Regression Methods in R

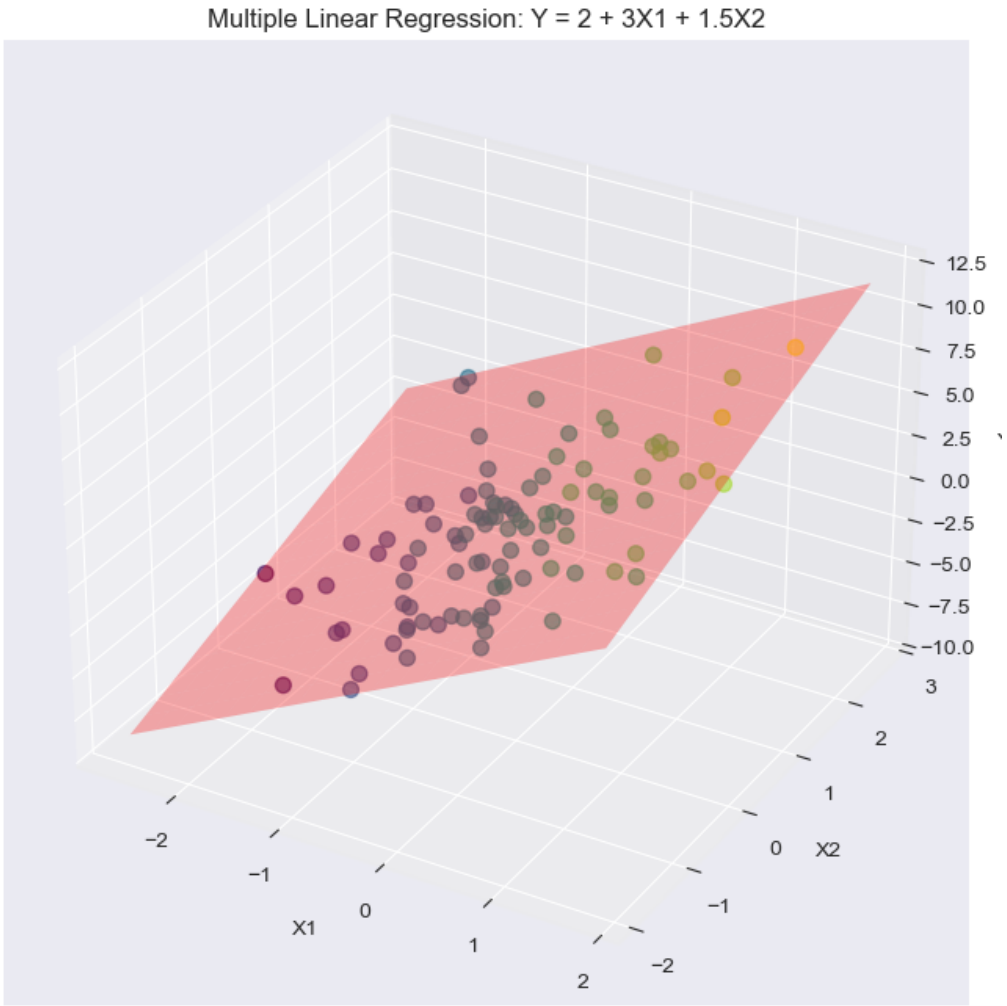
Multiple Linear, Logistic, and Survival Analysis

Statistical Data Analysis Course

MSc Level

Multiple Linear Regression - Introduction

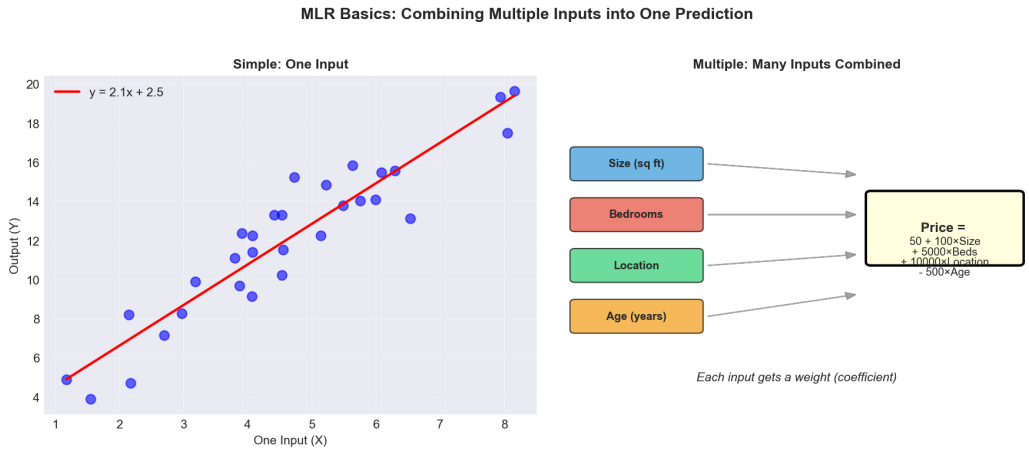
- **Goal**: Predict continuous outcome from multiple predictors
- **Example**: House price = base + (size effect) + (location effect) + (age effect)
- **Model**: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$



Key Questions: How do predictors relate? Which are significant? How well does model predict?

MLR - The Basics

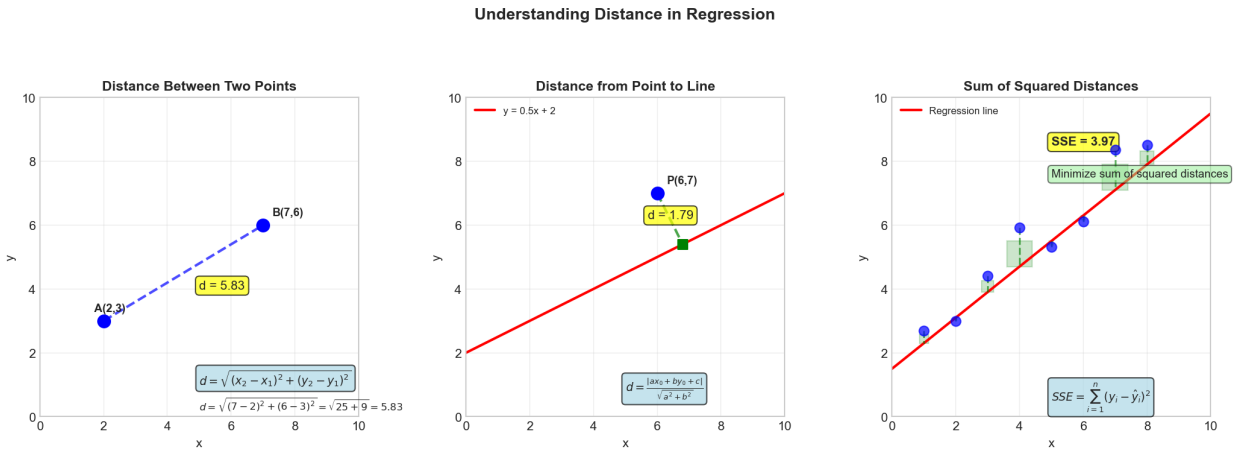
Core Idea: Combine Multiple Inputs to Predict One Output



Example: House (2000 sqft, 3 beds, good location, 10 yrs): Price = 50 + 100(2000) + 5000(3) + 10000(1) - 500(10) = $220,050
Key Point: Each feature contributes independently to prediction

Distance Basics - Foundation of Regression

How We Measure "Closeness" to Find the Best Line



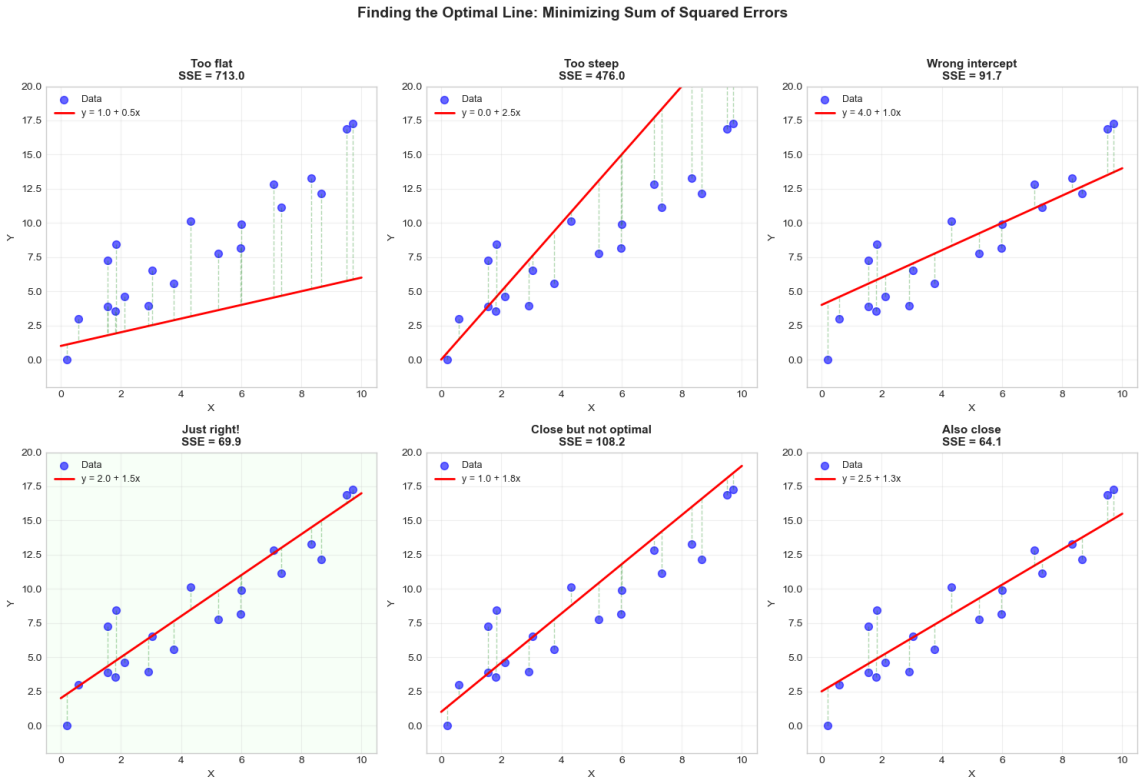
Three Key Distance Concepts:

1. **Point to Point**: Euclidean distance between observations
2. **Point to Line**: Perpendicular distance (single residual)
3. **Sum of Squares**: Total error to minimize (SSE/RSS)

Regression Goal: Find the line that minimizes the sum of squared vertical distances

Finding the Optimal Line

How do we find the "best" line through the data?

