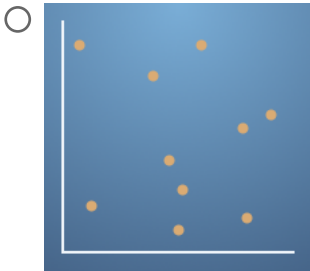


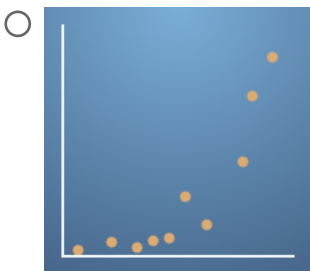
1. En el vídeo anterior viste cómo encajar una línea $y = mx + c$ a datos lineales. En este cuestionario, practicarás la identificación de datos apropiados para la regresión lineal e inicializarás algunos ajustes usted mismo.

1 / 1 punto

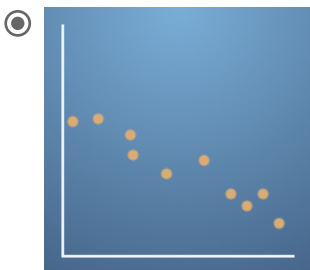
¿Cuál de las siguientes figuras parece contener datos sensibles para un ajuste lineal?



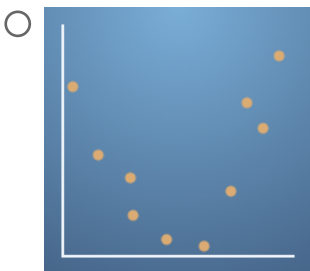
\$\$\$\$



\$\$\$\$



\$\$\$\$



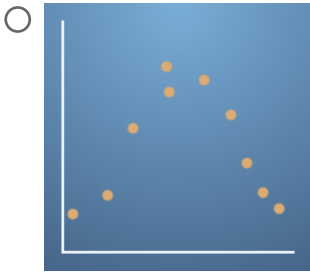
\$\$\$\$

☒ **Correcto**

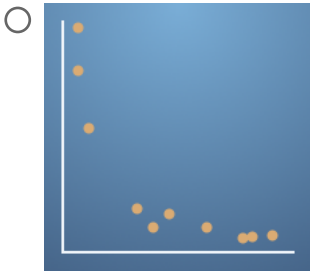
Estos datos parecen lineales, con gradiente negativo. *metro*

2. ¿Cuál de las siguientes figuras parece contener datos sensibles para un ajuste lineal?

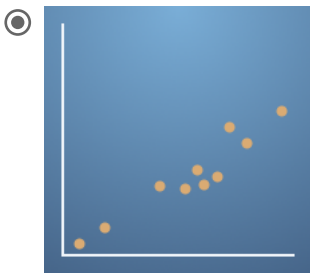
1 / 1 punto



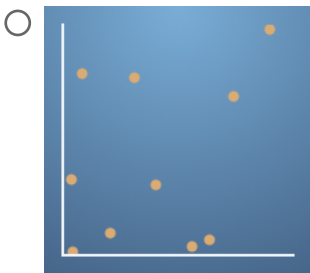
\$\$\$\$



\$\$\$\$



\$\$\$\$



\$\$\$\$

✓ Correcto

Parece que estos datos podrían ser lineales con positivometroy pequeñaC

3. Ahora que hemos identificado candidatos para la regresión lineal, podemos hacer algunos ajustes lineales nosotros mismos. El bloque de código a continuación traza algunos puntos de datos predefinidos y una regresión lineal con los valores[metro , do], dondemetroes el gradiente yCes el intercepto en y. También da la h^2 valor discutido en el video anterior, que es una medida de qué tan bueno es el ajuste.

1 / 1 punto

Juega con los valores demetroyCpara tener una idea de cómo afectan los diferentes ajustes lineales h^2 , luego intente encontrar el mejor ajuste posible a los datos.

```
1 # Veamos lo que m y c hacen al ajuste
2 m = -0,22 ; c = 0,76
3 pag = [m,c]
4 línea (pag)
```

Ejecutar

Restablecer

El mínimo h^2 el valor es 0.03819 a 4 cifras significativas. Trate de encontrar un ajuste con $h^2 \leq 0.04$ y luego ingrese estos valores en el siguiente bloque de código:

```
1 # Reemplace m y c con valores que minimicen  $\chi^2$ .
2 p = [ -.22 , 0.76 ]
```

Ejecutar

Restablecer

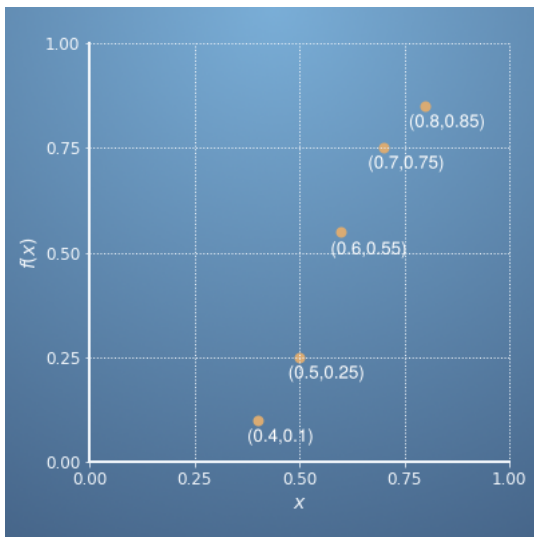
✓ **Correcto**

¡Bien hecho! Encontró un modelo que se ajusta aceptablemente bien a los datos según el criterio definido para χ^2 .

4. Ajustar a simple vista no es tan fácil, incluso para pequeños conjuntos de datos. Hagamos algunos ajustes lineales usando las matemáticas discutidas en el video anterior.

