

Practical Utility of Correlation and Regression through R

Dr. Kapil Kumar

Associate Professor, Department of Statistics
Chaudhary Charan Singh University, Meerut, India
kapilstats@gmail.com

One Week Online National Workshop on Statistical Data Analysis using SPSS & R
(SDASR), 6th Edition (17–21 March 2025)

Department of Statistics, Central University of Haryana, Mahendergarh

18 March 2025 · Online

Plan of the session

1. Correlation: ideas and measures
2. Computing and visualising correlation
3. Simulation for correlation analysis
4. Regression and its diagnostics
5. Case studies: house prices, real and simulated
6. Workflow, exercises and take-aways

By the end you can compute Pearson, Spearman and Kendall coefficients by hand and in R, test them, use simulation to see how much r varies by chance, fit and diagnose a regression, fix common problems, and read a multiple regression in plain words.

Packages used in this session

- **ggplot2**, **ggpubr**: scatterplots, Q-Q plots
- **corrplot**: correlation heat maps
- **Hmisc**: `rcorr()`, p -values for a correlation matrix
- **MASS**: `mvrnorm()` and the Boston data
- **lmtest**: `dwtest()`, `bptest()`, `coeftest()`
- **car**: `vif()`, `hccm()`; **nortest**: `lillie.test()`

```
# install once
# install.packages(c("ggplot2", "ggpubr",
#   "corrplot", "Hmisc", "MASS", "lmtest",
#   "car", "nortest"))
library(ggplot2); library(ggpubr)
library(corrplot); library(Hmisc)
library(MASS); library(lmtest)
library(car); library(nortest)

col_main <- "#1F5FA8" # plot colours
col_alt <- "#B45309"
```

PART 1

Correlation: ideas and measures

Correlation and regression answer different questions

Types of correlation (everyday illustrations)

- **Positive:** a company's share price and the market index tend to rise together
- **Negative:** unemployment usually falls when economic growth is strong
- **Zero:** rainfall in Mumbai and the stock index
- **Non-linear:** a population that grows fast, then levels off

None of these, on its own, shows that one variable causes the other.

Correlation strength and direction of association; symmetric in X and Y ; unit-free, between -1 and 1 .

Regression how the mean of Y changes with X ; asymmetric; slope in units of Y per unit of X ; used for prediction.

Three measures of association

Pearson $r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$

Spearman $r_s = \text{Pearson's } r \text{ of the ranks;}$
without ties $r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$

Kendall $\tau_a = \frac{C - D}{n(n - 1)/2}$, C, D :
concordant and discordant pairs

Measure	Use when
Pearson	linear relation, interval data, no gross outliers
Spearman	monotone relation, ordinal data, outliers
Kendall	as Spearman; small samples, many ties

R's `cor(method = "kendall")` returns τ_b , which corrects τ_a for ties.

By hand: Pearson's r and Spearman's r_s

```
x <- c(65, 66, 67, 67, 68, 69, 71, 73) # father (in)
y <- c(67, 68, 64, 69, 72, 70, 69, 73) # daughter (in)
n <- length(x)
sxy <- sum((x - mean(x)) * (y - mean(y)))
r <- sxy / sqrt(sum((x - mean(x))^2) * sum((y - mean(y))^2))
d <- rank(x) - rank(y) # ties get average ranks
rs <- 1 - 6 * sum(d^2) / (n * (n^2 - 1))
c(by_hand = r, R = cor(x, y))
c(by_hand = rs, R = cor(x, y, method = "spearman"))

## by_hand      R
## 0.6458 0.6458
## by_hand      R
## 0.7679 0.7651
```

Reading it Heights (inches) of 8 fathers and daughters, illustrative; the workshop code said cm. The formula reproduces `cor()` exactly.

Ties Shortcut 0.768, R 0.765: with ties only the rank correlation is exact.

By hand: Kendall's τ

```
# +1 where pair (i, j) moves the same way in x and y
s <- sign(outer(x, x, "-")) * sign(outer(y, y, "-"))
C <- sum(s[upper.tri(s)] > 0)      # concordant pairs
D <- sum(s[upper.tri(s)] < 0)      # discordant pairs
c(C = C, D = D, pairs = choose(n, 2))
(C - D) / choose(n, 2)            # tau-a
cor(x, y, method = "kendall")     # tau-b in R

##      C      D pairs
##    21     5    28
## [1] 0.5714
## [1] 0.5926
```

- `outer(x, x, "-")` holds every difference $x_i - x_j$
- Of 28 pairs, 21 are concordant, 5 discordant, 2 tied
- τ_b adjusts the denominator for ties, so it is a little larger

Scale $\tau = P(\text{concordant}) - P(\text{discordant})$; it is usually smaller than r_s .

