Energy and Insurance Cost Modelling

Ilke Kas

2025-04-(18-19)

Problem 1: Insurance Data

Attached is insurance.csv data file. This data consists of 1338 individual health insurance charges in different regional locations in the United States associated with 5 other predictors. The variable descriptions are given in Table 1.

Table 1: Insurance Data Variable Descriptions

No.	Variable	Description
1	charges	Insurance premium charge
2	age	Age (years)
3	sex	Sex (female/male)
4	bmi	Body mass index
5	children	Number of children
6	smoker	Smoking status (Y/N)
7	region	Location of the insured (northeast, northwest, southeast, southwest)

1. **Perform an exploratory analysis for this data.**

Solution:

```
# lets read the data first
insurance_data <- read.csv("insurance.csv")
head(insurance_data)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

As seen, we read the insurance data from the provided .csv file. There are 7 different columns. Let's look at the summary of this dataset:

```
# get the summary of the dataset
library(skimr)
skim(insurance_data)
```

Table 2: Data summary

Name	insurance_data
Number of rows	1338
Number of columns	7
Column type frequency:	
character	3
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
sex	0	1	4	6	0	2	0
smoker	0	1	2	3	0	2	0
region	0	1	9	9	0	4	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	39.21	14.05	18.00	27.00	39.00	51.00	64.00	
bmi	0	1	30.66	6.10	15.96	26.30	30.40	34.69	53.13	
children	0	1	1.09	1.21	0.00	0.00	1.00	2.00	5.00	
charges	0	1	13270.42	12110.01	1121.87	4740.29	9382.03	16639.91	63770.43	

In order to get the summary of this dataset, I used the skimr library since it is more organized compared to the summary. As seen from the data summary, the dataset contains 1,338 observations and 7 variables, including 3 categorical and 4 numeric features. There are no missing values, and all variables are fully complete. Among numeric variables, charges is highly skewed with a wide range (from ~1,122 to ~63,770), while age, bmi, and children are more symmetrically distributed. Categorical variables like sex, smoker, and region each have 2–4 unique levels, with no empty or whitespace issues.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

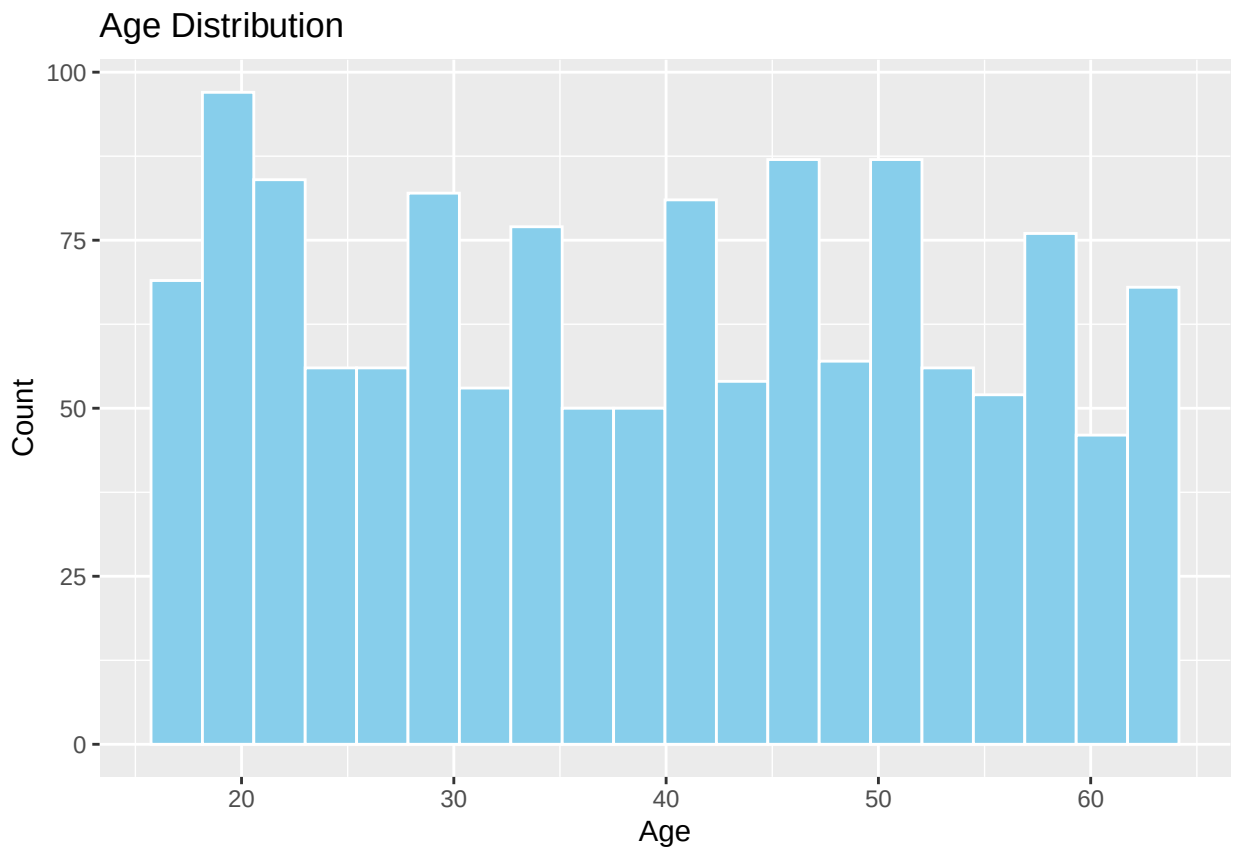
```
#lets check the categorical data values
categorical_data <- insurance_data %>%
  select(where(~ is.character(.x) || is.factor(.x)))
```

```r
for (col in names(categorical_data)) {
  cat("\n---", col, "---\n")
  print(table(categorical_data[[col]]))
  print(prop.table(table(categorical_data[[col]])))
}
```

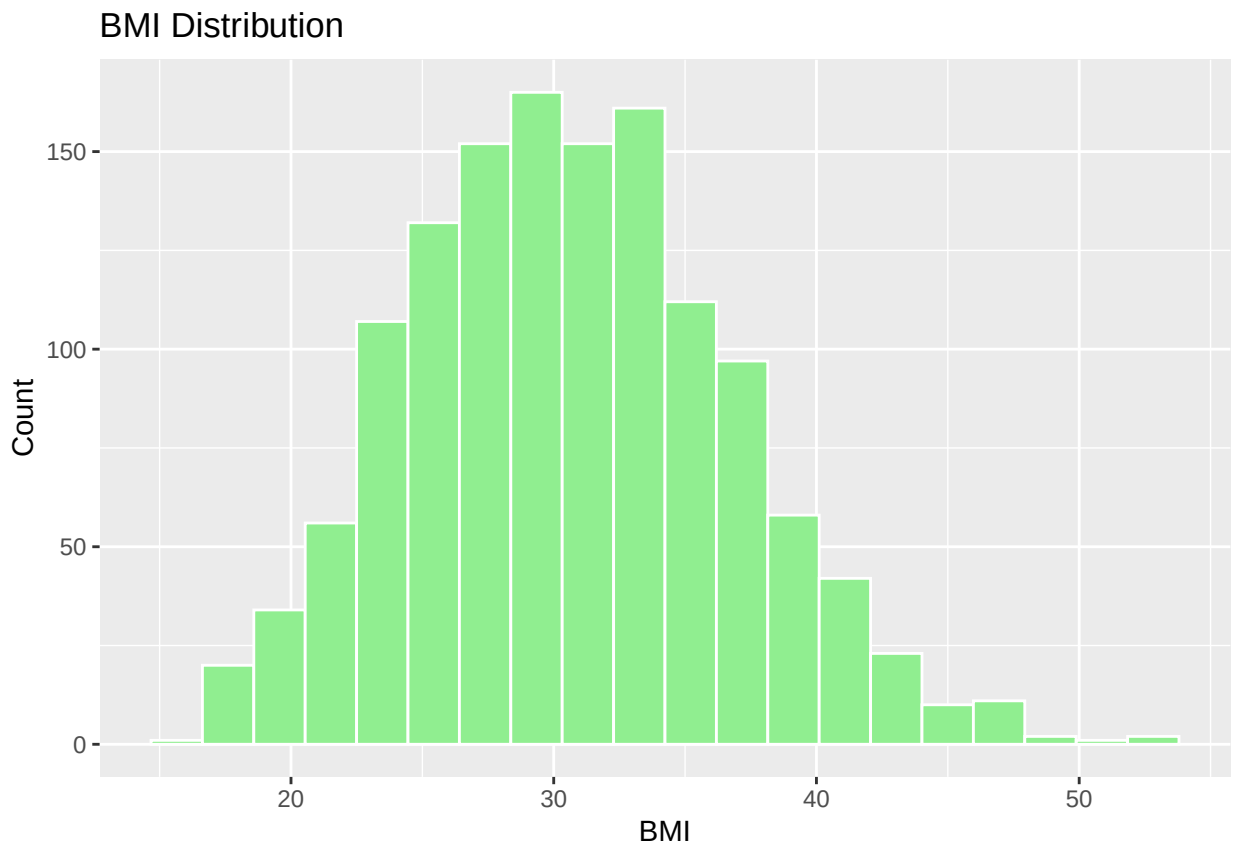
```
##
## --- sex ---
##
## female   male
##    662    676
##
##    female      male
## 0.4947683 0.5052317
##
## --- smoker ---
##
##   no  yes
## 1064  274
##
##        no       yes
## 0.7952167 0.2047833
##
## --- region ---
##
## northeast northwest southeast southwest
##       324       325       364       325
##
## northeast northwest southeast southwest
## 0.2421525 0.2428999 0.2720478 0.2428999
```

The dataset has a nearly equal distribution of sex, with 49.5% female and 50.5% male participants. A majority of individuals are non-smokers (79.5%), while only 20.5% are smokers. The region variable is fairly balanced across all four areas, with the southeast region having a slightly higher proportion (27.2%).

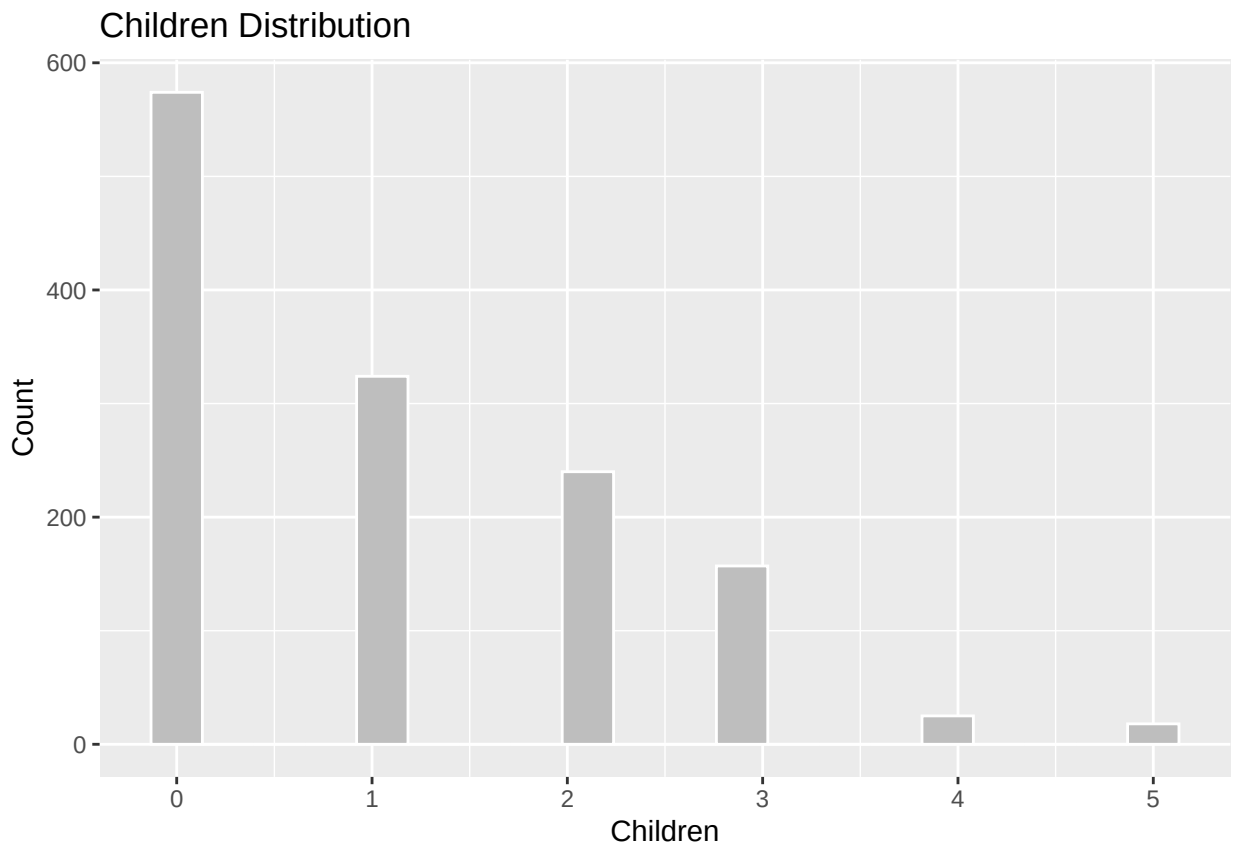
```r
# Lets check the histogram of the continuos variables
library(ggplot2)

# Example: Histogram for 'age'
ggplot(insurance_data, aes(x = age)) +
  geom_histogram(fill = "skyblue", color = "white", bins = 20) +
  labs(title = "Age Distribution", x = "Age", y = "Count")
```



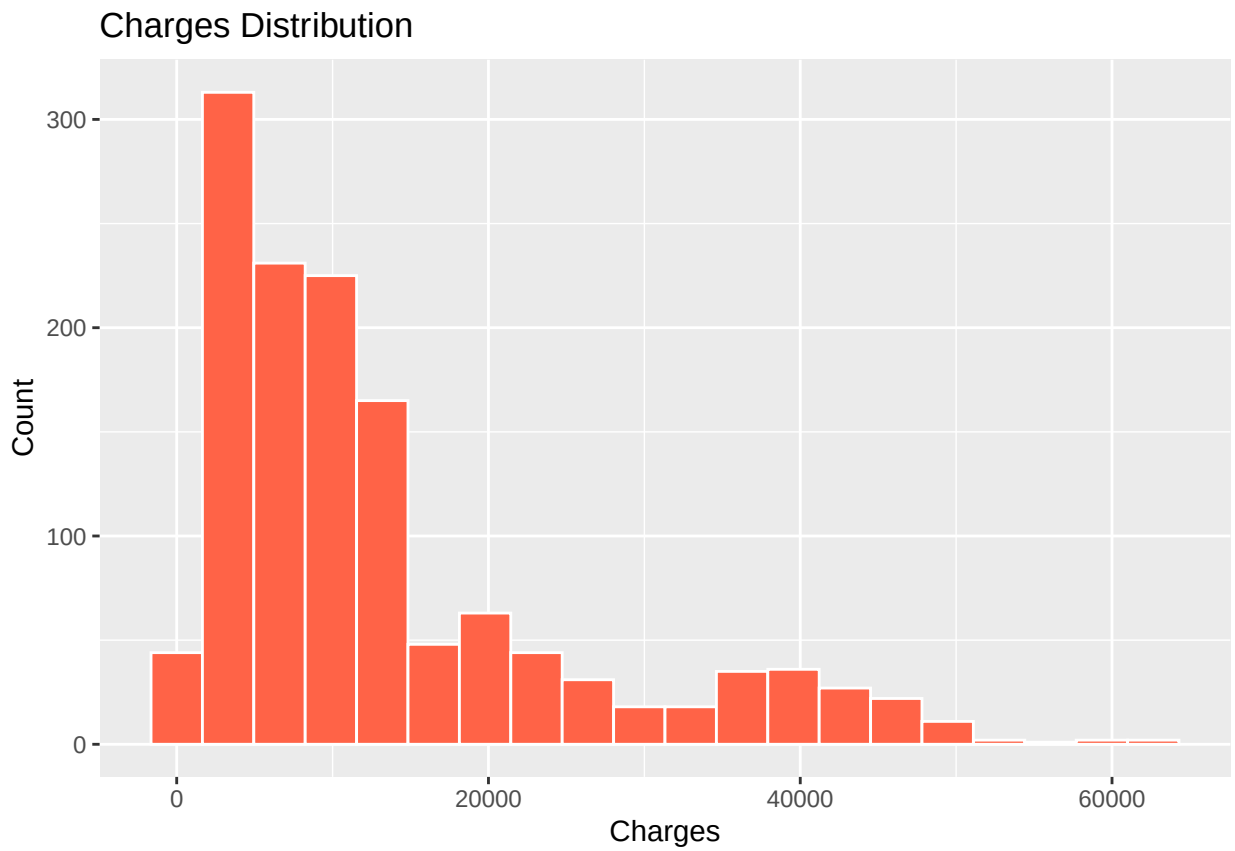

```
# Repeat for 'bmi' and 'charges'
ggplot(insurance_data, aes(x = bmi)) +
  geom_histogram(fill = "lightgreen", color = "white", bins = 20) +
  labs(title = "BMI Distribution", x = "BMI", y = "Count")
```




```
ggplot(insurance_data, aes(x = children)) +
  geom_histogram(fill = "gray", color = "white", bins = 20) +
  labs(title = "Children Distribution", x = "Children", y = "Count")
```




```
ggplot(insurance_data, aes(x = charges)) +
  geom_histogram(fill = "tomato", color = "white", bins = 20) +
  labs(title = "Charges Distribution", x = "Charges", y = "Count")
```



Age Distribution: The age variable is fairly uniformly distributed across the range of 18 to 64, with no strong skewness or concentration around a particular age group. This suggests a well-balanced representation across different age brackets in the dataset.

BMI Distribution: BMI shows a unimodal, right-skewed distribution centered around 30. Most individuals fall within the 25–35 BMI range. Based on my knowledge, high BMI indicates obesity (greater than 30). Therefore, this graph indicates a concentration in the overweight to mildly obese categories.

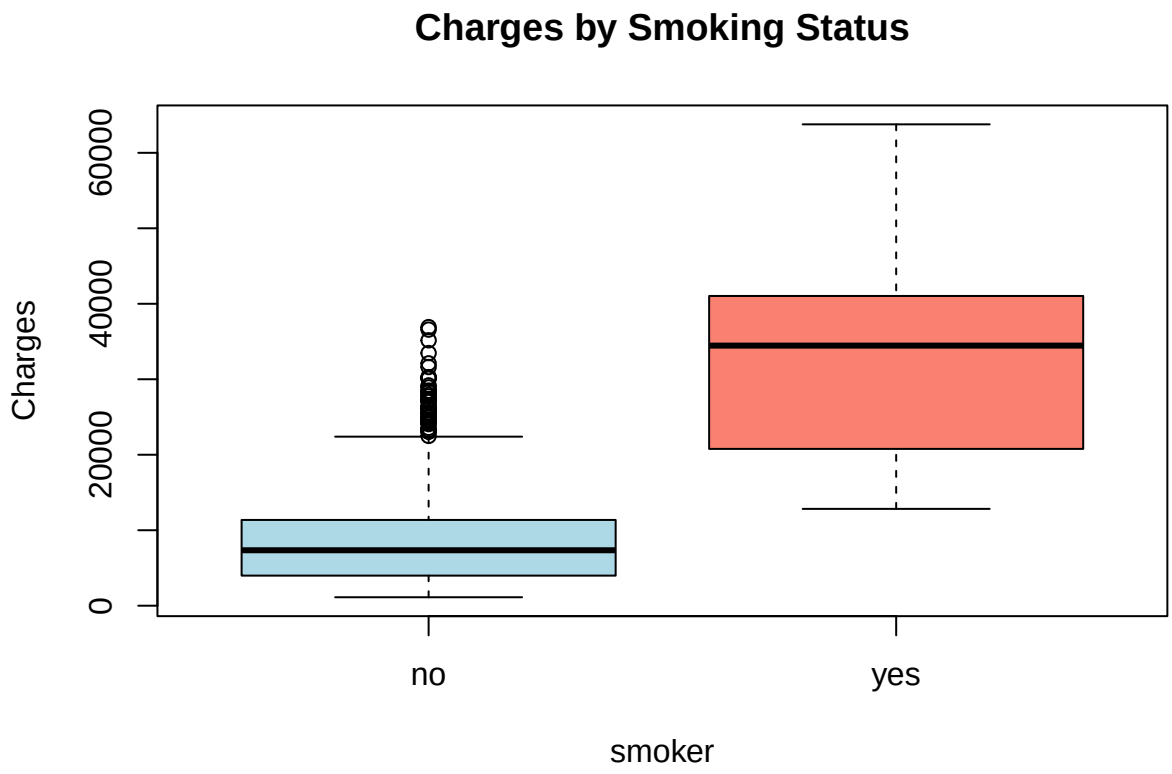
Children Distribution:

The number of children is heavily skewed toward zero, with the majority of individuals having no children. As the number of children increases, the count drops significantly, especially for 4 and 5 children, which are rare.

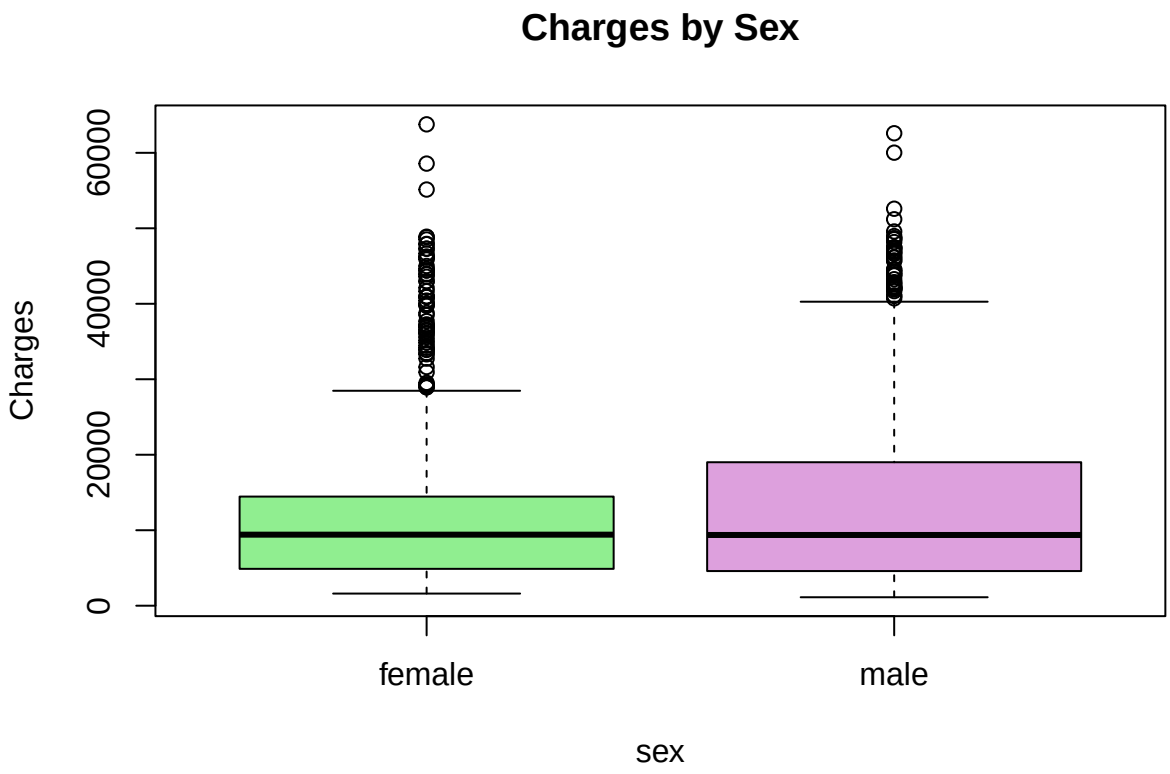
Charges Distribution: The distribution of insurance charges is highly right-skewed, with most individuals paying under $20,000. A small number of outliers with charges over $50,000 indicate significant variability, likely influenced by the predictor variables.

Since my label will be charges, I wanted to see how charges vary across different categorical variables like smoker, sex, region, and even children:

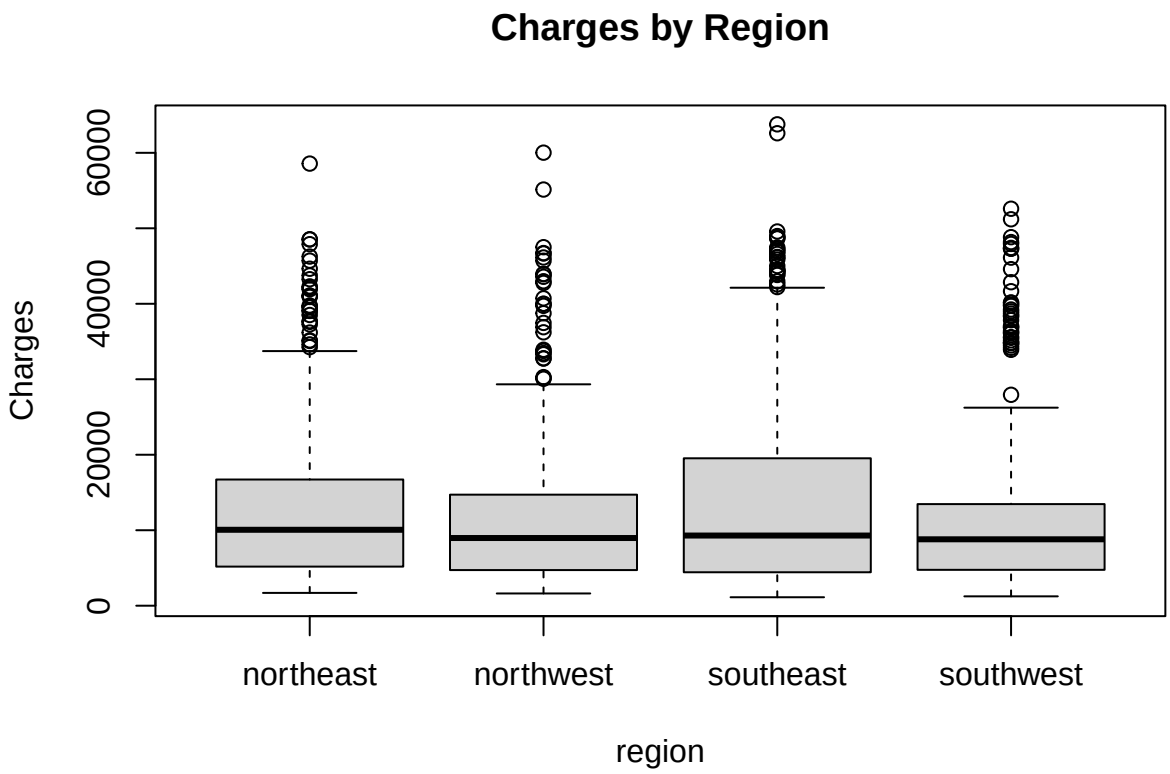
```
boxplot(charges ~ smoker, data = insurance_data,
        main = "Charges by Smoking Status", ylab = "Charges", col = c("lightblue", "salmon"))
```



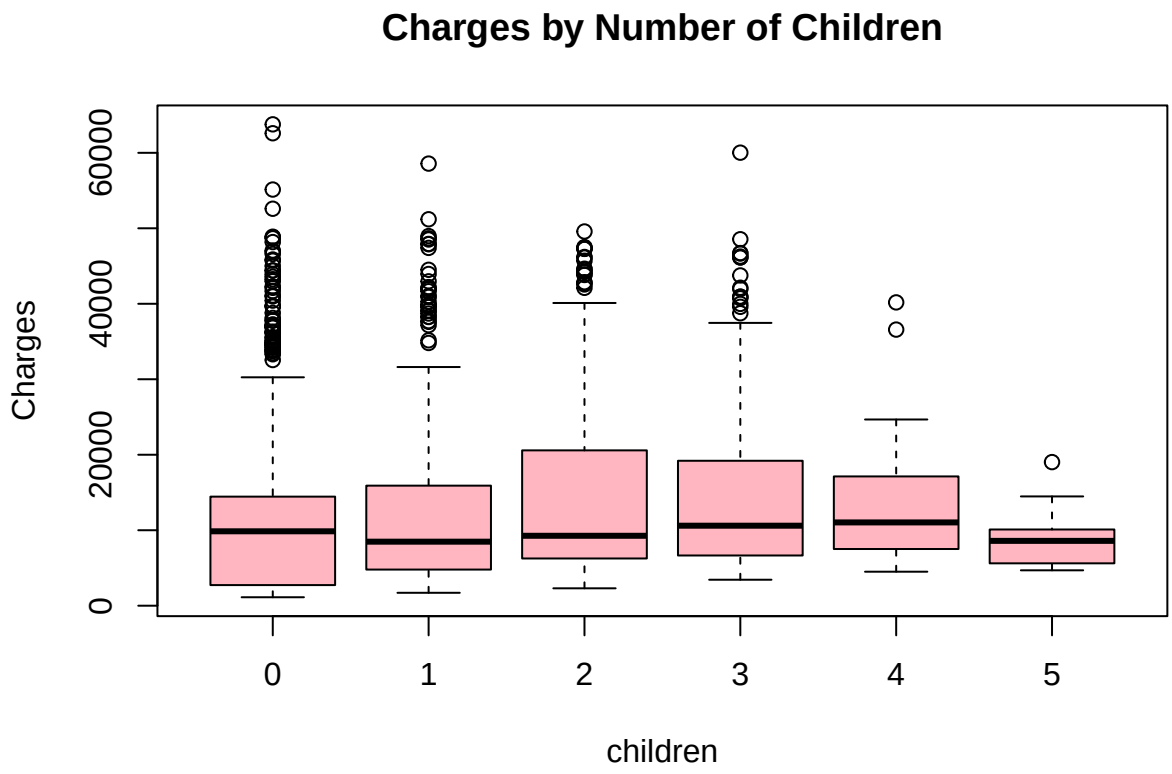

```
boxplot(charges ~ sex, data = insurance_data,
        main = "Charges by Sex", ylab = "Charges", col = c("lightgreen", "plum"))
```




```
boxplot(charges ~ region, data = insurance_data,
        main = "Charges by Region", ylab = "Charges", col = "lightgray")
```




