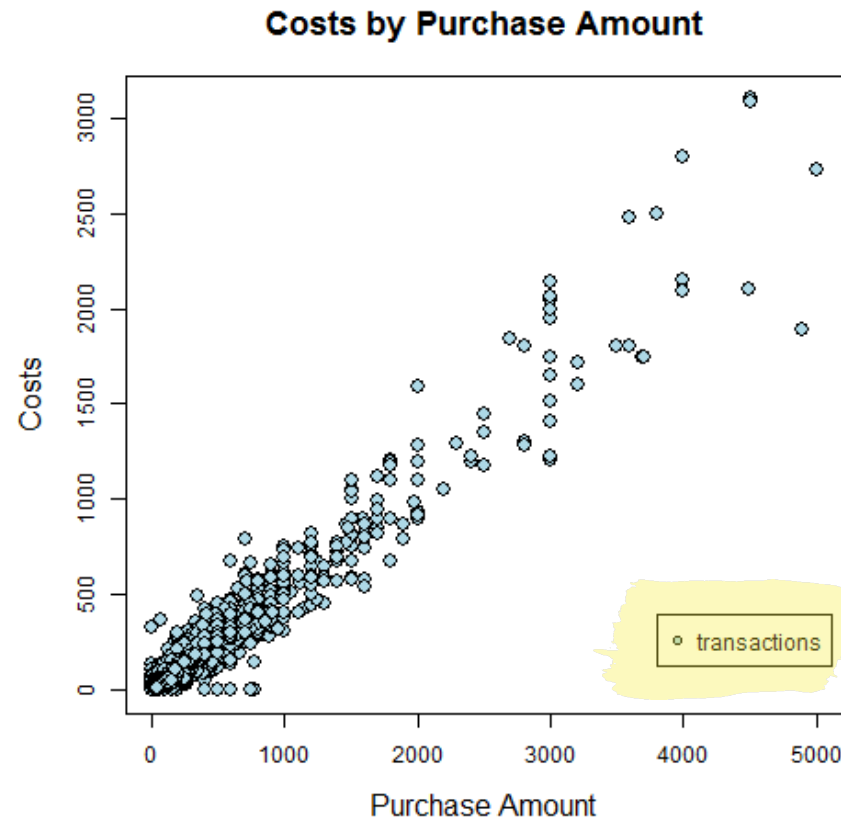


Adding further features to plots

Further improve the aesthetic features of a plot: Add a legend to make your plot self-explanatory



The first numbers indicate x and y coordinates of the legend (upper left edge). You can also specify, e.g., "bottomright".

```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)  
legend(3800, 400, "transactions", pch=21, pt.bg="lightblue")
```

Further improve the aesthetic features of a plot: Lines can be added for extra information



Draw the regression line: ①

$y = a + b \cdot x$

a = Slope

b = Intercept

```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)  
abline(lm(myData[, Cost] ~ myData[, PurchAmount]), col="red")
```

Further improve the aesthetic features of a plot: Lines can be added for extra information



