CALORIES BURNT PREDICTION

Sonam

23N0061

2023-11-16

INTRODUCTION Background

Calorie is a unit of heat energy. Health and fitness are becoming increasingly important to individuals and society as a whole. As people seek to live healthier lifestyles, they are turning to wearable devices and fitness trackers to monitor their physical activity and track their progress. One important metric that these devices track is the number of calories burnt during physical activity. Accurately predicting calorie burn can help individuals set and achieve fitness goals and can also inform health coaching and wellness tracking programs . The motivation for this research is to develop a model that can accurately predict calorie burn during physical activity. This has potential applications in a range of settings, including personalized health coaching, fitness tracking, and wellness programs. By developing an accurate calorie burn prediction model, we can help individuals make more informed decisions about their physical activity and improve their overall health and well-being

PROJECT OBJECTIVE

The objective of a calories burned prediction model is to estimate the number of calories an individual is likely to burn during a specific physical activity based on various input features. The goal is to create a model that can accurately predict calorie expenditure, considering factors such as user characteristics, activity details, and health metrics.

DATA

In this research, the dataset was collected from Kaggle, a popular platform for data scientists and machine learning practitioners to access and share datasets.

link : https://www.kaggle.com/code/muskanjha/calories-burnt-prediction/input?select=exercise.csv

In this work, the dataset contained over 15,000 records and 9 variables.

1. User_ID: Unique identifier for each user (integer).
2. Gender: Gender of the user (character: “male” or “female”).
3. Age: Age of the user (integer).
4. Height: Height of the user (integer).
5. Weight: Weight of the user (integer).
6. Duration: Duration of the activity (integer).

7. Heart_Rate: Heart rate of the user during the activity (integer).
8. Body_Temp: Body temperature of the user during the activity (numeric).
9. Calories: Number of calories burned during the activity (integer).

```
knitr::include_graphics("Web capture_25-11-2023_18446_.jpeg")
```

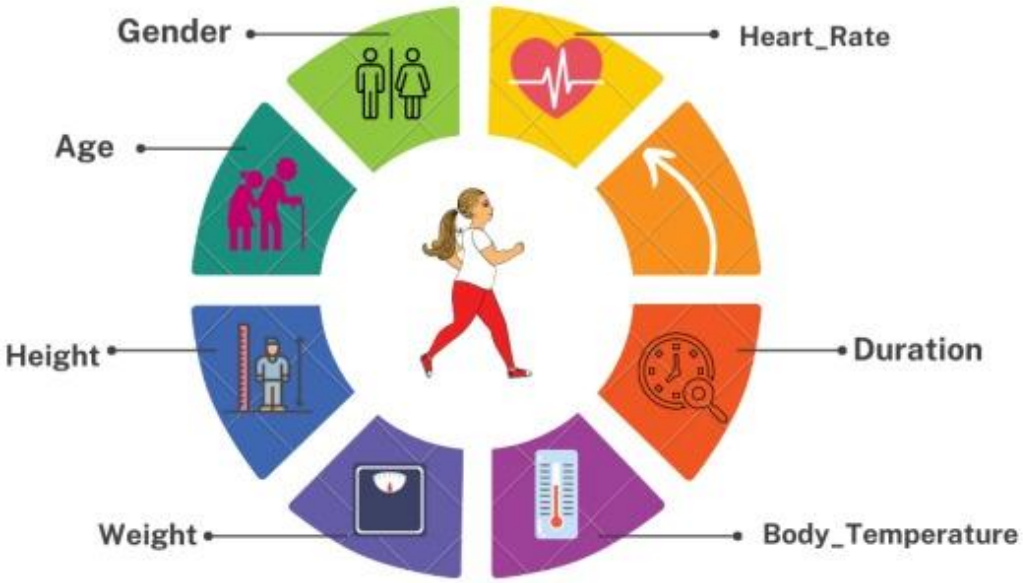
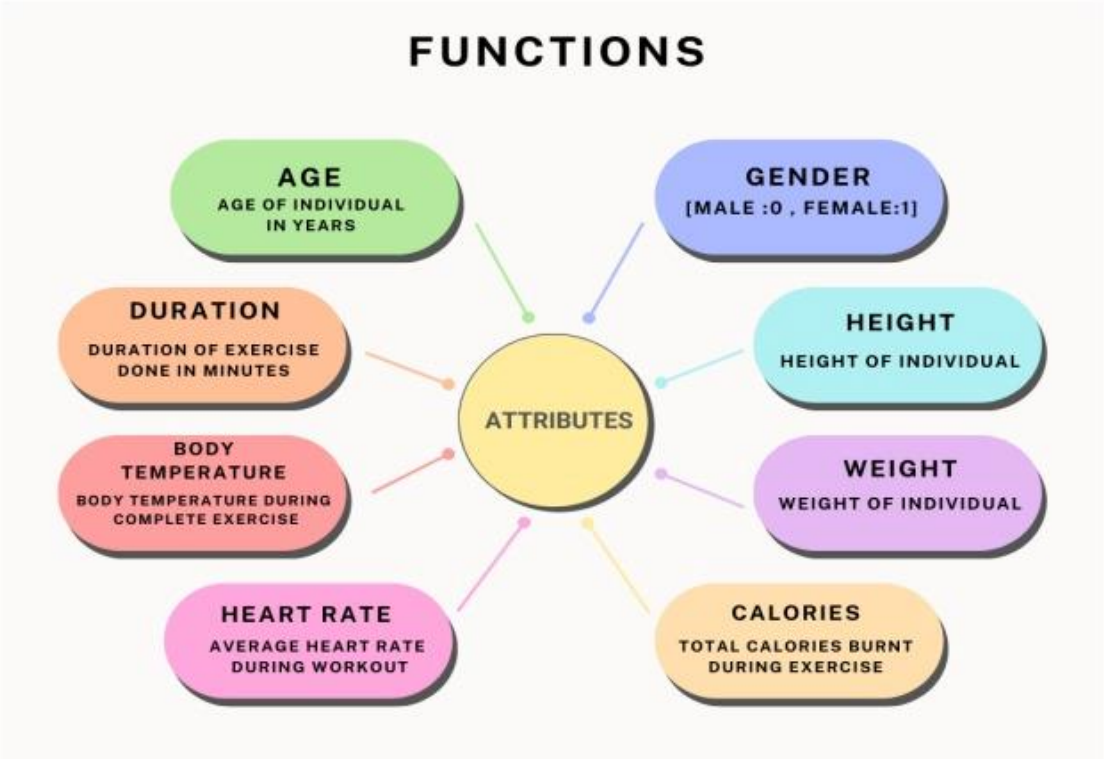


Fig-1 Attributes of calories burnt prediction

```
knitr::include_graphics("Web capture_25-11-2023_184735_.jpeg")
```




```
rm(list = ls())
```

Importing and viewing the dataset:

```
df = read.csv("exercise.csv")
df=as.data.frame(df)
view(df)
df %>% head() %>% knitr::kable()
```

User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
14733363	male	68	190	94	29	105	40.8	231
14861698	female	20	166	60	14	94	40.3	66
11179863	male	69	179	79	5	88	38.7	26
16180408	female	34	179	71	13	100	40.5	71
17771927	female	27	154	58	10	81	39.8	35
15130815	female	36	151	50	23	96	40.7	123

```
str(df)
```

```
## 'data.frame':    15000 obs. of  9 variables:
##  $ User_ID   : int  14733363 14861698 11179863 16180408 17771927 15130815
19602372 11117088 12132339 17964668 ...
##  $ Gender    : chr  "male" "female" "male" "female" ...
##  $ Age       : int  68 20 69 34 27 36 33 41 60 26 ...
##  $ Height    : int  190 166 179 179 154 151 158 175 186 146 ...
##  $ Weight    : int  94 60 79 71 58 50 56 85 94 51 ...
##  $ Duration  : int  29 14 5 13 10 23 22 25 21 16 ...
##  $ Heart_Rate: int  105 94 88 100 81 96 95 100 97 90 ...
##  $ Body_Temp : num  40.8 40.3 38.7 40.5 39.8 40.7 40.5 40.7 40.4 40.2 ...
##  $ Calories  : int  231 66 26 71 35 123 112 143 134 72 ...
```

Checking the number of missing values:

```
na_count = sapply(df, function(x) sum(length(which(is.na(x)))))
na_count = data.frame(na_count)
na_count %>% knitr::kable()
```

	na_count
User_ID	0
Gender	0
Age	0
Height	0
Weight	0
Duration	0
Heart_Rate	0
Body_Temp	0
Calories	0

Since there are no missing values , we can proceed with our analysis

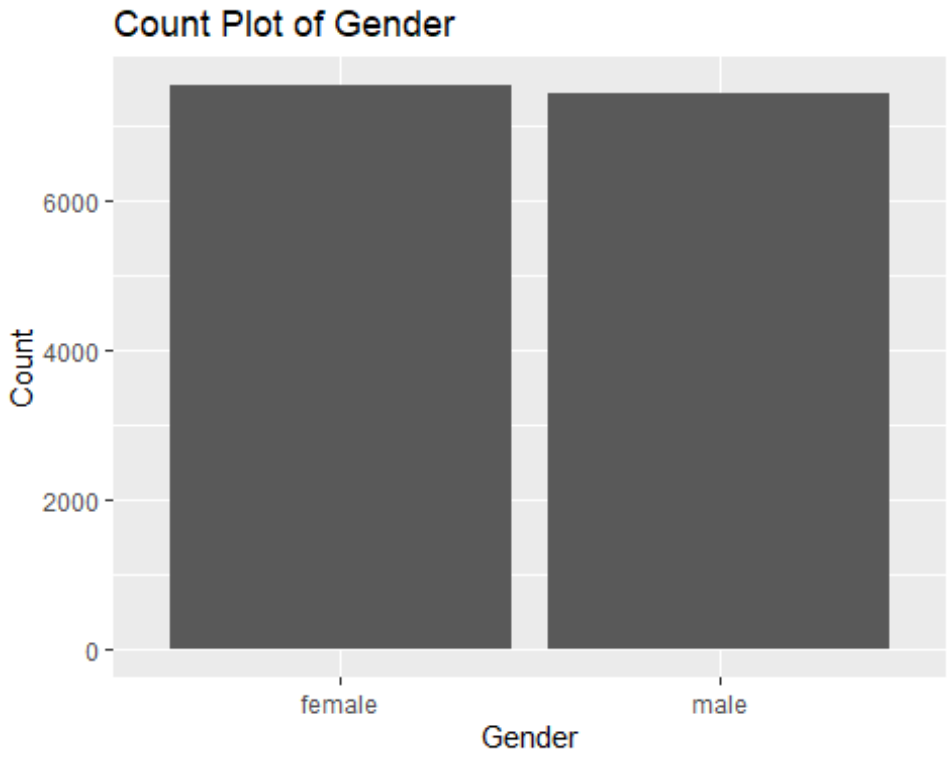
Summary Measures:

```
# Install and load the psych package
#install.packages("psych")
library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:DescTools':
##
##     AUC, ICC, SD

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
# Use describe() to get detailed descriptive statistics
describe(df)
```

```
##             vars     n        mean         sd     median     trimmed
mad
## User_ID        1 15000 14977358.54 2872851.45 14997285.0 14973538.80
3680613.06
## Gender*        2 15000        1.50       0.50        1.0        1.50
0.00
## Age            3 15000       42.79      16.98       39.0       41.60
19.27
## Height         4 15000      174.47      14.26      175.0      174.47
16.31
## Weight         5 15000       74.97      15.04       74.0       74.56
17.79
## Duration       6 15000       15.53       8.32       16.0       15.53
10.38
## Heart_Rate     7 15000       95.52       9.58       96.0       95.53
10.38
## Body_Temp      8 15000       40.03       0.78       40.2       40.12
0.74
## Calories       9 15000       89.54      62.46       79.0       85.12
74.13
##                   min         max     range  skew kurtosis       se
## User_ID    10001159.0 19999647.0 9998488.0  0.00    -1.19 23456.73
## Gender*           1.0        2.0       1.0  0.01    -2.00     0.00
## Age              20.0       79.0      59.0  0.47    -0.95     0.14
## Height          123.0      222.0      99.0 -0.01    -0.51     0.12
## Weight           36.0      132.0      96.0  0.23    -0.68     0.12
## Duration          1.0       30.0      29.0  0.00    -1.18     0.07
## Heart_Rate       67.0      128.0      61.0 -0.01    -0.64     0.08
## Body_Temp        37.1       41.5       4.4 -0.99     0.52     0.01
## Calories          1.0      314.0     313.0  0.51    -0.72     0.51
```

```r
# Load the ggplot2 package
library(ggplot2)

# Create a count plot
ggplot(df, aes(x = Gender)) +
  geom_bar() +
  ggtitle("Count Plot of Gender") +
  xlab("Gender") +
  ylab("Count")
```



Roughly balanced gender distribution

```
gender_counts <- table(df$Gender)
print(gender_counts)
```

```
##
## female   male
##   7553   7447
```

```
df$Gender <- ifelse(df$Gender == "male", 0, 1)

# Calculate the correlation matrix
correlation_matrix <- cor(df[2:9])

# Print the correlation matrix
print(correlation_matrix)
```

