Notebook

February 12, 2021

Question 1.a. To begin with, test whether players who play the guard position are paid the same as other players. Be sure to report the results of your test including the t-statistic and p-value.

This question is for your code, the next is for your explanation.

Hint: For those unfamiliar with American basketball, players are classified as playing one of three positions: guard, forward and center.

[4]:
```
nba_guard = nba[nba['guard'] == 1]
nba_not_guard =  nba[nba['guard'] == 0]

ttest_1a = stats.ttest_ind(nba_guard['wage'], nba_not_guard['wage'])

tstat_1a = ttest_1a.statistic
pval_1a = ttest_1a.pvalue

print("t-stat: {}".format(tstat_1a))
print("p-value: {}".format(pval_1a))
```

```
t-stat: -2.0530342128806582
p-value: 0.041043637620105405
```

Question 1.b. Explain.

If we test at a p level of .05, we can reject the null and assume that players who are guards are indeed paid less than those who are forwards and centers, by at least 2 t-stats (or roughly standard deviations).

Question 1.c. Do NBA players who complete college degree get paid more or less than those who do not? Test this hypothesis. Explain your results.

This question is for your code, the next is for your explanation.

Hint: Define a new variable degree to indicate whether the player completed 4 or more years of college.

[5]:
```
nba['degree'] = (nba['coll'] >= 4).astype(int)

ttest_1c = stats.ttest_ind(nba_degree['wage'], nba_no_degree['wage'])

tstat_1c = ttest_1c.statistic
pval_1c = ttest_1c.pvalue
```

```
print("t-stat: {}".format(tstat_1c))
print("p-value: {}".format(pval_1c))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-4245a30d6ac7> in <module>
      1 nba['degree'] = (nba['coll'] >= 4).astype(int)
      2
----> 3 ttest_1c = stats.ttest_ind(nba_degree['wage'], nba_no_degree['wage'])
      4
      5 tstat_1c = ttest_1c.statistic

NameError: name 'nba_degree' is not defined
```

Question 1.d. Explain.

If we are testing at a p level of .05, we reject the null, and accept the alternate hypothesis which tells us that those in the NBA who have a degree are more likely to be paid higher.

Question 1.e. Compute the *productivity* of each player in terms of the average number of points scored per minute of playing time. Note that the variable points is itself an average per game for the sampled season. Test whether guards are as productive as players who play other positions in this sense.

This question is for your code, the next is for your explanation.

[6]:
```
nba['productivity'] = nba['points']/nba['minutes']

nba_guard = nba[nba['guard'] == 1]
nba_not_guard =  nba[nba['guard'] == 0]

ttest_1e = stats.ttest_ind(nba_guard['wage'], nba_not_guard['wage'])

tstat_1e = ttest_1e.statistic
pval_1e = ttest_1e.pvalue

print("t-stat: {}".format(tstat_1e))
print("p-value: {}".format(pval_1e))
```

```
t-stat: -2.0530342128806582
p-value: 0.041043637620105405
```

Question 1.f. Explain.

At a p level of .05, we can reject the null and accept the alternate hypthesis. In terms of other players, guards tend to not to be as productive as other players in sense of scoring points, particularly since they prioritize more on defense over scoring points.

Question 1.g. Players do more on the court than just put the ball in the hoop. They also

rebound the ball and assist other players. Data on these two measures are given as a per-game average alongside points. Find the sample correlations between the three performance variables: points, rebounds, and assists.

Hint: The `.corr()` command is useful here.

[7]:
```
nba[['points', 'assists', 'rebounds']].corr()
```

[7]:

	points	assists	rebounds
points	1.000000	0.539269	0.563324
assists	0.539269	1.000000	0.059956
rebounds	0.563324	0.059956	1.000000

Question 1.h. To take all the performance measures into account, create a performance index as a weighted sum of the three measures: index = points + rebounds + 2 · assists. Using this index, test whether guards have the same performance as players at other positions.

This question is for your code, the next is for your explanation.

