Regression Analysis in R

STAT 4101L

Surya Lamichhane

In [3]:
```
df = read.csv("~/Desktop/STAT4101L_all_files/train_insurance.csv", header =
knitr::kable(head(df))
```

age	sex	bmi	children	smoker	region	charges	health_cond
19	female	27.900	0	yes	southwest	16884.924	high
18	male	33.770	1	no	southeast	1725.552	low
28	male	33.000	3	no	southeast	4449.462	low
33	male	22.705	0	no	northwest	21984.471	high
32	male	28.880	0	no	northwest	3866.855	low
31	female	25.740	0	no	southeast	3756.622	low

In [4]:
```
str(df)
```

```
'data.frame':   1000 obs. of  8 variables:
 $ age        : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex        : chr  "female" "male" "male" "male" ...
 $ bmi        : num  27.9 33.8 33 22.7 28.9 ...
 $ children   : int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker     : chr  "yes" "no" "no" "no" ...
 $ region     : chr  "southwest" "southeast" "southeast" "northwest" ...
 $ charges    : num  16885 1726 4449 21984 3867 ...
 $ health_cond: chr  "high" "low" "low" "high" ...
```

Motivation

- John works for an Insurance company. The company wants to understand the impact of various factors on medical charges to determine the insurance premium.
- John is assigned the task of developing a relationship between medical charges and various underlying factors that can influence the medical cost.

Approach:

- John need to understand the association between medical charges and explanatory variables
- John must understand the concepts of regression.

Learning Objectives

By the end of this lesson, you will be able to:

- Define correlation
- Perform regression analysis
- Build a regression model
- Analyze the assumptions of regression

Mathematical Model

George Box (Statistician 1919-2013, UK): "All models are wrong, but some are useful." This statement emphasizes the inherent limitation of models in representing complex real-world phenomena perfectly.

- Models built upon assumptions and approximations that might not capture every factor influencing a situation.
- No model can encompass the entirety of the complexities inherent in the real world.

Why models might be considered "wrong":

- Simplifications: Models often simplify complex relationships or systems to make them understandable and solvable. This simplification might miss out on certain complexity that exist in reality.
- Assumptions: Models are based on assumptions about the data and the relationships between variables. If these assumptions are violated, the model might not hold true.
- Incompleteness: Models might not incorporate all relevant variables or factors that influence the outcome. They might overlook hidden variables or interactions that play a role.
    - Changing Dynamics: The world is dynamic, and situations evolve over time. A model built on historical data might become less accurate as new trends or influences emerge.
    - Human Error and Bias: Models are created by humans and can carry biases or errors in the selection of variables, data collection, or methodology.

However, despite these limitations and potential inaccuracies, models can still be incredibly useful. They provide frameworks for understanding, making predictions, and guiding decisions. Even if they don't represent reality perfectly, they can offer valuable insights and help us navigate and make sense of complex systems.

Acknowledging the imperfection of models encourages a cautious and critical approach, prompting continuous improvement, refinement, and exploration of better models that can more accurately represent reality within a given context.

Correlation

Correlation coefficient measures the degree of association between two variables.

- Population correlation coefficient (Pearson's correlation coefficient) is denoted by ρ (rho), and sample correlation coefficient denoted by r.
- Values of r lie between -1 and 1, $-1 \leq r \leq 1$.
- Correlation does not mean cause-and-effect relationship, but
- The existence of causation always implies correlation.
- In R, $r = cor(x, y)$.

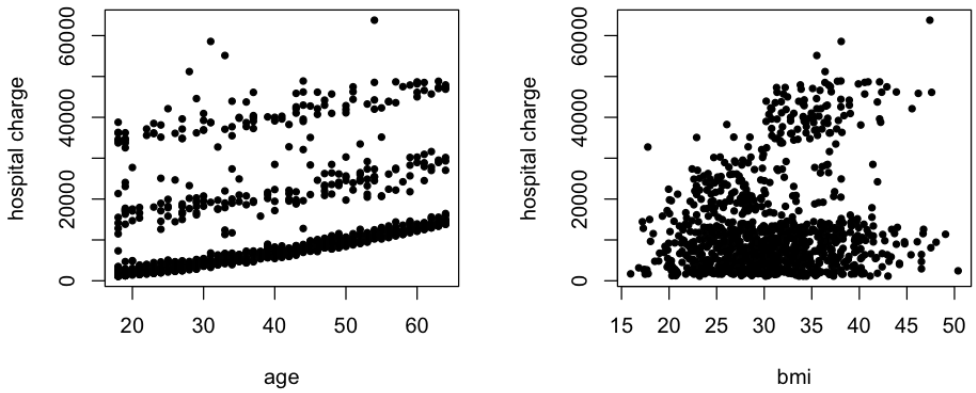
Sample correlation coefficient in R

In [5]:
```
cat("correlation coefficient between age and hospital charges")
cor(df$age, df$charges)
cat("correlation coefficient between bmi and hospital charges")
cor(df$bmi, df$charges)
```

```
correlation coefficient between age and hospital charges
```
0.330647056828519
```
correlation coefficient between bmi and hospital charges
```
0.188470399103015

Correlation diagnosis using scatter plot

In [6]:
```
options(repr.plot.width = 8, repr.plot.height = 4)
par(mfrow = c(1, 2))
plot(df$age, df$charges, xlab = 'age', ylab = 'hospital charge', type = 'p'
plot(df$bmi, df$charges,  xlab = 'bmi', ylab = 'hospital charge', type = 'p
```



Regression Analysis

It is a statistical technique of finding a functional relationship between variable of interest (target/dependent variable) to one or more independent or predictor variables.

The objective is to build a statistical model to:

- Describe the target and the variables that can explain the target variable called predictor/covariates/features.
- Find the association between target variable and predictor variable
- Predict the variable of interest
- Control the variable of interest

To explore additional concepts and more information, please see to the link1, or link2

1. Assumption of Linear Regression

- Linear relationship
- Independence of error
- Normality of error terms
- Equality of variance (Homoscedasticity)
- Predictors are linearly independent (No or little multi-collinearity)

Simple Linear Regression

Simple linear regression is a linear regression model with a single predictor variable.

- Regression line is denoted by $y = \beta_0 + \beta_1 X + \epsilon$, and
- the fitted line denoted by $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X$, where
- $\hat{\beta}_0$ and $\hat{\beta}_1$ are estimated values of parameters β_0 and β_1.

Multiple Linear Regression (MLR)

Multiple linear regression is a statistical technique used to model the relationship between multiple independent variables (predictors) and a single dependent variable (response). It extends simple linear regression, which models the relationship between one independent variable and a dependent variable, to cases where there are two or more independent variables.

The general form of multiple linear regression can be expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$$

Where:

- y is the dependent variable (response).
- $x_1, x_2, \ldots, x_p$ are the independent variables (predictors).
- β_0 is the intercept (the value of y when all predictors are zero).
- $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients (regression coefficients) representing the change in y associated with a one-unit change in each predictor, holding all other predictors constant.
- ϵ is the error term representing the discrepancy between the observed and predicted values.

In matrix notation, the equation can be written as:

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\,\boldsymbol{\beta} + \epsilon, \text{ so } \hat{\mathbf{y}} = \mathbf{X}\,\hat{\boldsymbol{\beta}}, && (1)\\ \implies E[\mathbf{y}] &= \mathbf{X}\,\boldsymbol{\beta} = E[\hat{\mathbf{y}}], && (2)\\ \implies \hat{E}[\hat{\mathbf{y}}] &= \mathbf{X}\,\hat{\boldsymbol{\beta}}. && (3) \end{aligned}$$

The final equation indicates that the estimated expected value of the predicted values $\hat{y}$ from the regression model is equal to $\mathbf{X}\,\hat{\boldsymbol{\beta}}$.

- The term ϵ is the error term, accounting for all other factors influencing y that cannot be captured by the model.
- Usual assumption, $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, having a mean of zero, being normally distributed, and having constant variance σ^2 (homoscedasticity)
- This assumption implies that $y_i \overset{\text{iid}}{\sim} \mathcal{N}(X_i^T \beta, \sigma^2)$, data values are independent and identically distributed (iid) and have normal distribution

Therefore, this assumption holds significant importance while building a regression model

Linear Regression in R

Syntax:

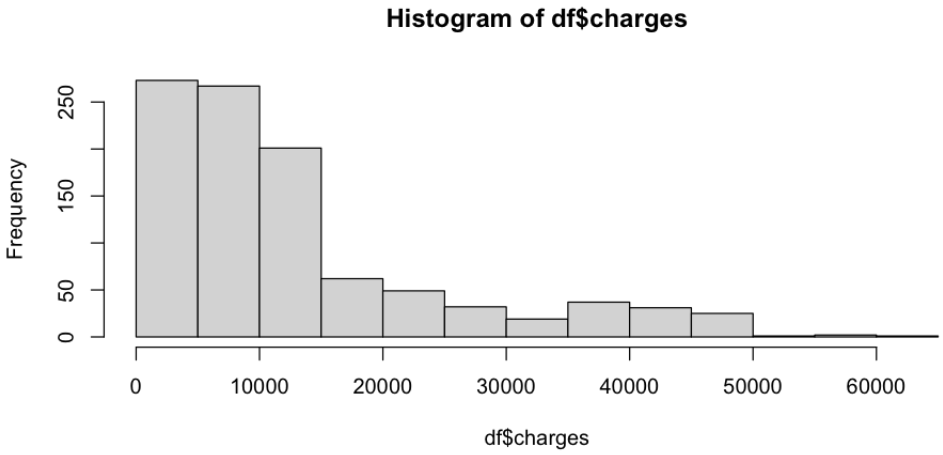
```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr
= TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

Example 1. Estimation of regression equation of hospital chargrs on age and regression output

---

First check the distribution of the target varaible (hospital chargrs) using histogram

In [7]:
```
hist(df$charges)
```

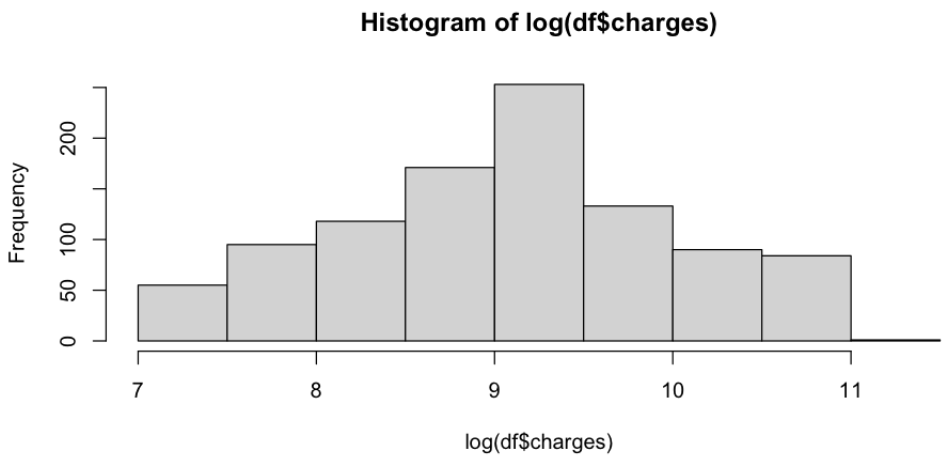


<div class="alert alert-danger" role="alert", style="margin-top: 0px">

- The above histogram shows the data values are far way from normally assumption (mild departures from normality is ok)
- If the data values significantly deviate from the normality assumption in a regression model, it can affect the reliability and validity of the model's results
- It's essential to consider alternative modeling approaches or transformations that could make the data more susceptible to the assumptions of linear regression.
- These might lead to use robust regression techniques such as data transformations (like log or square root transformations), or non-linear models that don't rely heavily on normality assumptions.

Check the distribution of the log of target varaible (hospital chargrs) using histogram

In [8]:
```
hist(log(df$charges))
```


Histogram of log(df$charges)
Frequency
0
50
100
200
7
8
9
10
11
log(df$charges)

