Basic Statistics with R

Dr. Kapil Kumar
Department of Statistics
Chaudhary Charan Singh University, Meerut, India
kapilstats@gmail.com

Online Faculty Development Programme on
Data Analytics and Decision Making with Statistical Software for Entrepreneurs and Innovators (27–31 August 2024)

Department of Mathematics & Statistics with the Directorate of E-Cell, Manipal University Jaipur

27 August 2024 · Online

Plan of the session

1. Getting started with R
2. Describing data
3. Measures of central tendency
4. Dispersion and shape
5. Correlation and regression

By the end you can import data into R, classify variables, compute and interpret averages, spread, skewness, correlation and a simple regression, and pick the summary that answers a business question honestly.

Running example 700 household records of monthly per capita consumption expenditure (MPCE).

PART 1

Getting started with R

Why R for data-driven decisions?

- Free and open source, on Windows, macOS and Linux
- Over 19,000 free add-on packages on CRAN (2024)
- A script records every step, so the analysis can be re-run, audited and shared
- Publication-quality graphics in a few lines
- Limitation: data are held in memory

A short history The **S** language was designed by John Chambers and colleagues at Bell Labs in the 1970s. Ross Ihaka and Robert Gentleman (Auckland) began **R**, a free language "not unlike S", in the early 1990s; version 1.0.0 appeared on 29 February 2000.

Installing R and RStudio

1. Install R from CRAN:
   `cran.r-project.org`
2. Install the free RStudio Desktop from Posit:
   `posit.co/download/rstudio-desktop`
3. Open RStudio; it finds R automatically

Script (top left): write and save code	**Environment** (top right): objects created so far
Console (bottom left): code runs, output appears	**Files, Plots, Help** (bottom right)

> **Tip** Select lines in the script and press `Ctrl+Enter` to run them. Saved scripts are your record of the analysis.

R as a calculator, objects and vectors

```
(2 + 3) * 5; sqrt(16)

## [1] 25
## [1] 4

age <- 1:6                    # ages of children
fr  <- c(5, 3, 7, 5, 4, 2)  # number at each age
age.data <- rep(age, times = fr)
age.data; length(age.data)

##  [1] 1 1 1 1 1 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4 5 5 5 5
## [25] 6 6
## [1] 26
```

- <- stores a result in a named object; R is case sensitive
- `c()` combines values; `1:6` and `seq()` make sequences; `rep()` repeats
- Arithmetic works on whole vectors; # starts a comment

Let R count `rep(age, times = fr)` rebuilds all 26 children from the table, with no typing slips.

Packages and colours used in this session

- Almost everything today is **base R**: `mean()`, `quantile()`, `cor()`, `lm()`, `plot()`
- **`moments`**: `skewness()`, `kurtosis()`
- **`arm`**: `display()`, a compact regression summary
- Two colours, defined once, are reused in every chart

```r
# install once
# install.packages(c("moments", "arm"))
library(moments)
library(arm)
col_main <- "#1F5FA8"   # blue
col_alt  <- "#B45309"   # amber
```

Reading a small CSV file

```
d1 <- read.csv("data/data1.csv")
d1

##    A  B C  D
## 1 10 12 1 14
## 2 25 24 2 36
## 3 36 36 5 69
## 4 14 85 9 58
## 5 10 12 9 NA

mean(d1$D); colMeans(d1, na.rm = TRUE)

## [1] NA
##     A     B     C     D
## 19.00 33.80  5.20 44.25
```

- Keep the script and a `data/` folder together and use relative paths, not drive-letter paths
- `d1$D` extracts one column of the data frame
- The empty cell in column `D` is read as `NA` (missing), so its mean is `NA` until we add `na.rm = TRUE`

The MPCE data set

```
mpce <- read.csv("data/MPCEData.csv",
                 fileEncoding = "UTF-8-BOM")
head(mpce, 3)

##   DISTT  MPCE weight hsize
## 1    43 452.2    668 1.463
## 2    43 972.6   1087 1.018
## 3    43 281.3   3872 1.416

dim(mpce); anyNA(mpce); sum(duplicated(mpce))

## [1] 700   4
## [1] FALSE
## [1] 0
```

