Question1-1

```
#import numpy as py
import pandas as pd

mroz = pd.read_fwf('../data/MROZ.txt', sep="/t", header=None)
mroz = mroz[mroz[6] != '.']
mroz[6] = pd.to_numeric(mroz[6])
mroz[20] = pd.to_numeric(mroz[20])
mroz = mroz[mroz[6] > 0]

inlf = mroz[0]          # =1 if in labor force, 1975
hours = mroz[1]    # hours worked, 1975
kidslt6 = mroz[2]                    # kids < 6 years
kidsge6 = mroz[3]                    # kids 6-18
age = mroz[4]                        # woman's age in yrs
educ = mroz[5]    # years of schooling
wage = mroz[6]                         # estimated wage from earns., hours
repwage = mroz[7]                    # reported wage at interview in 1976
hushrs = mroz[8]    # hours worked by husband, 1975
husage = mroz[9]    # husband's age
huseduc = mroz[10]    # husband's years of schooling
huswage = mroz[11]    # husband's hourly wage, 1975
faminc = mroz[12]    # family income, 1975
mtr = mroz[13]    # fed. marginal tax rate facing woman
motheduc = mroz[14]    # mother's years of schooling
fatheduc = mroz[15]    # father's years of schooling
unem = mroz[16]    # unem. rate in county of resid.
city = mroz[17]                      # =1 if live in SMSA
exper = mroz[18]    # actual labor mkt exper
nwifeinc = mroz[19]    # (faminc - wage*hours)/1000
lwage = mroz[20]    # log(wage)
expersq = mroz[21]    # exper^2
```

Question 1-2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import dataloader_p1 as dt_p1
plt.subplot(131)
plt.hist(dt_p1.wage,'auto',color = 'purple')
plt.title('salaire')

plt.subplot(132)
```

```
plt.hist(dt_p1.age,'auto',color = 'purple')
plt.title('age')

plt.subplot(133)
plt.hist(dt_p1.educ,'auto',color = 'purple')
plt.title('education')
plt.show()

print('The mean is : ', np.mean(dt_p1.wage))
print('The median is : ', np.median(dt_p1.wage))
print('The maximum is : ', np.max(dt_p1.wage))
print('The minimum is : ', np.min(dt_p1.wage))
print('The standard deviation is : ', np.std(dt_p1.wage))
print('The variance is : ', np.var(dt_p1.wage))

print('The mean is : ', np.mean(dt_p1.age))
print('The median is : ', np.median(dt_p1.age))
print('The maximum is : ', np.max(dt_p1.age))
print('The minimum is : ', np.min(dt_p1.age))
print('The standard deviation is : ', np.std(dt_p1.age))
print('The variance is : ', np.var(dt_p1.age))

print('The mean is : ', np.mean(dt_p1.educ))
print('The median is : ', np.median(dt_p1.educ))
print('The maximum is : ', np.max(dt_p1.educ))
print('The minimum is : ', np.min(dt_p1.educ))
print('The standard deviation is : ', np.std(dt_p1.educ))
print('The variance is : ', np.var(dt_p1.educ))

mediane_e = np.median(dt_p1.huswage)

plt.subplot(131)
plt.hist(dt_p1.wage[dt_p1.huswage>=mediane_e],'auto',color = 'r')
plt.title('salaire')

plt.subplot(132)
plt.hist(dt_p1.age[dt_p1.huswage>=mediane_e],'auto',color = 'r')
plt.title('age')

plt.subplot(133)
plt.hist(dt_p1.educ[dt_p1.huswage>=mediane_e],'auto',color = 'r')
plt.title('education')
plt.show()
```