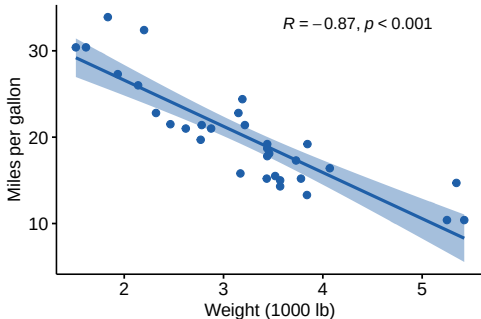
PART 2

Computing and visualising correlation

Weight and fuel economy of 32 cars

```
ggscatter(mtcars, x = "wt", y = "mpg",  
  add = "reg.line", conf.int = TRUE,  
  cor.coef = TRUE, color = col_main,  
  cor.coeff.args = list(p.accuracy = 0.001,  
    label.x = 3.6, label.y = 33),  
  xlab = "Weight (1000 lb)",  
  ylab = "Miles per gallon")
```

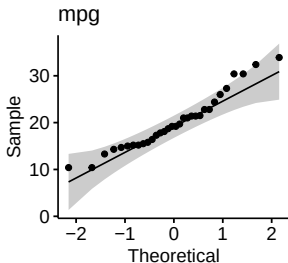
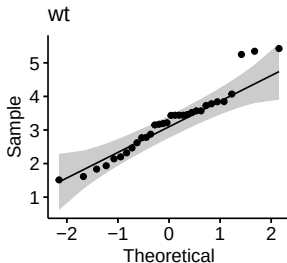
- mtcars: 32 cars road-tested by *Motor Trend* magazine (USA, 1974), built into R
- wt: weight in 1000 lb; mpg: miles per US gallon



Heavier cars travel fewer miles per gallon. The band is the 95% confidence band for the mean line.

Before Pearson's test: are the variables normal?

```
ggarrange(ggqqplot(mtcars$mpg, title = "mpg"),  
          ggqqplot(mtcars$wt, title = "wt"))
```



```
shapiro.test(mtcars$mpg)$p.value  
shapiro.test(mtcars$wt)$p.value
```

```
## [1] 0.1229  
## [1] 0.09265
```

Reading it Both $p > 0.05$, points inside the bands: no evidence against normality. (Strictly, the test needs *bivariate* normality.)

Three coefficients, three tests

```
w <- mtcars$wt; m <- mtcars$mpg
pe <- cor.test(w, m) # Pearson
sp <- cor.test(w, m, method = "spearman", exact = FALSE)
ke <- cor.test(w, m, method = "kendall", exact = FALSE)
est <- c(pe$estimate, sp$estimate, ke$estimate)
pv <- c(pe$p.value, sp$p.value, ke$p.value)
cbind(estimate = round(est, 3), p.value = signif(pv, 3))

##      estimate  p.value
## cor    -0.868 1.29e-10
## rho    -0.886 1.49e-11
## tau    -0.728 6.71e-09
```

Test:

$$t = r\sqrt{n-2}/\sqrt{1-r^2} \sim t_{n-2}$$

Fisher's z interval for ρ :

$$\tanh\left(\operatorname{atanh}(r) \pm 1.96/\sqrt{n-3}\right)$$

```
tanh(atanh(pe$estimate) +
      c(-1, 1) * 1.96 / sqrt(32 - 3))

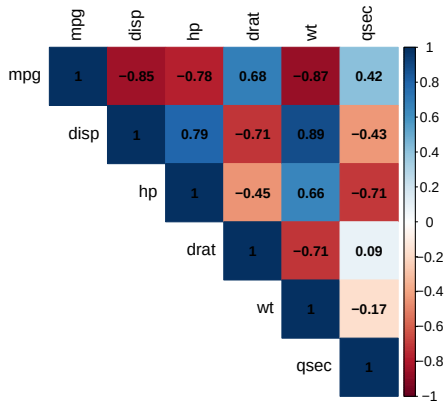
## [1] -0.9338 -0.7441
```

Same CI as `cor.test()`.

