Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №2
по дисциплине
«Методы машинного обучения»
на тему
«Изучение библиотек обработки данных»

Выполнил:
студент группы ИУ5-24М
Мельников К.

Москва — 2019 г.

0.1. mlcourse.ai – Open Machine Learning Course

Author:Yury Kashnitskiy. Translated and edited by Sergey Isaev, Artem Trunov, Anastasia Manokhina, and Yuanyuan Pao

#

Assignment #1 (demo) ##

Exploratory data analysis with Pandas

In this task you should use Pandas to answer a few questions about the Adult dataset. (You don't have to download the data – it's already in the repository). Choose the answers in the web-form.

Unique values of all features (for more information, please see the links above): - `age`: continuous. - `workclass`: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. - `fnlwgt`: continuous. - `education`: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. - `education-num`: continuous. - `marital-status`: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. - `occupation`: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. - `relationship`: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. - `race`: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. - `sex`: Female, Male. - `capital-gain`: continuous. - `capital-loss`: continuous. - `hours-per-week`: continuous. - `native-country`: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- `salary`: >50K,<=50K

```
In [1]: import numpy as np
        import pandas as pd
        pd.set_option('display.max.columns', 100)
        # to draw pictures in jupyter notebook
        %matplotlib inline
        import matplotlib.pyplot as plt
        import seaborn as sns
        # we don't like warnings
        # you can comment the following 2 lines if you'd like to
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [2]: data = pd.read_csv('adult.data.csv')
        data.head()
```

```
Out[2]:    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 salary
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
```

In [3]:
```
data.shape
```

Out[3]:
```
(32561, 15)
```

1. How many men and women (*sex* feature) are represented in this dataset?

In [4]:
```
# You code here
data['sex'].value_counts()
```

Out[4]:
```
Male      21790
Female    10771
Name: sex, dtype: int64
```

2. What is the average age (*age* feature) of women?

In [5]:
```
# You code here
data[data['sex']=='Female']['age'].mean()
```

Out[5]:
```
36.85823043357163
```

3. What is the percentage of German citizens (*native-country* feature)?

In [6]:
```
# You code here
"{0:.2%}".format(data[data['native-country']=='Germany'].shape[0]/data.sha
```

Out[6]:
```
'0.42%'
```

4-5. What are the mean and standard deviation of age for those who earn more than 50K per year (*salary* feature) and those who earn less than 50K per year?

In [7]:
```
# You code here
print(data[data['salary']=='>50K']['age'].mean())
print(data[data['salary']=='>50K']['age'].std())
print(data[data['salary']=='<=50K']['age'].mean())
print(data[data['salary']=='<=50K']['age'].std())
```

```
44.24984058155847
10.519027719851826
36.78373786407767
14.02008849082488
```

6. Is it true that people who earn more than 50K have at least high school education? (*education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters* or *Doctorate* feature)

```
In [8]: # You code here
        print(data[data['salary']=='>50K']['education'].value_counts())
        print('FALSE')
```

```
Bachelors       2221
HS-grad         1675
Some-college    1387
Masters          959
Prof-school      423
Assoc-voc        361
Doctorate        306
Assoc-acdm       265
10th              62
11th              60
7th-8th           40
12th              33
9th               27
5th-6th           16
1st-4th            6
Name: education, dtype: int64
FALSE
```

7. Display age statistics for each race (*race* feature) and each gender (*sex* feature). Use *groupby()* and *describe()*. Find the maximum age of men of *Amer-Indian-Eskimo* race.

```
In [9]: # You code here
        data.groupby(['race']).describe()
```

```
Out[9]:                       age
                            count       mean        std   min   25%   50%   75%
        race
        Amer-Indian-Eskimo  311.0  37.173633  12.447130  17.0  28.0  35.0  45.5
        Asian-Pac-Islander 1039.0  37.746872  12.825133  17.0  28.0  36.0  45.0
        Black              3124.0  37.767926  12.759290  17.0  28.0  36.0  46.0
        Other               271.0  33.457565  11.538865  17.0  25.0  31.0  41.0
        White             27816.0  38.769881  13.782306  17.0  28.0  37.0  48.0

                                  fnlwgt
                            max    count            mean              std        min
```

