

Data Analysis using R

Dr. Kapil Kumar

Assistant Professor, Department of Statistics
Central University of Haryana, Mahendergarh, India
kapilstats@gmail.com

Refresher Course in Statistical Computing in Social Sciences (Multidisciplinary)
Department of Statistics, University of Rajasthan, Jaipur

5 February 2022 · Online

Plan of the session

1. Getting started with R and RStudio
2. Data in R: vectors, data frames, files
3. Graphics: base R and `lattice`
4. Testing hypotheses: tables and t -tests
5. ANOVA and correlation
6. Regression and its diagnostics

By the end you can read your own data into R, test a hypothesis and read the output, fit a linear regression and check its assumptions with plots and formal tests.

For University teachers from many disciplines: no programming background assumed.

PART 1

Getting started

R and RStudio in brief

- R: free software for statistical computing and graphics, begun by Ross Ihaka and Robert Gentleman in the 1990s in the spirit of the S language
- On 3 February 2022 CRAN listed **18,861 packages**; 40 *Task Views* group them by field (social sciences, econometrics, psychometrics, ...)
- Install R first, then RStudio Desktop:
<https://cran.r-project.org>
<https://posit.co/download/rstudio-desktop/>

Why teachers use it Free for every student, scripts that make an analysis repeatable, and methods from every discipline in one place.

Packages Installed once with `install.packages()`; loaded in each session with `library()`.

Working in RStudio

Source (script) Write and save code in a .R file; Ctrl+Enter runs the current line or selection.	Environment / History Objects in memory (data sets, vectors, functions) and past commands.
Console Where code runs and output appears; typed commands are not saved.	Files / Plots / Packages / Help Your folder, graphs, installed packages and help pages.

Habits that save time Work in a script, comment it with #, and keep script and data in one folder (an RStudio project) so that paths like `data/file.csv` work.

Packages used in this session

- **lattice**: `trellis` graphics
- **arm**: `display()`, compact regression output
- **MASS**: `studres()`, studentized residuals
- **PASWR**: `checking.plots()`, four diagnostic plots
- **lmtest**: `dwtest()`, `bptest()`, `coeftest()`
- **e1071**: `skewness()`, `kurtosis()`
- **nortest**: `lillie.test()`; **car**: `hccm()`

```
# install once, e.g.  
# install.packages(c("lattice", "arm",  
#   "MASS", "PASWR", "lmtest", "e1071",  
#   "nortest", "car"))  
library(lattice); library(arm)  
library(MASS);    library(PASWR)  
library(lmtest);  library(e1071)  
library(nortest); library(car)  
col_main <- "#1F5FA8" # blue  
col_alt  <- "#B45309" # amber
```

Getting help

```
?mean           # help page
??correlation   # search all packages
help.search("variance")
help(package = "MASS")
example(wireframe) # run examples
?swiss          # about a data set
```

```
str(swiss[, 1:3], vec.len = 1)

## 'data.frame': 47 obs. of 3 variables:
## $ Fertility : num 80.2 83.1 ...
## $ Agriculture: num 17 45.1 ...
## $ Examination: int 15 6 ...
```

- `str()` shows the type and first values of every column
- A free online book: *R for Data Science*, <https://r4ds.hadley.nz>

PART 2

Data in R

Vectors and summary statistics

- `c()` combines values into a vector; `<-` assigns
- Summaries: `mean()`, `median()`, `var()`, `sd()`, `quantile()`, `summary()`
- Arithmetic and functions act on every element
- `x[2]`, `x[-2]`, `x[x > 5]` extract elements
- `seq(3, 5, by = 0.5)` and `rep(c(2, 3), 3)` build sequences

```
x <- c(2, 4, 3, 5, 7, 9, 15)
summary(x)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      2.00   3.50   5.00   6.43   8.00  15.00

c(mean = mean(x), sd = sd(x), n = length(x))

## mean    sd     n
## 6.429 4.467 7.000

round(log(x), 2)      # element by element

## [1] 0.69 1.39 1.10 1.61 1.95 2.20 2.71
```

