Master Guide - Chapter 3: Linear Regression & Correlation

DANA-4810 - Compiled Guide

2025-10-10

Contents

Topics covered

- Simple linear regression (least squares)
- Model diagnostics: residuals, coefficient of variation, R-squared
- Confidence intervals and prediction intervals for mean and individual responses
- Correlation and hypothesis tests for rho
- Multiple regression, interaction and quadratic terms (overview)

```
flowchart TD
  Start([Start: You have data]) --> A{Goal}
  A -->|Estimate relationship| B[Fit linear model: lm()]
  A -->|Test association| C[Use cor.test()]
  B --> D{Assess fit}
  D -->|Good fit| E[Report coef, CI, R-squared]
  D -->|Poor fit| F[Check residuals, transform or add terms]
  F --> G{Add terms}
  G -->|Interaction| H[Include x1*x2]
  G -->|Quadratic| I[Include I(x^2)]
  E --> End([Use predict() for CI/PI and interpret])
```

Decision notes: follow the flow above when deciding whether to fit a model, test correlation, or expand to multiple regression. The snippets below use the exact calls from the course Rmds.

1. Fitting the Model: Method of Least Squares

Loading example data (ADSALES):

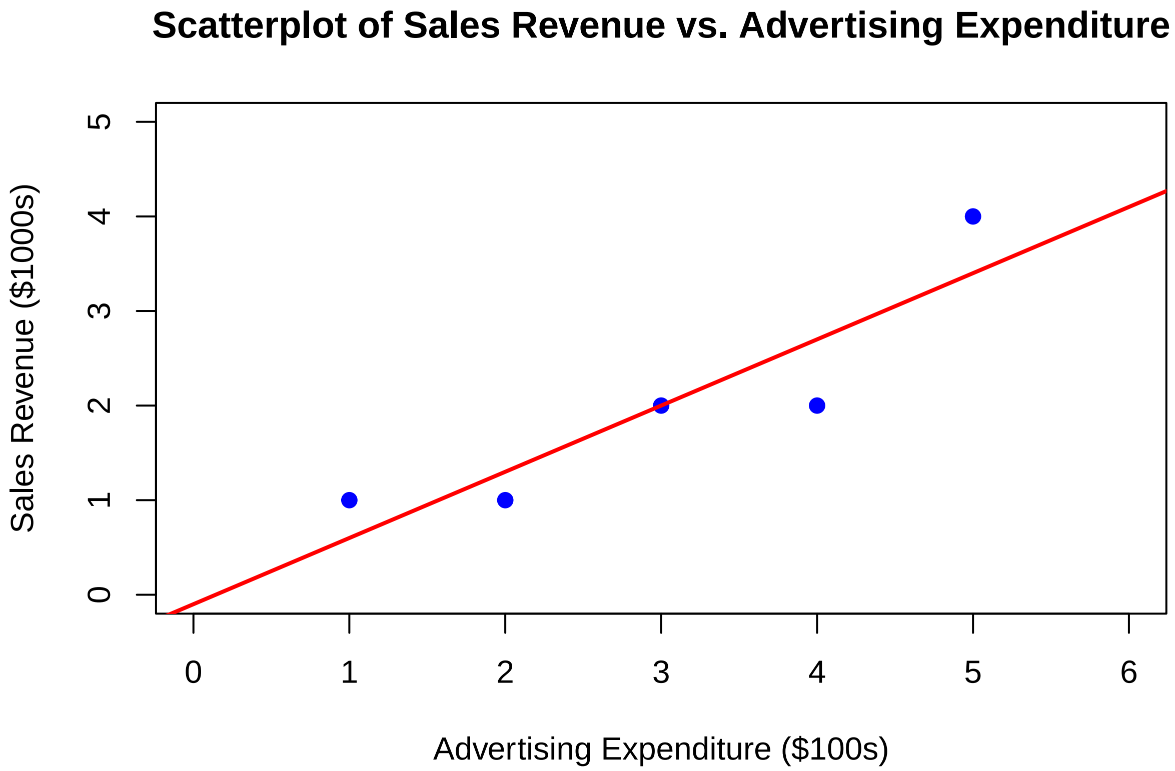
```r
ADSALES <- read.table("ADSALES.txt", header=TRUE)
# create variables used in later chunks to avoid attach()/detach() issues
ADVEXP_X <- ADSALES$ADVEXP_X
SALES_Y <- ADSALES$SALES_Y
head(ADSALES)
```

```
##   ADVEXP_X SALES_Y
## 1        1       1
## 2        2       1
## 3        3       2
## 4        4       2
## 5        5       4
```

Note: this file expects to be knit from the `Chapter3/R-Files` directory (the same folder as this Rmd). All datasets used in this guide are in this folder: `ADSALES.txt`, `FIREDAM.txt`, `CASINO.txt`, and `TIRES.txt`.

Scatterplot and regression line (ADSALES):

```r
plot(x=ADVEXP_X, y=SALES_Y, ylab = "Sales Revenue ($1000s)", xlab = "Advertising Expenditure ($100s)",
     main = "Scatterplot of Sales Revenue vs. Advertising Expenditure",
     col="blue", ylim = c(0,5), xlim = c(0,6), pch=19)
abline(lm(SALES_Y~ADVEXP_X), col="Red", lty=1, lwd=2)
```



Fit the linear model, show ANOVA, coefficients, and predicted values:

```
model=lm(SALES_Y~ADVEXP_X)
summary(model)
```

```
##
## Call:
## lm(formula = SALES_Y ~ ADVEXP_X)
##
## Residuals:
##          1          2          3          4          5
##  4.000e-01 -3.000e-01  6.478e-17 -7.000e-01  6.000e-01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1000     0.6351  -0.157   0.8849
## ADVEXP_X      0.7000     0.1915   3.656   0.0354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6055 on 3 degrees of freedom
## Multiple R-squared:  0.8167, Adjusted R-squared:  0.7556
## F-statistic: 13.36 on 1 and 3 DF,  p-value: 0.03535
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: SALES_Y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## ADVEXP_X   1    4.9  4.9000  13.364 0.03535 *
## Residuals  3    1.1  0.3667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coef(model)
```

```
## (Intercept)    ADVEXP_X
##        -0.1         0.7
```

```r
# Predictions for ADVEXP_X = 4,5 (i.e., $400 and $500 in the lecture scaling)
New=data.frame(ADVEXP_X=c(4,5))
predict(model,New)
```

```
##   1   2
## 2.7 3.4
```

Interpretation of this output (how to read what R prints):

- summary(model): shows coefficients (Estimate), their Std. Error, t value and Pr(>|t|). A small p-value for the slope indicates evidence that advertising (ADVEXP_X) is associated with SALES_Y.
- anova(model): partitions variance into regression and residual; the F-statistic and its p-value test whether the model explains a significant amount of variability compared to an intercept-only model.
- coef(model): returns the intercept and slope estimates used in predictions.
- predict(…, interval = "confidence"): returns the estimated mean response and a confidence interval for the population mean at the new x values.
- predict(…, interval = "prediction"): returns prediction intervals for individual future observations (wider than CI).

Typical next steps after these calls:

1. Check residuals and model sigma: `s = summary(model)$sigma` (an estimate of the residual standard deviation).
2. Compute coefficient of variation: `cv = (s/mean(SALES_Y))*100` to understand relative error.
3. If residuals show patterns, consider transformations or adding interaction/quadratic terms as shown later.

