Health Care Dataset

Ghazal Kalhor

***Abstract* — In this computer assignment, we want to perform statistical analysis on healthCare dataset. We will use methods that we learnt in Statistical Inference. Also, we will use R programming language to reach this goal. *Keywords* — Statistical Inference, R**

Introduction

The aim of this computer assignment is to perform analysis tasks on different columns of dataset too get a good view of it that will be valuable for next phase of this project.

Importing Libraries ¶

In this part, we will import some of the necessary libraries in order to use their helpful functions. Firstly, we will install related packages. Secondly, we will use `library()` function to import them.

In [ ]:

```
install.packages("plyr")
install.packages("e1071")
install.packages("psych")
install.packages("dplyr")
install.packages("hexbin")
install.packages("ggExtra")
install.packages("GGally")
install.packages("ggcorrplot")
install.packages("scatterplot3d")
```

In [ ]:

```
library(ggplot2)
library(plyr)
library(e1071)
library(psych)
library(dplyr)
library(hexbin)
library(ggExtra)
library(GGally)
library(ggcorrplot)
library(scatterplot3d)
```

Importing Data

In this part, file *HealthCare.csv* is coppied to the project directory, then we read and store it in a dataframe called *heathCare*.

In [3]:

```
healthCare <- read.csv("/content/HealthCare.csv")
```

`describe()` method from *psych* library is used in order to view some basic statistical details like max, min, median, mean, sd etc. of the dataframe.

In [4]:

```
describe(healthCare)
```

A psych: 13 × 13

	vars	n	mean	sd	median	trimmed	
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
id	1	5110	3.651783e+04	2.116172e+04	36932.000	36542.259051	2741
gender*	2	5110	1.414286e+00	4.930436e-01	1.000	1.392613	0
age	3	5110	4.322661e+01	2.261265e+01	45.000	43.607877	26
hypertension	4	5110	9.745597e-02	2.966067e-01	0.000	0.000000	0
heart_disease	5	5110	5.401174e-02	2.260630e-01	0.000	0.000000	0
ever_married*	6	5110	1.656164e+00	4.750335e-01	2.000	1.695205	0
work_type*	7	5110	3.495499e+00	1.278532e+00	4.000	3.619374	0
Residence_type*	8	5110	1.508023e+00	4.999845e-01	2.000	1.510029	0
avg_glucose_level	9	5110	1.061477e+02	4.528356e+01	91.885	97.846204	26
bmi	10	4909	2.889324e+01	7.854067e+00	28.100	28.342708	6
smoking_status*	11	5110	2.585519e+00	1.092522e+00	2.000	2.606898	1
stroke	12	5110	4.872798e-02	2.153199e-01	0.000	0.000000	0
health_bills	13	4909	3.138585e+03	8.247692e+02	3031.724	3052.745634	627

`summary()` method concise summary of dataset. It prints information about the dataframe such as min, max, quartiles, and mean.

In [5]:

```
summary(healthCare)
```

```
       id            gender              age          hypertension
 Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
 1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
 Median :36932   Mode  :character   Median :45.00   Median :0.00000
 Mean   :36518                      Mean   :43.23   Mean   :0.09746
 3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000
 Max.   :72940                      Max.   :82.00   Max.   :1.00000

 heart_disease     ever_married        work_type         Residence_t
ype
 Min.   :0.00000   Length:5110        Length:5110        Length:5110
 1st Qu.:0.00000   Class :character   Class :character   Class :char
acter
 Median :0.00000   Mode  :character   Mode  :character   Mode  :char
acter
 Mean   :0.05401
 3rd Qu.:0.00000
 Max.   :1.00000

 avg_glucose_level      bmi       smoking_status          stroke
 Min.   : 55.12    Min.   :10.30   Length:5110        Min.   :0.0000
0
 1st Qu.: 77.25    1st Qu.:23.50   Class :character   1st Qu.:0.0000
0
 Median : 91.89    Median :28.10   Mode  :character   Median :0.0000
0
 Mean   :106.15    Mean   :28.89                      Mean   :0.0487
3
 3rd Qu.:114.09    3rd Qu.:33.10                      3rd Qu.:0.0000
0
 Max.   :271.74    Max.   :97.60                      Max.   :1.0000
0
                   NA's   :201
  health_bills
 Min.   :  44.8
 1st Qu.:2628.8
 Median :3031.7
 Mean   :3138.6
 3rd Qu.:3474.4
 Max.   :9100.5
 NA's   :201
```

Cleaning Data

In this part, we will convert the column values that are in a wrong format to an appropriate format. Values in **hypertension**, **heart_disease**, and **stroke** are stored as Integer but they categorical variables and it would be better to store them as String. This can be done by using `mapvalues()` method from *plyr* library.

In [6]:

```
healthCare$hypertension <- mapvalues(healthCare$hypertension,
                                     from = c(0, 1),
                                     to = c("No", "Yes"))
```

In [7]:

```
healthCare$heart_disease <- mapvalues(healthCare$heart_disease,
                                      from = c(0, 1),
                                      to = c("No", "Yes"))
```

In [8]:

```
healthCare$stroke <- mapvalues(healthCare$stroke,
                                      from = c(0, 1),
                                      to = c("No", "Yes"))
```

Moreover, one task is left that we shoud do in cleaning data. For gender column there is a class called **Other**. At first, we have to count rows with this value.

In [9]:

```
healthCare[(healthCare$gender=="Other"),]
```

A data.frame: 1 × 13

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residen
	<int>	**<chr>**	**<dbl>**	**<chr>**	**<chr>**	**<chr>**	**<chr>**	
3117	56156	Other	26	No	No	No	Private	

Based on the result, there is only one row with this value. So, we can drop it because it is not informative.

In [10]:

```
healthCare <- healthCare[!(healthCare$gender=="Other"),]
```

Question 0

Part A

It is undeniable that we should pay attention to our health situation in each. Moreover, as we get older, we will face many health problems and without a good plan, we won't be able to overcome the expences of different treatments. So, we have to be familiar with important factors that can cause health issues for us. Maybe by having this knowledge we would be more careful about our habits to be in a better health situation.

This dataset provides some of the factors that we mentioned above. So, it be valuable to analyze it.

Part B

In this part we will take a look on the columns of our dataset to get a good sight about it. We have 13 features and 5110 observations in our dataset that we will discuss each of them in this part.

1. **id**

   It is a unique number that is assigned to the person in our dataset.

1. **gender**

   It is the person gender.

   - Female
   - Male

1. **age**

   It is the person age in years.

1. **hypertension**

   It is the whether person hypertension. It is also called high blood pressure that means the blood pressure is higher than normal.

   - Yes
   - No

1. **heart disease**

   It determines whether the person suffers from any heart disease or not. Heart disease refers to any condition affecting the heart.

   - Yes
   - No

1. **ever marrid**

   It determines whether the person has ever marrid or not.

   - Yes
   - No

1. **work type**

   It defines the type of work a person does.

   - Private
   - Self-employed
   - Govt_job
   - children
   - Never_worked

1. **residence type**

   It defines the type of residece a person lives in.

   - Urban

- Rural

1. **avg glucose level**

   It is the person average glucose level. Average glucose level is the measure of concentration of glucose present in the blood of humans. Its unit is mg/dL.

1. **bmi**

   It is the person body mass index which can be computed by dividing the body mass by the square of the body height. Its unit is kg/m2.

1. **smoking status**

   It is the person smoking status.

   - formerly smoked
   - never smoked
   - smokes
   - Unknown

1. **stoke**

   It determines whether the person is suffered from stroke or not. A stroke is a serious life-threatening medical condition that happens when the blood supply to part of the brain is cut off.

   - Yes
   - No

1. **health bills**

   It is the amount of money that the person spend yearly on her/his health in $.

In [11]:

```
str(healthCare)
```

```
'data.frame':	5109 obs. of  13 variables:
 $ id               : int  9046 51676 31112 60182 1665 56669 53882 1
0434 27419 60491 ...
 $ gender           : chr  "Male" "Female" "Male" "Female" ...
 $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
 $ hypertension     : chr  "No" "No" "No" "No" ...
 $ heart_disease    : chr  "Yes" "No" "Yes" "No" ...
 $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ work_type        : chr  "Private" "Self-employed" "Private" "Priv
ate" ...
 $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
 $ avg_glucose_level: num  229 202 106 171 174 ...
 $ bmi              : num  36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2
...
 $ smoking_status   : chr  "formerly smoked" "never smoked" "never s
moked" "smokes" ...
 $ stroke           : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ health_bills     : num  6012 NA 6385 5863 5461 ...
```

Based on the result, we are interested in health_bills variable and it would be valuable to predict it using other features.

Part C

In this part, we will use a combination of `colMeans()` and `is.na()` methods in order to compute proportion of nan values in each column.

In [12]:

```
colMeans(is.na(healthCare))
```

id: 0 **gender:** 0 **age:** 0 **hypertension:** 0 **heart_disease:** 0 **ever_married:** 0 **work_type:** 0 **Residence_type:** 0 **avg_glucose_level:** 0 **bmi:** 0.0393423370522607 **smoking_status:** 0 **stroke:** 0 **health_bills:** 0.0393423370522607

Based on the result, **bmi** and **health_bills** have about 4% nan values.

One of the methods to deal with these value is to replace them with a statistic of that column.

For **bmi** we will use mean to replace missing values, because it has an aproximately normal distribution as its median and mean are close to each other.

In [13]:

```
healthCare[c("bmi")][is.na(healthCare[c("bmi")])] <- mean(healthCare$bmi, na.rm
=TRUE)
```

For **health_bills** it seems that median can be a better statistic to replace nan values with.

In [14]:

```
healthCare[c("health_bills")][is.na(healthCare[
  c("health_bills")])] <- median(healthCare$health_bills, na.rm=TRUE)
```

Part D

It would be difficult to determine important features without performing any analysis on dataset. However, to my mind, **smoking_status** and **bmi** can be the two most important features in our dataset.

In [15]:

```
summary(healthCare)
```

```
       id            gender               age         hypertension
 Min.   :   67   Length:5109        Min.   : 0.08   Length:5109
 1st Qu.:17740   Class :character   1st Qu.:25.00   Class :character
 Median :36922   Mode  :character   Median :45.00   Mode  :character
 Mean   :36514                      Mean   :43.23
 3rd Qu.:54643                      3rd Qu.:61.00
 Max.   :72940                      Max.   :82.00
 heart_disease      ever_married        work_type          Residence_
type
 Length:5109        Length:5109        Length:5109        Length:510
9
 Class :character   Class :character   Class :character   Class :cha
racter
 Mode  :character   Mode  :character   Mode  :character   Mode  :cha
racter



 avg_glucose_level      bmi         smoking_status        stroke
 Min.   : 55.12    Min.   :10.30   Length:5109        Length:5109
 1st Qu.: 77.24    1st Qu.:23.80   Class :character   Class :charact
er
 Median : 91.88    Median :28.40   Mode  :character   Mode  :charact
er
 Mean   :106.14    Mean   :28.89
 3rd Qu.:114.09    3rd Qu.:32.80
 Max.   :271.74    Max.   :97.60
  health_bills
 Min.   :  44.8
 1st Qu.:2647.8
 Median :3032.2
 Mean   :3134.6
 3rd Qu.:3454.9
 Max.   :9100.5
```

Question 1

In this question we choose **BMI** feature as a numerical variable to perform following task with. Because it is one of the important factors that should be considered to predict health bills.

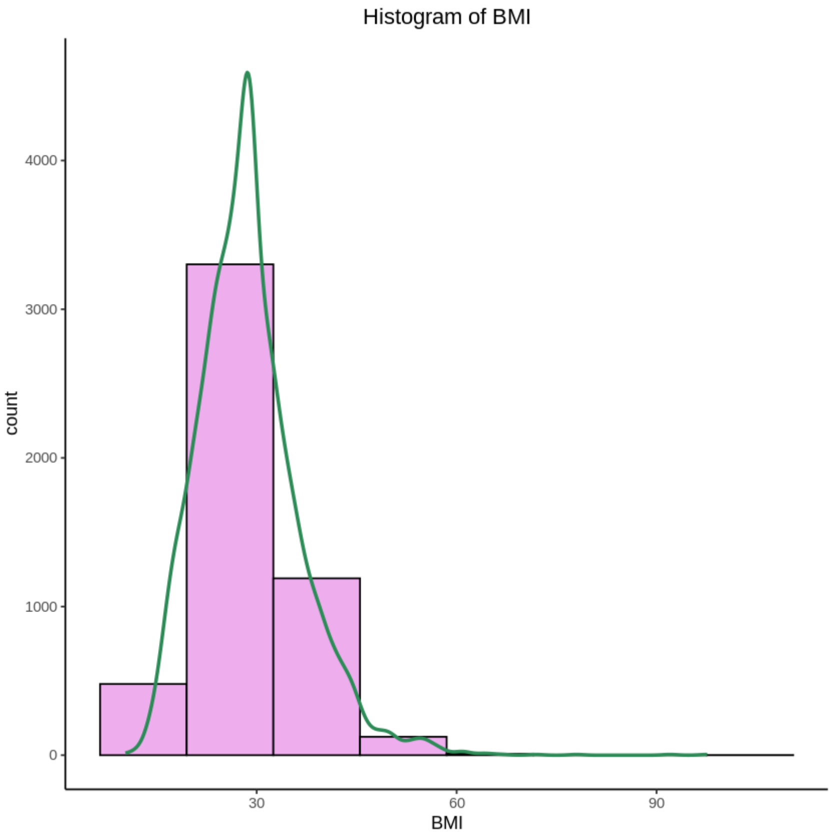
Part A

In this part,we will draw histogram along with density curve. For choosing bin with, we use square-root of n (number of observation) which we learnt in class.

Based on the result, the distribution of **bmi** is unimodal. In other words, it has one maximum.

In [16]:

```
bmi <- healthCare$bmi
n <- 1000
binwidth <- ceiling(log2(length(bmi)))
bmiDensHist <- ggplot(healthCare, aes(x=bmi))
bmiDensHist <- bmiDensHist + geom_histogram(aes(y=..count..), fill ="plum2", col
our="black",
                                                                          binwid
th = binwidth)
bmiDensHist <- bmiDensHist + geom_line(aes(y = ..density.. * n * binwidth), colo
r = "seagreen4",
                                                                   size = 1, stat
= 'density')
bmiDensHist <- bmiDensHist + xlab("BMI")
bmiDensHist <- bmiDensHist + ylab("count")
bmiDensHist <- bmiDensHist + ggtitle("Histogram of BMI")
bmiDensHist <- bmiDensHist + theme_classic()
bmiDensHist <- bmiDensHist + theme(plot.title = element_text(hjust = 0.5))
bmiDensHist
```

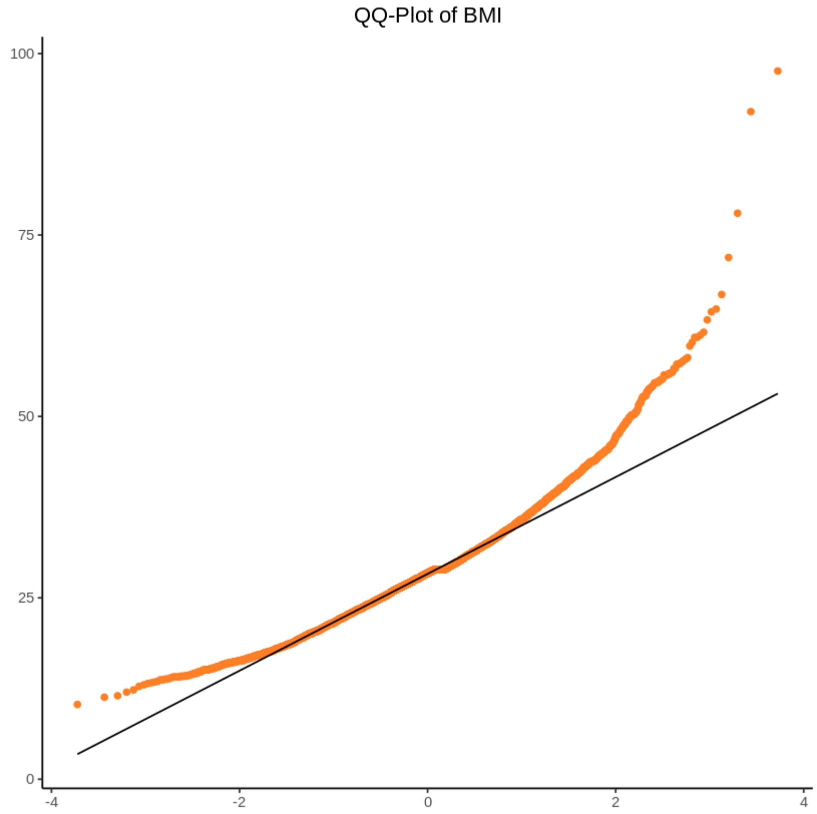


Part B

In this part, we use QQ-Plot in order to compare the distribution of this variable with normal distribution. As we can see, points bend up and to the left of the line. So, it is right-skewed.

In [17]:

```
bmiQQ <- ggplot(healthCare, aes(sample=bmi))
bmiQQ <- bmiQQ + geom_qq(col= "chocolate1")
bmiQQ <- bmiQQ + ggtitle("QQ-Plot of BMI")
bmiQQ <- bmiQQ + xlab("") + ylab("")
bmiQQ <- bmiQQ + theme_classic()
bmiQQ <- bmiQQ + theme(plot.title = element_text(hjust = 0.5))
bmiQQ <- bmiQQ + geom_qq_line(geom = "path", position = "identity")
bmiQQ
```



Part C

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.

When it is negative, it shows that the mean of data is less than the median and therefore we the distribution is left-skewed. Conversely if it is positive, it shows that the distribuion of data is right-skewed. Finally if it is zero, it means that the distribuion is symmetric.

In this part we will use `skewness()` method from *e1071* library to compute the skewness of **bmi**.

In [18]:

```
skewness(bmi)
```

```
1.07580550700259
```

Based on the result, the skewness of this vaariable is positive. So, the distribution of this variable is right-skewed. We guessed this point in previous parts and now by computing we are sure about it.

Part D

In this part, we will use the combination of `str()` and `boxplot.stats()` to get parameters of box plot such as whiskers, quartiles, outliers and etc. We should note that the first and last element of stats are whiskers and all the data points that are out of this range consider as outliers.

In [19]:

```
str(boxplot.stats(bmi))
```

```
List of 4
 $ stats: num [1:5] 11.3 23.8 28.4 32.8 46.2
 $ n    : int 5109
 $ conf : num [1:2] 28.2 28.6
 $ out  : num [1:126] 48.9 47.5 56.6 50.1 54.6 60.9 54.7 48.2 64.8 4
7.3 ...
```

Based on the result, we can see that **bmi** has 126 outliers.

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. These values can give us interesting information about data.

In this part, we will print all the outliers baseed on whiskers.

In [20]:

```
outliers <- c(bmi[bmi < 11.3], bmi[bmi > 46.2])
outliers
```

```
10.3 · 48.9 · 47.5 · 56.6 · 50.1 · 54.6 · 60.9 · 54.7 · 48.2 · 64.8 · 47.3 ·
46.5 · 46.6 · 54.7 · 49.8 · 60.2 · 51 · 51.5 · 71.9 · 50.2 · 47.8 · 54.6 ·
55.7 · 55.7 · 57.5 · 54.2 · 52.3 · 50.3 · 78 · 50.2 · 53.4 · 55.2 · 48.4 ·
50.6 · 49.5 · 55 · 54.8 · 50.2 · 47.5 · 52.8 · 66.8 · 55.1 · 48.5 · 55.9 ·
57.3 · 49.8 · 56 · 51.8 · 57.7 · 48.9 · 49.3 · 49.8 · 54 · 56.1 · 97.6 · 53.9 ·
49.4 · 48.5 · 49.2 · 48.7 · 48.9 · 53.8 · 46.5 · 48.8 · 52.7 · 52.8 · 55.7 ·
53.5 · 50.5 · 51.9 · 63.3 · 52.8 · 61.2 · 48 · 46.8 · 50.1 · 48.3 · 58.1 ·
49.3 · 50.4 · 52.7 · 48.3 · 49.3 · 51.9 · 53.4 · 50.3 · 59.7 · 47.4 · 52.5 ·
52.9 · 54.7 · 61.6 · 49.9 · 53.8 · 47.3 · 54.3 · 47.9 · 55 · 50.9 · 50.6 ·
57.2 · 64.4 · 92 · 50.8 · 55.9 · 57.9 · 47.6 · 55.7 · 48.8 · 57.2 · 47.5 ·
46.4 · 46.9 · 50.2 · 47.1 · 48.1 · 51.7 · 60.9 · 47.8 · 47.6 · 46.3 · 54.1 ·
56.6 · 49.5 · 47.6 · 46.9
```

This amount of outliers shows that there is a posibility of error in calculation of BMI that might be made by individuals if they reported it themselves. Moreover if we just asked the weight and height of people, it would be possible that they didn't report the correct values.

Part E

We use `mean()` function to compute the mean of bmi.

In [21]:

```
mean(bmi)
```

```
28.8945599022005
```

We use `median()` function to compute the median of bmi. It is the midpoint of the distribution. In other words, 50% of the individuals in our sample have bmi less than this value and 50% of them have bmi greater than it.

In [22]:

```
median(bmi)
```

```
28.4
```

We use `var()` function to compute the variance of bmi. It is roughly the average squared deviation from the mean. It indicates the degree of spread in the dataset. The more spread the data, the larger the variance is in relation to the mean. It shows that how much the individuals bmi are away from the mean bmi.

In [23]:

```
var(bmi)
```

```
59.2628239525037
```

We use `sd()` function to compute the standard deviation of bmi. It is roughly the average deviation from the mean that has the same units as the data. It provides an indication of how far the individuals bmi deviate from the mean bmi.

In [24]:

```
sd(bmi)
```

```
7.69823511933116
```

Part F

In this part, we cwill plot density plot of bmi distribution.

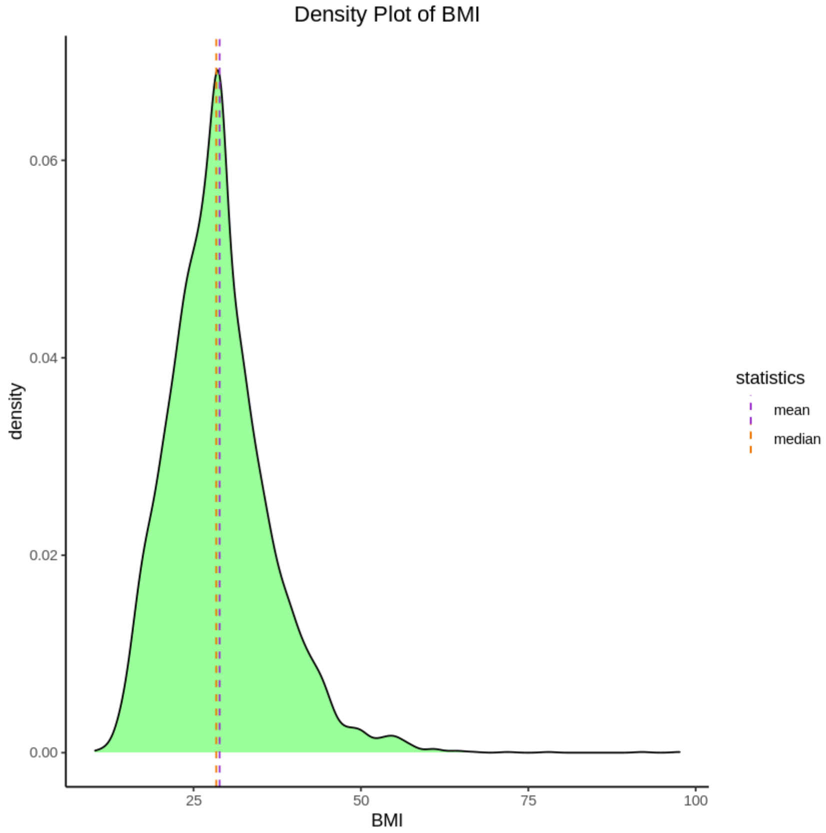
Median of a density curve is the equal-area point, the point with half the area under the curve to its left and the other half to its right.

Mean is the point at which the curve would balance if made of solid material.

In [25]:

```
bmiStatistics <- healthCare %>% summarize(mean = mean(bmi), median = median(bmi))

bmiDensity <- ggplot(healthCare, aes(x=bmi))
bmiDensity <- bmiDensity + geom_density(color="black", fill="palegreen1")
bmiDensity <- bmiDensity + geom_vline(data = bmiStatistics, aes(xintercept = mean, color= "mean"),
                                        linetype = "dashed")
bmiDensity <- bmiDensity + geom_vline(data = bmiStatistics, aes(xintercept = median, color= "median"),
                                        linetype = "dashed")
bmiDensity <- bmiDensity + scale_color_manual(name = "statistics",
                  values = c(mean = "darkorchid", median = "darkorange2"))
bmiDensity <- bmiDensity + xlab("BMI")
bmiDensity <- bmiDensity + ggtitle("Density Plot of BMI")
bmiDensity <- bmiDensity + theme_classic()
bmiDensity <- bmiDensity + theme(plot.title = element_text(hjust = 0.5))
bmiDensity
```



Part G

In this part, we want to categorize the **BMI** values into 4 classes as follows:

- Underweight:

  ```
  bmi <= 18.5
  ```

- Normal weight:

  ```
  bmi > 18.5 and bmi < 25
  ```

- Overweight:

  ```
  bmi >= 25 & bmi < 30
  ```

- Obesity:

  ```
  bmi > 30
  ```

Then, we will compute the percentage of each class and add it to labels using `paste0()` method.

