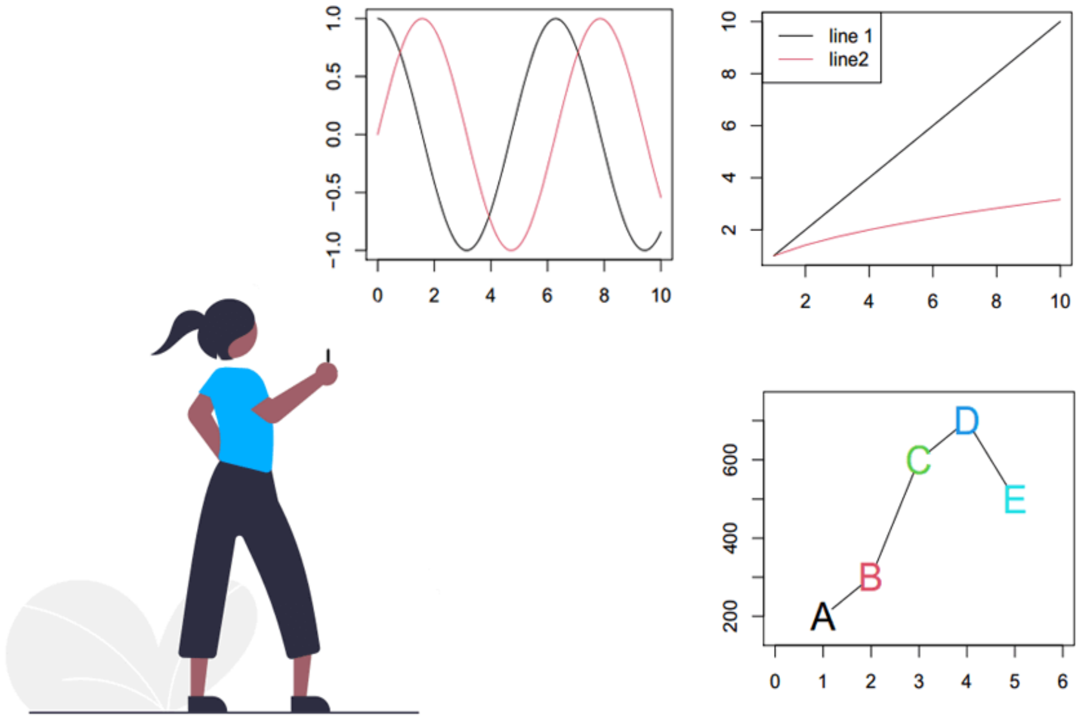
dot-plot, bar-plot, line-charts, box-plot and panels

Pablo E. Gutiérrez-Fonseca, PhD

BIOL-4994 & BIOL-4991 | BIOL-6994 & BIOL-6997
Fall 2025
2025-Aug-01 (updated: 2025-Oct-16)


1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
line 1
line2
10
8
6
4
2
2
4
6
8
10
A
B
C
D
E
600
400
200
0
1
2
3
4
5
6

Plots in Base R

- There are many types of statistical plots, but only five essential ones for beginners:
  - Bar plots – show summary statistics (e.g., counts, means, or proportions).
  - Scatter plots – show relationships among numerical variables.
  - Line graphs – show change over time.
  - Histograms – show data distributions.
  - Boxplots – show between-group and within-group variation.
- These five plots cover a broad range of data-science situations.

