```
In [70]:  #checking the outlier location before delete
          indices <- which(data$age == OutAge)

          # Print the resulting row indices
          print(indices)
```

```
[1] 1615 3295
```

Figure 1

After detecting the outliers from the Age attribute, we took the necessary steps to verify the location and impact of these outliers before proceeding with their deletion. This approach allowed us to observe the dataset both before and after removing the outliers.

```
In [499]:  #Number of rows
           nrow(data)
           #Number of column
           ncol(data)
```

Before

```
5110
12
```

Figure 2

```
In [33]:  #check after deleting
          #Number of rows
          nrow(data)
          #Number of column
          ncol(data)
```

After

```
5105
12
```

Figure 3

As a double check, we counted the rows before and after deletion and compared them to each other and confirmed that the deletion process is completed, as five rows were deleted according to the outlier of different attributes (Age, Glucose level, BMI).

```
In [49]: print(data[, c("age", "avg_glucose_level","bmi")])
```

```
1605 47.00              65.04 30.9
1606 35.00             151.25 28.4
1607 51.00             106.41 41.9
1608 60.00             197.09 34.3
1609 59.00              93.58 25.1
1610  1.24             122.04 10.3
1611 18.00              80.06 31.8
1612 81.00              84.93 31.8
1613 15.00              68.40   23
1614 73.00              62.99 25.4
1615  0.08             139.67 14.1
1616 53.00             113.40 35.1
1617 45.00             101.92 26.9
1618 70.00              65.98   33
1619 56.00              84.30 22.1
1620  7.00              61.42 20.8
1621 66.00              85.52   30
1622 53.00              83.79   44
1623 20.00              73.83 16.6
```

Before

Figure 4

```
In [77]: # outliers row is removed now
print(data[, c("age", "avg_glucose_level","bmi")])
```

```
1604 45.00             146.44 22.80000
1605 47.00              65.04 30.90000
1606 35.00             151.25 28.40000
1607 51.00             106.41 41.90000
1608 60.00             197.09 34.30000
1609 59.00              93.58 25.10000
1610  1.24             122.04 10.30000
1611 18.00              80.06 31.80000
1612 81.00              84.93 31.80000
1613 15.00              68.40 23.00000
1614 73.00              62.99 25.40000
1616 53.00             113.40 35.10000
1617 45.00             101.92 26.90000
1618 70.00              65.98 33.00000
1619 56.00              84.30 22.10000
1620  7.00              61.42 20.80000
1621 66.00              85.52 30.00000
1622 53.00              83.79 44.00000
1623 20.00              73.83 16.60000
1624 15.00              69.38 28.40000
```

After

Figure 5

We also printed the dataset before and after and compared the values. We noticed that the patient row whose age was 0.8 had disappeared, which indicates the success of the deletion.

In [78]:
```
indices <- which(data$age == OutAge)

# Print the resulting row indices
print(indices)
```

```
integer(0)
```

In [79]:
```
indices3 <- which(data$avg_glucose_level == OutAvg)

# Print the resulting row indices
print(indices3)
```

```
integer(0)
```

In [80]:
```
indices2 <- which(data$bmi == 97.6)

# Print the resulting row indices
print(indices2)
```

```
integer(0)
```

Figure 6

To make sure that the deletion was successful, we searched for the rows that contain the Outlier values, and the results were all zero, which confirms to us that the deletion was successful.

5- Normalize Data using Min-Max Scaling:

In [61]:
```
normalize <- function(x)
{
 return ((x - min(x))/ (max(x)- min(x)))
}

data$avg_glucose_level= normalize(data$avg_glucose_level)
data$age= normalize(data$age)
data$bmi= normalize(data$age)
head(data)
```

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	1	0.8167155	0	1	1	4	1	0.8162622	0.8167155	3	1
51676	2	0.7434018	0	0	1	3	2	0.6917325	0.7434018	2	1
31112	1	0.9755621	0	1	1	4	2	0.2389014	0.9755621	2	1
60182	2	0.5967742	0	0	1	4	1	0.5460403	0.5967742	4	1
1665	2	0.9633431	1	0	1	3	2	0.5596313	0.9633431	2	1
56669	1	0.9877810	0	0	1	4	1	0.6164880	0.9877810	3	1

Figure 7

a normalization step was performed to ensure consistent scaling of the data. The normalization technique applied was the max-min normalization. This technique rescales the values of specific attributes within a defined range between 0 and 1.

The following attributes were selected for normalization: age, average glucose level, and BMI (Body Mass Index). We can use the normalized dataset provides a more uniform and comparable representation of the attributes, enabling accurate analysis and modeling for stroke prediction with result as shown.