Reading a CSV file and summarising its columns

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

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

cv <- function(v) 100 * sd(v) / mean(v)
round(apply(data1, 2, cv), 1) # by column

##      A      B      C      D
## 55.1 81.6 84.6 55.1
```

- Save the sheet from Excel as CSV: names in the first row, one row per case; `read.table(..., sep = ",")` also reads it
- Use relative paths, never a full path such as `D:/...`
- `apply(data, 2, f)`: `f` on each column (1: each row)

Your own function `cv` computes $100 s/\bar{x}$: C is the most variable relative to its mean (84.6%).

Lists, factors and data frames

```
data2 <- list(x = c(2, 5, 4),
              Name = c("X", "Y", "Z"),
              Median = 4)
str(data2)

## List of 3
## $ x      : num [1:3] 2 5 4
## $ Name   : chr [1:3] "X" "Y" "Z"
## $ Median: num 4
```

A **list** holds objects of any type, e.g. test results.

```
score <- c(20, 20.5, 13, 16, 11,
           40, 42, 38, 56, 57)
group <- gl(2, 5, labels = c("batchA",
                             "batchB"))
data3 <- data.frame(score, group)
head(data3, 2)

##   score group
## 1  20.0 batchA
## 2  20.5 batchA
```

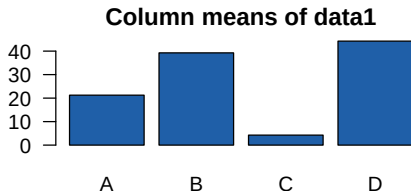
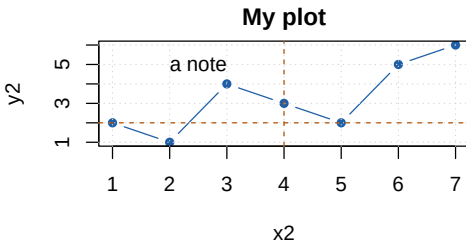
Data frame: a table of columns; **factor:** a categorical variable.

PART 3

Graphics

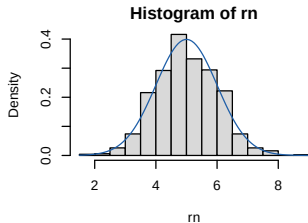
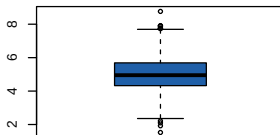
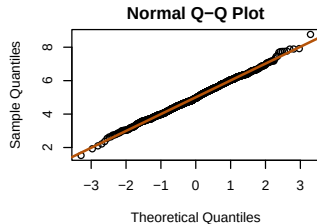
High-level and low-level graphics

```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
barplot(apply(data1, 2, mean), col = col_main, las = 1, main = "Column means of data1")
x2 <- c(1, 2, 3, 4, 5, 6, 7); y2 <- c(2, 1, 4, 3, 2, 5, 6)
plot(x2, y2, type = "b", pch = 16, col = col_main, main = "My plot") # high level
grid(); abline(h = 2, v = 4, col = col_alt, lty = 2); text(2.5, 5, "a note") # low level
```



Looking at a distribution

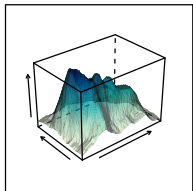
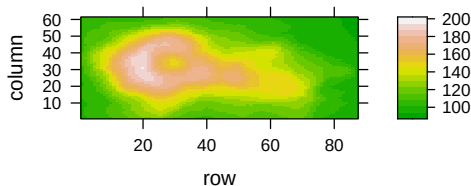
```
set.seed(2022); rn <- rnorm(1000, mean = 5, sd = 1)
par(mfrow = c(1, 3), mar = c(4, 4, 2, 1))
hist(rn, freq = FALSE, col = "grey85"); curve(dnorm(x, 5, 1), add = TRUE, col = col_main)
boxplot(rn, col = col_main); qqnorm(rn); qqline(rn, col = col_alt, lwd = 2)
```



Shape, five-number summary and agreement with the normal distribution.

Trellis graphics with lattice