Coefficient of variation and model sigma:

```r
s=summary(model)$sigma
cv=(s/mean(SALES_Y))*100
cv
```

```
## [1] 30.2765
```

95% confidence and 95% prediction intervals (example for ADVEXP_X=4):

```
predict(model,newdata=data.frame(ADVEXP_X=4), interval="confidence", level=0.95)
```

```
##   fit      lwr      upr
## 1 2.7 1.644502 3.755498
```

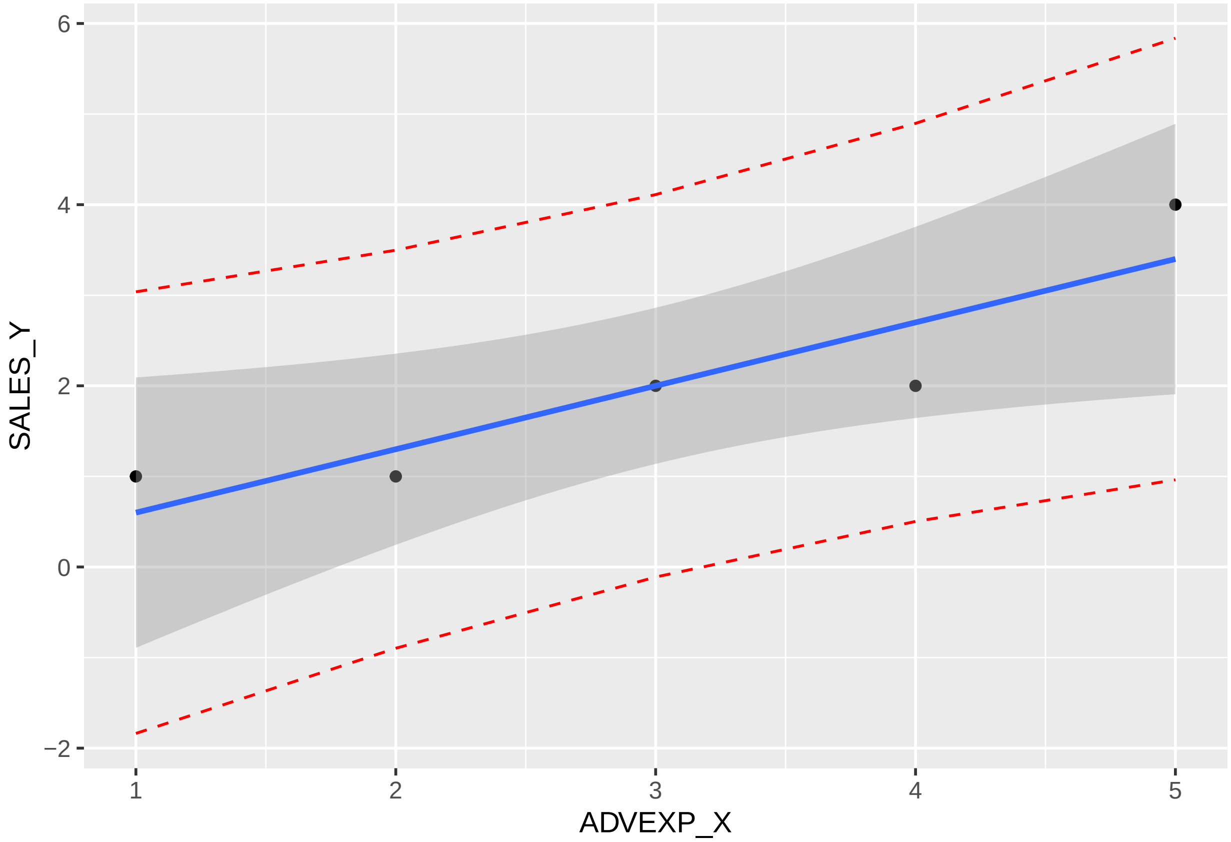
```
predict(model,newdata=data.frame(ADVEXP_X=4), interval="prediction", level=0.95)
```

```
##   fit       lwr      upr
## 1 2.7 0.5028056 4.897194
```

Alternative: ggplot approach used in the lecture material

```
library("ggplot2")
pred=predict(model, interval = "prediction")
new_df = cbind(ADSALES, pred)

ggplot(new_df, aes(ADVEXP_X, SALES_Y))+
        geom_point()+
        geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
        geom_line(aes(y=upr), color = "red", linetype = "dashed")+
        geom_smooth(method=lm, se=TRUE)
```



2. Model assessment and inference (examples from FIREDAM)

```
FIREDAM <- read.table("FIREDAM.txt", header=TRUE)
# avoid attach(): create explicit variables
DISTANCE <- FIREDAM$DISTANCE
DAMAGE <- FIREDAM$DAMAGE
model = lm(DAMAGE ~ DISTANCE)
summary(model)
```

```
##
## Call:
## lm(formula = DAMAGE ~ DISTANCE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4682 -1.4705 -0.1311  1.7915  3.3915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2779     1.4203   7.237 6.59e-06 ***
## DISTANCE      4.9193     0.3927  12.525 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.316 on 13 degrees of freedom
## Multiple R-squared:  0.9235, Adjusted R-squared:  0.9176
## F-statistic: 156.9 on 1 and 13 DF,  p-value: 1.248e-08
```

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: DAMAGE
##           Df Sum Sq Mean Sq F value    Pr(>F)
## DISTANCE   1 841.77  841.77  156.89 1.248e-08 ***
## Residuals 13  69.75    5.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
round(confint(model, level = 0.95), 3)
```

```
##             2.5 % 97.5 %
## (Intercept) 7.210 13.346
## DISTANCE    4.071  5.768
```

Examples: correlation tests (CASINO and TIRES datasets)

```
CASINO <- read.table("CASINO.txt", header=TRUE)
cor.test(x = CASINO$EMPLOYEES, y = CASINO$CRIMERAT)
```

```
##
##  Pearson's product-moment correlation
##
## data:  CASINO$EMPLOYEES and CASINO$CRIMERAT
## t = 17.39, df = 8, p-value = 1.219e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9441628 0.9970373
## sample estimates:
##       cor
## 0.9870298
```

```r
TIRES <- read.table("TIRES.txt", header=TRUE)
cor.test(x = TIRES$PRESS_X, y = TIRES$MILEAGE_Y)
```

