Ayudantía 2

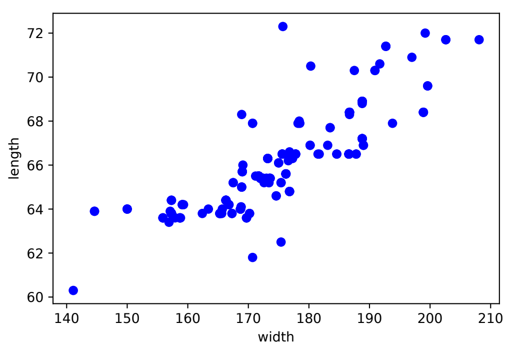
Jean Paul Maidana
Javier Palma
Taller de Modelamiento Matemático

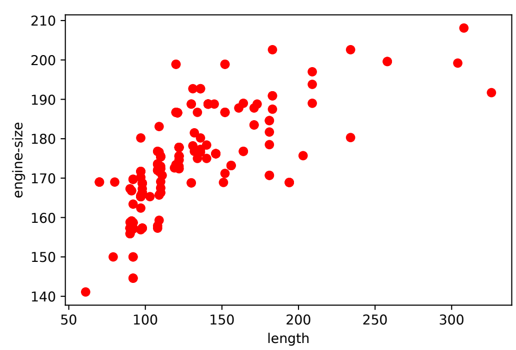
13 de julio de 2018

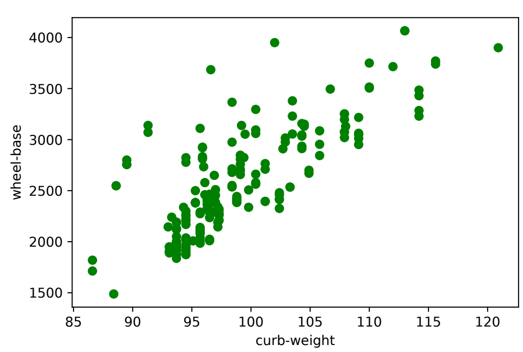
Ejercicio 1. En base a los datos de la página web

http://archive.ics.uci.edu/ml/datasets/Automobile

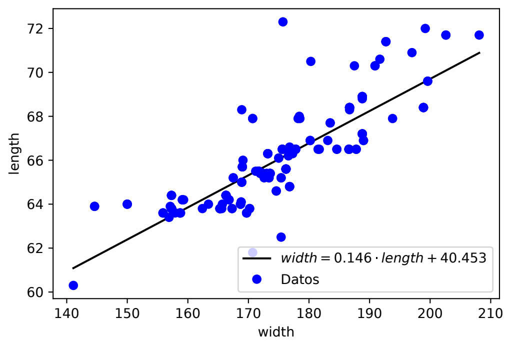
a) Describa al menos tres relaciones lineales entre las variables cuantitativas de la base de datos a través de gráficas apropiadas.
Existen varias relaciones aceptables, a continuación se presentan 3:

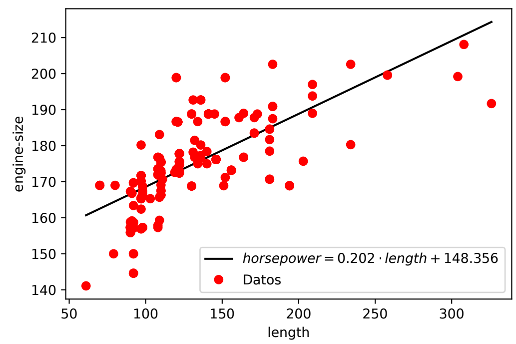


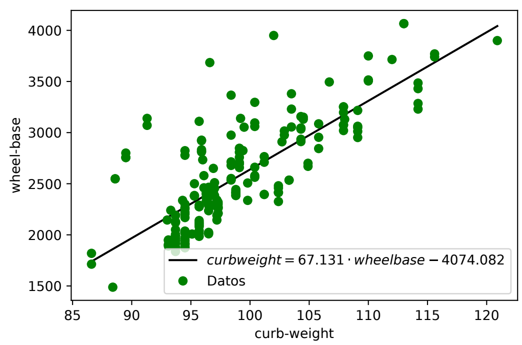




b) Para las relaciones encontradas en el punto anterior realice los respectivos ajustes de parámetros. Se realiza el ajuste de minimos cuadrados para el modelo de ecuación de una recta para cada una de las relaciones del punto anterior. Los valores de cada ajuste se presentan en el gráfico y los respectivos códigos se encuentran en el anexo.