```
p1 <- wireframe(volcano, shade = TRUE, xlab = "", ylab = "", zlab = "") # 3-D surface
p2 <- levelplot(volcano, col.regions = terrain.colors(100), aspect = "fill") # from above
print(p1, split = c(1, 1, 2, 1), more = TRUE); print(p2, split = c(2, 1, 2, 1))
```



volcano: built-in grid of heights of Maunga Whau, Auckland, seen in 3-D and from above.

PART 4

Testing hypotheses

The logic of a statistical test

1. State H_0 (no effect, no difference) and H_1
2. Choose a test whose assumptions fit the design and data
3. Compute the test statistic from the sample
4. **p -value**: the probability, *if H_0 were true*, of a statistic at least as extreme as the one observed
5. If $p \leq \alpha$ (often 0.05), reject H_0

A p -value is not the probability that H_0 is true, nor the size or importance of an effect.

Two errors Type I: rejecting a true H_0 (probability α). Type II: missing a real effect; small samples make it likely.

Always report the estimate and its confidence interval too.

Categorical data: diabetes, age and gender

```
sugar_data <- read.csv(
  "data/sugar_patients_data.csv")
tab_age <- xtabs(~ age + diabetes, sugar_data)
tab_gender <- xtabs(~ gender + diabetes,
                    sugar_data)
addmargins(tab_age)
```

```
##           diabetes
## age      no yes Sum
##  old      2  6  8
##  young   11  1 12
##  Sum     13  7 20
```

```
round(prop.table(tab_age, 1), 2)
```

```
##           diabetes
## age      no yes
##  old    0.25 0.75
##  young  0.92 0.08
```

```
round(prop.table(tab_gender, 1), 2)
```

```
##           diabetes
## gender      no yes
##  female  0.55 0.45
##  male    0.78 0.22
```

Hypothetical data on 20 patients.

Chi-square test, expected counts and Fisher's exact test

H_0 : the variables are independent

$$X^2 = \sum \frac{(O - E)^2}{E}$$

$E = \text{row total} \times \text{column total} / n$

Small expected counts The χ^2 approximation needs $E \geq 5$. Hiding R's warning with `suppressWarnings()` does not fix it; Fisher's exact test does.

```
round(chisq.test(tab_age)$expected, 1)
```

```
##          diabetes
## age          no yes
##   old      5.2 2.8
##  young    7.8 4.2
```

```
ft_age <- fisher.test(tab_age)
ft_gender <- fisher.test(tab_gender)
round(c(age = ft_age$p.value,
        gender = ft_gender$p.value), 4)
```

```
##   age gender
## 0.0044 0.3742
```

Reading the categorical analysis

- **Age:** Fisher's $p = 0.0044$. Diabetes is associated with age group: 6 of 8 old patients against 1 of 12 young ones
- **Gender:** $p = 0.37$: no evidence of an association
- The χ^2 p -value for age (0.0098) leads to the same decision here, but it rests on an approximation that these counts do not support

In words In these (hypothetical) data, older patients were much more often diabetic; there was no clear difference between men and women.

Remember “No evidence of association” with 20 patients is not evidence of *no* association.

Two independent groups: Welch and pooled t -tests

```
A1 <- c(20, 20.5, 13, 16, 11) # batch A
B1 <- c(40, 42, 38, 56, 57)   # batch B
welch <- t.test(A1, B1)
pooled <- t.test(A1, B1, var.equal = TRUE)
data.frame(test = c("Welch", "pooled"),
  t = c(welch$statistic, pooled$statistic),
  df = c(welch$parameter, pooled$parameter),
  p = c(welch$p.value, pooled$p.value))

##      test      t    df      p
## 1  Welch -6.774 5.605 0.0006718
## 2 pooled -6.774 8.000 0.0001415
```

$$H_0 : \mu_A = \mu_B$$

$$\text{Welch: } t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{s_A^2/n_A + s_B^2/n_B}}$$

$$\text{Pooled: } df = n_A + n_B - 2$$

Reading it B's mean is higher by 30.5 points (Welch 95% CI 19.3 to 41.7). Both tests reject H_0 ; Welch, the default, allows unequal variances.

Paired data: does an anti-smoking film reduce smoking?