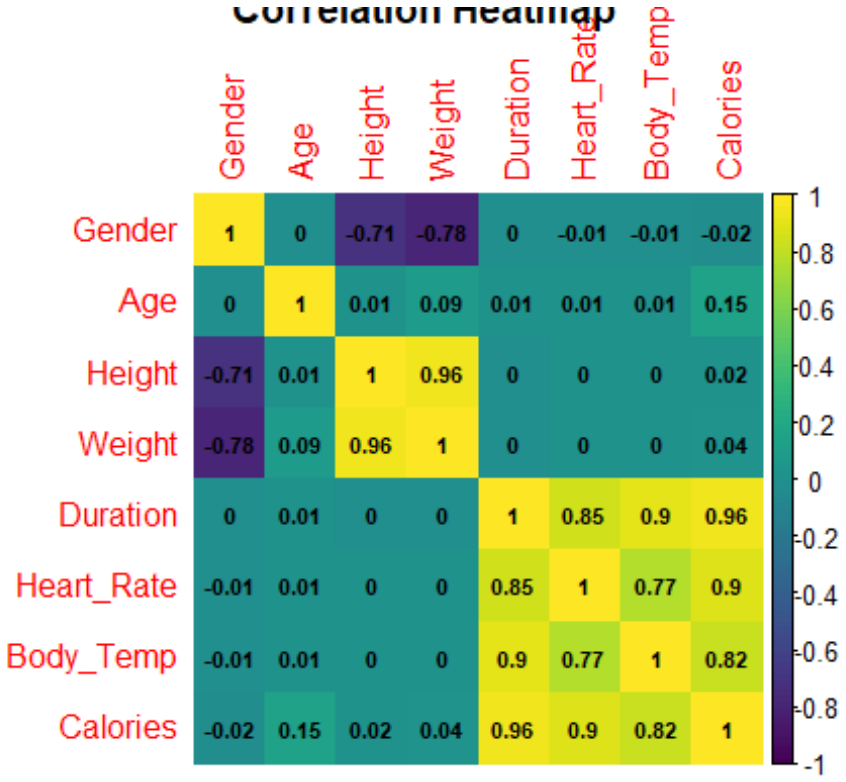
```
##                  Gender          Age        Height       Weight
Duration
## Gender      1.000000000 -0.003222434 -0.7105342171 -0.783186214 -
0.003439982
## Age        -0.003222434  1.000000000  0.0095538873  0.090094215
0.013246686
## Height     -0.710534217  0.009553887  1.0000000000  0.958450781 -
0.004625309
## Weight     -0.783186214  0.090094215  0.9584507810  1.000000000 -
0.001884338
## Duration   -0.003439982  0.013246686 -0.0046253089 -0.001884338
```

```
1.000000000
## Heart_Rate -0.011555216  0.010481592  0.0005280365  0.004311299
0.852868902
## Body_Temp  -0.007263838  0.013174736  0.0012002349  0.004095163
0.903166924
## Calories   -0.022357162  0.154395131  0.0175367679  0.035480582
0.955420533
##               Heart_Rate    Body_Temp    Calories
## Gender     -0.0115552158 -0.007263838 -0.02235716
## Age         0.0104815924  0.013174736  0.15439513
## Height      0.0005280365  0.001200235  0.01753677
## Weight      0.0043112993  0.004095163  0.03548058
## Duration    0.8528689016  0.903166924  0.95542053
## Heart_Rate  1.0000000000  0.771528551  0.89788206
## Body_Temp   0.7715285515  1.000000000  0.82455776
## Calories    0.8978820606  0.824557758  1.00000000
```

```
correlation <- cor(df[2:9])
#install.packages("viridis")
#install.packages("corrplot")
# Set up the plotting area
par(mfrow=c(1,1), mar=c(5, 4, 2, 2))


# Annotate the heatmap using the 'corrplot' package
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(correlation,
         method = 'color',
         col = viridis::viridis(100),  # Use the same color palette as the
heatmap
         addCoef.col = 'black',
         number.cex = 0.7,
         title = 'Correlation Heatmap')
```



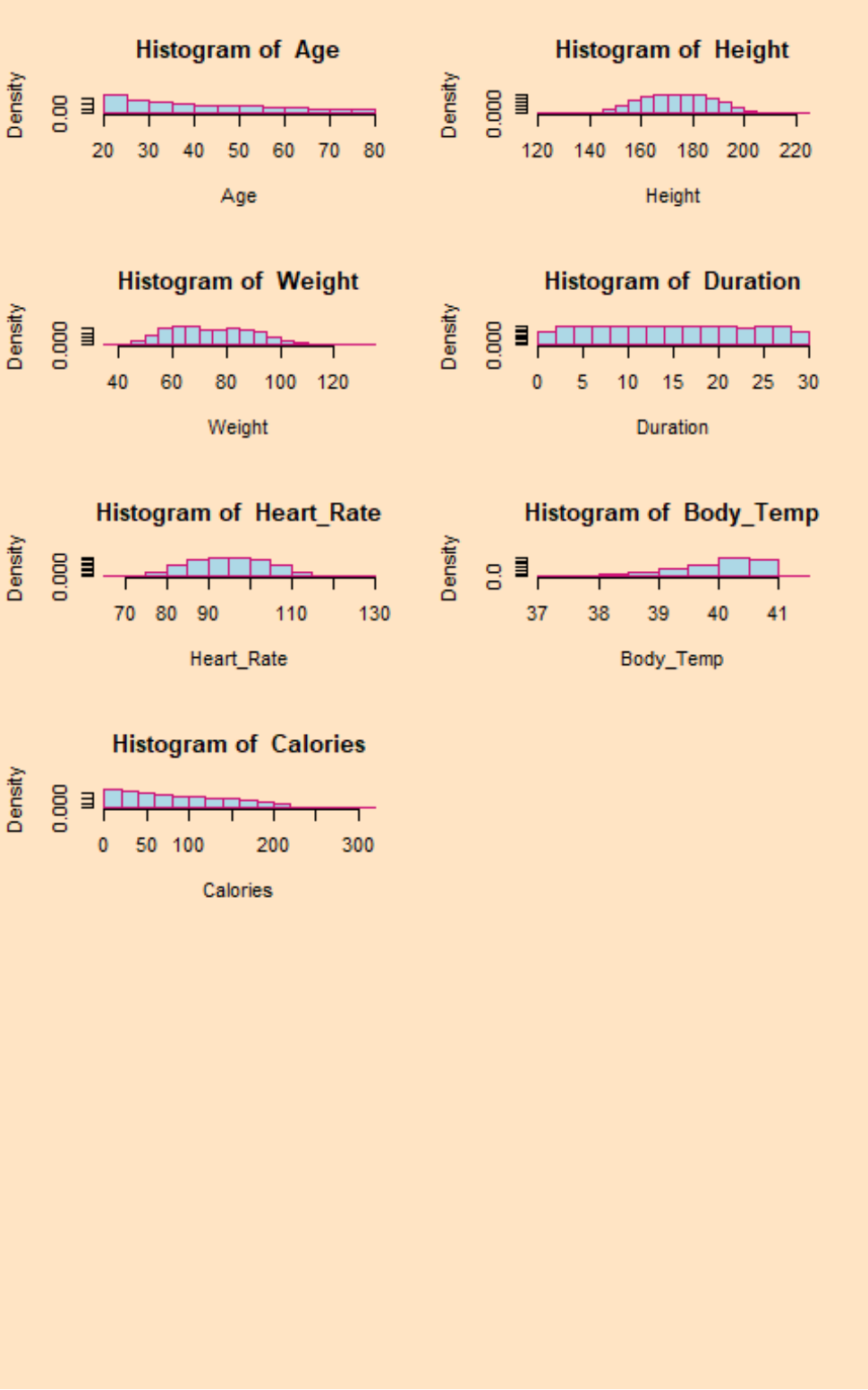

```r
# Adjust the layout
layout(matrix(c(1, 2), nrow = 1, byrow = TRUE), widths = c(4, 1))

# Show the plot
```

- **Gender:**
  - Strong negative correlation with height (-0.71) and weight (-0.78).
  - Indicates females generally have lower height and weight.
- **Age:**
  - Positive correlation with calories (0.15).
  - Suggests older individuals burn more calories during activities.
- **Duration:**
  - Strong positive correlation with calories (0.96).
  - Longer exercise durations associated with higher calorie burn.
  - High correlation with heart rate (0.85) and body temperature (0.90).
- **Heart Rate:**
  - Strong positive correlation with body temperature (0.77).
  - Increase in heart rate linked to higher body temperature.
  - High correlation with duration (0.85) and calories (0.90).
- **Body Temperature:**
  - Strong positive correlation with heart rate (0.77) and duration (0.90).
  - Indicates higher body temperatures during activities with longer durations and higher heart rates.

- **Weight and Height:**
  - Strong positive relationship (correlation of 0.96).
  - Indicates a strong positive association between weight and height.
- **Calories:**
  - Strongly associated with duration (0.96) and heart rate (0.90).
  - Various factors, including exercise duration and heart rate, strongly linked to calorie burn during physical activities.

Histogram of certain columns:

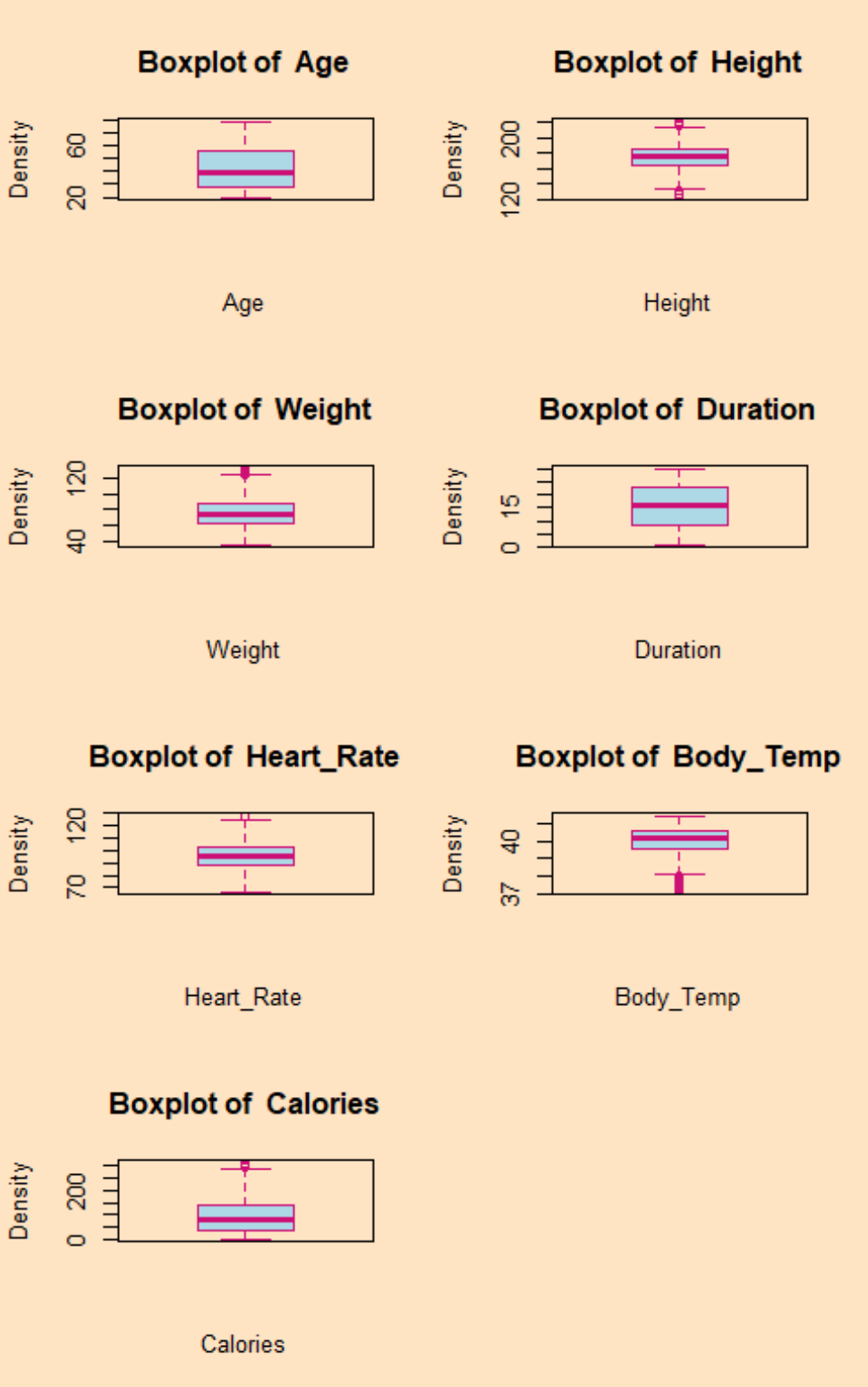
```
par(mfrow = c(3, 2), bg = "bisque1")
col_id = c(3:9)
for(j in 1:length(col_id))
{
  hist(df[, col_id[j]], col = "lightblue",
       border = "deeppink3", freq = F,
       main = paste("Histogram of ", colnames(df[,col_id])[j]),
       xlab = paste(colnames(df[,col_id][j])),
       ylab = "Density")
}
```



** Boxplot of certain columns**

```
par(mfrow = c(2, 2), bg = "bisque1")
col_id = c(3:9)
for(j in 1:length(col_id))
```

```
{
  boxplot(df[,col_id[j]], col = "lightblue",
       border = "deeppink3", freq = F,
       main = paste("Boxplot of ", colnames(df[,col_id])[j]),
       xlab = paste(colnames(df[,col_id][j])),
       ylab = "Density")
}
```



#OUTLIERS FOR HEIGHT

```
# Calculate IQR
data=df$Height
q1 <- quantile(data, 0.25)
```

```r
q3 <- quantile(data, 0.75)
iqr <- q3 - q1
iqr
```

```
## 75%
##  21
```

```r
# Define a threshold for outliers
threshold <- 1.5 * iqr

# Identify outliers
outliers <- data[data < (q1 - threshold) | data > (q3 + threshold)]
outliers
```

```
##  [1] 132 217 123 132 132 127 218 126 132 222 219 218 132 217
```

The average height for Indian men is 5.8 feet (177 cm), and that for women is 5.3 feet (162 cm) the height may differ,they are indeed genuine data points, it makes sense to retain them.

