MODS PROJECT

Imports

In [177…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from scipy.stats import f
from scipy.stats import spearmanr
from scipy.stats import norm
```

In [178…

```
df0 = pd.read_csv("MROZ.txt", delim_whitespace=True, header=None)
```

Naming columns

In [179…

```
df0 = df0.rename(columns={0: 'laborforce'
                          ,1: "hours worked"
                          , 2: "kids below 6"
                          ,3: 'kids between 6 and 18'
                          , 4: 'age women'
                          , 5: 'educ years of schooling'
                          , 6: 'wage estimated'
                          , 7: 'husbandWage'
                          , 8: 'hours worked by husband'
                          , 9: 'husband age'
                          , 10: 'huband year of schooling'
                          , 11: 'husbands hourly wage'
                          , 12: 'family income'
                          , 13: 'marginal tax rate facing woman'
                          , 14:'mothers year of schooling'
                          , 15: 'fathers year of schooling'
                          , 16:'rate in country of resid'
                          , 17: 'live in SMSA'
                          , 18: 'marginal tax rate facing woman'
                          , 19: 'nwifeinc'
                          , 20: 'log(wage)'
                          , 21: 'expersq'})
df0
```

Out[179…

	laborforce	hours worked	kids below 6	kids between 6 and 18	age women	educ years of schooling	wage estimated	husbandWage	hours worked by husband	husband age	...	family income	marginal tax rate facing woman	mothers year of schooling	fathers year of schooling	rate in country of resid	live in SMSA	marginal tax rate facing woman	nwifeinc	log(wage)	expersq
0	1	1610	1	0	32	12	3.354	2.65	2708	34	...	16310	0.7215	12	7	5.0	0	14	10.910060	1.210154	196
1	1	1656	0	2	30	12	1.3889	2.65	2310	30	...	21800	0.6615	7	7	11.0	1	5	19.499980	.3285121	25
2	1	1980	1	3	35	12	4.5455	4.04	3072	40	...	21040	0.6915	12	7	5.0	0	15	12.039910	1.514138	225
3	1	456	0	3	34	12	1.0965	3.25	1920	53	...	7300	0.7815	7	7	5.0	0	6	6.799996	.0921233	36
4	1	1568	1	2	31	14	4.5918	3.60	2000	32	...	27300	0.6215	12	14	9.5	1	7	20.100060	1.524272	49
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
748	0	0	0	2	40	13	.	0.00	3020	43	...	28200	0.6215	10	10	9.5	1	5	28.200000	.	25
749	0	0	2	3	31	12	.	0.00	2056	33	...	10000	0.7715	12	12	7.5	0	14	10.000000	.	196
750	0	0	0	0	43	12	.	0.00	2383	43	...	9952	0.7515	10	3	7.5	0	4	9.952000	.	16
751	0	0	0	0	60	12	.	0.00	1705	55	...	24984	0.6215	12	12	14.0	1	15	24.984000	.	225
752	0	0	0	3	39	9	.	0.00	3120	48	...	28363	0.6915	7	7	11.0	1	12	28.363000	.	144

753 rows × 22 columns

Question 1

Let's transform "." into real values

In [180…

```
s,df = df0['wage estimated'],df0.copy()
empty_rows = s[s=="." ].index
df = df.drop(empty_rows)
df0=df.copy()
```

In [181…

```
print("We removed ",len(df0)-len(df)," where the wage data was null or negative")
```

```
We removed  0  where the wage data was null or negative
```

Question 2

For all women:

In [182…

```
df['wage estimated']= pd.to_numeric(df['wage estimated'])
print('The statistics of women salaries whose husband earn more the median are:')
df[['wage estimated', 'age women', 'educ years of schooling']].describe()
```

```
The statistics of women salaries whose husband earn more the median are:
```

Out[182…

	wage estimated	age women	educ years of schooling
count	428.000000	428.000000	428.000000
mean	4.177682	41.971963	12.658879
std	3.310282	7.721084	2.285376
min	0.128200	30.000000	5.000000
25%	2.262600	35.000000	12.000000
50%	3.481900	42.000000	12.000000
75%	4.970750	47.250000	14.000000
max	25.000000	60.000000	17.000000

For women whose husband earn more than the median wage:

In [183…

```
print('The statistics of women salaries whose husband earn more the median are:')
df[['wage estimated', 'age women', 'educ years of schooling']][df['husbandWage']> np.median(df['husbandWage'])].describe()
```

```
The statistics of women salaries whose husband earn more the median are:
```

Out[183…

	wage estimated	age women	educ years of schooling
count	214.000000	214.000000	214.000000
mean	5.232559	41.841121	13.135514
std	2.937990	7.338431	2.238240
min	0.382600	30.000000	7.000000
25%	3.576825	35.000000	12.000000
50%	4.543150	43.000000	12.000000
75%	5.863100	47.000000	15.000000
max	25.000000	58.000000	17.000000

For women whose husband earn more than the median wage:

In [184…

```
print('The statistics of women salaries whose husband earn more the median are:')
df[['wage estimated', 'age women', 'educ years of schooling']][df['husbandWage']<=np.median(df['husbandWage'])].describe()
```