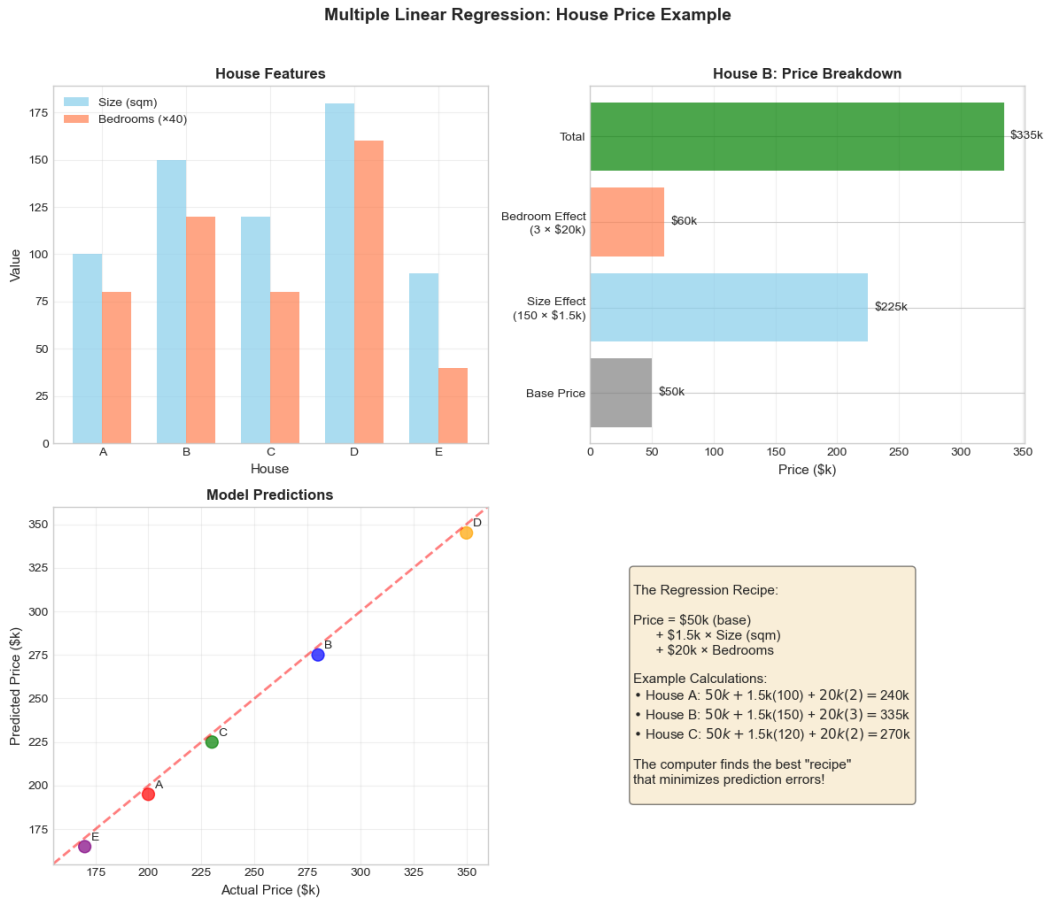


Least Squares Method: Try lines, calculate squared errors, sum them. Best line = smallest sum.
Why squares? Penalizes large errors more; mathematically convenient

MLR - How It Works (Intuitive Math)

Example: Predicting House Price

$$\text{Price} = \text{Base (\$50k)} + \$1{,}500/\text{sqm} + \$20{,}000/\text{bedroom}$$



House B check: $\$50\text{k} + (150 \times \$1.5\text{k}) + (3 \times \$20\text{k}) = \$275{,}000$
Key: Computer finds coefficients that minimize errors across all data

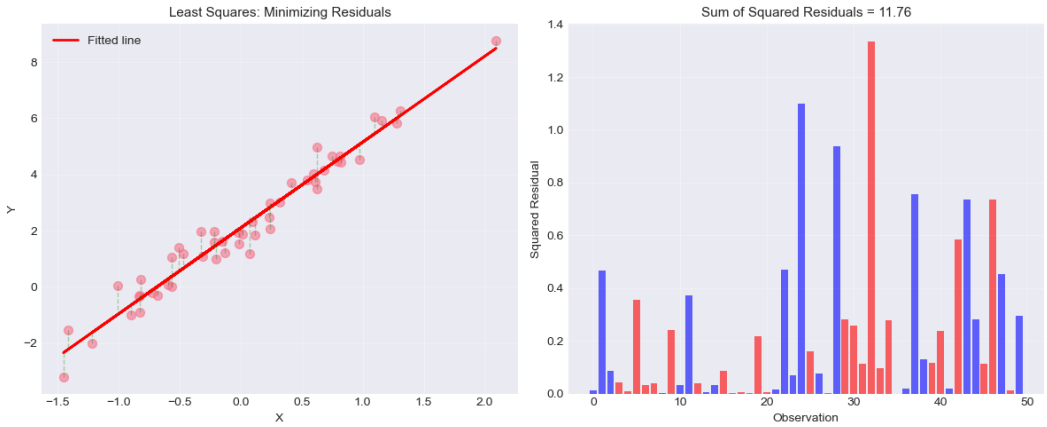
MLR - Theory and Estimation

Mathematical Foundation:

- Model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$
- Assumptions:
  1. Linearity: $E(Y|X) = X\beta$
  2. Independence: observations independent
  3. Homoscedasticity: $Var(\epsilon_i) = \sigma^2$
  4. Normality: $\epsilon_i \sim N(0, \sigma^2)$

Least Squares Estimation:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

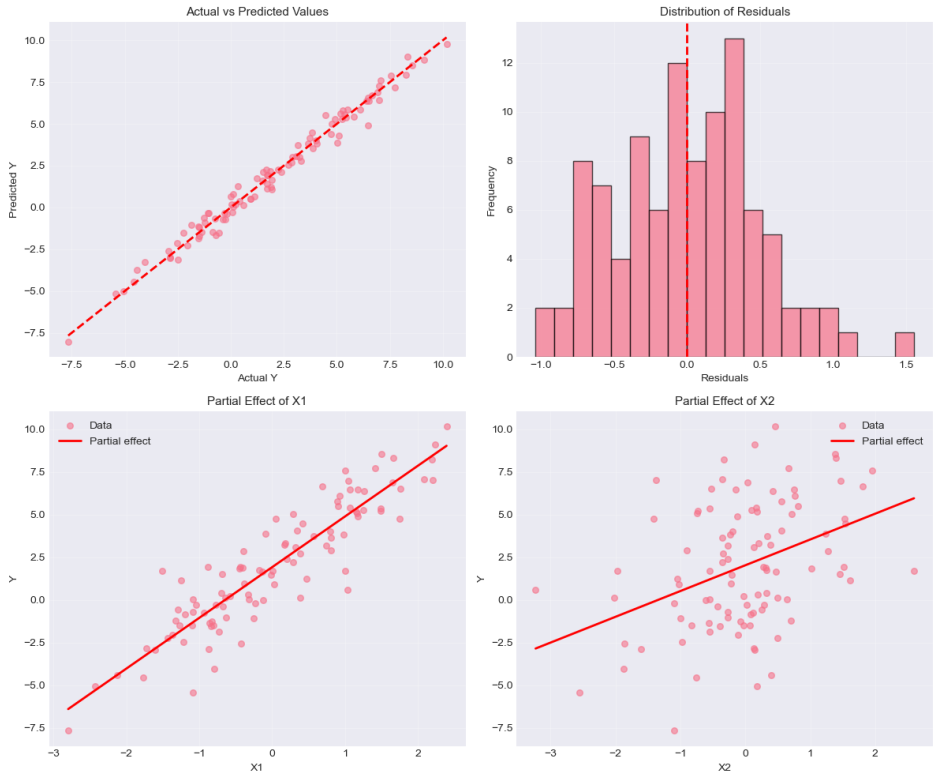


MLR - R Implementation

```r
# Generate sample data
set.seed(123)
n <- 100
x1 <- rnorm(n)
x2 <- rnorm(n)
y <- 2 + 3*x1 + 1.5*x2 + rnorm(n, sd=0.5)

# Fit multiple linear regression
model <- lm(y ~ x1 + x2)
summary(model)

# Predictions
new_data <- data.frame(x1=c(0, 1), x2=c(0, 1))
predict(model, new_data, interval="confidence")
```



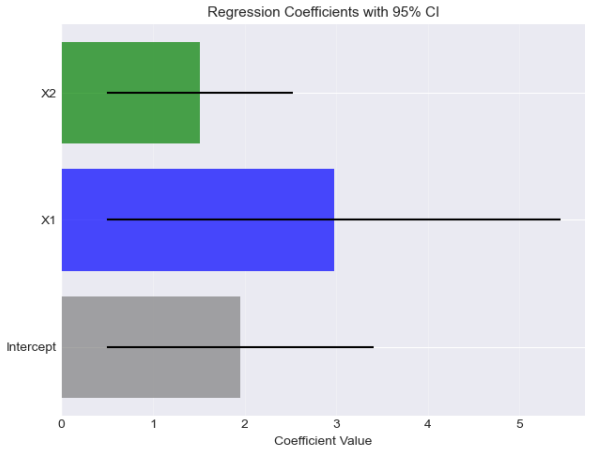
MLR - Interpretation

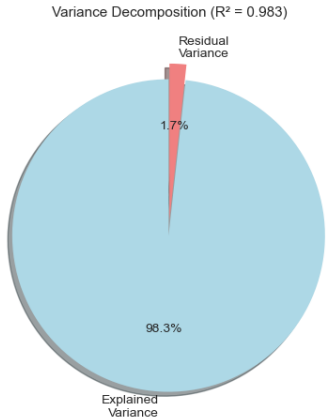
Coefficient Interpretation:

- β_j: Change in Y for 1-unit increase in X_j, holding others constant
- Standard Error: Uncertainty in coefficient estimate
- t-statistic: $t = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$
- p-value: Probability of observing effect if $\beta_j = 0$

Model Performance:

- $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$: Proportion of variance explained
- Adjusted R^2: Penalizes for number of predictors
- F-statistic: Overall model significance



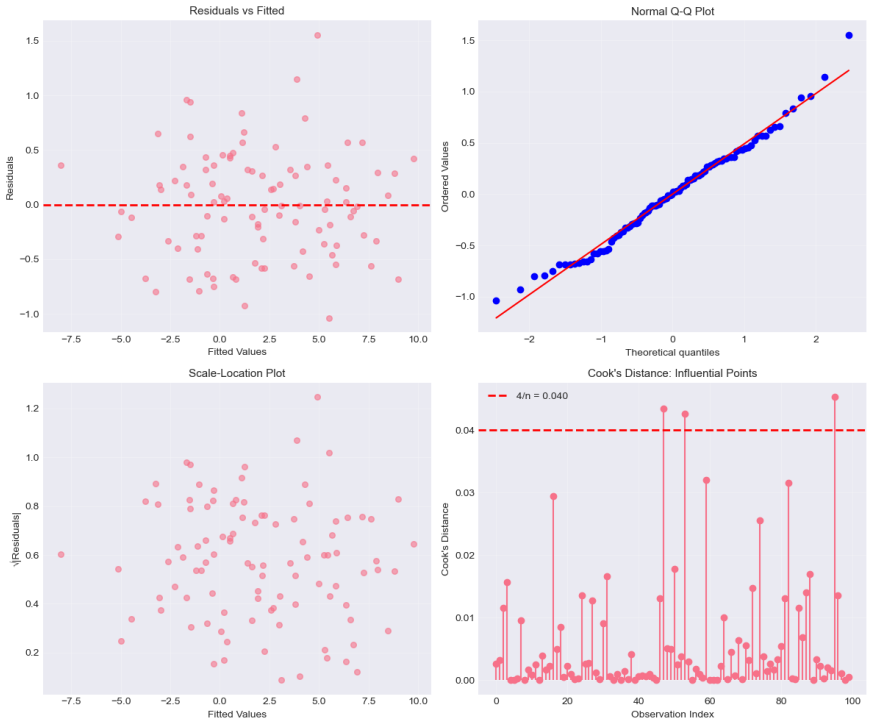


MLR - Diagnostics

Checking Assumptions:

```
# Diagnostic plots
par(mfrow=c(2,2))
plot(model)

# Additional diagnostics
library(car)
vif(model)  # Variance Inflation Factor
ncvTest(model)  # Heteroscedasticity test
```



Key Checks:

- Residuals vs Fitted: linearity, homoscedasticity
- Q-Q plot: normality of residuals
- Scale-Location: spread of residuals
- Residuals vs Leverage: influential points

MLR - Advanced Topics

Multicollinearity:

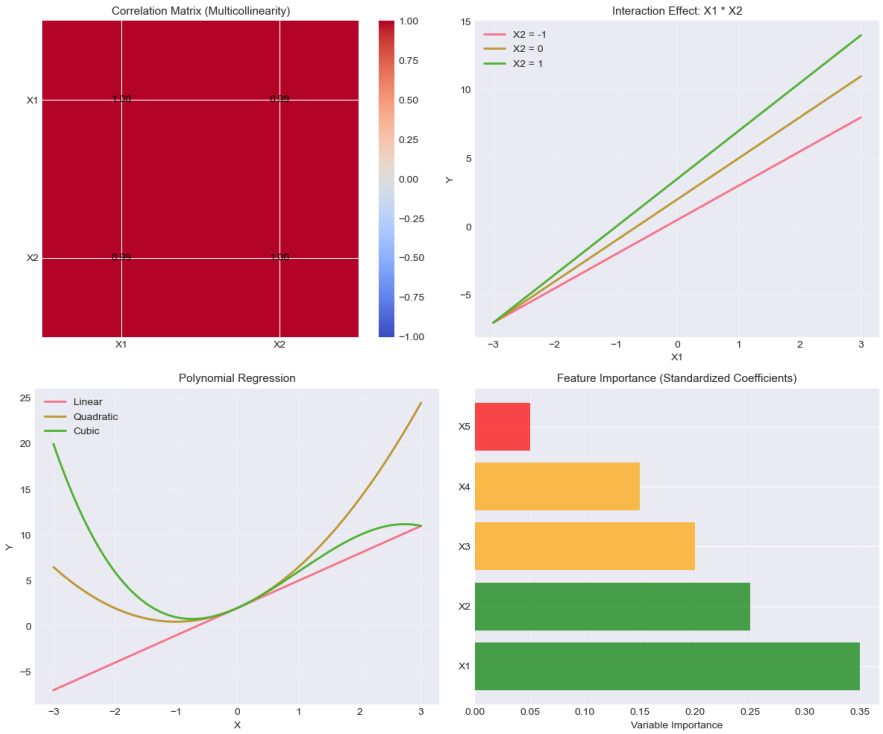
- VIF > 5: potential problem
- Solution: Remove/combine correlated predictors

Interaction Terms:

```
model_int <- lm(y ~ x1 * x2)
```

Polynomial Regression:

```
model_poly <- lm(y ~ poly(x1, 2) + x2)
```



Variable Selection:

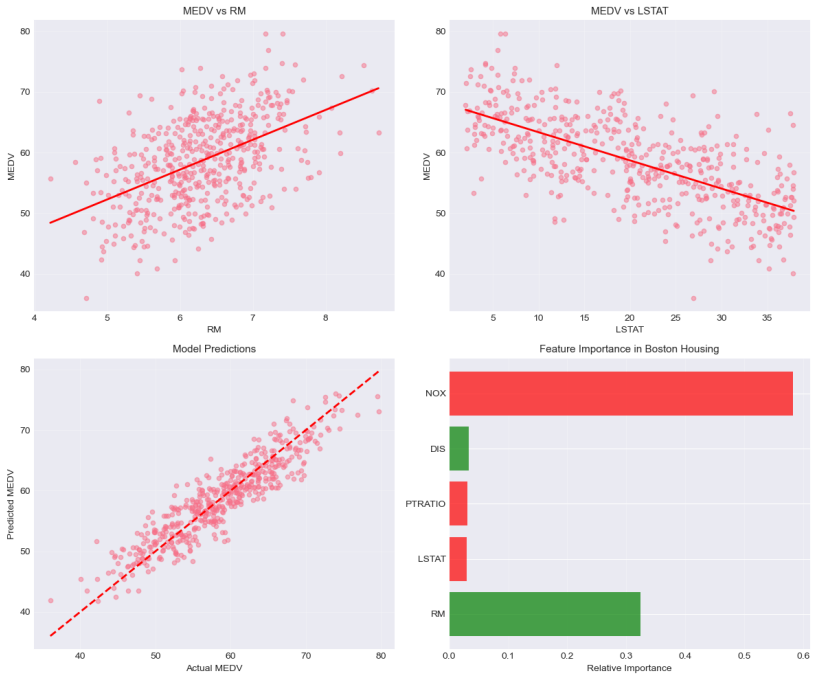
- Forward/Backward/Stepwise selection
- LASSO/Ridge regression for many predictors

MLR - Case Study

Boston Housing Data Analysis:

```
library(MASS)
data(Boston)
# Predict median home value
model_boston <- lm(medv ~ rm + lstat + ptratio +
                   dis + nox, data=Boston)
summary(model_boston)

# Cross-validation
library(caret)
cv_results <- train(medv ~ ., data=Boston,
                    method="lm",
                    trControl=trainControl(method="cv",
                                           number=10))
```



Key Findings:

- Room number strongest predictor (+)
- Lower status population strong predictor (-)
- Model $R^2 = 0.74$

Logistic Regression - Introduction

Binary Classification Problem: ("Will it happen or not?")

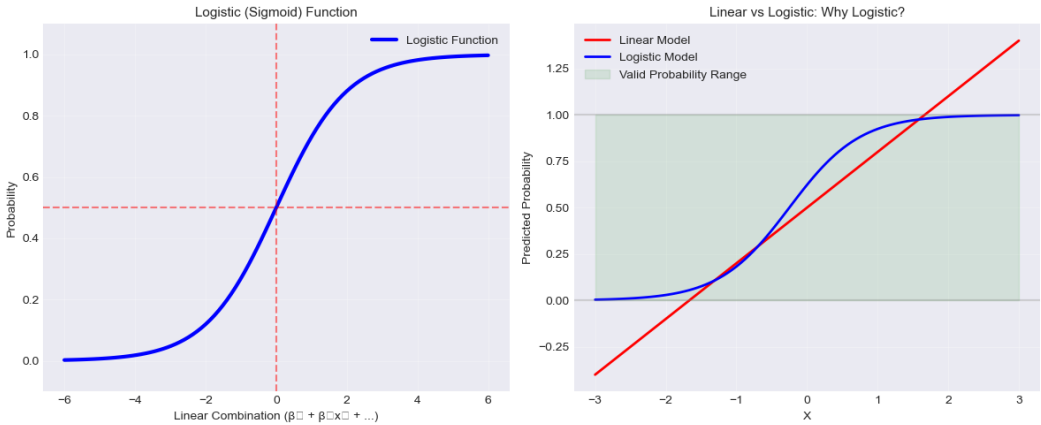
- Outcome: Binary (0/1, Yes/No, Success/Failure)
- Goal: Predict *probability* of success (between 0 and 1)
- Examples: Disease diagnosis, customer churn, loan default

Why not Linear Regression?