```
##
##  Pearson's product-moment correlation
##
## data:  TIRES$PRESS_X and TIRES$MILEAGE_Y
## t = -0.39647, df = 12, p-value = 0.6987
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6076297  0.4436352
## sample estimates:
##        cor
## -0.1137098
```

3. Choosing between correlation and simple linear regression (Chapter 3 scope)

Chapter 3 focuses on bivariate relationships and simple linear models. The examples in this chapter (ADSALES, FIREDAM, CASINO, TIRES) illustrate the common tasks:

- Correlation: quantify and test linear association between two variables (`cor.test()`).
- Simple linear regression: estimate the conditional mean of Y given X, test the slope, and predict (`lm()`, `summary()`, `anova()`, `predict()`).

When to use which:

- If the question asks only whether two variables move together (no predictive intent), use `cor.test()` and report Pearson's r, t-stat, and p-value.
- If the question asks how the expected value of Y changes with X, or asks for predictions at new X values, use `lm()` and report coefficients, CI, and prediction intervals.

Chapter 3 examples below follow this pattern. This chapter's R-Files do not include multi-predictor regression examples — focus on single-predictor models and the diagnostic/inference steps shown above.

Detailed study notes — Chapter 3 (linear regression & correlation)

These notes explain theory, assumptions, calculations, diagnostics and interpretation you need to study for the exam. All R code examples use only functions and data files present in the Chapter 3 folder (ADSALES, FIREDAM, CASINO, TIRES).

Short mathematical contract

- Input: paired numeric data (x, y) or a data frame with predictors and response.
- Output: fitted linear model object from `lm()`, summary statistics, confidence/prediction intervals, and model diagnostics.
- Error modes: missing files (fix by using correct relative path), non-numeric inputs, too few observations (df issues).

Key formulae (as used in the course)

- Estimated slope and intercept: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. Coefficients are printed by `coef(model)` or `summary(model)$coefficients`.
- Standard error of slope: $SE(\hat{\beta}_1)$ (R prints this). Manual t-statistic for slope: $t = \dfrac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$.
- Confidence interval for slope: $\hat{\beta}_1 \pm t_{1-\alpha/2,\,df} \cdot SE(\hat{\beta}_1)$ (use `confint(model)`).
- Residual standard error (sigma): printed as `Residual standard error` in `summary(model)` and available with `summary(model)$sigma`.
- Coefficient of determination: R^2 and adjusted R^2 appear in `summary(model)`.

Assumptions of the simple linear regression model

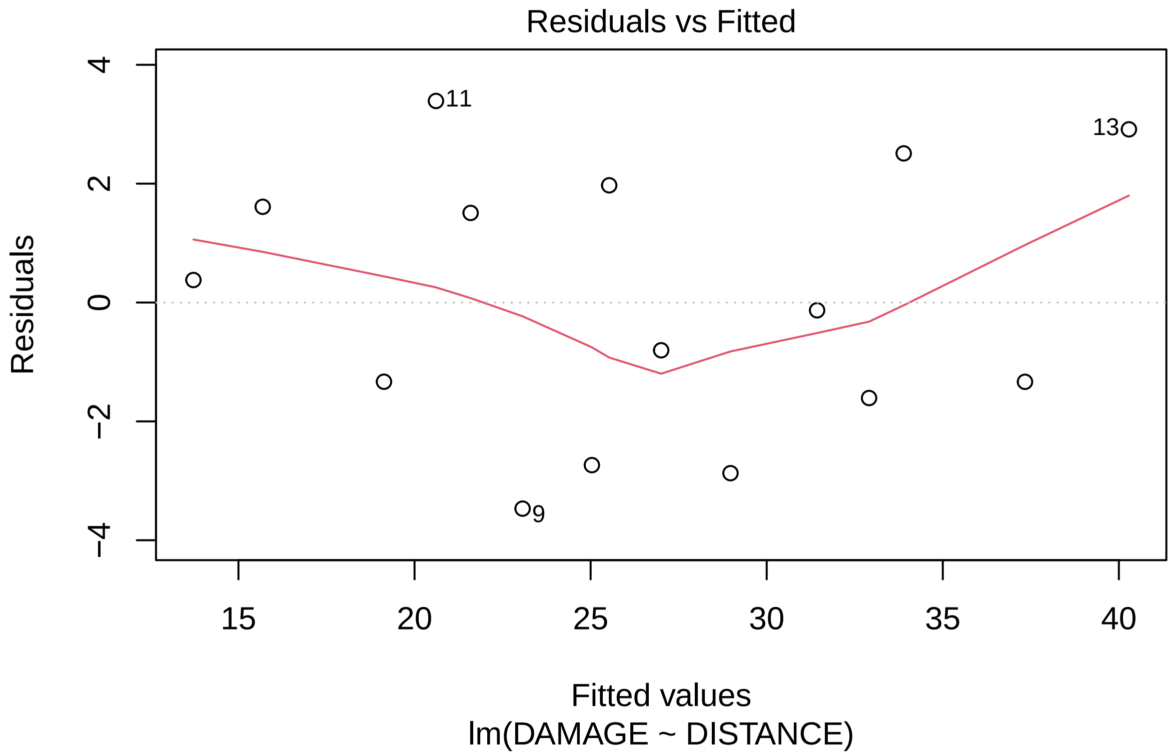
1. Linearity: the conditional mean of Y is linear in X.
2. Independence: observations are independent.
3. Constant variance (homoscedasticity): $\text{Var}(\varepsilon) = \sigma^2$ for all x.
4. Normality: residuals are approximately Normal (for small samples this matters for t and F inference).

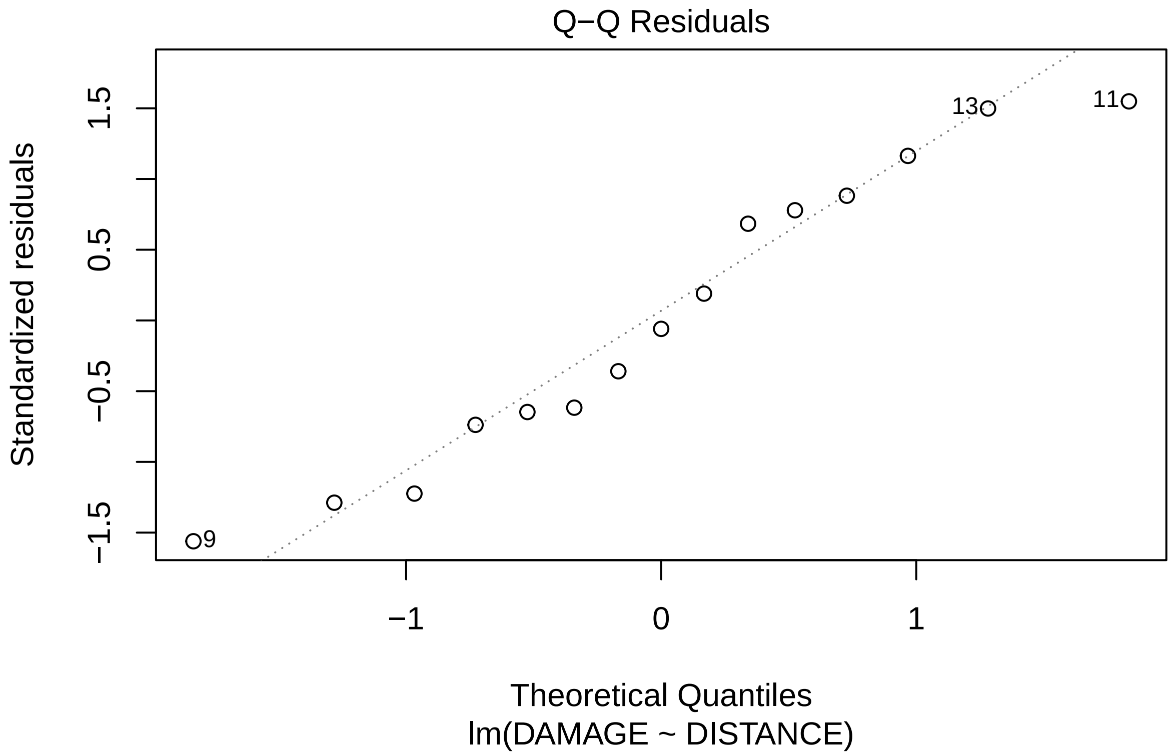
Exam tip: you must both state these assumptions and demonstrate them with diagnostics (residual plots and QQ plot).

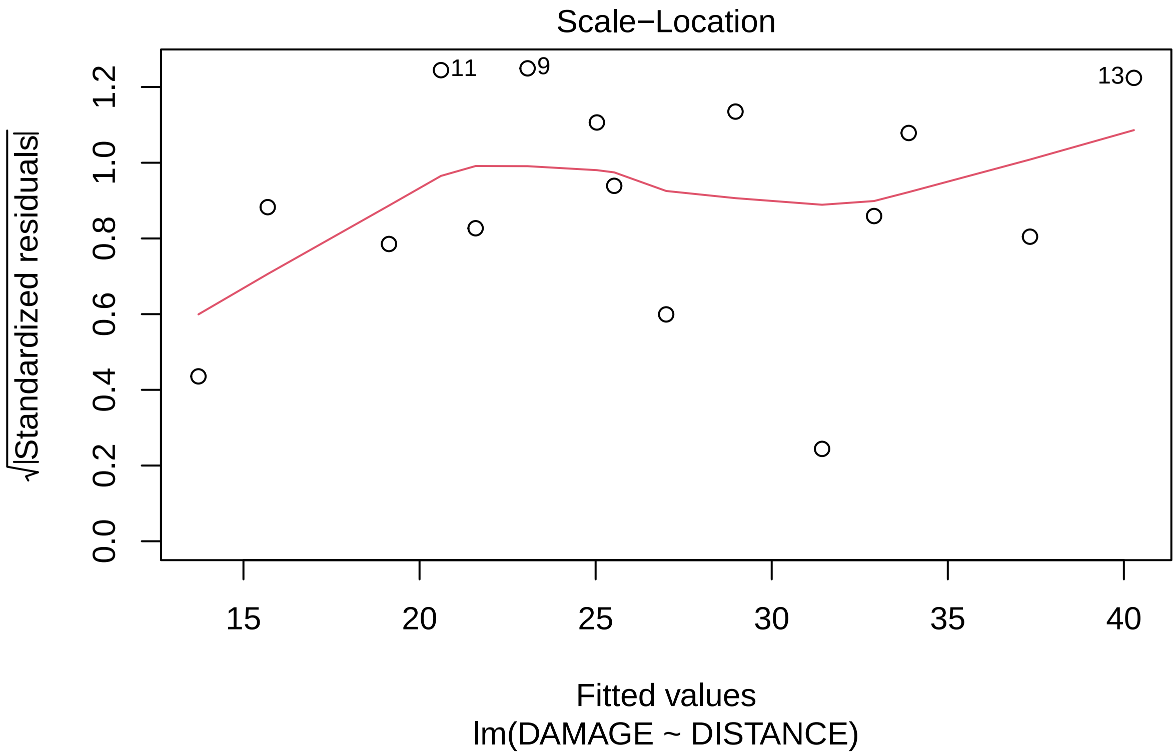
Diagnostics — how to check assumptions (exact code used in course Rmds)

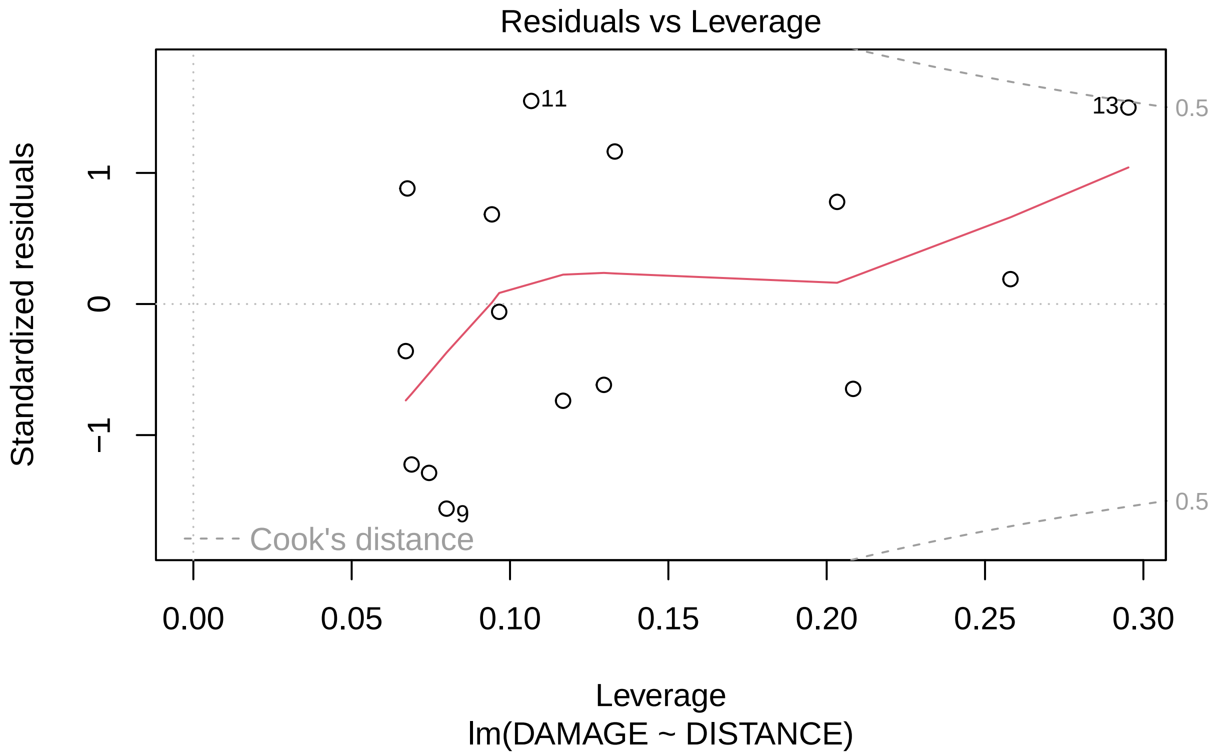
- Residual plots (look for nonlinearity, heteroscedasticity):