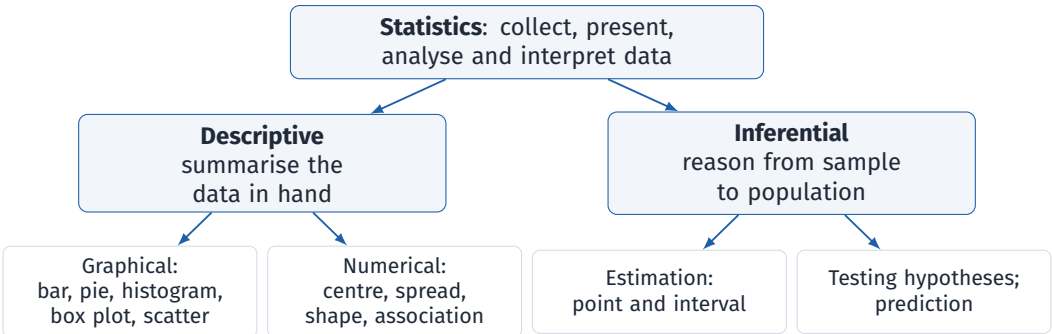
Data note `MPCE`: monthly per capita consumption expenditure (rupees, by the usual convention); `DISTT`: district code (27 districts). `weight` and `hsize` are undocumented (`hsize` is not a whole number). Survey and year are not stated: we treat the file as **illustrative data** and give unweighted summaries.

`fileEncoding` strips an invisible byte-order mark that would garble the first column name.

PART 2

Describing data

What statistics does



Today is mostly the left branch; regression at the end reaches into the right one.

Types of variables

Type	Scale	Business example	Summarise with
Categorical	Nominal	payment mode, region	counts, %, mode; bar chart
Categorical	Ordinal	rating: poor … excellent	counts, median; bar chart
Quantitative	Discrete	items per order	mean or median, SD; bar chart
Quantitative	Continuous	monthly spend	median or mean, IQR or SD; histogram, box plot

Rule A quantitative variable is described by its **centre**, **variability** and **shape**; a categorical one by the **frequency or percentage** of each category. In R, store categories as a `factor()`.

Frequency distributions

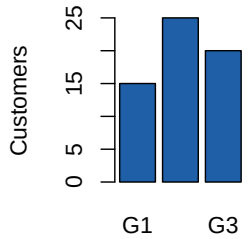
```
tab <- table(age.data)
rbind(freq = tab, cum = cumsum(tab),
      pct  = round(100 * prop.table(tab)))

##       1  2  3  4  5  6
## freq  5  3  7  5  4  2
## cum   5  8 15 20 24 26
## pct  19 12 27 19 15  8

grp <- cut(age.data, breaks = c(0, 2, 4, 6))
table(grp)

## grp
## (0,2] (2,4] (4,6]
##     8    12     6
```

- `table()` counts; `cumsum()` accumulates; `prop.table()` gives proportions
- `cut()` forms classes; `(0,2]` means $0 < x \leq 2$

Reading it Age 3 is the most common (7 children); 20 of the 26 children are aged 4 or younger.

Bar and pie charts for categories

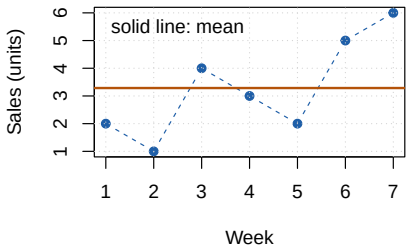
```
groups <- c(G1 = 15, G2 = 25, G3 = 20)
round(prop.table(groups), 3)

##    G1    G2    G3
## 0.250 0.417 0.333

par(mfrow = c(1, 2), mar = c(3, 4, 1, 1))
barplot(groups, col = col_main,
        ylab = "Customers")
pie(groups, col = c(col_main, col_alt,
                    "grey75"))
```



Prefer bars Lengths are compared more accurately than angles; keep pies for a few parts of one whole.