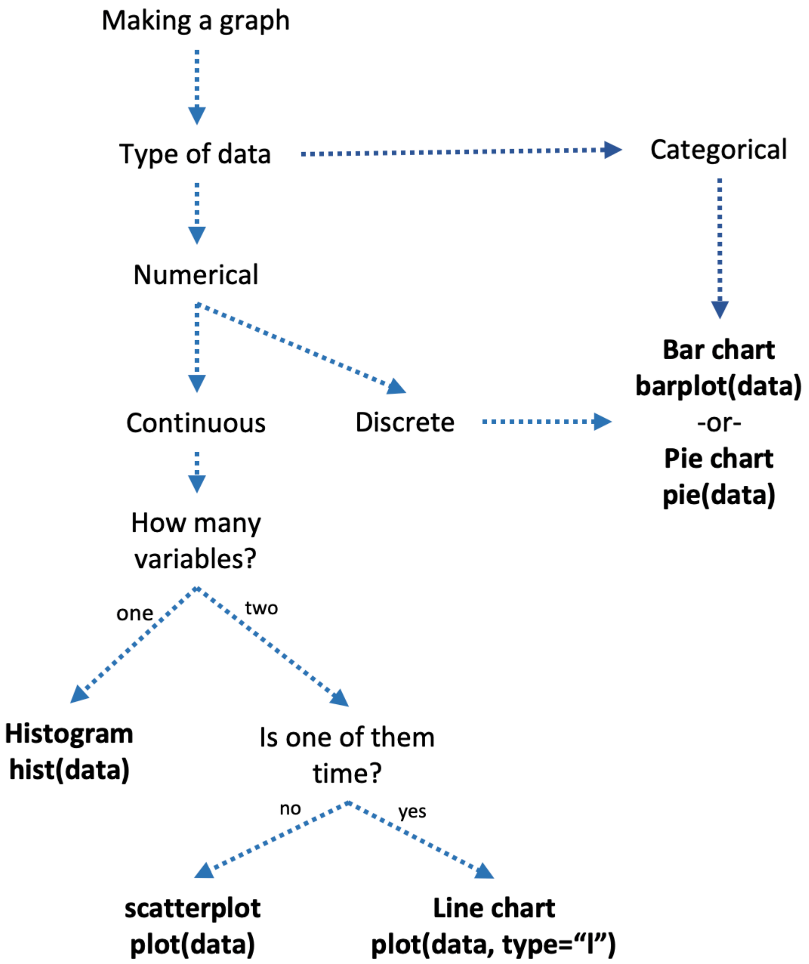
Plots in Base R

- In this lesson, you will learn to:
  - Understand the grammar of graphics.
  - Create the five basic plots listed above.
  - Use faceting and aesthetic variation (e.g., color) to represent multivariate information.
  - Customize plots with advanced control over their visual details.

Plots in Base R

- In this lesson, you will learn to:
  - Understand the grammar of graphics.
  - Create the five basic plots listed above.
  - Use faceting and aesthetic variation (e.g., color) to represent multivariate information.
  - Customize plots with advanced control over their visual details.
- Before we dive into data visualization, we’ll do a brief review (or introduction, for some of you) to statistics, since most plots are used to graph statistical results.

Plots



Making a graph
Type of data
Categorical
Numerical
Continuous
Discrete
Bar chart
barplot(data)
-or-
Pie chart
pie(data)
How many variables?
one
two
Histogram
hist(data)
Is one of them time?
no
yes
scatterplot
plot(data)
Line chart
plot(data, type="l")

Thinking like a statistician

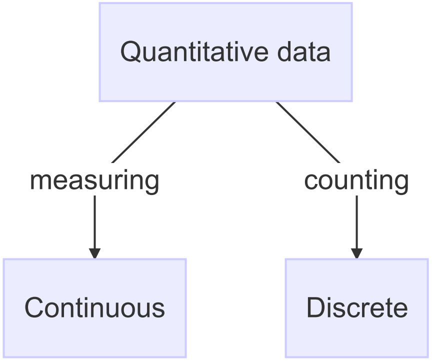
- **Variable:** a quantity counted or measured the characteristic that is being observed.

  1. Quantitative Variables
  2. Qualitative Variables

Thinking like a statistician

- **Quantitative Variables:** a measurable **amount**.

  1. **Continuous variable:** may assume any imaginable value within a certain range. Can (theoretically) have an infinite number of values.
     - Weights, Heights
  2. **Discrete Variables:** countable as integers (whole numbers). No values between two adjacent values are permissible.
     - Number of bicycles sold in a day.



Thinking like a statistician

- **Qualitative Variables:** descriptive characteristic assignable to a category.