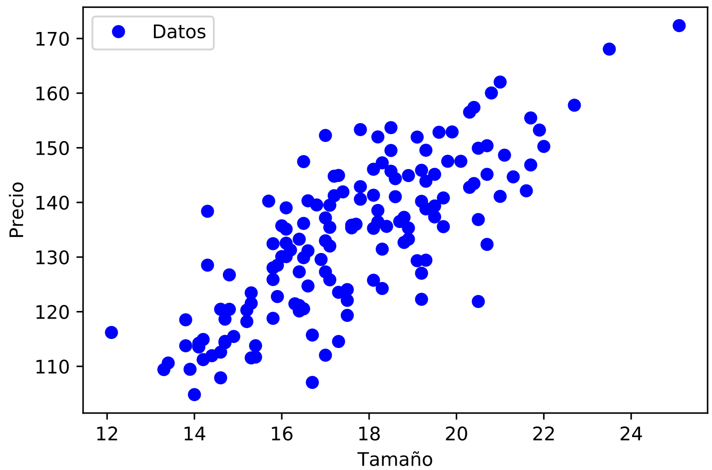






Ejercicio 2. Los datos contenidos en el archivo precios_casas.csv contiene cuatro columnas (valor,precio,tama,numpiez), donde las variables valor y precio están en miles de dolares, la variable tama está en cientos de pies^2 y la variable numpiez es el número de habitaciones.

a) Haga un gráfico de el precio versus el tamaño de la casa, ¿Existe alguna relación entre estas dos variables?

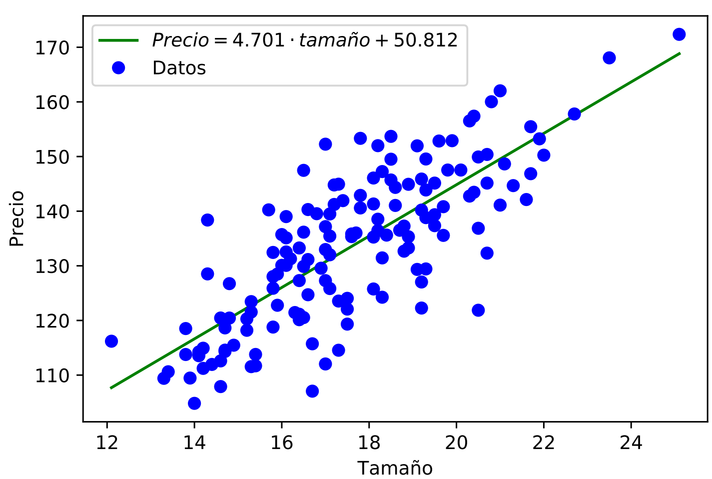


A partir de este gráfico se puede apreciar que una relación lineal de la forma

$$\text{precio} = a \cdot \text{tamaño} + b$$

donde a es la pendiente y b el intercepto.

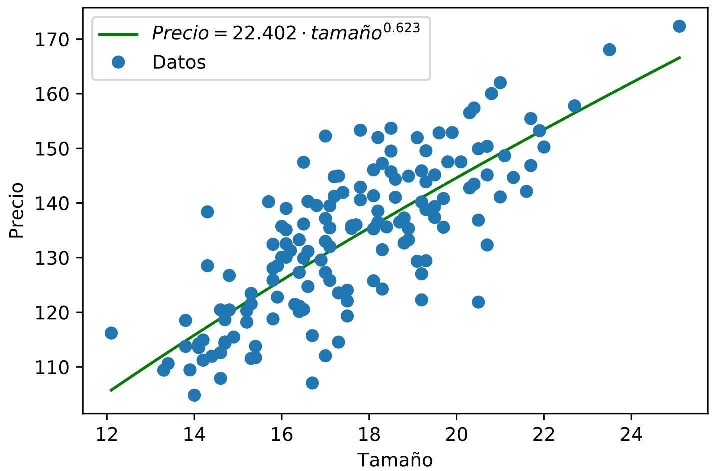
b) Proponga un modelo de regresión y ajuste sus datos con el método de mínimos cuadrados.
A partir del modelo propuesto en el punto anterior, se estima a y b por el método de mínimos cuadrados (ver anexo). En la siguiente figura se puede ver el ajuste respectivo:



c) Proponga un segundo modelo y ajuste sus datos con el método de mínimos cuadrados. Se puede proponer un modelo de la forma

$$\text{precio} = \alpha \cdot \text{tamaño}^N$$

con α y N a estimar desde la función de mínimos cuadrados que se encuentra en el anexo. La siguiente figura muestra dicho ajuste con los valores de α y N:



d) ¿Cuál es el precio de una casa de 2200pies^2 usando el modelo propuesto 1 y 2 del punto anterior.
Los precios para los dos modelos son:

Precio casa 2200 pies^2 en el modelo 1: US\$154244

Precio casa 2200 pies^2 en el modelo 2: US\$153450

e) Decida cuál de los dos modelos propuestos es el mejor a traves del criterio del error cuadrático medio.
Para determinar cual de los dos modelos propuestos se ajusta mejor, se puede calcular la suma de cuadrados del error, donde para el primer modelo (ecuación de una recta):

$$S = \sum_{i=1}^{149} (\text{precio}_i - a \cdot \text{tamaño}_i - b)^2 = 11391.3662$$

y para el caso de una función de tipo potencia:

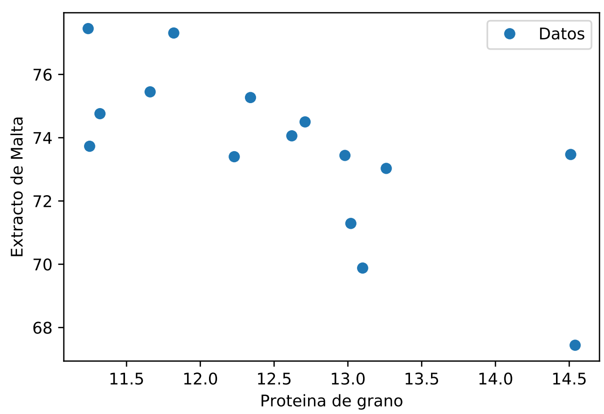
$$S = \sum_{i=1}^{149} \left(\text{precio}_i - \alpha \cdot \text{tamaño}_i^N\right)^2 = 11370.6136$$

Se puede ver que el modelo 2 es el que comete el menor error, por lo tanto se prefiere el modelo de tipo potencia por sobre el primer modelo.

Ejercicio 3. Desde los siguientes datos:

Proteína de grano (%)	Extracto de Malta (%)
13.02	71.29
11.32	74.76
12.23	73.40
11.82	77.31
14.51	73.47
11.66	75.45
13.26	73.03
12.71	74.50
11.25	73.73
11.24	77.45
13.10	69.88
14.54	67.44
12.98	73.44
12.34	75.27
12.62	74.06

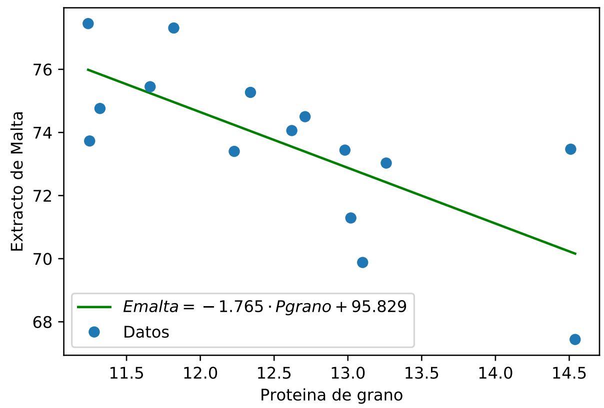
a) Grafique sus datos



b) Derive un modelo y ajuste a través de minimos cuadrados.
Observando los datos se puede ver que existe una relación lineal de la forma

$$\text{Emalta} = a \cdot \text{Pgrano} + b$$

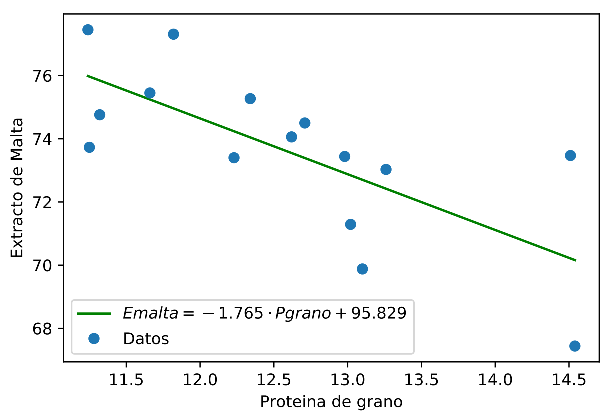
donde a es la pendiente y b es el intercepto de la recta que expresa la relación lineal entre el extracto de malta (Emalta) y la proteína de grano (Pgrano) en el modelo.
El ajuste se hace a través de la función de mínimos cuadrados que se encuentra en el anexo, para la cual los valores de a y b están representados en la figura a continuación:



c) ¿Puede proponer otro modelo que ajuste mejor sus datos?
Se puede proponer un modelo de la forma:

$$\text{Emalta} = \alpha \cdot \text{Pgrano}^N$$

con α y N a estimar desde la función de mínimos cuadrados que se encuentra en el anexo. La siguiente figura muestra dicho ajuste con los valores de α y N:



Para determinar cual de los dos modelos propuestos se ajusta mejor, se puede calcular la suma de cuadrados del error, donde para el primer modelo (ecuación de una recta):

$$S = \sum_{i=1}^{15} (\text{Emalta}_i - a \cdot \text{Pgrano}_i - b)^2 = 46.615$$

y para el caso de una función de tipo potencia:

$$S = \sum_{i=1}^{15} \left(\text{Emalta}_i - \alpha \cdot \text{Pgrano}_i^N\right)^2 = 46.947$$

En ambos caso el error es simular, aún así el que tiene el menor error es el primer modelo de una recta.

Anexo

Pregunta 1

Listing 1: Pregunta 1a

```
import pandas as pd
import matplotlib.pyplot as plt

data=pd.read_csv('Automobile.csv',sep=';')

plt.figure(1)
plt.plot(data['length'],data['width'],'o',color = 'blue')
plt.ylabel('length')
plt.xlabel('width')
plt.savefig('pregunta1_fig1.pdf',dpi=350)

plt.figure(2)
plt.plot(data['engine-size'],data['length'],'o',color = 'red')
plt.ylabel('engine-size')
plt.xlabel('length')
plt.savefig('pregunta1_fig2.pdf',dpi=350)

plt.figure(3)
plt.plot(data['Wheel-base'],data['curb-weight'],'o',color = 'green')
plt.ylabel('wheel-base')
plt.xlabel('curb-weight')
plt.savefig('pregunta1_fig3.pdf',dpi=350)
```

Listing 2: Pregunta 1b

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

def min_cuad(x,y):
### esta funcion entrega la pendiente y el intercepto
   m = len(x) #numero de datos
   Sumaxy = np.dot(x,y) #suma de los productos cruzados Suma(xi*yi)
   Sumax = np.sum(x) #Suma de los elementos de x
   Sumay = np.sum(y) #Suma de los elementos de y
   Sumaxx = np.dot(x,x) #Suma de los elementos de x al cuadrado
   a = (m*Sumaxy - Sumax*Sumay)/(m*Sumaxx - (Sumax)**2) ## esta es la pendiente
   b = (Sumaxx*Sumay - Sumaxy*Sumax)/(m*Sumaxx - (Sumax)**2)
   return a,b

data=pd.read_csv('Automobile.csv',sep=';')

plt.figure(1)
length=data['length']
width=data['width']
pendiente1,intercepto1=min_cuad(length,width)
xnew = np.linspace(np.min(length),np.max(length),100)
ynew = pendiente1*xnew + intercepto1
plt.plot(xnew,ynew,'-',color='black',label=r'$width=%0.3f\cdot length+%0.3f$'%(
    pendiente1,intercepto1))
plt.plot(length,width,'o',color = 'blue',label='Datos')
plt.ylabel('length')
plt.xlabel('width')
plt.legend(loc=4)
plt.savefig('pregunta1_ajuste_fig1.pdf',dpi=350)

plt.figure(2)
esize=data['engine-size']
length=data['length']
pendiente2,intercepto2=min_cuad(esize,length)
xnew = np.linspace(np.min(esize),np.max(esize),100)
ynew = pendiente2*xnew + intercepto2
plt.plot(xnew,ynew,'-',color='black',label=r'$horsepower=%0.3f\cdot  length+%0.3f$'%(
    pendiente2,intercepto2))
plt.plot(esize,length,'o',color = 'red',label='Datos')
plt.ylabel('engine-size')
plt.xlabel('length')
plt.legend(loc=4)
plt.savefig('pregunta1_ajuste_fig2.pdf',dpi=350)

plt.figure(3)
wbase=data['Wheel-base']
cweight=data['curb-weight']
pendiente3,intercepto3=min_cuad(wbase,cweight)
xnew = np.linspace(np.min(wbase),np.max(wbase),100)
ynew = pendiente3*xnew + intercepto3
plt.plot(xnew,ynew,'-',color='black',label=r'$curbweight=%0.3f\cdot wheelbase%0.3f$'
    %(pendiente3,intercepto3))
plt.plot(wbase,cweight,'o',color = 'green',label='Datos')
plt.ylabel('wheel-base')
plt.xlabel('curb-weight')
plt.legend(loc=4)
plt.savefig('pregunta1_ajuste_fig3.pdf',dpi=350)
```

Pregunta 2

Listing 3: Pregunta 2a y 2b