[8]:
```
nba['index'] = nba['points'] + nba['rebounds'] + 2 * nba["assists"]

nba_guard = nba[nba['guard'] == 1]
nba_not_guard =  nba[nba['guard'] == 0]

ttest_1h = stats.ttest_ind(nba_guard['index'], nba_not_guard['index'])

tstat_1h = ttest_1h.statistic
pval_1h = ttest_1h.pvalue

print("t-stat: {}".format(tstat_1h))
print("p-value: {}".format(pval_1h))
```

```
t-stat: 2.1909453801070837
p-value: 0.029320202223591406
```

Question 1.i. Explain.

Now we are still testing at a .05 p level, however this happens to be on the opposite end of the scale on the right end of the chart. This implies that they are more productive than other players on the team, mainly because the assists and rebounds tend to give more weight to the equation than that of nba players who are forwards or center, who primarily focus on scoring points.

Question 1.j. Finally, NBA general managers are very interested to know whether they are getting their money's worth, so want to know whether players are over or under paid given their performance. Compute a variable equal to the performance index per $1,000 of salary and again test whether guards are paid the same relative to performance as other positions.

This question is for your code, the next is for your explanation.

[9]:
```
nba['payoff'] = nba['index']/nba['wage']
```

```
nba_guard = nba[nba['guard'] == 1]
nba_not_guard =  nba[nba['guard'] == 0]

ttest_1j = stats.ttest_ind(nba_guard['payoff'], nba_not_guard['payoff'])

tstat_1j = ttest_1j.statistic
pval_1j = ttest_1j.pvalue

print("t-stat: {}".format(tstat_1j))
print("p-value: {}".format(pval_1j))
```

```
t-stat: 2.546161864697022
p-value: 0.011453727166139413
```

Question 1.k. Explain.

At a p level of .05, we can assume that they are getting their moneys worth, since productivity is high and wages are low. This tells us that general managers are generally paying more for forwards and centers than they do guards.

Question 2.a. Complete the following table of summary statistics. A code cell has been provided above for you to do your work for this question if you need it. An outline of the table is provided below. Replace the ... with actual numbers.

Hint: The command `.describe()` *will be useful. For example, if your dataset is called* `nba`*, the usage is* `nba.describe()`*.*

	Enrollment	Police	Crime
Number of observations	97	97	97
Sample mean	16076.35	20.49	394.45
Sample median	11990	16	187
Sample standard deviation	12298.98	15.63	460.78
Sample mean for public schools	17473.01	22.15	432.80
Sample mean for private schools	6183.33	8.75	122.83

Question 2.b. Compute the sample correlation between enrollment, police, and crime. Do the values you find make sense?

[12]:
```
crime[['enrollment', 'police', 'crime']].corr()
```

[12]:

	enrollment	police	crime
enrollment	1.000000	0.715053	0.836044
police	0.715053	1.000000	0.723310
crime	0.836044	0.723310	1.000000

Question 2.c. Do the values you find above make sense?

Yes they do. As enrollement goes up, this implies there are more individuals on campus so therefore police personelle and crime will go up. Police are also determined by how much crime there is as

well, so if there is more crime, the amount of police will go up to manage that.

Question 2.d. Test the hypothesis that the crime levels are the same in private and public schools by performing a t-test for equality of means of two subsamples. Is there a difference at the 5% significance level? At the 1% level? Do your conclusions depend on whether you assume the same and different variances for the two types of schools? Explain.

This question is for your code, the next is for your explanation.

[13]:
```
crime_public = crime[crime['private'] == 0]
crime_private = crime[crime['private'] == 1]

ttest_2d_unequal_var = stats.ttest_ind(crime_public['crime'],␣
↪crime_private['crime'], equal_var=False)

tstat_2d_unequal = ttest_2d_unequal_var.statistic
pval_2d_unequal = ttest_2d_unequal_var.pvalue

ttest_2d_equal_var = stats.ttest_ind(crime_public['crime'],␣
↪crime_private['crime'])

tstat_2d_equal = ttest_2d_equal_var.statistic
pval_2d_equal = ttest_2d_equal_var.pvalue

print("t-stat unequal variance: {}".format(tstat_2d_unequal))
print("p-value unequal variance: {}".format(pval_2d_unequal))
print("t-stat equal variance: {}".format(tstat_2d_equal))
print("p-value equal variance: {}".format(pval_2d_equal))
```

```
t-stat unequal variance: 4.8504897937789995
p-value unequal variance: 8.356792319851925e-06
t-stat equal variance: 2.225857620615201
p-value equal variance: 0.028388315564456677
```

Question 2.e. Explain.