  1. **Nominal Variables**: measurements fall into a particular class or category with no order implied.
     - sex (male or female), color (red, blue, green).
  2. **Ordinal Variables:** a ranking scale where order between categories is implied.
     - Likert scale (strongly agree, agree, neutral, disagree, strongly disagree).
  3. **Interval (ratio) Variables:** use a quantitative measurement to assign a specific qualitative category (these are still ordinal).
     - Temperature (cold, warm, hot), age (young, middle-aged, old).

Plots in Base R

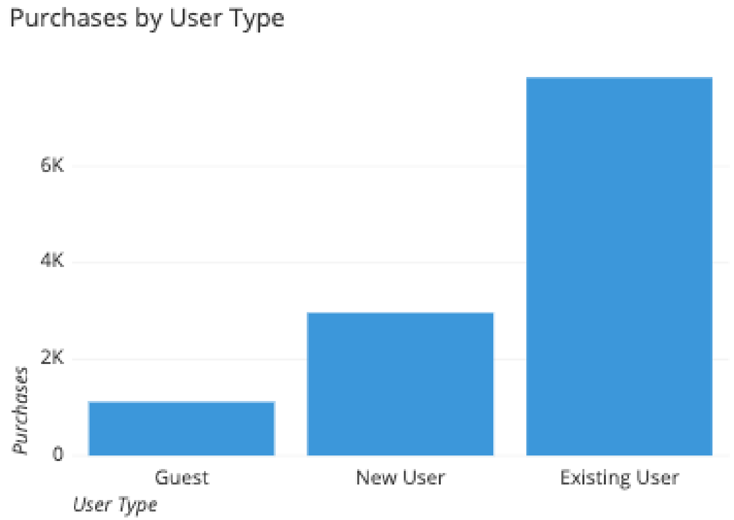
Barplots

Plots in Base R: Barplots

Plots in Base R: Barplots

In a bar chart, values are represented by the length of bars, each corresponding to a measured group.

- Bar charts can be vertical or horizontal:
  - Vertical bar charts are often called column charts.
  - Horizontal bar charts are ideal when:
    - You have many bars to plot, or
    - Labels need extra space to be legible.



Dataset: DNase (built-in in R)

- We will use barplot() function to create a barplot of the mean optical density for each substrate concentration.

```
# Load dataset
data(DNase)
```

- Variables:
- `conc` → substrate concentration (in mg/mL).
- `density` → optical density, representing the experimentally measured enzymatic activity.
- `Run` → experimental run or replicate number (factor with 11 levels).

`barplot()` function

- R Language uses the function barplot() to create bar charts. Here, both vertical and Horizontal bars can be drawn.

- Syntax: `barplot(H, names.arg, xlab, ylab, main, col)`

- Parameters:
  - H: This parameter is a vector or matrix containing numeric values which are used in bar chart.
  - names.arg: This parameter is a vector of names appearing under each bar in bar chart.
  - xlab: This parameter is the label for x axis in bar chart.
  - ylab: This parameter is the label for y axis in bar chart.
  - main: This parameter is the title of the bar chart.
  - col: This parameter is used to give colors to the bars in the graph.

Dataset: DNase (built-in in R)

```r
# Step 1: Inspect data
head(DNase, 3)
```

```
##   Run       conc density
## 1   1 0.04882812   0.017
## 2   1 0.04882812   0.018
## 3   1 0.19531250   0.121
```

```r
# Variables:
# conc = concentration of the substrate
# density = optical density (enzyme activity)
# Run = experimental replicate
```

Dataset: DNase (built-in in R)

- We use `aggregate()` to summarize the 11 experimental runs and visualize the average enzymatic activity at each concentration.

```
# Step 2: Calculate mean density per concentration
mean_density <- aggregate(density ~ conc, data = DNase, FUN = mean)
```

Dataset: DNase (built-in in R)