```
import pandas as pd
import matplotlib.pylab as plt
import numpy as np

def min_cuad(x,y):
### esta funcion entrega la pendiente y el intercepto
   m = len(x) #numero de datos
   Sumaxy = np.dot(x,y) #suma de los productos cruzados Suma(xi*yi)
   Sumax = np.sum(x) #Suma de los elementos de x
   Sumay = np.sum(y) #Suma de los elementos de y
   Sumaxx = np.dot(x,x) #Suma de los elementos de x al cuadrado
   a = (m*Sumaxy - Sumax*Sumay)/(m*Sumaxx - (Sumax)**2) ## esta es la pendiente
   b = (Sumaxx*Sumay - Sumaxy*Sumax)/(m*Sumaxx - (Sumax)**2)
   return a,b

precios_casas = pd.read_csv('precios_casas.csv',sep=',')
tama = precios_casas['tama']
precio = precios_casas['precio']

plt.figure(1)
plt.plot(tama,precio,'o',color='blue',label='Datos')
plt.xlabel('Tama o')
plt.ylabel('Precio')
plt.legend()
plt.savefig('pregunta2a.pdf',dpi=350)

plt.clf()
pendiente,intercepto=min_cuad(tama,precio)
xnew = np.linspace(np.min(tama),np.max(tama),100)
ynew = pendiente*xnew + intercepto
plt.figure(1)
plt.plot(xnew,ynew,'-',color='green',label=r'$Precio=%0.3f\cdot tama o+%0.3f$'%(
    pendiente,intercepto))
plt.plot(tama,precio,'o',color='blue',label='Datos')
plt.xlabel('Tama o')
plt.ylabel('Precio')
plt.legend()
plt.savefig('pregunta2b.pdf',dpi=350)

print 'Precio de una casa de 2200 pies^2=',pendiente*22+intercepto
print 'Suma de minimos cuadrados=',np.dot(precio-pendiente*tama-intercepto,precio-
    pendiente*tama-intercepto)
```

Listing 4: Pregunta 2c

```
import pandas as pd
import matplotlib.pylab as plt
import numpy as np

def min_cuadexpN(datax,datay):
    # se quiere ajustar un modelo de la forma y = alpha*x**n
    # se encontran a y n
    m = len(datax) #numero de datos
    ### se transforman los datos
    indices_0 = np.where(datay!=0)[0]
    lnx = np.log(datax[indices_0])
    lny = np.log(datay[indices_0])
    Sumaxy = np.dot(lnx,lny) #suma de los productos cruzados Suma(xi*yi)
    Sumax = np.sum(lnx) #Suma de los elementos de x
    Sumay = np.sum(lny) #Suma de los elementos de y
    Sumaxx = np.dot(lnx,lnx) #Suma de los elementos de x al cuadrado
    a = (m*Sumaxy - Sumax*Sumay)/(m*Sumaxx - (Sumax)**2) ## esta es la pendiente
    b = (Sumaxx*Sumay - Sumaxy*Sumax)/(m*Sumaxx - (Sumax)**2)
    n = a
    alpha = np.exp(b)
    return alpha,n

precios_casas = pd.read_csv('precios_casas.csv',sep=',')
tama = precios_casas['tama']
precio = precios_casas['precio']

alphaval,Nval=min_cuadexpN(tama,precio)
xnew = np.linspace(np.min(tama),np.max(tama),100)
ynew = alphaval*xnew**Nval
plt.figure(1)
plt.plot(xnew,ynew,'-',color='green',label=r'$Precio=%0.3f\cdot tama o^{%0.3f}$'%(
    alphaval,Nval))
plt.plot(tama,precio,'o',label='Datos')
plt.xlabel('Tama o')
plt.ylabel('Precio')
plt.legend()
plt.savefig('pregunta2c.pdf',dpi=350)

print 'Precio de una casa de 2200 pies^2=',alphaval*22**Nval
print 'Suma de minimos cuadrados=',np.dot(precio-alphaval*tama**Nval,precio-alphaval*
    tama**Nval)
```

Pregunta 3

Listing 5: Pregunta 3a y 3b

