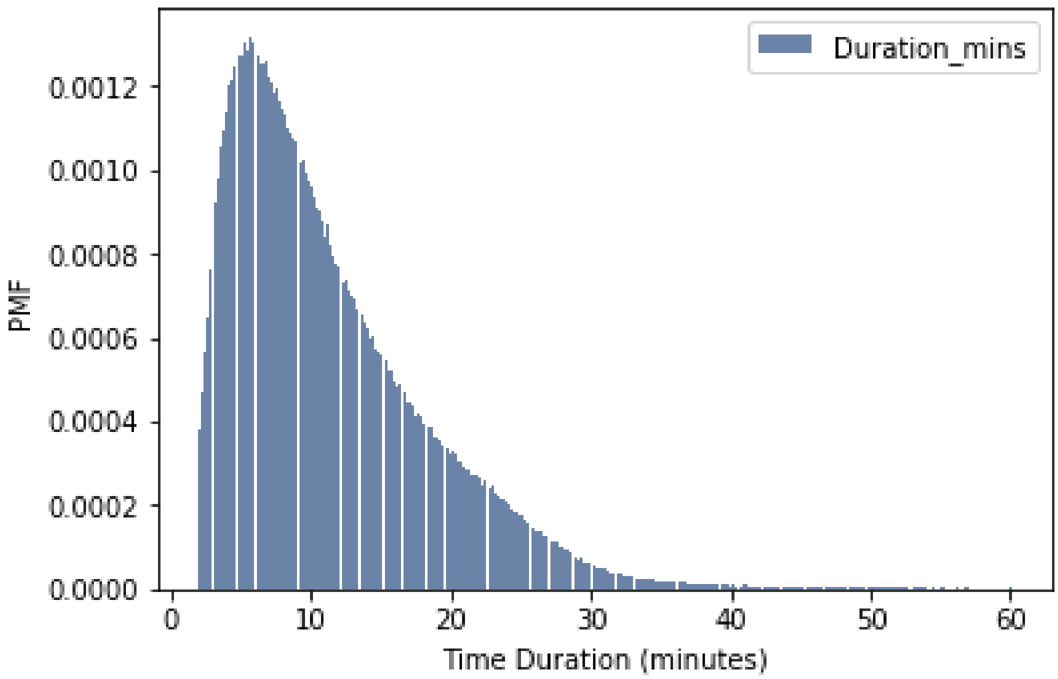



Normalize the distribution

```
In [7]: n = hist.Total()
        pmf = hist.Copy()
        for x, freq in hist.Items():
            pmf[x] = freq / n
```

Probability Mass Function (PMF).

```
In [8]: thinkplot.Hist(pmf)
        thinkplot.Config(xlabel='Time Duration (minutes)', ylabel='PMF')
```




```
In [9]: #OUTLIERS
        for pmf, freq in hist.Smallest(10):
            print(pmf, freq)

        #10 lowest values are roughly 2 minutes. Could be people commuting to the train
```

```
2.0 3370
2.0166666666666666 3493
2.033333333333333 3380
2.05 3588
2.066666666666667 3553
2.083333333333333 3721
2.1 3739
2.1166666666666667 3818
2.1333333333333333 3960
2.15 4017
```

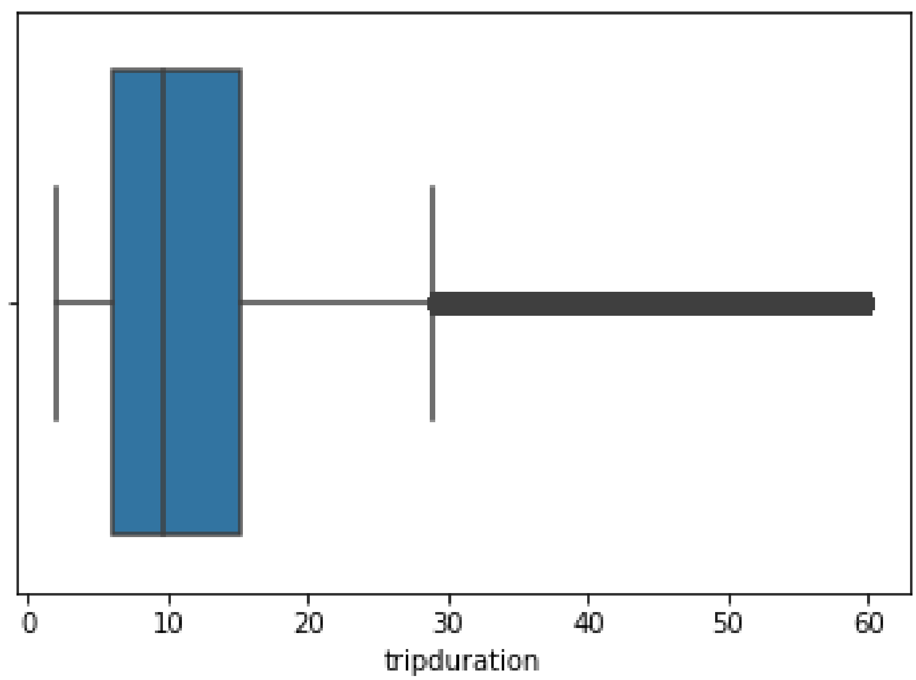
```
In [10]: for pmf, freq in hist.Largest(10):
             print(pmf, freq)
         #10 largest values are close to an hour, most likely due to the fact that divvy bikes
```

```
60.0 19
59.983333333333333 14
59.96666666666667 8
```

```
59.95 8
59.93333333333333 16
59.91666666666666 12
59.9 12
59.88333333333333 16
59.86666666666667 7
59.85 10
```

In [11]:
```python
#Outliers test for trip duration
#There are outliers for 30 minutes and on
sns.boxplot(x=df['tripduration'])
```

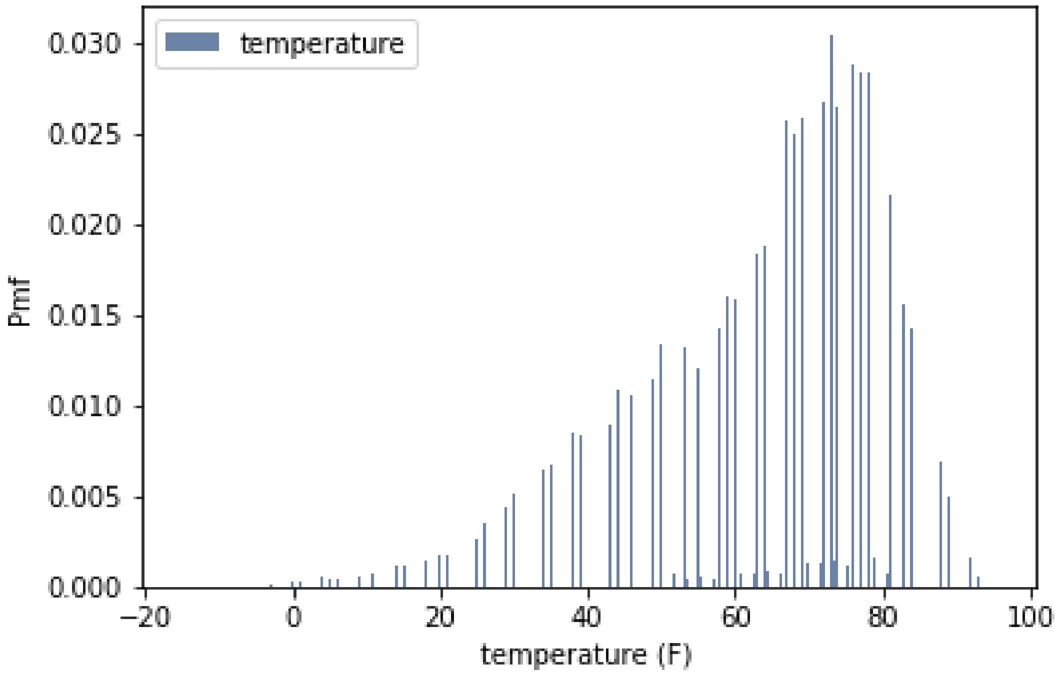
Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e3c5f36b00>
```




PMF for each of the variables

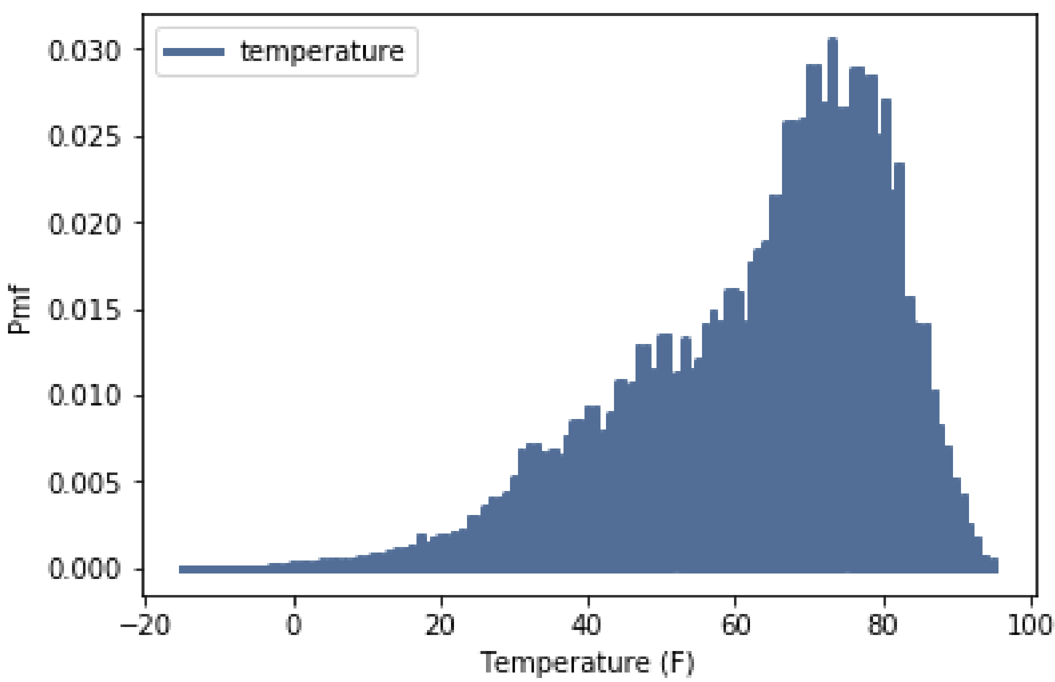
In [12]:
```python
pmf = thinkstats2.Pmf(df.temperature, label='temperature')
```

In [13]:
```python
thinkplot.Hist(pmf)
thinkplot.Config(xlabel='temperature (F)', ylabel='Pmf')
```




Stepped function PMF for Temperature

