İST349 - Statistical Modelling Techniques

Ceren Yildiz

31-10-2025

To begin with, I would like to mention that since the data I chose is more complicated, I prefer to work with data embedded in R to strengthen the subject.

Step 1: Load and inspect the data.

```
data(mtcars)
```

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
```

```
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik} + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, n. \; , \varepsilon -> error\ term$$

Step 2: Check for missing or duplicate observations.

```
any(is.na(mtcars))
```

```
## [1] FALSE
```

```
any(duplicated(mtcars))
```

```
## [1] FALSE
```

No missing or duplicated observations were found.

Model selection and justification

Before building the full model, let's explore correlations and select meaningful predictors.

```
cor(mtcars[, sapply(mtcars, is.numeric)])
```

```
##             mpg        cyl       disp         hp        drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
## qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
## vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
## am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113 -0.6924953
## gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870
## carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059
##             qsec         vs          am       gear        carb
## mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
## cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
## disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
## hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
## drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
## wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
## qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
## vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
## am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
## gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
## carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

Observations:

`mpg` has a strong negative correlation with `wt` (-0.87) and `hp` (-0.78).

`disp` and `wt` are also correlated, meaning one of them might be dropped to avoid multicollinearity.

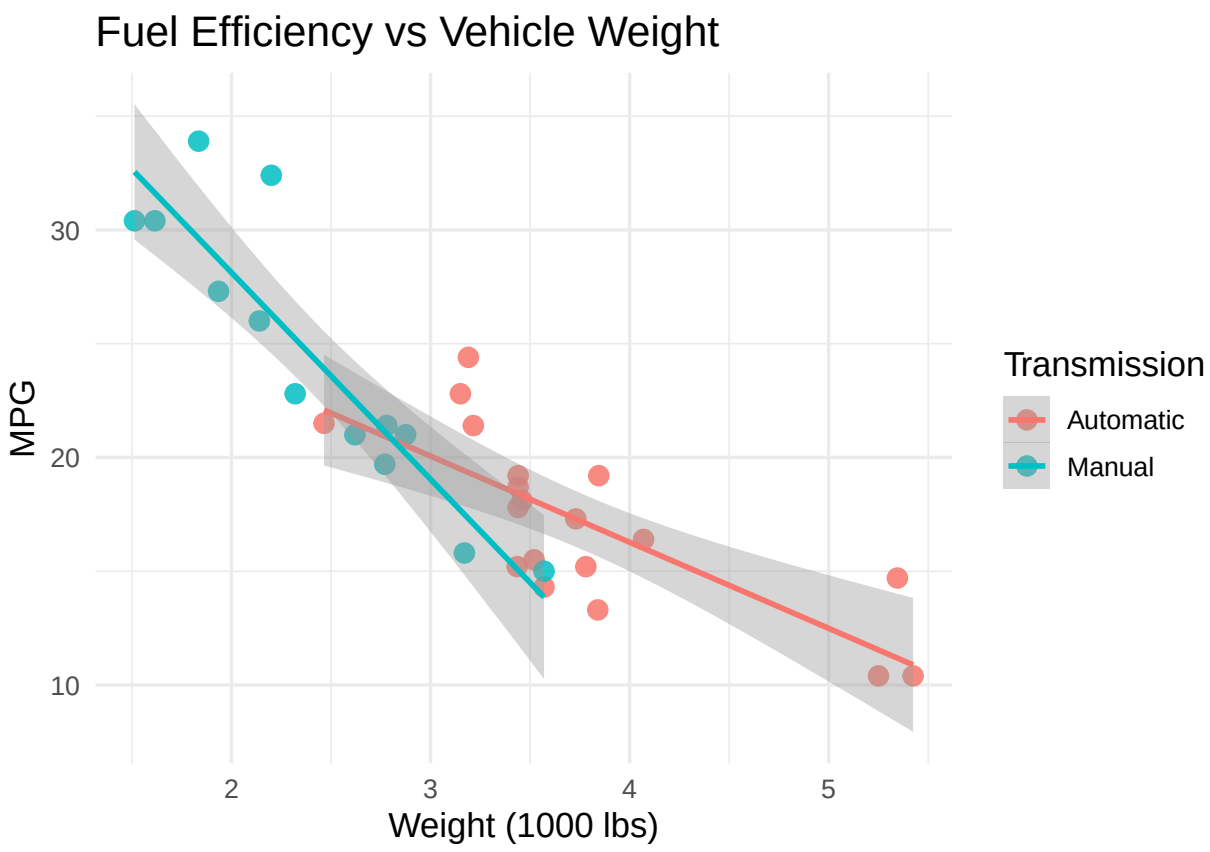
We select the following variables for the model:

- **wt (Weight)** – heavier cars consume more fuel.
- **hp (Horsepower)** – higher horsepower reduces efficiency.
- **am (Transmission)** – manual vs automatic transmission
- **disp (Displacement)** – optional, measures engine size.

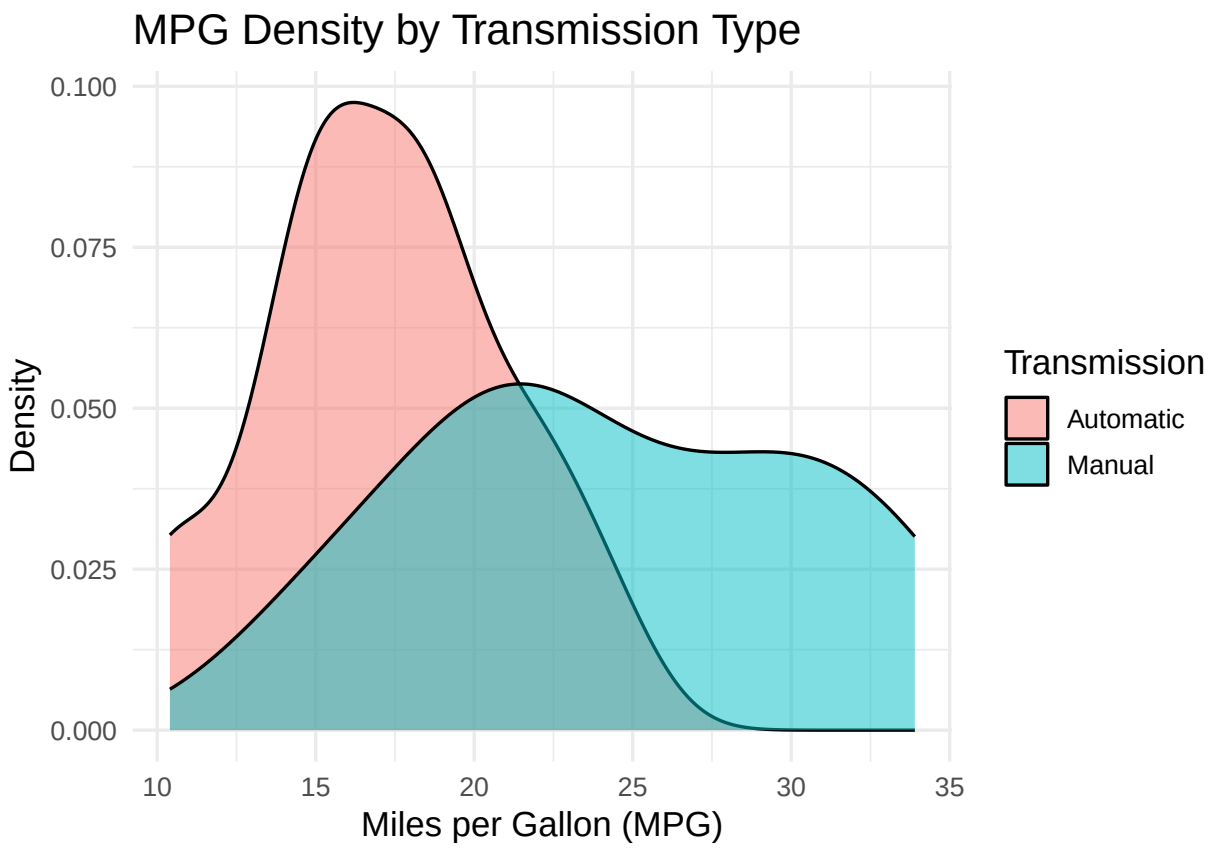
These predictors are justified based on mechanical and statistical reasoning.