```
smoke <- read.csv("data/smoker_data.csv")
smoke$diff <- smoke$Before - smoke$After
smoke
```

##	Before	After	diff
## 1	28	26	2
## 2	29	27	2
## 3	31	32	-1
## 4	44	44	0
## 5	35	35	0
## 6	20	16	4
## 7	50	47	3
## 8	25	23	2

- Mean daily cigarettes of 8 heavy smokers in the month before and the month after seeing the film (illustrative data)
- Same smokers twice: analyse the differences $d_i = \text{Before}_i - \text{After}_i$
- The question asks about a *decline*, so $H_1 : \mu_d > 0$ (one-sided, decided before seeing the data)

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \sim t_{n-1} \text{ under } H_0 : \mu_d = 0$$

The paired t -test, one-sided: $H_1 : \mu_d > 0$

```
t.test(smoke$Before, smoke$After, paired = TRUE, alternative = "greater")

##
## Paired t-test
##
## data:  smoke$Before and smoke$After
## t = 2.5, df = 7, p-value = 0.02
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  0.3678      Inf
## sample estimates:
## mean difference
##              1.5
```

Reading the paired test

```
shapiro.test(smoke$diff)$p.value  
  
## [1] 0.563  
  
two_sided <- t.test(smoke$Before, smoke$After,  
                    paired = TRUE)  
round(c(two_sided$p.value, two_sided$conf.int), 3)  
  
## [1] 0.040 0.087 2.913
```

- Differences look normal (Shapiro–Wilk $p = 0.56$)
- Two-sided, as in many textbooks: $p = 0.04$, 95% CI 0.09 to 2.91

Conclusion Mean decline 1.5 cigarettes a day (one-sided $p = 0.02$): significant at 5%, but small against roughly 33 a day.

Design caution No control group: a before–after fall cannot be attributed to the film alone.

PART 5

ANOVA and correlation

One-way ANOVA: three batches

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \varepsilon_{ij} \sim N(0, \sigma^2)$$

$$H_0 : \tau_1 = \tau_2 = \tau_3 = 0$$

$$F = MS_{\text{between}} / MS_{\text{within}}$$

`aov()` fits the model.

```
x1 <- c(20, 20.5, 13, 16, 11)
x2 <- c(40, 42, 56, 38, 57)
x3 <- c(10, 15, 12, 10, 3)
data4 <- data.frame(score = c(x1, x2, x3),
  group = gl(3, 5, labels = c("A", "B", "C")))
M2 <- aov(score ~ group, data = data4)
print(summary(M2), signif.stars = FALSE)
```

##	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## group	2	3845	1923	47.7	0.0000019
## Residuals	12	483	40		

Group means, Tukey's HSD and a box plot

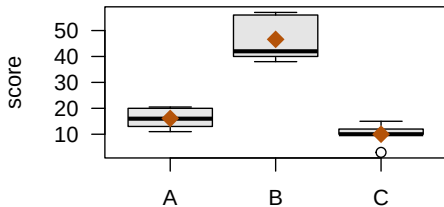
```
means <- tapply(data4$score, data4$group, mean)
means
```

```
##      A      B      C
## 16.1 46.6 10.0
```

```
round(TukeyHSD(M2)$group, 4)
```

```
##      diff      lwr      upr    p adj
## B-A   30.5  19.79  41.209 0.0000
## C-A   -6.1 -16.81   4.609 0.3167
## C-B  -36.6 -47.31 -25.891 0.0000
```

```
par(mar = c(2.5, 4, 0.5, 1))
boxplot(score ~ group, data = data4,
        col = "grey90", las = 1, xlab = "")
points(1:3, means, pch = 18, cex = 1.8,
      col = col_alt)      # group means
```



Tukey: family-wise error rate kept at 5%.

Reading the ANOVA

- $F = 47.7$ on 2 and 12 df, $p < 0.001$: the batch means are not all equal
- Tukey: batch B differs from A (by 30.5) and from C (by 36.6); A and C do not differ (adjusted $p = 0.32$)
- Family-wise 95% intervals keep the overall error rate at 5% across all three comparisons

In words Batch B scored clearly higher than the other two batches, which performed alike.

Check ANOVA assumes independent, normal, equal-variance errors; the regression diagnostics later apply here too.

Correlation