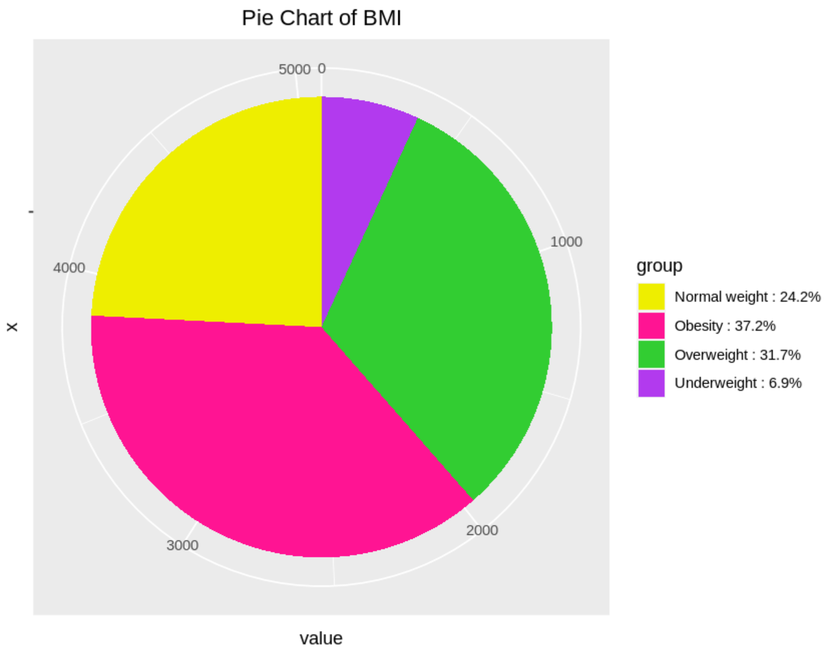
In [26]:

```
bmiGroups <- c(length(bmi[bmi <= 18.5]),
               length(bmi[bmi > 18.5 & bmi < 25]),
               length(bmi[bmi >= 25 & bmi < 30]),
               length(bmi[bmi > 30]))

bmiPercents <- round(100 * bmiGroups / sum(bmiGroups), 1)
bmiLabels = c("Underweight",
              "Normal weight",
              "Overweight",
              "Obesity")

data <- data.frame(group=paste0(bmiLabels, " : ", bmiPercents, "%"), value=bmiGr
oups)

bmiPie <- ggplot(data, aes(x="", y=value, fill=group))
bmiPie <- bmiPie + scale_fill_manual(values=c("yellow2", "deeppink", "limegreen"
, "darkorchid2"))
bmiPie <- bmiPie + geom_bar(stat="identity", width=1)
bmiPie <- bmiPie + coord_polar("y", start=0)
bmiPie <- bmiPie + ggtitle("Pie Chart of BMI")
bmiPie <- bmiPie + theme(plot.title = element_text(hjust = 0.5))
bmiPie
```

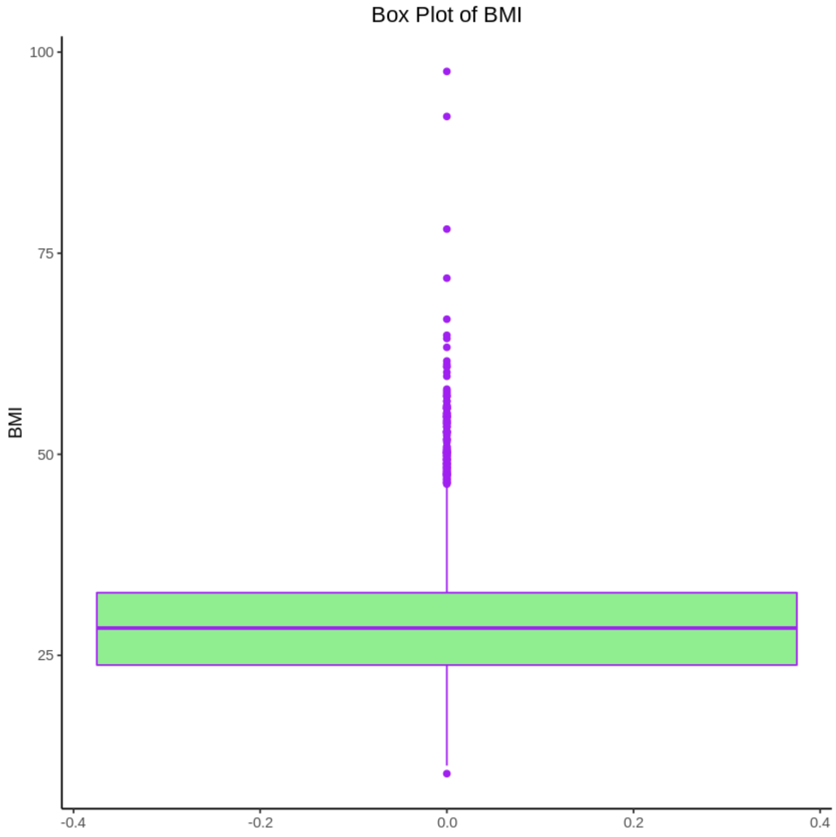


Part H

In this part, we will plot a box plot for **BMI** to find its parameters such as IQR, whiskers, and quartiles.

In [27]:

```
bmiBox <- ggplot(healthCare, aes(x = bmi))
bmiBox <- bmiBox + geom_boxplot(col="purple", fill="palegreen2")
bmiBox <- bmiBox + coord_flip()
bmiBox <- bmiBox + labs(x="BMI")
bmiBox <- bmiBox + ggtitle("Box Plot of BMI")
bmiBox <- bmiBox + theme_classic()
bmiBox <- bmiBox + theme(plot.title = element_text(hjust = 0.5))
bmiBox
```



By using a combination of `str()` and `boxplot.stats()` methods we can get the parameters of its boxplot.

In [28]:

```
str(boxplot.stats(bmi))
```

```
List of 4
 $ stats: num [1:5] 11.3 23.8 28.4 32.8 46.2
 $ n    : int 5109
 $ conf : num [1:2] 28.2 28.6
 $ out  : num [1:126] 48.9 47.5 56.6 50.1 54.6 60.9 54.7 48.2 64.8 4
7.3 ...
```

Based on the result, we can obtain these parameters:

- lower whisker = 11.3
- upper whisker = 46.2
- median = 28.4
- 1st-quartile = 23.8
- 3rd-quartile = 32.8

The difference between upper and lower whiskers shows the IQR of this boxplot. Also, we can compute it using `IQR()` method.

In [29]:

```
IQR(bmi)
```

9

Question 2

In this question we choose **Smoking Status** feature as a categorical variable to perform following task with. Because it is one of the important factors that affects our health situation. So, it should be considered to predict health bills.

Part A

In this part, we will apply `table()` method to get frequency of each class in our categorical variable. Then, we convert it to a dataframe using `data.fram()` method. Finally we add percentage column to this dataframe.

In [30]:

```
smokingStatus <- healthCare$smoking_status
smokingStatusTable <- table(smokingStatus)
smokingStatusTable <- data.frame(smokingStatusTable)
smokingStatusTable$Percentage <- smokingStatusTable$Freq / sum(smokingStatusTable$Freq) * 100
smokingStatusTable
```

A data.frame: 4 × 3

smokingStatus	Freq	Percentage
<fct>	<int>	<dbl>
formerly smoked	884	17.30280
never smoked	1892	37.03269
smokes	789	15.44334
Unknown	1544	30.22118

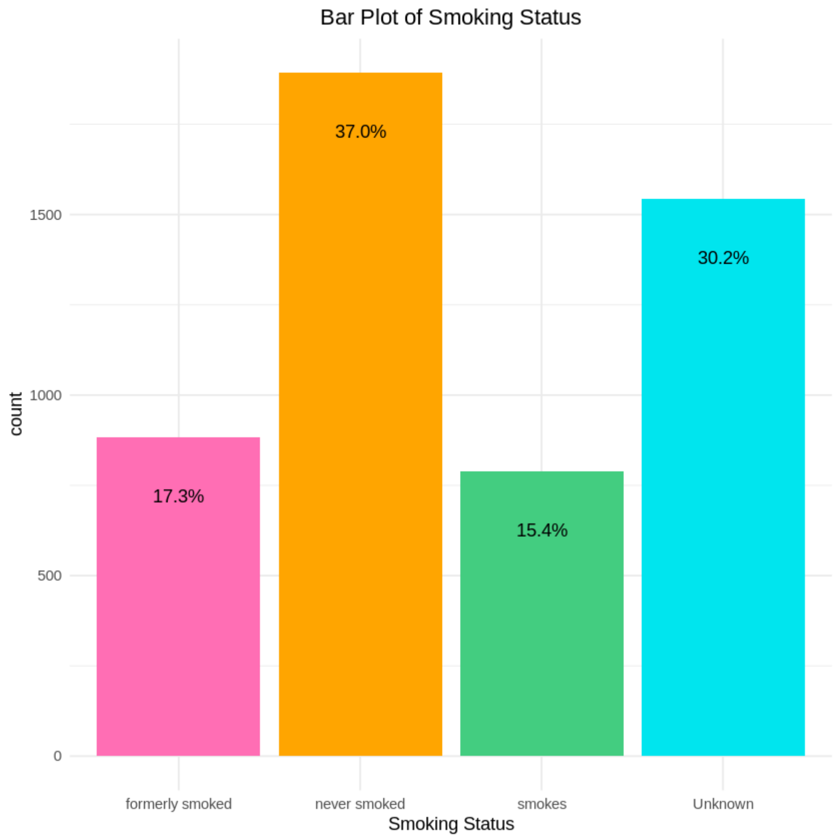
Part B

In this part, we will plot a bar plot for **Smoking Status**. We use different colors for each class by using fill. Moreover, by using `geom_text()` we can put percentage of each class on its bar.

In [31]:

```
colors <- c("hotpink1", "orange", "seagreen3", "turquoise2")

smokingStatusBar <- ggplot(data=healthCare, aes(x=smoking_status))
smokingStatusBar <- smokingStatusBar + geom_bar(fill = colors)
smokingStatusBar <- smokingStatusBar + geom_text(aes(label = scales::percent(..count../sum(..count..)), y= ..count.. ),
                                                  stat= "count", vjust = 5)
smokingStatusBar <- smokingStatusBar + ggtitle("Bar Plot of Smoking Status")
smokingStatusBar <- smokingStatusBar + theme_minimal()
smokingStatusBar <- smokingStatusBar + labs(x="Smoking Status")
smokingStatusBar <- smokingStatusBar + theme(plot.title = element_text(hjust = 0.5))
smokingStatusBar
```



Part C

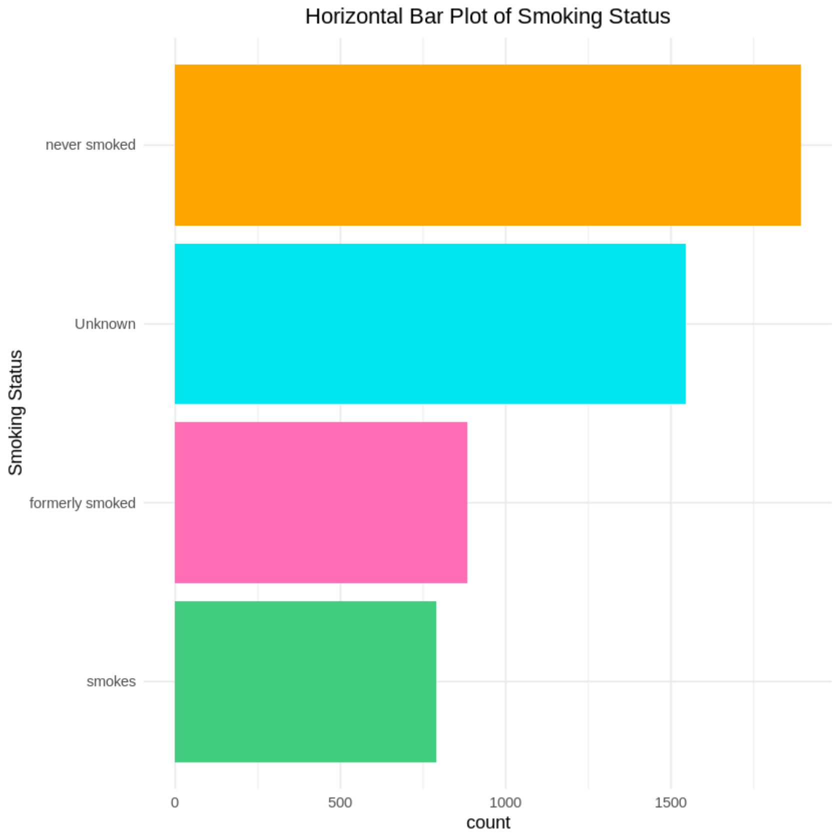
In this part, we want to sort bars of the previous plot and draw it horizontally. At first, we sort our dataset based on classes of **Smoking Status** and store it in a new dataframe. Finally, we use this new datafrane to plot the bar plot. We use `coord_flip()` to make it horizontal.

In [32]:

```
colors <- c("seagreen3", "hotpink1", "turquoise2", "orange")

sortedSmokingStatus <- within(healthCare,
                        smoking_status <- factor(
                                smoking_status,
                                levels=names(sort(table(smoking_status),
                                decreasing=FALSE))))

SmokingStatusBarH <- ggplot(data=sortedSmokingStatus, aes(smoking_status))
SmokingStatusBarH <- SmokingStatusBarH + geom_bar(fill=colors)
SmokingStatusBarH <- SmokingStatusBarH + theme_minimal()
SmokingStatusBarH <- SmokingStatusBarH + ggtitle("Horizontal Bar Plot of Smoking
Status")
SmokingStatusBarH <- SmokingStatusBarH + labs(x="Smoking Status")
SmokingStatusBarH <- SmokingStatusBarH + theme(plot.title = element_text(hjust =
0.5))
SmokingStatusBarH <- SmokingStatusBarH + coord_flip()
SmokingStatusBarH
```

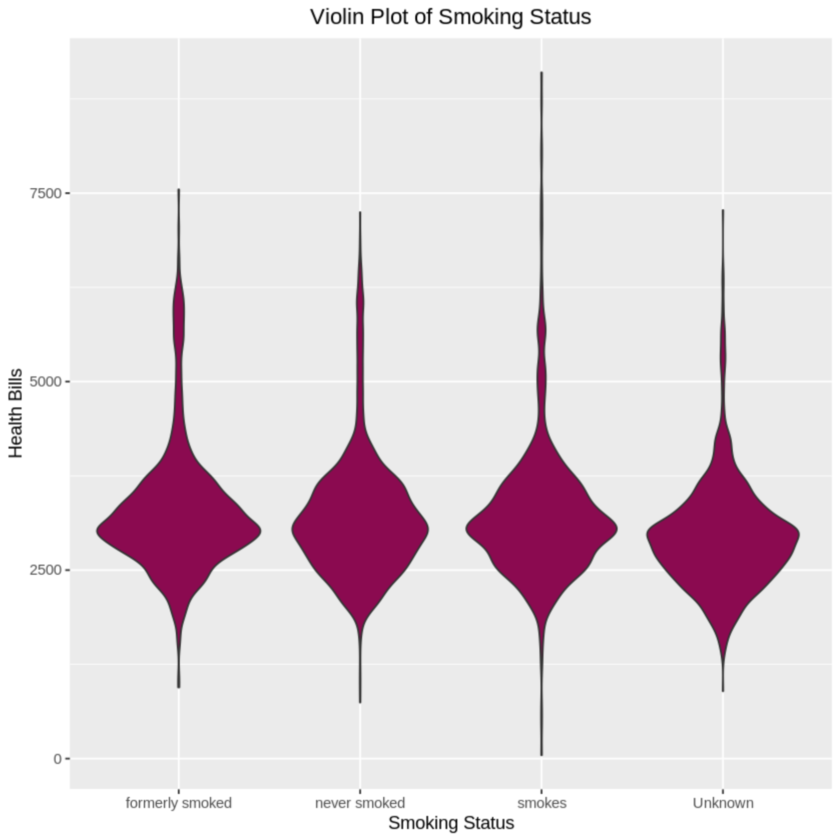


Part D

In this part, we will draw violin plot for **Smoking Status**. Also, we have to chose another column to be able to draw this plot. As we mentioned before, **Health Bills** is our taget column that we want to predict it. So, we choose this column as the second variable for out violin plot.

In [33]:

```
smokingStatusViolin <- ggplot(healthCare, aes(x=smoking_status, y=health_bills))
smokingStatusViolin <- smokingStatusViolin + geom_violin(fill = "deeppink4")
smokingStatusViolin <- smokingStatusViolin + ggtitle("Violin Plot of Smoking Status")
smokingStatusViolin <- smokingStatusViolin + labs(x="Smoking Status", y="Health Bills")
smokingStatusViolin <- smokingStatusViolin + theme(plot.title = element_text(hjust = 0.5))
smokingStatusViolin
```



Question 3

In this question, we choose **BMI** and **Health Bills** to perform following tasks with.

Part A

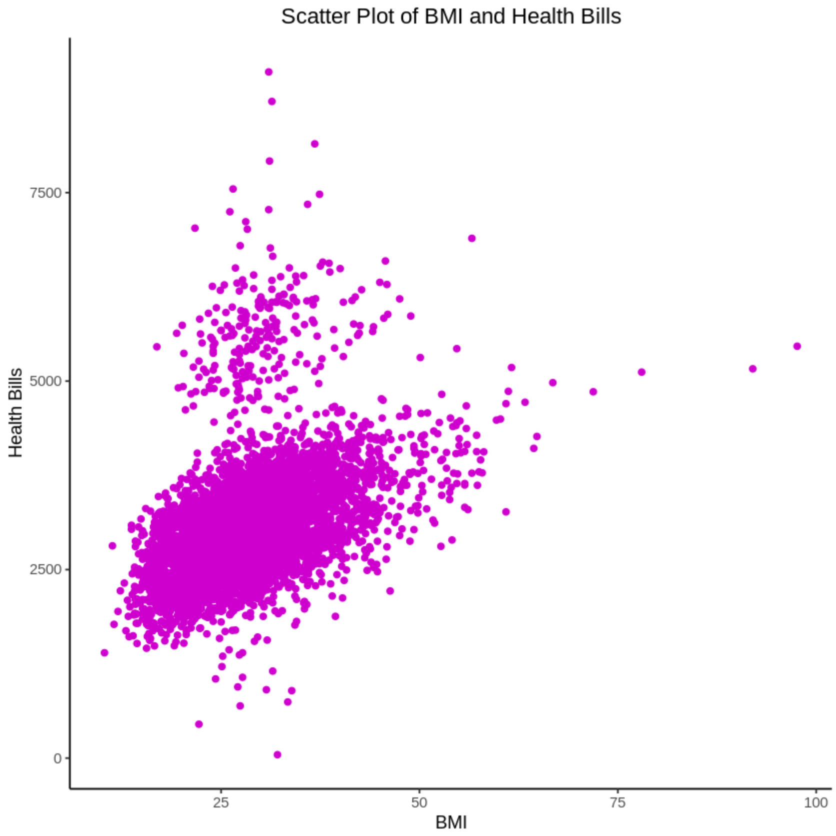
It is undeniable that being underweight or obesity can increase the risk of different types of disease for individuals. As the risk of diseases increases, the health bills that we should pay increases.

Part B

In this part, we draw scatter plot for these two variables. This is done by using `geom_point()` method from *ggplot* library.

In [34]:

```
scatter <- ggplot(healthCare, aes(x=bmi, y=health_bills))
scatter <- scatter + geom_point(color="magenta3")
scatter <- scatter + theme_classic()
scatter <- scatter + ggtitle("Scatter Plot of BMI and Health Bills")
scatter <- scatter + labs(x="BMI", y="Health Bills")
scatter <- scatter + theme(plot.title = element_text(hjust = 0.5))
scatter
```



Based on the result, it seems that there might be an association between these two variables. The data show an uphill pattern as we move from left to right, this indicates a positive relationship between them. As the BMI increase (move right), the health bills tend to increase (move up).

Part C

In this part, we compute the correlation coefficient betwwn these two variables using `cor()` method. By default it uses *pearson* method.

In [35]:

```
healthBills <- healthCare$health_bills
cor(bmi, healthBills)
```

```
0.422437011059867
```

Part D

Based on the computed correlation, it seems that these two variable are not independent and there is a positive association between them. This result is in good agreement with the answer of part A.

Part E

In this part, we use `cor.test()` method to run correlation test between these two variables.

In [36]:

```
cor.test(formula = ~ bmi + health_bills, data = healthCare)
```

```
        Pearson's product-moment correlation

data:  bmi and health_bills
t = 33.306, df = 5107, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3996445 0.4447075
sample estimates:
     cor 
0.422437 
```

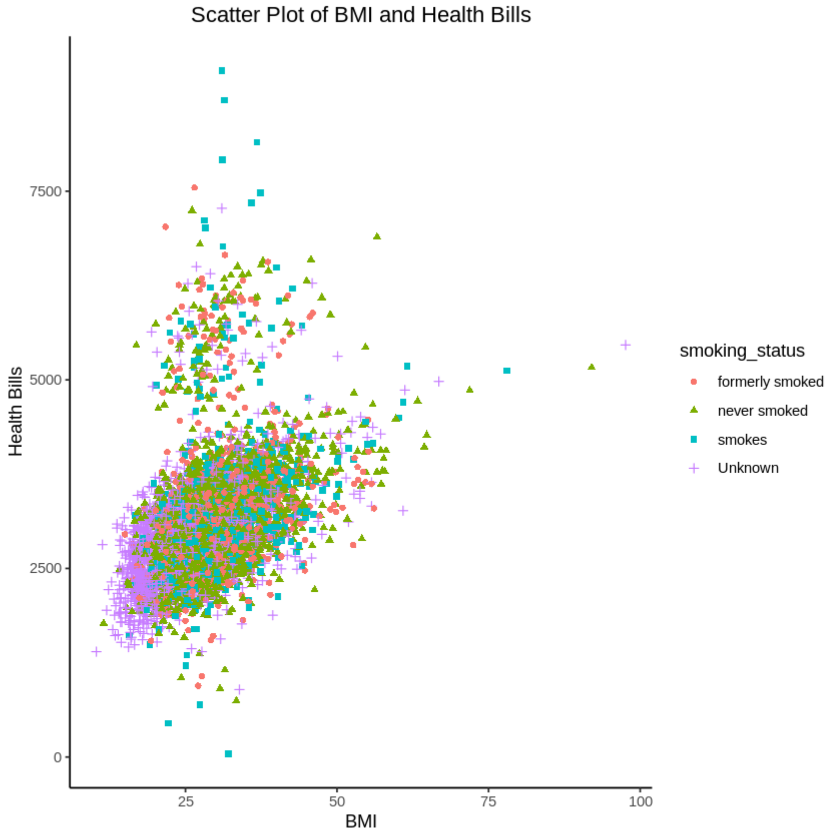
Based on the result, these two variables are correlated to each other. p-value = 2.2e-16 means these two variables are not independent (with the correlation equals to 0.4225391).

Part F

In this part, we choose **Smoking Status** as a categorical variable that we used in question 2. We assumed that it is one of the important factors which might have effect on **Health Bills**.

In [37]:

```
scatter <- ggplot(healthCare, aes(x=bmi, y=health_bills, shape=smoking_status, c
olor=smoking_status))
scatter <- scatter + geom_point()
scatter <- scatter + theme_classic()
scatter <- scatter + ggtitle("Scatter Plot of BMI and Health Bills")
scatter <- scatter + labs(x="BMI", y="Health Bills")
scatter <- scatter + theme(plot.title = element_text(hjust = 0.5))
scatter
```



Based on the result, our assumption in this part was not completely wrong and we can separate data based on **Smoking Status**.

Part G

In this part we use `geom_hex()` method to draw hexbin plot. Then, we use `geom_smooth()` to draw fitting curve. Finally we use `ggMarginal()` method to draw marginal histograms.

- It should be large enough to have data in most of the shapes.
- It should be small enough to allow us to start to see any relevant patterns.

In [38]:

```
hexBin <- ggplot(healthCare, aes(bmi, healthBills))
hexBin <- hexBin + theme_classic()
hexBin <- hexBin + scale_fill_gradient(low =  "palevioletred1", high = "palevioletred4")
hexBin <- hexBin + ggtitle("Hex Bin Plot of BMI and Health Bills")
hexBin <- hexBin + labs(x="BMI", y="Health Bills")
hexBin <- hexBin + theme(plot.title = element_text(hjust = 0.5))
hexBin <- hexBin + geom_point(col="transparent")
```

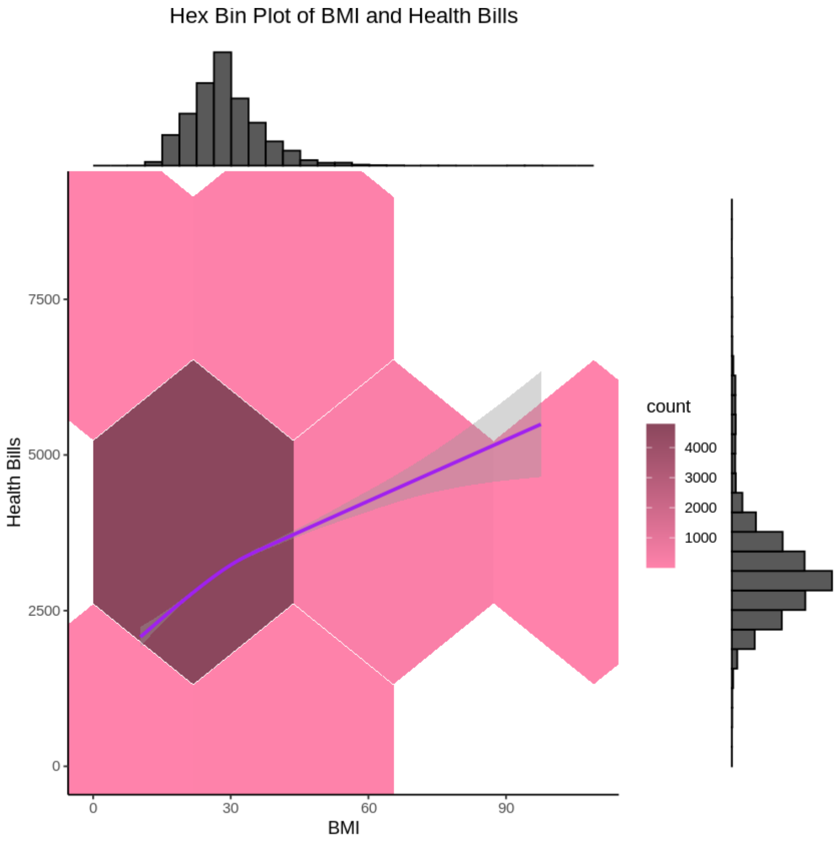
In [39]:

```
hexBin2 <- hexBin + geom_hex(bins=2)
hexBin2 <- hexBin2 + geom_smooth(col="purple")
ggMarginal(hexBin2, type = "histogram")
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'
```



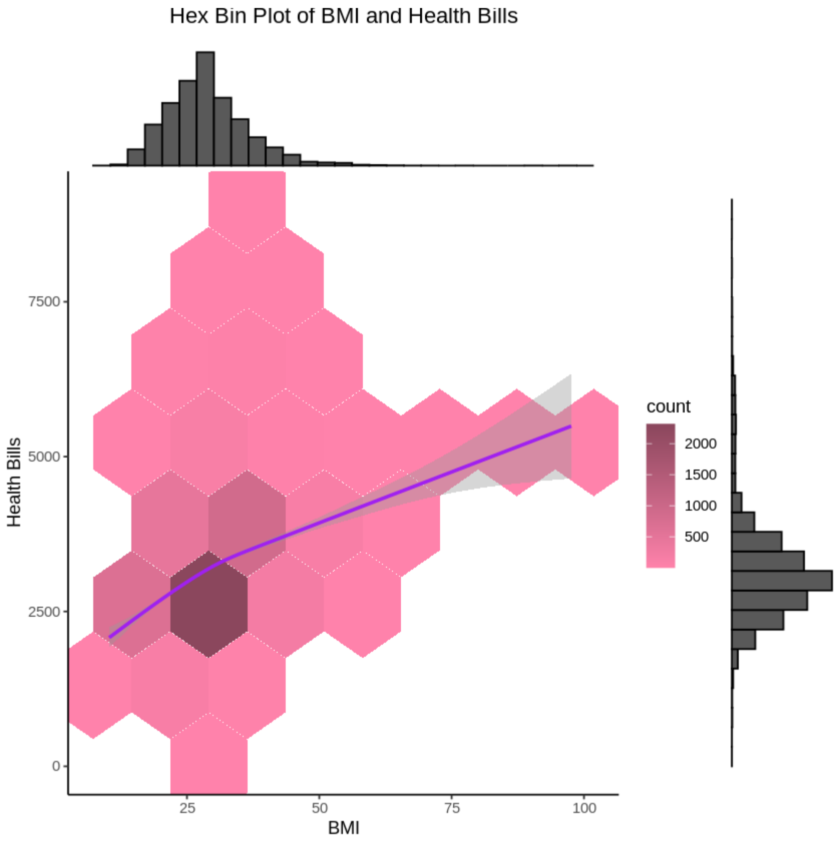
In [40]:

```
hexBin5 <- hexBin + geom_hex(bins=6)
hexBin5 <- hexBin5 + geom_smooth(col="purple")
ggMarginal(hexBin5, type = "histogram")
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'
```