- Linear model can predict -0.3 or 1.5 (impossible probabilities!)
- We need outputs constrained to [0,1]

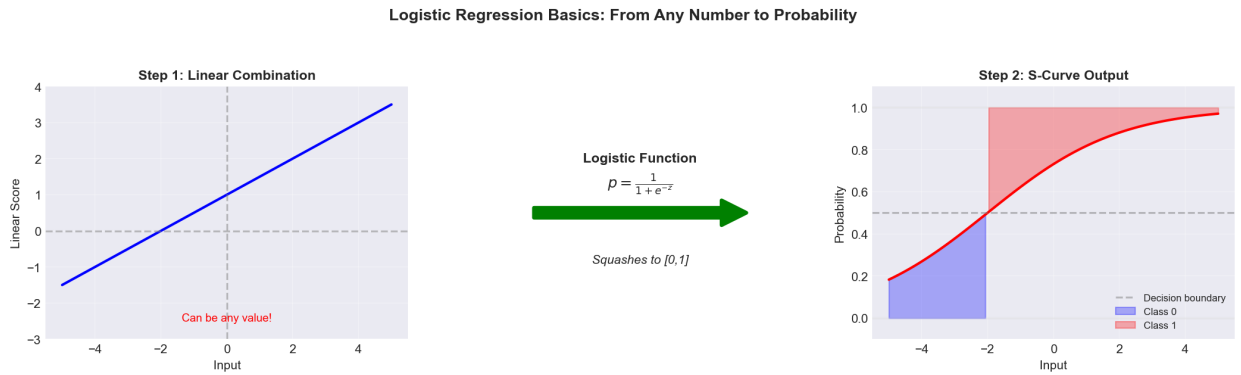
Solution: Logistic Function

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}}$$



Logistic Regression - The Basics

Core Problem: Need Probability (0 to 1), Not Any Number



Two-Step Process:

1. Calculate linear score (can be any value): $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$
2. Transform to probability using S-curve: $p = \frac{1}{1+e^{-z}}$

Example: Credit score=700, income=\$60k $\rightarrow$ z=2.5 $\rightarrow$ p=0.92 (92% chance of loan approval)

Logistic Regression - How It Works (Intuitive Math)

Example: Predicting Loan Default

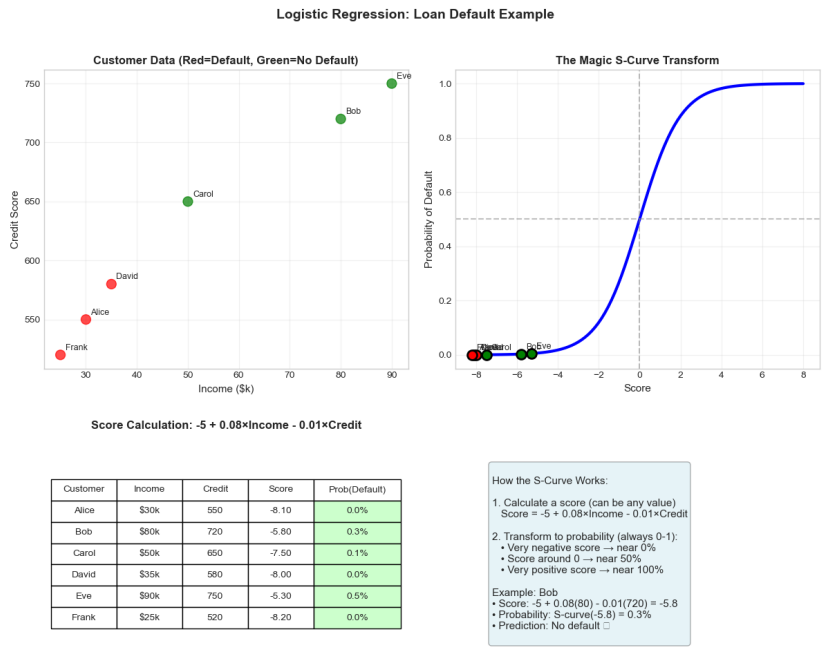
Where do these numbers come from?

- We have historical data: 1000 past loan applications
- We know who defaulted (200 people) and who didn't (800 people)
- Using Maximum Likelihood, we find coefficients that best separate the groups

Customer	Income	Credit Score	Default?
Alice	\$30k	550	Yes (1)
Bob	\$80k	720	No (0)
Carol	\$50k	650	No (0)
David	\$35k	580	Yes (1)

The Fitted Model:

- Intercept: -5 (baseline log-odds when income=0, credit=0)
- Income coefficient: 0.08 (higher income $\rightarrow$ lower default risk)
- Credit coefficient: -0.01 (higher credit $\rightarrow$ lower default risk)



Customer	Income	Credit	Score	Prob(Default)
Alice	\$30k	550	-8.10	0.0%
Bob	\$80k	720	-5.80	0.3%
Carol	\$50k	650	-7.50	0.1%
David	\$35k	580	-8.00	0.0%
Eve	\$90k	750	-5.30	0.5%
Frank	\$25k	520	-8.20	0.0%

Logistic Regression - Theory with Numbers

Maximum Likelihood Example:
Consider 5 customers with their outcomes:

Customer	Score	Default
1	-2.5	No (0)
2	-1.0	No (0)
3	0.5	Yes (1)
4	1.5	Yes (1)
5	2.0	Yes (1)

Calculate Likelihood:

- Customer 1: $P(Y = 0) = 1 - \frac{1}{1+e^{2.5}} = 0.924$
- Customer 2: $P(Y = 0) = 1 - \frac{1}{1+e^{1.0}} = 0.731$
- Customer 3: $P(Y = 1) = \frac{1}{1+e^{-0.5}} = 0.622$
- Customer 4: $P(Y = 1) = \frac{1}{1+e^{-1.5}} = 0.818$
- Customer 5: $P(Y = 1) = \frac{1}{1+e^{-2.0}} = 0.881$

Total Likelihood: $L = 0.924 \times 0.731 \times 0.622 \times 0.818 \times 0.881 = 0.303$
Log-Likelihood: $\ell = \log(0.303) = -1.194$
The algorithm finds coefficients that MAXIMIZE this likelihood!

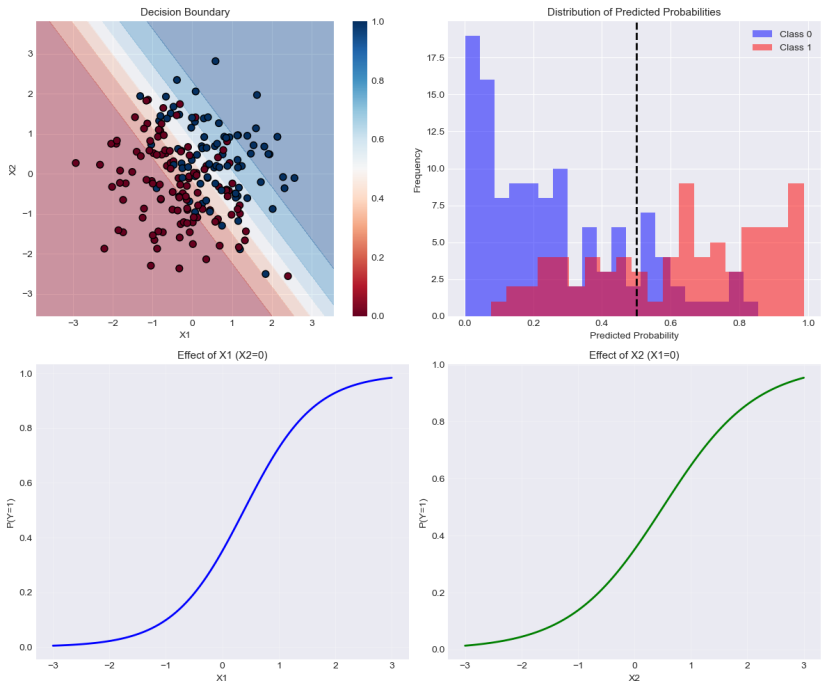
Logistic Regression - R Implementation

```r
# Generate binary outcome data
set.seed(456)
n <- 200
x1 <- rnorm(n)
x2 <- rnorm(n)
logit_p <- -1 + 2*x1 + 1.5*x2
p <- 1/(1 + exp(-logit_p))
y <- rbinom(n, 1, p)

# Fit logistic regression
log_model <- glm(y ~ x1 + x2, family=binomial)
summary(log_model)

# Predictions
pred_prob <- predict(log_model, type="response")
pred_class <- ifelse(pred_prob > 0.5, 1, 0)

# Odds ratios
exp(coef(log_model))
exp(confint(log_model))
```



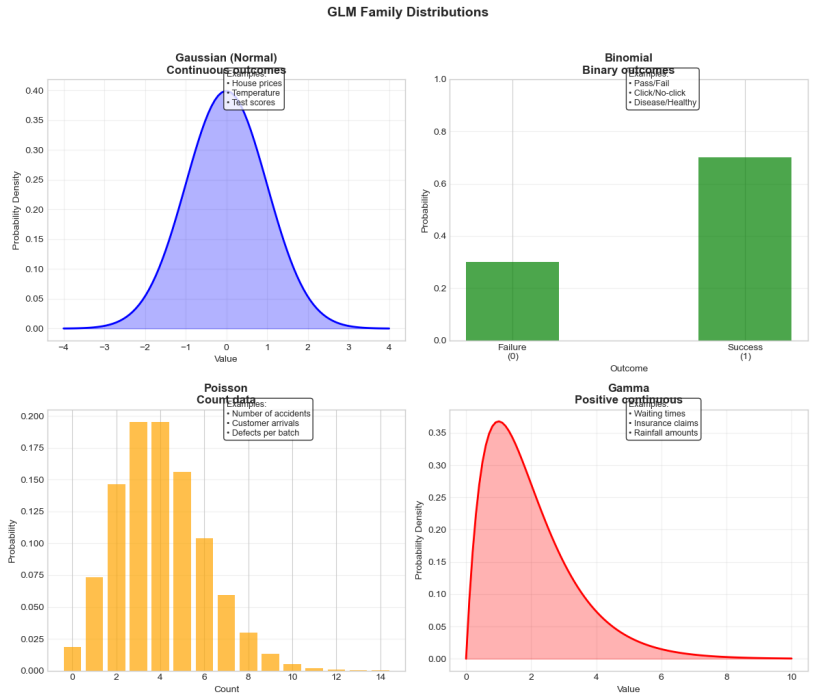
GLM Family Parameters

Generalized Linear Models - The "family" Parameter:

Family	Use Case	Link Function	R Code
gaussian	Continuous outcome (normal errors)	identity $\mu = X\beta$	`family=gaussian` (default for lm)
binomial	Binary outcome (0/1, Yes/No)	logit $\log(\frac{p}{1-p}) = X\beta$	`family=binomial`
poisson	Count data (0, 1, 2, ...)	log $\log(\lambda) = X\beta$	`family=poisson`
Gamma	Positive continuous (waiting times)	inverse $\frac{1}{\mu} = X\beta$	`family=Gamma`

Examples:

- **Gaussian**: House prices, temperature, height
- **Binomial**: Pass/fail, disease/healthy, click/no-click
- **Poisson**: Number of accidents, customer arrivals, defects
- **Gamma**: Insurance claims, rainfall amounts, service times



Logistic Regression - Interpretation

Coefficient Interpretation:

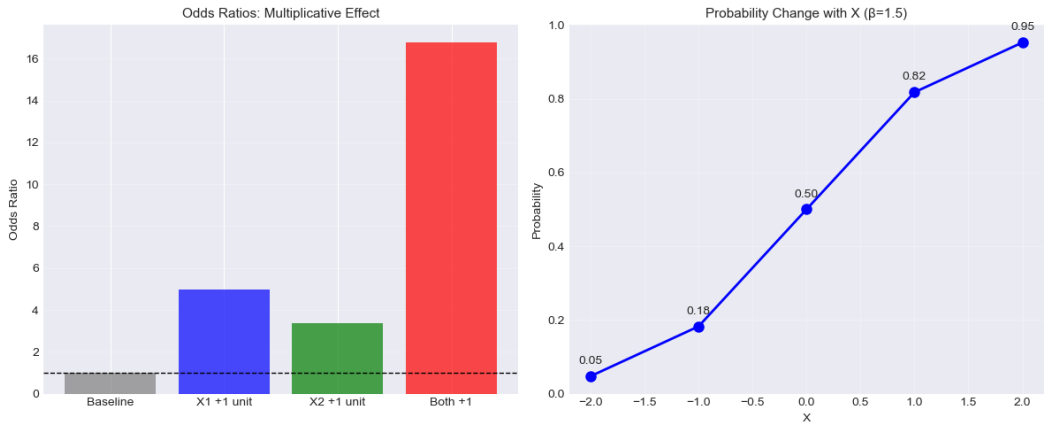
- β_j: Change in log-odds for 1-unit increase in x_j
- e^{β_j}: Odds ratio - multiplicative change in odds

Interpretation Example:

- Coefficient $\beta_1 = 0.693$
- Odds Ratio: $e^{0.693} = 2.0$
- Interpretation: 1-unit increase in x_1 doubles the odds of success

Probability Calculation:

- For given x: $p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\ldots)}}$
- Example: If logit $= 1.5$, then $p = \frac{1}{1+e^{-1.5}} = 0.82$



Confidence Intervals:

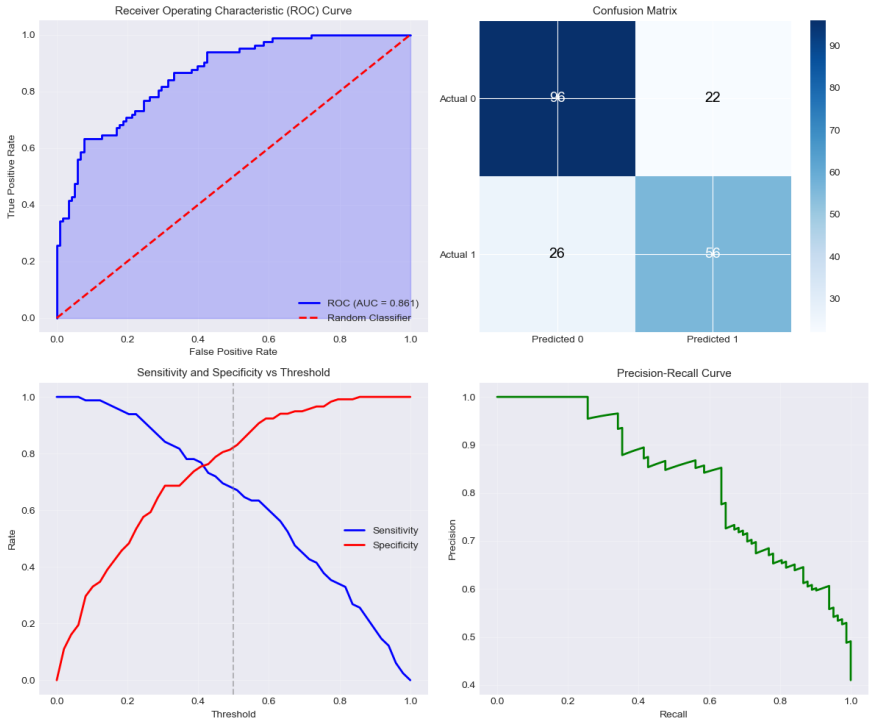
- For coefficients: Wald CI
- For odds ratios: Transform CI of coefficients

Logistic Regression - Model Evaluation

Classification Metrics:

```
# Confusion Matrix
table(Predicted=pred_class, Actual=y)

# ROC Curve and AUC
library(pROC)
roc_obj <- roc(y, pred_prob)
plot(roc_obj, main="ROC Curve")
auc(roc_obj)
```



Performance Measures:

- Accuracy: $(TP + TN)/Total$
- Sensitivity (Recall): $TP/(TP + FN)$
- Specificity: $TN/(TN + FP)$
- AUC: Area Under ROC Curve (0.5-1.0)

Model Evaluation - Deep Dive

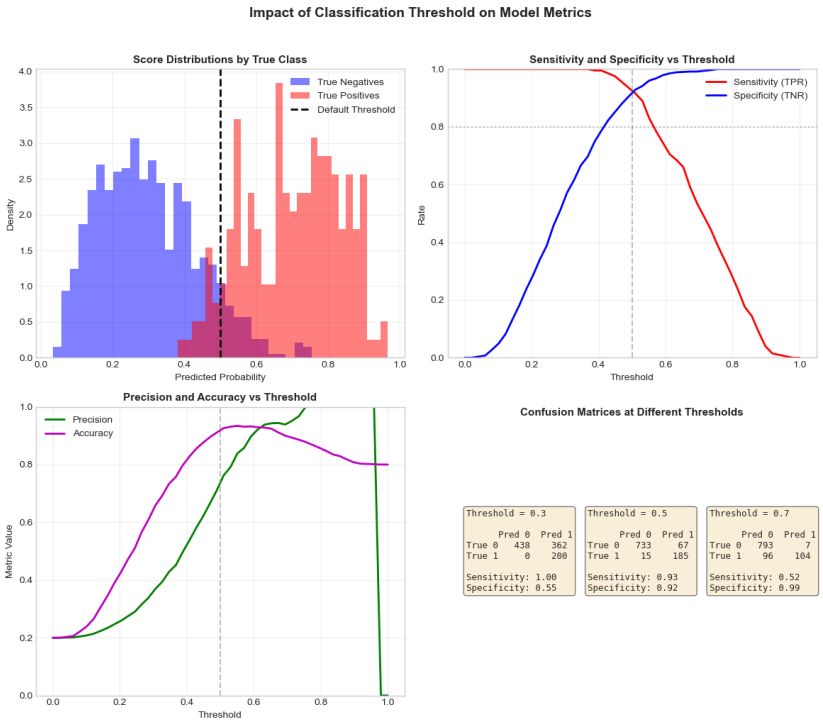
Example: 100 Patients, 20 Have Disease

Predicted	Actual		Total
	Disease	Healthy	
Positive	15 (TP)	10 (FP)	25
Negative	5 (FN)	70 (TN)	75
Total	20	80	100

Metrics Calculation:

- **Accuracy**: $(15 + 70)/100 = 85\%$ - Overall correct
- **Sensitivity**: $15/20 = 75\%$ - Detected 75% of diseased
- **Specificity**: $70/80 = 87.5\%$ - Correctly identified 87.5% healthy
- **Precision**: $15/25 = 60\%$ - 60% of positive predictions correct
- **NPV**: $70/75 = 93.3\%$ - 93.3% of negative predictions correct

Threshold Impact:



Moving threshold from 0.5 to 0.3: More positives → Higher sensitivity, Lower specificity

Logistic Regression - Extensions

Multinomial Logistic Regression:

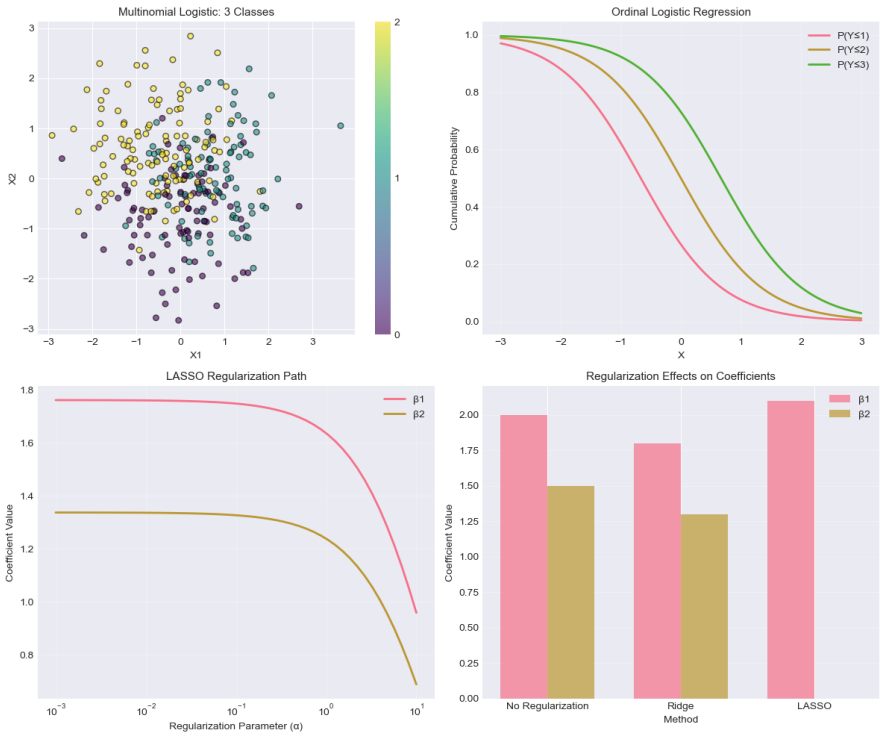
```r
library(nnet)
# For 3+ categories
multinom_model <- multinom(category ~ x1 + x2)
```

Ordinal Logistic Regression:

```r
library(MASS)
# For ordered categories
ordinal_model <- polr(ordered_y ~ x1 + x2)
```

Regularized Logistic Regression:

```r
library(glmnet)
# LASSO/Ridge for variable selection
cv_model <- cv.glmnet(X, y, family="binomial")
```

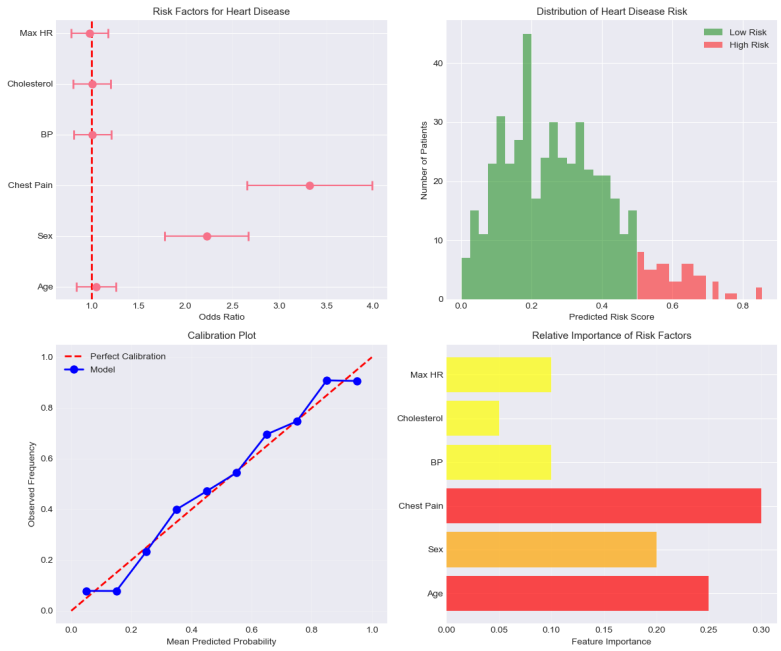


Logistic Regression - Case Study

Heart Disease Prediction:

```
# Load heart disease data
heart <- read.csv("heart.csv")
# Predict presence of heart disease
heart_model <- glm(target ~ age + sex + cp +
                   trestbps + chol + thalach,
                   data=heart, family=binomial)

# Model evaluation
library(caret)
ctrl <- trainControl(method="cv", number=10,
                     summaryFunction=twoClassSummary,
                     classProbs=TRUE)
cv_model <- train(target ~ ., data=heart,
                  method="glm", family="binomial",
                  trControl=ctrl, metric="ROC")
```



Results:

- Chest pain type strongest predictor
- Model AUC = 0.88
- 85% cross-validated accuracy

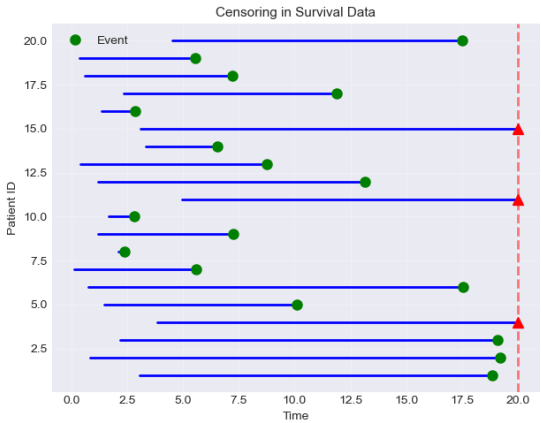
Survival Analysis - Introduction

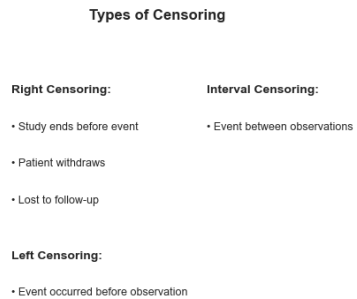
Time-to-Event Data: ("How long until something happens?")

- Outcome: Time until event occurs (not just "did it happen?")
- Event: Death, disease recurrence, machine failure, customer churn
- Key feature: **Censoring** - some subjects haven't had the event yet!

Types of Censoring:

- Right censoring: Event not observed by study end
- Left censoring: Event occurred before observation started
- Interval censoring: Event occurred between observations



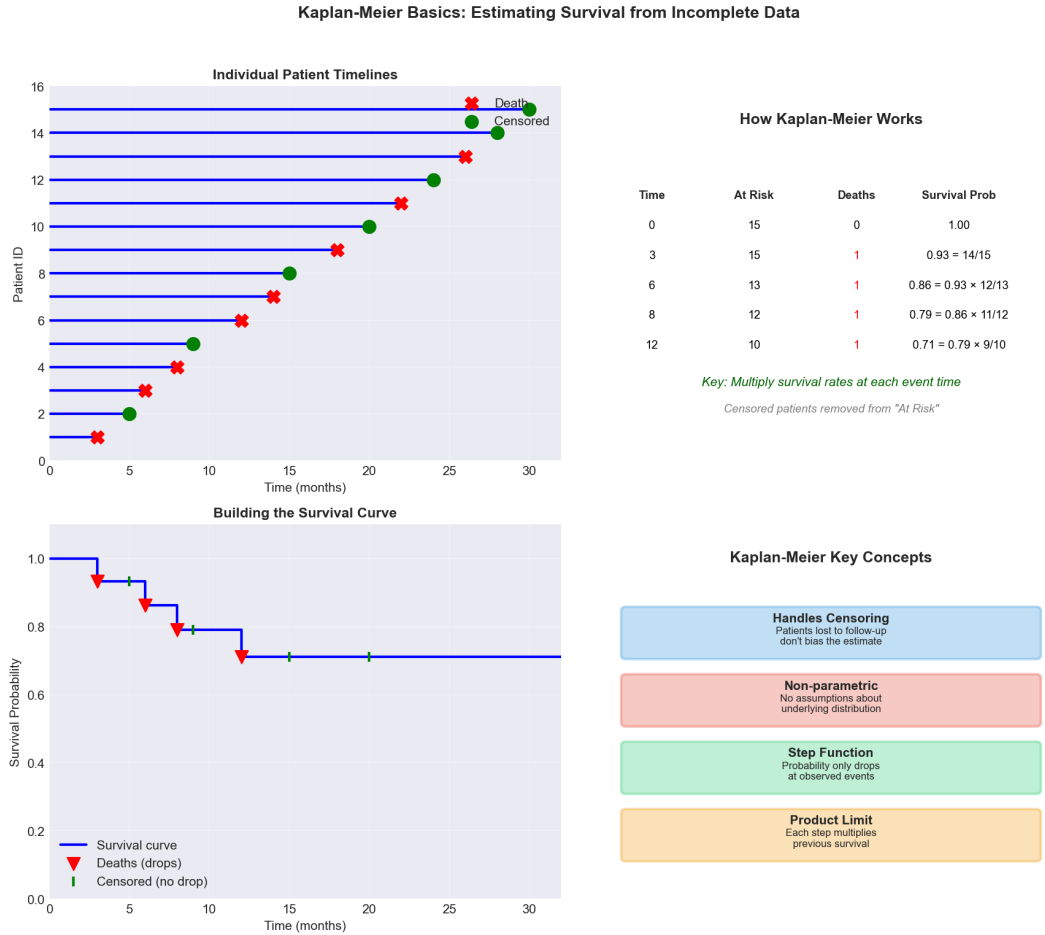


Key Functions:

- Survival function: $S(t) = P(T > t)$
- Hazard function: $h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t+\Delta t | T \geq t)}{\Delta t}$

Kaplan-Meier - The Basics

Core Challenge: How to Estimate Survival When Some Patients Are Still Alive?



