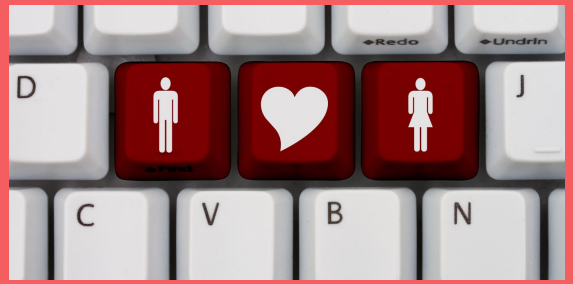
Date_A_Scientist Project
by Zoe Chumakova



Age group

```
df['age'] = df['age'].replace(np.nan, '', regex=True)
#df['income'] = df['income'].replace(-1, '', regex=True)
#print(df['income'])
#income = df['income']
#age = df['age']
```

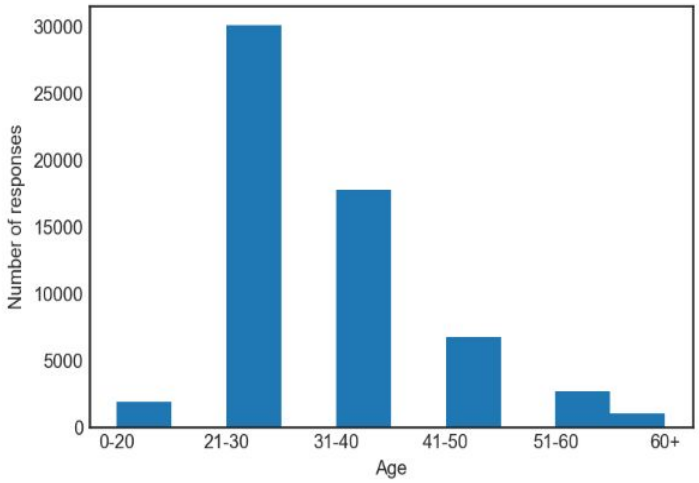
```
df['age_group'] = pd.cut(df.age,[0,20,30,40,50,60, 110], labels = ['0-20', '21-30', '31-40', '41-50', '51-60','60+'])
print(df.age_group.value_counts())
age_group = df['age_group']
```

```
21-30    30017
31-40    17727
41-50     6745
51-60     2618
0-20      1873
60+        966
Name: age_group, dtype: int64
```

Age group

In [80]:
```
plt.hist(age_group, bins=10)
plt.xlabel("Age")
plt.ylabel("Number of responses")
```

Out[80]:
```
Text(0,0.5,'Number of responses')
```



Transformed drinks, drugs, smokes, sex, status into numerical value

In [81]:
```
print(df.drinks.value_counts())
drink_mapping = {"not at all": 0, "rarely": 1, "socially": 2, "often" : 3, "very often":4, "desperately": 5}
df["drinks_code"] = df.drinks.map(drink_mapping)
```

```
socially       41780
rarely          5957
often           5164
not at all      3267
very often       471
desperately      322
Name: drinks, dtype: int64
```

In [82]:
```
print(df.drinks_code.value_counts())
```

```
2.0    41780
1.0     5957
3.0     5164
0.0     3267
4.0      471
5.0      322
Name: drinks_code, dtype: int64
```

In [83]:
```
#same for smokes
print(df.smokes.value_counts())
smokes_mapping = {"no": 0, "sometimes":1, "when drinking": 2, "yes": 3, "trying to quit": 4}
df["smokes_code"] = df.smokes.map(smokes_mapping)
```

Transformed drinks, drugs, smokes, sex, status into numerical value

```
#same for smokes
print(df.smokes.value_counts())
smokes_mapping = {"no": 0, "sometimes":1, "when drinking": 2, "yes": 3, "trying to quit": 4}
df["smokes_code"] = df.smokes.map(smokes_mapping)
```

```
no               43896
sometimes         3787
when drinking     3040
yes               2231
trying to quit    1480
Name: smokes, dtype: int64
```

```
print(df.smokes_code.value_counts())
```

```
0.0    43896
1.0     3787
2.0     3040
3.0     2231
4.0     1480
Name: smokes_code, dtype: int64
```

