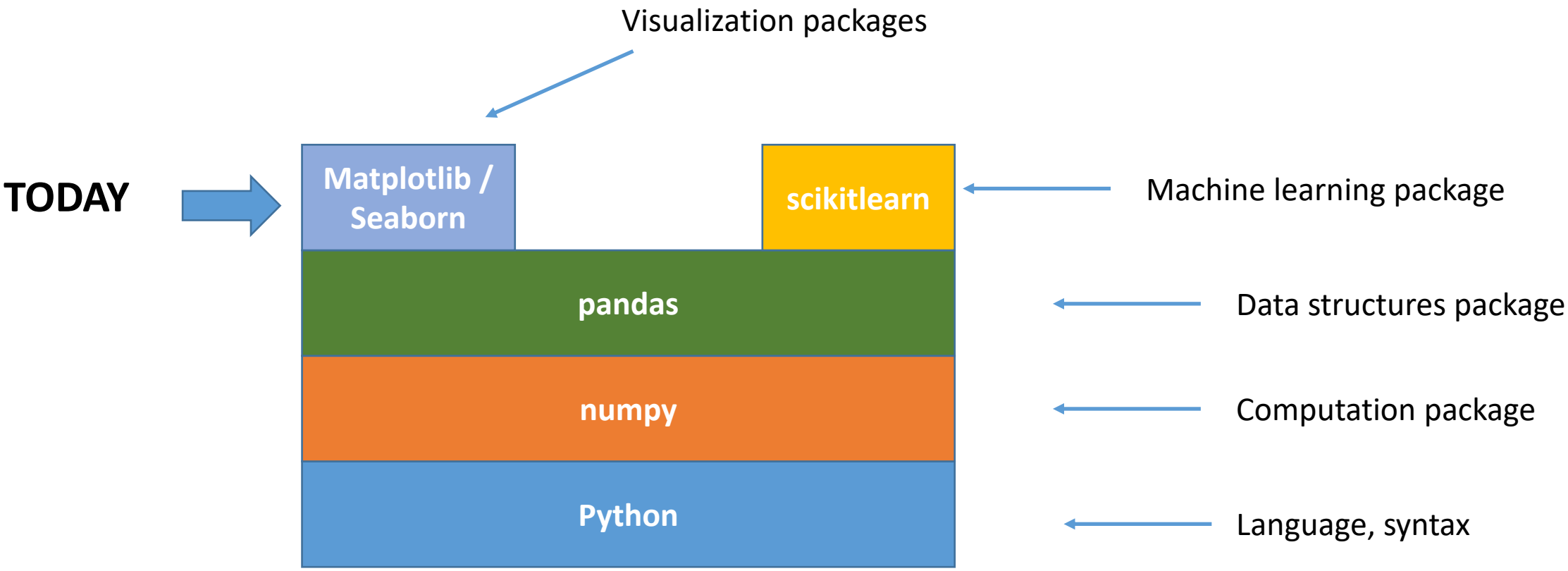
# Visualization - Seaborn

module 9

# Announcement

- **E-mail me your top 3 dataset candidates for your Final Project**
  - **URLs**
  - **High level description on data set**
  - **High level ideas what type of unexpected/interesting things you want to look for**
- **Final Exam:**
  - **5/11 Sat.**
  - **Lucas 309**

# This course


Visualization packages
TODAY
Matplotlib / Seaborn
scikitlearn
Machine learning package
pandas
Data structures package
numpy
Computation package
Python
Language, syntax

# Today's data set

- https://www.kaggle.com/uciml/adult-census-income
- This data was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). *The prediction task is to determine whether a person makes over $50K a year*.

# Types of variables

- **Numeric or continuous**: (e.g., age) a variable with many (sometimes infinite) possible numeric values

- **Categorical or nominal**: (e.g., sex, race) a variable with a finite set of values. There is no intrinsic order between them (sexes or races cannot be sorted from smaller to larger)

- **Ordinal**: a variable with a finite set of values that can be sorted. Example in this data set: income (“< 50k”, “>= 50k”)

# Discretization

- We can turn a numeric variable into an ordinal one through two functions.
  - For **equal-width binning**, use **pandas.cut**. All bins will be of the same width.
  - For **equal-frequency binning**, use **pandas.qcut**. All bins will (attempt to) have the same number of records.

## EQUAL-WIDTH BINNING: CUT

```
df2 = df.copy()
df2['discretized_age'] = pd.cut(df2.age, 5)
```

Number of bins

Series to discretize

| | age | discr_age |
|---|---|---|
| **9646** | 24 | (16.936, 29.8] |
| **709** | 34 | (29.8, 42.6] |
| **7385** | 18 | (16.936, 29.8] |
| **16671** | 44 | (42.6, 55.4] |
| **21932** | 27 | (16.936, 29.8] |

## EQUAL-FREQUENCY BINNING: QCUT

```
df2 = df.copy()
df2['discretized_age'] = pd.qcut(df2.age, 5)
```

| | age | discretized_age |
|---|---|---|
| **9646** | 24 | [17, 25] |
| **709** | 34 | (32, 42] |
| **7385** | 18 | [17, 25] |
| **16671** | 44 | (42, 51] |
| **21932** | 27 | (25, 32] |