The Kaplan-Meier Solution:

- At each death time: Calculate fraction surviving = (survivors / at risk)
- Multiply all fractions together for cumulative survival
- Censored patients: Remove from "at risk" but don't count as deaths
- Result: Unbiased survival estimate despite incomplete data

Survival Analysis - How It Works (Intuitive Math)

Example: Patient Survival After Treatment

10 Patients Started Treatment:

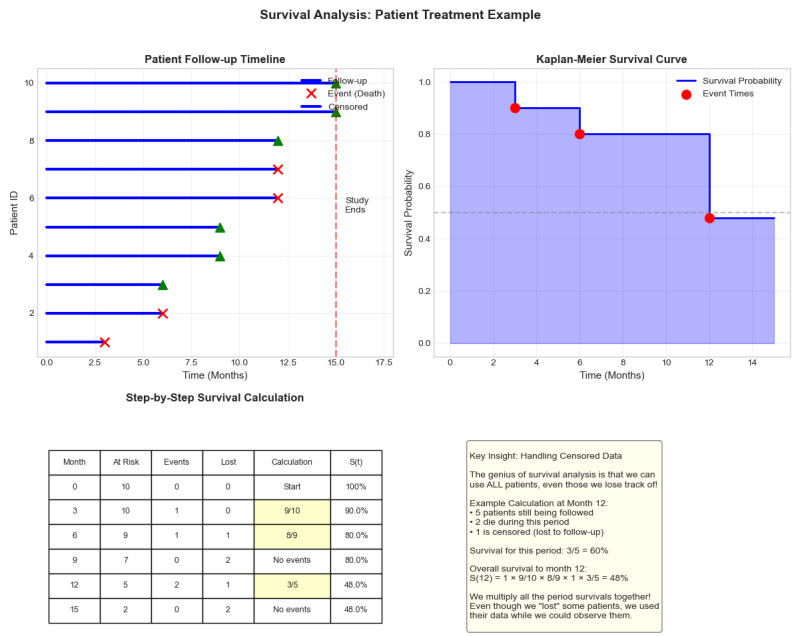
Month	At Risk	Events	Lost/Censored
0	10	0	0
3	10	1	0
6	9	1	1
9	7	0	2
12	5	2	1
15	2	0	2 (study ends)

Calculating Survival Probability:

- Month 0: 100% alive
- Month 3: 1 died out of 10 $\rightarrow$ 9/10 = 90% survive this period
- Month 6: 1 died out of 9 $\rightarrow$ 8/9 survive this period
- Month 12: 2 died out of 5 $\rightarrow$ 3/5 survive this period

Overall Survival: Multiply the probabilities!

$$S(12) = 1 \times \frac{9}{10} \times \frac{8}{9} \times 1 \times \frac{3}{5} = 0.48 \text{ (48\% survive to 12 months)}$$



Month	At Risk	Events	Lost	Calculation	S(t)
0	10	0	0	Start	100%
3	10	1	0	9/10	90.0%
6	9	1	1	8/9	80.0%
9	7	0	2	No events	80.0%
12	5	2	1	3/5	48.0%
15	2	0	2	No events	48.0%

Key Insight: Handling Censored Data

The genius of survival analysis is that we can use ALL patients, even those we lose track of!

Example Calculation at Month 12:
- 5 patients still being followed
- 2 die during this period
- 1 is censored (lost to follow-up)

Survival for this period: 3/5 = 60%

Overall survival to month 12:
S(12) = 1 × 9/10 × 8/9 × 1 × 3/5 = 48%

We multiply all the period survivals together! Even though we "lost" some patients, we used their data while we could observe them.

Key Insight: We can estimate survival even when some patients are "lost" (censored) - we just use them while we can see them!

Kaplan-Meier Estimator

Non-parametric Survival Estimation:

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

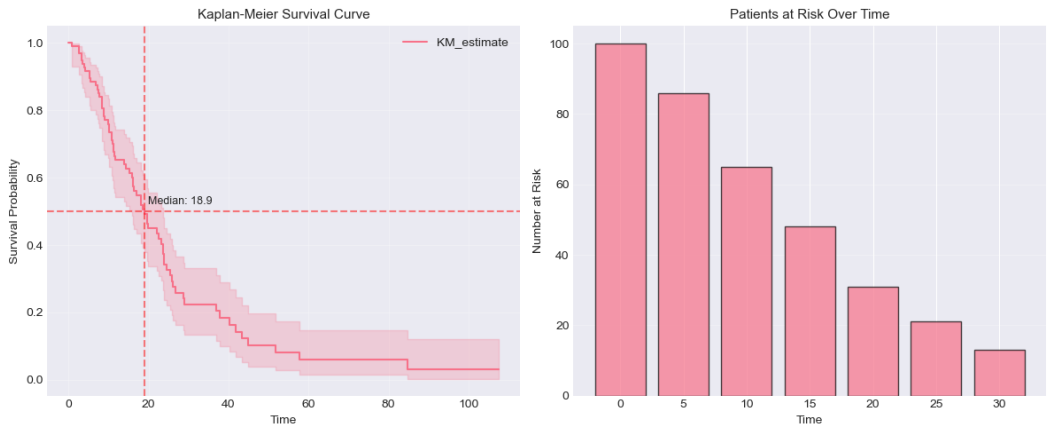
where d_i = events at time t_i, n_i = at risk at time t_i

```r
library(survival)
library(survminer)

# Create survival object
surv_obj <- Surv(time, status)

# Kaplan-Meier estimate
km_fit <- survfit(surv_obj ~ 1)
summary(km_fit)

# Plot survival curve
ggsurvplot(km_fit,
           conf.int = TRUE,
           risk.table = TRUE,
           palette = "Dark2")
```



Log-Rank Test

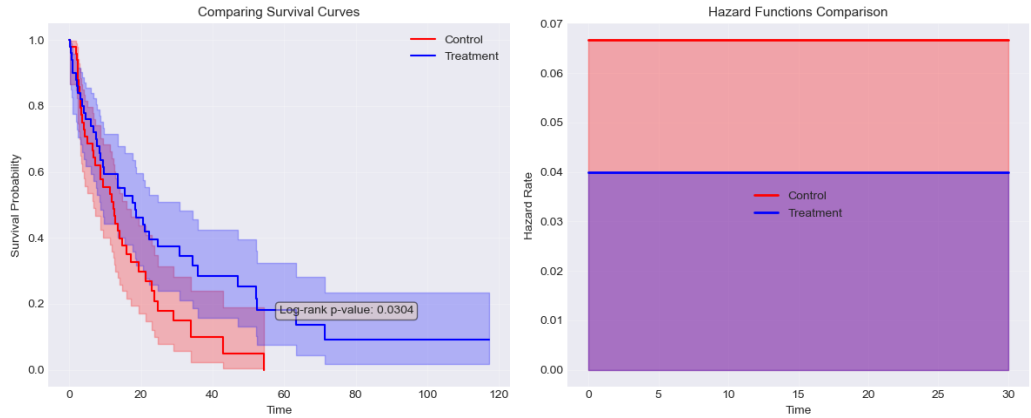
Comparing Survival Curves:

- Null hypothesis: No difference between groups
- Test statistic: Weighted sum of observed-expected events

```
# Compare survival between treatment groups
km_group <- survfit(Surv(time, status) ~ treatment)

# Log-rank test
survdiff(Surv(time, status) ~ treatment)

# Visualization
ggsurvplot(km_group,
           pval = TRUE,
           conf.int = TRUE,
           risk.table = TRUE,
           legend.labs = c("Control", "Treatment"))
```



Interpretation:

- p-value < 0.05: Significant difference
- Median survival times by group
- Confidence intervals for survival probabilities

Cox Proportional Hazards Model

Semi-parametric Regression: ("Regression for survival times")

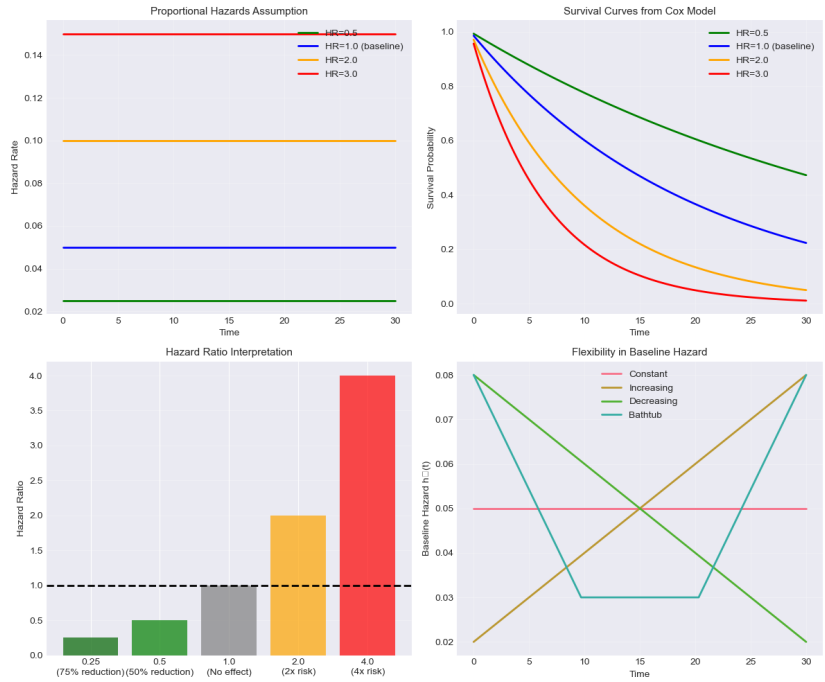
$$h(t|x) = h_0(t)\exp(\beta_1 x_1 + ... + \beta_p x_p)$$

Key Features:

- $h_0(t)$: Baseline hazard (we don't specify this - just compare to it!)
- $\exp(\beta_j)$: Hazard ratio (HR) - how much does factor j change the risk?
- Example: HR $= 2$ means "twice the risk of event"

Interpretation (like odds ratios):

- HR > 1: Increased hazard (worse survival)
- HR < 1: Decreased hazard (better survival)
- HR $= 1$: No effect