```
In [14]: thinkplot.Pmf(pmf)
         thinkplot.Config(xlabel='Temperature (F)', ylabel='Pmf')
```

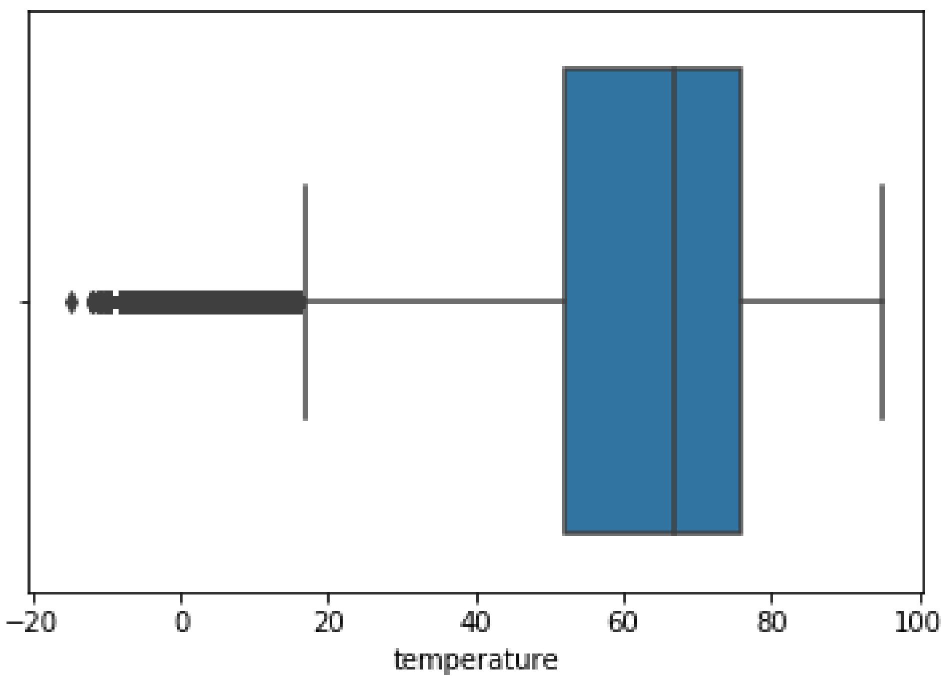


Distributions for month, week,day and hour

In [15]:
```
timeduration_pmf = thinkstats2.Pmf(df.tripduration, label = 'trip duration')
temperature_pmf = thinkstats2.Pmf(df.temperature, label = 'temperature')
month_pmf = thinkstats2.Pmf(df.month, label='month')
week_pmf = thinkstats2.Pmf(df.week, label='week')
day_pmf = thinkstats2.Pmf(df.day, label='day')
hour_pmf = thinkstats2.Pmf(df.hour, label='hour')
```

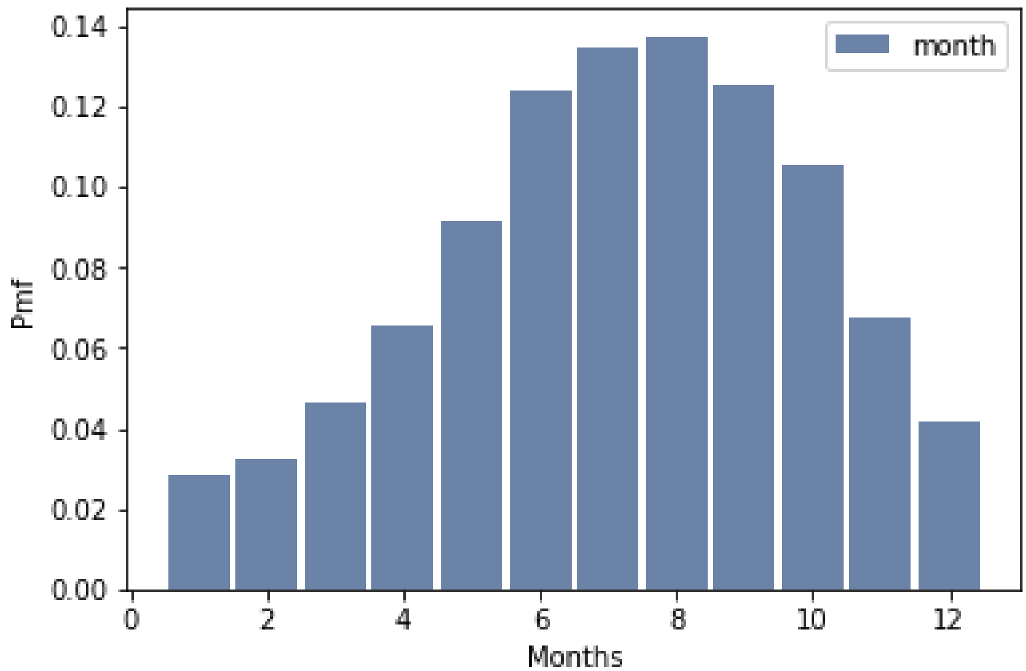
In [16]:
```
sns.boxplot(x=df['temperature'])
#20 and below are shown as outliers, but Chicago winters do get very cold so I will k
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1e3c6234a90>

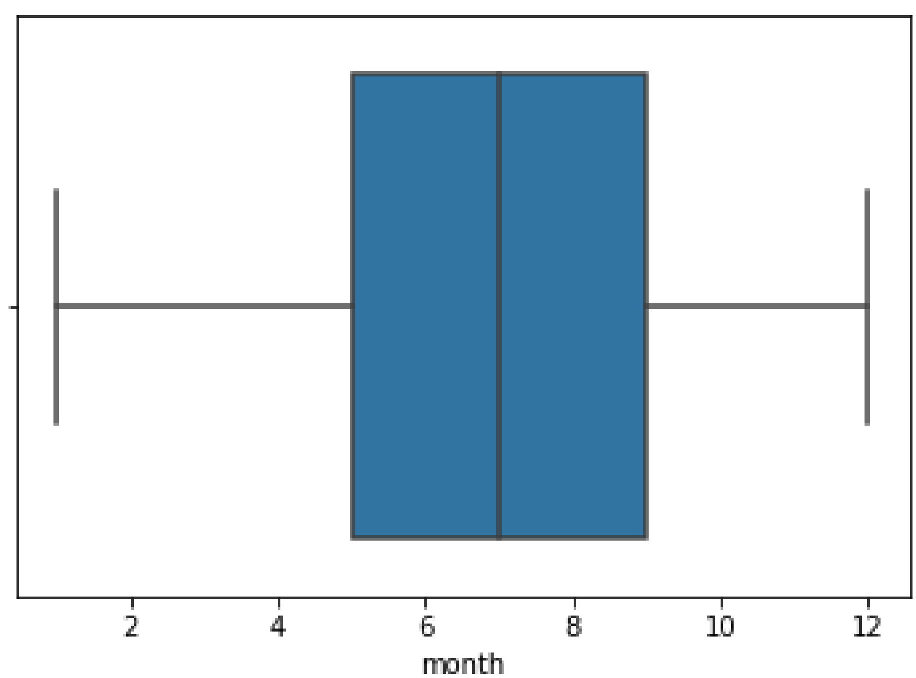



Month Histogram

In [17]:
```
thinkplot.Hist(month_pmf)
thinkplot.Config(xlabel='Months', ylabel='Pmf')
```