Many variables: a correlation heat map

```
cars6 <- mtcars[, c("mpg", "disp", "hp",  
                    "drat", "wt", "qsec")]  
corrplot(cor(cars6), method = "color",  
          type = "upper", addCoef.col = "black",  
          tl.col = "black", number.cex = 0.8)
```

- disp: engine size; hp: horsepower; drat: rear axle ratio; qsec: quarter-mile time
- mpg, disp, hp and wt form one block: big, heavy, powerful cars use more fuel



Fifteen tests at once: adjust the p -values

```
P <- rcorr(as.matrix(cars6))$P # Hmisc
p_raw <- P[upper.tri(P)] # 15 pairs
p_holm <- p.adjust(p_raw, method = "holm")
c(raw = sum(p_raw < 0.05),
  holm = sum(p_holm < 0.05))

## raw holm
## 13 11
```

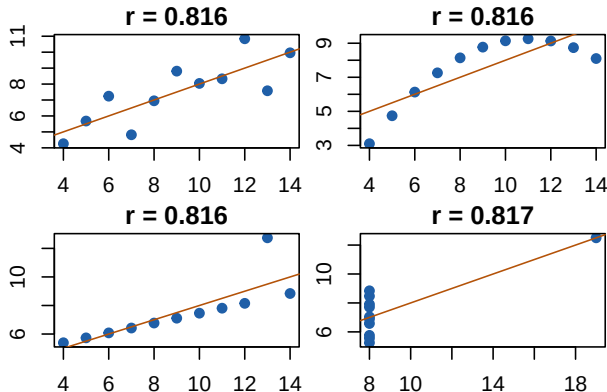
Reading it Unadjusted, 13 of the 15 correlations are significant at 5%; after Holm's adjustment, 11. The weaker links mpg-qsec and disp-qsec drop out.

Why adjust With 15 tests, a false positive somewhere is far more likely than 5%.

Always plot: Anscombe's quartet

```
par(mfrow = c(2, 2), cex = 0.8,  
    mar = c(2, 2, 1.5, 0.5))  
for (i in 1:4) { # x1..x4, y1..y4  
  xi <- anscombe[[i]]  
  yi <- anscombe[[i + 4]]  
  r <- round(cor(xi, yi), 3)  
  plot(xi, yi, pch = 19, col = col_main,  
        xlab = "", ylab = "",  
        main = paste("r =", r))  
  abline(lm(yi ~ xi), col = col_alt)  
}
```

Same r ; only set 1 suits a line.



Four ways a correlation misleads

1. **Curvature:** r measures linear association only
2. **Outliers:** one point can make or break r
3. **Unpaired data:** the workshop's batch A and batch B scores belong to *different* students, so pairing rows is arbitrary (right)
4. **Confounding:** a third variable can drive both, e.g. hospitals and deaths both grow with population

```
d3 <- read.table("data/data3.txt",  
                 header = TRUE)  
cor(d3$batchA, d3$batchB)  
  
## [1] -0.8299  
  
set.seed(1) # random re-pairings  
round(replicate(6, cor(d3$batchA,  
                      sample(d3$batchB))), 2)  
  
## [1] -0.46  0.93  0.83  0.57  0.05 -0.23
```

For unpaired scores r changes with the arbitrary row order: it means nothing.

PART 3

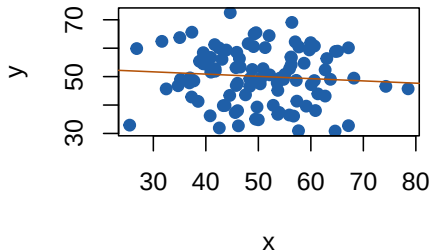
Simulation for correlation analysis

Unrelated variables: the workshop simulation

```
set.seed(2025)
x <- rnorm(100, mean = 50, sd = 10)
y <- rnorm(100, mean = 50, sd = 10) # independent of x
round(c(pearson = cor(x, y),
        spearman = cor(x, y, method = "spearman"),
        kendall = cor(x, y, method = "kendall")), 3)

## pearson spearman kendall
## -0.084 -0.079 -0.048

par(mar = c(4, 4, 1, 1))
plot(x, y, pch = 19, col = col_main)
abline(lm(y ~ x), col = col_alt)
```

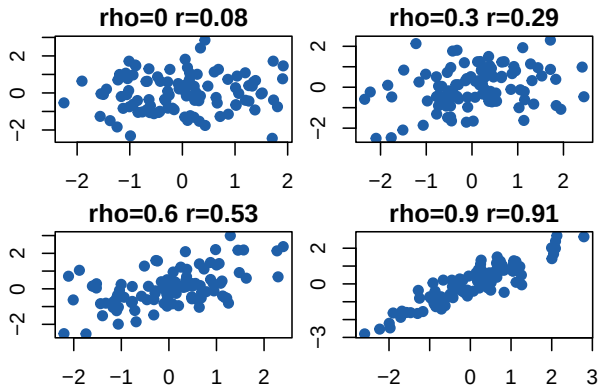


Reading it No pattern; r near, not at, 0.