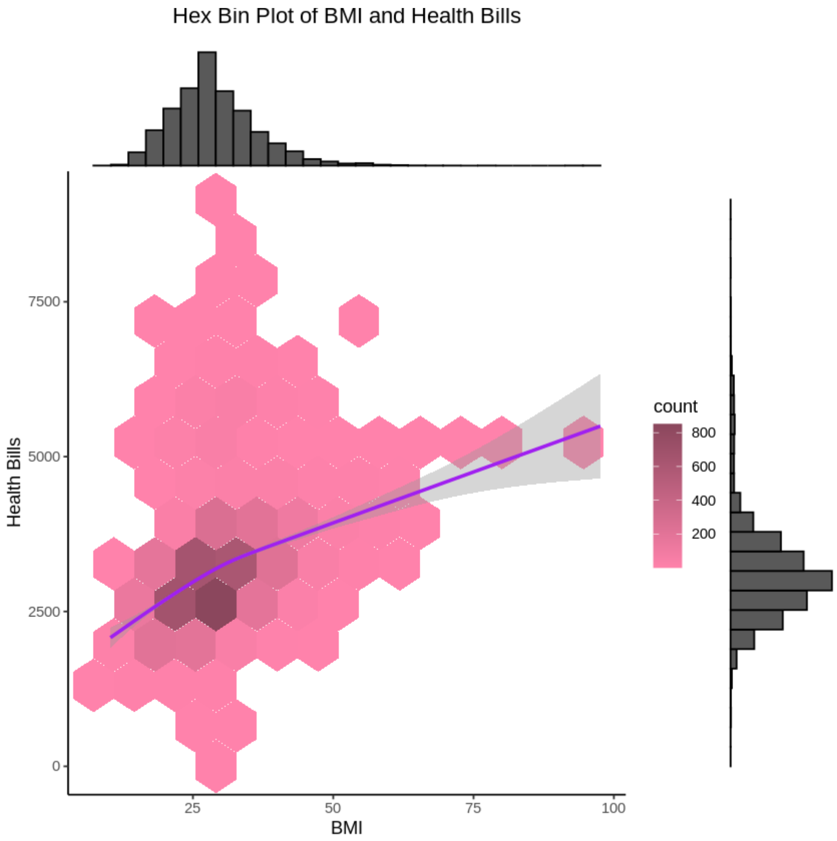
In [41]:

```
hexBin8 <- hexBin + geom_hex(bins=12)
hexBin8 <- hexBin8 + geom_smooth(col="purple")
ggMarginal(hexBin8, type = "histogram")
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'
```



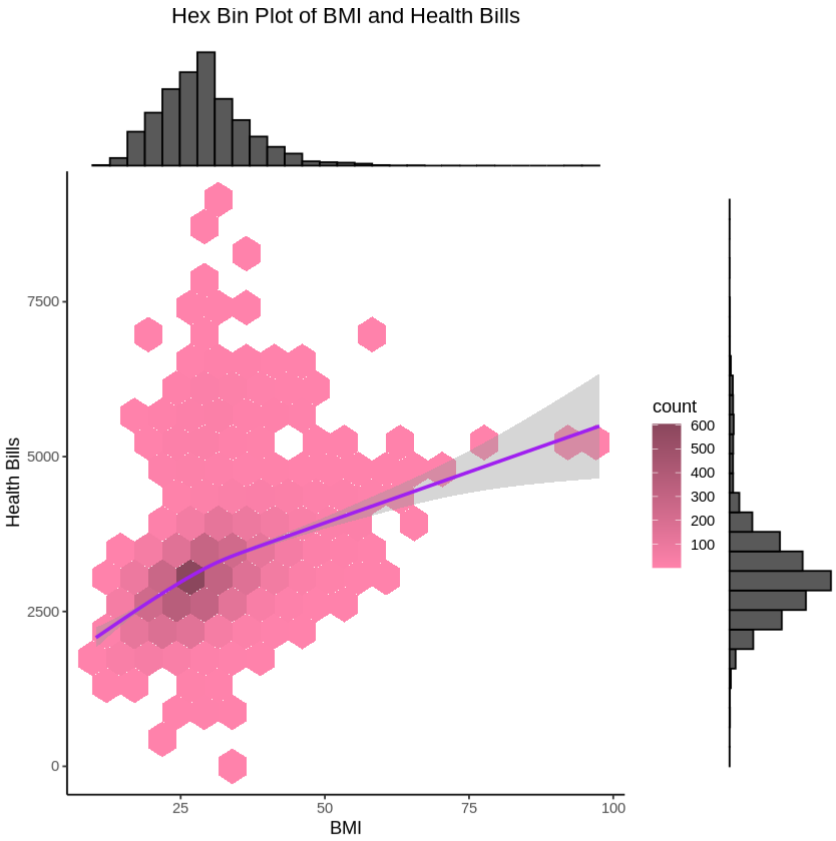
In [42]:

```
hexBin8 <- hexBin + geom_hex(bins=18)
hexBin8 <- hexBin8 + geom_smooth(col="purple")
ggMarginal(hexBin8, type = "histogram")
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'

`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "c
s")'
```



Based on the result, a large portion of data is located at **BMI** around 27 and **Health Bills** less than 2800.

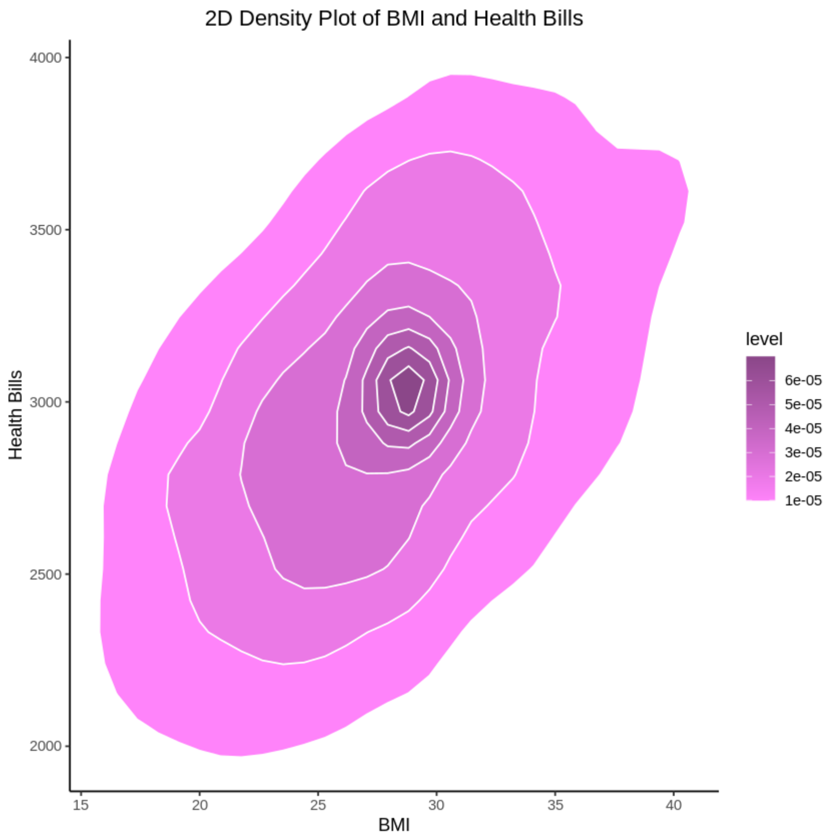
Moreover, we see that by decreasing the bin size it composed a larger amount of samples with each other. It seems that 6 and 12 are good size for bins and we can get good information from the variables by using them.

Part H

In this part, we use `stat_density_2d()` method to draw 2D density plot of these two variables.

In [43]:

```
density2D <- ggplot(healthCare, aes(x=bmi, y=healthBills))
density2D <- density2D + stat_density_2d(aes(fill = ..level..), geom = "polygon"
, colour="white")
density2D <- density2D + theme_classic()
density2D <- density2D + scale_fill_gradient(low =  "orchid1", high = "orchid4")
density2D <- density2D + ggtitle("2D Density Plot of BMI and Health Bills")
density2D <- density2D + labs(x="BMI", y="Health Bills")
density2D <- density2D + theme(plot.title = element_text(hjust = 0.5))
density2D
```



Based on the result, it is in a good agreement with the result of last part.

Advantages & Disadvantages of 2D Density & Hexbin

If we want to draw a scatter plot for a huge dataset, our result would be like a messy dark blob in the center with a smattering of distinguishable points around its surrounding. Therefore it is not very informative. In this situation **Hexbin** can be more useful. It automatically returns values using a color gradient for density.

2D density plot is very useful to avoid overplotting in a scatterplot.

As a disadvantage, if we do not select the bin size carefully, it may generate some uninformative results. Sometimes these results may have too much information that we can not obtain an abstract view from data and sometime we can not gain any information.

Question 4

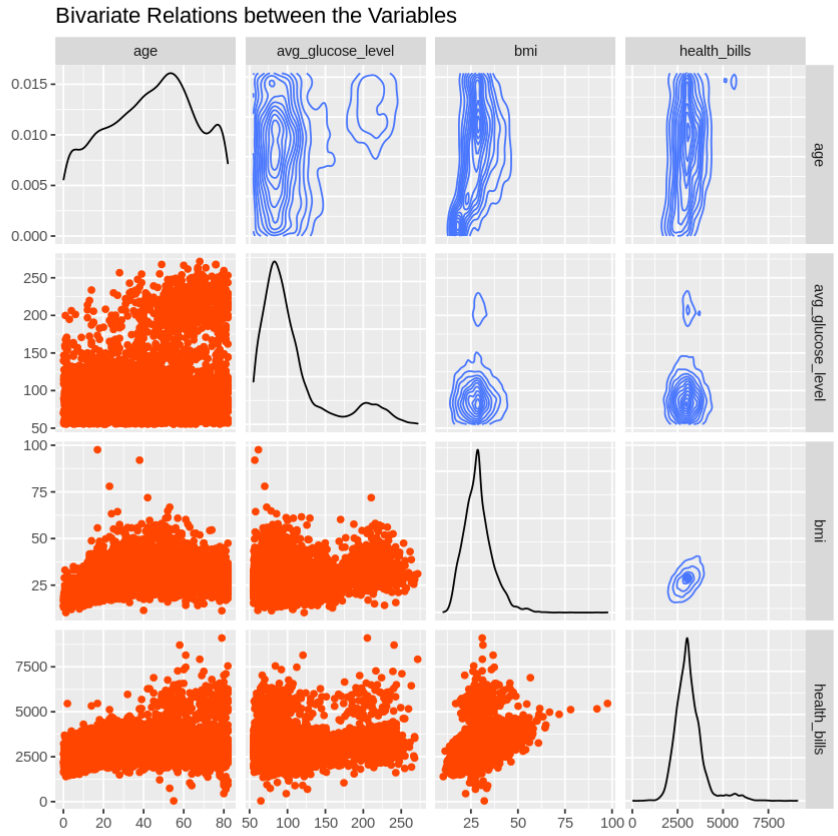
In this question, we consider **Age**, **Average Glucose Level**, **BMI**, and **Health Bills** as a group of 4 numerical variables.

Part A

In this part, we will use `ggpairs()` method from *GGally* library to plot pairwise scatterplots. This method allows us to build a great scatterplot matrix. Scatterplots of each pair of numeric variable are drawn on the left part of the figure. Pearson correlation is displayed on the right. Variable distribution is available on the diagonal.

In [44]:

```
ggpairs(healthCare[c("age", "avg_glucose_level", "bmi", "health_bills")],
 title="Bivariate Relations between the Variables ",
 lower=list(coreSize=10, continuous = wrap("points", color= "orangered")),
 upper=list(coreSize=10, continuous = wrap("density", color= "royalblue1")))
```



Based on the result, we can conclude that **BMI** is the most correlated variable to **Health Bills** among the numerical variables. This is a confirmation for our assumptions in previous part.

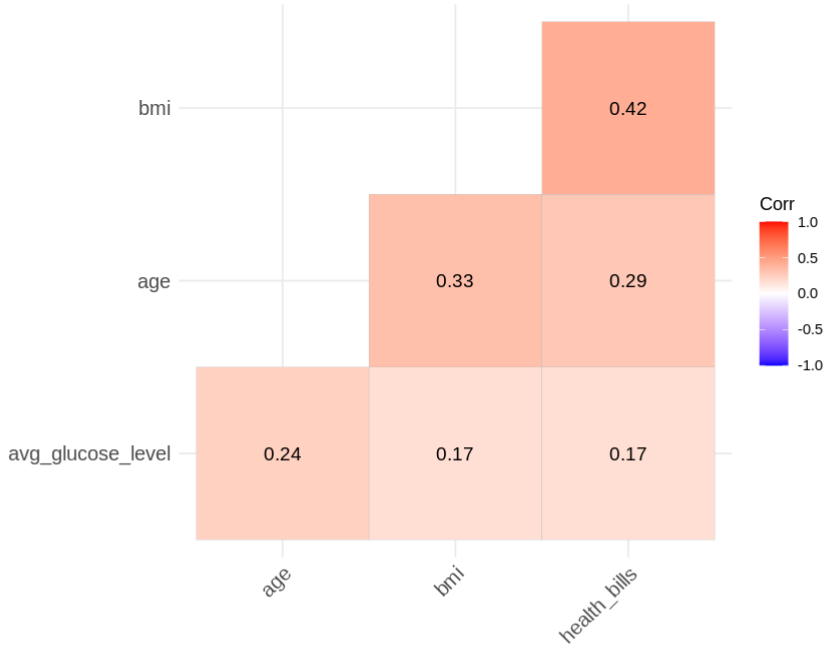
Moreover, **Age** is somehow correlated with **BMI** and there is a positive association between them.

Part B

In this part, we use `ggcorrplot()` from *ggcorrplot* library to plot to create a heatmap correlogram from our features. Also, we use `cor_pmat()` method to compute matrix of p-value. Finally, we set the significance level to 0.05.

In [45]:

```
numericVars <- healthCare[c("age", "avg_glucose_level", "bmi", "health_bills")]
corr <- cor(numericVars)
p.mat <- cor_pmat(numericVars)
ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE, p.mat = p.mat, sig
.level = 0.05)
```



In [46]:

```
p.mat
```

A matrix: 4 × 4 of type dbl

	age	avg_glucose_level	bmi	health_bills
age	0.000000e+00	6.647636e-67	1.108772e-126	1.387186e-102
avg_glucose_level	6.647636e-67	0.000000e+00	5.205739e-34	1.242458e-34
bmi	1.108772e-126	5.205739e-34	0.000000e+00	2.725930e-220
health_bills	1.387186e-102	1.242458e-34	2.725930e-220	0.000000e+00

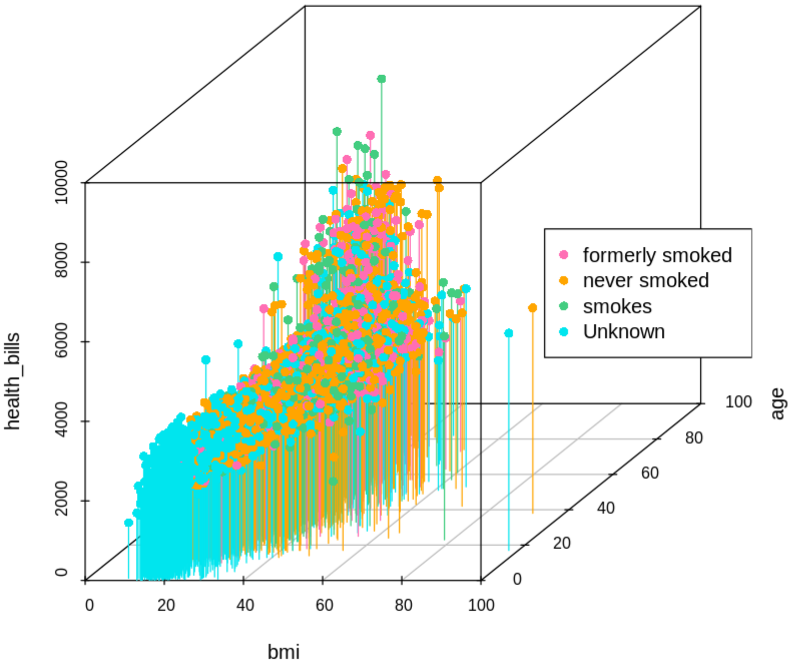
Based on the result, the highest correlation is between **BMI** and **Health Bills** that wew discussed in previous parts.

Part C

In this part, we use `scatterplot3d()` method from *scatterplot3d* library to draw 3D scatterplot for our dataset. We choose **BMI**, **Age**, and **Health Bills** as three numerical variables. Also, we choose **Smoking Status** as a categorical variable for coloring of the plot.

In [47]:

```
colors <- c("hotpink1", "orange", "seagreen3", "turquoise2")
smokingFactor <- factor(healthCare$smoking_status)
colors <- colors[as.numeric(smokingFactor)]
scatter3D <- scatterplot3d(healthCare[c("bmi", "age", "health_bills")],
                pch = 16, color=colors, type="h")
legend("right", legend = levels(smokingFactor),
      col = c("hotpink1", "orange", "seagreen3", "turquoise2"), pch = 16)
```



Based on the result, **Smoking Status** can separate data points into groups. Moreover, it shows that people who smoke should pay more for their health bills.

Question 5

Part A

A contingency is a tabular mechanism with at least two rows and two columns used in statistics to present categorical data in terms of frequency counts.

For this chart, we consider **Gender** and **Smoking Status** featuresa of our dataset. We use `table()` method to get the contingency table of these two variables. Also, we use `addmargins()` method to row and column for sum.

In [48]:

```
tableColors <- c("red", "red","red","red")
contingencyTable <- table(healthCare$gender, healthCare$smoking_status)
addmargins(contingencyTable)
```

A table: 3 × 5 of type dbl

	formerly smoked	never smoked	smokes	Unknown	Sum
Female	477	1229	452	836	2994
Male	407	663	337	708	2115
Sum	884	1892	789	1544	5109

Part B

A grouped bar chart extends the bar chart, plotting numeric values for levels of two categorical variables instead of one. Bars are grouped by position for levels of one categorical variable, with color indicating the secondary category level within each group.

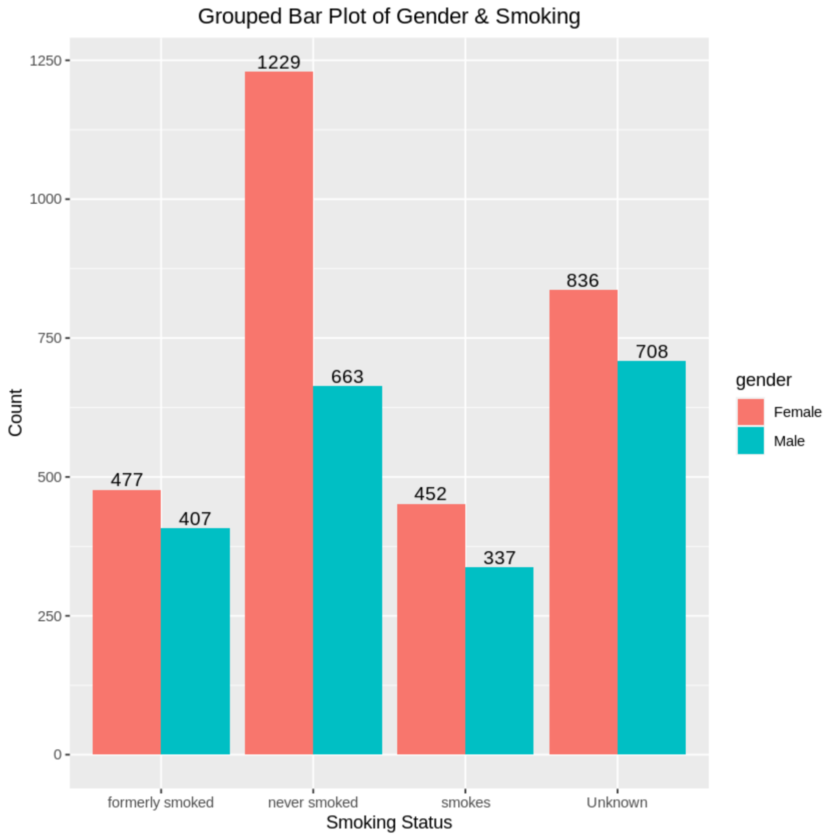
For this chart, we consider **Gender** and **Smoking Status** featuresa of our dataset. We use `geom_bar()` method with dodge position to draw this plot.

In [49]:

```
barData <- healthCare %>% group_by(smoking_status, gender) %>% summarise(Count=n
())

groupedBar <- ggplot(barData, aes(fill=gender, y=Count, x=smoking_status))
groupedBar <- groupedBar + geom_bar(position="dodge", stat="identity")
groupedBar <- groupedBar + geom_text(aes(label=Count), position = position_dodge
(width = 0.9),
                vjust = -0.25, size = 4)
groupedBar <- groupedBar + ggtitle("Grouped Bar Plot of Gender & Smoking")
groupedBar <- groupedBar + xlab("Smoking Status")
groupedBar <- groupedBar + theme(plot.title = element_text(hjust = 0.5))
groupedBar
```

```
`summarise()` has grouped output by 'smoking_status'. You can overri
de using the `.groups` argument.
```



Part C

A segmented Bar chart is one kind of stacked bar chart, but each bar will show 100% of the discrete value.

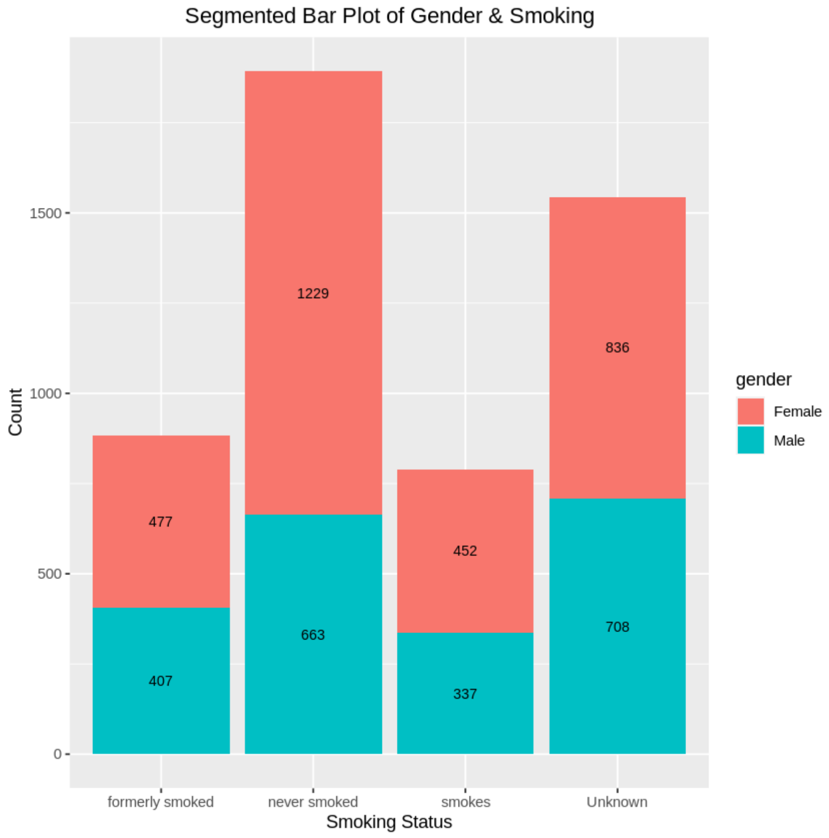
For this chart, we consider **Gender** and **Smoking Status** featuresa of our dataset. We use `geom_bar()` method with stack position to draw this plot.

In [50]:

```
barData <- healthCare %>% group_by(smoking_status, gender) %>% summarise(Count=n
())

segmentedBar <- ggplot(barData, aes(fill=gender, y=Count, x=smoking_status))
segmentedBar <- segmentedBar + geom_bar(stat="identity")
segmentedBar <- segmentedBar + geom_text(size = 3, aes(label=Count),
                              position = position_stack(vjust = 0.5))
segmentedBar <- segmentedBar + ggtitle("Segmented Bar Plot of Gender & Smoking")
segmentedBar <- segmentedBar + xlab("Smoking Status")
segmentedBar <- segmentedBar + theme(plot.title = element_text(hjust = 0.5))
segmentedBar
```

```
`summarise()` has grouped output by 'smoking_status'. You can overri
de using `.groups` argument.
```



Part D

A mosaic plot is a graphical display that allows you to examine the relationship among two or more categorical variables.

The mosaic plot starts as a square with length one. The square is divided first into horizontal bars whose widths are proportional to the probabilities associated with the first categorical variable. Then each bar is split vertically into bars that are proportional to the conditional probabilities of the second categorical variable. Additional splits can be made if wanted using a third, fourth variable, etc

For this chart, we consider **Gender** and **Smoking Status** featuresa of our dataset. We use `facet_grid()` method to draw this plot.