With equal variance, we can see that we can reject the null that the crime is the same at a p level of .05, but cannot at a .01 level since it comes out to about .028 rate. Without equal variance, it is overwhelmingly different between the private and public schools at both a .05 and .01 significance level.

Question 2.f. Since it is likely that more crimes occur on bigger campuses, generate a new variable called "crimerate," defined as the number of crimes per 1,000 students. Test whether private and public schools have different crime rates (allowing for potentially unequal variances). Is there a difference at the 5% level? At the 1% level? Do your conclusions depend on whether you assume the same and different variances for the two types of schools? Explain.

This question is for your code, the next is for your explanation.

[14]:
```
crime['crimerate'] = crime['crime']/(crime['enrollment']/1000)

crime_public = crime[crime['private'] == 0]
crime_private = crime[crime['private'] == 1]

ttest_2f_unequal_var = stats.ttest_ind(crime_public['crimerate'],␣
↪crime_private['crimerate'], equal_var=False)

tstat_2f_unequal = ttest_2f_unequal_var.statistic
pval_2f_unequal = ttest_2f_unequal_var.pvalue

ttest_2f_equal_var = stats.ttest_ind(crime_public['crimerate'],␣
↪crime_private['crimerate'])

tstat_2f_equal = ttest_2f_equal_var.statistic
pval_2f_equal = ttest_2f_equal_var.pvalue

print("t-stat unequal variance: {}".format(tstat_2f_unequal))
print("p-value unequal variance: {}".format(pval_2f_unequal))
print("t-stat equal variance: {}".format(tstat_2f_equal))
print("p-value equal variance: {}".format(pval_2f_equal))
```

```
t-stat unequal variance: 0.19395693977979903
p-value unequal variance: 0.849083848844537
t-stat equal variance: 0.21220073218647473
p-value equal variance: 0.8324050631018755
```

Question 2.g. Explain.

At either .05 significance level or .01 signifance level, we can safely assume that the crime rates at both public and private schools are roughly the same and do not vary as much. We can deduce this from the extemely high p level that we are given as well as how close the test stat is to the test mean.

Question 3.a. Create another table with the joint and marginal probabilities associated with this sample. Create another table with the conditional distribution of employment status given whether or not the resident has returned to their home. An outline of the tables is provided below. Replace the ... with actual numbers. Make sure to show your work for how you derived the tables!

You may either: * Write your math in the cell below using LaTeX typesetting (recommended). * Write your math on a tablet and include the exported pdf with the pdf submission of this assignment. * Write your math on paper and include a scan with the pdf submission of this assignment.

Joint	Employed	Unemployed	Marginal
Returned to pre-storm address	0.56	.03	.59
Have not yet returned	0.32	.09	.41
Marginal	0.88	.12	1

Conditional	Employed	Unemployed
Given returned to pre-storm address	.67	.25
Given have not yet returned	.36	.75

Type your answer here, replacing this text.

Question 3.b. Using this last table find the expectation of a resident being employed, conditional on returning to their home. To do this, assign values to the two variables: 1 = returned to home and 0 = did not; 1 = employed and 0 = unemployed. Using the same table, confirm the law of iterated expectations.

You may either: * Write your math in the cell below using LaTeX typesetting (recommended). * Write your math on a tablet and include the exported pdf with the pdf submission of this assignment. * Write your math on paper and include a scan with the pdf submission of this assignment.

$ E(3b) = .67 * 1 + .25 * 0 = .67 $

Question 3.c. Compute the sample covariance of return status and employment status.

You may either: * Write your math in the cell below using LaTeX typesetting (recommended). * Write your math on a tablet and include the exported pdf with the pdf submission of this assignment. * Write your math on paper and include a scan with the pdf submission of this assignment.