Base graphics: high-level and low-level functions

```r
x2 <- c(1, 2, 3, 4, 5, 6, 7)
y2 <- c(2, 1, 4, 3, 2, 5, 6)
par(mar = c(4, 4, 1, 1))
plot(x2, y2, type = "b", pch = 19, lty = 2,
     col = col_main, xlab = "Week",
     ylab = "Sales (units)")
grid()
abline(h = mean(y2), col = col_alt, lwd = 2)
text(2.5, 5.5, "solid line: mean")
```



- **High-level** functions start a plot: `plot()`, `hist()`, `boxplot()`, `barplot()`
- **Low-level** functions add to it: `abline()`, `text()`, `points()`, `grid()`

PART 3

Measures of central tendency

The arithmetic mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{(population mean: } \mu\text{)}$$

$\sum(x_i - \bar{x}) = 0$, and $\sum(x_i - m)^2$ is smallest at $m = \bar{x}$

- Uses every observation
- Mean $\times$ count gives the total, e.g. revenue

```
x <- c(1, 5, 4, 3, 2)
sum(x) / length(x); mean(x)

## [1] 3
## [1] 3

sum(x - mean(x))   # deviations cancel

## [1] 0
```

Sensitivity One extreme value can pull the mean a long way (next slides).

Trimmed mean and missing values

```r
C <- c(1, 1, 1, 1, 1, 2, 2, 2, 4, 4, 4,
       4, 4, 4, 4, 5, 5, 5, 5, 6, 6)   # n = 21
c(mean   = mean(C),
  trim10 = mean(C, trim = 0.1),  # 2 off each end
  trim2  = mean(C, trim = 2),    # NOT 2 values
  median = median(C))

##   mean trim10  trim2 median
##  3.381  3.353  4.000  4.000

C_na <- c(C, NA)                 # one missing
c(mean(C_na), mean(C_na, na.rm = TRUE))

## [1]    NA 3.381
```

`trim` $= \alpha$ removes the $\lfloor n\alpha \rfloor$ smallest and largest values, then averages the rest.

Careful `trim` is a *proportion* in $[0, 0.5)$, not a count: `trim = 2` silently returns the median. To drop two values from each end of 21, use `trim = 0.1`.

The median

Sort: $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

n odd: median $= x_{((n+1)/2)}$

n even: mean of $x_{(n/2)}$ and $x_{(n/2+1)}$

- The 50th percentile: half the values lie below it
- Hardly moved by a few extreme values, so preferred for incomes, prices and waiting times

```
s1 <- c(10, 8, 14, 15, 7, 3, 3, 8, 12, 10, 9)
sort(s1); median(s1)   # 6th of 11 values

##  [1]  3  3  7  8  8  9 10 10 12 14 15
## [1] 9

s2 <- c(24, 18, 19, 42, 16, 12)
sort(s2); median(s2)   # mean of 3rd and 4th

## [1] 12 16 18 19 24 42
## [1] 18.5
```

The mode: writing a small function

```
mode_value <- function(x) {
  u <- unique(x)
  u[which.max(tabulate(match(x, u)))]
}
A3 <- c(2, 1, 2, 3, 1, 2, 3, 4, 1, 5, 5, 3, 2, 3, 2)
A4 <- c("o", "it", "the", "it", "it")
mode_value(A3); mode_value(A4)

## [1] 2
## [1] "it"
```

- The mode is the most frequent value, the only average for **nominal** data (most used payment method)
- Base R has no statistical mode: `mode()` reports the storage type

Ties For bimodal or multimodal data this function returns only the first mode, so look at `table(A3)` as well.

Mean or median?

```
orders <- c(20, 30, 40, 990)
mean(orders)

## [1] 270

median(orders)

## [1] 35
```

Three of the four values lie between 20 and 40, yet the mean is 270, pulled up by the single value 990. The median, 35, describes the typical order.