race					
Amer-Indian-Eskimo	82.0	311.0	120831.147910	88338.636336	12285.0
Asian-Pac-Islander	90.0	1039.0	159940.609240	85122.307505	14878.0
Black	90.0	3124.0	228013.124200	123600.677422	19752.0
Other	77.0	271.0	197124.191882	88856.775370	24562.0
White	90.0	27816.0	187298.064280	103124.944196	18827.0

					education-num
	25%	50%	75%	max	count
race					
Amer-Indian-Eskimo	35036.5	102541.0	176142.0	445168.0	311.0
Asian-Pac-Islander	93321.0	143098.0	194456.0	506329.0	1039.0
Black	148212.0	205934.0	281693.0	1268339.0	3124.0
Other	138103.0	188793.0	240091.0	481175.0	271.0
White	116902.5	177627.0	233542.5	1484705.0	27816.0

\

	mean	std	min	25%	50%	75%	max
race							
Amer-Indian-Eskimo	9.311897	2.310387	2.0	9.0	9.0	10.0	16.0
Asian-Pac-Islander	10.960539	2.811582	1.0	9.0	10.0	13.0	16.0
Black	9.486236	2.297893	1.0	9.0	9.0	10.0	16.0
Other	8.841328	3.226153	1.0	7.5	9.0	10.0	16.0
White	10.135246	2.570307	1.0	9.0	10.0	13.0	16.0

	capital-gain					
	count	mean	std	min	25%	50%
race						
Amer-Indian-Eskimo	311.0	625.266881	2753.238961	0.0	0.0	0.0
Asian-Pac-Islander	1039.0	1478.358037	9986.156906	0.0	0.0	0.0
Black	3124.0	609.940461	5139.653447	0.0	0.0	0.0
Other	271.0	934.660517	8625.128995	0.0	0.0	0.0
White	27816.0	1121.660375	7504.533302	0.0	0.0	0.0

		capital-loss				
	max	count	mean	std	min	25%
race						
Amer-Indian-Eskimo	27828.0	311.0	34.176849	245.583106	0.0	0.0
Asian-Pac-Islander	99999.0	1039.0	97.222329	423.556931	0.0	0.0
Black	99999.0	3124.0	60.385083	337.394121	0.0	0.0
Other	99999.0	271.0	61.070111	322.452705	0.0	0.0
White	99999.0	27816.0	90.806155	410.833347	0.0	0.0

				hours-per-week		
	50%	75%	max	count	mean	std
race						
Amer-Indian-Eskimo	0.0	0.0	1980.0	311.0	40.048232	11.695364
Asian-Pac-Islander	0.0	0.0	2457.0	1039.0	40.127045	12.556816
Black	0.0	0.0	4356.0	3124.0	38.422855	10.315646
Other	0.0	0.0	2179.0	271.0	39.468635	11.143755

```
White                0.0  0.0  4356.0       27816.0  40.689100  12.544796


                     min   25%   50%   75%   max
race
Amer-Indian-Eskimo   3.0  40.0  40.0  40.0  84.0
Asian-Pac-Islander   1.0  40.0  40.0  40.0  99.0
Black                1.0  37.0  40.0  40.0  99.0
Other                5.0  36.0  40.0  40.0  98.0
White                1.0  40.0  40.0  45.0  99.0
```

```
In [10]: data[data['race']=='Amer-Indian-Eskimo']['age'].max()
```

```
Out[10]: 82
```