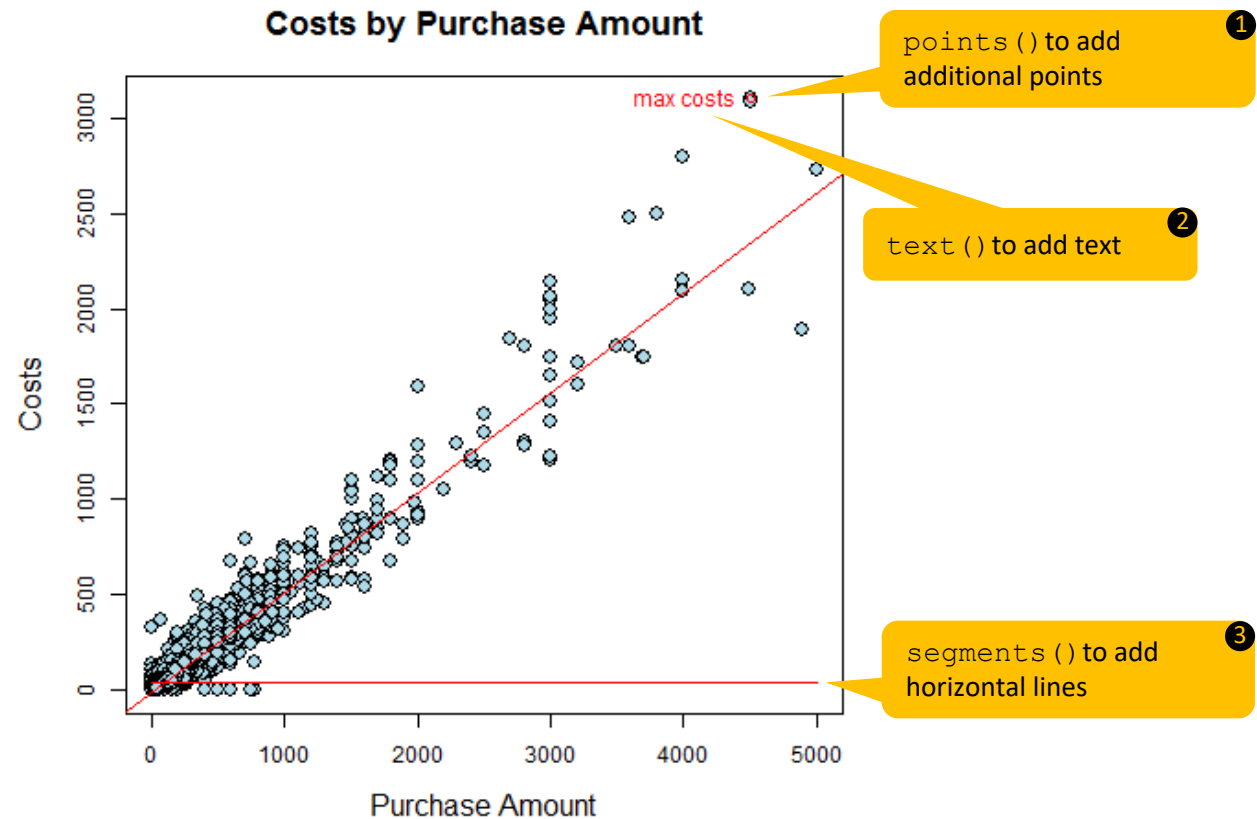
1 Draw the regression line:
 $y = a + b \cdot x$
 a = Slope
 b = Intercept

3 Specify the line color

```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)
abline(lm(myData[, Cost] ~ myData[, PurchAmount]), col="red")
```

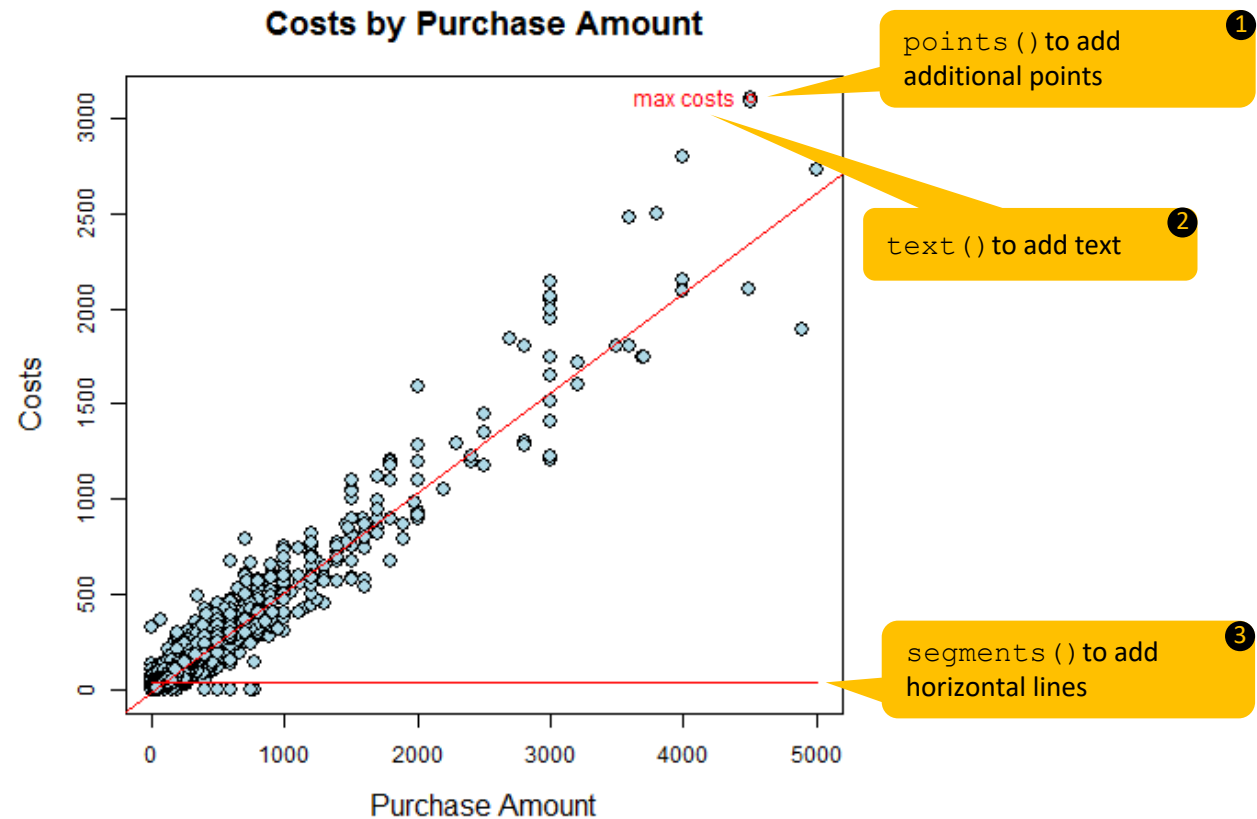
2 To the left of the ~-operator you specify the dependent variable and to the right the explanator variables

