Final Project Part B: Explore Factors Effecting Income in NLSY '79 Data

Team Members

- Member 1: Gina O'Riordan
- Member 2: Marisa Roman
- Member 3: Justin Saslaw

Introduction ¶

In this document, we survey the data from National Longitudinal Survey of Youth 1979 (NLSY79) to review the relationship of three variables to income:

1. Years of Education
2. Gender
3. Race

This EDA analysis will conclude with hypothesis for further analysis. We limit our analysis to income data from 2008 - 2014, as we want to focus on recent information. Note: We originally chose the range 2004 - 2014, but discovered that significant portions of education data were missing in 2004 and 2006, so we further restricted our range.

Description of the Data

We are using three datasets from the National Longitudinal Survey of Youth 1979 (NLSY79). The datasets have been provided to us in a tidied and cleaned form, though we will inspect the data in our chosen variables to determine whether further cleaning is necessary.

To start, let's load our libraries:

In [231]:

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn')

# Display floats with two decimal places
pd.options.display.float_format = '{:,.2f}'.format
```

Income

This analysis will review income as the dependent variable. First load the data and examine it.

In [232]:

```
income_data = pd.read_csv('./data/income_data_nlsy79.csv',usecols=['CASEID','income','year'])
```

In [233]:

```
income_data.head()
```

Out[233]:

	CASEID	income	year
0	1	nan	1982
1	2	10,000.00	1982
2	3	7,000.00	1982
3	4	1,086.00	1982
4	5	2,300.00	1982

Seen in the code below, the survey data ranges from 1982 to 2014. From analyzing the unique year values, it appears the study contains data from even years within the time frame to be studied.

```
In [234]: income_data['year'].unique()
```

```
Out[234]: array([1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992,
                 1993, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012,
                 2014])
```

After truncating the income dataset to retain only years 2008 through 2014 (inclusive)...

```
In [235]: # Truncate income_data so that only years 2008 - 2014 are retained and show unique values
          income_data_restrict = income_data[income_data['year'] >= 2008]
          income_data_restrict['year'].unique()
```

```
Out[235]: array([2008, 2010, 2012, 2014])
```

there are 50,744 entries remaining, 28,433 non-null. Once the data sets are combined into a consolidated dataframe for analysis, the null income values will be removed.

```
In [236]: income_data_restrict.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50744 entries, 241034 to 291777
Data columns (total 3 columns):
CASEID    50744 non-null int64
income    28433 non-null float64
year      50744 non-null int64
dtypes: float64(1), int64(2)
memory usage: 1.5 MB
```

Distribution of Income Data

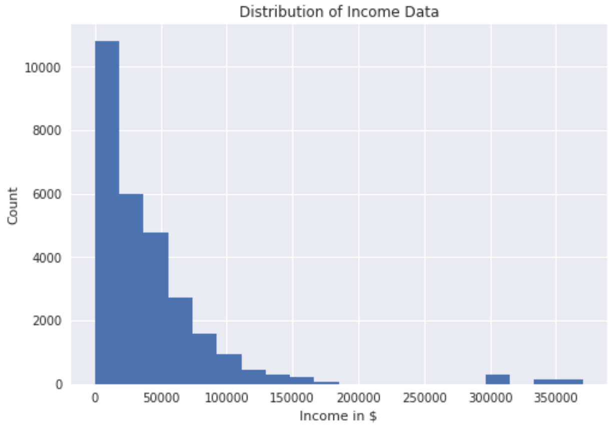
A histogram of the income data reveals a right-skewed distribution, with values over $200K as candidates for outliers. For now, we will leave these values in place and we can filter them out if necessary later on when graphing and analyzing to see the full picture of the data.

In [237]:

```
# Generate histogram of income values for our income dataset to review the distribution
figure, axes = plt.subplots()

# NaN values are dropped to produce the plot
axes.hist(income_data_restrict['income'].dropna(), bins=20)

# Titles and labels added
axes.set_title('Distribution of Income Data')
axes.set_xlabel('Income in $')
axes.set_ylabel('Count')
```

Out[237]:

```
<matplotlib.text.Text at 0x7fe6062879e8>
```



Truncation of Income Data

The data dictionary truncates the income data by taking the top 2% of income each year and assigning it the minimum income value for the top 2% in that year. For example, the maximum income value for 2014 is $370,314.

Repondents with incomes above that amount were recorded as having an income of $370,314. As noted above, we will keep this data in place in its truncated form as to not eliminate the top 2% of income values. We can choose whether to exclude these values if necessary later on when graphing and analyzing to see the full picture of the data.

The code below reviews the maximum income values by year.

In [238]:

```
# remove NaN values
income_data_values = income_data_restrict.dropna()

# groupby year
income_data_values_byyear = income_data_values.groupby(['year'])

# show maximum income per year with new column label
max_income = income_data_values_byyear[['income']].max()
max_income.rename(columns={'income': 'max income'},
                       inplace=True)
max_income
```

Out[238]:

	max income
year	
2008	307,823.00
2010	312,324.00
2012	343,830.00
2014	370,314.00

Gender

The first independent variable studied is gender; once the data is loaded and cleaned, its relationship to income during the years 2008-2014 will be analyzed. Note that the imported columns include CASEID, year, race, and sex. Also note that gender is represented by the ‘sex’ variable in this dataset.