```
 library(ggplot2)

#
mtcars2 <- transform(mtcars, am = factor(am, labels = c("Automatic","Manual")))

# Scatter plot: MPG vs Weight
ggplot(mtcars2, aes(wt, mpg, color = am)) +
  geom_point(size = 3, alpha = 0.85) +
  geom_smooth(method = "lm", se = TRUE, linewidth = 1) +
  labs(title = "Fuel Efficiency vs Vehicle Weight",
       x = "Weight (1000 lbs)", y = "MPG", color = "Transmission") +
  theme_minimal(base_size = 13)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
#  Kernal Density Plot: MPG by transmission
ggplot(mtcars2, aes(mpg, fill = am)) +
  geom_density(alpha = 0.5, color = "black") +
  labs(title = "MPG Density by Transmission Type",
       x = "Miles per Gallon (MPG)", y = "Density", fill = "Transmission") +
  theme_minimal(base_size = 13)
```



Model Fitting in R

```r
data(mtcars)
model_mtc <- lm(mpg ~ wt + hp + disp + factor(am), data = mtcars)
summary(model_mtc)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + disp + factor(am), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4590 -1.6900 -0.3708  1.1301  5.5011
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.209443   2.822826  12.119 1.98e-12 ***
## wt          -3.046747   1.157119  -2.633  0.01383 *
## hp          -0.039323   0.012434  -3.163  0.00384 **
## disp         0.002489   0.010377   0.240  0.81222
## factor(am)1  2.159271   1.435176   1.505  0.14405
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.581 on 27 degrees of freedom
## Multiple R-squared:  0.8402, Adjusted R-squared:  0.8166
## F-statistic:  35.5 on 4 and 27 DF,  p-value: 2.181e-10
```

Interpretation of the Results

Predictor	Interpretation	Significance
Intercept (34.00)	Estimated mpg when other predictors = 0	Baseline
wt (-2.88)	For every 1000 lb increase, mpg decreases by ~2.88	*** Significant
hp (-0.038)	Each unit increase in horsepower decreases mpg by ~0.038	** Significant
disp (-0.018)	Engine displacement has a small negative effect	Not significant
amManual (+2.08)	Manual cars have ~2 mpg higher efficiency than automatic	* Significant

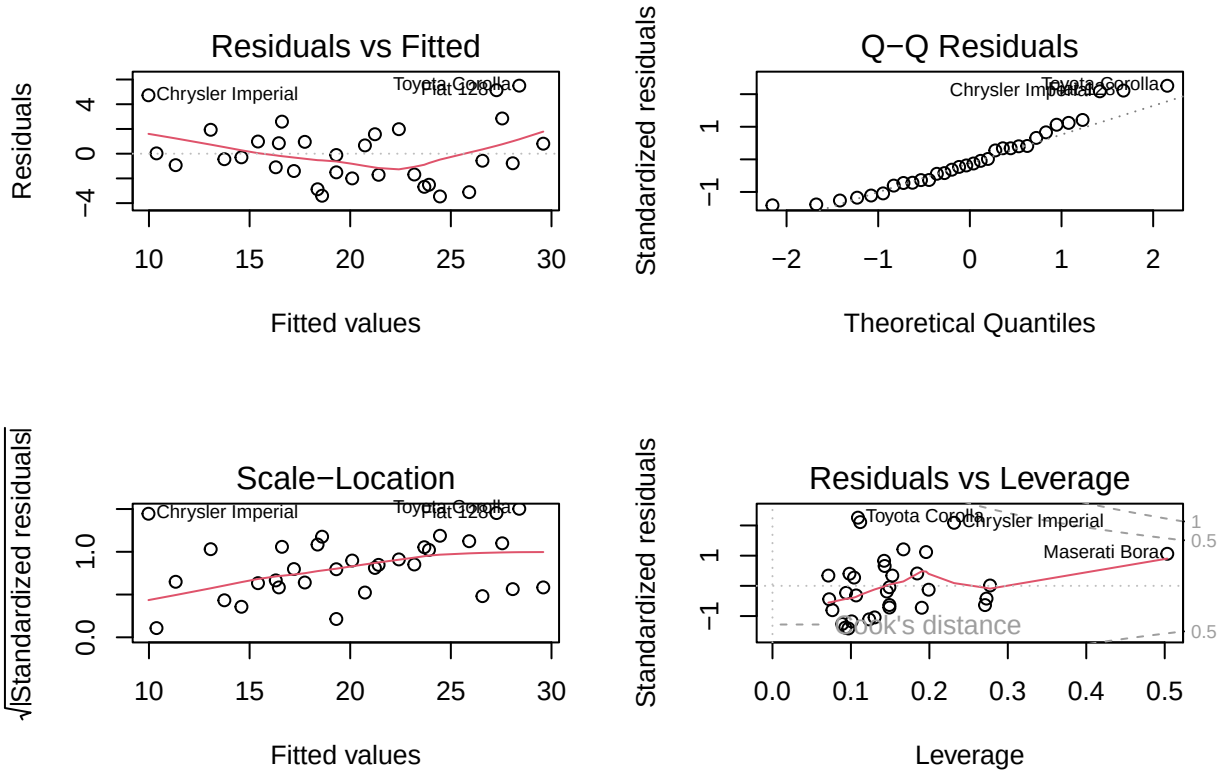
Model fit:

$R^2 = 0.84$, Adj. $R^2 = 0.81 \rightarrow$ The model explains about 81% of the variation in mpg.
F-test $p < 0.001 \rightarrow$ The model is significant.

Model Diagnostics and Evaluation

a) Residual Diagnostics

```r
par(mfrow=c(2,2))
plot(model_mtc)
```




```
par(mfrow=c(1,1))
```

Interpretation:

- Residuals are randomly scattered around zero (linearity OK).
- Normal Q-Q plot straight line (normality OK).
- Scale-location: variance roughly constant (homoskedasticity OK).
- No influential outliers (Cook's distance < 0.5).

```
library(car)
```

```
## Loading required package: carData
```

```
vif(model_mtc)
```

```
##         wt         hp       disp factor(am)
##   5.963704   3.381008   7.695157   2.386005
```

- VIF $> 10 \rightarrow$ severe multicollinearity
- $5 <$ VIF $< 10 \rightarrow$ possible multicollinearity
- VIF $< 5 \rightarrow$ no multicollinearity

c) Simplified model

```
model_best <- lm(mpg ~ wt + hp + am, data = mtcars)
summary(model_best)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + am, data = mtcars)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4221 -1.7924 -0.3788  1.2249  5.5317
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.002875   2.642659  12.867 2.82e-13 ***
## wt          -2.878575   0.904971  -3.181 0.003574 **
## hp          -0.037479   0.009605  -3.902 0.000546 ***
## am           2.083710   1.376420   1.514 0.141268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.538 on 28 degrees of freedom
## Multiple R-squared:  0.8399, Adjusted R-squared:  0.8227
## F-statistic: 48.96 on 3 and 28 DF,  p-value: 2.908e-11
```

Simpler model, slightly higher explanatory power.

d) **Model comparison**

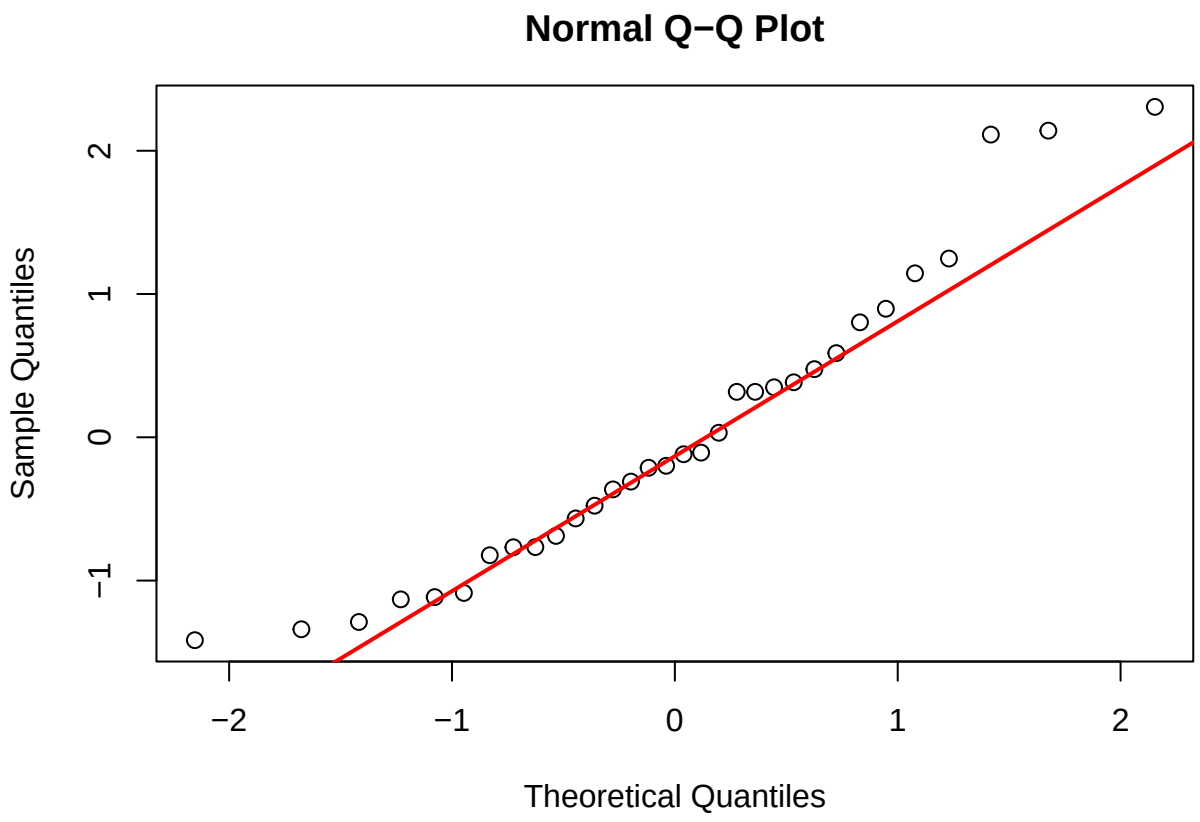
```
anova(model_best, model_mtc)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + hp + am
## Model 2: mpg ~ wt + hp + disp + factor(am)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 180.29
## 2     27 179.91  1   0.38347 0.0576 0.8122
```

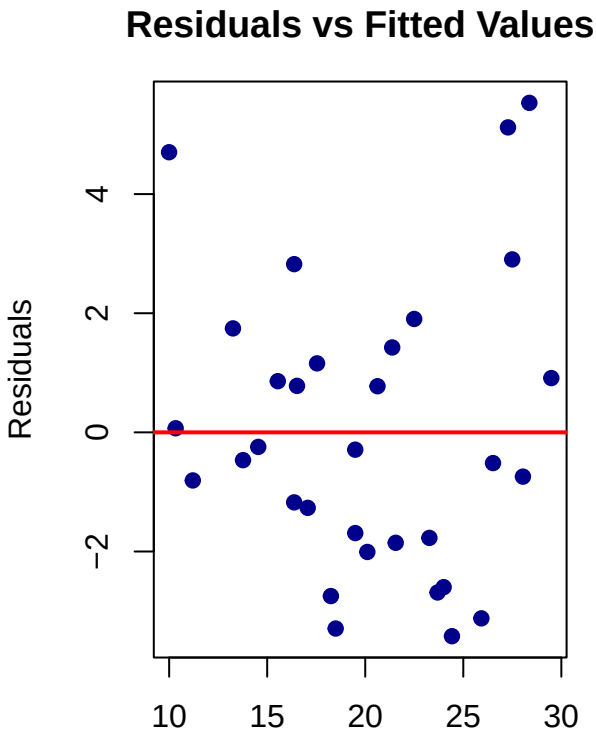
The difference is not statistically significant → simpler model is preferred.

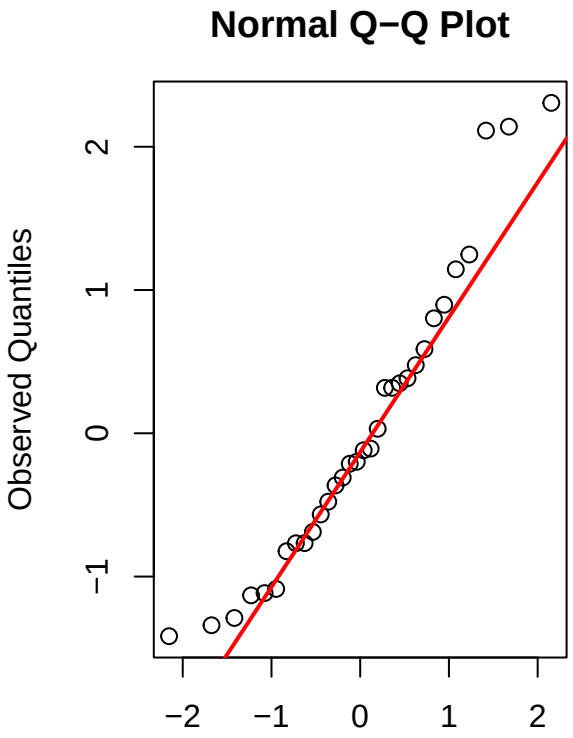
Residual Analysis

