dataMining02-data_exploration-adults

October 29, 2019

1 From UCI Machine Learning Repository

1.1 Adult dataset

This data file does not have a header with column names. Look at the ".names" text file in the Data Folder and use the same procedure used for Iris

Print als the types of the columns using the `types` attribute

names = ['age','workclass','fnlwgt','education','education-num','marital-status','occupation','relationship','race','sex','capital-gain','capital-loss','hours-per-week','native-country', 'high-income']

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'

Load the data in the dataframe `df` and then show the column types with the `.dtypes` attribute of a Pandas DataFrame

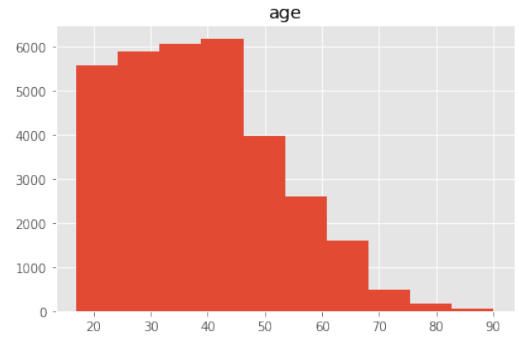
```
age               int64
workclass        object
fnlwgt            int64
education        object
education-num     int64
marital-status   object
occupation       object
relationship     object
race             object
sex              object
capital-gain      int64
capital-loss      int64
hours-per-week    int64
native-country   object
high-income      object
dtype: object
```

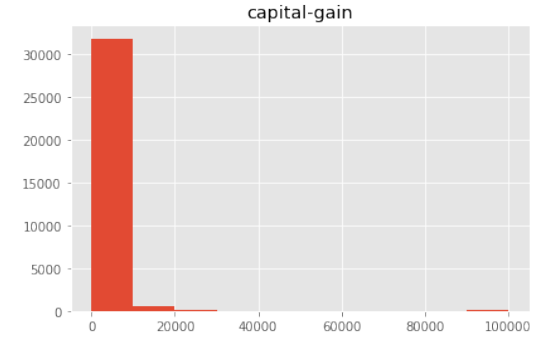
Show the head and then generate the histograms for all the columns

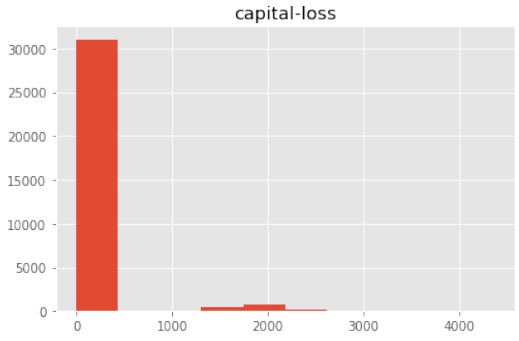
[4]:

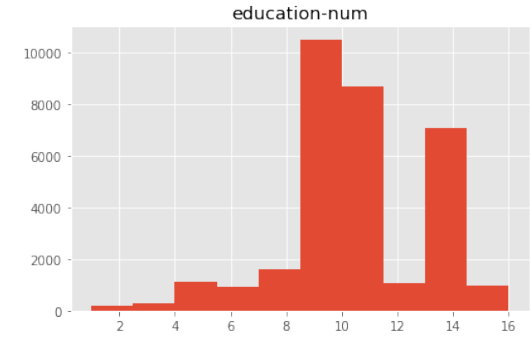
	age	workclass	fnlwgt	education	education-num	\
0	39	State-gov	77516	Bachelors	13	
1	50	Self-emp-not-inc	83311	Bachelors	13	
2	38	Private	215646	HS-grad	9	

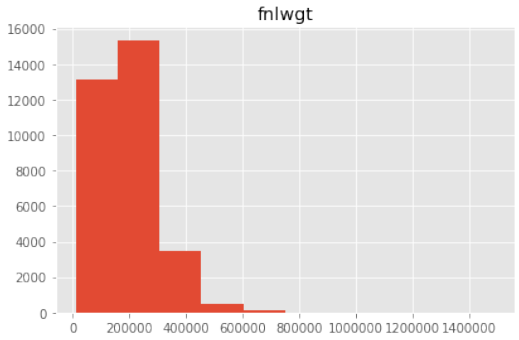
```
3   53            Private  234721        11th             7
4   28            Private  338409   Bachelors            13

       marital-status          occupation    relationship    race      sex  \
0       Never-married        Adm-clerical   Not-in-family   White     Male
1   Married-civ-spouse     Exec-managerial         Husband   White     Male
2             Divorced   Handlers-cleaners   Not-in-family   White     Male
3   Married-civ-spouse   Handlers-cleaners         Husband   Black     Male
4   Married-civ-spouse      Prof-specialty            Wife   Black   Female

   capital-gain  capital-loss  hours-per-week  native-country high-income
0          2174             0              40   United-States       <=50K
1             0             0              13   United-States       <=50K
2             0             0              40   United-States       <=50K
3             0             0              40   United-States       <=50K
4             0             0              40            Cuba       <=50K
```

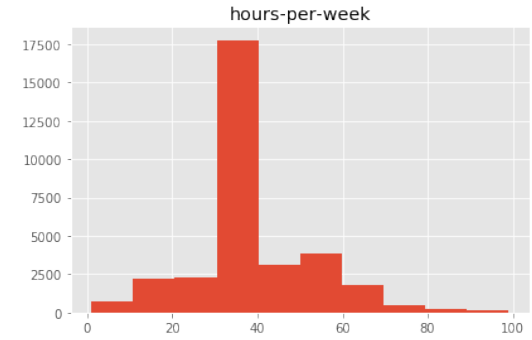






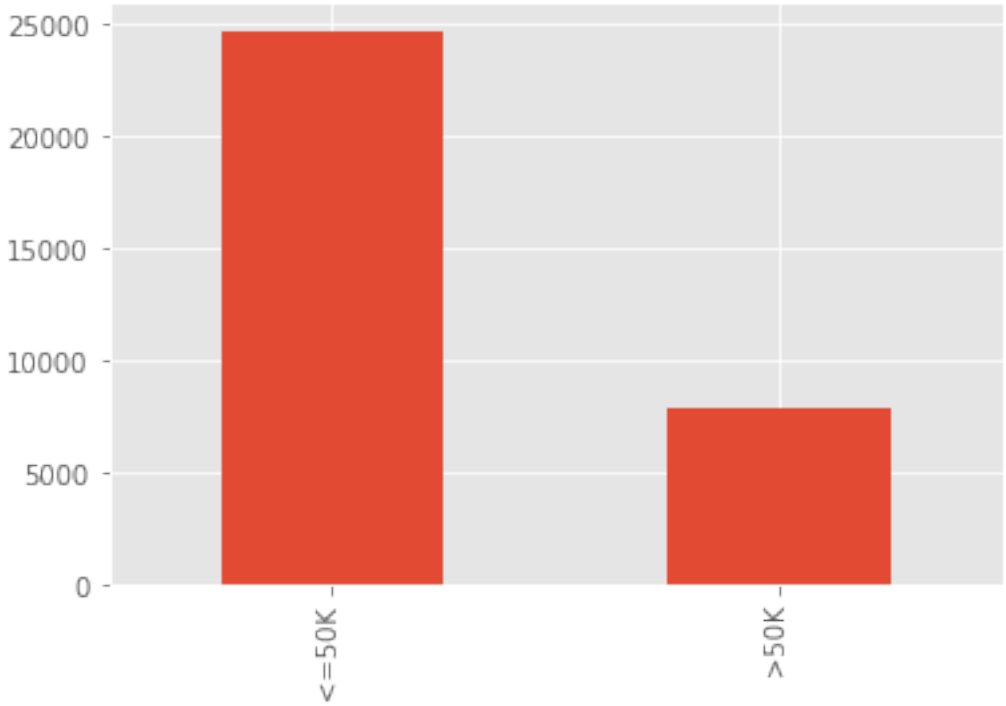






Show a bar graph with the value counts of the attribute `high-income`. Use the method `value_counts` of Pandas, then `plot` with the option `kind = 'bar'`

[15]: `<matplotlib.axes._subplots.AxesSubplot at 0x228c9a0f5c0>`



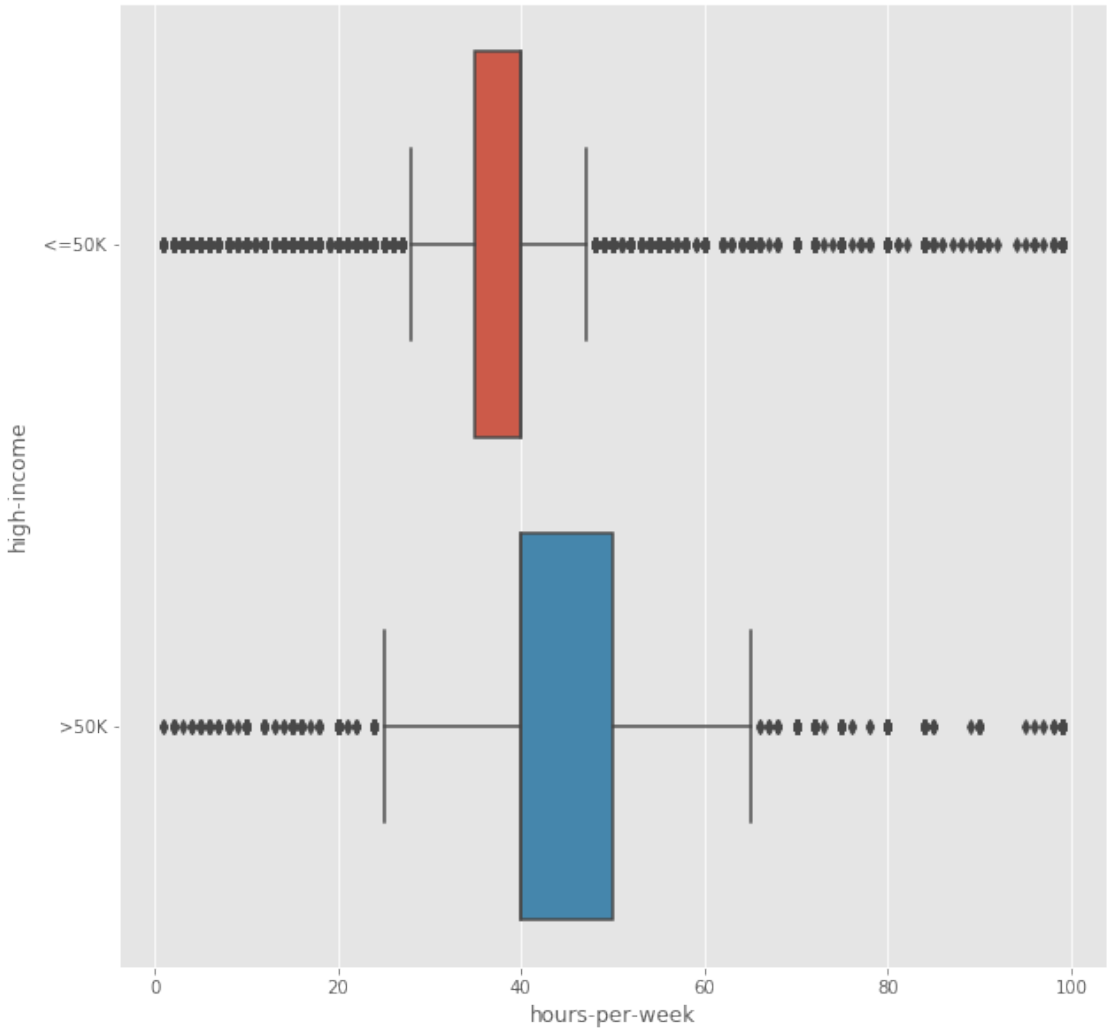
1.1.1 More examples of figures

Boxplot

More on boxplots

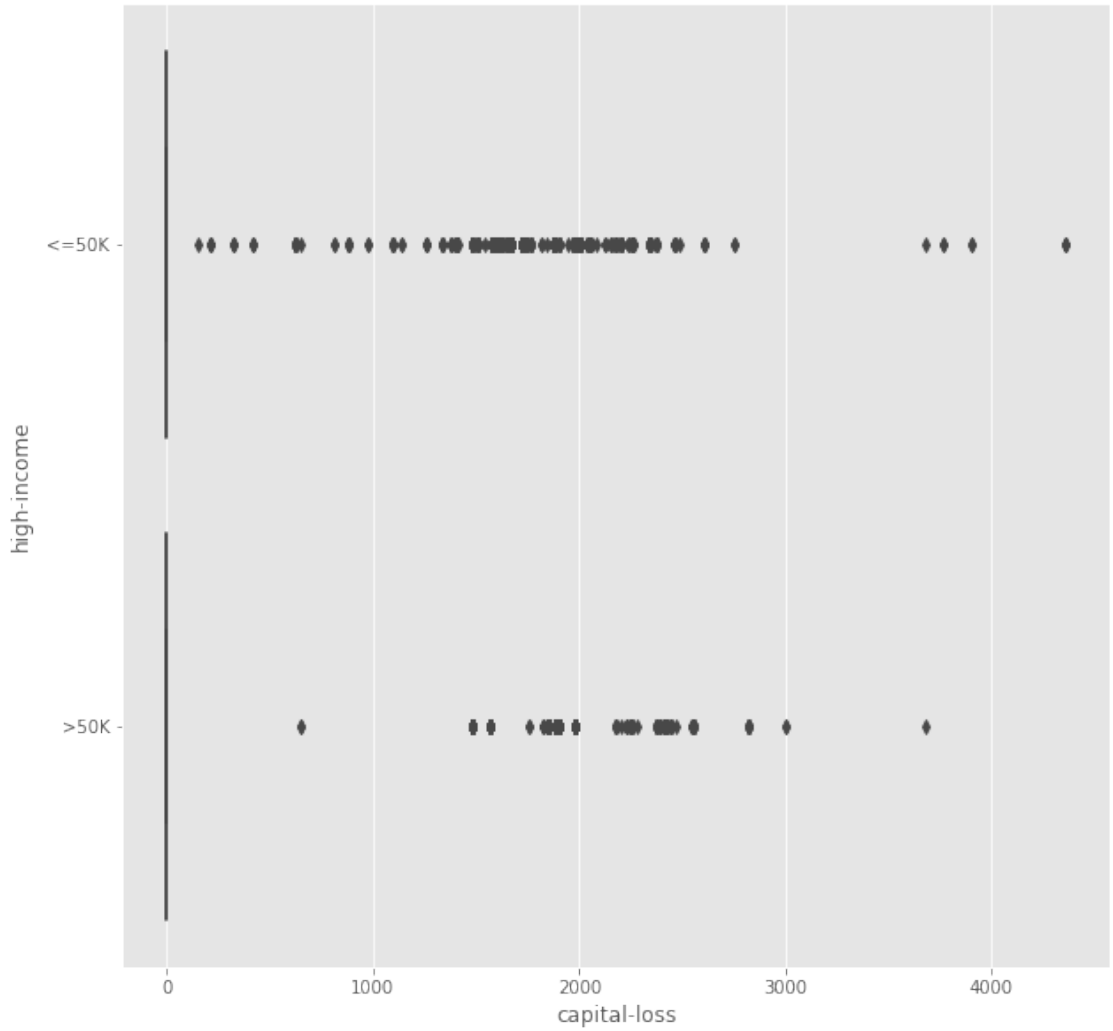
Use the `boxplot` method of Seaborn with `hours-per-week` in the `x` axis and `high-income` in the `y` axis. The columns are extracted with the `loc` method of Pandas DataFrames, with index expression `[:,'attribute-name']` (means all the elements of column `attribute-name`)

[16]: `<matplotlib.axes._subplots.AxesSubplot at 0x228c99a1240>`



Similar boxplot for `'capital-loss'` and `'high-income']`

[17]: `<matplotlib.axes._subplots.AxesSubplot at 0x228c99576a0>`



Something is wrong, the figure does not look like a proper boxplot.