```
In [239]: physical_data = pd.read_csv('./data/physical_data_nlsy79.csv',
                                      usecols=['CASEID','year','race','sex'])
```

```
In [240]: physical_data.head()
```

Out[240]:

	CASEID	year	race	sex
0	1	1981	NBNH	female
1	2	1981	NBNH	female
2	3	1981	NBNH	female
3	4	1981	NBNH	female
4	5	1981	NBNH	male

Truncation of Gender Data

Then we truncate the data to view only the decade from 2008-2014 and view it.

```
In [241]: # Truncate physical_data so that only years 2008 - 2014 are retained and show unique values
          physical_data_restrict = physical_data[physical_data['year'] >= 2008]
          physical_data_restrict['year'].unique()
```

```
Out[241]: array([2008, 2010, 2012, 2014])
```

Distribution of Gender Data

There are two values for gender: female and male. There are no missing gender values. There are 480 more male data points than female in each year under consideration. We will keep this in mind when analyzing income by gender.

```
In [242]: physical_data_restrict.groupby('sex')['CASEID'].count()
```

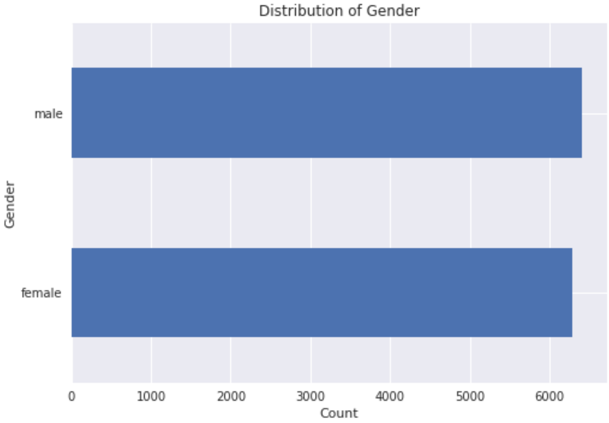
```
Out[242]: sex
          female    25132
          male      25612
          Name: CASEID, dtype: int64
```

In [243]:

```
# Create unique value plot of gender data grouped by sex
df = physical_data_restrict[['CASEID','sex']].groupby('sex').CASEID.nunique().plot(kind='barh')

# Title and labels
df.set_title('Distribution of Gender')
df.set_ylabel('Gender')
df.set_xlabel('Count')
```

Out[243]:

```
<matplotlib.text.Text at 0x7fe6061b1cf8>
```



Note that there are no null values to remove.

In [244]:

```
sum(physical_data_restrict['sex'].isnull())
```

Out[244]:

```
0
```

Race

Race is available in the physical_data_nlsy79 dataset previously loaded.

Truncation of Race Data

Note that the dataset has already been limited to years 2008-2014 for the Gender variable truncation above.

Distribution of Race Data

There are three possible values for race: Black, Hispanic, and NBNH (not black or Hispanic). There are no missing values, though the proportion of non-black/non-Hispanic subjects is much greater than that of black or Hispanic subjects. Again, we will keep this proportion in mind when performing our analyses.

In [245]:

```
# unique race groups
physical_data_restrict['race'].unique()
```

Out[245]:

```
array(['NBNH', 'hispanic', 'black'], dtype=object)
```

In [246]:

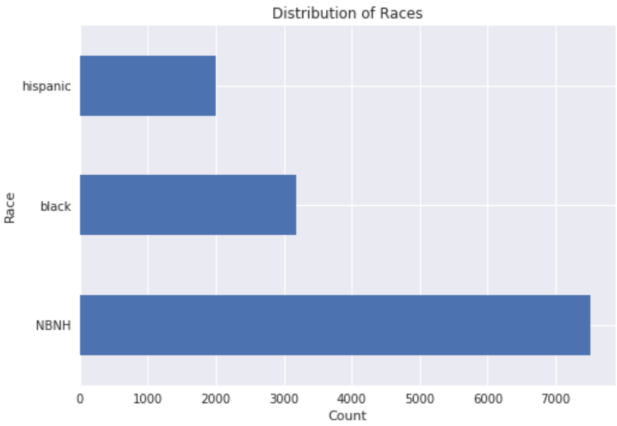
```
# count of unique race groups
physical_data_restrict.groupby('race')['CASEID'].count()
```

Out[246]:

```
race
NBNH        30040
black       12696
hispanic     8008
Name: CASEID, dtype: int64
```

```
In [247]: # Create unique value plot of race data grouped by race
          df = physical_data_restrict[['CASEID','race']].groupby('race').CASEID.nunique().plot(kind='barh')

          # Title and labels
          df.set_title('Distribution of Races')
          df.set_ylabel('Race')
          df.set_xlabel('Count')
```

```
Out[247]: <matplotlib.text.Text at 0x7fe60618d5f8>
```



Now that the physical data has been restricted, it should be noted there are no null values in the dataframe.