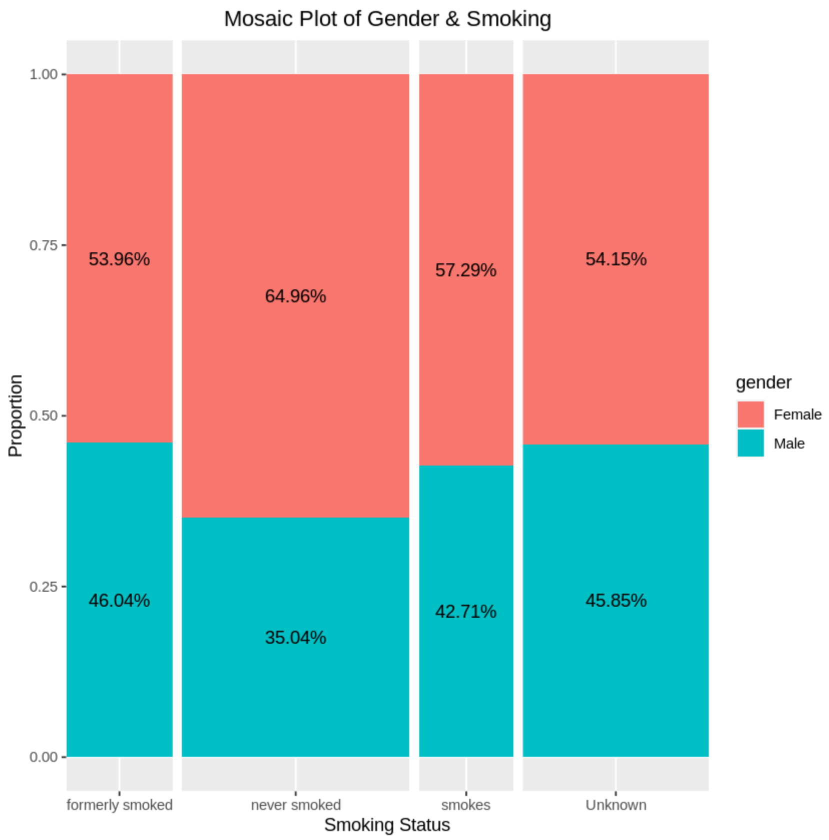
In [51]:

```
mosaicData <- healthCare %>%
group_by(smoking_status, gender) %>%
summarise(count = n()) %>%
mutate(smoking_status.count = sum(count),
        prop = count/sum(count)) %>%
ungroup()

mosaicPlot <- ggplot(mosaicData, aes(x = smoking_status, y = prop,
                      width = smoking_status.count, fill = gender))
mosaicPlot <- mosaicPlot + geom_bar(stat = "identity")
mosaicPlot <- mosaicPlot + geom_text(aes(label = scales::percent(prop)),
                                    position = position_stack(vjust = 0.5))
mosaicPlot <- mosaicPlot + facet_grid(~smoking_status, scales = "free_x", space
= "free_x")
mosaicPlot <- mosaicPlot + ggtitle("Mosaic Plot of Gender & Smoking")
mosaicPlot <- mosaicPlot + xlab("Smoking Status")
mosaicPlot <- mosaicPlot + ylab("Proportion")
mosaicPlot <- mosaicPlot + theme(plot.title = element_text(hjust = 0.5),
                                  strip.background = element_blank(),
                                  strip.text.x = element_blank())

mosaicPlot
```

```
`summarise()` has grouped output by 'smoking_status'. You can overri
de using the `.groups` argument.
```



Question 6

As we mentioned in previous questions, our target variable is **Health Bills** and we are interested to predict it in future. So, we choose it for this question to perform following tasks on it.

Part A

In this part, we want to build a 95% confidence interval for the mean of **Health Bills**. At the first step, we compute and store some statistics that we need in the calculation of confidence interval. Moreover, we take a sample from the data of size 100.

In [52]:

```
billPopulation <- healthCare$health_bills
sampleSize <- 100
set.seed(2)
billSampleIndex <- sample(1:nrow(healthCare), sampleSize)
billSample <- healthCare[c("health_bills")][billSampleIndex, ]
sampleMean <- mean(billSample)
sampleMean
```

```
3034.02130995571
```

Now it is time to compute lower and upper bounds of confident interval.

$$\bar{x} \pm z^* \frac{s}{\sqrt{n}}$$

Conditions for Confidence Interval

1. **Independence:** As we used random sampling and the size of the sample (100) is less than 10% of the population, we can conclude that this condition is satisfied.
2. **Sample size/skew:** Size of the sample is greater than 30 and our population is not skewed. So, we met this condition.

In [53]:

```
sampleMean <- mean(billSample)
populationSd <- sd(billPopulation)

SE <- populationSd / sqrt(sampleSize)

ME <- qnorm(0.975) * SE

low <- sampleMean - ME
up <- sampleMean + ME

print(paste("the 95% CI=(",low,"up to ",up,")"), quote = FALSE)
```

```
[1] the 95% CI=( 2875.53291146154 up to  3192.50970844987 )
```

Part B

We are 95% confident that the mean health bills for all individuals in the given population is somewhere between 2856.15 and 3173.14.

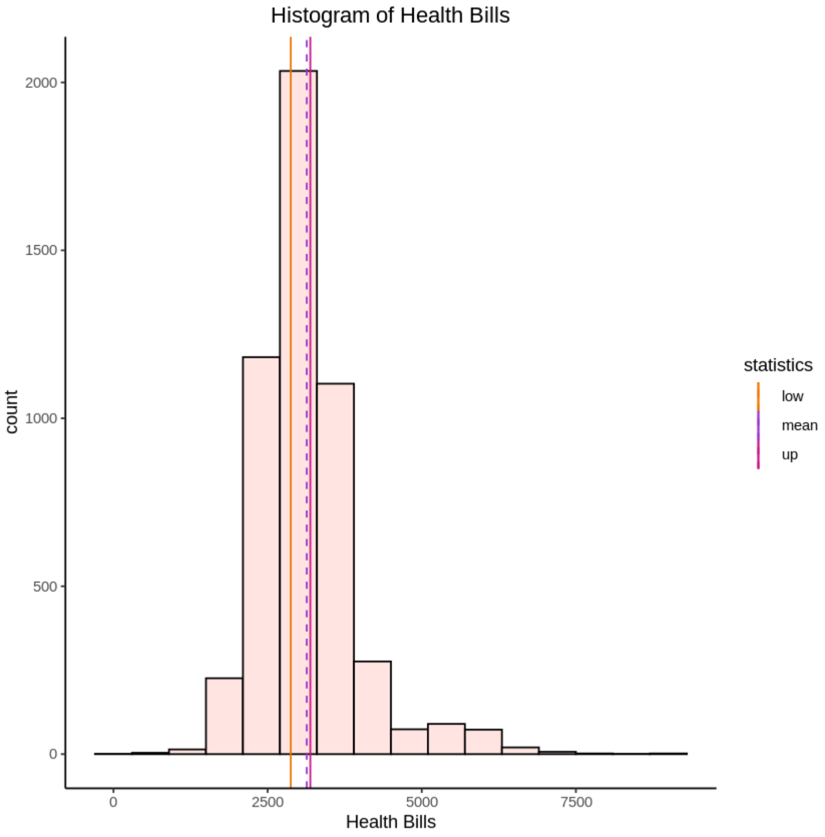
Part C

In this part, we use `geom_vline()` method from *ggplot* library to add mean and lower and upper bounds of confidence interval to our histogram. We draw them with different colors and also define a legend for them.

In [54]:

```
billData <- healthCare %>%
  summarize(mean = mean(health_bills),
            up = up,
            low = low)

binwidth <- 600
billHist <- ggplot(healthCare, aes(x=health_bills))
billHist <- billHist + geom_histogram(fill ="mistyrose", colour="black", binwidt
h = binwidth)
billHist <- billHist + geom_vline(data = billData, aes(xintercept = low, color =
"low"))
billHist <- billHist + geom_vline(data = billData, aes(xintercept = up, color =
"up"))
billHist <- billHist + geom_vline(data = billData, aes(xintercept = mean, , colo
r = "mean"),
                                                            linetype = "dashed")
billHist <- billHist + scale_color_manual(name = "statistics",
                  values = c(mean = "darkorchid", low = "darkorange2", up = "med
iumvioletred"))
billHist <- billHist + xlab("Health Bills")
billHist <- billHist + ggtitle("Histogram of Health Bills")
billHist <- billHist + theme_classic()
billHist <- billHist + theme(plot.title = element_text(hjust = 0.5))
billHist
```



Part D

In this part, the question that we design and want to answer is as follows:

Do individuals pay on average have paid more than 3000 for their health bills?

To answer this question, we statethe null and alternative hypotheses.

$$\begin{cases} H_0 : \mu = 3000 \\ H_A : \mu > 3000 \end{cases}$$

Now we compute the p-value based on the statistic that we find in last part.

In [55]:

```
mu_0 <- 3000
#test statistics
z <- (sampleMean - mu_0) / SE
p_value <- pnorm(z, lower.tail = FALSE)
p_value
```

```
0.336976774968319
```

$$p-value = 0.337 > 0.05 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills is greater than 3000.

Part E

Yes. null hypothesis (3000) is within the confidence interval. So, we fail to reject null hypothesis. In other words, the data do not support the hypothesis that the mean of health bills is greater than 3000.

Part F

In this part, we want to calculate type 2 error.

$$\beta = P(\text{Fail to reject } H_0 | \mu = \mu_a)$$
$$Power = 1 - \beta$$

Based on the above formulas, we can compute the power at the first step and then subtract it from 1 to obtain type 2 error.

In [56]:

```
actualMean <- mean(billPopulation)
alpha <- 0.05
zAlpha <- qnorm(alpha, lower.tail = FALSE)
zStatistics <- ((SE * zAlpha + mu_0) - actualMean) / SE
power <- pnorm(zStatistics, lower.tail = FALSE)
beta <- 1 - power
beta
```

```
0.492259941455642
```

We can see that type 2 error is about 49.23%. In other words, the probabilty that we fail to reject null hypothesis given that it null hypothesis is false is 0.4923.

Part G

In [57]:

```
power
```

```
0.507740058544358
```

We compute it in last part and it is 50.77%. Power is the probability of correctly rejecting null hypothesis,

The effect size tells us something about how relevant the relationship between two variables is in practice. There are two types of effect sizes:

- Effect size based on the proportion of explained variance: the proportion of explained variance is often indicated by one of the following terms: R² or eta squared, partial eta squared or omega squared. These forms are discussed later in the summary.
- Effect size based on the difference in averages. This is often referred to using Cohen's d.

The statistical power of a significance test depends 3 factors:

- The sample size (n): when n increases, the power increases;
- The significance level (α): when α increases, the power increases;
- The effect size (explained below): when the effect size increases, the power increases.

Question 7

Part A

In this part, we take a sample of size 25 from the data and then we perform following tasks on it.

In [58]:

```
sampleSize = 25
set.seed(123)
index <- sample(1:nrow(healthCare), sampleSize)
pairSample = healthCare[c("bmi", "age")][index, ]
```

Part a

Here, we will use t-test because the sample size is less than 30 and we can not meet the condition of sample size for z-test.

As we used randome sampling and the sample size is less than the 10% of population, the indepence condtion of t-test is satisfied and we can use it for this task.

Part b

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{diff} = 0$$
$$H_A : \mu_{diff} \neq 0$$

Average difference between the **Age** and **BMI** of all individuals in our population.

We have to mention that these two sample (age & bmi) are paired. So, they are dependent and we have to look at the difference in outcomes of each pair of observations and then run the t-test on it.

In [59]:

```
diff <- pairSample$age - pairSample$bmi
t.test(diff, mu = 0)
```

```
	One Sample t-test

data:  diff
t = 2.3996, df = 24, p-value = 0.02453
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  1.421751 18.901449
sample estimates:
mean of x 
  10.1616 
```

$$p - value = 0.024 < 0.05 \implies$$

We reject null hypothesis and conclude that there is significant evidence in the average of difference between the **Age** and **BMI**. And their difference is not equal to zero.

Part B

In this part, the two samples are independent. So, we have the codition of indepence between and within groups. Therefore, we can run t-test on these two groups.

Our hypotheses for this part are as follows:

$$H_0 : \mu_{age} = \mu_{bmi}$$
$$H_A : \mu_{age} \neq \mu_{bmi}$$

Firstly, we take the two samples from the poulation. One sample for **Age** and another sample for **BMI**.

In [60]:

```
sampleSize = 100
set.seed(123)
ageSampleIndex <- sample(1:nrow(healthCare), sampleSize)
bmiSampleIndex <- sample(1:nrow(healthCare), sampleSize)

ageSample <- healthCare[c("age")][ageSampleIndex, ]
bmiSample <- healthCare[c("bmi")][bmiSampleIndex, ]
```

Now it is time to run t-test on these groups.

In [61]:

```
t.test(ageSample, bmiSample)
```

```
	Welch Two Sample t-test

data:  ageSample and bmiSample
t = 5.4366, df = 127.96, p-value = 2.649e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  8.349213 17.904313
sample estimates:
mean of x mean of y 
 43.52760  30.40084 
```

$$p - value = 2.65e^{-7} < 0.05 \implies$$

We reject null hypothesis and conclude that there is significant evidence in the mean drop in **Age** and **BMI** groups.

As we can see, zero is not in the 95% confidence interval. Therefor we reject the null hypothesis. This is the same as what we obtained by using the p-value. This two methods are always in agreement with each other.

Question 8

In this question, we choose **BMI** as a numerical variable to do following tasks with.

Part A

In this part, we will compute a 95% confidence interval for the mean of this variable using percentile method. At first, we take 1000 samples of size 100 from the population without replacement. Then we compute the lower and upper bound of this interval as follows.

In [62]:

```
bmi = healthCare$bmi
CI = 0.95
sampleSize = 100
numBootSamples = 1000
set.seed(4)
boot <- replicate(numBootSamples, sample(bmi, size = sampleSize))
means <- sort(apply(X = boot, MARGIN = 2, FUN = mean))
lowIndex <- (1 - CI)/2 * numBootSamples
upIndex <- numBootSamples - (1 - CI)/2 * numBootSamples
low <- means[lowIndex]
up <- means[upIndex]
print(paste("the 95% CI=(",low,"up to ",up,")"), quote = FALSE)
```

```
[1] the 95% CI=( 27.451619193154 up to  30.468836797066 )
```

Part B

In this part, we will compute a 95% confidence interval for the mean of this variable using standard error method. At first, we take 1000 samples of size 20 from the population with replacement. Then we compute the lower and upper bound of this interval using bootstrap method.

In [63]:

```
sampleSize = 20
df = numBootSamples - 1
set.seed(4)
mySample <- sample(bmi, size = sampleSize, replace = TRUE)
tStar <- qt(0.975, df)
SE <- sd(means)
ME <- tStar * SE
low <- mean(mySample) - ME
up <- mean(mySample) + ME
print(paste("the 95% CI=(",low,"up to ",up,")"), quote = FALSE)
```

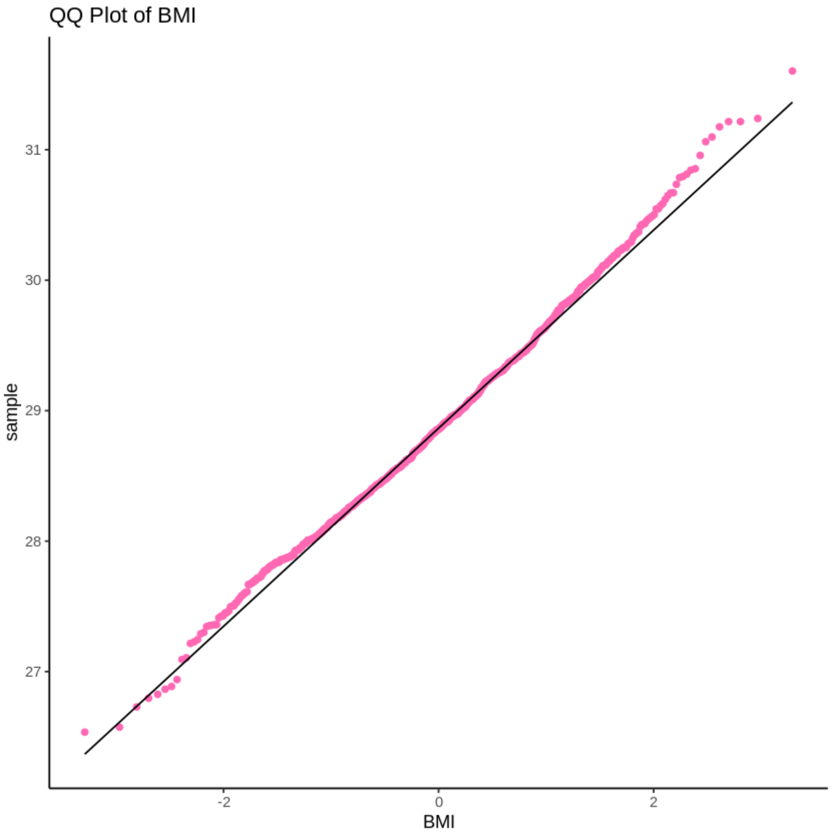
```
[1] the 95% CI=( 26.505772866177 up to  29.5136831240431 )
```

Part C

In this part, we draw a QQ Plot for the distribution of means. Based on the result, the little difference between the interval of two methods refers to the difference between the distribution of mean (bootstrap distribution) and the standard normal distribution.

In [78]:

```
meansDf <- data.frame(mean=means)
qqBMI <- ggplot(meansDf, aes(sample=mean))
qqBMI <- qqBMI + stat_qq(col="hotpink") + geom_qq_line()
qqBMI <- qqBMI + labs(x="BMI", title="QQ Plot of BMI")
qqBMI <- qqBMI + theme_classic()
qqBMI
```



Question 9

In this question, we want to compare the mean of health bill in different work type groups. At first, we put health bills of each group in a vactor.

In [65]:

```
private <- healthCare[healthCare$work_type == "Private",]$health_bills
selfEmployed <- healthCare[healthCare$work_type == "Self-employed",]$health_bil
ls
govtJob <- healthCare[healthCare$work_type == "Govt_job",]$health_bills
children <- healthCare[healthCare$work_type == "children",]$health_bills
neverWorked <- healthCare[healthCare$work_type == "Never_worked",]$health_bills
```

In this step, we state hypotheses as follows:

$$H_0 : \mu_{private} = \mu_{selfEmployed} = \mu_{govtJob} = \mu_{children} = \mu_{neverWorked}$$
$$H_A : \text{At least one pair of means are different from each other.}$$

It is time to run ANOVA test to evaluate our hypotheses.

In [66]:

```
y <- c(private, selfEmployed, govtJob, children, neverWorked)
n <- c(length(private), length(selfEmployed), length(govtJob), length(children),
        length(neverWorked))
group = rep(1:5, n)
tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x), n = length(x))
tapply(y, group, tmpfn)
data = data.frame(y = y, group = factor(group))
fit = lm(y ~ group, data)
anova(fit)
```

$`1`

sum: 9386642.07525402 **mean:** 3210.20590808961 **var:** 656619.842498027 **n:** 2924

$`2`

sum: 2649141.94144312 **mean:** 3234.6055451076 **var:** 780956.804122752 **n:** 819

$`3`

sum: 2092579.89243997 **mean:** 3185.05310873663 **var:** 604932.015547214 **n:** 657

$`4`

sum: 1820051.71765501 **mean:** 2649.27469818778 **var:** 275628.467431976 **n:** 687

$`5`

sum: 66136.2891785441 **mean:** 3006.19496266109 **var:** 344227.236998166 **n:** 22

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
group	4	188756793	47189198.2	76.43072	4.868617e-63
Residuals	5104	3151267768	617411.4	NA	NA

Based on the result, the p-value is less than the significance level 0.05. So, we can conclude that there are significant differences between at least two groups.

Bonus

For this part, we draw boxplot of **Health Bills** for each work type class. As we learnt in class, in multiple comparion there are two important faxctors that we should consider them:

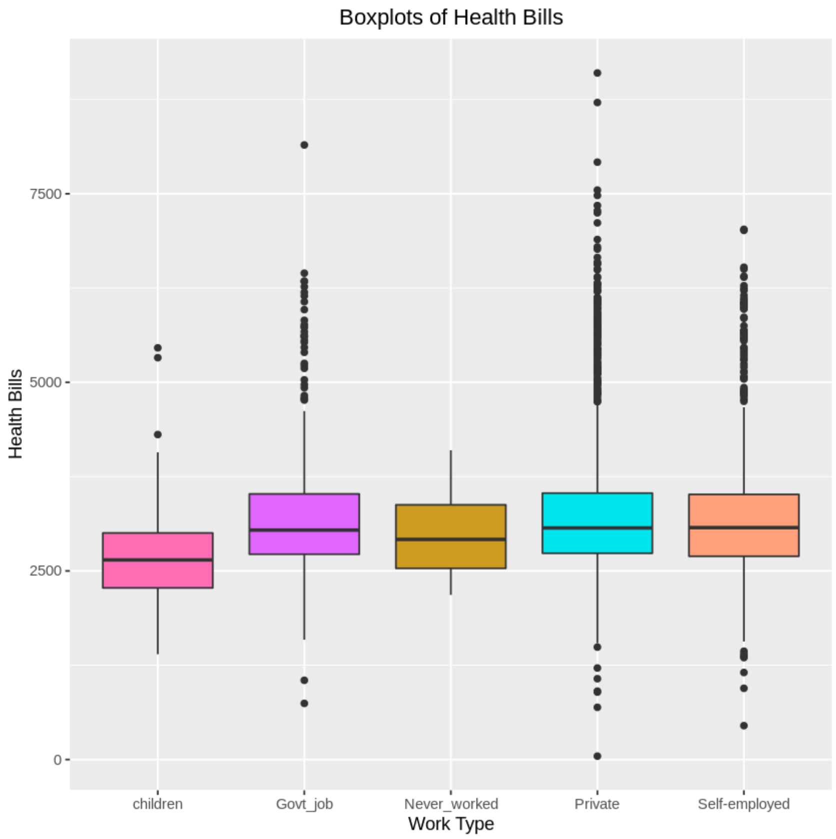
- **Variability within groups:** We can analyze this by considering variance in each group. In other words, compactness of box can be agood metric for this analysis.
- **Variability between groups:** We can analyze this by comparing medians of these groups. The reason is that the skewness for an aproximately normal distribution is to small. As a result, the mean and median of such a distribution will be too close to each other. So, we can compare their medians.

Moreover, the more overlap exists between boxes,the less differene in means they have.

In [67]:

```
colors <- c("hotpink1", "mediumorchid1", "goldenrod3", "turquoise2", "lightsalmon1")

boxPlot <- ggplot(healthCare, aes(x = work_type, y = health_bills))
boxPlot <- boxPlot + geom_boxplot(fill = colors)
boxPlot <- boxPlot + ggtitle("Boxplots of Health Bills")
boxPlot <- boxPlot + xlab("Work Type")
boxPlot <- boxPlot + ylab("Health Bills")
boxPlot <- boxPlot + theme(plot.title = element_text(hjust = 0.5))
boxPlot
```



Based on the result, we can see that the median of **children** group has a significant difference with another groups. Moreover, the compactness of boxes are appropriate. So, we can conclude tha there is a significant difference between means of at least two groups. This conclusion is in agreement with the result of ANOVA test.

Now we have to find these pairs.

In this part, we compute significant level for our tests as follows:

$$K = \frac{k.\,(k-1)}{2} = \frac{5.(5-1)}{2} = 10$$
$$\alpha^* = \frac{\alpha}{K} = \frac{0.05}{10} = 0.005$$

Private vs Self Employed

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{private} = \mu_{selfEmployed}$$
$$H_A : \mu_{private} \neq \mu_{selfEmployed}$$

Now we will evaluate our hypothesis with t-test.

In [68]:

```
t.test(private, selfEmployed, data = y)
```

```
	Welch Two Sample t-test

data:  private and selfEmployed
t = -0.71087, df = 1229.6, p-value = 0.4773
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -91.73900  42.93972
sample estimates:
mean of x mean of y 
 3210.206  3234.606 
```

$$p - value = 0.471 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Private** and **Government Job** groups are different.

Private vs Government Job

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{private} = \mu_{governmentJob}$$
$$H_A : \mu_{private} \neq \mu_{governmentJob}$$

Now we will evaluate our hypothesis with t-test.

In [69]:

```
t.test(private, govtJob, data = y)
```

```
	Welch Two Sample t-test

data:  private and govtJob
t = 0.74323, df = 1001.6, p-value = 0.4575
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -41.25742  91.56302
sample estimates:
mean of x mean of y 
 3210.206  3185.053 
```

$$p - value = 0.253 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Self Employed** and **Government Job** groups are different.

Private vs Children

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{private} = \mu_{children}$$
$$H_A : \mu_{private} \neq \mu_{children}$$

Now we will evaluate our hypothesis with t-test.

In [70]:

```
t.test(private, children, data = y)
```

```
	Welch Two Sample t-test

data:  private and children
t = 22.423, df = 1554.5, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 511.8638 609.9986
sample estimates:
mean of x mean of y 
 3210.206  2649.275 
```

$$p - value = 2.2e^{-16} < 0.005 \implies$$

We reject null hypothesis and conclude that there is significant evidence in the mean drop in health bills between the **Children** and **Private** groups.

Private vs Never Worked

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{private} = \mu_{neverWorked}$$
$$H_A : \mu_{private} \neq \mu_{neverWorked}$$

Now we will evaluate our hypothesis with t-test.

In [71]:

```
t.test(private, neverWorked, data = y)
```

```
        Welch Two Sample t-test

data:  private and neverWorked
t = 1.6194, df = 21.607, p-value = 0.1199
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -57.53375 465.55564
sample estimates:
mean of x mean of y 
 3210.206  3006.195 
```

$$p - value = 0.253 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Self Employed** and **Government Job** groups are different.

Self Employed vs Government Job

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{selfEmployed} = \mu_{governmentJob}$$
$$H_A : \mu_{selfEmployed} \neq \mu_{governmentJob}$$

Now we will evaluate our hypothesis with t-test.

In [72]:

```
t.test(selfEmployed, govtJob, data = y)
```

```
        Welch Two Sample t-test

data:  selfEmployed and govtJob
t = 1.1446, df = 1461.4, p-value = 0.2526
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -35.37094 134.47582
sample estimates:
mean of x mean of y 
 3234.606  3185.053 
```

$$p - value = 0.253 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Self Employed** and **Government Job** groups are different.

Self Employed vs Children

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{selfEmployed} = \mu_{children}$$
$$H_A : \mu_{selfEmployed} \neq \mu_{children}$$

Now we will evaluate our hypothesis with t-test.

In [73]:

```
t.test(selfEmployed, children, data = y)
```

```
	Welch Two Sample t-test

data:  selfEmployed and children
t = 15.903, df = 1363.4, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 513.1264 657.5353
sample estimates:
mean of x mean of y 
 3234.606  2649.275 
```

$$p - value = 2.2e^{-16} < 0.005 \implies$$

We reject null hypothesis and conclude that there is significant evidence in the mean drop in health bills between the **Children** and **Self Employed** groups.

Self Employed vs Never Worked

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{selfEmployed} = \mu_{neverWorked}$$
$$H_A : \mu_{selfEmployed} \neq \mu_{neverWorked}$$

Now we will evaluate our hypothesis with t-test.

In [74]:

```
t.test(selfEmployed, neverWorked, data = y)
```

```
	Welch Two Sample t-test

data:  selfEmployed and neverWorked
t = 1.7728, df = 23.635, p-value = 0.08915
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -37.72337 494.54454
sample estimates:
mean of x mean of y 
 3234.606  3006.195 
```

$$p - value = 0.089 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Self Employed** and **Never worked** groups are different.

Government Job vs Children

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{governmentJob} = \mu_{children}$$
$$H_A : \mu_{governmentJob} \neq \mu_{children}$$

Now we will evaluate our hypothesis with t-test.