```r
plot(model)   # the base plot method gives Residuals vs Fitted, QQ, Scale-Location, Cook's distance
```


Residuals vs Fitted
Residuals
4
2
0
−2
−4
11
13
9
15
20
25
30
35
40
Fitted values
lm(DAMAGE ~ DISTANCE)

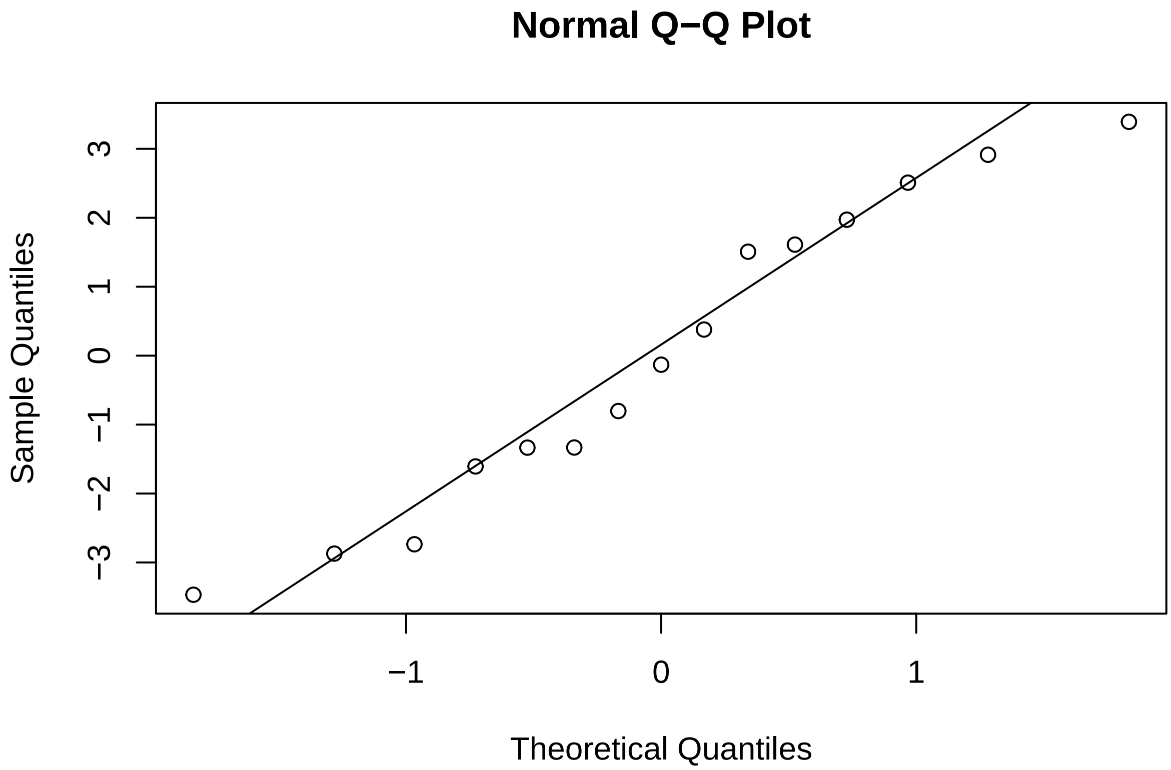

Q−Q Residuals
Standardized residuals
1.5
0.5
−0.5
−1.5
−1
0
1
Theoretical Quantiles
lm(DAMAGE ~ DISTANCE)
9
13
11


Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
11
9
13
15
20
25
30
35
40
Fitted values
lm(DAMAGE ~ DISTANCE)



- QQ-plot for residuals (explicit):

```
qqnorm(resid(model)); qqline(resid(model))
```



- Check the residual standard error and CV:

```
s = summary(model)$sigma
cv = (s/mean(model$model[[2]]))*100  # model$model[[2]] is the response vector
s; cv
```