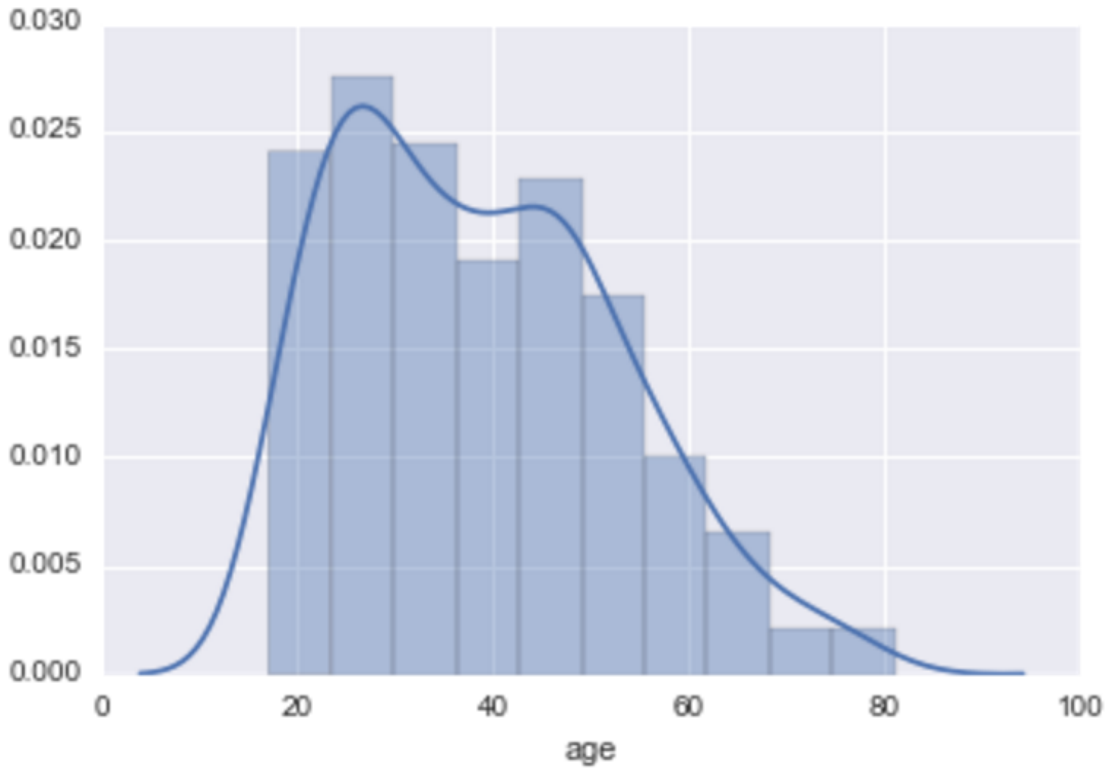
# Visualization by combination and type of variables

# One variable

# One numeric variable

```
sns.distplot(df.age, bins=10)
```

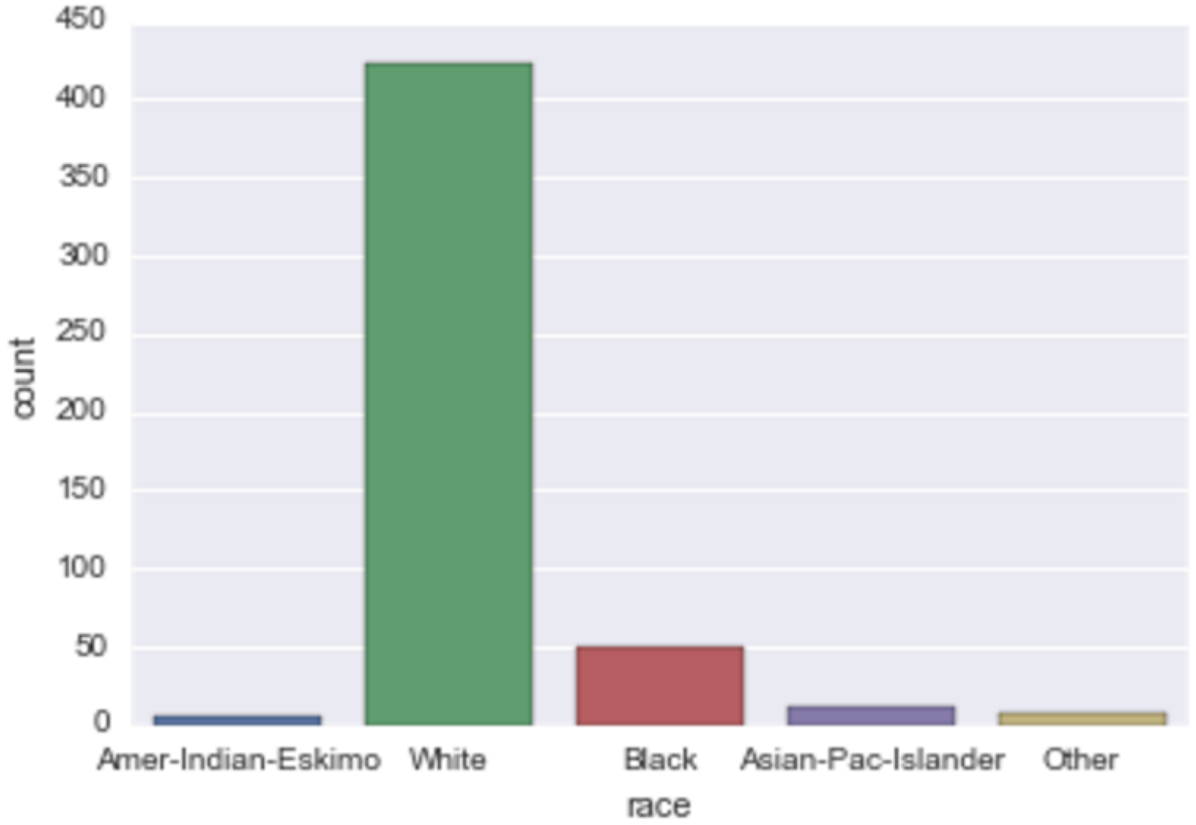
```
<matplotlib.axes._subplots.AxesSubplot at 0x160769e8>
```