```
boxplot(charges ~ children, data = insurance_data,
        main = "Charges by Number of Children", ylab = "Charges", col = "lightpink")
```



**Charges by Smoking Status:* *Smokers have significantly higher charges compared to non-smokers, with their median charges around $42,000. Non-smokers have a much lower median. The spread is also wider for smokers. This indicates more variability in their charges. This suggests smoking can be a strong predictor of insurance cost.

Charges by Sex: The median charges for males and females are relatively similar. Both distributions show a high number of outliers. There is no clear difference between sexes in terms of insurance charges. Therefore, I believe that sex might not be a strong factor on its own.

Charges by Region: Charges are fairly consistent across all four regions, with no major shifts in medians. However, all regions exhibit numerous outliers. This shows a common presence of high-cost individuals across geographic areas.

Charges by Children: The median charges do not vary dramatically with the number of children. However, there's a slight decrease for those with 4 or 5 children. Outliers are present across all groups. Variability is highest for those with 0–3 children since there are not many people that have 4-5 children. This suggests number of children also may have minimal influence on charges.

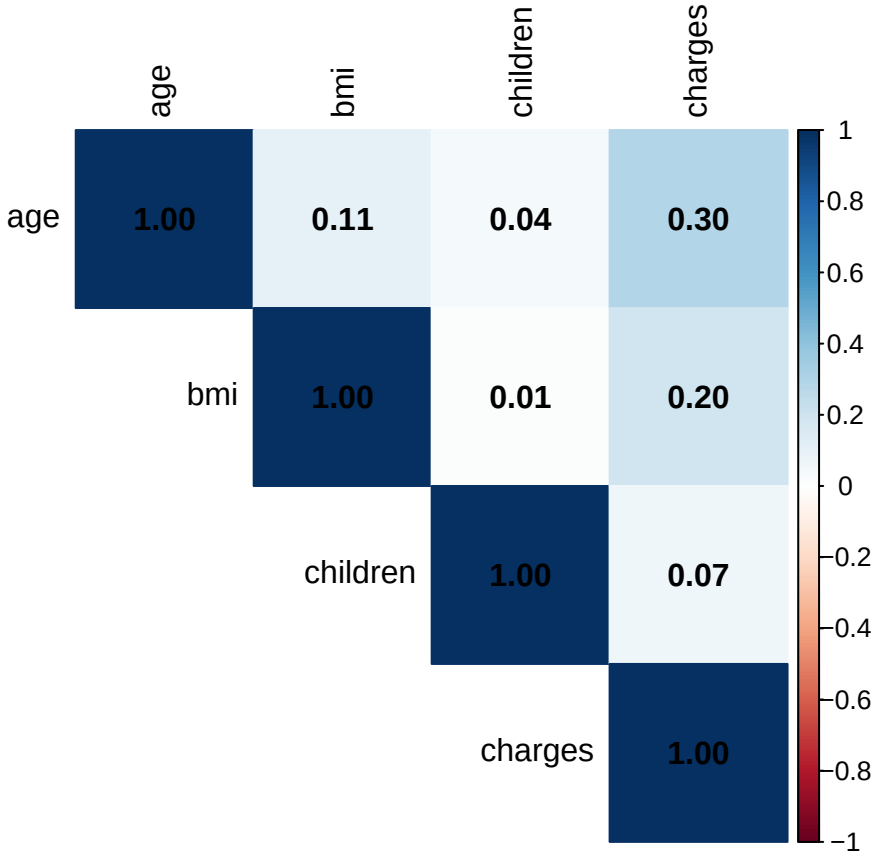
```
numeric_data <- insurance_data[, sapply(insurance_data, is.numeric)]
cor_matrix <- cor(numeric_data)
print(cor_matrix)
```

```
##                age       bmi   children    charges
## age      1.0000000 0.1092719 0.04246900 0.29900819
## bmi      0.1092719 1.0000000 0.01275890 0.19834097
## children 0.0424690 0.0127589 1.00000000 0.06799823
## charges  0.2990082 0.1983410 0.06799823 1.00000000
```

```r
library(corrplot)
```

```
## corrplot 0.95 loaded
```

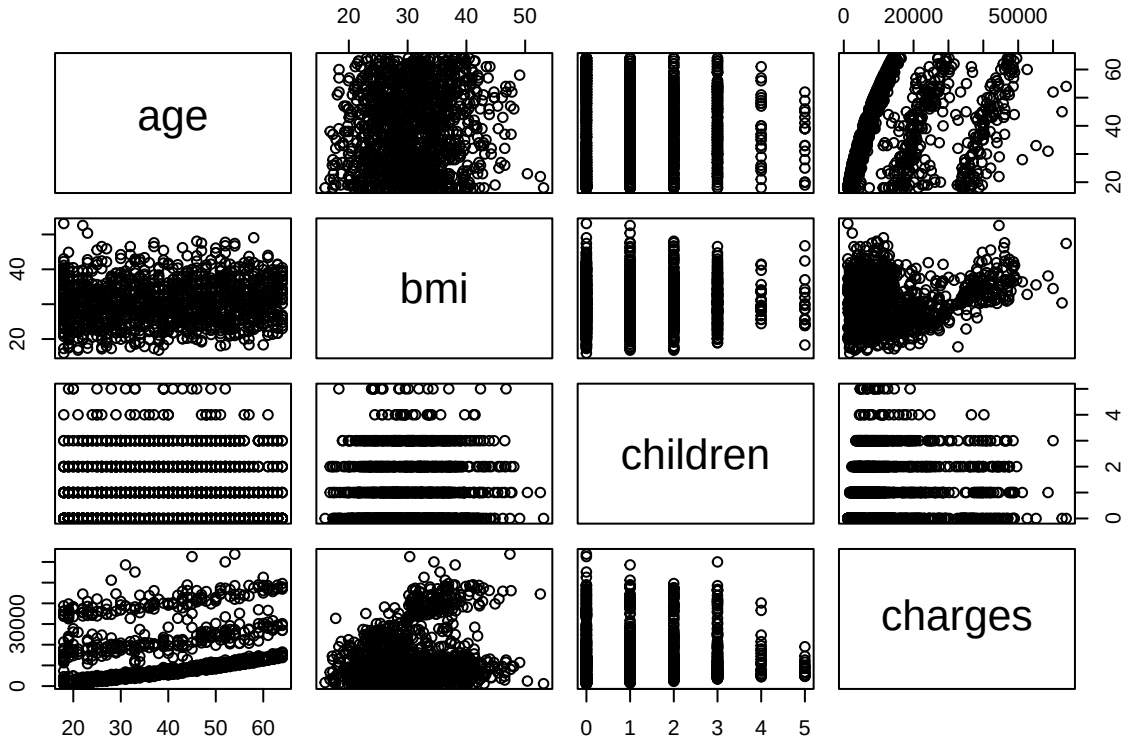
```r
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", addCoef.col = "black")
```



The correlation matrix shows how the numeric variables in the dataset relate to each other. Among them, age has the strongest positive correlation with charges (0.30). So, we can interpret this as older people tend to have higher insurance costs. BMI also has a weak positive correlation with charges (0.20), suggesting a slight tendency for people with higher BMI to pay more. The number of children has almost no correlation with charges (0.07). Therefore, I could say that it doesn't seem to affect insurance costs much. Overall, the correlations between the predictors themselves are low, so multicollinearity doesn't seem to be an issue.

However, we know that the correlation matrix only shows linear relationships. For example, age and charges have a moderate linear correlation, but for other variables like children or BMI, even if their correlation is low, it doesn't mean they have no impact — they might still affect charges in a nonlinear way or interact with other variables.

```r
pairs(insurance_data[, c("age", "bmi", "children", "charges")])
```



Based on the scatter plots, we can say that the relationship between age and charges appears to be positively curved, with older individuals tending to have higher charges, especially after a certain age. For BMI and charges, there isn't a strong trend overall, but there are some individuals with high BMI and very high charges, which could suggest interaction effects (like high BMI + smoker). The number of children mostly creates horizontal bands because it's a discrete variable, and it doesn't seem to influence charges much on its own.

```
library(dplyr)

insurance_data %>%
  group_by(sex, region,smoker) %>%
  summarize(mean_charges = mean(charges), .groups = "drop")%>%
  arrange(mean_charges)
```

```
## # A tibble: 16 x 4
##    sex    region    smoker mean_charges
##    <chr>  <chr>     <chr>         <dbl>
##  1 male   southeast no            7609.
##  2 male   southwest no            7779.
##  3 female southwest no            8234.
##  4 male   northwest no            8321.
##  5 female southeast no            8440.
##  6 male   northeast no            8664.
##  7 female northwest no            8787.
##  8 female northeast no            9640.
##  9 female northeast yes          28032.
## 10 female northwest yes          29671.
```

```
## 11 male   northwest yes          30713.
## 12 male   northeast yes          30926.
## 13 female southwest yes          31688.
## 14 male   southwest yes          32599.
## 15 female southeast yes          33035.
## 16 male   southeast yes          36030.
```

From the table, I can see that non-smokers have much lower average charges compared to smokers, no matter their sex or region. For example, non-smoking males in the southeast have the lowest average charges at around $7,600. Even among non-smokers, males usually pay a bit less than females, but the differences aren't that big.

Once I look at smokers, the average charges jump a lot. Female smokers in the southeast are paying over $33,000, and male smokers in the southeast have the highest at around $36,000. This shows that smoking is a huge factor in determining insurance charges — way more than sex or region.

I look at this to understand how charges change across different groups and see which combinations lead to higher or lower costs.

2. Fit an appropriate linear model with charges as response and the remaining variables as predictors.

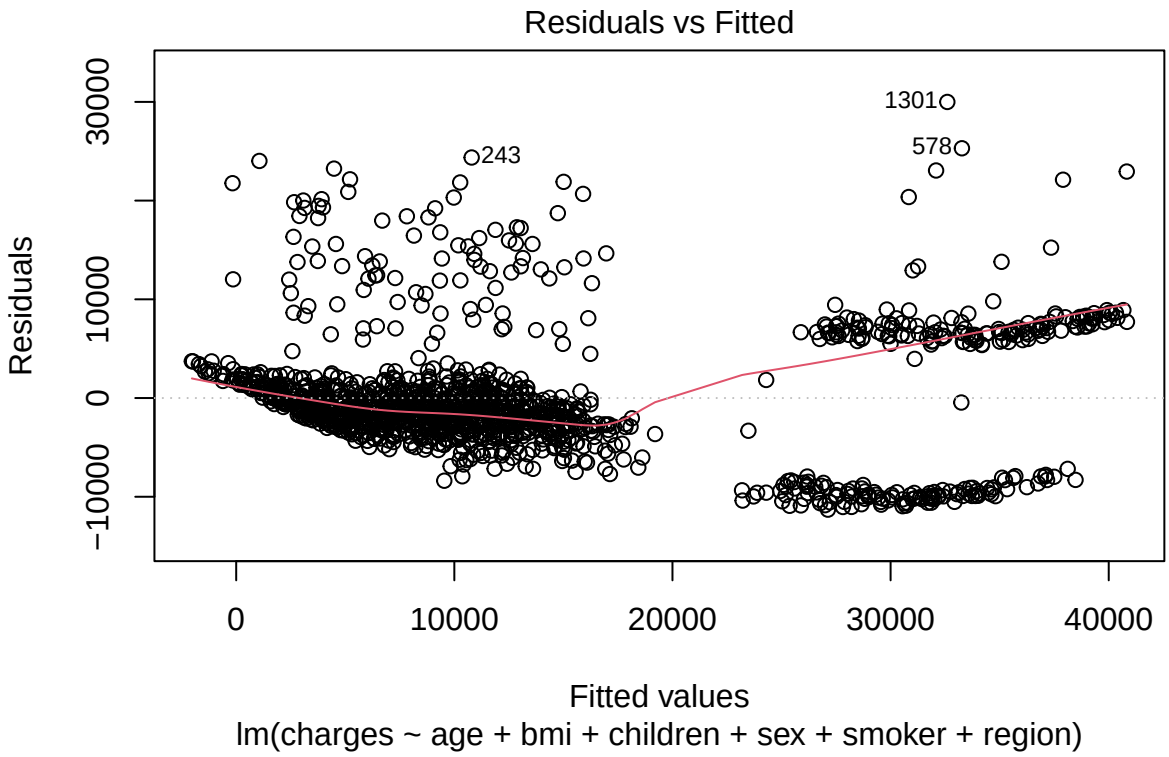
Solution:

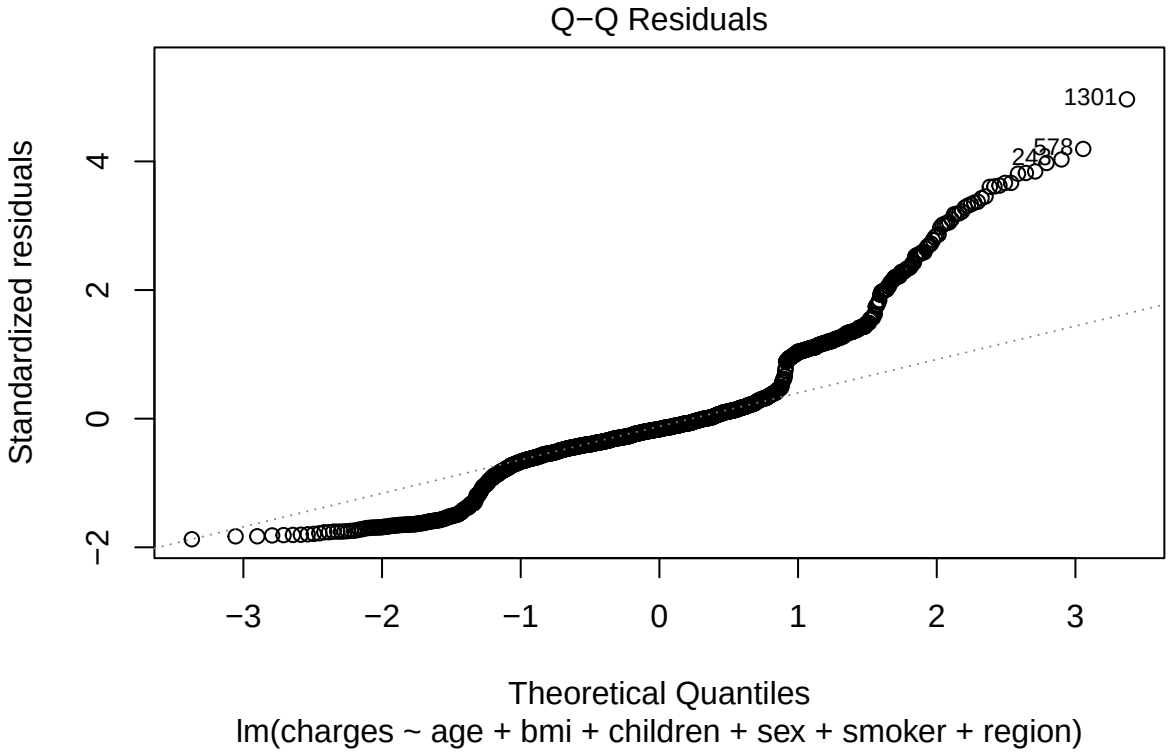
In this example, I did not continue with the best linear model. However, I tried different linear models:

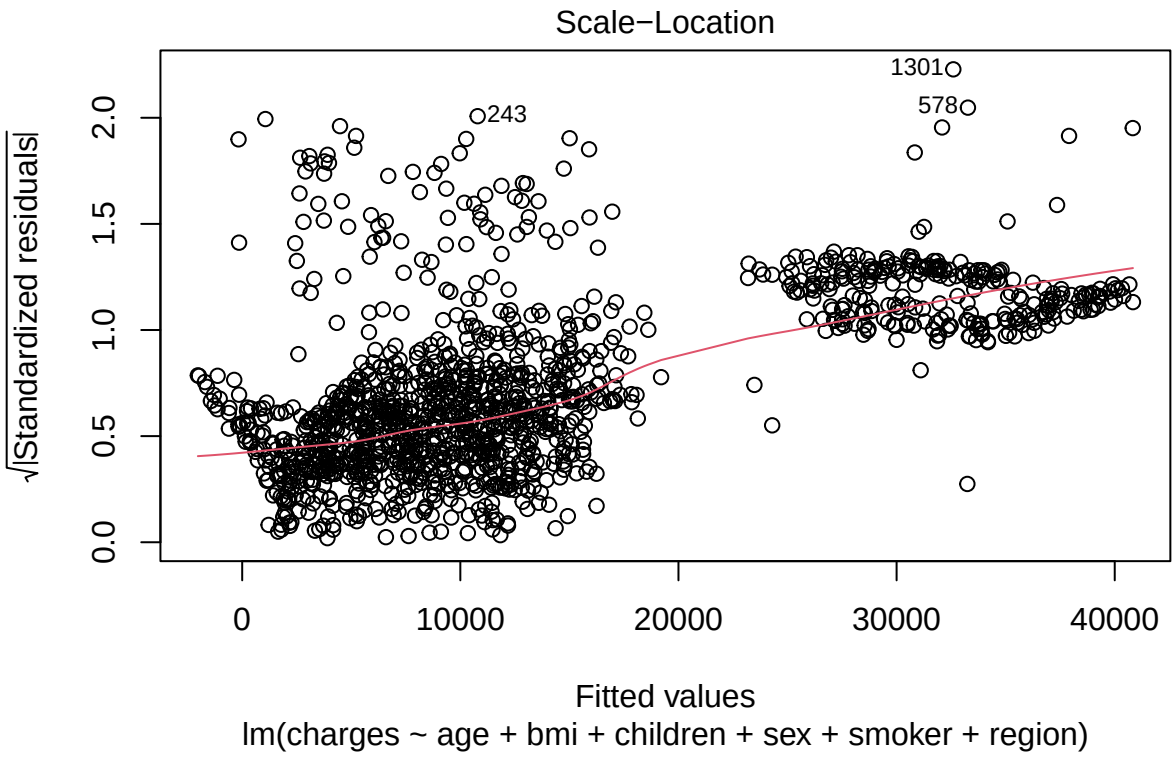
```
basic_lm_model <- lm(charges ~ age + bmi + children + sex + smoker + region, data = insurance_data)
summary(basic_lm_model)
```

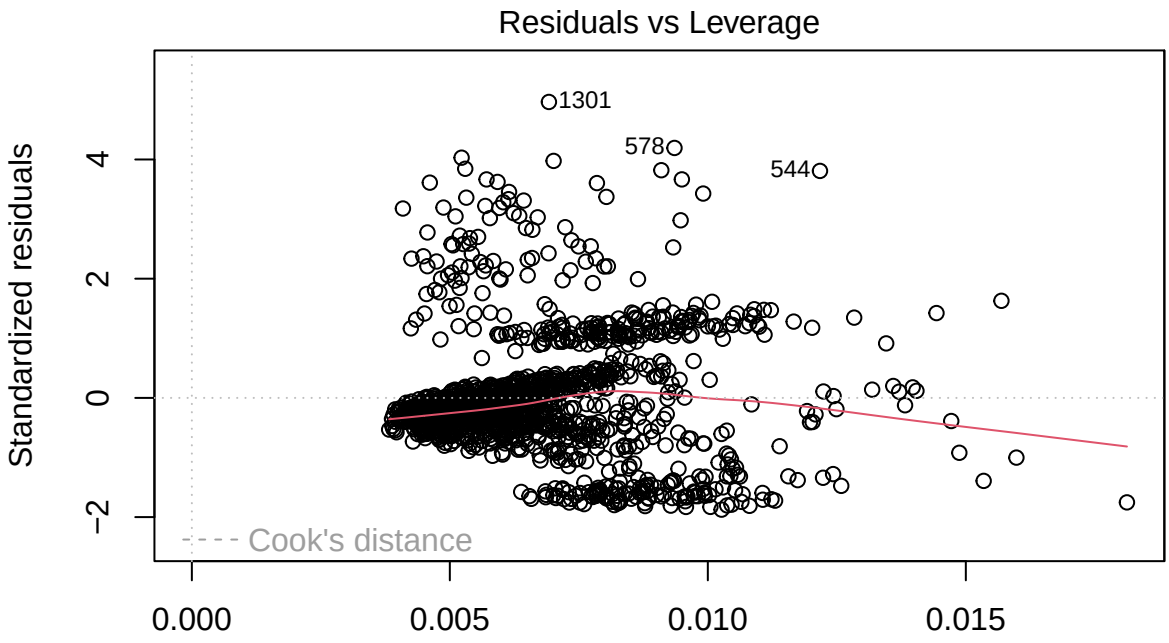
```
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## bmi                339.2       28.6  11.860  < 2e-16 ***
## children           475.5      137.8   3.451 0.000577 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```
plot(basic_lm_model)
```


Residuals vs Fitted
Residuals
30000
10000
0
−10000
1301
578
243
0
10000
20000
30000
40000
Fitted values
lm(charges ~ age + bmi + children + sex + smoker + region)


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(charges ~ age + bmi + children + sex + smoker + region)
1301
578
−2
0
2
4
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
10000
20000
30000
40000
1301
578
243
Fitted values
lm(charges ~ age + bmi + children + sex + smoker + region)

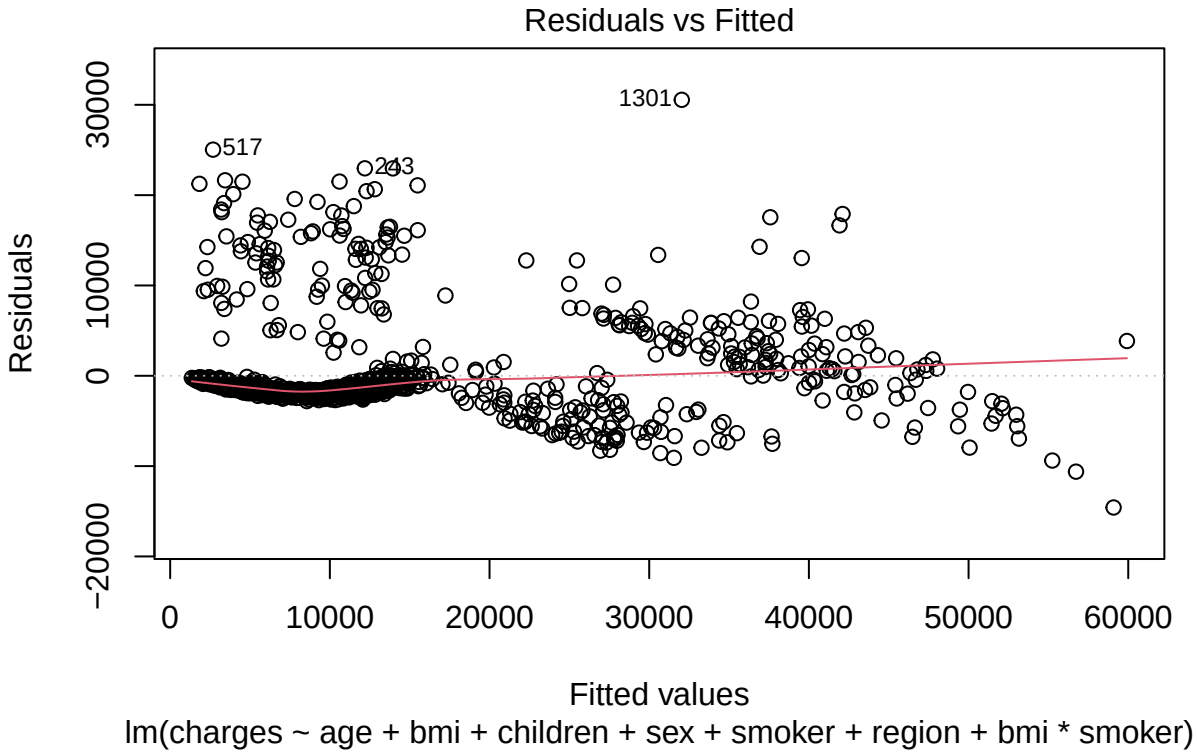


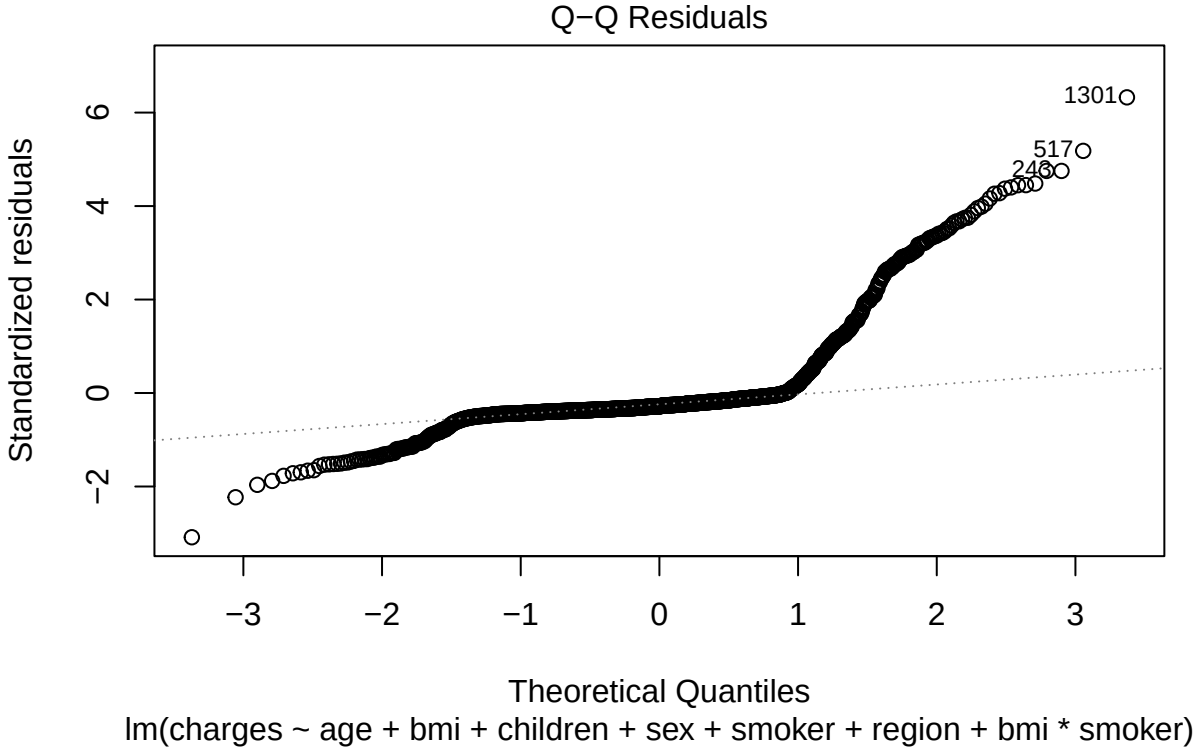

```
interaction_lm <- lm(charges ~ age + bmi + children + sex + smoker + region + bmi*smoker, data = insurar
summary(interaction_lm )
```

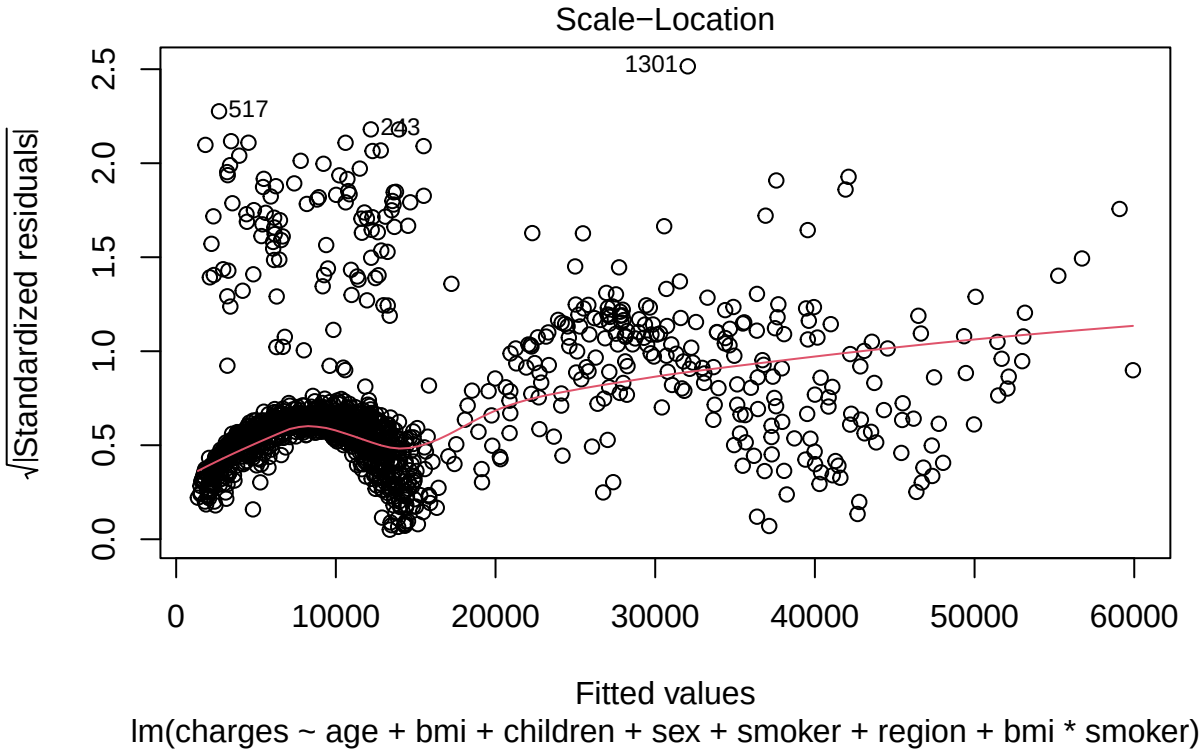
```
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region + bmi * smoker, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14580.7  -1857.2  -1360.8   -475.7  30552.4
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2223.454    865.611  -2.569  0.01032 *
## age                263.620      9.516  27.703  < 2e-16 ***
## bmi                 23.533     25.601   0.919  0.35814
## children           516.403    110.179   4.687 3.06e-06 ***
## sexmale           -500.146    266.518  -1.877  0.06079 .
## smokeryes       -20415.611   1648.277 -12.386  < 2e-16 ***
## regionnorthwest   -585.478    380.859  -1.537  0.12447
## regionsoutheast  -1210.131    382.750  -3.162  0.00160 **
## regionsouthwest  -1231.108    382.218  -3.221  0.00131 **
## bmi:smokeryes     1443.096     52.647  27.411  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

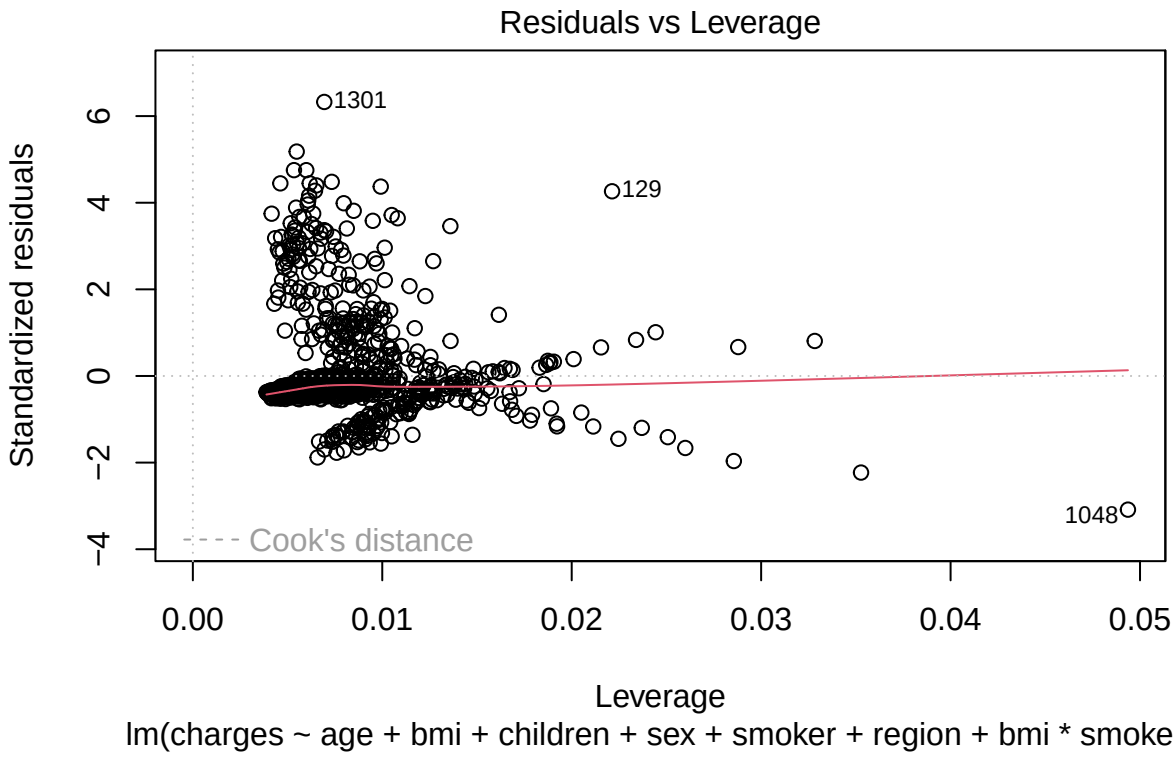
```
##
## Residual standard error: 4846 on 1328 degrees of freedom
## Multiple R-squared:  0.8409, Adjusted R-squared:  0.8398
## F-statistic:   780 on 9 and 1328 DF,  p-value: < 2.2e-16
```

```
plot(interaction_lm )
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(charges ~ age + bmi + children + sex + smoker + region + bmi * smoker)
1301
517
243
−2
0
2
4
6
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
1301
517
243
0
10000
20000
30000
40000
50000
60000
Fitted values
lm(charges ~ age + bmi + children + sex + smoker + region + bmi * smoker)

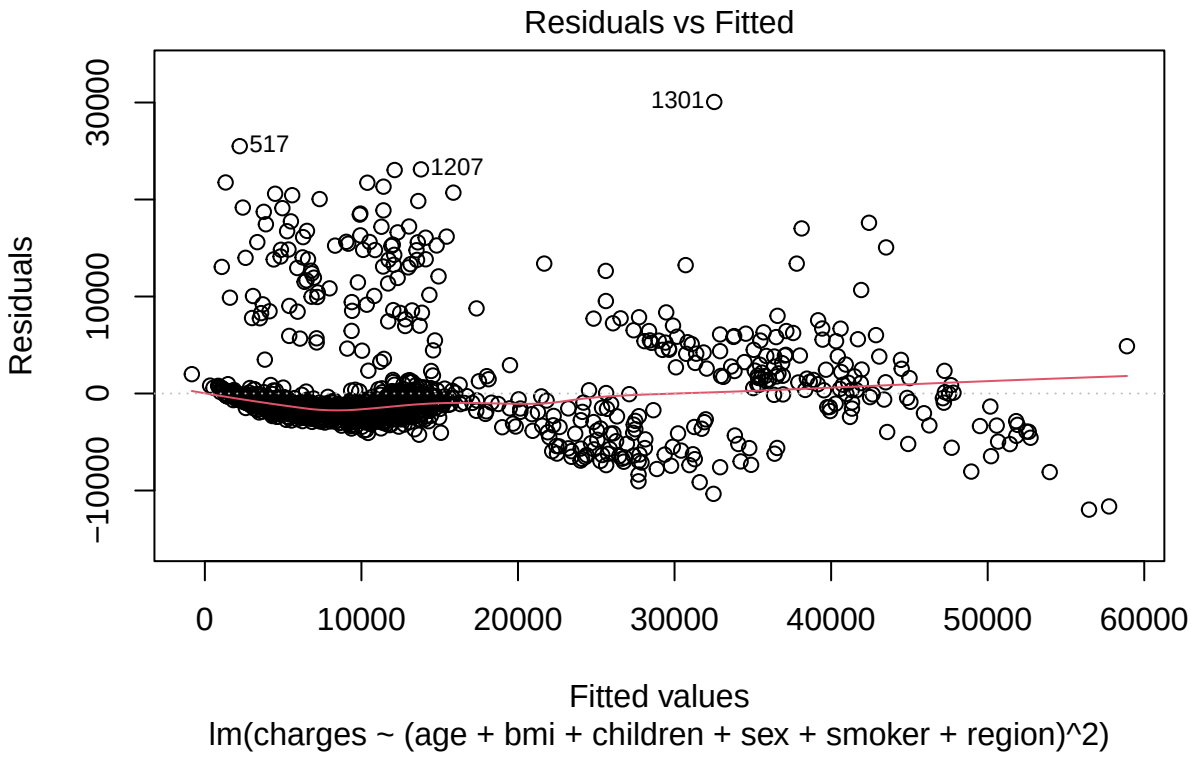


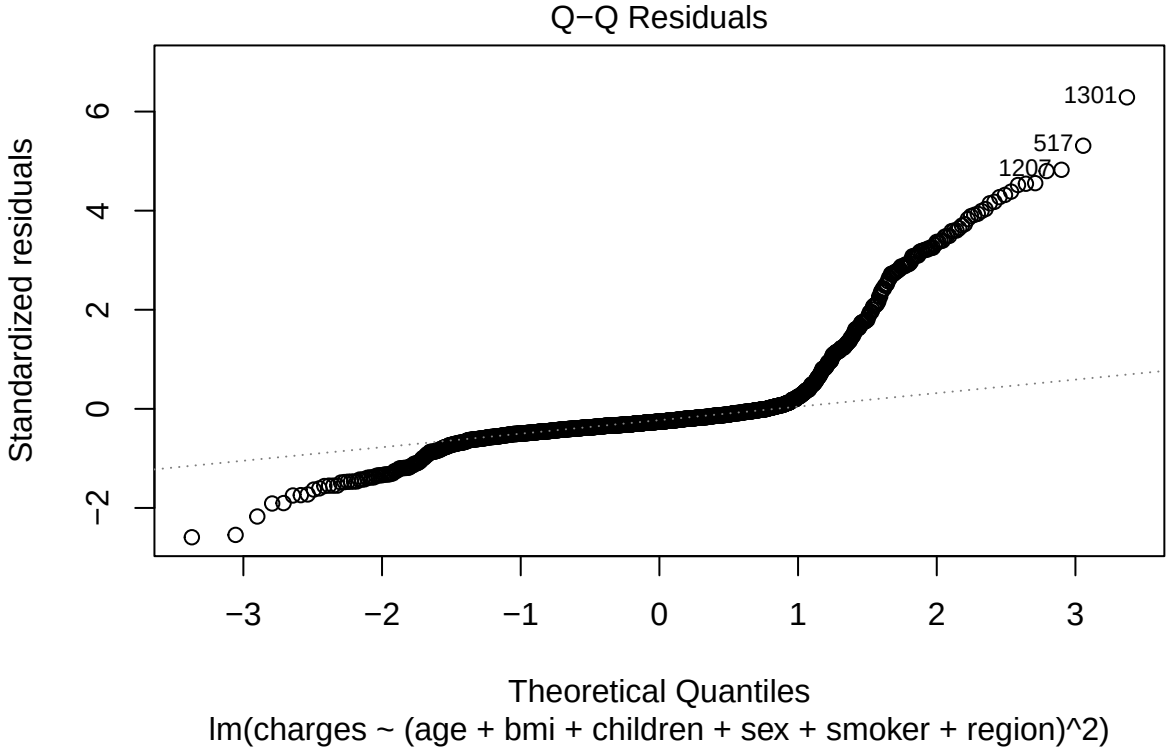

```
squared_lm <- lm(charges ~ (age + bmi + children + sex + smoker + region)^2, data = insurance_data)
summary(squared_lm)
```

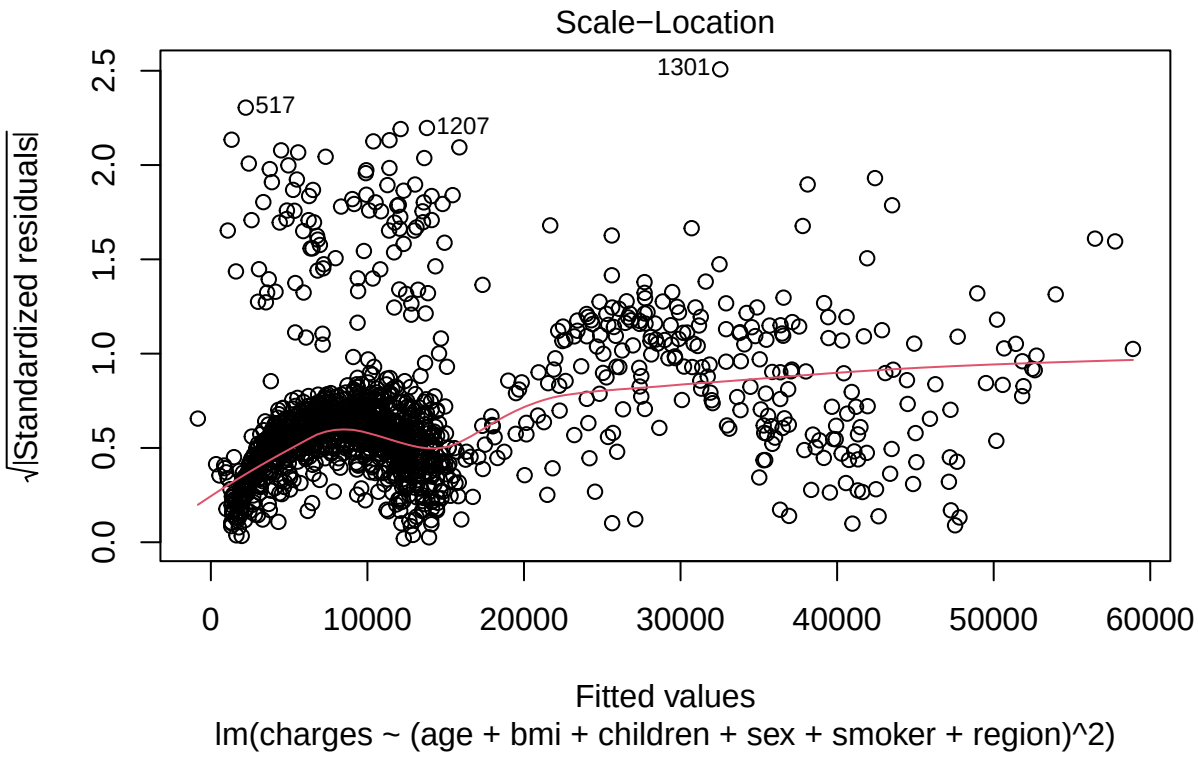
```
##
## Call:
## lm(formula = charges ~ (age + bmi + children + sex + smoker +
##     region)^2, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11971.4  -1979.7  -1233.8   -212.7  30056.8
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.682e+03  2.591e+03  -0.649 0.516186
## age                         1.919e+02  5.292e+01   3.626 0.000298 ***
## bmi                         4.958e+01  8.392e+01   0.591 0.554788
## children                    9.058e+02  6.729e+02   1.346 0.178485
## sexmale                    -1.260e+03  1.581e+03  -0.797 0.425510
## smokeryes                  -2.103e+04  1.937e+03 -10.856  < 2e-16 ***
## regionnorthwest            -1.611e+03  2.294e+03  -0.702 0.482761
## regionsoutheast             2.941e+03  2.193e+03   1.341 0.180147
## regionsouthwest            -4.435e+02  2.191e+03  -0.202 0.839615
## age:bmi                     1.168e+00  1.635e+00   0.714 0.475086
## age:children               -3.123e+00  8.607e+00  -0.363 0.716839
## age:sexmale                 1.577e+01  1.915e+01   0.823 0.410556
```

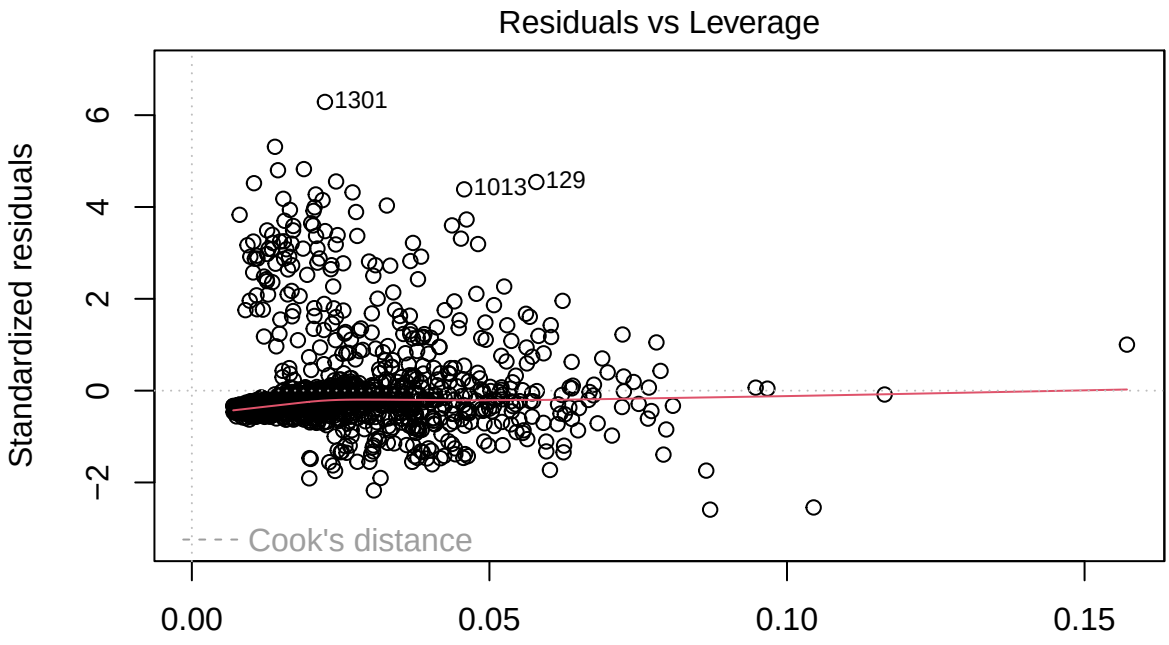
```
## age:smokeryes             -3.251e+00  2.400e+01  -0.135 0.892263
## age:regionnorthwest        1.975e+01  2.741e+01   0.721 0.471192
## age:regionsoutheast        5.074e+01  2.749e+01   1.846 0.065153 .
## age:regionsouthwest        4.764e+01  2.787e+01   1.709 0.087656 .
## bmi:children               4.843e-01  1.916e+01   0.025 0.979840
## bmi:sexmale                3.576e+00  4.636e+01   0.077 0.938528
## bmi:smokeryes              1.487e+03  5.649e+01  26.330  < 2e-16 ***
## bmi:regionnorthwest       -7.191e+00  7.019e+01  -0.102 0.918420
## bmi:regionsoutheast       -1.893e+02  6.113e+01  -3.097 0.001995 **
## bmi:regionsouthwest       -8.750e+01  6.727e+01  -1.301 0.193584
## children:sexmale          -2.456e+02  2.236e+02  -1.098 0.272196
## children:smokeryes        -3.848e+02  2.878e+02  -1.337 0.181483
## children:regionnorthwest   2.625e+02  3.256e+02   0.806 0.420147
## children:regionsoutheast  -2.073e+02  3.229e+02  -0.642 0.520963
## children:regionsouthwest  -3.803e+02  3.100e+02  -1.227 0.220035
## sexmale:smokeryes         -2.702e+01  6.792e+02  -0.040 0.968275
## sexmale:regionnorthwest    3.828e+02  7.660e+02   0.500 0.617308
## sexmale:regionsoutheast    6.368e+02  7.705e+02   0.826 0.408682
## sexmale:regionsouthwest    2.631e+02  7.702e+02   0.342 0.732700
## smokeryes:regionnorthwest -1.610e+02  9.740e+02  -0.165 0.868728
## smokeryes:regionsoutheast -1.145e+03  9.273e+02  -1.235 0.217237
## smokeryes:regionsouthwest  1.016e+03  9.872e+02   1.029 0.303475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4835 on 1304 degrees of freedom
## Multiple R-squared:  0.8445, Adjusted R-squared:  0.8406
## F-statistic: 214.7 on 33 and 1304 DF,  p-value: < 2.2e-16
```

```
plot(squared_lm)
```


Residuals vs Fitted
1301
517
1207
Residuals
30000
10000
0
−10000
0
10000
20000
30000
40000
50000
60000
Fitted values
lm(charges ~ (age + bmi + children + sex + smoker + region)^2)


Q−Q Residuals
Standardized residuals
−2
0
2
4
6
1301
517
1207
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(charges ~ (age + bmi + children + sex + smoker + region)^2)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
1301
517
1207
0
10000
20000
30000
40000
50000
60000
Fitted values
lm(charges ~ (age + bmi + children + sex + smoker + region)^2)

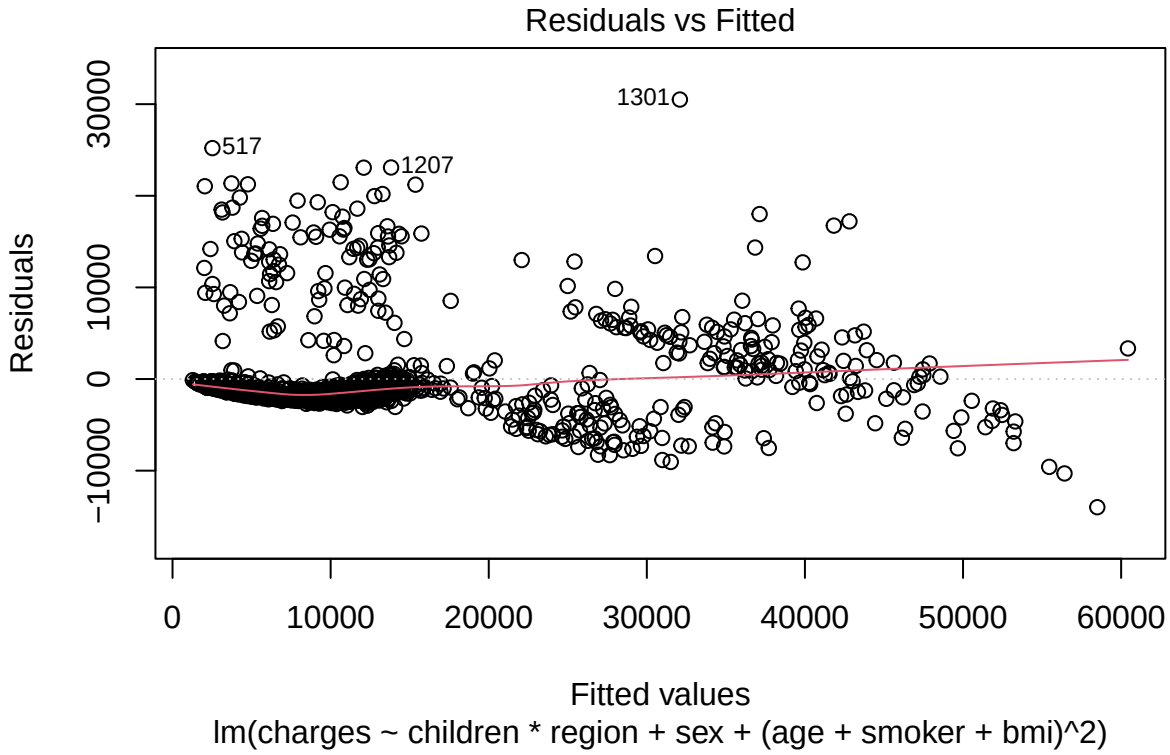


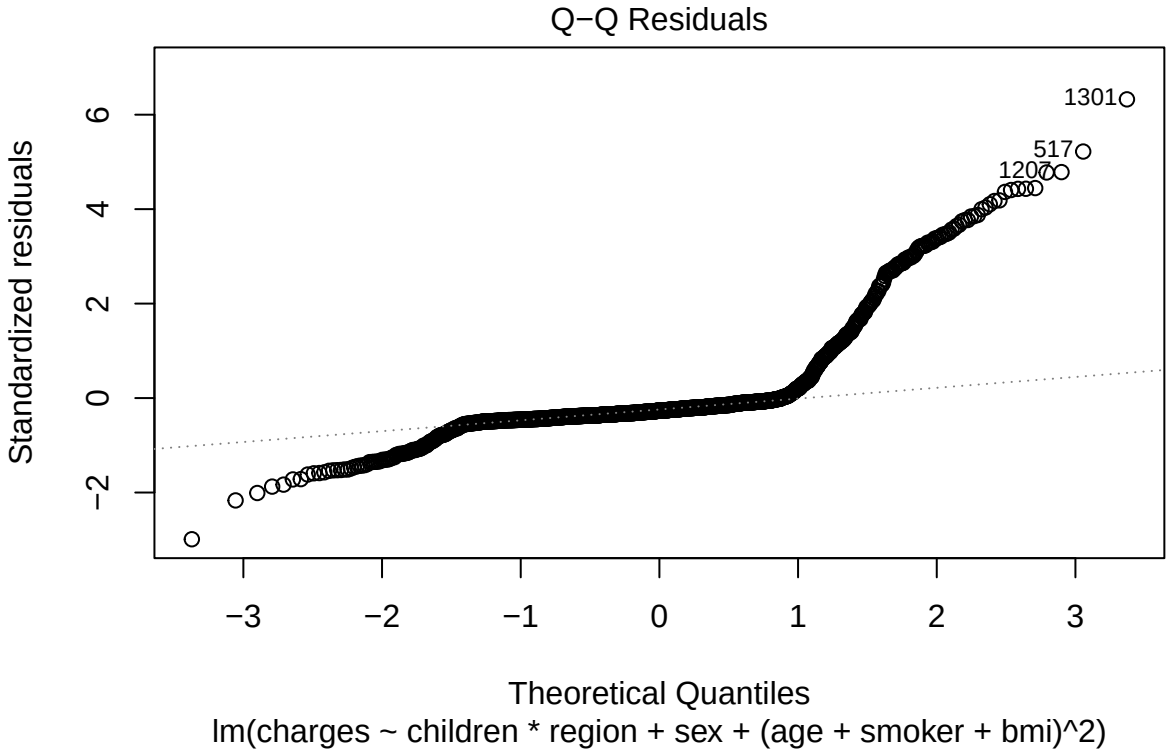

```
my_model <- lm(charges ~  children * region + sex + (age+smoker+bmi)^2 , data = insurance_data)
summary(my_model)
```

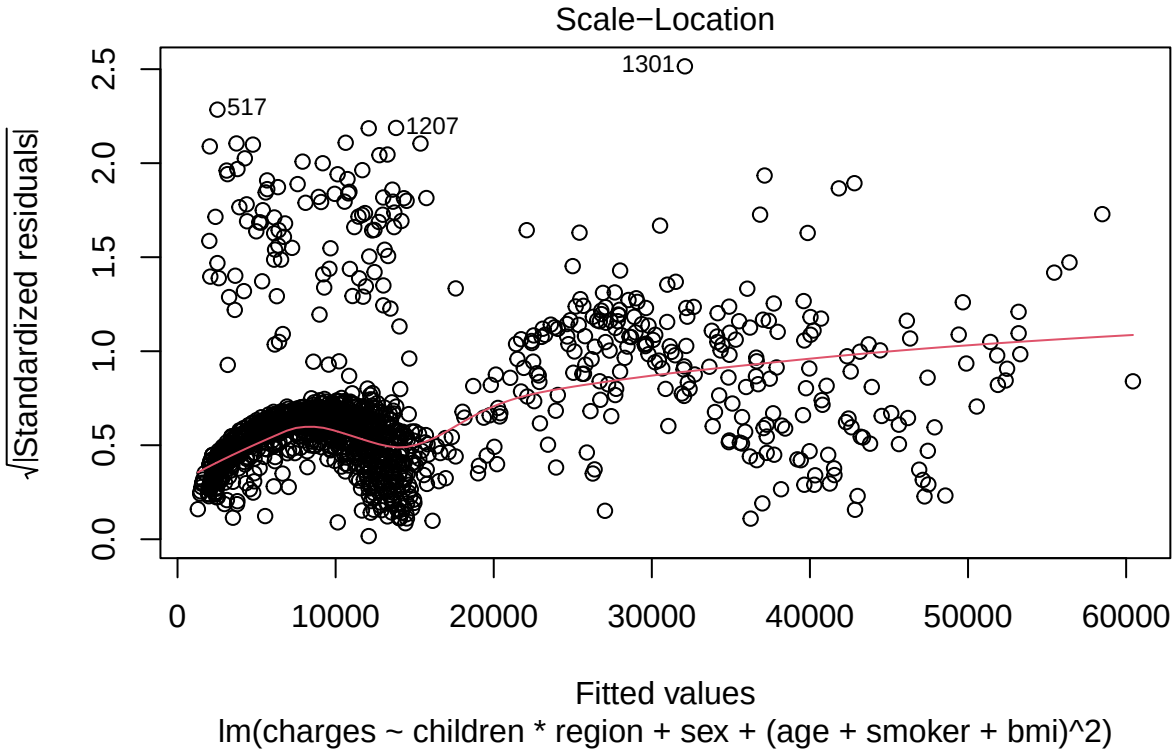
```
##
## Call:
## lm(formula = charges ~ children * region + sex + (age + smoker +
##     bmi)^2, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13990.2  -1914.0  -1289.1   -415.7  30501.7
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -145.918   2058.480  -0.071  0.94350
## children                     583.255    225.009   2.592  0.00964 **
## regionnorthwest             -921.915    520.078  -1.773  0.07652 .
## regionsoutheast            -1035.425    505.722  -2.047  0.04081 *
## regionsouthwest             -836.990    509.317  -1.643  0.10055
## sexmale                     -510.003    266.472  -1.914  0.05585 .
## age                          208.075     49.074   4.240 2.39e-05 ***
## smokeryes                 -20323.904   1831.499 -11.097  < 2e-16 ***
## bmi                           -47.089     65.779  -0.716  0.47420
## children:regionnorthwest      296.591    321.856   0.922  0.35696
## children:regionsoutheast     -147.522    312.686  -0.472  0.63715
## children:regionsouthwest     -358.930    308.717  -1.163  0.24518
```

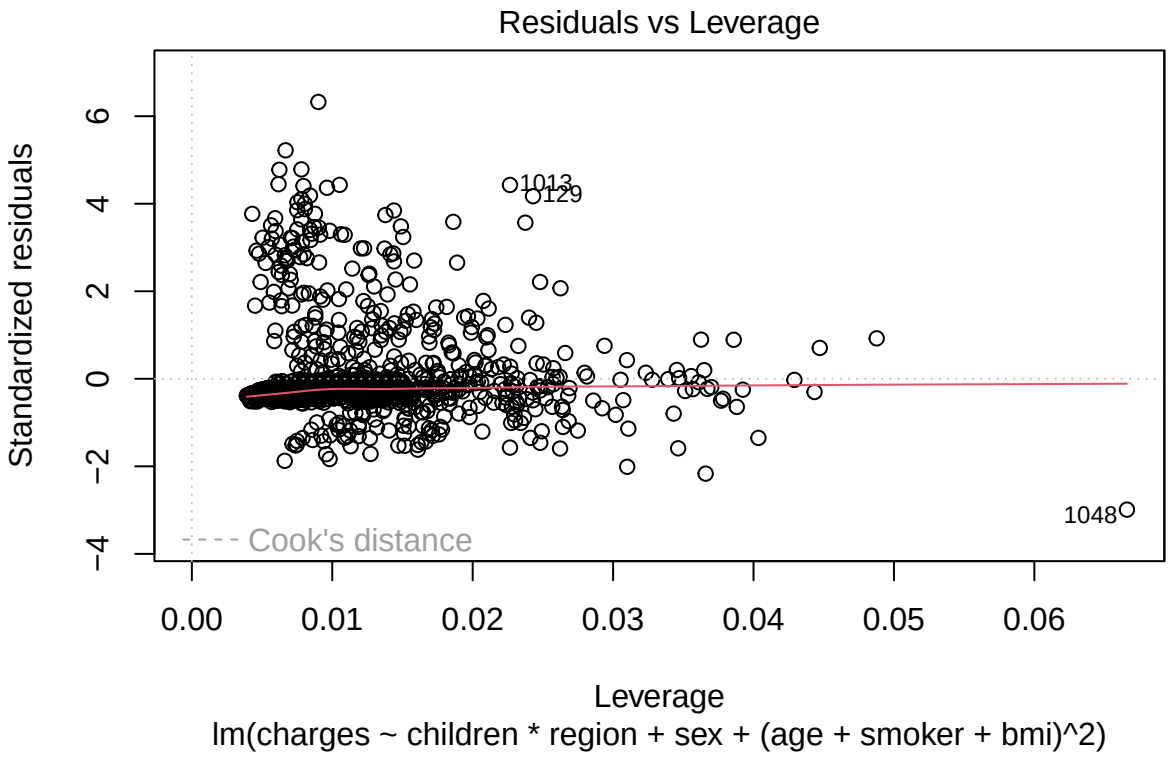
```
## age:smokeryes                -3.479     23.728  -0.147  0.88347
## age:bmi                       1.812      1.555   1.165  0.24434
## smokeryes:bmi              1444.472     52.863  27.325  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4844 on 1323 degrees of freedom
## Multiple R-squared:  0.8417, Adjusted R-squared:   0.84
## F-statistic: 502.3 on 14 and 1323 DF,  p-value: < 2.2e-16
```

```
plot(my_model)
```




Q−Q Residuals
Standardized residuals
−2
0
2
4
6
1301
517
1207
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(charges ~ children * region + sex + (age + smoker + bmi)^2)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
517
1207
1301
0
10000
20000
30000
40000
50000
60000
Fitted values
lm(charges ~ children * region + sex + (age + smoker + bmi)^2)



- Based on the p-value of the F-stat, the model is useful overall. That is, at least one predictor is relevant in predicting the charges value Based on the R2 -Based on the adjusted, about 84% of the total variations in charges can be explained by its linear relationship with the predictors in the model.
- It appears childen number and region are not significant in the model based on individual p-values. However, do not remove them from the model based on these marginal p-values.

```
AIC(basic_lm_model,interaction_lm,squared_lm,my_model)
```

```
##                df      AIC
## basic_lm_model 10 27115.51
## interaction_lm 11 26517.57
## squared_lm     35 26534.81
## my_model       16 26521.33
```

my_model is the best because it has the lowest AIC and the highest adjusted R^2, indicating strong model performance with good generalizability. Its residual plot shows the most random and balanced pattern, suggesting that it fits the data better than the other models. While there's still some variance at higher charge values, overall, it provides the best balance between accuracy and interpretability.

Based on the exploratory data analysis, I observed strong effects of variables like smoker, BMI, and age, as well as possible interactions. That's why in my_model, I included interaction terms like (age + smoker + bmi)^2 and children * region, which reflect patterns seen in the boxplots, scatterplots, and grouped summaries. This model aligns well with the data structure and captures both main effects and interactions that appear important in predicting charges.

However, to clearly observe the effects of the clustering and diagnostic steps in the later parts of this analysis, I used the simpler basic_lm_model instead of the more complex one for this analysis. So here are the few points for basic lm model:

```r
summary(basic_lm_model)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## bmi                339.2       28.6  11.860  < 2e-16 ***
## children           475.5      137.8   3.451 0.000577 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

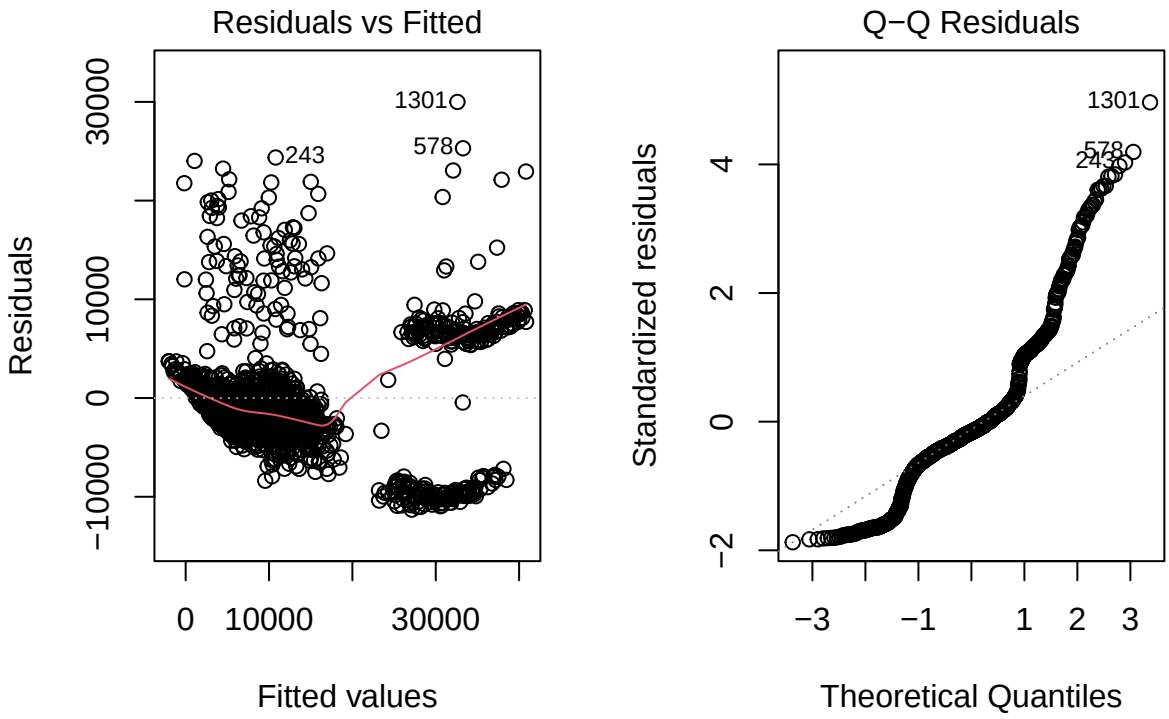
- Based on the p-value of the F-stat, the model is useful overall. That is, at least one predictor is relevant in predicting the charges value Based on the R2. -Based on the adjusted, about 75% of the total variations in charges can be explained by its linear relationship with the predictors in the model.
- It appears that sex and the region are not significant in the model based on individual p-values. However, do not remove them from the model based on these marginal p-values.
- Age, BMI, number of children, and smoker status are significant predictors of insurance charges.
- Smoking having the largest impact (increasing charges by ~$23,800).

3. Perform residual diagnostics for the model in (ii) and explain your findings.

Solution:

Residual QQ Plot:

```r
par(mfrow=c(1,2))
plot(basic_lm_model,1:2)
```



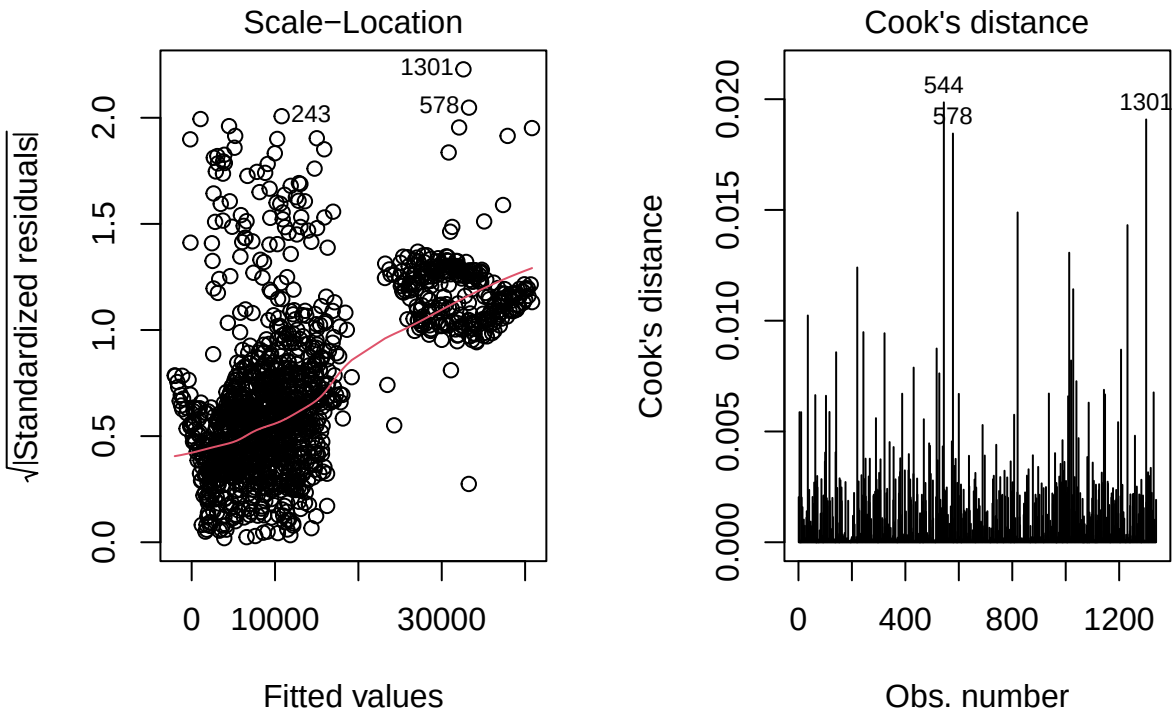
Based on the Normal Q-Q Plot:

- The residuals deviate from the diagonal line at both ends. This shows that the residuals are not normally distributed.
- It shows heavy right tail (positive skew).

Based on the Residuals vs. Fitted VAlues Graph:

- Residuals are not randomly scattered. I can see a curve pattern, especially in the middle and higher fitted values. This shows some nonlinearity or potential model misspecification.
- Points like 1301, 578, and 243 stand out as potential outliers.

```
par(mfrow=c(1,2))
plot(basic_lm_model,3:4)
```



Based on the scale-location plot: - The red line have trends upward. This means increasing variance in the residuals at higher fitted values. It is actually sign of heteroscedasticity. (Model is uneven spread for predicted values). So, error variance is not constant in opposite to the key assumption of linear regression.

- Ideally, points should be randomly scattered with a flat red line in the scale location plot.
- A few observations such as 1301, 243, and 578 show higher standardized residuals.Again, these are potential outliers.

Based on the Cook's distance:

- I searched and find that "Cook's distance measures how much each observation influences the overall regression model."

- In this model, most points have low influence, but a few like 544, 578, and 1301 stand out slightly (again the outliers).
- So, to sum up, The model is not very good because the residual plots show non-constant variance, non-normal errors, and a few influential outliers, which all violate key assumptions of linear regression. It works, but it could definitely be improved.

4. Extract the residuals ^ and the model matrix (X) without the intercept from the model in (ii)

Solution:

```
# extract the residuals
residuals <- resid(basic_lm_model)
head(residuals)
```

```
##          1          2          3          4          5          6
```

```
## -8408.7890 -1723.0505 -2257.5265 18229.6404 -1725.6382    36.7958
```

```r
# extract the model matrix X without the intercept of my_Model in q2
X <- model.matrix(basic_lm_model)[, -1]
head(X)
```

```
##   age    bmi children sexmale smokeryes regionnorthwest regionsoutheast
## 1  19 27.900        0       0         1               0               0
## 2  18 33.770        1       1         0               0               1
## 3  28 33.000        3       1         0               0               1
## 4  33 22.705        0       1         0               1               0
## 5  32 28.880        0       1         0               1               0
## 6  31 25.740        0       0         0               0               1
##   regionsouthwest
## 1               1
## 2               0
## 3               0
## 4               0
## 5               0
## 6               0
```

5. Fit a linear model with ^2 (squared residuals) as the response and X as the predictor

Solution:

```r
#  find squared residuals
squared_residuals <- residuals^2

# fit linear model  with ^2 (squared residuals) as the response and X as the predictor
squared_model <- lm(squared_residuals ~ X)
summary(squared_model)
```

```
##
## Call:
## lm(formula = squared_residuals ~ X)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -85348778 -24657420 -19379170  -9699370 819100619
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19083878   12271092   1.555    0.120
## Xage               -18306     147812  -0.124    0.901
## Xbmi               273306     355274   0.769    0.442
## Xchildren         1352802    1711859   0.790    0.430
## Xsexmale         -2747960    4135984  -0.664    0.507
## Xsmokeryes       57938894    5132359  11.289   <2e-16 ***
## Xregionnorthwest   254394    5916492   0.043    0.966
## Xregionsoutheast -1283334    5946510  -0.216    0.829
## Xregionsouthwest -7884726    5937079  -1.328    0.184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75310000 on 1329 degrees of freedom
## Multiple R-squared:  0.09099,    Adjusted R-squared:  0.08552
## F-statistic: 16.63 on 8 and 1329 DF,  p-value: < 2.2e-16
```

- Smoking is the only variable that significantly affects the squared residuals, meaning the model makes bigger errors when predicting charges for smokers. This shows that charges for smokers are harder to predict and more variable compared to non-smokers.
- All other predictors (age, BMI, children, sex, and region) are not statistically significant, meaning they do not explain the variation in residuals.
- The R^2 is pretty low (0.09), which means the predictors only explain about 9% of the differences in the residuals. So, most of the variation in the model's errors is still not explained.

6. What does the result in (v) suggest in terms of the assumption of independence between and X?

Solution:

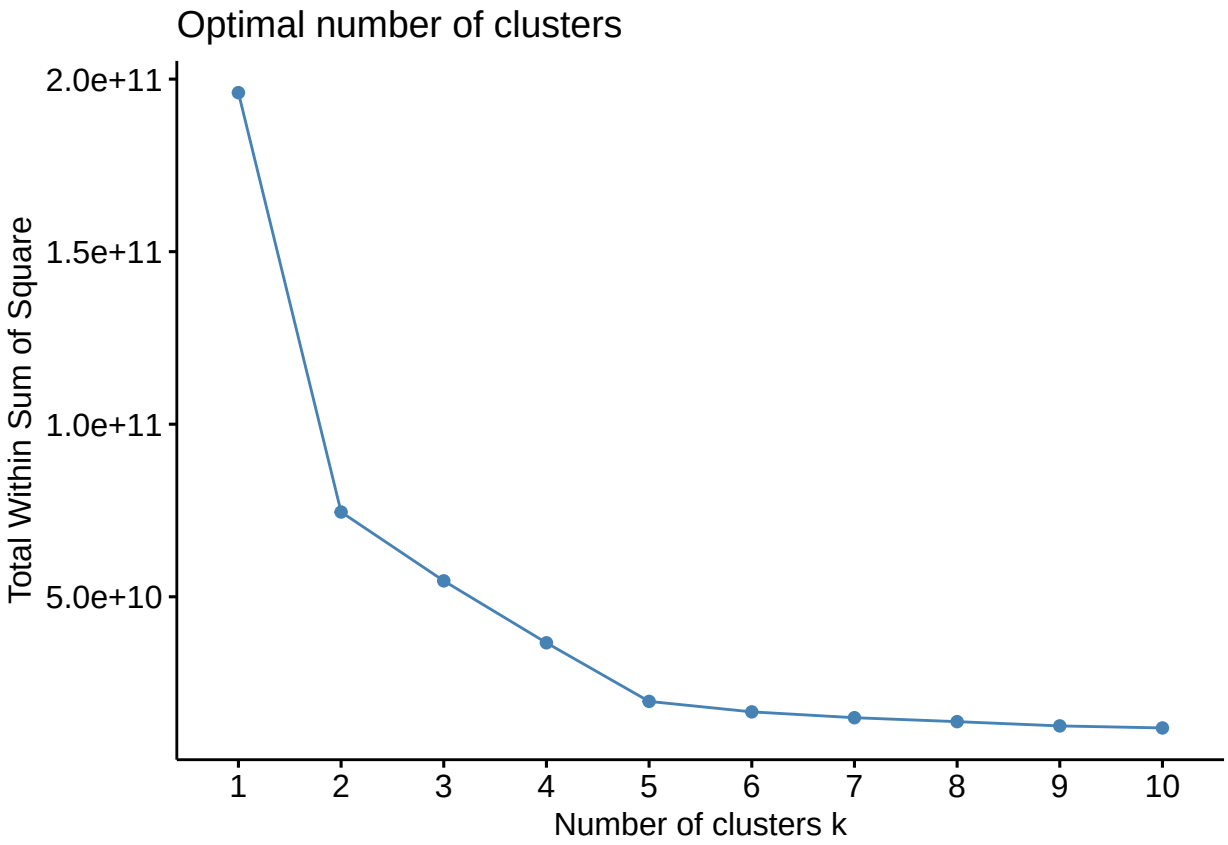
- We know that based on the key assumption of linear regression, the residuals ( ) are independent of the predictors (X). However, when we fit a model where the squared residuals were predicted using the original predictors in part (v), we see that smoker status was a significant predictor of the residual size. In other words, model tends to make larger errors for smokers. And this is statistically significant since the p-value is small. So, I can say that there is a statistically significant relationship between the residuals and the predictors (smoker variable).
- So, this result suggests a violation of the independence assumption, since at least one predictor (smoker) appears to influence the residuals. Even though, this does only explain the 9% of the differences in the residuals, it is still an important observation.

7. Perform cluster analysis on the matrix of residuals and fitted values, i.e., [ˆ, Xˆ] from model (ii) using an appropriate clustering method and report the cluster frequencies. Solution:

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# find the fitted values fitted values
fitted_vals <- fitted(basic_lm_model)

# recreate the matrix  [residuals, fitted values]
resid_fit_matrix <- cbind(residuals, fitted_vals)
fviz_nbclust(resid_fit_matrix, kmeans, method="wss")
```



- The elbow plot suggests that 3 clusters is the optimal choice, as adding more beyond that gives only minor improvement in model fit.

```
#  k-means clustering
set.seed(42)
kmeans_result <- kmeans(resid_fit_matrix, centers = 3)

# cluster frequencies
table(kmeans_result$cluster)
```

```
##
##   1   2   3
## 628 436 274
```

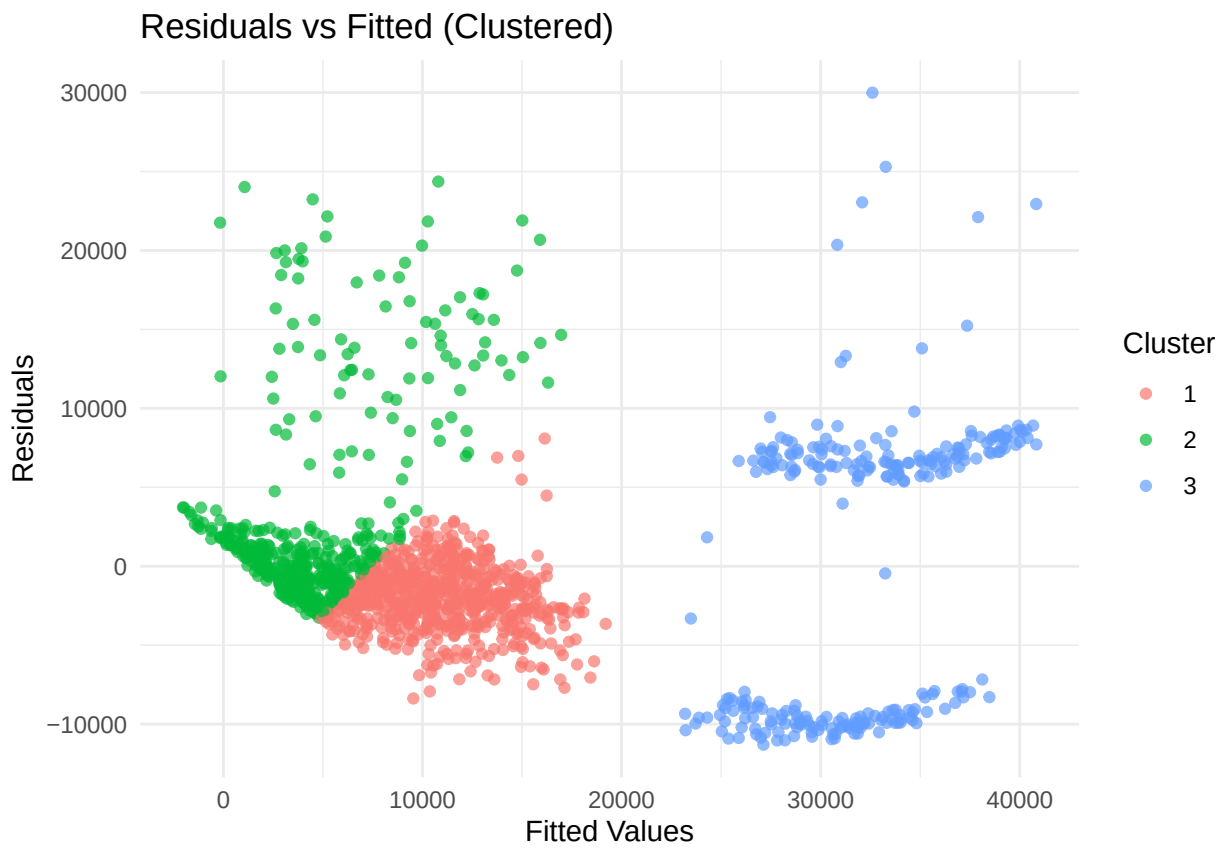
Based on the result of the k-means, data is cluster in 3 groups:

- First cluster has 628 observations
- Second cluster have 436 observations and
- Third cluster have 274 observations
- By looking into this distribution, one can easily say that majority of the observations are in cluster 1.
- Meanwhile, second and third clusters capture smaller subgroups with different residual–fitted value patterns.
- The clustering on residuals and fitted values shows that there are different groups in the data where the model performs differently. This means the model might be missing some patterns, and there could be subgroups that aren't fully captured by the current predictors.

8. Repeat the residual plots in (iii) and label the points based on the cluster assignments in (vii).

Solution:

```
library(ggplot2)

# data frame with fitted values, residuals, and cluster assignments
resid_df <- data.frame(
  Fitted = fitted(basic_lm_model),
  Residuals = resid(basic_lm_model),
  Cluster = as.factor(kmeans_result$cluster)
)
ggplot(resid_df, aes(x = Fitted, y = Residuals, color = Cluster)) +
  geom_point(alpha = 0.7) +
  labs(title = "Residuals vs Fitted (Clustered)", x = "Fitted Values", y = "Residuals") +
  theme_minimal()
```

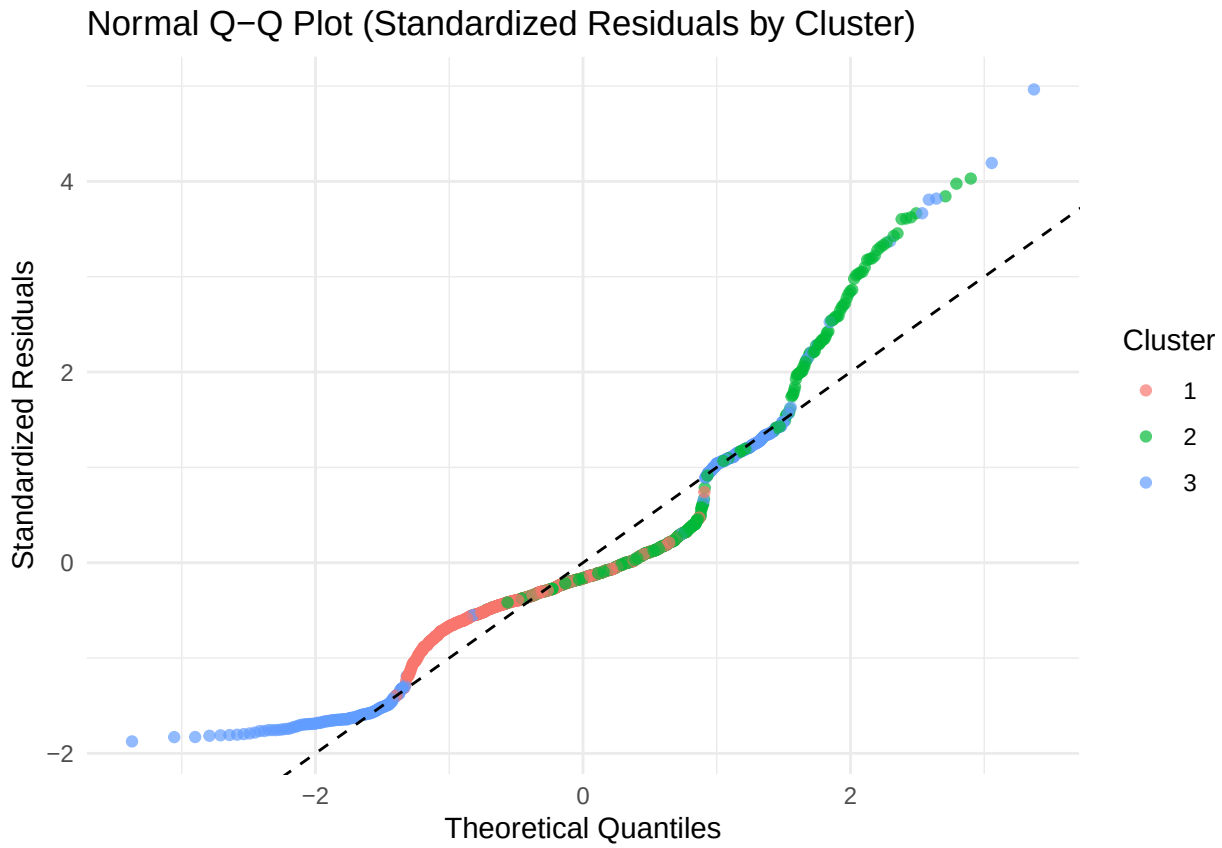


- Based on the clustered plot of residual vs. fitted value, red and green clusters have lower fitted values. For the green cluster (cluster 2), they have much larger positive residuals. THis may mean that my model is underpredicting charges for these cases.

- On the other hand, Cluster 3 (blue) is concentrated at higher fitted values. I interpreted this as the model's predictions are less accurate and more variable for those high-charge individuals.

```
# need to get standardized residuals
standardized_resid <- rstandard(basic_lm_model)

# theoretical quantiles from qqnorm()
qq_data <- qqnorm(standardized_resid, plot.it = FALSE)
```

```r
# into a data frame
resid_df <- data.frame(
  Theoretical = qq_data$x,
  Sample = qq_data$y,
  Cluster = as.factor(kmeans_result$cluster)
)

# with labels and line
library(ggplot2)
ggplot(resid_df, aes(x = Theoretical, y = Sample, color = Cluster)) +
  geom_point(alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  labs(title = "Normal Q-Q Plot (Standardized Residuals by Cluster)",
       x = "Theoretical Quantiles", y = "Standardized Residuals") +
  theme_minimal()
```

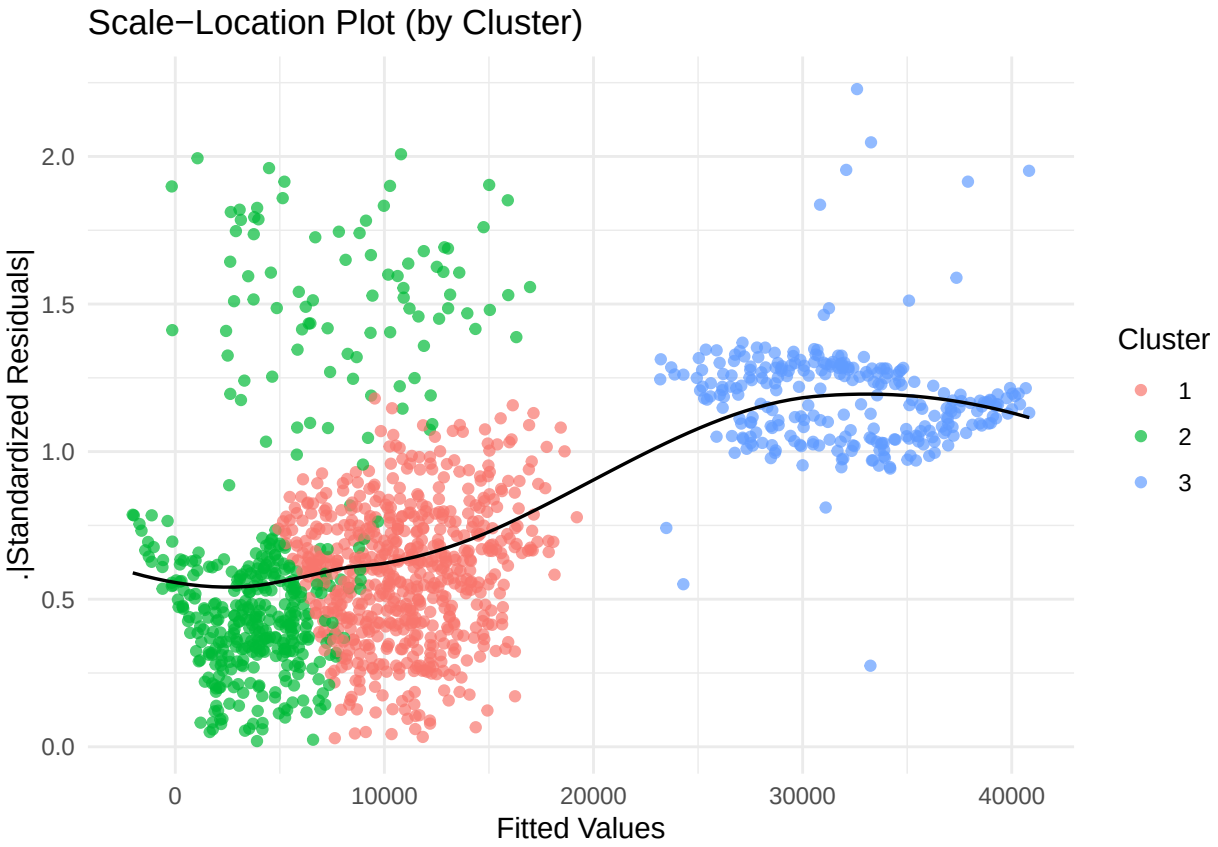


Based on the Normal Q-Q plot: - Cluster 1 has residuals close to normal, while Clusters 2 and 3 deviate from normality with large positive residuals. Again this confirms that the model underestimates charges for those groups.

```r
resid_df <- data.frame(
  Fitted = fitted(basic_lm_model),
  Std_Resid = rstandard(basic_lm_model),
  Cluster = as.factor(kmeans_result$cluster)
)

#square root of absolute standardized residuals
```

```r
resid_df$Sqrt_Abs_Std_Resid <- sqrt(abs(resid_df$Std_Resid))

ggplot(resid_df, aes(x = Fitted, y = Sqrt_Abs_Std_Resid, color = Cluster)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "black", linewidth = 0.6) +
  labs(title = "Scale-Location Plot (by Cluster)",
       x = "Fitted Values",
       y = "√|Standardized Residuals|") +
  theme_minimal()
```

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

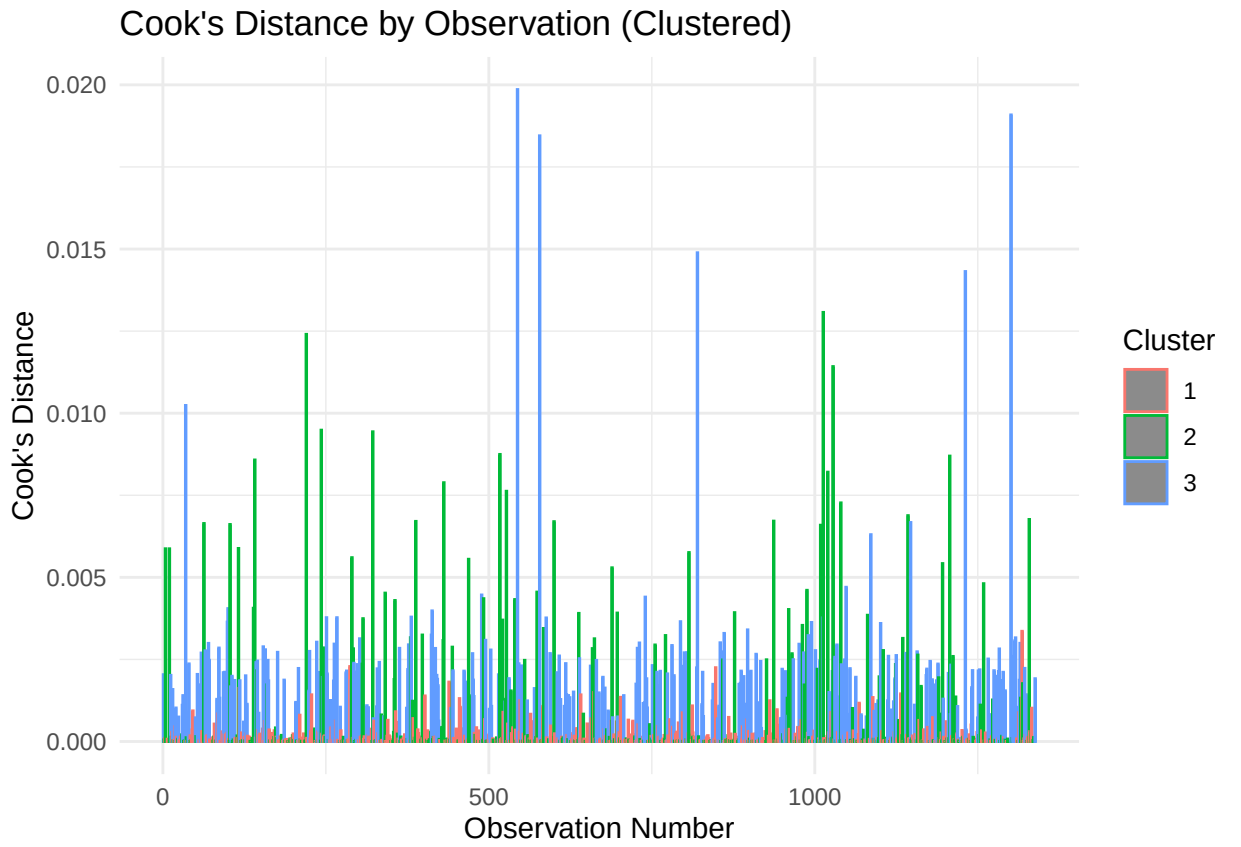


Based on the scale-location plot:

- The model makes more scattered and less stable predictions for people with higher predicted charges, especially in Cluster . This shows the model's error grows as charges increase.