```
In [248]: sum(physical_data_restrict['race'].isnull())
```

```
Out[248]: 0
```

Education

The variable that is reported here as education is the one on the survey described as “highest grade completed as of May 1 of survey year.” Therefore, this is a cumulative accounting of the achieved level of education.

Truncation of Education Data

First, we load the data -

```
In [249]: education_data = pd.read_csv('./data/education_data_nlsy79.csv',
                                       usecols=['CASEID','education','year'])
```

```
In [250]: education_data.head()
```

Out[250]:

	CASEID	education	year
0	1	12.00	1979
1	2	9.00	1979
2	3	10.00	1979
3	4	9.00	1979
4	5	13.00	1979

then limit the education records to those from 2008 onward:

```
In [251]: # Truncate education_data so that only years 2008 - 2014 are retained and show unique values
          education_data_restrict = education_data[education_data['year'] >= 2008]
          education_data_restrict['year'].unique()

Out[251]: array([2008, 2010, 2012, 2014])
```

Distribution of Education Data

Education is coded as a numeric value ranging from 0 to 20 inclusive (code below), with an additional possible value of 95 (explained below). A high school degree is 12 years of education, an associate college degree is 14 years, a bachelor degree is 16 years, and postgraduate education of 4 years or more is 20.

```
In [252]: education_data_restrict['education'].sort_values().unique()

Out[252]: array([  1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,  11.,
                  12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  95.,  nan])
```

The value of 95 corresponds to an “ungraded” education level, so we will replace ‘95’ values with NA then remove the NA values.

```
In [253]: # Replace education values of 95 with NA since 95 corresponds to "ungraded" and check unique values
          education_data_restrict = education_data_restrict.replace({95:np.NaN})
          education_data_restrict['education'].sort_values().unique()

Out[253]: array([  1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,  11.,
                  12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  nan])
```

```
In [254]: # Remove NaN 'education' values
          education_data_restrict = education_data_restrict.dropna(subset=['education'])

          # To confirm removal
          sum(physical_data_restrict['sex'].isnull())

Out[254]: 0
```

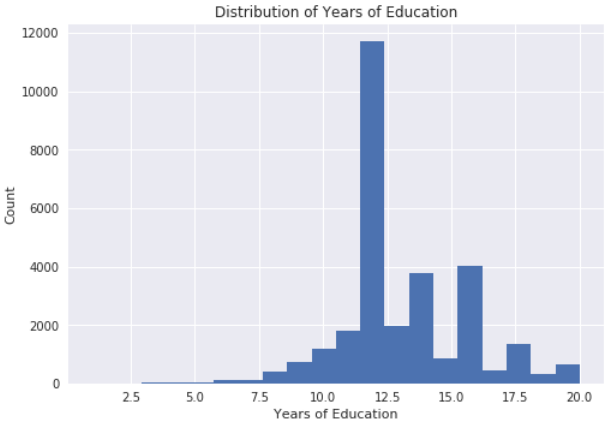
This plot will show there is a significant amount of respondents who have 12 years of education, signifying a high school education.

In [255]:

```
# Plot distribution of years of education
figure, axes = plt.subplots()
axes.hist(education_data_restrict['education'], bins=20)

# Titles and labels added
axes.set_title('Distribution of Years of Education')
axes.set_xlabel('Years of Education')
axes.set_ylabel('Count')
```

Out[255]:

```
<matplotlib.text.Text at 0x7fe6060479b0>
```



EDA: Factors Affecting Income

Joining the Data into a New Dataset

By joining the income, education, and physical data sets to produce a single data set, we have one set of data to reference going forward. The dataset will be called consolidated_data, and the records for which income is NA are excluded since income is the focus of the analysis.

In [256]:

```
# Create consolidated dataset for use in analysis
# Join income & education datasets
consolidated_data_part1 = pd.merge(income_data_restrict, education_data_restrict,
                                  how = 'inner')
# Join Part 1 with physical dataset to complete the merging of data
consolidated_data_part2 = pd.merge(consolidated_data_part1, physical_data_restrict,
                                  how = 'inner')
# Remove NaN from income data
consolidated_data = consolidated_data_part2.dropna(subset=['income'])

