

```
In [1]: %use lets-plot
```

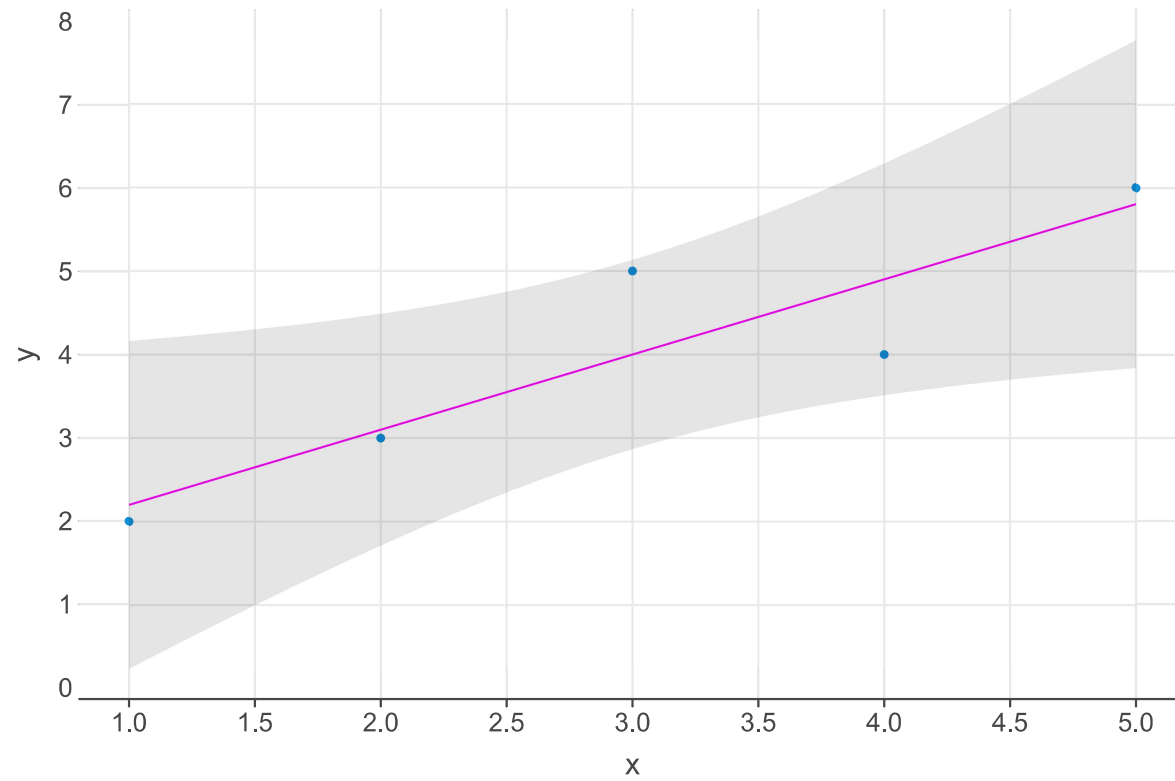
```
In [2]: // Crear un conjunto de datos con los valores de x e y para utilizar en el objeto letsPlot
val d = mapOf(
    "x" to listOf(1.0, 2.0, 3.0, 4.0, 5.0),
    "y" to listOf(2.0, 3.0, 5.0, 4.0, 6.0)
)
```

```
In [3]: // Creamos un objeto plot, que es una representación gráfica de los datos:
//      La función recibe dos parámetros: el primero es el mapa de datos `dat`,
//                                             y el segundo es una expresión lambda que indica qué variables usar
//                                             para el eje x y el eje y del gráfico.
val p = letsPlot(d) {x="x"; y="y"}

// Se usa el operador `+` para añadir dos capas geométricas al objeto plot:
//      La función `geomPoint()` dibuja un punto por cada fila de los datos,
//      y la función `geomSmooth()` dibuja una línea suavizada que muestra la tendencia de los datos.
//mpg_plot + geomPoint() + geomSmooth()

//En este caso forzamos que suavice con -> statSmooth(method = "lm", level = 0.95)
//      El método "lm" significa regresión lineal
//      Establecemos un nivel de confianza (entre 0 y 1) del 95%
//      nota: nivel de confianza establece un rango de valores que es probable que contenga el verdadero valor
//      del parámetro que estamos estimando
p + geomPoint() + statSmooth(method = "lm", level = 0.95)
```

Out[3]:



In [ ]: