Biostatistics with R

Jamalludin Ab Rahman MD MPH

Table of Contents

BASIC CONCEPTS IN BIOSTATISTICS

Biostatistics is the application of statistics in medicine, public health and biology.

Population and sample

Variables and level of measurements

Distribution of data

Statistical inference

Causality

QUICK INTRODUCTION TO R

R is an open source language environment used especially for statistical computing and graphics. It is free but it has a steep learning curve. RStudio will make the experience using R more bearable. RStudio is like a skin for R.

Installing R and RStudio

Latest stable version of R can be downloaded from the Comprehensive R Archive Network (CRAN) (https://cran.r-project.org/mirrors.html). The installation process is pretty straight forward. Once R has been installed, you can also install RStudio (https://rstudio.com/products/rstudio/download/) which serve as a nice and user friendly interface to R scripts. There are many tutorials and videos showing the step-by-step process of installing and running R, so I will not be repeating them here. However I would like to highlight some tips that I feel will help you to navigate R and RStudio better.

RStudio interface

By default, RStudio will show various panes in three divided areas. Console is where you type your codes or commands. If you execute the command (by pressing enter or return key), text results will appear in the same console pane, graphics will appear in the Plot pane and any new data structure or object will appear in the Environment pane.

You can also use a Script pane (by going to File>New File>R Script) to write your command. But you need to press Command+Return (or Control+Enter in Windows) to execute it. There are also various other panes in RStudio including Packages (where you can install and update them), Files (where you can navigate directory in your computer) and Help (to get some references obviously).

R command structure

In general, to form a command, you apply function to an object. Functions in R are "first class objects", which means that they can be treated like any other object.

e.g. object <- functions()

R packages

In R you can perform a function in many ways. You can use the built-in functions (Base) or you may use 'customised' functions (R packages). R packages complement the Base R in performing functions.

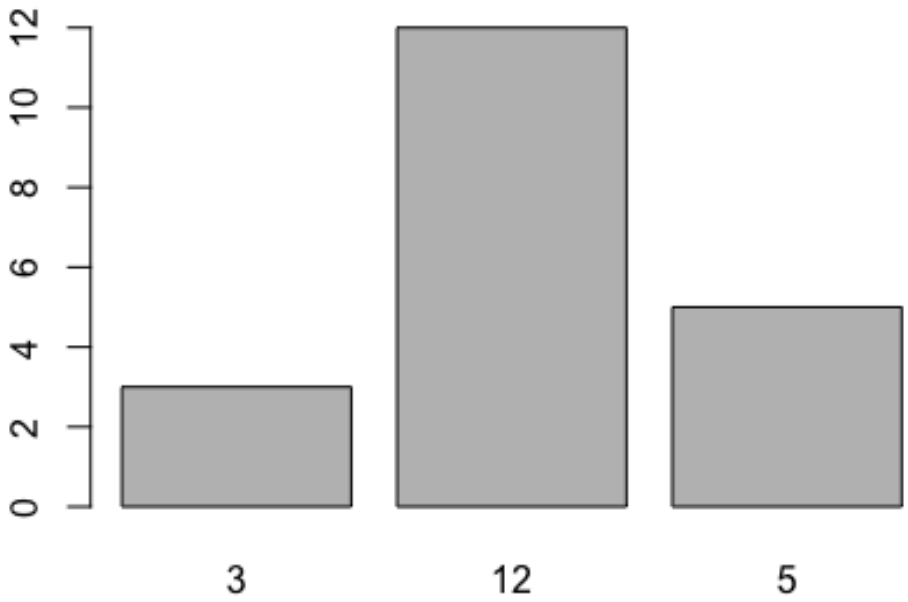
For an example, if you wish to plot a bar chart, you can use

barplot(thedata)

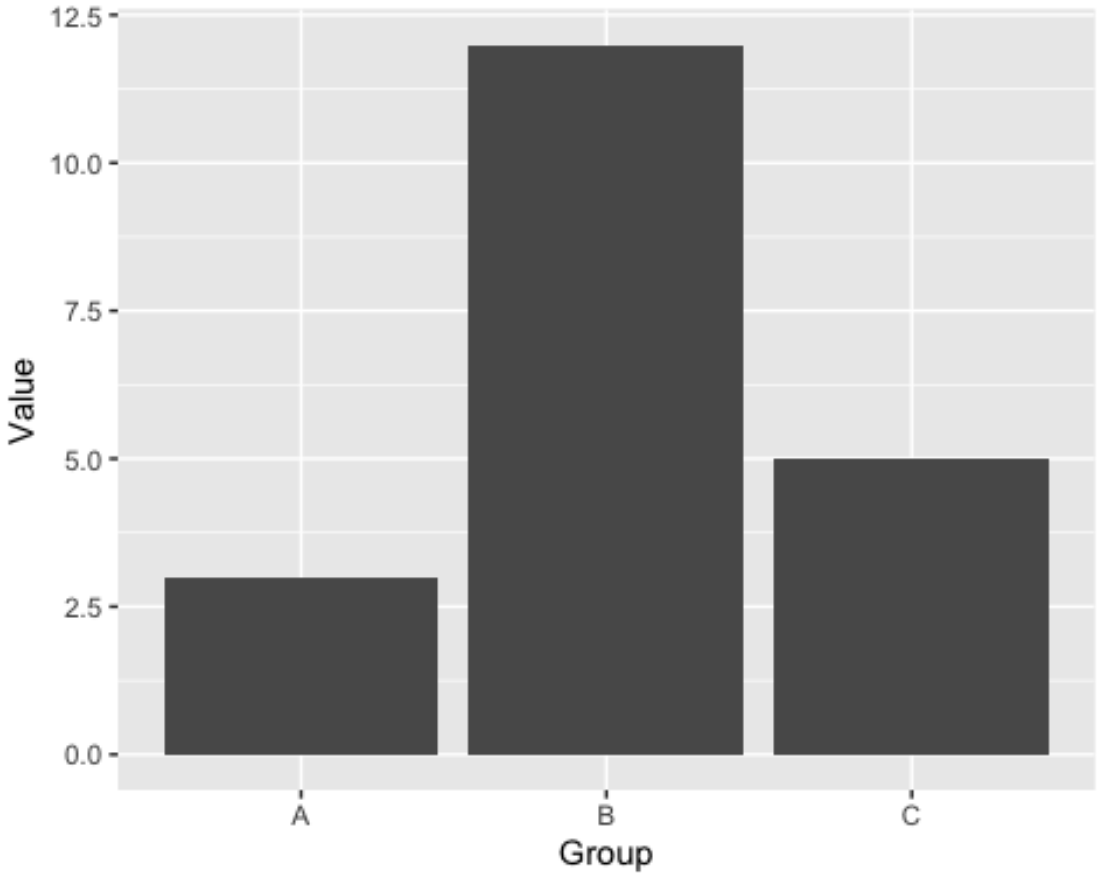
where, 'barplot' is a function in Base R; or alternatively you can also use a package called ggplot2:

ggplot(data=thedata, aes(x=variable_x, y=variable_y))

```
# Create the data frame
thedata <- data.frame(
  Group=c("A","B","C") ,
  Value=c(3,12,5)
  )

# Bar plot using Base R
barplot(height = thedata$Value, names = thedata$Value)

# Bar plot using R Package called ggplot2
library(ggplot2)
```




```
ggplot(thedata, aes(x=Group, y=Value)) +
  geom_bar(stat = "identity")
```



In both methods, we are able to produce a bar plot but **ggplot2** requires a complicated command structure to achieve the same outcome. But of course it has its advantages.

In this guideline, most convenient functions will be used. If a package is required, you need to install and then to load it first. Some of the recommended packages are listed below:

Tidyverse

Tidyverse (https://www.tidyverse.org) is not just a package but an environment and dubbed as a modern R, which include many packages:

1. dplyr
2. ggplot2
3. readr
4. forcats
5. stringr
6. purrr
7. tidyr
8. tibble

gtsummary

To create publication-ready analytical and summary tables.
http://www.danieldsjoberg.com/gtsummary/

ggpubr

This is another excellent package for data visualisation i.e. graphics, in R.
https://rpkgs.datanovia.com/ggpubr/

labelled

https://larmarange.github.io/labelled/articles/intro_labelled.html

Data type and structure

The six basic data types in R are:

1. Character - similar like text in Excel
2. Numeric - any numbers including decimals
3. Integer - only accept whole number
4. Logical - TRUE, FALSE or NA
5. Complex - combination of data type
6. Date

For those who are not used to use R for statistical analysis, you may find this as something new. Unlike SPSS which accept only one data structure, R can accept various data structure. Data frame is akin to SAV file in SPSS. The other types of data structure available in R are:

1. Atomic vector
2. List
3. Matrix
4. Data frame - contain multiple variables
5. Factors

DATA MANAGEMENT IN R

In this first part of the exercise using R, we will analyse a set of data to describe relationship of systolic blood pressure or **SBP** (as the dependent variable or outcome) with **Age**, **Sex**, **Exercise intensity**, **Smoking** and **BMI status**. So we will learn data management in R using this scenario.

Working directory

It is a good habit to set a specific working directory for every project we do. This can be done by using the menu in RStudio by going to **Session>Set Working Directory** and choose where to location. Or if you remember the path, you can type it using **setwd()** command.

Data creation

Data can be created using R manually especially for small databases. For big database, we can create them outside R and we can 'read' the data using R. For this exercise we will import the data from
https://raw.githubusercontent.com/profjamal/biostatistics/main/healthstatus.csv

For this purpose, we will use a library called **tidyverse**.

Viewing data

```
View(healthstatus)
```

This is a hypothetical dataset created just for this exercise. It contains 15 variable and 153 cases.

```
names(healthstatus)
```

```
##  [1] "id"       "age"      "sex"      "exercise" "smoking"  "wt"
##  [7] "ht"       "sbp"      "dbp"      "hba1c"    "hcy"      "wt2"
## [13] "wt3"      "sbp2"     "sbp3"
```

Preparing the data

We will select the relevant variables for the first part of this exercise and we will also create two new variables; **bmi** and **bmistat**

```
mydata <- healthstatus %>%
  dplyr::select(age, sex, exercise, smoking, wt, ht, sbp) %>%
  mutate(bmi = wt/(ht/100)^2) %>%
  mutate(bmistat = if_else(bmi < 25, "Normal",
                           if_else(bmi<30, "Overweight", "Obese")))
```

You can observe that there are originally two types of data, **numeric** and **character**. We should convert them to their proper data type i.e. **factor** for sex, exercise, smoking and bmistat.

- age - numeric
- sex - factor with 2 categories, Male and Female
- exercise - factor with 3 categories, Low, Moderate and High
- smoking - factor with 2 categories, Yes and No
- wt - numeric
- ht - numeric
- sbp - numeric
- bmi - numeric
- bmistat - factor with 3 categories

Factors are used for categorical data. R will assign Female as 1, and Male as 2 because F comes earlier than M. So before you change the value label, you must know the value level.

```
mydata$sex <- factor(mydata$sex, labels = c("Female", "Male"))
levels(mydata$sex)

## [1] "Female" "Male"

nlevels(mydata$sex)

## [1] 2
```

Checking the data

```
table(mydata$sex)

##
## Female   Male
##     70     83

table(mydata$exercise)

##
##      Low Moderate     High
##       74       61       18

table(mydata$smoking)

##
## Yes  No
##  63  90

table(mydata$bmistat)

##
##     Normal Overweight      Obese
##         81         48         24

head(mydata)

## # A tibble: 6 × 9
##     age sex    exercise smoking    wt    ht   sbp   bmi bmistat
##   <dbl> <fct>  <fct>    <fct>   <dbl> <dbl> <dbl> <dbl> <fct>
## 1    36 Male   Moderate Yes      59.5   145   123  28.3 Overweight
## 2    49 Male   Moderate Yes      77.6   166   122  28.2 Overweight
## 3    56 Male   Low      No       75     145   136  35.7 Obese
## 4    61 Female Low      Yes      62.3   158   127  25.0 Normal
## 5    40 Female Low      Yes      55.3   150   151  24.6 Normal
## 6    42 Female Moderate No       54.8   144   128  26.4 Overweight
```

Labeling the variables

It is a good to label each variable properly.

```
library(labelled)
var_label(mydata) <- list(
                       age ="Age(years)",
```

```
                        sex = "Sex",
                        exercise = "Exercise intensity",
                        smoking = "Smoking",
                        wt = "Weight (kg)",
                        ht = "Height (cm)",
                        sbp = "SBP (mmHg)",
                        bmi = "BMI (kg/m2)",
                        bmistat = "BMI status"
)
# Checking the outcome of the above command
var_label(mydata)

## $age
## [1] "Age(years)"
##
## $sex
## [1] "Sex"
##
## $exercise
## [1] "Exercise intensity"
##
## $smoking
## [1] "Smoking"
##
## $wt
## [1] "Weight (kg)"
##
## $ht
## [1] "Height (cm)"
##
## $sbp
## [1] "SBP (mmHg)"
##
## $bmi
## [1] "BMI (kg/m2)"
##
## $bmistat
## [1] "BMI status"
```

DESCRIPTIVE STATISTICS

When the analysis involved only one variable, we called it, but not always, a descriptive statistics. In descriptive statistics, we are not interested to compare one value to another. We are interested to see the magnitude of the values without making any comparison or reference to any other value. For example, we want to know about the baseline characteristics of the study population by describing the distribution of the samples by age, sex, education and economic status. We are not interested to know if the samples are older, or have more male, more educated or at a better economic status. If we are interested to compare those values, we should perform analytical statistics.

Descriptive statistics depends on the type of variable we want to summarise.

Summarising numerical values

For numerical variables, we will start by knowing their distribution. In most biological variables, values are distributed in what is known as Normal distribution. It is a bell-shaped symmetrical line curve with a single peak. Once done, we then summarise the variable using central measures like mean or median, and dispersion measures like SD, SE or IQR depending on the distribution.

Check for Normal distribution

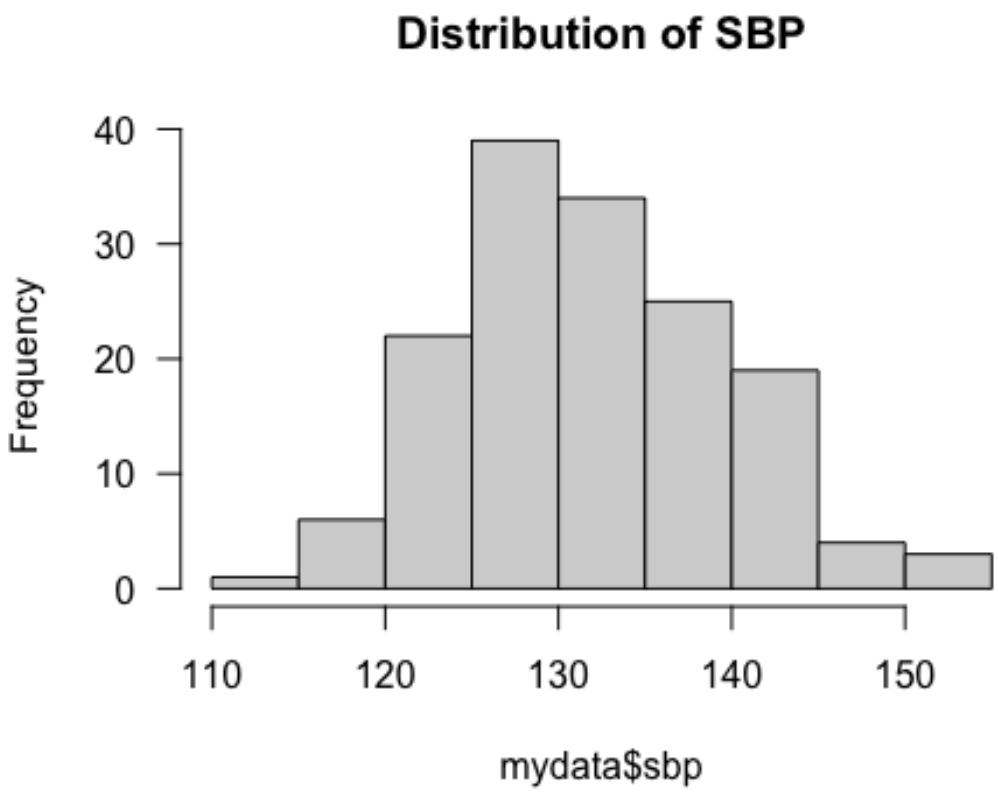
There are basically two approaches to determine the distribution; inspect visually or using the Normality tests. Assessing the distribution visually like checking for the overall smooth symmetrical line curve, comparing mean to median, and checking values for skewness and kurtosis values, is already enough to determine the distribution.

Normality test in the other hand, while being very objective, is very sensitive to sample size. Small sample always pass the test, and a small deviation in large sample will flag the test to show not normally distributed.

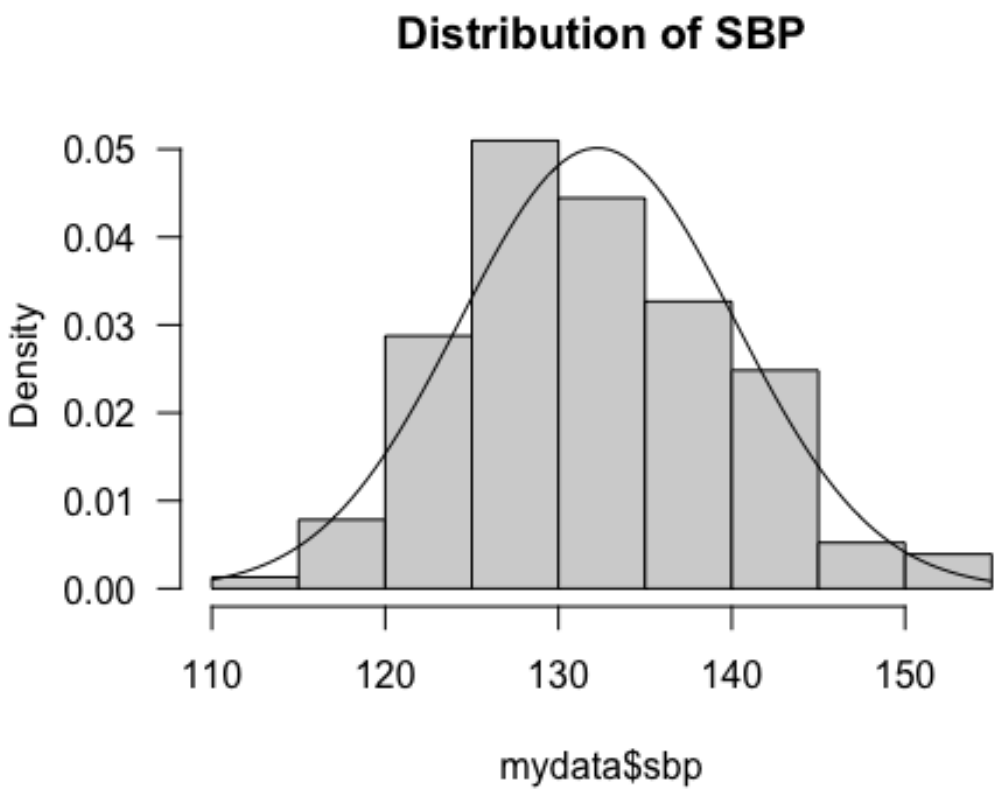
1) Visualising histogram

The followings are method to determine distribution for **sbp**.

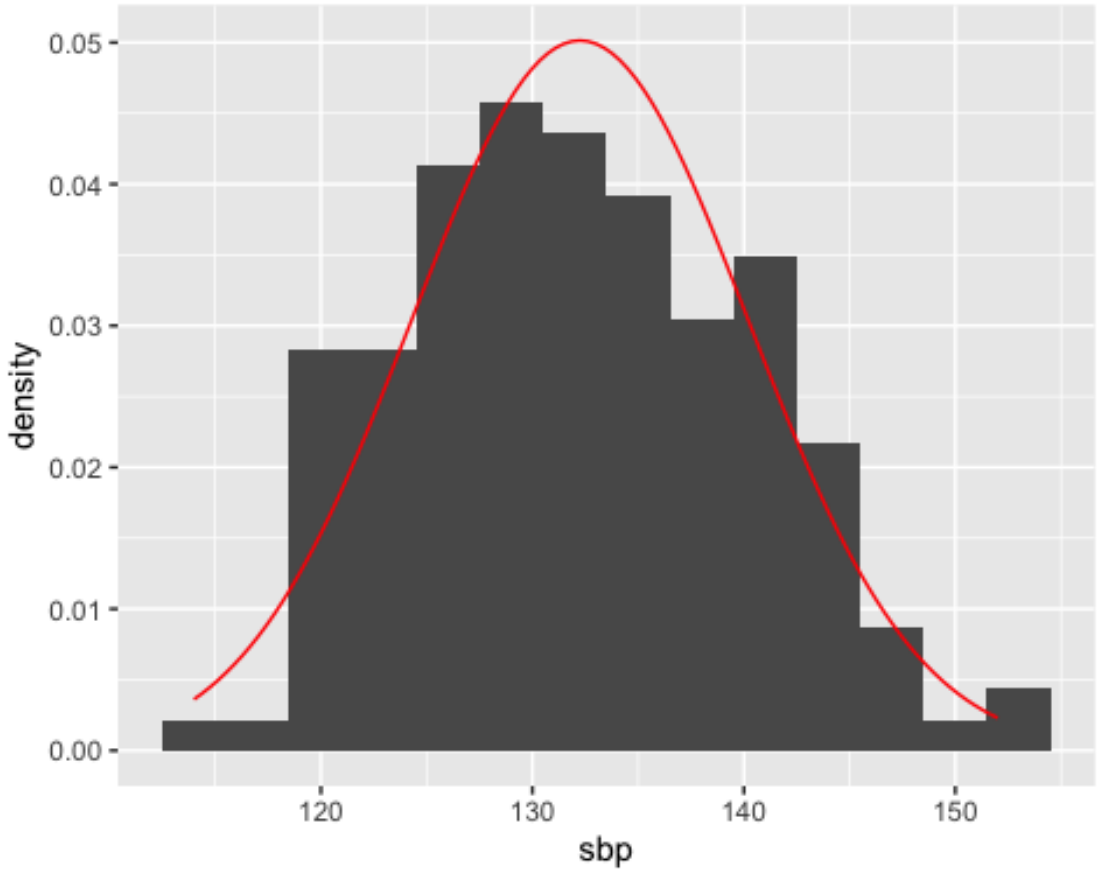
```
 # Display histogram with y-axis showing the frequency
hist(mydata$sbp, main="Distribution of SBP", las=1)
```




```r
# Display the density, then only the normal curve line can be shown
hist(mydata$sbp, main="Distribution of SBP", las=1, prob=TRUE)

# Showing Normal curve (based on density)
curve(dnorm(x, mean=mean(mydata$sbp), sd=sd(mydata$sbp)), add=TRUE)
```

