Notebook

February 9, 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.

Based on the t-stat, the observed value of the wage is deviating about 2.1 standard errors away from the expected value if guards were paid the same wage. The p-value suggests that NBA players in the guard position are unlikely to be paid the same.

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

nba_degree = nba[nba['degree'] == True]
nba_no_degree = nba[nba['degree'] == False]
```

```
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))
```

```
t-stat: -2.3705333284217867
p-value: 0.0184730432847849
```

Question 1.d. Explain.

Based on the data given, it suggests a negative association between college degree and wages. The t-stat shows that the observed value of the wage for NBA players with a college degree deviates about 2.4 standard errors away. The p-value infers that NBA players with a college degree are unlikely to be paid the same.

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['productivity'],␣
 ↪nba_not_guard['productivity'])

tstat_1e = ttest_1e.statistic
pval_1e = ttest_1e.pvalue

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

```
t-stat: -1.7767355680677752
p-value: 0.07675030716366808
```

Question 1.f. Explain.

The t-stat shows that the observed productivity is deviating about 1.8 standard errors away from the expected productivity if we assume that they have the same productivity as other positions. Based on the p-value, we cannot reject the notion that guards have the same productivity as players who play other positions.

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.

According to the metric index, NBA guards display a higher performance index than players at other positions. The t-stat shows that the observed performance index is deviating about 2.2 standard errors away from the expected performance index assuming that other players have the same performance index. The p-value infers that the performance index of guards is unlikely to be the same as players from other positions.

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.

Given the data, NBA guards have a better payoff compared to players from other positions. The t-stat shows that the observed payoff is deviating about 2.5 standard errors away from the expected payoff assuming that players from other positions have the same payoff. The p-value infers that guards are under paid based on their performance index compared to players from other positions.

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()`*.*

[12]:
```
private_school = crime[crime['private'] == 1].describe()
private_school
```

[12]:

	enrollment	private	police	crime
count	12.000000	12.0	12.000000	12.000000
mean	6183.333333	1.0	8.750000	122.833333
std	3347.160678	0.0	6.538348	129.444150
min	1799.000000	1.0	2.000000	15.000000
25%	4174.750000	1.0	5.000000	43.750000
50%	5073.000000	1.0	7.000000	63.500000
75%	7975.750000	1.0	9.000000	161.500000
max	13570.000000	1.0	26.000000	426.000000

[13]:
```
public_school = crime[crime['private'] == 0].describe()
public_school
```

[13]:

	enrollment	private	police	crime
count	85.000000	85.0	85.000000	85.000000
mean	17473.011765	0.0	22.152941	432.800000

```
std     12468.025117      0.0   15.847116   477.945225
min      1859.000000      0.0    1.000000     1.000000
25%      7758.000000      0.0   10.000000   106.000000
50%     15669.000000      0.0   18.000000   239.000000
75%     24011.000000      0.0   28.000000   525.000000
max     56350.000000      0.0   74.000000  2052.000000
```

	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	12298.99	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?

[14]:
```
crime[['enrollment', 'police', 'crime']].corr()
```

[14]:
```
            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?

The strong correlation between number of enrolled students and the number of police employed makes sense because you need more policemen for a larger population of enrolled students. The strong correlation between enrollment and crime could makes sense because a larger population is located in a larger city, which would have higher crime rates. The strong correlation between police and crime because of the third variable of enrollment having a correlation with both policemen and crime.

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.

[15]:
```
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.

Based on the data, it rejects that the crime levels are the same in private and public schools. If we assume that the variance is the same between the two types of schools, then we reject at the 5 percent that the crime levels are the same in private and public schools. If the variance is different between the two schools, then we reject at the 1 percent that the crime levels are the same in private and public schools. Looking at the two equations for finding the t-stat with unequal and equal variance, the we see that equal variance uses population standard deviation and unequal variance uses sample standard deviation.

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.

[16]:
```
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.

Based on the data, we cannot reject that private and public schools the same crime rates. There is insignificant difference whether or not we assume that the variances are equal between public and private schools.