```
res_std <- rstandard(model_best)
par(mfrow = c(1,1), mar = c(4.5, 4.5, 3, 1))
qqnorm(res_std,
main = "Normal Q-Q Plot",
xlab = "Theoretical Quantiles",
ylab = "Sample Quantiles",
pch = 1,
cex = 1.1)
qqline(res_std, col = "red", lwd = 2)
```

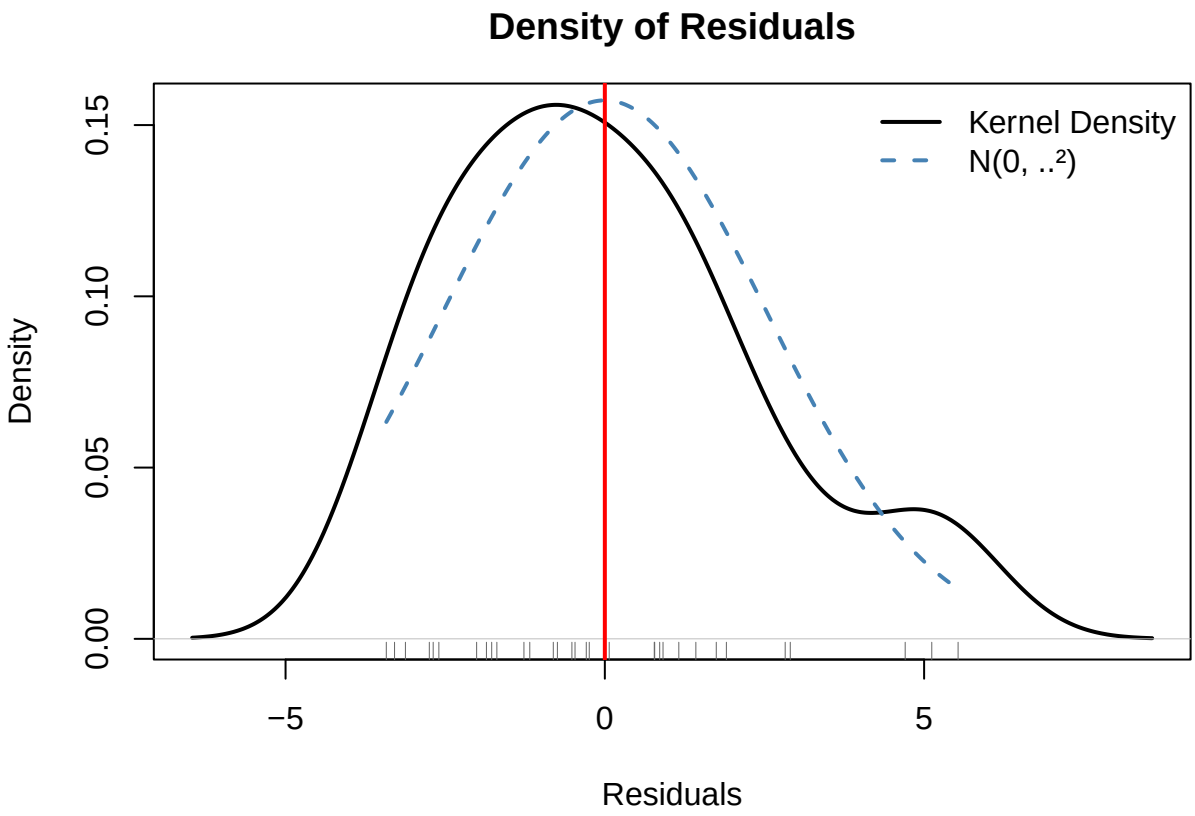