```
The statistics of women salaries whose husband earn more the median are:
```

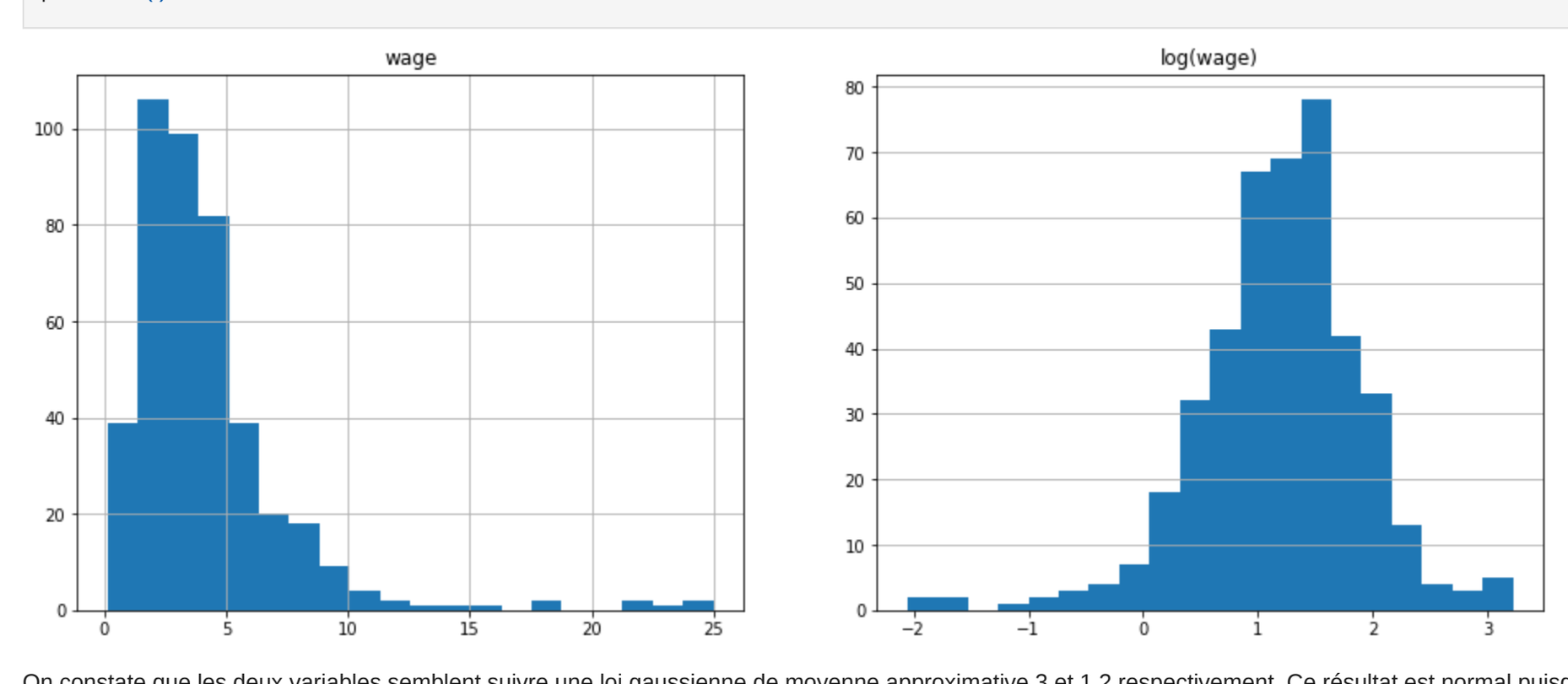
Out[184…

	wage estimated	age women	educ years of schooling
count	214.000000	214.000000	214.000000
mean	3.122804	42.102804	12.182243
std	3.331609	8.100814	2.237005
min	0.128200	30.000000	5.000000
25%	1.697225	35.000000	12.000000
50%	2.369150	41.000000	12.000000
75%	3.328950	48.000000	13.000000
max	25.000000	60.000000	17.000000

Question 3

In [185…

```
plt.rcParams["figure.figsize"] = (16, 6)

plt.subplot(1,2,1)
plt.hist(df['wage estimated'],20)
plt.title('wage')
plt.grid()

plt.subplot(1,2,2)
df['log(wage)']= np.log(df['wage estimated'])
plt.hist(df['log(wage)'],20)
plt.title('log(wage)')
plt.grid(axis='y')
plt.show()
```



On constate que les deux variables semblent suivre une loi gaussienne de moyenne approximative 3 et 1.2 respectivement. Ce résultat est normal puisque tout échantillon de donné converge vers une telle loi lorsque n devient assez grand.

Question 4

In [186…

```
motheduc = df['mothers year of schooling']
fatheduc = df['fathers year of schooling']
print('La correlation entre ces deux variables est: \n', spearmanr(df['mothers year of schooling'], df['fathers year of schooling'])[0])
print('La p-value de rejet \n', spearmanr(df['mothers year of schooling'], df['fathers year of schooling'])[1])
```