```
log transformation shows much better result than original
data, so we use it as a response variable.
```

If applying a log transformation makes the histogram of the data much closer to a normal distribution, it can often improve the performance of linear models, as these models typically assume that the data follows a normal distribution.

However, whether it's "better" depends on the specific context of your analysis and the goals of your modeling. Here are a few considerations:

- Model Assumptions: Linear models assume certain relationships between the variables, including normality of residuals. If the log transformation makes the data more consistent with these assumptions, it might lead to more reliable model estimates.
- Robustness: Sometimes, transformations can make the model more sensitive to outliers. While a log transformation often reduces the impact of extreme values, it's not immune to this issue. So, you should assess the robustness of your model to outliers after applying the transformation.
- Context: Consider the specific context of your analysis and whether a log transformation makes sense theoretically. For example, if your data represents counts or monetary values, a log transformation might align better with the underlying process.
- Comparison: Before and after applying the transformation, compare the performance of your model using appropriate metrics (e.g., R-squared, RMSE) and assess whether the improvement justifies the transformation.

In [9]:
```
df$log_charges = log(df$charges)
fit1 = lm(log_charges~age, data = df)
summary(fit1)
```

```
Call:
lm(formula = log_charges ~ age, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2563 -0.4112 -0.2973  0.3446  2.2869

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.610892   0.071364  106.65   <2e-16 ***
age         0.037122   0.001697   21.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7589 on 998 degrees of freedom
Multiple R-squared:  0.3242,    Adjusted R-squared:  0.3235
F-statistic: 478.8 on 1 and 998 DF,  p-value: < 2.2e-16
```

In [12]:
```
#coefficients(fit1)
coef(fit1)
```

(Intercept): 7.61089156172694 **age:** 0.0371215505756222

Example 1. Estimation of regression equation of hospital chargrs on age. * The estimated regression equation of y on x is

$$\hat{y} = 7.7442 + 0.0345\ x$$

* that is, $\log(\widehat{charges}) = 7.7442 + 0.0345\ x$

---

2. Interpretation of coefficients for numerical covariates

- The line $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X$ is called the line of best fit or line of least squares
- $E[\hat{y}] = y$,
- $\hat{E}[\hat{\mathbf{y}}] = b_0 + \hat{\beta}_1 X$ at given x: The predicted value of y at given X is $\hat{y}$ is the estimated expected value at x.

β_0 (Intercept)

- represents the average estimated value for the response variable when predictor variables are equal to zero (a mathematical meaning than practical meaning) - it is the value of the dependent variable when all other factors have no effect. - in our above example $\hat{\beta} = 7.7442$ represents the estimated average log(hospital-charges) for customers whose age is 0 (Less practical meaning) - Sometimes, the intercept might not have a direct, tangible interpretation but serves as a reference point for the regression equation.

β (Slope corresponding to numerical covariate)

- β is also called a marginal rate of change in y - represents the change in the estimated expected value of the response variable for each unit increase in the corresponding independent variable, assuming all other variables remain constant. - regression coefficient for age is 0.0345. For each one-unit increase in x, the $\hat{E}[\hat{y}]$ increases by 0.0345 units. - This means that, the estimated expected value of log(hospital charge) increases by $0.0345 for each unit increase in the age variable, assuming all other variables remain constant. - change in log(hospital-charge) = 0.0345 implies estimated expected value of hospital charge increases by a factor of exp(0.0345) = 1.035

---

3. Choosing the Best Line (Ordinary Least Square Regression)

- predict $y = \beta_0 + \beta\, x$, so that $\sum(\hat{y}_i - y_i)^2$ is minimum, that is
- estimate β_0 and β for which $\sum \left((\beta_0 + \beta x_i) - y_i\right)^2$ is minimum.

Linear Regression Diagnostic

Diagnosing a linear regression model involves assessing its assumptions and examining various aspects to ensure the model's validity and reliability. Here are some common diagnostics used for linear regression models

1. Metrics to determine how well regression model fit the data (goodness of fit)
   a. R-squared
   b. Adjusted R-squared
   c. Likelihood
   d. AIC
   e. BIC

1. Residual Analysis: Residuals are the differences between observed and predicted values. Analyzing residuals helps check if they meet the assumptions of linear regression:
   a. Residuals vs. Fitted Values Plot: A scatter plot of residuals against predicted values to check for patterns or heteroscedasticity (unequal variance).
   b. Normality of Residuals: Checking if residuals follow a normal distribution using a histogram or a Q-Q plot.
   c. Statistical test for normality of residuals

1. Homoscedasticity: Verifying that the residuals have constant variance across all levels of the predictors. Heteroscedasticity might indicate a violation of the regression assumptions.

1. Linearity: Assessing if the relationship between predictors and the response variable is adequately captured by the linear model.
   a. This can involve plotting the predictor against the response variable before modeling.
   b. Test overall model significane,
   c. Coefficient Estimates: Assessing the coefficients of the predictors for significance and interpreting their impact on the dependent variable.
   d. Test for significance of predictor: Test whether the predictor variable is statistically significant or not in predicting the response varaible

1. Colinearity: Examining the correlation between predictor variables to ensure they are not highly correlated, as high colinearity can affect the model's stability and interpretation.
   a. Variance influence factor (VIF): a measure used in regression analysis to detect multicollinearity among predictor variables
   b. If VIF is greater than 5 (strong assumption), we consider that the corresponding varaible is linearly explained by other variables c. VIF > 10, loose assumption

1. Outlier and Influential Detection: Identifying influential data points or outliers that can significantly impact the regression results.
   a. Cook's Distance: Measures the influence of each observation on the model's coefficients.
   b. Outlier Analysis: Identifying observations with unusually large residuals

1. Leverage and Cook's Distance: Identifying observations with high leverage (extreme values in predictor variables) and their impact on the regression coefficients.

1. Model usefulness (Model predictability at new value of explanatory variables)

Eestimated data variance in R

In [13]:
```
#compute varaince for model 1
summary(fit1)$sigma^2
```

0.576005084976164

Multiple linear regression with numerical covariates

In [14]:
```
### Multiple linear regression
MLR_model = log_charges ~ age + bmi + children
fit2 = lm(log_charges ~ age + bmi + children, data = df)
```

In [15]:
```
#compute varaince for model 2
summary(fit2)$sigma^2
```

0.559150101667072

1. Check how good is regression.

Regression is assessed based on the residual deviations.

- Total deviation = Unexplained deviation + Explained deviation, i.e.
- $y_i - \bar{y} = (y_i - \hat{y_i}) + (\hat{y_i} - \bar{y})$.
- Once the linear relationship is determined, one can analyze the strength of the relationship

$$\underbrace{\Sigma(y_i - \bar{\mathrm{y}})^2}_{\mathrm{SST\ (Total\ sum\ of\ squared)}} = \underbrace{\Sigma(y_i - \hat{\mathrm{y}}_i)^2}_{\mathrm{SSE\ (Sum\ of\ squares\ of\ error)}} + \mathrm{SS}$$

Coefficient of determination (R-squared)

R-squared shows the strength of linear relationship between target variable with its predictors.

- It is the proportion of the variation in y that is explained by the regression.
- It is can be computed as

$$r^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

- R-squared = 0.279 means that 27.9% of the total variability of log(hospital-charges) in our original data is explained by the simple linear regression model (i.e., fit1) of log(hospital-charges) on age.
- R-squared always increases when the number of predictor variables included in the regression model increases, hence, we do not use R-squared to check how well multiple linear regression fits the data,
- Adjusted R-squared is most appropriate for multiple linear regression (MLR)

Adjusted R-squared

Adjusted-R^2 is the coefficient of determination corrected for degree of freedom.

- Adjusted R-squared doesn't always increase when new variables are introduced in the regression model.
- It will increase if increase in R^2 is higher than expected while adding variables in the model, otherwise decreases.
- It is defined as:

$$\text{adjusted } R^2 = 1 - \frac{\frac{SSE}{n-(k+1)}}{\frac{SST}{(n-1)}},$$

where n = Sample size, k = No. of parameters = p + 2, (if you have p many parameters).

In [16]:
```
summary(fit2)
```

```
Call:
lm(formula = log_charges ~ age + bmi + children, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2555 -0.4147 -0.3034  0.4091  2.2708 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.238357   0.134822  53.688  < 2e-16 ***
age         0.036384   0.001684  21.611  < 2e-16 ***
bmi         0.009514   0.003940   2.415   0.0159 *  
children    0.100104   0.019750   5.069 4.78e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7478 on 996 degrees of freedom
Multiple R-squared:  0.3453,    Adjusted R-squared:  0.3433 
F-statistic: 175.1 on 3 and 996 DF,  p-value: < 2.2e-16
```

The estimated model is

$$\widehat{\log(\text{Hospital charge})} = 7.2384 + 0.0364\ age + 0.0095\ bmi + 0.1001\ cl$$

Likelihood

The likelihood of a regression model refers to the probability of observing the given data points given the parameters and assumptions of the regression model.

- It measures how well the model, with its specified parameters, explains or predicts the observed data.
- A higher likelihood indicates that the model does a better job of explaining the observed data.

$$\text{Likelihood} = p(y|model)$$

In [17]:
```
log_lik_SLR = logLik(fit1) # of previous simple model
log_lik_MLR = logLik(fit2) # of Multiple linear regression mod

cat("Log-likelihood of previous simple model is: ", log_lik_SI
cat("Log-likelihood of MLR is: ", log_lik_MLR, "\n")
```