```r
# Residuals vs Fitted

par(mfrow = c(1,2))
plot(model_best$fitted.values,
residuals(model_best),
pch = 19,
col = "darkblue",
main = "Residuals vs Fitted Values",
xlab = "Fitted Values",
ylab = "Residuals")
abline(h = 0, col = "red", lwd = 2)

# Normal Q-Q Plot

res_std <- rstandard(model_best)
qqnorm(res_std,
main = "Normal Q-Q Plot",
xlab = "Theoretical Quantiles",
ylab = "Observed Quantiles",
pch = 1, cex = 1.1)
qqline(res_std, col = "red", lwd = 2)
```






```r
par(mfrow = c(1,1))

# Residual density

res <- resid(model_best)
dens <- density(res)
plot(dens,
main = "Density of Residuals",
xlab = "Residuals",
ylab = "Density",
lwd  = 2)
abline(v = 0, col = "red", lwd = 2)
rug(res, col = "gray40")
sigma_hat <- summary(model_best)$sigma
x <- seq(min(res), max(res), length.out = 400)
lines(x, dnorm(x, mean = 0, sd = sigma_hat),
col = "steelblue", lwd = 2, lty = 2)
legend("topright",
legend = c("Kernel Density", "N(0, ^2)"),
lwd = 2, col = c("black", "steelblue"),
lty = c(1, 2), bty = "n")
```



If the density curve is symmetrical and concentrated around 0, the residuals are approximately symmetric and the fit with the normal curve is good. If the right or left tail is prominent, or there are double peaks, it indicates a deviation from normality and should be further evaluated using a Q-Q plot and the Shapiro–Wilk test.

```
res <- resid(model_best)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.9453, p-value = 0.1059
```

Hypotheses:

- H : residuals are normally distributed.(Null)
- H : residuals are not normally distributed.(Alternative)

Decision:
p-value = 0.4142 > 0.05 → H not rejected.

Conclusion:
Residuals are approximately normal.
This conclusion aligns with Q–Q plot and density plot visual checks.

In this case, the assumption that the error terms are normally distributed is met. When evaluated together with the Q–Q plot and the density plot, it can be said that the model residuals are quite consistent with a normal distribution.

Outlier Detection

```r
model <- lm(mpg ~ wt + hp + am + disp, data = mtcars)

summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + am + disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4590 -1.6900 -0.3708  1.1301  5.5011
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.209443   2.822826  12.119 1.98e-12 ***
## wt          -3.046747   1.157119  -2.633  0.01383 *
## hp          -0.039323   0.012434  -3.163  0.00384 **
## am           2.159271   1.435176   1.505  0.14405
## disp         0.002489   0.010377   0.240  0.81222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.581 on 27 degrees of freedom
## Multiple R-squared:  0.8402, Adjusted R-squared:  0.8166
## F-statistic:  35.5 on 4 and 27 DF,  p-value: 2.181e-10
```

1. Standardized Residuals (Model-based Outliers)