```r
# Cook's distances
cooks <- cooks.distance(basic_lm_model)

# Create a data frame with observation index
cooks_df <- data.frame(
  Obs = 1:length(cooks),
  Cook = cooks,
  Cluster = as.factor(kmeans_result$cluster)
)

# Plot
```

```r
ggplot(cooks_df, aes(x = Obs, y = Cook, color = Cluster)) +
  geom_bar(stat = "identity", alpha = 0.7) +
  labs(title = "Cook's Distance by Observation (Clustered)",
       x = "Observation Number",
       y = "Cook's Distance") +
  theme_minimal()
```



- This plot shows that most observations have low Cook's distance, meaning they don't heavily influence the model. However, a few (mostly from Cluster 3) stand out slightly. This shows that they may have more impact on the model's estimates.

9. Re-fit your linear model in (ii) and include the cluster labels as a categorical predictor. Perform residual diagnostics and report your findings.

Solution:

```r
# add cluster to dataset
insurance_data$cluster <- as.factor(kmeans_result$cluster)

# refit the model from (ii)
model_with_cluster <- lm(charges ~ age + bmi + children + sex + smoker + region + cluster, data = insura

summary(model_with_cluster)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region + cluster, data = insurance_data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11225  -3475   -312   2908  29944
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -27494.10    1180.09 -23.298  < 2e-16 ***
## age                383.60      12.34  31.098  < 2e-16 ***
## bmi                566.53      27.77  20.400  < 2e-16 ***
## children           726.24     122.18   5.944 3.54e-09 ***
## sexmale           -276.25     293.65  -0.941 0.347007
## smokeryes        27469.53     408.72  67.209  < 2e-16 ***
## regionnorthwest   -238.49     419.96  -0.568 0.570206
## regionsoutheast  -1512.39     422.76  -3.577 0.000359 ***
## regionsouthwest   -906.60     421.39  -2.151 0.031622 *
## cluster2          8475.24     433.84  19.535  < 2e-16 ***
## cluster3               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5345 on 1328 degrees of freedom
## Multiple R-squared:  0.8065, Adjusted R-squared:  0.8052
## F-statistic: 615.1 on 9 and 1328 DF,  p-value: < 2.2e-16
```

- We can easily see from the summary that the model is improved. Adjusted R^2 improved from 0.749 to 0.805, meaning the new model explains over 80% of the variation in charges — a significant improvement. However, it has still lower adjuster R^2 compared to my_model i found in (ii).
- The residual standard error decreased from around 6062 to 5345, indicating smaller prediction errors on average.
- The new variable cluster2 is highly significant. THis shows that it captures meaningful differences in error patterns across groups that the original predictors missed. It is statistically significant.
- One can see that the cluster 3 is dropped since it did not provide new information which was already explained by the other predictors.

```
# see why cluster 3 dropped
model.matrix(~ cluster, data = insurance_data)[1:5, ]
```

```
##   (Intercept) cluster2 cluster3
## 1           1        0        1
## 2           1        1        0
## 3           1        0        0
## 4           1        1        0
## 5           1        1        0
```

```
insurance_data %>%
  group_by(cluster) %>%
  summarise(
    avg_age = mean(age),
    avg_bmi = mean(bmi),
    avg_charges = mean(charges),
    smoker_rate = mean(smoker == "yes"),
    male_rate = mean(sex == "male"),
    .groups = "drop"
```

```
  )
```

```
## # A tibble: 3 x 6
##   cluster avg_age avg_bmi avg_charges smoker_rate male_rate
##   <fct>     <dbl>   <dbl>       <dbl>       <dbl>     <dbl>
## 1 1          46.3    33.0       8978.           0     0.481
## 2 2          29.5    27.3       7651.           0     0.493
## 3 3          38.5    30.7      32050.           1     0.580
```

```
X <- model.matrix(charges ~ age + bmi + children + sex + smoker + region + cluster, data = insurance_da

qr(X)$rank # actual rank
```

```
## [1] 10
```

```
ncol(X)          # total number of columns in the design matrix
```

```
## [1] 11
```

Cluster 3 was dropped because it only contains smokers, and their characteristics — like higher charges, moderate age, and high BMI — are already explained by other predictors in the model. So, R found that adding cluster3 didn't add any new information, making it redundant.

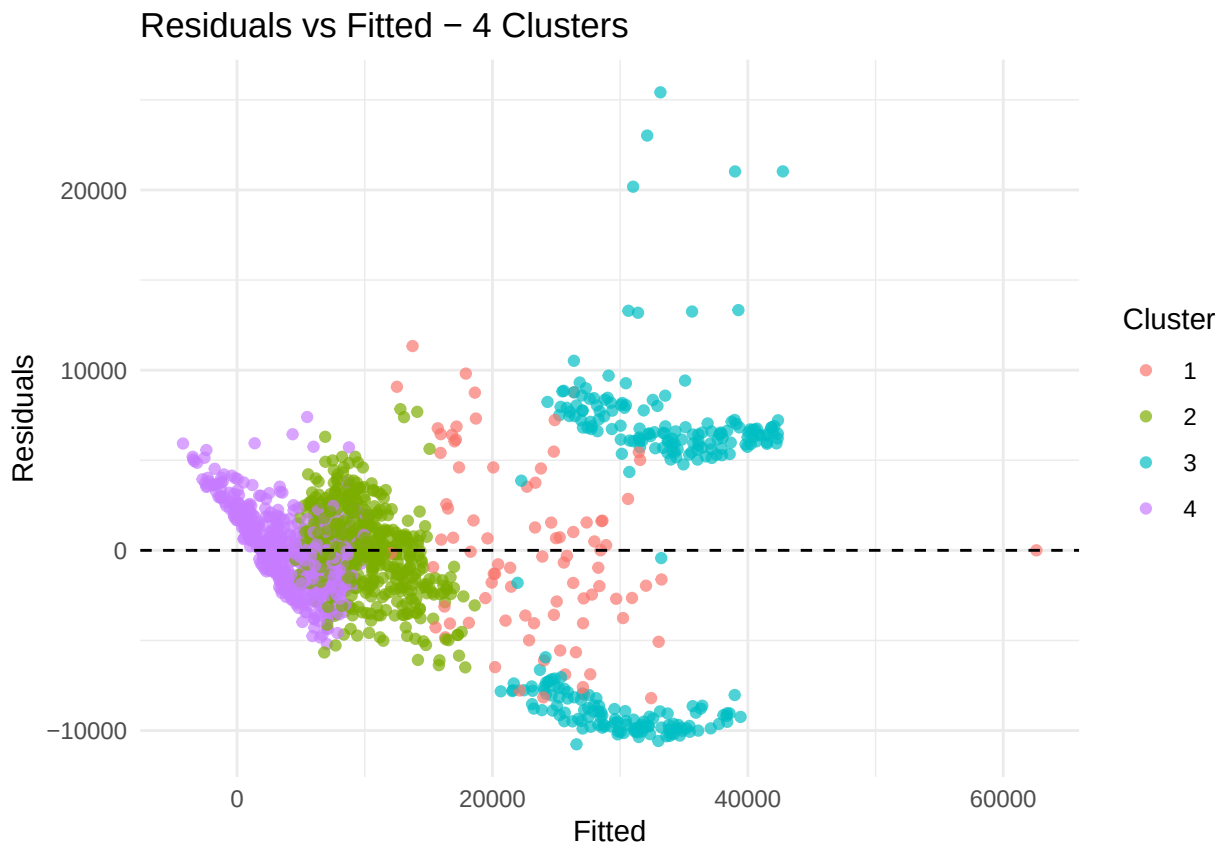
10. If the residual plots in (ix) are not adequate, increase the number of clusters in (vii) and repeat (ix) until the model is adequate. Once the model assumptions are fairly satisfied, proceed to summarize the results from model (xi) in one paragraph.

Solution:

```
library(ggplot2)

# Matrix for clustering
resid_fit_matrix <- cbind(resid(basic_lm_model), fitted(basic_lm_model))

# by looking at the graph in part (vii), optimal number of clusters, decided to test clusters from 4 to
for (k in 4:6) {
  cat("\nFitting model with", k, "clusters...\n")

  # perform k-means
  set.seed(42)
  kmeans_result <- kmeans(resid_fit_matrix, centers = k)
  insurance_data$cluster <- as.factor(kmeans_result$cluster)

  # fit model with clusters
  model_k <- lm(charges ~ age + bmi + children + sex + smoker + region + cluster, data = insurance_data
  summary_stats <- summary(model_k)

  # print summary info
  cat("Adjusted R-squared:", summary_stats$adj.r.squared, "\n")
  cat("Residual Std. Error:", summary_stats$sigma, "\n")
  print(summary(model_k))

  # plot Residuals vs Fitted
  resid_df <- data.frame(
    Fitted = fitted(model_k),
    Residuals = resid(model_k),
    Cluster = insurance_data$cluster
```

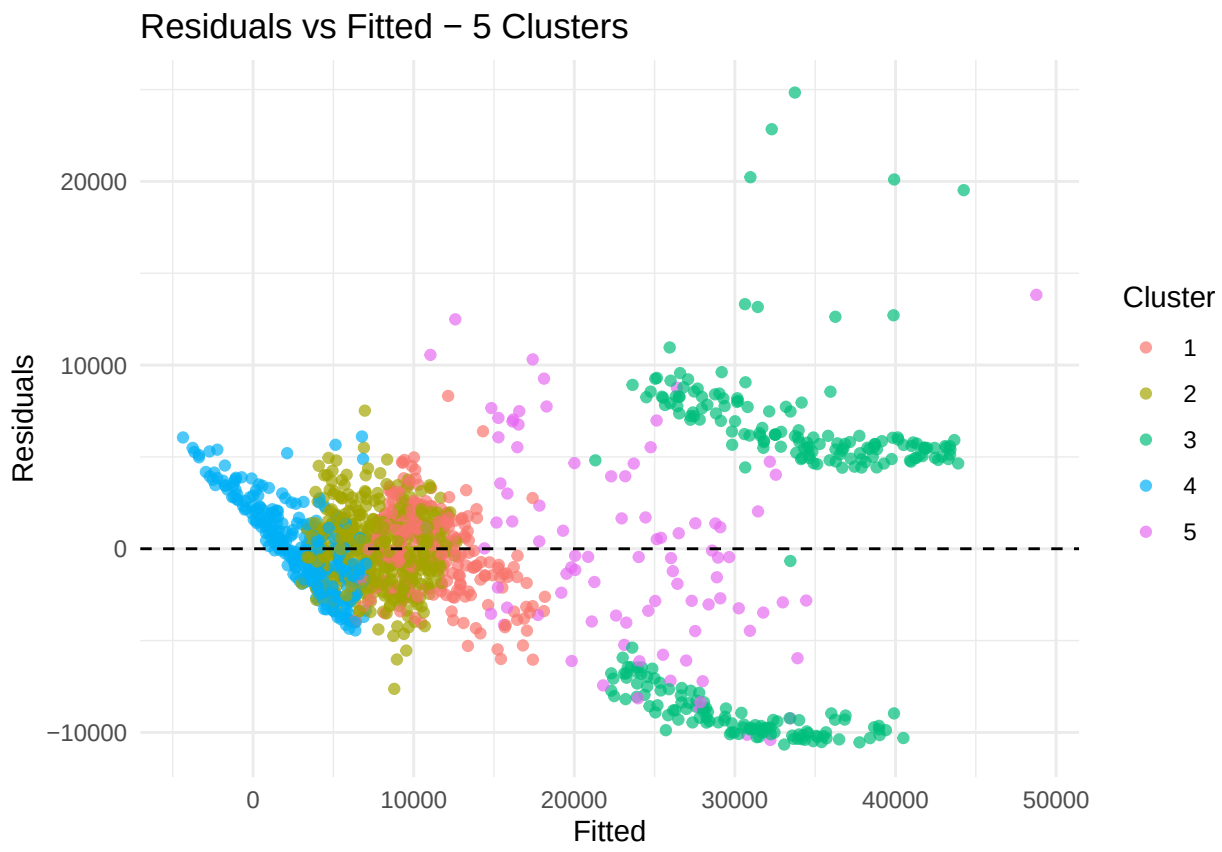
```
  )

  print(
    ggplot(resid_df, aes(x = Fitted, y = Residuals, color = Cluster)) +
      geom_point(alpha = 0.7) +
      geom_hline(yintercept = 0, linetype = "dashed") +
      labs(title = paste("Residuals vs Fitted -", k, "Clusters")) +
      theme_minimal()
  )
}
```

```
##
## Fitting model with 4 clusters...
## Adjusted R-squared: 0.8625721
## Residual Std. Error: 4489.334
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region + cluster, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10763.4  -1998.4     -5.7   2238.3  25415.9
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2434.78    1042.22  -2.336 0.019632 *
## age                318.76      11.67  27.317  < 2e-16 ***
## bmi                419.52      23.17  18.107  < 2e-16 ***
## children           444.79     102.70   4.331 1.60e-05 ***
## sexmale            -74.63     246.67  -0.303 0.762286
## smokeryes        39228.83    4524.15   8.671  < 2e-16 ***
## regionnorthwest   -153.89     352.85  -0.436 0.662804
## regionsoutheast  -1207.33     355.62  -3.395 0.000707 ***
## regionsouthwest   -398.15     354.60  -1.123 0.261731
## cluster2        -17478.77     543.93 -32.134  < 2e-16 ***
## cluster3        -29946.77    4505.00  -6.647 4.34e-11 ***
## cluster4        -14156.95     545.23 -25.965  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4489 on 1326 degrees of freedom
## Multiple R-squared:  0.8637, Adjusted R-squared:  0.8626
## F-statistic: 763.9 on 11 and 1326 DF,  p-value: < 2.2e-16
```



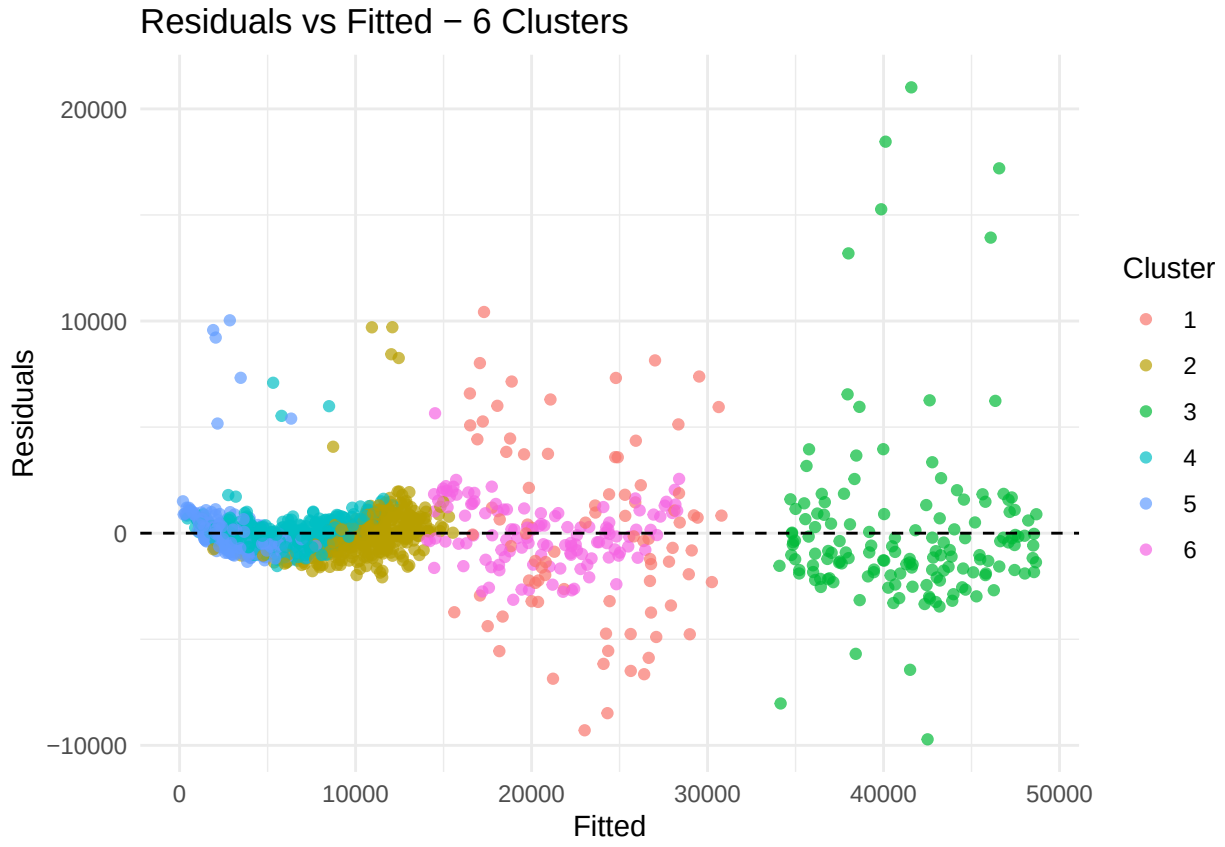

```
##
## Fitting model with 5 clusters...
## Adjusted R-squared: 0.867855
## Residual Std. Error: 4402.201
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region + cluster, data = insurance_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10658.8  -1962.0    -27.1   2054.9  24837.1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -24835.06    1298.19 -19.130  < 2e-16 ***
## age                349.53      12.36  28.277  < 2e-16 ***
## bmi                480.71      24.86  19.337  < 2e-16 ***
## children           540.57     102.59   5.269 1.60e-07 ***
## sexmale           -137.58     241.93  -0.569   0.5697
## smokeryes        25539.47    1773.03  14.404  < 2e-16 ***
## regionnorthwest   -365.38     346.72  -1.054   0.2922
## regionsoutheast  -1464.01     350.62  -4.175 3.17e-05 ***
## regionsouthwest   -767.64     349.23  -2.198   0.0281 *
## cluster2           3395.22     423.94   8.009 2.52e-15 ***
## cluster3           3340.85    1717.31   1.945   0.0519 .
```

```
## cluster4          6643.90     604.87  10.984  < 2e-16 ***
## cluster5         19343.76     590.77  32.743  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4402 on 1325 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8679
## F-statistic: 732.7 on 12 and 1325 DF,  p-value: < 2.2e-16
```




```
##
## Fitting model with 6 clusters...
## Adjusted R-squared: 0.971596
## Residual Std. Error: 2040.958
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region + cluster, data = insurance_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9716.8  -694.7  -161.0   435.3 21014.1
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8283.535    558.027  14.844  < 2e-16 ***
## age                287.619      5.854  49.131  < 2e-16 ***
```

```
## bmi                114.873     12.444   9.231  < 2e-16 ***
## children           441.431     47.083   9.376  < 2e-16 ***
## sexmale           -446.487    112.307  -3.976 7.40e-05 ***
## smokeryes        -1417.191    295.925  -4.789 1.86e-06 ***
## regionnorthwest    -88.955    160.541  -0.554 0.579608
## regionsoutheast   -788.534    162.704  -4.846 1.41e-06 ***
## regionsouthwest   -607.150    161.784  -3.753 0.000182 ***
## cluster2        -16406.246    266.089 -61.657  < 2e-16 ***
## cluster3         19517.022    274.789  71.025  < 2e-16 ***
## cluster4        -16062.752    246.185 -65.247  < 2e-16 ***
## cluster5        -14655.606    289.462 -50.630  < 2e-16 ***
## cluster6                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2041 on 1325 degrees of freedom
## Multiple R-squared:  0.9719, Adjusted R-squared:  0.9716
## F-statistic:  3812 on 12 and 1325 DF,  p-value: < 2.2e-16
```



- The model with 6 clusters gave the highest adjusted R^2 (0.97). This means it explains almost all the variation in charges. It is much better than the earlier models where k=3,4 and 5.
- Most predictors like age, BMI, number of children, and smoking status are still important and significant in this model.
- The residuals are now more evenly spread around the zero line, with less curve or funnel shape, suggesting the linearity and constant variance assumptions are better satisfied.
- Each cluster seems to group points with similar fitted values and residual patterns, helping to explain

different behaviors in the data that the original model didn't fully capture.
- Cluster 6 is dropped probably due to its effect is already covered by other predictors again.

11. Compare and explain the value of the R^2-adjusted in models (ii) and (ix).

Solution:

- In model (ii) (the original model without clusters), the adjusted R^2 was about 0.749, meaning the model explained around 74.9% of the variability in insurance charges after adjusting for the number of predictors.
- In model (ix) (with 6 clusters added), the adjusted R^2 jumped to 0.9716, which shows that the updated model can now explain over 97% of the variation in charges.

However, I am afraid whether the model over fits or not. Therefore, as a final touch, I will perform predictions with both models (basic linear and clustered with k = 6).

```
library(Metrics)

# data split into train and test functions
set.seed(42)
sample_idx <- sample(nrow(insurance_data), 0.8 * nrow(insurance_data))
train_data <- insurance_data[sample_idx, ]
test_data <- insurance_data[-sample_idx, ]

# get residuals and fitted values for training set (basic model)
basic_model_train <- lm(charges ~ age + bmi + children + sex + smoker + region, data = train_data)
resid_fit_matrix <- cbind(resid(basic_model_train), fitted(basic_model_train))

# perform k-means clustering on training residuals and fitted values
set.seed(42)
kmeans_result <- kmeans(resid_fit_matrix, centers = 6)
train_data$cluster <- as.factor(kmeans_result$cluster)

#train both models
model_basic <- lm(charges ~ age + bmi + children + sex + smoker + region, data = train_data)
model_clustered <- lm(charges ~ age + bmi + children + sex + smoker + region + cluster, data = train_da

# assign clusters to test set based on nearest center (this is approximate)
basic_model_test <- lm(charges ~ age + bmi + children + sex + smoker + region, data = test_data)
test_resid_fit <- cbind(resid(basic_model_test), fitted(basic_model_test))

# assign clusters to test set (find nearest cluster center)
assign_clusters <- function(points, centers) {
  apply(points, 1, function(p) {
    which.min(colSums((t(centers) - p)^2))
  })
}
test_data$cluster <- as.factor(assign_clusters(test_resid_fit, kmeans_result$centers))

# predict on test set
pred_basic <- predict(model_basic, newdata = test_data)
pred_clustered <- predict(model_clustered, newdata = test_data)

print("Basic Linear Model (ii)")
```

```
## [1] "Basic Linear Model (ii)"
```

```r
do.call(cbind, pracma::rmserr(test_data$charges, pred_basic))
```

```
##           mae      mse     rmse      mape      nmse      rstd
## [1,] 4148.585 35129129 5926.983 0.4647483 0.2424743 0.4546794
```

```r
print("Clustered Model k = 6 Model (iv)")
```

```
## [1] "Clustered Model k = 6 Model (iv)"
```

```r
do.call(cbind, pracma::rmserr(test_data$charges, pred_clustered))
```

```
##           mae     mse     rmse      mape       nmse      rstd
## [1,] 1115.609 3802839 1950.087 0.1203759 0.02624861 0.1495979
```

To sum up, the clustered model is a major improvement. It's more accurate, more consistent, and better at capturing complex patterns in the data. While the basic model struggled with large errors and group differences, adding clusters helped the model learn those differences and make predictions that are both closer to reality and more reliable.

Problem 2: Energy Data

Attached is energy.csv data file. This data is simulated in Ecotect for assessing the heating load requirements of 12 different building shapes. The data consist of 768 observations with 9 variables described in Table 2.

Table 2: Energy Data Variable Descriptions

No.	Variable Description
1	heating load
2	relative compactness
3	surface area
4	wall area
5	roof area
6	overall height
7	orientation
8	glazing area
9	glazing area distribution

1. Perform an exploratory analysis for this data.

Solution:

```r
# lets read the data first
energy_data <- read.csv("energy.csv")
head(energy_data)
```

```
##   relative_compactness surface_area wall_area roof_area overall_height
## 1                 0.98        514.5     294.0    110.25              7
## 2                 0.98        514.5     294.0    110.25              7
## 3                 0.98        514.5     294.0    110.25              7
## 4                 0.98        514.5     294.0    110.25              7
## 5                 0.90        563.5     318.5    122.50              7
## 6                 0.90        563.5     318.5    122.50              7
##   orientation glazing_area glazing_distribution HeatingLoad
## 1           2            0                    0       21.33
```

```
## 2           3           0                   0       21.33
## 3           4           0                   0       21.33
## 4           5           0                   0       21.33
## 5           2           0                   0       28.28
## 6           3           0                   0       25.38
```

```
# get the summary of the dataset
library(skimr)
skim(energy_data)
```

Table 6: Data summary

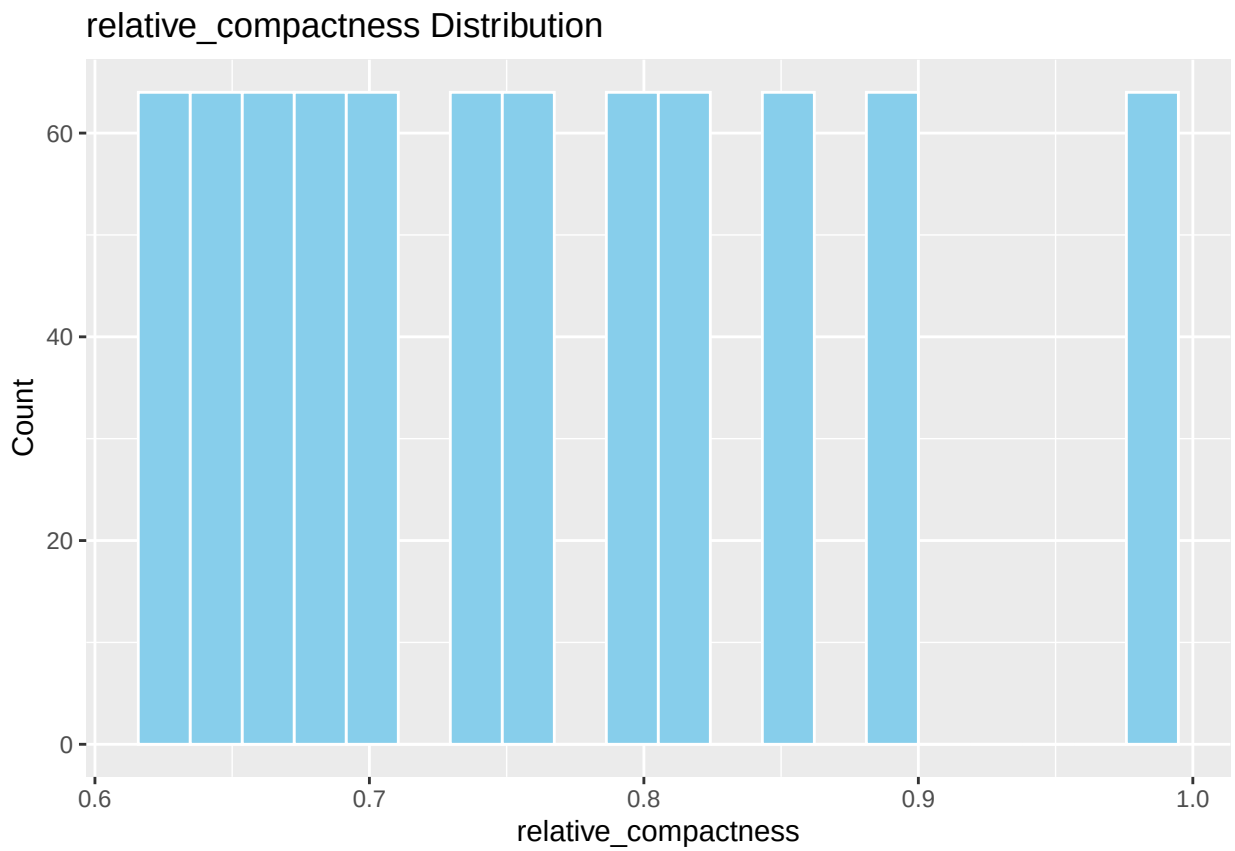
Name	energy_data
Number of rows	768
Number of columns	9
Column type frequency:	
numeric	9
Group variables	None

Variable type: numeric

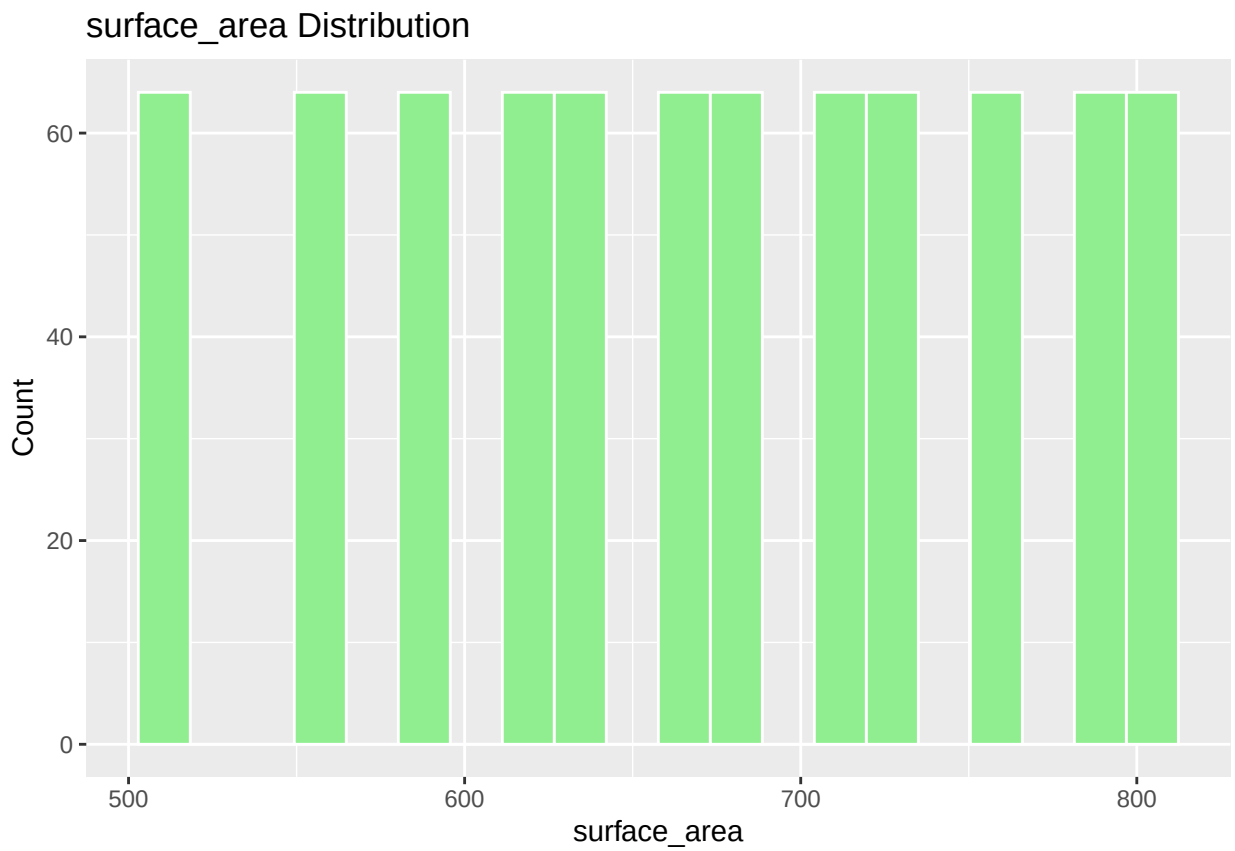
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
relative_compactness	0	1	0.76	0.11	0.62	0.68	0.75	0.83	0.98	
surface_area	0	1	671.71	88.09	514.50	606.38	673.75	741.12	808.50	
wall_area	0	1	318.50	43.63	245.00	294.00	318.50	343.00	416.50	
roof_area	0	1	176.60	45.17	110.25	140.88	183.75	220.50	220.50	
overall_height	0	1	5.25	1.75	3.50	3.50	5.25	7.00	7.00	
orientation	0	1	3.50	1.12	2.00	2.75	3.50	4.25	5.00	
glazing_area	0	1	0.23	0.13	0.00	0.10	0.25	0.40	0.40	
glazing_distribution	0	1	2.81	1.55	0.00	1.75	3.00	4.00	5.00	
HeatingLoad	0	1	24.59	9.51	10.90	15.62	22.08	33.13	48.03	

To explore the energy dataset, I used the skimr package because it gives a more organized and detailed summary than the basic summary() function. The dataset contains 768 observations and 9 numeric variables, with no missing values, meaning it's completely clean. The target variable, HeatingLoad, is right-skewed and ranges from around 10.9 to 48.0, showing that some buildings require much more energy than others. Variables like relative compactness, surface area, and wall area are fairly symmetric, while overall height appears to be bimodal, likely representing two types of building designs. Other features like orientation, glazing area, and glazing distribution also vary in structured ways, which could be helpful for predicting energy efficiency.

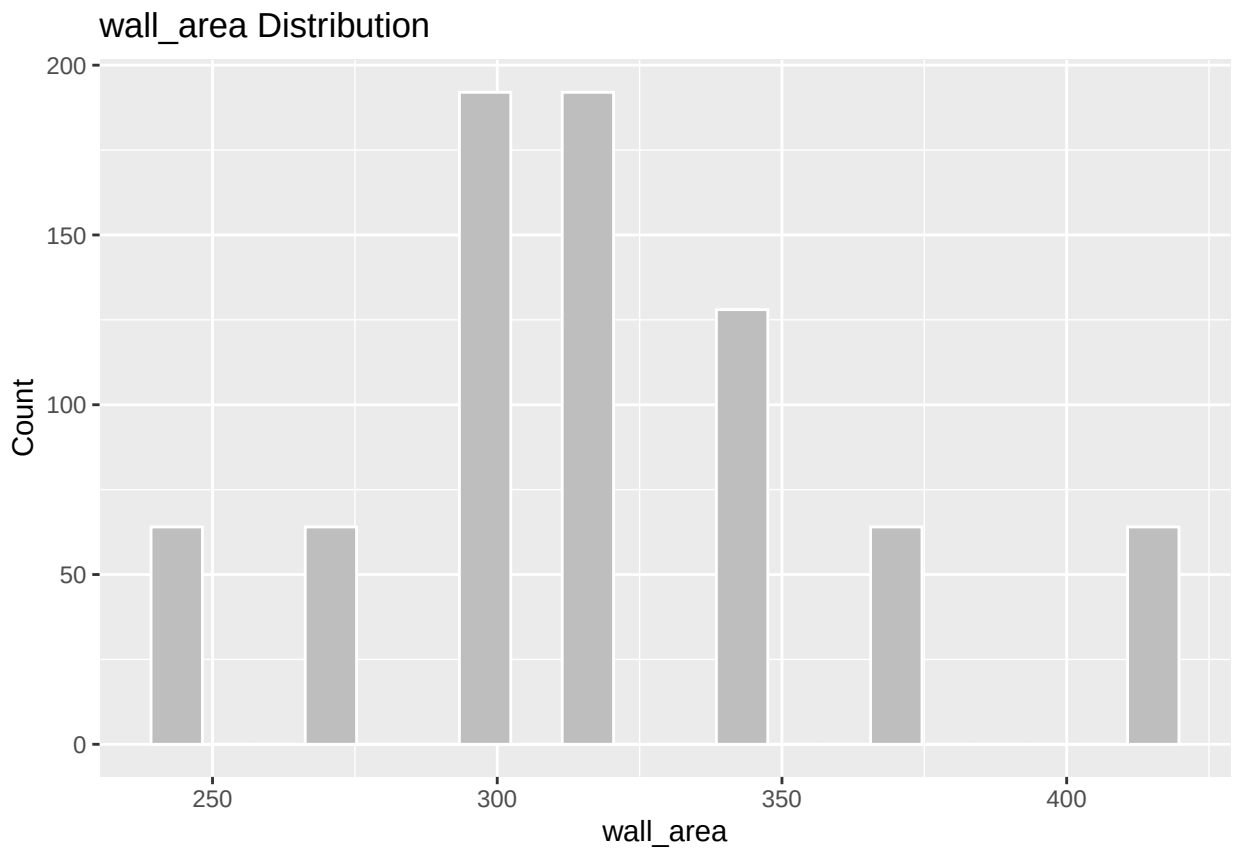
```
# Lets check the histogram of the continuos variables
library(ggplot2)

# Example: Histogram for 'age'
ggplot(energy_data, aes(x = relative_compactness)) +
  geom_histogram(fill = "skyblue", color = "white", bins = 20) +
  labs(title = "relative_compactness Distribution", x = "relative_compactness", y = "Count")
```



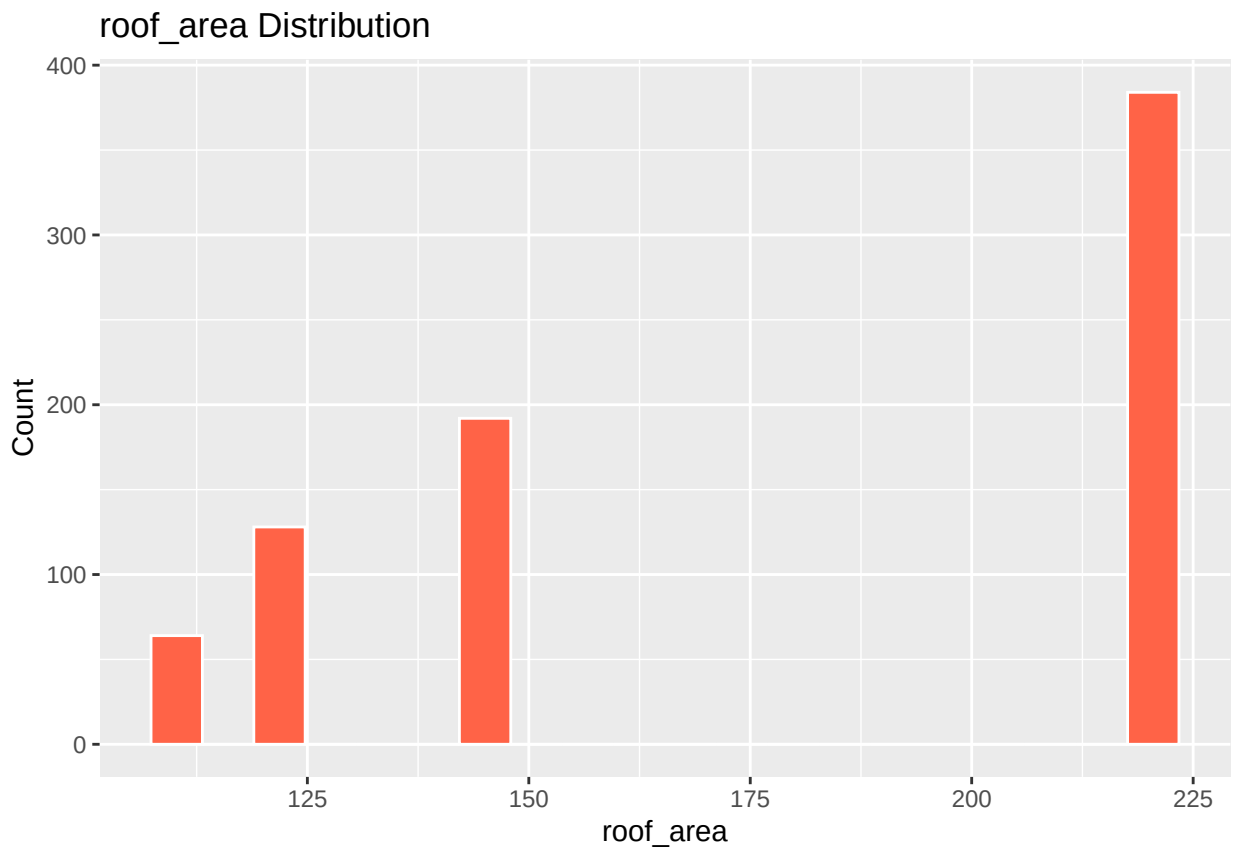

```
# Repeat for 'bmi' and 'charges'
ggplot(energy_data, aes(x = surface_area)) +
  geom_histogram(fill = "lightgreen", color = "white", bins = 20) +
  labs(title = "surface_area Distribution", x = "surface_area", y = "Count")
```



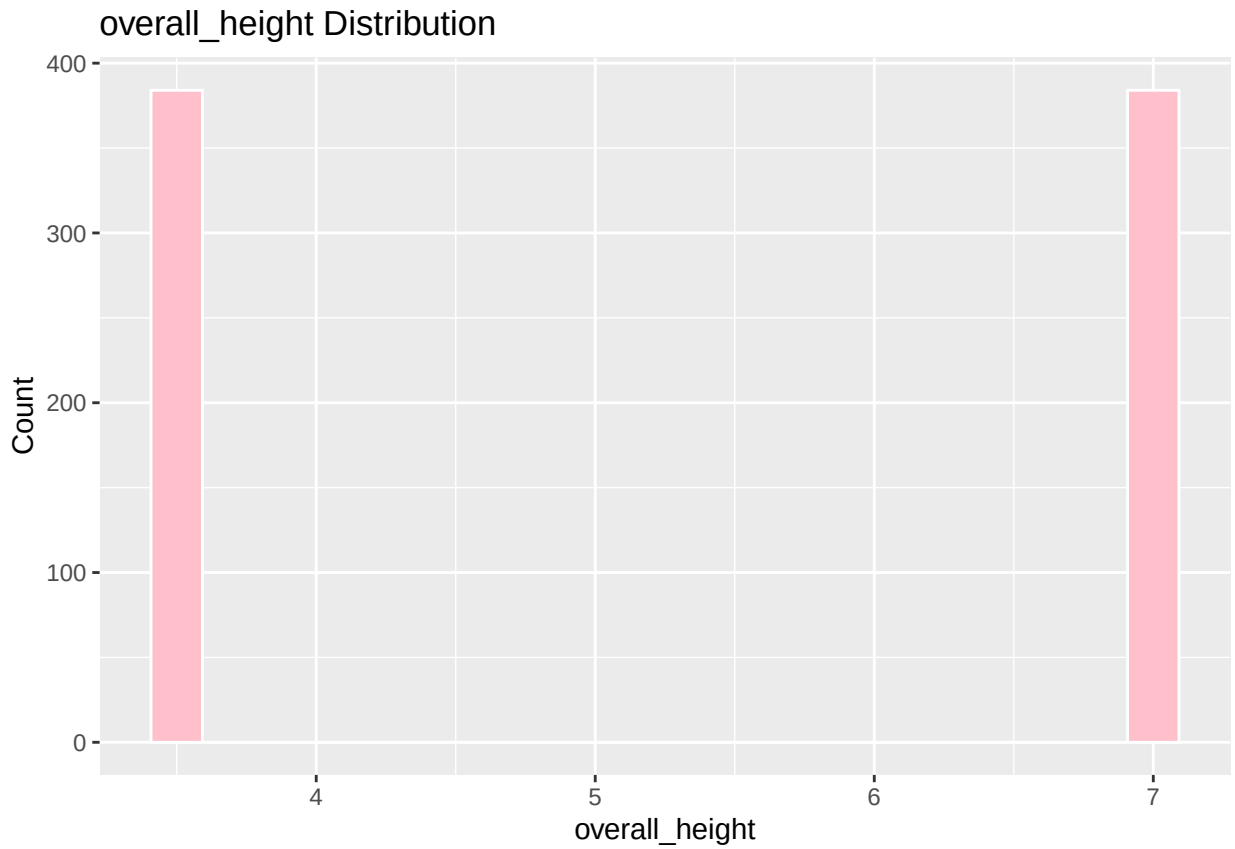

```
ggplot(energy_data, aes(x = wall_area)) +
  geom_histogram(fill = "gray", color = "white", bins = 20) +
  labs(title = "wall_area Distribution", x = "wall_area", y = "Count")
```



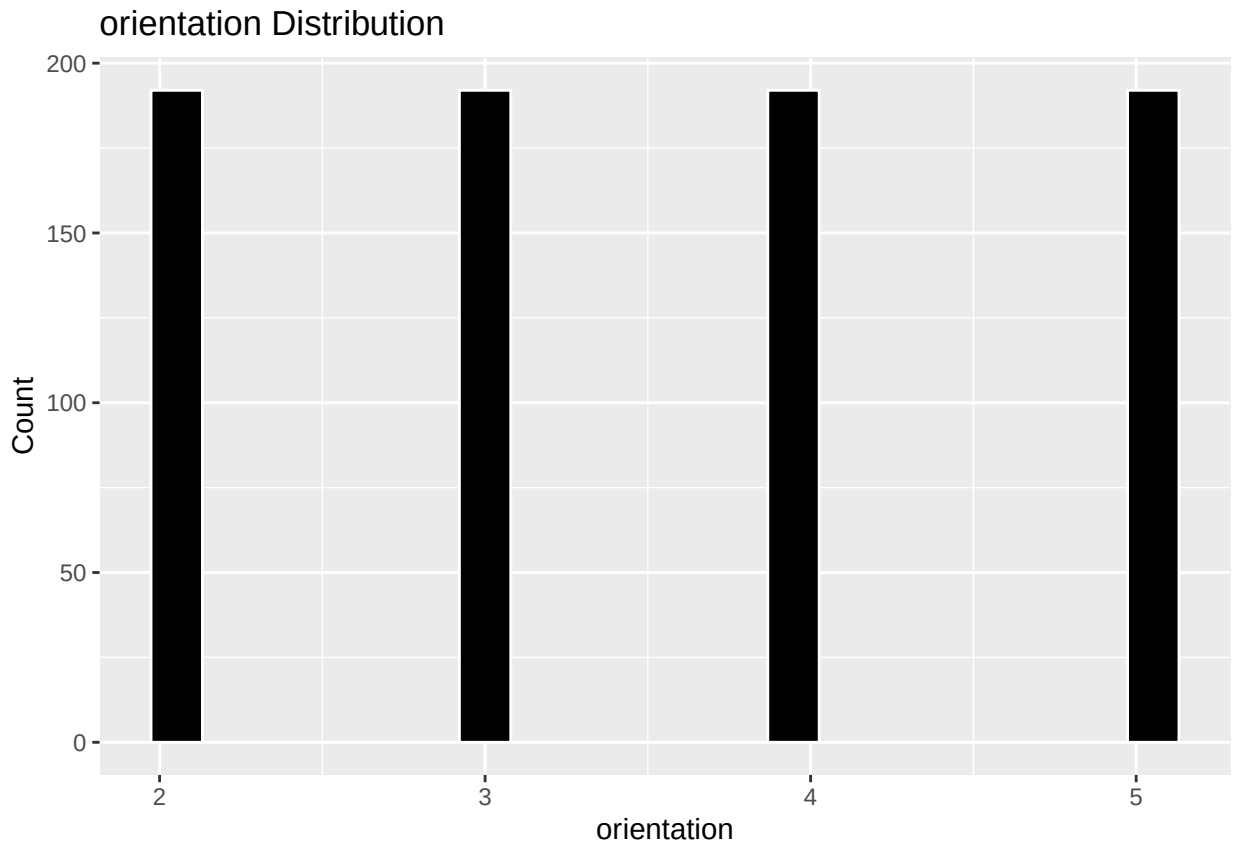

```
ggplot(energy_data, aes(x = roof_area)) +
  geom_histogram(fill = "tomato", color = "white", bins = 20) +
  labs(title = "roof_area Distribution", x = "roof_area", y = "Count")
```



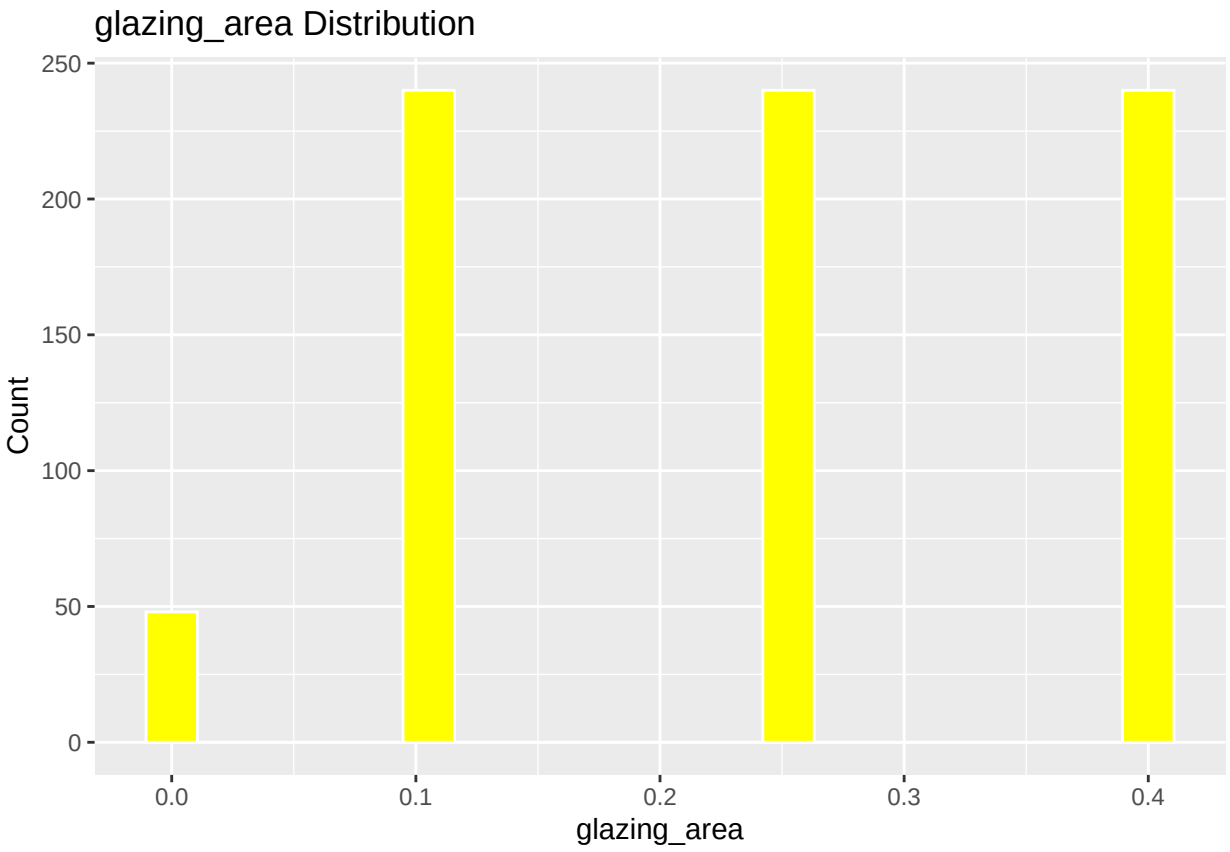

```
ggplot(energy_data, aes(x = overall_height)) +
  geom_histogram(fill = "pink", color = "white", bins = 20) +
  labs(title = "overall_height Distribution", x = "overall_height", y = "Count")
```



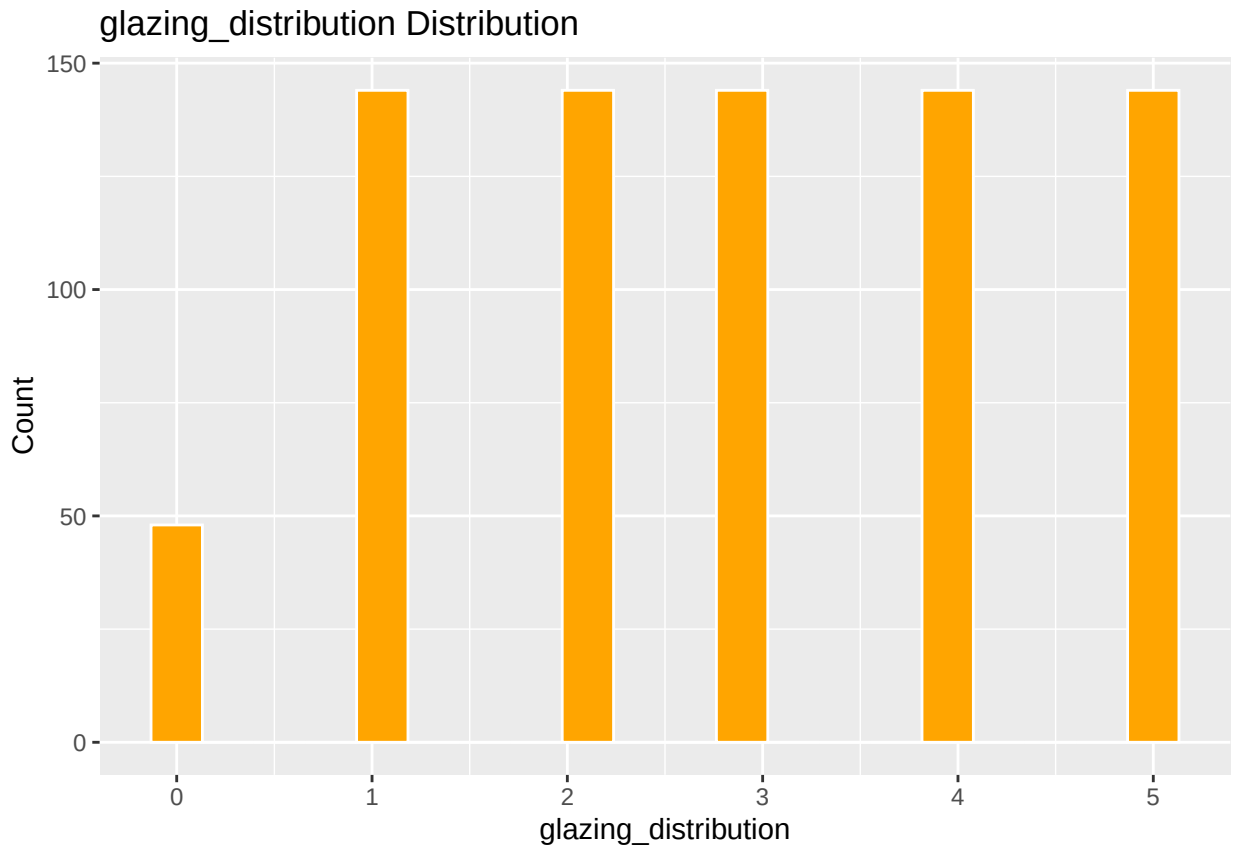

```
ggplot(energy_data, aes(x = orientation)) +
  geom_histogram(fill = "black", color = "white", bins = 20) +
  labs(title = "orientation Distribution", x = "orientation", y = "Count")
```




```
ggplot(energy_data, aes(x = glazing_area)) +
  geom_histogram(fill = "yellow", color = "white", bins = 20) +
  labs(title = "glazing_area Distribution", x = "glazing_area", y = "Count")
```




```
ggplot(energy_data, aes(x = glazing_distribution)) +
  geom_histogram(fill = "orange", color = "white", bins = 20) +
  labs(title = "glazing_distribution Distribution", x = "glazing_distribution", y = "Count")
```



Relative Compactness Distribution: The distribution of relative compactness appears uniform across its range from ~0.62 to ~0.98, with each bin containing a similar number of observations. This suggests the dataset was likely designed or sampled to equally represent different building compactness levels.

Surface Area Distribution: Like relative compactness, surface area is uniformly distributed from ~510 to ~810, meaning all surface sizes are equally represented. This might help in evaluating the impact of surface area on heating load without bias toward a specific size range.

Wall Area Distribution: Wall area shows a bimodal distribution with two peaks around 300 and 310, and fewer buildings at the lower and higher ends. This indicates that most buildings have medium-sized wall areas, with some variety on both ends.

Roof Area Distribution: This variable has a discrete distribution with just a few distinct values, and a very strong peak at ~220. That means most buildings share the same roof area, likely due to specific architectural standards or simulation design.

Overall Height Distribution: The dataset includes only two height levels (3.5 and 7), with equal representation. This clearly indicates that buildings are either single- or double-story, and both are evenly distributed.

Orientation Distribution: Orientation values range from 2 to 5 and are perfectly balanced, with an equal number of observations for each. This ensures that each directional orientation is equally studied, useful for analyzing sunlight effects.

Glazing Area Distribution: The glazing area has four dominant levels (0, 0.1, 0.25, and 0.4), with no values in between. Most buildings include some glazing, but the variety is limited to discrete options, which is common in simulation datasets.

Glazing Distribution: Glazing distribution also follows a discrete pattern from 0 to 5, with a slightly lower count at 0 and equal representation across levels 1–5. This shows variation in how glazing is applied

spatially on buildings, offering insights into design variation.

In this analysis, I will treat orientation and glazing_area_distribution as categorical variables because they represent discrete, non-continuous values that correspond to distinct categories rather than numeric magnitudes. Orientation indicates the direction a building faces (e.g., north, east), which has qualitative implications on energy use rather than a linear numeric relationship. Similarly, glazing_area_distribution represents predefined types of window placements, which are categorical in nature and should not be interpreted as continuous numerical values. Converting these variables to factors allows the model to properly account for their distinct levels without imposing a false numeric order. I also treat overall_height as categorical.

```
# Convert to factors
energy_data$orientation <- as.factor(energy_data$orientation)
energy_data$glazing_distribution <- as.factor(energy_data$glazing_distribution)
energy_data$overall_height <- as.factor(energy_data$overall_height)

# Check structure to confirm
str(energy_data)
```

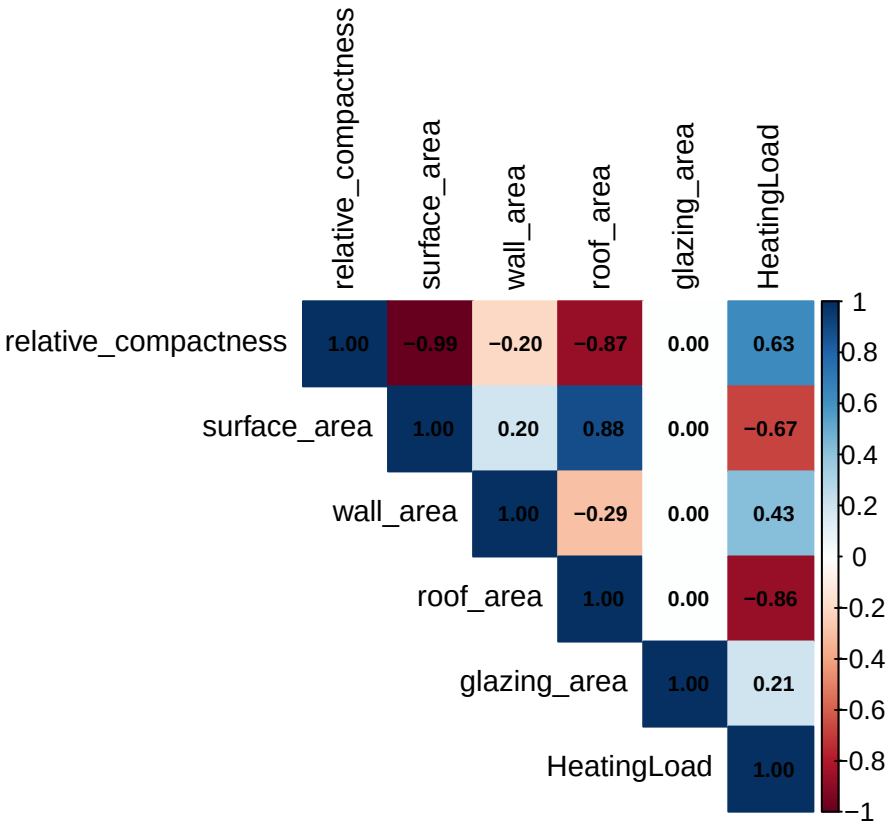
```
## 'data.frame':    768 obs. of  9 variables:
##  $ relative_compactness: num  0.98 0.98 0.98 0.98 0.9 0.9 0.9 0.9 0.86 0.86 ...
##  $ surface_area        : num  514 514 514 514 564 ...
##  $ wall_area           : num  294 294 294 294 318 ...
##  $ roof_area           : num  110 110 110 110 122 ...
##  $ overall_height      : Factor w/ 2 levels "3.5","7": 2 2 2 2 2 2 2 2 2 2 ...
##  $ orientation         : Factor w/ 4 levels "2","3","4","5": 1 2 3 4 1 2 3 4 1 2 ...
##  $ glazing_area        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ glazing_distribution: Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ HeatingLoad         : num  21.3 21.3 21.3 21.3 28.3 ...
```

```
numeric_data <- energy_data[, sapply(energy_data, is.numeric)]
cor_matrix <- cor(numeric_data)
print(cor_matrix)
```

```
##                      relative_compactness  surface_area  wall_area
## relative_compactness         1.000000e+00 -9.919015e-01 -0.2037817
## surface_area                -9.919015e-01  1.000000e+00  0.1955016
## wall_area                   -2.037817e-01  1.955016e-01  1.0000000
## roof_area                   -8.688234e-01  8.807195e-01 -0.2923165
## glazing_area                 7.617400e-20  4.664140e-20  0.0000000
## HeatingLoad                  6.343391e-01 -6.729989e-01  0.4271170
##                            roof_area  glazing_area HeatingLoad
## relative_compactness -8.688234e-01  7.617400e-20   0.6343391
## surface_area          8.807195e-01  4.664140e-20  -0.6729989
## wall_area            -2.923165e-01  0.000000e+00   0.4271170
## roof_area             1.000000e+00 -1.197187e-19  -0.8625466
## glazing_area         -1.197187e-19  1.000000e+00   0.2075050
## HeatingLoad          -8.625466e-01  2.075050e-01   1.0000000
```

```
library(corrplot)
# Bigger and clearer corrplot
corrplot(cor_matrix,
         method = "color",
         type = "upper",
         tl.col = "black",
         tl.cex = 0.9,          # text label size
         number.cex = 0.7,      # correlation coefficient size
```

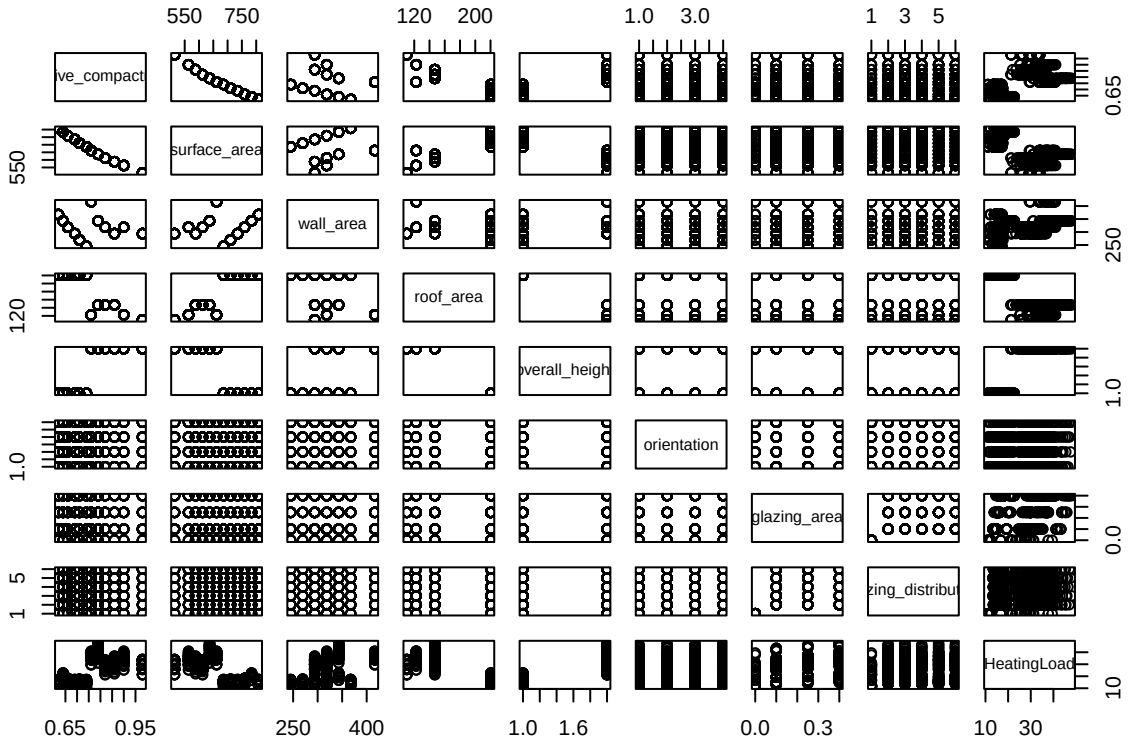
```
        addCoef.col = "black",
        mar = c(0, 0, 1, 0))    # reduce margins if needed
```



- There are strong correlations among some building design features — for example, relative compactness, surface area, roof area are highly related to each other.

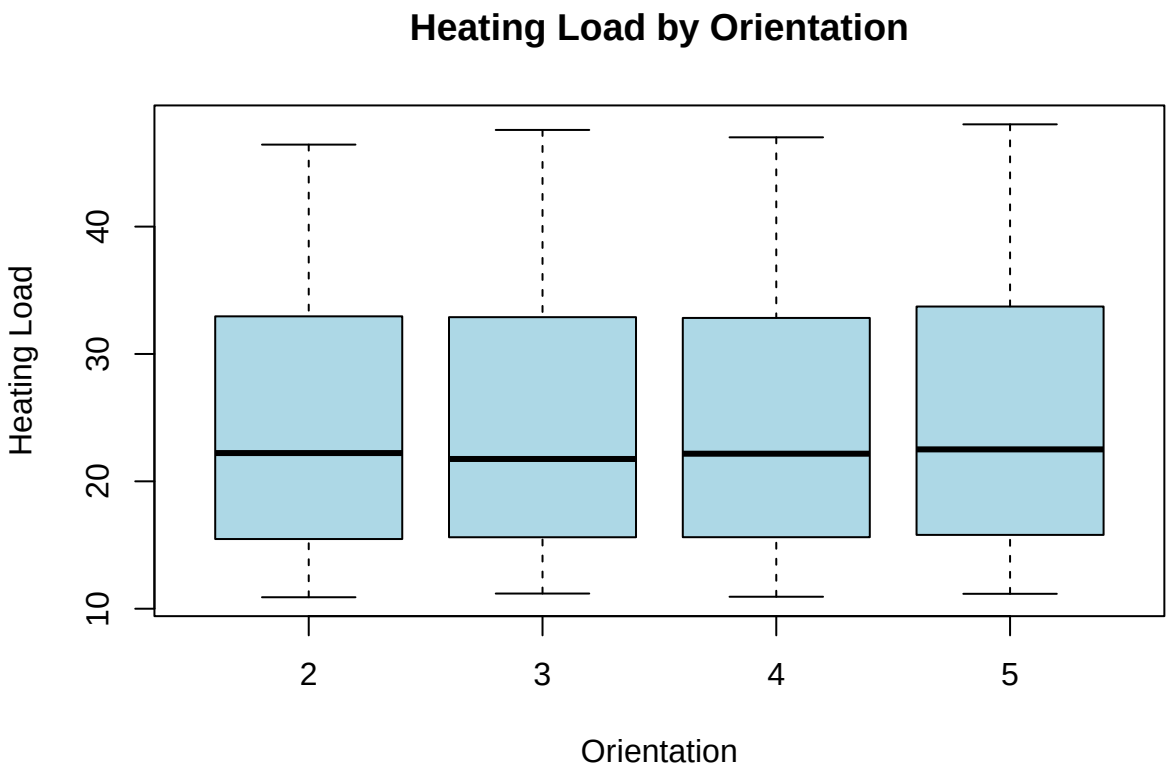
- HeatingLoad is most positively correlated with relative compactness (0.63), and most negatively correlated with roof area (-0.86) and surface area (-0.67).