```
print('The mean is : ', np.mean(dt_p1.wage[dt_p1.huswage>=mediane_e]))
print('The median is : ', np.median(dt_p1.wage[dt_p1.huswage>=mediane_e]))
print('The maximum is : ', np.max(dt_p1.wage[dt_p1.huswage>=mediane_e]))
print('The minimum is : ', np.min(dt_p1.wage[dt_p1.huswage>=mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.wage[dt_p1.huswage>=mediane_e]))
print('The variance is : ', np.var(dt_p1.wage[dt_p1.huswage>=mediane_e]))

print('The mean is : ', np.mean(dt_p1.age[dt_p1.huswage>=mediane_e]))
print('The median is : ', np.median(dt_p1.age[dt_p1.huswage>=mediane_e]))
print('The maximum is : ', np.max(dt_p1.age[dt_p1.huswage>=mediane_e]))
print('The minimum is : ', np.min(dt_p1.age[dt_p1.huswage>=mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.age[dt_p1.huswage>=mediane_e]))
print('The variance is : ', np.var(dt_p1.age[dt_p1.huswage>=mediane_e]))

print('The mean is : ', np.mean(dt_p1.educ[dt_p1.huswage>=mediane_e]))
print('The median is : ', np.median(dt_p1.educ[dt_p1.huswage>=mediane_e]))
print('The maximum is : ', np.max(dt_p1.educ[dt_p1.huswage>=mediane_e]))
print('The minimum is : ', np.min(dt_p1.educ[dt_p1.huswage>=mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.educ[dt_p1.huswage>=mediane_e]))
print('The variance is : ', np.var(dt_p1.educ[dt_p1.huswage>=mediane_e]))

plt.subplot(131)
plt.hist(dt_p1.wage[dt_p1.huswage<mediane_e],'auto',color = 'b')
plt.title('salary')

plt.subplot(132)
plt.hist(dt_p1.age[dt_p1.huswage<mediane_e],'auto',color = 'b')
plt.title('age')

plt.subplot(133)
plt.hist(dt_p1.educ[dt_p1.huswage<mediane_e],'auto',color = 'b')
plt.title('education')
plt.show()

print('The mean is : ', np.mean(dt_p1.wage[dt_p1.huswage<mediane_e]))
print('The median is : ', np.median(dt_p1.wage[dt_p1.huswage<mediane_e]))
print('The maximum is : ', np.max(dt_p1.wage[dt_p1.huswage<mediane_e]))
print('The minimum is : ', np.min(dt_p1.wage[dt_p1.huswage<mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.wage[dt_p1.huswage<mediane_e]))
print('The variance is : ', np.var(dt_p1.wage[dt_p1.huswage<mediane_e]))

print('The mean is : ', np.mean(dt_p1.age[dt_p1.huswage<mediane_e]))
print('The median is : ', np.median(dt_p1.age[dt_p1.huswage<mediane_e]))
print('The maximum is : ', np.max(dt_p1.age[dt_p1.huswage<mediane_e]))
```

```
print('The minimum is : ', np.min(dt_p1.age[dt_p1.huswage<mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.age[dt_p1.huswage<mediane_e]))
print('The variance is : ', np.var(dt_p1.age[dt_p1.huswage<mediane_e]))

print('The mean is : ', np.mean(dt_p1.educ[dt_p1.huswage<mediane_e]))
print('The median is : ', np.median(dt_p1.educ[dt_p1.huswage<mediane_e]))
print('The maximum is : ', np.max(dt_p1.educ[dt_p1.huswage<mediane_e]))
print('The minimum is : ', np.min(dt_p1.educ[dt_p1.huswage<mediane_e]))
print('The standard deviation is : ', np.std(dt_p1.educ[dt_p1.huswage<mediane_e]))
print('The variance is : ', np.var(dt_p1.educ[dt_p1.huswage<mediane_e]))
```

Question 1-3

```
plt.hist(dt_p1.wage,'auto')
plt.title('histogram of women salary')
plt.show()

log_wage = np.log(dt_p1.wage)
plt.hist(log_wage,'auto')
plt.title('histogram of women salary in log')
plt.show()
```

Question 1-4

```
corr = np.corrcoef(dt_p1.motheduc, dt_p1.fatheduc)
print(corr)
```

Question 1-5