#OUTLIERS FOR WEIGHT

```r
# Calculate IQR
data=df$Weight
q1 <- quantile(data, 0.25)
q3 <- quantile(data, 0.75)
iqr <- q3 - q1
iqr
```

```
## 75%
##  24
```

```r
# Define a threshold for outliers
threshold <- 1.5 * iqr

# Identify outliers
outliers <- data[data < (q1 - threshold) | data > (q3 + threshold)]
outliers
```

```
## [1] 124 132 128 126 126 124
```

Healthy Weight: 65kg to 75kg. Overweight: 75kg to 95kg. Obese: 95kg to 125kg. Very Obese: More than 125kg They are indeed genuine data points, it makes sense to retain them.these are the weights of obese

#OUTLIERS FOR Heart_rate

```r
# Calculate IQR
data=df$Heart_Rate
q1 <- quantile(data, 0.25)
q3 <- quantile(data, 0.75)
```

```r
iqr <- q3 - q1
iqr
```

```
## 75%
##  15
```

```r
# Define a threshold for outliers
threshold <- 1.5 * iqr

# Identify outliers
outliers <- data[data < (q1 - threshold) | data > (q3 + threshold)]

outliers
```

```
## [1] 128
```

A normal resting heart rate should be between 60 to 100 beats per minute, but it can vary from minute to minute. Our age and general health can also affect our pulse rate, so it's important to remember that a 'normal' pulse can vary from person to person. Yes, it's normal for our heart rate to increase to 130 to 150 beats per minute or more when we exercise – this is because your heart is working to pump more oxygen-rich blood around your body.

#OUTLIERS FOR Body Temperature

```r
# Calculate IQR
data=df$Body_Temp
q1 <- quantile(data, 0.25)
q3 <- quantile(data, 0.75)
iqr <- q3 - q1
iqr
```

```
## 75%
##   1
```

```r
# Define a threshold for outliers
threshold <- 1.5 * iqr

# Identify outliers
outliers <- data[data < (q1 - threshold) | data > (q3 + threshold)]
outliers
```

```
##   [1] 37.8 37.7 38.0 37.9 37.6 37.9 37.7 37.9 37.9 37.8 37.7 37.9 37.8
38.0 37.7
##  [16] 37.8 37.7 38.0 37.5 38.0 37.7 37.7 38.0 38.0 37.5 38.0 37.9 38.0
37.8 37.4
##  [31] 37.9 37.8 37.7 38.0 37.9 37.6 37.7 37.7 38.0 37.7 38.0 37.3 37.8
37.7 38.0
##  [46] 37.6 37.8 37.8 37.9 37.9 37.8 37.9 38.0 37.4 38.0 37.8 37.7 37.7
38.0 37.7
##  [61] 38.0 38.0 37.5 37.9 37.6 37.9 37.8 37.5 37.9 37.9 37.7 37.9 37.7
```

```
37.8 38.0
##  [76] 37.9 37.9 37.7 37.7 37.9 37.9 37.7 37.7 38.0 37.6 37.7 37.8 37.4
37.9 37.5
##  [91] 37.9 37.9 37.8 38.0 37.9 37.7 37.8 37.9 37.3 37.8 37.8 37.8 37.8
37.9 37.7
## [106] 37.7 37.3 37.5 37.6 37.5 37.7 37.8 37.3 37.6 37.8 38.0 38.0 38.0
38.0 37.6
## [121] 38.0 38.0 38.0 37.8 38.0 37.9 37.7 37.7 37.8 37.4 38.0 37.4 37.2
37.9 37.8
## [136] 37.8 37.9 37.8 37.9 37.8 38.0 37.7 38.0 37.9 37.8 37.7 37.6 37.8
37.7 37.8
## [151] 37.6 37.9 37.8 37.8 38.0 37.8 37.8 37.5 38.0 37.9 37.7 37.9 38.0
37.9 37.1
## [166] 37.8 37.8 37.9 37.8 37.7 37.9 37.8 37.9 38.0 37.7 38.0 38.0 37.5
37.7 37.7
## [181] 37.2 37.3 38.0 37.4 38.0 37.8 38.0 38.0 38.0 37.7 37.7 37.7 37.8
37.9 37.7
## [196] 37.9 37.9 37.6 38.0 37.9 37.4 37.8 37.5 37.5 38.0 37.9 37.4 37.9
37.5 38.0
## [211] 37.8 38.0 38.0 38.0 38.0 37.8 37.9 37.6 37.9 38.0 38.0 38.0 37.8
37.4 37.9
## [226] 37.7 37.8 37.6 37.5 37.8 37.6 37.8 37.9 37.9 38.0 37.7 37.7 37.7
37.7 37.3
## [241] 37.8 37.8 37.6 38.0 37.8 37.9 38.0 37.6 38.0 37.8 37.6 38.0 37.4
37.6 37.7
## [256] 38.0 37.9 38.0 37.9 37.6 37.9 37.7 37.8 38.0 37.8 38.0 37.7 37.9
37.3 37.9
## [271] 37.6 37.9 37.5 37.8 37.7 37.9 37.9 38.0 38.0 37.4 37.9 37.9 37.8
37.5 37.7
## [286] 38.0 37.6 37.6 38.0 38.0 38.0 37.8 37.8 37.7 37.8 37.6 37.9 37.5
37.5 37.8
## [301] 37.7 38.0 37.8 37.8 37.7 38.0 37.7 37.9 38.0 37.4 37.9 37.7 38.0
37.9 38.0
## [316] 37.6 38.0 37.9 38.0 37.2 37.5 37.7 37.4 37.7 37.5 37.5 38.0 37.6
38.0 37.8
## [331] 38.0 37.8 38.0 38.0 37.7 37.7 37.9 37.9 38.0 37.8 37.4 38.0 37.9
37.8 37.6
## [346] 37.9 37.9 37.5 38.0 38.0 38.0 37.9 37.8 37.9 37.7 38.0 37.9 37.4
37.8 37.4
## [361] 37.8 37.7 37.5 38.0 37.9 37.9 38.0 38.0 37.4
```

The average body temperature is 37 C. A high-grade fever is present when the oral temperature is above 38.5°C but here we are gettting outliers below 38.5 ,that is quiet normal so,we will not drop these data points

#OUTLIERS FOR CALORIES

```
# Calculate IQR
data=df$Calories
q1 <- quantile(data, 0.25)
```

```r
q3 <- quantile(data, 0.75)
iqr <- q3 - q1
iqr

## 75%
## 103

# Define a threshold for outliers
threshold <- 1.5 * iqr
# Identify outliers
outliers <- data[data < (q1 - threshold) | data > (q3 + threshold)]
outliers

## [1] 314 295 300 295
```

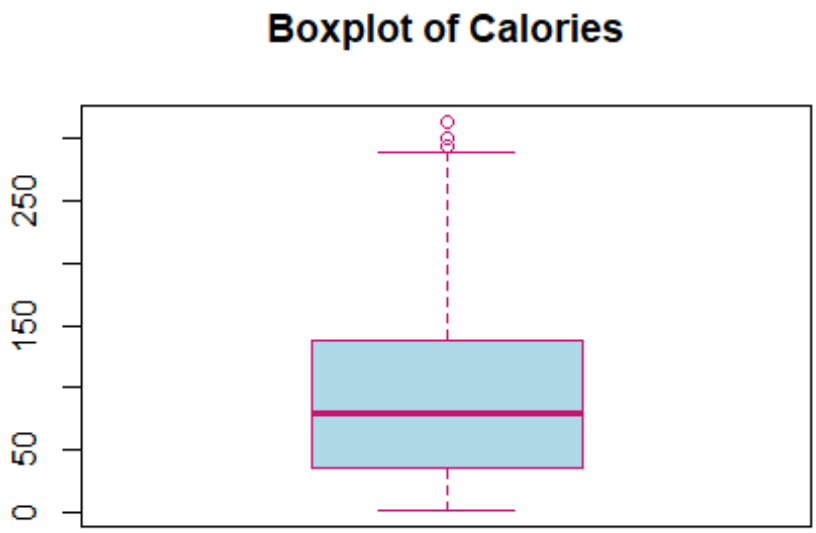
Histogram of Calories

```r
hist(df[,"Calories"], col = "lightblue",
       border = "deeppink3", freq = F,
       main = paste("Histogram of Calories"),
       xlab = 'Calories',
       ylab = "Density")
```

The data of Calories is positively skewed

Boxplot of Calories

```
boxplot(df[,"Calories"], col = "lightblue",
        border = "deeppink3", freq = F,
        main = "Boxplot of Calories")
```



Linear Regression Linear regression models the relationships between at least one explanatory variable (independent) and an outcome variable (dependent). When there is one independent variable, the procedure is known as simple linear regression.

Simple linear regression: Simple linear regression is defined by the linear function: Y= β0 + β1X + Ɛ Y is the predicted value of the dependent variable (Y) for any given value of the independent variable (X). β0 is the intercept, the predicted value of Y when the Xis 0. β1is the regression coefficient – how much we expect Y to change as X increases. X is the independent variable. Ɛis the error of the estimate, or how much variation there is in our estimate of the regression coefficient.

Assumption of Data: We can use R to check that our data meet the four main assumption of linear model.

1. Independence of observation:The first assumption of linear regression is the independence of observations. Independence means that there is no relation between the independents variables. We find correlation between independents variable and make sure they are not too highly correlated. Because we have only one

independent variable at a time, we don't need to test for any hidden relation between them.

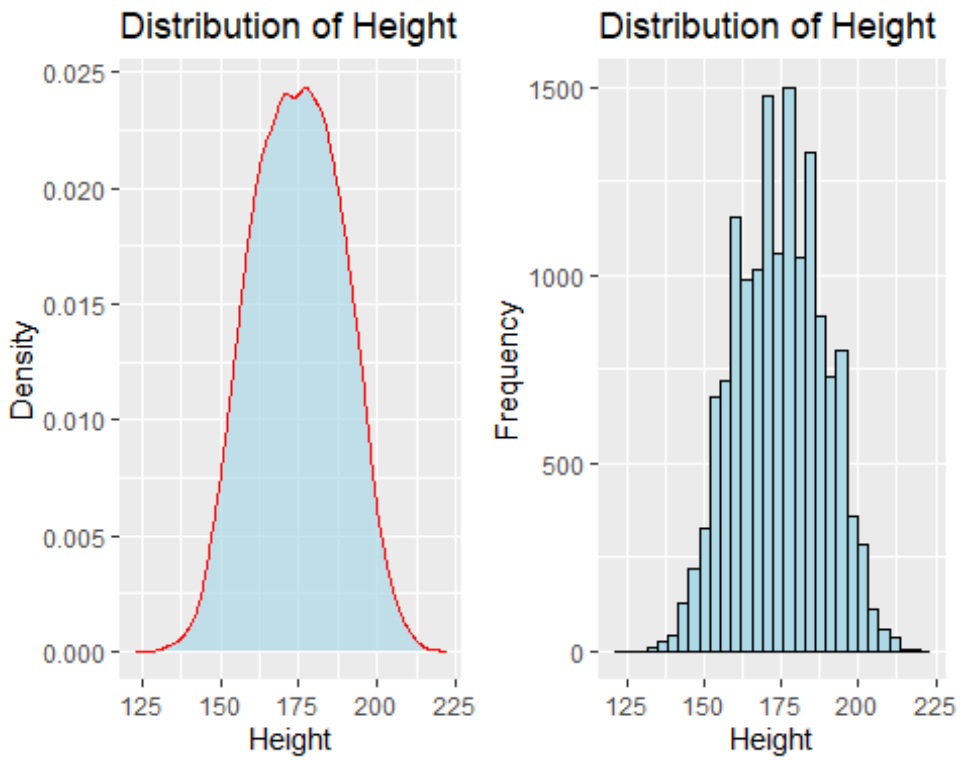
2. Normality: The second assumption of Linear Regression is that the residuals should follow a normal distribution. Once you obtain the residuals from your model, this is relatively easy to test using either a histogram or a QQ Plot. To check whether the dependent variable follows the normal distribution we use histogram.

3. Linearity: The relationship between dependent and independent variable must be linear. We can test this visually with a scatter plot to see if the distribution of data point could be described with a straight line.

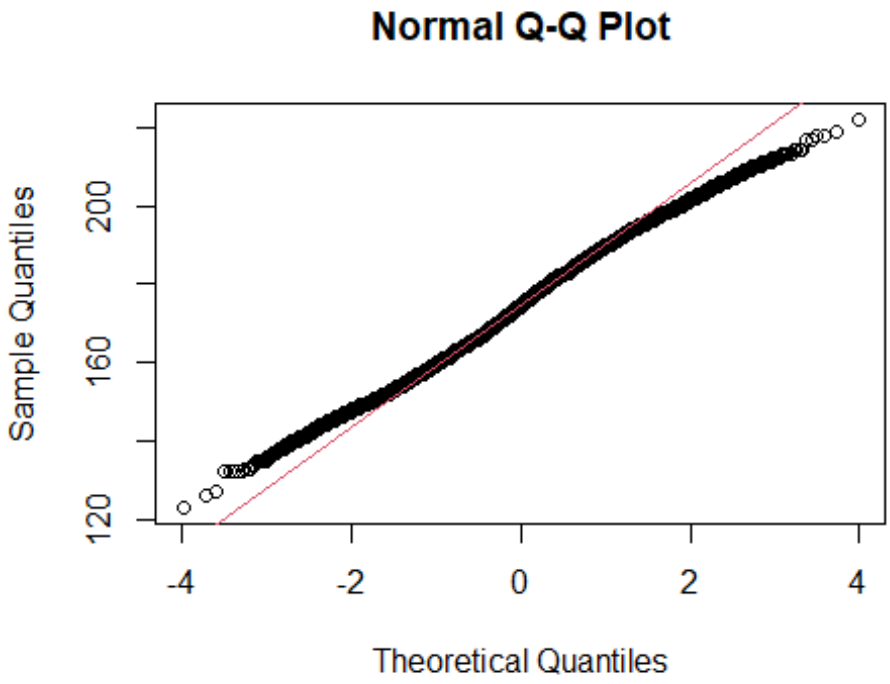
4. Homoscedasticity: Homoscedasticity in a model means that the error is constant along the values of the dependent variable. The best way for checking homoscedasticity is to make a scatterplot with the residuals against the dependent variable.