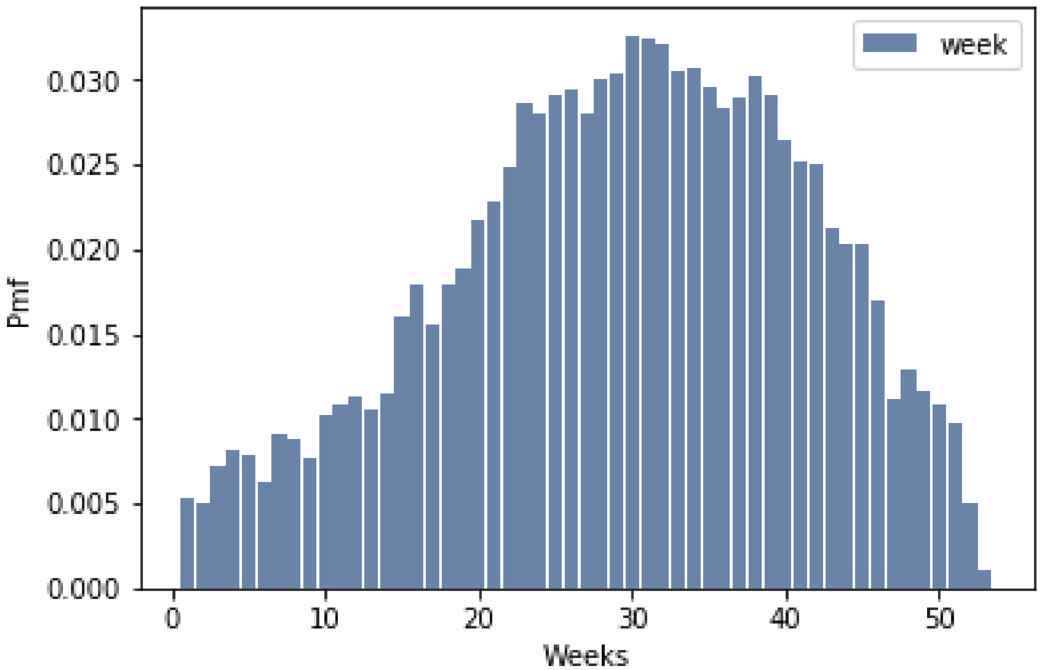



```
In [18]: sns.boxplot(x=df['month']) #No outliers for months

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1e3c633ee48>
```

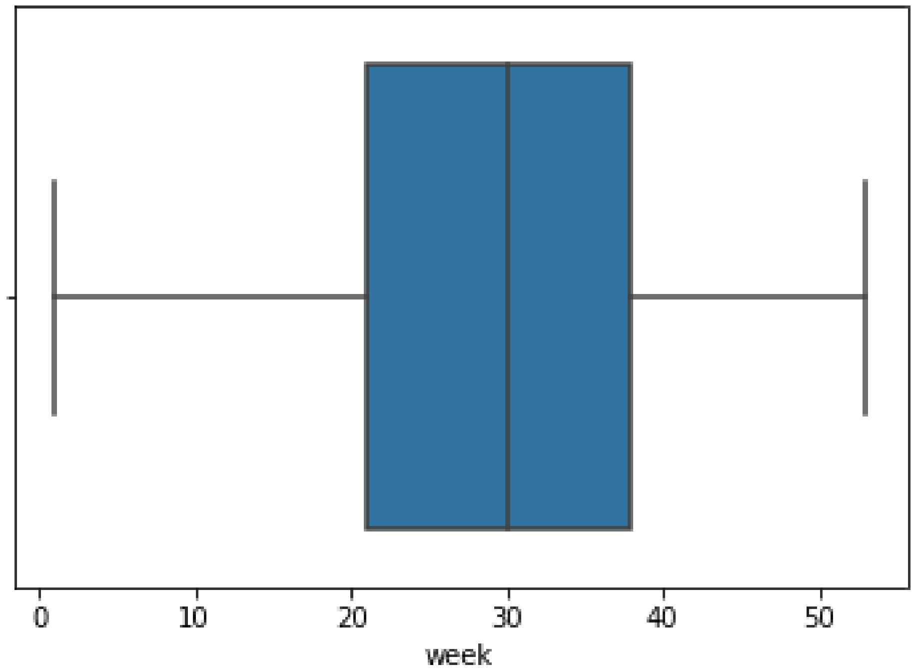


Week Histogram

```
In [19]: thinkplot.Hist(week_pmf)
         thinkplot.Config(xlabel='Weeks', ylabel='Pmf')
```

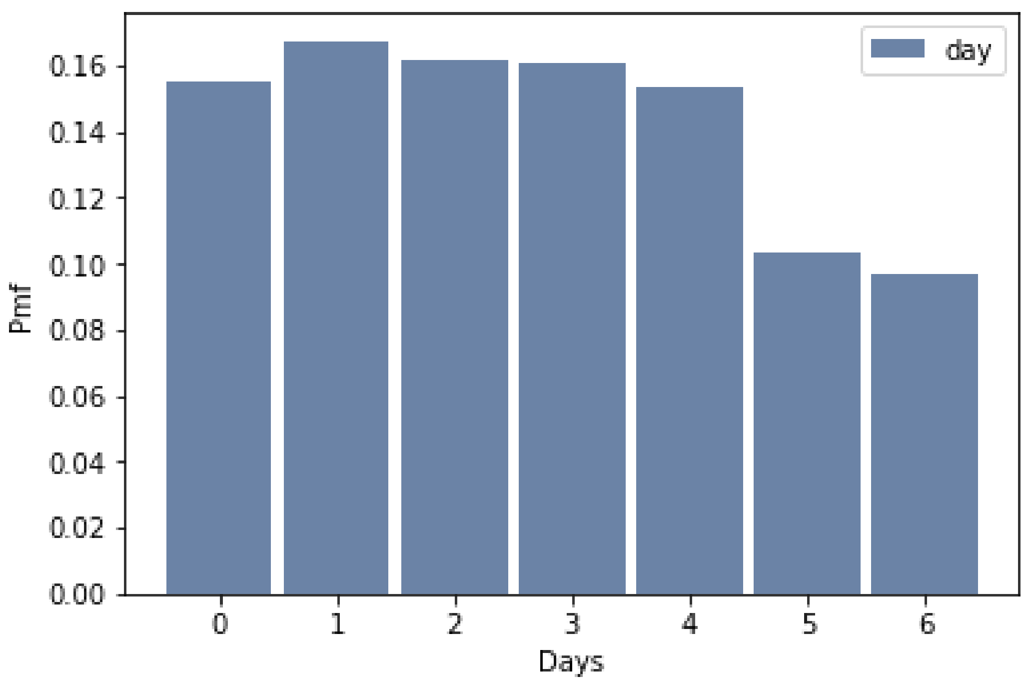



```
In [20]: sns.boxplot(x=df['week']) #no outliers in weeks

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1e3c643e898>
```