```
La correlation entre ces deux variables est:
 0.550732782239588
La p-value de rejet
 2.580005695516474e-35
```

On remarque une corrélation assez forte, ou du moins non-négligeable. Cela peut s'expliquer par le fait que les membres d'un couple sont souvents issus d'un niveau d'études assez similaire à cause de la non-mixité des classes sociales.

Cette corrélation peut donc introduire un biais de multicolinéarité.

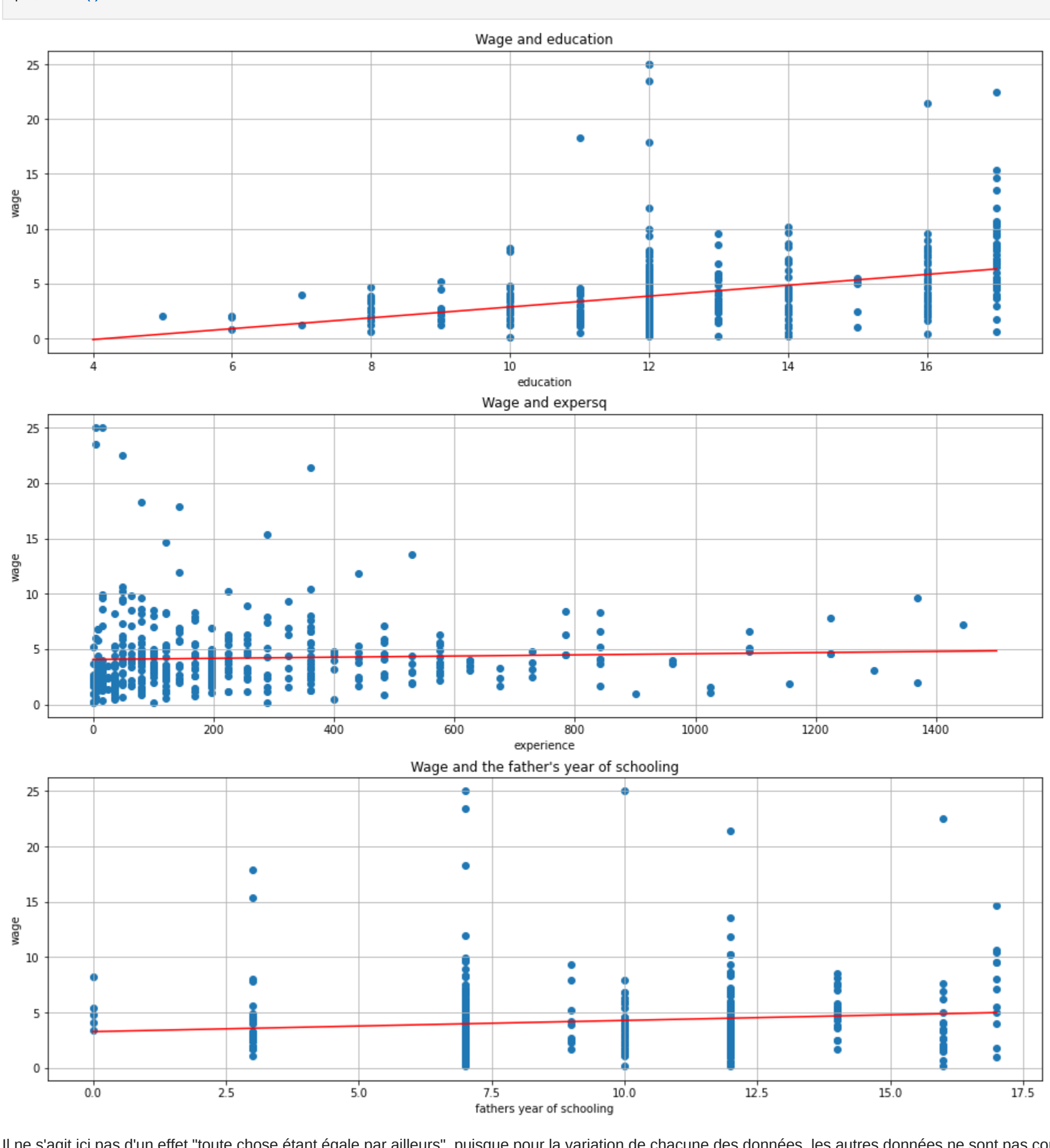
Cependant, puisque la p-value vaut environ 0 (<1%), on rejette l'hypothèse "présence de multicolinérarité" entre les 2 variables.

Question 5

Wage and education / expersq / education of the father

In [187…

```
plt.rcParams["figure.figsize"] = (16, 17)

plt.subplot(3,1,1)
plt.scatter(df['educ years of schooling'],df['wage estimated'])
plt.grid()
plt.title('Wage and education')
fit1 = np.polyfit(df['educ years of schooling'],df['wage estimated'] , 1)
tt1=np.linspace(4,17,100)
plt.plot(tt1,fit1[0]*tt1+fit1[1], 'r')
plt.xlabel('education')
plt.ylabel('wage')


plt.subplot(3,1,2)
plt.scatter(df['expersq'],df['wage estimated'])
plt.grid()
plt.title('Wage and expersq')
fit2 = np.polyfit(df['expersq'],df['wage estimated'] , 1)
tt2=np.linspace(0,1500,100)
plt.plot(tt2,fit2[0]*tt2+fit2[1], 'r')
plt.xlabel('experience')
plt.ylabel('wage')


plt.subplot(3,1,3)
plt.scatter(df['fathers year of schooling'],df['wage estimated'])
plt.title("Wage and the father's year of schooling")
fit3 = np.polyfit(df['fathers year of schooling'],df['wage estimated'] , 1)
tt3=np.linspace(0,17,100)
plt.plot(tt3,fit3[0]*tt3+fit3[1], 'r')
plt.xlabel('fathers year of schooling')
plt.ylabel('wage')
plt.grid()

