

Asignación 1

David Nogales Pérez

17 de octubre de 2021

Problema 2: Mínimos cuadrados

Tenemos una secuencia de números reales x_1, \dots, x_n . Definimos la función:

$$f(x) = \frac{1}{n} \sum_{i=1}^n (x - x_i)^2$$

(a) Demostrar que f tiene un único mínimo, obtenerlo e interpretar el resultado.

1era. derivada:

aplicando la regla de la cadena

$$f'(x) = \frac{1}{n} \sum_{i=1}^n 2(x - x_i)(1)$$

$$f'(x) = \frac{1}{n} \sum_{i=1}^n 2x - 2x_i$$

$$f'(x) = \frac{\sum_{i=1}^n 2x}{n} - \frac{\sum_{i=1}^n 2x_i}{n}$$

$$f'(x) = \frac{2x\mathcal{N}}{\mathcal{N}} - \frac{\sum_{i=1}^n 2x_i}{n}$$

$$f'(x) = 2x - 2 \frac{\sum_{i=1}^n x_i}{n}$$

Igualando a cero:

$$2x - 2 \frac{\sum_{i=1}^n x_i}{n} = 0 (\div 2)$$
$$x = \frac{\sum_{i=1}^n x_i}{n}$$

El mínimo obtenido de la función es igual a la media de la muestra.

2da. derivada:

$$f''(x) = 2 - 2 \frac{\sum_{i=1}^n \cancel{x_i}}{n}$$
$$f''(x) = 2$$

Dado que $f''(x) > 0$ el valor encontrado en la primera derivada es un mínimo y es único.

(b) Considerando la siguiente función:

$$f(x) = \sum_{i=1}^n p_i (x - x_i)^2$$

Donde ponderamos cada termino por un factor $p_i > 0$, de manera que $\sum_{i=1}^n p_i = 1$. Recalcular la solución.

1era. derivada:

$$f'(x) = \sum_{i=1}^n 2p_i (x - x_i)(1)$$

$$f'(x) = \sum_{i=1}^n 2p_i x - 2p_i x_i$$

$$f'(x) = 2x \sum_{i=1}^n p_i - 2 \sum_{i=1}^n p_i x_i$$

Igualando a cero:

$$\cancel{2}x \sum_{i=1}^n \cancel{p_i} - \cancel{2} \sum_{i=1}^n p_i x_i = 0(\div 2)$$

$$x = \sum_{i=1}^n p_i x_i$$

El valor obtenido de la función es igual a la suma de los elementos ponderados de la muestra.

2da. derivada:

$$f''(x) = 2\cancel{x} \sum_{i=1}^n p_i$$

$$f''(x) = 2 \sum_{i=1}^n \cancel{p_i}$$

$$f''(x) = 2$$

Dado que $f''(x) > 0$ el valor encontrado en la primera derivada es un mínimo y es único.

(c) **[PROG]** Sea $n = 100$. Aplicar el resultado precedente a una secuencia de números escogida. Usar una ponderación basada en números independientes uniformes en $(0, 1)$. Dibujar la función y el mínimo.