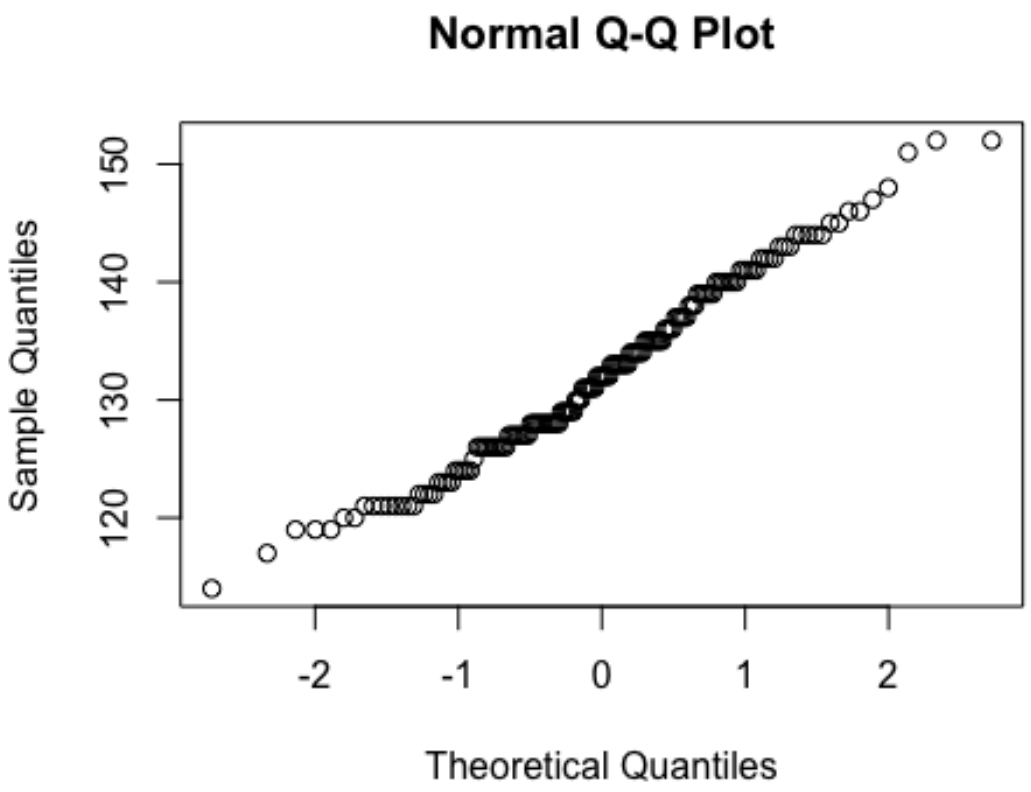



```
histo.sbp <- ggplot(mydata, aes(sbp)) +
                geom_histogram(aes(y=..density..), binwidth = 3) +
                stat_function(fun = dnorm, colour = "red",
                              args = list(mean = mean(mydata$sbp),
                                          sd = sd(mydata$sbp)))


histo.sbp
```

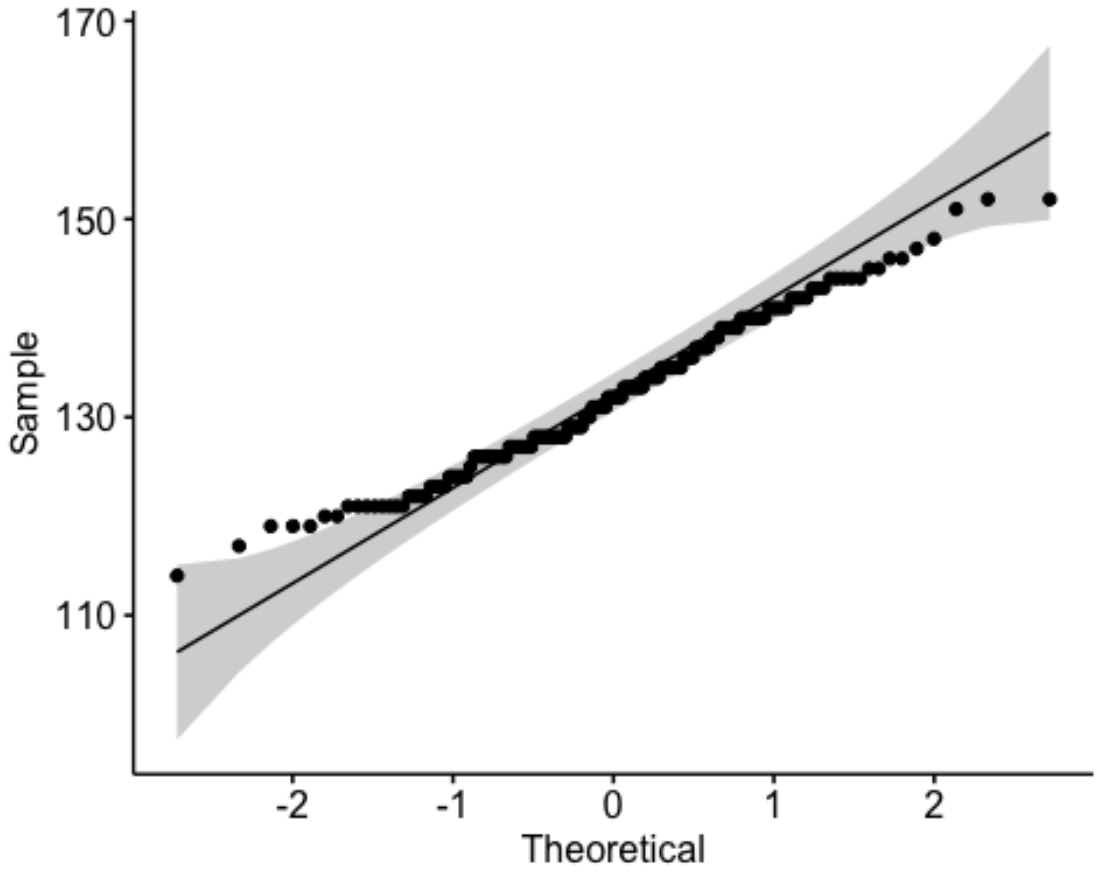


2) Visualising Q-Q plot

```
qqnorm(mydata$sbp) # From Base
```

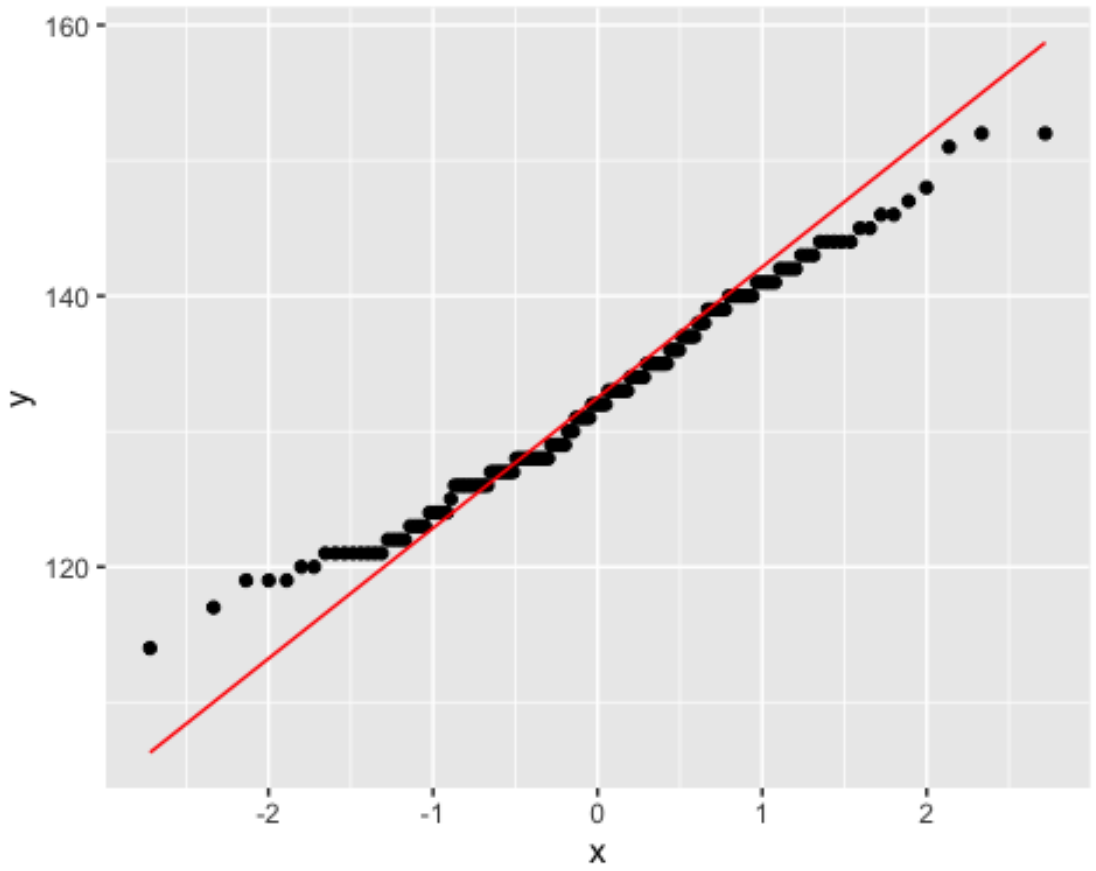



```
library(ggpubr)
ggqqplot(mydata$sbp) # From ggpubr package
```



From the q-q plot we can observed that the observed **sbp** are within the acceptable Normal distribution.

```
qqplot.sbp <- ggplot(mydata, aes(sample=sbp)) +
              geom_qq() + geom_qq_line(color="red")

qqplot.sbp
```



Most of the observations are distributed along the straight line. We can deduce that **sbp** is normally distributed.

3) Normality test

We can also check normality using a Normality test like Shapiro Test.

```
shapiro.test(mydata$sbp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mydata$sbp
## W = 0.98403, p-value = 0.07418
```

Shapiro-wilk test shows that **sbp** is normally distributed (P=0.074). When the P>0.05, we do not reject the Ho that the **sbp** distribution is no different from the Normal distribution, which means that **sbp** is normally distributed.

Now, we should do the same for **age** and *all the numerical variables* in the dataset. First inspect visually, then you may try to run the Shapiro-Wilk test. We use mean(SD) to describe the normally distributed numerical variable and median(IQR) to describe the not normally distributed numerical variable.

We can summarise the whole variables in a table format using **gtsummary** package. A more coomplicated coding is needed but it is worth it.

```
library(gtsummary)
mydata %>%
  select(age, wt, ht, sbp) %>%
  tbl_summary(
    type = all_continuous() ~ "continuous2",
    statistic = all_continuous() ~ c("{mean} ({sd})",
                                     "{median} ({p25}, {p75})",
                                    "{min} - {max}"),
    digits = all_continuous() ~ 1)

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	**N = 153**
Age(years)	
Mean (SD)	42.2 (8.9)
Median (IQR)	42.0 (36.0, 47.0)
Range	21.0 - 64.0
Weight (kg)	
Mean (SD)	60.9 (8.3)
Median (IQR)	59.1 (55.4, 64.2)
Range	42.6 - 82.0
Height (cm)	
Mean (SD)	155.8 (8.9)
Median (IQR)	156.0 (148.0, 162.0)
Range	140.0 - 176.0
SBP (mmHg)	
Mean (SD)	132.2 (8.0)
Median (IQR)	132.0 (126.0, 139.0)
Range	114.0 - 152.0

In the table above, both mean and median are presented. We can actually decide what measure to show by customising the command.

Summarising categorical values

Categorical value is described using count and percentage.