```
plt.scatter(dt_p1.wage, dt_p1.educ)
plt.xlabel('wage')
plt.ylabel('education year')
plt.title('cloud map of wage & education')
plt.show()

plt.scatter(dt_p1.wage, dt_p1.exper)
plt.xlabel('wage')
plt.ylabel('work experience year')
plt.title('cloud map of wage & experience')
plt.show()

plt.scatter(dt_p1.wage, dt_p1.fatheduc)
plt.xlabel('wage')
plt.ylabel('father educcation year')
plt.title('cloud map of wage & father education')
plt.show()
```

Question 1-7

```
def linear_reg(X,y):
    beta = np.linalg.inv(X.T @ X)@X.T@y
    u=y-X@beta
    n,k = np.shape(X)
    sig2=u.T@u/(n-k)
    Var=sig2*np.linalg.inv(X.T @ X)
    std=np.sqrt(np.diag(Var))
    return beta, u , sig2, Var, std

y = dt_p1.wage
const = np.ones(np.shape(y))
X=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper,
dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))

beta_7, u_7, sig2_7, Var_7, std_7 = linear_reg(X,y)

plt.hist(u_7,'auto')
plt.ylabel('cumulative number')
plt.xlabel('erreur')
plt.title('erreur')
plt.show()
```

Question 1-8

```
logy = dt_p1.lwage
const = np.ones(np.shape(logy))

X=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper,
dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))

beta_8, u_8, sig2_8, Var_8, std_8 = linear_reg(X,logy)

u_8.T@u_8

plt.hist(u_8,'auto')
plt.ylabel('cumulative number')
plt.xlabel('erreur')
plt.title('erreur')
plt.show()

std_8

import statsmodels.api as sm
model=sm.OLS(logy,X)
```

```
results = model.fit()
print(results.summary())

plt.figure(figsize=(7, 4))
sns.distplot(results.resid)
plt.title("Résidus du modèle", fontsize=14)
plt.show()
```

Question 1-9

```
n,k = np.shape(X)
test_9 = beta_8[4]/std_8[4]
2*t.sf(test_9,n-k)
test_9
seuil_signifi = [0.01, 0.05,0.1]

for i    in seuil_signifi:
   print(i,'left side:',t.ppf(i/2,n-k,loc = beta_8[4], scale = std_8[4]),'right side:',t.ppf(1-i/2,n-k,loc = beta_8[4], scale = std_8[4]))
```

Question 1-10

```
test_10 = (beta_8[4]-0.01)/std_8[4]
2*(1-t.sf(test_10,n-k))
```

Question 1-11

```
y = dt_p1.wage
const = np.ones(np.shape(y))
X0=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper, dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
y = y - 0.01*X0[:,4]-0.05*X0[:,1]
X = np.column_stack((const, dt_p1.educ,dt_p1.exper,dt_p1.kidslt6,dt_p1.kidsgt6))
beta_11, u_11, sig2_11, Var_11, std_11 = linear_reg(X,y)
model=sm.OLS(y,X)
results = model.fit()
print(results.summary())

SSR_0 = u_7.T@u_7
SSR_11 = u_11.T@u_11
n,k=np.shape(X0)
F=((SSR_11-SSR_0)/2)/(SSR_0/(n-k))
f.sf(F,2,n-k)
```

Question 1-12

```
xxx_educ = np.linspace(5,20,100)
yyy_wage = xxx_educ*beta_7[2]+beta_7[0]
```

```
plt.scatter(dt_p1.educ,dt_p1.wage)
plt.plot(xxx_educ,yyy_wage, color = 'r')
plt.xlabel('education')
plt.ylabel('wage')
plt.title('variation of wage with education')

xxx_exper = np.linspace(0,40,100)
yyy_wage = xxx_exper*beta_7[3]+beta_7[0]
plt.scatter(dt_p1.exper,dt_p1.wage)
plt.plot(xxx_exper,yyy_wage, color = 'r')
plt.xlabel('experience professionnelle')
plt.ylabel('wage')
plt.title('variation of wage with experience professionnelle')