# One categorical variable

```
sns.countplot(x='race',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x261843c8>
```



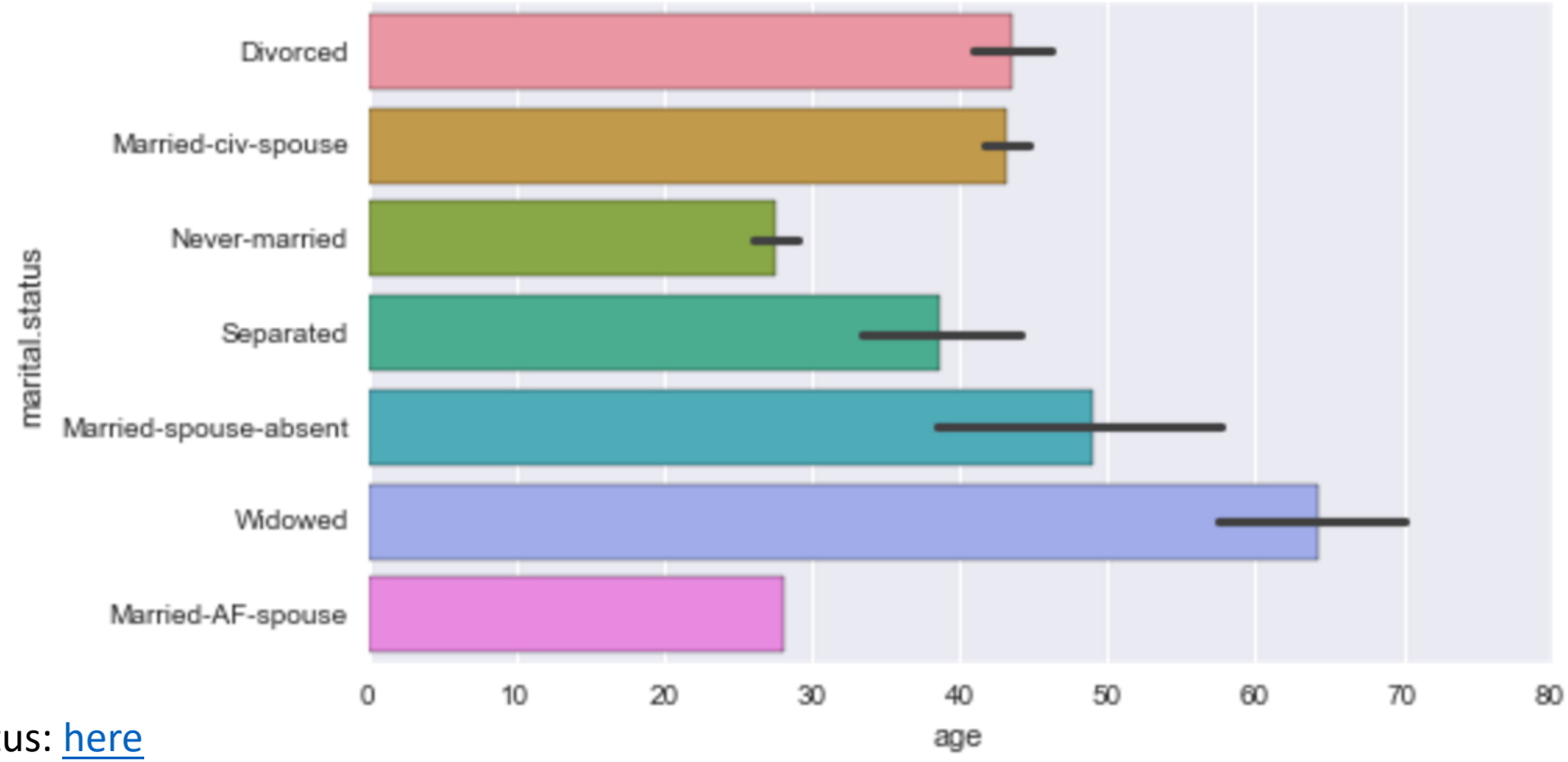
# Two variables

# One categorical and one numeric variables

```
sns.factorplot(x='age',y='marital.status', data=df, kind='bar', aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x205e9208>
```

