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(inf\_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]