Additional graphical elements can be added in the same fashion



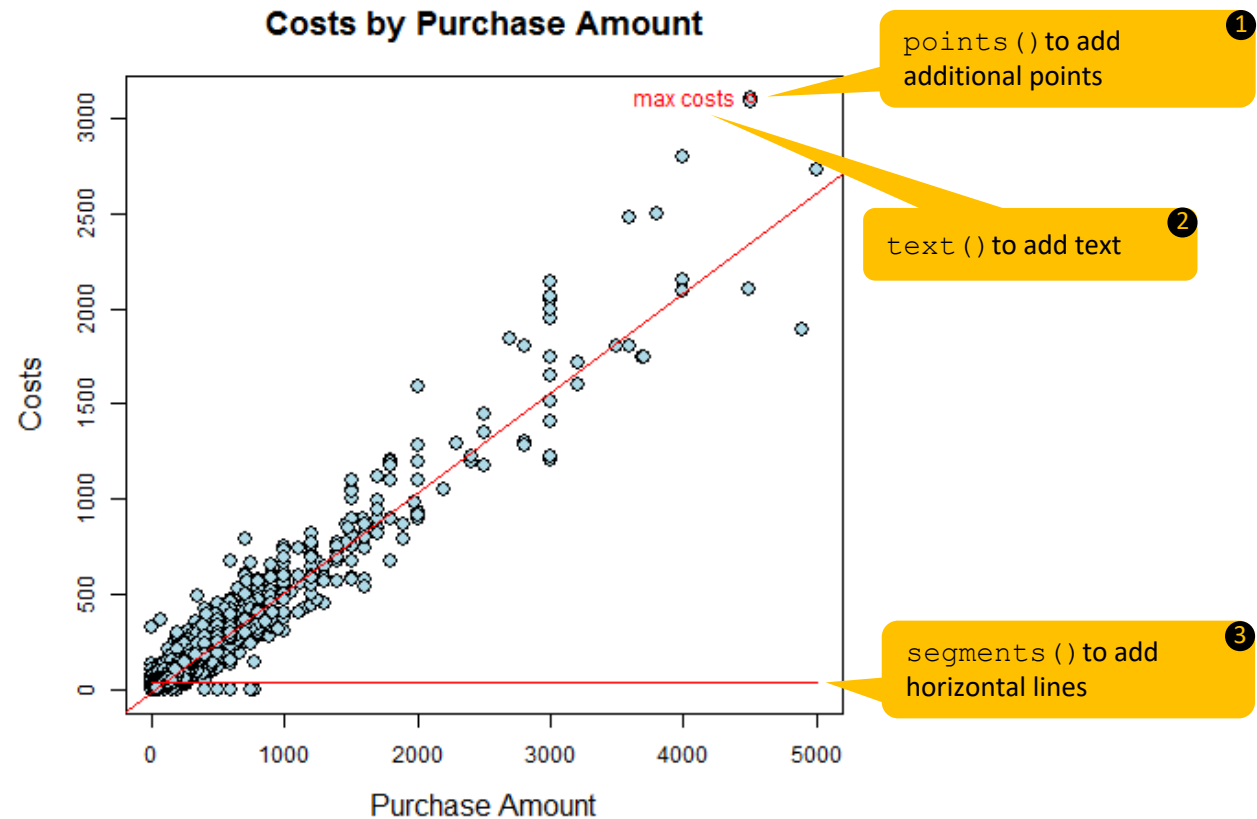
```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)
  points(4500, 3100, col="red")
text(4500,3100, "max costs", col="red", cex=1, pos=2)
segments(0,39,5000,39,col="red")
```