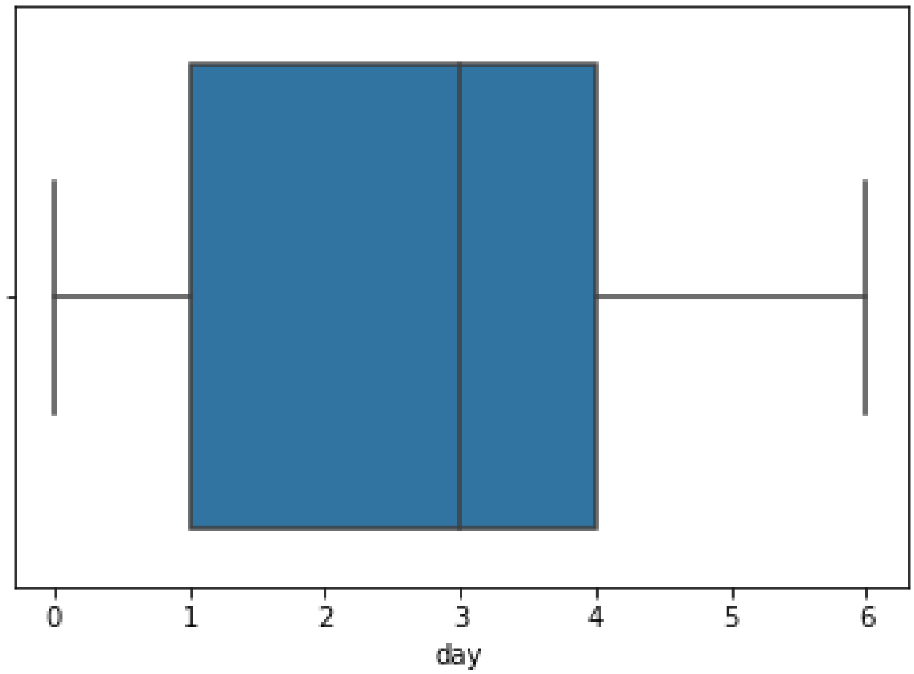

Days Histogram

```
In [21]: thinkplot.Hist(day_pmf)
         thinkplot.Config(xlabel='Days', ylabel='Pmf')
```



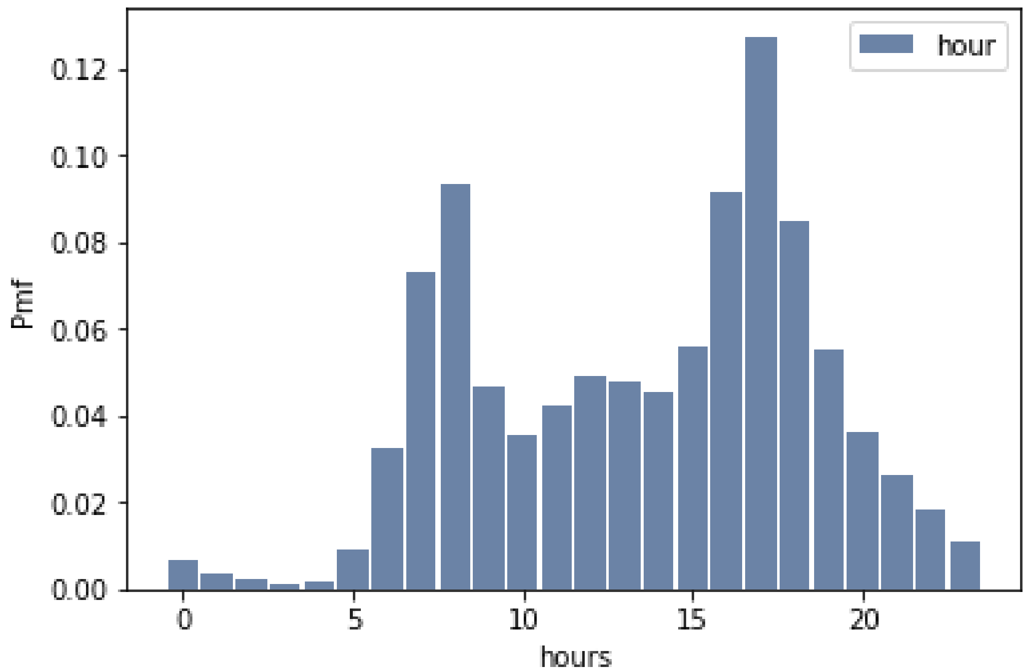
In [22]:
```
sns.boxplot(x=df['day']) #no outliers in day
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1e3c6334208>

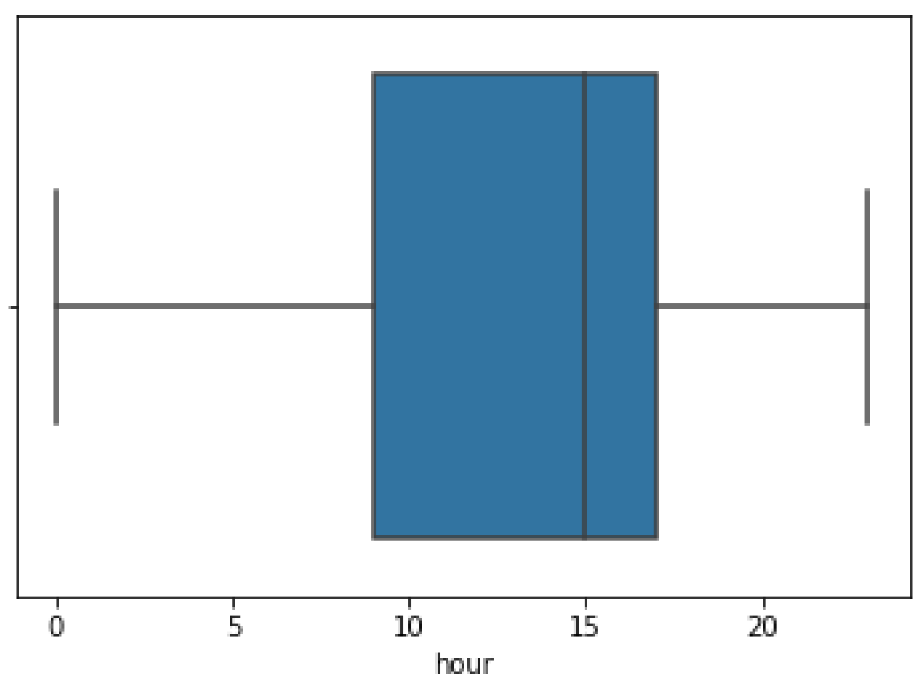



Hours Histogram

In [23]:
```
thinkplot.Hist(hour_pmf)
thinkplot.Config(xlabel='hours', ylabel='Pmf')
```

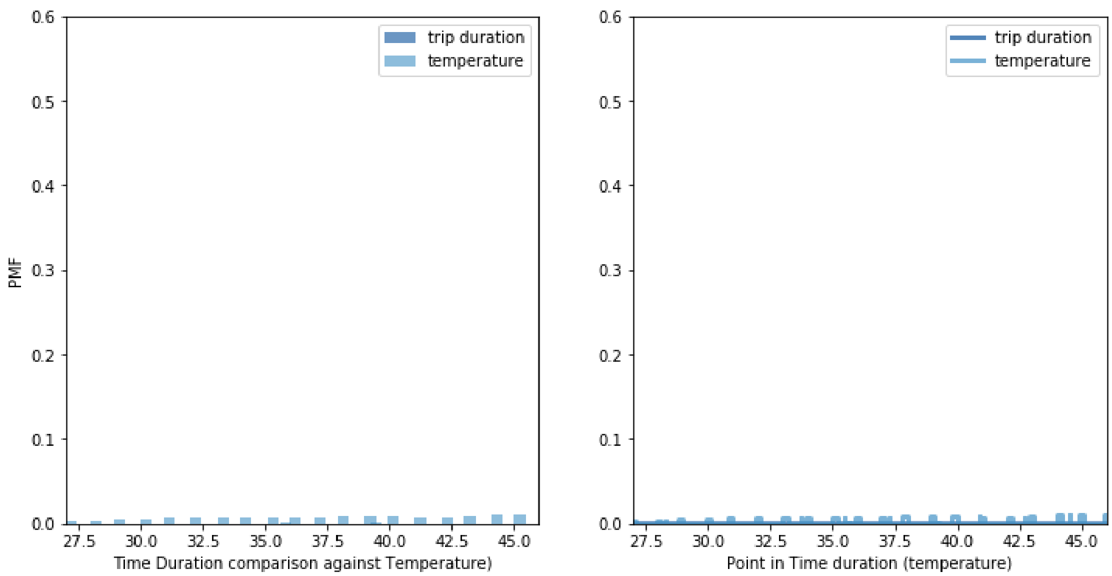