```
## [1] 2.316346
```

```
## [1] 70.62031
```

Interpretation guidance:

- Residuals vs Fitted: no pattern → linearity and constant variance plausible.
- QQ-plot: points near the line → residuals approx Normal.
- Large Cook's distances indicate influential observations to examine.

Inference — coefficients and hypothesis tests

When you run `summary(model)` R prints coefficient estimates, standard errors, t-values and p-values. Use these rules for exam answers:

- Null for slope: $H_0 : \beta_1 = 0$. Reject if p-value small (e.g., < 0.05).
- Two-sided CI for slope from `confint(model)`: if it excludes 0, slope is significant at that level.

- The F-test in `anova(model)` or the model F-statistic in `summary(model)` tests the null that all slope coefficients are 0 (for simple regression this is the same as testing the slope).

Manual t example (how to show work on exam using R outputs):

1. From `summary(model)` read Estimate and Std. Error for slope, e.g. Estimate = 0.8, SE = 0.12.
2. Compute $t = 0.8/0.12$ and compare to $t_{1-\alpha/2,df}$ (use `qt()`), or report p-value from R.

Confidence vs Prediction intervals (practical difference)

- Confidence interval for mean response at x_0: use `predict(model, newdata = data.frame(x = x0), interval = "confidence")`.
- Prediction interval for a new observation at x_0: use `predict(model, newdata = data.frame(x = x0), interval = "prediction")`.
- Prediction intervals are wider because they include variability of individual outcomes plus uncertainty in mean estimate.

Correlation (bivariate association)

Use `cor.test(x, y)` (as in `CASINO` and `TIRES` examples). R prints Pearson's r, t-statistic, df and p-value.

- Exam interpretation: report the correlation estimate, the p-value, and comment on strength/direction (e.g., r = 0.7 strong positive).

Extensions beyond single-predictor models (note)

Chapter 3 material and the provided R-Files concentrate on single-predictor regression and correlation. If you later study multiple regression, interactions or quadratic terms, the general approach is similar: specify the model with `lm()` (using `x1*x2` or `I(x^2)`), and compare nested models with `anova()`. Those topics are covered in later chapters; for Chapter 3 focus on mastering single-predictor inference, diagnostics and interpretation.

Practical exam checklist (step-by-step for a typical question)

1. State the model and clearly define variables and parameter of interest.
2. Fit the model with `lm()` and report estimates with `coef()` and `confint()`.
3. Check assumptions: residuals vs fitted, QQ-plot, residual standard error and CV.
4. Perform tests: slope test (t), model F-test (anova or summary), correlation test (`cor.test`).
5. Compute intervals for new x: `predict(..., interval = "confidence")` and `predict(..., interval = "prediction")`.
6. Interpret results in context: mention units, direction, strength, and practical significance.

Common pitfalls (short)

- Forgetting to check residuals before trusting p-values.
- Confusing CI for mean vs PI for an individual.
- Using `attach()` without specifying dataset in code examples — it simplifies interactive exploration but is not recommended in reproducible scripts; course Rmds use it, so you can use the same when following examples.

Practical analyst workflow — step-by-step (apply these in this order)

This section gives a compact, repeatable workflow you can follow in assignments or on the exam. Each step shows the exact R commands (from the course files) to run, what to look for, and alternatives when things go wrong.

1) Define the question and inspect the data

- Goal: be explicit about the parameter of interest (e.g., relationship between ADVEXP_X and SALES_Y, predict SALES_Y for a given ADVEXP_X).
- Code (ADSALES example):

```
ADSALES <- read.table("ADSALES.txt", header = TRUE)
# View(ADSALES)
attach(ADSALES)
head(ADSALES)
```

```
##   ADVEXP_X SALES_Y
## 1        1       1
## 2        2       1
## 3        3       2
## 4        4       2
## 5        5       4
```

```
summary(ADSALES)
```

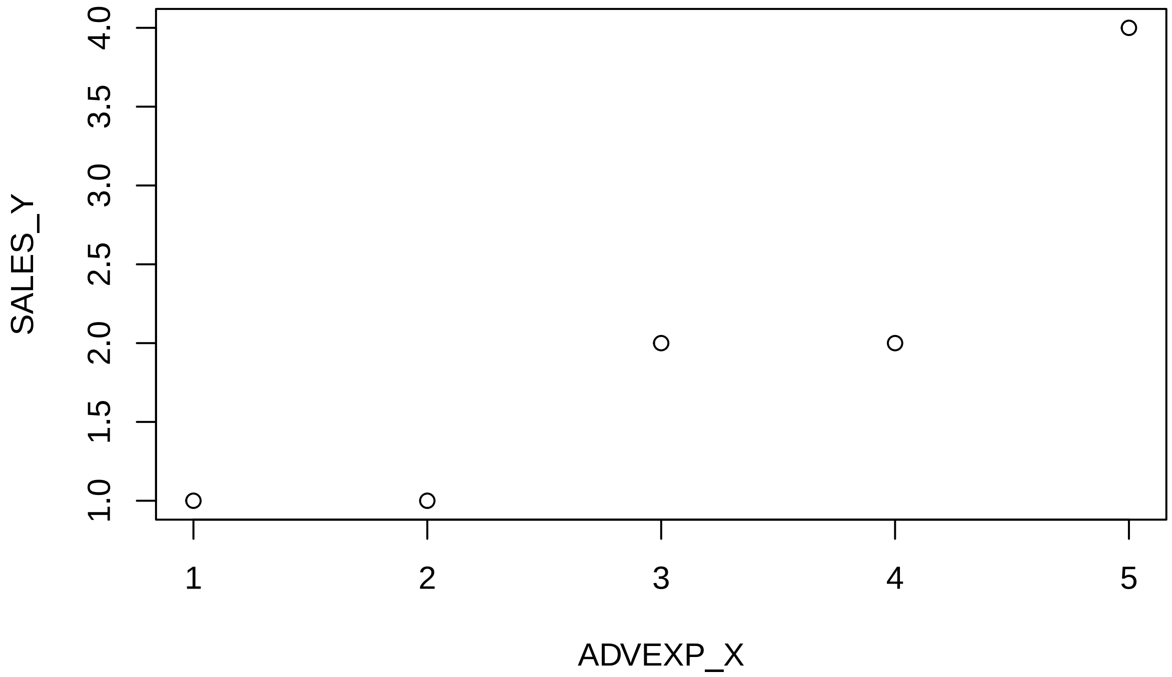
```
##     ADVEXP_X    SALES_Y
##  Min.   :1   Min.   :1
##  1st Qu.:2   1st Qu.:1
##  Median :3   Median :2
##  Mean   :3   Mean   :2
##  3rd Qu.:4   3rd Qu.:2
##  Max.   :5   Max.   :4
```

- Look for: sample size, missing values, variable types.
- Alternative: if variables are not numeric or have NA, clean/convert before modeling (e.g., use `na.omit()` or `as.numeric()` where appropriate).

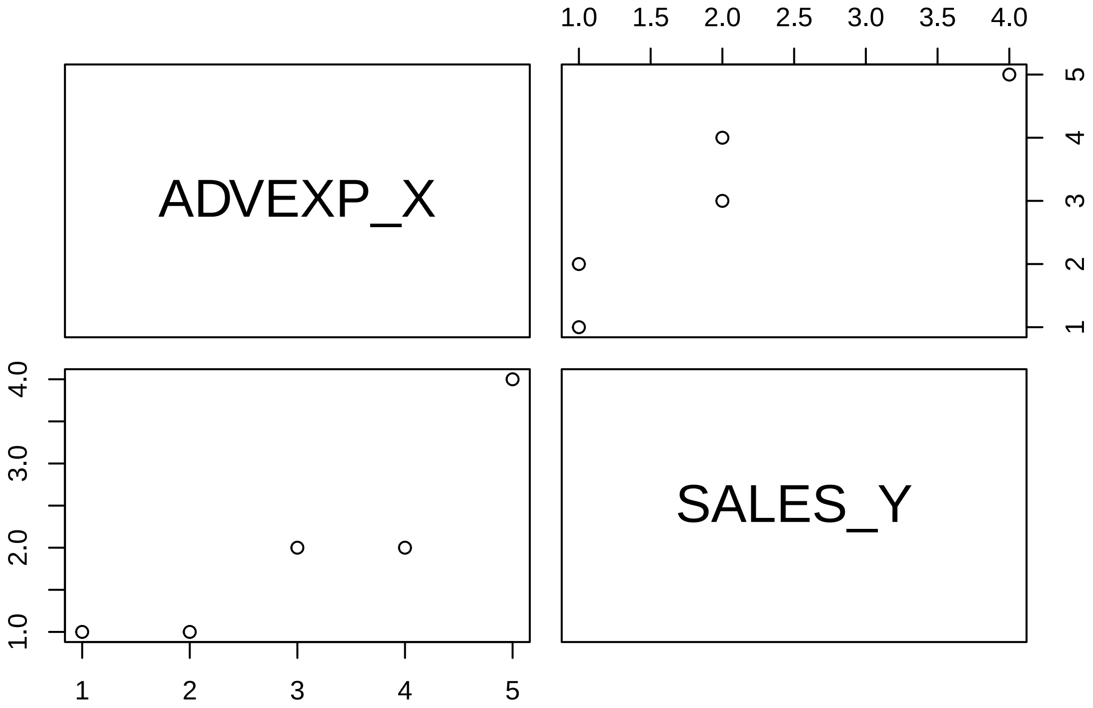
2) Visualize relationships and distribution

- Goal: detect linearity, outliers, groups, heteroscedasticity.
- Code:

```
plot(x = ADVEXP_X, y = SALES_Y)
```



```r
pairs(~ADVEXP_X+SALES_Y)  # use pairs() for multivariate view
```



- Look for: straight-line pattern (good), curvature (consider transformation), outliers (flag for later).
- Alternative: use boxplots or ggplot2 (course shows ggplot, but base plots are fine for the exam).

3) Quick association test (if the question asks about correlation)

- Goal: test whether two variables are linearly associated.
- Code:

```
CASINO <- read.table("CASINO.txt", header = TRUE)
# View(CASINO)
attach(CASINO)
head(CASINO)
```

```
##   EMPLOYEES CRIMERAT
## 1        15     1.35
## 2        18     1.63
## 3        24     2.33
## 4        22     2.41
## 5        25     2.63
## 6        29     2.93
```

```
summary(CASINO)
```

```
##    EMPLOYEES       CRIMERAT
##  Min.   :15.0   Min.   :1.350
##  1st Qu.:22.5   1st Qu.:2.350
##  Median :27.0   Median :2.780
##  Mean   :26.8   Mean   :2.773
##  3rd Qu.:31.5   3rd Qu.:3.373
##  Max.   :38.0   Max.   :4.150
```

```r
cor.test(x = EMPLOYEES, y = CRIMERAT)  # from CASINO example
```

```
##
##  Pearson's product-moment correlation
##
## data:  EMPLOYEES and CRIMERAT
## t = 17.39, df = 8, p-value = 1.219e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9441628 0.9970373
## sample estimates:
##       cor
## 0.9870298
```

- Read: report Pearson's r (estimate), t statistic and p-value. If assumptions (normality) are violated or data are rank-based, an alternative is Spearman (`cor.test(..., method = "spearman")`).

4) Fit a simple linear regression (when estimating relationship/prediction)

- Goal: estimate slope/intercept and test if slope differs from zero.
- Code:

```r
model = lm(SALES_Y ~ ADVEXP_X)
summary(model)
```

```
##
## Call:
## lm(formula = SALES_Y ~ ADVEXP_X)
##
## Residuals:
##          1          2          3          4          5
##  4.000e-01 -3.000e-01  6.478e-17 -7.000e-01  6.000e-01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1000     0.6351  -0.157   0.8849
## ADVEXP_X      0.7000     0.1915   3.656   0.0354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6055 on 3 degrees of freedom
## Multiple R-squared:  0.8167, Adjusted R-squared:  0.7556
## F-statistic: 13.36 on 1 and 3 DF,  p-value: 0.03535
```

```
anova(model)
```

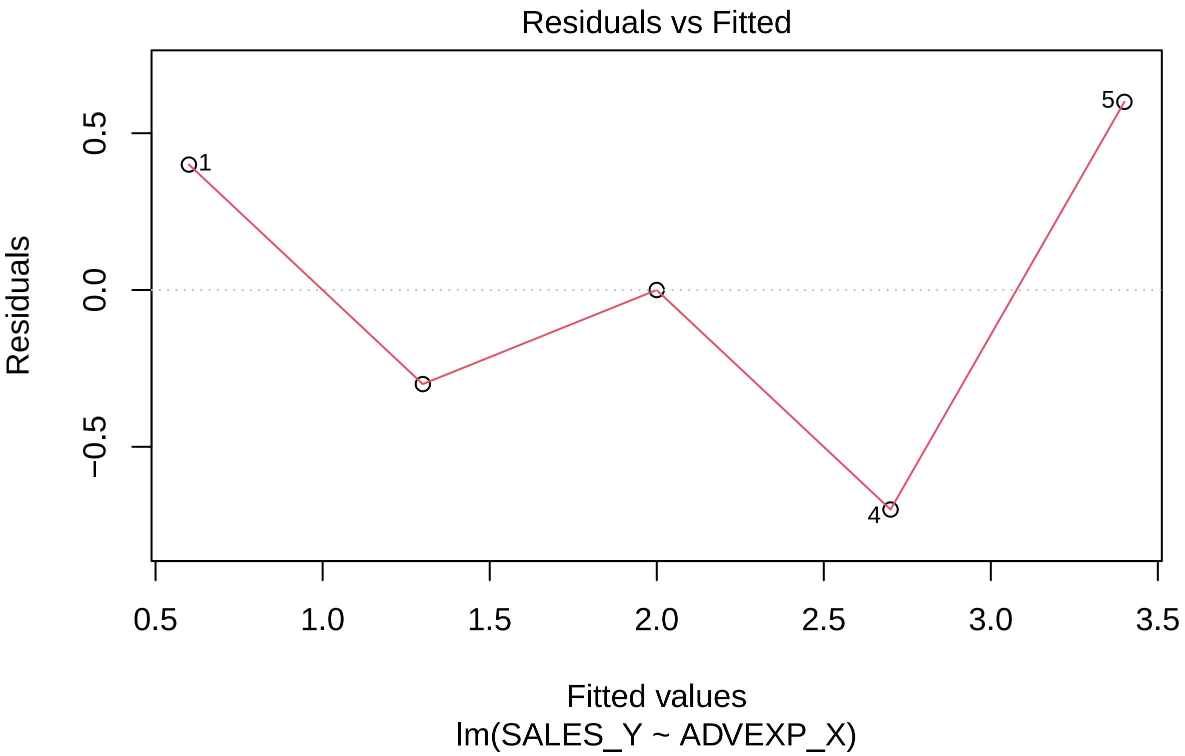
```
## Analysis of Variance Table
##
## Response: SALES_Y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## ADVEXP_X   1    4.9  4.9000  13.364 0.03535 *
## Residuals  3    1.1  0.3667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