Additional graphical elements can be added in the same fashion



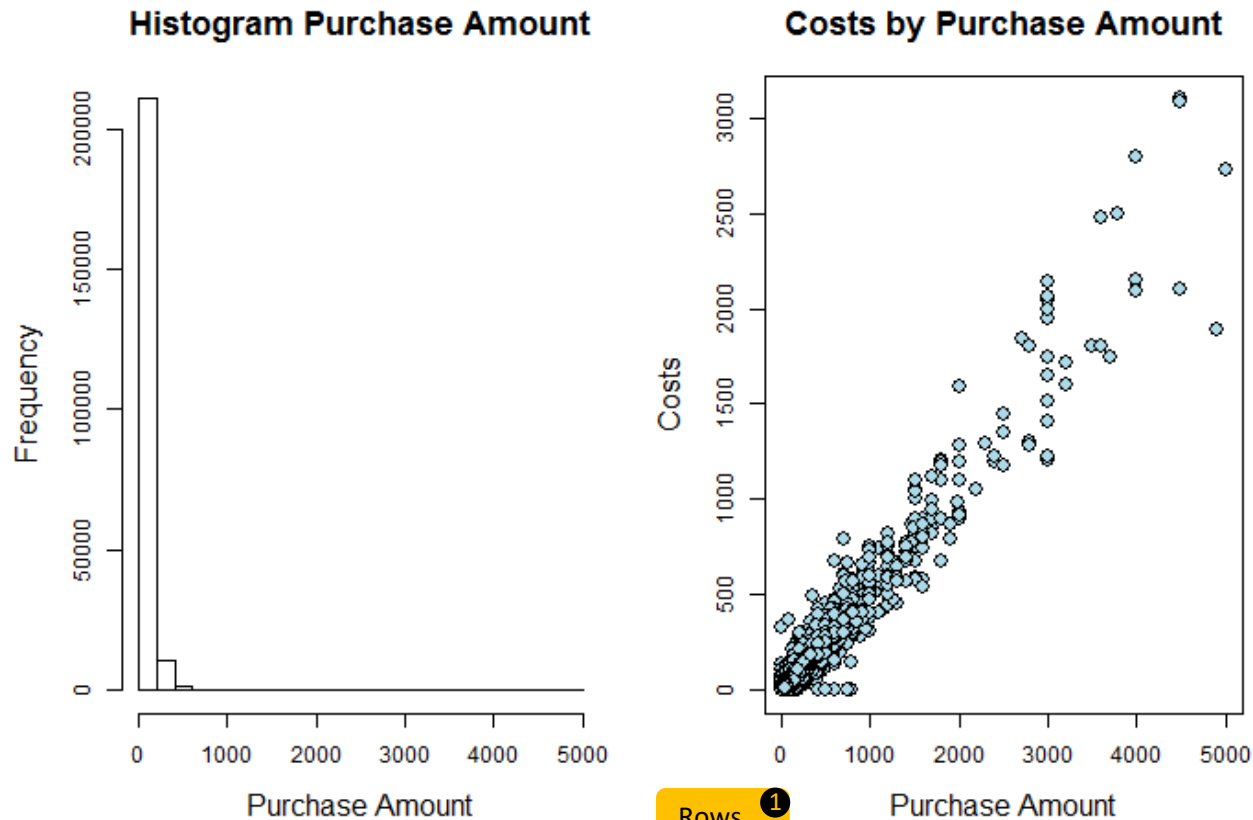
```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)
      points(4500, 3100, col="red")
text(4500,3100, "max costs", col="red", cex=1, pos=2)
segments(0,39,5000,39,col="red")
```

Additional graphical elements can be added in the same fashion



```
plot(x=myData[, PurchAmount], y=myData[, Cost],.....)
      points(4500, 3100, col="red")
text(4500,3100, "max costs", col="red", cex=1, pos=2)
      segments(0,39,5000,39,col="red")
```