Simulating a chosen correlation

```
set.seed(2025)
par(mfrow = c(2, 2), cex = 0.85,
    mar = c(2, 2, 1.5, 0.5))
for (rho in c(0, 0.3, 0.6, 0.9)) {
  S <- matrix(c(1, rho, rho, 1), 2)
  xy <- mvrnorm(100, c(0, 0), Sigma = S)
  r <- round(cor(xy)[1, 2], 2)
  plot(xy, pch = 19, col = col_main,
       xlab = "", ylab = "",
       main = paste0("rho=", rho, " r=", r))
}
```

`MASS::mvrnorm()` sets ρ .

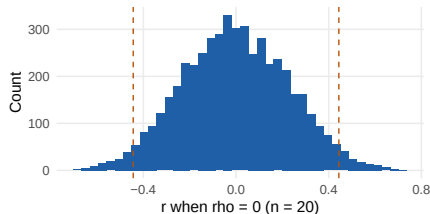


How large can r be by chance alone?

```
set.seed(2025)
r0 <- replicate(5000, cor(rnorm(20), rnorm(20)))
t_c <- qt(0.975, df = 18)
r_crit <- t_c / sqrt(18 + t_c^2) # n = 20
c(r_crit, share = mean(abs(r0) > r_crit))

##          share
## 0.4438 0.0502
```

```
ggplot(data.frame(r0), aes(r0)) +
  geom_histogram(bins = 40, fill = col_main) +
  geom_vline(xintercept = c(-1, 1) * r_crit,
             colour = col_alt, linetype = 2) +
  labs(x = "r when rho = 0 (n = 20)", y = "Count")
```



5000 samples of 20 unrelated pairs.
Values beyond ± 0.44 occur in 5% of
samples: the 5% Type I error rate
the test promises.

Significant is not the same as strong

```
n_vals <- c(10, 20, 50, 100, 500)
t_c <- qt(0.975, df = n_vals - 2)
r_crit <- t_c / sqrt(n_vals - 2 + t_c^2)
data.frame(n = n_vals,
           r_crit = round(r_crit, 3),
           r2_pct = round(100 * r_crit^2, 1))
```

##	n	r_crit	r2_pct
## 1	10	0.632	39.9
## 2	20	0.444	19.7
## 3	50	0.279	7.8
## 4	100	0.197	3.9
## 5	500	0.088	0.8

- r_{crit} : the smallest $|r|$ that is significant at 5%
- With $n = 10$ we need $|r| > 0.63$; with $n = 500$ an r of 0.09 already “works”, yet it explains under 1% of the variation (r^2)

Report the size of r with its confidence interval, not just “significant”. Judge practical importance from r and r^2 .

PART 4

Regression and its diagnostics

The linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \varepsilon_i \text{ independent} \\ N(0, \sigma^2)$$

$$\text{Matrix form: } \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\text{Least squares: } \hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \\ \widehat{\text{Var}}(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2 (\mathbf{X}^\top \mathbf{X})^{-1}$$

$$R^2 = 1 - \text{SSE}/\text{SST}; \text{ in simple regression} \\ R^2 = r^2$$

Assumptions: LINE Linear mean, Independent errors, Normal errors, Equal variance. Estimates need L; tests and intervals need all four.

Which variable on the left?

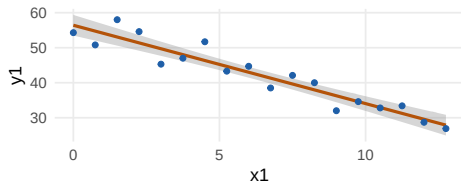
The one you want to explain or predict. Regressing X on Y gives a different line.

A first fit: 18 observations

```
x1 <- 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)
y1 <- 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)
M1 <- lm(y1 ~ x1)
coef(summary(M1))
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	56.418	1.3921	40.53	1.481e-17
## x1	-2.238	0.1864	-12.01	2.040e-09

```
ggplot(data.frame(x1, y1), aes(x1, y1)) +
  geom_smooth(method = "lm",
             colour = col_alt) +
  geom_point(colour = col_main)
```



Illustrative workshop data:

$$\hat{y} = 56.42 - 2.24x, R^2 = 0.9.$$

Under the hood: least squares in matrix form

```
X <- cbind(1, x1) # design matrix
XtX_inv <- solve(t(X) %*% X)
beta <- XtX_inv %*% t(X) %*% y1
s2 <- sum(residuals(M1)^2) / (18 - 2) # sigma^2 hat
se <- sqrt(diag(s2 * XtX_inv))
round(cbind(beta, se), 4)
round(confint(M1), 3)