```
# Using base R to summarise categorical variable
table(mydata$sex)

##
## Female   Male
##     70     83

barplot(table(mydata$sex),
        main="Distribution of sex")
```


```
# Showing %
prop.table(table(mydata$sex))*100

##
##   Female     Male
## 45.75163 54.24837

library(gtsummary)
mydata %>%
  select(sex, smoking, exercise, bmistat) %>%
  tbl_summary()
```

```
## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	N = 153
Sex	
Female	70 (46%)
Male	83 (54%)
Smoking	63 (41%)
Exercise intensity	
Low	74 (48%)
Moderate	61 (40%)
High	18 (12%)
BMI status	
Normal	81 (53%)
Overweight	48 (31%)
Obese	24 (16%)

Presenting baseline characteristics

We can now combine all the variables, and present them in one table.

```
mydata %>%
  tbl_summary(
    statistic = list(all_continuous() ~ "{mean} ({sd})",
                     all_categorical() ~ "{n}/{N} ({p}%)"),
    digits = all_continuous() ~ 1
  )

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	N = 153
Age(years)	42.2 (8.9)
Sex	

Characteristic	N = 153
Female	70/153 (46%)
Male	83/153 (54%)
Exercise intensity	
Low	74/153 (48%)
Moderate	61/153 (40%)
High	18/153 (12%)
Smoking	63/153 (41%)
Weight (kg)	60.9 (8.3)
Height (cm)	155.8 (8.9)
SBP (mmHg)	132.2 (8.0)
BMI (kg/m2)	25.3 (4.3)
BMI status	
Normal	81/153 (53%)
Overweight	48/153 (31%)
Obese	24/153 (16%)

ANALYTICAL STATISTICS

Analytical statistics used when we are interested to compare values. We are interested to say if one value is higher than another. For example, if the prevalence of hypertension in male is higher compared to female, or if the prevalence is higher this year compared to last year. In the former example, the two variables involved are hypertension status (yes or no) and sex (male and female), while in the later example, hypertension status (yes or no) and year (this year and last year).

So for analytical statistics, we will compare two variables. In fact, we can only compare two variables at any one time. Even if we are interested to compare prevalence of hypertension between Malay, Chinese and Indian; at any one time, we can compare two races. It can be Malay vs. Chinese, or Malay vs. Indian or Chinese vs. Indian. We can't compare Malay vs. Chinese vs. Indian at the same time. Let say the prevalences are 40%, 35% and 45% for Malay, Chinese and Indian respectively. We may say the highest prevalence is observed among Indian, but to tell how much it is higher, we need to set one reference point only at a time. The prevalence is higher by 5% compared to Malay and 10% compared to Chinese.

Comparing numerical values

Compare two means

Let us compare **sbp** in male and **sbp** in female (comparing two means). Since **sbp** is normally distributed, and **sex** has only two options, we use **Independent sample-t test**.

```
t.test(mydata$sbp ~ mydata$sex)
```

```
##
##  Welch Two Sample t-test
##
## data:  mydata$sbp by mydata$sex
## t = 0.4972, df = 141.8, p-value = 0.6198
## alternative hypothesis: true difference in means between group Female and
group Male is not equal to 0
## 95 percent confidence interval:
##  -1.928972  3.225357
## sample estimates:
## mean in group Female   mean in group Male
##             132.6000             131.9518
```

The analysis shows that there is no difference of mean **sbp** by **sex** (t(141.8)=0.426, P=0.6198).

We can also use **gtsummary** package to show the result is a nice table format

```
mydata %>%
  dplyr::select(sbp, sex) %>%
  tbl_summary(
    by = sex,
    statistic = sbp ~ c("{mean} ({sd})")) %>%
  add_p(test= sbp ~ "t.test") %>%
  add_overall()

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

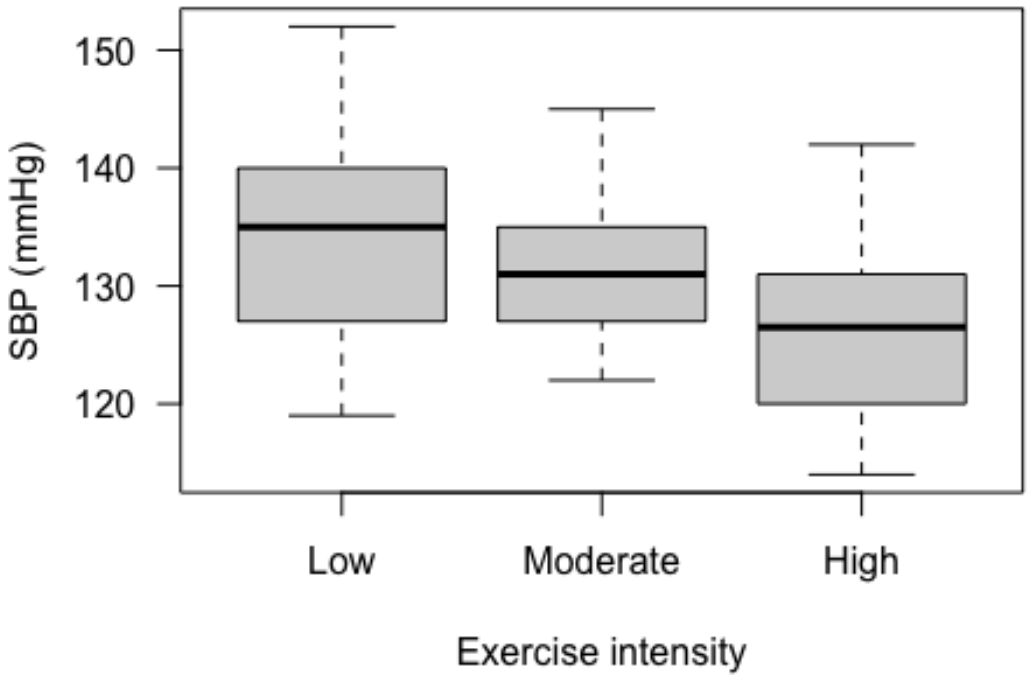
Characteristic	**Overall**, N = 153	**Female**, N = 70	**Male**, N = 83	**p-value**
SBP (mmHg)	132 (8)	133 (8)	132 (8)	0.6

How about **sbp** and **exercise**?

Compare more than two means

Since exercise has more than two categories, we should run **one-way ANOVA**.

```
boxplot(mydata$sbp~mydata$exercise,
        xlab = "Exercise intensity", ylab = "SBP (mmHg)", las=1)
```




```
aov(sbp~exercise, data=mydata)

## Call:
##    aov(formula = sbp ~ exercise, data = mydata)
##
## Terms:
##                 exercise Residuals
## Sum of Squares  1064.026  8558.536
## Deg. of Freedom        2       150
##
## Residual standard error: 7.553602
## Estimated effects may be unbalanced

summary(aov(mydata$sbp~mydata$exercise))

##                  Df Sum Sq Mean Sq F value   Pr(>F)
## mydata$exercise   2   1064   532.0   9.324 0.000152 ***
## Residuals       150   8559    57.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The one-way ANOVA shows that there is at least one pairwise comparison that is different significantly. It could be between Low-Moderate, Low-High or Moderate-High.

Therefore we need to identify which levels are actually different by running a post-hoc test. The preferred post-hoc test depends on the homogeneity of variances between variables. If the variances are equally distributed, **Tukey's HSD** is the recommended test, and if the equal variance can't be assumed, we use **Bonferroni's test**.

So first, let us determine its homogeneity.

```
library(car)

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.2

##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some

leveneTest(sbp ~ exercise, mydata, center=mean)

## Levene's Test for Homogeneity of Variance (center = mean)
##        Df F value  Pr(>F)
## group   2  4.2458 0.01608 *
##       150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This (P=0.016) shows that equal variances cannot be assumed, and therefore we should run Bonferroni's test.

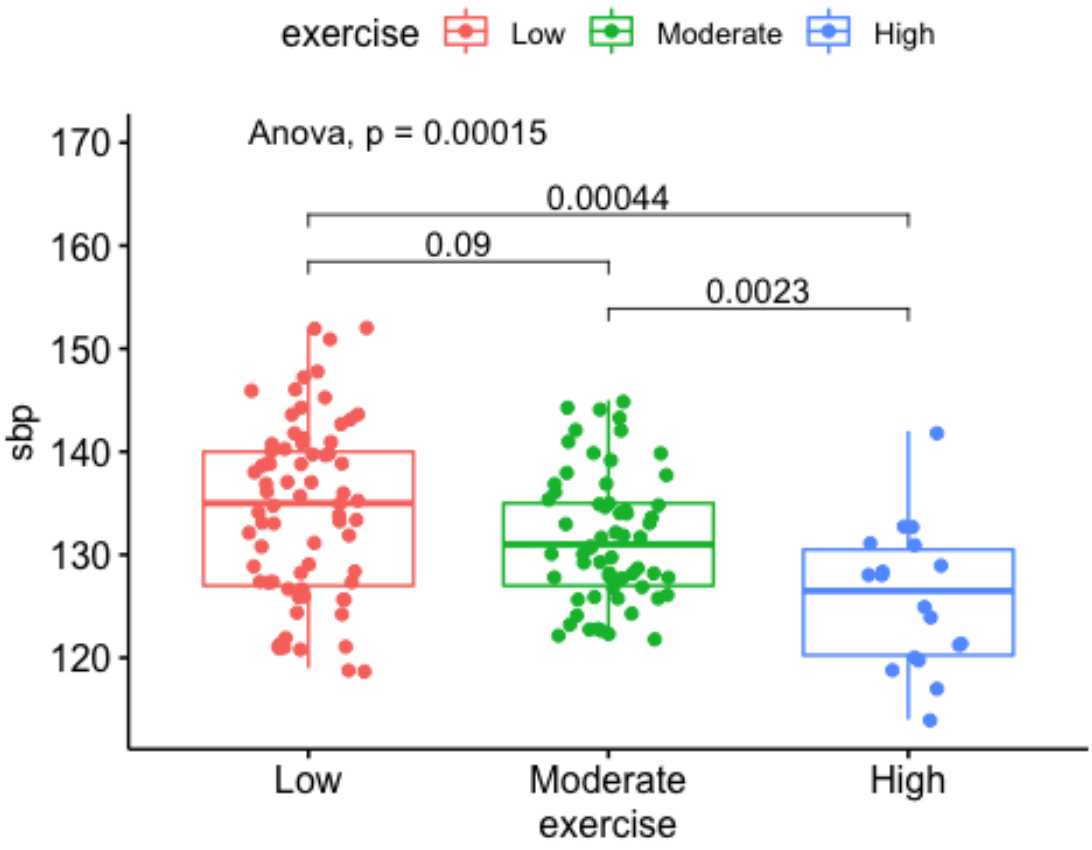
```
# Post-hoc test using Bonferroni
pairwise.t.test(mydata$sbp, mydata$exercise, p.adj = "bonf")

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mydata$sbp and mydata$exercise
##
##          Low     Moderate
## Moderate 0.17101 -
## High     0.00011 0.01135
##
## P value adjustment method: bonferroni

# Experimenting with Tukey's HSD
TukeyHSD(aov(sbp~exercise, data=mydata))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = sbp ~ exercise, data = mydata)
##
## $exercise
##                     diff        lwr        upr     p adj
## Moderate-Low  -2.505538  -5.597805  0.5867282 0.1371181
## High-Low      -8.465465 -13.164759 -3.7661716 0.0001043
## High-Moderate -5.959927 -10.756192 -1.1636620 0.0105140

# You can also plot the post-hoc test
plot(TukeyHSD(aov(sbp~exercise, data=mydata))) ## The Moderate-Low paiwise crosses 0, and therefore considered as not significant
```