You might want to plot multiple graphs in one image



Rows ①

Columns ②

```
par(mfrow = c(1, 2))
```

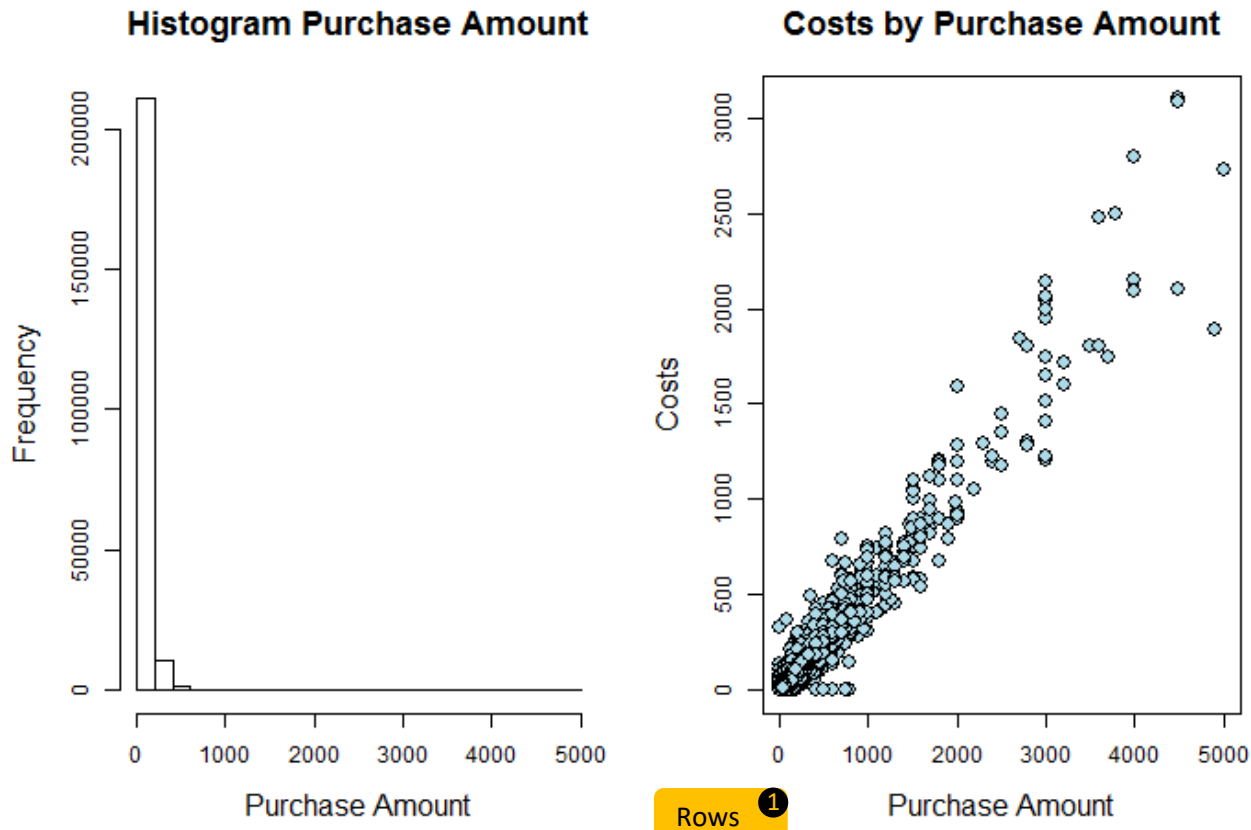
```
hist(myData[, PurchAmount], .....)
```

```
plot(x=myData[, PurchAmount], y=myData[, Cost], .....)
```

```
par(mfrow = c(1, 1))
```