plt.show()
```



Il ne s'agit ici pas d'un effet "toute chose étant égale par ailleurs", puisque pour la variation de chacune des données, les autres données ne sont pas constantes.

Question 6

L'hypothèse est que:

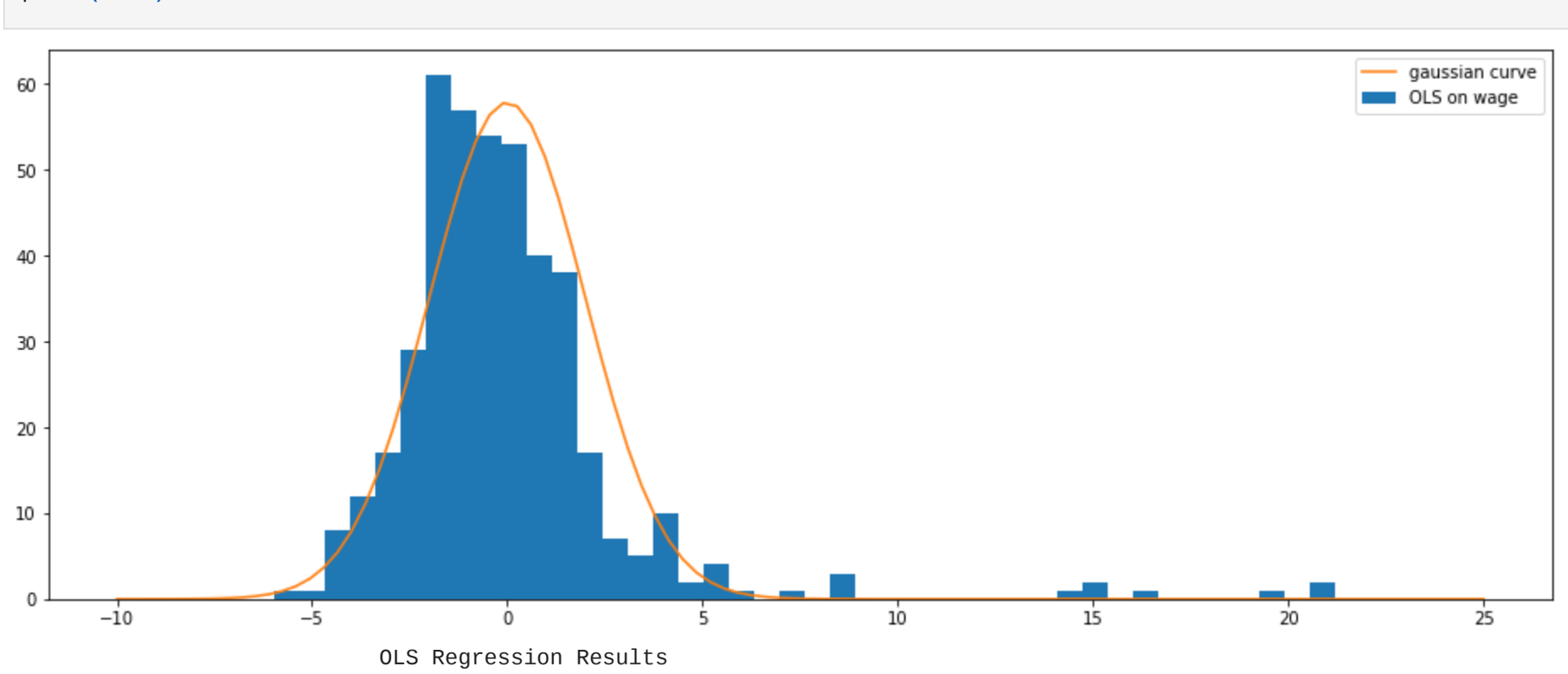
```
E(U|X) = 0

Since:    E(β_hat) = E(β) + (x-x')E(U|X)

                              E(β_hat) = E(β)
```

Le biais de variable omise est le biais lié à la non prise en compte dans le modèle d'une varibale importante.

Question 7

In [190…

```
plt.rcParams["figure.figsize"] = (16, 6)
a = np.ones(len(df['wage estimated']))

stack =np.column_stack((a,
                        df['live in SMSA'],
                        df['educ years of schooling'],
                        df['expersq'],
                        df['nwifeinc'],
                        df['kids below 6'],
                        df['kids between 6 and 18']))

mod0 = sm.OLS(df['wage estimated'], stack)
mod0 = mod0.fit()
res0 = mod0.resid
SSR0 = res0.T@res0


plt.hist(res0,'auto', label='OLS on wage')


x = np.linspace(-10, 25, 100)
y = norm.pdf(x,0,2)*290  ## We mutliply to normalize the 2 curves
plt.plot(x,y, label = 'gaussian curve')
plt.legend()

plt.show()