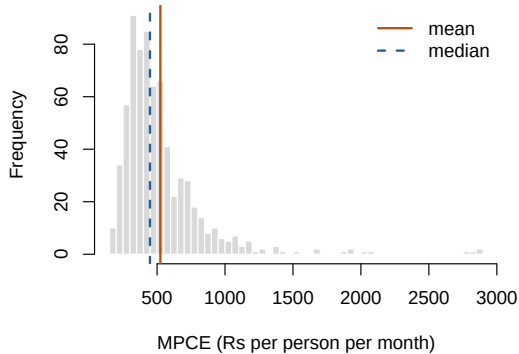
Symmetric: mean $\approx$ median $\approx$ mode
Right skew: mode $<$ median $<$ mean
Left skew: mean $<$ median $<$ mode
(typical orderings, not laws)

For decision makers Report the **median** as the typical income, price or order value. Use the **mean** when totals matter: mean $\times$ customers $=$ revenue.

Averages of MPCE

```
summary(mpce$MPCE); mean(mpce$MPCE, trim = 0.1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     156     340     448     524     600    2878
## [1] 474
```

Reading it Mean 524 $>$ trimmed mean 474 $>$ median 448: a few households spend far more than the rest. A typical household spends about Rs 448 per person per month.



Comparing districts

```
med_d <- tapply(mpce$MPCE, mpce$DISTT, median)
n_d   <- table(mpce$DISTT)
sm <- sort(med_d)
round(c(head(sm, 3), tail(sm, 3)))  # low, high

##  44  56  67  62  51  46
## 277 313 319 534 541 670

n_d[c("44", "46")]                  # group sizes

##
## 44 46
##  2 22
```

- `tapply(x, group, FUN)` summarises each group
- Medians range from Rs 277 to Rs 670 across 27 districts

Check group sizes District 44 has the lowest median but only 2 records: that is not evidence. District 46 (22 records) is a more reliable high-spending market.

Saving a summary table

```
by_district <- data.frame(DISTT  = names(med_d),
                          n      = as.vector(n_d),
                  median = round(as.vector(med_d), 1))
write.csv(by_district, "data/mpce_by_district.csv",
          row.names = FALSE)
check <- read.csv("data/mpce_by_district.csv")
head(check, 4)
```

```
##   DISTT  n median
## 1    43 23  373.5
## 2    44  2  277.3
## 3    45 41  468.6
## 4    46 22  669.7
```

- `write.csv()` saves a data frame that opens in Excel
- `row.names = FALSE` avoids an extra unnamed column
- Read the file back once to confirm what was written
- Also: `read.table()` for text files, `readxl::read_excel()` for Excel

PART 4

Dispersion and shape

Same average, different spread

```
A <- c(30, 50, 70)
B <- c(40, 50, 60)
c(mean(A), mean(B))

## [1] 50 50

c(diff(range(A)), diff(range(B)))

## [1] 40 20

c(sd(A), sd(B))

## [1] 20 10
```

- Both means are 50, but the values of A lie further from it, so 50 represents B better
- Range = largest − smallest: simple, but uses only two values

> **Rule** Report a measure of centre *with* a measure of spread: two shops with the same average daily sales can carry very different risks.

Variance and standard deviation

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \qquad s = \sqrt{s^2}$$

- s is in the units of the data (rupees, days)
- The divisor $n-1$ makes s^2 unbiased for the population variance σ^2

```
v <- c(5, 7, 3)
var(v); sd(v)

## [1] 4
## [1] 2

sum((v - mean(v))^2) / length(v)  # divisor n

## [1] 2.667
```

Which divisor? `var()` and `sd()` divide by $n-1$ (variance 4, SD 2); dividing by n gives 2.67. Say which you use.