**Mean of age grouped by marital.status**



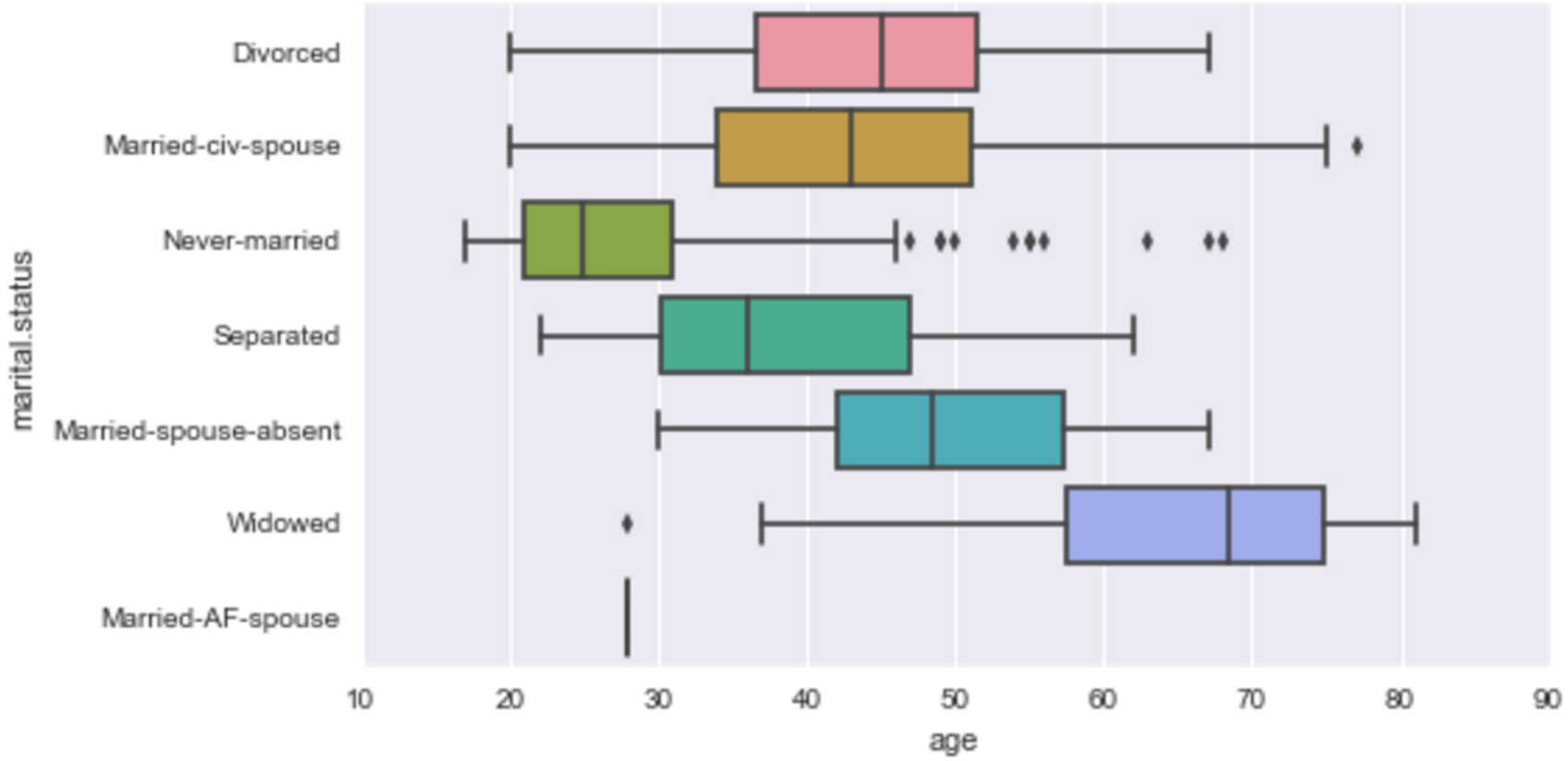

More info on marital.status: [here]

# One categorical and one numeric variables

**Mean of age grouped by marital.status**

```
sns.factorplot(x='age',y='marital.status', data=df, kind='box', aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x1f508b38>
```

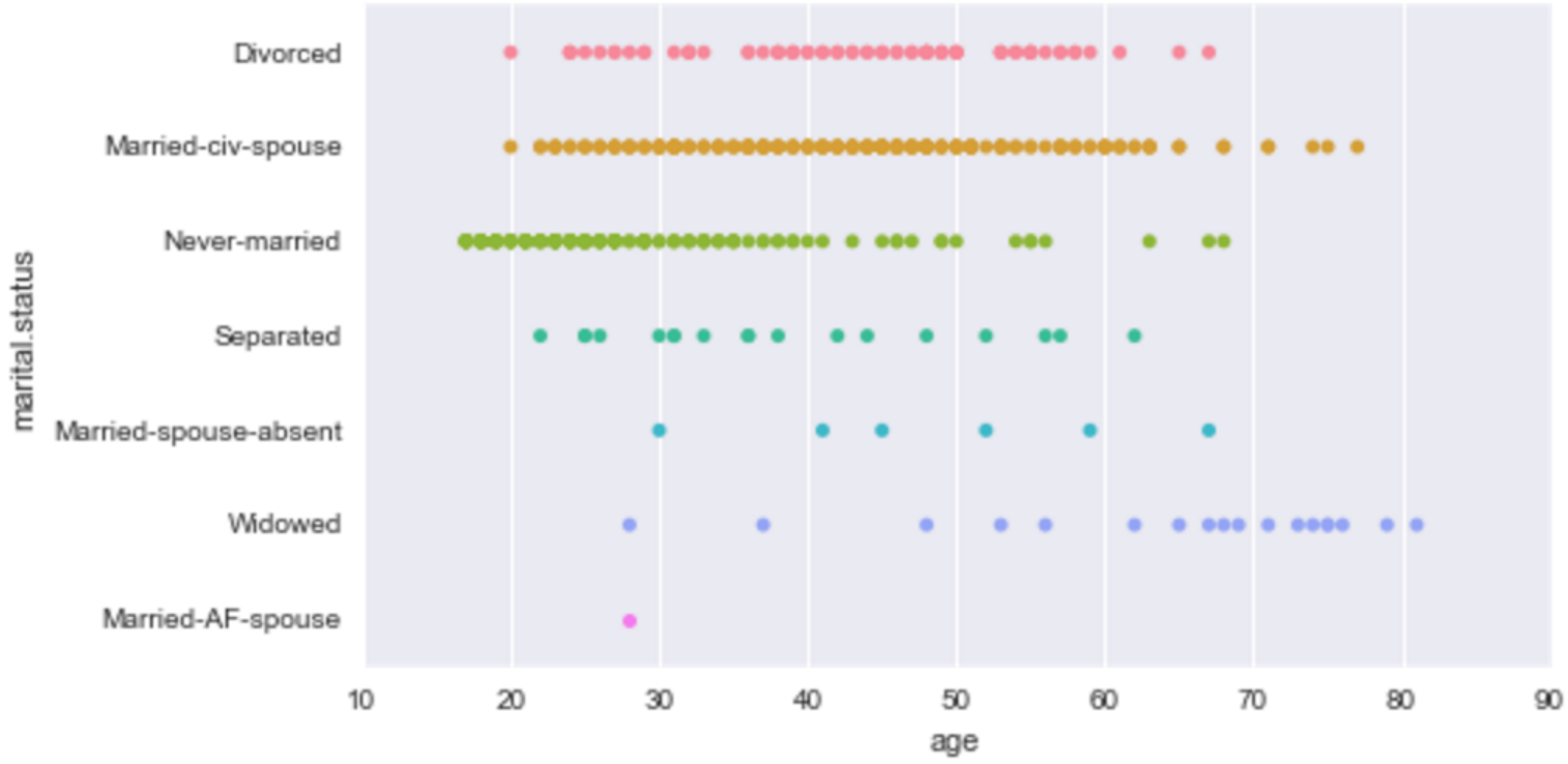


# One categorical and one numeric variables

**Mean of age grouped by [marital.status]**

```
sns.factorplot(x='age',y='marital.status', data=df, kind='strip', aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x210c4320>
```

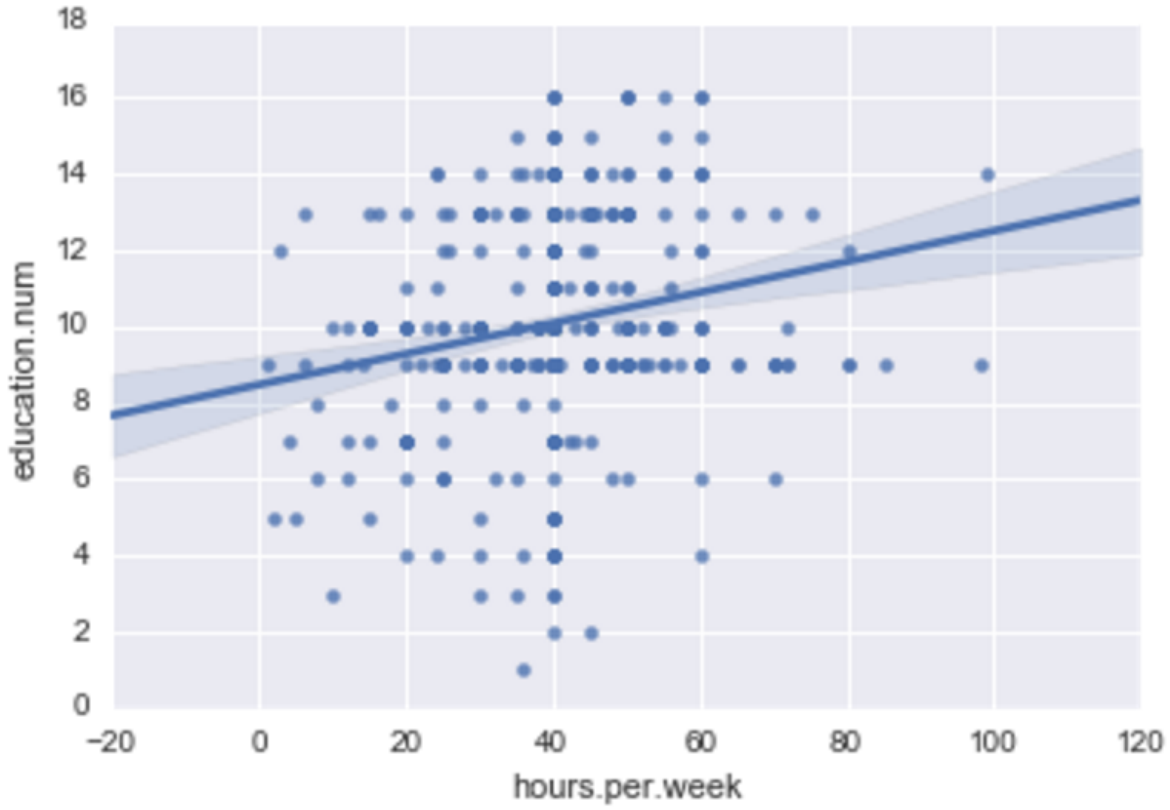


# Two numeric variables: Regression

**Regression**
**Hours.per.week → education.num**

```
sns.regplot(x=df['hours.per.week'], y=df['education.num'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x149d9908>
```



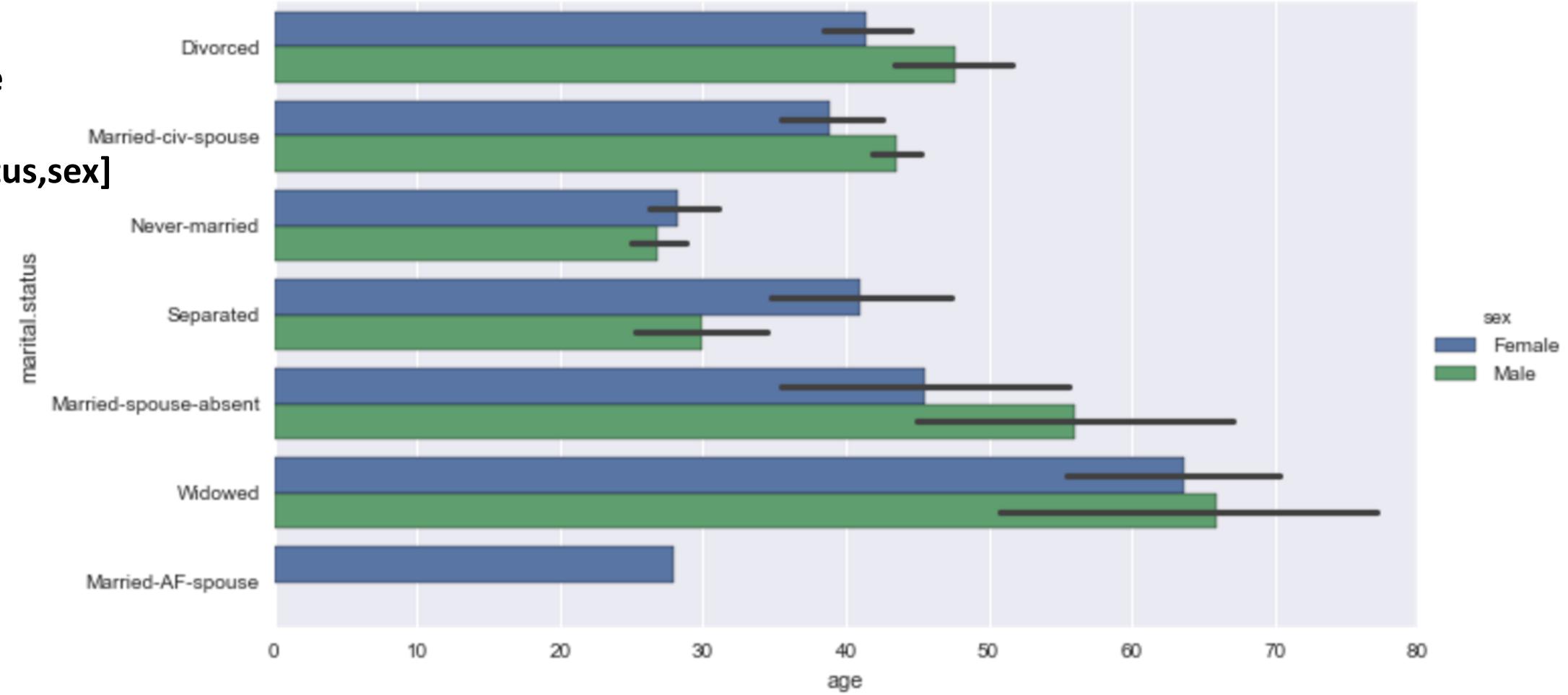
# Three variables

# Two categorical and one numeric variables

```
sns.factorplot(y='marital.status',x='age',data=df,hue='sex',kind='bar', size=5, aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x21685c50>
```