Transformed drinks, drugs, smokes, sex, status into numerical value

```
#same for drugs
print(df.drugs.value_counts())
drugs_mapping = {"never": 0, "sometimes": 1, "often": 2}
df["drugs_code"] = df.drugs.map(drugs_mapping)
```

```
never        37724
sometimes     7732
often          410
Name: drugs, dtype: int64
```

```
print(df.drugs_code.value_counts())
```

```
0.0    37724
1.0     7732
2.0      410
Name: drugs_code, dtype: int64
```

```
print(df.sex.value_counts())
```

```
m    35829
f    24117
Name: sex, dtype: int64
```

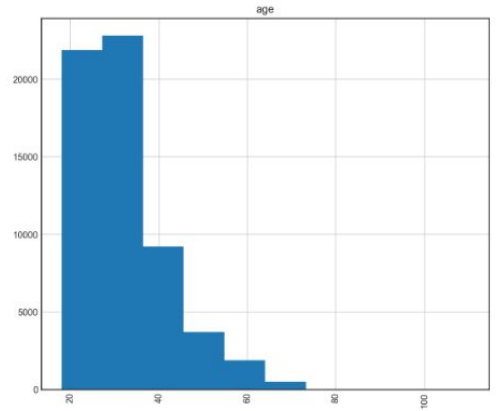
Creating new data set with necessary columns

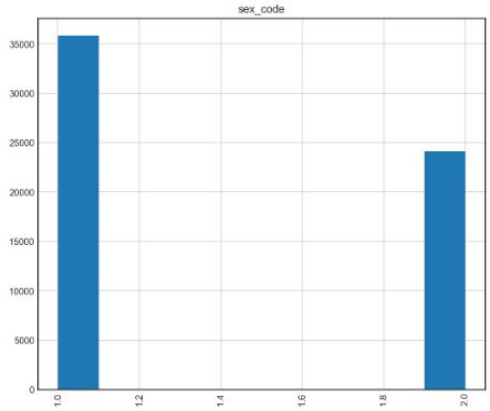
```
import statsmodels as sm
import sklearn as skl
import sklearn.preprocessing as preprocessing
import sklearn.linear_model as linear_model
import sklearn.cross_validation as cross_validation
import sklearn.metrics as metrics
import sklearn.tree as tree
import seaborn as sns

import math

future_data=df[["age", "sex_code", "income", "education", "status", "drinks_code","drugs_code", "smokes_code","job"]]

print(future_data)

fig = plt.figure(figsize=(30,30))
cols = 3
rows = math.ceil(float(future_data.shape[1]) / cols)
for i, column in enumerate(future_data.columns):
    ax = fig.add_subplot(rows, cols, i + 1)
    ax.set_title(column)
    if future_data.dtypes[column] == np.object:
        future_data[column].value_counts().plot(kind="bar", axes=ax)
    else:
        future_data[column].hist(axes=ax)
        plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.5, wspace=0.2)
```

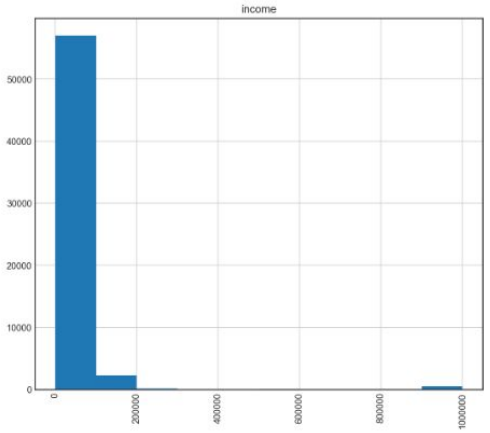
Making graphs

```
    if future_data.dtypes[column] == np.object:
        future_data[column].value_counts().plot(kind="bar", axes=ax)
    else:
        future_data[column].hist(axes=ax)
        plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.5, wspace=0.2)
```

```
59945                  medicine / health

[59946 rows x 9 columns]
```