In [75]:

```
t.test(govtJob, children, data = y)
```

```
	Welch Two Sample t-test

data:  govtJob and children
t = 14.736, df = 1144.5, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 464.4412 607.1156
sample estimates:
mean of x mean of y 
 3185.053  2649.275 
```

$$p - value = 2.2e^{-16} < 0.005 \implies$$

We reject null hypothesis and conclude that there is significant evidence in the mean drop in health bills between the **Children** and **Government Job** groups.

Government Job vs Never Worked

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{governmentJob} = \mu_{neverWorked}$$
$$H_A : \mu_{governmentJob} \neq \mu_{neverWorked}$$

Now we will evaluate our hypothesis with t-test.

In [76]:

```
t.test(govtJob, neverWorked, data = y)
```

```
	Welch Two Sample t-test

data:  govtJob and neverWorked
t = 1.3896, df = 23.542, p-value = 0.1777
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -87.06965 444.78594
sample estimates:
mean of x mean of y 
 3185.053  3006.195 
```

$$p - value = 0.178 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Government Job** and **Never worked** groups are different.

Never Worked vs Children

At first, we state the null and alternative hypotheses:

$$H_0 : \mu_{neverWorked} = \mu_{children}$$
$$H_A : \mu_{neverWorked} \neq \mu_{children}$$

Now we will evaluate our hypothesis with t-test.

In [77]:

```
t.test(children, neverWorked, data = y)
```

```
	Welch Two Sample t-test

data:  children and neverWorked
t = -2.8175, df = 22.09, p-value = 0.01
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -619.57687  -94.26366
sample estimates:
mean of x mean of y 
 2649.275  3006.195 
```

$$p - value = 0.01 > 0.005 \implies$$

We fail to reject null hypothesis. The data do not support the hypothesis that the mean of health bills between the **Children** and **Never worked** groups are different.

Health Care Dataset

Ghazal Kalhor

***Abstract* — In this computer assignment, we want to perform statistical analysis on healthCare dataset. We will use methods that we learnt in Statistical Inference. Also, we will use R programming language to reach this goal. *Keywords* — Statistical Inference, R**

Importing Libraries

In this part, we will import some of the necessary libraries in order to use their helpful functions. Firstly, we will install related packages. Secondly, we will use `library()` function to import them.

In [ ]:

```
options(warn=-1)
install.packages("plyr")
install.packages("GGally")
install.packages("caret")
install.packages("ggcorrplot")
```

In [126]:

```
library(plyr)
library(ggplot2)
library(GGally)
library(caret)
library(ggcorrplot)
library(pROC)
```

Importing Data

In this part, file *HealthCare.csv* is coppied to the project directory, then we read and store it in a dataframe called *heathCare*.

In [ ]:
```
healthCare <- read.csv("/content/HealthCare.csv")
```

In [ ]:
```
summary(healthCare)
```

```
       id           gender                age         hypertension    
 Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000  
 1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000  
 Median :36932   Mode  :character   Median :45.00   Median :0.00000  
 Mean   :36518                      Mean   :43.23   Mean   :0.09746  
 3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000  
 Max.   :72940                      Max.   :82.00   Max.   :1.00000  
                                                                     
 heart_disease     ever_married        work_type          Residence_ty
pe
 Min.   :0.00000   Length:5110        Length:5110        Length:5110  
 1st Qu.:0.00000   Class :character   Class :character   Class :chara
cter
 Median :0.00000   Mode  :character   Mode  :character   Mode  :chara
cter
 Mean   :0.05401                                                      
 3rd Qu.:0.00000                                                      
 Max.   :1.00000                                                      
                                                                      
 avg_glucose_level      bmi        smoking_status         stroke       
 Min.   : 55.12    Min.   :10.30   Length:5110        Min.   :0.00000  
 1st Qu.: 77.25    1st Qu.:23.50   Class :character   1st Qu.:0.00000  
 Median : 91.89    Median :28.10   Mode  :character   Median :0.00000  
 Mean   :106.15    Mean   :28.89                      Mean   :0.04873  
 3rd Qu.:114.09    3rd Qu.:33.10                      3rd Qu.:0.00000  
 Max.   :271.74    Max.   :97.60                      Max.   :1.00000  
                   NA's   :201                                         
  health_bills   
 Min.   :  44.8  
 1st Qu.:2628.8  
 Median :3031.7  
 Mean   :3138.6  
 3rd Qu.:3474.4  
 Max.   :9100.5  
 NA's   :201     
```

Cleaning Data

In this part, we will convert the column values that are in a wrong format to an appropriate format. Values in **hypertension**, **heart_disease**, and **stroke** are stored as Integer but they categorical variables and it would be better to store them as String. This can be done by using `mapvalues()` method from *plyr* library.

In [ ]:
```
healthCare$hypertension <- mapvalues(healthCare$hypertension,
                           from = c(0, 1),
                           to = c("No", "Yes"))
```

In [ ]:

```
healthCare[(healthCare$gender=="Other"),]
```

A data.frame: 1 × 13

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_
	<int>	<chr>	<dbl>	<chr>	<int>	<chr>	<chr>	<
3117	56156	Other	26	No	0	No	Private	

In [ ]:

```
healthCare <- healthCare[!(healthCare$gender=="Other"),]
```

Handling NA Values

In this part, we will use a combination of `colMeans()` and `is.na()` methods in order to compute proportion of nan values in each column.

In [ ]:

```
colMeans(is.na(healthCare))
```

```
id: 0 gender: 0 age: 0 hypertension: 0 heart_disease: 0 ever_married: 0 work_type: 0
 Residence_type: 0 avg_glucose_level: 0 bmi: 0.0393423370522607 smoking_status: 0
 stroke: 0 health_bills: 0.0393423370522607
```

Based on the result, **bmi** and **health_bills** have about 4% nan values.

One of the methods to deal with these value is to replace them with a statistic of that column.

For **bmi** we will use mean to replace missing values, because it has an aproximately normal distribution as its median and mean are close to each other.

In [ ]:

```
healthCare[c("bmi")][is.na(healthCare[c("bmi")])] <- mean(healthCare$
bmi, na.rm=TRUE)
```

For **health_bills** it seems that median can be a better statistic to replace nan values with.

In [ ]:

```
healthCare[c("health_bills")][is.na(healthCare[
  c("health_bills")])] <- median(healthCare$health_bills, na.rm=TRUE)
```

Question 1

In this question, we have to choose two categorical feature with more than two levels. Based on our dataset, only **Work Type** and **Smoking Status** satisfy this criteria. So, we choose them to do following tasks.

Before doing next parts, let take a look on the levels of these two features.

1. **work type**

   It defines the type of work a person does.

   - Private
   - Self-employed
   - Govt_job
   - children
   - Never_worked

1. **smoking status**

   It is the person smoking status.

   - formerly smoked
   - never smoked
   - smokes
   - Unknown

Part A

In this part, we have two choose a level from each feature to be able to compare their proportion by the methods that we have learnt so far. We choose **Govt_job** from **Work Type** and **smokes** from **Smoking Status**. We will take a sample for each categorical variable.

Firstly, we have to check conditions for inference:

Independence

- **within groups**

  Based on the documentation, data is gathered using random sampling technique.

  Sampling method that is used is without replacement. But, 300 (sample size) is less than 10 percent of population (500). Therefore, this condition is met.
- **between groups**

  Two groups must be independent of each other (non-paired). We will take samples without replacement to satisfy this condition.

Sample size/skew

We should at least 10 successes and 10 failures for each group. Now we can check this condition.

In [ ]:

```r
n1 <- 300
n2 <- 300
set.seed(123)
smokeRows <- sample(nrow(healthCare), n1)
smokeSample <- healthCare[smokeRows,]

workRows <- sample(nrow(healthCare[-smokeRows,]), n2)
workSample <- healthCare[workRows,]
```

In [ ]:

```r
p1 <- sum(smokeSample$smoking_status == "smokes") / n1
p2 <- sum(workSample$work_type == "Govt_job") / n2
```

In [ ]:

```r
(n1 * (1 - p1) >= 10 && n1 * p1 >= 10)
```

```
TRUE
```

In [ ]:

```r
(n2 * (1 - p2) >= 10 && n2 * p2 >= 10)
```

```
TRUE
```

Based on the result, sample size/skew condition is satisfied.

Now it is time to compute confidence interval for the difference between proportion of these two groups. At first, we have to compute standard error for the difference.

$$SE_{(\hat{p_1}-\hat{p_2})} = \sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}$$

In [ ]:

```
SE <- sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2)
SE
```

0.0289948909931337

The formula for calculating confidence interval is as follows:

$$(\hat{p_1}-\hat{p_2}) \pm z^* SE_{(\hat{p_1}-\hat{p_2})}$$

In [ ]:

```
diff <- p1 - p2

CI = 0.95
zStar <- qnorm((1 - CI) / 2, lower.tail = FALSE)
ME <- SE * zStar

low <- diff - ME
up <- diff + ME

print(paste("the 95% CI=(",low,"up to ",up,")"), quote = FALSE)
```

```
[1] the 95% CI=( -0.0268289420822068 up to  0.0868289420822068 )
```

We are 95% confident that the difference in the population proportion of individuals who have never work and the population proportion of individuals who smoke lies between -0.027 and 0.087.

Part B

In this part, we use chi-square test to check whether these groups are independent or not.

Let's check the conditions for this test.

Independence

- The sampling technique based on the documentation is random.
- The sample size is 300 which is less than 10 percent of the population.
- Each case only contributes to one cell in the table.

Sample size

Each cell must have at least 5 expected cases. We can check this condition by constructing contingency table.

In [ ]:

```r
smokingVsWorkType <- table(smokeSample$smoking_status, workSample$work_type)
smokingVsWorkType
```

```
                 children Govt_job Private Self-employed
  formerly smoked        5        9      33            14
  never smoked          21        9      55            18
  smokes                 5       13      24             7
  Unknown               12        9      52            14
```

In [ ]:

```r
chisq.test(smokingVsWorkType)$expected
```

A matrix: 4 × 4 of type dbl

	children	Govt_job	Private	Self-employed
formerly smoked	8.743333	8.133333	33.34667	10.776667
never smoked	14.763333	13.733333	56.30667	18.196667
smokes	7.023333	6.533333	26.78667	8.656667
Unknown	12.470000	11.600000	47.56000	15.370000

Based on the result, the value of each cell is at least 5. Therefore sample size condition for the test is met.

Now it is time to state our hypothesis for independence test.

$$H_0\,(\text{nothing going on}) : \text{Work type and smoking status are independent.}$$
$$H_A\,(\text{something going on}) : \text{Work type and smoking status are dependent.}$$

```
In [ ]: chisq.test(smokingVsWorkType)
```

```
	Pearson's Chi-squared test

data:  smokingVsWorkType
X-squared = 15.689, df = 9, p-value = 0.07367
```

$$p - value = 0.07367 > 0.05 \implies$$

Since p-value is greater than 0.05, we fail to reject null hypothesis. The data do not provide convincing evidence that work type and smoking status are dependent.

Question 2

In this question, we have to choose a binary categorical feature. Based on our dataset, one of the variables that satisfies this condition is **Ever Married**. Its levels are *Yes* and *No*. We consider *Yes* as success in this context.

Firstly, we take a random sample of size 12 from our data and keep the target column.

```
In [ ]: set.seed(123)
        sampleSize <- 12
        rows <- sample(1:nrow(healthCare), sampleSize)
        smallSample <- healthCare[rows, ]["ever_married"]
        pObserved <- sum(smallSample$ever_married == "Yes") / sampleSize
        pObserved
```

```
0.583333333333333
```

$$H_0 : p = 0.5$$
$$H_A : p > 0.5$$

Independence

- The sampling technique is random.
- The sample size is 12 which is less than 10 percent of the population.

Sample size / skew

12 × 0.5 = 6 → not met

distribution of sample proportions cannot be assumed to be nearly normal

Now we will take 3000 random samples from data and run our simulation to compute the p-value.

```
In [ ]: set.seed(123)
        simCount <- 3000
        nullSample <- c(rep(1, 6), rep(0, 6))
        simSamples <- replicate(simCount, sample(nullSample, size = sampleSiz
        e, replace = TRUE))
        proportions <- colSums(simSamples) / sampleSize
        pValue <- sum(proportions >= pObserved) / simCount
        pValue
```

```
0.384
```

$$p - value = 0.384 > 0.05 \implies$$

We fail to reject null hypothesis. Results from the simulations look like the data → the proportion of ever married was due to chance.

Question 3

Part A

In this part, we choose **Smoking Status** as a categorical variable which has 4 levels.

- formerly smoked
- never smoked
- smokes
- Unknown

Now we use `table()` method to get frequency of each level in dataset. Then, we divide it by n to get the probability distribution of this variable.

```
In [ ]: n <- length(healthCare$smoking_status)
        smokingDist <- table(healthCare$smoking_status) / n
        smokingDist
```

```
formerly smoked    never smoked          smokes         Unknown 
      0.1730280       0.3703269       0.1544334       0.3022118 
```

Bsed on the result we can see the probability distribution of **Smoking Status** in the whole dataset. The probability of each possible outcome is specified.

In this part, we have to compare the probability distribution of each sample with the original dataset. As a result, we should perform ***Goodness of Fit*** test. At first, we must check the conditions for this test.

Independence

- The sampling technique is random. (with or without bias)
- The sample size is 100 which is less than 10 percent of the population.
- Each case only contributes to one cell in the table.

Sample size

Each cell must have at least 5 expected cases. We can check this condition by constructing contingency table.

Random Sample

In this part, we randomly select 100 data point from our dataset.We have to select indices of these data points using `sample()` function. Then we will filter dataset by these indices and **Smoking Status** feature.

```
In [ ]: sampleSize <- 100
        set.seed(123)
        randomRows <- sample(nrow(healthCare), sampleSize)
        randomSample <- healthCare[randomRows,]["smoking_status"]
```

Here, by calling `table()` function on random sample we can see its frequency table.

```
In [ ]: randomDist <- table(randomSample$smoking_status)
        randomDist
```

```
formerly smoked    never smoked          smokes         Unknown 
             21              31              16              32 
```

The value in each cell is greater than 5. So, the sample size condition is met.

Now it is time to state our hypothesis for independence test.
$H_0\,\text{(nothing going on)} : \text{The random sample follows the same smoking status distribution in the p}$
$H_A\,\text{(something going on)} : \text{The random sample does not follow the same smoking status distribut}$

Let's perform our test by calling `chisq.test()` function on the random sample distribution and original distribution as probability.

```
In [ ]: chisq.test(randomDist, p=smokingDist)
```

```
	Chi-squared test for given probabilities

data:  randomDist
X-squared = 1.8975, df = 3, p-value = 0.5939
```

$$p - value = 0.5939 > 0.05 \implies$$

We fail to reject null hypothesis. The data do not provide convincing evidence that the random sample distribution differs from the original distribution.

Biased Sample

In this part, we will make our biased sample. For this goal, we select data point in a way that **never smoked** users have more chance to be in our sample.

In [ ]:

```
set.seed(123)
prob <- ifelse(healthCare$smoking_status=="never smoked", 0.7, 0.3)
biasedRows <- sample(nrow(healthCare), sampleSize, prob = prob)
biasedSample <- healthCare[biasedRows,]["smoking_status"]
```

In this part we will repreat the steps that we we have done for random sample.

In [ ]:

```
biasedDist <- table(biasedSample$smoking_status)
biasedDist
```

```
formerly smoked    never smoked         smokes          Unknown
             14              59             11               16
```

The value in each cell is greater than 5. So, the sample size condition is met.

Now it is time to state our hypothesis for independence test.
$H_0\,(\text{nothing going on}) : \text{The biased sample follows the same smoking status distribution in the po}$
$H_A\,(\text{something going on}) : \text{The biased sample does not follow the same smoking status distributio}$

In [ ]:

```
chisq.test(biasedDist, p=smokingDist)
```

```
	Chi-squared test for given probabilities

data:  biasedDist
X-squared = 21.632, df = 3, p-value = 7.782e-05
```

$$p - value = 7.782e - 05 < 0.05 \implies$$

We reject null hypothesis. The data provide convincing evidence that the biased sample distribution differs from the original distribution.

Part B

In this part, we choose **Ever married** as the second categorical variable. We use chi-square test to check whether these groups are independent or not.

We take two non-paired samples of size 200 for our test to meet independence condition.

In [ ]:

```
nSmoking <- 200
nMarriage <- 200
set.seed(123)
smokingRows <- sample(nrow(healthCare), nSmoking)
smokingSample <- healthCare[smokingRows,]

marriageRows <- sample(nrow(healthCare[-smokingRows,]), nMarriage)
marriageSample <- healthCare[marriageRows,]
```

Let's check the conditions for this test.

Independence

- The sampling technique based on the documentation is random.
- The sample size is 200 which is less than 10 percent of the population.
- Each case only contributes to one cell in the table.

Sample size

Each cell must have at least 5 expected cases. We can check this condition by constructing contingency table.

In [ ]:

```
marriageVsSmoking <- table(marriageSample$ever_married, smokingSample
$smoking_status)
marriageVsSmoking
```

```
     formerly smoked never smoked smokes Unknown
  No              17           30      8      21
  Yes             26           38     23      37
```

In [ ]:

```
chisq.test(marriageVsSmoking)$expected
```

A matrix: 2 × 4 of type dbl

	formerly smoked	never smoked	smokes	Unknown
No	16.34	25.84	11.78	22.04
Yes	26.66	42.16	19.22	35.96

Based on the result, the value of each cell is at least 5. Therefore sample size condition for the test is met.

Now it is time to state our hypothesis for independence test.

$$H_0\,(\text{nothing going on}) : \text{Ever married and smoking status are independent.}$$
$$H_A\,(\text{something going on}) : \text{Ever married and smoking status are dependent.}$$

```
In [ ]: chisq.test(marriageVsSmoking)
```

```
	Pearson's Chi-squared test

data:  marriageVsSmoking
X-squared = 3.1587, df = 3, p-value = 0.3678
```

$$p - value = 0.3678 > 0.05 \implies$$

Since p-value is greater than 0.05, we fail to reject null hypothesis. The data do not provide convincing evidence that ever married and smoking status are dependent.

Question 4

In this question, I select **Health Bills** as a response variable, because I believe that this is one of the important factors in our financial decisions.

I select **BMI** and **Age** as exaplanatory variables for our model. Based on the results of *Phase 1* we concluded that there is an association between **BMI** and our response variable. Moreover, it is undeniable that the older we get, the more health bills we should pay.

Part A

In *Phase 1* we saw that **BMI** is the most associated feature with our response variable in our dataset. As a result, I guess that it would be better predictor in our model.

Part B

In this part, we will discuss questions a to b for *BMI* and *Age* respectively.

BMI

a

Least Squares Regression

In this question, we use `lm()` method in order to compute linear regression model for the response variable using *BMI* as the only explanatory variable.

In [ ]:

```
bmiModel <- lm(health_bills ~ bmi, data = healthCare)
summary(bmiModel)
```

```
Call:
lm(formula = health_bills ~ bmi, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3232.0  -429.8  -102.3   258.3  5872.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1852.434     39.838   46.50   <2e-16 ***
bmi           44.373      1.332   33.31   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 733 on 5107 degrees of freedom
Multiple R-squared:  0.1785,	Adjusted R-squared:  0.1783 
F-statistic:  1109 on 1 and 5107 DF,  p-value: < 2.2e-16
```

Based on the result, p-value for this predictor is about 0 that is less than 0.05. Therefore we can conclude that it is a good predictor for the response variable.

b

Predictive Equation

The predictive equation for this linear model is as follows:

$$healthBills = 1856.626 + 44.373 \times bmi$$

Intercept

When *BMI = 0*, *Health Bills* is expected to equal 1856.626. In this model, having bmi = 0 is somehow meaningless.

Slope

For each unit increase in *BMI*, *Health Bills* is expected to be higher on average by 44.373.

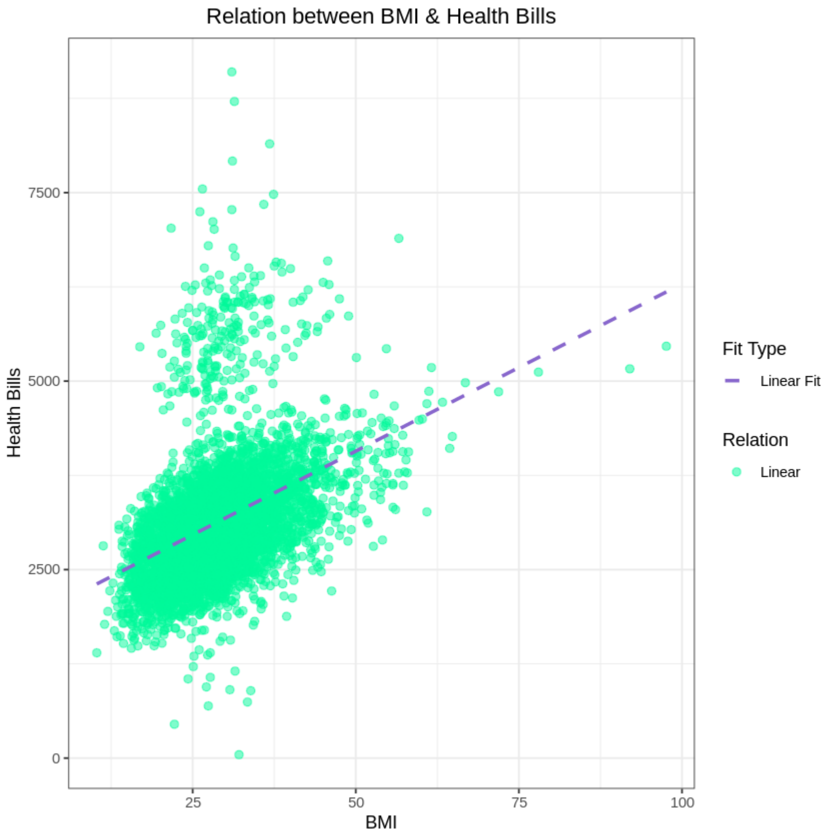
c

Scatter Plot

In this question, we use `geom_point()` method to show data points in our scatter plot. Moreover, we use `stat_smooth()` method to draw the least-squares fit that we obtained in a and b.

In [ ]:

```
bmiScatter <- ggplot(healthCare, aes(x = bmi))
bmiScatter <- bmiScatter + geom_point(aes(y = health_bills, color = "Linear"), size = 2, alpha = 0.5)
bmiScatter <- bmiScatter + stat_smooth(aes(x = bmi, y = health_bills, linetype = "Linear Fit"),
              method = "lm", formula = y ~ x, se = F, color = "mediumpurple3")
bmiScatter <- bmiScatter + scale_color_manual(name = "Relation", values = c("mediumspringgreen", "thistle1"))
bmiScatter <- bmiScatter + scale_linetype_manual(name = "Fit Type", values = c(2, 2))
bmiScatter <- bmiScatter + xlab("BMI")
bmiScatter <- bmiScatter + ylab("Health Bills")
bmiScatter <- bmiScatter + ggtitle("Relation between BMI & Health Bills")
bmiScatter <- bmiScatter + theme_bw()
bmiScatter <- bmiScatter + theme(plot.title = element_text(hjust = 0.5))
bmiScatter
```



Age

a

Least Squares Regression

In this question, we use the same code as what we explained for *BMI* to reach the linear model for this new predictor.

In [ ]:

```
ageModel <- lm(health_bills ~ age, data = healthCare)
summary(ageModel)
```

```
Call:
lm(formula = health_bills ~ age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3213.6  -465.1   -90.5   338.2  5589.6 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2679.7026    23.3308   114.9   <2e-16 ***
age           10.5222     0.4782    22.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 772.9 on 5107 degrees of freedom
Multiple R-squared:  0.08659,	Adjusted R-squared:  0.08641 
F-statistic: 484.1 on 1 and 5107 DF,  p-value: < 2.2e-16
```

Based on the result, p-value for this predictor is about 0 that is less than 0.05. Therefore we can conclude that it is a good predictor for the response variable.

b

Predictive Equation

The predictive equation for this linear model is as follows:

$$healthBills = 2663.58 + 11.0847 \times age$$

Intercept

When *Age = 0*, *Health Bills* is expected to equal 2663.58. **Slope**

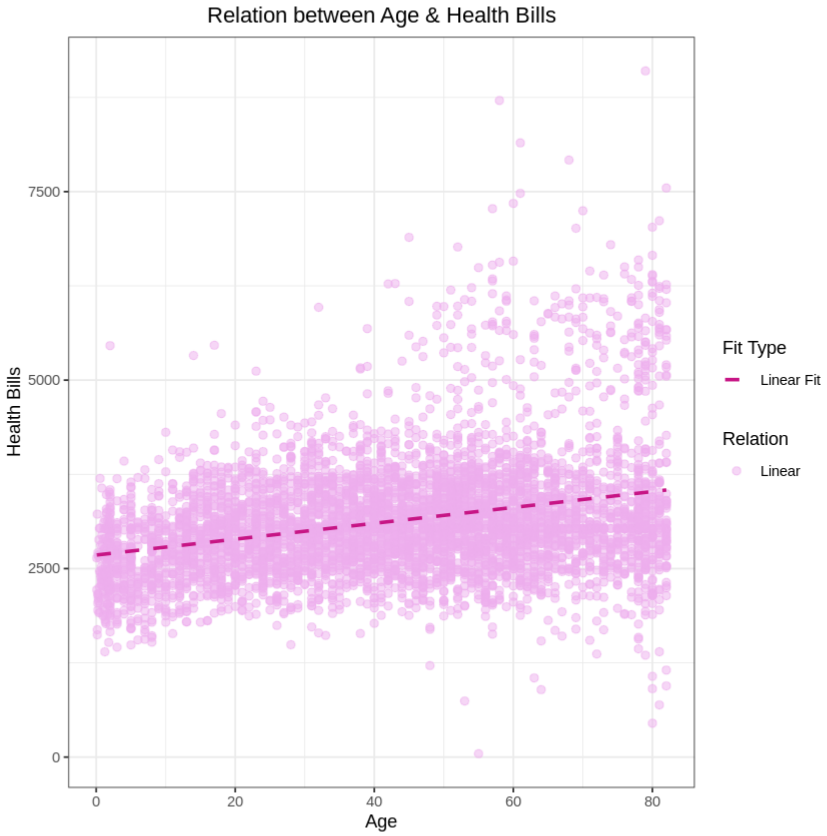
For each unit increase in *Age*, *Health Bills* is expected to be higher on average by 11.0847.

c

Scatter Plot

In this question, again we use the same code as what we explained for *BMI* to draw scatter plot and fit line for the linear model.

In [ ]:
```
ageScatter <- ggplot(healthCare, aes(x = age))
ageScatter <- ageScatter + geom_point(aes(y = health_bills, color =
"Linear"), size = 2, alpha = 0.5)
ageScatter <- ageScatter + stat_smooth(aes(x = age, y = health_bills,
linetype = "Linear Fit"),
              method = "lm", formula = y ~ x, se = F, color = "medium
violetred")
ageScatter <- ageScatter + scale_color_manual(name = "Relation", valu
es = c("plum2", "thistle1"))
ageScatter <- ageScatter + scale_linetype_manual(name = "Fit Type", v
alues = c(2, 2))
ageScatter <- ageScatter + xlab("Age")
ageScatter <- ageScatter + ylab("Health Bills")
ageScatter <- ageScatter + ggtitle("Relation between Age & Health Bil
ls")
ageScatter <- ageScatter + theme_bw()
ageScatter <- ageScatter + theme(plot.title = element_text(hjust = 0.
5))
ageScatter
```



Part C

One of the important metrics in comparing two predictors is p-value. For both variables this value is less than 2.2e-16. The lower the p-value is, The better the predictor will be. The parameter is identical for these two variables. Therefore we can not compare them in this way.