# Review the results
consolidated_data.head()
```

Out[256]:

	CASEID	income	year	education	race	sex
0	2	5,000.00	2008	13.00	NBNH	female
1	3	30,000.00	2008	12.00	NBNH	female
2	6	86,000.00	2008	16.00	NBNH	male
3	7	32,500.00	2008	12.00	NBNH	male
4	8	41,000.00	2008	14.00	NBNH	female

In [257]:

```
# Note there are an equal amount of non-null values (28366) in all 6 columns
consolidated_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28366 entries, 0 to 29624
Data columns (total 6 columns):
CASEID       28366 non-null object
income       28366 non-null float64
year         28366 non-null object
education    28366 non-null float64
race         28366 non-null object
sex          28366 non-null object
dtypes: float64(2), object(4)
memory usage: 1.5+ MB
```

Effect of Education on Income

For the first part of the analysis, we will review the effect of the independent variable education on income.

Mean and Median Income by Years of Education

First we will create income_by_education by grouping income data by year and years of education. Then we create a dataframe of the grouping and add the mean income of that data, calling the new dataframe income_by_education_mean. The same is done for the median income of the data, calling that dataframe income_by_education_median.

In [258]:

```
# Group consolidated_data by year then by education years
income_by_education = consolidated_data.groupby(['year', 'education'])

# Add mean income for year & education years to groupby
income_by_education_mean = income_by_education[['income']].mean()

# Rename columns and rearrange for easy viewing
income_by_education_mean.rename(columns={'income': 'mean income'}, inplace=True)
income_by_education_mean.unstack(level=0)
```

Out[258]:

	mean income			
year	**2008**	**2010**	**2012**	**2014**
education				
1.00	0.00	25,000.00	18,000.00	nan
2.00	9,333.33	10,000.00	22,550.00	43,333.33
3.00	12,514.40	14,048.92	5,000.00	5,375.00
4.00	34,081.82	66,813.78	26,394.00	20,655.42
5.00	15,957.14	11,650.00	12,833.33	4,195.00
6.00	11,533.91	11,308.57	9,317.86	11,194.64
7.00	13,630.93	12,917.91	10,406.25	4,389.71
8.00	12,919.07	11,576.71	8,297.71	14,909.39
9.00	16,842.42	15,869.69	15,617.94	13,807.92
10.00	15,963.67	15,330.79	15,409.27	16,601.63
11.00	18,710.81	17,992.54	16,887.27	17,258.41
12.00	31,133.84	29,108.40	30,085.96	30,027.54
13.00	32,561.47	32,713.20	32,314.72	35,096.93
14.00	41,327.90	39,390.55	40,663.53	40,814.96
15.00	39,910.28	41,324.70	43,843.14	41,090.62
16.00	67,749.32	67,858.44	72,579.91	75,685.17
17.00	60,070.62	61,272.54	62,129.72	68,220.42
18.00	75,039.57	77,941.61	82,235.32	85,562.75
19.00	90,583.42	85,681.13	94,379.43	112,548.16
20.00	109,877.65	124,386.67	121,833.84	111,111.80

In [259]:

```
# Add median income for year & education years to groupby
income_by_education_median = income_by_education[['income']].median()

# Rename columns and rearrange for easy viewing
income_by_education_median.rename(columns={'income':'median income'}, inplace=True)
income_by_education_median.unstack(level=0)
```

Out[259]:

	median income			
year	**2008**	**2010**	**2012**	**2014**
education				
1.00	0.00	25,000.00	18,000.00	nan
2.00	0.00	10,000.00	33,000.00	42,000.00
3.00	14,000.00	12,000.00	0.00	0.00
4.00	20,000.00	22,000.00	23,250.00	15,500.00
5.00	3,300.00	6,000.00	8,000.00	0.00
6.00	7,000.00	6,500.00	0.00	3,300.00
7.00	12,000.00	1,200.00	0.00	0.00
8.00	250.00	0.00	0.00	0.00
9.00	10,000.00	6,500.00	3,320.00	0.00
10.00	9,300.00	6,000.00	300.00	3,930.00
11.00	12,000.00	9,346.50	5,000.00	2,000.00
12.00	27,000.00	25,000.00	25,000.00	24,000.00
13.00	28,000.00	27,000.00	30,000.00	24,000.00
14.00	37,000.00	35,000.00	34,000.00	32,000.00
15.00	30,000.00	34,000.00	40,000.00	32,500.00
16.00	50,000.00	50,395.50	53,500.00	55,000.00
17.00	51,500.00	47,000.00	56,000.00	54,042.50
18.00	59,000.00	60,000.00	65,000.00	65,000.00
19.00	61,000.00	64,000.00	63,868.00	70,000.00
20.00	77,000.00	90,000.00	80,000.00	68,000.00

Now we can combine the two sets of data to plot them for analysis.

In [260]:

```
# Combine the data via concatenation
income_by_education_combined = pd.concat([income_by_education_mean,
                                          income_by_education_median],
                                        axis = 1)

# Quick view of new dataset
income_by_education_combined.head()
```

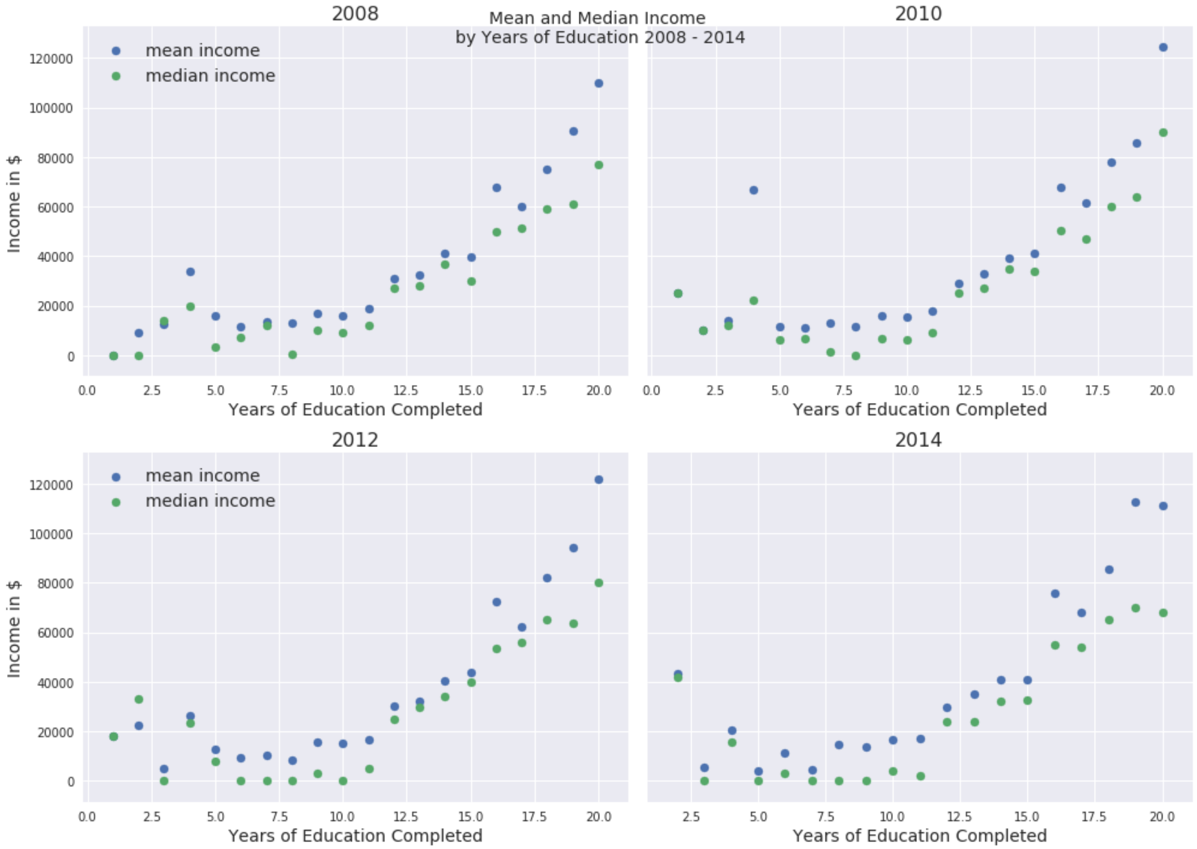
Out[260]:

		mean income	median income
year	**education**		
2008	**1.00**	0.00	0.00
	2.00	9,333.33	0.00
	3.00	12,514.40	14,000.00
	4.00	34,081.82	20,000.00
	5.00	15,957.14	3,300.00

The plot below shows the mean and median income data in dollars by years of education. The mean income is higher than the median (as expected for our right-skewed data) for each of the 4 years studied.

In [261]:

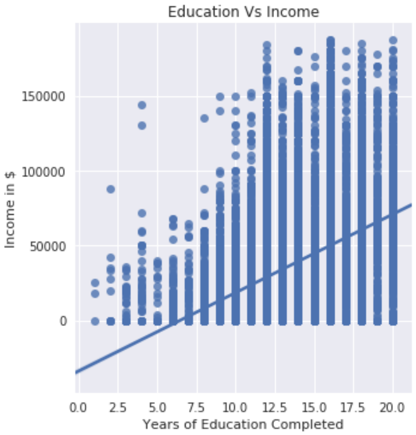
```
# Plot mean and median income by years of education for each year surveyed in subplots with shared y access
figure, axes = plt.subplots(2, 2, figsize=(14,10), sharey=True)
df = income_by_education_combined.reset_index()