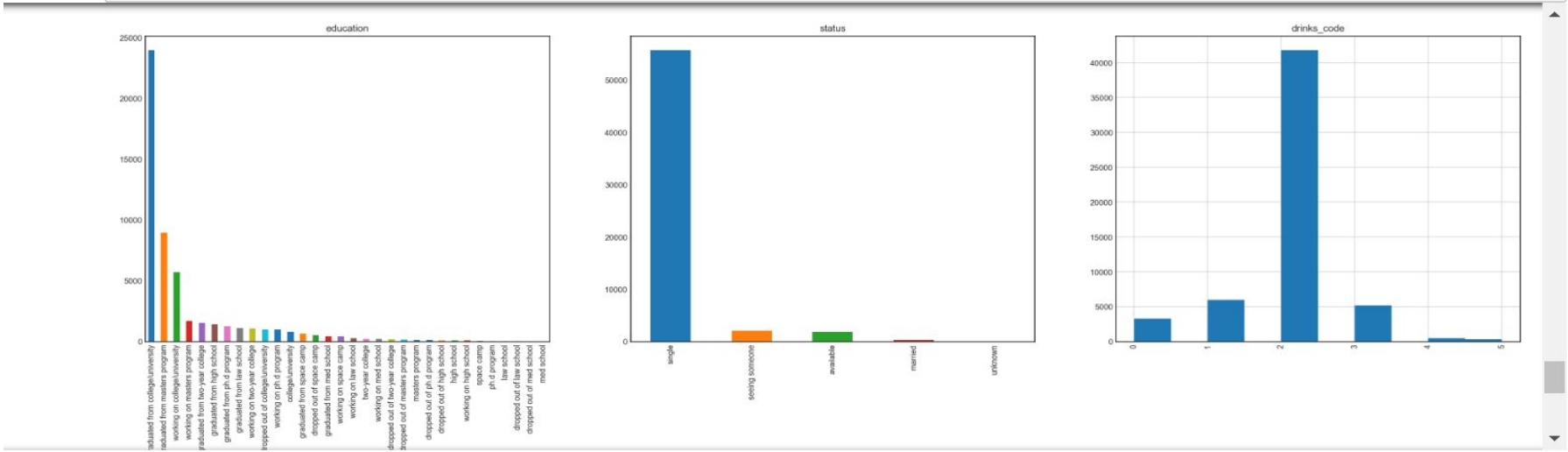






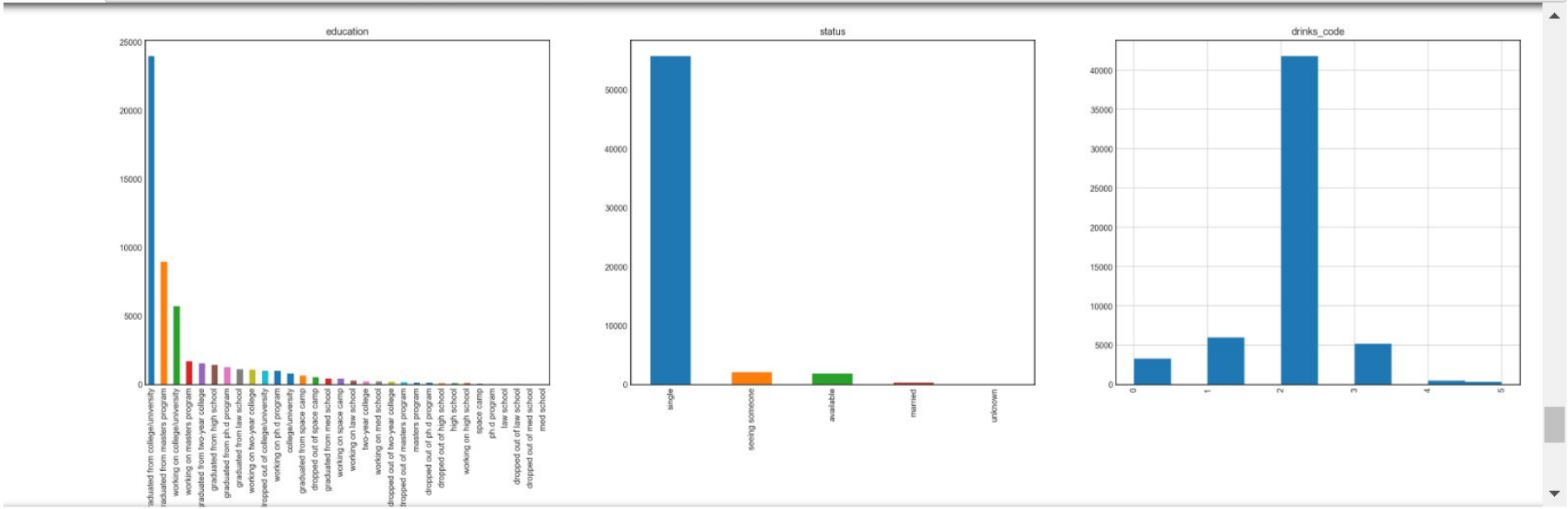
Making graphs

```
    if future_data.dtypes[column] == np.object:
        future_data[column].value_counts().plot(kind="bar", axes=ax)
    else:
        future_data[column].hist(axes=ax)
        plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.5, wspace=0.2)
```