import numpy as np
import matplotlib.pyplot as mp
from mpl_toolkits.mplot3d import axes3d
mp.figure("3D Scatter", facecolor="lightgray")
ax3d = mp.gca(projection="3d")

mp.title('3D Scatter', fontsize=20)
ax3d.set_xlabel('education', fontsize=14)
ax3d.set_ylabel('experience', fontsize=14)
ax3d.set_zlabel('salary', fontsize=14)
mp.tick_params(labelsize=10)

zzz_wage = xxx_exper*beta_7[3]+beta_7[0]+xxx_educ*beta_7[2]
ax3d.scatter(dt_p1.educ, dt_p1.exper,dt_p1.wage, marker="o")
ax3d.plot(xxx_educ, xxx_exper, zzz_wage, color = 'r')

mp.show()
```

Question 1-13

```
y = dt_p1.wage
const = np.ones(np.shape(y))
#X0=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper, dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
X = np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper,dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6-dt_p1.kidslt6))

model=sm.OLS(y,X)
results = model.fit()
print(results.summary())
```

```
beta_13, u_13, sig2_13, Var_13, std_13 = linear_reg(X,y)
test_13 = beta_13[6]/std_13[6]
2*(1-t.sf(test_13,n -k))
```

Question 1-14

```
y = dt_p1.wage
const = np.ones(np.shape(y))
X=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper,
dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
model=sm.OLS(y,X)
results = model.fit()
print(results.summary())

u = results.resid
u2= u**2
y = u2
model = sm.OLS(y,X)
results    = model.fit()
print(results.summary())

city2 = (dt_p1.city == 0)
y = dt_p1.wage
const = np.ones(np.shape(y))
X=np.column_stack((const, city2, (dt_p1.educ)**4, (dt_p1.expersq)**4,
dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
model=sm.OLS(y,X)
results = model.fit()
print(results.summary())
u = results.resid
u2= u**2
y = u2
model = sm.OLS(y,X)
results    = model.fit()
print(results.summary())
```

Question 1-15

```
X0=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper,
dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
s_g = dt_p1.age>=43
y = dt_p1.lwage
X = X0[s_g,:]
y = y[s_g]
n,k = np.shape(X)
model_g = sm.OLS(y,X)
```

```
results_g = model_g.fit()
u_g = results_g.resid
SSR15_g = u_g.T@u_g


s_l = dt_p1.age<43
y = dt_p1.lwage
X = X0[s_l,:]
y = y[s_l]
n,k = np.shape(X)
model_l = sm.OLS(y,X)
results_l = model_l.fit()
u_l = results_l.resid
SSR15_l = u_l.T@u_l
print(results_g.summary())
print(results_l.summary())

F = (SSR_8- (SSR_G+SSR_15))/k/((SSR_G+SSR_15)/(n-2*k))
f.sf(F,2,n-k)
```

Question 1-16

```
X0=np.column_stack((const, dt_p1.city, dt_p1.educ,dt_p1.exper, dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
y = dt_p1.lwage
femegreateduc = (dt_p1.age>=43)*dt_p1.educ
femegreatexper = (dt_p1.age>=43)*dt_p1.exper
X = np.column_stack((const, femegreateduc, femegreatexper, dt_p1.city, dt_p1.educ,dt_p1.exper, dt_p1.nwifeinc,dt_p1.kidslt6,dt_p1.kidsgt6))
model = sm.OLS(y,X)
results  = model.fit()
print(results.summary())
SSR16_1 = results.resid.T@results.resid
SSR16_0 = u_8.T@u_8
n_16,k_16 = np.shape(X)
F = ((SSR16_0-SSR16_1)/2)/(SSR16_1/(n_16-k_16))
f.sf(F,2,n_16-k_16)
```

Question 2-1