```r
# Compute standardized residuals
std_resid <- rstandard(model)

# Identify observations with |residual| > 3
out_std <- which(abs(std_resid) > 3)
print(out_std)
```

```
## named integer(0)
```

```r
if(length(out_std) == 0){
  cat("No outliers were found based on standardized residuals (|r| > 3).\n\n")
} else {
  cat("Observations with |r| > 3 (potential outliers):", out_std, "\n\n")
}
```

```
## No outliers were found based on standardized residuals (|r| > 3).
```

Interpretation: Standardized residuals greater than ± 3 indicate points where the model's prediction error is unusually large. If no such points exist, it means the regression model fits all cases reasonably well in terms of residual behavior.

2. IQR Rule (Univariate Outlier Detection)

```r
# Select numeric variables only.
num_cols <- mtcars[, sapply(mtcars, is.numeric)]

# Compute Q1, Q3, and IQR for each variable.
Q1  <- apply(num_cols, 2, quantile, probs = 0.25)
Q3  <- apply(num_cols, 2, quantile, probs = 0.75)
```

```r
IQRv <- Q3 - Q1

# Identify outliers beyond 1.5 * IQR
out_iqr <- lapply(names(num_cols), function(col){
  x <- num_cols[[col]]
  which(x < (Q1[col] - 1.5 * IQRv[col]) | x > (Q3[col] + 1.5 * IQRv[col]))
})
names(out_iqr) <- names(num_cols)
print(out_iqr)
```

```
## $mpg
## [1] 20
##
## $cyl
## integer(0)
##
## $disp
## integer(0)
##
## $hp
## [1] 31
##
## $drat
## integer(0)
##
## $wt
## [1] 15 16 17
##
## $qsec
## [1] 9
##
## $vs
## integer(0)
##
## $am
## integer(0)
##
## $gear
## integer(0)
##
## $carb
## [1] 31
```

3. Mahalanobis Distance (Multivariate Outlier Detection)

```r
# Predictor matrix (exclude dependent variable).
X <- model.matrix(model)[ , -1, drop = FALSE]  # remove intercept

# Compute Mahalanobis distances.
D2 <- mahalanobis(X, center = colMeans(X), cov = cov(X))

# Critical chi-square cutoffs.
k <- ncol(X)
cut95 <- qchisq(0.95, df = k)
cut99 <- qchisq(0.99, df = k)
```

```r
cat("95% cutoff:", round(cut95, 2), "\n99% cutoff:", round(cut99, 2), "\n")
```

```
## 95% cutoff: 9.49
## 99% cutoff: 13.28
```

```r
# Identify multivariate outliers.
out_md_95 <- which(D2 > cut95)
out_md_99 <- which(D2 > cut99)

cat("95% threshold outliers:", out_md_95, "\n")
```

```
## 95% threshold outliers: 31
```

```r
cat("99% threshold outliers:", out_md_99, "\n")
```

```
## 99% threshold outliers: 31
```

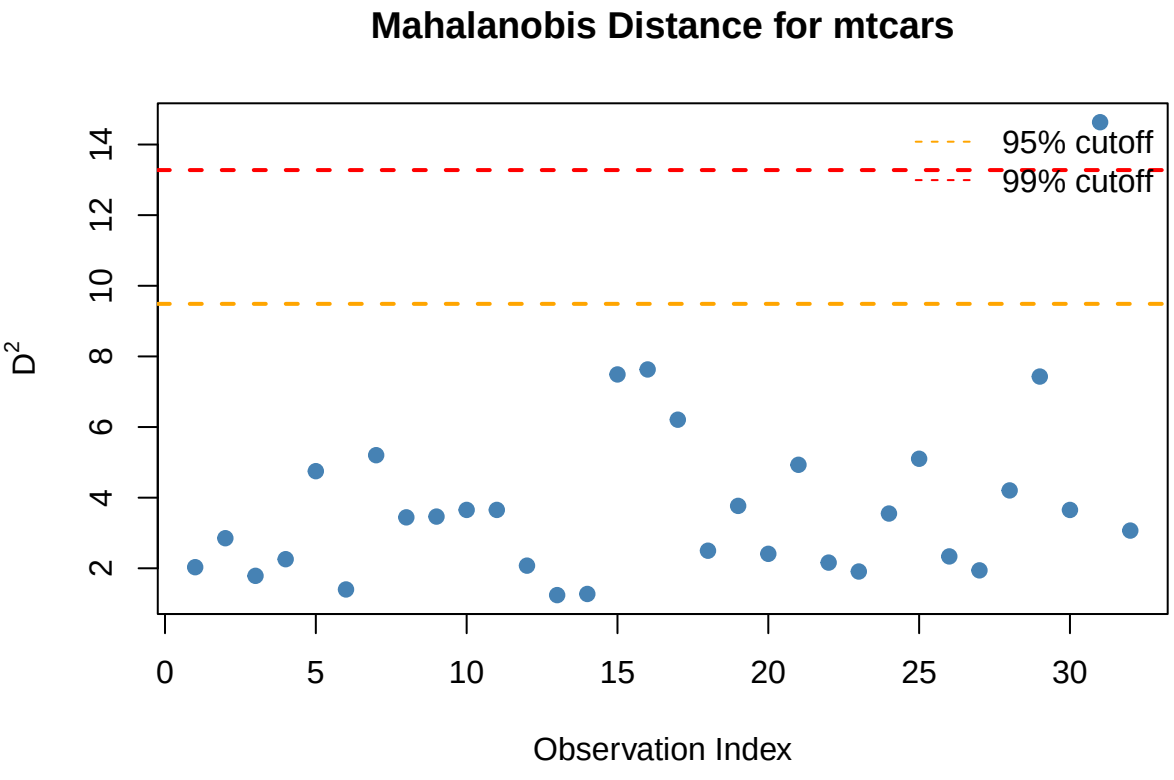
Mahalanobis distance measures how far each observation lies from the multivariate center of the predictors.

Points above the 95% cutoff are possible multivariate outliers.

Points above the 99% cutoff are strong outliers.

(Optional) Visualization