Another thing that we can consider is variability of data points around the least squares line. It should be constant. As we can see, vriability in *BMI* scatter plot is less than *Age*. So, it seems that *BMI* is better predictor than *Age*.

Part D

Adjusted R-squared

One of the important metrics in comparing two predictors is the value of adjusted R-squared. Based on the results of the previous part, this metric for *BMI* is 0.1784 and for *Age* is 0.09172. The greater the adjusted R-squared is, the better the predictor is. As a result, **BMI** is better predictor than *Age*.

ANOVA table We have following formula to compute adjusted R-squared from ANOVA table.

$$R^2_{adj} = 1 - (\frac{SSE}{SST} \times \frac{n-1}{n-k-1})$$

k: number of predictors

At first, we make ANOVA table for *BMI* predictor. For this goal, we will use `anova()` method.

In [ ]:

```
bmiANOVA <- anova(bmiModel)
bmiANOVA
```

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
bmi	1	596037498	596037497.5	1109.321	2.72593e-220
Residuals	5107	2743987063	537299.2	NA	NA

In this step, we will write a method to compute adjusted R-squared from ANOVA table based on the above formula.

In [ ]:
```
getAdjR2 <- function(anovaTable, pred) {
  n <- length(healthCare[pred])
  SSR <- anovaTable[pred, "Sum Sq"]
  SSE <- anovaTable["Residuals", "Sum Sq"]
  SST <- SSE + SSR
  adjR2 <- 1 - (SSE / SST * (5109 - 1) / (5109 - 1 - 1))
  return(adjR2)
}
```

Now we call this function for *BMI* model.

In [ ]:
```
getAdjR2(bmiANOVA, "bmi")
```

```
0.178292161469316
```

Based on the result, computed adjusted R-squared is as same as what we get by calling `lm()` method.

Now will repeat these tasks for *Age* as a predictor.

In [ ]:
```
ageANOVA <- anova(ageModel)
ageANOVA
```

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
age	1	289202015	289202015.3	484.1169	1.387186e-102
Residuals	5107	3050822546	597380.6	NA	NA

In [ ]:
```
getAdjR2(ageANOVA, "age")
```

```
0.086407937853144
```

As we can see, thse is no difference between this value and the value obtained from `lm()` .

To wrap up, *BMI* is a better predictor than *Age*.

Part E

Based on the results of previous part, I made a list of features of a good predictorl.

- It cause the model to have lower p-value.
- Variability of data points around its least squares line is fewer.
- It increases the adjusted R-squared of the model.

Part F

a

In [ ]:

```r
nSample <- 100
nTrain <- 90
set.seed(123)
sampleRows <- sample(nrow(healthCare), nSample)
sampleData <- healthCare[sampleRows,]
```

In [ ]:

```r
# Split data into train and test
index <- createDataPartition(sampleData$health_bills, p = .90, list =
FALSE)
train <- sampleData[index, ]
test <- sampleData[-index, ]
```

In [ ]:

```
bmiSampleModel <- lm(health_bills ~ bmi, data = train)
summary(bmiSampleModel)
```

```
Call:
lm(formula = health_bills ~ bmi, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-1060.9  -427.2   -48.7   354.2  3349.3 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2228.30     242.38   9.194 1.36e-14 ***
bmi            28.41       8.04   3.534 0.000649 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 664.2 on 90 degrees of freedom
Multiple R-squared:  0.1218,	Adjusted R-squared:  0.1121 
F-statistic: 12.49 on 1 and 90 DF,  p-value: 0.0006492
```

In [ ]:

```
ageSampleModel <- lm(health_bills ~ age, data = train)
summary(ageSampleModel)
```

```
Call:
lm(formula = health_bills ~ age, data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-1331.58  -427.62   -52.61   336.63  2883.09 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2630.800    158.497  16.598  < 2e-16 ***
age            9.521      3.230   2.947  0.00408 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 676.8 on 90 degrees of freedom
Multiple R-squared:  0.08802,	Adjusted R-squared:  0.07789 
F-statistic: 8.687 on 1 and 90 DF,  p-value: 0.004082
```

b

In [ ]:

```
confint(bmiSampleModel)
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	1746.77759	2709.82396
bmi	12.43735	44.38155

In [ ]:

```
confint(ageSampleModel)
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	2315.918807	2945.68110
age	3.103285	15.93904

c

In [259]:

```
bmiModelPreds <- predict(bmiSampleModel, newdata = test, type=c("response"))
```

d

In [264]:

```
summary(healthCare$health_bills)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   44.8  2647.8  3032.2  3134.6  3454.9  9100.5 
```

In [260]:

```
actual <- test$health_bills
```

In [263]:

```
abs(bmiModelPreds - actual)
```

526: 818.180086602123 **2986:** 1134.55799793978 **2980:** 485.222855429379 **555:** 52.8026754752932 **277:** 325.826375713676 **1006:** 367.376624469863 **2339:** 361.751479787056 **4262:** 468.77067579832

We consider for difference the threshold of 600 and it is obtained from what we can see in the summary. So, the success rate would be 70% which is above 50% and it seems that we have a good model.

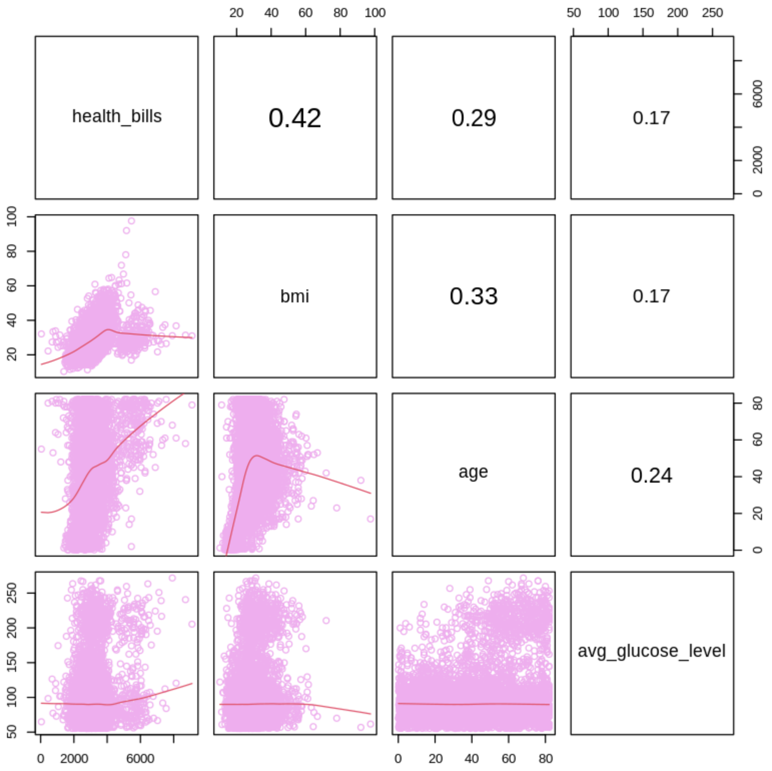
Question 5

Part A

In this part, we will plot correlogram for numerical variables. In first figure, we used `pairs()` method and in second figure, we we used ggcorrplot() from ggcorrplot library to plot to create a heatmap correlogram from our features. Also, we used cor_pmat() method to compute matrix of p-value. Finally, we set the significance level to 0.05.

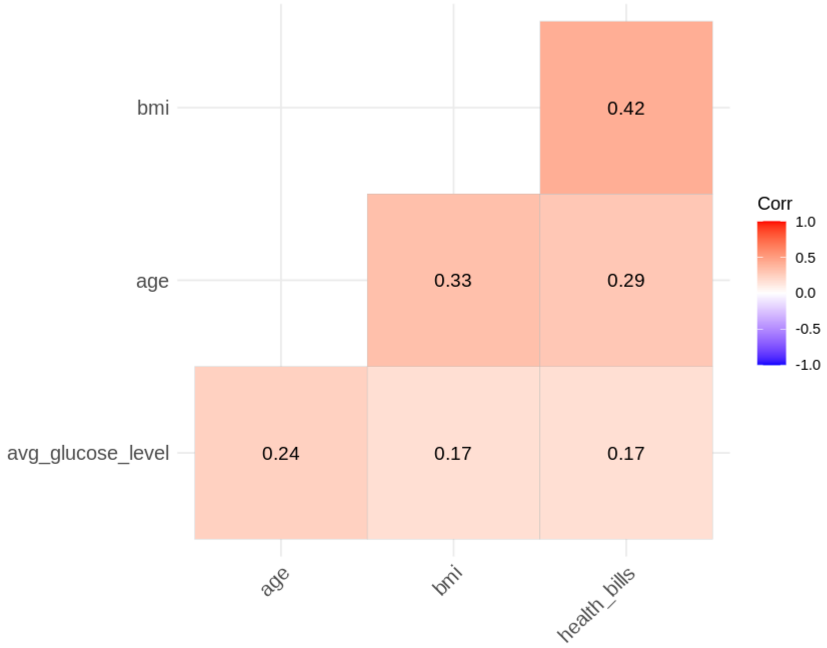
In [ ]:
```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    Cor <- abs(cor(x, y))
    txt <- paste0(prefix, format(c(Cor, 0.123456789), digits = digits
)[1])
    if(missing(cex.cor)) {
        cex.cor <- 0.4 / strwidth(txt)
    }
    text(0.5, 0.5, txt,
         cex = 1 + cex.cor * Cor)
}

pairs(health_bills ~ bmi + age + avg_glucose_level, data=healthCare,
      upper.panel = panel.cor,
      lower.panel = panel.smooth, col="plum2")
```



In [ ]:

```r
numericVars <- healthCare[c("age", "avg_glucose_level", "bmi", "health_bills")]
corr <- cor(numericVars)
p.mat <- cor_pmat(numericVars)
ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE, p.mat = p.mat, sig.level = 0.05)
```



Based on the results, the highest correlation is between BMI and Health Bills. Is means that BMI is the most effective predictor for this value. Also, it would be of importance to consider Age as a predictor. Because it has a high correlation with our response variable.

Part B

In this part, we consider *BMI* and *Age* as predictors in our multiple linear regression model.

In [ ]:

```
selectedModel <- lm(health_bills ~ bmi + age, data = healthCare)
summary(selectedModel)
```

```
Call:
lm(formula = health_bills ~ bmi + age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3286.5  -421.1   -80.5   281.5  5661.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1754.8765    39.8535   44.03   <2e-16 ***
bmi           38.3761     1.3856   27.70   <2e-16 ***
age            6.2651     0.4717   13.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 720.7 on 5106 degrees of freedom
Multiple R-squared:  0.2059,	Adjusted R-squared:  0.2056 
F-statistic: 661.9 on 2 and 5106 DF,  p-value: < 2.2e-16
```

Part C

Based on the results of previous parts, R-squared is 0.2059. Therefore we can conclude that 20.59% of variability in health bills is explained by our model.

Part D

It is a good model. Because its p-value is too small and we can conclude that our model is statistcally significant. Its adjsusted R-squared is approriate but there is a potential to reach higher value by considering more predictors.

Part E

Backwards Elimination - Adjusted R-squared

In this method, we start with the full model. At each step, we drop one variable at a time and record adjusted R-squared of each smaller model. Then, we pick the model with the highest increase in adjusted R-squared.

We repeat until none of the models yield an increase in adjusted R-squared.

```
In [ ]: full <- lm(health_bills ~ bmi + age + avg_glucose_level, data = healt
        hCare)
        summary(full)
```

```
Call:
lm(formula = health_bills ~ bmi + age + avg_glucose_level, data = hea
lthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3225.0  -422.1   -76.8   286.2  5554.4 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1663.8185    42.9576  38.732  < 2e-16 ***
bmi                 37.6063     1.3884  27.086  < 2e-16 ***
age                  5.7387     0.4797  11.964  < 2e-16 ***
avg_glucose_level    1.2819     0.2298   5.579 2.54e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 718.6 on 5105 degrees of freedom
Multiple R-squared:  0.2107,	Adjusted R-squared:  0.2102 
F-statistic: 454.3 on 3 and 5105 DF,  p-value: < 2.2e-16
```

We can see that adjusted R-squared for the full model is 0.2102.

Step 1

```
In [ ]: step1dropBMI <- lm(health_bills ~ age + avg_glucose_level, data = hea
        lthCare)
        summary(step1dropBMI)
```

```
Call:
lm(formula = health_bills ~ age + avg_glucose_level, data = healthCar
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-3124.6  -475.7   -82.4   342.7  5433.5 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2517.2050    31.2260  80.612  < 2e-16 ***
age                  9.6152     0.4896  19.640  < 2e-16 ***
avg_glucose_level    1.9004     0.2445   7.773 9.18e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 768.4 on 5106 degrees of freedom
Multiple R-squared:  0.09727,	Adjusted R-squared:  0.09692 
F-statistic: 275.1 on 2 and 5106 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1dropAge <- lm(health_bills ~ bmi + avg_glucose_level, data = hea
        lthCare)
        summary(step1dropAge)
```

```
Call:
lm(formula = health_bills ~ bmi + avg_glucose_level, data = healthCar
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-3151.1  -425.4   -79.4   265.0  5695.6 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1711.3171    43.3650   39.46  < 2e-16 ***
bmi                 42.5622     1.3435   31.68  < 2e-16 ***
avg_glucose_level    1.8225     0.2284    7.98 1.79e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 728.5 on 5106 degrees of freedom
Multiple R-squared:  0.1886,	Adjusted R-squared:  0.1883 
F-statistic: 593.3 on 2 and 5106 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1dropGlucose <- lm(health_bills ~ bmi + age, data = healthCare)
        summary(step1dropGlucose)
```

```
Call:
lm(formula = health_bills ~ bmi + age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3286.5  -421.1   -80.5   281.5  5661.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1754.8765    39.8535   44.03   <2e-16 ***
bmi           38.3761     1.3856   27.70   <2e-16 ***
age            6.2651     0.4717   13.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 720.7 on 5106 degrees of freedom
Multiple R-squared:  0.2059,	Adjusted R-squared:  0.2056 
F-statistic: 661.9 on 2 and 5106 DF,  p-value: < 2.2e-16
```

Based on the result, none of the above models yield an increase in adjusted R-squared. So, it means that full model is the best model in this method.

Backwards Elimination - p-value

In this method, we start with the full model. At each step we drop the variable with the highest p-value and refit a smaller model.

We repeat until all variables left in the model are significant.

```
In [ ]: full <- lm(health_bills ~ bmi + age + avg_glucose_level, data = healt
        hCare)
        summary(full)
```

```
Call:
lm(formula = health_bills ~ bmi + age + avg_glucose_level, data = hea
lthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3225.0  -422.1   -76.8   286.2  5554.4 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1663.8185    42.9576  38.732  < 2e-16 ***
bmi                 37.6063     1.3884  27.086  < 2e-16 ***
age                  5.7387     0.4797  11.964  < 2e-16 ***
avg_glucose_level    1.2819     0.2298   5.579 2.54e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 718.6 on 5105 degrees of freedom
Multiple R-squared:  0.2107,    Adjusted R-squared:  0.2102 
F-statistic: 454.3 on 3 and 5105 DF,  p-value: < 2.2e-16
```

Based on the result, all the predictors in our full model are significant. So, we report this model as the best model that can be achieved by this method.

Forward Selection - Adjusted R-squared

In this method, we start with single predictor regressions of response vs. each explanatory variable. we pick the model with the highest adjusted R-squared, add the remaining variables one at a time to the existing model, and pick the model with the highest adjusted R-squared.

we repeat until the addition of any of the remaining variables does not result in a higher adjusted R-squared.

Step 1

```
In [ ]: step1selectBMI <- lm(health_bills ~ bmi, data = healthCare)
        summary(step1selectBMI)
```

```
Call:
lm(formula = health_bills ~ bmi, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3232.0  -429.8  -102.3   258.3  5872.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1852.434     39.838   46.50   <2e-16 ***
bmi           44.373      1.332   33.31   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 733 on 5107 degrees of freedom
Multiple R-squared:  0.1785,	Adjusted R-squared:  0.1783 
F-statistic:  1109 on 1 and 5107 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1selectAge <- lm(health_bills ~ age, data = healthCare)
        summary(step1selectAge)
```

```
Call:
lm(formula = health_bills ~ age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3213.6  -465.1   -90.5   338.2  5589.6 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2679.7026    23.3308   114.9   <2e-16 ***
age           10.5222     0.4782    22.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 772.9 on 5107 degrees of freedom
Multiple R-squared:  0.08659,	Adjusted R-squared:  0.08641 
F-statistic: 484.1 on 1 and 5107 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1selectGlucose <- lm(health_bills ~ avg_glucose_level, data = hea
        lthCare)
        summary(step1selectGlucose)
```

```
Call:
lm(formula = health_bills ~ avg_glucose_level, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-2964.3  -480.4   -98.0   329.3  5664.0 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2811.4146    28.4115   98.95   <2e-16 ***
avg_glucose_level    3.0447     0.2462   12.37   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 796.9 on 5107 degrees of freedom
Multiple R-squared:  0.02907,	Adjusted R-squared:  0.02888 
F-statistic: 152.9 on 1 and 5107 DF,  p-value: < 2.2e-16
```

Based on what we can see, bmi is the predictor with the highest adjusted R-squared. So, we select it for this step.

Step 2

```
In [ ]: step2selectAge <- lm(health_bills ~ bmi + age, data = healthCare)
        summary(step2selectAge)
```

```
Call:
lm(formula = health_bills ~ bmi + age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3286.5  -421.1   -80.5   281.5  5661.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1754.8765    39.8535   44.03   <2e-16 ***
bmi           38.3761     1.3856   27.70   <2e-16 ***
age            6.2651     0.4717   13.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 720.7 on 5106 degrees of freedom
Multiple R-squared:  0.2059,	Adjusted R-squared:  0.2056 
F-statistic: 661.9 on 2 and 5106 DF,  p-value: < 2.2e-16
```

```
In [ ]: step2selectGlucose <- lm(health_bills ~ bmi + avg_glucose_level, data
        = healthCare)
        summary(step2selectGlucose)
```

```
Call:
lm(formula = health_bills ~ bmi + avg_glucose_level, data = healthCar
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-3151.1  -425.4   -79.4   265.0  5695.6 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1711.3171    43.3650   39.46  < 2e-16 ***
bmi                 42.5622     1.3435   31.68  < 2e-16 ***
avg_glucose_level    1.8225     0.2284    7.98 1.79e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 728.5 on 5106 degrees of freedom
Multiple R-squared:  0.1886,	Adjusted R-squared:  0.1883 
F-statistic: 593.3 on 2 and 5106 DF,  p-value: < 2.2e-16
```

Based on what we can see, by adding age we can reach the highest adjusted R-squared. So, we select it for this step.

Step 3

```
In [ ]: step3selectGlucose <- lm(health_bills ~ bmi + age + avg_glucose_level
        , data = healthCare)
        summary(step3selectGlucose)
```

```
Call:
lm(formula = health_bills ~ bmi + age + avg_glucose_level, data = hea
lthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3225.0  -422.1   -76.8   286.2  5554.4 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1663.8185    42.9576  38.732  < 2e-16 ***
bmi                 37.6063     1.3884  27.086  < 2e-16 ***
age                  5.7387     0.4797  11.964  < 2e-16 ***
avg_glucose_level    1.2819     0.2298   5.579 2.54e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 718.6 on 5105 degrees of freedom
Multiple R-squared:  0.2107,	Adjusted R-squared:  0.2102 
F-statistic: 454.3 on 3 and 5105 DF,  p-value: < 2.2e-16
```

Based on what we can see, by adding glucose we can reach greater adjusted R-squared in comparison to last step. So, we select it for this step.

As you see, we reach our full model. It means that this is the best model than develop from our data.

Forward Selection - p-value

In this method, we start with single predictor regressions of response vs. each explanatory variable, pick the variable with the lowest significant p-value. we add the remaining variables one at a time to the existing model, and pick the variable with the lowest significant p-value.

We repeat until any of the remaining variables do not have a significant p-value.

Step 1

In [ ]:

```
step1selectBMI <- lm(health_bills ~ bmi, data = healthCare)
summary(step1selectBMI)
```

```
Call:
lm(formula = health_bills ~ bmi, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3232.0  -429.8  -102.3   258.3  5872.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1852.434     39.838   46.50   <2e-16 ***
bmi           44.373      1.332   33.31   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 733 on 5107 degrees of freedom
Multiple R-squared:  0.1785,    Adjusted R-squared:  0.1783 
F-statistic:  1109 on 1 and 5107 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1selectAge <- lm(health_bills ~ age, data = healthCare)
        summary(step1selectAge)
```

```
Call:
lm(formula = health_bills ~ age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3213.6  -465.1   -90.5   338.2  5589.6 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2679.7026    23.3308   114.9   <2e-16 ***
age           10.5222     0.4782    22.0   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 772.9 on 5107 degrees of freedom
Multiple R-squared:  0.08659,	Adjusted R-squared:  0.08641 
F-statistic: 484.1 on 1 and 5107 DF,  p-value: < 2.2e-16
```

```
In [ ]: step1selectGlucose <- lm(health_bills ~ avg_glucose_level, data = healthCare)
        summary(step1selectGlucose)
```

```
Call:
lm(formula = health_bills ~ avg_glucose_level, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-2964.3  -480.4   -98.0   329.3  5664.0 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2811.4146    28.4115   98.95   <2e-16 ***
avg_glucose_level    3.0447     0.2462   12.37   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 796.9 on 5107 degrees of freedom
Multiple R-squared:  0.02907,	Adjusted R-squared:  0.02888 
F-statistic: 152.9 on 1 and 5107 DF,  p-value: < 2.2e-16
```

Based on the resut, p-value for all the predictors is less than 2.2e-16. So, there is no difference and we can choose one of them for this step. I want to choose bmi.

Step 2

```
In [ ]: step2selectAge <- lm(health_bills ~ bmi + age, data = healthCare)
        summary(step2selectAge)
```

```
Call:
lm(formula = health_bills ~ bmi + age, data = healthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3286.5  -421.1   -80.5   281.5  5661.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1754.8765    39.8535   44.03   <2e-16 ***
bmi           38.3761     1.3856   27.70   <2e-16 ***
age            6.2651     0.4717   13.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 720.7 on 5106 degrees of freedom
Multiple R-squared:  0.2059,	Adjusted R-squared:  0.2056 
F-statistic: 661.9 on 2 and 5106 DF,  p-value: < 2.2e-16
```

```
In [ ]: step2selectGlucose <- lm(health_bills ~ bmi + avg_glucose_level, data
        = healthCare)
        summary(step2selectGlucose)
```

```
Call:
lm(formula = health_bills ~ bmi + avg_glucose_level, data = healthCar
e)

Residuals:
    Min      1Q  Median      3Q     Max 
-3151.1  -425.4   -79.4   265.0  5695.6 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1711.3171    43.3650   39.46  < 2e-16 ***
bmi                 42.5622     1.3435   31.68  < 2e-16 ***
avg_glucose_level    1.8225     0.2284    7.98 1.79e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 728.5 on 5106 degrees of freedom
Multiple R-squared:  0.1886,	Adjusted R-squared:  0.1883 
F-statistic: 593.3 on 2 and 5106 DF,  p-value: < 2.2e-16
```

Based on what we can see that age has the lowest p-value. So, we choose it for this step.

Step 3

In [ ]:

```
step3selectGlucose <- lm(health_bills ~ bmi + age + avg_glucose_level
, data = healthCare)
summary(step3selectGlucose)
```

```
Call:
lm(formula = health_bills ~ bmi + age + avg_glucose_level, data = hea
lthCare)

Residuals:
    Min      1Q  Median      3Q     Max 
-3225.0  -422.1   -76.8   286.2  5554.4 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1663.8185    42.9576  38.732  < 2e-16 ***
bmi                 37.6063     1.3884  27.086  < 2e-16 ***
age                  5.7387     0.4797  11.964  < 2e-16 ***
avg_glucose_level    1.2819     0.2298   5.579 2.54e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 718.6 on 5105 degrees of freedom
Multiple R-squared:  0.2107,	Adjusted R-squared:  0.2102 
F-statistic: 454.3 on 3 and 5105 DF,  p-value: < 2.2e-16
```

Based on the result, glucose has a significant p-value. So, we add it to our model. This was the last predictor. So, we reached our best model which is the full model.

Conclusion

Our full model (with numerical predictors only) is the best model that we can develop to predict health bills.