```
In [24]: sns.boxplot(x=df['hour']) #no outlier in hours

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1e46ba90c88>
```



Exploratory Analysis of the Data

In [25]:
```
width=0.45
axis = [27, 46, 0, 0.6]
thinkplot.PrePlot(2, cols=2)
thinkplot.Hist(timeduration_pmf, align='right', width=width)
thinkplot.Hist(temperature_pmf, align='left', width=width)
thinkplot.Config(xlabel='Time Duration comparison against Temperature)', ylabel='PMF'

thinkplot.PrePlot(2)
thinkplot.SubPlot(2)
thinkplot.Pmfs([timeduration_pmf, temperature_pmf])
thinkplot.Config(xlabel='Point in Time duration (temperature)', axis=axis)
```




Additional Exploratory Analysis

In [26]:
```
df.temperature.mean() #62.999
df.tripduration.mean() #11.44
mode(df.month) #8, People ride the most in August
mode(df.week) #30 #People rode the most the 30th week of the month
mode(df.day) #1, people rode the most on the first of the month
mode(df.hour) #17, people mostly ride at 5PM
```

Out[26]: 17

Standard Deviation

In [27]:
```
temp_std = df.temperature.std()
print(temp_std)
```

```
17.200858851614083
```

```
In [28]: tripduration_std = df.tripduration.std()
         print(tripduration_std)
```

```
7.206061042256747
```

Variance

```
In [29]: temp_var = df.temperature.var()
         print(temp_var)
```

```
295.8695452331505
```

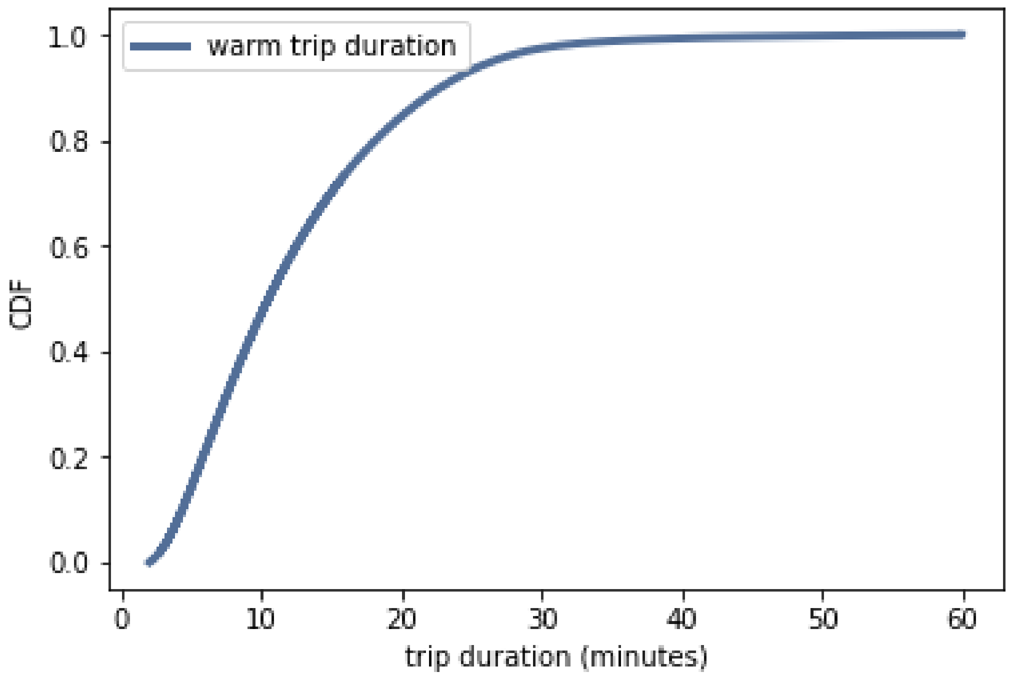
```
In [30]: tripduration_var = df.tripduration.var()
         print(tripduration_var)
```

```
51.92731574473039
```

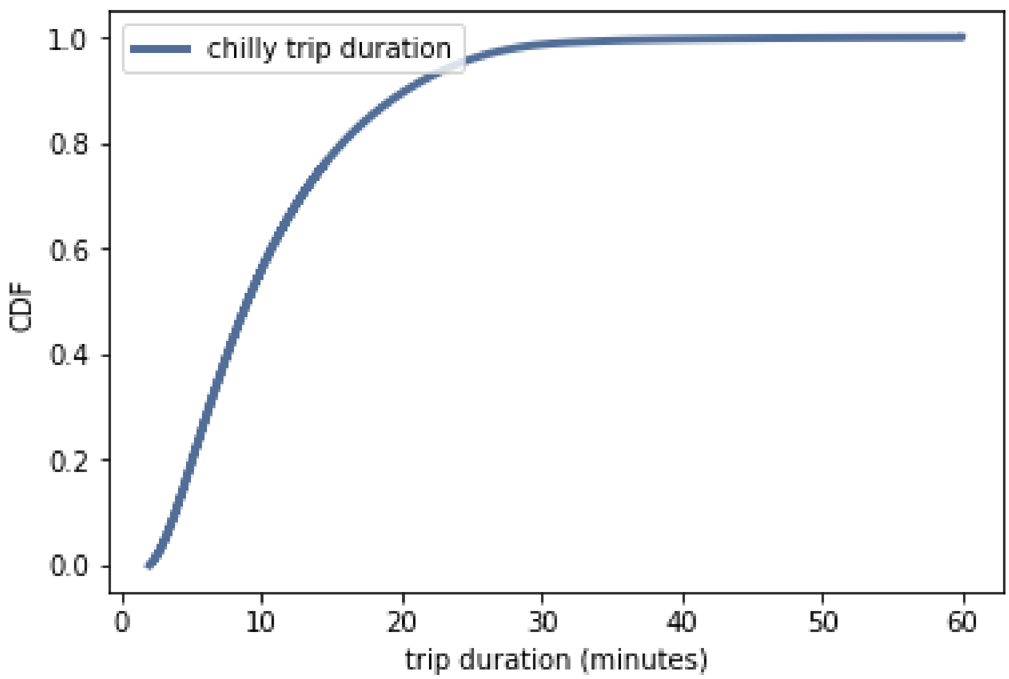
Cumulative Distributive Function

```
In [31]: # Create variable with TRUE if temperature >= 70
         warm = df['temperature'] >= 70

         # Create variable with TRUE if temperature < 70
         chilly = df['temperature'] < 70


         warm_df = df[warm]
         chilly_df = df[chilly]
```

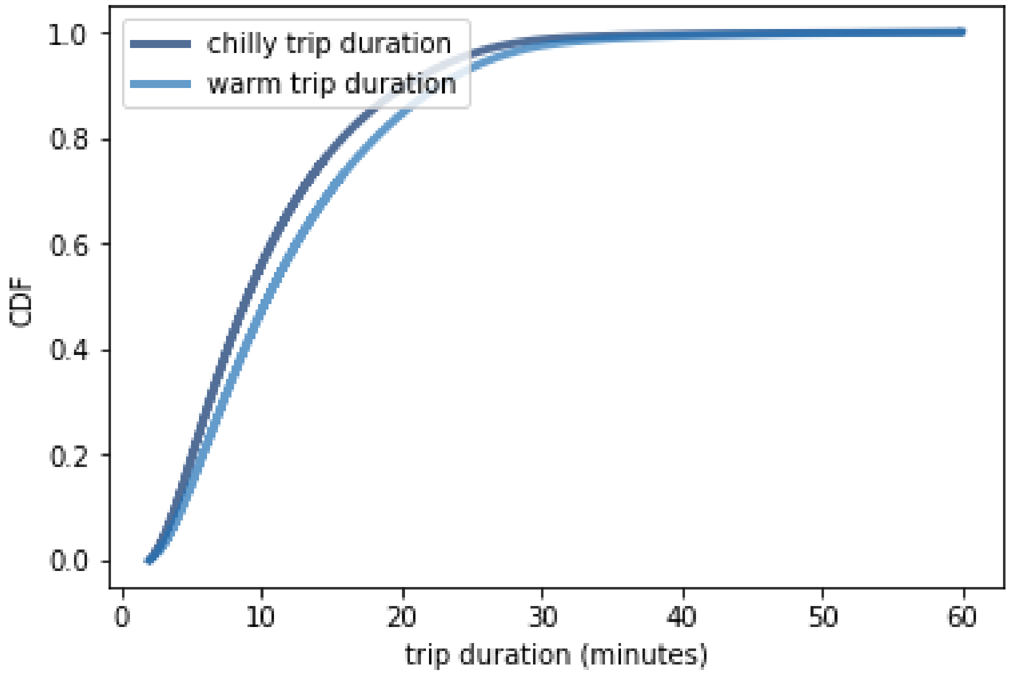
```
In [32]: warm_tripduration_cdf = thinkstats2.Cdf(warm_df.tripduration, label='warm trip duratio
         thinkplot.Cdf(warm_tripduration_cdf)
         thinkplot.Config(xlabel='trip duration (minutes)', ylabel='CDF', loc='upper left')
```