print(mod0.summary())
print(SSR0)
```




```
                            OLS Regression Results
==============================================================================
Dep. Variable:         wage estimated   R-squared:                       0.127
Model:                            OLS   Adj. R-squared:                  0.114
Method:                 Least Squares   F-statistic:                     10.18
Date:                Fri, 19 Nov 2021   Prob (F-statistic):           1.59e-10
Time:                        15:42:35   Log-Likelihood:                -1090.1
No. Observations:                 428   AIC:                             2194.
Df Residuals:                     421   BIC:                             2223.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.2446      0.926     -2.425      0.016      -4.064      -0.425
x1             0.3762      0.327      1.152      0.250      -0.266       1.018
x2             0.4617      0.070      6.569      0.000       0.324       0.600
x3             0.0006      0.001      1.026      0.305      -0.001       0.002
x4             0.0148      0.015      0.954      0.340      -0.016       0.045
x5             0.0203      0.396      0.051      0.959      -0.758       0.798
x6            -0.0700      0.124     -0.562      0.574      -0.315       0.175
==============================================================================
Omnibus:                      343.215   Durbin-Watson:                   2.058
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6326.629
Skew:                           3.359   Prob(JB):                         0.00
Kurtosis:                      20.596   Cond. No.                     2.21e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.21e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
4086.2143777901156
```

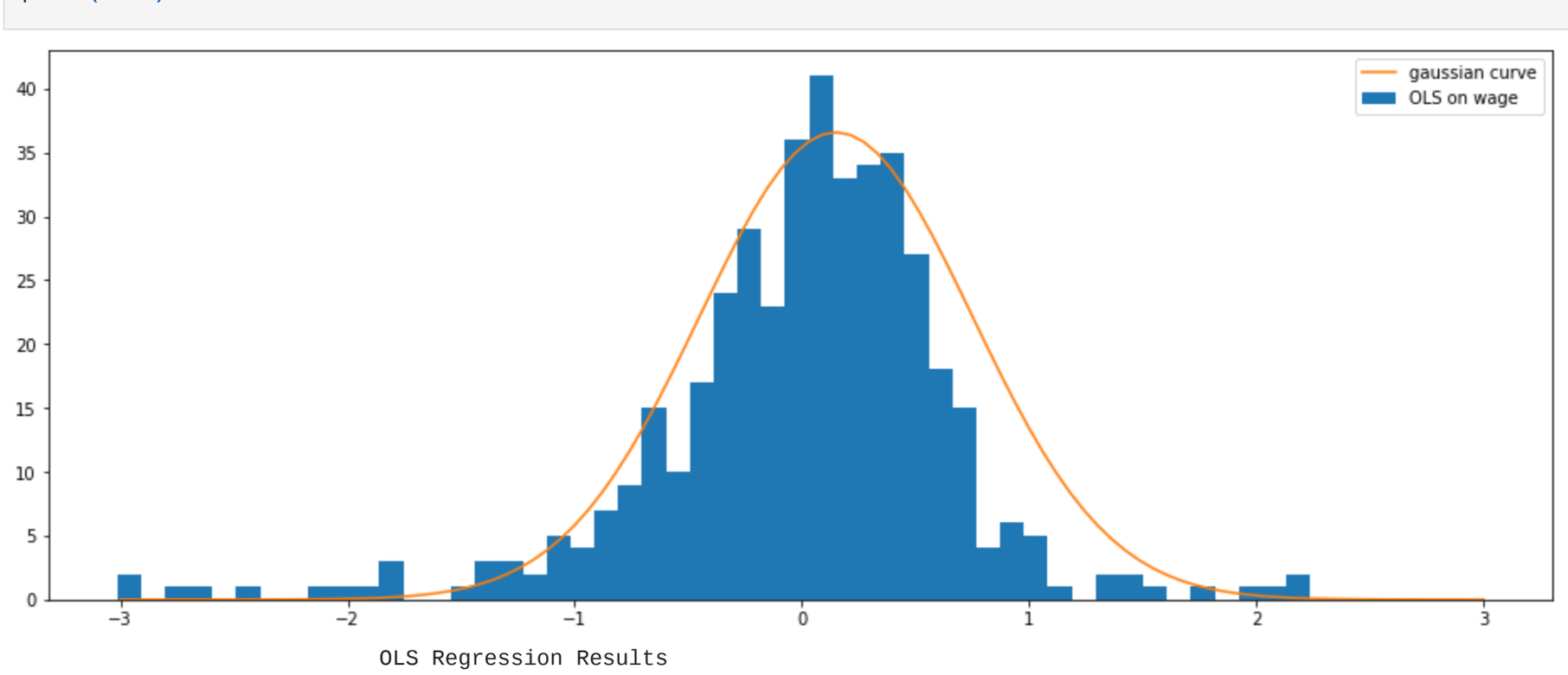
A partir de l'histogramme, nous pouvons interpréter que les résidus suivent une lois gaussienne centrée en 0.

Cependant, il y a une légère erreur introduisant une asymétrie pour des valeurs d'abscisse négatives proches de 0, et de plus, certaines valeurs me semblent incohérentes puisque la courbe de la gaussienne n'est pas parfaite: certains points semblent très éloignés des autres -ce qui est inhabituel pour une telle loi-.

Question 8

In [193…

```
stack =np.column_stack((a,
                        df['live in SMSA'],
                        df['educ years of schooling'],
                        df['expersq'],
                        df['nwifeinc'],
                        df['kids below 6'],
                        df['kids between 6 and 18']))


mod1 = sm.OLS(np.asarray(df['log(wage)']), stack)
mod1 = mod1.fit()
res1 = mod1.resid
SSR1 = res1.T@res1


plt.hist(res1,50, label='OLS on wage')

x = np.linspace(-3, 3, 100)
y = norm.pdf(x,0.15,0.6)*55  ## We mutliply to normalize the 2 curves
plt.plot(x,y, label = 'gaussian curve')
plt.legend()

plt.show()

print(mod1.summary())
print(SSR1)
```




```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.145
Model:                            OLS   Adj. R-squared:                  0.133
Method:                 Least Squares   F-statistic:                     11.90
Date:                Fri, 19 Nov 2021   Prob (F-statistic):           2.37e-12
Time:                        15:59:52   Log-Likelihood:                -434.57
No. Observations:                 428   AIC:                             883.1
Df Residuals:                     421   BIC:                             911.6
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2662      0.200     -1.331      0.184      -0.660       0.127
x1             0.0398      0.071      0.564      0.573      -0.099       0.179
x2             0.1033      0.015      6.798      0.000       0.073       0.133
x3             0.0003      0.000      2.568      0.011    8.01e-05       0.001
x4             0.0043      0.003      1.286      0.199      -0.002       0.011
x5            -0.0624      0.086     -0.730      0.466      -0.231       0.106
x6            -0.0219      0.027     -0.813      0.417      -0.075       0.031
==============================================================================
Omnibus:                       83.417   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              288.443
Skew:                          -0.851   Prob(JB):                     2.32e-63
Kurtosis:                       6.644   Cond. No.                     2.21e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.21e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
190.93732173097357
```

Nous pouvons encore observer une gaussienne centrée en 0 (ou plutot juste au dessus de 0), mais cette fois ci les valeurs sont bien mieux distribuées et elles sont cohérentes: il y a beaucoup moins de valeurs seules loin de l'origine (grâce au log)

Question 9

In [162…