##              se
## 56.417 1.3921
## x1 -2.238 0.1864
##              2.5 % 97.5 %
## (Intercept) 53.466 59.369
## x1          -2.633 -1.843
```

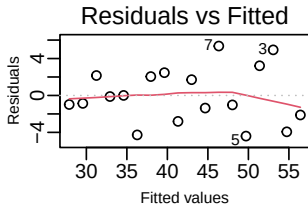
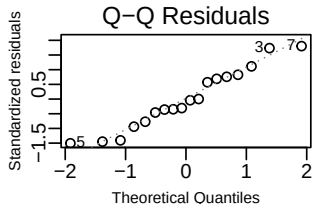
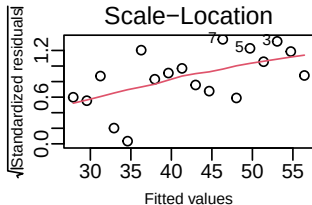
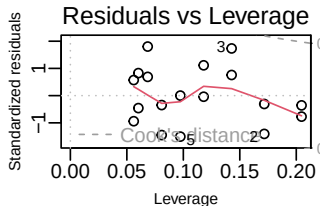
Reading it The matrix formulas reproduce `lm()`'s estimates and standard errors. Each unit increase in x lowers the mean of y by 2.24, 95% CI 1.84 to 2.63.

The same two lines compute a multiple regression: only X gains columns.

Four diagnostic plots in one call

```
par(mfrow = c(2, 2), cex = 0.75,  
    mar = c(2.6, 2.6, 1.4, 0.4),  
    mgp = c(1.5, 0.45, 0),  
    cex.lab = 0.8)  
plot(M1, cex.caption = 0.9)
```

- **L**: no clear curve in residuals vs fitted
- **N**: Q-Q points near the line
- **E**: scale-location rises a little
- No point beyond Cook's distance 0.5



The formal tests side by side

```
e <- residuals(M1)
p <- c(dwtest(M1)$p.value, bptest(M1)$p.value,
      shapiro.test(e)$p.value, lillie.test(e)$p.value)
names(p) <- c("DW", "BP", "SW", "Lilliefors")
round(p, 3)
```

##	DW	BP	SW	Lilliefors
##	0.763	0.035	0.540	0.781

H_0 : no autocorrelation (DW); constant variance (BP); normal errors (Shapiro–Wilk, Lilliefors).

Reading it Independence and normality are fine; Breusch–Pagan ($p = 0.035$) hints at unequal variance.

Corrected from the source `ks.test()` with an estimated mean and SD is invalid: use Lilliefors. Shapiro–Wilk is not limited to $n < 50$.

A remedy: heteroscedasticity-robust standard errors

```
se_ols <- sqrt(diag(vcov(M1)))
se_hc3 <- sqrt(diag(hccm(M1, type = "hc3"))) # car
round(cbind(estimate = coef(M1), se_ols, se_hc3), 4)

##           estimate se_ols se_hc3
## (Intercept)  56.417  1.3921  1.7948
## x1          -2.238  0.1864  0.1956

rob <- coeftest(M1, vcov = hccm(M1, type = "hc3"))
rob["x1", , drop = FALSE]

##      Estimate Std. Error t value Pr(>|t|)
## x1    -2.238      0.1956  -11.44 4.087e-09
```

- `hccm()` estimates the covariance of $\hat{\beta}$ without assuming equal variance
- `coeftest()` redoes the t tests with it

Verdict The robust standard error of the slope is close to the usual one and the slope stays highly significant: the conclusion stands.

Fuel economy: confidence vs prediction intervals

```
M2 <- lm(mpg ~ wt, data = mtcars)
round(cbind(coef(M2), confint(M2)), 3)
new <- data.frame(wt = 3)           # a 3000 lb car
predict(M2, new, interval = "confidence")
predict(M2, new, interval = "prediction")

##                2.5 % 97.5 %
## (Intercept) 37.285 33.450 41.120
## wt          -5.344 -6.486 -4.203
##      fit   lwr   upr
## 1 21.25 20.12 22.38
##      fit   lwr   upr
## 1 21.25 14.93 27.57
```

Reading it Each extra 1000 lb lowers mpg by 5.34 on average. The *mean* mpg of 3000 lb cars is between 20.1 and 22.4; a *single* new car could lie anywhere from 14.9 to 27.6.

The workshop code regressed wt on mpg; fuel use depends on weight, so mpg belongs on the left.

Interpolation vs extrapolation

```
marketing <- data.frame(
  sales = c(2, 4, 6, 9, 12, 34, 45),
  youtube = c(1, 2, 4, 7, 9, 11, 15))
model <- lm(sales ~ youtube, data = marketing)
new <- data.frame(youtube = c(2, 5, 0, 1000))
round(predict(model, new, interval = "prediction"), 1)

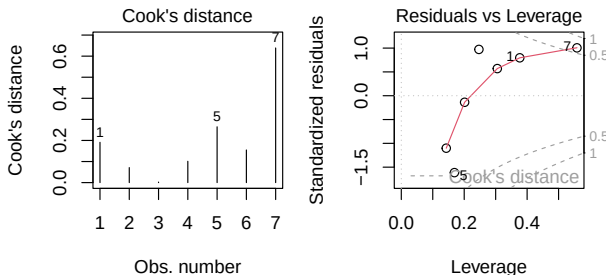
##      fit    lwr    upr
## 1    0.7  -19.4   20.9
## 2    9.9   -9.2   29.0
## 3   -5.4  -26.7   16.0
## 4 3046.6 1634.5 4458.6
```

- Seven periods of advertising spend and sales (illustrative); slope 3.05, $R^2 = 0.86$
- Spend 2 and 5 lie inside the data (1 to 15): usable, if wide, intervals
- Spend 0 predicts *negative* sales; 1000 is far outside the data

Rule A fitted line is a description of the observed range, not a law of nature.

One period drives the slope

```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))  
plot(model, 4)      # Cook's distance  
plot(model, 5)      # residuals vs leverage
```



```
no7 <- marketing[-7, ]    # drop period 7  
coef(model)["youtube"]  
coef(lm(sales ~ youtube, no7))[2]  
  
## youtube  
## 3.052  
## youtube  
## 2.513
```

Reading it Period 7: leverage 0.56, Cook's distance 0.64. Without it the slope drops by about 18%: report both fits.

PART 5

Case studies: house prices

Case study 1: Boston house prices

- MASS::Boston: 506 census tracts of the Boston area, 1970 census (Harrison and Rubinfeld, 1978); a classic real data set
- Response `medv`: median value of owner-occupied homes (\$1000)
- Predictors used: `lstat` (% lower-status population), `rm` (rooms), `ptratio` (pupil-teacher ratio), `dis` (distance to jobs), `nox` (air pollution), `crim` (crime rate), `chas` (on the Charles river)

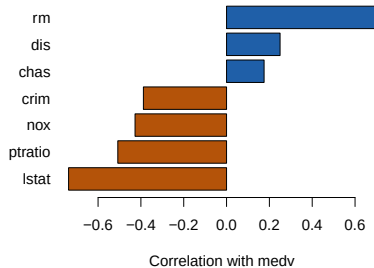
```
anyNA(Boston)
summary(Boston$medv)
sum(Boston$medv == 50) # top-coded

## [1] FALSE
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      5.0   17.0   21.2   22.5   25.0   50.0
## [1] 16
```

Censored response Values above \$50,000 were recorded as 50: `lm()` treats them as exact.

Which variables move with price?

```
vars <- c("lstat", "rm", "ptratio", "dis",  
          "nox", "crim", "chas")  
r_medv <- sort(cor(Boston)[vars, "medv"])  
round(r_medv, 2)  
  
##   lstat ptratio    nox   crim   chas    dis  
##   -0.74  -0.51  -0.43  -0.39   0.18   0.25  
##      rm  
##    0.70  
  
par(mar = c(4, 5, 1, 1))  
barplot(r_medv, horiz = TRUE, las = 1,  
        col = ifelse(r_medv < 0, col_alt, col_main),  
        xlab = "Correlation with medv")
```



Prices fall with the lower-status share ($r = -0.74$) and rise with rooms ($r = 0.7$). Predictors overlap, so we need a joint model.

A multiple regression for house prices

```
fit_h <- lm(medv ~ lstat + rm + ptratio + dis +  
            nox + crim + chas, data = Boston)  
coef(summary(fit_h))
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	35.16877	4.61150	7.626	1.234e-13
## lstat	-0.54523	0.04858	-11.223	3.264e-26
## rm	4.19428	0.40740	10.295	1.150e-22
## ptratio	-0.96348	0.11372	-8.472	2.734e-16
## dis	-1.16305	0.16629	-6.994	8.651e-12
## nox	-17.81329	3.24268	-5.493	6.299e-08
## crim	-0.06545	0.02982	-2.195	2.863e-02
## chas	3.12385	0.88160	3.543	4.322e-04

Reading it, others held fixed One more room: +\$4,194. One point more lower-status share: -\$545. A tract on the river: +\$3,124. Pupil-teacher ratio, pollution, distance and crime all lower prices. $R^2 = 0.718$.