# Label the graph with a title
figure.suptitle("Mean and Median Income \nby Years of Education 2008 - 2014", fontsize=14)

# Year 2008 subset data and scatterplot
df_2008 = df[df['year']==2008]
axes[0, 0].scatter(df_2008['education'], df_2008['mean income'])
axes[0, 0].scatter(df_2008['education'], df_2008['median income'])

# Year 2008 titles, legend, labels added
axes[0, 0].set_title("2008", fontsize=16)
axes[0, 0].legend(fontsize=14)
axes[0, 0].set_ylabel("Income in $", fontsize=14)
axes[0, 0].set_xlabel("Years of Education Completed", fontsize=14)

# Year 2010 subset data and scatterplot
df_2010 = df[df['year']==2010]
axes[0, 1].scatter(df_2010['education'], df_2010['mean income'])
axes[0, 1].scatter(df_2010['education'], df_2010['median income'])

# Year 2010 titles, x label added
axes[0, 1].set_title("2010", fontsize=16)
axes[0, 1].set_xlabel("Years of Education Completed", fontsize=14)

# Year 2012 subset data and scatterplot
df_2012 = df[df['year']==2012]
axes[1, 0].scatter(df_2012['education'], df_2012['mean income'])
axes[1, 0].scatter(df_2012['education'], df_2012['median income'])

# Year 2012 titles, legend, labels added
axes[1, 0].set_title("2012", fontsize=16)
axes[1, 0].legend(fontsize=14)
axes[1, 0].set_ylabel("Income in $", fontsize=14)
axes[1, 0].set_xlabel("Years of Education Completed", fontsize=14)

# Year 2014 subset data and scatterplot
df_2014 = df[df['year']==2014]
axes[1, 1].scatter(df_2014['education'], df_2014['mean income'])
axes[1, 1].scatter(df_2014['education'], df_2014['median income'])

# Year 2014 titles, x label added
axes[1, 1].set_title("2014", fontsize=16)
axes[1, 1].set_xlabel("Years of Education Completed", fontsize=14)

# Automatically spaces subplots
plt.tight_layout()
```



The following plot shows the effect of education on income. Here we have chosen to exclude the top 2% data; by choosing to only include incomes less than $300000, we ensure to incorporate the data for all 4 years (see explanation in income analysis above). With the trend line, it is clear that there is a positive correlation between income and education.

In [262]:

```
# Education vs Income Scatter Plot, top 2% data excluded
educ_vs_income = sns.lmplot(x='education', y="income", data=consolidated_data[(consolidated_data['income'] <300000)])

# Title and labels added
plt.title("Education Vs Income");
plt.xlabel("Years of Education Completed");
plt.ylabel("Income in $");
```



Effect of Gender on Income

For the second part of the analysis, we will review the effect of the independent variable gender on income.

Mean Income by Gender

First we will create income_by_gender data with a created field for mean income in the same way as above with income_by_education. The data is grouped by year and gender then we add a column for mean income; it seems to demonstrate relatively no change year over year for female respondents versus more fluctuations year over year for male repsondents.

In [263]:

```
# Group consolidated_data by year then gender
income_by_gender = consolidated_data.groupby(['year', 'sex'])

# Add mean income for year & gender to groupby
income_by_gender_mean = income_by_gender[['income']].mean()

# Rename columns and rearrange for easy viewing
income_by_gender_mean.rename(columns={'income':'mean income'}, inplace=True)
income_by_gender_mean.unstack(level=0)
```

Out[263]:

	mean income			
year	**2008**	**2010**	**2012**	**2014**
sex				
female	29,547.44	29,832.22	29,565.55	29,487.16
male	51,306.43	50,054.71	53,503.74	56,170.20

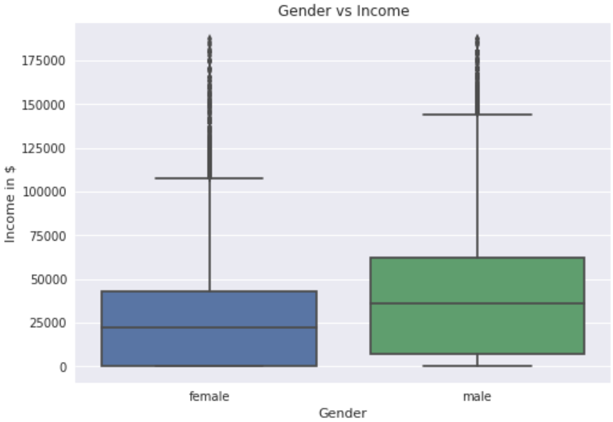
The plot below compares median and quartile income values for men and women. The median income for men is higher than the median income for women, and the other quartiles are similarly offset. Note the data does not include the top 2% of incomes by limiting the consolidated_data dataframe to less than $300000 incomes.

In [264]:

```
#Income by Gender without top 2% of incomes
df = consolidated_data[consolidated_data['income']<300000]
gender_income = sns.boxplot(df.sex,df.income)

# Title and labels added
gender_income.set_title('Gender vs Income')
gender_income.set_xlabel('Gender')
gender_income.set_ylabel('Income in $')
```

Out[264]:

```
<matplotlib.text.Text at 0x7fe60dd52860>
```



Additionally, we see that a higher percentage of men earn incomes higher than the overall mean.

In [265]:

```
# Percentages of Males and Females that earn over the mean income, grouped, and aggregated
group_1 = consolidated_data[consolidated_data['income']>
                            consolidated_data.income.mean()].groupby('sex').agg({'CASEID':'nunique'})
group_2 = consolidated_data.groupby('sex').agg({'CASEID':'nunique'})

# Columns renamed for clarity
group_1.rename(columns={'CASEID':'% of gender group having income > mean'}, inplace=True)
group_2.rename(columns={'CASEID':'% of gender group having income > mean'}, inplace=True)

# Data percentage calculated
group_1 / group_2 * 100
```

Out[265]:

	% of gender group having income > mean
sex	
female	36.69
male	57.85

Effect of Race on Income

For the third part of the analysis, we will review the effect of the independent variable race on income.

Mean Income by Race

First we will create income_by_race data with a created field for mean income in the same way as above with income_by_education and income_by_gender. This data shows the large disparity between black, hispanic, and non black non hispanic respondent groups.

In [266]:

```python
# Group consolidated_data by year then race
income_by_race = consolidated_data.groupby(['year', 'race'])

# Add mean income for year & race to groupby
income_by_race_mean = income_by_race[['income']].mean()

# Rename columns and rearrange for easy viewing
income_by_race_mean.rename(columns={'income':'mean income'}, inplace=True)
income_by_race_mean.unstack(level=0)
```

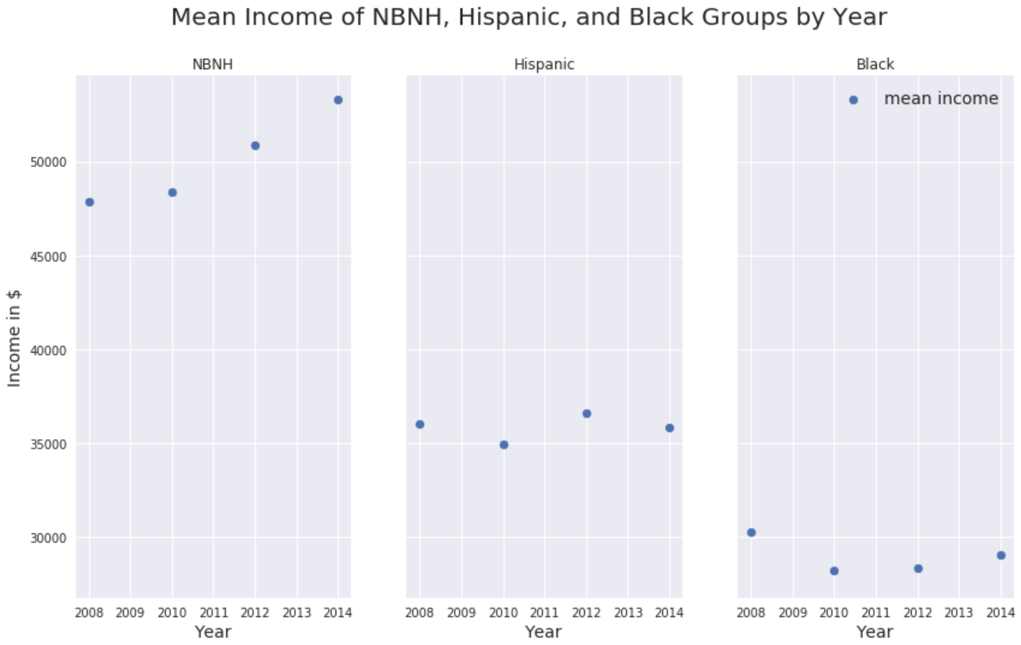
Out[266]:

	mean income			
year	**2008**	**2010**	**2012**	**2014**
race				
NBNH	47,877.29	48,359.17	50,876.36	53,318.91
black	30,268.35	28,186.68	28,364.36	29,026.54
hispanic	36,038.13	34,935.00	36,608.16	35,840.37

The graph below demonstrates the disparity visually, with the NBNH race group making significantly higher mean incomes each year than black and hispanic groups, as well as seeing an increase in mean income while the other groups see stagnant or negative growth.

In [267]:
```
# Plot mean income by race for each year surveyed: 2008, 2010, 2012, and 2014
figure, axes = plt.subplots(1, 3, figsize=(14,8), sharey=True)
df = income_by_race_mean.reset_index()