Set back to default ③

You might want to plot multiple graphs in one image



Rows ①

Columns ②

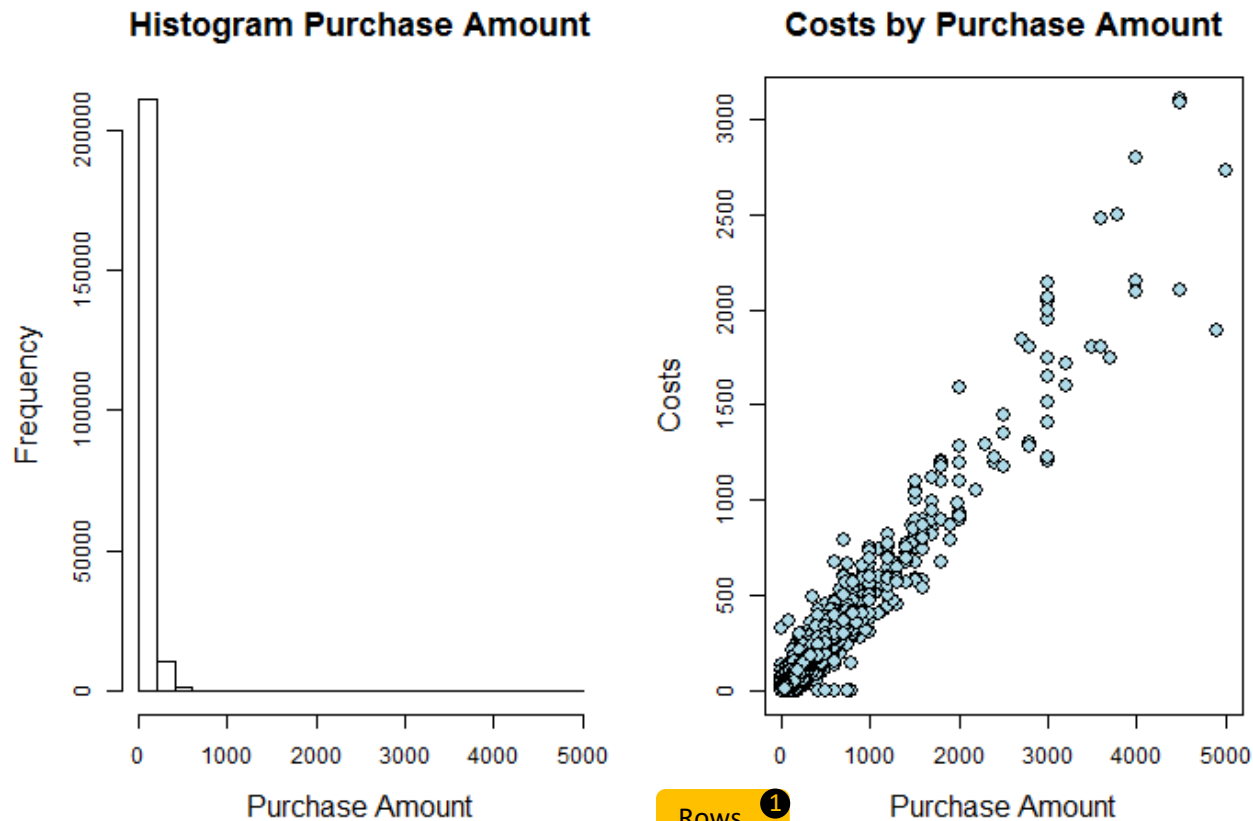
```
par(mfrow = c(1, 2))
hist(myData[, PurchAmount], .....)
```

```
plot(x=myData[, PurchAmount], y=myData[, Cost], .....)
```

```
par(mfrow = c(1, 1))
```

Set back to default ③

You might want to plot multiple graphs in one image



Rows ①

Columns ②

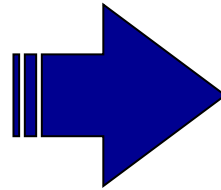
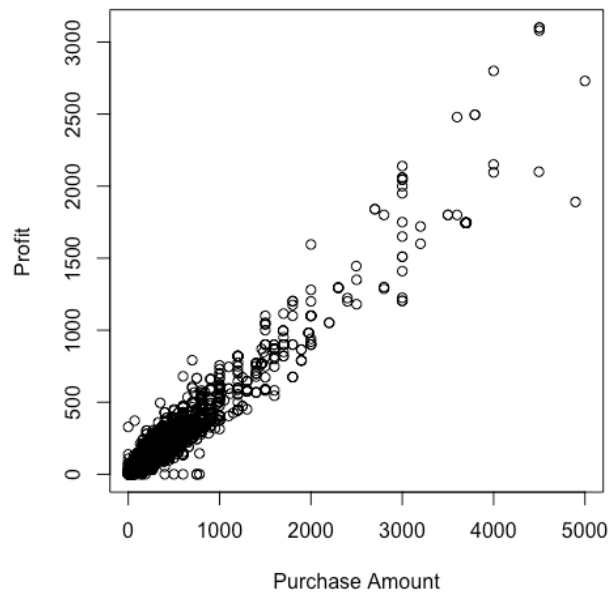
```
par(mfrow = c(1, 2))
hist(myData[, PurchAmount], .....)
```

```
plot(x=myData[, PurchAmount], y=myData[, Cost], .....)
```

```
par(mfrow = c(1, 1))
```