Normality: Use the hist() function to test whether our dependent variable follows a normal distribution HEIGHT

```
# Set up the plot size
options(repr.plot.width=6, repr.plot.height=6)

# Load required libraries
#library(ggplot2)
 library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

# Create a density plot using ggplot2
density_plot <- ggplot(df, aes(x=Height)) +
  geom_density(fill="lightblue", color="red", alpha=0.7) +
  labs(title="Distribution of Height", x="Height", y="Density")

# Create a histogram using ggplot2
hist_plot <- ggplot(df, aes(x=Height)) +
  geom_histogram(fill="lightblue", color="black", bins=30) +
  labs(title="Distribution of Height", x="Height", y="Frequency")

# Arrange the plots in a 1x2 grid
grid.arrange(density_plot, hist_plot, ncol=2)
```




```r
# Show the plots

qqnorm(df$Height)
qqline(df$Height, col = 2)
```



Since, qq-plot is very much close to the actual line , so the distribution of Height in our data is normally distributed. 😊

The distribution of observations is bell-shaped, so we can proceed with the linear regression

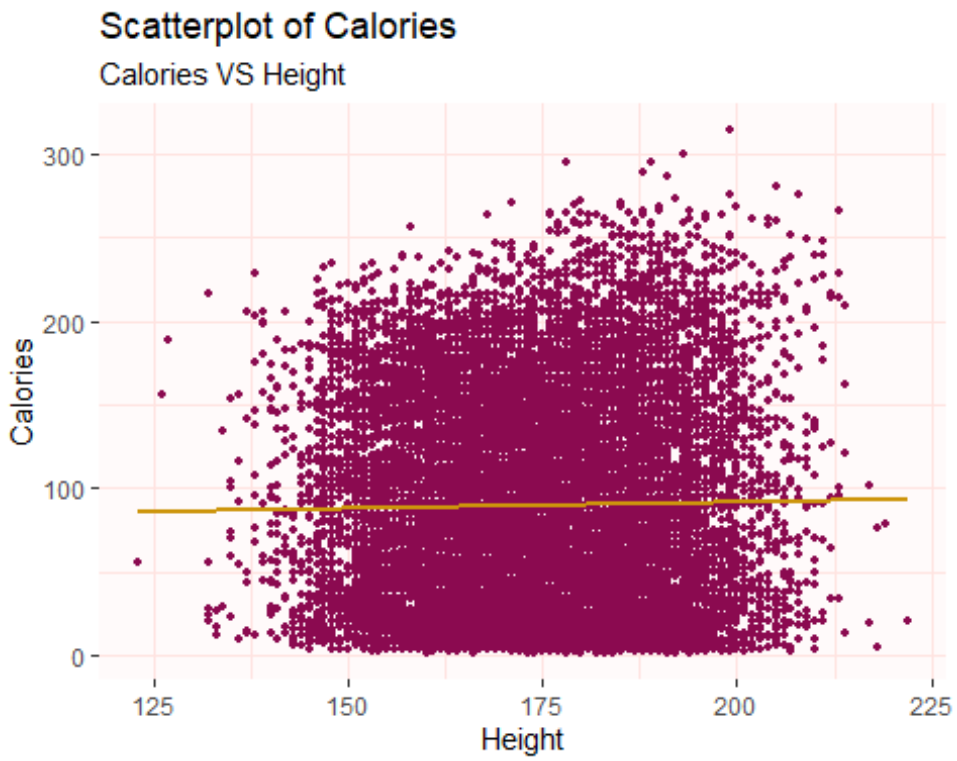
Linearity:We can check this by using scatterplot between Calories and Height

```
#Response = Calories
# Predictor = Height
 cor(df$Calories,df$Height)
```

```
## [1] 0.01753677
```

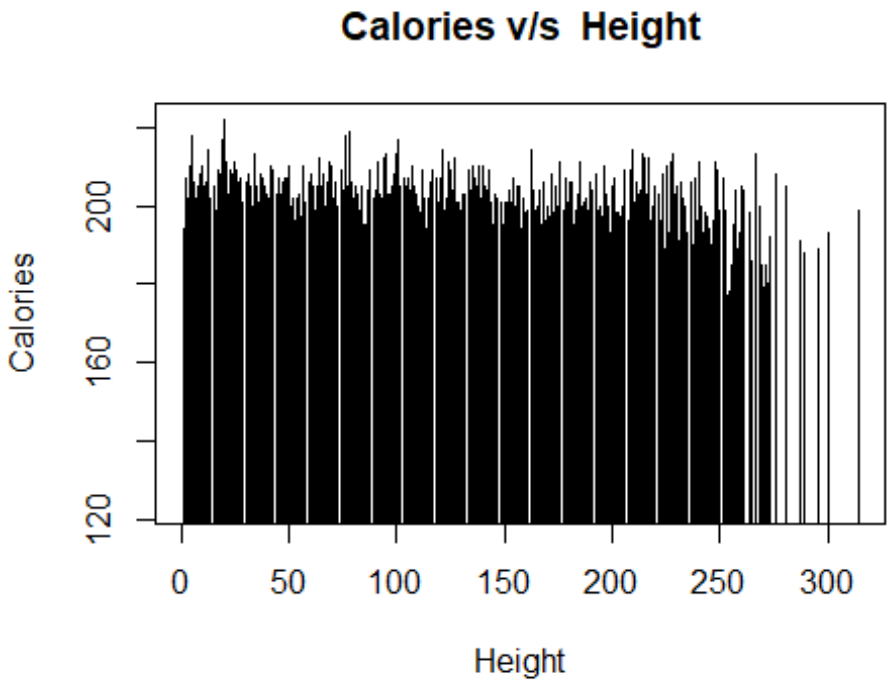
```
df %>%
  ggplot(aes(Height, Calories)) +
  geom_point(pch = 20, col = "deeppink4", size = 2) +
  geom_smooth(method = lm, se = F, col = "darkgoldenrod3") +
  labs(title = "Scatterplot of Calories",
       subtitle = "Calories VS Height") +
  xlab("Height") +
  ylab("Calories") +
  theme(panel.background = element_rect(fill = 'snow1'),
        panel.grid.major = element_line(color = 'mistyrose'),
        panel.grid.minor = element_line(color = 'mistyrose'))
```

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



Comment: Here, The correlation coefficient between Calories and Height is 0.01753677. That is, there's a weak positive correlation between these two variables. From the fitted scatterplot,the result is also verified.

```
plot(df$Calories,df$Height,main = "Calories v/s  Height",ylab =
"Calories",xlab = "Height",type = "h")
```



Although the relationship between Calories and Height is a bit less clear, it still appears linear. We can proceed with linear regression.

Fitting a linear model between Calories vs Height$

```
model = df %>%
  lm(formula =Calories~Height)
model %>% summary()

##
## Call:
## lm(formula = Calories ~ Height, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -90.04 -54.52 -10.50  48.03 222.58
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 76.13730    6.26022  12.162   <2e-16 ***
## Height       0.07682    0.03576   2.148   0.0317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.45 on 14998 degrees of freedom
## Multiple R-squared:  0.0003075,  Adjusted R-squared:  0.0002409
## F-statistic: 4.614 on 1 and 14998 DF,  p-value: 0.03173
```

The linear regression model suggests a statistically significant but modest relationship between Height and Calories:

Intercept: Predicted Calories when Height is zero is 76.14, but this might not have practical meaning.

Height (Slope): For each one-unit increase in Height, predicted Calories increase by approximately 0.077 units.

Statistical Significance: The relationship is statistically significant (p = 0.0317), but the p-value is higher compared to other models.

Model Fit: The model has limited explanatory power (low R-squared), indicating that Height explains only a negligible portion of the variability in Calories.

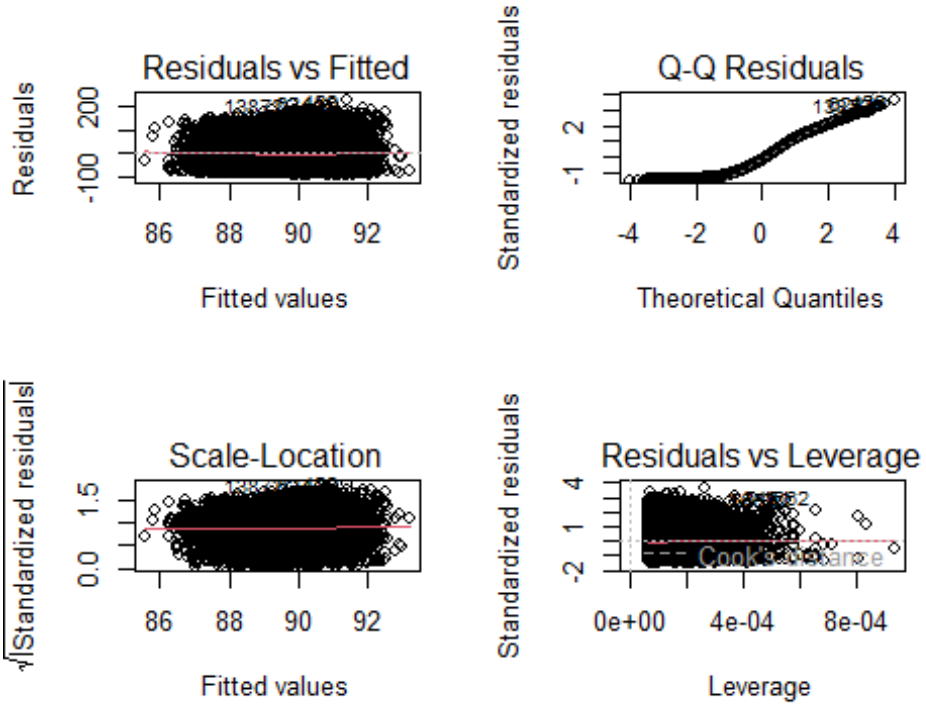
Overall Model Significance: The overall model is statistically significant, but the effect size is relatively small.

Conclusion: While statistically significant, the practical impact of Height on predicting Calories appears limited. Consider exploring additional factors for a more comprehensive understanding of calorie prediction

Check for homoscedasticity:

Before proceeding with data visualization, we should make sure that our models fit the homoscedasticity assumption of the linear model. We should check that our model is actually a good fit for the data, and that we don't have large variation in the model error, by running this code:

```
par(mfrow=c(2,2))
plot(lm(formula =df$Calories~df$Height))
```

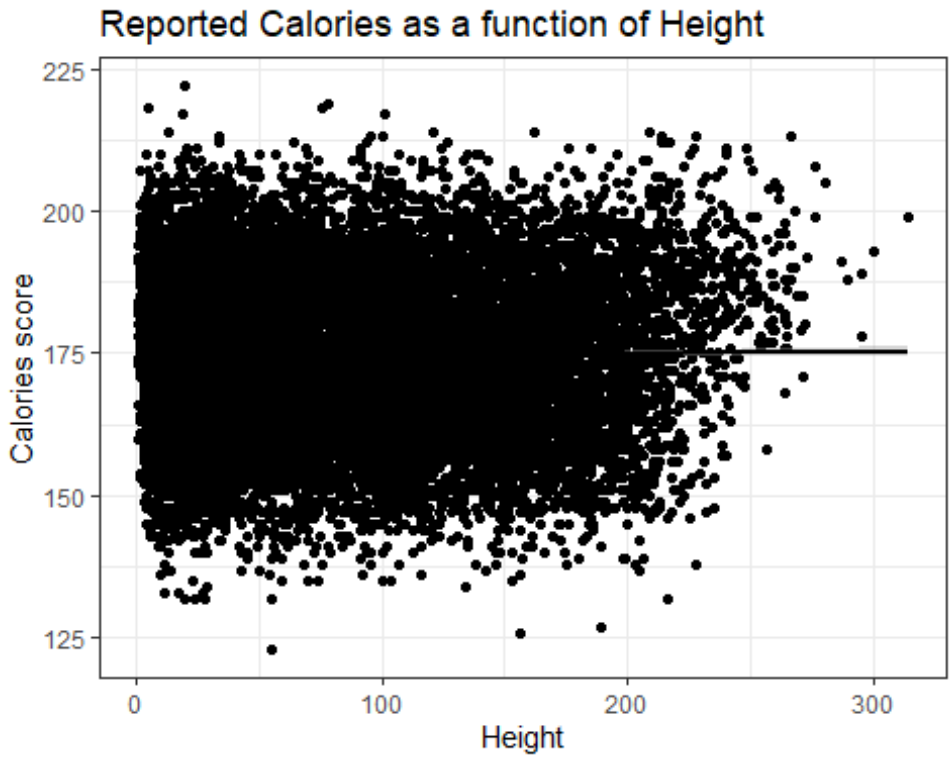


The most important thing to look for is that the red lines representing the mean of the residuals are all basically horizontal and centered around zero. This means there are no outliers or biases in the data that would make a linear regression invalid.