R Code   Plot

```r
# Step 3: Create barplot
barplot(
  height = mean_density$density, # height is a vector of numeric values that determines the h
  names.arg = mean_density$conc, # names.arg is a vector of names of each bar
  col = "lightblue",
  border = "gray30",
  main = "Mean DNase I Activity by Concentration", # Main title of the plot
  xlab = "Concentration (mg/mL)", # Label for the X-axis
  ylab = "Mean Optical Density", # Label for the Y-axis
  las = 2,          # rotate labels for clarity
  cex.names = 0.8  # adjust label size
) # Don't forget to close the parenthesis
```

Dataset: DNase (built-in in R)

- Simple Horizontal Bar Plot

R Code   Plot

```r
# Step 3: Create barplot
barplot(
  height = mean_density$density,
  names.arg = mean_density$conc,
  col = "lightblue",
  border = "gray30",
  main = "Mean DNase I Activity by Concentration",
  xlab = "Concentration (mg/mL)",
  ylab = "Mean Optical Density",
  las = 2,
  cex.names = 0.8,
  horiz = TRUE      # make bars horizontal
)
```

Scatter plots

Plots in Base R: Scatter plots

Plots in Base R: Scatter plots

- Scatter plots are used to display the **relationship between two continuous variables**.
- Each point on the plot represents one observation, with its position defined by values on the x and y axes.
- They are especially useful for detecting patterns, trends, clusters, or outliers in your data.
- In R, scatter plots can be created easily using the `plot()` function, allowing you to customize colors, symbols, and labels to enhance interpretation.

Dataset: `coronary`

- The dataset contains the total cholesterol level, their individual characteristics and intervention groups in a hypothetical clinical trial. The dataset contains 200 observations for nine variables:

```r
library(readxl)
# Load in R:
coronary <- read_excel("data/coronary.xlsx")
```

Dataset: `coronary`

- Examine our data:

```
head(coronary, 3)
```

```
## # A tibble: 3 × 9
##      id cad      sbp   dbp  chol   age   bmi race   gender
##   <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  <chr>
## 1     1 no cad   106    68  6.57    60  38.9 indian woman
## 2    14 no cad   130    78  6.32    34  37.8 malay  woman
## 3    56 no cad   136    84  5.97    36  40.5 malay  woman
```

Dataset: `coronary`

- The dataset contains the total cholesterol level, their individual characteristics and intervention groups in a hypothetical clinical trial. The dataset contains 200 observations for nine variables:

- Variables

  - id: Subjects’ ID.
  - cad: Coronary artery disease status (categorical) {no cad, cad}.
  - sbp : Systolic blood pressure in mmHg (numerical).
  - dbp : Diastolic blood pressure in mmHg (numerical).
  - chol: Total cholesterol level in mmol/L (numerical).
  - age: Age in years (numerical).
  - bmi: Body mass index (numerical).
  - race: Race of the subjects (categorical) {malay, chinese, indian}.
  - gender: Gender of the subjects (categorical) {woman, man}.

Dataset: `coronary`

R Code   Plot

```r
# Create a line chart
plot(
  coronary$dbp ~ coronary$chol,
  type = "p",                      # line plot
  col = "blue",                    # line color
  lwd = 2,                         # line width
  xlab = "Total Cholesterol (mmol/L)",  # x-axis label
  ylab = "Diastolic Blood Pressure (mmHg)", # y-axis label
  main = "Relationship between Cholesterol and Diastolic BP"
)
```

Spearman Correlation: chol vs dbp

- The Spearman correlation evaluates the monotonic (rank-based) relationship between cholesterol and diastolic blood pressure.

```r
spearman_result <- cor.test(
  coronary$chol,
  coronary$dbp,
  method = "spearman",
  exact = FALSE # avoids warnings with tied ranks
)
```

Spearman Correlation: chol vs dbp

```
# Print results
spearman_result
```

```
##
##     Spearman's rank correlation rho
##
## data:  coronary$chol and coronary$dbp
## S = 841173, p-value = 0.00000007518
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3691043
```

Spearman Correlation: chol vs dbp

- Spearman’s rho (ρ) indicates the strength and direction of the monotonic relationship.
- p-value assesses whether this relationship is statistically significant.
- For example:
  - $\rho > 0$ → higher cholesterol tends to accompany higher diastolic BP.
  - $\rho < 0$ → higher cholesterol tends to accompany lower diastolic BP.