Let's look at the **capital-loss** column with the `describe` method

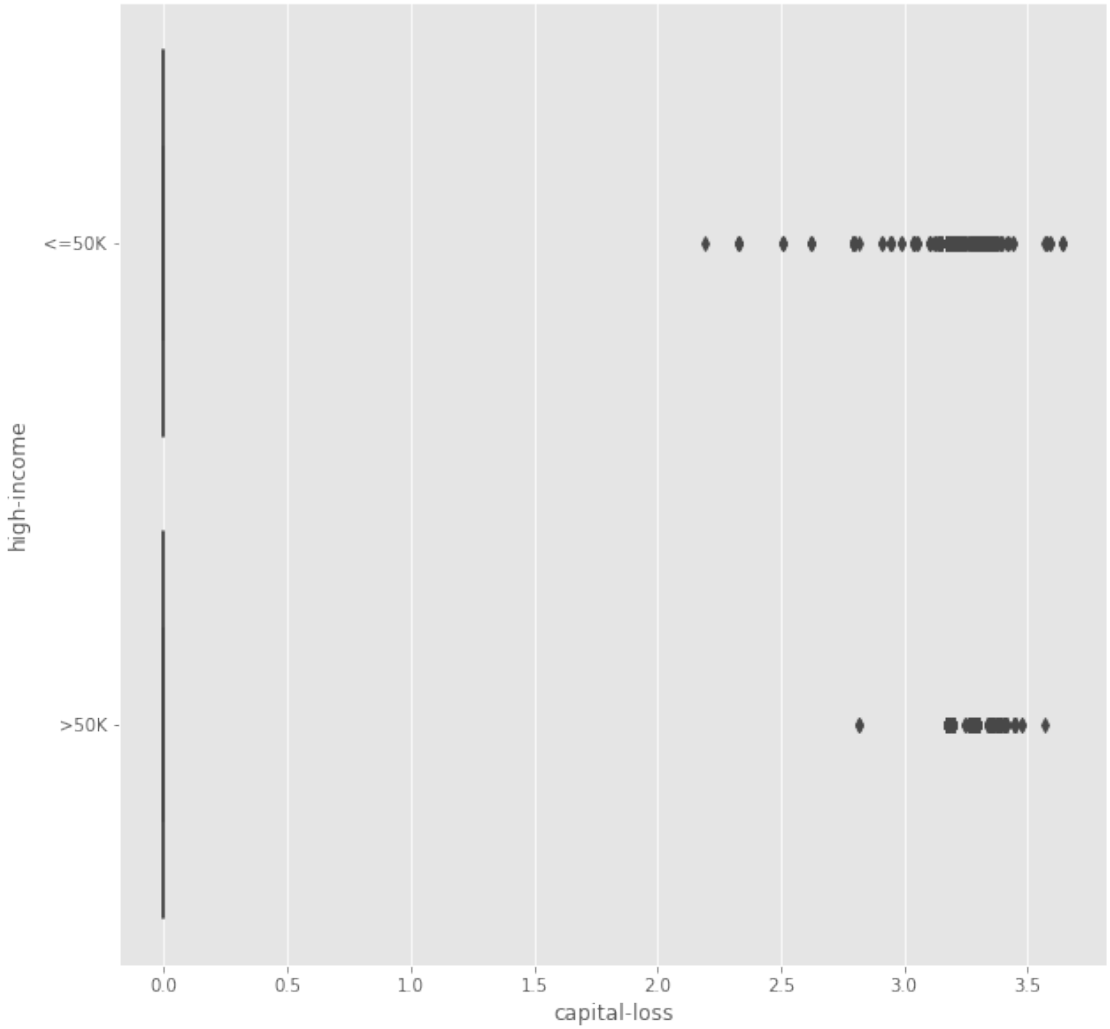
[18]:
```
count    32561.000000
mean        87.303830
std        402.960219
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max       4356.000000
Name: capital-loss, dtype: float64
```

The three quartiles are all zero, and there are no left outliers.