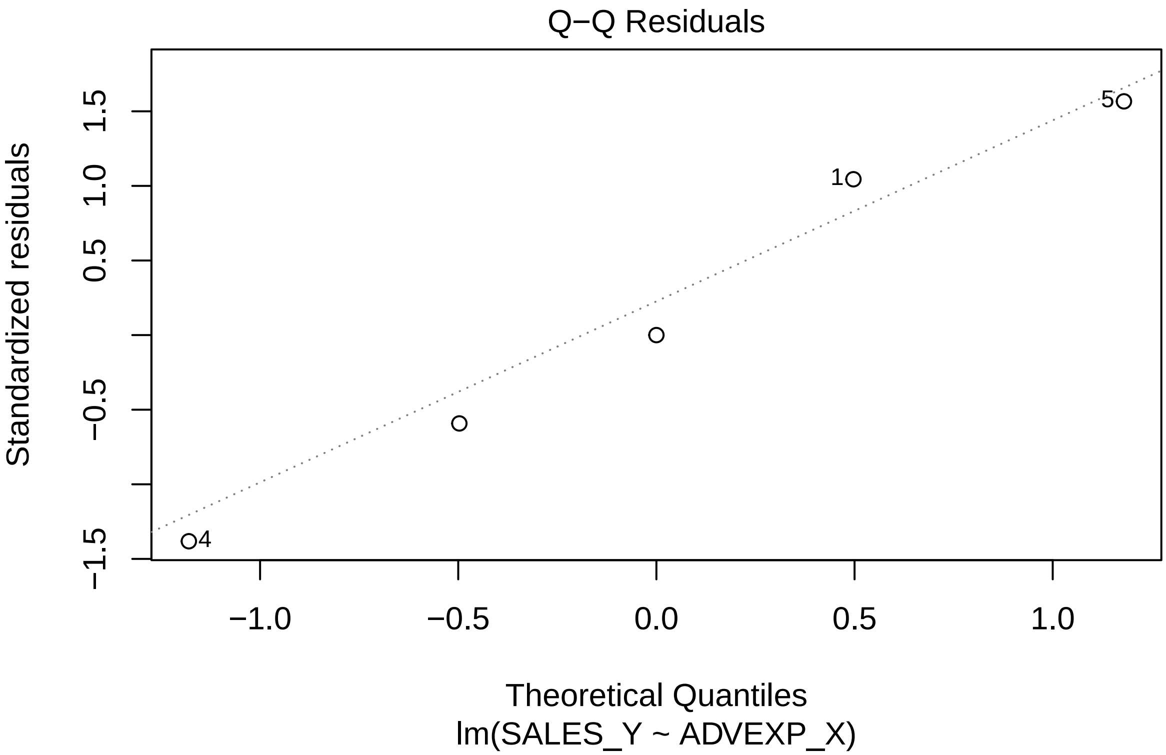
- Read: report Estimate, Std. Error, t value and Pr(>|t|) for slope; Residual standard error and R-squared.
- Alternative: if sigma known (rare in practice), z-based procedures exist (Chapter 1), but in these Rmds we use `lm()`/t-based inference.

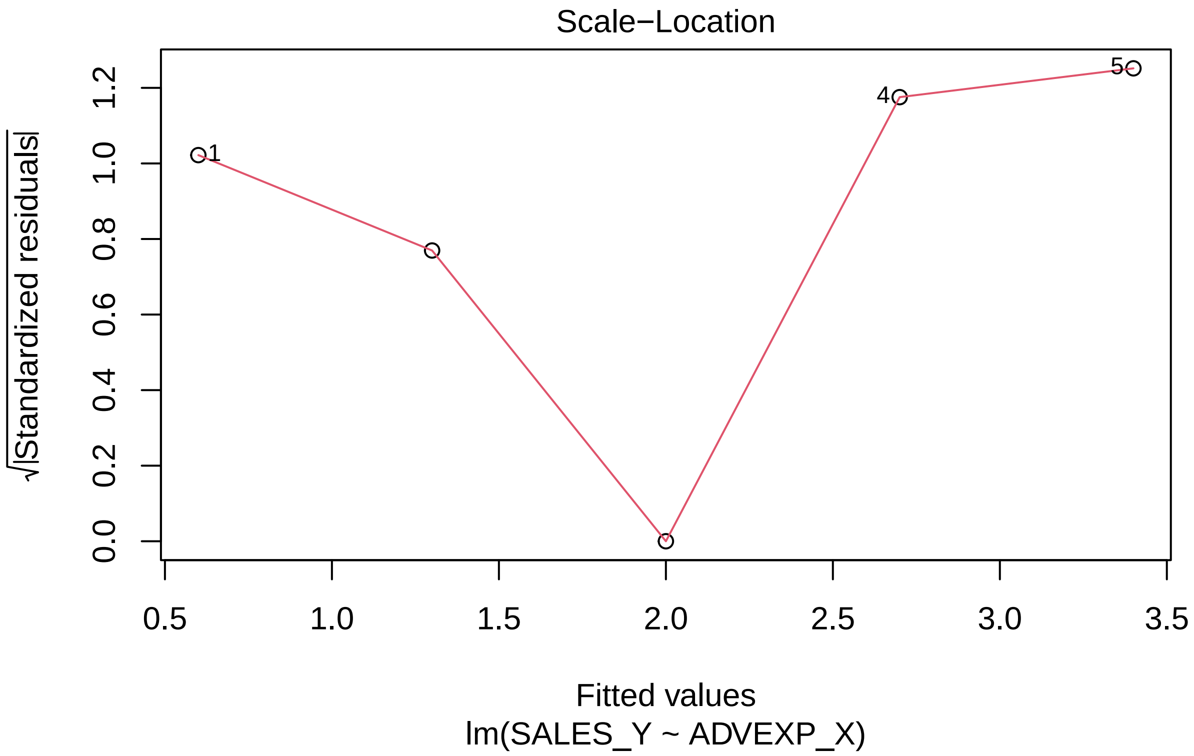
5) Diagnostics — always after fitting a model

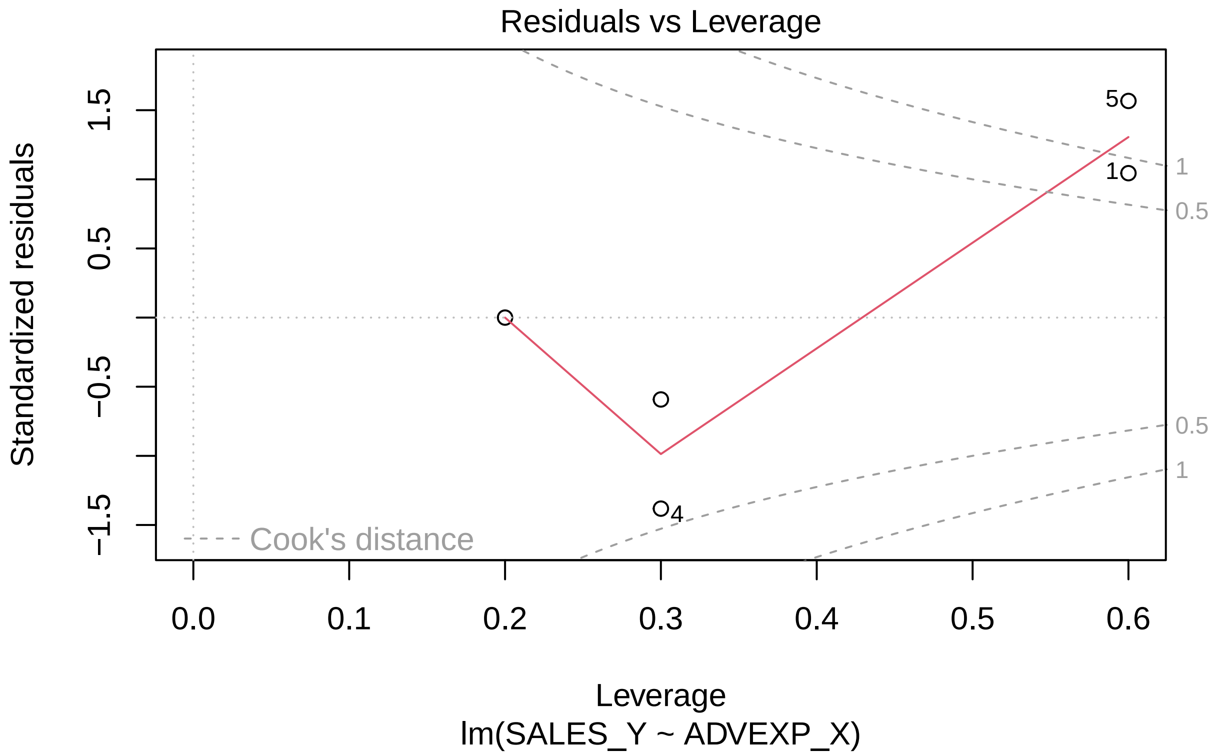
- Goal: check linearity, homoscedasticity, and normality of residuals.
- Code (course pattern):

```
plot(model)                # Residuals vs Fitted, QQ, Scale-Location, Cook's distance
```