$\frac{1}{n-1}\sum_n^1(x_i-\bar{x})(y_i-\bar{y}) = \frac{1}{249-1}((0.56-.59)(0.32-.41)+(.03-.59)(.09-.41) = \frac{1}{248}(.0027+.1792) = 0.00083$

Question 3.d. Is current employment status statistically independent of the return status? Along with part 3.c, what does this say about the relationship between return status and employment status? Justify your answer.

No, it seems that it is more likely for one to return to their original residence if they are employed rather than unemployed.

Question 3.e. Give two plausible reasons that could explain the difference in employment status between the residents who returned to their homes and those who did not.

Some reasons may include that damages are far too great to be able to pay and repair for

Question 4.a. The variable `oecd` is a dummy indicator of each country's membership in the Organization for Economic Cooperation and Development (OECD): 1 = a member in OECD, 0 = not a member. The OECD consists of several dozen of the largest, most developed economies in the world. Compare the sample mean and standard deviation of per-capita GDP between the OECD and non-OECD countries. Do the same with per-capita CO2 emissions. A code cell has been provided for you above to do your work for this question.

Compare the sample mean and standard deviation of per-capita GDP between the OECD and non-OECD countries. Between OECD and non-OECD countries, the population means seem to be fairly close together however the standard deviation of non-OECD countries is twice that of OECD countries. We can assume this is the case since most larger and more developed countries that are involved tend to have more consistently higher populations.

Do the same with per-capita CO2 emissions. The per capita CO2 emmisions of non-OECD counties are more less than half that of OECD countries (4.07 vs 8.98, respectively) and the standard deviations are a little more extreme on the non-OECD countries, however this is expected since non-OECD countries have a higher count of those in the sample, so it can easily be skewed if there are a couple outliers. The reason that the emissions are more in OECD countries are most likely due to industrialism and the use of cars which increases CO2 emmisions.

Question 4.b. Conduct a t-test of whether the sample means of CO2 emissions per-capita are significantly different between the OECD and non-OECD countries. Did you choose to assume variances of the two groups were equal or unequal? Explain why.

This question is for your code, the next is for your explanation.

[18]:
```
pollution_oecd = pollution[pollution['oecd'] == 1]
pollution_no_oecd = pollution[pollution['oecd'] == 0]

ttest_4b = stats.ttest_ind(pollution_oecd['co2pc'], pollution_no_oecd['co2pc'],␣
 ↪equal_var=False, nan_policy='omit')

tstat_4b = ttest_4b.statistic
pval_4b = ttest_4b.pvalue

print("t-stat: {}".format(tstat_4b))
print("p-value: {}".format(pval_4b))
```

```
t-stat: 5.682465161974535
p-value: 2.6835812931694e-07
```

Question 4.c. Explain.

According to what we have been given, at a .01 confidence interval, I can confirm that the non-OECD countries CO2 per capita emissions is signficantly less than that of OECD countries. The reason that the emissions are more in OECD countries are most likely due to industrialism and the use of cars which increases CO2 emmisions.

Question 4.d. Approximate the growth rates of GDP and CO2 by first generating variables that are the natural logarithms of the two variables. Why would we examine the growth rates instead of the absolute levels of emissions and GDP? The function `np.log()` will be helpful. It can take a column as an argument in the parentheses. For example, if we wanted to take the natural log of a number instead of a column we would do `np.log(10)`. `np` is the shortcut for `numpy`, which is another useful package for doing math.

This question is for your code, the next is for your explanation.

[19]:
```
pollution['log_gdp'] = np.log(pollution['gdp'])
pollution['log_co2'] = np.log(pollution['co2'])

pollution.head()
```

[19]:

	year	countryname	countrycode	gdp	gdppc	co2	\
0	2010	Zambia	ZMB	9.799629e+09	741.4421	2427.554	
1	2010	French Polynesia	PYF	NaN	NaN	883.747	
2	2010	Monaco	MCO	NaN	NaN	NaN	
3	2010	Ukraine	UKR	9.057726e+10	1974.6212	304804.720	
4	2010	Venezuela, RB	VEN	1.750000e+11	6010.0270	201747.340	

	co2pc	population	oecd	log_gdp	log_co2
0	0.183669	13216985	0.0	23.005610	7.794639
1	3.296764	268065	0.0	NaN	6.784171
2	NaN	36845	0.0	NaN	NaN
3	6.644867	45870700	0.0	25.229469	12.627427
4	6.946437	29043283	0.0	25.888052	12.214771

Question 4.e. Explain.