```
import numpy as np
import matplotlib.pylab as plt
import pandas as pd

def min_cuad(x,y):
### esta funcion entrega la pendiente y el intercepto
   m = len(x) #numero de datos
   Sumaxy = np.dot(x,y) #suma de los productos cruzados Suma(xi*yi)
   Sumax = np.sum(x) #Suma de los elementos de x
   Sumay = np.sum(y) #Suma de los elementos de y
   Sumaxx = np.dot(x,x) #Suma de los elementos de x al cuadrado
   a = (m*Sumaxy - Sumax*Sumay)/(m*Sumaxx - (Sumax)**2) ## esta es la pendiente
   b = (Sumaxx*Sumay - Sumaxy*Sumax)/(m*Sumaxx - (Sumax)**2)
   return a,b

data = pd.read_csv('ejercicio3.csv',delimiter=';')
Pgrano = data['Proteina_de_grano']
Emalta = data['Extracto_de_malta']

plt.figure(1)
plt.plot(Pgrano,Emalta,'o',label='Datos')
plt.xlabel('Proteina␣de␣grano')
plt.ylabel('Extracto␣de␣Malta')
plt.legend()
plt.savefig('pregunta3a.pdf',dpi=350)

pendiente,intercepto=min_cuad(Pgrano,Emalta)
xnew = np.linspace(np.min(Pgrano),np.max(Pgrano),100)
ynew = pendiente*xnew + intercepto
plt.figure(2)
plt.plot(xnew,ynew,'-',color='green',label=r'$Emalta=%0.3f\cdot␣Pgrano+%0.3f$'%(
    pendiente,intercepto))
plt.plot(Pgrano,Emalta,'o',label='Datos')
plt.xlabel('Proteina␣de␣grano')
plt.ylabel('Extracto␣de␣Malta')
plt.legend()
plt.savefig('pregunta3b.pdf',dpi=350)


print 'Suma␣de␣minimos␣cuadrados=',np.dot(Emalta-(pendiente*Pgrano+intercepto),Emalta
    -(pendiente*Pgrano+intercepto))
```

Listing 6: Pregunta 3c

```python
import numpy as np
import matplotlib.pylab as plt
import pandas as pd

def min_cuadexpN(datax,datay):
    # se quiere ajustar un modelo de la forma y = alpha*x**n
    # se encontran a y n
    m = len(datax) #numero de datos
    ### se transforman los datos
    indices_0 = np.where(datay!=0)[0]
    lnx = np.log(datax[indices_0])
    lny = np.log(datay[indices_0])
    Sumaxy = np.dot(lnx,lny) #suma de los productos cruzados Suma(xi*yi)
    Sumax = np.sum(lnx) #Suma de los elementos de x
    Sumay = np.sum(lny) #Suma de los elementos de y
    Sumaxx = np.dot(lnx,lnx) #Suma de los elementos de x al cuadrado
    a = (m*Sumaxy - Sumax*Sumay)/(m*Sumaxx - (Sumax)**2) ## esta es la pendiente
    b = (Sumaxx*Sumay - Sumaxy*Sumax)/(m*Sumaxx - (Sumax)**2)
    n = a
    alpha = np.exp(b)
    return alpha,n

data = pd.read_csv('ejercicio3.csv',delimiter=';')
Pgrano = data['Proteina_de_grano']
Emalta = data['Extracto_de_malta']

alphaval,Nval=min_cuadexpN(Pgrano,Emalta)
xnew = np.linspace(np.min(Pgrano),np.max(Pgrano),100)

ynew = alphaval*xnew**Nval
plt.figure(1)
plt.plot(xnew,ynew,'-',color='green',label=r'$Emalta=%0.3f\cdot Pgrano^{%0.3f}$'%(
    alphaval,Nval))
plt.plot(Pgrano,Emalta,'o',label='Datos')
plt.xlabel('Proteina de grano')
plt.ylabel('Extracto de Malta')
plt.legend()
plt.savefig('pregunta3c.pdf',dpi=350)

print 'Suma de minimos cuadrados=',np.dot(Emalta-alphaval*Pgrano**Nval,Emalta-
    alphaval*Pgrano**Nval)
```