8. Among whom is the proportion of those who earn a lot (>50K) greater: married or single men (*marital-status* feature)? Consider as married those who have a *marital-status* starting with *Married* (Married-civ-spouse, Married-spouse-absent or Married-AF-spouse), the rest are considered bachelors.

```
In [11]: data[data['salary']=='>50K'].groupby('marital-status').describe()
```

```
Out[11]:                           age
                           count       mean        std   min    25%   50%
marital-status
Divorced                   463.0  45.645788   8.554373  24.0  40.00  45.0
Married-AF-spouse           10.0  31.300000   6.700746  22.0  27.50  29.5
Married-civ-spouse        6692.0  44.436192  10.383282  19.0  37.00  44.0
Married-spouse-absent       34.0  47.323529  10.803256  28.0  41.25  47.5
Never-married              491.0  38.217923  10.262840  19.0  30.50  36.0
Separated                   66.0  42.348485  10.043739  23.0  34.25  42.0
Widowed                     85.0  58.588235  11.536994  29.0  51.00  58.0

                                fnlwgt
                         max   count           mean            std      mi
marital-status
Divorced                69.0   463.0  185425.691145   98586.532618  20296
Married-AF-spouse       43.0    10.0  166411.200000  125626.413447  26892
Married-civ-spouse      90.0  6692.0  187893.779737  102330.197150  14878
Married-spouse-absent   77.0    34.0  159234.735294   84719.581980  27444
Never-married           90.0   491.0  196258.124236  111550.846060  23438
Separated               64.0    66.0  210674.075758  100937.924364  27766
Widowed                 81.0    85.0  159583.705882   82429.176123  23074

                                                                     \
                              25%       50%        75%        max
marital-status
Divorced                119602.00  176037.0  228547.50   682947.0
Married-AF-spouse        52022.25  179044.0  221776.75   436341.0
Married-civ-spouse      119101.00  176026.5  231386.00  1226583.0
Married-spouse-absent   109759.50  164552.0  207886.50   344415.0
```

```
Never-married          122622.50  178134.0  235195.50  1033222.0
Separated              141621.25  195111.5  270711.75   581071.0
Widowed                102359.00  150389.0  214627.00   411007.0

                      education-num
                              count       mean       std  min    25%   5
marital-status
Divorced                      463.0  11.881210  2.339417  5.0  10.00  13
Married-AF-spouse              10.0  11.000000  1.825742  9.0   9.25  10
Married-civ-spouse           6692.0  11.525852  2.375349  2.0   9.00  12
Married-spouse-absent          34.0  12.088235  2.597990  4.0  10.00  13
Never-married                 491.0  12.529532  2.307186  3.0  11.00  13
Separated                      66.0  12.166667  2.291008  6.0  10.00  13
Widowed                        85.0  11.047059  2.586162  4.0   9.00  11

                                   capital-gain
                         75%   max        count         mean           s
marital-status
Divorced               14.00  16.0        463.0  5781.812095  16205.0921
Married-AF-spouse      13.00  13.0         10.0   729.800000   2307.8302
Married-civ-spouse     13.00  16.0       6692.0  3678.163927  14262.5303
Married-spouse-absent  13.75  16.0         34.0  6836.647059  18411.1691
Never-married          14.00  16.0        491.0  6137.576375  16295.8947
Separated              14.00  16.0         66.0  6614.727273  18159.7521
Widowed                13.00  16.0         85.0  5071.117647  12848.6627

                                                       capital-loss  \
                       min  25%  50%      75%      max        count
marital-status
Divorced               0.0  0.0  0.0  4934.00  99999.0        463.0
Married-AF-spouse      0.0  0.0  0.0     0.00   7298.0         10.0
Married-civ-spouse     0.0  0.0  0.0     0.00  99999.0       6692.0
Married-spouse-absent  0.0  0.0  0.0  3590.25  99999.0         34.0
Never-married          0.0  0.0  0.0  8614.00  99999.0        491.0
Separated              0.0  0.0  0.0  4934.00  99999.0         66.0
Widowed                0.0  0.0  0.0  4934.00  99999.0         85.0


                             mean         std  min  25%  50%  75%     ma
marital-status
Divorced               137.853132  550.166480  0.0  0.0  0.0  0.0  3004.
Married-AF-spouse        0.000000    0.000000  0.0  0.0  0.0  0.0     0.
Married-civ-spouse     198.975194  593.202167  0.0  0.0  0.0  0.0  2415.
Married-spouse-absent  164.705882  552.696504  0.0  0.0  0.0  0.0  2472.
Never-married          188.971487  630.983524  0.0  0.0  0.0  0.0  3683.
Separated              232.621212  743.917236  0.0  0.0  0.0  0.0  2824.
Widowed                234.129412  710.191704  0.0  0.0  0.0  0.0  2824.

                      hours-per-week
                               count       mean        std  min   25%
```

```
marital-status
Divorced                   463.0  47.336933  10.811794   5.0  40.0  4
Married-AF-spouse           10.0  42.600000   4.115013  40.0  40.0  4
Married-civ-spouse        6692.0  45.303945  11.045102   1.0  40.0  4
Married-spouse-absent       34.0  45.058824  15.273152  16.0  40.0  4
Never-married              491.0  46.678208   9.923293   7.0  40.0  4
Separated                   66.0  46.212121   9.162659  24.0  40.0  4
Widowed                     85.0  41.600000  13.424569   2.0  40.0  4


                         75%   max
marital-status
Divorced               50.00  99.0
Married-AF-spouse      43.50  50.0
Married-civ-spouse     50.00  99.0
Married-spouse-absent  53.75  80.0
Never-married          50.00  90.0
Separated              50.00  80.0
Widowed                50.00  80.0
```

```
In [12]: # You code here`
         data[(data['salary']=='>50K') & (~data['marital-status'].str.contains('Ma

Out[12]: (1105, 15)

In [13]: data[(data['salary']=='>50K') & (data['marital-status'].str.contains('Ma

Out[13]: (6736, 15)
```