```
In [33]: chilly_tripduration_cdf = thinkstats2.Cdf(chilly_df.tripduration, label='chilly trip d
         thinkplot.Cdf(chilly_tripduration_cdf)
         thinkplot.Config(xlabel='trip duration (minutes)', ylabel='CDF', loc='upper left')
```



```
In [34]: #comparison
         thinkplot.Cdfs([chilly_tripduration_cdf,warm_tripduration_cdf])
         thinkplot.Show(xlabel='trip duration (minutes)',ylabel='CDF')
```



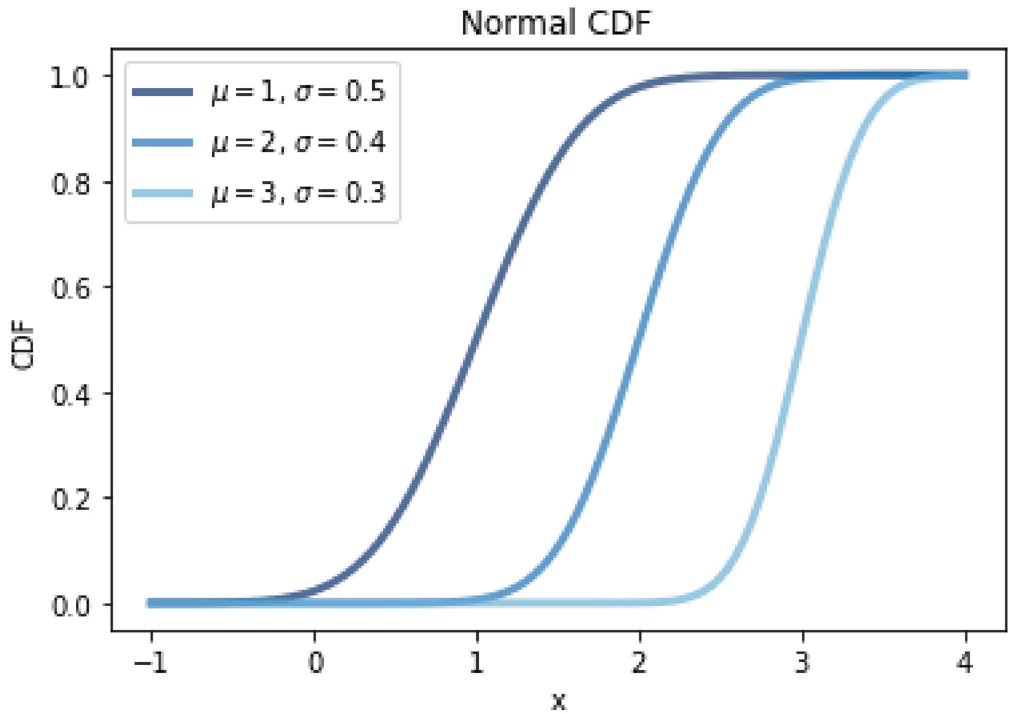

```
<Figure size 576x432 with 0 Axes>
```

By comparing colder and warmer temperatures with their duration, we can see that chilly bike rides are slightly shorter than warmer bike rides

Analytical Distribution

```
In [35]: #NORMAL CDF to for visual
         thinkplot.PrePlot(3)

         mus = [1.0, 2.0, 3.0]
         sigmas = [0.5, 0.4, 0.3]
         for mu, sigma in zip(mus, sigmas):
             xs, ps = thinkstats2.RenderNormalCdf(mu=mu, sigma=sigma,
                                                  low=-1.0, high=4.0)
             label = r'$\mu=%g$, $\sigma=%g$' % (mu, sigma)
             thinkplot.Plot(xs, ps, label=label)
```

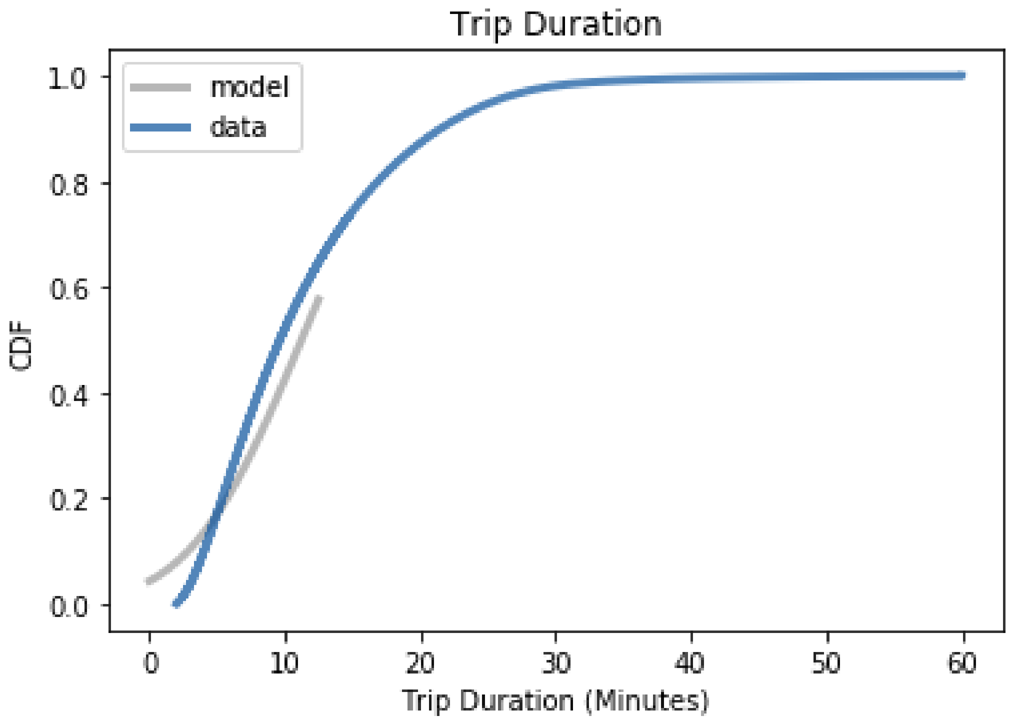
```
thinkplot.Config(title='Normal CDF', xlabel='x', ylabel='CDF',
                 loc='upper left')
```




In [71]:
```
#OBSERVED CDF AND MODEL
# estimate parameters: trimming outliers yields a better fit
mu, var = thinkstats2.TrimmedMeanVar(df.tripduration, p=0.01)
print('Mean, Var', mu, var)

# plot the model
sigma = np.sqrt(var)
print('Sigma', sigma)
xs, ps = thinkstats2.RenderNormalCdf(mu, sigma, low=0, high=12.5)
thinkplot.Plot(xs, ps, label='model', color='0.6')

# plot the data
cdf = thinkstats2.Cdf(df.tripduration, label='data')

thinkplot.PrePlot(1)
thinkplot.Cdf(cdf)
thinkplot.Config(title='Trip Duration',
                 xlabel='Trip Duration (Minutes)',
                 ylabel='CDF')
```

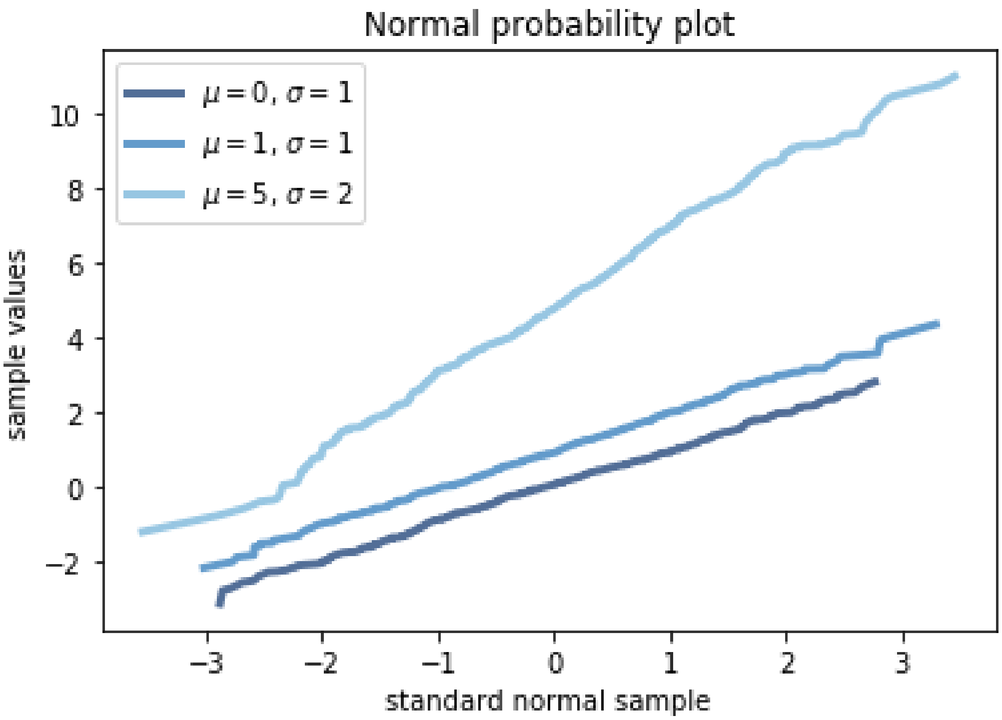
```
Mean, Var 11.237980359502187 42.62121185708795
```