We can also run one-way ANOVA using a different package, and the best part about using **ggpubr** is that we can compare the mean *sbp* between intensity of **exercise** graphically.

```
library(ggpubr)

# Run Anova
ggpubr::compare_means(sbp ~ exercise, method="anova", data=mydata)
```

```
## # A tibble: 1 × 6
##   .y.          p   p.adj p.format p.signif method
##   <chr>    <dbl>   <dbl> <chr>    <chr>    <chr>
## 1 sbp   0.000152 0.00015 0.00015  ***      Anova

# Run pairwise statistics
ggpubr::compare_means(sbp ~ exercise, data=mydata)

## # A tibble: 3 × 8
##   .y.   group1   group2          p  p.adj p.format p.signif method
##   <chr> <chr>    <chr>       <dbl>  <dbl> <chr>    <chr>    <chr>
## 1 sbp   Low      Moderate 0.0896   0.09   0.08965  ns       Wilcoxon
## 2 sbp   Low      High     0.000435 0.0013 0.00044  ***      Wilcoxon
## 3 sbp   Moderate High     0.00229  0.0046 0.00229  **       Wilcoxon

# Specify the pairwise comparisons
mycompare <- list(c("Moderate","High"), c("Low", "Moderate"), c("Low","High")
)

# Plot the box-plot
ggboxplot(mydata, x="exercise", y="sbp",
          color = "exercise",
          add="jitter")+
  stat_compare_means(comparison = mycompare) +
  stat_compare_means(method="anova", label.y = 170)
```



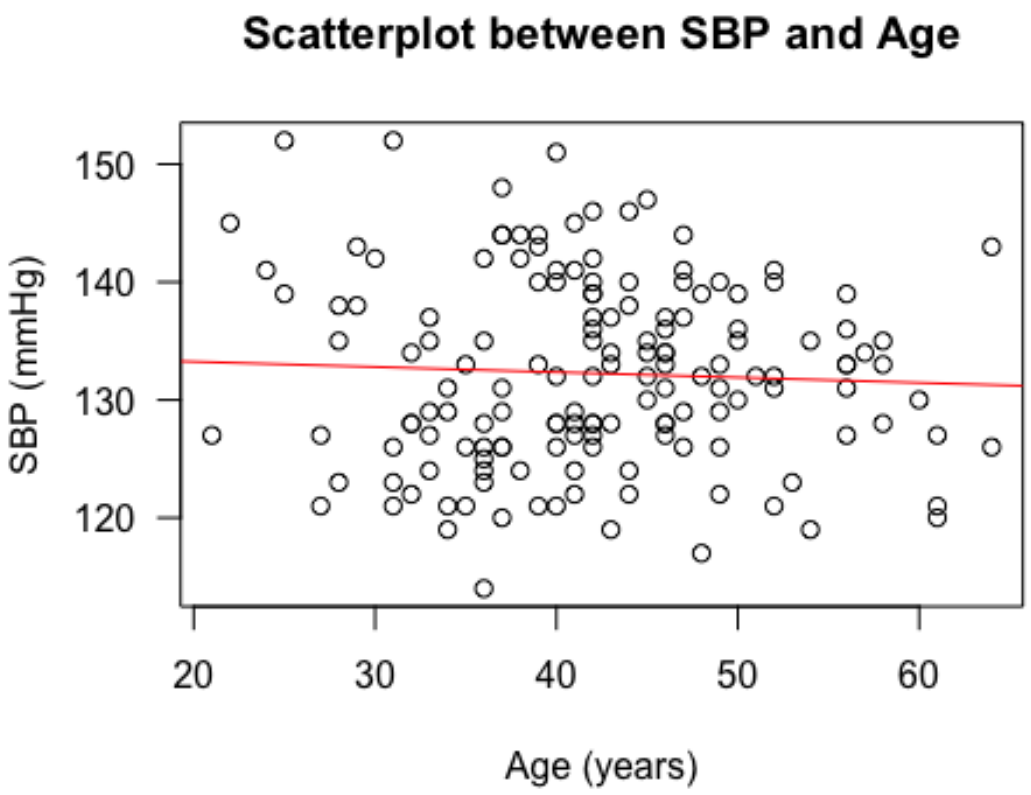
The analysis shows that those who reported doing high exercise intensity had lower **sbp** compared to those with Moderate and Low (P<0.1 and P=0.1 respectively). No difference of mean **sbp** noted between Moderate and Low exercise intensity.

Correlation

For this exercise we will use **sbp** and **age**. Both are numericals and the statistics used is Correlation Coefficient Test. There are two types of Correlation Test, either Pearson or Spearman Correlation Coefficient Tests depending on the distribution of the variables. If any of the variables is not normally distributed, we use Spearman's.

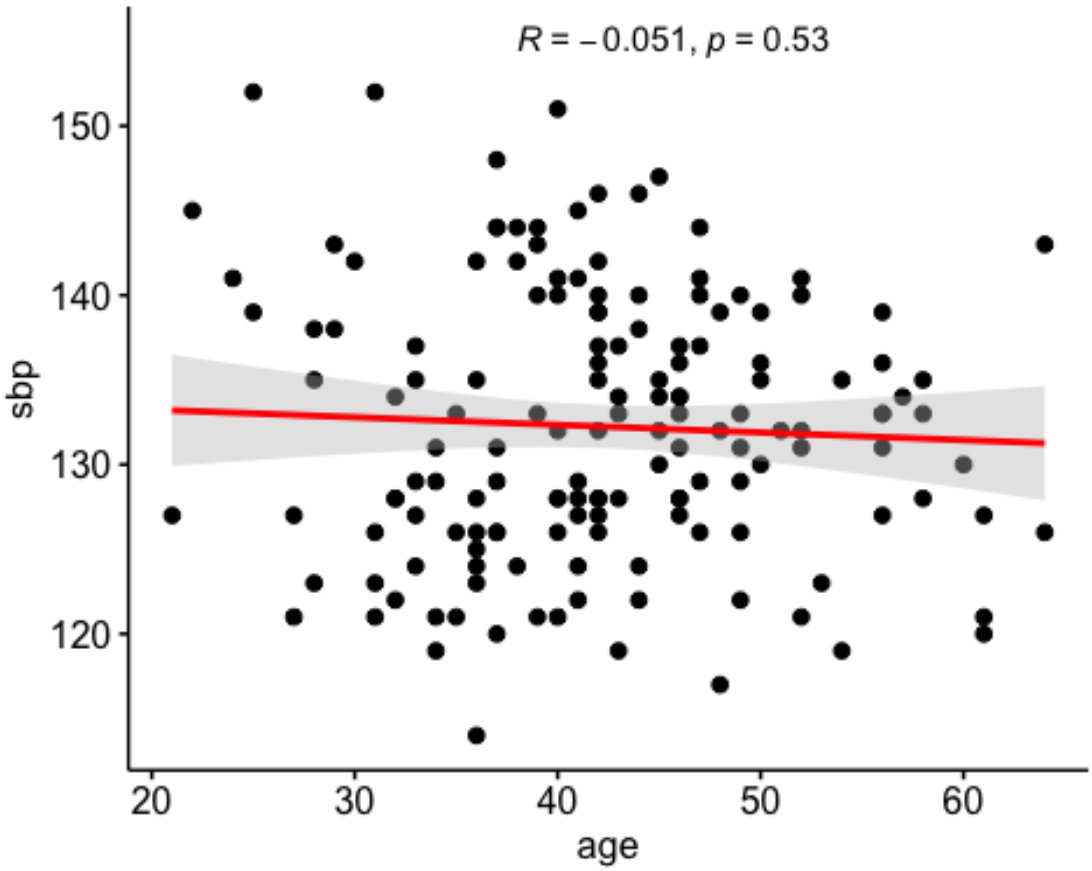
We can start by visualising the distribution using Scatterplot.

```
plot(mydata$sbp~mydata$age,
     main="Scatterplot between SBP and Age",
     xlab="Age (years)",
     ylab="SBP (mmHg)",
     las=1)
abline(lm(mydata$sbp~mydata$age), col="red")
```




```
ggscatter(mydata, x="age", y="sbp",
          color = "black",
          add = "reg.line",
          add.params = list(color = "red", fill = "grey"),
          conf.int = TRUE,
          cor.coef = TRUE,
          cor.coeff.args = list(method = "pearson", label.x = 38, label.y =
155)
          )
```

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



And to check whether there is indeed a statistically significant correlation, we can run Pearson's Correlation Coefficient Test because both sbp and age are normally distributed.

```
cor.test(mydata$sbp, mydata$age, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  mydata$sbp and mydata$age
## t = -0.62213, df = 151, p-value = 0.5348
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2075763  0.1089889
## sample estimates:
##         cor
## -0.05056365
```

The test shows that there is no significant correlation (rho=-0.050, P=0.535) between **sbp** and **age**. We will only describe the correlation value (rho) if we find significant correlation. Usually when it is not significant, the rho value is approaching zero. The highest correlation is either 1 or -1. Negative value show that when one value increases, the other decreases.

Comparing proportions

To illustrate this exercise we need two categorical variables. Since the previous exercise dealing with **sbp** as the dependent variable, there is no need for us to compare two categorical variables.

So now let us create a new set of dataset we named as **mydata2** which has additional variable, Blood pressure status (**bp**) which is formed based on **sbp** and **dbp**.

```
mydata2 <- healthstatus %>%
  dplyr::select(age,sex, exercise, smoking, ht, wt, sbp, dbp) %>%
  mutate(bmi = wt/(ht/100)^2) %>%
  mutate(bmistat = if_else(bmi < 25, "Normal",
                           if_else(bmi<30, "Overweight", "Obese"))) %>%
  mutate(bp = if_else(sbp < 140 & dbp < 90, "Normal", "High"))

# Updating the level
mydata2$sex <- factor(mydata2$sex, levels = c("Male","Female"))
mydata2$exercise <- factor(mydata2$exercise, levels = c("Low", "Moderate",
"High"))
mydata2$smoking <- factor(mydata2$smoking, levels = c("Yes", "No"))
mydata2$bmistat <- factor(mydata2$bmistat, levels = c("Normal", "Overweight",
"Obese"))
mydata2$bp <- factor(mydata2$bp, levels = c("High", "Normal"))

library(labelled)
var_label(mydata2) <- list(
                        age ="Age(years)",
                        sex = "Sex",
                        exercise = "Exercise intensity",
                        smoking = "Smoking",
                        wt = "Weight (kg)",
                        ht = "Height (cm)",
                        sbp = "SBP (mmHg)",
                        dbp = "DBP (mmHg)",
                        bmi = "BMI (kg/m2)",
                        bmistat = "BMI status",
                        bp = "Blood pressure"
)
```

Now let us cross tabulate **bp** with **bmistat**

```
# Using base R
crosstab1 <- table(mydata2$bp, mydata2$bmistat)
prop.table(crosstab1) # To produce proportion

##
##               Normal Overweight      Obese
##   High   0.11764706 0.15032680 0.11764706
##   Normal 0.41176471 0.16339869 0.03921569
```

```
chisq.test(crosstab1)

##
##  Pearson's Chi-squared test
##
## data:  crosstab1
## X-squared = 24.351, df = 2, p-value = 5.155e-06

# Using tidyverse and gtsummary
mydata2 %>%
  select(bp,bmistat) %>%
  tbl_summary(by=bmistat) %>%
  add_p(pvalue_fun = ~style_pvalue(.x, digits = 3))

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	**Normal**, N = 81	**Overweight**, N = 48	**Obese**, N = 24	**p-value**
Blood pressure				<0.001
High	18 (22%)	23 (48%)	18 (75%)	
Normal	63 (78%)	25 (52%)	6 (25%)	

The statistics show that there is no significant relationship between BMI and blood pressure status (chi-sq(2)=0.867, P=0.648).

Now we check the relationship of BP with sex, exercise and smoking.

