Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

DEPARTMENT OF CIVIL ENGINEERING

SHC 798

APPLIED STATISTICAL METHODS AND OPTIMISATION

Multiple Linear Regression & ANOVA

RICHARD LUBEGA

Full names

25585089

Student number

2

Assignment

DECLARATION

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this submission is my original work. Wherever other people's work has been used (either from a printed source, the internet or any other source) thishas been properly acknowledged and referenced in accordance with departmental requirements.
3. I declare that I have used AI-based tools (*ChatGPT*, *Grok* and *Manus*) to help interpret and debug my R code for attempting the assignment questions.
4. I have not used another student's current or past written work to hand in as myown.
5. I have not allowed and will not allow anyone to copy my work to pass it off as his or her work.

Signature:

Date: 06-10-2025

TABLE OF CONTENTS

1 Part 1: Multiple Linear Analysis (MLR)

1.1 Question 1

Concrete Strength Data

Question 1: Concrete Strength

```
pacman::p_load(tidymodels)

# Getting started with the dataset in concrete.csv
concrete <- read.csv(file.choose(), header = TRUE, na.strings = c("NA")) # open dataset
head(concrete) # View first few rows of the dataset
```

```
##   cement  wcr age strength
## 1  369.9 0.48  14     12.7
## 2  344.5 0.49  28     12.7
## 3  375.9 0.44  14     14.7
## 4  410.9 0.44  28     25.0
## 5  340.6 0.54  28      7.9
## 6  340.6 0.57  28      4.9
```

1.1.1 Part a): Data Preparation

a) Histograms and Marginal Distributions

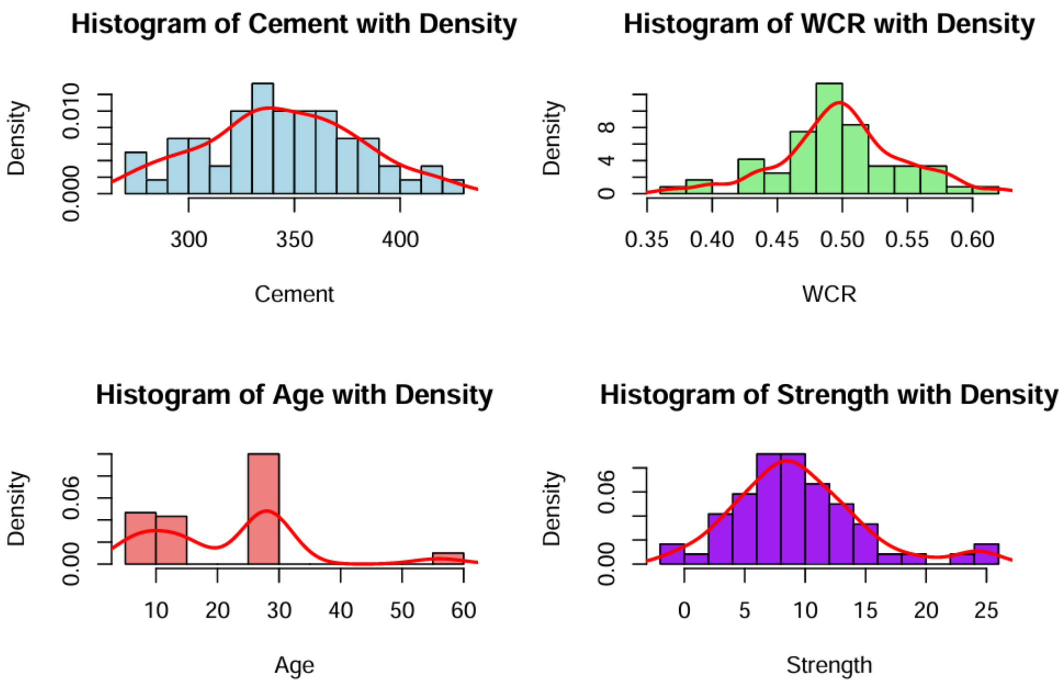
```
# [a-1] Histograms with overlaid marginal density distributions
par(mfrow = c(2, 2))

# cement
hist(concrete$cement, main = "Histogram of Cement with Density",
     xlab = "Cement", col = "lightblue", probability = TRUE, breaks = 15)
lines(density(concrete$cement), col = "red", lwd = 2)

# wcr
hist(concrete$wcr, main = "Histogram of WCR with Density",
     xlab = "WCR", col = "lightgreen", probability = TRUE, breaks = 15)
lines(density(concrete$wcr), col = "red", lwd = 2)

# age
hist(concrete$age, main = "Histogram of Age with Density",
     xlab = "Age", col = "lightcoral", probability = TRUE, breaks = 15)
lines(density(concrete$age), col = "red", lwd = 2)

# strength
hist(concrete$strength, main = "Histogram of Strength with Density",
     xlab = "Strength", col = "purple", probability = TRUE, breaks = 15)
lines(density(concrete$strength), col = "red", lwd = 2)
```



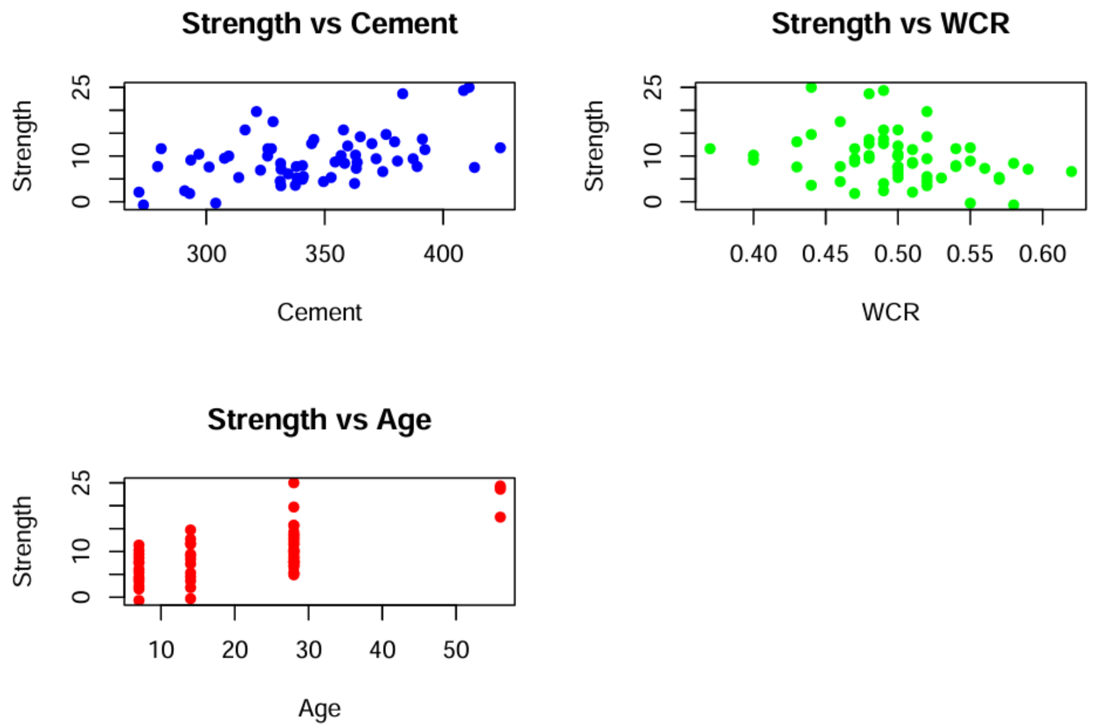
b) Scatter Plots

```
# [a-2] Scatter Plots of Strength against each Predictor
par(mfrow = c(2, 2))

# Strength vs Cement
plot(concrete$cement, concrete$strength, main = "Strength vs Cement",
     xlab = "Cement", ylab = "Strength", pch = 16, col = "blue")


# Strength vs WCR
plot(concrete$wcr, concrete$strength, main = "Strength vs WCR",
     xlab = "WCR", ylab = "Strength", pch = 16, col = "green")

# Strength vs Age
plot(concrete$age, concrete$strength, main = "Strength vs Age",
     xlab = "Age", ylab = "Strength", pch = 16, col = "red")
```



<u>Commenting on the Trend and Need for Variable Transformation</u>

The marginal plots are not skewed and there is <u>no</u> warranted <u>need for variable transformations</u>.

The scatter plot for `strength` vs `age` indicates has distinct values (`7`, `14`, `28`, `56`) which suggests a discrete or categorical nature rather than continuous. The marginal plots for `age` also show spikes at these specific ages rather than a smooth distribution.

Therefore, age may be as a categorical variable (factor) in regression to account for its discrete levels. Including *interaction terms* (e.g., `cement:age`, `wcr:age`) in such a regression model may be also necessary.

1.1.2 Part b): Multicollinearity

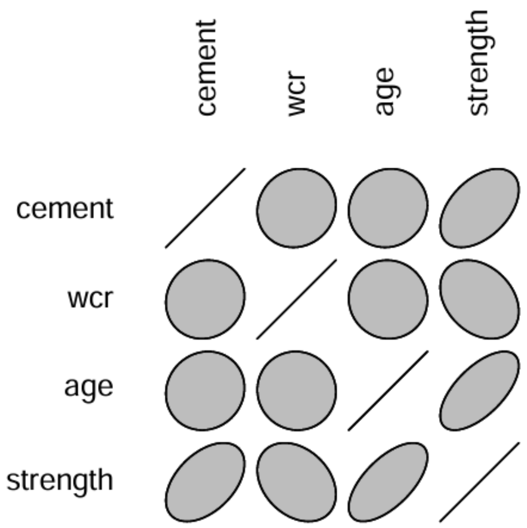
1.1.2.1 Pearson correlation coefficients

```r
# (i) Pearson correlation coefficients
cor(concrete, method = "pearson")
```

```
##              cement         wcr         age   strength
## cement   1.00000000  0.08414330  0.07698239  0.4657863
## wcr      0.08414330  1.00000000 -0.02466868 -0.3063764
## age      0.07698239 -0.02466868  1.00000000  0.6345642
## strength 0.46578632 -0.30637643  0.63456425  1.0000000
```

1.1.2.2 Ellipse plot to visualise collinearity

```r
# (ii) An ellipse plot to visualise collinearity
pacman::p_load(ellipse)
plotcorr(cor(concrete))
```



1.1.2.3 Variance Inflation Factors (VIFs)

```r
# (iii) Variance Inflation Factors (VIFs)
pacman::p_load(car)
conc_model <- lm(strength ~ cement + wcr + age, data = concrete)
vif(conc_model)
```

```
##   cement      wcr      age
## 1.013514 1.008121 1.006951
```

Comment on the findings

From the above collinearity audit checks (Pearson correlation coefficients and the ellipse plot), the somewhat elongated ellipses, particularly between `strength` and `cement` (0.46578632), and `strength` and `age` (0.6345642), suggest potential multicollinearity among these predictors.

This indicates that these predictors may be highly correlated with each other and with the response variable, but Since all VIF values are very close to 1 (well below 5), there is <u>no significant multicollinearity</u> among the predictors. This suggests that the predictors are largely independent of each other, which is ideal for a stable regression model.

1.1.3 Part c) Model Output

1.1.3.1 Multiple Regression Model

The Model [`conc_model`]: `strength ~ cement + wcr + age`

```
conc_model <- lm(strength ~ cement + wcr + age, data = concrete)
summary(conc_model)
```

```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

1.1.3.2 Model Output, Adequacy & Appropriateness of Fit

a) **Regression Coefficients**

The **slope** coefficients (`cement: 0.06657`, `wcr: -37.44811`, and `age: 0.26614`) indicate the respective change (increase [+] or decrease [-]) in the concrete strength when each of the predictors increase by 1 unit, but all other predictors remain unchanged.

- The p-values in summary(`conc_model`) determine whether the different response-predictor relationships are statistically significant. The p-value are all below 0.05, so we reject the null hypothesis on a 5% significance level and conclude that all the variables (`cement`, `wcr`, and `age`) significantly affect concrete strength. A zero slope coefficient is implausible for all the predictors.

The **intercept** coefficient corresponds to the estimated (theoretical) concrete strength value when all the predictors (`cement`, `wcr`, and `age`) are equal to zero.

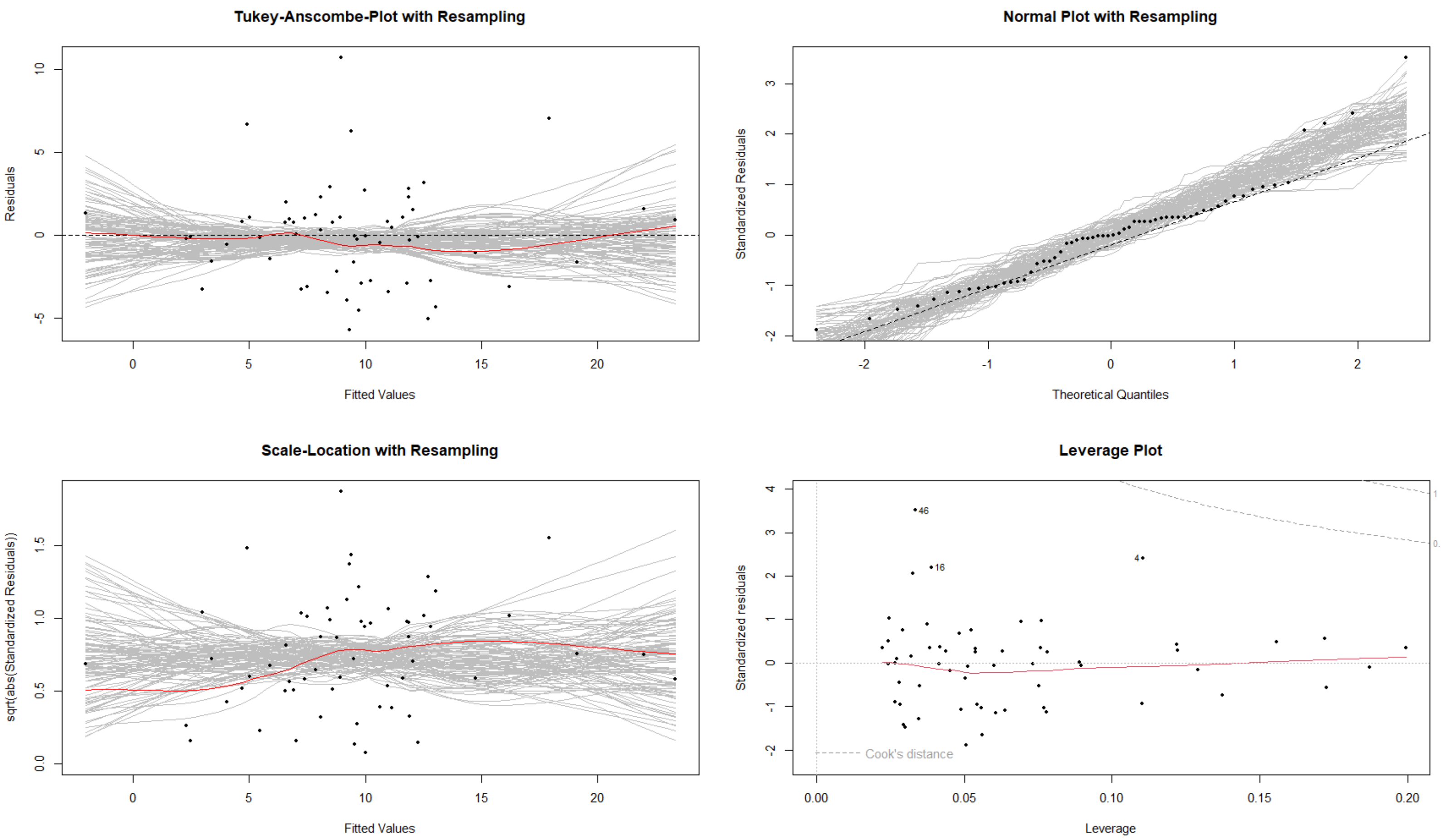
- It's p-value (`0.942`) is not statistically significant at the 5% level, and an intercept of zero is plausible.

- However, interpreting this is not practically rational but ensures the regression hyperplane fits the data best within the observed predictor values range. It is not meaningful to extrapolate the predictors to zero.

b) **Model Significance**

From the summary (the global F-Statistic), we gather that p-value is very small (`4.441e-14`) and that the model is highly significant at the 5% level.

c) **Appropriateness of Fit [Model Diagnostics]**



(i) **Linearity: E [E_i] = 0**

The Tukey-Anscombe residual plot shows that the smoother does not deviate from the x-axis except for a slight kink for fitted values between 10 and 20 but this deviation can be attributed to randomness. Using the resampling approach by the R function, `resplot()`, the original red smoother is within what can be generated by random sampling. It is thus imperative to that we accept the linearity hypothesis E [E_i] = 0.

Hence, there is no systematic error and the hyperplane is the correct fit.

(ii) Homoskedasticity, Var (E_i) = σ^2_E

From the Scale-Location plot, the red smoother is generally horizontal with a gentle kink (between 5 and 17 of the fitted values) which can be considered random. Using the resampling approach, the smoother line is well within the confidence region. We can consider that there is no heteroscedasticity.

(iii)No Correlation: Cov (E_i, E_j) = 0

Since the concrete dataset observations are not directly affected by temporal variation (in the `age` variable), the errors may be autocorrelated. The Durbin-Watson test run to check this.

```
# Autocorrelation using the Durbin-Watson test
pacman::p_load(lmtest)
dwtest(conc_model)
```

```
##
##  Durbin-Watson test
##
## data:  conc_model
## DW = 1.8426, p-value = 0.2733
## alternative hypothesis: true autocorrelation is greater than 0
```

The Durbin-Watson statistic (`1.8426`) is close to 2 and the high p-value (`0.2733`) implies that the small deviation from 2 could easily be due to random chance. Thus, Meaning: we fail to reject the null hypothesis of positive autocorrelation in residuals. There is no statistically significant evidence of autocorrelation in the residuals of the model.

The residuals may be considered independent and uncorrelated.

(iv)Normality: E_i ~ N(0,σ^2_E)

From the Normal Q-Q Plot, the bulk of the residuals (largely in the central region) are approximately Gaussian distributed. A noticeable deviation (3 outliers) at the upper tail indicates right skewness and departure from normality but because all residuals from the concrete dataset fall within the resampling based confidence region, there is no systematic deviation from the normal distribution. Therefore, the *i.i.d.* assumption holds.

d) **Adequacy of Fit [R^2]**

The R-squared from summary (`conc_model`) indicates how much variation in concrete strength is explained by the three predictors as per the regression hyperplane. Here, multiple R^2 = `0.6852` (the adjusted R^2 = `0.6684`), meaning that 69% of the variation in concrete strength is explained by predictors (`cement`, `wcr`, and `age`), while the remaining 31% is due to other factors not included in the model.

Summary: From the R^2 value (`0.6852`), the regression model (hyperplane) is **adequate** because it accounts for a large portion of the total variation in the concrete strength. The model is also **appropriate** because of the good model diagnostics.

1.1.4 Part c): Variable Selection

Starting from the initial (`conc_model`).

a) **Backward Elimination Model** (`conc.back`)

```
# Backward Elimination with AIC
conc.back <- stats::step(conc_model, direction="backward")
```

```
summary(conc.back)
```

```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##     Min      1Q Median      3Q     Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

b) Forward Selection **Model** (`conc.forw`)

```
# Forward Selection with AIC
conc_null <- lm(strength ~ 1, data = concrete) # Intercept-only model
sc <- list(lower=conc_null, upper=conc_model)
conc.forw <- stats::step(conc_null, scope=sc, direction="forward", k=2)
```

```r
summary(conc.forw)
```

```
##
## Call:
## lm(formula = strength ~ age + cement + wcr, data = concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

c) AIC Stepwise **Models** [`conc.b1`, `conc.b2`, and `conc.b3`]

```r
# AIC Stepwise Model Search: Both Directions Approach
# starting with the null model
conc.b1 <- stats::step(conc_null, scope = sc, direction = "both")
```

Models `conc.b1`

```r
summary(conc.b1)
```

```
##
## Call:
## lm(formula = strength ~ age + cement + wcr, data = concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

Models conc.b2

```r
summary(conc.b2)
```

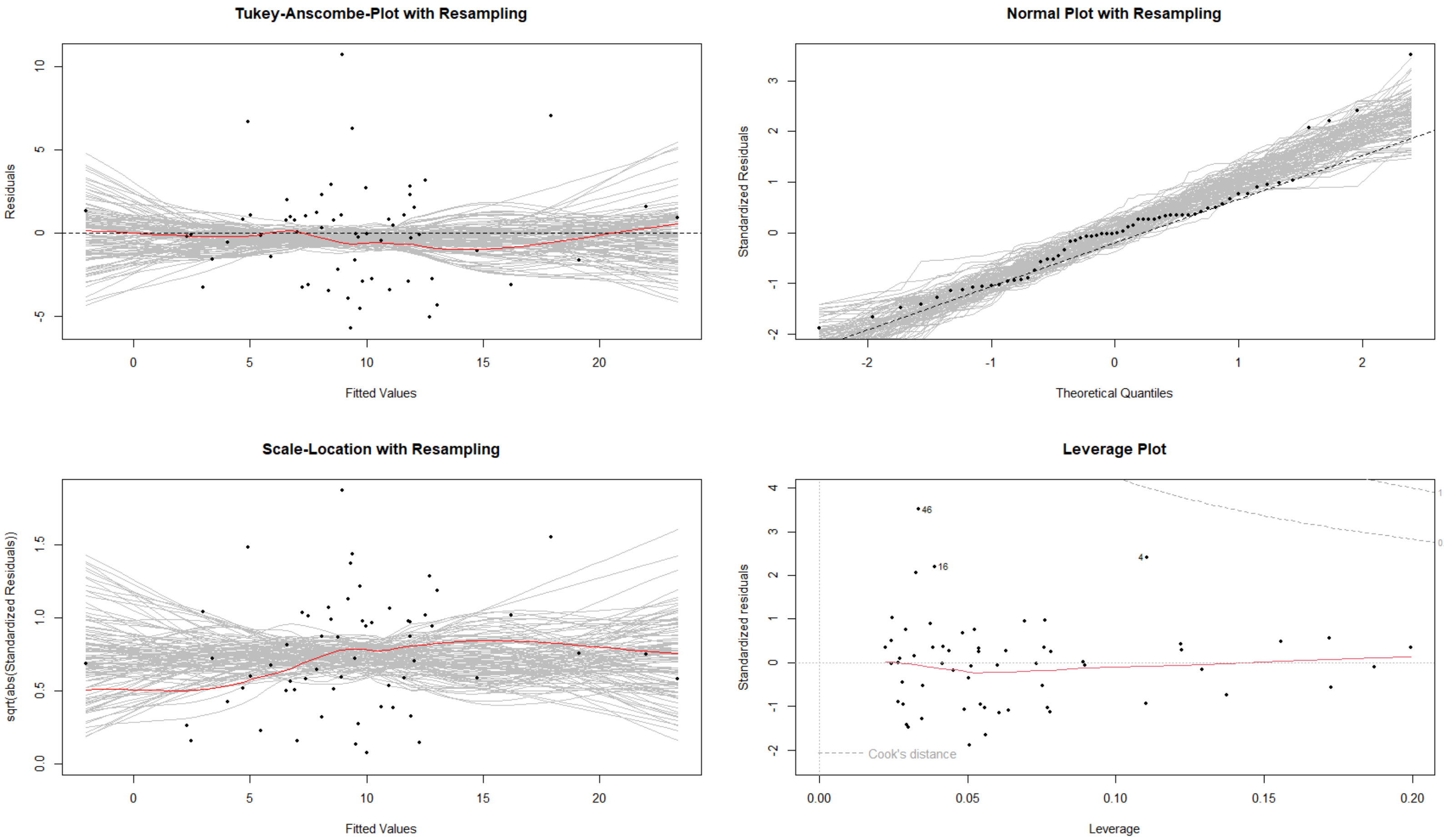
```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

Models conc.b3

```r
summary(conc.b3)
```

```
##
## Call:
## lm(formula = strength ~ wcr + age + cement, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

Residual plots for `conc.forw`



Commenting on Variable Selection Results

For all the variable selection methods, all the 3 predictor variables (`cement`, `wcr`, and `age`) are retained in all the 6 models (the initial model and the 5 test models).

There are no also major improvements in residual plots for all the models (residual plots for model `conc.forw` are shown above). Also, there is no noticeable changes on predictor significance or model fit.

Therefore, I would recommend the initial model (`conc_model`) before variable selection was conducted.

1.1.5 Part d): 5-fold Cross Validation & MSPE

The 5-fold cross-validation loop code

```
# Full Model is (strength ~ cement + wcr + age)
# Reduced Model is (strength ~ cement + age); wcr is dropped to see effect on prediction performance

set.seed(123) # Set seed for reproducibility

n <- nrow(concrete) # Number of observations
k <- 5 # Number of folds
sb <- round(seq(0, n, length = (k + 1)))  # Fold boundaries

# Initialize vectors to store MSPE for each model
mspe_full <- numeric(k)
mspe_reduced <- numeric(k)

# 5-fold cross-validation for full model (strength ~ cement + wcr + age)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]
  train <- (1:n)[-test]
  fit_full <- lm(strength ~ cement + wcr + age, data = concrete[train, ])
  pred_full <- predict(fit_full, newdata = concrete[test, ])
  mspe_full[i] <- mean((concrete$strength[test] - pred_full)^2, na.rm = TRUE)
}

# 5-fold cross-validation for reduced model (strength ~ cement + age)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]  # Same fold split comparability
  train <- (1:n)[-test]
  fit_reduced <- lm(strength ~ cement + age, data = concrete[train, ])
  pred_reduced <- predict(fit_reduced, newdata = concrete[test, ])
  mspe_reduced[i] <- mean((concrete$strength[test] - pred_reduced)^2, na.rm = TRUE)
}

# Calculating overall MSPE for each model
mspe_full_mean <- mean(mspe_full, na.rm = TRUE)
mspe_reduced_mean <- mean(mspe_reduced, na.rm = TRUE)
```

<u>**MSPE values**</u> for both the full and reduced models

- Full Model

```
## MSPE for Full Model: 10.63511
```

- Reduced Model

```
## MSPE for Reduced Model: 13.32577
```

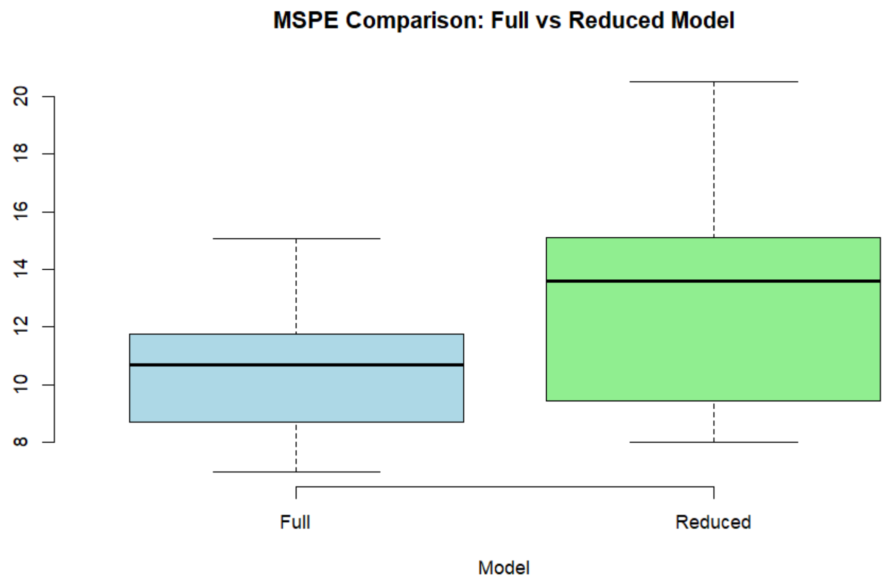
Comparing change in MSPEs (Full to Reduced)

```
## Relative increase in MSPE (%): 25.29973
```

Visualising MSPEs with Box Plots

```
# Combining MSPEs into a data frame for plotting
mspe_data <- data.frame(
  MSPE = c(mspe_full, mspe_reduced),
  Model = factor(rep(c("Full", "Reduced"), each = k))
)

# Generating box plots
boxplot(MSPE ~ Model, data = mspe_data,
        main = "MSPE Comparison: Full vs Reduced Model",
        ylab = "Mean Squared Prediction Error (MPa²)",
        col = c("lightblue", "lightgreen"),
        border = "black")
```



Comparing the models

From the cross-validation exercise above, The MSPE for the reduced model is substantially higher (`25.29973%`) than the full model. Therefore, the variable, `wcr`, adds predictive power and the full model is preferable for prediction purposes.

The full model with the variable, `wcr`, is therefore recommended.

1.1.6 Part e): Prediction

```r
predict(conc_model, newdata = conc.str, interval = "pred")
```

```
##        fit      lwr      upr
## 1 11.62271 5.324389 17.92103
```

The model predicts a mean of `11.62271` MPa. The prediction interval spans over 12.6 MPa (from `5.324389` to `17.92103`) which reflects high variability in strength for a single batch given the inputs. For structural design, this constitutes a very large uncertainty and the mix may not consistently meet design requirements.

Practically, this result is not fully reliable for decision-making about a specific batch without further testing or improving the model.

1.2 .Question 2

Energy consumption data from 80 office buildings

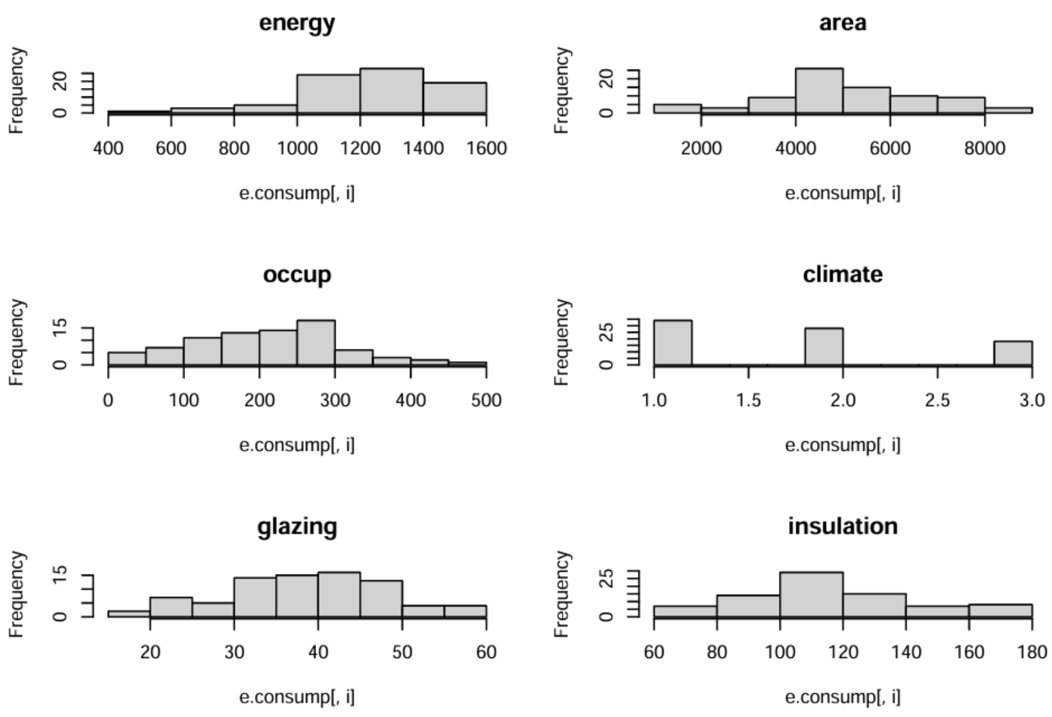
Question 2: # Energy consumption data from 80 office buildings

```
pacman::p_load(tidymodels)

# Getting started with the dataset in energy.csv :
e.consump <- read.csv(file.choose(), header = TRUE, na.strings = c("NA"))
head(e.consump) # View first few rows of the dataset
```

```
##   energy area occup climate glazing insulation
## 1 1083.5 1887   174       2    47.2      108.5
## 2 1560.9 5445   331       1    41.8      101.4
## 3 1103.5 5576   246       1    24.2      115.3
## 4 1239.7 6304   132       3    47.9      124.9
## 5 1423.2 5749   260       1    32.7       61.7
## 6 1056.0 4778   102       1    49.3       79.0
```

```
# View variables
par(mfrow=c(3,2))
for (i in 1:6) hist(e.consump[,i], main=names(e.consump)[i])
```




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

1.2.1 Part a): Multicollinearity

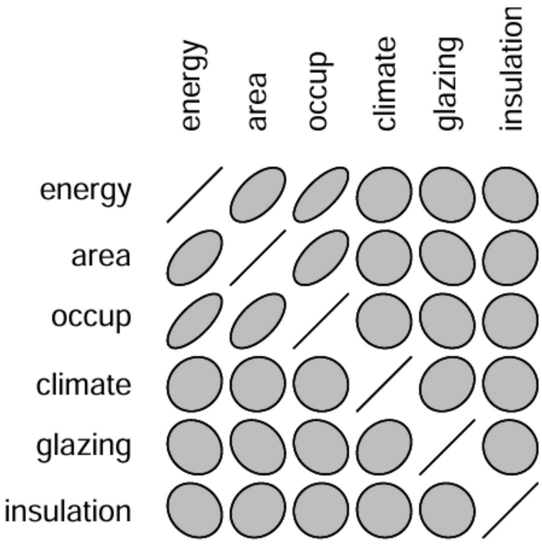
1.2.1.1 Pearson correlation coefficients

```
# (i) Pearson correlation coefficients
cor(e.consump, method = "pearson")
```

```
##                energy        area       occup     climate    glazing
## energy      1.0000000  0.56727188  0.71535501  0.12451307 -0.1348882
## area        0.5672719  1.00000000  0.60076867  0.03597627 -0.2360775
## occup       0.7153550  0.60076867  1.00000000 -0.03118426 -0.1716492
## climate     0.1245131  0.03597627 -0.03118426  1.00000000  0.2001212
## glazing    -0.1348882 -0.23607748 -0.17164920  0.20012116  1.0000000
## insulation -0.1681677  0.13148613  0.02944892 -0.03287315 -0.0447956
##             insulation
## energy     -0.16816767
## area        0.13148613
## occup       0.02944892
## climate    -0.03287315
## glazing    -0.04479560
## insulation  1.00000000
```

1.2.1.2 Ellipse plot to visualise collinearity

```
# (ii) An ellipse plot to visualise collinearity
pacman::p_load(ellipse)
plotcorr(cor(e.consump))
```



1.2.1.3 Variance Inflation Factors (VIFs)

```
# (iii) Variance Inflation Factors (VIFs)
pacman::p_load(car)
engy_model <- lm(energy ~ area + occup + climate + glazing + insulation, data = e.consump)
vif(engy_model)
```

```
##       area     occup   climate   glazing insulation
##   1.661848  1.579343  1.055096  1.111013   1.023478
```

Commenting on Multicollinearity

In the correlogram (ellipse plot), narrow/elongated ellipses indicate stronger correlation. Energy has elongated ellipses with area (`0.5672719`) and occupancy (`0.71535501`), indicating moderate to strong positive correlation. Also, area and occupancy are noticeably correlated with narrow tilted ellipse (`0.60076867`) which indicates collinearity. Therefore, there is some multicollinearity between area and occupancy, and to a lesser extent between energy and these two variables.

Since all VIF values are very well below 5, there is no significant multicollinearity among the predictors for the model, `engy_model`. This suggests that the predictors can be considered independent of each other for this regression model.

1.2.2 Part b): Model and Predictor Linearity

1.2.2.1 Initial Model Output, Adequacy & Appropriateness of Fit

Multiple Regression Model

```
# Initial Model Output
summary(engy_model)
```

```
##
## Call:
## lm(formula = energy ~ area + occup + climate + glazing + insulation,
##     data = e.consump)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -522.35  -74.40   11.52   93.61  367.70
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 936.47055  115.55537   8.104 8.23e-12 ***
## area          0.03186    0.01257   2.534  0.01338 *
## occup         1.26073    0.19992   6.306 1.87e-08 ***
## climate      36.38855   21.04601   1.729  0.08798 .
## glazing      -0.32620    1.73189  -0.188  0.85112
## insulation   -1.73568    0.60284  -2.879  0.00521 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 143.1 on 74 degrees of freedom
## Multiple R-squared:  0.6042, Adjusted R-squared:  0.5774
## F-statistic: 22.59 on 5 and 74 DF,  p-value: 1.101e-13
```

The model is [`energy ~ area + occup + climate + glazing + insulation`]

a) **Regression Coefficients**

The **slope** coefficients in the `engy_model` summary above indicate the respective change (increase [+] or decrease [-]) in energy consumption when each of the predictors increase by 1 unit while all other predictors remain unchanged.

- The p-values determine whether the different response-predictor relationships are statistically significant. Only 3 predictors (`area`, `occup`, and `insulation`) have p-values are all below 0.05 (where we reject the null hypothesis on a 5% significance level) Therefore, these variables significantly affect energy consumption. A zero slope coefficient is plausible for the other predictors (`climate` and `glazing`). Hence, they likely do not affect energy consumption

The **intercept** coefficient corresponds to the estimated (theoretical) energy consumption value when all the predictors are equal to zero.

- It's p-value (`8.23e-12`) is statistically significant at the 5% level, and an intercept of zero is not plausible.
- Although interpreting this is not practically rational, it ensures the regression hyperplane fits the data best within the observed predictor values range. It is not meaningful to extrapolate the predictors to zero.

b) **Model Significance**

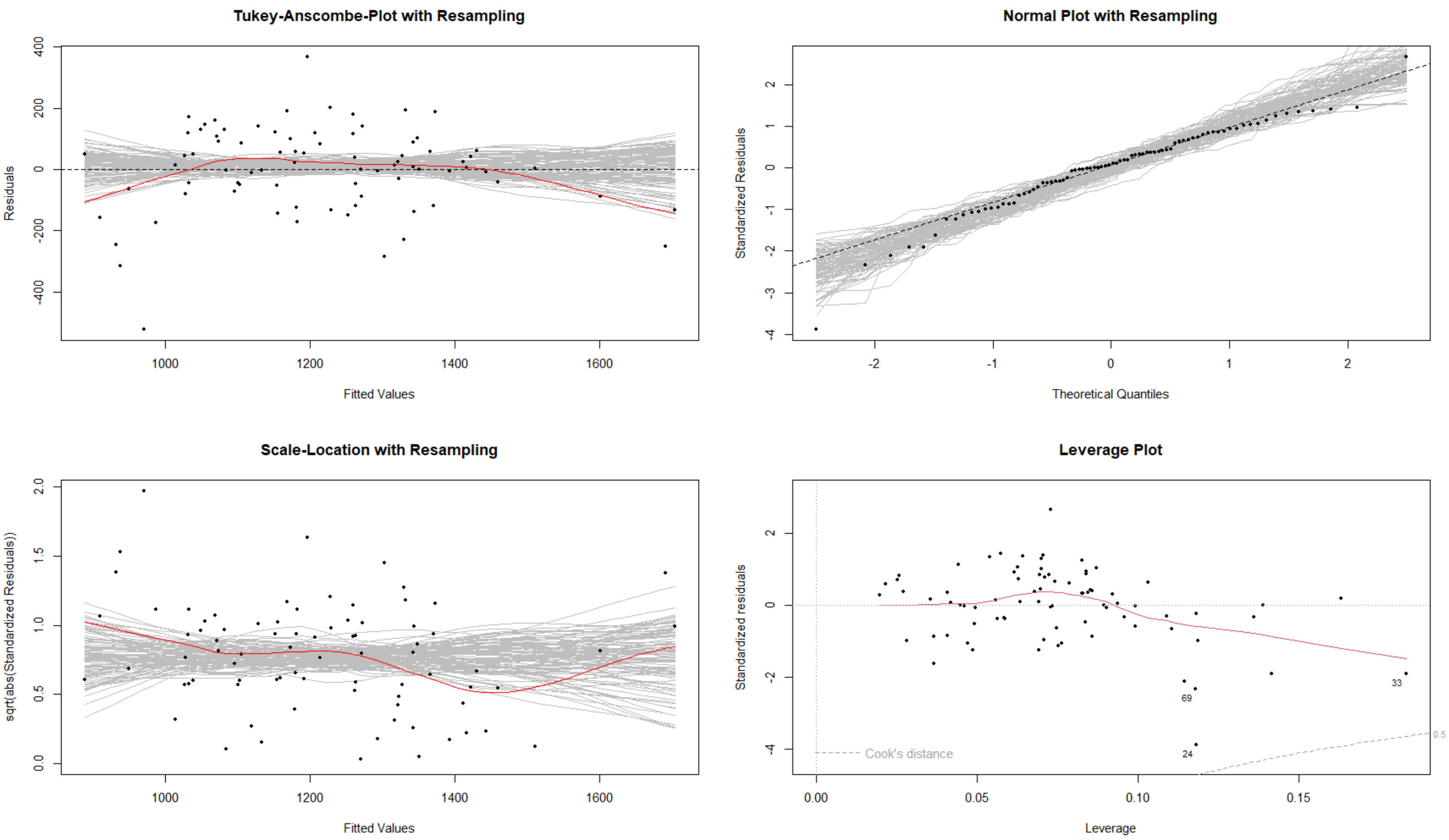
From the summary (the global F-Statistic), we gather that p-value is very small (`1.101e-13`) and that the model is significant at the 5% level.

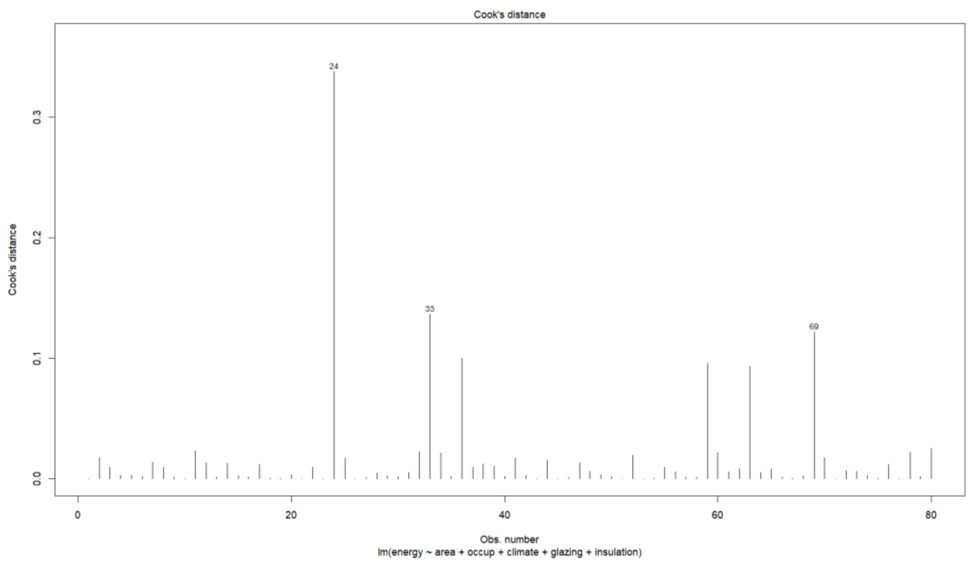
c) **Adequacy of Fit [R^2]**

The R-squared from summary (`engy_model`) indicates how much variation in energy consumption is explained by the five predictors as per the regression hyperplane. Here, multiple R^2 = `0.6042`, (the adjusted R^2 = `0.5774`), meaning that 61% of the variation in energy consumption is explained by predictors (`area`, `occup`, `climate`, `glazing`, and `insulation`), while the remaining 39% is due to other factors not included in the model.

d) **Appropriateness of Fit [Model Diagnostics]**

Residual Plots





(i) Linearity: E [E_i] = 0

The Tukey-Anscombe residual plot shows that the smoother noticeably deviates from the x-axis at low and high fitted values. From the resampling approach by the R function, `resplot()`, this deviation may be attributed to randomness because the original red smoother is within what can be generated by random sampling. We accept the linearity assumption.

(ii) Homoskedasticity, Var (E_i) = σ^2_E

From the Scale-Location plot, the red smoother is generally horizontal and the slight kink (between 1400 and 1600 of the fitted values) can be considered random because the smoother line is well within the resampling confidence region. There is no worrying heteroscedasticity.

(iii)No Correlation: Cov (E_i, E_j) = 0

The energy dataset observations are not affected by temporal or spatial variation. Thus, the errors can be considered independent and uncorrelated.

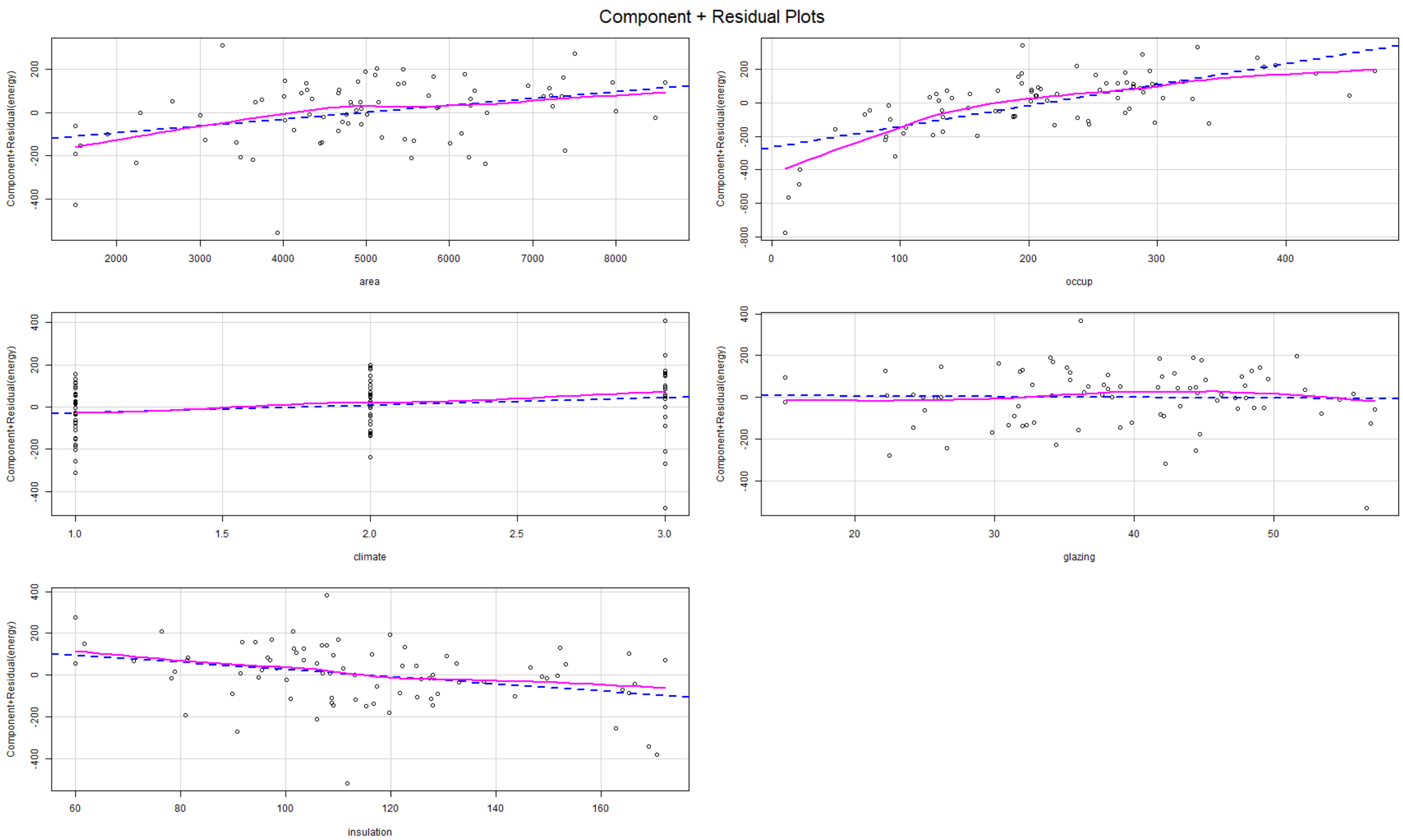
(iv)Normality: $E_i \sim N(0,\sigma^2_E)$

From the Normal Q-Q Plot, the bulk of the residuals (largely in the central region) are approximately normally distributed. There are some outliers at both tails which may imply departure from normality. All residuals from this dataset fall within the resampling confidence region, which means that deviations are random. The normality. assumption holds.

Summary: The model is also **appropriate** because of its associated residual plots are acceptable. The R^2 value (`0.6042`) implies that the regression model (hyperplane) is **adequate** because it accounts for a large portion of the total variation in the energy consumption.

1.2.2.2 Predictor Linearity

The partial residual plots are shown for the initial/original model (`engy_model`).



From the partial plots of the initial/original (`engy_model`) above, predictors, the variables `area` and `occupancy` clearly deviate from the blue dotted line which indicates non-linearity.

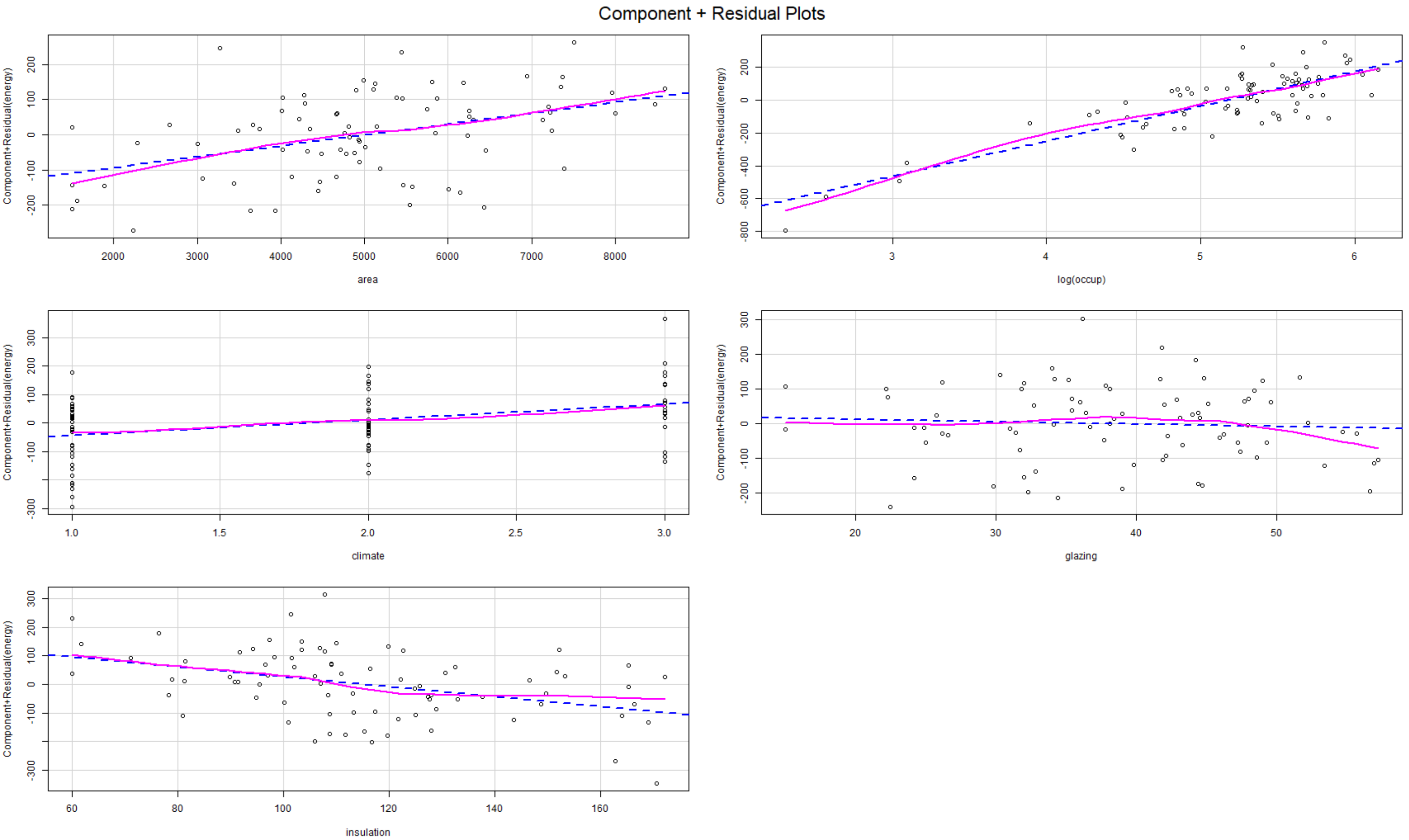
1.2.2.3 Transformed Model, Adequacy & Appropriateness of Fit

Transformed Model 1 (`engy_model2`). Here the `occup` is log-transformed

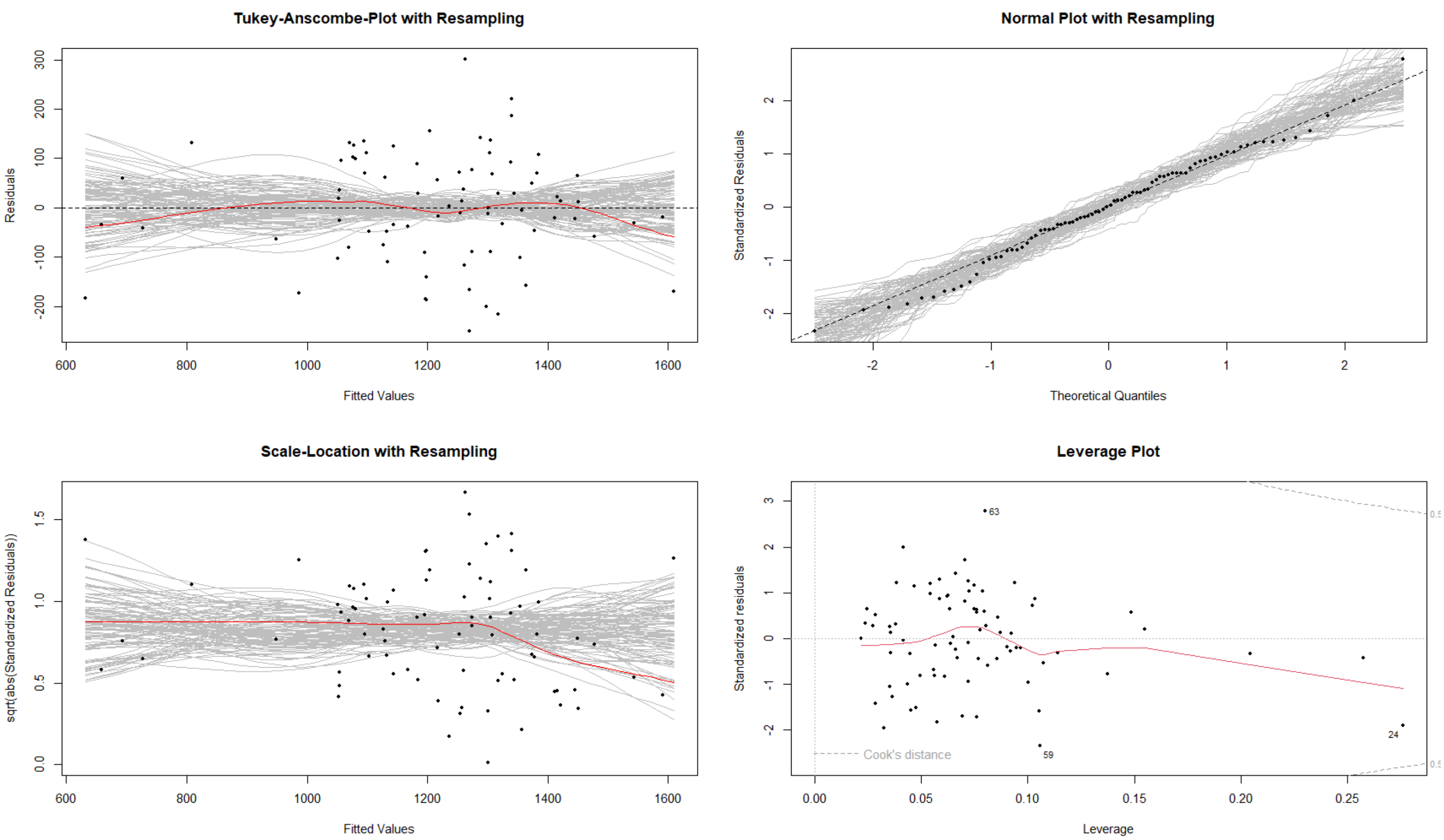
```
# Transformed Model 1
engy_model2 <- lm(energy ~ area + log(occup) + climate + glazing + insulation, data = e.consump)
summary(engy_model2)
```

```
##
## Call:
## lm(formula = energy ~ area + log(occup) + climate + glazing +
##     insulation, data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -250.129  -66.554    1.599   72.610  301.157
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  89.134103 127.287741   0.700 0.485963
## area          0.031008   0.009216   3.365 0.001217 **
## log(occup)  212.531379  20.365584  10.436 3.42e-16 ***
## climate      55.078048  16.765693   3.285 0.001559 **
## glazing      -0.694371   1.365478  -0.509 0.612602
## insulation   -1.744152   0.474978  -3.672 0.000452 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.9 on 74 degrees of freedom
```

Partial Plots for `engy_model2`



Residual Plots for `engy_model2`

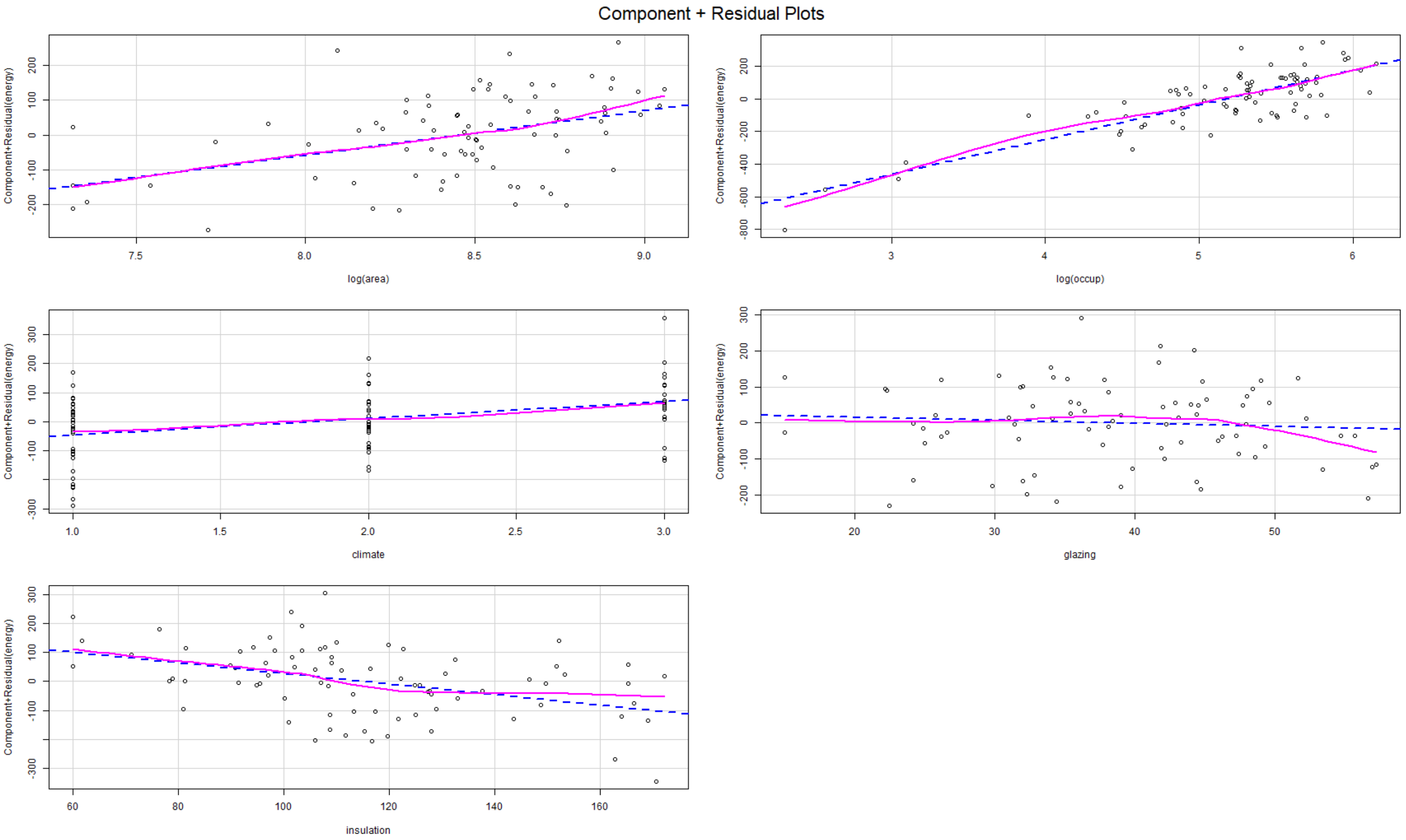


<u>Transformed Model 2</u> (`engy_model3`). In this model, `area` and `occup` are log-transformed

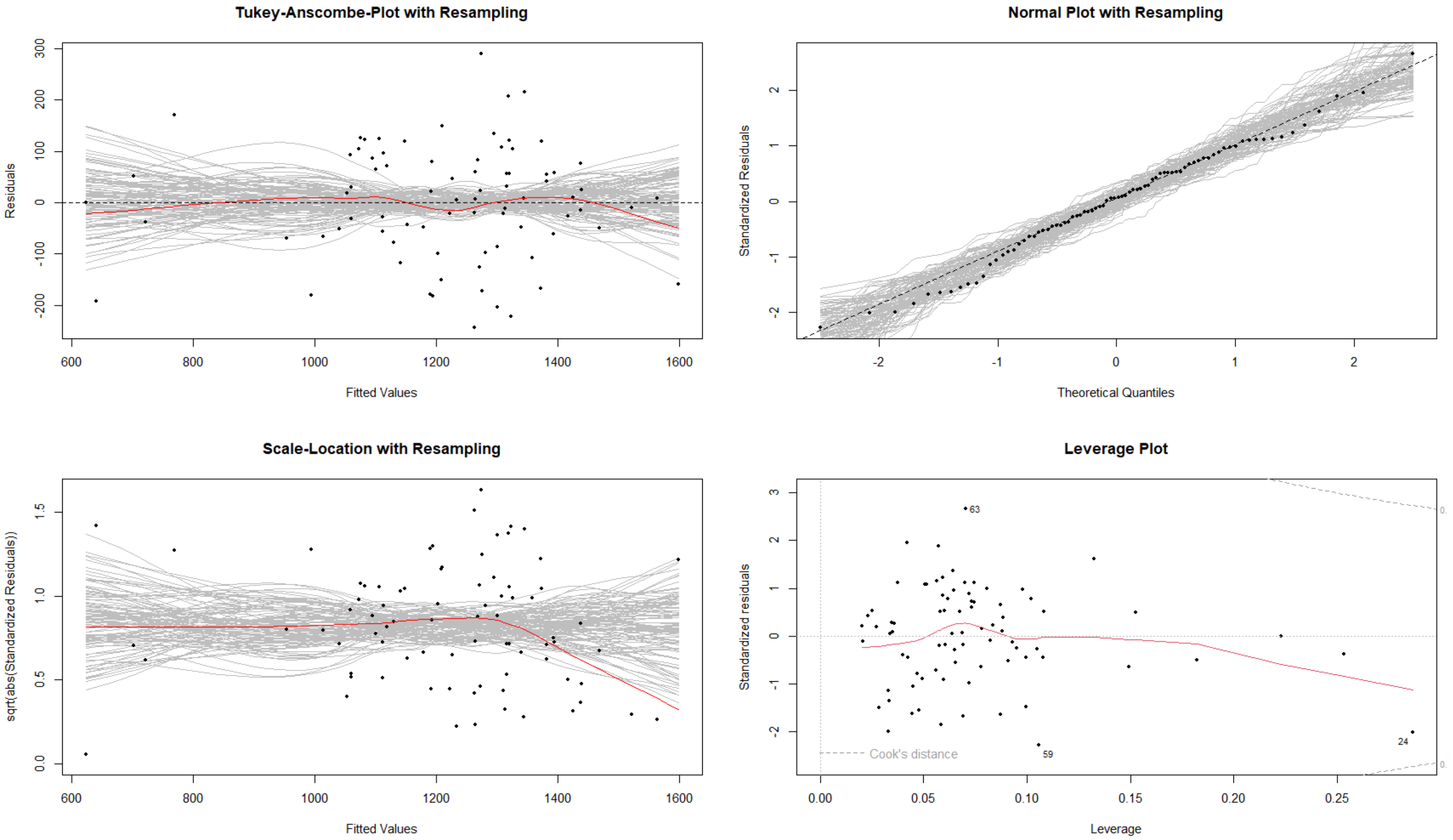
```r
# Transformed Model 2
engy_model3 <- lm(energy ~ log(area) + log(occup) + climate + glazing + insulation, data = e.consump)
summary(engy_model3)
```

```
##
## Call:
## lm(formula = energy ~ log(area) + log(occup) + climate + glazing +
##     insulation, data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -243.595  -62.706    6.811   77.199  290.031
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -829.3248   295.0838  -2.810 0.006327 **
## log(area)    128.5629    38.1860   3.367 0.001209 **
## log(occup)   212.6420    20.3453  10.452 3.19e-16 ***
## climate       56.9442    16.6877   3.412 0.001047 **
## glazing       -0.8729     1.3548  -0.644 0.521390
## insulation    -1.8293     0.4790  -3.819 0.000276 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.9 on 74 degrees of freedom
```

Partial Plots for `engy_model3`



Residual Plots for `engy_model3`



<u>Commenting on Model outputs, adequacy of fit and appropriateness of fit</u>.

In the first transformed model (`engy_model2`), the linearity of both variables are seen to improve. Also, the model diagnostics (appropriateness of fit) are much better for this transformed model. From the Adjusted R^2, this model also fits the data better (`0.7371`) than the original/initial model (`0.5774`).

In the second transformed model (`engy_model3`), the variable linearity, residual plots (appropriateness of fit) and model fit are better than both the original and the first transformed model (`engy_model2`)

Therefore, this model (`engy_model3`), is taken as the most appropriate in this case.

1.2.3 Part c): Variable Selection

Starting from the appropriately transformed model (`engy_model3`).

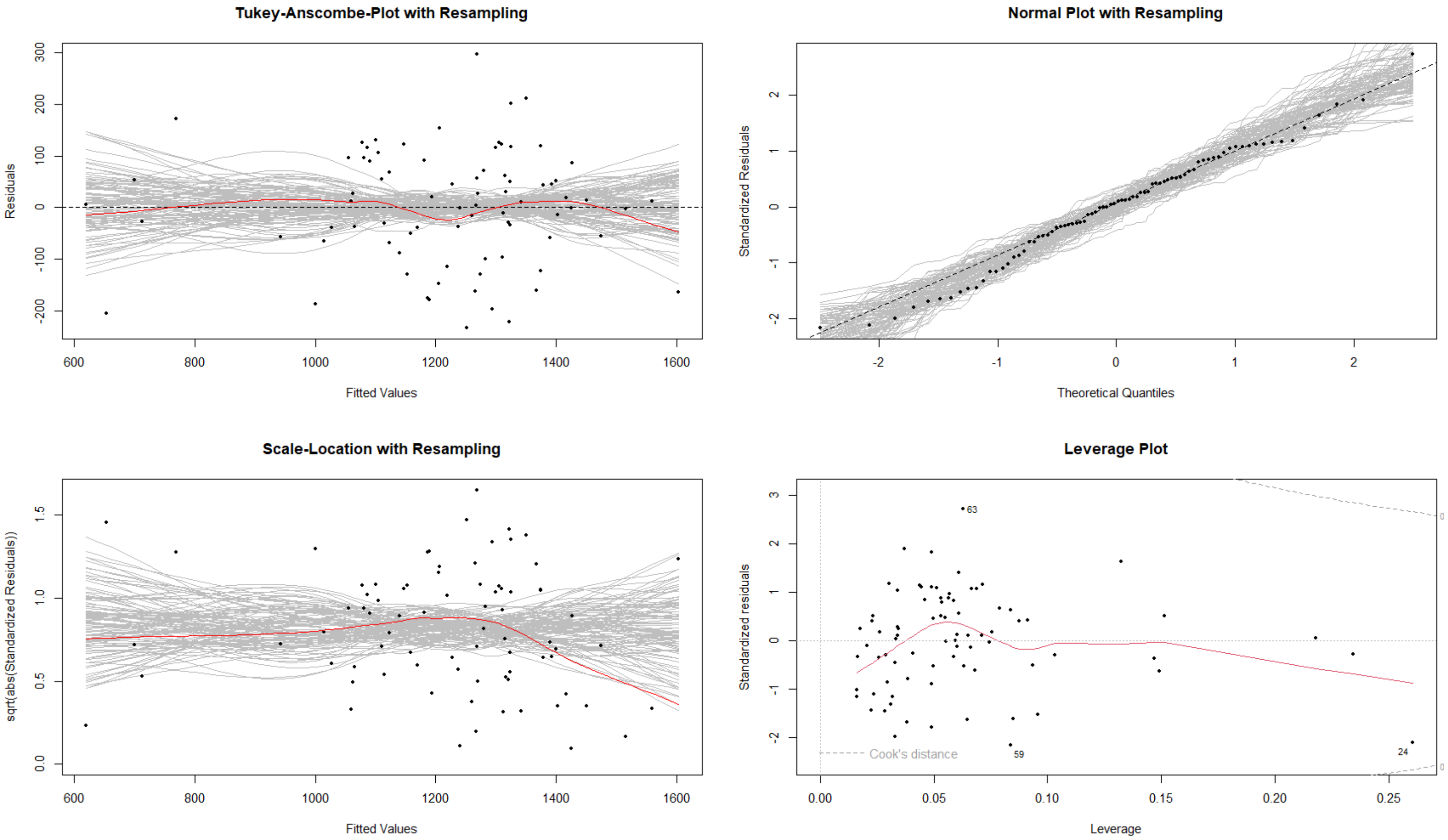
d) **Backward Elimination Model** (`engy.back`)

```
# Backward Elimination with AIC
engy.back <- stats::step(engy_model3, direction="backward")
```

```
summary(engy.back)
```

```
##
## Call:
## lm(formula = energy ~ log(area) + log(occup) + climate + insulation,
##     data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -232.755  -60.103    8.202   75.275  296.391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -897.3616   274.4641  -3.270 0.001628 **
## log(area)    132.9696    37.4216   3.553 0.000662 ***
## log(occup)   212.8157    20.2640  10.502  < 2e-16 ***
## climate       54.7563    16.2747   3.365 0.001211 **
## insulation    -1.8288     0.4771  -3.833 0.000261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.4 on 75 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7392
## F-statistic: 56.99 on 4 and 75 DF,  p-value: < 2.2e-16
```

Model Diagnostics (Residual Plots) for `engy.back`



e) **AIC Stepwise Models** [`engy.b1`, `engy.b2`, and `engy.b3`]

```r
# AIC Stepwise Model Search: Both Directions Approach
# starting with the null model
engy_null <- lm(energy ~ 1, data = e.consump) # Intercept-only model
sc <- list(lower=engy_null, upper=engy_model3)
engy.b1 <- stats::step(engy_null, scope = sc, direction = "both")
```

Model `engy.b1`

```r
summary(engy.b1)
```

```
##
## Call:
## lm(formula = energy ~ log(occup) + climate + insulation + log(area),
##     data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -232.755  -60.103    8.202   75.275  296.391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -897.3616   274.4641  -3.270 0.001628 **
## log(occup)   212.8157    20.2640  10.502  < 2e-16 ***
## climate       54.7563    16.2747   3.365 0.001211 **
## insulation    -1.8288     0.4771  -3.833 0.000261 ***
## log(area)    132.9696    37.4216   3.553 0.000662 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.4 on 75 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7392
## F-statistic: 56.99 on 4 and 75 DF,  p-value: < 2.2e-16
```

Model `engy.b2`

```r
summary(engy.b2)
```

```
##
## Call:
## lm(formula = energy ~ log(area) + log(occup) + climate + insulation,
##     data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -232.755  -60.103    8.202   75.275  296.391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -897.3616   274.4641  -3.270 0.001628 **
## log(area)    132.9696    37.4216   3.553 0.000662 ***
## log(occup)   212.8157    20.2640  10.502  < 2e-16 ***
## climate       54.7563    16.2747   3.365 0.001211 **
## insulation    -1.8288     0.4771  -3.833 0.000261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.4 on 75 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7392
## F-statistic: 56.99 on 4 and 75 DF,  p-value: < 2.2e-16
```

Model `engy.b3`

```r
summary(engy.b3)
```

```
##
## Call:
## lm(formula = energy ~ climate + log(occup) + insulation + log(area),
##     data = e.consump)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -232.755  -60.103    8.202   75.275  296.391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -897.3616   274.4641  -3.270 0.001628 **
## climate       54.7563    16.2747   3.365 0.001211 **
## log(occup)   212.8157    20.2640  10.502  < 2e-16 ***
## insulation    -1.8288     0.4771  -3.833 0.000261 ***
## log(area)    132.9696    37.4216   3.553 0.000662 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 112.4 on 75 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7392
## F-statistic: 56.99 on 4 and 75 DF,  p-value: < 2.2e-16
```

Comparing results

In all the reduced models from applying variable selection (i.e., `engy.back`, `engy.b1`, `engy.b2` and `engy.b3`), the variable, `glazing`, was dropped.

There are no major improvements in residual plots for all the models (here, only plots for the model `engy.back` are shown). Also, no noticeable changes (improvements) on the remaining predictor significance or model fit as compared to the full transformed model (`engy_model3`).

1.2.4 Part d): 5-fold cross-validation & MSPE

Compute MSPE for both the full and the reduced model. Which performs better for prediction?