9. What is the maximum number of hours a person works per week (*hours-per-week* feature)? How many people work such a number of hours, and what is the percentage of those who earn a lot (>50K) among them?

```
In [14]: # You code here
         maxx = data['hours-per-week'].max()
         workers = data[data['hours-per-week'] == maxx]
         print(workers.count()[0])
         print(workers[workers['salary']=='>50K'].count()[0]/workers.count()[0])

85
0.29411764705882354
```

10. Count the average time of work (*hours-per-week*) for those who earn a little and a lot (*salary*) for each country (*native-country*). What will these be for Japan?

```
In [15]: # You code here
         for (country, salary), sub_df in data.groupby(['native-country', 'salary
             print(country, salary, round(sub_df['hours-per-week'].mean(), 2))
```

```
? <=50K 40.16
? >50K 45.55
Cambodia <=50K 41.42
Cambodia >50K 40.0
Canada <=50K 37.91
Canada >50K 45.64
China <=50K 37.38
China >50K 38.9
Columbia <=50K 38.68
Columbia >50K 50.0
Cuba <=50K 37.99
Cuba >50K 42.44
Dominican-Republic <=50K 42.34
Dominican-Republic >50K 47.0
Ecuador <=50K 38.04
Ecuador >50K 48.75
El-Salvador <=50K 36.03
El-Salvador >50K 45.0
England <=50K 40.48
England >50K 44.53
France <=50K 41.06
France >50K 50.75
Germany <=50K 39.14
Germany >50K 44.98
Greece <=50K 41.81
Greece >50K 50.62
Guatemala <=50K 39.36
Guatemala >50K 36.67
Haiti <=50K 36.33
Haiti >50K 42.75
Holand-Netherlands <=50K 40.0
Honduras <=50K 34.33
Honduras >50K 60.0
Hong <=50K 39.14
Hong >50K 45.0
Hungary <=50K 31.3
Hungary >50K 50.0
India <=50K 38.23
India >50K 46.48
Iran <=50K 41.44
Iran >50K 47.5
Ireland <=50K 40.95
Ireland >50K 48.0
Italy <=50K 39.62
Italy >50K 45.4
Jamaica <=50K 38.24
Jamaica >50K 41.1
Japan <=50K 41.0
Japan >50K 47.96
Laos <=50K 40.38
```

```
Laos >50K 40.0
Mexico <=50K 40.0
Mexico >50K 46.58
Nicaragua <=50K 36.09
Nicaragua >50K 37.5
Outlying-US(Guam-USVI-etc) <=50K 41.86
Peru <=50K 35.07
Peru >50K 40.0
Philippines <=50K 38.07
Philippines >50K 43.03
Poland <=50K 38.17
Poland >50K 39.0
Portugal <=50K 41.94
Portugal >50K 41.5
Puerto-Rico <=50K 38.47
Puerto-Rico >50K 39.42
Scotland <=50K 39.44
Scotland >50K 46.67
South <=50K 40.16
South >50K 51.44
Taiwan <=50K 33.77
Taiwan >50K 46.8
Thailand <=50K 42.87
Thailand >50K 58.33
Trinadad&Tobago <=50K 37.06
Trinadad&Tobago >50K 40.0
United-States <=50K 38.8
United-States >50K 45.51
Vietnam <=50K 37.19
Vietnam >50K 39.2
Yugoslavia <=50K 41.6
Yugoslavia >50K 49.5
```

```
In [16]: import time
         import pandasql as ps
```

```
In [17]: data.shape
```

```
Out[17]: (32561, 15)
```

```
In [18]: data['indexx'] = data.index
```

```
In [19]: dataLeft = pd.DataFrame([data['age'],data['workclass'],data['fnlwgt'], da
```

```
In [20]: dataRight = pd.DataFrame([data['relationship'],data['race'],data['sex'],
```

```
In [21]: dataLeft = dataLeft.sample(frac=1).reset_index()
         dataRight = dataRight.sample(frac=1).reset_index()
```

```
In [22]: dataLeft
```