```
pairs(energy_data)
```

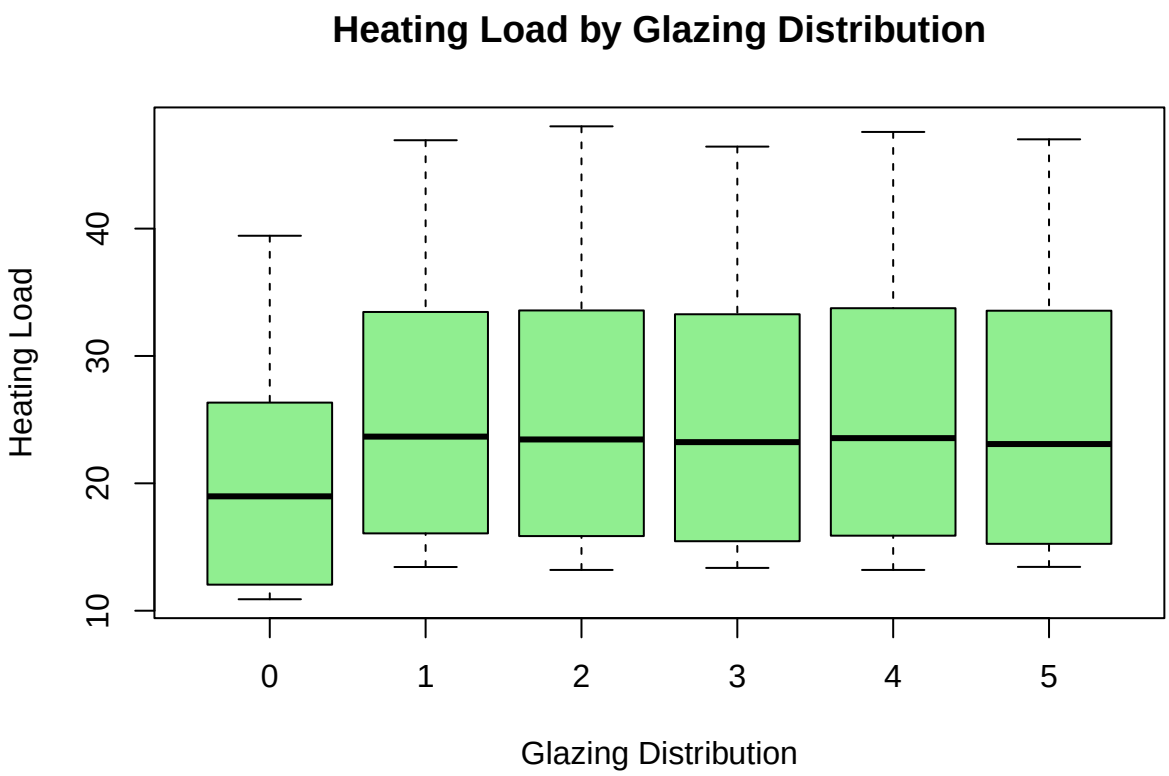


- There's a clear negative linear relationship between relative compactness and surface area, and also between relative compactness and roof area.
- Surface area and roof area are positively correlated.
- HeatingLoad increases with relative compactness, consistent with earlier correlations.
- Wall area, orientation, glazing area, and glazing distribution don't show clear linear relationships with HeatingLoad, indicating their effects may be weaker or non-linear.

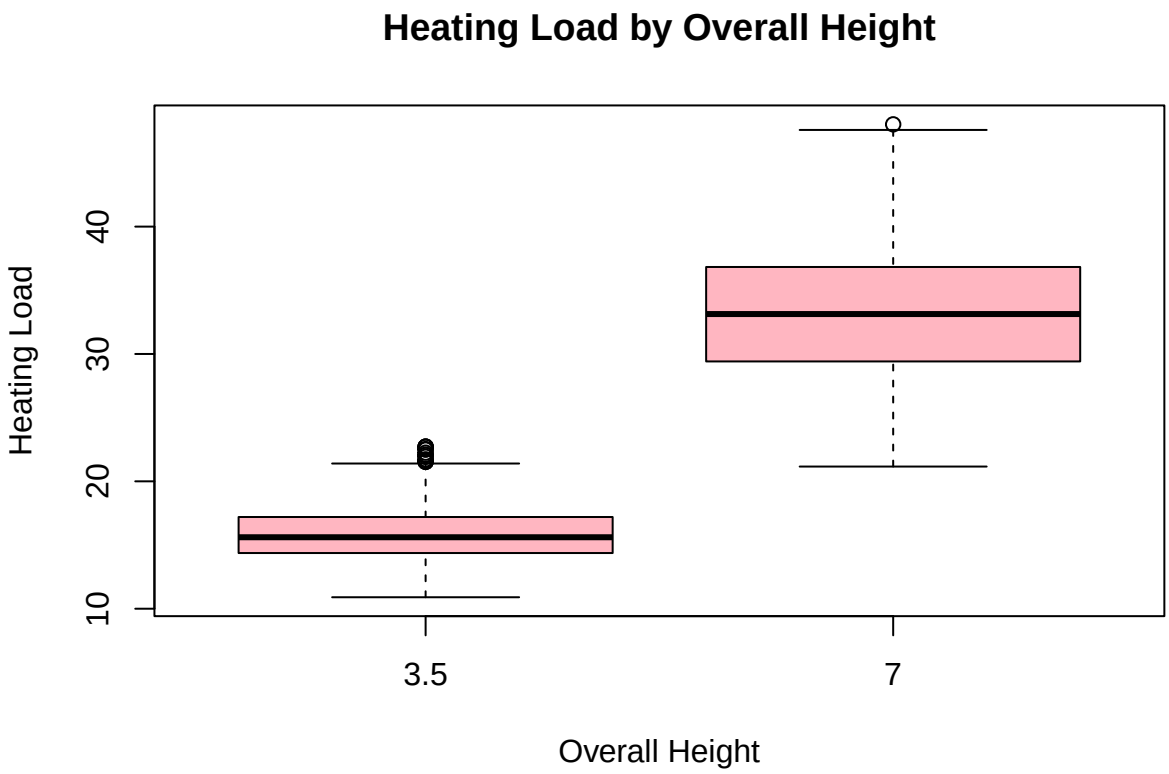
```r
# Orientation vs Heating Load
boxplot(HeatingLoad ~ orientation, data = energy_data,
        main = "Heating Load by Orientation",
        ylab = "Heating Load", xlab = "Orientation",
        col = "lightblue")
```




```r
# Glazing Distribution vs Heating Load
boxplot(HeatingLoad ~ glazing_distribution, data = energy_data,
        main = "Heating Load by Glazing Distribution",
        ylab = "Heating Load", xlab = "Glazing Distribution",
        col = "lightgreen")
```




```r
# Overall Height vs Heating Load
boxplot(HeatingLoad ~ overall_height, data = energy_data,
        main = "Heating Load by Overall Height",
        ylab = "Heating Load", xlab = "Overall Height",
        col = "lightpink")
```



- Orientation: Heating load is consistent across all orientation categories.It shows no strong influence of orientation on heating demand.

- Glazing Distribution: Higher glazing distribution levels generally have a slightly higher median heating load, especially when compared to 0 (no glazing).
- Overall Height: Buildings with an overall height of 7 have significantly higher heating loads than those with a height of 3.5. THis shows that height is an important factor.

2. Fit an appropriate linear model with heating load as response and the remaining variables as predictors.

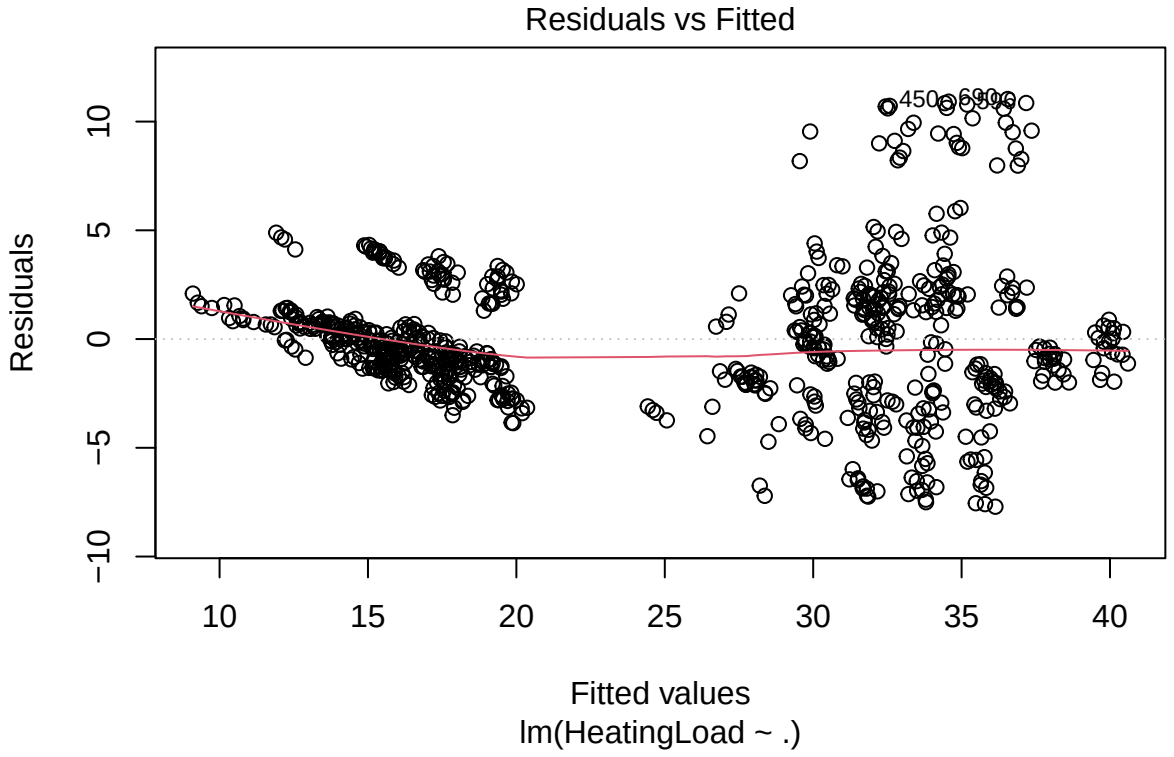
Solution: In this example, I did not continue with the best linear model. I used the most basic linear model:

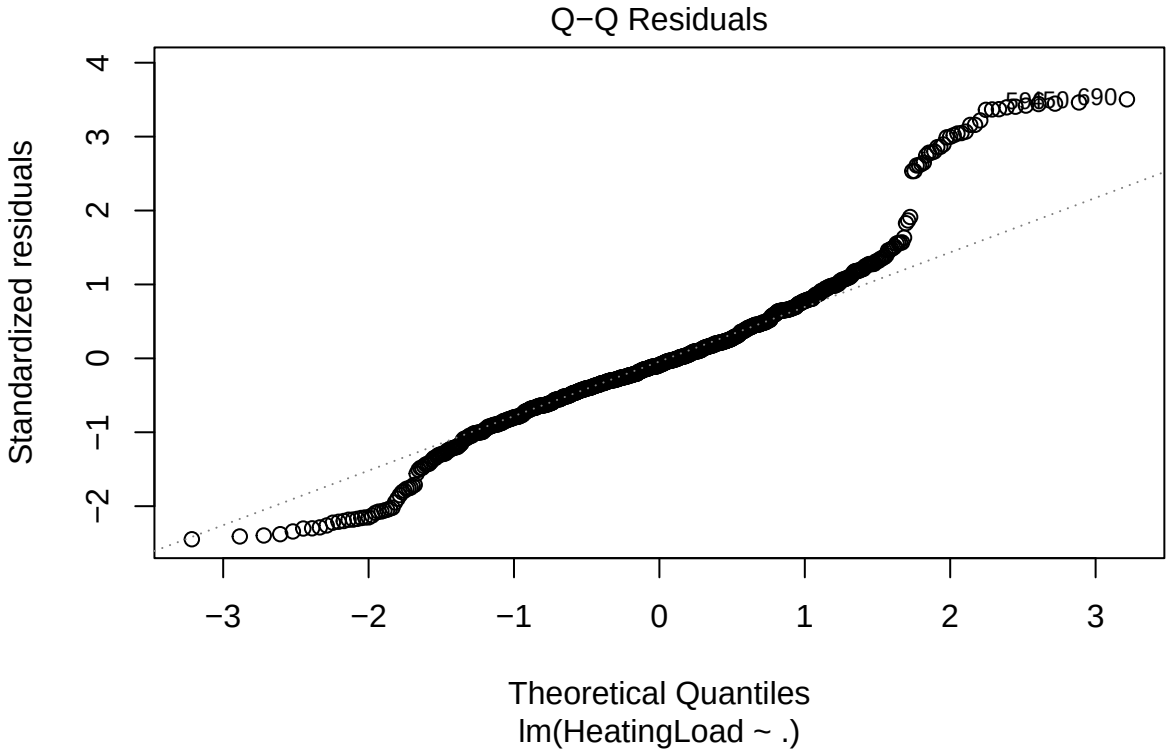
```
basic_lm_model  <- lm(HeatingLoad ~ ., data = energy_data)
summary(basic_lm_model)
```

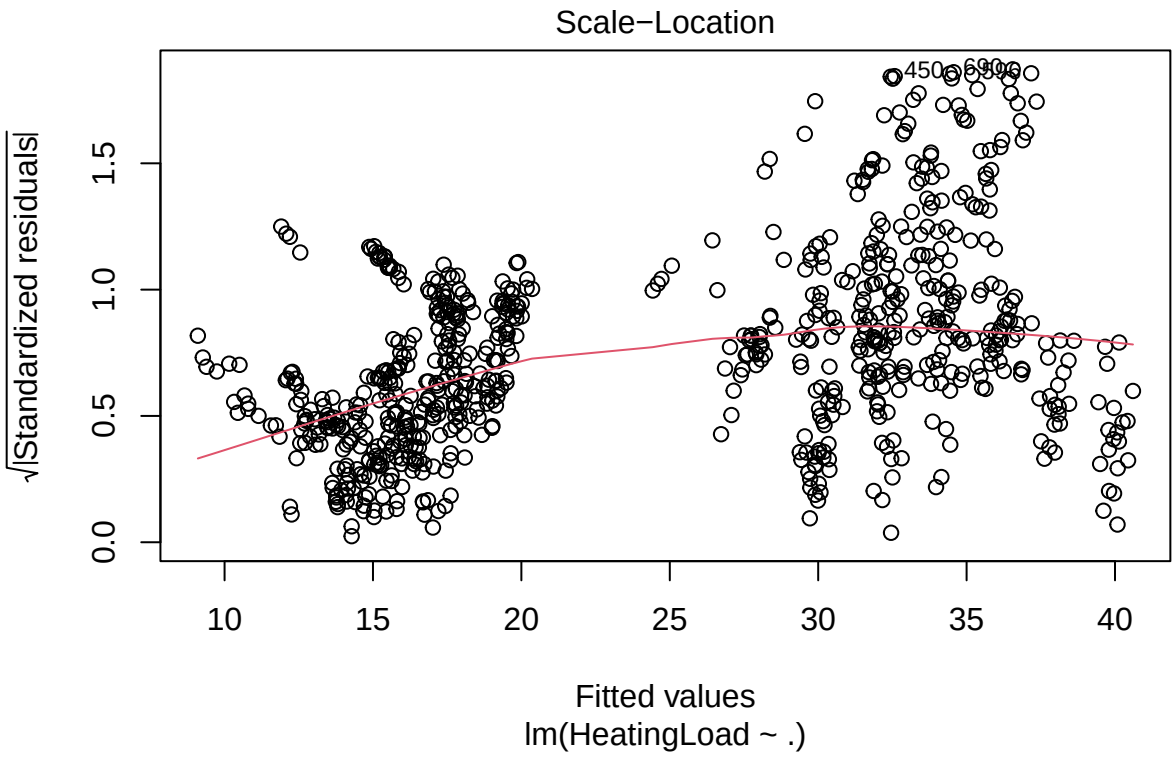
```
##
## Call:
## lm(formula = HeatingLoad ~ ., data = energy_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7082 -1.6996 -0.2814  1.4302 11.0365
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           111.363522  19.655078   5.666 2.08e-08 ***
## relative_compactness  -70.787707  11.138038  -6.355 3.59e-10 ***
```

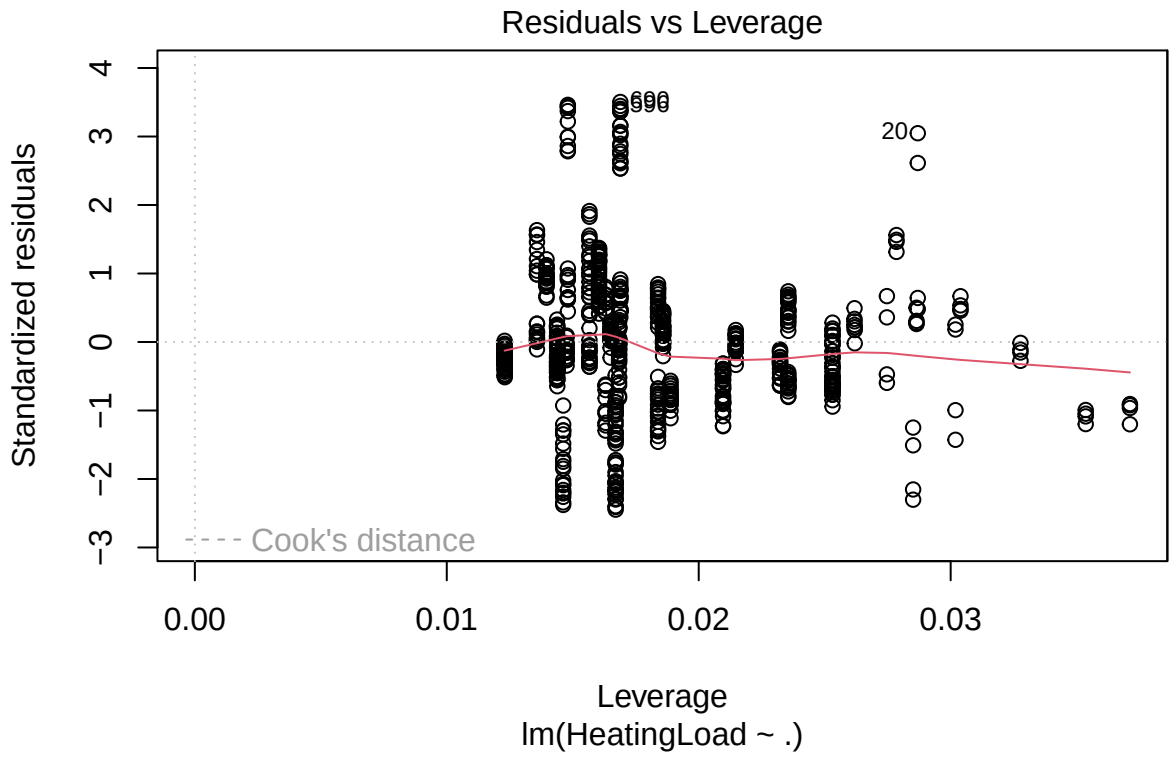
```
## surface_area           -0.088245   0.018484  -4.774 2.17e-06 ***
## wall_area               0.044682   0.007196   6.209 8.80e-10 ***
## roof_area                     NA         NA      NA       NA
## overall_height7        14.993452   1.280527  11.709  < 2e-16 ***
## orientation3           -0.291979   0.324182  -0.901 0.368054
## orientation4           -0.124219   0.324182  -0.383 0.701697
## orientation5            0.349115   0.324182   1.077 0.281865
## glazing_area           13.252917   0.966523  13.712  < 2e-16 ***
## glazing_distribution1   2.160035   0.581924   3.712 0.000221 ***
## glazing_distribution2   1.977396   0.581924   3.398 0.000714 ***
## glazing_distribution3   1.639965   0.581924   2.818 0.004956 **
## glazing_distribution4   1.995660   0.581924   3.429 0.000638 ***
## glazing_distribution5   1.695521   0.581924   2.914 0.003678 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.176 on 754 degrees of freedom
## Multiple R-squared:  0.8904, Adjusted R-squared:  0.8885
## F-statistic: 471.3 on 13 and 754 DF,  p-value: < 2.2e-16
```

```
plot(basic_lm_model)
```




Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
3
4
−3
−2
−1
0
1
2
3
690
Theoretical Quantiles
lm(HeatingLoad ~ .)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
15
20
25
30
35
40
Fitted values
lm(HeatingLoad ~ .)



3. What is the likely cause of the NA's in the parameter estimates? Explain!

Solution:

- roof_area is dropped since it can be linearly linearly dependent on variables like relative_compactness, surface_area, or overall_height as we saw in the correlation matrix on part (i). When one variable can be exactly or nearly exactly calculated from others, R automatically drops it from the model to avoid singularity issues in matrix inversion during model estimation. That is why we saw NA in the roof_area part in the model summary.

4. What formal check can be used to confirm the issue you stated in (iii)?

Solution:

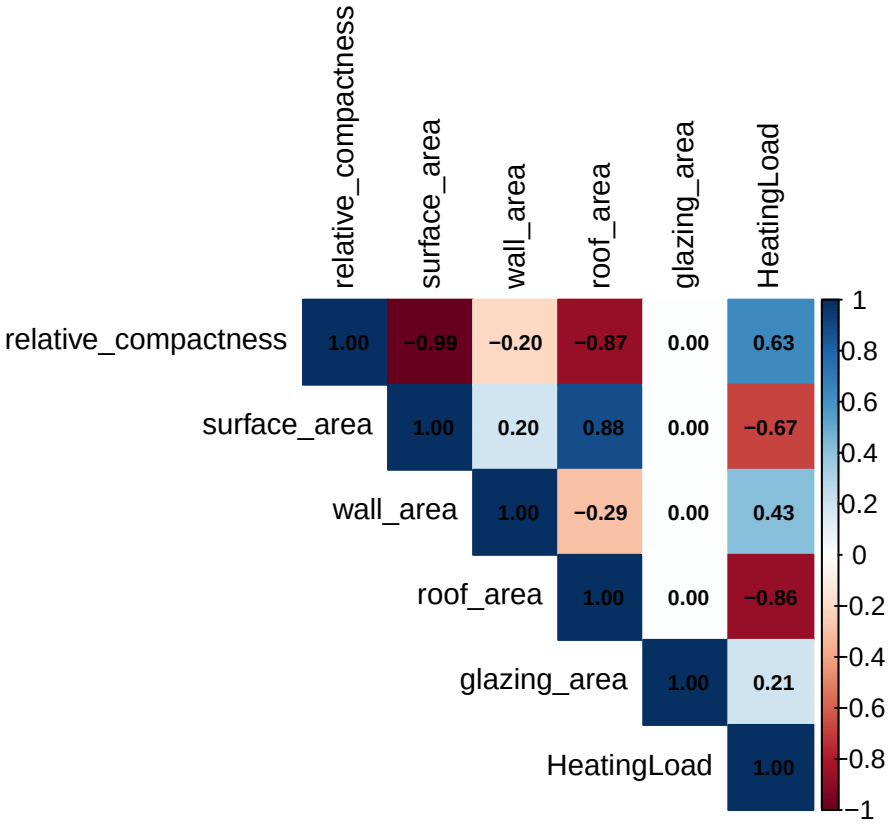
As a formal check to the issue stated in part (iii), as mentioned correlation can be performed.

```
numeric_data <- energy_data[, sapply(energy_data, is.numeric)]
cor_matrix <- cor(numeric_data)
print(cor_matrix)
```

```
##                      relative_compactness  surface_area  wall_area
## relative_compactness         1.000000e+00 -9.919015e-01 -0.2037817
## surface_area                -9.919015e-01  1.000000e+00  0.1955016
## wall_area                   -2.037817e-01  1.955016e-01  1.0000000
## roof_area                   -8.688234e-01  8.807195e-01 -0.2923165
## glazing_area                 7.617400e-20  4.664140e-20  0.0000000
## HeatingLoad                  6.343391e-01 -6.729989e-01  0.4271170
##                         roof_area  glazing_area HeatingLoad
## relative_compactness -8.688234e-01  7.617400e-20   0.6343391
```

```
## surface_area          8.807195e-01  4.664140e-20  -0.6729989
## wall_area            -2.923165e-01  0.000000e+00   0.4271170
## roof_area             1.000000e+00 -1.197187e-19  -0.8625466
## glazing_area         -1.197187e-19  1.000000e+00   0.2075050
## HeatingLoad          -8.625466e-01  2.075050e-01   1.0000000
```

```r
library(corrplot)
# Bigger and clearer corrplot
corrplot(cor_matrix,
         method = "color",
         type = "upper",
         tl.col = "black",
         tl.cex = 0.9,          # text label size
         number.cex = 0.7,      # correlation coefficient size
         addCoef.col = "black",
         mar = c(0, 0, 1, 0))   # reduce margins if needed
```



- As seen, there is a stong relationship between roof_area and relative_compactness: -0.87, roof_area and surface_area: 0.88 and. These corrlation values show us the linearly dependence between the predictor variables.

As a second formal check, the rank of the model matrix can be calculated.

```r
# Get the model matrix
X <- model.matrix(basic_lm_model)

# Compute the rank
qr(X)$rank
```

```
## [1] 14
```

```
ncol(X)
```

```
## [1] 15
```

- qr(X)$rank This computes the rank of the model matrix X using QR decomposition. The rank tells you how many linearly independent columns are in the matrix. If the rank is less than the number of columns, that means some variables are perfectly collinear (i.e., redundant).
- ncol(X) This returns the number of columns in the model matrix, which includes the intercept and all predictors used in the model.

Comparing these two helped me to detect perfect multicollinearity. If qr(X)$rank < ncol(X), one or more predictors are not estimable and are dropped automatically by R during model fitting.