```
data6 <- read.table("data/data15.txt",
                    header = TRUE)
round(cor(data6), 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

r_ab <- cor.test(data6$batchA, data6$batchB)
c(r_ab$estimate, p = r_ab$p.value)

##      cor      p
## -0.82987 0.08206
```

Pairs, and enough of them

Correlation needs two measurements on the same units: read each row as one student's marks in three papers (illustrative). With $n = 5$ only very strong correlations reach significance.

Reading it $r = -0.83$ between A and B looks strong, but $p = 0.082$: not significant at 5%.

PART 6

Regression diagnostics

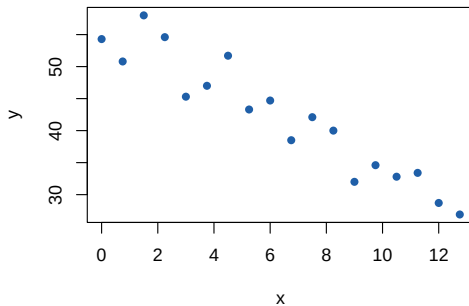
The regression data

```
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)
length(x); cor(x, y)
```

```
## [1] 18
## [1] -0.9487
```

Illustrative data: 18 observations in order, at equally spaced x .

```
par(mar = c(4, 4, 1, 1))
plot(x, y, pch = 16, col = col_main)
```



Fitting the simple linear regression

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

ε_i independent, $N(0, \sigma^2)$

Four assumptions: a **L**inear mean, **I**ndependent errors, **N**ormal errors, **E**qual variance (LINE). Estimates can always be computed; p -values and intervals are trustworthy only if the assumptions hold.

```
M1 <- lm(y ~ x)
display(M1) # arm

## lm(formula = y ~ x)
##               coef.est coef.se
## (Intercept)  56.42      1.39
## x           -2.24      0.19
## ---
## n = 18, k = 2
## residual sd = 3.08, R-Squared = 0.90
```

Reading the fitted model

```
round(confint(M1), 3)
```

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

```
summary(aov(y ~ x))[[1]][, 1:4]
```

```
##           Df Sum Sq Mean Sq F value  
## x           1   1365    1365     144  
## Residuals   16    151         9
```

- $\hat{y} = 56.42 - 2.238x$: each unit of x lowers the mean of y by 2.24 (95% CI 1.84 to 2.63)
- $R^2 = 0.90$: the line explains 90% of the variation in y
- ANOVA $F = 144.2$ on 1 and 16 df; for one predictor $F = t^2$ of the slope

But first All of this relies on the LINE assumptions: check them next.

The assumptions and how to check them

Assumption	If it fails	Plot	Test in R
Linear mean Independence Normality	biased fit and predictions SEs and p -values too small small-sample tests, intervals off	residuals vs fitted residuals vs order normal Q-Q plot	RESET: <code>resettest()</code> Durbin-Watson: <code>dwtest()</code> Shapiro-Wilk: <code>shapiro.test()</code>
Equal variance	SEs and p -values wrong	residuals vs fitted	Breusch-Pagan: <code>bptest()</code>

Plots first A plot shows *how* an assumption fails; a test only says *whether* the sample is large enough to detect it.

Tests and n With small n tests miss real problems; with very large n they flag trivial ones.

Residuals and studentized residuals

```
resi <- residuals(M1)
y.hat <- fitted(M1)
st_res <- studres(M1)          # MASS
round(summary(st_res), 2)

##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##    -1.57  -0.68   -0.17     0.00   0.73     1.96

which(abs(st_res) > 2)