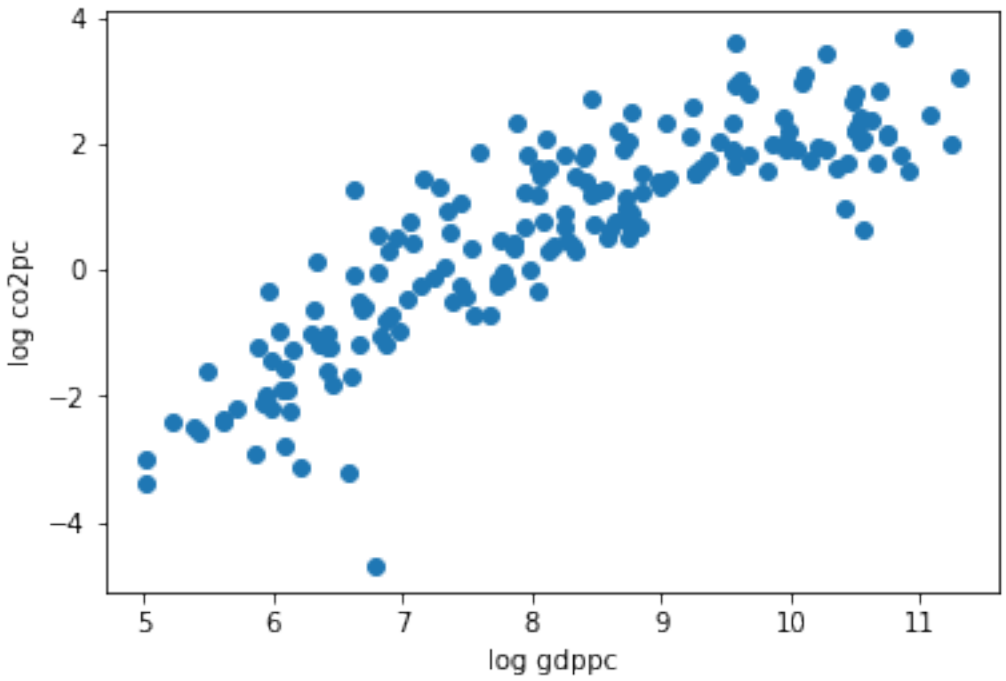
By using log, we can put it in terms that make it easier to identify growth and a linear trend if we needed to.

Question 4.f. Some countries with relatively high GDP might argue that their large population size – rather than carbon-intensive technology – drives high emissions. As a quick test on this claim, draw a scatterplot with the growth rates of per-capita GDP on the x-axis and the growth rates of per-capita emissions on the y-axis. You will need to generate these new variables/columns. To generate the plot you can mimic the code above but with your new variables in the right parts. Compare the resulting figure with the previous figure. Do you think the claim of the high-GDP countries is convincing? Explain.

This question is for your code, the next is for your explanation.

[21]:
```
pollution['log_gdppc'] = np.log(pollution['gdppc'])
pollution['log_co2pc'] = np.log(pollution['co2pc'])

plt.scatter(pollution['log_gdppc'], pollution['log_co2pc'])
plt.xlabel("log gdppc")
plt.ylabel("log co2pc");
```



Question 4.g. Explain.

Now that we have put the GDP and CO2 emissions per capita and taken the log of that to establish a linear trend, we can see that it is true that regardless of the population size (as per capita standardizes all of them), carbon intensive technology is far more likely to blame rather than large population size, since the per capita CO2 is high to begin with.

Question 5.a. Display summary statistics for this dataset that include at least the mean and interquartile range for each variable. You learned a command earlier in this assignment that does this.