Test for Normality

- Let’s check if chol and dbp are normally distributed (as a comparison to why we might prefer Spearman over Pearson).

```
shapiro.test(coronary$chol)
```

```
##
##      Shapiro-Wilk normality test
##
## data:  coronary$chol
## W = 0.98617, p-value = 0.04786
```

Test for Normality

- Let’s check if chol and dbp are normally distributed (as a comparison to why we might prefer Spearman over Pearson).

```
shapiro.test(coronary$dbp)
```

```
##
##      Shapiro-Wilk normality test
##
## data:  coronary$dbp
## W = 0.97816, p-value = 0.003288
```

Visualization

R Code   Plot

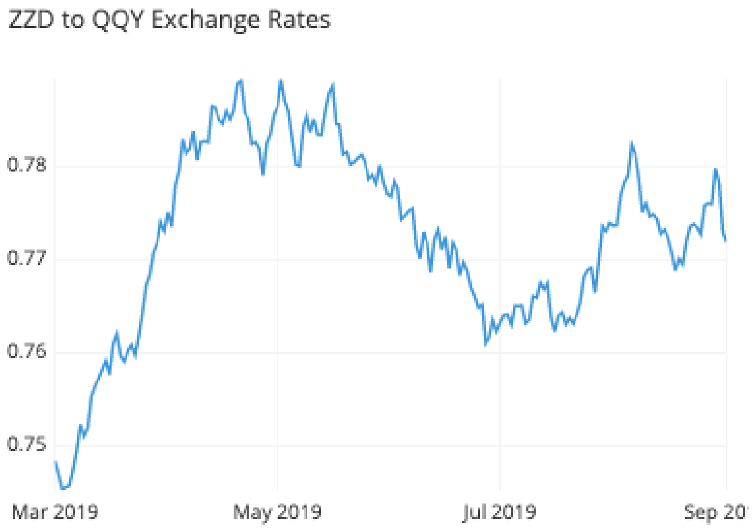
```r
# Create a line chart
plot(
  coronary$dbp ~ coronary$chol,
  type = "p",                       # line plot
  col = "blue",                     # line color
  lwd = 2,                          # line width
  xlab = "Total Cholesterol (mmol/L)",  # x-axis label
  ylab = "Diastolic Blood Pressure (mmHg)", # y-axis label
  main = "Relationship between Cholesterol and Diastolic BP"
)

abline(lm(dbp ~ chol, data = coronary), col = "red", lwd = 2,lty = 2)
```

Line charts

Plots in Base R: Line charts

Plots in Base R: Line charts

- Line charts show changes in value **across continuous measurements**, such as over time.
- Movement of the line up or down highlights positive or negative changes.
- Line charts can expose overall trends and help make predictions or projections.
- Multiple line charts can also give rise to related charts like sparklines or ridgeline plots.



`plot()` function

- To create a line graph in R, we use the `plot()` function.
- Syntax: `plot(v, type, col, xlab, ylab, main)`
- Parameters:
  - v: A vector containing the numeric values to be plotted.
  - type: Specifies the type of graph ("p" only points, "l" only lines, "o" both points and lines).
  - xlab: Label for the x-axis.
  - ylab: Label for the y-axis.
  - main: Title of the chart.
  - col: Specifies the color for the points and lines.

`plot()` function

R Code   Plot

```r
# Asegúrate de que los datos estén ordenados por edad
coronary <- coronary[order(coronary$age), ]
# Gráfico de líneas básico
plot(coronary$age, coronary$chol,
     type = "l", # "l" = line plot
     col = "blue",
     lwd = 2,
     xlab = "Age (years)",
     ylab = "Cholesterol (mmol/L)",
     main = "Cholesterol vs Age")
```

Histograms

Plots in Base R: Histograms

Plots in Base R: Histograms

- Histograms are used to visualize the distribution of a single continuous variable.
- They show how often data values fall within specific intervals (bins).
- Useful for identifying the shape of the data (e.g., normal, skewed, bimodal) and detecting outliers or spread in the dataset.
- In R, histograms can be created using the `hist()` function, which allows customization of the number of bins, colors, and labels.