Is a smaller model good enough? Collinearity check

```
fit_all <- lm(medv ~ ., data = Boston)
summary(fit_all)$adj.r.squared # all 13
summary(fit_h)$adj.r.squared  # our 7
round(vif(fit_h), 2)          # car

## [1] 0.7338
## [1] 0.7145
##      lstat      rm ptratio      dis      nox      crim
##      2.52      1.71      1.27      2.56      2.95      1.38
##      chas
##      1.05
```

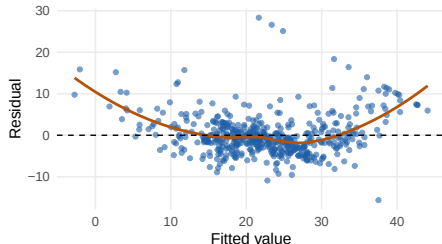
- The workshop model used all 13 predictors (including a race-based index we prefer not to use); six extra variables add only 0.019 to adjusted R^2
- Variance inflation factors below 5: the 7 predictors are not seriously collinear

$VIF_j = 1/(1 - R_j^2)$, R_j^2 from regressing x_j on the other predictors

Checking the house-price model

```
ggplot(data.frame(fitted = fitted(fit_h),  
                  resid = residuals(fit_h)),  
       aes(fitted, resid)) +  
  geom_point(colour = col_main, alpha = 0.6) +  
  geom_smooth(se = FALSE, colour = col_alt) +  
  geom_hline(yintercept = 0, linetype = 2) +  
  labs(x = "Fitted value", y = "Residual")  
bptest(fit_h)$p.value
```

```
##          BP  
## 0.0000005965
```



A U-shaped trend (curvature in `lstat` is missed), uneven spread, and a diagonal streak from the top-coded tracts.

A better model on the log scale

```
fit_log <- lm(log(medv) ~ log(lstat) + rm + ptratio +  
              dis + nox + crim + chas, data = Boston)  
summary(fit_log)$adj.r.squared  
rob <- coeftest(fit_log, vcov = hccm(fit_log))  
rob[c("log(lstat)", "rm", "crim"), ]  
  
## [1] 0.7816  
##           Estimate Std. Error t value Pr(>|t|)  
## log(lstat) -0.39066    0.035104 -11.128 7.650e-26  
## rm         0.07274    0.026192   2.777 5.689e-03  
## crim       -0.01077    0.002062  -5.224 2.576e-07
```

Reading it Adjusted R^2 rises from 0.715 to 0.782. Coefficients are now relative changes: one more room raises the median value by about 7.5%; 10% more lower-status residents lowers it by about 3.7%.

Robust (`hccm()`) standard errors guard against the remaining unequal variance.

Case study 2: simulated “Indian” examples

```
set.seed(2025) # seed added for reproducibility
house <- data.frame(price = rnorm(100, 5e6, 1e6),
                    size = rnorm(100, 1500, 300))
cpi <- data.frame(CPI = rnorm(100, 150, 10),
                  spending = rnorm(100, 50000, 5000))
fit_p <- lm(price ~ size, data = house)
fit_c <- lm(spending ~ CPI, data = cpi)
round(rbind(size = coef(summary(fit_p))["size", ],
            CPI = coef(summary(fit_c))["CPI", ]), 3)

##      Estimate Std. Error t value Pr(>|t|)
## size -296.940    354.74  -0.837   0.405
## CPI   -1.756     50.81  -0.035   0.973
```

Honest label The workshop’s “housing prices in India” and “CPI and spending” data are *simulated*, each pair drawn independently.

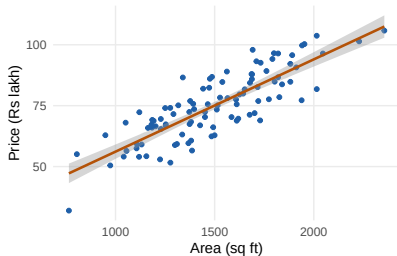
So the result is right
Neither slope is significant ($R^2 \approx 0$).

