Let’s see these **ggplot/plotnine statements** line by line how do they work together.

**The Statement**

plot = (

ggplot(df, aes(x="Hours Worked", y="Money Earned"))

+ geom\_line()

+ labs(

title="Hours Worked vs Money Earned",

x="Hours Worked",

y="Money Earned (₹)"

)

)

**Step 1: ggplot(df, aes(...))**

ggplot(df, aes(x="Hours Worked", y="Money Earned"))

* **ggplot()** → this is the **main plotting function**. Think of it as creating a **blank canvas** for your plot.
* **df** → the **data** you want to plot. In ggplot, you always give it a **DataFrame**.
* **aes()** → short for **aesthetics**, it tells ggplot **what to put on x-axis and y-axis**.
  + x="Hours Worked" → put the "Hours Worked" column on the X-axis
  + y="Money Earned" → put the "Money Earned" column on the Y-axis

It’s like we have a blank paper (ggplot) and we specify which columns go on X and Y (aes).”

**Step 2: geom\_line()**

+ geom\_line()

* **geom\_line()** → tells ggplot **how to draw the data**.
* “geom” = geometry → **the shape/representation of data**.
  + geom\_line() → connect the data points with a **line**
  + geom\_point() → would draw **points**
  + geom\_bar() → would draw **bars**

We have a paper with data mapped to axes. Now we decide, should we draw points, lines, bars…? Here we choose a line.”

**Step 3: labs()**

+ labs(

title="Hours Worked vs Money Earned",

x="Hours Worked",

y="Money Earned (₹)"

)

* **labs()** → short for **labels**
* Adds **title, x-axis label, y-axis label** to the plot
* Makes the plot **readable and meaningful** for humans

We drew the line, now we put a title and label our axes so everyone understands the graph.”

**Step 4: plot = (...)**

* The **whole combination** is stored in a variable called plot
* It is **not yet drawn**. It’s like a **recipe**: it describes what the plot should look like, but nothing appears until we **draw it**.
* We can display it using:

plot.draw() # in PyCharm / scripts

or

print(plot) # in Jupyter notebook

**Summary:-**

1. ggplot(df, aes(...)) → creates canvas and map data
2. geom\_line() → decides how data is drawn (line)
3. labs(...) → adds labels and title
4. Assigns to plot → store the “recipe” for the plot

This is the **Grammar of Graphics approach**: data + aesthetics + geometry + labels → final plot.