**Mean of age grouped by [marital.status,sex]**

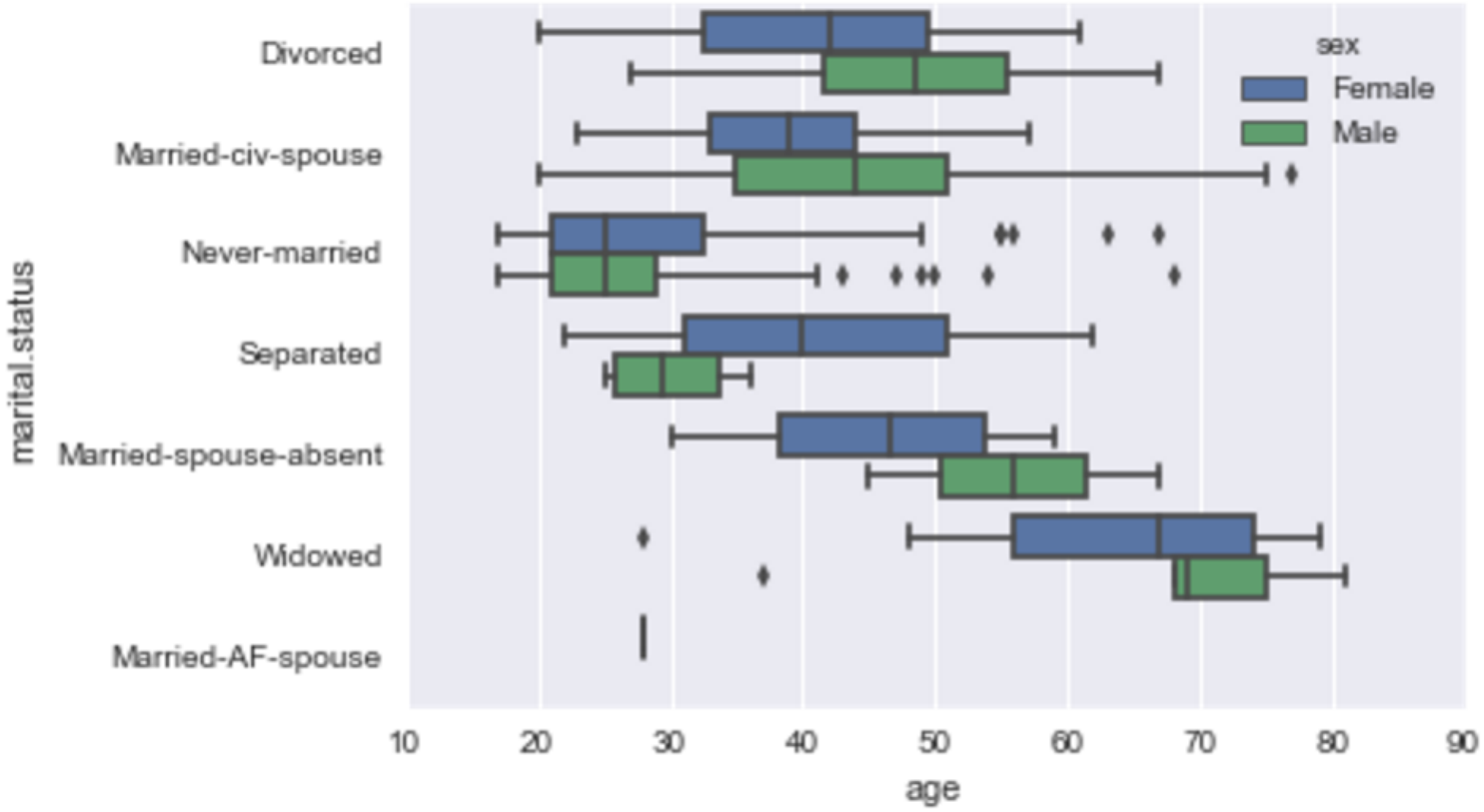


# Two categorical and one numeric variables

**Mean of age grouped by [marital.status,sex]**

```
sns.boxplot(x=df.age,y=df['marital.status'],hue=df.sex)
```

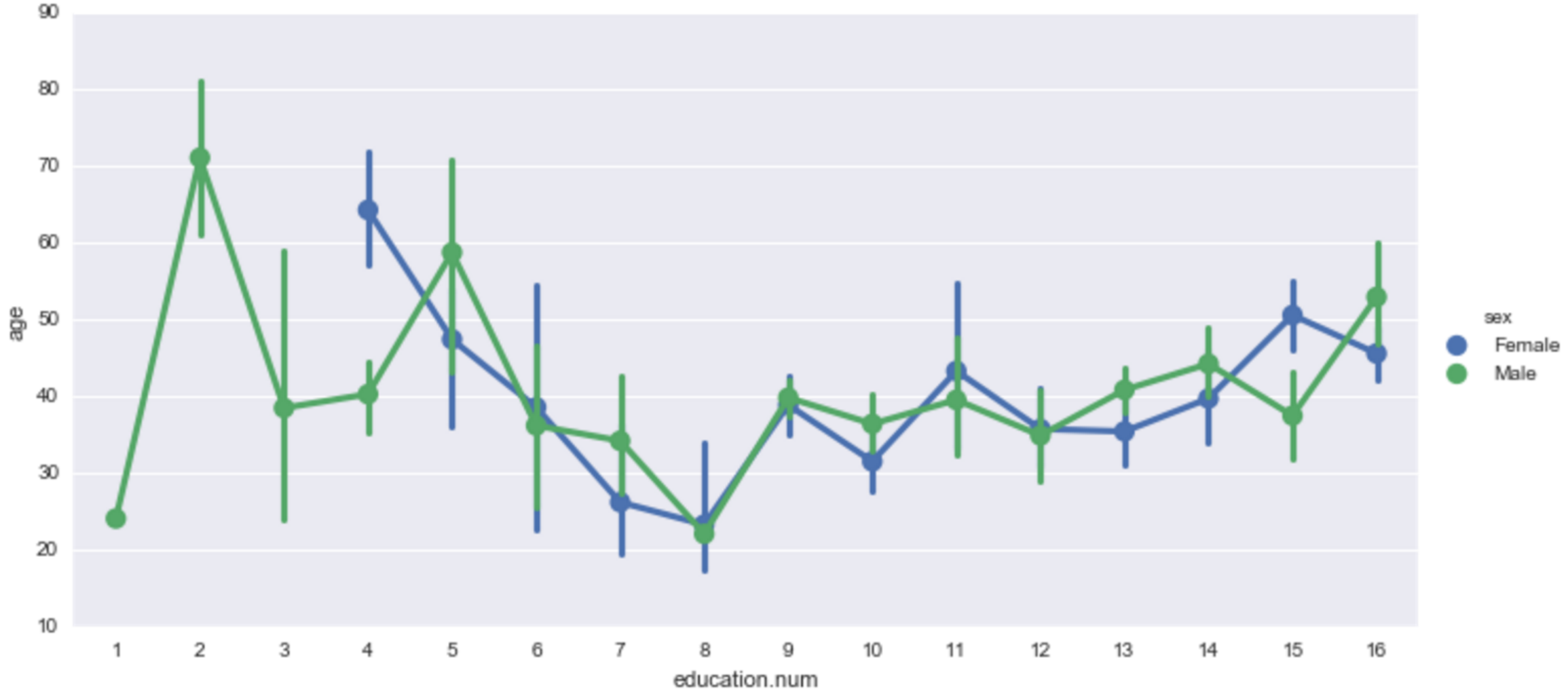
```
<matplotlib.axes._subplots.AxesSubplot at 0xdef7748>
```