## named integer(0)
```

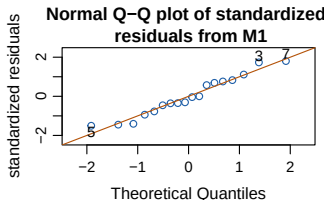
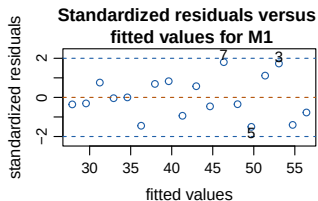
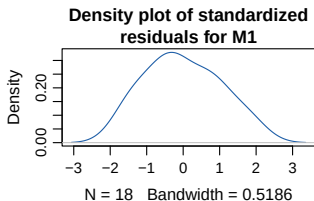
Residual: $e_i = y_i - \hat{y}_i$

Studentized: $t_i = \frac{e_i}{s_{(i)}\sqrt{1 - h_{ii}}}$

h_{ii} : leverage; $s_{(i)}$: residual SD without point i . Look at $|t_i| > 2$; $|t_i| > 3$ suggests an outlier.

Reading it Largest $|t_i| = 1.96$: no outliers.

Four diagnostic plots in one call



```
par(mar = c(4, 4, 2.6, 0.5),  
    mgp = c(2.2, 0.7, 0),  
    cex.lab = 1.1, cex.axis = 1.05)  
checking.plots(M1, # PASWR  
               COL = c(col_main, col_alt))
```

Standardized residuals: against order (top left) and fitted values (bottom left), a normal Q-Q plot and a density estimate. Dashed lines at ± 2 , ± 3 .

Reading the diagnostic plots

- **Residuals vs order** (top left): no runs of same-signed residuals and no trend, so no sign of dependence along the sequence
- **Normal Q-Q** (top right): points lie close to the line
- **Residuals vs fitted** (bottom left): no curve, so a straight line is adequate; the vertical spread looks somewhat larger at higher fitted values
- **Density** (bottom right): roughly symmetric and bell-shaped

Next Back up the visual impressions with a formal test for each assumption: Durbin-Watson, Breusch-Pagan and Shapiro-Wilk.

`checking.plots()` uses standardized residuals and dashed lines at ± 2 and ± 3 . Base R's `plot(M1)` gives a similar set.

Independence: the Durbin–Watson test

$$d = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2} \approx 2(1 - r_1)$$

$H_0 : \rho = 0$ vs $H_1 : \rho > 0$ (default)

r_1 : lag-1 autocorrelation of the residuals.
 $d \approx 2$: none; d near 0: positive; d near 4: negative autocorrelation.

```
dwtest(M1)                                # lmtest

##
## Durbin-Watson test
##
## data:  M1
## DW = 2.4, p-value = 0.8
## alternative hypothesis: true autocorrelation is greater than 0
```

Reading the Durbin–Watson test

```
r1 <- cor(resi[-1], resi[-length(resi)])  
round(r1, 3)  
  
## [1] -0.248  
  
dwtest(M1, alternative = "two.sided")$p.value  
  
## [1] 0.4748
```

- $d = 2.45$, one-sided $p = 0.76$: no evidence of positive autocorrelation
- $d > 2$ reflects a slightly negative r_1 ; the two-sided test ($p = 0.47$) does not detect it

When is DW meaningful? Only when rows have a natural order (time, sequence, space). In cross-sectional survey data the row order is arbitrary.

If it fails Model the dependence or use HAC standard errors; `bgtest()` checks higher-order autocorrelation.

Constant variance: the Breusch-Pagan test

H_0 : $\text{Var}(\varepsilon_i) = \sigma^2$ for all i

H_1 : the variance changes with x

Regress e_i^2 on x_i : $\text{BP} = nR_{\text{aux}}^2 \sim \chi_k^2$

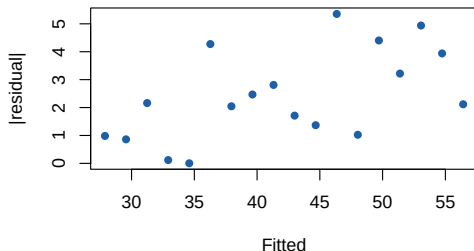
k : number of regressors in the auxiliary regression (here 1). `bptest()` uses Koenker's studentized version, valid even when the errors are not normal.