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	0.558	0.032	0.590
Have not yet returned	0.317	0.092	0.410
Marginal	0.876	0.124	1

Conditional	Employed	Unemployed
Given returned to pre-storm address	0.946	0.054
Given have not yet returned	0.775	0.225

Joint PD (Employed and Returned to pre-storm address) = (139/249) = 0.5582329317269076
Joint PD (Employed and Have not yet returned) = (79/249) = 0.3172690763052209
Joint PD (Unemployed and Returned to pre-storm address) = (8/249) = 0.0321285140562249
Joint PD ((Unemployed and Have not yet returned) = (23/249) = 0.09236947791164658

Marginal PD (employed) = 218/249 = (0.87550201)
Marginal PD (unemployed) = 31/249 = (0.12449799)
Marginal PD (returned to pre-storm address) = (147/249) = (0.5903614457831325)

Marginal PD (have not yet returned) = (102/249) = (0.40963855421686746)

Conditional PD (Employed | returned to pre-storm) address = 139/147 = 0.9455782312925171
Conditional PD (Employed | have not yet returned) = 79/102 = 0.7745098039215687
Conditional PD (Unemployed | returned to pre-storm address) = 8/147 = 0.05442176870748299
Conditional PD (Unemployed | have not yet returned) = 23/102 = 0.22549019607843138

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.

X = returned to home | X = 0 - did not return | X = 1 - returned
Y = employed | Y = 0 - unemployed | Y = 1 - employed

We first calculate the probability of being employed, conditioned with returning to their home:

$$\frac{0.558}{0.590} = 0.946 \tag{1}$$

with 0.558 being the joint probability of being employed and returned to pre-storm address and 0.590 being the marginal probability of returning to pre-storm address.

Next, we calculate the probability of being unemployed, conditioned with not returning to their home:

$$\frac{0.317}{0.410} = 0.775 \tag{2}$$

with 0.317 being the joint probability of being unemployed and have not yet returned and 0.410 being the marginal probability having not yet returned.

The law of iterexted expectation is:

$$E[Y] = E_X[E_Y[Y|X] \tag{3}$$

Using the table above from **Question 3.a**, we can confirm the law of iterated expectation

$$\mathbb{E}[Y|X=1] = \mathbb{P}(Y=1|X=1)*1 + \mathbb{P}(Y=0|X=1)*0 = 0.946 \tag{4}$$

$$\mathbb{E}[Y|X=0] = \mathbb{P}(Y=1|X=0)*1 + \mathbb{P}(Y=0|X=0)*0 = 0.775 \tag{5}$$

$$\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{P}(X=1)*\mathbb{E}[Y|X=1] + \mathbb{P}(X=0)*\mathbb{E}[Y|X=0] = 0.590*0.946 + 0.410*0.775 = 0.876 \tag{6}$$

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.

$$S_{XY} = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y}) \tag{7}$$

$$Cov(X, Y) = E(XY) - E(X) * E(Y) \tag{8}$$

$$E(XY) = (XY) * P(XY) = 1 * 0.558 = 0.558 \tag{9}$$

$$E(X) = \sum P(X) * X^2 = 1 * 0.876 + 0 * 0.124 = 0.876 \tag{10}$$

$$E(Y) = \sum P(Y) * Y^2 = 1 * 0.590 + 0 * 0.410 = 0.590 \tag{11}$$

$$E(XY) - E(X) * E(Y) = 0.558 - 0.876 * 0.590 = 0.0415 \tag{12}$$

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.

Current employment status would be dependent on the return status because the probability of being employed is different for those who have returned home and those who have not returned. The small value of our covariance suggests that there is a weak relationship between return status and employment status.

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.

1.)Those who returned home are likely wealthier, and thus may have more marketable skills in terms of seeking employment, therefore a higher rate of employment. In contrast, those that are less educated are more likely to have trouble finding a job.

2.) The people who returned to their homes have a higher employment rate because they likely returned to their previous work place where they were already employed, whereas those who did not return home were forced to find new jobs.

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.

The mean gdppc for the oecd countries is much higher than the mean for non-oecd countries. Likewise, the gdppc standard deviation for oecd countries is much higher than the standard deviation for non-oecd countries. In contrast, the mean CO2 is significantly lower in the oecd countries than the non-oecd countries, and the standard deviation of CO2 is lower for oecd countries is slightly lower than the non-oecd countries.

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.