The 5-fold cross-validation loop code

```
set.seed(123) # Set seed for reproducibility
n <- nrow(e.consump) # Number of observations and folds
k <- 5 # Number of folds
sb <- round(seq(0, n, length = (k + 1)))  # Fold boundaries

# Initialize vectors to store MSPE for each model
mspe_full <- numeric(k)
mspe_reduced <- numeric(k)

# 5-fold cross-validation for full model (engy_model3)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]
  train <- (1:n)[-test]
  fit_full <- lm(energy ~ log(area) + log(occup) + climate + glazing + insulation, data = e.consump[train, ])
  pred_full <- predict(fit_full, newdata = e.consump[test, ])
  mspe_full[i] <- mean((e.consump$energy[test] - pred_full)^2, na.rm = FALSE)
}

# 5-fold cross-validation for reduced model (dropping glazing)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]  # Same fold split for comparability
  train <- (1:n)[-test]
  fit_reduced <- lm(energy ~ log(area) + log(occup) + climate + insulation, data = e.consump[train, ])
  pred_reduced <- predict(fit_reduced, newdata = e.consump[test, ])
  mspe_reduced[i] <- mean((e.consump$energy[test] - pred_reduced)^2, na.rm = FALSE)
}

# Calculate overall MSPE for each model
mspe_full_mean <- mean(mspe_full, na.rm = TRUE)
mspe_reduced_mean <- mean(mspe_reduced, na.rm = TRUE)
```

MSPE values for both the full and reduced models

- Full Model

```
## MSPE for Full Model: 15405.86
```

- Reduced Model

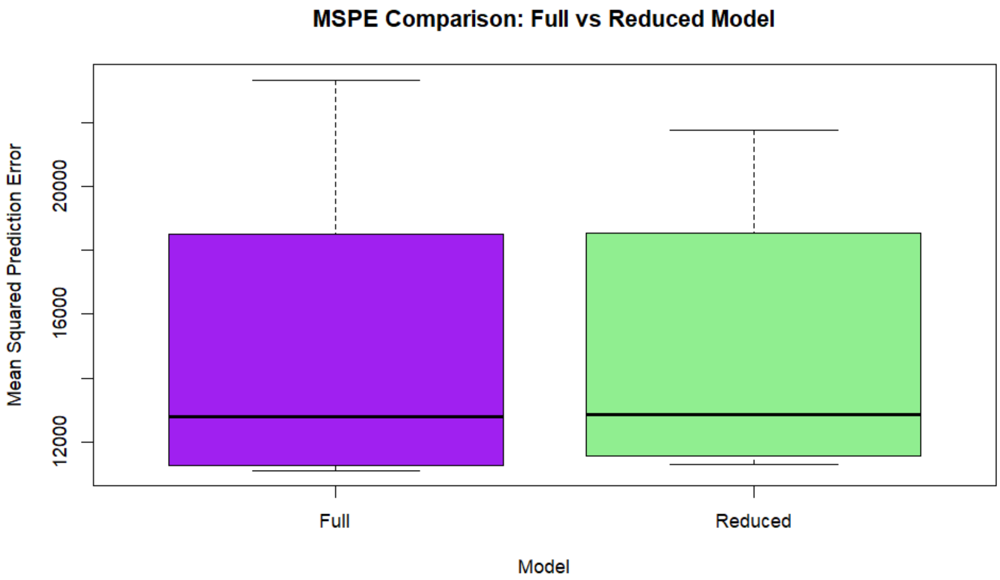
```
## MSPE for Reduced Model: 15209.33
```

- Comparing change in MSPEs (Full to Reduced)

```
## Relative increase in MSPE (%): -1.275735
```

Visualising MSPEs with Box Plots

```
# Combining MSPEs into a data frame for plotting
mspe_data <- data.frame(
  MSPE = c(mspe_full, mspe_reduced),
  Model = factor(rep(c("Full", "Reduced"), each = k))
)
# Generating box plots
boxplot(MSPE ~ Model, data = mspe_data,
        main = "MSPE Comparison: Full vs Reduced Model",
        ylab = "Mean Squared Prediction Error",
        col = c("purple", "lightgreen"),
        border = "black")
```



Comparing the models

From the cross-validation exercise, The MSPE for the *reduced* model is less (`-1.275735%`) than the *full* model. This implies that the variable, `glazing`, can be said to reduce the predictive power in model.

Thus, in this case, the reduced model is preferable for prediction purposes.

1.3 Question 3

Multiple Linear Regression *theory questions*

1.3.1 Q 3.1: MCQ Answer

B. Multicollinearity is present among the predictors.

1.3.2 Q 3.2: MCQ Answer

D. Cross-validation can help compare models based on predictive accuracy

2 Part 2: Analysis of Variance (ANOVA)

2.1 Question 4

Timber bending stiffness results for 3 species

Question 4: # Timber bending stiffness results

```
pacman::p_load(tidymodels)

# Getting started with the dataset in timber.csv :
timber <- read.csv(file.choose(), header = TRUE, na.strings = c("NA"))
# timber
head(timber)
```

```
##   species stiffness
## 1    pine    7897.6
## 2    pine    8239.5
## 3    pine    7740.3
## 4    pine    7722.1
## 5    pine    8982.9
## 6    pine    8696.7
```

```
## Convert species column to a factor
timber$species <- factor(timber$species)
## Check levels
levels(timber$species)
```

```
## [1] "cedar" "gum"   "pine"
```

```
print(summary_stats)
```

```
## # A tibble: 3 x 9
##   species  Mean    SD Median   IQR    Q1    Q3   Min    Max
##   <fct>   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 cedar   9288.  506.  9403.  638. 8934. 9571. 8220. 10075.
## 2 gum     9398.  607.  9365.  635. 9050. 9684. 8315. 11124.
## 3 pine    8156.  506.  8139.  728. 7806. 8534. 6999.  8983.
```

```
print("Outliers:")
```

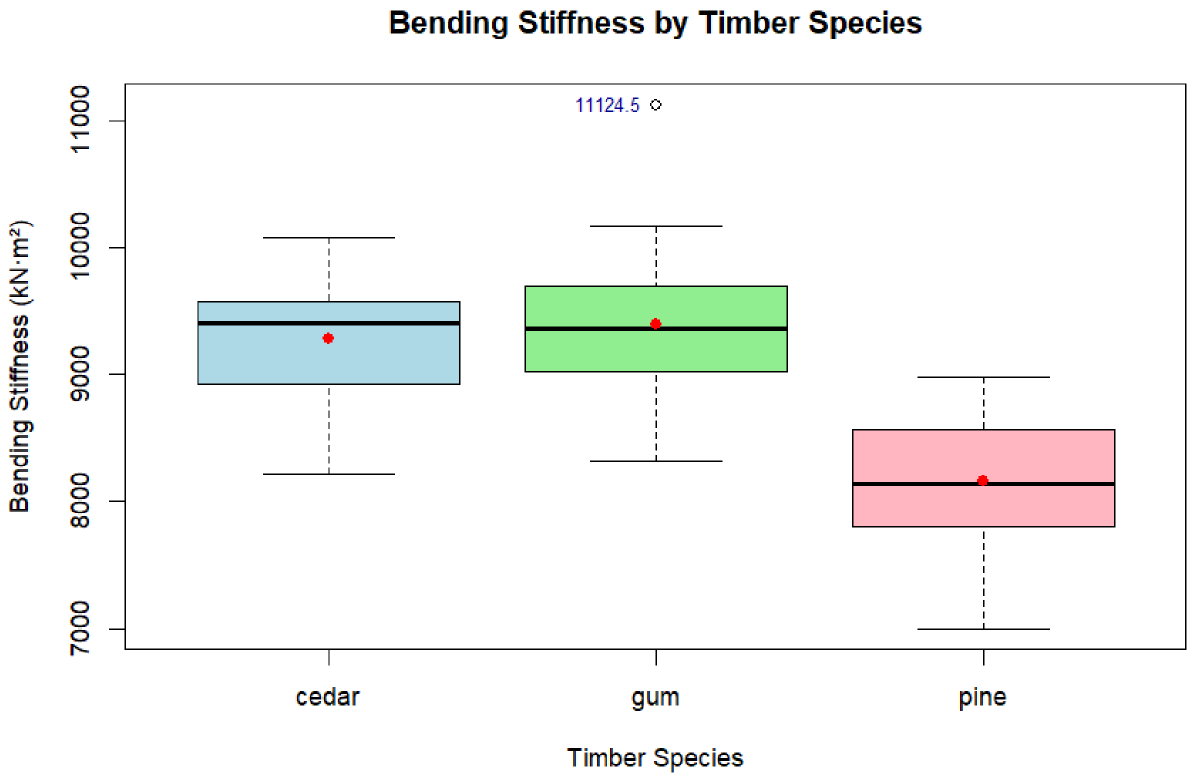
```
## [1] "Outliers:"
```

```
print(outliers)
```

```
## # A tibble: 1 x 2
## # Groups:   species [1]
##   species stiffness
##   <fct>       <dbl>
## 1 gum        11124.
```

2.1.1 Part a): Box Plots

```r
# Part a): # Box Plots
boxplot(stiffness ~ species,
        data = timber,
        main = "Bending Stiffness by Timber Species",
        xlab = "Timber Species",
        ylab = "Bending Stiffness (kN·m²)",
        col = c("lightblue", "lightgreen", "lightpink"),
        border = "black")
#Adding means as points
means <- tapply(timber$stiffness, timber$species, mean)
points(1:3, means, pch = 19, col = "red")
```



Commenting on Variability and Outliers

<u>Variability</u>

The variability of stiffness across species is assessed using the box plots and some summary statistics (standard deviation, range, and the interquartile range IQR).

As per standard deviation (SD), gum has the highest variability (`607.08` kN·m²), followed by pine (`506.4` kN·m²), and cedar has the lowest (`505.8` kN·m²). This suggests that gum's stiffness values are more spread out compared to pine and cedar.

Comparing the interquartile range (IQR), pine has the highest IQR (`728` kN·m²) which implies a slightly wider spread of the middle 50% of stiffness values compared to gum (`635` kN·m²) and cedar (`638` kN·m²). These differences in IQR are small and the spread central data across species is quite comparable.

The range (max - min) is largest for gum (`2809.6` kN·m²), followed by pine (`1983.7` kN·m²), and cedar (`1854.6` kN·m²). This reinforces that gum has the most extreme values.

From the box plot, cedar has the highest median stiffness (`9402.6` kN·m²), followed by gum (`9365.3` kN·m²), and pine (`8139.2` kN·m²) which indicates that gum and cedar generally have higher bending stiffness than pine.

Therefore, gum exhibits the highest variability in bending stiffness, as seen in its larger standard and range. This suggests less consistency in this property for gum compared to pine and cedar which have similar variability.

Outliers

From the box plot, only gum has an upper bound outlier (`11124.5` kN·m²) which means it can exhibit extreme (stronger) stiffness values.

2.1.2 Part b): A one-way ANOVA test

```
# Part b):# Fit a one-way ANOVA test
timber$species <- relevel(timber$species, ref = "gum")
options(contrasts = c("contr.sum", "contr.poly"))
# options(contrasts = c("contr.treatment", "contr.poly")) # used as default anyway

stiff <- aov(stiffness ~ species, data = timber)
summary(stiff) ## ANOVA table including F-test
```

```
##             Df   Sum Sq Mean Sq F value   Pr(>F)    
## species      2 18889629 9444815   32.17 4.45e-10 ***
## Residuals   57 16734248  293583                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Interpretation

The null hypothesis (H_o): $\mu_{pine} = \mu_{gum} = \mu_{cedar}$ (All species have the same mean bending stiffness)

The alternative hypothesis (H_A): At least one species has a different mean stiffness.

F-statistic: `32.17`; This is a large F-value which indicates that between-species variability is much greater than within-species variability.

p-value: `4.45e-10`; This extremely small (< 0.001) and so, we reject H_o at all the conventional significance levels like 0.05 and 0.01.

Conclusion: There is ***strong statistical evidence*** that the mean bending stiffness differs significantly between timber species.

2.1.3 Part c): A pairwise two-sample t-test

A pairwise two-sample t-test (with multiple comparison correction)

```
tapply(timber$stiffness, timber$species, var) # check for group var
```

```
##      gum    cedar     pine
## 368474.5 255812.0 256463.4
```

```
pairwise_results <- pairwise.t.test(timber$stiffness, timber$species,
                                    p.adjust.method = "bonferroni",
                                    pool.sd = FALSE, # Welch's t-test (unequal variances)
                                    paired = FALSE,  # Independent samples
                                    conf.level = 0.95)