```
mydata2 %>%
  select(sex, exercise, smoking, bmistat, bp) %>%
  tbl_summary(by = bp, percent = "row") %>%
  add_p(pvalue_fun = ~style_pvalue(.x, digits = 3))

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	**High**, N = 59	**Normal**, N = 94	**p-value**
Sex			0.737
Male	31 (37%)	52 (63%)	

Characteristic	**High**, N = 59	**Normal**, N = 94	**p-value**
Female	28 (40%)	42 (60%)	
Exercise intensity			<0.001
Low	39 (53%)	35 (47%)	
Moderate	19 (31%)	42 (69%)	
High	1 (5.6%)	17 (94%)	
Smoking	36 (57%)	27 (43%)	<0.001
BMI status			<0.001
Normal	18 (22%)	63 (78%)	
Overweight	23 (48%)	25 (52%)	
Obese	18 (75%)	6 (25%)	

The output shows that Exercise intensity, Smoking and BMI are related with Blood pressure.

Those with high blood pressure are those with lower exercise intensity, smoking and overweight.

REGRESSION

Regression is the extension of correlation. Regression shows how much changes in independent variable (X) influence changes in the dependent variable (Y). In contrast, correlation does not take into consideration which variable is the cause, and which is the effect.

Regression

Linear regression assumes linear relationship between independent and dependent variables. General linear model (GLM) assumes the residuals or errors follow a normal

distribution while generalized linear model (GzLM, and some use GLIM as the acronym) does not. Example of GLM includes ANOVA, ANCOVA, MANOVA and MANCOVA.

LINEAR REGRESSION

What are the factors that significantly affecting systolic blood pressure (sbp)? We can examine the factors (X) that affect sbp by using linear regression because sbp, which is the dependent variable (Y), is a numerical values. To run linear regression, we need to fulfill few assumptions:

1. Linearity: The relationship between X and the mean of Y is linear.
2. Homoscedasticity: The variance of residual is the same for any value of X.
3. Independence: Observations are independent of each other.
4. Normality: For any fixed value of X, Y is normally distributed.

(Ref: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html)

To illustrate this analysis we will use the same data as above to determine factors that are related with systolic blood pressure (sbp). There are 5 independent variables (predictors) in this exercise: age, sex, exercise, smoking and bmi.

Simple linear regression

Simple linear regression is a bivariable analysis because there are only two variables involved in the analysis. Previously we analysed sbp with age using correlation test. We can also test the relationship using linear regression.

```
model1 <- lm(sbp~age, data=mydata2)
summary(model1)
```

```
##
## Call:
## lm(formula = sbp ~ age, data = mydata2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.5260  -5.9404  -0.1206   5.9786  19.2488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 134.14745    3.11985  42.998   <2e-16 ***
## age          -0.04504    0.07240  -0.622    0.535
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.973 on 151 degrees of freedom
## Multiple R-squared:  0.002557,   Adjusted R-squared:  -0.004049
## F-statistic: 0.387 on 1 and 151 DF,  p-value: 0.5348
```

```
Anova(model1)
```

```
## Anova Table (Type II tests)
##
## Response: sbp
##           Sum Sq  Df F value Pr(>F)
## age         24.6   1   0.387 0.5348
## Residuals 9598.0 151
```

Age is not significantly related with SBP (P=0.535).

```
model2 <- lm(sbp ~ sex, data=mydata2)
summary(model2)
```

```
##
## Call:
## lm(formula = sbp ~ sex, data = mydata2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.952  -5.952  -0.600   6.400  19.400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 131.9518     0.8755 150.715   <2e-16 ***
## sexFemale     0.6482     1.2944   0.501    0.617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.976 on 151 degrees of freedom
## Multiple R-squared:  0.001658,   Adjusted R-squared:  -0.004953
## F-statistic: 0.2508 on 1 and 151 DF,  p-value: 0.6173
```

```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: sbp
##            Df Sum Sq Mean Sq F value Pr(>F)
## sex         1   16.0  15.955  0.2508 0.6173
## Residuals 151 9606.6  63.620
```

There is no linear association between sbp and sex (F(1, 151)=0.251, P=0.617).

```
model3 <- lm(sbp ~ exercise, data=mydata2)
summary(model3)
```

```
##
## Call:
## lm(formula = sbp ~ exercise, data = mydata2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -15.2432  -5.7377   0.2623   5.2623  17.7568
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      134.2432     0.8781 152.881  < 2e-16 ***
## exerciseModerate  -2.5055     1.3063  -1.918    0.057 .
## exerciseHigh      -8.4655     1.9852  -4.264 3.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.554 on 150 degrees of freedom
## Multiple R-squared:  0.1106, Adjusted R-squared:  0.09872
## F-statistic: 9.324 on 2 and 150 DF,  p-value: 0.0001525
```

```
Anova(model3)
```

```
## Anova Table (Type II tests)
##
## Response: sbp
##           Sum Sq  Df F value    Pr(>F)
## exercise  1064.0   2  9.3243 0.0001525 ***
## Residuals 8558.5 150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lower the intensity, the higher is the SBP. The differences observed is significant (F(2,150)=9.32, P=0.0002).

Next is determining relationship between sbp and smoking.

```
model4 <- lm(sbp ~ smoking, data=mydata2)
summary(model4)
```

```
##
## Call:
## lm(formula = sbp ~ smoking, data = mydata2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3444  -5.5397  -0.3444   6.4603  20.6556
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 133.5397     0.9964  134.03   <2e-16 ***
## smokingNo    -2.1952     1.2991   -1.69   0.0931 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.908 on 151 degrees of freedom
```

```
## Multiple R-squared:  0.01856,    Adjusted R-squared:  0.01206
## F-statistic: 2.855 on 1 and 151 DF,  p-value: 0.09313
```

```
Anova(model4)
```

```
## Anova Table (Type II tests)
##
## Response: sbp
##           Sum Sq  Df F value  Pr(>F)
## smoking    178.6   1  2.8555 0.09313 .
## Residuals 9444.0 151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is no relationship between sbp and smoking (F(1, 151)=2.855, P=0.093).

Now we need to analyse the relationship between sbp and bmistat.

```
model5 <- lm(sbp ~ bmistat, data=mydata2)
summary(model5)
```

```
##
## Call:
## lm(formula = sbp ~ bmistat, data = mydata2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4198  -5.8125   0.1875   4.7083  21.5802
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       130.4198     0.8475 153.895  < 2e-16 ***
## bmistatOverweight   2.3927     1.3893   1.722 0.087080 .
## bmistatObese        6.8719     1.7726   3.877 0.000158 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.627 on 150 degrees of freedom
## Multiple R-squared:  0.09317,    Adjusted R-squared:  0.08108
## F-statistic: 7.706 on 2 and 150 DF,  p-value: 0.0006521
```

```
anova(model5)
```

```
## Analysis of Variance Table
##
## Response: sbp
##            Df Sum Sq Mean Sq F value    Pr(>F)
## bmistat     2  896.6  448.28   7.706 0.0006521 ***
## Residuals 150 8726.0   58.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

BMI is significantly related with SBP (F(2,150)=7.71, P=0.0007).

So far based on these simple linear regressions significant factors that are related with SBP are exercise and BMI.

```
model1 %>%
  tbl_regression(pvalue_fun = ~style_pvalue(.x, digits = 3),)

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	Beta	95% CI	p-value
Age(years)	-0.05	-0.19, 0.10	0.535

```
model2 %>%
  tbl_regression(pvalue_fun = ~style_pvalue(.x, digits = 3),)

## • Install "flextable" with the code below.
## install.packages("flextable")
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	Beta	95% CI	p-value
Sex			
Male	—	—	
Female	0.65	-1.9, 3.2	0.617

```
model3 %>%
  tbl_regression(pvalue_fun = ~style_pvalue(.x, digits = 3),)

## • Install "flextable" with the code below.
## install.packages("flextable")
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	Beta	95% CI	p-value
Exercise intensity			
Low	—	—	
Moderate	-2.5	-5.1, 0.08	0.057
High	-8.5	-12, -4.5	<0.001

```
model4 %>%
  tbl_regression(pvalue_fun = ~style_pvalue(.x, digits = 3),)
```

```
## • Install "flextable" with the code below.
## install.packages("flextable")
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	Beta	95% CI	p-value
Smoking			
Yes	—	—	
No	-2.2	-4.8, 0.37	0.093

```
model5 %>%
  tbl_regression(pvalue_fun = ~style_pvalue(.x, digits = 3),)
```

```
## • Install "flextable" with the code below.
## install.packages("flextable")
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

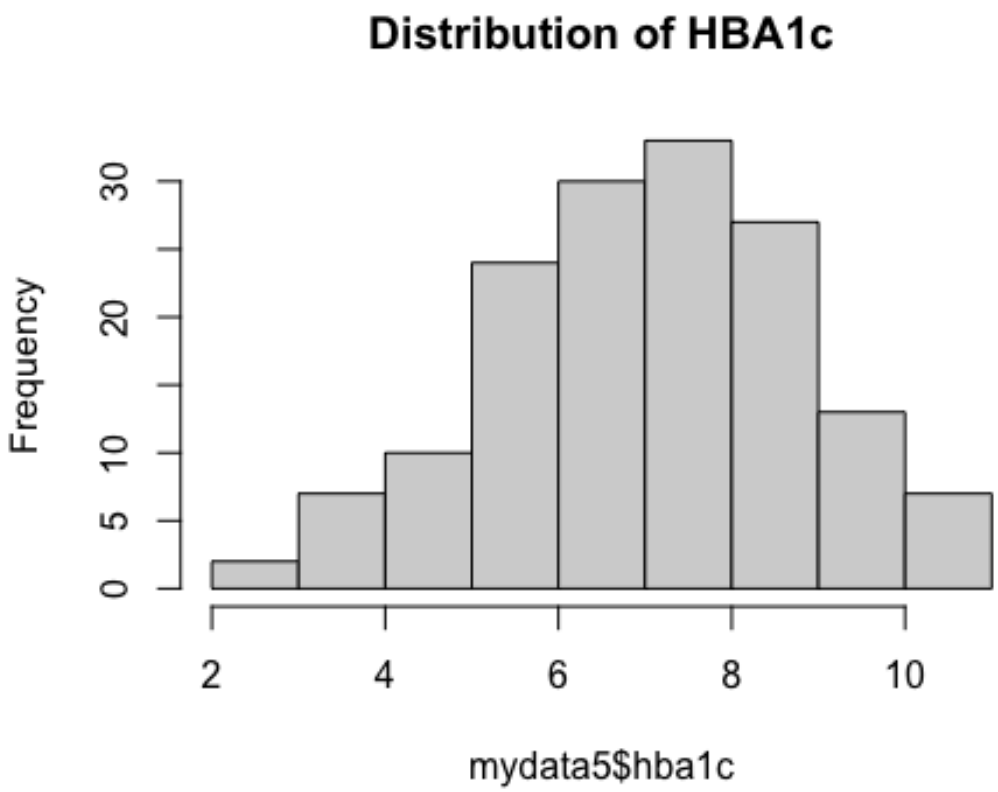
Characteristic	Beta	95% CI	p-value
BMI status			
Normal	—	—	
Overweight	2.4	-0.35, 5.1	0.087
Obese	6.9	3.4, 10	<0.001