[20]:
```
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.

We assume that the variance is not equal since the standard deviation squared for oecd and non-oecd countries are significantly different. With the t-stat, we reject the null hypothesis, that the CO2 emissions per-capita are the same between oecd and non-oecd countries.

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.

[21]:
```
pollution['log_gdp'] = np.log(pollution['gdp'])
pollution['log_co2'] = np.log(pollution['co2'])

pollution.head()
```

[21]:

	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.

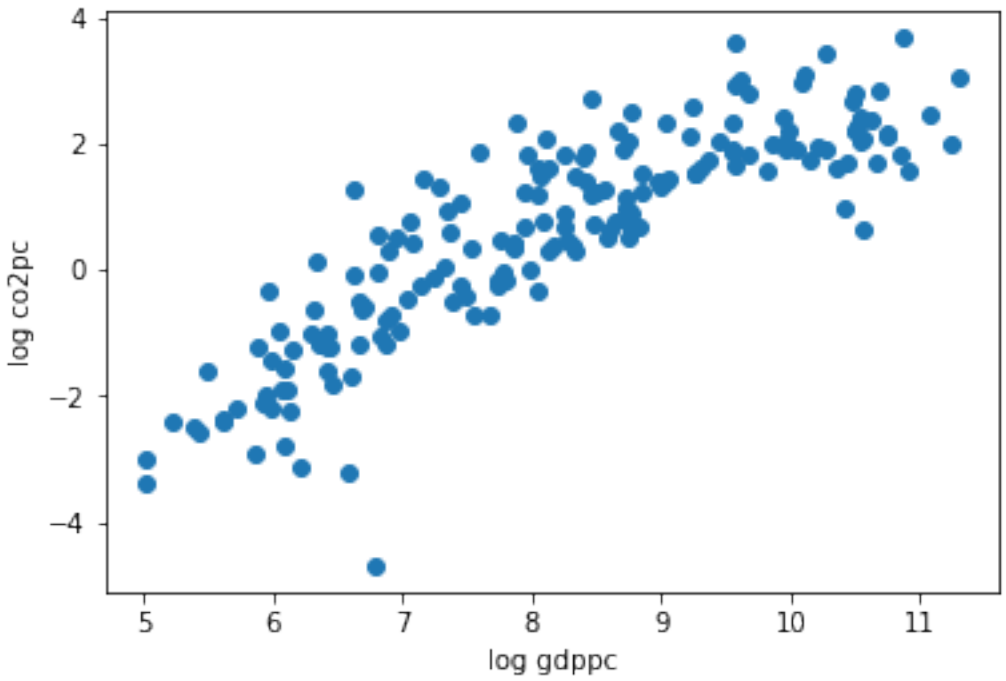
Looking at the growth rates of gdp and co2, we want to estimate these variables because it gives us a better representation of the data, which helps us make better conclusions. If we were only looking at the absolute levels, our data would be very clumped together and would poorly portray the emissions and GDP from country to country due to the extreme data values.

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.

[23]:
```
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.

Comparing the two graphs, both suggests that as gdp increases, co2 also increases. However, the second graph displays that regardless of a larger population, the higher the gdp per capita, the higher the co2 emissions. This disproves the suggestion that some countries with high gdp are producing more co2 due to a larger population size rather than carbon intensive technology.

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.

[25]:
```
la.describe()
```

[25]:
```
          hispanic     citizen       black         exp        wage      female  \
count   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
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

        education
count  863.000000
mean    12.864426
```