# Print the results
print("Pairwise t-test results with Bonferroni correction:")
```

```
print(pairwise_results)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  timber$stiffness and timber$species
##
##       gum     cedar
## cedar 1       -
## pine  8.1e-08 6.0e-08
##
## P value adjustment method: bonferroni
```

Test Interpretation

Test method: Welch-adjusted pairwise t-test because the groups have unequal variances. This adjusts the degrees of freedom for each pair according to Welch's formula.

The *null hypothesis* (H_o): $\mu_{pine} = \mu_{gum}$ [or $\mu_{pine} = \mu_{cedar}$ or $\mu_{gum} = \mu_{cedar}$] (mean bending stiffness is the same) and the *alternative hypothesis* (H_A): Means differ. We reject H_o if $p < 0.05$.

Interpretation for each pair

pine vs gum: p = `8.1e-08` < 0.05 (significant) implying that mean stiffness differs between pine and gum, hence gum is stiffer than pine (looking at raw data: gum = `9398.3` kN·m² vs pine = `8156. 5` kN·m²).

pine vs cedar: p = `6.0e-08` < 0.05 (significant) implying that mean stiffness differs between pine and cedar hence cedar is stiffer than pine (cedar = `9287.5` kN·m²).

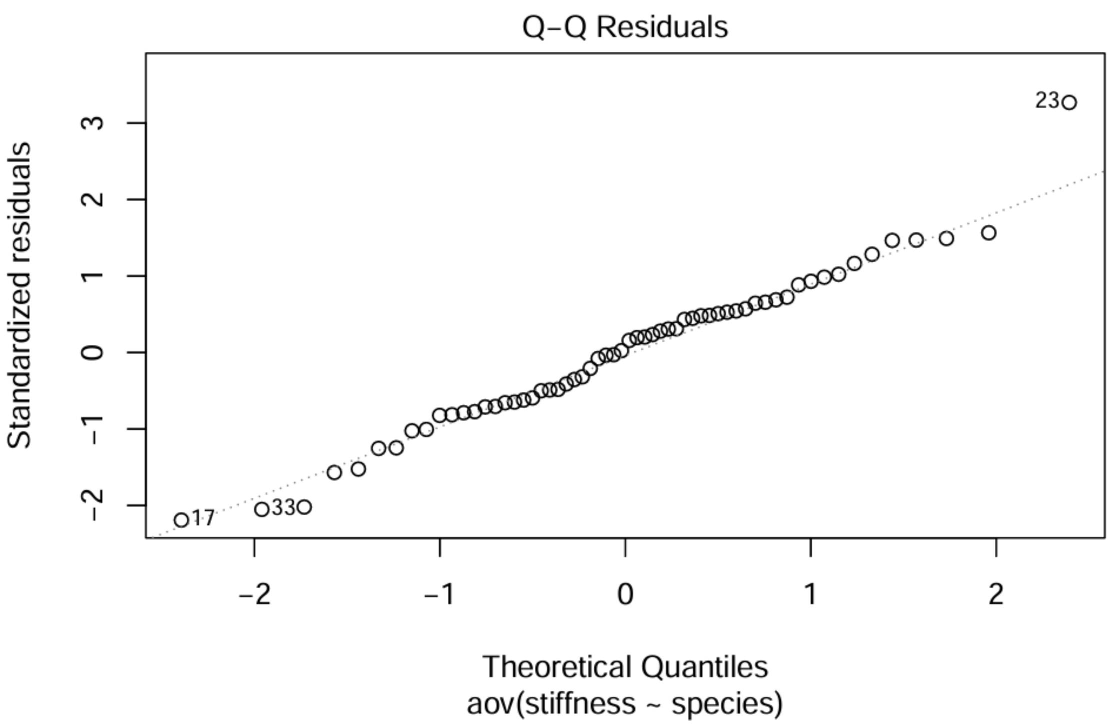
gum vs cedar: p = `1` > 0.05 (not significant) implying that there is no evidence that gum and cedar differ in mean stiffness. Their stiffness values are roughly similar (gum = `9398.3` kN·m², cedar = `9287.5` kN·m²).

Therefore, practically, pine is the softest whereas gum and cedar have similar higher stiffness.

2.1.4 Part d): Residual Diagnostics

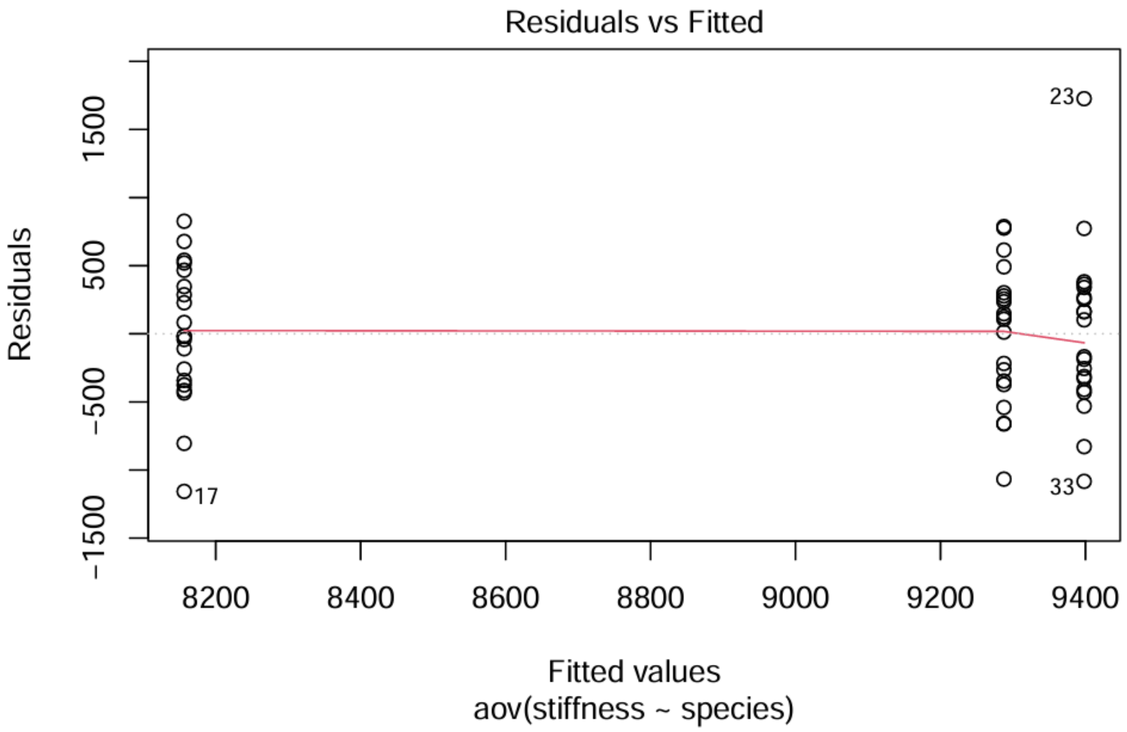
<u>Normal Q-Q Plot</u>

```
# Part d) ## Residual Diagnostics
plot(stiff, which = 2)
```



Tukey-Anscombe Plot

```
plot(stiff, which = 1)
```



Model Assumptions

From the residual plots, error variance is constant and error can be expected to be zero (Tukey-Anscombe Plot). Errors are *i .i. d.* (from Q-Q plot). No autocorrelation is present.

The ANOVA model meets the required assumptions.

2.2 Question 5

Compressive strength for concrete cured under different methods.

Question 5: # Compressive strength for concrete

```
pacman::p_load(tidymodels)

# Getting started with the dataset in curing.csv :
curing <- read.csv(file.choose(), header = TRUE, na.strings = c("NA"))

head(curing)
```

```
##   method strength
## 1  water     44.5
## 2  water     37.7
## 3  water     38.5
## 4  water     40.9
## 5  water     43.9
## 6  water     41.9
```

```
## Convert species column to a factor
curing$method <- factor(curing$method)
## Check levels
levels(curing$method)
```

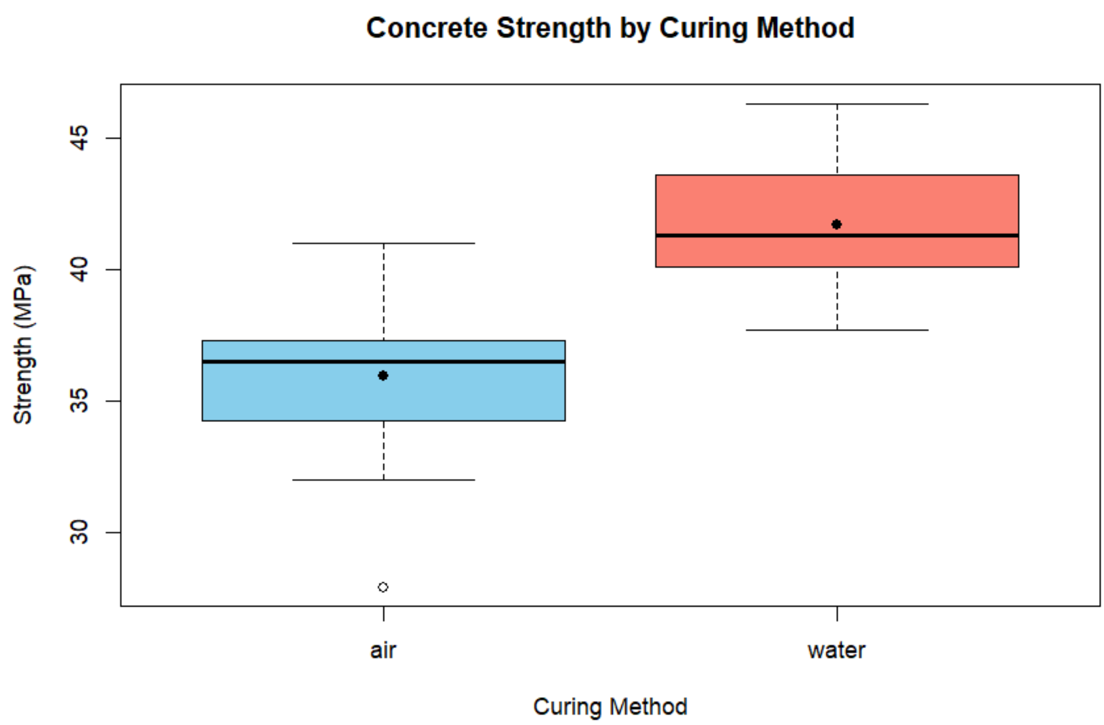
```
## [1] "air"   "water"
```

2.2.1 Part a): Box Plots

```
# Part a): # Box Plots
# Box plots

# Side-by-side boxplots of strength by curing method
boxplot(strength ~ method, data = curing,
        main = "Concrete Strength by Curing Method",
        xlab = "Curing Method",
        ylab = "Strength (MPa)",
        col = c("skyblue", "salmon"))

#Adding means as points
means <- tapply(curing$strength, curing$method, mean)
points(1:2, means, pch = 19, col = "black")
```



Inspecting Difference in Strength

From the box plots, the box for water is higher than the box for air. The Whiskers indicate that the upper range of air overlaps slightly with the lower range of water, but most water values are consistently higher.

Based on this, it appears likely that the two curing methods would produce significantly different strengths where water curing produces higher strengths than air curing.

2.2.2 Part b): A two-sample t-test

```
# Part b): # A two-sample t-test
tapply(curing$strengt, curing$method, sd) # check for group SD
```

```
##      air    water
## 3.296362 2.508234
```

```
tapply(curing$strengt, curing$method, var) # check for group var
```

```
##       air     water
## 10.866000  6.291238
```

```
# t.test(strength ~ method, data = curing, var.equal = TRUE)
t.test(strength ~ method, data = curing, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  strength by method
## t = -5.392, df = 26.141, p-value = 1.178e-05
## alternative hypothesis: true difference in means between group air and group water is not equal to 0
## 95 percent confidence interval:
##  -7.964463 -3.568871
## sample estimates:
##   mean in group air mean in group water
##            35.92000            41.68667
```

Model Interpretation

Test method: The Welch's t-test (does not assume equal variances).

The ***null hypothesis*** (H_O): $\mu_{water} = \mu_{air}$ (mean compressive strength is the same for both curing methods).

The ***alternative hypothesis*** (H_A): $\mu_{water} \neq \mu_{air}$ (mean compressive strength differs between the two curing methods).

2.2.3 Part c): Test statistic, p-value, and conclusion

Test Results

t-statistic: `-5.392`

p-value: `1.178e-05`

Conclusion: The p-value is much smaller than 0.05, so we reject the null hypothesis. There is *strong evidence* that the mean strengths **differ** between the two curing methods. Water curing results in significantly higher mean strength than air curing.

2.2.4 Part d): Practical significance

```
# Part d) # practical significance

Mean.diff <- 41.68667- 35.92000
print("Difference in means is:")
```

```
## [1] "Difference in means is:"
```

```
print(Mean.diff)
```

```
## [1] 5.76667
```

Water curing consistently produces higher strength than air curing across all samples. From the test output, the difference in mean approximately 5.8 (`41.68667 – 35.920`) and such an increase could be materially important in concrete performance. In construction, even small differences in concrete strength can affect structural safety, durability, or compliance with standards.

Therefore, the difference is both statistically significant (very low p-value) and practically significant because it represents a meaningful improvement in strength due to water curing

2.3 Question 6

Analysis Of Variance theory questions

2.3.1 Q 6.1 MCQ Answer

B. At least one species has a mean stiffness significantly different from the others.

2.3.2 Q 6.2 MCQ Answer

D. The sample sizes must be equal.

2.3.3 Q.6.3 MCQ Answer

A. The ratio of between-group variance to within-group variance.

2.3.4 Q6.4 MCQ Answer

B. ANOVA avoids increasing the risk of Type I error from multiple t-tests.