```
SSR1 = res1.T@res1
stack_troncated =np.column_stack((a,
                                  df['live in SMSA'],
                                  df['educ years of schooling'],
                                  df['expersq'],
                                  df['nwifeinc'],
                                  df['kids below 6'],
                                  df['kids between 6 and 18']))

n,k =np.shape(stack1)

mod2 =sm.OLS(df['log(wage)'],stack_troncated)
mod2 =mod2.fit()
res2 =mod2.resid
SSR2 =res2.T@res2


F2 =((SSR2-SSR1)/SSR1)*(n-k)/1
fisher2 = f.sf(F2,1,n-k)
print("The p-value is:     p-value =",fisher2)
```

```
The p-value is:     p-value = 0.14340799213513744
```

On ne rejette donc PAS H0 pour aucun des seuils (14>10>5>1)

Question 10

In [112…

```
stack3 = np.column_stack((a,
                       df['live in SMSA'],
                       df['educ years of schooling'],
                       df['expersq'],
                       df['nwifeinc'],
                       df['kids below 6'],
                       df['kids between 6 and 18']))


data3 = df['log(wage)']-0.01*df['nwifeinc']
mod3 = sm.OLS(data3,stack3)
mod3 = mod3.fit()
res3 = mod3.resid
SSR3 = res3.T@res
n,k = np.shape(stack)
f3 = ((SSR3 - SSR1)/SSR1) *(n-k)/1
fisher3 = f.sf(f3,1,n-k)
print("The p-val is : ", fisher3)
```

```
The p-val is :  0.12442539027343329
```

On ne rejette donc pas l'hypothèse

Question 11

In [118…

```
stack4 = np.column_stack((a,
                       df['educ years of schooling'],
                       df['expersq'],
                       df['kids below 6'],
                       df['kids between 6 and 18']))