Quartiles and the inter-quartile range

```
q15 <- c(3, 6, 7, 11, 13, 22, 30, 40, 44, 50,
         52, 61, 68, 80, 94)
quantile(q15, type = 6)   # (n+1)p rule

##   0%  25%  50%  75% 100%
##    3   11   40   61   94

IQR(q15, type = 6)

## [1] 50

quantile(q15)             # R default: type 7

##   0%  25%  50%  75% 100%
##  3.0 12.0 40.0 56.5 94.0
```

Q_k = the $k(n+1)/4$-th ordered value;
IQR $= Q_3 - Q_1$

Reading it The middle half of the data lie between $Q_1 = 11$ and $Q_3 = 61$, so IQR $= 50$. Rules differ between programs: say which one you used.

Percentiles and the five-number summary of MPCE

```
quantile(mpce$MPCE,
         probs = c(0.10, 0.25, 0.50, 0.75, 0.90))

##   10%   25%   50%   75%   90%
## 279.5 339.6 447.8 600.0 837.4

fivenum(mpce$MPCE)   # min, Q1, median, Q3, max

## [1]  155.9  339.6  447.8  600.0 2878.2

IQR(mpce$MPCE)

## [1] 260.4
```

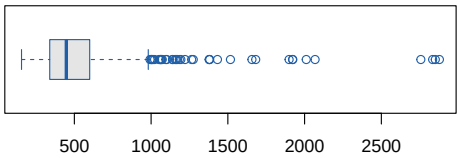
- Percentiles cut ordered data into 100 parts, deciles into 10, quartiles into 4
- Five-number summary: min, Q_1, median, Q_3, max

For a business plan 10% of households spend under Rs 280 per person per month; the top 10% spend over Rs 837. Percentiles define price tiers directly.

Box plot and outliers

```
par(mar = c(4, 1, 1, 1))
boxplot(mpce$MPCE, horizontal = TRUE,
        col = "grey90", border = col_main,
        xlab = "MPCE (Rs per person per month)")
out <- boxplot.stats(mpce$MPCE)$out
length(out); range(out)

## [1] 40
## [1]  993 2878
```

Fences: $Q_1 - 1.5\,\mathrm{IQR}$ and $Q_3 + 1.5\,\mathrm{IQR}$



Reading it 40 of 700 records lie beyond the upper fence, none below: a long right tail of high spenders. An outlier is not an error; check it before removing it.

Coefficient of variation

```r
100 * sd(mpce$MPCE) / mean(mpce$MPCE)

## [1] 59.73

sd_d   <- tapply(mpce$MPCE, mpce$DISTT, sd)
mean_d <- tapply(mpce$MPCE, mpce$DISTT, mean)
cv_d   <- 100 * sd_d / mean_d
round(range(cv_d))

## [1] 26 88
```

$$\text{CV} = 100 \times \frac{s}{\bar{x}} \, \% \quad \text{(unit free)}$$

Reading it Overall CV is 60%. Spending is most uniform in district 50 (26%) and most unequal in district 47 (88%). The CV compares variability on different scales, e.g. the sales of two products.

Shape: skewness and kurtosis

With $m_r = \frac{1}{n}\sum(x_i - \bar{x})^r$:

skewness $g_1 = m_3/m_2^{3/2}$ (0: symmetric)

kurtosis $b_2 = m_4/m_2^2$ (3: normal)

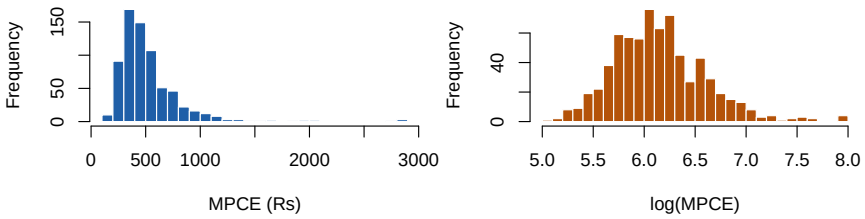
- $g_1 > 0$: longer right tail; $g_1 < 0$: longer left tail
- Excess kurtosis $b_2 - 3$: 0 for the normal, positive for heavier tails

```
skewness(mpce$MPCE); kurtosis(mpce$MPCE)

## [1] 3.492
## [1] 21.81

log_mpce <- log(mpce$MPCE)
skewness(log_mpce); kurtosis(log_mpce)

## [1] 0.7316
## [1] 4.239
```

Note `kurtosis()` returns b_2; "0 for the normal" means *excess* kurtosis.

Seeing the shape: original and log scale

```
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
hist(mpce$MPCE, breaks = 30, col = col_main, border = "white", main = "", xlab = "MPCE (Rs)")
hist(log_mpce, breaks = 30, col = col_alt, border = "white", main = "", xlab = "log(MPCE)")
```



Skewness falls from 3.49 to 0.73 on the log scale: spending varies in proportion rather than in fixed steps, as money variables usually do.

Choosing a summary

Data look like	Report
Symmetric, no outliers	n, mean, SD
Skewed, or outliers	n, median, IQR
Categories	counts, %, mode
Different scales	coefficient of variation

MPCE in one line $n = 700$; median Rs 448 (IQR 340–600); mean Rs 524; strongly right-skewed with 40 high outliers.

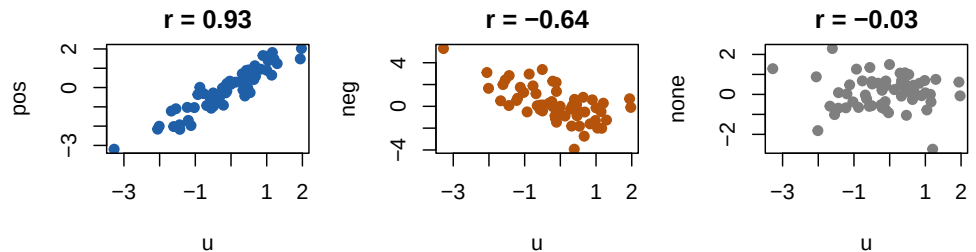
Always Plot first: the histogram or box plot tells you which row applies.

PART 5

Correlation and regression

Scatter diagrams: direction and strength

```r
set.seed(2024)            # each panel below: plot(u, y, pch = 19)
u <- rnorm(60); pos <- u + rnorm(60, sd = 0.4)   # strong positive
neg <- -u + rnorm(60, sd = 1.2); none <- rnorm(60) # negative; none
```



Positive: advertising spend and sales. Negative: price and quantity demanded.

Karl Pearson's correlation coefficient

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}, \quad -1 \le r \le 1$$

- Measures **linear** association only
- Unchanged by a change of origin and scale (the sign flips for a negative scale factor)
- $r = \pm\sqrt{b_{yx}\, b_{xy}}$; sensitive to outliers

```
X <- 1:8
Y <- 3 + 2 * X                  # exact line
c(line   = cor(X, Y),
  scaled = cor(X, (Y - 10) / 5),
  curve  = cor(X, X^2 - 9 * X))

##   line scaled  curve
##      1      1      0
```

The last pair is perfectly related, yet $r = 0$: r misses curved patterns.

A correlation matrix

```
scores <- read.table("data/data15.txt",
                     header = TRUE)
scores; round(cor(scores), 3)

##   batchA batchB batchC
## 1   20.0     40     10
## 2   20.5     42     15
## 3   13.0     56     12
## 4   16.0     38     10
## 5   11.0     57      3
##        batchA batchB batchC
## batchA  1.000 -0.830  0.696
## batchB -0.830  1.000 -0.476
## batchC  0.696 -0.476  1.000
```

- Scores of five candidates in three batches (data from the live session)
- `cor()` on a data frame gives every pairwise r

Reading it A and B move in opposite directions ($r = -0.83$); A and C move together ($r = 0.7$). With five pairs these are descriptions, not yet evidence.