Assumptions:

- Proportional hazards
- Log-linearity
- Independence of observations

Cox Model - R Implementation

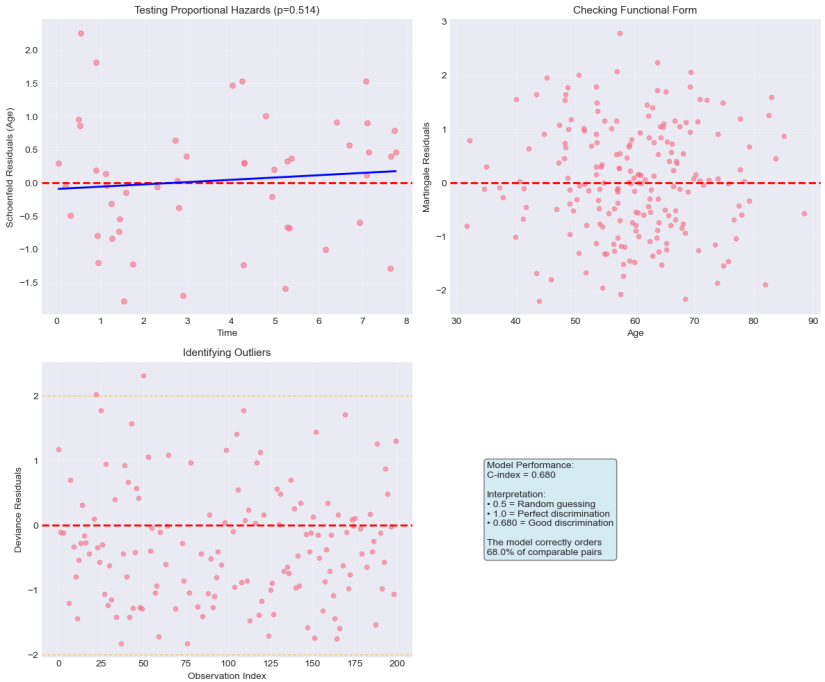
```r
# Fit Cox model
cox_model <- coxph(Surv(time, status) ~ age + sex +
                   treatment + stage, data=survival_data)
summary(cox_model)

# Hazard ratios with CI
exp(cbind(HR = coef(cox_model),
          confint(cox_model)))

# Test proportional hazards assumption
cox.zph(cox_model)

# Predicted survival for new patients
new_patient <- data.frame(age=60, sex="M",
                          treatment=1, stage=2)
surv_pred <- survfit(cox_model, newdata=new_patient)
plot(surv_pred)

# Model diagnostics
ggcoxdiagnostics(cox_model, type = "deviance")
```



Survival Analysis - Case Study

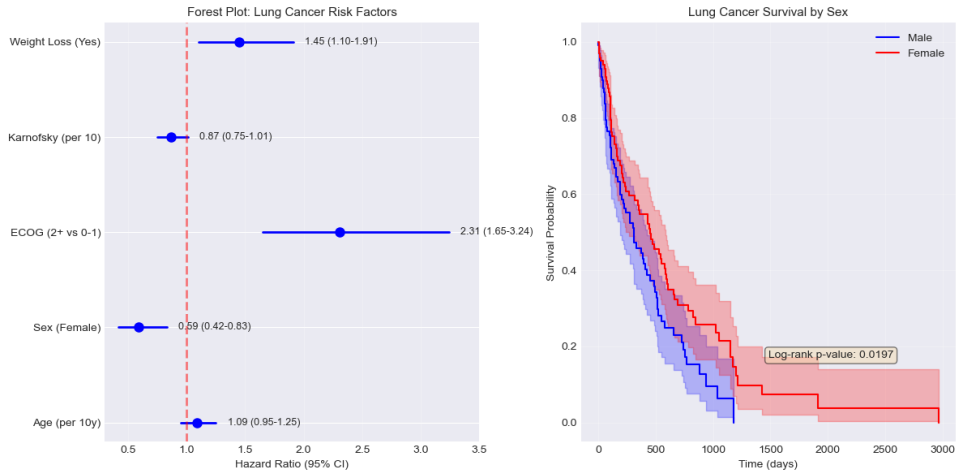
Lung Cancer Survival Analysis:

```
# Load lung cancer data
data(lung)
lung$status <- lung$status - 1  # 0=censored, 1=dead

# Full analysis
cox_lung <- coxph(Surv(time, status) ~ age + sex +
                  ph.ecog + ph.karno + pat.karno +
                  meal.cal + wt.loss, data=lung)

# Stratified K-M curves
km_sex <- survfit(Surv(time, status) ~ sex, data=lung)
ggsurvplot(km_sex, pval=TRUE, risk.table=TRUE,
           legend.labs=c("Male", "Female"))

# Forest plot of hazard ratios
ggforest(cox_lung, data=lung)
```



Key Findings:

- ECOG performance status strongest predictor
- Females have better survival (HR = 0.59)
- Model C-index = 0.68

Summary - Methods Comparison

Regression Methods Comparison Chart

Method	Outcome Type	Key Assumption	Main Output	Evaluation	R Function
Multiple Linear Regression	Continuous	Linearity, Normality Independence,	Coefficients, R^2	RMSE, R^2	lm()
Logistic Regression	Binary	Linearity in log-odds	Odds Ratios, Probabilities	AUC, Accuracy	glm(family= binomial)
Survival Analysis	Time to Event	Proportional Hazards	Hazard Ratios, Survival Curves	C-index, Log-rank	coxph(), survfit()

Decision Flow:
1. What type of outcome? → Continuous (MLR) | Binary (Logistic) | Time-to-event (Survival)
2. Check assumptions → Plot diagnostics | Check independence | Verify proportional hazards
3. Interpret results → Coefficients & p-values | Odds ratios & probabilities | Hazard ratios & survival curves

Decision Guide:

- **Continuous outcome?** → Multiple Linear Regression
- **Binary outcome?** → Logistic Regression
- **Time to event with censoring?** → Survival Analysis
- **Count data?** → Poisson/Negative Binomial (GLM)

Summary - Key Takeaways

Linear Regression:

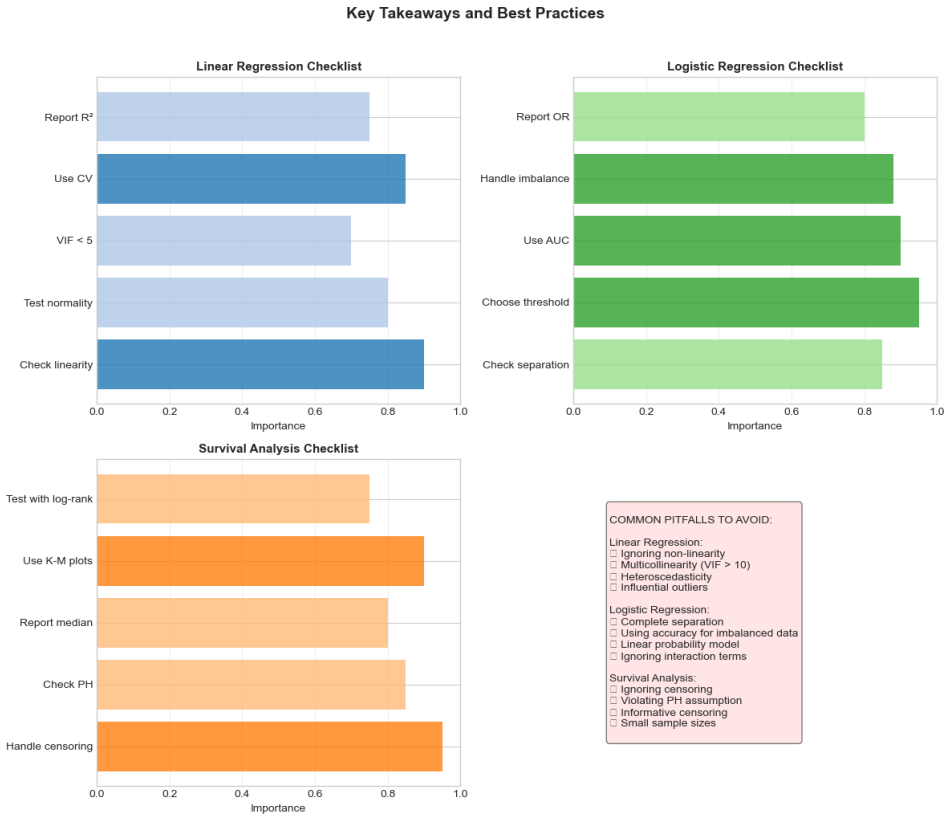
- Minimize squared errors
- Check assumptions
- Watch for multicollinearity
- Use adjusted R^2
- Validate with CV

Logistic Regression:

- Model probabilities
- Interpret odds ratios
- Choose threshold wisely
- Use AUC for evaluation
- Consider class imbalance

Survival Analysis:

- Handle censoring properly
- K-M for description
- Cox for regression
- Check PH assumption
- Report median survival



Common Pitfalls to Avoid:

- Ignoring assumptions in linear regression
- Using accuracy for imbalanced data in logistic regression
- Treating censored observations as complete in survival analysis

Summary - R Implementation Cheatsheet

Method	R Function	Key Output
Linear Regression	`lm(y ~ x1 + x2)`	coefficients, R^2, p-values
Logistic Regression	`glm(y ~ x, family=binomial)`	odds ratios, AUC
Survival (K-M)	`survfit(Surv(time, status) ~ 1)`	median survival, curve
Survival (Cox)	`coxph(Surv(time, status) ~ x)`	hazard ratios, C-index

Essential Packages:

```r
# Installation
install.packages(c("survival", "survminer",
                   "pROC", "caret", "glmnet"))

# Model diagnostics
library(car)      # VIF, diagnostic tests
library(broom)    # Tidy model output
library(ggplot2) # Visualization
```