Based on these residuals, we can say that our model meets the assumption of homoscedasticity.

Visualize the result with graph:

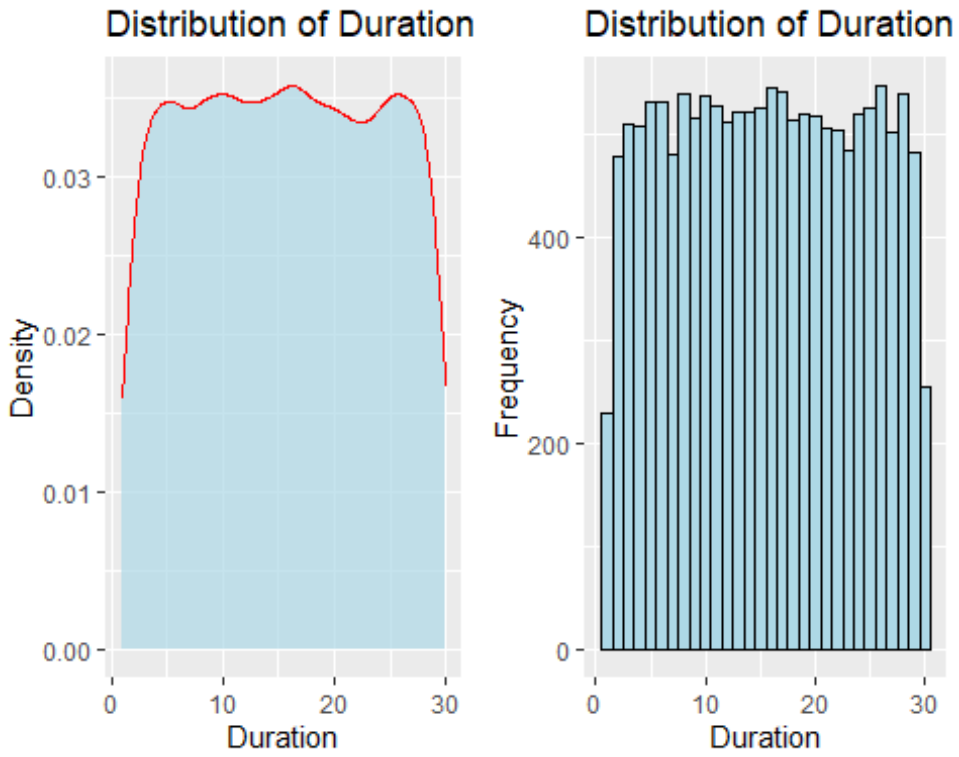
```
Calories.graph<-ggplot(df, aes(x=Calories, y=Height))+
 geom_point()
Calories.graph <- Calories.graph + geom_smooth(method="lm",
col="black")
## `geom_smooth()` using formula = 'y ~ x'
Calories.graph +
 theme_bw() +
 labs(title = "Reported Calories as a function of Height",
 x = "Height",
 y = "Calories score")+
annotate(geom="text",label="Calories = 76.13730 + (0.07682*Height)")

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



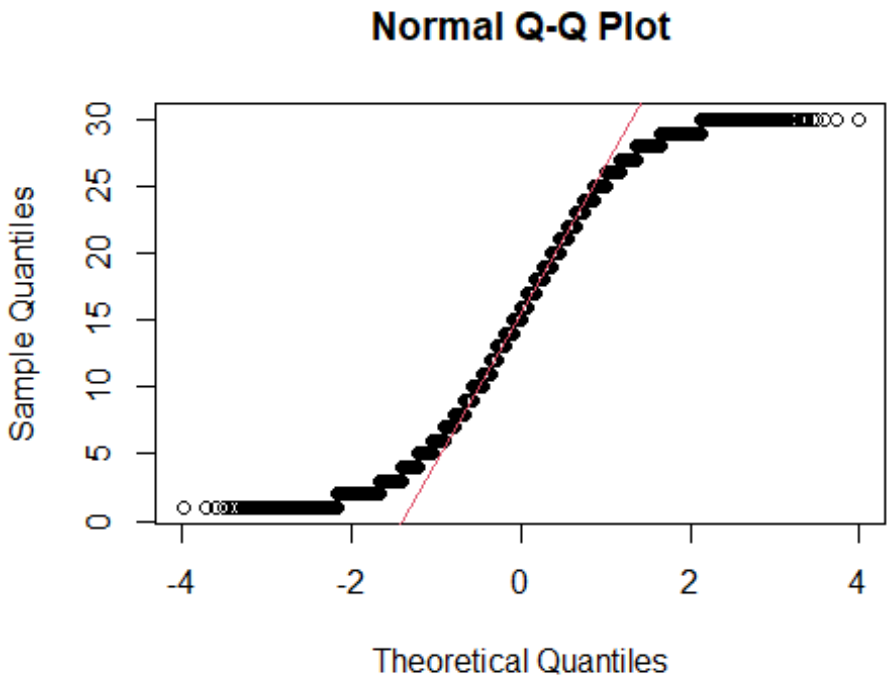
We found a significant relationship between Calories and Height, with a 0.07682-unit increase in reported Calories for every 1 unit increase in Height

DURATION

```r
# Set up the plot size
options(repr.plot.width=6, repr.plot.height=6)

# Load required libraries
library(ggplot2)
library(gridExtra)

# Create a density plot using ggplot2
density_plot <- ggplot(df, aes(x=Duration)) +
  geom_density(fill="lightblue", color="red", alpha=0.7) +
  labs(title="Distribution of Duration", x="Duration", y="Density")

# Create a histogram using ggplot2
hist_plot <- ggplot(df, aes(x=Duration)) +
  geom_histogram(fill="lightblue", color="black", bins=30) +
  labs(title="Distribution of Duration", x="Duration", y="Frequency")

# Arrange the plots in a 1x2 grid
grid.arrange(density_plot, hist_plot, ncol=2)
```




```r
# Show the plots

qqnorm(df$Duration)
qqline(df$Duration, col = 2)
```



Since, qq-plot is not that much close to the actual line , so the distribution of Duration the data may deviate from a normal distribution.

S-shaped Curve: An S-shaped curve in the QQ plot may suggest skewness in the data.

so, we can still proceed with the linear regression

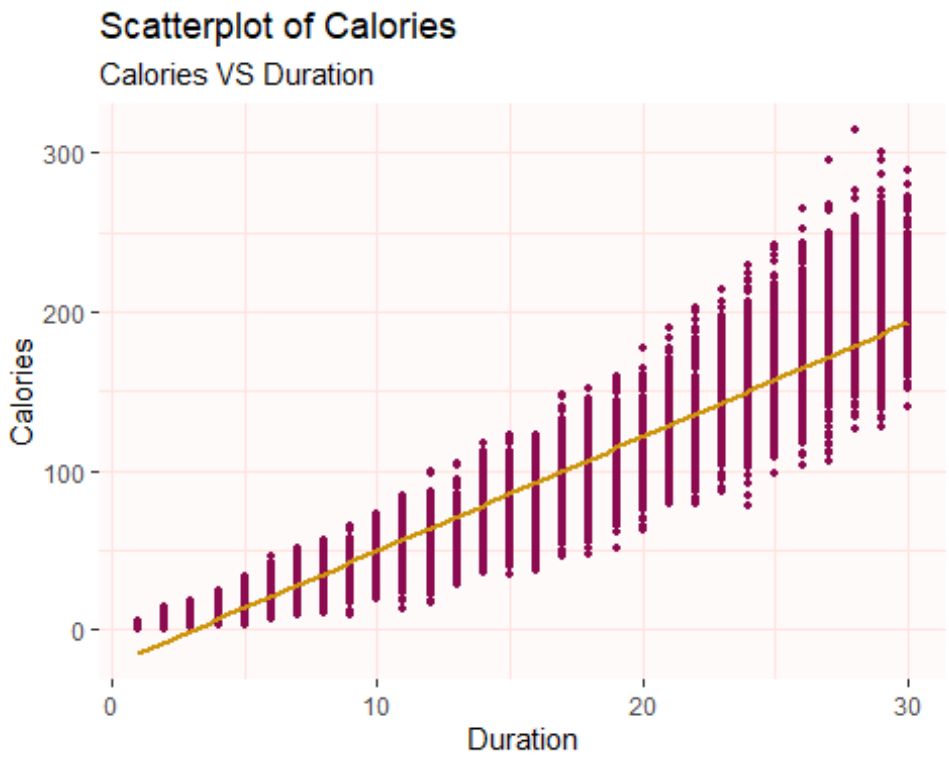
Linearity:We can check this by using scatterplot between Calories and Duration

```
#Response = Calories
# Predictor = Duration
 cor(df$Calories,df$Duration)
```

```
## [1] 0.9554205
```

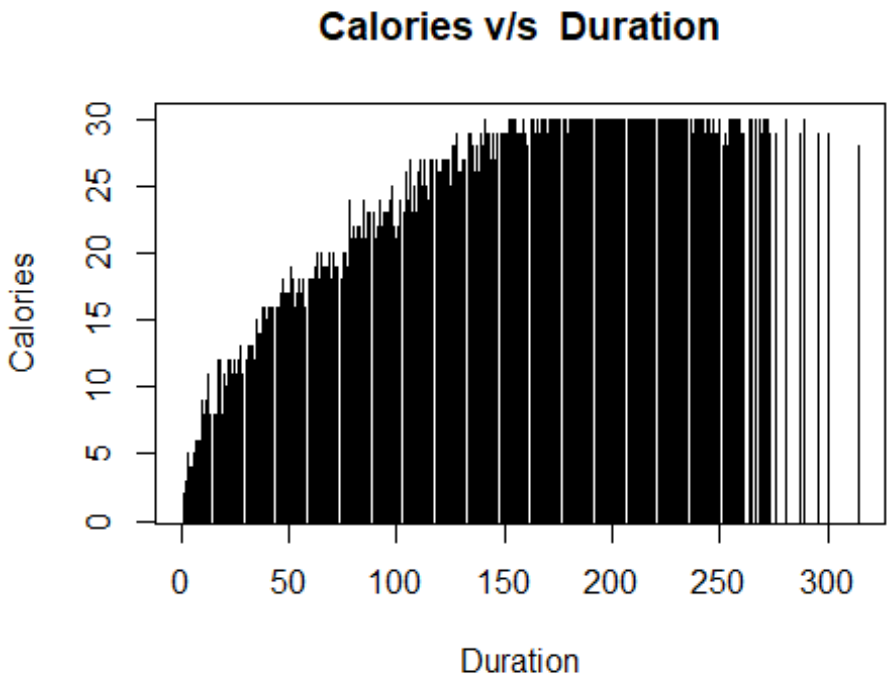
```
df %>%
  ggplot(aes(Duration, Calories)) +
  geom_point(pch = 20, col = "deeppink4", size = 2) +
  geom_smooth(method = lm, se = F, col = "darkgoldenrod3") +
  labs(title = "Scatterplot of Calories",
       subtitle = "Calories VS Duration") +
  xlab("Duration") +
  ylab("Calories") +
  theme(panel.background = element_rect(fill = 'snow1'),
        panel.grid.major = element_line(color = 'mistyrose'),
        panel.grid.minor = element_line(color = 'mistyrose'))
```

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



Comment: Here, The correlation coefficient between Calories and Height is 0.9554205. Strong positive correlation with calories (0.96).Longer exercise durations associated with higher calorie burn. From the fitted scatterplot,the result is also verified.

```
plot(df$Calories,df$Duration,main = "Calories v/s  Duration",ylab =
"Calories",xlab = "Duration",type = "h")
```



the relationship between Calories and Duration appears linear. We can proceed with linear regression.

Fitting a linear model between Calories vs Duration$

```
model = df %>%
  lm(formula =Calories~Duration)
model %>% summary()

##
## Call:
## lm(formula = Calories ~ Duration, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.290 -11.215  -0.215   9.995 135.019
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.8597     0.3189  -68.55   <2e-16 ***
## Duration      7.1729     0.0181  396.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.44 on 14998 degrees of freedom
## Multiple R-squared:  0.9128, Adjusted R-squared:  0.9128
## F-statistic: 1.571e+05 on 1 and 14998 DF,  p-value: < 2.2e-16
```

**The linear regression model indicates a highly significant and strong relationship between Duration and Calories. Specifically:*

Intercept: Predicted Calories when Duration is zero is -21.86, but interpret cautiously.

Duration (Slope): For each one-unit increase in Duration, predicted Calories increase by approximately 7.17 units.

Statistical Significance: Both intercept and Duration are highly significant ($p < 2e\text{-}16$).

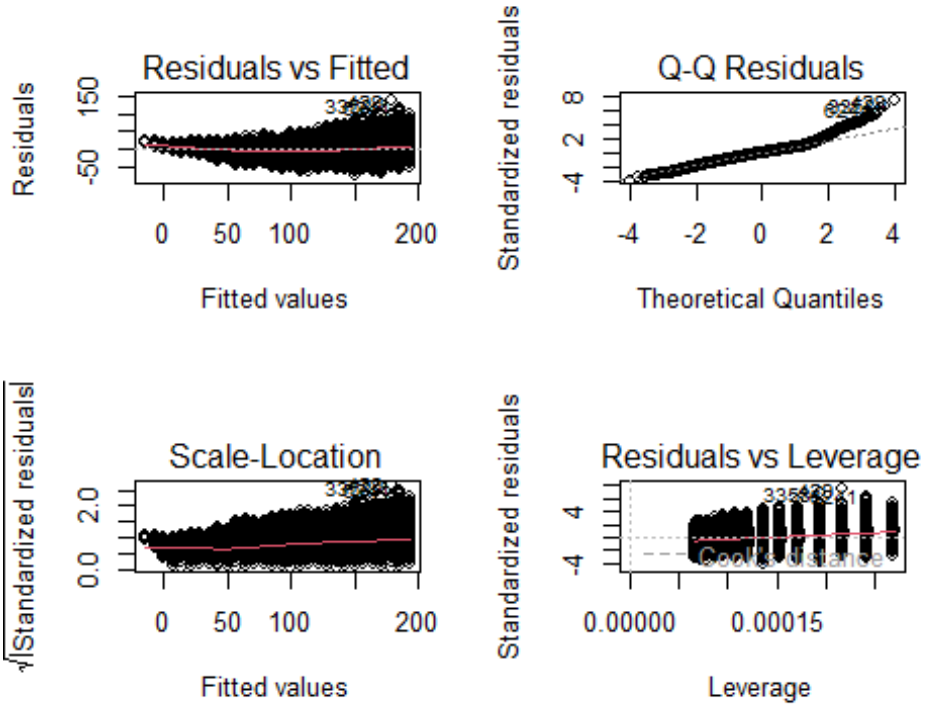
Model Fit: The model explains about 91.28% of the variability in Calories based on the high R-squared value.

Overall Model Significance: The overall model is statistically significant, suggesting its usefulness in predicting Calories based on Duration.

Check for homoscedasticity:

Before proceeding with data visualization, we should make sure that our models fit the homoscedasticity assumption of the linear model. We should check that our model is actually a good fit for the data, and that we don't have large variation in the model error, by running this code:

```
par(mfrow=c(2,2))
plot(lm(formula =df$Calories~df$Duration))
```



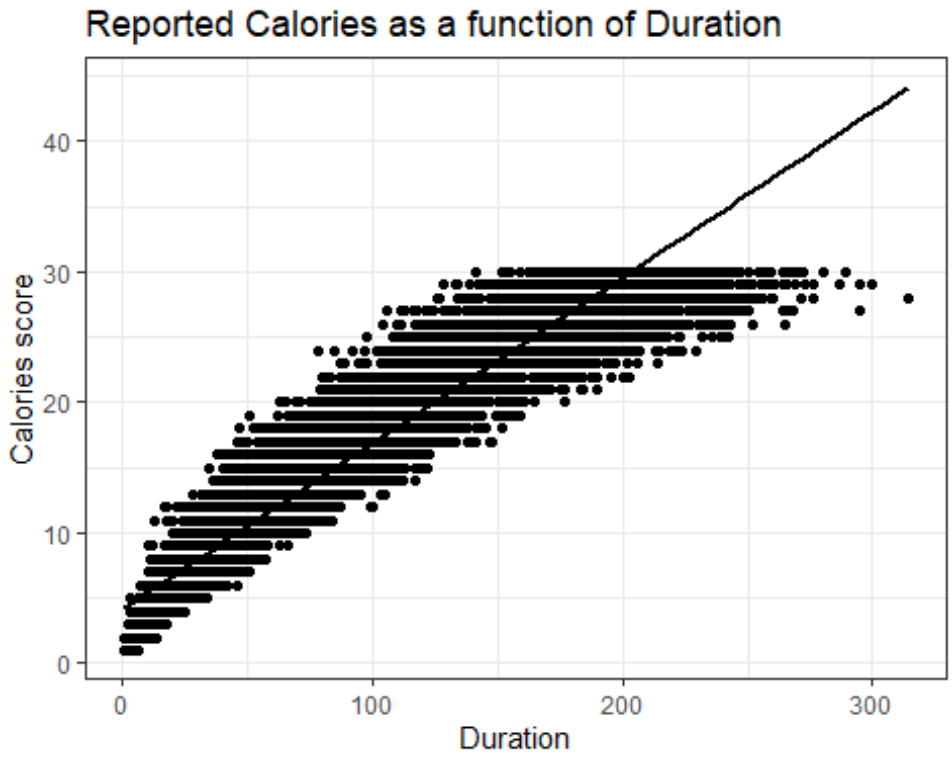
The most important thing to look for is that the red lines representing the mean of the residuals are

all basically horizontal and centered around zero. This means there are no outliers or biases in the data that would make a linear regression invalid.

Based on these residuals, we can say that our model meets the assumption of homoscedasticity.

Visualize the result with graph:

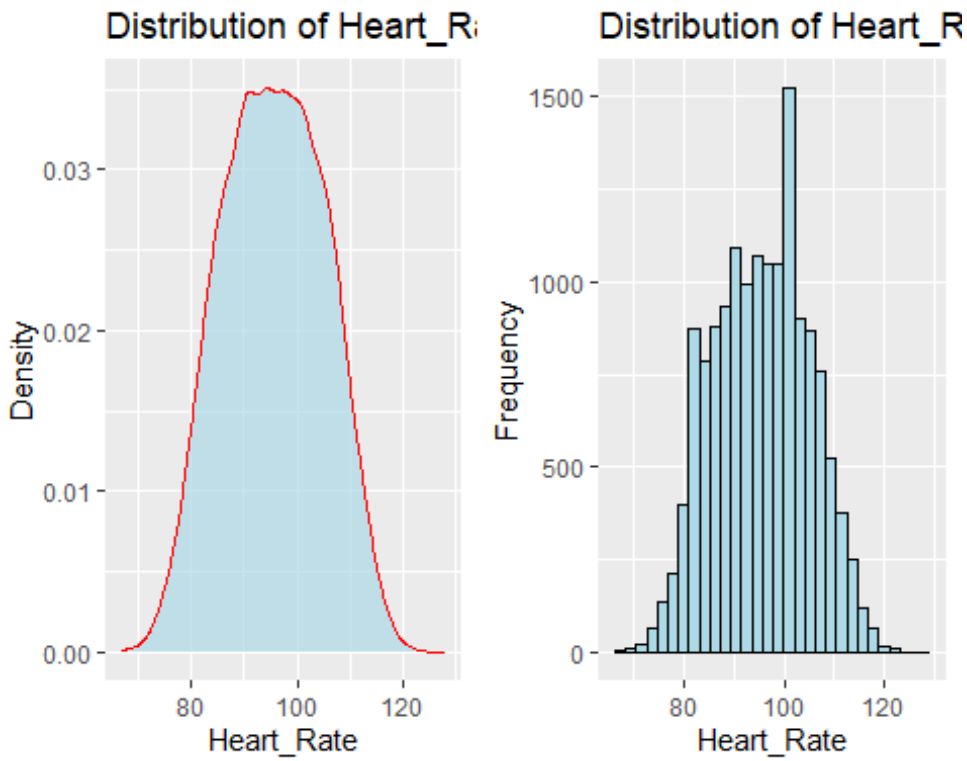
```
Calories.graph<-ggplot(df, aes(x=Calories, y=Duration))+
 geom_point()
Calories.graph <- Calories.graph + geom_smooth(method="lm",
col="black")
## `geom_smooth()` using formula = 'y ~ x'
Calories.graph +
 theme_bw() +
 labs(title = "Reported Calories as a function of Duration",
 x = "Duration",
 y = "Calories score")+
annotate(geom="text",label="Calories = -21.8597 + (7.1729*Duration)")

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



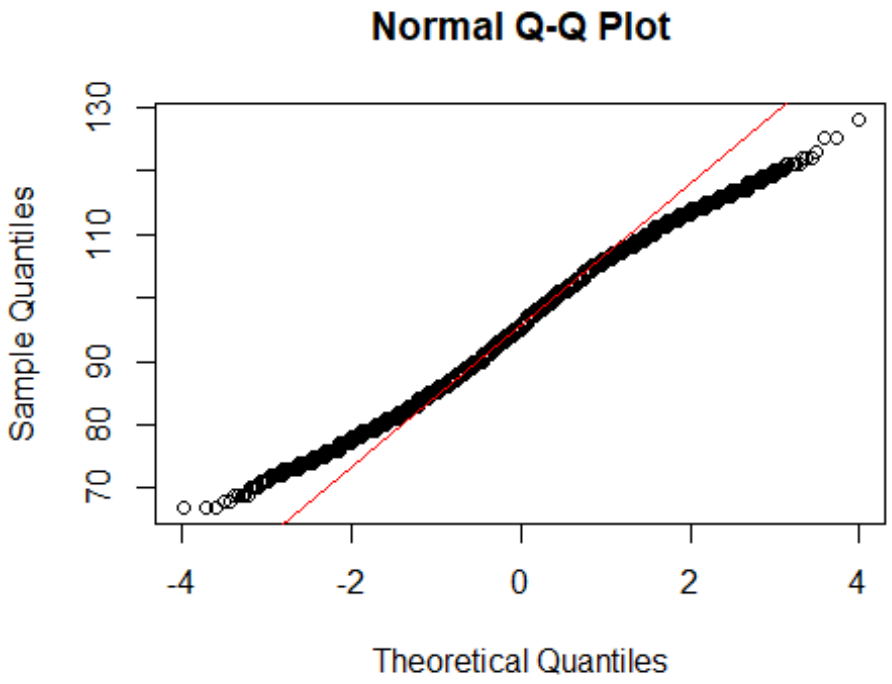
We found a significant relationship between Calories and Duration, with a 7.1729-unit increase in reported Calories for every 1 unit increase in Duration

```
Heart_Rate
```

```r
# Set up the plot size
options(repr.plot.width=6, repr.plot.height=6)

# Load required libraries
library(ggplot2)
library(gridExtra)

# Create a density plot using ggplot2
density_plot <- ggplot(df, aes(x=Heart_Rate)) +
  geom_density(fill="lightblue", color="red", alpha=0.7) +
  labs(title="Distribution of Heart_Rate", x="Heart_Rate", y="Density")

# Create a histogram using ggplot2
hist_plot <- ggplot(df, aes(x=Heart_Rate)) +
  geom_histogram(fill="lightblue", color="black", bins=30) +
  labs(title="Distribution of Heart_Rate", x="Heart_Rate", y="Frequency")

# Arrange the plots in a 1x2 grid
grid.arrange(density_plot, hist_plot, ncol=2)
```




```r
# Show the plots

qqnorm(df$Heart_Rate)
qqline(df$Heart_Rate, col = 'red')
```



Since, qq-plot deviates a little from the central line, so the distribution of Heart_Rate deviates a bit from the normal distribution.

The distribution of observations is bell-shaped, so we can proceed with the linear regression

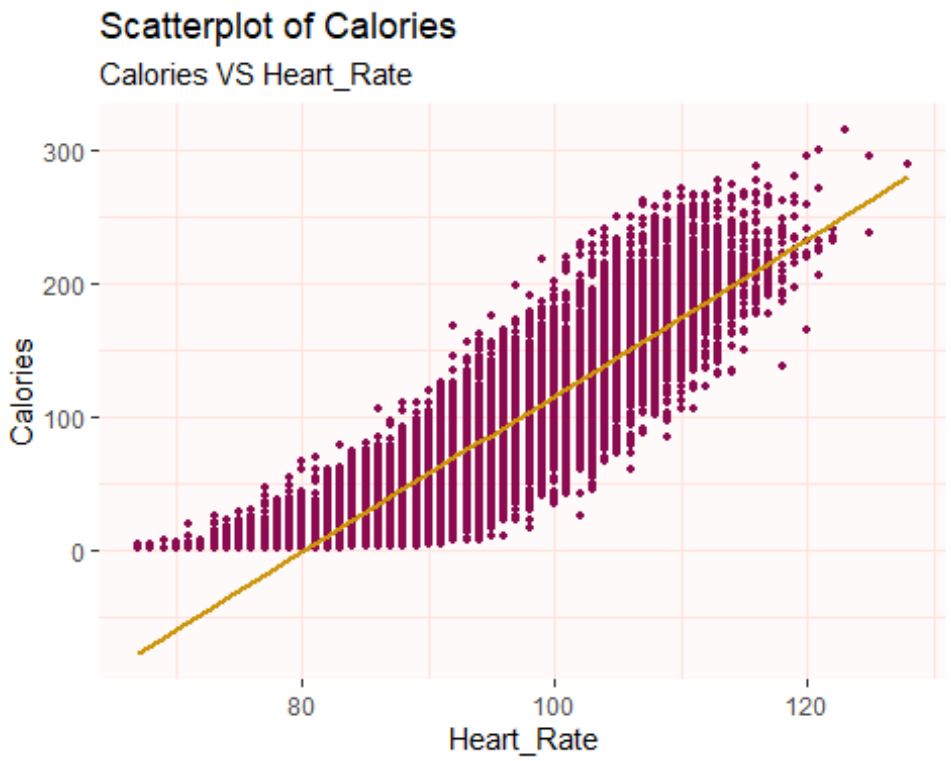
Linearity:We can check this by using scatterplot between Calories and Heart_Rate

```
#Response = Calories
# Predictor =Heart_Rate
 cor(df$Calories,df$Heart_Rate)
```

```
## [1] 0.8978821
```