```
import numpy as py
import pandas as pd

quarterly = pd.read_excel('../data/quarterly.xls')
```

Question 2-2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from    scipy.stats import f
from scipy.stats import t
import statsmodels.api as sm

import dataloader_p2 as dt_p2

CPI = dt_p2.quarterly.CPI
year = pd.to_datetime(dt_p2.quarterly.DATE)
n = len(CPI)

inf = []
for i in range(1,n):
        inter = (CPI[i] - CPI[i-1])/CPI[i-1]
        inf.append(inter)

plt.plot(year[1:n],inf)
plt.xlabel('years')
plt.ylabel('taux')
plt.title('taux de CPI dans les annees')
plt.show()
```

Question 2-3

```
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
plot_acf(np.array(inf))
plt.show()
plot_pacf(np.array(inf))
plt.show()
```

Question 2-5

```
import statsmodels.tsa.api as smt
minf = smt.AR(inf).fit(maxlag = 4)
```

```
print(minf.params)
print(minf.aic)
print(minf.bic)
```

Question 2-6

```
y = dt_p2.quarterly.Unemp[1:]
const = np.ones(np.shape(y))
X = np.column_stack((const,np.array(inf)*100))

model=sm.OLS(y,X)
results = model.fit()
print(results.summary())
```

Question 2-7

```
u = results.resid
plt.plot(u)
import statsmodels
statsmodels.stats.stattools.durbin_watson(u, axis=0)
acf(u)
plot_acf(np.array(u))
plt.show()
```

Question 2-8

```
rho = results1.params[0]
transform_u = np.array(u[1:n])-np.array(rho*u_1)
```

Question 2-9

```
n = len(dt_p2.quarterly.Unemp)
y = dt_p2.quarterly.Unemp[1:int((n/2))]
const = np.ones(np.shape(y))
X = np.column_stack((const,np.array(inf[:int((n/2))-1])))

model=sm.OLS(y,X)
results = model.fit()
print(results.summary())

n = len(dt_p2.quarterly.Unemp)
y = dt_p2.quarterly.Unemp[int(n/2):]
const = np.ones(np.shape(y))
X = np.column_stack((const,np.array(inf[int(n/2)-1:])))

model=sm.OLS(y,X)
results = model.fit()
print(results.summary())
```

Question 2-10

```
n =len(dt_p2.quarterly.Unemp)
y = dt_p2.quarterly.Unemp[5:n]
const = np.ones(np.shape(y))
inf_1 = inf[3:n-2]
inf_2 = inf[2:n-3]
inf_3 = inf[1:n-4]
inf_4 = inf[0:n-5]
cho_1 = dt_p2.quarterly.Unemp[4:n-1]
cho_2 = dt_p2.quarterly.Unemp[3:n-2]
cho_3 = dt_p2.quarterly.Unemp[2:n-3]
cho_4 = dt_p2.quarterly.Unemp[1:n-4]

X =
np.column_stack((const,np.array(inf_1),np.array(inf_2),np.array(inf_3),np.array(inf_4),cho_1,cho
_2,cho_3,cho_4))
X =
np.column_stack((const,np.array(inf_1)*100,np.array(inf_2)*100,np.array(inf_3)*100,np.array(in
f_4)*100,cho_1,cho_2,cho_3,cho_4))

model=sm.OLS(y,X)
results = model.fit()
print(results.summary())

X = np.column_stack((dt_p2.quarterly.Unemp[0:n-1],np.array(inf)*100))
statsmodels.tsa.stattools.grangercausalitytests(X,5)
```

Question 2-11

```
d_inf = (results.params[1],results.params[2],results.params[3],results.params[4])
x = (1,2,3,4)
plt.bar(x,d_inf)
plt.title('effect of lag on inf')
plt.xlabel('lag on inf')
plt.ylabel('coef')
plt.show()

d_cho = (results.params[5],results.params[6],results.params[7],results.params[8])
x = (1,2,3,4)
plt.bar(x,d_cho)
plt.title('effect of lag on chomage')
plt.xlabel('lag on chomage')
plt.ylabel('coef')
plt.show()
```

results.params[1]+results.params[2]+results.params[3]+results.params[4]