Out[22]:

	index	age	workclass	fnlwgt	education	indexx
0	11042	50	Private	337606	HS-grad	11042
1	24752	20	?	220115	HS-grad	24752
2	22664	19	Private	331433	HS-grad	22664
3	22438	28	Private	197905	HS-grad	22438
4	31230	39	Private	314007	10th	31230
5	26420	43	Private	191982	Assoc-voc	26420
6	31987	22	Private	228411	Some-college	31987
7	6832	48	Private	304791	Some-college	6832
8	13395	42	Private	222884	Bachelors	13395
9	30412	41	?	213416	5th-6th	30412
10	17866	21	Private	147655	HS-grad	17866
11	12285	29	Private	234447	Some-college	12285
12	6262	23	Private	204209	Some-college	6262
13	16579	18	Private	201613	12th	16579
14	3453	18	?	139003	HS-grad	3453
15	11971	46	Private	28334	HS-grad	11971
16	15844	59	Self-emp-not-inc	223131	HS-grad	15844
17	26975	63	Private	30813	Masters	26975
18	21183	63	Private	181153	HS-grad	21183
19	5442	35	Private	48123	12th	5442
20	7664	23	Private	215395	HS-grad	7664
21	19629	20	Private	218178	HS-grad	19629
22	7410	21	Private	226145	Some-college	7410
23	21615	25	Private	297531	Bachelors	21615
24	11232	35	Private	241001	HS-grad	11232
25	16456	46	?	37672	HS-grad	16456
26	26982	65	?	240857	Bachelors	26982
27	28457	60	Local-gov	255711	Bachelors	28457
28	32335	18	?	156608	11th	32335
29	24026	49	Local-gov	98738	HS-grad	24026
…	…	..	…	…	…	…
32531	26800	27	Private	119793	Some-college	26800
32532	23901	40	Private	140559	HS-grad	23901
32533	7254	32	Private	199963	11th	7254
32534	20638	39	Private	99146	Masters	20638
32535	7439	37	Local-gov	328301	Bachelors	7439
32536	27150	47	Private	94809	Assoc-voc	27150
32537	2497	32	Private	158438	Some-college	2497
32538	4309	58	?	142158	HS-grad	4309
32539	32336	32	Private	172415	HS-grad	32336
32540	5924	43	Private	104660	Bachelors	5924
32541	26266	49	Private	379779	Bachelors	26266
32542	18332	42	Local-gov	100793	Bachelors	18332
32543	13523	27	Private	153291	Prof-school	13523
32544	21024	43	Private	182437	Bachelors	21024
32545	13079	33	Private	159929	HS-grad	13079
32546	30953	34	Private	244064	HS-grad	30953
32547	547	35	Local-gov	233327	Some-college	547
32548	12956	18	Private	100875	11th	12956

```
32549   8140  27        Private  357348       HS-grad   8140
32550  19854  26        Private  248612     Bachelors  19854
32551  17139  33        Private  209317           9th  17139
32552   3604  42        Private  138634       HS-grad   3604
32553  32287  54      Local-gov   34832     Doctorate  32287
32554   3084  23        Private  113601  Some-college   3084
32555   9783  39        Private  179481       HS-grad   9783
32556   9947  25        Private  362826       HS-grad   9947
32557   3580  63        Private  106023       HS-grad   3580
32558  11556  39        Private  324231       HS-grad  11556
32559  20652  32    Federal-gov  115066       HS-grad  20652
32560  12557  34        Private  191834     Assoc-voc  12557

[32561 rows x 6 columns]
```

```
In [23]: from pandasql import sqldf
         pysqldf = lambda q: sqldf(q, globals())
```

```
In [39]: def concTime(dfs, length):
             start = time.time()
             result = pd.concat([dfs[0][:length], dfs[1][:length]], ignore_index=T
             end = time.time()
             result = result
             return end - start
```

```
In [37]: def joinTime(dfs, length):
             datae = dfs[0][:length]
             datad = dfs[1][:length]
             q = """
             SELECT *
             FROM (select * from datae) as e
             join (select * from datad) as d on d.indexx = e.indexx
             ;
             """
             start = time.time()
             result = sqldf(q, locals())
             end = time.time()
             return end - start
```

```
In [26]: def concdfTest(dfs, length):
             timeP = concTime(dfs,length)
             timeS = joinTime(dfs,length)
             return [timeP - timeS,timeP,timeS]
```

```
In [27]: def comparisonJoin():
             comp = (100,1000,2000,5000,10000,20000,30000)
             results = [[0,0,0,0]]
             for i in comp:
                 results += [concdfTest((dataLeft,dataRight),i)+[i]]
             return results[1:]
```