data4 = df['log(wage)']-0.01*df['nwifeinc']-0.05*df['live in SMSA']
mod4 = sm.OLS(data4,stack4)
mod4 = mod4.fit()
res4 = mod4.resid
SSR4 = res4.T@res4
n,k = np.shape(stack)
f4 = ((SSR4 - SSR1)/SSR1) *(n-k)/2
fisher4 = f.sf(f4,2,n-k)
print("The p-val is : ", fisher4)
```

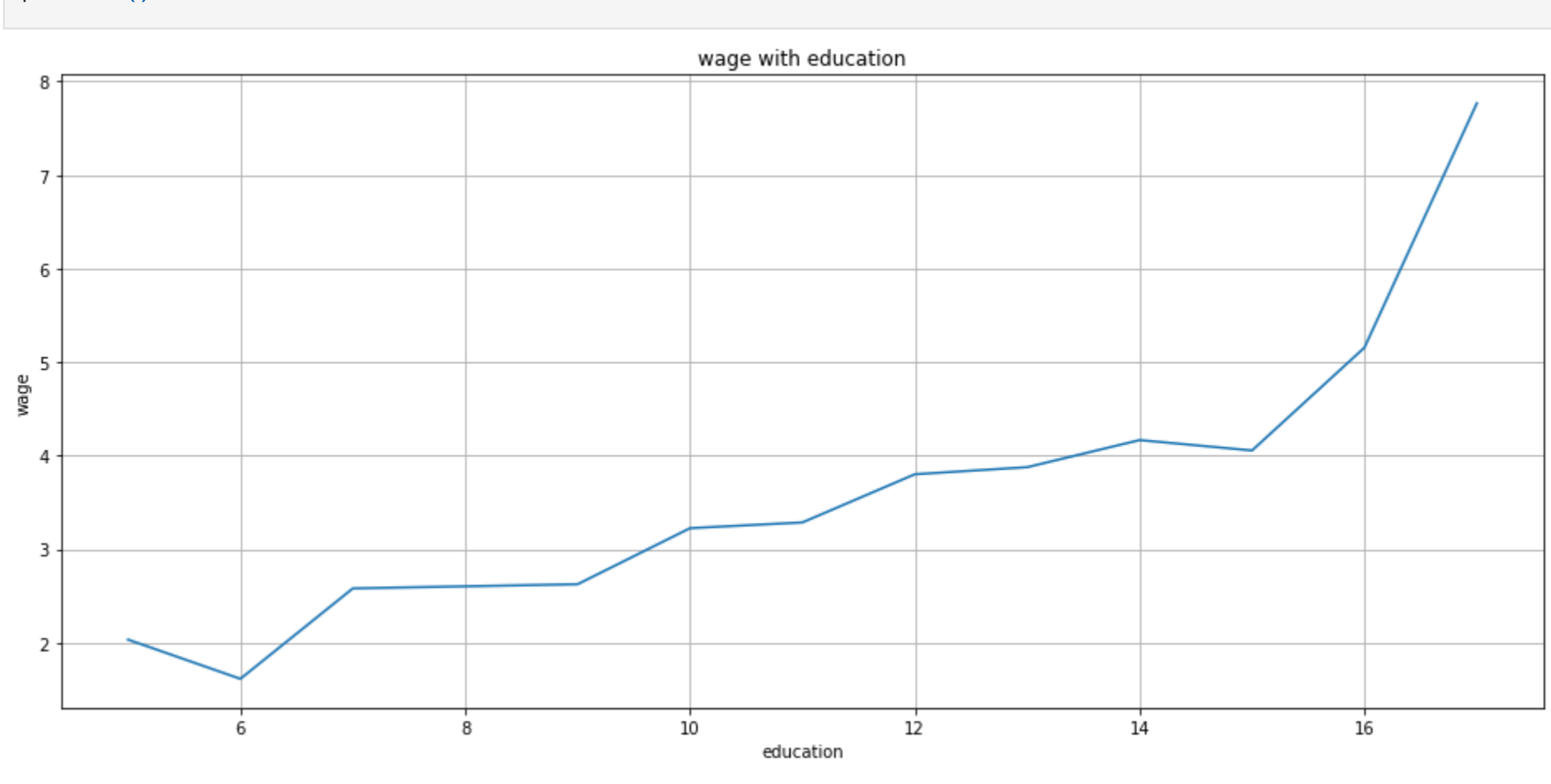
```
The p-val is :  0.2636174065428352
```

On ne rejette donc pas l'hypothèse à 5%

Question 12

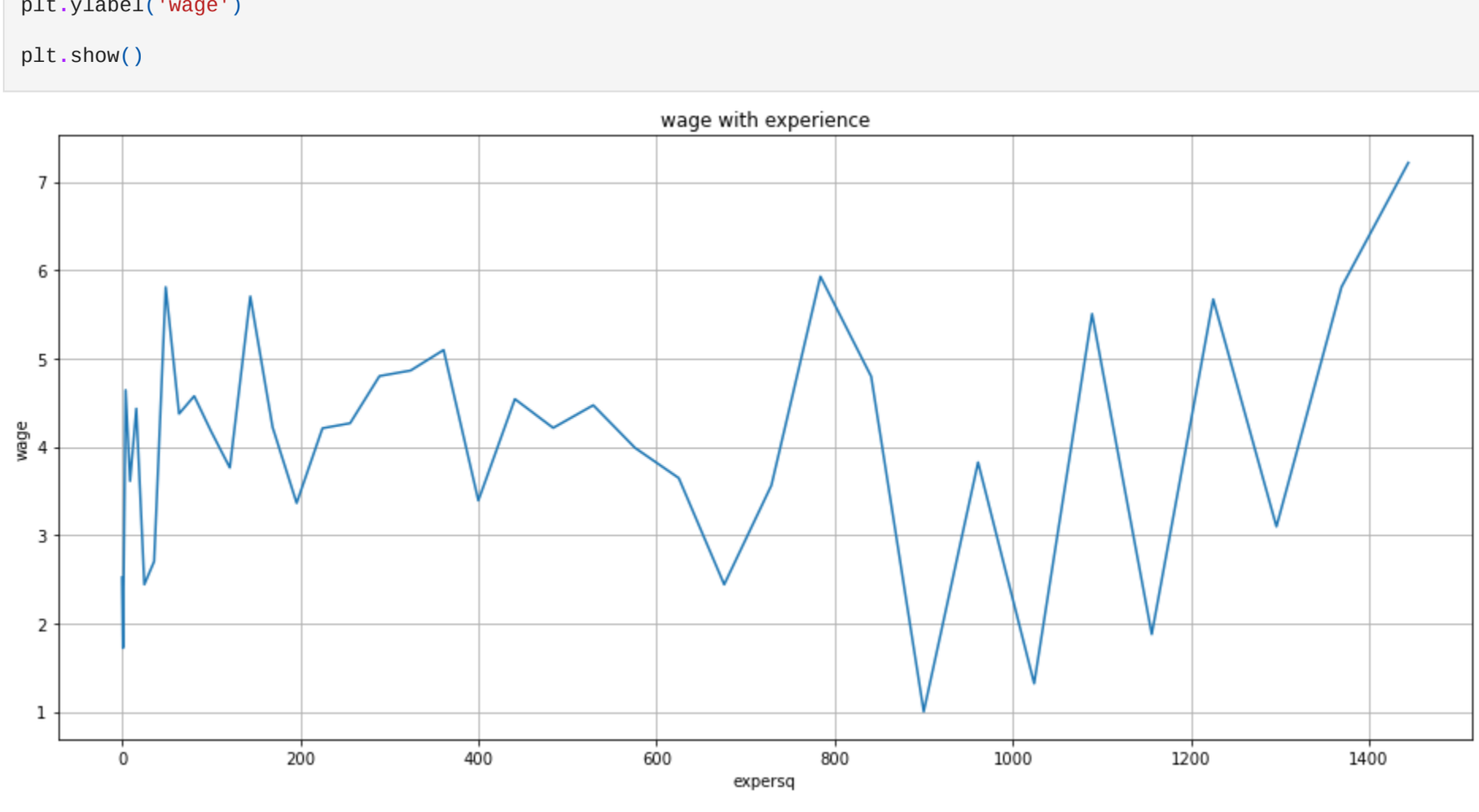
In [119…

```
plt.rcParams["figure.figsize"] = (16, 7)

plt.grid()
plt.title('wage with education')
both = df.groupby(['educ years of schooling']).mean()
both = both.sort_values(by=['educ years of schooling'])
plt.plot(both.index, both['wage estimated'])
plt.xlabel('education')
plt.ylabel('wage')
plt.show()
```



In [96]:

```
plt.grid()
plt.title('wage with experience')
both = df.groupby(['expersq']).mean()
both = both.sort_values(by=['expersq'])
plt.plot(both.index, both['wage estimated'])
plt.xlabel('expersq')
plt.ylabel('wage')

plt.show()
```



On remarque que le salaire dépend clairement plus de l'éducation que de l'experience (puisque la courbe selon l'experience oscille autour d'une valeur centrale)

Question 13

In [123…