```
Log-likelihood of previous simple model is:  -1142.118
Log-likelihood of MLR is:  -1126.266
```

AIC and BIC

Likelihood always increases when adding more parameters in the model (so is covariates), but model complexity also increases. Adding predictor variables is not always beneficial due to the model complexity. Hence, in order to balance the goodness of fit and model complexity other statistical measure that are in prectice are

a. AIC (Akaike Information Criterion):

$$AIC = -2\hat{L} + 2K$$

Where k represents the number of parameters in the model, and $\hat{L}$ is the maximum value of the likelihood function of the model.

- k = p + 2, if there are p covaraites in the model
- parameters for 1 intercept, 1 variance, p coefficients for covariates

b. BIC (Bayesian Information Criterion):

$$BIC = -2\hat{L} + k\log(n)$$

- A model with good fit has lowest AIC and BIC.
- We choose model based on either AIC or BIC or both

Comparision of SLR Model1 with MLR model2

In [18]:
```
n = nrow(df)
results = data.frame(
    matric = c("R-squared", "Adj-R^2", "Log-lik", "AIC", "BI(
    Model1 = c(summary(fit1)$r.squared, summary(fit1)$adj.r.s
    Model2 = c(summary(fit2)$r.squared, summary(fit2)$adj.r.s
)

print(results)
```

```
     matric        Model1       Model2
1 R-squared     0.3242095     0.345299
2   Adj-R^2     0.3235323     0.343327
3   Log-lik -1142.1181367 -1126.265861
4       AIC  2290.2362735  2262.531721
5       BIC  2304.9595393  2287.070498
```

Observed vs fitted plot

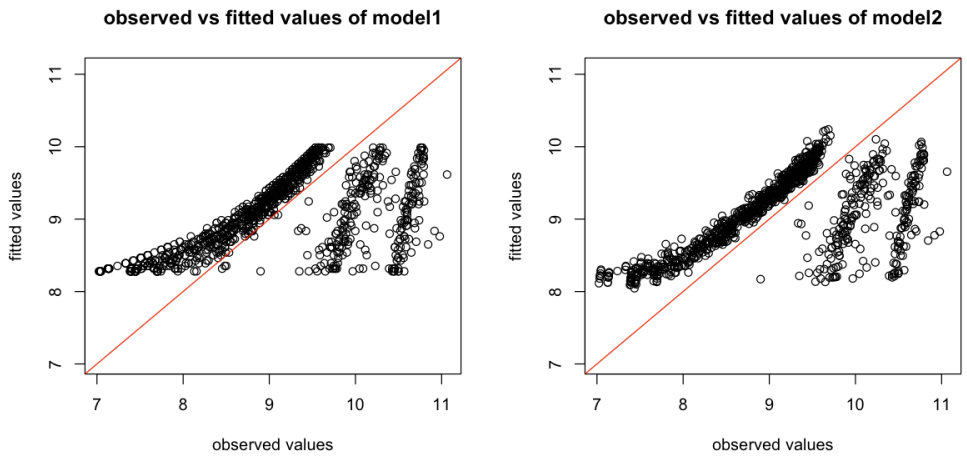
Visualization of observed vs fitted values:

Plotting observed versus fitted values in a regression analysis is a common diagnostic tool to assess the model's performance. This plot compares the actual observed values of the dependent variable against the predicted or fitted values obtained from the regression equation.

$$\text{Fitted values: } \hat{y} = \hat{\beta}_0 + \hat{\beta}X.$$

In [19]:
```
fit_values1 = fit1$fitted.values
fit_values2 = fitted(fit2)
```

In [20]:
```
options(repr.plot.width = 10, repr.plot.height = 5)
par(mfrow = c(1,2))
plot(df$log_charges, fit_values1, xlim = range(df$log_charg
     ylim = range(df$log_charges, fit_values1), xlab = "obs
     ylab = "fitted values", type = 'p', main = "observed v
# x = y line
abline(0,1, col = 'red')
plot(df$log_charges, fit_values2, xlim = range(df$log_charg
     ylim = range(df$log_charges, fit_values2), xlab = "obs
     ylab = "fitted values", type = 'p', main = "observed v
# x = y line
abline(0,1, col = 'red')
```



2. Residual Analysis (Model1)

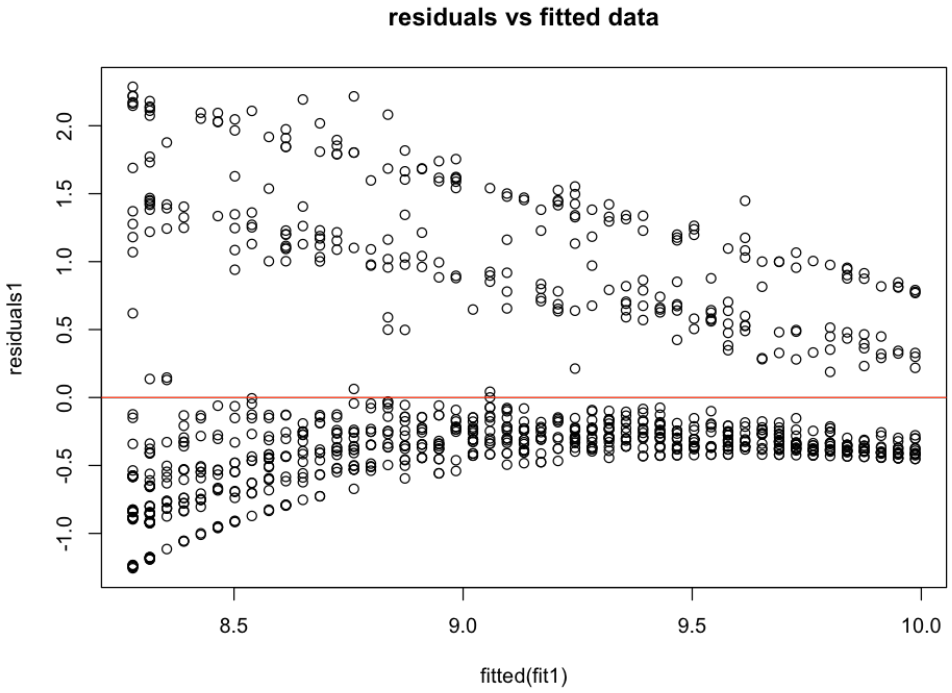
residuals = Observed - predicted

$$\text{residuals} = y - \hat{y}$$

1. Scatter plot of residuals against predicted values

- check for patterns or heteroscedasticity (unequal variance)

In [21]:
```
options(repr.plot.width = 8, repr.plot.height = 6)
residuals1 = fit1$residuals
#produce residual vs. fitted plot
plot(fitted(fit1), residuals1, main = "residuals vs fitted
#add a horizontal line at 0
abline(0,0, col = "red")
```



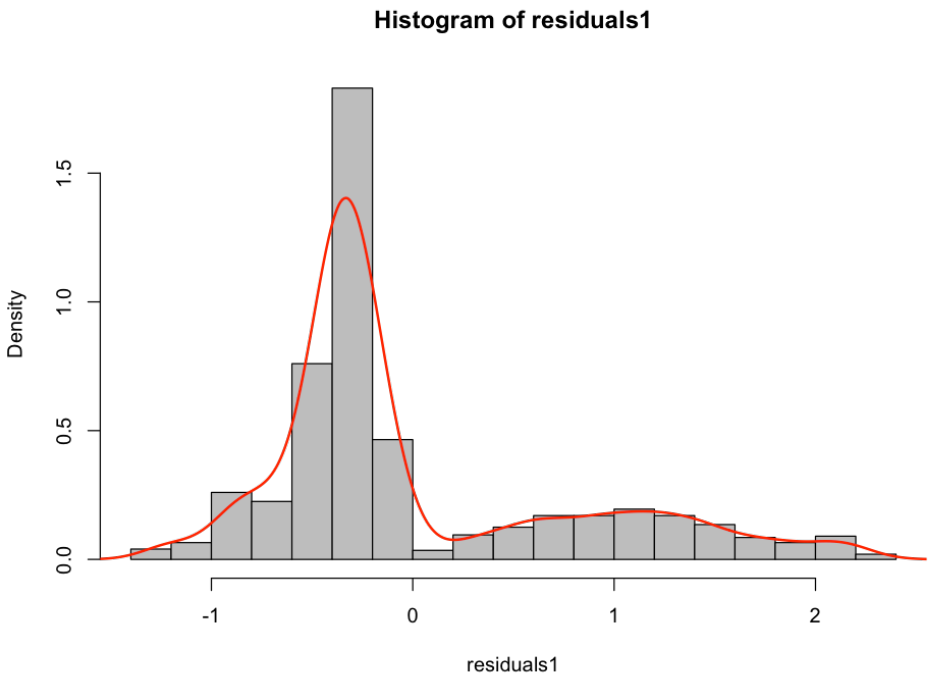
<div class="alert alert-danger" role="alert", style="margin-top: 0px">

- Residual plot fails to show equality of variance (Homoscedasticity assumption), not good!

2. Test for normality using visualization

Histogram of standardize residuals

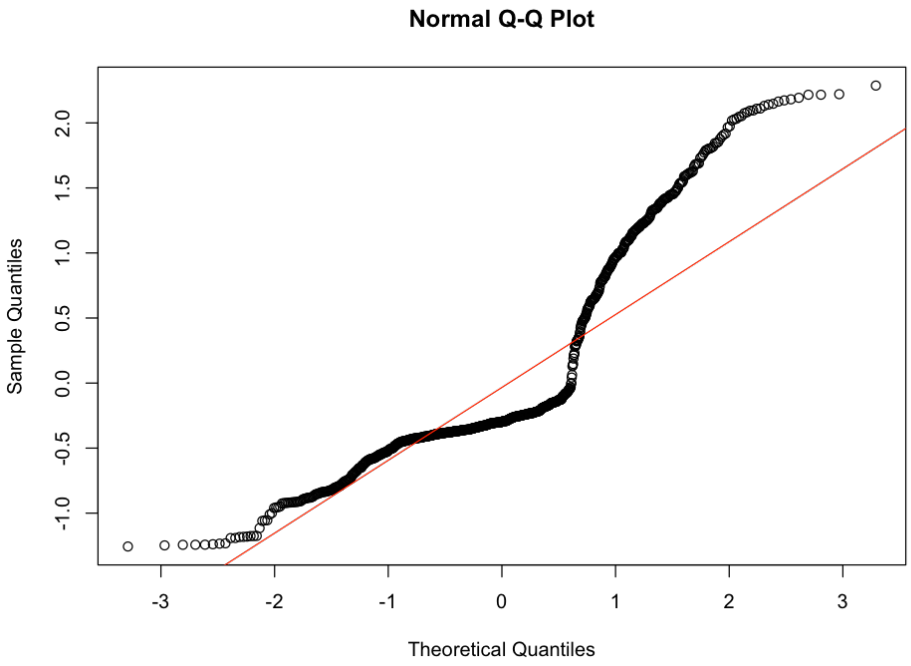
In [22]:
```
hist(residuals1, breaks = 15, prob = TRUE, col = 'grey' )
lines(density(residuals1), lwd = 2, col = 'red')
```



the condition that the error terms are normally distributed is not met.

Q-Q plot of residuals

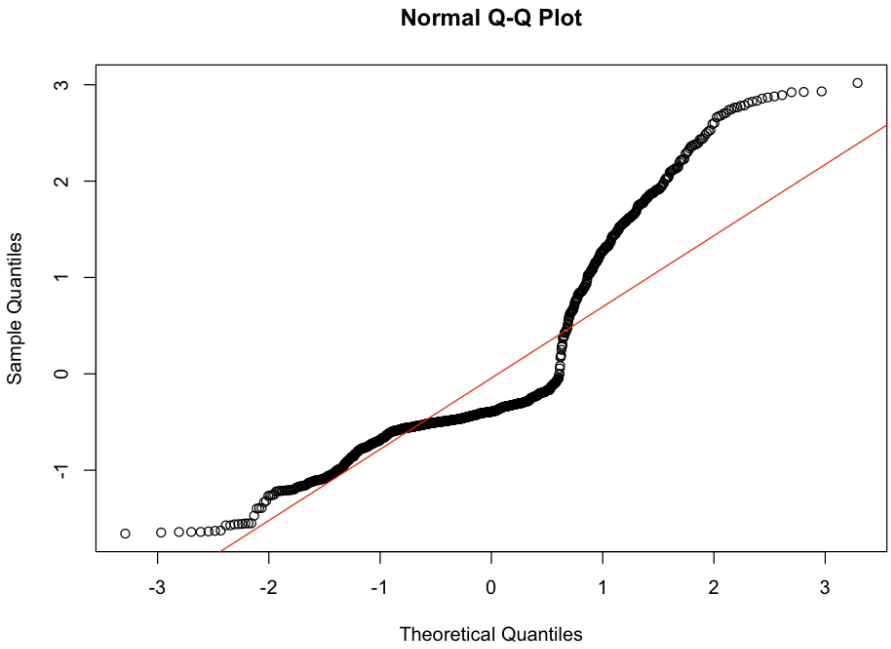
In [23]:
```
#create Q-Q plot for residuals
qqnorm(residuals1)
#add a straight diagonal line to the plot
qqline(residuals1, col = "red")
```



* The relationship between the sample percentiles and theoretical percentiles is not linear. * the condition that the error terms are normally distributed is not met.

Q-Q plot of standardize residuals

In [24]:
```
res_standard = rstandard(fit1) # standardized residuals
#create Q-Q plot for residuals
qqnorm(res_standard)
#add a straight diagonal line to the plot
qqline(res_standard, col = "red")
```



3. Test for normality (Shapiro-Wilk test)

Shapiro-Wilk test of normality assesses normality statistically. The null and alternative of the Shapiro-Wilk test are:

```
Null        H0: residuals are normally
distributed
Alternative H1: residuals are not normally
distributed
```

- a low p-value indicates that the residuals data do not follow the normal distribution.

In [25]:
```
shapiro.test(residuals1) # Shapiro-Wilk test in R
```

```
        Shapiro-Wilk normality test

data:  residuals1
W = 0.83778, p-value < 2.2e-16
```

In [26]:
```
shapiro.test(rstandard(fit1)) # Shapiro-Wilk test in R
```

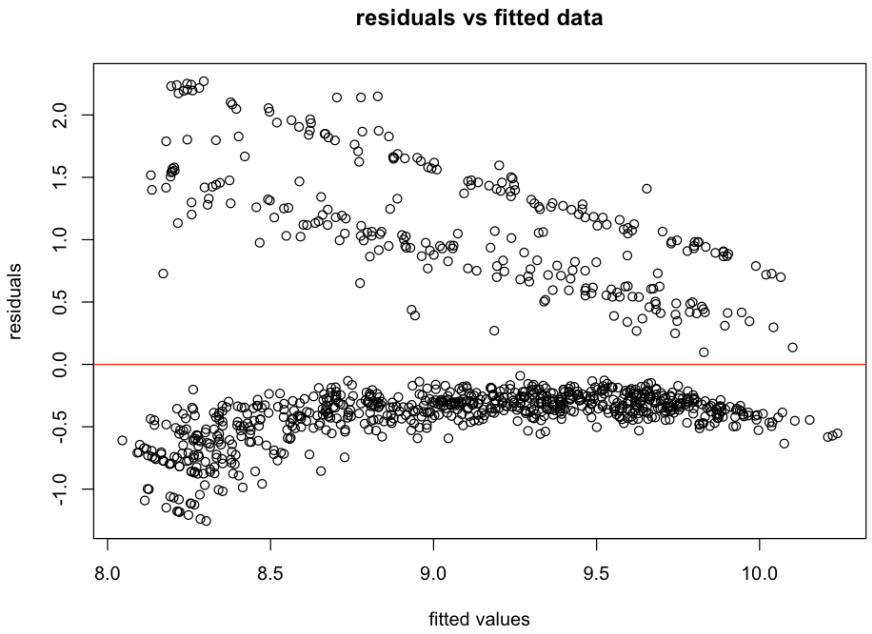
```
        Shapiro-Wilk normality test

data:  rstandard(fit1)
W = 0.83789, p-value < 2.2e-16
```

The above result shows the residuals are not normally distributed.

Residual Analysis (Model2)

1. Scatter plot of residuals against predicted values check for patterns or heteroscedasticity (unequal variance)

In [27]:
```
#produce residual vs. fitted plot
residuals2 = resid(fit2)
plot(fitted(fit2), residuals2, xlab = 'fitted values',
#add a horizontal line at 0
abline(0,0, col = "red")
```

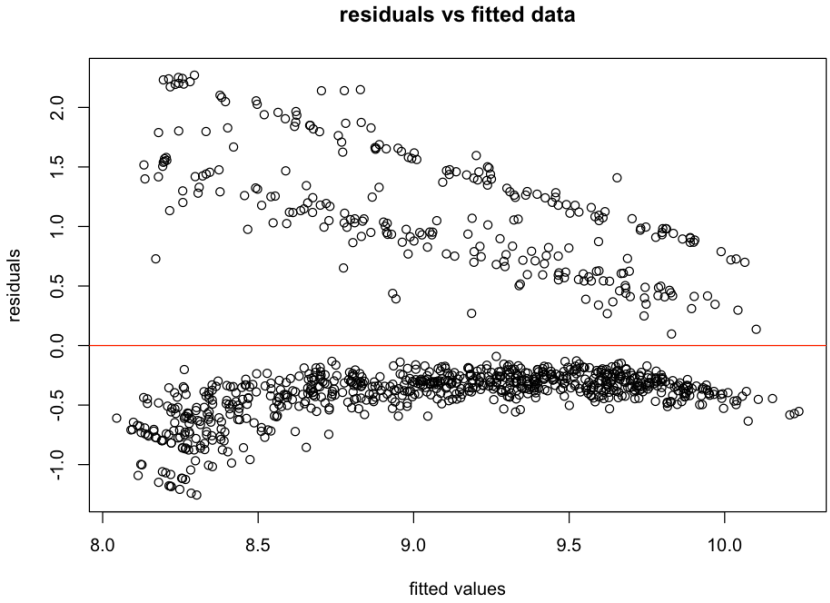


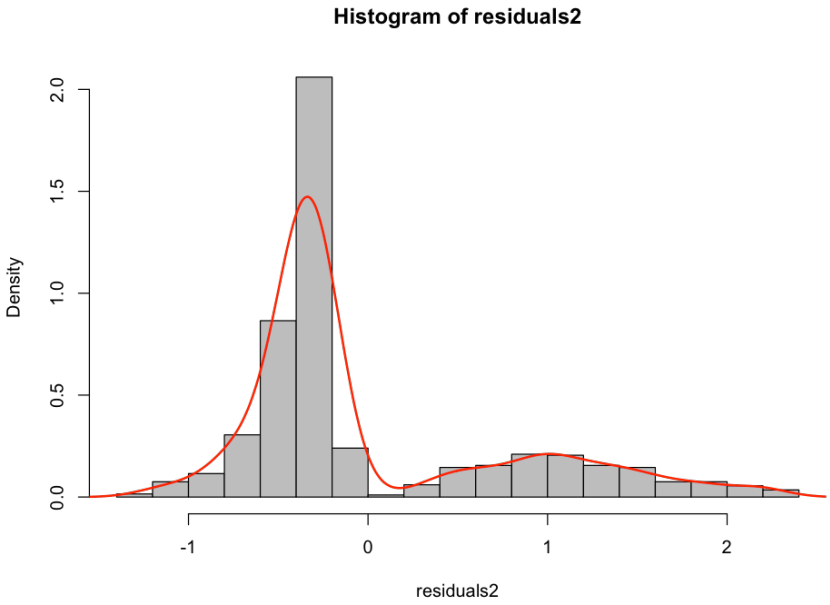
residuals are not showing randomness, but showing some weired pattern, not good!.

1. Test for normality using visualization

Histogram of standardize residuals

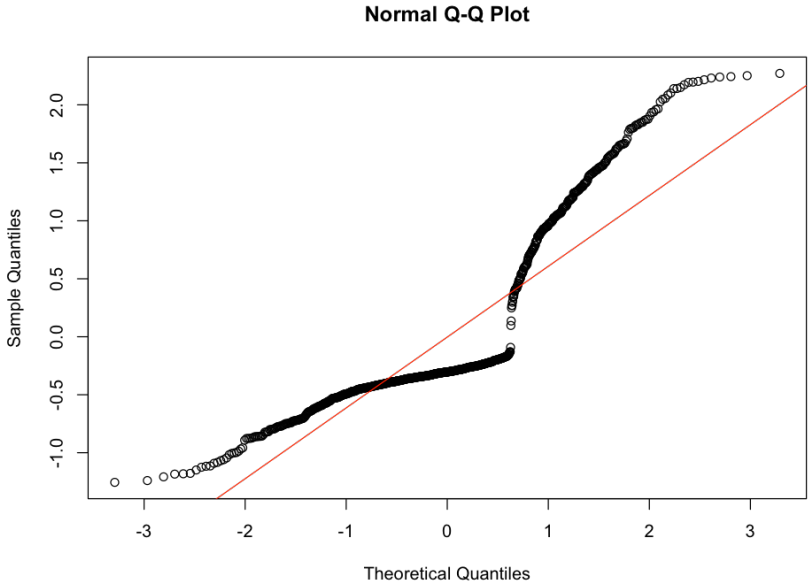
In [28]:
```
#produce residual vs. fitted plot
residuals2 = resid(fit2)
plot(fitted(fit2), residuals2, xlab = 'fitted values
#add a horizontal line at 0
abline(0,0, col = "red")
hist(residuals2, breaks = 15, prob = TRUE, col = 'gr
lines(density(residuals2), lwd = 2, col = 'red')
```





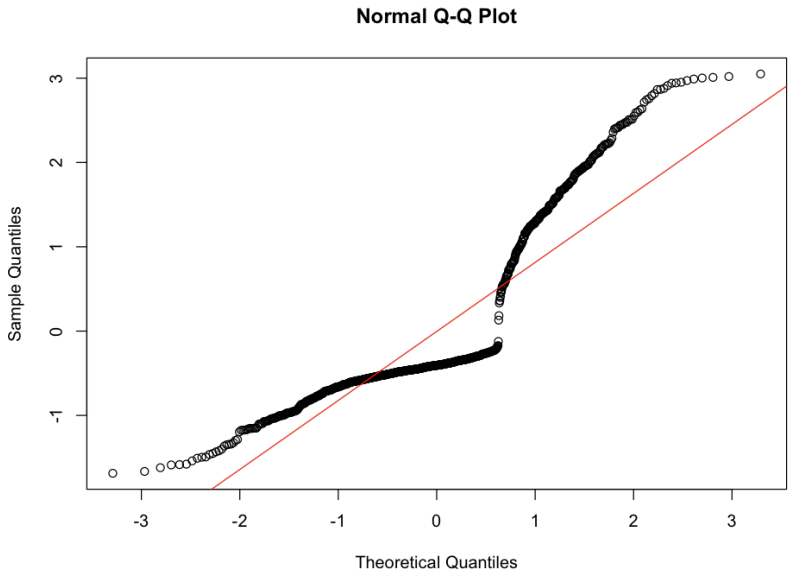
Q-Q plot of residuals

In [29]:
```
#create Q-Q plot for residuals
qqnorm(residuals2)
#add a straight diagonal line to the plot
qqline(residuals2, col = "red")
```

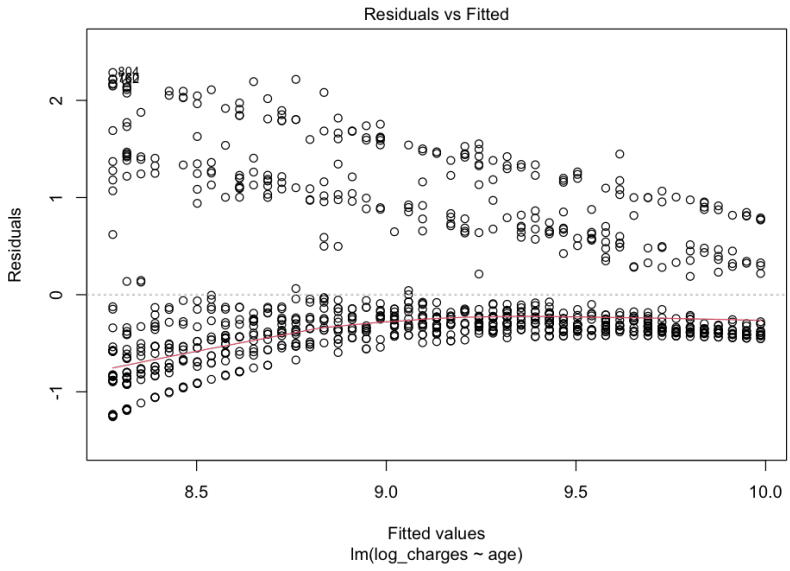


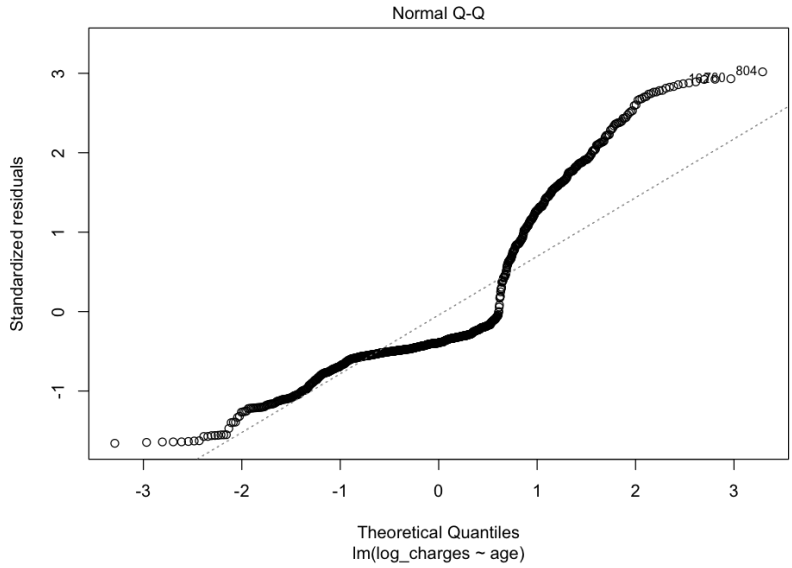
Q-Q plot of standardize residuals2

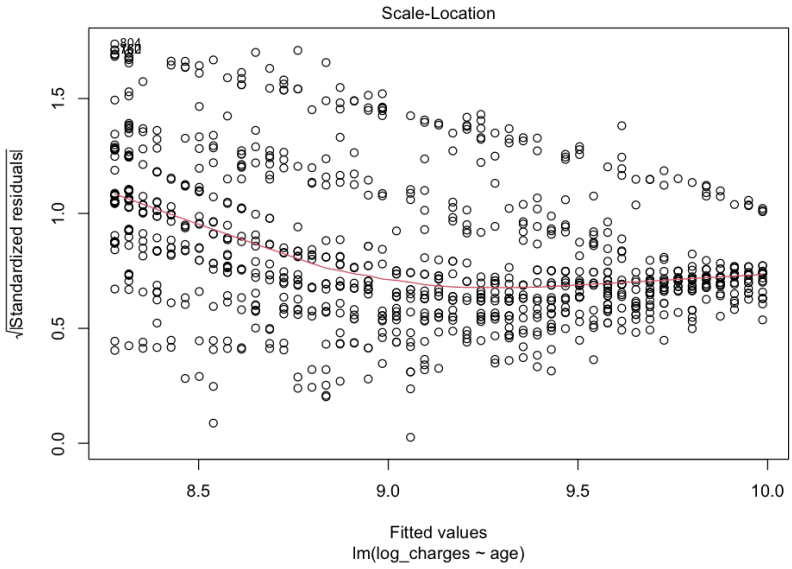
In [30]:
```
res_standard = rstandard(fit2) # standardized res
#create Q-Q plot for residuals
qqnorm(res_standard)
#add a straight diagonal line to the plot
qqline(res_standard, col = "red")
```

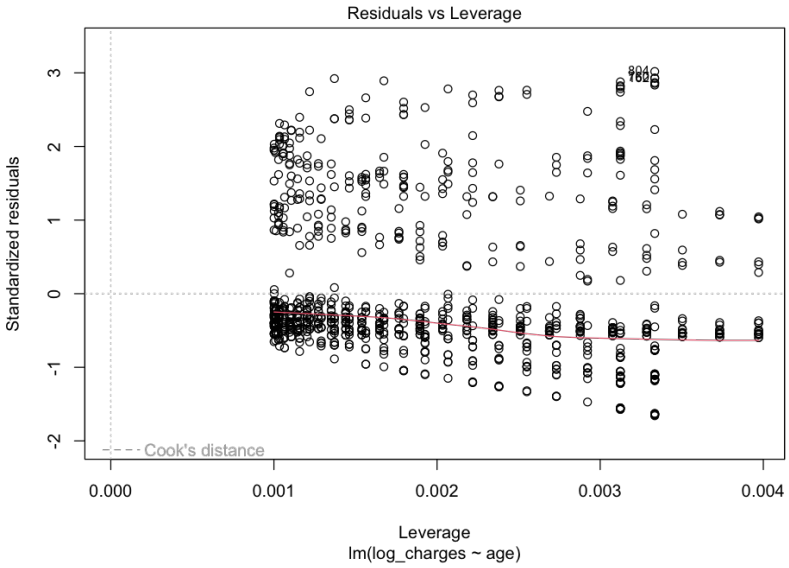


In [31]:
```
plot(fit1)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(log_charges ~ age)


Scale-Location
√|Standardized residuals|
Fitted values
lm(log_charges ~ age)



1. Test for normality (Shapiro-Wilk test)

Shapiro-Wilk test of normality assesses normality statistically. The null and alternative of the Shapiro-Wilk test are:

```
Null        H0: residuals are normally distributed
Alternative H1: residuals are not normally distributed
```

- a low p-value indicates that the residuals data do not follow the normal distribution.

In [32]:
```
shapiro.test(residuals2) # Shapiro-Wilk test in
```

```
        Shapiro-Wilk normality test

data:  residuals2
W = 0.80638, p-value < 2.2e-16
```

residuals are not showing randomness (independent) and normality test fails, not good!.

Evaluation Metrics for model's predictive performance

Evaluation metrics measure how good a model performs and how well it defines the relationships.

The predictability of the model can measured usinng evaluation metrics:

- MSE: Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

- MAE: Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

- RMSE: Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}.$$

A lower value of these metrics indicates a better model.

1. evaluation metrics for Simple Reg model (Model1)

In [33]:
```
n = nrow(df)
k = 1 + 2
fit_values = fit1$fitted.values
MSE = 1/n*sum((df$log_charges - fit_values)^2
MAE = 1/n*sum(abs(df$log_charges - fit_values
RMSE = sqrt(1/n * sum((df$log_charges - fit_v

evaluation_metrics = c( "R-sq" = summary(fit1
                        "Adj.R-sq" = summary(f
                        "log-Lik" = logLik(fit
                        "AIC" = AIC(fit1),
                        "BIC" = BIC(fit1),
                        "MSE" = MSE, "MAE" = M
model1_eval = data.frame(model1 = round(evalu
print(model1_eval)
```

```
            model1
R-sq         0.324
Adj.R-sq     0.324
log-Lik  -1142.118
AIC       2290.236
BIC       2304.960
MSE          0.575
MAE          0.601
RMSE         0.758
```

1. evaluation metrics for MLR (Model2)

In [34]:
```
n = nrow(df)
k = 3+2
fit_values = fit2$fitted.values
MSE = 1/n*sum((df$log_charges - fit_values)^2
MAE = 1/n*sum(abs(df$log_charges - fit_values
RMSE = sqrt(1/n * sum((df$log_charges - fit_v
evaluation_metrics = c( "R-sq" = summary(fit2
                        "Adj.R-sq" = summary(f
                        "log-Lik" = logLik(fit
                        "AIC" = AIC(fit2),
                        "BIC" = BIC(fit2),
                        "MSE" = MSE, "MAE" = M
model2_eval = data.frame(model2 = round(evalu
print(model2_eval)
```

```
            model2
R-sq         0.345
Adj.R-sq     0.343
log-Lik  -1126.266
AIC       2262.532
BIC       2287.070
MSE          0.557
MAE          0.600
RMSE         0.746
```

In [35]:
```
results = data.frame(model1_eval, model2_eval
print(results)
```

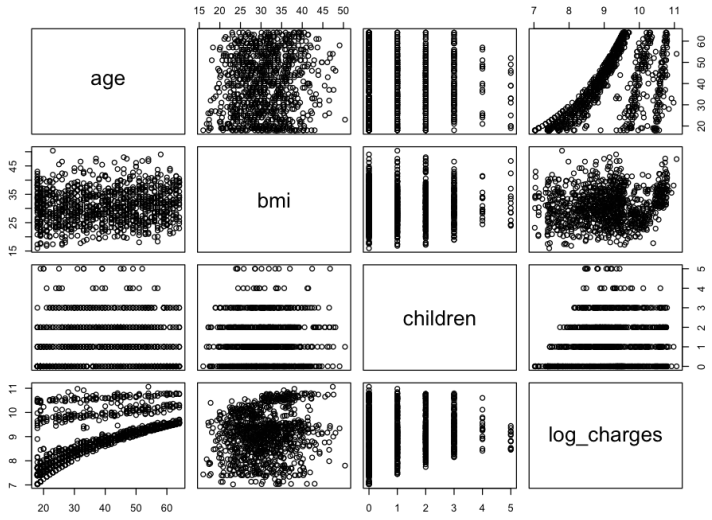
```
            model1    model2
R-sq         0.324     0.345
Adj.R-sq     0.324     0.343
log-Lik  -1142.118 -1126.266
AIC       2290.236  2262.532
BIC       2304.960  2287.070
MSE          0.575     0.557
MAE          0.601     0.600
RMSE         0.758     0.746
```

4. Linearity and model significance

Assessing if the relationship between predictors and the response variable is adequately captured by the linear model.

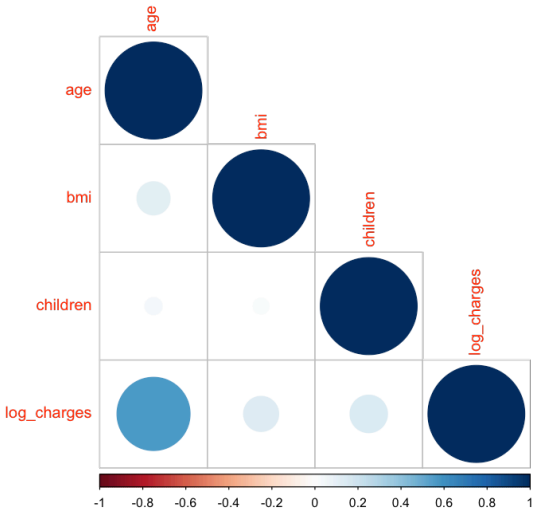
a. Plot the predictor against the response variable before modeling.

In [36]:
```
# select numeric varaibles and check the line
df_numeric_mat = subset(df, select= c("age",
plot(df_numeric_mat)
```



In [37]:
```
# check sample correlation
#install.packages("corrplot")
library(corrplot)
corrplot(cor(df_numeric_mat),method = "circle
```

```
corrplot 0.92 loaded
```



b. Test overall model significane

We can assess the significance of the overall model fit using methods like the F-test or examining the p-value associated with the overall model. The corresponding test for k many predictor variable and for the model

$$y = \beta_0 + \beta_1 x1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon$$

.

$$\text{Null } H_0 : \ \beta_1 = \beta_2 = \ldots = \beta_k = 0, (\text{i.e. a}$$
$$\text{Alternative } H_1 : \ \text{at least one } \beta_j \neq 0, j = 1, 2, . .$$

- The test statistic is F-score
- Low p-value indicates Linear regression model is significant.

In [38]:
```
fit2 = lm(log_charges ~ age + bmi + children
summary(fit2)
```

```
Call:
lm(formula = log_charges ~ age + bmi + chil
dren, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2555 -0.4147 -0.3034  0.4091  2.2708

Coefficients:
            Estimate Std. Error t value Pr(
>|t|)
(Intercept) 7.238357   0.134822  53.688  <
2e-16 ***
age         0.036384   0.001684  21.611  <
2e-16 ***
bmi         0.009514   0.003940   2.415
0.0159 *
children    0.100104   0.019750   5.069 4.7
8e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’
0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7478 on 996 degr
ees of freedom
Multiple R-squared:  0.3453,    Adjusted R-
squared:  0.3433
F-statistic: 175.1 on 3 and 996 DF,  p-valu
e: < 2.2e-16
```

```
Residual standard error: 0.7478 on
996 degrees of freedom
Multiple R-squared:  0.3453,
Adjusted R-squared:  0.3433
F-statistic: 175.1 on 3 and 996 DF,
p-value: < 2.2e-16
```

The value of F-score and p-value above indicate that Regression model is significant.

c. Coefficient Estimates: Assessing the coefficients of the predictors for significance and interpreting their impact on the dependent variable.

- magnitude of the coefficient of predictor varaible shows strength of its impact on the response
- larger absolute values indicate a stronger impact or influence; however,
- if variables are on different scales, it is challenging to compare their impacts directly.
- scaling is necessary if we want compare effects directly using the coefficient terms.
- most widely used scaling methods are standardization(or normalization) and min-max transforamtion.

Usual sclaing methods are

- Centered 0: $z = (x - \mu)$
- Normalization: $z = \frac{(x-\mu)}{sd}$
- min max method: $z = \frac{(x-max)}{(max-min)}$

Let's scale the numerical covariates

Here we use normalization method to scale the numerical method as follows:

In [39]:
```
num_covariates = subset(df, select = c("a
#cat_covariates = subset(df, select = c("
num_std_covariates = scale(num_covariates
new_df = df
new_df[,c("age", "bmi", "children")] = nu
head(new_df)
```

A data

	age	sex	bmi	children	s
	<dbl>	<chr>	<dbl>	<dbl>	
1	-1.4564882	female	-0.4900222	-0.90092736	
2	-1.5271401	male	0.4806364	-0.06673536	
3	-0.8206214	male	0.3533098	1.60164864	
4	-0.4673621	male	-1.3490633	-0.90092736	
5	-0.5380140	male	-0.3279702	-0.90092736	
6	-0.6086658	female	-0.8471981	-0.90092736	

In [40]:
```
fit3 = lm(log_charges ~  age + bmi + chil
summary(fit3)
```

```
Call:
lm(formula = log_charges ~ age + bmi + ch
ildren, data = new_df)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2555 -0.4147 -0.3034  0.4091  2.2708

Coefficients:
             Estimate Std. Error t value P
r(>|t|)
(Intercept)  9.08146    0.02365 384.053
< 2e-16 ***
age          0.51497    0.02383  21.611
< 2e-16 ***
bmi          0.05754    0.02383   2.415
0.0159 *
children     0.12000    0.02368   5.069
4.78e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01
‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7478 on 996 de
grees of freedom
Multiple R-squared:  0.3453,    Adjusted
R-squared:  0.3433
F-statistic: 175.1 on 3 and 996 DF,  p-va
lue: < 2.2e-16
```

In [41]:
```
fit_values = fit3$fitted.values
MSE = 1/n*sum((df$log_charges - fit_value
MAE = 1/n*sum(abs(df$log_charges - fit_va
RMSE = sqrt(1/n * sum((df$log_charges - f

evaluation_metrics = c( "R-sq" =  summary
                        "Adj.R-sq" =  sum
                       "log-Lik" = logLik
                       "AIC" = AIC(fit3),
                       "BIC" = BIC(fit3),
                       "MAE" = MAE, "RMSE
model3_eval = data.frame(model3 = round(e
print(model3_eval)
```

```
           model3
R-sq        0.345
Adj.R-sq    0.343
log-Lik  -1126.266
AIC       2262.532
BIC       2287.070
MSE          0.557
MAE          0.600
RMSE         0.746
```

In [ ]:
```

```

In [42]:
```
results = data.frame(model1_eval, model2_
print(results)
```

```
           model1    model2    model3
R-sq        0.324     0.345     0.345
Adj.R-sq    0.324     0.343     0.343
log-Lik  -1142.118 -1126.266 -1126.266
AIC       2290.236  2262.532  2262.532
BIC       2304.960  2287.070  2287.070
MSE          0.575     0.557     0.557
MAE          0.601     0.600     0.600
RMSE         0.758     0.746     0.746
```

Scaling of covariates (independent variables) is often needed in multiple linear regression (MLR) for several reasons:

- Magnitude Differences: When the variables in your dataset have different scales, the coefficients of the regression model may be biased towards variables with larger scales. Scaling helps to ensure that all

variables contribute equally to the regression model.

- Gradient Descent Optimization: If you're using optimization algorithms such as gradient descent to estimate the coefficients of the regression model, scaling can help speed up convergence. Variables with larger scales can dominate the optimization process, causing it to take longer to converge.
- Interpretability: Scaling does not affect the relationship between variables, but it does affect the interpretation of the coefficients. When variables are on different scales, the coefficients represent the change in the dependent variable for a one-unit change in the predictor. Scaling ensures that these coefficients are comparable.
- Regularization Techniques: Regularization techniques like Ridge regression and Lasso regression penalize the size of the coefficients. Scaling ensures that all variables are penalized equally, regardless of their scales.

However, there are some cases where scaling may not be necessary or may not have a significant impact:

- Categorical Variables: If your dataset contains categorical variables (e.g., gender, region), scaling may not be necessary as these variables do not have a scale.

- Regularization with Interaction Terms: If you're including interaction terms or polynomial terms in your regression model along with regularization, the need for scaling may vary depending on the specific context.
- Tree-based Models: Decision trees and ensemble methods like Random Forests and Gradient Boosting Machines are not affected by the scale of the features.

In summary, while scaling is not always necessary in MLR, it can often improve model performance, convergence, and interpretability, especially when dealing with variables with different scales.

d. Test for significance of predictor: Test whether the predictor variable is statistically significant or not in predicting the response varaible

- not all the predictor varaibles in MLR will be equally important in explaining y.
- there might be some varaibles which have no significant effect on y.
- these can be assessed from the scatter plot of response y vs predictor varaible X_j.
- the statistical method of testing for the significance of such varaible X_j can be defined as

$$\text{Null } H_0 : \ \beta_j = 0, \ X_j \text{ has no effect}$$
$$\text{Alternative } H_1 : \ \beta_j \neq 0, X_j \text{ has some effe}$$

- The test statistic is t-score
- Low p-value indicates variable X_j is significant.

Removing insignificant varaible from the model

- less significant variables in the models are variables that have high p-value (usual cutoff is 0.05)
- remove one varaible that has highest p-value, then run the model using rest of the value and check p-values
- if you still find high p-value, remove the varible one at a time until you come to the predefined cutoff.

Variable selection methods

Variable selection methods in regression analysis help in identifying the most relevant predictors to include in a model. Here are some common techniques:

- Forward Selection: Starts with an empty model and iteratively adds variables that improve model fit based on a chosen criterion (e.g., p-values, AIC, BIC) until a stopping rule is met.
- Backward Elimination: Begins with a model containing all predictors and iteratively removes the least significant variable based on a chosen criterion until a stopping rule is met.
- Stepwise Selection: Combines forward and backward selection, adding or removing variables at each step based on a chosen criterion until the best subset of variables is identified.
- Lasso Regression (L1 Regularization): Uses regularization to penalize coefficients, effectively shrinking some coefficients to zero, thus performing variable selection automatically by excluding less influential predictors.
- Ridge Regression (L2 Regularization): Similar to Lasso but doesn't set coefficients to exactly zero, reducing the impact of less important predictors rather than entirely

excluding them.

- Elastic Net: Combines Lasso and Ridge regularization techniques to benefit from both variable selection (like Lasso) and handling correlated predictors (like Ridge).

What other factors can affect the model's predictibility?

The predictability of a regression model refers to its ability to accurately forecast or estimate outcomes based on input data. Several factors affect a model's predictability:

- Quality of Data: Clean, relevant, and representative data usually leads to better predictions. If the data is noisy, incomplete, or biased, it can affect the model's performance.
- Model Complexity: The complexity of the model matters. A too simple model may not capture all the patterns in the data, while an overly complex model might overfit (perform well on training data but poorly on new data).
- Feature Selection: Choosing the right features (variables) for the model is crucial. Irrelevant or redundant features can negatively impact predictability.
- Training Size: Having a sufficient amount of data for training is essential. More data often leads to

better model performance, allowing it to generalize well to unseen data.