```
alias(basic_lm_model)
```

```
## Model :
## HeatingLoad ~ relative_compactness + surface_area + wall_area +
##     roof_area + overall_height + orientation + glazing_area +
##     glazing_distribution
##
## Complete :
##           (Intercept) relative_compactness surface_area wall_area
## roof_area    0           0                  1/2          -1/2
##           overall_height7 orientation3 orientation4 orientation5 glazing_area
## roof_area    0               0            0            0            0
##           glazing_distribution1 glazing_distribution2 glazing_distribution3
## roof_area    0                     0                     0
##           glazing_distribution4 glazing_distribution5
## roof_area    0                     0
```

- alias(basic_lm_model) shows which variables are perfectly predictable from others in the model — a condition known as perfect multicollinearity.
- It confirms that roof_area is a linear combination of surface_area and wall_area. That's why it was dropped from the model

5. Refit the model in (ii) without the variable(s) whose parameters are not estimable.

Solution:

```
# Refit the model without roof_area
model_refit <- lm(HeatingLoad ~ . -roof_area,
                  data = energy_data)

# View summary
summary(model_refit)
```

```
##
## Call:
## lm(formula = HeatingLoad ~ . - roof_area, data = energy_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7082 -1.6996 -0.2814  1.4302 11.0365
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)           111.363522  19.655078   5.666 2.08e-08 ***
## relative_compactness  -70.787707  11.138038  -6.355 3.59e-10 ***
## surface_area           -0.088245   0.018484  -4.774 2.17e-06 ***
## wall_area               0.044682   0.007196   6.209 8.80e-10 ***
## overall_height7        14.993452   1.280527  11.709  < 2e-16 ***
## orientation3           -0.291979   0.324182  -0.901 0.368054
## orientation4           -0.124219   0.324182  -0.383 0.701697
## orientation5            0.349115   0.324182   1.077 0.281865
## glazing_area           13.252917   0.966523  13.712  < 2e-16 ***
## glazing_distribution1   2.160035   0.581924   3.712 0.000221 ***
## glazing_distribution2   1.977396   0.581924   3.398 0.000714 ***
## glazing_distribution3   1.639965   0.581924   2.818 0.004956 **
## glazing_distribution4   1.995660   0.581924   3.429 0.000638 ***
## glazing_distribution5   1.695521   0.581924   2.914 0.003678 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.176 on 754 degrees of freedom
## Multiple R-squared:  0.8904, Adjusted R-squared:  0.8885
## F-statistic: 471.3 on 13 and 754 DF,  p-value: < 2.2e-16
```

After removing roof_area, the model became full rank and all parameters were estimable.

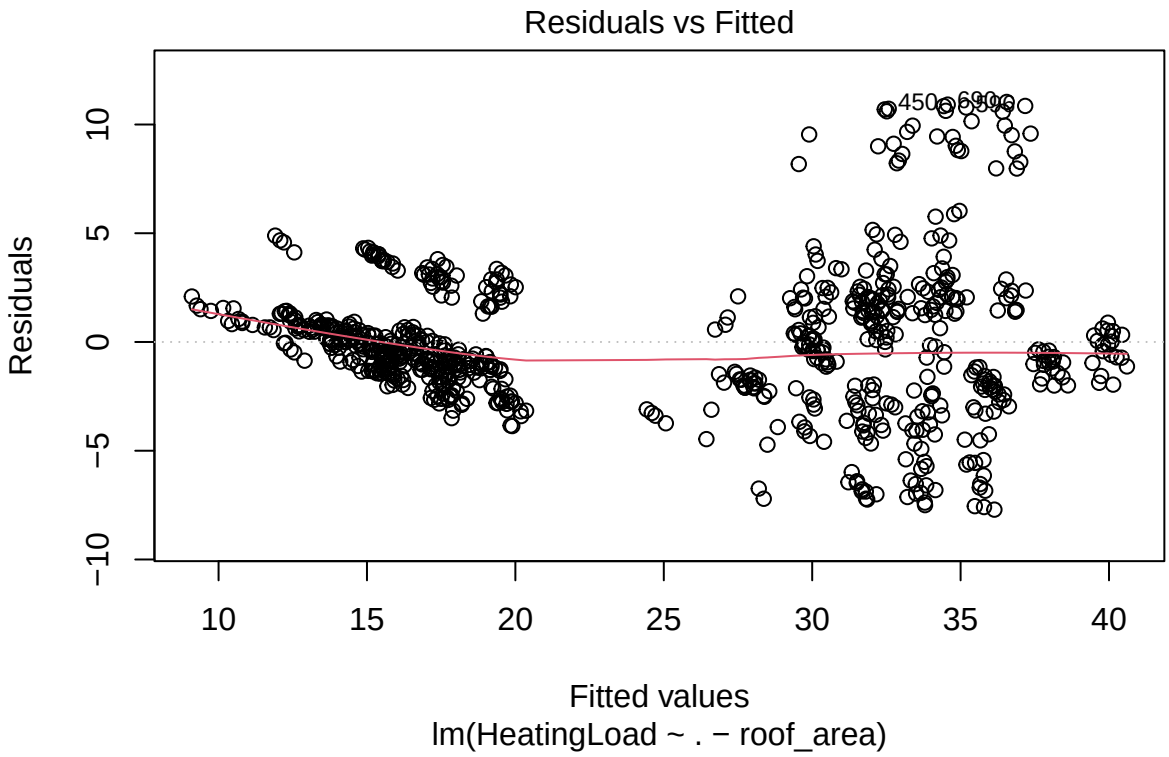
-Based on the p-value of the F-statistic, the model is statistically significant overall. That is, at least one predictor is useful in explaining variation in the heating load.

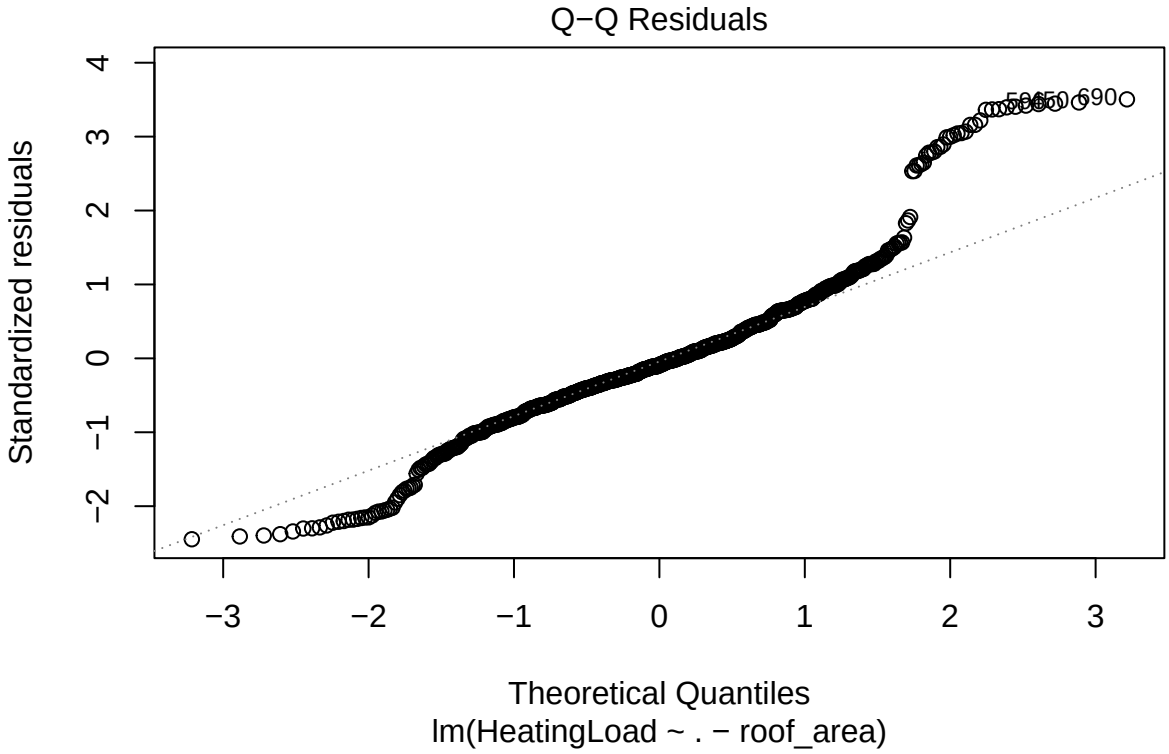
- Based on the p-value of the F-statistic, the overall model is highly significant — meaning at least one of the predictors helps explain the variation in heating load.
- The adjusted R^2 is approximately 88.9%, indicating that a large portion of the variability in heating load is explained by the model's predictors.
- Some orientation variables (levels 3, 4, and 5) appear statistically insignificant based on their individual p-values. However, we should not remove them just based on these p-values — it's better to test their collective contribution with model comparison or an F-test.

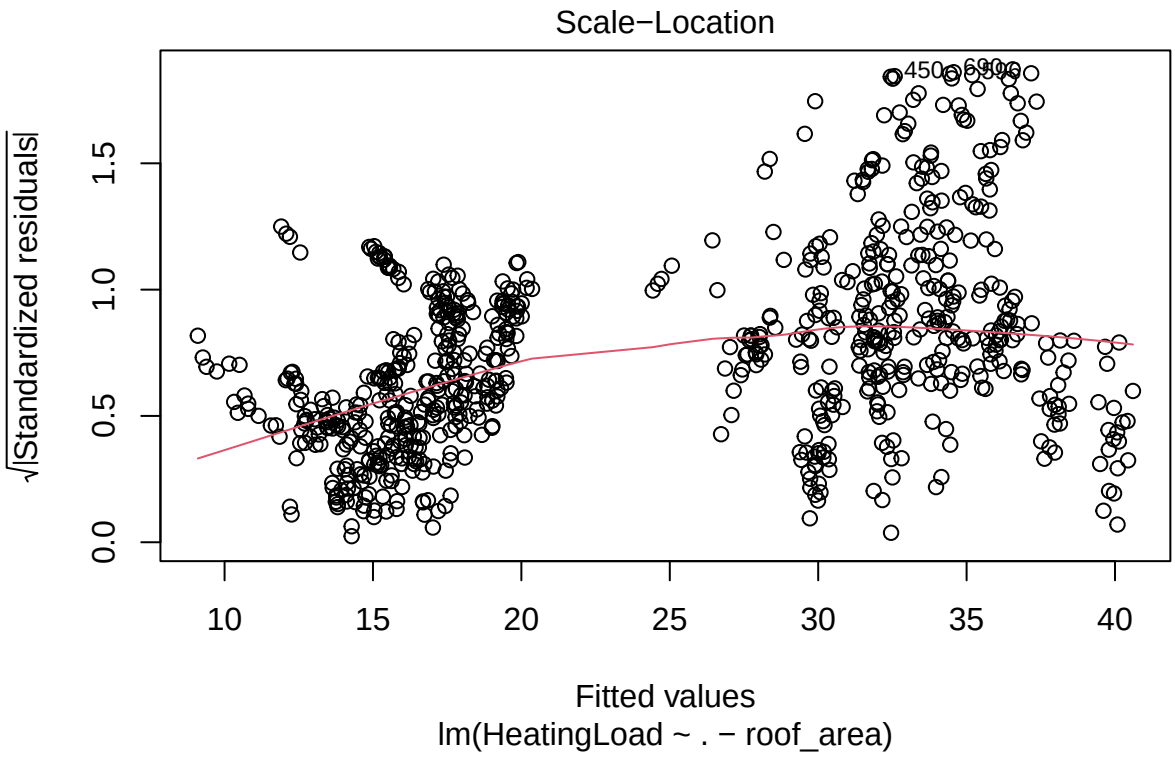
6. Perform a residual diagnostics and explain your findings.

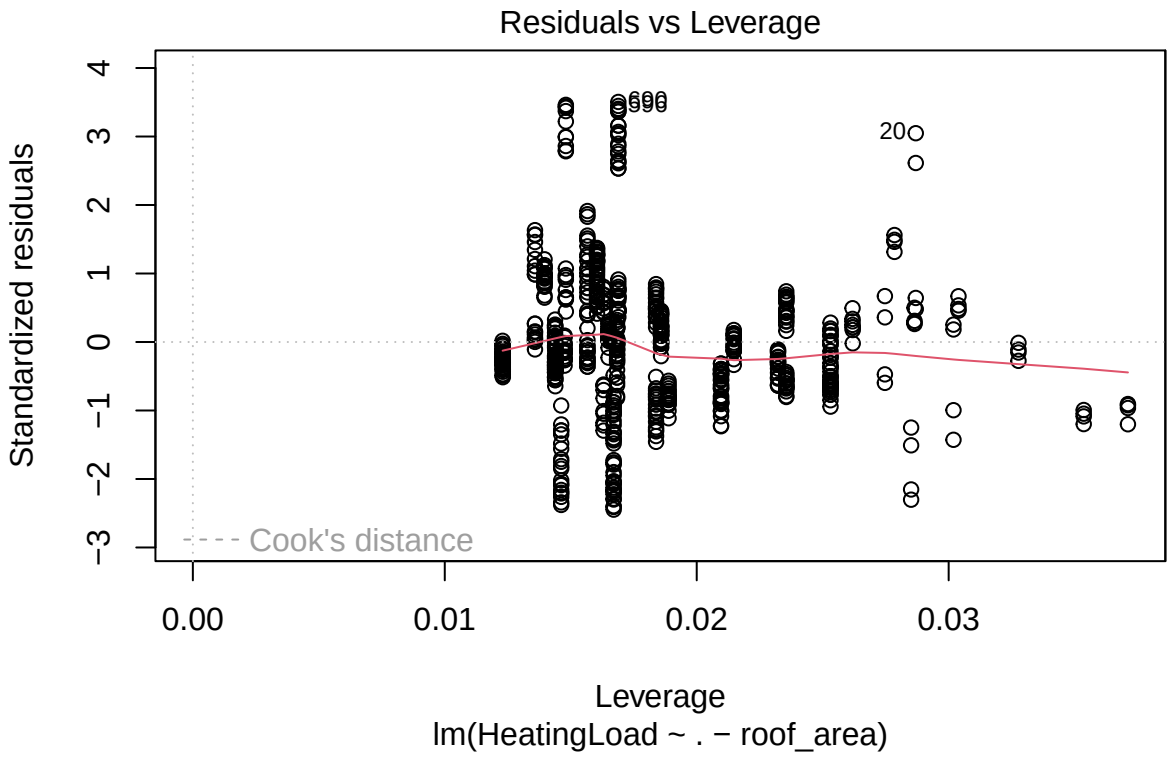
Solution:

```
plot(model_refit)
```


Residuals vs Fitted
Residuals
10
5
0
−5
−10
450
690
10
15
20
25
30
35
40
Fitted values
lm(HeatingLoad ~ . − roof_area)


Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
3
4
−3
−2
−1
0
1
2
3
690
Theoretical Quantiles
lm(HeatingLoad ~ . − roof_area)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
15
20
25
30
35
40
Fitted values
lm(HeatingLoad ~ . − roof_area)



Residuals vs Fitted Plot

- The residuals do not appear evenly scattered around zero — instead, they show a clear pattern with curved structure, indicating potential non-linearity.
- This suggests the linear model may not fully capture the relationship between the predictors and the response.

Q-Q Plot of Standardized Residuals

-The residuals deviate significantly from the diagonal line, especially at the tails — both the lower and upper ends curve away.

- This indicates that the residuals are not normally distributed, violating the normality assumption. It has right-skewness.

Scale-Location Plot

-The red trend line gradually increases as fitted values grow, which suggests slightly increasing variance of residuals at higher predicted heating load levels. It indicates mild heteroscedasticity.

Residuals vs Leverage Plot - Few points (like 20 and 690) has high leverage and a large residual, meaning it might have a strong influence on the model's predictions.

7. How will you resolve the issue with the residuals? Explain!

Solution:

Firstly, I would try more flexible model with interaction and polnomial terms first as in the following.

```
trial <-lm(HeatingLoad ~ (relative_compactness + surface_area + wall_area + glazing_area )^2 +
   overall_height + orientation+ glazing_distribution, data = energy_data)
summary(trial)
```

```
##
## Call:
## lm(formula = HeatingLoad ~ (relative_compactness + surface_area +
##     wall_area + glazing_area)^2 + overall_height + orientation +
##     glazing_distribution, data = energy_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0532 -1.5522 -0.1182  1.5508 10.0900
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -8.489e+02  2.265e+02  -3.748 0.000192 ***
## relative_compactness               2.787e+02  1.382e+02   2.017 0.044069 *
## surface_area                       7.222e-01  1.686e-01   4.282 2.09e-05 ***
## wall_area                          1.325e+00  6.198e-01   2.138 0.032807 *
## glazing_area                      -8.843e+01  9.249e+01  -0.956 0.339292
## overall_height7                    2.893e+01  1.899e+00  15.234  < 2e-16 ***
## orientation3                      -2.920e-01  2.958e-01  -0.987 0.323875
## orientation4                      -1.242e-01  2.958e-01  -0.420 0.674619
## orientation5                       3.491e-01  2.958e-01   1.180 0.238236
## glazing_distribution1              2.160e+00  5.309e-01   4.068 5.24e-05 ***
## glazing_distribution2              1.977e+00  5.309e-01   3.724 0.000210 ***
## glazing_distribution3              1.640e+00  5.309e-01   3.089 0.002084 **
## glazing_distribution4              1.996e+00  5.309e-01   3.759 0.000184 ***
## glazing_distribution5              1.696e+00  5.309e-01   3.194 0.001464 **
## relative_compactness:surface_area  3.209e-01  7.426e-02   4.322 1.76e-05 ***
## relative_compactness:wall_area    -4.516e-01  4.199e-01  -1.075 0.282519
## relative_compactness:glazing_area  7.954e+01  5.869e+01   1.355 0.175696
## surface_area:wall_area            -1.482e-03  4.647e-04  -3.189 0.001485 **
## surface_area:glazing_area          4.033e-02  7.035e-02   0.573 0.566675
## wall_area:glazing_area             4.337e-02  1.843e-02   2.354 0.018855 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.898 on 748 degrees of freedom
## Multiple R-squared:  0.9095, Adjusted R-squared:  0.9072
## F-statistic: 395.7 on 19 and 748 DF,  p-value: < 2.2e-16
```

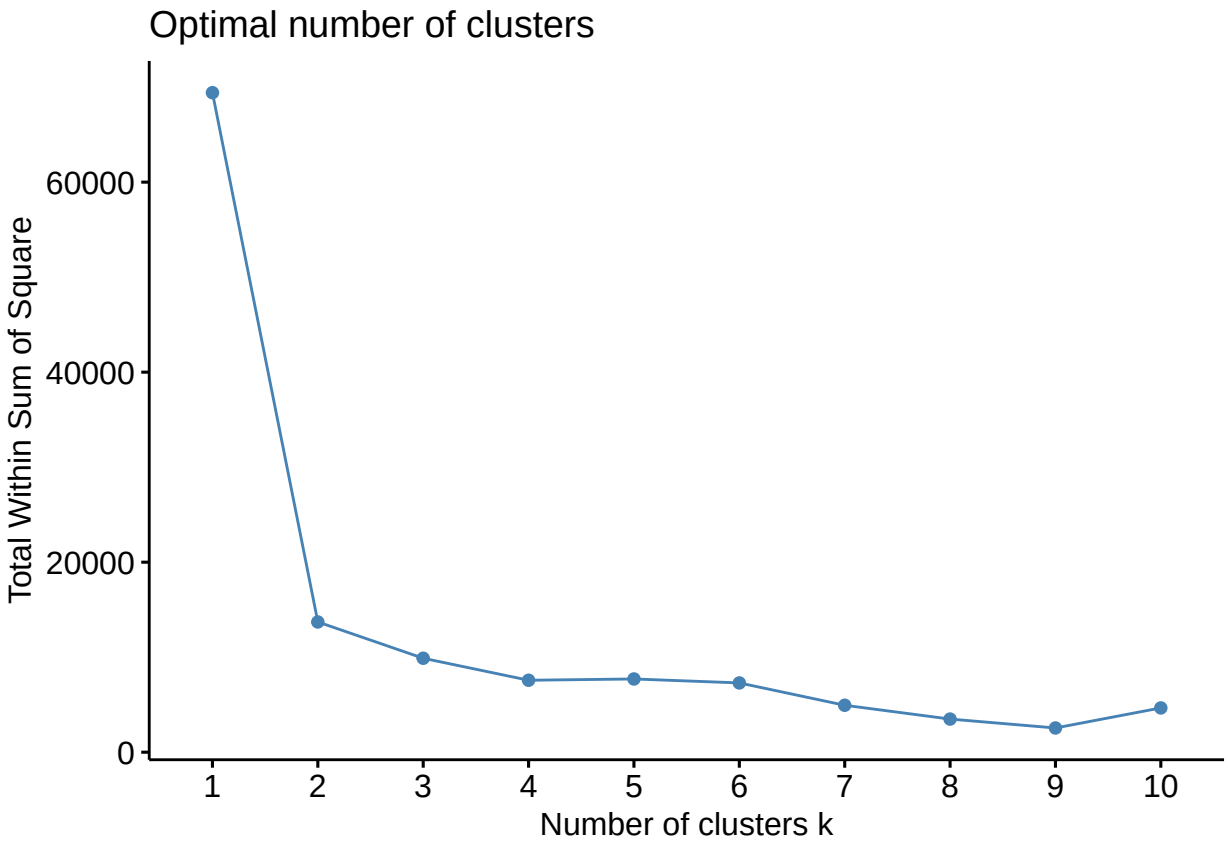
-After including second-order interaction terms between key continuous predictors, the adjusted R^2 increased to 90.7%, meaning the model explains more variation in heating load compared to the basic linear model.

-The inclusion of interactions like relative_compactness:glazing_area or surface_area:wall_area helps the model better capture complex relationships, improving both fit and predictive power.

Secondly, as we did in the question 1, I would apply to clustering to the residuals and do the same exact thing we did in question 1.

```r
library(factoextra)
resid_trial <- residuals(model_refit)
fitted_trial <- fitted(model_refit)

# Create matrix for clustering
resid_fit_matrix <- cbind(resid_trial, fitted_trial)
```

```r
fviz_nbclust(resid_fit_matrix, kmeans, method="wss")
```




```r
set.seed(42)
kmeans_result <- kmeans(resid_fit_matrix, centers = 3)

# Add cluster labels to your dataset
energy_data$cluster <- as.factor(kmeans_result$cluster)
table(energy_data$cluster)
```

```
##
##   1   2   3
## 187 201 380
```

```r
trial_clustered <- lm(HeatingLoad ~ relative_compactness + surface_area + wall_area + glazing_area +
                        overall_height + orientation + glazing_distribution + cluster,
                      data = energy_data)

summary(trial_clustered)
```

```
##
## Call:
## lm(formula = HeatingLoad ~ relative_compactness + surface_area +
##     wall_area + glazing_area + overall_height + orientation +
##     glazing_distribution + cluster, data = energy_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -8.4451 -1.5614 -0.2046  1.2627 11.3557
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             71.076412  19.823430   3.585 0.000358 ***
## relative_compactness   -49.386041  11.174097  -4.420 1.13e-05 ***
## surface_area            -0.055178   0.018476  -2.986 0.002914 **
## wall_area                0.042382   0.006986   6.067 2.07e-09 ***
## glazing_area            17.014910   1.083246  15.707  < 2e-16 ***
## overall_height7         14.615226   2.174233   6.722 3.54e-11 ***
## orientation3            -0.433054   0.314424  -1.377 0.168832
## orientation4            -0.194756   0.313923  -0.620 0.535186
## orientation5             0.490189   0.314424   1.559 0.119415
## glazing_distribution1    1.729445   0.579848   2.983 0.002951 **
## glazing_distribution2    1.490376   0.580130   2.569 0.010390 *
## glazing_distribution3    1.058896   0.580856   1.823 0.068702 .
## glazing_distribution4    1.546260   0.579929   2.666 0.007834 **
## glazing_distribution5    1.076831   0.581236   1.853 0.064323 .
## cluster2                 2.708635   0.393290   6.887 1.20e-11 ***
## cluster3                 3.135602   1.647325   1.903 0.057363 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.074 on 752 degrees of freedom
## Multiple R-squared:  0.8976, Adjusted R-squared:  0.8956
## F-statistic: 439.5 on 15 and 752 DF,  p-value: < 2.2e-16
```

- Adding residual clusters as a predictor helped explain previously unaccounted variation, improving model accuracy and capturing subtle structure in the data. This suggests the original residuals contained meaningful group differences the base model missed.

8. Based on the model in (v), select the important variables needed to predict the heating load using a non-shrinkage technique.

Solution:

```r
# Stepwise selection
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
step_model <- stepAIC(model_refit, direction = "both", trace = FALSE)

# View selected model
summary(step_model)
```

```
##
## Call:
## lm(formula = HeatingLoad ~ relative_compactness + surface_area +
##     wall_area + overall_height + glazing_area + glazing_distribution,
##     data = energy_data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8241 -1.6634 -0.3236  1.3974 11.2196
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           111.346751  19.669830   5.661 2.14e-08 ***
## relative_compactness  -70.787707  11.146967  -6.350 3.70e-10 ***
## surface_area           -0.088245   0.018499  -4.770 2.21e-06 ***
## wall_area               0.044682   0.007202   6.204 9.05e-10 ***
## overall_height7        14.993452   1.281553  11.699  < 2e-16 ***
## glazing_area           13.252917   0.967297  13.701  < 2e-16 ***
## glazing_distribution1   2.160035   0.582390   3.709 0.000223 ***
## glazing_distribution2   1.977396   0.582390   3.395 0.000721 ***
## glazing_distribution3   1.639965   0.582390   2.816 0.004990 **
## glazing_distribution4   1.995660   0.582390   3.427 0.000644 ***
## glazing_distribution5   1.695521   0.582390   2.911 0.003705 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.179 on 757 degrees of freedom
## Multiple R-squared:  0.8898, Adjusted R-squared:  0.8883
## F-statistic: 611.2 on 10 and 757 DF,  p-value: < 2.2e-16
```

I used stepwise regression based on AIC, which is a non-shrinkage method that selects variables by adding or removing them based on their contribution to model fit. Starting from the full model, the algorithm selected the most important predictors without penalizing coefficients. The final model includes:

- relative_compactness
- surface_area
- wall_area
- overall_height
- glazing_area
- All levels of glazing_distribution

These predictors showed statistically significant effects and helped achieve an adjusted R^2 of 88.8%, confirming their relevance in predicting heating load. Orientation was excluded, suggesting it did not contribute meaningfully to the model.

9. Based on the model in (v), select the important variables needed to predict the heating load using a shrinkage technique.

Solution:

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
# Prepare the data
X <- model.matrix(model_refit)[, -1]  # Remove the first column (intercept)
y <- energy_data$HeatingLoad

# Fit LASSO model using cross-validation
cv_lasso <- cv.glmnet(X, y, alpha = 1)
```

```r
# Get best lambda
best_lambda <- cv_lasso$lambda.min
print(best_lambda)
```

```
## [1] 0.001967382
```

```r
# Fit final LASSO model
lasso_model <- glmnet(X, y, alpha = 1, lambda = best_lambda)

# Show coefficients
coef(lasso_model)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## (Intercept)             98.27509265
## relative_compactness   -63.35745415
## surface_area            -0.07609106
## wall_area                0.04154291
## overall_height7         15.60238304
## orientation3            -0.28516395
## orientation4            -0.11740354
## orientation5             0.34911458
## glazing_area            13.30254810
## glazing_distribution1    2.08209994
## glazing_distribution2    1.89946105
## glazing_distribution3    1.56203050
## glazing_distribution4    1.91772494
## glazing_distribution5    1.61758605
```

```r
# Predict using the LASSO model
lasso_preds <- predict(lasso_model, s = best_lambda, newx = X)

# Compute SSE and SST
sse <- sum((y - lasso_preds)^2)
sst <- sum((y - mean(y))^2)

# Compute R-squared
r_squared <- 1 - (sse / sst)

# Count the number of non-zero coefficients (excluding the intercept)
p <- sum(coef(lasso_model)[-1] != 0)
n <- length(y)

# Compute adjusted R-squared
adj_r_squared <- 1 - (1 - r_squared) * (n - 1) / (n - p - 1)

# Print adjusted R-squared
cat("Adjusted R-squared for LASSO:", round(adj_r_squared, 4), "\n")
```

```
## Adjusted R-squared for LASSO: 0.8885
```

- The LASSO model selected all predictors, meaning none were shrunk to exactly zero — but their coefficients are slightly smaller than in the regular linear model due to penalization.
- key variables like relative_compactness, glazing_area, and overall_height7 still have strong influence, with large absolute coefficients, showing they are the most important predictors.

- no predictors are eliminated in the shrinkage method.

10. Select the best model based on your results in (viii) and (ix) and justify your answer.

Solution:

Based on the results of (viii) and (ix), I would choose LASSO Model due to this reasons: **Stepwise Regression** -Selects predictors based on AIC without applying any penalty, resulting in a simpler model by excluding non-contributing variables like orientation.

-Achieved a strong adjusted R^2 of 0.8883, indicating good explanatory power with fewer predictors.

LASSO Regression -Retains all predictors but shrinks less important coefficients, helping reduce overfitting while keeping model complexity under control.

-Slightly higher adjusted R^2 of 0.8885, and better generalization through regularization.

I would choose LASSO because: Although both models performed similarly, LASSO is preferred because it offers slightly better accuracy and includes built-in regularization, making it more robust for prediction on new data. As a prrof of it, I also did comparision on train-test data for both models:

```
energy_data <- read.csv("energy.csv")
energy_data$orientation <- as.factor(energy_data$orientation)
energy_data$glazing_distribution <- as.factor(energy_data$glazing_distribution)
energy_data$overall_height <- as.factor(energy_data$overall_height)
set.seed(42)
sample_idx <- sample(1:nrow(energy_data), size = 0.8 * nrow(energy_data))
train_data <- energy_data[sample_idx, ]
test_data <- energy_data[-sample_idx, ]
step_model <- stepAIC(lm(HeatingLoad ~ relative_compactness + surface_area + wall_area + glazing_area +
                           overall_height + orientation + glazing_distribution,
                         data = train_data),
                      direction = "both", trace = FALSE)

step_preds <- predict(step_model, newdata = test_data)
library(glmnet)

# Model matrix for training
X_train <- model.matrix(HeatingLoad ~ relative_compactness + surface_area + wall_area + glazing_area +
                          overall_height + orientation + glazing_distribution, data = train_data)[, -1]
y_train <- train_data$HeatingLoad

# Model matrix for test
X_test <- model.matrix(HeatingLoad ~ relative_compactness + surface_area + wall_area + glazing_area +
                         overall_height + orientation + glazing_distribution, data = test_data)[, -1]
y_test <- test_data$HeatingLoad

# Cross-validation for LASSO
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1)
best_lambda <- cv_lasso$lambda.min

# Final LASSO model and prediction
lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = best_lambda)
lasso_preds <- predict(lasso_model, newx = X_test)
library(Metrics)

print("LASSO")
```

```
## [1] "LASSO"
```

```
do.call(cbind, pracma::rmserr(y_test, lasso_preds))
```

```
##           mae      mse     rmse       mape      nmse      rstd
## [1,] 2.173616 9.546098 3.089676 0.08776032 0.1060729 0.1237883
```

```
print("Stepwise")
```

```
## [1] "Stepwise"
```

```
do.call(cbind, pracma::rmserr(y_test, step_preds))
```

```
##           mae      mse     rmse       mape      nmse      rstd
## [1,] 2.197992 9.671307 3.109873 0.08844074 0.1074642 0.1245975
```

- The LASSO model achieved slightly better performance across all metrics compared to stepwise regression — lower RMSE (3.09 vs. 3.11), MAE (2.17 vs. 2.20), MAPE (8.78% vs. 8.84%), and NMSE (0.1061 vs. 0.1075) — which supports its selection as the preferred model.

11. Partition your data into 60% training set and 40% test set. Report the average root mean squared error (RMSE) and mean absolute error (MAE) for 200 random partitions.

Solution:

```
library(glmnet)

# Prepare data
X <- model.matrix(HeatingLoad ~ ., data = energy_data)[, -1]  # remove intercept
y <- energy_data$HeatingLoad

set.seed(42)
n <- nrow(energy_data)
n_train <- floor(0.6 * n)

rmse_list <- c()
mae_list <- c()

for (i in 1:200) {
  train_idx <- sample(1:n, n_train)
  test_idx <- setdiff(1:n, train_idx)

  X_train <- X[train_idx, ]
  y_train <- y[train_idx]
  X_test <- X[test_idx, ]
  y_test <- y[test_idx]

  # Cross-validated LASSO on training set
  cv_fit <- cv.glmnet(X_train, y_train, alpha = 1)
  best_lambda <- cv_fit$lambda.min

  # Predict
  pred <- predict(cv_fit, s = best_lambda, newx = X_test)

  # metrics
  rmse_list[i] <- sqrt(mean((y_test - pred)^2))
  mae_list[i] <- mean(abs(y_test - pred))
}
```

```r
#  average metrics
mean_rmse <- mean(rmse_list)
mean_mae <- mean(mae_list)

cat("Average RMSE over 200 splits:", round(mean_rmse, 4), "\n")
```

```
## Average RMSE over 200 splits: 3.2091
```

```r
cat("Average MAE over 200 splits:", round(mean_mae, 4), "\n")
```

```
## Average MAE over 200 splits: 2.2869
```

After performing 200 random train-test splits (with 60% for training and 40% for testing), the LASSO regression model produced the following average performance metrics:

- Average Root Mean Squared Error (RMSE): 3.2091 This value indicates the typical size of prediction errors. A lower RMSE means the model's predictions are close to the actual heating load values. Compared to earlier values, this suggests consistent and reliable performance across different data partitions.
- Average Mean Absolute Error (MAE): 2.2869 MAE measures the average magnitude of errors in a more interpretable way. The average predicted heating load differs from the actual value by around 2.29 units. This is low, which supports the accuracy of the model.

These averaged metrics across multiple splits confirm that the LASSO model generalizes well to unseen data.