[23]:
```
la.describe()
```

[23]:

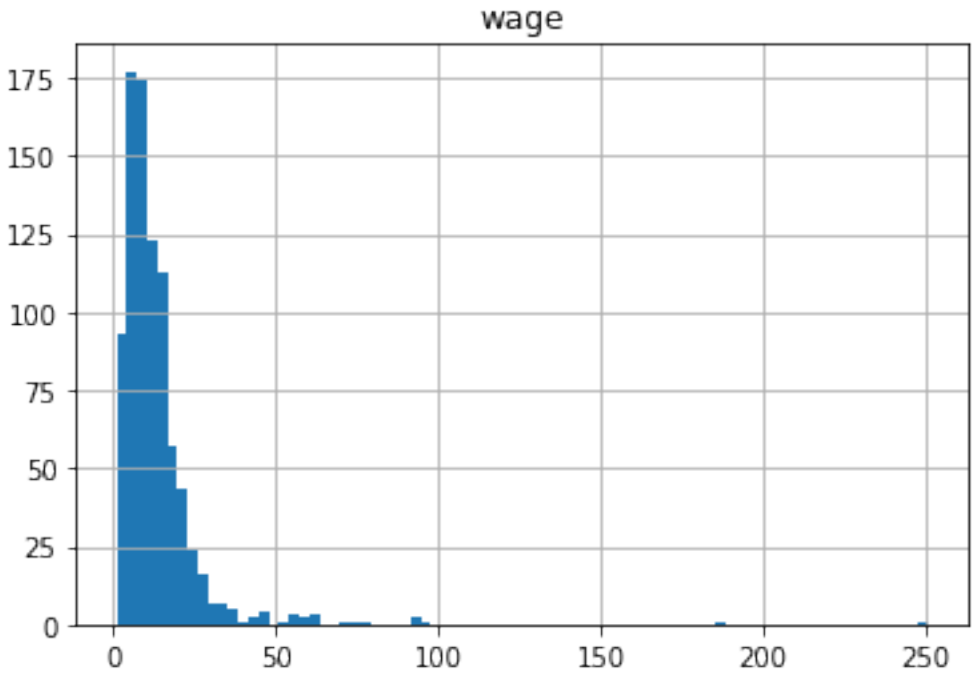
	hispanic	citizen	black	exp	wage	female	education
count	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000
mean	0.367323	0.717265	0.088065	12.494044	13.413861	0.424102	12.864426
std	0.482355	0.450590	0.283554	1.605983	14.659279	0.494492	
min	0.000000	0.000000	0.000000	10.000000	1.250000	0.000000	
25%	0.000000	0.000000	0.000000	11.000000	6.500000	0.000000	
50%	0.000000	1.000000	0.000000	12.700000	10.576923	0.000000	
75%	1.000000	1.000000	0.000000	14.000000	15.723952	1.000000	
max	1.000000	1.000000	1.000000	15.000000	250.661540	1.000000	

```
std      3.194241
min      0.000000
25%     12.000000
50%     13.000000
75%     16.000000
max     20.000000
```

Question 5.b. Display a histogram *with 80 bins* of the wage variable. The histogram must have 80 bins. This is the documentation for the `.hist()` command for `pandas`, which may be helpful. The documentation for the `pandas` library is excellent and you are highly encouraged to reference it throughout the course.

Hint: If you wanted to generate a 40-bin histogram for the ***exp*** *variable, the command would look like* `la.hist(column='exp', bins=40)`.

[24]:
```
la.hist(column='wage', bins=80);
```



Question 5.c. Is the distribution skewed in any way? You don't have to code for this part if you don't want to, a qualitative description is enough, but you can check quantitatively by doing `la.skew()`. If you do want to try this, use the code cell provided.

Yes, the data is exremely skewed. The value the skew function gives us for wage is 8.19 which is significantly higher than the rest of the values. Visually we can also confirm the skewness of the graph as it is right skewed.

Question 5.d. Wages and earnings are often studied by first taking logarithms. Transform wage by taking its natural logarithm.