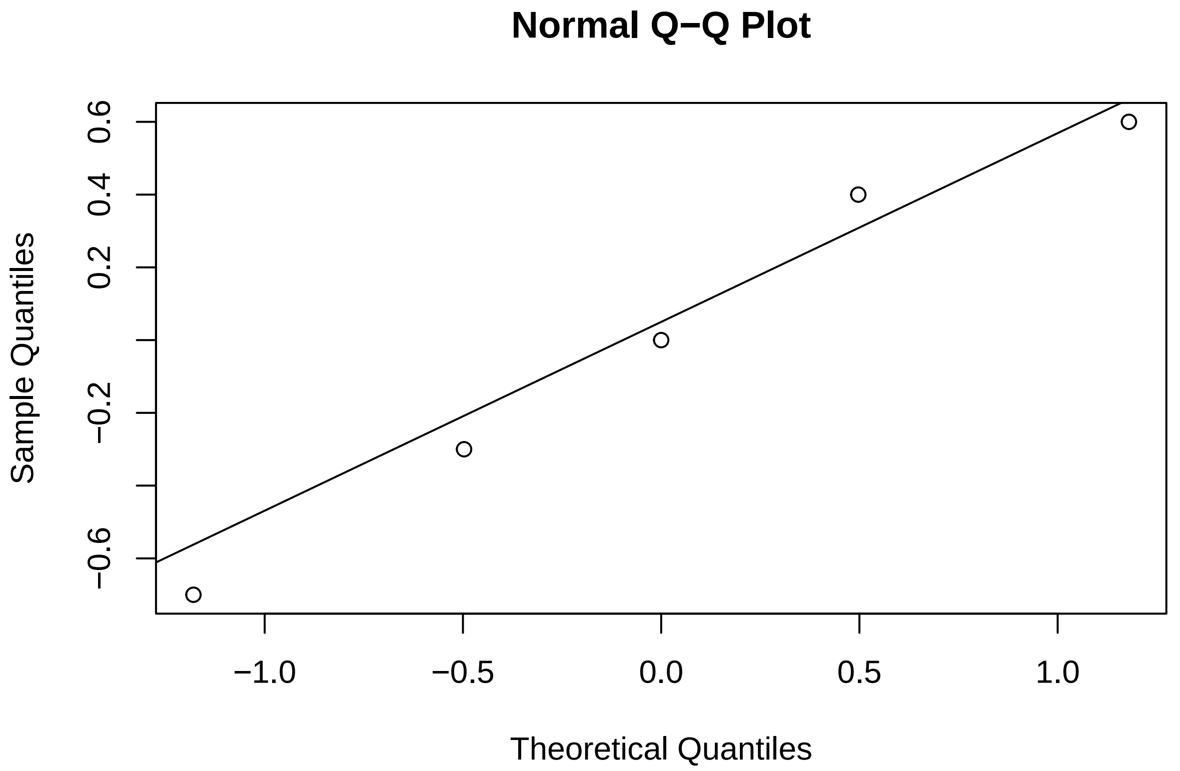



Q−Q Residuals
Standardized residuals
1.5
1.0
0.5
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
5
1
4
Theoretical Quantiles
lm(SALES_Y ~ ADVEXP_X)


Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1
4
5
Fitted values
lm(SALES_Y ~ ADVEXP_X)



```r
qqnorm(resid(model)); qqline(resid(model))
```



- Look for: no visible pattern in Residuals vs Fitted; QQ points near line; no extremely large Cook's distances.
- Alternatives when assumptions fail:
  - Nonlinearity: consider adding quadratic terms `I(x^2)` or transforming variables (e.g., `log()`), or use polynomial regression as in Chapter 3 examples.
  - Heteroscedasticity: consider transforming response (log) or use weighted least squares (not in course Rmds; mention as advanced alternative).
  - Non-normal residuals: with large sample sizes t-tests are robust; with small samples consider nonparametric alternatives (Spearman correlation for association questions).

6) Interval estimation and prediction

- Goal: produce CI for mean response and PI for individual predictions.
- Code:

```
predict(model, newdata = data.frame(ADVEXP_X = 4), interval = "confidence")
```

```
##   fit      lwr      upr
## 1 2.7 1.644502 3.755498
```

```
predict(model, newdata = data.frame(ADVEXP_X = 4), interval = "prediction")
```

```
##   fit       lwr      upr
## 1 2.7 0.5028056 4.897194
```

- Report both with units; explain that PI is wider because it includes individual variability.

7) Multiple predictors, interactions, and model comparison

- Goal: extend model when more predictors are relevant (e.g., GASTURBINE example).
- Code pattern:

```r
modelFull = lm(HEATRATE ~ RPM + CPRATIO + RPM*CPRATIO + I(RPM^2) + I(CPRATIO^2),
               data = read.table("../../Chapter4/R-Files/GASTURBINE.txt", header = TRUE))
modelReduced = lm(HEATRATE ~ RPM + CPRATIO + RPM*CPRATIO,
                  data = read.table("../../Chapter4/R-Files/GASTURBINE.txt", header = TRUE))
anova(modelReduced, modelFull)
```

```
## Analysis of Variance Table
##
## Model 1: HEATRATE ~ RPM + CPRATIO + RPM * CPRATIO
## Model 2: HEATRATE ~ RPM + CPRATIO + RPM * CPRATIO + I(RPM^2) + I(CPRATIO^2)
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     63 25310639
## 2     61 19370350  2   5940289 9.3534 0.0002864 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Read: if anova comparison p-value is small, the extra terms improve the model.
- Alternatives: if multicollinearity or complexity is an issue, simplify model or use variable selection techniques (not covered in course Rmds).

8) Report results and interpret in context — what to write on the exam

- State model and sample size.
- Report key estimates: slope (with CI), intercept (if meaningful), residual standard error, R-squared (or adjusted R-squared), and p-values for relevant tests.
- Provide interpretation in plain language and units, e.g., "An additional 100 dollars in advertising (ADVEXP_X = 1 unit) is associated with an estimated increase of 0.8 (thousand dollars) in sales (p = 0.02)."
- Provide caution: if assumptions are violated, mention the diagnostic evidence and alternative approach you would take.

Quick exam checklist (one-page summary)

- Data: `read.table()` and `head()`, `summary()`
- Visualize: `plot()`, `pairs()`
- Association: `cor.test()`
- Fit: `lm()`, `summary()`, `anova()`
- Diagnostics: `plot(model)`, `qqnorm()`
- Intervals: `confint()`, `predict(..., interval = "prediction")`
- Advanced: interactions `x1*x2`, quadratic `I(x^2)`, compare with `anova()`

Appendix: handy snippets

- t-value quantiles: `qt(0.025, df, lower.tail = FALSE)`
- confint: `confint(model, level = 0.95)`
- predict with intervals: `predict(model, newdata = data.frame(...), interval = "prediction")`

Notes: This master guide for Chapter 3 collects the main code patterns used in the course Rmds (ADSALES, CASINO, TIRES, FIREDAM). It intentionally reuses the exact function calls and workflows from those Rmd files.

Quick reference: interpreting common outputs

- `cor.test(x, y)`: returns the correlation estimate (Pearson's r), t-statistic, df, and p-value. Look at `estimate` and `p.value` to decide whether `rho != 0`.
- `confint(model, level = 0.95)`: returns CIs for parameters. If a CI for a slope excludes 0, that suggests evidence of an effect at the stated level.
- `qt(p, df)`: quantiles of the t distribution; used to compute critical values when performing manual tests.

Try it (small checklist before inferential steps)

1. Inspect data: `head()`, `summary()`, `plot()` or `pairs()`
2. Fit model with `lm()` and check `summary()` and `anova()`
3. Visualize residuals: `plot(model)` or `qqnorm(resid(model)); qqline(resid(model))`
4. Report estimates with `coef()` and `confint()` and interpret in context

““