- Model Evaluation: Using appropriate evaluation metrics helps assess how well the model is performing. Metrics like Mean Squared Error (MSE), R-squared, or Root Mean Squared Error (RMSE) help measure prediction accuracy.
- Assumptions: Regression models rely on certain assumptions about the data. Violations of these assumptions can affect predictability. For instance, if the relationship between variables is non-linear but a linear model is used, predictability might be compromised.
- Regularization Techniques: Applying regularization techniques (like Lasso, Ridge regression) can help prevent overfitting, enhancing a model's generalizability.
- Cross-Validation: Utilizing techniques like k-fold cross-validation aids in assessing how the model performs on different subsets of the data, helping to gauge its reliability.

Remember, no model is perfect, and there's always a trade-off between bias and variance. Ensuring a balance between the two is key to improving the predictability of a regression model. Additionally, the context of the problem and the domain expertise play a significant role in determining

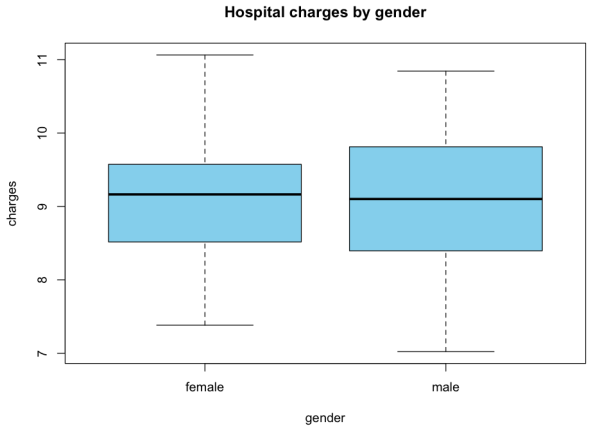
whether the model's predictions are practically useful.

Regression model with categorical Covariates

In [43]:
```
tb = table(df$sex, df$smoker)
tb
```

```
        no yes
female 414  81
male   390 115
```

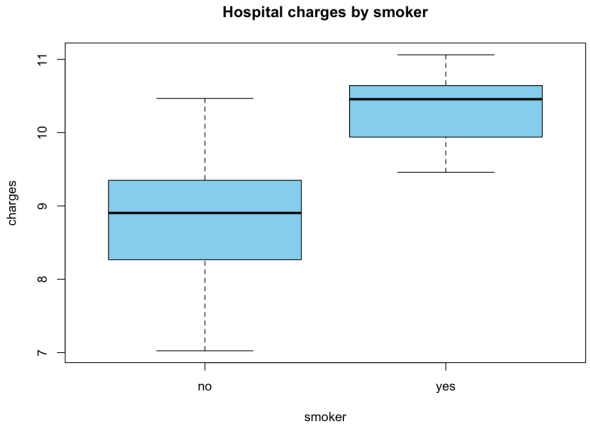
In [44]:
```
boxplot(df$log_charges~df$sex, col="
        main='Hospital charges by ge
        xlab="gender", ylab="charges
```



In [45]:
```
t.test(log_charges~sex, data = df)
```

```
        Welch Two Sample t-test

data:  log_charges by sex
t = 0.59852, df = 976.39, p-value =
0.5496
alternative hypothesis: true differe
nce in means between group female an
d group male is not equal to 0
95 percent confidence interval:
 -0.07949215  0.14926010
sample estimates:
mean in group female   mean in group
male
            9.099078             9.0
64194
```

In [46]:
```
boxplot(df$log_charges~df$smoker, co
        main='Hospital charges by sm
        xlab="smoker", ylab="charges
```



- t-test suggests that the effect of gender on the log-hospital charge is not significant, it implies that there is no statistically significant difference in log-hospital charges between different genders.
- In such cases, including the gender variable in the regression model may not be meaningful or necessary.

In [47]:
```
t.test(log_charges~smoker, data = d
```

```
        Welch Two Sample t-test

data:  log_charges by smoker
t = -39.573, df = 592.84, p-value
< 2.2e-16
alternative hypothesis: true diffe
rence in means between group no an
d group yes is not equal to 0
95 percent confidence interval:
 -1.598141 -1.447011
sample estimates:
 mean in group no mean in group ye
s
         8.783037         10.30561
3
```

- t-test suggests that the effect of smoker on the log-hospital charge is significant,
- In such cases, including the smoker variable in the regression model is meaningful.

Check interaction between gender and smokers

In [48]:
```
aov1 = aov(log_charges ~  sex*smok
summary(aov1)
```

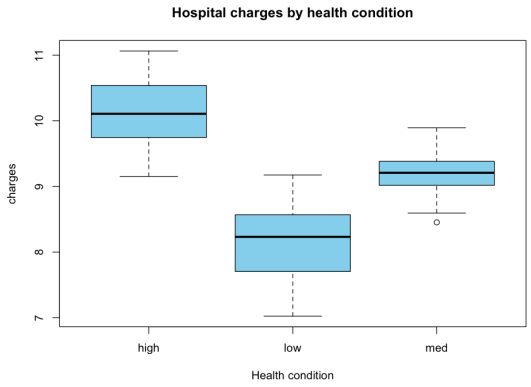
```
             Df Sum Sq Mean Sq F
value Pr(>F)
sex           1    0.3     0.3
0.633 0.4265
smoker        1  369.4   369.4 76
8.627 <2e-16 ***
sex:smoker    1    2.2     2.2
4.567 0.0328 *
Residuals   996  478.7     0.5
---
Signif. codes:  0 ‘***’ 0.001
‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- interaction between gender and smokers is significant for significance level (α = 0.05).
- including the interaction term in regression model might be meaningful

In [49]:
```
tb = table(df$sex,df$health_cond
tb
```

```
        high low med
  female  146 185 164
  male    164 199 142
```

In [50]:
```
boxplot(df$log_charges~df$health_
        main='Hospital charges b
        xlab="Health condition",
```



In [51]:
```
anova_health = aov(log_charges ~
summary(anova_health)
```

```
             Df Sum Sq Mean Sq F
value Pr(>F)
health_cond   2  657.4   328.7
1696 <2e-16 ***
Residuals   997  193.2     0.2
---
Signif. codes:  0 ‘***’ 0.001
‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- ANOVA-test suggests that the effect of *health condition* on the log-hospital charge is significant,
- In such cases, including the *health condition* variable in the regression model is meaningful.

Model building using categorical Covariates

- First we need to convert categorical variable into dummy variables.
- exclude one of the dummy and define a regression model including all the numerical and categorical dummy variables
- excluded category is called the baseline (or reference) category
- if you specify the variable as categorical (or factor), and your data is in dataframe format, lm function will automatically creates dummy variables and creates regression models based on those dummy variables.

1. We include the gender variable in the regression as $Sex - male$.
2. We include the smoker using indicator $smoker : yes$, so $smoker : no$ is a reference category
3. We include interaction of gender and smoker $sex. smoker$
4. We include the dummy varaibles $health - high$, $health - med$, so $health - low$ is a reference category

In [52]:
```
#Manually create indicator (or
Sex_male = as.numeric(new_df$s
Smoker_yes = as.numeric(new_df
sex.smoker =  as.numeric(new_d
Health_low =  as.numeric(df$he
Health_med =  as.numeric(df$he
dummy_df = cbind(num_std_covar
dummy_df = data.frame(dummy_df
head(dummy_df)
```

	age	bmi	chil
	<dbl>	<dbl>	<(
1	-1.4564882	-0.4900222	-0.90092
2	-1.5271401	0.4806364	-0.06673
3	-0.8206214	0.3533098	1.60164
4	-0.4673621	-1.3490633	-0.90092
5	-0.5380140	-0.3279702	-0.90092
6	-0.6086658	-0.8471981	-0.90092

In [53]:
```
fit4 = lm(log_charges ~ ., dat
summary(fit4)
```

```
Call:
lm(formula = log_charges ~ ., 
data = dummy_df)

Residuals:
    Min      1Q  Median      
3Q     Max 
-0.8478 -0.2195  0.0220  0.21
03  1.1649 

Coefficients:
             Estimate Std. Err
or t value Pr(>|t|)    
(Intercept)  9.57299    0.033
23 288.050  < 2e-16 ***
age          0.27884    0.014
57  19.135  < 2e-16 ***
bmi         -0.03716    0.011
19  -3.321 0.000931 ***
children     0.10117    0.010
50   9.633  < 2e-16 ***
Sex_male    -0.08735    0.023
30  -3.749 0.000188 ***
Smoker_yes   0.78299    0.047
44  16.504  < 2e-16 ***
Health_low  -1.14518    0.043
07 -26.588  < 2e-16 ***
Health_med  -0.56166    0.033
24 -16.899  < 2e-16 ***
sex.smoker   0.09337    0.053
39   1.749 0.080611 .  
---
Signif. codes:  0 ‘***’ 0.001
‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 0.32
93 on 991 degrees of freedom
Multiple R-squared:  0.8737,
Adjusted R-squared:  0.8726
F-statistic: 856.7 on 8 and 9
91 DF,  p-value: < 2.2e-16
```

In [54]:
```
fit4 = lm(log_charges~ age + b
summary(fit4)
```

```
Call:
lm(formula = log_charges ~ age + bmi + children + sex + smoker +
    sex * smoker + health_cond, data = new_df)

Residuals:
    Min      1Q  Median      3Q     Max
-0.8478 -0.2195  0.0220  0.2103  1.1649

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         9.57299    0.03323 288.050  < 2e-16 ***
age                 0.27884    0.01457  19.135  < 2e-16 ***
bmi                -0.03716    0.01119  -3.321 0.000931 ***
children            0.10117    0.01050   9.633  < 2e-16 ***
sexmale            -0.08735    0.02330  -3.749 0.000188 ***
smokeryes           0.78299    0.04744  16.504  < 2e-16 ***
health_condlow     -1.14518    0.04307 -26.588  < 2e-16 ***
health_condmed     -0.56166    0.03324 -16.899  < 2e-16 ***
sexmale:smokeryes   0.09337    0.05339   1.749 0.080611 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3293 on 991 degrees of freedom
Multiple R-squared:  0.8737,    Adjusted R-squared:  0.8726
F-statistic: 856.7 on 8 and 991 DF,  p-value: < 2.2e-16
```

The interaction term seems to be insignificant, we fit a model after removing it.

In [55]:
```
fit4 = lm(log_charges~ age +
summary(fit4)
```

```
Call:
lm(formula = log_charges ~ a
ge + bmi + children + sex +
smoker +
    health_cond, data = new_
df)

Residuals:
     Min       1Q   Median
3Q      Max
-0.85564 -0.21042  0.01992
0.20762  1.15133

Coefficients:
               Estimate Std.
Error t value Pr(>|t|)
(Intercept)     9.56781
0.03314 288.744  < 2e-16 ***
age             0.27830
0.01458  19.082  < 2e-16 ***
bmi            -0.03683
0.01120  -3.287 0.001046 **
children        0.10217
0.01050   9.733  < 2e-16 ***
sexmale        -0.06950
0.02097  -3.314 0.000951 ***
smokeryes       0.83285
0.03796  21.942  < 2e-16 ***
health_condlow -1.14947
0.04305 -26.703  < 2e-16 ***
health_condmed -0.56565
0.03319 -17.042  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.00
1 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

Residual standard error: 0.3
296 on 992 degrees of freedo
m
Multiple R-squared:  0.8733,
Adjusted R-squared:  0.8724
F-statistic: 976.6 on 7 and
992 DF,  p-value: < 2.2e-16
```

All other covariates in the above model are statistically significant.