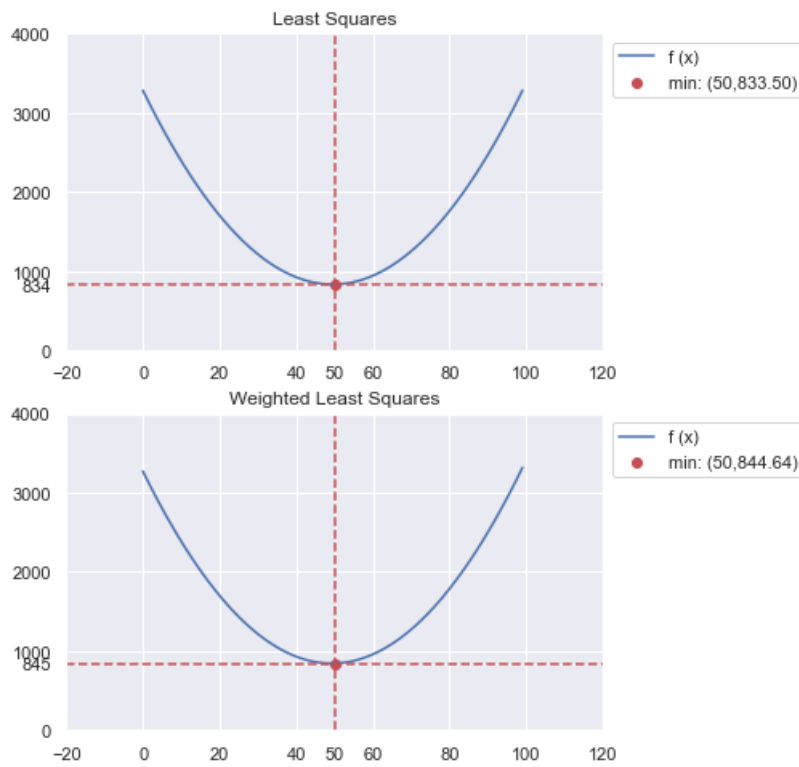


Figura 1: Gráficas de las funciones de los apartados **a)** y **b)** (Código en Anexo I)

Anexo I

Código con funciones y librerías usadas:

```
1 import numpy as np
2 import seaborn as sns
3 import pandas as pd
4 import matplotlib.pyplot as plt
5
6 def least_squares(x, df):
7     return ((x-df["x_i"])**2).sum()/len(df["x_i"])
8
9 def weighted_least_squares(x, df):
10    return (df["p_i"]*((x-df["x_i"])**2)).sum()
11
12 def make_plot(new_ax, minimum_x, minimum_y, df, plot_title):
13    sns.set_style("whitegrid")
14    df.plot(ax=new_ax, label='f (x)', title=plot_title)
15    point_label = "min: ({}, {:.2f})"
16    point_label = point_label.format(minimum_x, minimum_y)
17    new_ax.scatter(x=minimum_x, y=minimum_y, c='r', label=point_label)
18    new_ax.axvline(minimum_x, color='r', ls='—')
19    new_ax.axhline(minimum_y, color='r', ls='—')
20    myticks = np.append(new_ax.get_xticks(), [minimum_x])
21    new_ax.set_xticks(myticks)
22    myticks = np.append(new_ax.get_yticks(), [minimum_y])
23    new_ax.set_yticks(myticks)
24    new_ax.legend(bbox_to_anchor=(1, 1))
25    return new_ax
```

Código para generar los datos y obtener las soluciones:

```
1 data = np.arange(start=1, stop=101)
2
3 #Generating weights with sum 1
4 np.random.seed(1234)
5 number_ponderation = np.random.uniform(low=1, high=2, size=len(data))
6 df = pd.DataFrame(data={'x_i': data, 'p_i': number_ponderation})
7 df['p_i'] = df['p_i']/df['p_i'].sum()
8 weights_sum = "Distribution sum: {:.2f}"
9 print(weights_sum.format(df['p_i'].sum()))
10
11 #Applying functions to data sample and calculating the minimum values
12 df_ls = df["x_i"].apply(lambda x: least_squares(x, df))
13 minimum_ls_x = round(df["x_i"].mean())
14 minimum_ls_y = least_squares(minimum_ls_x, df)
15
16 df_wls = df["x_i"].apply(lambda x: weighted_least_squares(x, df))
17 minimum_wls_x = round((df["p_i"]*df["x_i"]).sum())
18 minimum_wls_y = weighted_least_squares(minimum_wls_x, df)
```

Código para generar los gráficos:

```
1 #Generating plots of functions
2 fig, (ax1, ax2) = plt.subplots(2, 1)
3 ax1 = make_plot(new_ax=ax1, minimum_x=minimum_ls_x,
4                 minimum_y=minimum_ls_y, df=df_ls, plot_title='Least Squares')
5 ax2 = make_plot(new_ax=ax2, minimum_x=minimum_wls_x,
6                 minimum_y=minimum_wls_y, df=df_wls, plot_title='Weighted Least Squares')
7 fig.set_size_inches(6, 8)
```