```
tot_kids = df['kids below 6'].add(df['kids between 6 and 18'])

stack_kids =np.column_stack((a,
                        np.asarray(df['live in SMSA']),
                        np.asarray(df['educ years of schooling']),
                        np.asarray(df['expersq']),
                        np.asarray(df['kids below 6']),
                        np.asarray(tot_kids)))

mod5 = sm.OLS(df['log(wage)'], stack_kids)
mod5 = mod5.fit()
res5= mod5.resid

print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              log(wage)   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.132
Method:                 Least Squares   F-statistic:                     13.93
Date:                Fri, 19 Nov 2021   Prob (F-statistic):           1.30e-12
Time:                        15:00:18   Log-Likelihood:                -435.41
No. Observations:                 428   AIC:                             882.8
Df Residuals:                     422   BIC:                             907.2
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2564      0.200     -1.282      0.201      -0.650       0.137
x1             0.0595      0.069      0.862      0.389      -0.076       0.195
x2             0.1085      0.015      7.400      0.000       0.080       0.137
x3             0.0003      0.000      2.394      0.017    5.63e-05       0.001
x4            -0.0522      0.091     -0.576      0.565      -0.230       0.126
x5            -0.0210      0.027     -0.779      0.436      -0.074       0.032
==============================================================================
Omnibus:                       84.273   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              293.606
Skew:                          -0.857   Prob(JB):                     1.76e-64
Kurtosis:                       6.677   Cond. No.                     2.21e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.21e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

On remarque que pour la varibale x4 (kids bellow 6): P>|t| = 0,56

La p-value est donc 56% ce qui est plus grand que 5%, donc on rejette l'hyptohese d'égalité des coefficients 'kidsbt6' et 'kidsgt6'.

Question 14

In [200…

```
stack6 = np.column_stack((a,
                        np.asarray(df['educ years of schooling']),
                        np.asarray(df['expersq'])))

mod6 = sm.OLS(df['wage estimated'],stack6)
mod6 = mod6.fit()
res6 = mod6.resid


mod66 = res6**2
mod66 = sm.OLS(mod66, stack6)
mod66 = mod66.fit()
res66 = mod66.resid
SSR66 = res66.T @ res66
R6 = mod66.rsquared

nbis,kbis = np.shape(stack6)
fbis = (R6/kbis) / ((1-R6) / (nbis-kbis-1))
p_values = mod66.f_pvalue

print("statistique de fischer = ", mod66.fvalue )
print("valeur critique= ",f.isf(0.05,2,n-k))
print('p-value = ', p_values)
```

```
statistique de fischer =  1.585263938652012
valeur critique=  3.0171506509713275
p-value =  0.147253803625398
```

14>10>5>1 Donc on rejette l'hypothèse pour tous les seuils.

Question 15

In [205…

```
def SSR(plusOrMoins):

    if plusOrMoins == 'moins':
        filtre = df['age women']<=45
    elif plusOrMoins == 'plus':
        filtre = df['age women']>45

    data7 = df['log(wage)'][filtre]
    stack7 = np.column_stack((a[filtre],
                            np.asarray(df['live in SMSA'][filtre]),
                            np.asarray(df['educ years of schooling'][filtre]),
                            np.asarray(df['expersq'][filtre]),
                            np.asarray(df['nwifeinc'][filtre])))

    mod7 = sm.OLS(data7,stack7)
    mod7 = mod7.fit()
    p_value = mod7.pvalues
    res7 = mod7.resid
    SSR7 = res7.T @ res7

    return SSR7
```

In [224…

```
SSR_Moins43 = SSR('moins')
SSR_plus43 = SSR('plus')
SSR_tot = SSR('tot')

SSR_sum = SSR_Moins43 + SSR_Plus43
print('SSR_Moins43:', SSR_Moins43,
      'SSR_plus43:', SSR_plus43,
      'SSR_tot:', SSR_tot)

f7 = ((SSR_Moins43 - SSR_sum)/SSR_sum)*((n-2*7)/7)
p_value7 = f.sf(f7,7,n-7)
print('Ficher chow:',f7,'P_value chow: ',p_value7)
print('P_value chow: ',p_value7)
```

```
SSR_Moins43: 105.36297749294738 SSR_plus43: 80.03719784649275 SSR_tot: 187.16273662839203
Ficher chow: 1.162737927566472
 P_value chow:  0.31373828930284803
```

Question 16

In [225…

```
def SSR_bis(plusOrMoins):
    if plusOrMoins == 'moins':
        filtre = df['age women']<=45
    elif plusOrMoins == 'plus':
        filtre = df['age women']>45

    data8 = df['log(wage)'][filtre]
    stack8 = np.column_stack((a[filtre],
                            np.asarray(df['educ years of schooling'][filtre]),
                            np.asarray(df['expersq'][filtre])))

    mod8 = sm.OLS(data8,stack8)
    mod8 = mod8.fit()
    p_value = result.pvalues
    res8 = mod8.resid
    SSR8 = res8.T @ res8

    return SSR8
```

In [226…

```
SSR_Moins43_bis = SSR_bis('moins')
SSR_Plus43_bis = SSR_bis('plus')


SSR_sum_bis = SSR_Moins43_bis + SSR_Plus43_bis
print('SSR_Moins43_bis:', SSR_Moins43_bis,
      'SSR_Plus43_bis:', SSR_Plus43_bis)

f8 = ((SSR_Moins43_bis - SSR_sum_bis)/SSR_sum_bis)*((n-2*7)/7)
p_value8 = f.sf(f8,7,n-7)
print('Ficher chow:',f8,'P_value chow: ',p_value8)
```

```
SSR_Moins43_bis: 105.50372924028493 SSR_Plus43_bis: 81.99284013840149
Ficher chow: 0.8527829272729232 P_value chow:  0.5318200188810824
```