`hist()` function

R Code

Plot

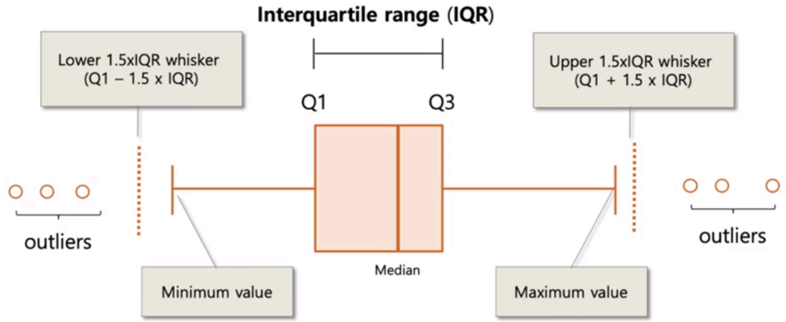
```r
hist(coronary$chol,
main = "Distribution of Cholesterol",
xlab = "Cholesterol (mmol/L)",
col = "lightblue",
border = "white",
# breaks = 10,    # You can cange the number
)
```

Boxplots

Plots in Base R: Boxplots

- Boxplots (or box-and-whisker plots) are used to visualize the distribution of a numeric variable and detect outliers.

- They display:
  - Median (central line inside the box)
  - Interquartile range (IQR) — the box spans from the 25th (Q1) to the 75th percentile (Q3)
  - Whiskers, which extend up to 1.5×IQR from the box
  - Points beyond whiskers represent potential outliers


Interquartile range (IQR)
Lower 1.5xIQR whisker (Q1 – 1.5 x IQR)
Upper 1.5xIQR whisker (Q1 + 1.5 x IQR)
Q1
Q3
outliers
outliers
Median
Minimum value
Maximum value

`boxplot()` function

- In R, boxplots can be created using the `boxplot()` function, which allows customization of colors, labels, and grouping by categorical variables.

R Code

Plot

```r
boxplot(coronary$chol,
        main = "Cholesterol Levels",
        ylab = "Cholesterol (mmol/L)",
        col  = "lightgreen",
        border = "darkgreen"
)
```

`boxplot()` function

R Code

Plot

```r
# Boxplot de SBP por grupo de edad
boxplot(sbp ~ gender,
        data = coronary,
        main = "Systolic Blood Pressure by Age Group",
        xlab = "Age Group (years)",
        ylab = "Systolic BP (mmHg)",
        col = "lightblue",
        border = "darkblue"
)
```

Analisys of Varianza (`anova()`)

- We can use ANOVA to test whether mean systolic blood pressure (SBP) differs across age groups or genders.

```r
anova_model <- aov(sbp ~ gender, data = coronary)
summary(anova_model)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## gender        1     16    15.7    0.04  0.842
## Residuals   198  78100   394.4
```

Assumptions of ANOVA:

- Normality – residuals should be approximately normally distributed. We check this using the Shapiro-Wilk test: `shapiro.test(residuals(anova_model))`.

```
# 1. Shapiro-Wilk test on residuals
shapiro.test(residuals(anova_model))
```

```
##
##      Shapiro-Wilk normality test
##
## data:  residuals(anova_model)
## W = 0.96669, p-value = 0.0001127
```

Assumptions of ANOVA:

- Homogeneity of variance – variance across groups should be similar (can be checked with `bartlett.test()`).

```
bartlett.test(sbp ~ gender, data = coronary)
```

```
##
##      Bartlett test of homogeneity of variances
##
## data:  sbp by gender
## Bartlett's K-squared = 4.6837, df = 1, p-value = 0.03045
```

- Since the p-value < 0.05, we reject the null hypothesis of equal variances.
- This means the variance of SBP differs between genders, violating the homogeneity assumption of ANOVA.