In [ ]:

```
best <- full
```

Part F

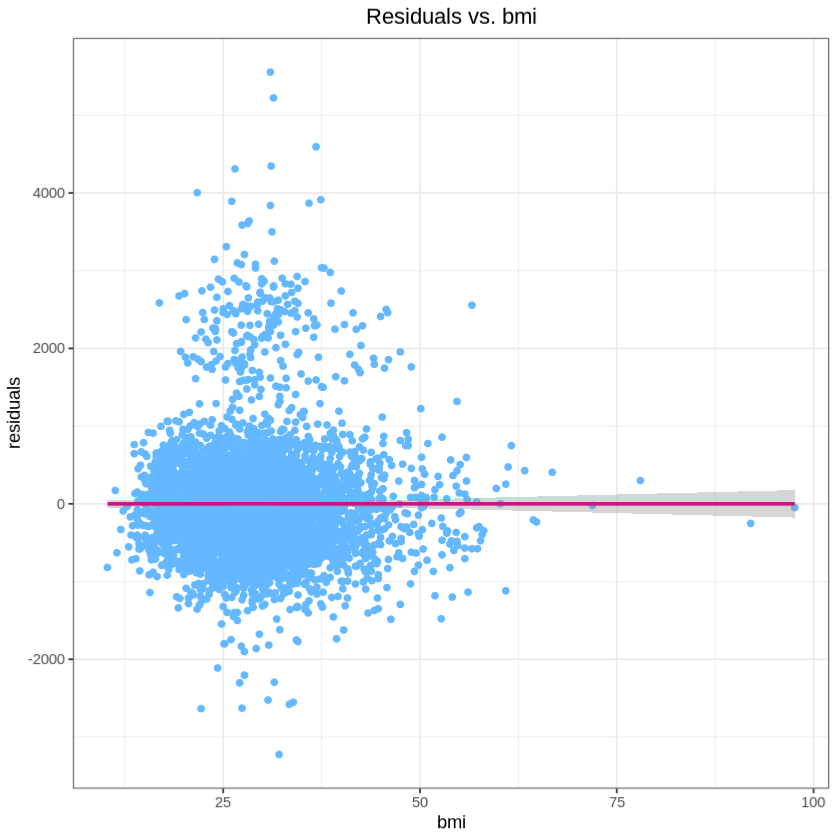
Linearity

Each explanatory variable should be linearly related to the response variable. We will check this condition using residuals plots. We are looking for a random scatter around 0.

```
In [ ]: data <- data.frame(residuals=best$residuals, bmi=healthCare$bmi)

        bmiRes <- ggplot(data = data, aes(bmi, residuals))
        bmiRes <- bmiRes + geom_point(color = "steelblue1")
        bmiRes <- bmiRes + stat_smooth(method = lm, color="mediumvioletred")
        bmiRes <- bmiRes + ggtitle("Residuals vs. bmi")
        bmiRes <- bmiRes + theme_bw()
        bmiRes <- bmiRes + theme(plot.title = element_text(hjust = 0.5))
        bmiRes
```

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

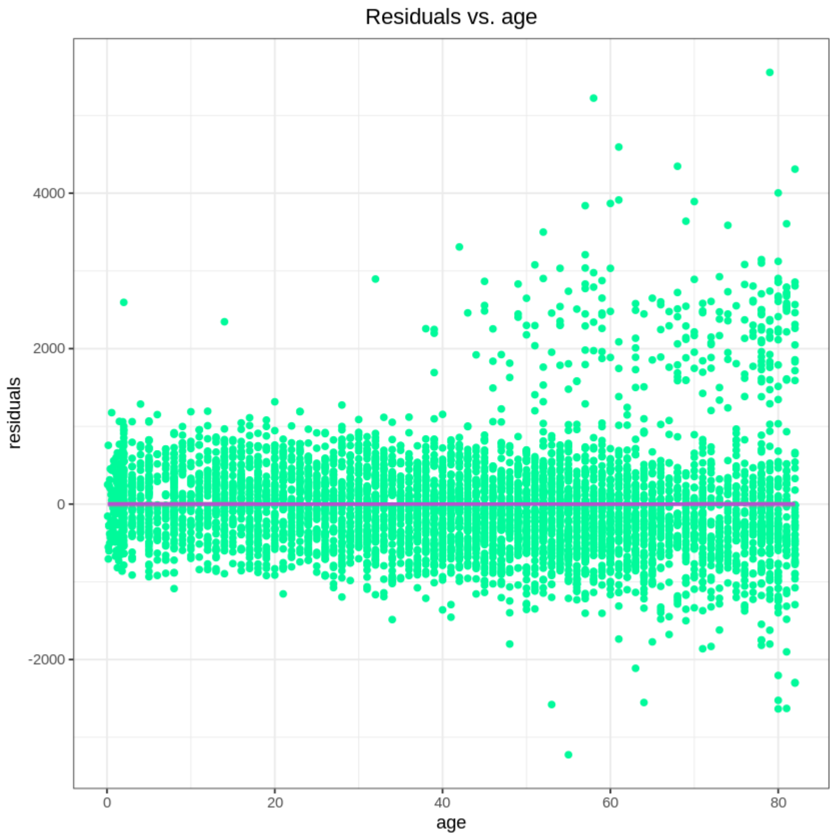


In [ ]:

```
data <- data.frame(residuals=best$residuals, age=healthCare$age)

ageRes <- ggplot(data = data, aes(age, residuals))
ageRes <- ageRes + geom_point(color = "mediumspringgreen")
ageRes <- ageRes + stat_smooth(method = lm, color="mediumorchid3")
ageRes <- ageRes + ggtitle("Residuals vs. age")
ageRes <- ageRes + theme_bw()
ageRes <- ageRes + theme(plot.title = element_text(hjust = 0.5))
ageRes
```

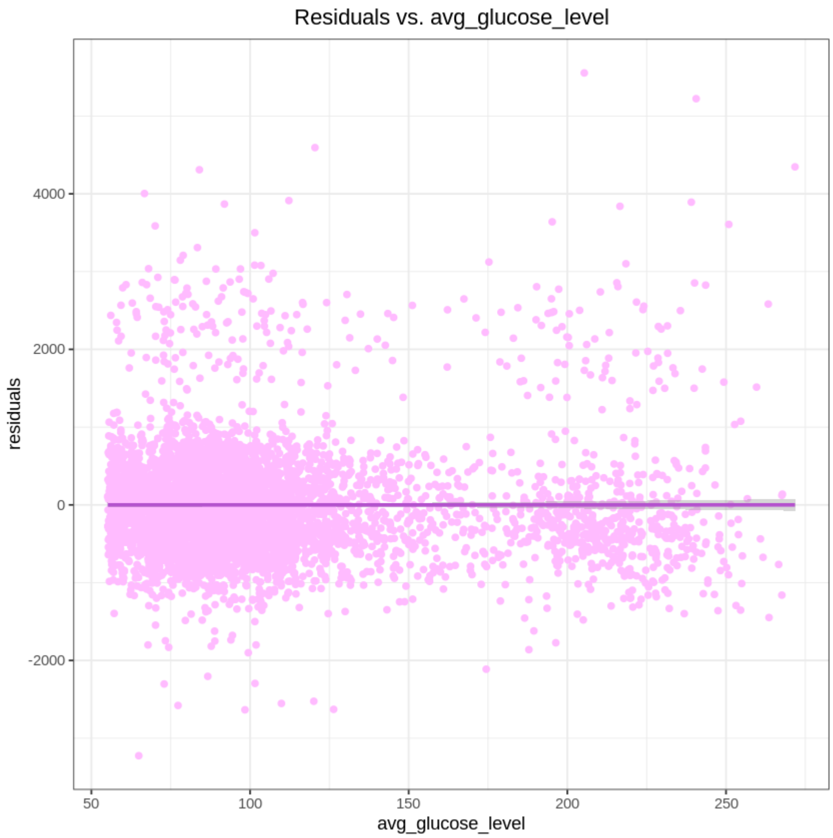
```
`geom_smooth()` using formula 'y ~ x'
```



```
In [ ]: data <- data.frame(residuals=best$residuals, avg_glucose_level=healthCare$avg_glucose_level)

glucoseRes <- ggplot(data = data, aes(avg_glucose_level, residuals))
glucoseRes <- glucoseRes + geom_point(color = "plum1")
glucoseRes <- glucoseRes + stat_smooth(method = lm, color="mediumorchid3")
glucoseRes <- glucoseRes + ggtitle("Residuals vs. avg_glucose_level")
glucoseRes <- glucoseRes + theme_bw()
glucoseRes <- glucoseRes + theme(plot.title = element_text(hjust = 0.5))
glucoseRes
```

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



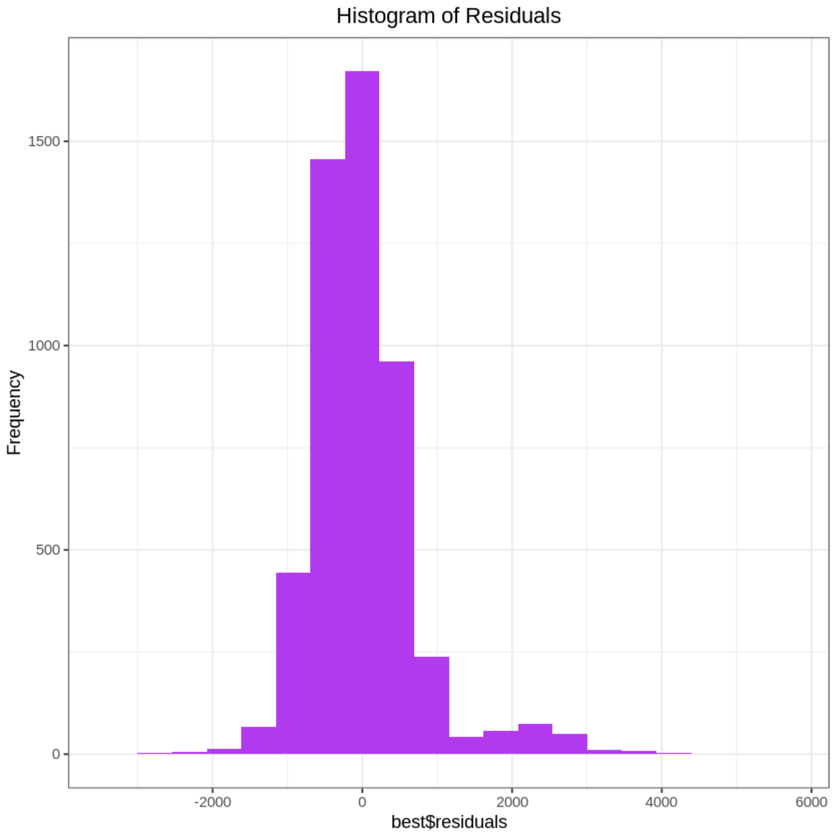
Baed on the result, in each plot there is a horizontal line with the zero slope which can be fit on residuals. Therefore, we can comclude that linearity condtion is satisfied.

Nearly normal residuals

We will check this condition using histogram and normal probability plot.

Histogram

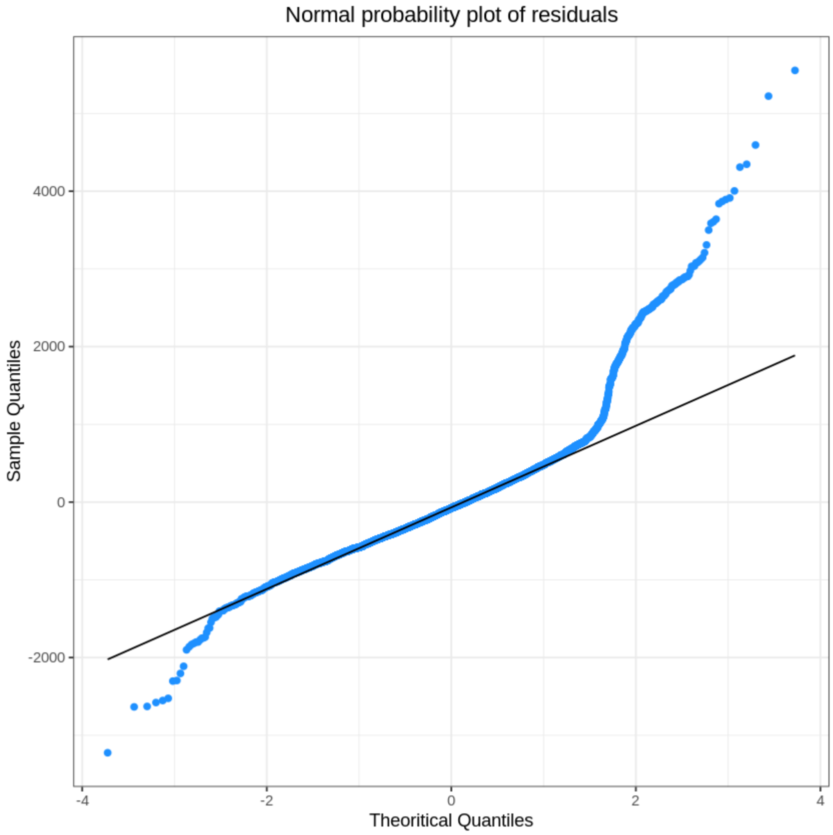
```
In [ ]: modelHist <- ggplot(data=best, aes(best$residuals))
        modelHist <- modelHist + geom_histogram(bins=20, fill="darkorchid2")
        modelHist <- modelHist + ylab("Frequency")
        modelHist <- modelHist + ggtitle("Histogram of Residuals")
        modelHist <- modelHist + theme_bw()
        modelHist <- modelHist + theme(plot.title = element_text(hjust = 0.5
        ))
        modelHist
```



Based on the result, we can say that residuals have a nearly normal distribution. So, this condition is met.

QQ Plot

```
In [ ]: modelQQ <- ggplot(best, aes(sample=best$residuals))
        modelQQ <- modelQQ + stat_qq(col="dodgerblue")
        modelQQ <- modelQQ + stat_qq_line()
        modelQQ <- modelQQ + ylab("Sample Quantiles")
        modelQQ <- modelQQ + xlab("Theoritical Quantiles")
        modelQQ <- modelQQ + ggtitle("Normal probability plot of residuals")
        modelQQ <- modelQQ + theme_bw()
        modelQQ <- modelQQ + theme(plot.title = element_text(hjust = 0.5))
        modelQQ
```



Based on the result, tails are a bit different from normal plot. But we will check the remaining condition and test our model later.

Constant variability

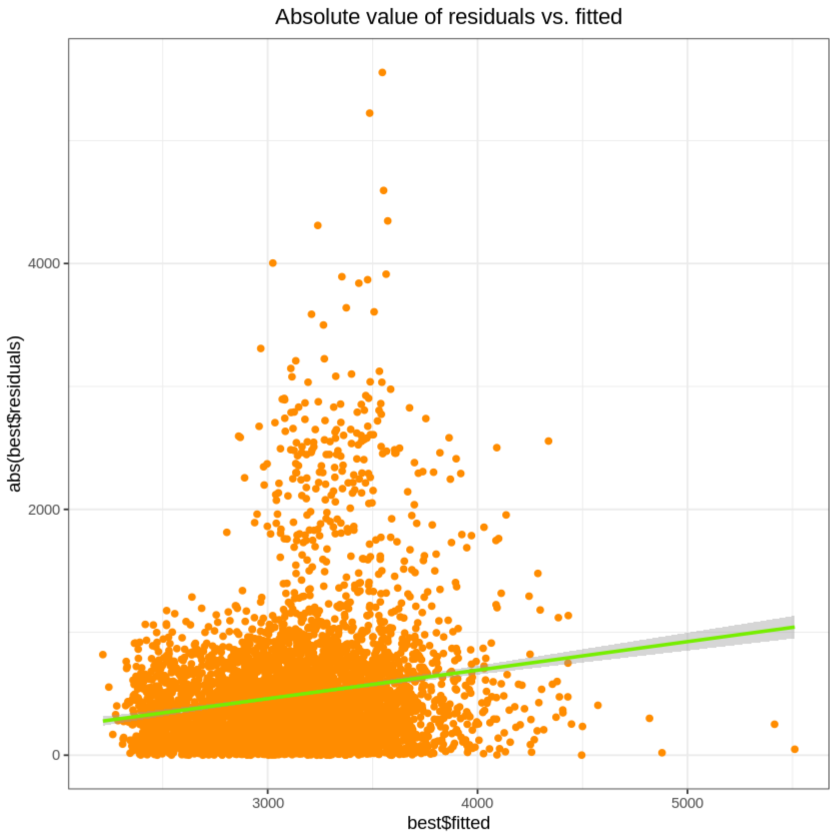
Residuals should be equally variable for low and high values of the predicted response variable.

We will check it using residuals plots of residuals vs. predicted.

In [ ]:

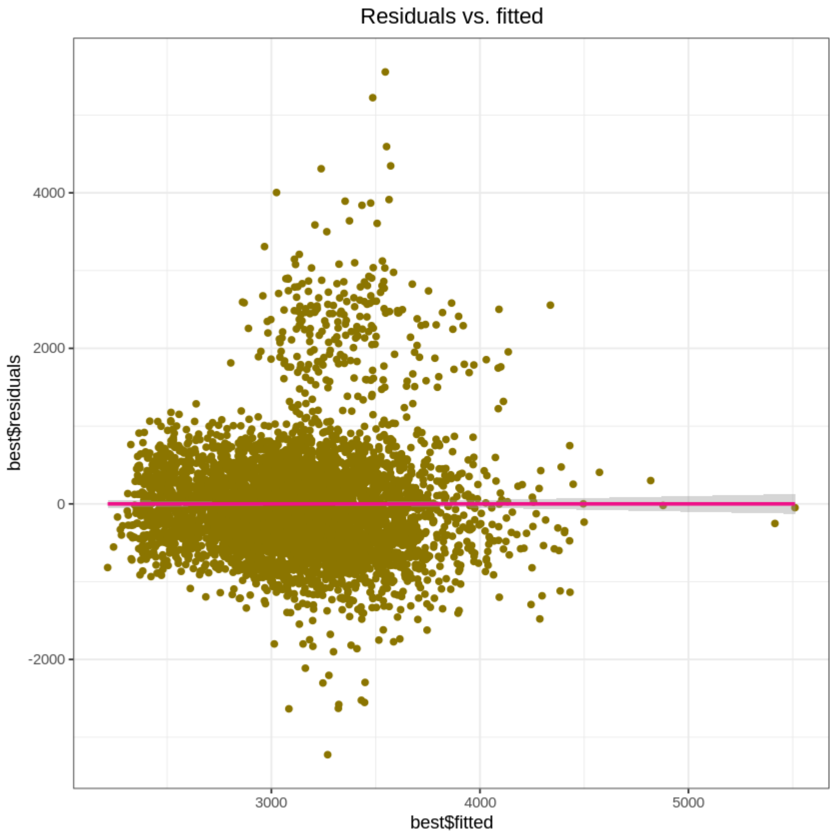
```
modelRes <- ggplot(data = best, aes(best$fitted, abs(best$residuals)))
modelRes <- modelRes + geom_point(color = "darkorange")
modelRes <- modelRes + stat_smooth(method = lm, color="chartreuse2")
modelRes <- modelRes + ggtitle("Absolute value of residuals vs. fitted")
modelRes <- modelRes + theme_bw()
modelRes <- modelRes + theme(plot.title = element_text(hjust = 0.5))
modelRes
```

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



```
In [ ]: modelRes <- ggplot(data = best, aes(best$fitted, best$residuals))
        modelRes <- modelRes + geom_point(color = "gold4")
        modelRes <- modelRes + stat_smooth(method = lm, color="deeppink2")
        modelRes <- modelRes + ggtitle("Residuals vs. fitted")
        modelRes <- modelRes + theme_bw()
        modelRes <- modelRes + theme(plot.title = element_text(hjust = 0.5))
        modelRes
```

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



Based on the results, there is a linear model with the slope equals zero that can be fit on residuals. To sum up, we can say all the conditions are met for this model and we can rely on it predictions.

Part G

K-Fold Cross Validation

K-Fold Cross Validation is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point.

In 5-Fold cross validation(K=5), the data set is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set.

We use `trainControl()` function to define 5-fold cv as our control method.

```
In [ ]: trainControl <- trainControl(method="cv", number=5)
```

Now, we use `train()` method to run this test on the model that we defined in part B.

```
In [ ]: myCrossVal <- train(health_bills ~ bmi + age, data=healthCare,
                            trControl=trainControl, method="lm")
        myCrossVal
```

```
Linear Regression

5109 samples
   2 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 4089, 4086, 4085, 4087, 4089
Resampling results:

  RMSE     Rsquared   MAE
  720.609  0.2055299  490.8399

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Based on the result, RMSE for the model of part B is 720.2921.

The formula of this metric is as follows:

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

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. It measures how spread out the residuals are. In other words, it is the standard deviation of the unexplained variance. One of its benefits is that it has the same unit as data.

By using this metric, we can compare the accuracy of our models. The smaller the RMSE is, the more successful our model is in predicting response variable.

In [ ]:

```
bestCrossVal <- train(health_bills ~ bmi + age + avg_glucose_level, data=healthCare,
                      trControl=trainControl, method="lm")
bestCrossVal
```

```
Linear Regression

5109 samples
   3 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 4088, 4085, 4087, 4088, 4088
Resampling results:

  RMSE      Rsquared   MAE
  718.4328  0.2107306  491.6144

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Based on the result, RMSE for the model of part E is 718.4328.

Its RMSE is less than what we found for model of part B. It shows that it is better model and we can rely on the statical methods that we used to reach this model.

Question 6

For this question, we select **Heart Disease** as a categorical response variable, **Age** as a numerical explanatory variable, and **Gender** as a categogorical explanatory variable.

Part A

Wew will use `glm()` method to create our model.

In [146]:
```
myModel <- glm(heart_disease ~ age + gender, family = binomial, data
= healthCare)
summary(myModel)
```

```
Call:
glm(formula = heart_disease ~ age + gender, family = binomial,
    data = healthCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.9895  -0.3358  -0.1654  -0.0717   3.7020  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -7.880882   0.352513 -22.356  < 2e-16 ***
age          0.079898   0.004962  16.100  < 2e-16 ***
genderMale   0.869806   0.133519   6.514 7.29e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1667.2  on 5106  degrees of freedom
AIC: 1673.2

Number of Fisher Scoring iterations: 7
```

By using `exp()` we get odds ratio of predictors.

In [147]:
```
exp(cbind(coef(myModel)))
```

A matrix: 3 × 1 of type dbl

(Intercept)	0.0003778995
age	1.0831765490
genderMale	2.3864489565

Gender slope

In terms of log odds ratio:

When the other predictors are held constant, the log odds ratio of having heart disease for men are 0.869806 higher than women.

In terms of odds ratio:

When the other predictors are held constant, the odds ratio of having heart disease for males is 2.3864489565 times of the odds ratio of having heart disease for females.

Age slope

In terms of log odds ratio:

When the other predictors are held constant, for a unit increase in age (being 1 year older) the log odds ratio of having heart disease increases on average by 0.079898.

In terms of odds ratio:

When the other predictors are held constant, for a unit increase in age (being 1 year older) the odds ratio of having heart disease will be multiplied by 1.0831765490.

Part B

In this part, we consider gender as our categorical explanatory variable and we draw its OR curve using ggplot.

In [219]:

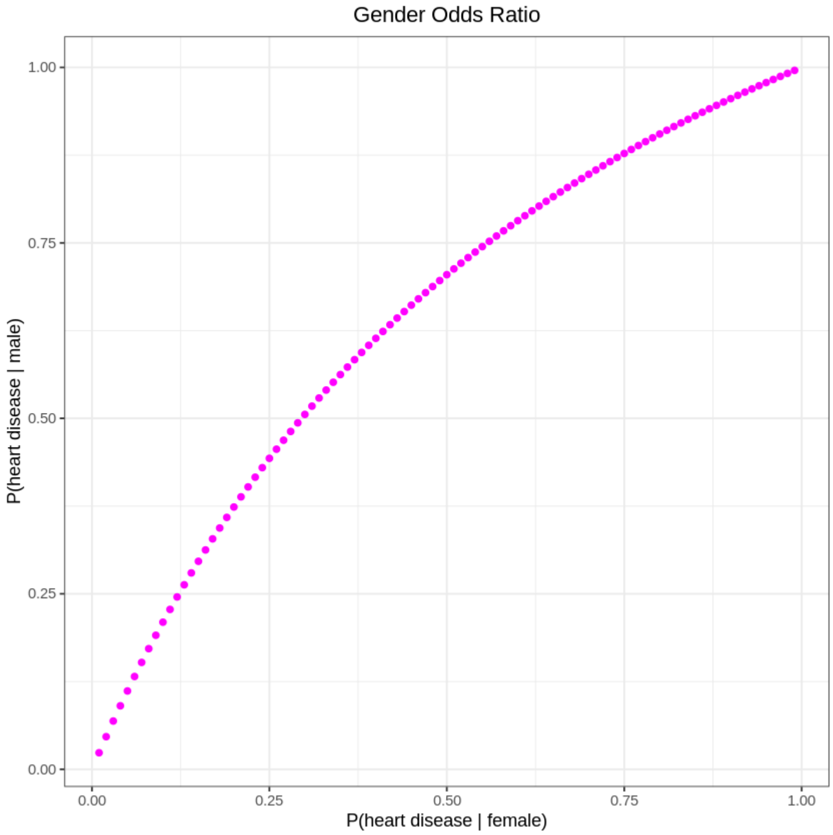
```
maleProb <- function(female) {
    OR <- exp(coef(myModel)["genderMale"])
    return ((OR * (female) / (1 - female)) / (1 + (OR * (female) / (1 - female) )))
}
```

In [222]:

```
female <- matrix(c(1:99) / 100, nrow = 1, ncol = 99)
male <- apply(female, 2, maleProb)
```

In [223]:
```
data <- data.frame(male=male, female=c(1:99) / 100)

orPlot <- ggplot(data = data, aes(female, male))
orPlot <- orPlot + geom_point(color="magenta")
orPlot <- orPlot + ylab("P(heart disease | male)")
orPlot <- orPlot + xlab("P(heart disease | female)")
orPlot <- orPlot + ggtitle("Gender Odds Ratio")
orPlot <- orPlot + theme_bw()
orPlot <- orPlot + theme(plot.title = element_text(hjust = 0.5))
orPlot
```



Based on the result, we can see the odds ratio curve for this variable. It confirms what we said about the odds ratio of having heart disease for male and female. As we see, it is higher for Male.

Part C

ROC shows the trade off in sensitivity and specificity for all possible thresholds.

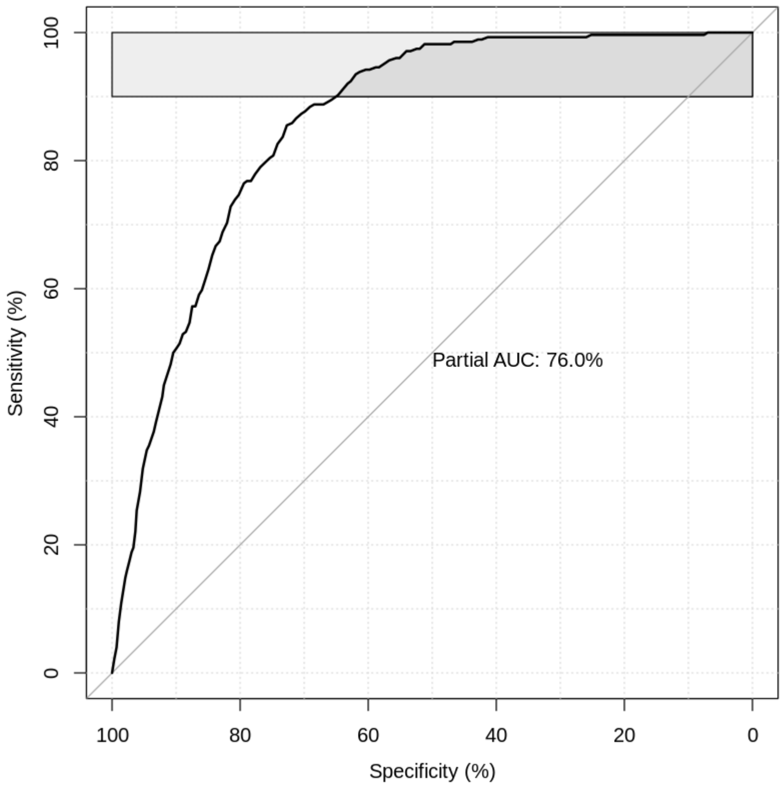
We can use the area under the curve (AUC) as an assessment of the predictive ability of a model.