Is the correlation significant?

```
ct <- cor.test(scores$batchA, scores$batchB)
round(c(ct$estimate, ct$statistic,
       df = ct$parameter[[1]], p = ct$p.value), 3)

##    cor      t     df      p
## -0.830 -2.576  3.000  0.082

round(ct$conf.int[1:2], 3)   # 95% CI for rho

## [1] -0.988  0.196
```

`cor.test()` returns a list; `$` extracts its parts.

$$H_0 : \rho = 0, \quad t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \sim t_{n-2}$$

Reading it $p = 0.082 > 0.05$, and the 95% interval (-0.99, 0.2) contains 0. Even $r = -0.83$ is not significant with $n = 5$.

Spearman's rank correlation

$$R = 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)}$$

D_i: difference between the two ranks

With ties, R uses average ranks and Pearson's r of the ranks. Use R for ordinal data (two judges ranking products) and monotone curved trends; it is robust to outliers.

```
rA <- rank(scores$batchA); rB <- rank(scores$batchB)
rA; rB

## [1] 4 5 2 3 1
## [1] 2 3 4 1 5

D <- rA - rB
1 - 6 * sum(D^2) / (5 * (5^2 - 1))

## [1] -0.6

cor(scores$batchA, scores$batchB, method = "spearman")

## [1] -0.6
```

Correlation, causation and r^2

- Correlation does **not** imply causation: ice-cream and air-conditioner sales rise together because both follow temperature
- Nor does a causal link guarantee a large r: a strong *curved* effect can give $r \approx 0$
- Causal claims need design: experiments, A/B tests

Coefficient of determination:

$$r^2 = \frac{\text{explained variation}}{\text{total variation}}$$

Compare on the r^2 scale $r = 0.6$ explains 36% of the variation, $r = 0.3$ only 9%: four times as much, not twice.

Simple linear regression

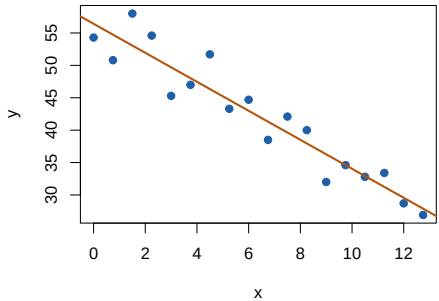
$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
errors ε_i independent $N(0, \sigma^2)$
Least squares: minimise $\sum (y_i - b_0 - b_1 x_i)^2$
$b_1 = r\,\dfrac{s_y}{s_x}, \qquad b_0 = \bar{y} - b_1 \bar{x}$

Residuals $y_i - \hat{y}_i$ sum to zero; the line passes through $(\bar{x}, \bar{y})$.

- Correlation is symmetric; regression has a response y and an explanatory x
- y on x and x on y are two lines; they coincide only when $|r| = 1$
- Gives the size of an effect: y changes by b_1 per unit of x

Fitting the line in R

```r
x <- c(0, 0.75, 1.5, 2.25, 3, 3.75, 4.5,
       5.25, 6, 6.75, 7.5, 8.25, 9, 9.75,
       10.5, 11.25, 12, 12.75)
y <- c(54.3, 50.8, 58, 54.6, 45.3, 47,
       51.7, 43.3, 44.7, 38.5, 42.1, 40,
       32, 34.6, 32.8, 33.4, 28.7, 26.9)
fit <- lm(y ~ x)
par(mar = c(4, 4, 1, 1))
plot(x, y, pch = 19, col = col_main)
abline(fit, col = col_alt, lwd = 2)
```



Illustrative data from the live session: the points fall close to a straight line with a negative slope, $r = -0.95$.