```
Sigma 6.528492311176291
```



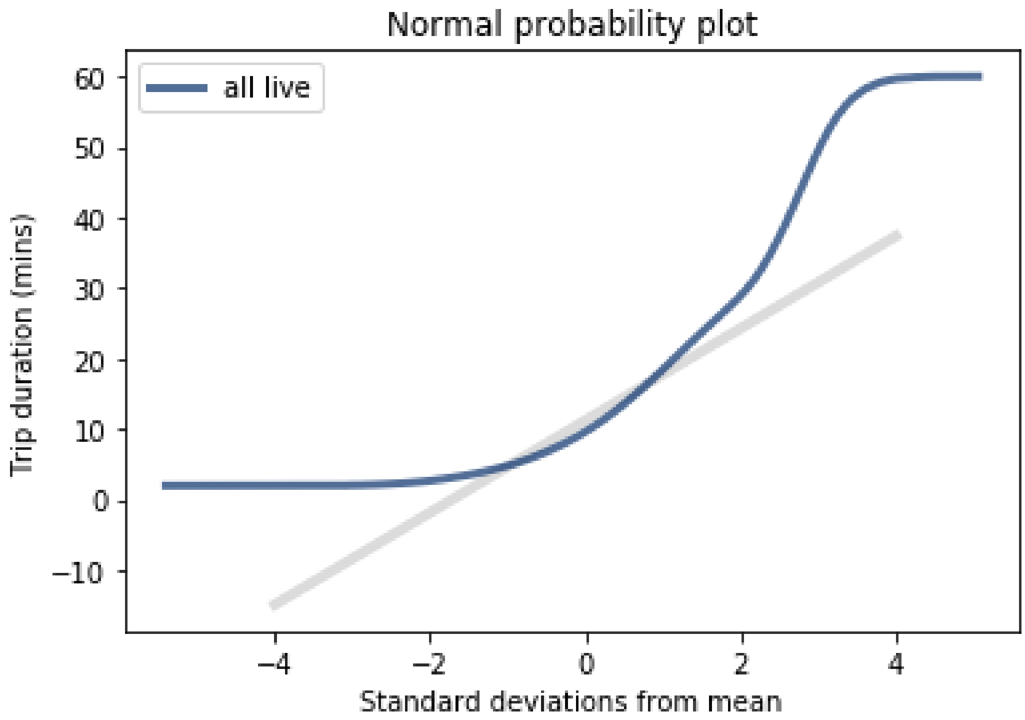

Model fits data except for in th left tail up to 15 minutes
If data is from a normal distribution then the plot will be straight

```
In [72]: n = 1000
         thinkplot.PrePlot(3)

         mus = [0, 1, 5]
         sigmas = [1, 1, 2]

         for mu, sigma in zip(mus, sigmas):
             sample = np.random.normal(mu, sigma, n)
             xs, ys = thinkstats2.NormalProbability(sample)
             label = '$\mu=%d$, $\sigma=%d$' % (mu, sigma)
             thinkplot.Plot(xs, ys, label=label)

         thinkplot.Config(title='Normal probability plot',
                          xlabel='standard normal sample',
                          ylabel='sample values')
```




Normal Probability Plot for trip duration

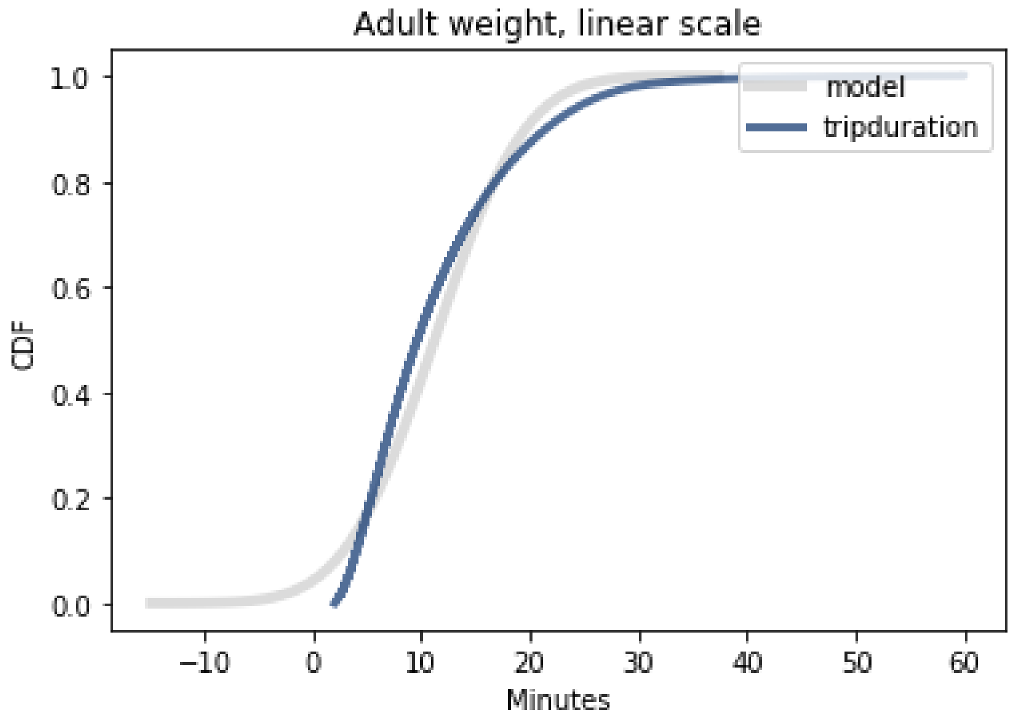
```
In [73]: mean, var = thinkstats2.TrimmedMeanVar(df.tripduration, p=0.01)
         std = np.sqrt(var)

         xs = [-4, 4]
         fxs, fys = thinkstats2.FitLine(xs, mean, std)
         thinkplot.Plot(fxs, fys, linewidth=4, color='0.8')

         xs, ys = thinkstats2.NormalProbability(df.tripduration)
         thinkplot.Plot(xs, ys, label='trip duration')

         thinkplot.Config(title='Normal probability plot',
                          xlabel='Standard deviations from mean',
                          ylabel='Trip duration (mins)')
```




The results cause me to not choose a normal probability plot

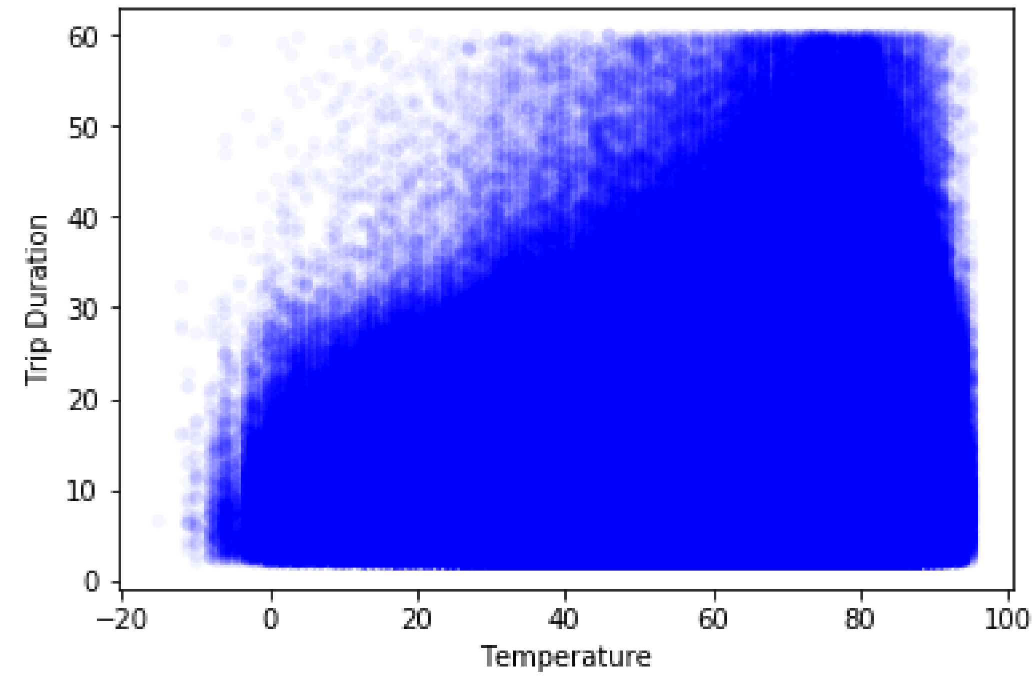
```
In [80]: tripduration = df.tripduration
         #Testing for normal lognormal distribution
         def MakeNormalModel(tripduration):
             """Plots a CDF with a Normal model.