Making graphs

```
    if future_data.dtypes[column] == np.object:
        future_data[column].value_counts().plot(kind="bar", axes=ax)
    else:
        future_data[column].hist(axes=ax)
        plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.5, wspace=0.2)
```

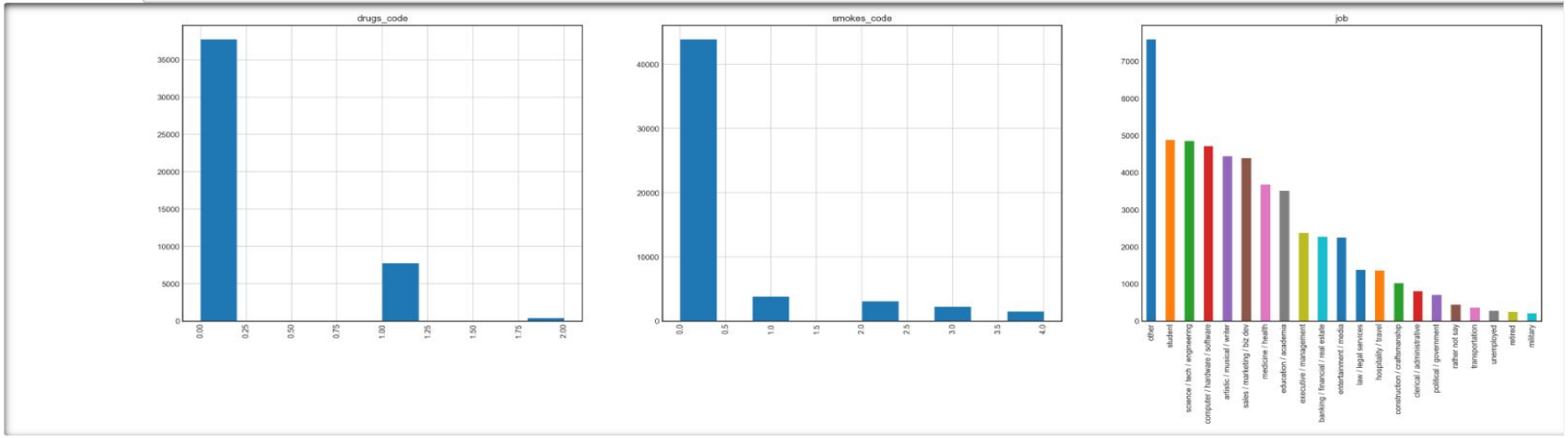


Making graphs

File Edit View Insert Cell Kernel Widgets Help Trusted Pytho

Run Code

```
    if future_data.dtypes[column] == np.object:
        future_data[column].value_counts().plot(kind="bar", axes=ax)
    else:
        future_data[column].hist(axes=ax)
        plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.5, wspace=0.2)
```