Set back to default ③

Step 6: Save your plot using the command line



Open a graphic device

1

Create the plot

3

Close the graphic device

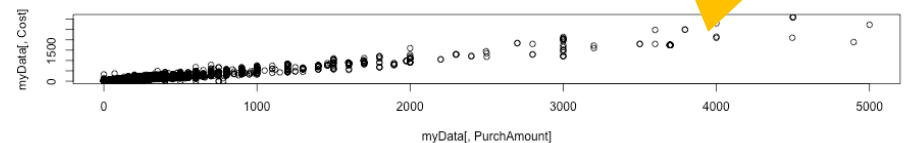
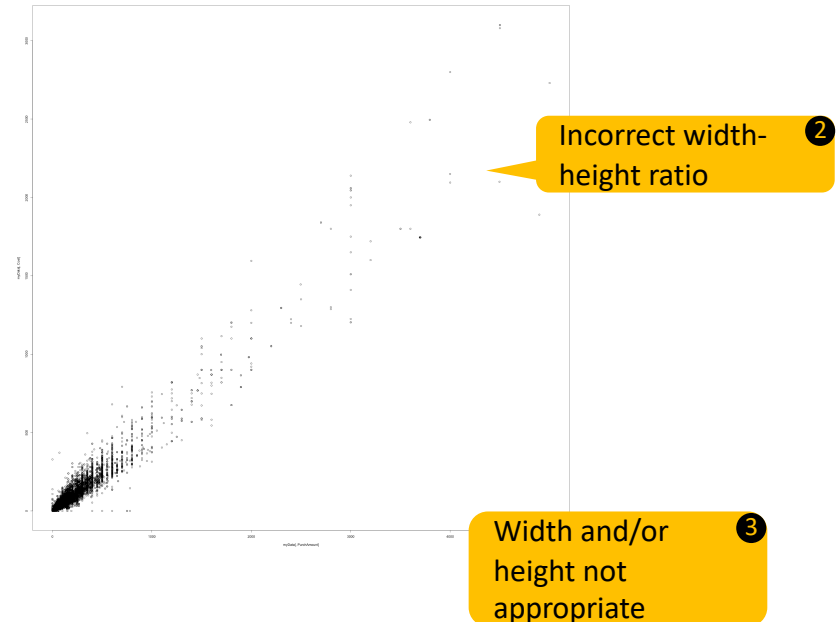
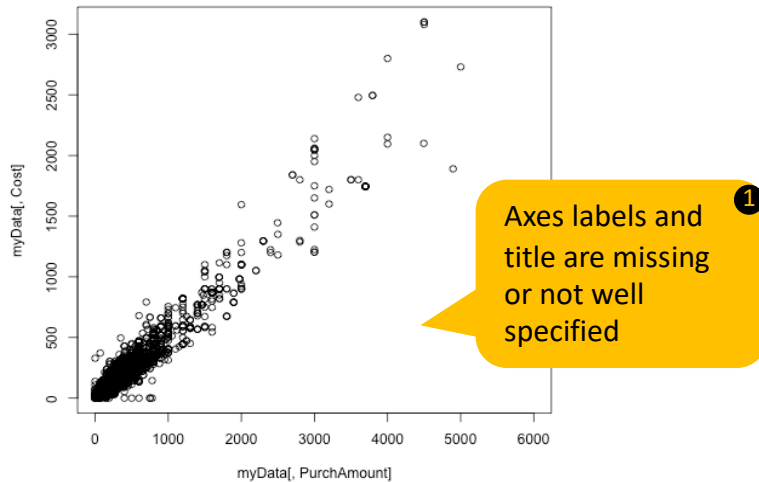
4

```
pdf("output.pdf")  
plot(x)  
dev.off()
```

Use `jpg()` or `png()` to
save as .jpg or .png.

2

Be careful: Saving the plot with the wrong specifications can ruin your hard work!



- **Advice 1:** The point-and-click method helps to avoid this.
- **Advice 2:** Always save and comment your code, so that you can modify simple changes with few effort.

Adding further features to plots