# Two numeric and one categorical variables

```
sns.factorplot(x='education.num',y='age',data=df,hue='sex',kind='point',size=5,aspect=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x11325e48>
```

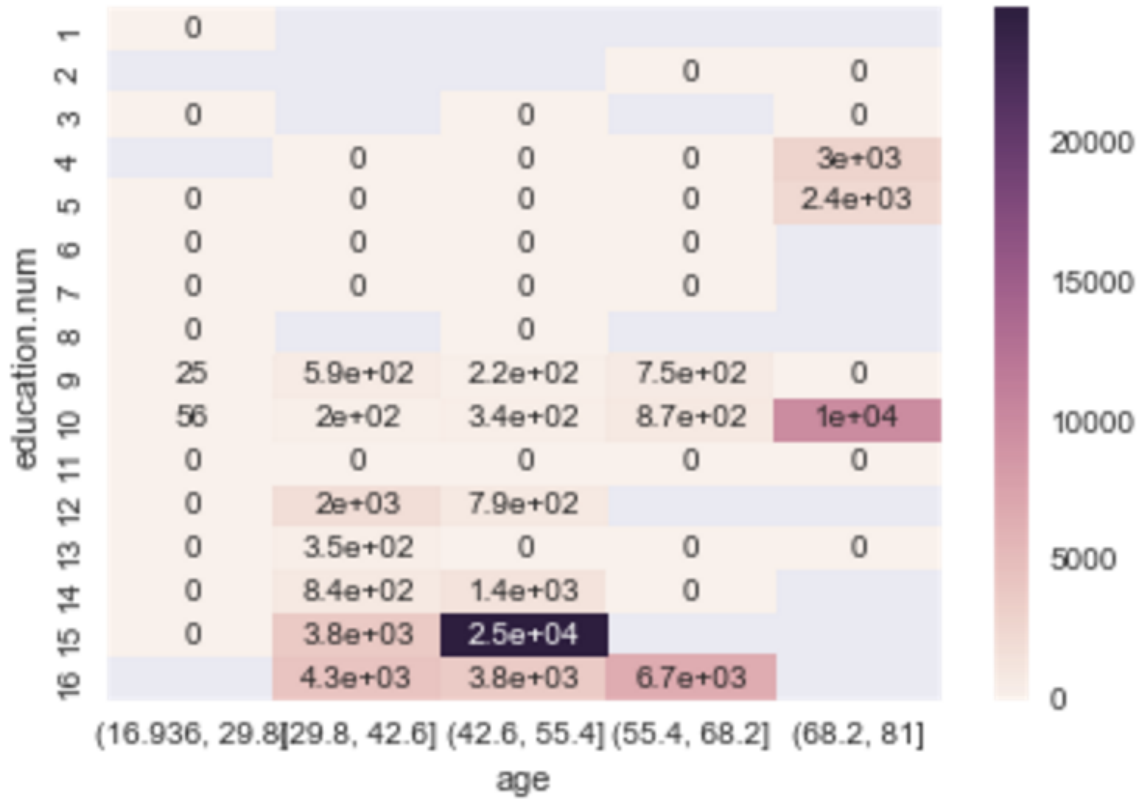


# Two ordinal and one numeric variables

```
df2 = df.copy()
df2['age'] = pd.cut(df2.age, 5)
gr = df2.groupby(['education.num','age'])['capital.gain'].mean()
gr2 = gr.unstack()
sns.heatmap(gr2,annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x23d485c0>
```

**Mean of capital.gain grouped by [discretized age, education.num]**



# Four variables

# Two categorical and two numeric variables

```
sns.factorplot(x='education.num',y='age',data=df,hue='sex',col='income',kind='point',aspect = 1)
```

```
<seaborn.axisgrid.FacetGrid at 0x26942390>
```

# Three categorical and one numeric variables

```
sns.factorplot(x='sex',y='age',data=df,hue='race',col='income',kind='bar')
```

```
<seaborn.axisgrid.FacetGrid at 0x26522e10>
```