             weights: sequence
             """
             cdf = thinkstats2.Cdf(tripduration, label='tripduration')

             mean, var = thinkstats2.TrimmedMeanVar(tripduration)
             std = np.sqrt(var)
             print('n, mean, std', len(tripduration), mean, std)

             xmin = mean - 4 * std
             xmax = mean + 4 * std

             xs, ps = thinkstats2.RenderNormalCdf(mean, std, xmin, xmax)
             thinkplot.Plot(xs, ps, label='model', linewidth=4, color='0.8')
             thinkplot.Cdf(cdf)

In [82]: MakeNormalModel(tripduration)
         thinkplot.Config(title='Adult weight, linear scale', xlabel='Minutes',
                          ylabel='CDF', loc='upper right')
```

```
n, mean, std 9495235 11.237980359502187 6.528492311176291
```

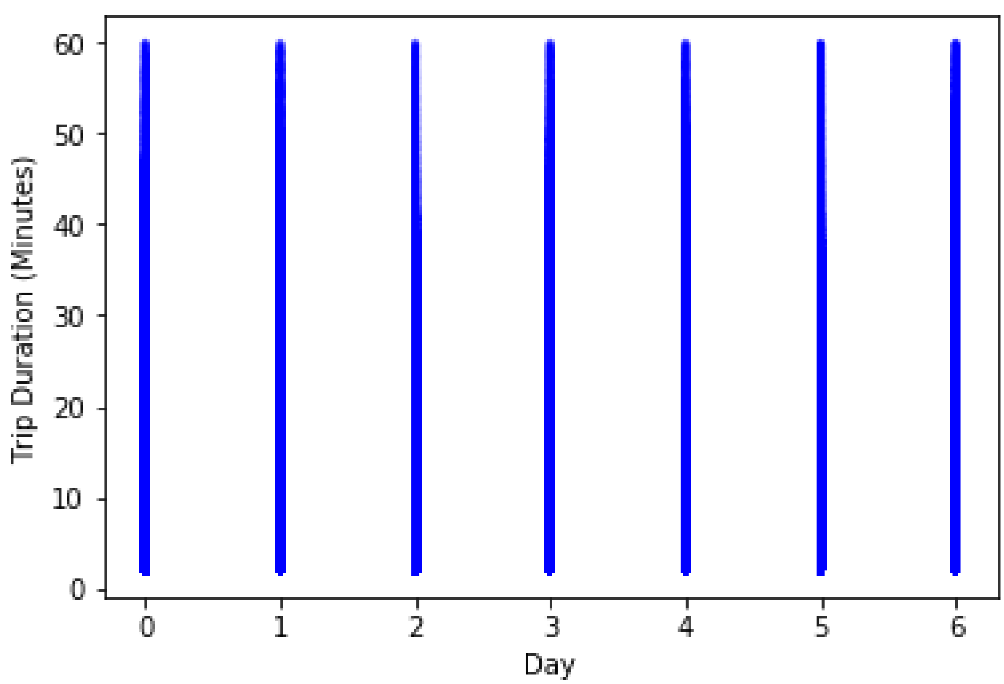



Scatter Plots

```
In [85]: thinkplot.Scatter(df.temperature, df.tripduration, alpha=.04)
         thinkplot.Config(xlabel='Temperature',
                          ylabel='Trip Duration',
                          legend=False)
```




```
In [86]: thinkplot.Scatter(df.day, df.tripduration, alpha=0.04, s=10)
         thinkplot.Config(xlabel='Day',
                          ylabel='Trip Duration (Minutes)',
                          legend=False)
```



```
In [90]: #Covariance
         np.corrcoef(df.tripduration, df.temperature)
         np.corrcoef(df.tripduration, df.day)

Out[90]: array([[1.        , 0.05348869],
                [0.05348869, 1.        ]])

In [93]: #Pearson's Correlation is more robust
         def SpearmanCorr(xs, ys):
             xs = pd.Series(xs)
             ys = pd.Series(ys)
             return xs.corr(ys, method='spearman')

In [94]: SpearmanCorr(df.tripduration, df.temperature)

Out[94]: 0.13020937038867972

In [95]: SpearmanCorr(df.tripduration, df.day)

Out[95]: 0.04083534974976528
```

Hypothesis Testing

```
In [99]: #Test Correlation
         class CorrelationPermute(thinkstats2.HypothesisTest):

             def TestStatistic(self, data):
                 xs, ys = data
                 test_stat = abs(thinkstats2.Corr(xs, ys))
                 return test_stat

             def RunModel(self):
                 xs, ys = self.data
                 xs = np.random.permutation(xs)
                 return xs, ys

In [97]: class DivvyTest(HypothesisTest):

             def TestStatistic(self, data):
                 df.tripduration, df.temperature = data
                 test_stat = abs(df.tripduration - df.temperature)
                 return test_stat

             def RunModel(self):
                 df.tripduration, df.temperature = self.data
                 n = df.tripduration + df.temperature
```

```
            sample = [random.choice('HT') for _ in range(n)]
            hist = thinkstats2.Hist(sample)
            data = hist['H'], hist['T']
            return data
```

```
In [101]: def RunTests(live, iters=1000):
              """Runs the tests from Chapter 9 with a subset of the data.

              live: DataFrame
              iters: how many iterations to run
              """
              n = len(df)
              chilly_df = df[chilly]
              warm_df = df[warm]
              chilly_df = df[chilly]

              # Test
              data = chilly_df.values, warm_df.values
              ht = DiffMeansPermute(data)
              p1 = ht.PValue(iters=iters)

              ht = CorrelationPermute(data)
              p3 = ht.PValue(iters=iters)

              # chi-squared
              data = chilly_df.values, warm_df.values
              ht = DivvyTest(data)
              p4 = ht.PValue(iters=iters)

              print('%d\t%0.2f\t%0.2f\t%0.2f\t%0.2f' % (n, p1, p2, p3, p4))
```

Regression Test

```
In [103]: import statsmodels.formula.api as smf

In [106]: formula = 'tripduration ~ temperature'
          model = smf.ols(formula, data=df)
          results = model.fit()
          results.summary()

C:\Users\Paulina\Anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1543: Runt
  return 1 - self.ssr/self.centered_tss
C:\Users\Paulina\Anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1554: Runt
  return self.ess/self.df_model


Out[106]: <class 'statsmodels.iolib.summary.Summary'>
          """
                                      OLS Regression Results
```

```
==============================================================================
Dep. Variable:           tripduration   R-squared:                        -inf
Model:                            OLS   Adj. R-squared:                   -inf
Method:                 Least Squares   F-statistic:                      -inf
Date:                Sun, 03 Mar 2019   Prob (F-statistic):                nan
Time:                        18:24:47   Log-Likelihood:             1.9932e+08
No. Observations:             9495235   AIC:                        -3.986e+08
Df Residuals:                 9495234   BIC:                        -3.986e+08
Df Model:                           0
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept       0.0116   4.96e-18   2.33e+15      0.000       0.012       0.012
temperature     1.2726   5.46e-16   2.33e+15      0.000       1.273       1.273
==============================================================================
Omnibus:                        1.000   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.607   Jarque-Bera (JB):          3560713.125
Skew:                           0.000   Prob(JB):                         0.00
Kurtosis:                       0.000   Cond. No.                     1.50e+19
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
[2] The smallest eigenvalue is 5.13e-28. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```