In [131]:
```
predictions <- predict(myModel, type=c("response"))
```

```
In [136]: roc_curve <- roc(heart_disease ~ predictions, data = healthCare, perc
ent=TRUE,
    partial.auc=c(100, 90),
    partial.auc.correct=TRUE,
    partial.auc.focus="sens",
    plot=TRUE,
    auc.polygon=TRUE,
    max.auc.polygon=TRUE,
    grid=TRUE,
    print.auc=TRUE,
    show.thres=TRUE)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



Based on the result, AUC of our model is 76.09%. As we learnt in class, if it was above 90% we could say that this model is very good at classification, between 90% to 90% is good. But the AUC of our model is below 80% and it is not good.

Part D

Based on the result, p-value for age is less than p-value for gender. It means that age is more significant predictor than gender. It is undeniable that the older we get, the chance of getting heart disease increases and it is more effective than gender.

Part E

In [ ]:

```
fullLogit <- glm(heart_disease ~ . - id, family = binomial, data = he
althCare)
summary(fullLogit)
```

```
Call:
glm(formula = heart_disease ~ . - id, family = binomial, data = healt
hCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4677  -0.3156  -0.1510  -0.0717   3.4115  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -8.910e+00  1.077e+00  -8.270  < 2e-16 ***
genderMale                  7.981e-01  1.382e-01   5.777 7.62e-09 ***
age                         7.931e-02  5.864e-03  13.524  < 2e-16 ***
hypertensionYes             1.005e-01  1.686e-01   0.596  0.55121    
ever_marriedYes            -3.008e-01  2.206e-01  -1.363  0.17280    
work_typeGovt_job          -2.347e-01  1.094e+00  -0.215  0.83011    
work_typeNever_worked      -9.651e+00  3.051e+02  -0.032  0.97476    
work_typePrivate           -1.834e-01  1.083e+00  -0.169  0.86557    
work_typeSelf-employed     -2.936e-01  1.098e+00  -0.267  0.78915    
Residence_typeUrban        -5.515e-02  1.354e-01  -0.407  0.68373    
avg_glucose_level           5.130e-03  1.160e-03   4.423 9.74e-06 ***
bmi                        -1.509e-02  1.201e-02  -1.256  0.20918    
smoking_statusnever smoked -2.044e-01  1.760e-01  -1.162  0.24542    
smoking_statussmokes        5.152e-01  1.999e-01   2.577  0.00996 ** 
smoking_statusUnknown      -4.938e-02  2.076e-01  -0.238  0.81203    
stroke                     -6.914e-01  2.895e-01  -2.388  0.01692 *  
health_bills                4.330e-04  9.315e-05   4.649 3.34e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1598.3  on 5092  degrees of freedom
AIC: 1632.3

Number of Fisher Scoring iterations: 14
```

Step 1

In [ ]:

```
step1dropWork <- glm(heart_disease ~ . - id - work_type, family = binomial, data = healthCare)
summary(step1dropWork)
```

```
Call:
glm(formula = heart_disease ~ . - id - work_type, family = binomial, 
    data = healthCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4491  -0.3164  -0.1518  -0.0708   3.4534  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -9.068e+00  6.045e-01 -15.002  < 2e-16 ***
genderMale                  8.010e-01  1.381e-01   5.801 6.59e-09 ***
age                         7.823e-02  5.453e-03  14.347  < 2e-16 ***
hypertensionYes             9.703e-02  1.683e-01   0.576  0.56434    
ever_marriedYes            -2.996e-01  2.181e-01  -1.374  0.16949    
Residence_typeUrban        -5.411e-02  1.353e-01  -0.400  0.68932    
avg_glucose_level           5.157e-03  1.159e-03   4.451 8.55e-06 ***
bmi                        -1.541e-02  1.190e-02  -1.295  0.19523    
smoking_statusnever smoked -2.032e-01  1.759e-01  -1.155  0.24803    
smoking_statussmokes        5.149e-01  1.995e-01   2.581  0.00985 ** 
smoking_statusUnknown      -4.409e-02  2.070e-01  -0.213  0.83128    
stroke                     -6.863e-01  2.894e-01  -2.372  0.01770 *  
health_bills                4.356e-04  9.304e-05   4.682 2.84e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1598.9  on 5096  degrees of freedom
AIC: 1624.9

Number of Fisher Scoring iterations: 7
```

In [ ]:

```
step2dropResidence <- glm(heart_disease ~ . - id - work_type - Reside
nce_type, family = binomial, data = healthCare)
summary(step2dropResidence)
```

```
Call:
glm(formula = heart_disease ~ . - id - work_type - Residence_type, 
    family = binomial, data = healthCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4343  -0.3166  -0.1519  -0.0710   3.4437  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -9.096e+00  6.004e-01 -15.149  < 2e-16 ***
genderMale                  8.020e-01  1.381e-01   5.809 6.29e-09 ***
age                         7.813e-02  5.443e-03  14.353  < 2e-16 ***
hypertensionYes             9.884e-02  1.683e-01   0.587   0.5569    
ever_marriedYes            -2.967e-01  2.180e-01  -1.361   0.1734    
avg_glucose_level           5.153e-03  1.159e-03   4.447 8.71e-06 ***
bmi                        -1.544e-02  1.190e-02  -1.298   0.1944    
smoking_statusnever smoked -2.001e-01  1.757e-01  -1.139   0.2549    
smoking_statussmokes        5.132e-01  1.995e-01   2.573   0.0101 *  
smoking_statusUnknown      -4.304e-02  2.070e-01  -0.208   0.8353    
stroke                     -6.898e-01  2.892e-01  -2.385   0.0171 *  
health_bills                4.369e-04  9.303e-05   4.696 2.65e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1599.0  on 5097  degrees of freedom
AIC: 1623

Number of Fisher Scoring iterations: 7
```

In [ ]:

```
step3dropHyper <- glm(heart_disease ~ . - id - work_type - Residence_
type - hypertension, family = binomial,
    data = healthCare)
summary(step3dropHyper)
```

```
Call:
glm(formula = heart_disease ~ . - id - work_type - Residence_type -
    hypertension, family = binomial, data = healthCare)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4524  -0.3186  -0.1527  -0.0709   3.4473

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -9.128e+00  5.984e-01 -15.255  < 2e-16 ***
genderMale                  8.019e-01  1.381e-01   5.808 6.31e-09 ***
age                         7.852e-02  5.404e-03  14.530  < 2e-16 ***
ever_marriedYes            -2.984e-01  2.180e-01  -1.369  0.17097
avg_glucose_level           5.222e-03  1.152e-03   4.532 5.86e-06 ***
bmi                        -1.504e-02  1.188e-02  -1.265  0.20572
smoking_statusnever smoked -1.953e-01  1.755e-01  -1.113  0.26584
smoking_statussmokes        5.140e-01  1.995e-01   2.577  0.00998 **
smoking_statusUnknown      -4.989e-02  2.066e-01  -0.241  0.80917
stroke                     -6.877e-01  2.893e-01  -2.377  0.01743 *
health_bills                4.386e-04  9.299e-05   4.716 2.40e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1599.4  on 5098  degrees of freedom
AIC: 1621.4

Number of Fisher Scoring iterations: 7
```

In [ ]:

```
step4dropBmi <- glm(heart_disease ~ . - id - work_type - Residence_type - hypertension - bmi, family = binomial,
    data = healthCare)
summary(step4dropBmi)
```

```
Call:
glm(formula = heart_disease ~ . - id - work_type - Residence_type - 
    hypertension - bmi, family = binomial, data = healthCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4626  -0.3193  -0.1519  -0.0678   3.4891  

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -9.5187748  0.5226793 -18.212  < 2e-16 ***
genderMale                  0.8062653  0.1378273   5.850 4.92e-09 ***
age                         0.0797838  0.0053719  14.852  < 2e-16 ***
ever_marriedYes            -0.3078652  0.2174427  -1.416  0.15682    
avg_glucose_level           0.0049528  0.0011299   4.383 1.17e-05 ***
smoking_statusnever smoked -0.1933605  0.1752187  -1.104  0.26979    
smoking_statussmokes        0.5210566  0.1990764   2.617  0.00886 ** 
smoking_statusUnknown      -0.0442510  0.2062031  -0.215  0.83008    
stroke                     -0.6019900  0.2791973  -2.156  0.03107 *  
health_bills                0.0004033  0.0000882   4.573 4.81e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1601.0  on 5099  degrees of freedom
AIC: 1621

Number of Fisher Scoring iterations: 7
```

In [ ]:

```
step5dropMarried <- glm(heart_disease ~ . - id - work_type - Residenc
e_type - hypertension - bmi - ever_married,
    family = binomial, data = healthCare)
summary(step5dropMarried)
```

```
Call:
glm(formula = heart_disease ~ . - id - work_type - Residence_type -
    hypertension - bmi - ever_married, family = binomial, data = heal
thCare)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4644  -0.3240  -0.1530  -0.0623   3.5446

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -9.727e+00  5.096e-01 -19.088  < 2e-16 ***
genderMale                  7.964e-01  1.374e-01   5.797 6.75e-09 ***
age                         7.876e-02  5.391e-03  14.611  < 2e-16 ***
avg_glucose_level           4.853e-03  1.125e-03   4.312 1.61e-05 ***
smoking_statusnever smoked -1.852e-01  1.748e-01  -1.059  0.28943
smoking_statussmokes        5.230e-01  1.987e-01   2.632  0.00848 **
smoking_statusUnknown      -3.463e-02  2.058e-01  -0.168  0.86640
stroke                     -6.021e-01  2.795e-01  -2.155  0.03120 *
health_bills                4.072e-04  8.817e-05   4.618 3.87e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1602.9  on 5100  degrees of freedom
AIC: 1620.9

Number of Fisher Scoring iterations: 7
```

In [ ]:

```
bestGlm <- glm(heart_disease ~ gender + age + avg_glucose_level + hea
lth_bills,
    family = binomial, data = healthCare)
summary(bestGlm)
```

```
Call:
glm(formula = heart_disease ~ gender + age + avg_glucose_level + 
    health_bills, family = binomial, data = healthCare)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4524  -0.3220  -0.1580  -0.0712   3.5430  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)       -9.0617821  0.4049907 -22.375  < 2e-16 ***
genderMale         0.8324774  0.1355936   6.140 8.28e-10 ***
age                0.0744448  0.0051096  14.570  < 2e-16 ***
avg_glucose_level  0.0048683  0.0011180   4.354 1.33e-05 ***
health_bills       0.0002748  0.0000599   4.588 4.47e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2147.7  on 5108  degrees of freedom
Residual deviance: 1621.6  on 5104  degrees of freedom
AIC: 1631.6

Number of Fisher Scoring iterations: 7
```

Part F

Our utility function is as follows:

$$utility = TP - FP + TN - 4 \times FN$$

In [252]:

```
prob = predict(bestGlm, type=c("response"))
```

In [238]:

```
healthCare$heart <- mapvalues(healthCare$heart_disease,
                              from = c(0, 1),
                              to = c("No", "Yes"))
```

In [250]:

```
utility <- function(thresholds) {
    confusion_matrix = table(healthCare$heart, prob <= thresholds)
    utility = confusion_matrix["Yes", "TRUE"] - confusion_matrix["Yes", "FALSE"] +
            confusion_matrix["No", "TRUE"] -  4 * confusion_matrix["No", "FALSE"]
    return (utility)
}
```

In [254]:

```
thresholds <- matrix(c(1:5109) / 5110, nrow = 1, ncol = 5110)
```

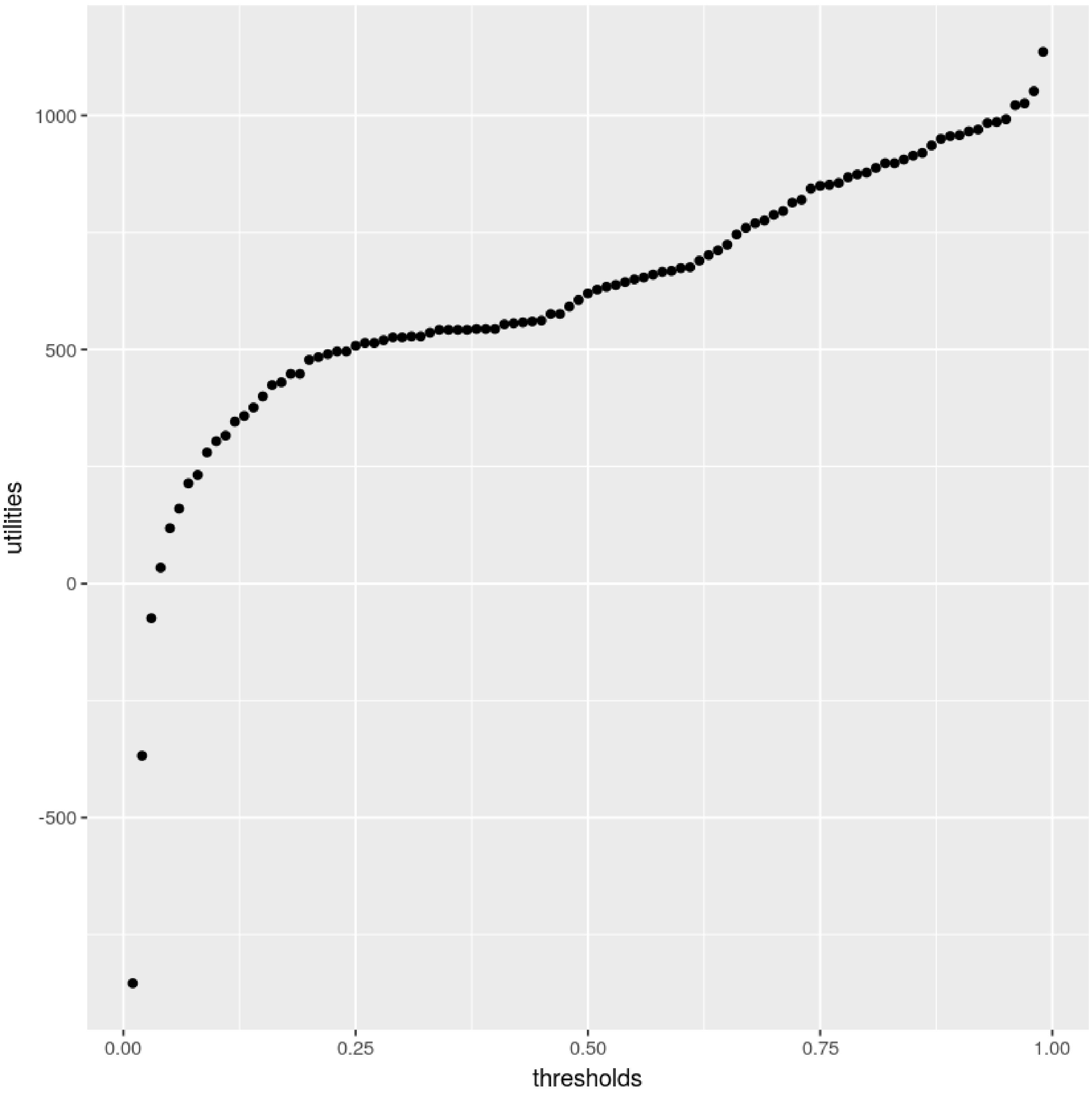
In [ ]:

```
utilities = apply(thresholds, 2, utility)
```

In [ ]:

```
data <- data.frame(utilities=utilities, thresholds=c(1:99) / 100)

ggplot(data = data, aes(thresholds, utilities)) + geom_point()
```


1000
500
0
-500
utilities
0.00
0.25
0.50
0.75
1.00
thresholds

Based on the result, we can say that about the threshold of 0.24 this curve is broke, so we will choose it as our threshold.

Question 7

At first, we want to choose our threshold for this new variable. We will consider median of *Health Bills*, because this variable is related to high medical costs. Also, we do not consider mean because mean is sensitive to outliers.

In [ ]:
```
summary(healthCare$health_bills)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   44.8  2647.8  3032.2  3134.6  3454.9  9100.5 
```

Therefore, our threshold would be 3032.2.

In [ ]:
```
newHealth <- healthCare
newHealth$high_medical_costs <- healthCare$health_bills > 3032.2
```

Now, we can see the new column in our data.

In [ ]:

```
summary(newHealth)
```

```
       id            gender                age          hypertension
 Min.   :   67   Length:5109        Min.   : 0.08   Length:5109
 1st Qu.:17740   Class :character   1st Qu.:25.00   Class :character
 Median :36922   Mode  :character   Median :45.00   Mode  :character
 Mean   :36514                      Mean   :43.23
 3rd Qu.:54643                      3rd Qu.:61.00
 Max.   :72940                      Max.   :82.00
 heart_disease     ever_married        work_type         Residence_ty
pe
 Min.   :0.00000   Length:5109        Length:5109        Length:5109
 1st Qu.:0.00000   Class :character   Class :character   Class :chara
cter
 Median :0.00000   Mode  :character   Mode  :character   Mode  :chara
cter
 Mean   :0.05402
 3rd Qu.:0.00000
 Max.   :1.00000
 avg_glucose_level      bmi        smoking_status         stroke
 Min.   : 55.12    Min.   :10.30   Length:5109        Min.   :0.00000
 1st Qu.: 77.24    1st Qu.:23.80   Class :character   1st Qu.:0.00000
 Median : 91.88    Median :28.40   Mode  :character   Median :0.00000
 Mean   :106.14    Mean   :28.89                      Mean   :0.04874
 3rd Qu.:114.09    3rd Qu.:32.80                      3rd Qu.:0.00000
 Max.   :271.74    Max.   :97.60                      Max.   :1.00000
  health_bills    high_medical_costs
 Min.   :  44.8   Mode :logical
 1st Qu.:2647.8   FALSE:2454
 Median :3032.2   TRUE :2655
 Mean   :3134.6
 3rd Qu.:3454.9
 Max.   :9100.5
```

In this part, we create a logistic regression model with all features except *Health Bills*, because we add this new column by that and it can affect our model which is not the goal of question.

```
In [ ]: medicalModel <- glm(high_medical_costs ~ .-health_bills - id,family =
        binomial, data = newHealth)
        summary(medicalModel)
```

```
Call:
glm(formula = high_medical_costs ~ . - health_bills - id, family = bi
nomial,
    data = newHealth)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7679  -0.9517   0.0755   1.0119   2.1735

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -4.2537275  0.2079310 -20.457  < 2e-16 ***
genderMale                  0.0910628  0.0654769   1.391   0.1643
age                         0.0020072  0.0023920   0.839   0.4014
hypertensionYes             0.2363633  0.1200193   1.969   0.0489 *
heart_disease               0.2673097  0.1538717   1.737   0.0823 .
ever_marriedYes            -0.1010322  0.0936619  -1.079   0.2807
work_typeGovt_job          -0.1432738  0.1694210  -0.846   0.3977
work_typeNever_worked       0.3203345  0.4839838   0.662   0.5081
work_typePrivate           -0.0008468  0.1430892  -0.006   0.9953
work_typeSelf-employed     -0.0333286  0.1723389  -0.193   0.8467
Residence_typeUrban        -0.0590059  0.0639299  -0.923   0.3560
avg_glucose_level           0.0019173  0.0007712   2.486   0.0129 *
bmi                         0.1408764  0.0060884  23.139  < 2e-16 ***
smoking_statusnever smoked -0.0105407  0.0941127  -0.112   0.9108
smoking_statussmokes        0.0517804  0.1121237   0.462   0.6442
smoking_statusUnknown      -0.0934114  0.1060111  -0.881   0.3782
stroke                      5.4492345  1.0048062   5.423 5.86e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5722.2  on 5092  degrees of freedom
AIC: 5756.2

Number of Fisher Scoring iterations: 8
```

Based on the result, bmi has has the most impact on the prediction. Because, it has the lowest p-value among all explanatory variables. But we will perform a backward elimination with p-value to ensure that our answer is correct.

Step1

In this step we drop all the levels of work_type, because it has the highest p-value.

In [ ]:

```
step1dropWorkType <- glm(high_medical_costs ~ .-health_bills - id - w
ork_type, family = binomial, data = newHealth)
summary(step1dropWorkType)
```

```
Call:
glm(formula = high_medical_costs ~ . - health_bills - id - work_type,
    family = binomial, data = newHealth)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7613  -0.9513   0.0763   1.0137   2.1734

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -4.2505429  0.2018691 -21.056  < 2e-16 ***
genderMale                  0.0920392  0.0652716   1.410   0.1585
age                         0.0016331  0.0021561   0.757   0.4488
hypertensionYes             0.2382650  0.1196297   1.992   0.0464 *
heart_disease               0.2730052  0.1536299   1.777   0.0756 .
ever_marriedYes            -0.1111663  0.0926955  -1.199   0.2304
Residence_typeUrban        -0.0594561  0.0638747  -0.931   0.3519
avg_glucose_level           0.0019339  0.0007692   2.514   0.0119 *
bmi                         0.1406312  0.0058728  23.946  < 2e-16 ***
smoking_statusnever smoked -0.0100991  0.0938636  -0.108   0.9143
smoking_statussmokes        0.0505771  0.1117433   0.453   0.6508
smoking_statusUnknown      -0.0904361  0.1038166  -0.871   0.3837
stroke                      5.4533433  1.0047960   5.427 5.72e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5724.8  on 5096  degrees of freedom
AIC: 5750.8

Number of Fisher Scoring iterations: 8
```

Step2

In this step we drop all the levels of smoking_status, because it has the highest p-value.

```
In [ ]: step2dropSmoke <- glm(high_medical_costs ~ .-health_bills - id - work
        _type - smoking_status,
            family = binomial, data = newHealth)
        summary(step2dropSmoke)
```

```
Call:
glm(formula = high_medical_costs ~ . - health_bills - id - work_type 
- 
    smoking_status, family = binomial, data = newHealth)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.7581  -0.9491   0.0775   1.0141   2.1596  

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)         -4.3253011  0.1746554 -24.765  < 2e-16 ***
genderMale           0.0894797  0.0649716   1.377   0.1684    
age                  0.0020234  0.0021115   0.958   0.3379    
hypertensionYes      0.2440560  0.1195460   2.042   0.0412 *  
heart_disease        0.2723423  0.1536648   1.772   0.0763 .  
ever_marriedYes     -0.0999066  0.0924104  -1.081   0.2796    
Residence_typeUrban -0.0578438  0.0638185  -0.906   0.3647    
avg_glucose_level    0.0019300  0.0007693   2.509   0.0121 *  
bmi                  0.1416467  0.0058311  24.292  < 2e-16 ***
stroke               5.4470636  1.0047655   5.421 5.92e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5726.8  on 5099  degrees of freedom
AIC: 5746.8

Number of Fisher Scoring iterations: 8
```

Step3

In this step we drop all the levels of Residence_type, because it has the highest p-value.

In [ ]:

```
step3dropResidence <- glm(high_medical_costs ~ .-health_bills - id -
 work_type - smoking_status - Residence_type,
    family = binomial, data = newHealth)
summary(step3dropResidence)
```

```
Call:
glm(formula = high_medical_costs ~ . - health_bills - id - work_type
-
    smoking_status - Residence_type, family = binomial, data = newHea
lth)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7503  -0.9474   0.0764   1.0144   2.1715

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -4.3535894  0.1719809 -25.314  < 2e-16 ***
genderMale         0.0897993  0.0649669   1.382   0.1669
age                0.0019950  0.0021114   0.945   0.3447
hypertensionYes    0.2450470  0.1195289   2.050   0.0404 *
heart_disease      0.2730259  0.1536559   1.777   0.0756 .
ever_marriedYes   -0.0998645  0.0924064  -1.081   0.2798
avg_glucose_level  0.0019336  0.0007692   2.514   0.0119 *
bmi                0.1416368  0.0058316  24.288  < 2e-16 ***
stroke             5.4462686  1.0047789   5.420 5.95e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5727.6  on 5100  degrees of freedom
AIC: 5745.6

Number of Fisher Scoring iterations: 8
```

Step4

In this step we drop age, because it has the highest p-value.

```
In [ ]: step4dropAge <- glm(high_medical_costs ~ .-health_bills - id - work_t
        ype - smoking_status - Residence_type - age,
            family = binomial, data = newHealth)
        summary(step4dropAge)
```

```
Call:
glm(formula = high_medical_costs ~ . - health_bills - id - work_type 
- 
    smoking_status - Residence_type - age, family = binomial, 
    data = newHealth)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.7656  -0.9462   0.0763   1.0160   2.1658  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)       -4.3372372  0.1710429 -25.358  < 2e-16 ***
genderMale         0.0871303  0.0648912   1.343   0.1794    
hypertensionYes    0.2656335  0.1177090   2.257   0.0240 *  
heart_disease      0.3068456  0.1496034   2.051   0.0403 *  
ever_marriedYes   -0.0459523  0.0725667  -0.633   0.5266    
avg_glucose_level  0.0020127  0.0007648   2.632   0.0085 ** 
bmi                0.1424794  0.0057737  24.677  < 2e-16 ***
stroke             5.4817265  1.0041470   5.459 4.79e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5728.5  on 5101  degrees of freedom
AIC: 5744.5

Number of Fisher Scoring iterations: 8
```

Step5

In this step we drop all the levels of ever_married, because it has the highest p-value.

In [ ]:

```
step5dropMarried <- glm(high_medical_costs ~ gender + hypertension +
 heart_disease + avg_glucose_level + bmi + stroke,
    family = binomial, data = newHealth)
summary(step5dropMarried)
```

```
Call:
glm(formula = high_medical_costs ~ gender + hypertension + heart_dise
ase +
    avg_glucose_level + bmi + stroke, family = binomial, data = newHe
alth)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7622  -0.9468   0.0762   1.0173   2.1712

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -4.3325226  0.1708329 -25.361  < 2e-16 ***
genderMale         0.0886886  0.0648507   1.368  0.17144
hypertensionYes    0.2576044  0.1169454   2.203  0.02761 *
heart_disease      0.2984543  0.1489254   2.004  0.04506 *
avg_glucose_level  0.0019826  0.0007631   2.598  0.00938 **
bmi                0.1413567  0.0054847  25.773  < 2e-16 ***
stroke             5.4720879  1.0040051   5.450 5.03e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5728.9  on 5102  degrees of freedom
AIC: 5742.9

Number of Fisher Scoring iterations: 8
```

Step6

In this step we drop all the levels of gender, because it has the highest p-value.

In [ ]:

```
step6dropGender <- glm(high_medical_costs ~ hypertension + heart_dise
ase + avg_glucose_level + bmi + stroke,
    family = binomial, data = newHealth)
summary(step6dropGender)
```

```
Call:
glm(formula = high_medical_costs ~ hypertension + heart_disease + 
    avg_glucose_level + bmi + stroke, family = binomial, data = newHe
alth)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.7515  -0.9484   0.0764   1.0197   2.1562  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)       -4.2985950  0.1689538 -25.442  < 2e-16 ***
hypertensionYes    0.2593195  0.1168976   2.218  0.02653 *  
heart_disease      0.3137647  0.1484217   2.114  0.03451 *  
avg_glucose_level  0.0020211  0.0007625   2.651  0.00804 ** 
bmi                0.1413040  0.0054911  25.733  < 2e-16 ***
stroke             5.4655951  1.0039463   5.444 5.21e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7074.7  on 5108  degrees of freedom
Residual deviance: 5730.8  on 5103  degrees of freedom
AIC: 5742.8

Number of Fisher Scoring iterations: 8
```

As we can see, all the predictors are significant. But, bmi has the lowest p-value among them. So it is the best predictor for it.

As what we saw in previous parts, bmi was the most correlated variable with health bills. Moreover, we built this new feature from health bills. So, we can conclude that bmi would be the best predictor for it.