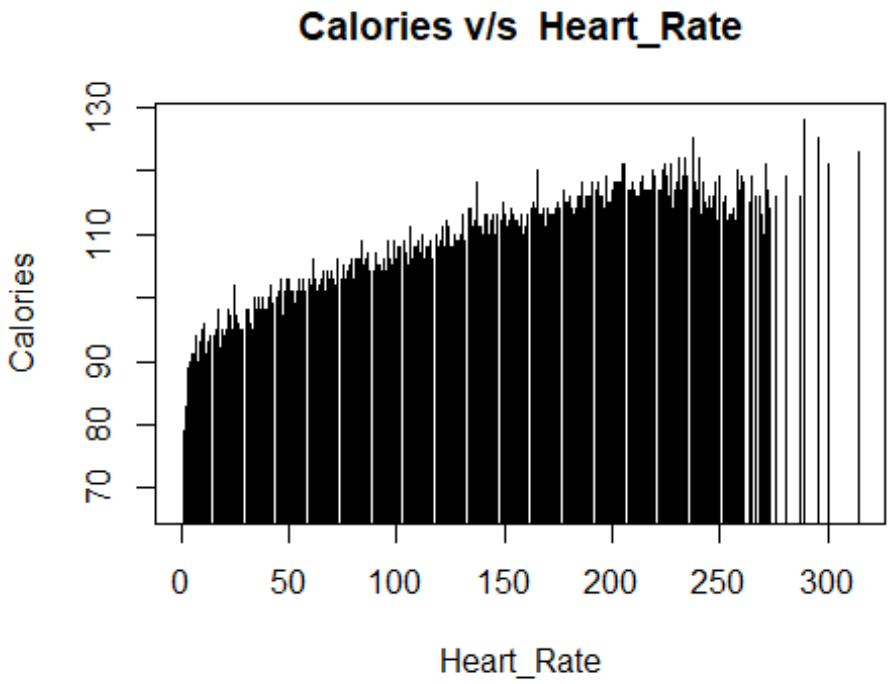
```
df %>%
  ggplot(aes(Heart_Rate, Calories)) +
  geom_point(pch = 20, col = "deeppink4", size = 2) +
  geom_smooth(method = lm, se = F, col = "darkgoldenrod3") +
  labs(title = "Scatterplot of Calories",
       subtitle = "Calories VS Heart_Rate") +
  xlab("Heart_Rate") +
  ylab("Calories") +
  theme(panel.background = element_rect(fill = 'snow1'),
        panel.grid.major = element_line(color = 'mistyrose'),
        panel.grid.minor = element_line(color = 'mistyrose'))
```

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



Comment: Here, The correlation coefficient between Calories and Heart_Rate is 0.8978821. That is, there's a strong positive correlation between these two variables. From the fitted scatterplot,the result is also verified.

```
plot(df$Calories,df$Heart_Rate,main = "Calories v/s  Heart_Rate",ylab =
"Calories",xlab = "Heart_Rate",type = "h")
```



the relationship between Calories and Heart_Rate appears linear. We can proceed with linear regression.

Fitting a linear model between Calories vs Heart_Rate$

```
model = df %>%
  lm(formula =Calories~Heart_Rate)
model %>% summary()

##
## Call:
## lm(formula = Calories ~ Heart_Rate, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.467  -18.505   -0.691   17.763  108.088
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -469.40865    2.24903  -208.7   <2e-16 ***
## Heart_Rate     5.85172    0.02343   249.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.5 on 14998 degrees of freedom
## Multiple R-squared:  0.8062, Adjusted R-squared:  0.8062
## F-statistic: 6.239e+04 on 1 and 14998 DF,  p-value: < 2.2e-16
```

The linear regression model indicates a strong and statistically significant relationship between Heart_Rate and Calories. Specifically:

Intercept: The estimated intercept is -469.41, representing the predicted Calories when Heart_Rate is zero. Interpretation of the intercept may not be meaningful in our context.

Heart_Rate: For each one-unit increase in Heart_Rate, the predicted Calories increase by approximately 5.85 units.

Statistical Significance: Both intercept and Heart_Rate coefficients are highly significant ($p < 2e-16$), providing strong evidence against the null hypothesis that their values are zero.

Model Fit: The model explains about 80.62% of the variability in Calories based on the R-squared value. The residuals have a standard deviation of 27.5, indicating the typical difference between observed and predicted values.

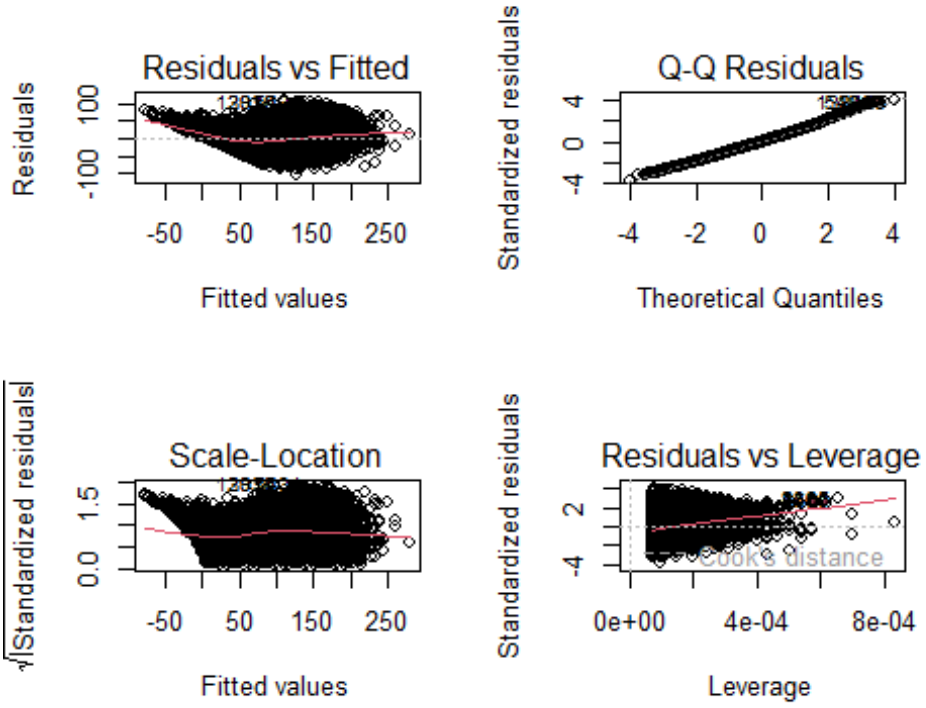
Overall Model Significance: The F-statistic is very high (6.239e+04) with a small p-value (<2.2e-16), indicating that the overall model is statistically significant.

Conclusion: The linear regression model is a robust predictor of Calories based on Heart_Rate, and the strong statistical significance supports the assertion that the relationship is meaningful in the data.

*Check for homoscedasticity:

Before proceeding with data visualization, we should make sure that our models fit the homoscedasticity assumption of the linear model. We should check that our model is actually a good fit for the data, and that we don't have large variation in the model error, by running this code:

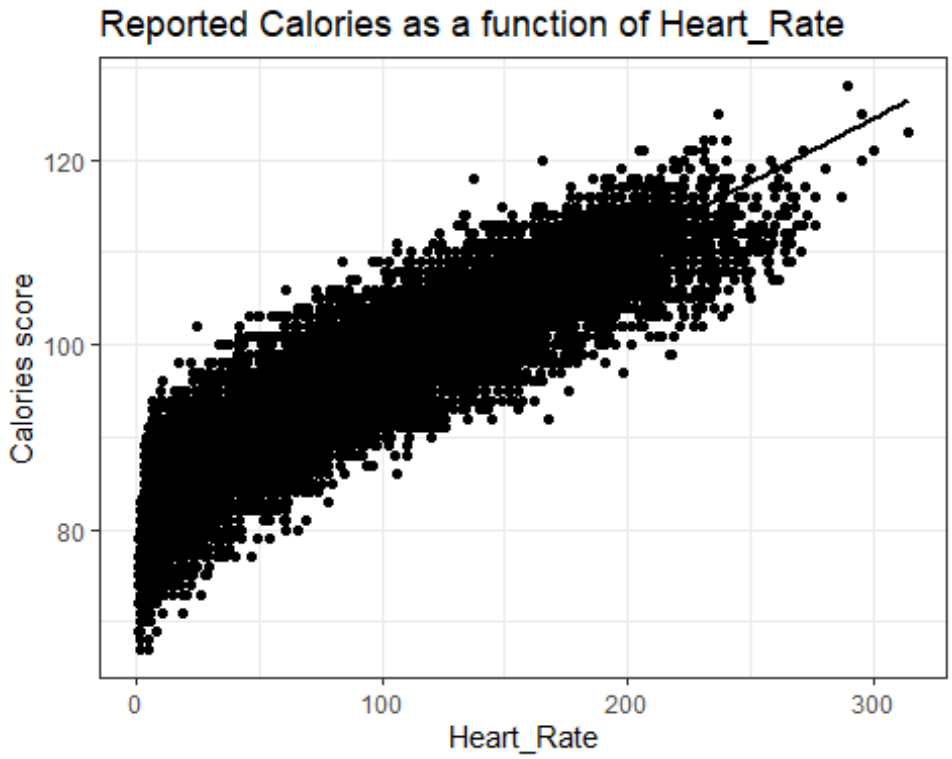
```
par(mfrow=c(2,2))
plot(lm(formula =df$Calories~df$Heart_Rate))
```



Based on these residuals, we can say that our model meets the assumption of homoscedasticity.

Visualize the result with graph:

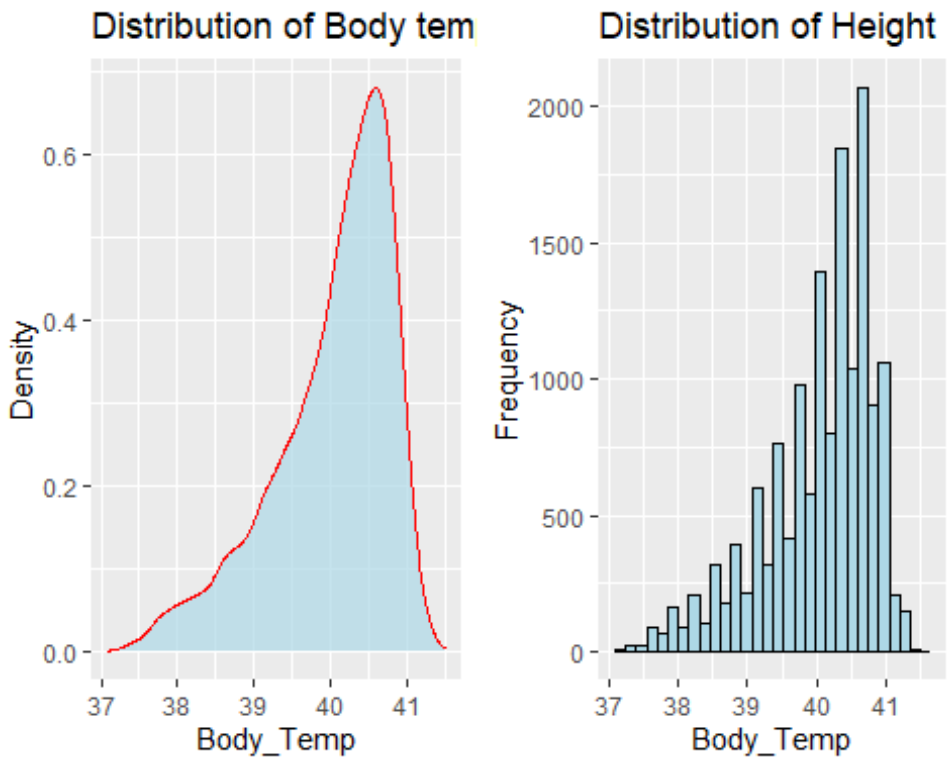
```
Calories.graph<-ggplot(df, aes(x=Calories, y=Heart_Rate))+
 geom_point()
Calories.graph <- Calories.graph + geom_smooth(method="lm",
col="black")
## `geom_smooth()` using formula = 'y ~ x'
Calories.graph +
 theme_bw() +
 labs(title = "Reported Calories as a function of Heart_Rate",
 x = "Heart_Rate",
 y = "Calories score")+
annotate(geom="text",label="Calories = -469.40865 + (5.85172*Heart_Rate)")

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



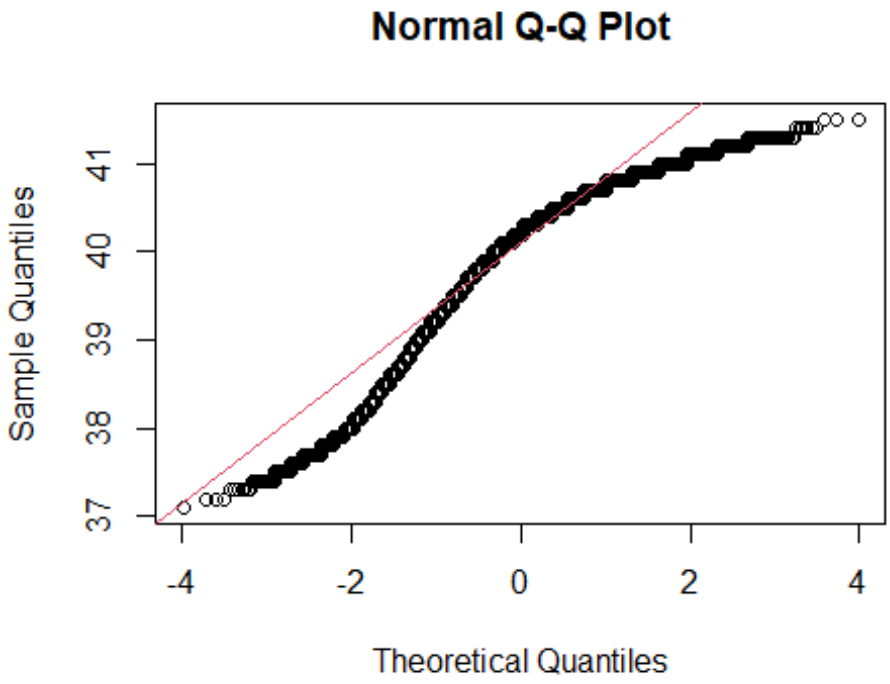
We found a significant relationship between Calories and Heart_Rate.

BODY TEMPERATURE

```r
# Set up the plot size
options(repr.plot.width=6, repr.plot.height=6)

# Load required libraries
library(ggplot2)
library(gridExtra)

# Create a density plot using ggplot2
density_plot <- ggplot(df, aes(x=Body_Temp)) +
  geom_density(fill="lightblue", color="red", alpha=0.7) +
  labs(title="Distribution of Body temperature", x="Body_Temp", y="Density")

# Create a histogram using ggplot2
hist_plot <- ggplot(df, aes(x=Body_Temp)) +
  geom_histogram(fill="lightblue", color="black", bins=30) +
  labs(title="Distribution of Height", x="Body_Temp", y="Frequency")

# Arrange the plots in a 1x2 grid
grid.arrange(density_plot, hist_plot, ncol=2)
```




```r
# Show the plots

qqnorm(df$Body_Temp)
qqline(df$Body_Temp, col = 2)
```



Since, qq-plot deviates a little from the central line, so the distribution of Body_Temp deviates a bit from the normal distribution.😕

The distribution of observations is approximately bell-shaped, so we can proceed with the linear regression

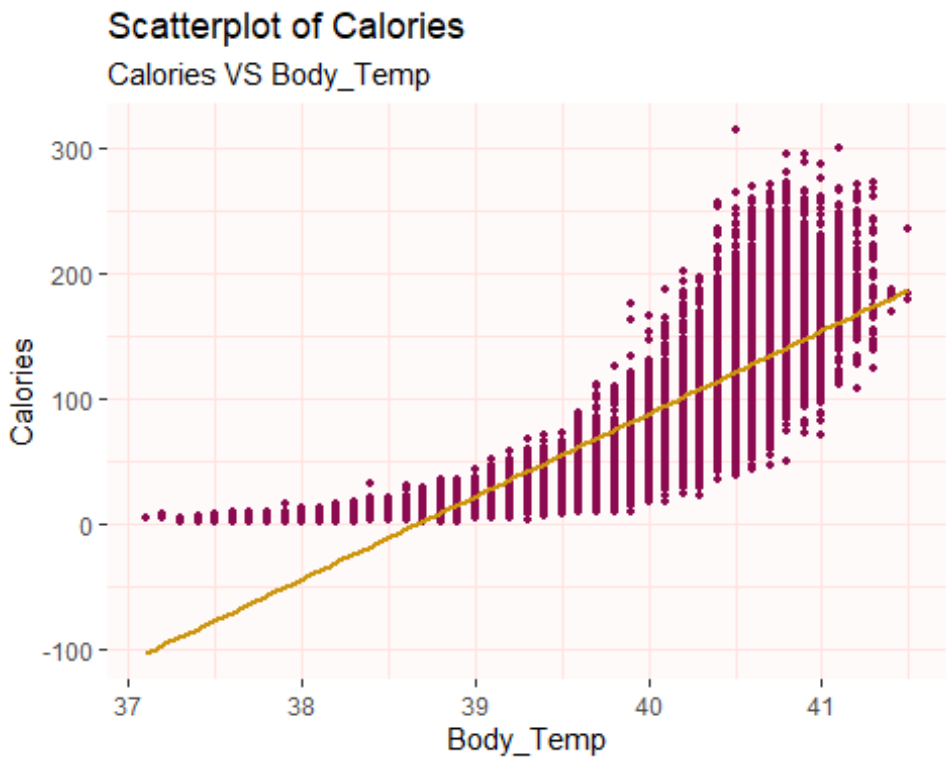
Linearity:We can check this by using scatterplot between Calories and Body_Temp

```
#Response = Calories
# Predictor = Body_Temp
 cor(df$Calories,df$Body_Temp)
```

```
## [1] 0.8245578
```

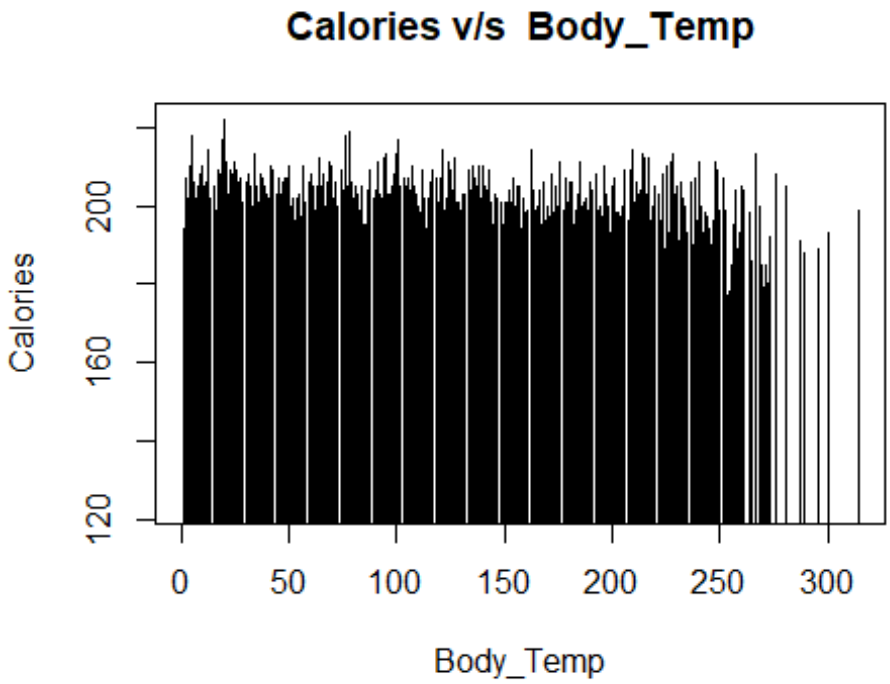
```
df %>%
  ggplot(aes(Body_Temp, Calories)) +
  geom_point(pch = 20, col = "deeppink4", size = 2) +
  geom_smooth(method = lm, se = F, col = "darkgoldenrod3") +
  labs(title = "Scatterplot of Calories",
       subtitle = "Calories VS Body_Temp") +
  xlab("Body_Temp") +
  ylab("Calories") +
  theme(panel.background = element_rect(fill = 'snow1'),
        panel.grid.major = element_line(color = 'mistyrose'),
        panel.grid.minor = element_line(color = 'mistyrose'))
```

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



Comment: Here, The correlation coefficient between Calories and Height is 0.8245578. That is, there's a strong positive correlation between these two variables. From the fitted scatterplot,the result is also verified.

```
plot(df$Calories,df$Height,main = "Calories v/s  Body_Temp",ylab =
"Calories",xlab = "Body_Temp",type = "h")
```



Although the relationship between Calories and Height is a bit less clear, it still appears linear. We can proceed with linear regression.

Fitting a linear model between Calories vs Body_Temp$

```
model = df %>%
  lm(formula =Calories~Body_Temp)
model %>% summary()

##
## Call:
## lm(formula = Calories ~ Body_Temp, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.729 -25.639  -6.729  21.271 193.098
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2555.7468    14.8239  -172.4   <2e-16 ***
## Body_Temp      66.0901     0.3703   178.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.34 on 14998 degrees of freedom
## Multiple R-squared:  0.6799, Adjusted R-squared:  0.6799
## F-statistic: 3.186e+04 on 1 and 14998 DF,  p-value: < 2.2e-16
```

The linear regression model indicates a highly significant and strong relationship between Body_Temp and Calories:

Intercept: Predicted Calories when Body_Temp is zero is -2555.75, but interpret cautiously.

Body_Temp (Slope): For each one-unit increase in Body_Temp, predicted Calories increase by approximately 66.09 units.

Statistical Significance: Both intercept and Body_Temp are highly significant ($p < 2e-16$).

Model Fit: The model explains about 67.99% of the variability in Calories based on the high R-squared value.

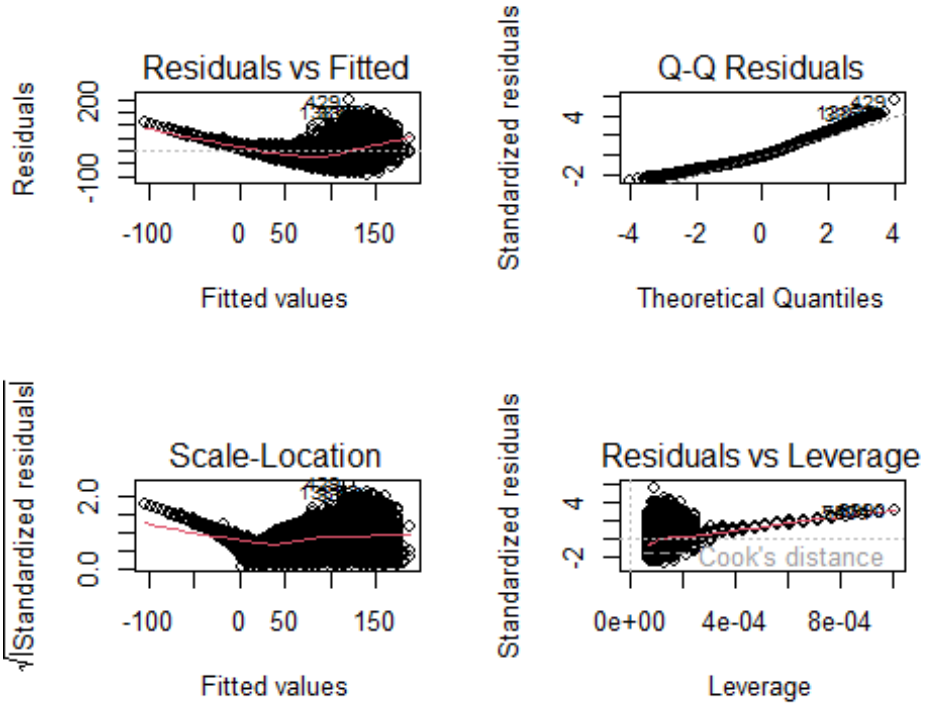
Overall Model Significance: The overall model is statistically significant, suggesting its usefulness in predicting Calories based on Body_Temp.

The linear regression model is highly significant, suggesting a strong and meaningful relationship between Body_Temp and Calories. The high R-squared value indicates that a substantial portion of the variability in Calories is explained by the linear relationship with Body_Temp. This model can be used to predict Calories based on body temperature.

Check for homoscedasticity:

Before proceeding with data visualization, we should make sure that our models fit the homoscedasticity assumption of the linear model. We should check that our model is actually a good fit for the data, and that we don't have large variation in the model error, by running this code:

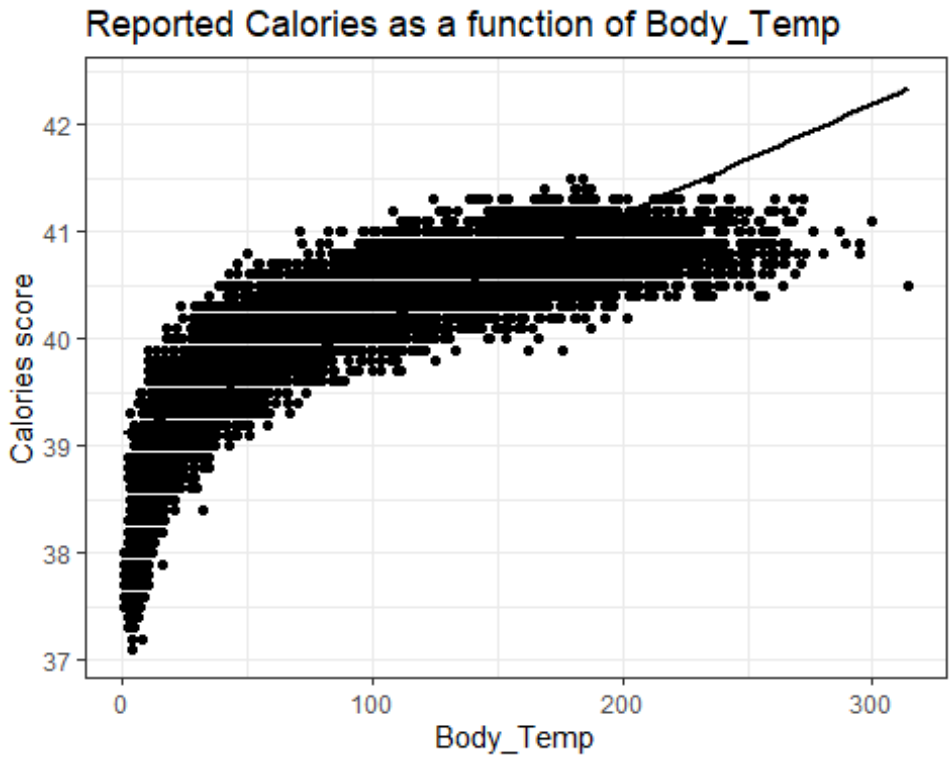
```
par(mfrow=c(2,2))
plot(lm(formula =df$Calories~df$Body_Temp))
```



Based on these residuals, we can say that our model meets the assumption of homoscedasticity.

Visualize the result with graph:

```
Calories.graph<-ggplot(df, aes(x=Calories, y=Body_Temp))+
 geom_point()
Calories.graph <- Calories.graph + geom_smooth(method="lm",
col="black")
## `geom_smooth()` using formula = 'y ~ x'
Calories.graph +
 theme_bw() +
 labs(title = "Reported Calories as a function of Body_Temp",
 x = "Body_Temp",
 y = "Calories score")+
annotate(geom="text",label="Calories = -2555.7468 + (66.0901*Body_Temp)")

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



We found a significant relationship between Calories and Body temperature, with a 66.0901-unit increase in reported Calories for every 1 unit increase in Body temperature.

CONCLUSION

Practical Recommendations:

For individuals aiming to maximize calorie burn, focusing on activities with longer durations, increased heart rates, and elevated body temperatures could prove beneficial