Reading the regression output

```
display(fit, detail = TRUE)

## lm(formula = y ~ x)
##             coef.est coef.se t value Pr(>|t|)
## (Intercept)  56.42     1.39   40.53    0.00
## x            -2.24     0.19  -12.01    0.00
## ---
## n = 18, k = 2
## residual sd = 3.08, R-Squared = 0.90

confint(fit)

##               2.5 % 97.5 %
## (Intercept) 53.466 59.369
## x           -2.633 -1.843
```

Interpretation $\hat{y} = 56.42 - 2.24\,x$: each unit increase in x lowers y by 2.24 on average (95% CI -2.63 to -1.84). $R^2 = 0.9$: x accounts for 90% of the variation in y; slope $p < 0.001$.

`summary(fit)` gives the same table with more detail.

ANOVA table and the two regression lines

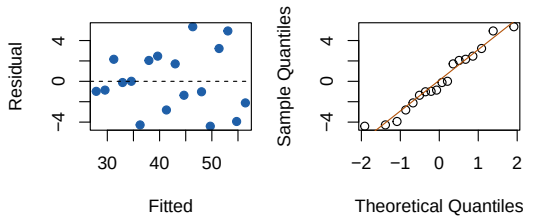
```
print(anova(fit), signif.stars = FALSE)

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## x          1   1365    1365     144  2e-09
## Residuals 16    151       9

b_yx <- coef(lm(y ~ x))[[2]]
b_xy <- coef(lm(x ~ y))[[2]]
c(b_yx, b_xy, b_yx * b_xy, cor(x, y)^2)

## [1] -2.2380 -0.4022  0.9001  0.9001
```

- ANOVA splits total variation into explained and residual parts
- Last line: b_{yx}, b_{xy}, their product, and r^2
- One predictor: $F = t^2$, here $144.2 = (-12.01)^2$

Two coefficients $b_{yx}\, b_{xy} = r^2$; both share the sign of r. They are unaffected by a change of origin but not of scale.

Checking the assumptions

```r
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
plot(fitted(fit), resid(fit), pch = 19,
     col = col_main, xlab = "Fitted",
     ylab = "Residual")
abline(h = 0, lty = 2)
qqnorm(resid(fit), main = "")
qqline(resid(fit), col = col_alt)
sw <- shapiro.test(resid(fit))
```



Reading it No curve or funnel in the residuals; the Q–Q points follow the line. Shapiro–Wilk $p = 0.54$: no evidence against normal errors (but 18 points give such checks little power).

Prediction with intervals

```
new <- data.frame(x = c(5, 10))
predict(fit, new, interval = "confidence")

##     fit   lwr   upr
## 1 45.23 43.60 46.86
## 2 34.04 31.94 36.14

predict(fit, new, interval = "prediction")

##     fit   lwr   upr
## 1 45.23 38.50 51.95
## 2 34.04 27.18 40.89
```

Two intervals **Confidence**: for the *average* y at that x. **Prediction**: for one *new* observation, so always wider.

Do not extrapolate The data cover $0 \leq x \leq 12.75$; beyond that the line is an assumption.

Key take-aways

1. **Know your variable types** before choosing a chart or a summary.
2. **Plot first**: the histogram or box plot shows whether the mean or the median is the honest "typical" value.
3. **Centre needs spread**: report the SD or IQR; use the CV to compare scales.
4. **Check the details**: `na.rm`, the `trim` proportion, the variance divisor and the quantile rule all change the numbers.
5. **Correlation is not causation**, and with small n a large r may be noise.
6. **Regression** gives effect sizes and predictions with intervals, after checking the residuals and within the range of the data.

References

- Dalgaard, P. (2008). *Introductory Statistics with R*, 2nd Ed., Springer, New York.
- Verzani, J. (2014). *Using R for Introductory Statistics*, 2nd Ed., CRC Press, Boca Raton.
- Jones, O., Harden, S., and Crawley, M. J. (2022). *The R Book*, 3rd Ed., Wiley, Hoboken.
- Gelman, A., and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge University Press, Cambridge.
- Montgomery, D. C., Peck, E. A., and Vining, G. G. (2021). *Introduction to Linear Regression Analysis*, 6th Ed., Wiley, Hoboken.
- R Core Team. *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna. `https://www.R-project.org`

Thank you

Questions and discussion