Multiple linear regression

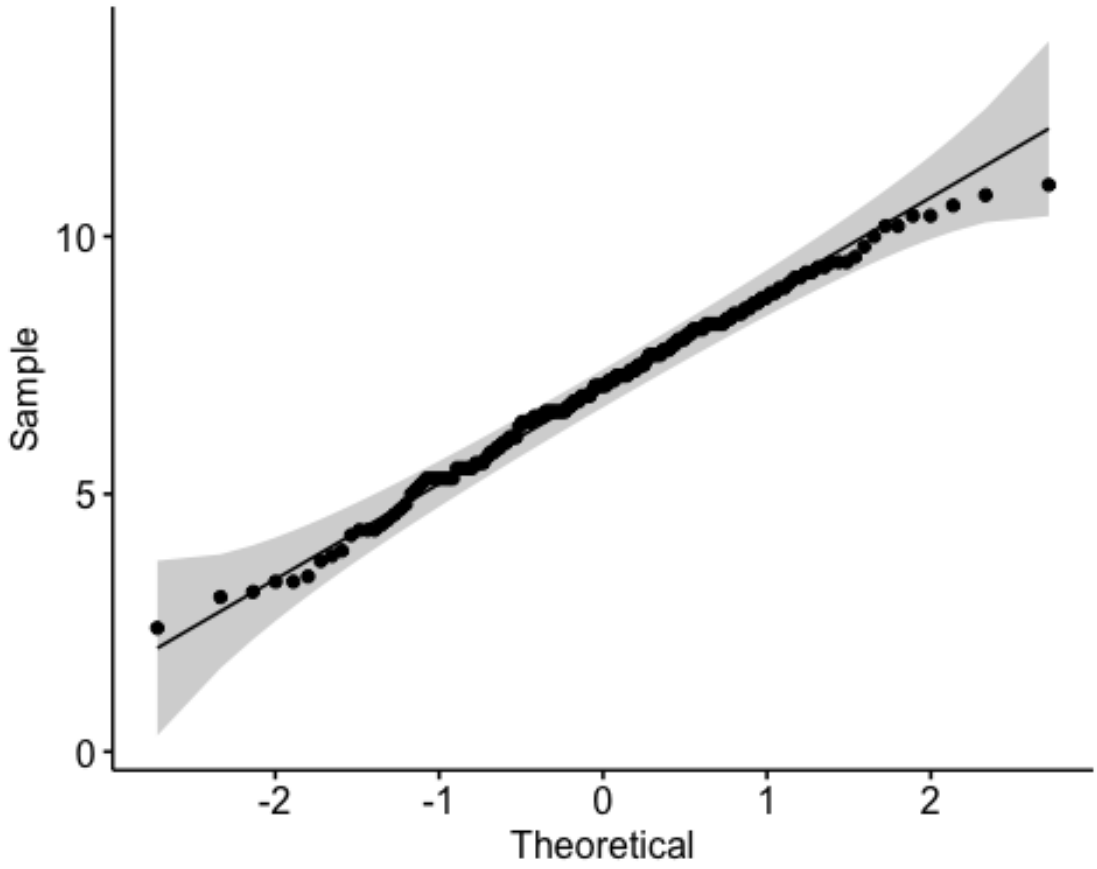
In this exercise, we will use HbA1c as the outcome of the study, and the independent variables are age, sex, exercise, smoking, bmi status and blood pressure status.

First we build the mydata5 which contain age, sex, exercise, smoking, bmicat, bp and hba1c

```
mydata3 <- healthstatus %>%
  dplyr::select(age, sex, exercise, smoking, hba1c)

mydata4 <- mydata2 %>%
  dplyr::select(bmistat, bp)

mydata5 <- cbind(mydata3,mydata4)

hist(mydata5$hba1c, main="Distribution of HBA1c")
```




```
ggqqplot(mydata5$hba1c)
```




```
shapiro.test(mydata5$hba1c)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mydata5$hba1c
## W = 0.99205, p-value = 0.555
```

Based on all the 3 analyses, we can conclude that hba1c is normally distributed (W=0.991, P=0.542).

Univariate model (unadjusted)

Let's check factors that are related to hba1c, one-by-one (or univariately).

```
summary(lm(hba1c~age, data=mydata5))
```

```
##
## Call:
## lm(formula = hba1c ~ age, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7208 -1.2129  0.1073  1.2295  3.9439
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.35087    0.70030  10.497   <2e-16 ***
## age         -0.00719    0.01625  -0.442    0.659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.79 on 151 degrees of freedom
## Multiple R-squared:  0.001295,   Adjusted R-squared:  -0.005319
## F-statistic: 0.1958 on 1 and 151 DF,  p-value: 0.6588
```

```
summary(lm(hba1c~sex, data=mydata5))
```

```
##
## Call:
## lm(formula = hba1c ~ sex, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0024 -1.0271 -0.0024  1.0976  4.3729
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6271     0.2089  31.719  < 2e-16 ***
## sexMale       0.7753     0.2837   2.733  0.00702 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.748 on 151 degrees of freedom
## Multiple R-squared:  0.04713,    Adjusted R-squared:  0.04082
## F-statistic: 7.469 on 1 and 151 DF,  p-value: 0.007025
```

Male has significantly higher hba1c compared to female (P=0.007).

```
summary(lm(hba1c~exercise, data=mydata5))
```

```
##
## Call:
## lm(formula = hba1c ~ exercise, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8676 -1.2676  0.0778  1.2324  3.7324
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.5222     0.4194  15.551   <2e-16 ***
## exerciseLow        0.7453     0.4676   1.594    0.113
## exerciseModerate   0.4138     0.4773   0.867    0.387
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.779 on 150 degrees of freedom
## Multiple R-squared:  0.01922,    Adjusted R-squared:  0.006145
## F-statistic:  1.47 on 2 and 150 DF,  p-value: 0.2332
```

Lower exercise intensity is found to have higher HbA1c (P=0.037).

```
summary(lm(hba1c~smoking, data=mydata5))

##
## Call:
## lm(formula = hba1c ~ smoking, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3633 -1.2633  0.1367  1.2367  4.2367
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.7633     0.1853  36.503   <2e-16 ***
## smokingYes    0.6906     0.2887   2.392    0.018 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.758 on 151 degrees of freedom
## Multiple R-squared:  0.0365, Adjusted R-squared:  0.03012
## F-statistic: 5.721 on 1 and 151 DF,  p-value: 0.01799
```

Smoking is significanly related with higher HbA1c (P<0.001).

```
summary(lm(hba1c~bmistat, data=mydata5))

##
## Call:
## lm(formula = hba1c ~ bmistat, data = mydata5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.422 -1.122  0.000  1.200  4.178
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.8222     0.1966  34.701   <2e-16 ***
## bmistatOverweight   0.2778     0.3223   0.862   0.3901
## bmistatObese        0.8819     0.4112   2.145   0.0336 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.769 on 150 degrees of freedom
```

```
## Multiple R-squared:  0.03014,    Adjusted R-squared:  0.0172
## F-statistic:  2.33 on 2 and 150 DF,  p-value: 0.1008
```

HbA1c is however not related with BMI status.

```
summary(lm(hba1c~bp, data=mydata5))

##
## Call:
## lm(formula = hba1c ~ bp, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1734 -1.0734  0.0266  1.1266  3.8266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.8034     0.2195  35.545  < 2e-16 ***
## bpNormal     -1.2300     0.2801  -4.391 2.11e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 151 degrees of freedom
## Multiple R-squared:  0.1133, Adjusted R-squared:  0.1074
## F-statistic: 19.29 on 1 and 151 DF,  p-value: 2.107e-05
```

HbA1c is found to be lower among those with normal blood pressure (P<0.001).

The simple linear regressions show that sex, exercise, smoking and high blood pressure are associated with hba1c. Are they independent predictors for hba1c?

Multivariate model

```
model3 <- lm(hba1c ~ sex + exercise + smoking + bp, data=mydata5)
summary(model3)

##
## Call:
## lm(formula = hba1c ~ sex + exercise + smoking + bp, data = mydata5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5126 -1.1762  0.0958  1.0958  3.6915
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.17251    0.52439  13.678  < 2e-16 ***
## sexMale           0.77249    0.27494   2.810 0.005635 **
## exerciseLow       0.13597    0.45731   0.297 0.766638
## exerciseModerate  0.09368    0.45301   0.207 0.836453
## smokingYes        0.17201    0.29512   0.583 0.560894
## bpNormal         -1.16833    0.30668  -3.810 0.000204 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.657 on 147 degrees of freedom
## Multiple R-squared:  0.1669, Adjusted R-squared:  0.1385
## F-statistic: 5.889 on 5 and 147 DF,  p-value: 5.435e-05

anova(model3)

## Analysis of Variance Table
##
## Response: hba1c
##            Df Sum Sq Mean Sq F value   Pr(>F)
## sex         1  22.82  22.824  8.3168 0.004519 **
## exercise    2   8.41   4.206  1.5324 0.219437
## smoking     1   9.74   9.744  3.5507 0.061494 .
## bp          1  39.83  39.829 14.5134 0.000204 ***
## Residuals 147 403.41   2.744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

library(car)
Anova(model3, type=3)

## Anova Table (Type III tests)
##
## Response: hba1c
##             Sum Sq  Df  F value    Pr(>F)
## (Intercept) 513.42   1 187.0860 < 2.2e-16 ***
## sex          21.66   1   7.8942  0.005635 **
## exercise      0.25   2   0.0453  0.955749
## smoking       0.93   1   0.3397  0.560894
## bp           39.83   1  14.5134  0.000204 ***
## Residuals   403.41 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the multiple linear regression, sex and bp are related with HbA1c.

```
library(gvlma)
gvlma::gvlma(model3)

##
## Call:
## lm(formula = hba1c ~ sex + exercise + smoking + bp, data = mydata5)
##
## Coefficients:
##      (Intercept)           sexMale       exerciseLow  exerciseModerate
##          7.17251           0.77249           0.13597           0.09368
##       smokingYes          bpNormal
##          0.17201          -1.16833
```

```
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma::gvlma(x = model3)
##
##                      Value p-value                Decision
## Global Stat        2.198741  0.6993 Assumptions acceptable.
## Skewness           0.235729  0.6273 Assumptions acceptable.
## Kurtosis           0.757979  0.3840 Assumptions acceptable.
## Link Function      0.002713  0.9585 Assumptions acceptable.
## Heteroscedasticity 1.202320  0.2729 Assumptions acceptable.
```

How to intepret gvlma?

Global Stat Are the relationships between your X predictors and Y roughly linear?. Rejection of the null ($p < .05$) indicates a non-linear relationship between one or more of your X's and Y

Skewness Is your distribution skewed positively or negatively, necessitating a transformation to meet the assumption of normality? Rejection of the null ($p < .05$) indicates that you should likely transform your data.

Kurtosis Is your distribution kurtotic (highly peaked or very shallowly peaked), necessitating a transformation to meet the assumption of normality? Rejection of the null ($p < .05$) indicates that you should likely transform your data.

Link Function Is your dependent variable truly continuous, or categorical? Rejection of the null ($p < .05$) indicates that you should use an alternative form of the generalized linear model (e.g. logistic or binomial regression).

Heteroscedasticity Is the variance of your model residuals constant across the range of X (assumption of homoscedasticity)? Rejection of the null ($p < .05$) indicates that your residuals are heteroscedastic, and thus non-constant across the range of X. Your model is better/worse at predicting for certain ranges of your X scales.

LOGISTIC REGRESSION

Logistic regression is one of the form of generalised linear regression. We use logistic regression when the outcome or dependent variable is categorical.

Preparing the data

In this exercise, assuming we would like to study variables that are associated with High blood pressure. Blood pressure is a dichotomous variable with High and Normal as the categories.

Let's look at the data we are using for this exercise.