```
bptest(M1)                                # lmtest

##
## studentized Breusch-Pagan test
##
## data:  M1
## BP = 4.4, df = 1, p-value = 0.04
```

Reading the Breusch–Pagan test

```
par(mar = c(4, 4, 0.5, 1))  
plot(y.hat, abs(resi), pch = 16,  
     col = col_main, xlab = "Fitted",  
     ylab = "|residual|")
```



- $BP = 4.44$ on 1 df, $p = 0.035$: at 5% we reject constant variance
- The plot agrees: larger residuals at larger fitted values (small x)
- Moderate evidence only: 18 points, p near 0.05

Why it matters The slope stays unbiased, but the usual standard errors, t -tests and intervals can be wrong.

A remedy: heteroscedasticity-consistent standard errors

```
coeftest(M1, vcov = hccm(M1, type = "hc3"))      # lmtest + car

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   56.418      1.795    31.4 8.2e-16 ***
## x             -2.238      0.196   -11.4 4.1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reading it HC3 “sandwich” standard errors allow non-constant variance: the slope SE moves from 0.186 to 0.196 and the slope stays highly significant.

Normality of the residuals

H_0 : the errors are normal

Shapiro–Wilk W measures how well the ordered residuals match those expected from a normal sample (like a straight Q–Q plot); small W signals non-normality

A normal sample has skewness and excess kurtosis near 0.

```
shapiro.test(resi)
```

```
##
```

```
##  Shapiro-Wilk normality test
```

```
##
```

```
## data:  resi
```

```
## W = 0.96, p-value = 0.5
```

```
lillie.test(resi)$p.value      # nortest
```

```
## [1] 0.7814
```

```
c(skewness(resi), kurtosis(resi)) # e1071
```

```
## [1] 0.1763 -1.1473
```

Reading the normality checks

- Shapiro–Wilk $p = 0.54$, Lilliefors $p = 0.78$: no evidence against normality
- Skewness 0.18, excess kurtosis -1.15 : a little flatter than normal
- In R Shapiro–Wilk works for $3 \leq n \leq 5000$; the rule “Shapiro if $n < 50$, else KS” is not needed

```
ks.test(resi, "pnorm", mean(resi),  
        sd(resi))$p.value
```

```
## [1] 0.9542
```

Not plain KS `ks.test()` with mean and SD estimated from the same data gives a p -value that is too large (0.95 vs 0.78); Lilliefors corrects it.

Perspective Normality matters least of the four, especially in large samples; independence and equal variance matter more.

Diagnostics summary for the fitted line

Assumption	Evidence	Result	Verdict
Linear mean	plot; RESET test	no curvature, $p = 0.69$	satisfied
Independence	Durbin–Watson	$d = 2.45$, $p = 0.76$	satisfied
Equal variance	Breusch–Pagan	$BP = 4.44$, $p = 0.035$	doubtful: use robust SEs
Normality	Shapiro–Wilk	$W = 0.957$, $p = 0.54$	satisfied
Outliers	studentized residuals	$\max t_i = 1.96$	none

Overall A straight line describes the data well; y falls by about 2.24 per unit of x , and this conclusion survives the one doubtful assumption.

Key take-aways

1. **Scripts and relative paths** make an analysis repeatable by anyone.
2. **Check a test's conditions:** small expected counts call for Fisher's exact test, not `suppressWarnings()`.
3. **The design picks the test:** independent or paired, one- or two-sided, decided in advance.
4. **A regression is not finished until its residuals are checked:** plots first, then Durbin-Watson, Breusch-Pagan and Shapiro-Wilk.
5. **Know the remedies:** robust standard errors, transformations, modelling the dependence.
6. **Report estimates with intervals**, and say what they mean in words.

References

- Breusch, T. S., and Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5), 1287–1294.
- Durbin, J., and Watson, G. S. (1950). Testing for serial correlation in least squares regression. I. *Biometrika*, 37(3/4), 409–428.
- Fox, J., and Weisberg, S. (2019). *An R Companion to Applied Regression*, 3rd Ed., Sage, Thousand Oaks.
- Lilliefors, H. W. (1967). On the Kolmogorov–Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318), 399–402.
- Montgomery, D. C., Peck, E. A., and Vining, G. G. (2021). *Introduction to Linear Regression Analysis*, 6th Ed., Wiley, Hoboken.
- Shapiro, S. S., and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591–611.
- 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