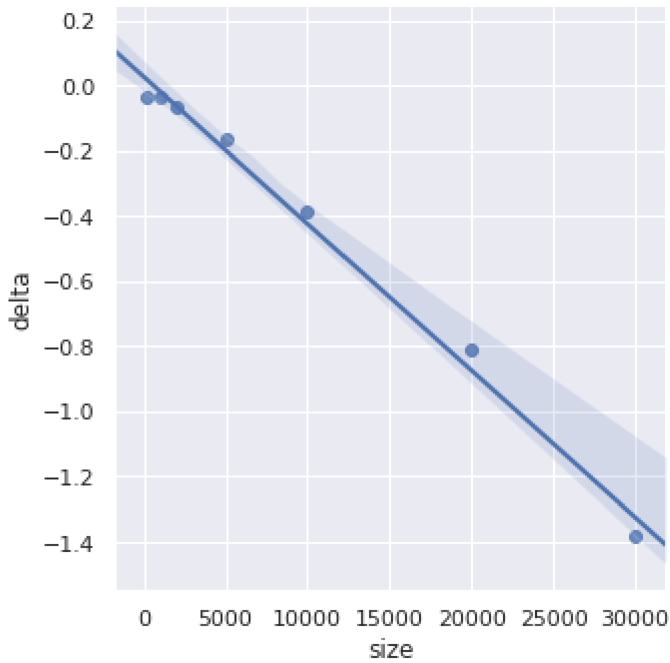
```
In [40]: dd = pd.DataFrame(comparisonJoin(), columns=['delta','concat','pandasql'
         dd
```

Out[40]:

	delta	concat	pandasql	size
0	-0.011437	0.009196	0.020633	100
1	-0.035377	0.011209	0.046585	1000
2	-0.059393	0.013589	0.072982	2000
3	-0.141453	0.021303	0.162756	5000
4	-0.298575	0.034997	0.333571	10000
5	-0.826091	0.059677	0.885768	20000
6	-1.418078	0.087115	1.505193	30000

In [29]:
```
import seaborn as sns; sns.set(color_codes=True)

g = sns.lmplot(x="size", y="delta", data=dd)
```



In [30]:
```
def aggTime(dfs, length):
    start = time.time()
    result = dfs[:length].groupby(by='sex').agg({'hours-per-week':'mean'
    end = time.time()
    result = result
    return end - start
```

In [31]:
```
def sqlTime(dfs, length):
    dfd = dfs[:length]
    q = """
    SELECT d.sex, AVG(d."hours-per-week") as avghoursperweek, AVG(d."edu
    FROM (select * from dfd) as d
    GROUP BY d.sex
    ;
    """
    start = time.time()
```

```
        result = sqldf(q, locals())
        end = time.time()
        return end - start
```

In [32]:
```
def aggTest(dfs, length):
    timeP = aggTime(dfs,length)
    timeS = sqlTime(dfs,length)
    return [timeP - timeS,timeP,timeS]
```

In [33]:
```
def comparisonAgg():
    comp = (100,1000,2000,5000,10000,20000,30000)
    results = [[0,0,0,0]]
    for i in comp:
        results += [aggTest(data,i)+[i]]
    return results[1:]
```

In [34]:
```
dd = pd.DataFrame(comparisonAgg(), columns=['delta','agg','pandasql','si
dd
```

Out[34]:

	delta	agg	pandasql	size
0	-0.011532	0.008934	0.020465	100
1	-0.034741	0.006961	0.041701	1000
2	-0.064239	0.008065	0.072304	2000
3	-0.141361	0.007666	0.149027	5000
4	-0.288953	0.007934	0.296887	10000
5	-0.657823	0.009386	0.667209	20000
6	-0.872482	0.010822	0.883304	30000

In [35]:
```
g = sns.lmplot(x="size", y="delta", data=dd)
```