Let's try with a logarithmic transformation (add +1 to deal with the zero values) - use the `log10`

function of `numpy` to transform the `capital-loss+1` - prepare a plot figure of size [10,10] - boxplot with Seaborn

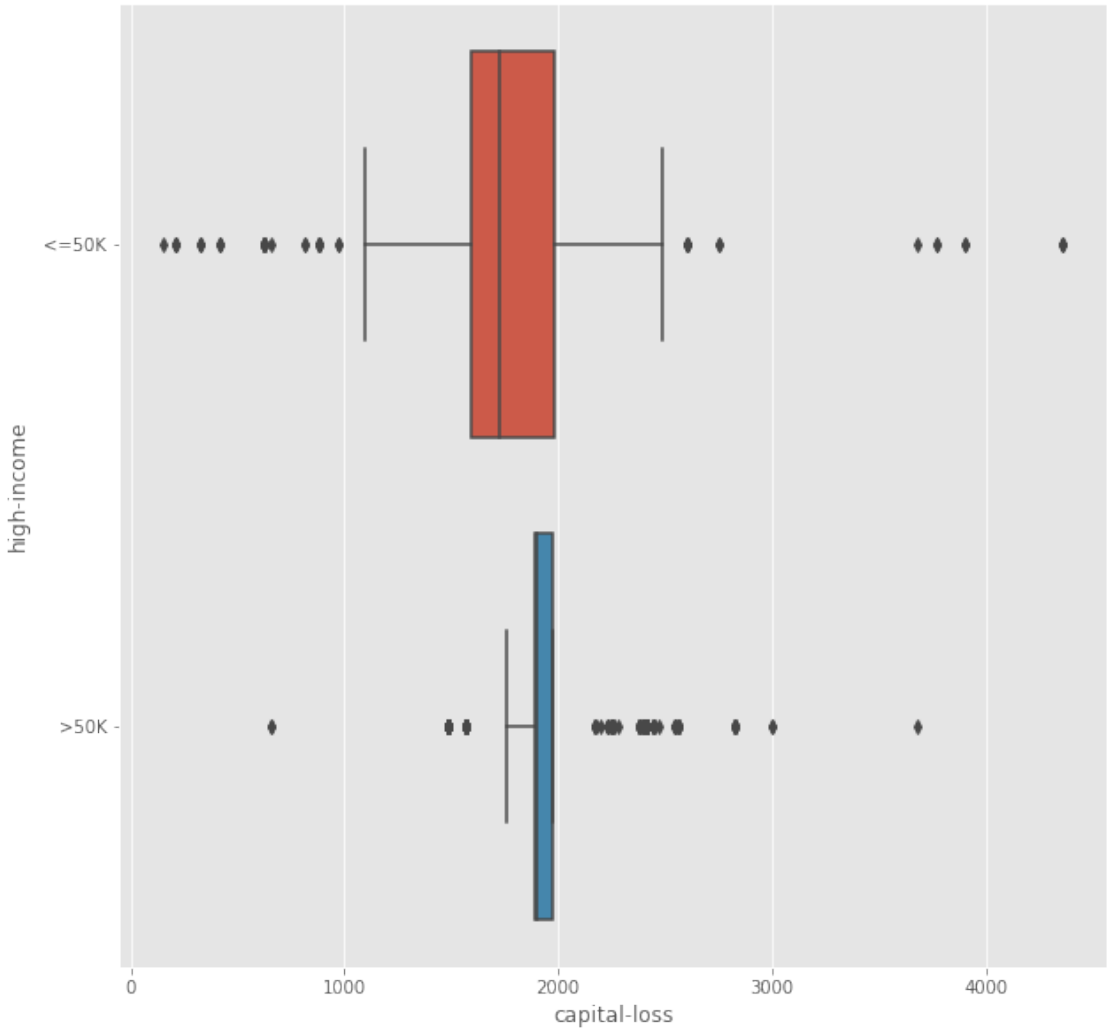
[19]: `<matplotlib.axes._subplots.AxesSubplot at 0x228c98b95c0>`



You can observe that a most of the data are 'compressed' at 0 - it is due to the zero values to which we added 1, whose log is 0 again

Look at the rows with non-zero values: in the x values, instead of the `:` indicating 'all the rows' we must use a 'selector expression', in this case `df['capital-loss']!=0`