In [56]:
```
logLik(fit4)
```

```
'log Lik.' -305.1729 (df=9)
```

In [57]:
```
summary(fit4)$adj.r.squared
```

0.872384181562898

Interpretation of coefficient of categorical covaraites

The estimated model is

$$\widehat{\log(\text{Hospital charge})} = 9.573$$

- 0.279 for age: For each one-year increase in age, the expected logarithm of hospital charges is estimated to increase by 0.4878 units, holding other variables constant.
- Here missing category from sex varaible is $female$, is called a reference category.
- Simply, (-0.087) is interpreted as the

expected change in the predicted value in y (loghospital-charge) for each unit change in $\mathbb{I}_{male}$ if all others remains constant. However, since $\mathbb{I}_{male}$ is a categorical variable coded as 0 or 1, so a unit change represents switching to category 'male' from the reference category 'female'.

- Being 'male' (compared to 'female', which is the reference category) is associated with a decrease of (-0.087) units in the logarithm of hospital charges, holding other variables constant.
- Hence, (-0.087) is the expected change in predicted value in y when category of sex variable changes from 'Female' to 'Male'.
- β_{cati}: esimated

expected change in the response moving from reference level to the ith dummy variable.

'Female', 0, effect = 9.57299,

Male, 8.8191+(-0.087) = 8.7321

In [58]:
```
n = nrow(df)
k = 8 + 2 # 8 covariates,
fit_values = fit4$fitted.
MSE = 1/n*sum((df$log_cha
MAE = 1/n*sum(abs(df$log_
RMSE = sqrt(1/n * sum((df
evaluation_metrics = c("R
                       "
                      "J
                      "R
                      "M
model4_eval = data.frame(
print(model4_eval)
```

```
           model4
R-sq        0.873
Adj.R-sq    0.872
log-Lik  -305.173
AIC       628.346
BIC       672.516
MSE         0.108
MAE         0.262
RMSE        0.328
```

In [ ]:
```
```

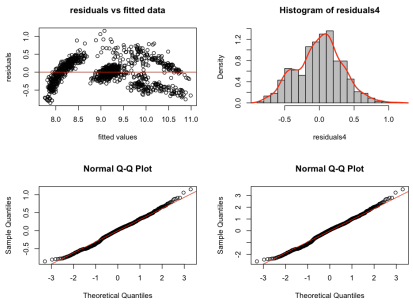
In [60]:
```
results = data.frame(mode
knitr::kable(results, "si
```

```
                 model1
model2       model3      m
odel4
---------  ----------  -
---------  ----------  -
--------
R-sq             0.324
0.345        0.345
0.873
Adj.R-sq         0.324
0.343        0.343
0.872
log-Lik      -1142.118
-1126.266    -1126.266
-305.173
AIC           2290.236
2262.532     2262.532
628.346
BIC           2304.960
2287.070     2287.070
672.516
MSE              0.575
0.557        0.557
0.108
MAE              0.601
0.600        0.600
0.262
RMSE             0.758
0.746        0.746
0.328
```

Residual Analysis (Model4)

1. Scatter plot of residuals against predicted values check for patterns or heteroscedasticity (unequal variance)

In [61]:
```
par(mfrow = c(2, 2))
#produce residual vs. fit
residuals4 = resid(fit4)
plot(fitted(fit4), residu
#add a horizontal line at
abline(0,0, col = "red")
hist(residuals4, breaks =
lines(density(residuals4)
#create Q-Q plot for resi
qqnorm(residuals4)
#add a straight diagonal
qqline(residuals4, col =

res_standard = rstandard(
#create Q-Q plot for resi
qqnorm(res_standard)
#add a straight diagonal
qqline(res_standard, col
```



In [62]:
```
shapiro.test(residuals4)
```

```
        Shapiro-Wilk nor
mality test

data:  residuals4
W = 0.99518, p-value =
0.002967
```

Prediction at new data

We have

$$\hat{y} = \beta_0 1 + \beta_1 X_1 + \ldots + \beta_{p-}$$

where
$X = [1, X_1, X_2, \ldots, X_p]$
is matrix of p covaraites (predictors).

Let $[\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p]$ be the estimated values for set of parameters. Then, for any given covaraites $X^{new} = [1, X_1, X_2, \ldots, X_p]$ we have

$$\hat{y}_{new} = \beta_0 1 + \beta_1 X_1^{new} + \ldots$$

Test the model predictibility for the new data

In [63]:
```
test_data = read.csv("~/I
head(test_data)
```

	age	sex	bmi	childr
	<int>	<chr>	<dbl>	<ir
1	30	male	22.99	
2	24	male	32.70	
3	24	male	25.80	
4	48	male	29.60	
5	47	male	19.19	
6	29	male	31.73	

In [64]:
```
test_num_covariates = sub
test_cat_covariates = sub
```

- If we trained the model with scaled (or standardized) covarites, then we should scale the test covariates with same scaling method
- Model1 and model2 are built without scaling, and model3 is built with scaled numerical covariates.

Prediction using model

```
predict(model, newdata)
```

In [65]:
```
test_cov = subset(test_
test_response = log(tes
t = length(test_respons
t
```

338

Model1 (Simple linear regression model with age as a covariate)

$$y = \beta_0 + \beta_1 \, age$$

In [66]:
```
predict_y = predict(fit
```

In [67]:
```
Test_MSE = 1/t*sum((tes
Test_MAE = 1/t*sum(abs(
Test_RMSE = sqrt(1/t * 
pred_measure = c("Test 
model1_pred = data.fram
model1_pred
```

A data.frame: 3 × 1

	model1
	<dbl>
Test MSE	0.7213
Test MAE	0.6671
Test RMSE	0.8493

Model2 (Full model without scaled covaraites)

$$y = \beta_0 + \beta_1\ age + \beta_2\ bmi$$

In [68]:
```
predict_y = predict(fit
```

In [69]:
```
Test_MSE = 1/t*sum((tes
Test_MAE = 1/t*sum(abs(
Test_RMSE = sqrt(1/t * 
pred_measure = c("Test 
model2_pred = data.fram
model2_pred
```

A data.frame: 3 × 1

	model2
	<dbl>
Test MSE	0.6923
Test MAE	0.6521
Test RMSE	0.8320

Model3 (Model with scaled numerical covariates)

$$y = \beta_0 + \beta_1\ age + \beta_2\ bmi$$

- normalization: $x_{normalize} = \frac{x - mean}{sd}$, we can use R-function scale for the training data
- Test data should be normalized by using the same center and scale of train data.

In [70]:
```
train_cov = subset(df, 
for (j in 1:ncol(train_
    train_meanj = mean(
    train_sdj = sd(trai
    test_cov[, j] = (te
}
head(test_cov)
```

A

	age	bmi
	<dbl>	<dbl>
1	-0.6793177	-1.3019359
2	-1.1032289	0.3037020
3	-1.1032289	-0.8372765
4	0.5924159	-0.2089115
5	0.5217640	-1.9303009
6	-0.7499696	0.1433036

In [71]:
```
#summary(fit3)
beta_vec = fit3$coeffic
round(beta_vec,4)
```

(Intercept): 9.0815 **age:** 0.515 **bmi:** 0.0575 **children:** 0.12

In [72]:
```
predict_y =  predict(fi
```

In [73]:
```
Test_MSE = 1/t*sum((tes
Test_MAE = 1/t*sum(abs(
Test_RMSE = sqrt(1/t * 
pred_measure = c("Test 
model3_pred = data.fram
model3_pred
```

A data.frame: 3 × 1

	model3
	<dbl>
Test MSE	0.6923
Test MAE	0.6521
Test RMSE	0.8320

Model4 (Model with numerical covariates and categorical covariates)

$$y = \beta_0 + \beta_1\ age + \beta_2\ bmi$$

In [74]:
```
beta_vec = fit4$coeffic
round(beta_vec,3)
```

(Intercept): 9.568 **age:** 0.278 **bmi:** -0.037 **children:** 0.102 **sexmale:** -0.069 **smokeryes:** 0.833 **health_condlow:** -1.149 **health_condmed:** -0.566

In [75]:
```
predict_y = predict(fit
```

In [76]:
```
Test_MSE = 1/t*sum((tes
Test_MAE = 1/t*sum(abs(
Test_RMSE = sqrt(1/t * 
pred_measure = c("Test 
model4_pred = data.fram
model4_pred
```

A data.frame: 3 × 1

	model4
	<dbl>
Test MSE	0.1197
Test MAE	0.2731
Test RMSE	0.3459

In [77]:
```
pred_results = data.fra
pred_results
```

A data.frame: 3 × 4

	model1	model2	mod
	<dbl>	<dbl>	<d
Test MSE	0.7213	0.6923	0.6
Test MAE	0.6671	0.6521	0.6
Test RMSE	0.8493	0.8320	0.8

In [78]:
```
train_test_results = rb
train_test_results
```

A data.frame

	model1	model
	<dbl>	<dbl
R-sq	0.3240	0.345
Adj.R-sq	0.3240	0.343
log-Lik	-1142.1180	-1126.266
AIC	2290.2360	2262.532
BIC	2304.9600	2287.070
MSE	0.5750	0.557
MAE	0.6010	0.600
RMSE	0.7580	0.746
Test MSE	0.7213	0.692
Test MAE	0.6671	0.652
Test RMSE	0.8493	0.832

- model2 and model3 are same except considering the scaling of numerical covaraites
- scaling makes easier to compare the effects of predictors on response
- All the predictor variables in model2/model4 are significant

The above results indicate that model2/model4 is the best model.

Prediction interval and corresponding plot

See the link, for more details.

- $(1-\alpha)100\%$ Confidence interval of new predicted values = $\hat{y_h} \pm t_{\frac{\alpha}{2},n-(k+1)} * \mathrm{SE}($

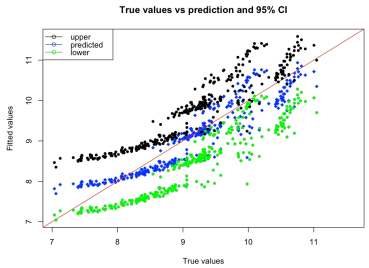
In [79]:
```
predict_output = predi
head(predict_output$fi
```

A matrix: 6 × 3 of type d

	fit	lwr	
1	9.702814	9.051131	10.
2	9.920901	9.271048	10.
3	7.980596	7.332153	8.
4	9.578822	8.928554	10.
5	8.558308	7.908377	9.
6	8.213253	7.564926	8.

Prediction plot

In [80]:
```
predict_interval = pre
plot(test_response, pr
     ylim =  range(pre
    type = "p", pch =
abline(0, 1, col = "re

points(test_response,p
points(test_response,p
legend("topleft", lege
```



- The best model satisfies the normality assumption in our example.
- More sophisticated models are possible for the insurance data.
- Models beyond linear regression, such as nonlinear model, or other factors that can influence hospital charges linearly are still considerable