Simulating a relationship we can recover

```
set.seed(2025)
area <- rnorm(100, mean = 1500, sd = 300) # sq ft
price <- 15 + 0.04 * area + rnorm(100, 0, 8) # lakh
fit_s <- lm(price ~ area)
round(confint(fit_s)["area", ], 4)

## 2.5 % 97.5 %
## 0.0328 0.0429

ggplot(data.frame(area, price), aes(area, price)) +
  geom_point(colour = col_main) +
  geom_smooth(method = "lm", colour = col_alt) +
  labs(x = "Area (sq ft)", y = "Price (Rs lakh)")
```



True slope 0.04 lakh (Rs 4000) per sq ft, chosen for illustration, not an estimate of any market. Estimate: Rs 3,787; the 95% CI contains the truth.

Monte Carlo check: does least squares deliver?

```
set.seed(2025)
slopes <- numeric(1000)
covered <- logical(1000)
for (i in 1:1000) {
  a <- rnorm(100, mean = 1500, sd = 300)
  p <- 15 + 0.04 * a + rnorm(100, 0, 8)
  f <- lm(p ~ a)
  slopes[i] <- coef(f)[2]
  ci <- confint(f)["a", ]
  covered[i] <- ci[1] < 0.04 & ci[2] > 0.04
}
c(mean = mean(slopes), sd = sd(slopes), cover = mean(covered))

##      mean      sd    cover
## 0.040044 0.002759 0.943000
```

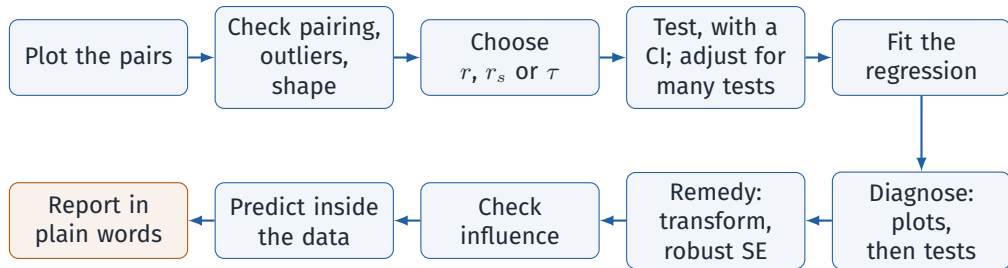
Reading it Over 1000 simulated data sets the slope averages 0.04004 (truth 0.04): unbiased. Its spread, 0.00276, is close to the theoretical $\sigma/(\sigma_x\sqrt{n-1}) = 8/(300\sqrt{99}) = 0.00268$. The 95% intervals cover the truth 94.3% of the time.

The same recipe tests any estimator under any assumption you choose.

PART 6

Workflow and take-aways

The workflow we followed



Simulation sits beside every step: when unsure whether a method works for your design, generate data with a known answer and check.

Exercises

1. **Correlation.** For `airquality` (built into R), compute Pearson, Spearman and Kendall correlations of `Ozone` and `Temp` (use `= "complete.obs"`). Plot first: which coefficient suits the shape?
2. **Multiple regression on real data.** Fit `lm(mpg ~ wt + hp, data = mtcars)`. Interpret each slope with the other held fixed, check `vif()`, and compare adjusted R^2 with the weight-only model.
3. **Diagnostics.** Draw `plot()` for that model and run `dwtest()`, `bptest()` and `shapiro.test()` on it. Which assumption, if any, is in doubt?
4. **Simulation.** Repeat the null-distribution experiment with $n = 50$ and with `method = "spearman"`. Does the 5% rejection rate still hold?

Key take-aways

1. **Plot before you correlate:** Anscombe's four data sets share one r .
2. **Pick the coefficient for the data:** Pearson for linear relations, Spearman or Kendall for ranks, monotone curves and outliers; correlate paired data only.
3. **Significant is not strong:** report r with its interval and adjust for many tests.
4. **Diagnose every regression**, then remedy: logs, robust standard errors, a look at influential points; never extrapolate.
5. **Simulate** to understand a method, and label simulated data as simulated.

References

- Fox, J., and Weisberg, S. (2019). *An R Companion to Applied Regression*, 3rd Ed., Sage, Thousand Oaks.
- Montgomery, D. C., Peck, E. A., and Vining, G. G. (2021). *Introduction to Linear Regression Analysis*, 6th Ed., Wiley, Hoboken.
- Faraway, J. J. (2014). *Linear Models with R*, 2nd Ed., CRC Press, Boca Raton.
- Rizzo, M. L. (2019). *Statistical Computing with R*, 2nd Ed., CRC Press, Boca Raton.
- Breusch, T. S., and Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5), 1287–1294.
- Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27(1), 17–21.
- Harrison, D., and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1), 81–102.

Thank you

Questions and discussion