Lecture 4: Normal Linear Models — Notation and R Fitting

MATH3823 Generalised Linear Models

Richard P Mann

MATH3823 Generalised Linear Models

Reading

Course notes: Chapter 2, Sections 2.5–2.7

`www.richardpmann.com/MATH3823`

R Formula Notation

The ~ operator: "is modeled by"

Symbol	Meaning
`y ~ x`	y is modeled by x
`1`	Intercept (included by default)
`E`	Main effect of variable E
`E + F`	Main effects of E and F
`E : F`	Interaction between E and F
`E * F`	Main effects + interaction ($= E + F + E:F$)
`-E`	Remove term E from model
`-1` or `0`	Remove intercept

More Formula Notation

Symbol	Meaning
`E / F`	Nested factors: $E + E : F$
`poly(E, ℓ)`	Polynomial of degree ℓ in E
`I()`	"As is" — use arithmetic literally
`.`	All other variables in data frame

Examples:

- `y ~ x + I(x^2)`: quadratic regression
- `y ~ poly(x, 3)`: cubic polynomial (orthogonal)
- `y ~ . - z`: all variables except `z`

Formula Examples

Birth weight example:

- `weight ~ age`: simple regression on age
- `weight ~ age + sex`: parallel lines model
- `weight ~ age * sex`: separate lines (with interaction)
- `weight ~ age + sex + age:sex`: same as above

More examples:

- `yield ~ variety + poly(rainfall, 2)`: ANCOVA with quadratic term
- `score ~ school/class`: students nested in classes within schools
- `profit ~ investment - 1`: regression through the origin

The `lm()` Function

Basic syntax:

```
my.lm <- lm(formula, data = mydata)
```

Example:

```
# Fit the parallel lines model
model2 <- lm(weight ~ age + sex, data = birthwt)

# View results
summary(model2)
```

Note: R automatically handles:

- Design matrix construction
- Dummy coding for factors
- Parameter estimation

Extracting Information from `lm` Objects

```
# Coefficient estimates
coefficients(my.lm)    # or coef(my.lm)

# Fitted values
fitted.values(my.lm)   # or fitted(my.lm)

# Residuals
residuals(my.lm)       # or resid(my.lm)

# Residual degrees of freedom
df.residual(my.lm)

# Design matrix
model.matrix(my.lm)
```

The summary() Output

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1773.324    820.906  -2.160   0.0423 *
age           120.894     21.423   5.643 1.26e-05 ***
sex           163.039     72.814   2.239   0.0360 *

Residual standard error: 172.2 on 21 degrees of freedom
Multiple R-squared:  0.6387,  Adjusted R-squared:  0.6043
F-statistic: 18.56 on 2 and 21 DF,  p-value: 2.237e-05
```

Key elements:

- **Estimates:** $\hat{\boldsymbol{\beta}}$ values
- **Std. Error:** $\sqrt{\text{diag}(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1})}$
- t **value:** Estimate / Std. Error
- p**-value:** Test of $H_0 : \beta_j = 0$

The `anova()` Function

Single model: Sequential analysis of variance

```
anova(my.lm)
```

Multiple models: Compare nested models

```
anova(model1, model2, model3)
```

Output includes:

- Sum of squares for each term
- Degrees of freedom
- Mean squares
- F-statistics and p-values

Making Predictions

Predict at new covariate values:

```
# Create new data
new_data <- data.frame(age = c(38, 40, 42),
                       sex = c(0, 1, 1))

# Point predictions
predict(model2, newdata = new_data)

# With confidence intervals
predict(model2, newdata = new_data,
        interval = "confidence")

# With prediction intervals
predict(model2, newdata = new_data,
        interval = "prediction")
```

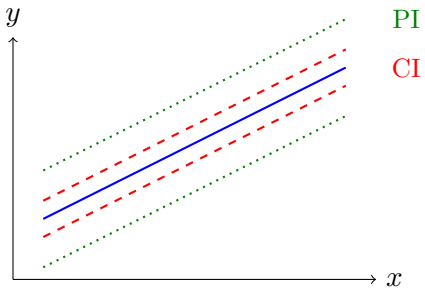
Confidence vs. Prediction Intervals

Confidence interval for $\mathbb{E}[Y \mid \mathbf{x}]$:

- Uncertainty about the *mean* response
- Narrower

Prediction interval for new Y:

- Uncertainty about an *individual* observation
- Wider (includes σ^2)



Model Diagnostics: Residual Plots

Key diagnostic plots:

1. **Residuals vs. fitted:** Check linearity, constant variance
2. **Q-Q plot:** Check normality of residuals
3. **Scale-location:** Check homoscedasticity
4. **Residuals vs. leverage:** Identify influential points

```
# All four plots
par(mfrow = c(2, 2))
plot(my.lm)
```

What to Look For

Residuals vs. Fitted:

- Good: Random scatter around 0
- Bad: Curved pattern $\Rightarrow$ nonlinearity
- Bad: Funnel shape $\Rightarrow$ heteroscedasticity

Q-Q Plot:

- Good: Points follow diagonal line
- Bad: S-shape $\Rightarrow$ heavy/light tails
- Bad: Curve $\Rightarrow$ skewness

If assumptions violated:

- Transform response (log, sqrt)
- Add polynomial terms
- Use a different model (e.g., GLM)

Declaring Factors in R

Problem: R may treat numeric codes as continuous.

```
# Sex coded as 0/1
sex <- c(0, 0, 1, 1, 0, 1)

# Wrong: treated as quantitative
lm(y ~ sex)  # Only one coefficient for sex

# Right: declare as factor
sex <- as.factor(sex)
lm(y ~ sex)  # Now categorical
```

Best practice: Always use `factor()` or meaningful labels:

```
sex <- factor(c("F", "F", "M", "M", "F", "M"))
```

Summary

Key points:

- R formula notation provides concise model specification
- `lm()` fits normal linear models
- `summary()` gives coefficient estimates and tests
- `anova()` compares nested models
- `predict()` generates predictions with intervals
- Diagnostic plots check model assumptions
- Declare categorical variables as factors

Next lecture: Introduction to GLM theory — the exponential family.