[20]: `<matplotlib.axes._subplots.AxesSubplot at 0x228ca1afc50>`

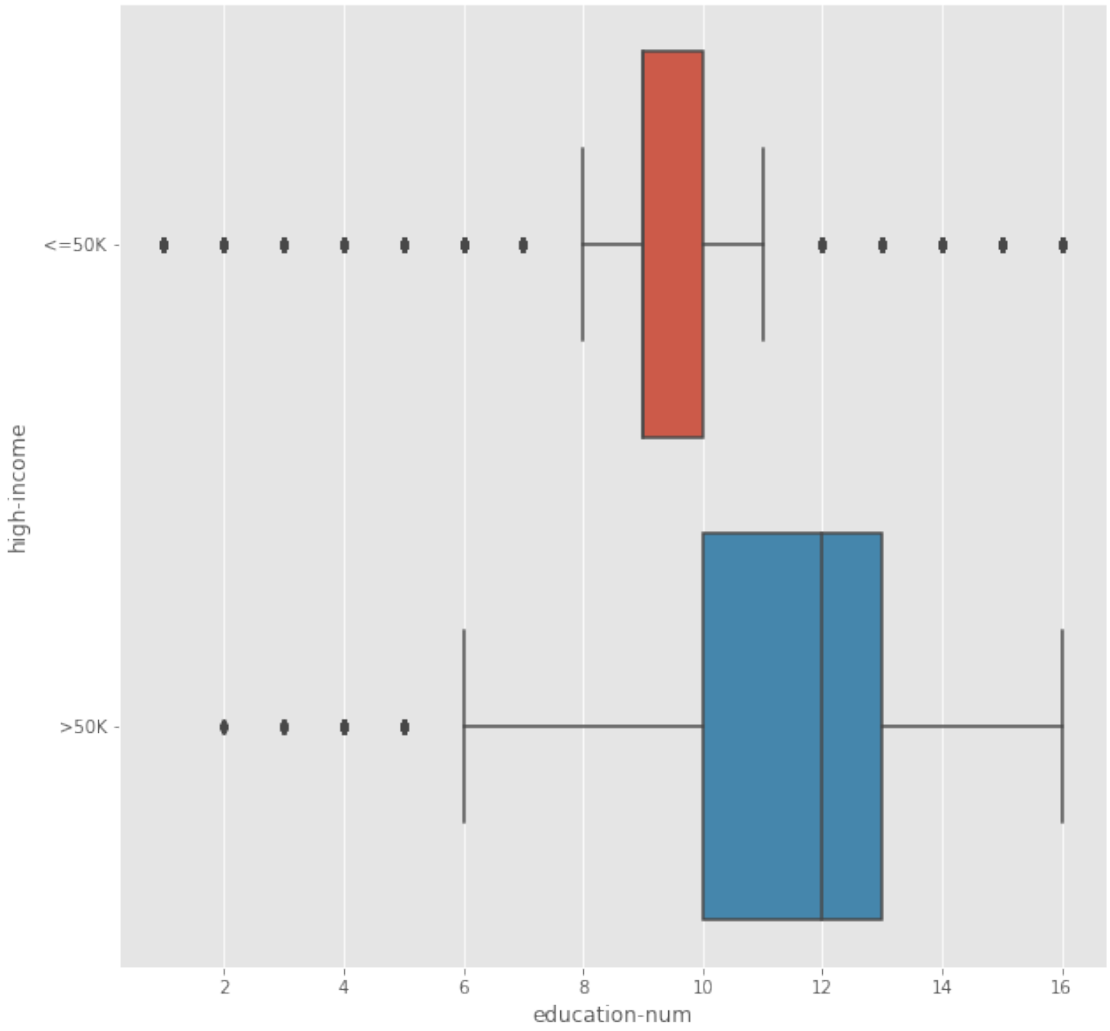


Now we see that the non-zero values have some structure

1.1.2 Plot another pair of columns

`education-num` and `high-income`

[21]: `<matplotlib.axes._subplots.AxesSubplot at 0x228c9ac1240>`


<=50K
>50K
high-income
2
4
6
8
10
12
14
16
education-num