```
glimpse(mydata5)
```

```
## Rows: 153
## Columns: 7
## $ age      <dbl> 36, 49, 56, 61, 40, 42, 44, 41, 46, 32, 38, 28, 27, 22,
44, 3…
## $ sex      <chr> "Male", "Male", "Male", "Female", "Female", "Female",
"Male",…
## $ exercise <chr> "Moderate", "Moderate", "Low", "Low", "Low", "Moderate",
"Low…
## $ smoking  <chr> "Yes", "Yes", "No", "Yes", "Yes", "No", "Yes", "Yes",
"No", "…
## $ hba1c    <dbl> 9.3, 8.5, 6.8, 5.5, 4.6, 3.3, 8.9, 9.0, 7.7, 5.5, 5.5,
4.3, 7…
## $ bmistat  <fct> Overweight, Overweight, Obese, Normal, Normal,
Overweight, Ov…
## $ bp       <fct> Normal, High, High, Normal, High, Normal, High, High,
Normal,…
```

mydata5 has 7 variables, where:

Age - Numerical Sex - Character *Exercise - Character* Smoking status - Character *HbA1c - Numerical* BMI Status - Factor with 3 levels - Normal, Overweight & Obese *Blood pressure - Factor with 2 levels - Normal, High

We can observe that some variables do not assigned propert data type. So let's assign Sex, Exercise and Smoking status to categorical variable with their proper level and label.

```
# When changing value label, it is very important to specify the labels
according their order e.g. F before M, H before, L etc
mydata5$sex <- factor(mydata5$sex, labels = c("Female", "Male"))
mydata5$exercise <- factor(mydata5$exercise, labels = c("High", "Low",
"Moderate"))
mydata5$smoking <- factor(mydata5$smoking, labels = c("No", "Yes"))

glimpse(mydata5)
```

```
## Rows: 153
## Columns: 7
## $ age      <dbl> 36, 49, 56, 61, 40, 42, 44, 41, 46, 32, 38, 28, 27, 22,
44, 3…
## $ sex      <fct> Male, Male, Male, Female, Female, Female, Male, Male,
Female,…
## $ exercise <fct> Moderate, Moderate, Low, Low, Low, Moderate, Low, Low,
Modera…
## $ smoking  <fct> Yes, Yes, No, Yes, Yes, No, Yes, Yes, No, No, No, No,
Yes, No…
## $ hba1c    <dbl> 9.3, 8.5, 6.8, 5.5, 4.6, 3.3, 8.9, 9.0, 7.7, 5.5, 5.5,
4.3, 7…
```

```
## $ bmistat  <fct> Overweight, Overweight, Obese, Normal, Normal,
Overweight, Ov…
## $ bp       <fct> Normal, High, High, Normal, High, Normal, High, High,
Normal,…
```

Now we can see that all categorical variables have been changed to factor.

Running bivariable analyses

Again, to practice, when building any multivariate model, we need to decide what variables we want to include in the final model. For this exercise our equation is:

'bp <- age + sex + exercise + smoking + hba1c + bmistat'

We need to decide what variables to be included. The usual approach is to do bivariable analyses first, i.e.

bp ~ age bp ~ sex *bp ~ exercise* bp ~ smoking *bp ~ hba1c

Even though, we could straight away build the final model based on our literature review.

```
mydata5 %>%
  tbl_summary(by = bp,
              statistic = list(all_continuous() ~ "mean ({sd})",
                               all_categorical() ~ "{n}/{N} ({p}%)"),
              percent = "row") %>%
  add_p(pvalue_fun = ~style_pvalue(.x, digits = 3))

## • Install "flextable" with the code below.

## install.packages("flextable")

## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in the code chunk
header.
```

Characteristic	**High**, N = 59	**Normal**, N = 94	**p-value**
age	mean (9)	mean (9)	0.987
sex			0.737
Female	28/70 (40%)	42/70 (60%)	
Male	31/83 (37%)	52/83 (63%)	
exercise			<0.001
High	1/18 (5.6%)	17/18 (94%)	
Low	39/74 (53%)	35/74 (47%)	
Moderate	19/61 (31%)	42/61 (69%)	
smoking	36/63 (57%)	27/63 (43%)	<0.001
hba1c	mean (1.56)	mean (1.76)	<0.001

Characteristic	**High**, N = 59	**Normal**, N = 94	**p-value**
BMI status			<0.001
Normal	18/81 (22%)	63/81 (78%)	
Overweight	23/48 (48%)	25/48 (52%)	
Obese	18/24 (75%)	6/24 (25%)	

The analyses show that Exercise, Smoking, HbA1c and BMI are related with Blood pressure. So now we can build the logistic regression model as:

bp <- exercise + smoking + hba1c + bmistat

Simple Logistic Regression

Simple logistic regression involves only one dependent variable and one independent variable.

```
contrasts(mydata5$bp)

##        Normal
## High        0
## Normal      1
```

We can see that based on Blood Pressure value label, Normal is coded "1", and this make our logistic regression to predict Normal blood pressure instead of High blood pressure. That is not what we want. We should be predicting High blood pressure. Therefore we need to change the value level. R uses the first value label that appears alphabetically as the reference point, and this case it is High, because "H" appears first before "N". We can change this.

```
mydata$bp <- relevel(mydata5$bp, ref = "Normal")
contrasts(mydata$bp)

##        High
## Normal    0
## High      1
```

Now we can see that the reference point is Normal, and our model will be predicting High blood pressure.

```
# BP ~ Exercise intensity
slrexercise <- glm(bp ~ exercise, data=mydata5, family="binomial")
summary(slrexercise)

##
## Call:
## glm(formula = bp ~ exercise, family = "binomial", data = mydata5)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4043  -1.1318   0.8639   0.8639   1.2237
```

```
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.833      1.029   2.753   0.0059 **
## exerciseLow         -2.941      1.055  -2.788   0.0053 **
## exerciseModerate    -2.040      1.065  -1.915   0.0555 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 204.03  on 152  degrees of freedom
## Residual deviance: 185.77  on 150  degrees of freedom
## AIC: 191.77
##
## Number of Fisher Scoring iterations: 5
```

```
summary(slrexercise)$coef
```

```
##                   Estimate Std. Error   z value    Pr(>|z|)
## (Intercept)       2.833213   1.028970  2.753445 0.005897159
## exerciseLow      -2.941427   1.054984 -2.788124 0.005301432
## exerciseModerate -2.039983   1.065467 -1.914636 0.055538885
```

```
coef(slrexercise)
```

```
##      (Intercept)      exerciseLow exerciseModerate
##         2.833213        -2.941427        -2.039983
```

We can do the same for Smoking, HbA1c and BMI status.

```
# BP ~ Smoking
slrsmoking <- glm(bp ~ smoking, data=mydata5, family="binomial")
summary(slrsmoking)
```

```
##
## Call:
## glm(formula = bp ~ smoking, family = "binomial", data = mydata5)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6519  -1.0579   0.7683   0.7683   1.3018
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0692     0.2417   4.424 9.68e-06 ***
## smokingYes   -1.3569     0.3510  -3.865 0.000111 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 204.03  on 152  degrees of freedom
## Residual deviance: 188.35  on 151  degrees of freedom
## AIC: 192.35
##
## Number of Fisher Scoring iterations: 4

# BP ~ HbA1c
slrhba1c <- glm(bp ~ hba1c, data=mydata5, family="binomial")
summary(slrhba1c)

##
## Call:
## glm(formula = bp ~ hba1c, family = "binomial", data = mydata5)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8974  -1.1506   0.6646   0.9690   1.5977
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.6564     0.8437   4.334 1.47e-05 ***
## hba1c        -0.4428     0.1123  -3.942 8.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 204.03  on 152  degrees of freedom
## Residual deviance: 185.50  on 151  degrees of freedom
## AIC: 189.5
##
## Number of Fisher Scoring iterations: 3

# BP ~ BMI status
slrbmistat <- glm(bp ~ bmistat, data=mydata5, family="binomial")
summary(slrbmistat)

##
## Call:
## glm(formula = bp ~ bmistat, family = "binomial", data = mydata5)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.734  -1.213   0.709   0.709   1.665
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.2528     0.2673   4.687 2.77e-06 ***
## bmistatOverweight  -1.1694     0.3936  -2.971  0.00297 **
## bmistatObese       -2.3514     0.5419  -4.339 1.43e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 204.03  on 152  degrees of freedom
## Residual deviance: 179.26  on 150  degrees of freedom
## AIC: 185.26
##
## Number of Fisher Scoring iterations: 4
```

Even though we are using simple logistic regression for the above analyses, they are still bivariable analyses. To discover independent predictor to high blood pressure, we should do multiple logistic regression.

Multiple logistic regression

```
mlr <- glm(bp ~ exercise + smoking + hba1c + bmistat, data=mydata5, family =
binomial)
summary(mlr)

##
## Call:
## glm(formula = bp ~ exercise + smoking + hba1c + bmistat, family =
binomial,
##     data = mydata5)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4620  -0.7583   0.3037   0.7236   2.3718
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         7.2466     1.5865   4.568 4.93e-06 ***
## exerciseLow        -3.3381     1.1920  -2.800 0.005103 **
## exerciseModerate   -2.3521     1.1973  -1.965 0.049469 *
## smokingYes         -1.2906     0.4265  -3.026 0.002478 **
## hba1c              -0.3610     0.1279  -2.822 0.004770 **
## bmistatOverweight  -1.3057     0.4676  -2.792 0.005233 **
## bmistatObese       -2.4810     0.6441  -3.852 0.000117 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 204.03  on 152  degrees of freedom
## Residual deviance: 139.53  on 146  degrees of freedom
## AIC: 153.53
##
## Number of Fisher Scoring iterations: 6
```

```
summary(mlr)$coef

##                     Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)        7.2465596  1.5864980  4.567645 4.932347e-06
## exerciseLow       -3.3381426  1.1919994 -2.800457 5.103037e-03
## exerciseModerate  -2.3521309  1.1973007 -1.964528 4.946887e-02
## smokingYes        -1.2905595  0.4264920 -3.025987 2.478226e-03
## hba1c             -0.3609756  0.1279072 -2.822169 4.770003e-03
## bmistatOverweight -1.3056740  0.4675922 -2.792335 5.232918e-03
## bmistatObese      -2.4809774  0.6441388 -3.851619 1.173393e-04

coef(mlr)

##       (Intercept)       exerciseLow  exerciseModerate        smokingYes
##         7.2465596        -3.3381426        -2.3521309        -1.2905595
##             hba1c bmistatOverweight      bmistatObese
##        -0.3609756        -1.3056740        -2.4809774
```

REPEATED MEASURE ANOVA

-TBA-