[26]:
```
la['log_wage'] = np.log(la['wage'])

la.head()
```

[26]:

	hispanic	citizen	black	exp	wage	female	education	log_wage
0	1	1	0	14.0	5.288462	1	9	1.665527
1	0	1	0	14.7	8.461538	1	13	2.135531
2	0	1	0	14.7	10.416667	1	13	2.343407
3	0	1	0	14.0	21.634615	1	14	3.074295
4	1	0	0	12.0	3.365385	1	12	1.213542

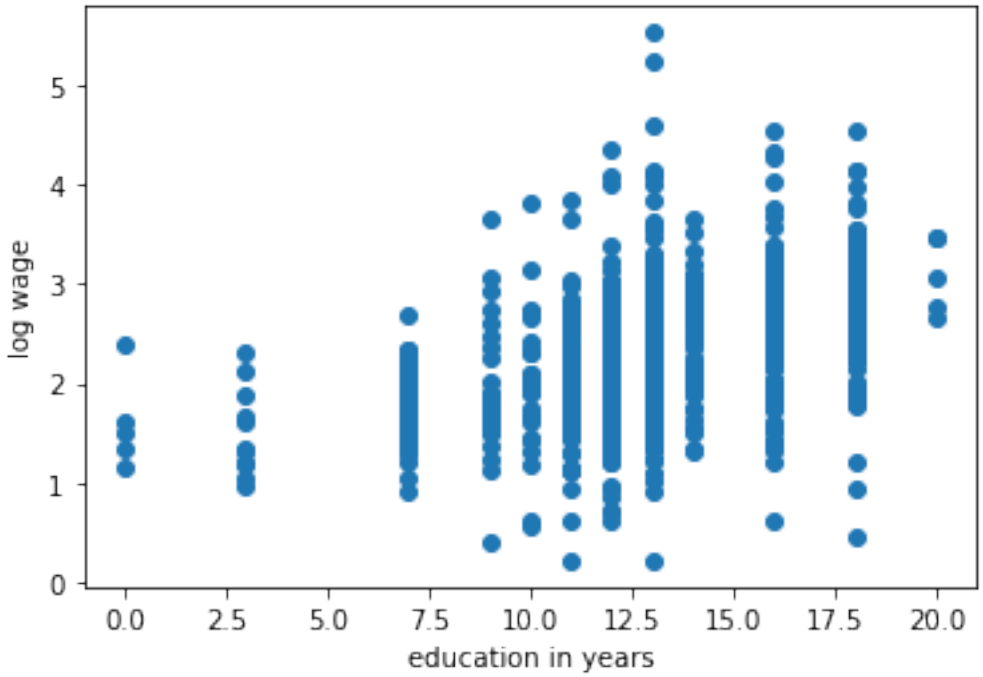
Question 5.e. Compute the frequencies of years of schooling (education). The `.value_counts()` method will be useful. Here is the documentation. For the `subset` parameter, you will want to set that equal to `'education'` (with the quote marks on either side; if you don't know why we need the quote marks, it might be worthwhile revisiting the Python assignment). You will also probably want to set the `sort` parameter equal to `False` because otherwise it will sort by count instead of education level, which might not be as helpful in this context. Set `normalize` to `True`. This will give proportions instead of counts. You don't have to do specify anything for the `ascending` parameter, it will default to `False` if you don't tell `pandas` what to do there.

[27]:
```
la.value_counts(subset = 'education', sort = False, normalize = True)
```

[27]:
```
education
0     0.005794
3     0.012746
7     0.068366
9     0.033604
10    0.030127
11    0.086906
12    0.201622
13    0.224797
14    0.063731
16    0.179606
18    0.086906
20    0.005794
dtype: float64
```

Question 5.f. Construct a scatterplot of log wages and years of schooling, with schooling on the x-axis and wages on the y-axis. Refer to how we used `plt.scatter()` earlier in the problem set if you don't recall how to make a scatterplot. Make sure to label your axes!

[28]:
```
plt.scatter(la['education'], la['log_wage'])
plt.xlabel("education in years")
plt.ylabel("log wage");
```



0.1 Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

[ ]:
```
# Save your notebook first, then run this cell to export your submission.
grader.to_pdf(pagebreaks=False, display_link=True)
```