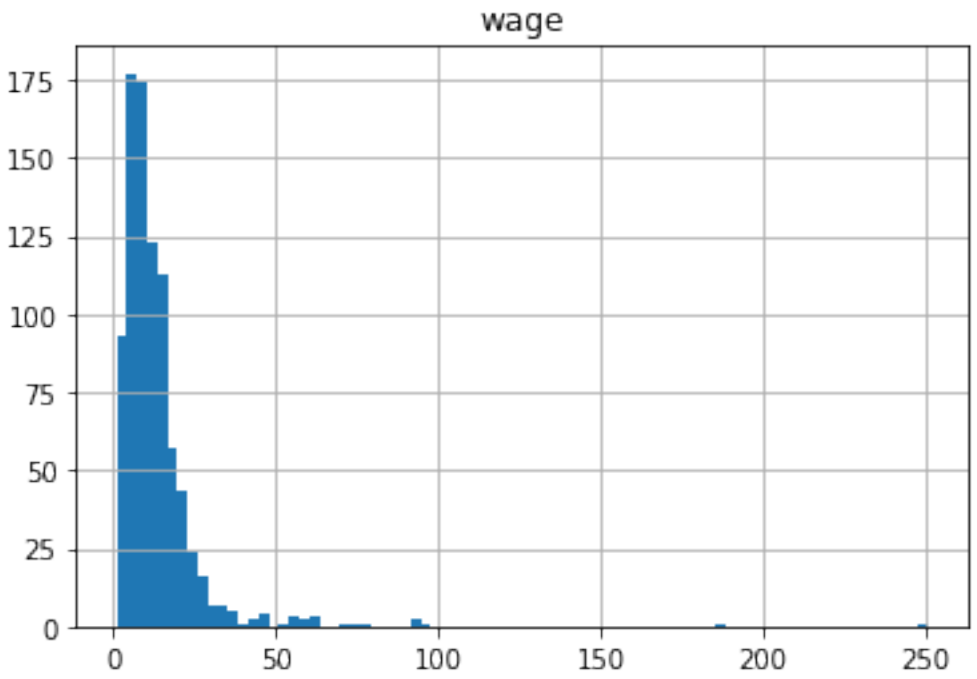
```
std      3.194241
min      0.000000
25%     12.000000
50%     13.000000
75%     16.000000
max     20.000000
```

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)`.

[26]:
```
la.hist(column='wage',bins=80)
```

[26]:
```
array([[<AxesSubplot:title={'center':'wage'}>]], dtype=object)
```



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.

The distribution is skewed to the right.

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

[28]:
```
la['log_wage'] = np.log(la['wage'])
la.head()
```

[28]:

	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

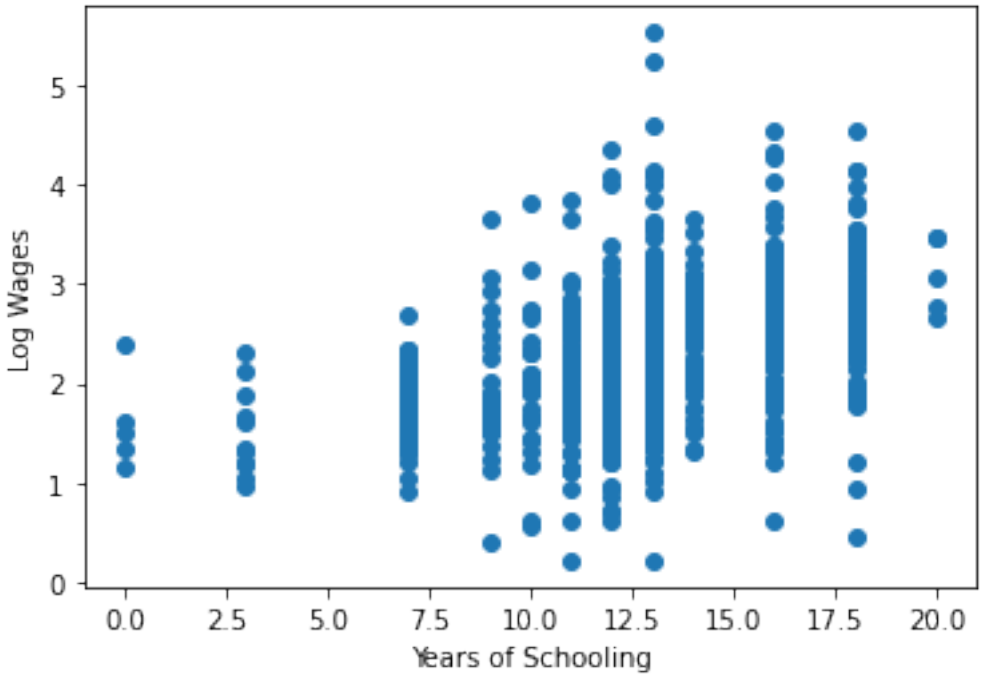
Question 5.e. Compute the frequencies of years of schooling (education). The `.value_counts()` method will be useful. Here is the documentation. For the `subset` parameter, you will want to set that equal to `'education'` (with the quote marks on either side; if you don't know why we need the quote marks, 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.

[29]:
```
la['education'].value_counts(normalize=True,sort=False)
```

[29]:
```
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
Name: education, 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!

[30]:
```
plt.scatter(la['education'],la['log_wage'])
plt.xlabel("Years of Schooling")
plt.ylabel("Log Wages");
```



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)
```