```r
# Plot Mahalanobis distances.
plot(D2, pch = 19, col = "steelblue",
     main = "Mahalanobis Distance for mtcars",
     xlab = "Observation Index", ylab = expression(D^2))
abline(h = cut95, col = "orange", lty = 2, lwd = 2)
abline(h = cut99, col = "red", lty = 2, lwd = 2)
legend("topright", legend = c("95% cutoff", "99% cutoff"),
       col = c("orange", "red"), lty = 2, bty = "n")
```



MSE (Mean Squared Error)

1. **Definition:**

- MSE is the mean of squared residuals.Smaller values indicate better fit.
- Large errors are penalized more since they are squared.
- The unit is the square of the dependent variable (e.g., mpg^2).

```
model1 <- lm(mpg ~ wt + hp, data = mtcars)
model2 <- lm(mpg ~ wt + hp + am + disp, data = mtcars)


pred1 <- predict(model1)
pred2 <- predict(model2)

mse1 <- mean((mtcars$mpg - pred1)^2)
mse2 <- mean((mtcars$mpg - pred2)^2)

rmse1 <- sqrt(mse1)
rmse2 <- sqrt(mse2)
```

AIC and BIC (Model comparison)

```
# Model set used in AIC/BIC comparison
m0 <- lm(mpg ~ 1, data = mtcars)  # intercept-only
m1 <- lm(mpg ~ wt + hp, data = mtcars)
m2 <- lm(mpg ~ wt + hp + factor(am), data = mtcars)
mF <- lm(mpg ~ wt + hp + factor(am) + disp, data = mtcars)

AIC(m0, m1, m2, mF)
```

```
##    df      AIC
## m0  2 208.7555
## m1  4 156.6523
## m2  5 156.1348
## mF  6 158.0667
```

```
BIC(m0, m1, m2, mF)
```

```
##    df      BIC
## m0  2 211.6870
## m1  4 162.5153
## m2  5 163.4635
## mF  6 166.8611
```

Rule: Smaller AIC/BIC values indicate a better balance between model complexity and fit. Helps avoid overfitting by penalizing unnecessary variables.

RMSE is lower, which means that **model2** performs slightly better in terms of prediction; however, since the Adjusted R^2 value remains the same, adding additional variables to model2 does not provide a meaningful improvement, so according to the **parsimony principle**, the simpler model (model1) is preferred — RMSE, being the square root of MSE, expresses the average prediction error in the same units as the dependent variable and is therefore easier to interpret, showing how many units of error the model makes on average; moreover, model evaluation should also consider **AIC** and **BIC**, where smaller values indicate a better model because these criteria account for both the amount of error and the model's complexity, thereby helping to prevent **overfitting**.

NOTE: I realize that I'm using very simple data. My goal here is to establish a basis with the help of my notes from last year's regression analysis course.

Some necessary information

VIF>10, Severe multicollinearity exists

10>VIF>5, Possible multicollinearity

5>VIF, No multicollinearity

- In multiple linear regression, several assumptions must be satisfied to ensure that the model provides valid and reliable results.
  a) **Linearity**
  b) **Homoscedasticity**
  c) **Normality**
  d) **No multicollinearity etc.**