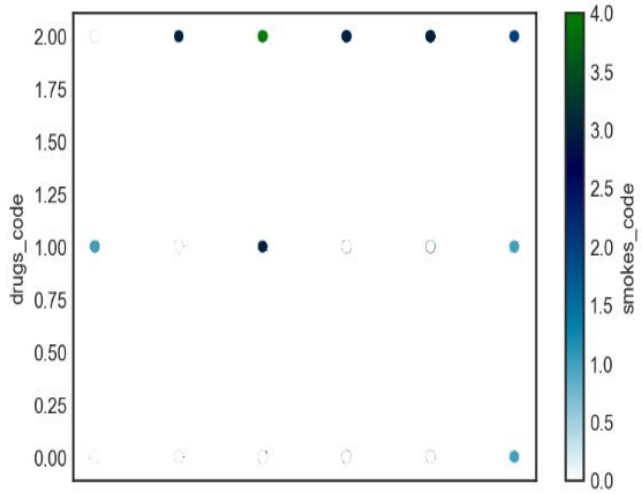


Correlation between drinks, drugs and smokes

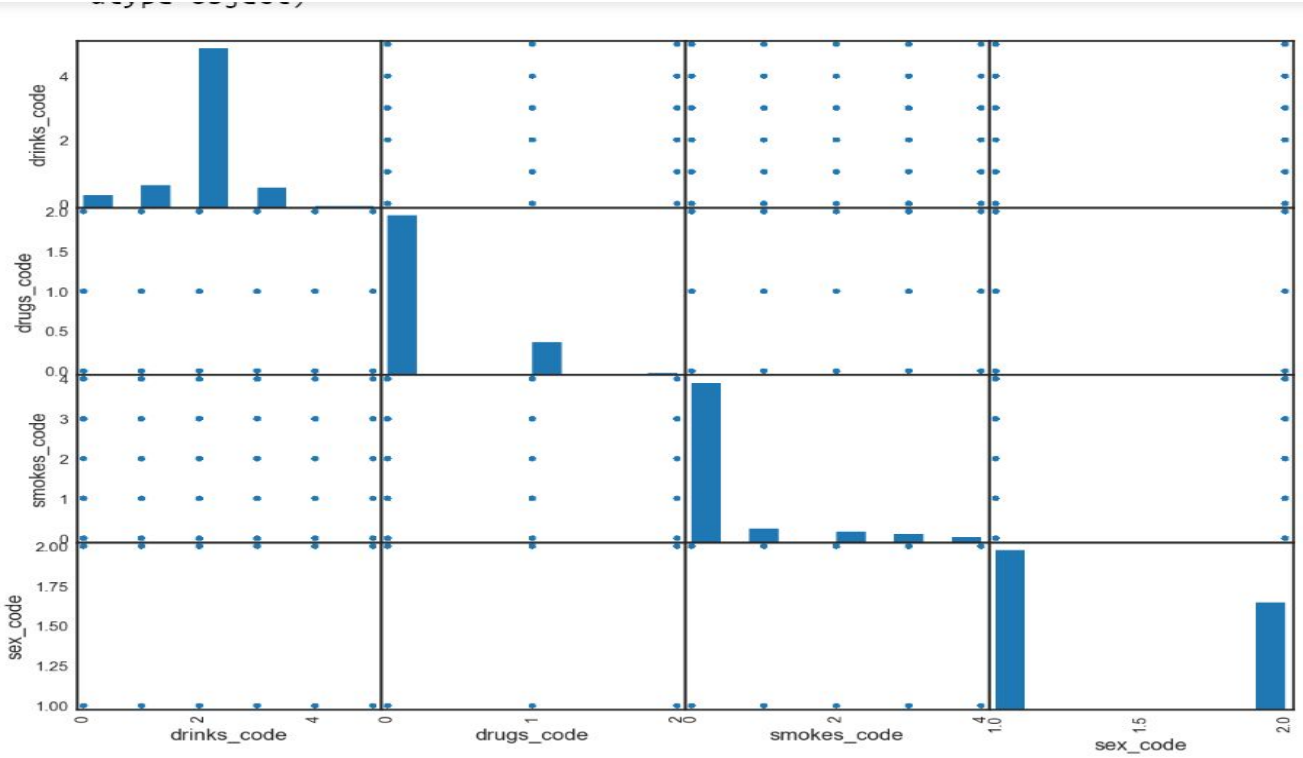
```
Name: status_code, dtype: int64
```

```
In [118]: future_data.plot(kind = 'scatter', x ='drinks_code', y = 'drugs_code', c = 'smokes_code', colormap = 'ocean_r')
```

```
Out[118]: <matplotlib.axes._subplots.AxesSubplot at 0x1c42a5cfba8>
```



Making graphs



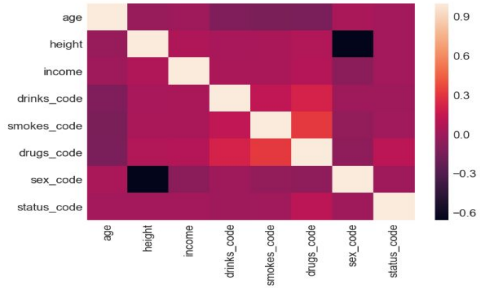
```
In [140]: df.corr()
```

Out[140]:

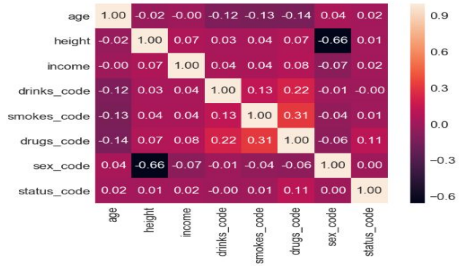
	age	height	income	drinks_code	smokes_code	drugs_code	sex_code	status_code
age	1.000000	-0.022262	-0.001004	-0.124017	-0.134493	-0.141840	0.041481	0.015221
height	-0.022262	1.000000	0.065049	0.034801	0.041142	0.070707	-0.655448	0.011826
income	-0.001004	0.065049	1.000000	0.042515	0.037264	0.081115	-0.074601	0.016152
drinks_code	-0.124017	0.034801	0.042515	1.000000	0.131600	0.218327	-0.007057	-0.003094
smokes_code	-0.134493	0.041142	0.037264	0.131600	1.000000	0.314700	-0.043324	0.005964
drugs_code	-0.141840	0.070707	0.081115	0.218327	0.314700	1.000000	-0.056780	0.113470
sex_code	0.041481	-0.655448	-0.074601	-0.007057	-0.043324	-0.056780	1.000000	0.001136
status_code	0.015221	0.011826	0.016152	-0.003094	0.005964	0.113470	0.001136	1.000000

```
In [148]: sns.heatmap(df.corr())
```

Out[148]: <matplotlib.axes._subplots.AxesSubplot at 0x1c42adadb38>

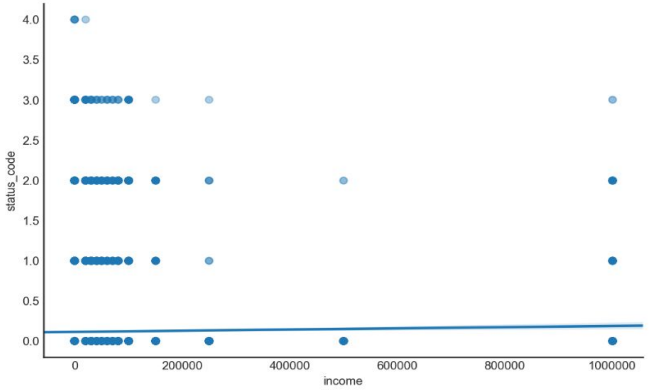



```
In [143]: sns.heatmap(df.corr(),
                      cbar=True,
                      annot=True,
                      square=True,
                      fmt='.2f',
                      annot_kws={'size':10},)
plt.show()
```



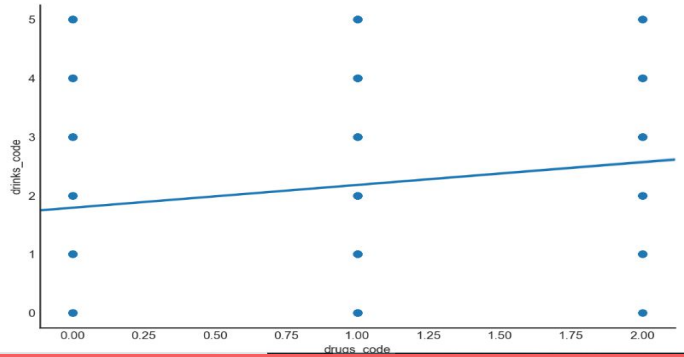
In [161]:
```
sns.lmplot(x='income',
           y='status_code',
           data=df,
           aspect=1.5,
           scatter_kws={'alpha':0.2})
```

Out[161]: <seaborn.axisgrid.FacetGrid at 0x1c42f4fd240>



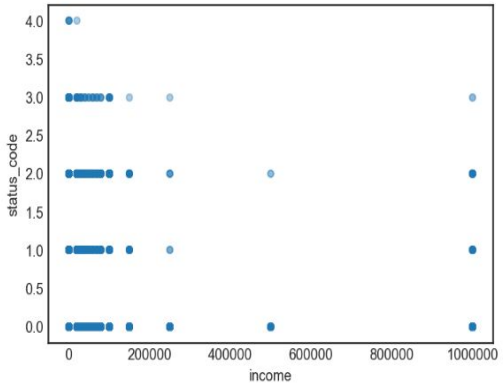
In [151]:
```
sns.lmplot(x='drugs_code',
           y='drinks_code',
           data=df,
           aspect=1.5,
           scatter_kws={'alpha':0.2})
```

Out[151]: <seaborn.axisgrid.FacetGrid at 0x1c42ae0f9e8>



In [158]:
```
df.plot(kind='scatter',x='income',y= 'status_code',alpha=0.2)
```

Out[158]: <matplotlib.axes._subplots.AxesSubplot at 0x1c42dd154e0>



In [149]:
```
df.plot(kind='scatter',x='income',y= 'sex_code',alpha=0.2)
```

Out[149]: <matplotlib.axes._subplots.AxesSubplot at 0x1c42ae0f630>