# Subset dataframes for each race
df_nbnh = df[df['race']=='NBNH']
df_hispanic = df[df['race']=='hispanic']
df_black = df[df['race']=='black']

# Create scatterplots
axes[0].scatter(df_nbnh['year'], df_nbnh['mean income'])
axes[1].scatter(df_hispanic['year'], df_hispanic['mean income'])
axes[2].scatter(df_black['year'], df_black['mean income'])

# Set titles
axes[0].set_title("NBNH")
axes[1].set_title("Hispanic")
axes[2].set_title("Black")

# Set labels and legend
axes[0].set_ylabel("Income in $", fontsize=14)
axes[0].set_xlabel("Year", fontsize=14)
axes[1].set_xlabel("Year", fontsize=14)
axes[2].set_xlabel("Year", fontsize=14)
axes[2].legend(fontsize=14)

# Label the graph with a title
figure.suptitle("Mean Income of NBNH, Hispanic, and Black Groups by Year", fontsize=20)
```

Out[267]:
```
<matplotlib.text.Text at 0x7fe60c34aba8>
```



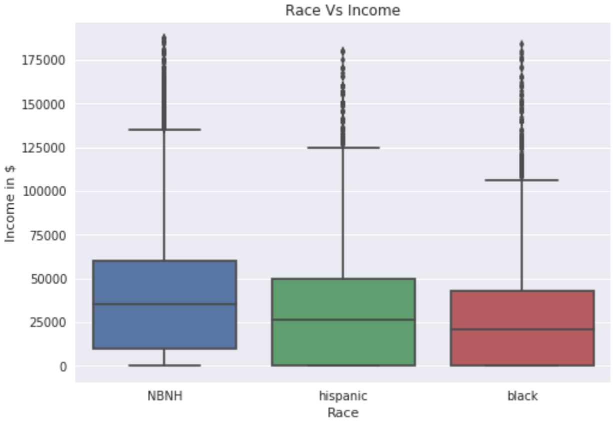
The plot below compares median and quartile income values for each race group. The median income for the non-black, non-Hispanic group is higher than the mean income for the Hispanic and black groups. Note the top 2% of data has been excluded as outliers.

In [268]:

```
# Dataframe set and outliers excluded
df = consolidated_data
race_income = sns.boxplot(df[ df['income']< 200000].race,df[ df['income']< 200000].income)

# Titles and labels
race_income.set_title('Race Vs Income')
race_income.set_xlabel('Race')
race_income.set_ylabel('Income in $')
```

Out[268]:

```
<matplotlib.text.Text at 0x7fe6055e42e8>
```



Additionally, we see that the percentage of non-black, non-Hispanic individuals with incomes higher than the overall mean is greater than the percentage for the Hispanic and black groups, respectively.

In [269]:

```
# Percentage of individuals in each race group with income > overall mean income, grouped, and aggregated
group_1 = consolidated_data[consolidated_data['income']>
                            consolidated_data.income.mean()].groupby('race').agg({'CASEID':'nunique'})
group_2 = consolidated_data.groupby('race').agg({'CASEID':'nunique'})

# Columns renamed for clarity
group_1.rename(columns={'CASEID':'% of race group having income > mean'}, inplace=True)
group_2.rename(columns={'CASEID':'% of race group having income > mean'}, inplace=True)

# Data percentage calculated
group_1 / group_2 * 100
```

Out[269]:

	% of race group having income > mean
race	
NBNH	54.85
black	37.20
hispanic	43.20

Hypotheses for Further Analysis

Based on the plots and the grouped summary analyses we state the following hypotheses:

1. Mean income is positively correlated with years of education completed.
2. Mean income shows a gender-based disparity, with men having a higher mean income than women.
3. Mean income shows racial disparity, with NBNH respondents having the highest (and majority) mean income, followed by Hispanics, followed by Blacks.
4. Though the overall respondent mean income rose, only the NBNH segment had consistent mean income growth during the 2008-2014 time period.

Testing the statistical significance of these hypotheses requires statistical analyses to be learned later. Our team would also be interested to analyze other factors such as profession and industry of the respondents for greater insight into the data.

Further Data Points for Analysis

During this analysis, it came to our attention that more detail from the respondents on the following topics would create clearer hypotheses:

- Type of college and graduate degrees (law, medical, humanities, etc.)
- GED versus 18-year-old high school graduate for the 12 year educated respondents
- Type of location (urban, suburban, versus country)
- Physical location (city, state)
- Industry (technology, pharmaceuticals, etc.)