1 / 1 punto

La siguiente es una figura con 5 puntos de datos, etiquetados con sus (X, y) coordenadas:



Ajustemos una regresión lineal a esta pequeña muestra a mano. Recordemos que podemos usar h^2 para medir qué tan bueno es nuestro ajuste, definido por

$$h^2 = \sum (y_i - mx_i - c)^2,$$

y que podemos encontrar el mínimo de h^2 diferenciándolo y poniéndolo a cero. Esto nos lleva a las ecuaciones de metro y C,

$$metro = \frac{\sum (X_i - \bar{X})y_i}{\sum (X_i - \bar{X})^2}, \quad C = \bar{y} - metro\bar{X},$$

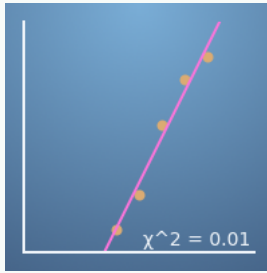
que minimizan h^2 .

Utilice estas ecuaciones para calcular la metro y C que minimizan h^2 para los 5 puntos de datos dados arriba y seleccione los valores correctos a continuación:

- ☐ metro = -1.8, C = 0.7
- ☐ metro = 1.8, C = -0.6
- ☐ metro = 1.6, C = -0.5
- ☒ metro = 2, C = -0.7

✓ Correcto

Aquí está el ajuste:



5. Como ha visto, puede ser un gran esfuerzo ajustar incluso 5 puntos de datos al hacer los cálculos a mano. A menudo, es necesario trabajar con conjuntos de datos mucho más grandes, así que consideremos un nuevo ejemplo con 50 puntos de datos. En lugar de hacerlo a mano, implementaremos una función para que haga los cálculos por nosotros.

1 / 1 punto

Ejecute primero el siguiente bloque de código para ver los datos sin ningún tipo de ajuste lineal. La función *linfit* se define dentro del bloque de código. Su tarea es editar la definición para que *linfit* tome la matriz de datos *x*, *xdat*, y la matriz de datos *y*, *ydat*, y devuelva la correcta *metroyC* para crear un ajuste lineal que minimice h^2 .

el cálculo de \bar{x} , \bar{y} , \bar{y} , \bar{y} , ya está dado. Como puede ver, *numpy* se ha importado como *np*.

```
1 # Aquí se define la función
2 def linfit(xdat,ydat):
3     imprimir (xdat)
4     imprimir (ydat)
5     # Aquí se calculan xbar y ybar
6     xbarra = np. suma (xdat)/ len (xdat)
7     ybarra = np. sum (ydat)/ len (ydat)
8
9     # Inserte el cálculo de m y c aquí. Si no hay nada aquí, los datos se trazará
10
11     # Devuelve tus valores como [m, c]
12     volver [m, c]
13
14 # Produce la trama - no pongas esto en el siguiente bloque de código
15 línea()
16
```

Ejecutar

Restablecer

Utilice el bloque de código anterior para probar su código. Cuando esté seguro de haber definido correctamente la función, colóquela en el siguiente bloque de código y ejecútela, teniendo cuidado de no incluir *line()* en su respuesta.

```
1 # Aquí se define la función
2 def linfit(xdat,ydat):
3     imprimir (xdat)
4     imprimir (ydat)
5     # Aquí se calculan xbar y ybar
6     xbarra = np. suma (xdat)/ len (xdat)
7     ybarra = np. sum (ydat)/ len (ydat)
8
9     # Inserte el cálculo de m y c a continuación
10    x_numerador = np.sum((xdat-xbar)*ydat)
11    x_denominador = np.sum((xdat-xbar)**2)
12    m = x_numerador/x_denominador
13    c = ybarra - (m * xbarra)
14
15    # Devuelve tus valores como [m, c]
16    volver [m, c]
17    # No incluyas line() en esta respuesta
```

Ejecutar

Restablecer

✓ Correcto

Bien hecho, has encontrado correctamente los valores de m y c.

6. Si bien es informativo escribir el código nosotros mismos, como en la pregunta anterior, en la práctica, las funciones que realizan varios tipos de regresión se implementan en muchos lenguajes de programación. Hay varios de estos en python.

1 / 1 punto

Uno de esos ejemplos es el método `scipy.stats.linregress()`, que toma matrices de datos `x` e `y` exactamente de la misma manera que la función `linfit()` que definió en la pregunta anterior. Como salida da la pendiente `metroe` interceptarCasí como algunas medidas estadísticas útiles como el error estándar.

En el siguiente bloque de código, los datos `x` se almacenan nuevamente en la matriz `xdat` y los datos `y` en la matriz `ydat`. Llame al método `stats.linregress()` con los argumentos de datos y luego pase el resultado a `line()` para trazar la regresión.

```
1 de las estadísticas de importación de Scipy
2
3 # Use el método stats.linregress() para evaluar la regresión
4 regresión = stats.linregress(xdat, ydat)
5
6 línea (regresión)
7
```

Ejecutar

Restablecer

Esperemos que quede claro que `linregress()` hace todo lo que hizo `linfit()` y más, ¡sin tener que escribirlo usted mismo!

Una vez que esté satisfecho de haber implementado las cosas correctamente en el bloque de código anterior, repita lo mismo en el siguiente bloque de código sin la última línea para completar la pregunta.

```
1 de las estadísticas de importación de Scipy
2
3 # Use el método stats.linregress() para evaluar la regresión
4 regresión = stats.linregress(xdat, ydat)
5
6 # No use línea (regresión) en este cuadro de código
```

Ejecutar

Restablecer

✓ Correcto

Bien hecho,
ha asignado 'regresión' correctamente

