In [1]:

```
# DSC530-T302
# Stephen Smitshoek
# Week04
# Exercise 4-1
```

In [2]:

```
import sys
import numpy as np
import thinkstats2
import math
```

In [3]:

```
def ReadFemPreg(dct_file='2002FemPreg.dct',
                dat_file='2002FemPreg.dat.gz'):
    """Reads the NSFG pregnancy data.

    dct_file: string file name
    dat_file: string file name

    returns: DataFrame
    """
    dct = thinkstats2.ReadStataDct(dct_file)
    df = dct.ReadFixedWidth(dat_file, compression='gzip')
    CleanFemPreg(df)
    return df
```

In [4]:

```
def CleanFemPreg(df):
    """Recodes variables from the pregnancy frame.

    df: DataFrame
    """
    # mother's age is encoded in centiyears; convert to years
    df.agepreg /= 100.0

    # birthwgt_lb contains at least one bogus value (51 lbs)
    # replace with NaN
    df.loc[df.birthwgt_lb > 20, 'birthwgt_lb'] = np.nan

    # replace 'not ascertained', 'refused', 'don't know' with NaN
    na_vals = [97, 98, 99]
    df.birthwgt_lb.replace(na_vals, np.nan, inplace=True)
    df.birthwgt_oz.replace(na_vals, np.nan, inplace=True)
    df.hpagelb.replace(na_vals, np.nan, inplace=True)

    df.babysex.replace([7, 9], np.nan, inplace=True)
    df.nbrnaliv.replace([9], np.nan, inplace=True)

    # birthweight is stored in two columns, lbs and oz.
    # convert to a single column in lb
    # NOTE: creating a new column requires dictionary syntax,
    # not attribute assignment (like df.totalwgt_lb)
    df['totalwgt_lb'] = df.birthwgt_lb + df.birthwgt_oz / 16.0

    # due to a bug in ReadStataDct, the last variable gets clipped;
    # so for now set it to NaN
    df.cmintvw = np.nan
```

In [5]:

```
def data_split(preg_df):
    # Find all the live births and split them into first babies and other babies
    live = preg_df[preg_df.outcome==1]
    first = live[live.birthord == 1]
    other = live[live.birthord != 1]

    return first, other
```

In [6]:

```
def calc_perc_rank(your_weight_lbs, nsfg_data):
    i = 0
    for birth_weight in nsfg_data.totalwgt_lb:
        if your_weight_lbs >= birth_weight:
            i += 1 # increase counter for every weight that is less than or equal to y

    # Calculate the percentage of weights that are less than or equal to your weight
    perc_rank = i / len(nsfg_data.totalwgt_lb) * 100

    return perc_rank
```

In [7]:

```
def main():
    preg_df = ReadFemPreg() # Retrive the pregnacy dataframe
    CleanFemPreg(preg_df) # Clean up the data in the dataframe
    first, other = data_split(preg_df) # Split the data into first live births and oth

    my_weight = 8.1 # My birth weight to be compared to the dataframe

    perc_rank = calc_perc_rank(my_weight, other) # Calculate my percentage rank in the
    print('My percential rank: {}'.format(round(perc_rank, 1)))
```

In [8]:

```
if __name__ == '__main__':
    main()
```

```
My percential rank: 71.4
```