

In [8]: %use lets-plot

In [11]: `// Crear un conjunto de datos con los valores de x e y que quieres graficar`
`val d = mapOf(`
 `"x" to listOf(1.0, 3.0, 5.0, 7.0, 9.0, 11.0, 13.0, 15.0, 17.0, 19.0),`
 `"y" to listOf(2.0, 3.0, 5.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0, 20.0)`
`)`

In [12]: `// 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 la transformación`
`// 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.`
`//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`
`// del parámetro que estamos estimando`
`p + geomPoint() + statSmooth(method = "lm", level = 0.95)`

Out[12]:

