

R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
date() # Current system time and date.
```

```
## [1] "Tue Jul 16 08:48:13 2024"
```

```
Sys.time() # Current system time and date (redundant).
```

```
## [1] "2024-07-16 08:48:13 +07"
```

```
R.version.string # R version and version release date.
```

```
## [1] "R version 4.2.3 (2023-03-15 ucrt)"
```

```
options(digits=6) # Confirm default digits.  
options(scipen=999) # Suppress scientific notation.  
options(width=60) # Confirm output width.  
ls() # List all objects in the working # directory.
```

```
## character(0)
```

```
rm(list = ls()) # CAUTION: Remove all files in the #working directory. If this action is not desired, u  
ls.str() # List all objects with finite detail.  
getwd() # Identify the current working directory
```

```
## [1] "C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics"
```

```
setwd("C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics") # Set to a new wor  
getwd()# Confirm the working directory.
```

```
## [1] "C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics"
```

```
list.files()# List files at the PC directory
```

```

## [1] "assignment.csv"
## [2] "BreedMilkLb365.txt"
## [3] "Chapter-1-Import Files.Rmd"
## [4] "Chapter-2-exploration-visualization.Rmd"
## [5] "Chapter-3-distribution-correlation.nb.html"
## [6] "Chapter-3-distribution-correlation.Rmd"
## [7] "CPIIISectionLbsGender.csv"
## [8] "data-exploration-SBP.html"
## [9] "data-exploration-SBP.Rmd"
## [10] "datasets"
## [11] "distribution-correlation.pdf"
## [12] "EmpWellAge21to79.csv"
## [13] "GenderEndurance.csv"
## [14] "graphs"
## [15] "nhanes_2015_2016.csv"
## [16] "Rplot-Frequency of variables.png"
## [17] "Rplot - Distribution of factor-type object variables gender, RaceEthnicity, BMIStatus, and obe
## [18] "Rplot - Distribution of numeric-type object variable age.png"
## [19] "SmartWayVehicleListMY2008.csv"
## [20] "Sorghum2012to2016.xlsx"
## [21] "YearSoilTypeCropRainYieldBushelsPerAcreNoHeader.txt"

```

```
.libPaths()# Library pathname
```

```

## [1] "C:/Users/linan/AppData/Local/R/win-library/4.2"
## [2] "C:/Program Files/R/R-4.2.3/library"

```

```
.Library# Library pathname.
```

```

## [1] "C:/PROGRA~1/R/R-42~1.3/library"

```

```
sessionInfo() # R version, locale, and packages.
```

```

## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22631)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Indonesia.utf8
## [2] LC_CTYPE=English_Indonesia.utf8
## [3] LC_MONETARY=English_Indonesia.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_Indonesia.utf8
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets
## [6] methods    base
##
## loaded via a namespace (and not attached):
## [1] compiler_4.2.3    fastmap_1.1.1    cli_3.6.1

```

```
## [4] tools_4.2.3      htmltools_0.5.8    rstudioapi_0.16.0
## [7] yaml_2.3.8       rmarkdown_2.26     knitr_1.45
## [10] xfun_0.40        digest_0.6.31     rlang_1.1.1
## [13] evaluate_0.23
```

```
search() # Attached packages and objects.
```

```
## [1] ".GlobalEnv"      "package:stats"
## [3] "package:graphics" "package:grDevices"
## [5] "package:utils"     "package:datasets"
## [7] "package:methods"   "Autoloads"
## [9] "package:base"
```

```
searchpaths() # Attached packages and objects.
```

```
## [1] ".GlobalEnv"
## [2] "C:/Program Files/R/R-4.2.3/library/stats"
## [3] "C:/Program Files/R/R-4.2.3/library/graphics"
## [4] "C:/Program Files/R/R-4.2.3/library/grDevices"
## [5] "C:/Program Files/R/R-4.2.3/library/utils"
## [6] "C:/Program Files/R/R-4.2.3/library/datasets"
## [7] "C:/Program Files/R/R-4.2.3/library/methods"
## [8] "Autoloads"
## [9] "C:/PROGRA~1/R/R-42~1.3/library/base"
```

```
#####
#####
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```

library(gmodels)
library(ggplot2)
library(tidyverse)

## -- Attaching core tidyverse packages ---- tidyverse 2.0.0 --
## vforcats    1.0.0      vstringr    1.5.1
## vlubridate  1.9.3      vtibble     3.2.1
## vpurrr      1.0.2      vtidyrm    1.3.1
## vreadr      2.1.5

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x Hmisc::src()       masks dplyr::src()
## x Hmisc::summarize() masks dplyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(beeswarm)

setwd("C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics")

EmployeeBiometric.df <- read.table(file="EmpWellAge21to79.csv", header=TRUE, dec=".\"", sep = ",") 

str(EmployeeBiometric.df)

## 'data.frame': 3945 obs. of 10 variables:
## $ EmployeeID : chr "ID10005" "ID10010" "ID10016" "ID10028" ...
## $ Gender      : int 1 1 1 2 1 1 1 2 1 1 ...
## $ AgeYears    : int 34 67 41 57 35 40 40 34 40 50 ...
## $ RaceEthnicity: int 5 5 5 1 5 3 2 5 2 2 ...
## $ TotalCholesterolmgdL: int 221 175 224 217 139 198 NA 205 87 149 ...
## $ SBPmmHg     : int 102 NA NA 90 86 114 NA 92 120 126 ...
## $ DBPmmHg     : int 44 NA NA 42 42 74 NA 48 80 72 ...
## $ BMIMetric   : num 18.4 18.7 21.6 20.6 18.7 ...
## $ BMISstatus  : int 1 2 2 2 2 2 2 1 2 ...
## $ Obesity     : int 1 1 1 1 1 1 1 1 1 1 ...

dim(EmployeeBiometric.df)

## [1] 3945 10

head(EmployeeBiometric.df)

##   EmployeeID Gender AgeYears RaceEthnicity
## 1 ID10005      1      34          5
## 2 ID10010      1      67          5
## 3 ID10016      1      41          5
## 4 ID10028      2      57          1
## 5 ID10041      1      35          5

```

```

## 6     ID10051      1      40      3
##   TotalCholesterolmgdL SBPmmHg DBPmmHg BMIMetric BMIStatus
## 1             221       102      44    18.35      1
## 2             175        NA      NA    18.66      2
## 3             224        NA      NA    21.61      2
## 4             217        90      42    20.59      2
## 5             139        86      42    18.66      2
## 6             198       114      74    19.73      2
##   Obesity
## 1     1
## 2     1
## 3     1
## 4     1
## 5     1
## 6     1

```

Check for duplicate in individual level = no duplicate ID

```
table(duplicated(EmployeeBiometric.df$EmployeeID))
```

```

##
## FALSE
## 3945

```

```
summary(EmployeeBiometric.df)
```

```

##   EmployeeID          Gender      AgeYears
##   Length:3945      Min. :1.00    Min. :21.0
##   Class :character  1st Qu.:1.00   1st Qu.:34.0
##   Mode  :character  Median :1.00   Median :47.0
##                      Mean   :1.47   Mean   :47.7
##                      3rd Qu.:2.00  3rd Qu.:61.0
##                      Max.   :2.00   Max.   :79.0
##
##   RaceEthnicity TotalCholesterolmgdL      SBPmmHg
##   Min.   :1.0    Min.   : 78      Min.   : 74
##   1st Qu.:2.0   1st Qu.:151     1st Qu.:108
##   Median :3.0   Median :175     Median :118
##   Mean   :3.4   Mean   :180     Mean   :122
##   3rd Qu.:5.0   3rd Qu.:205     3rd Qu.:132
##   Max.   :5.0   Max.   :380     Max.   :222
##   NA's   :894    NA's   :209     NA's   :209
##
##   DBPmmHg          BMIMetric      BMIStatus
##   Min.   :34.0    Min.   :13.4   Min.   :1.00
##   1st Qu.:60.0    1st Qu.:22.6   1st Qu.:2.00
##   Median :70.0    Median :26.4   Median :3.00
##   Mean   :69.3    Mean   :27.7   Mean   :2.86
##   3rd Qu.:78.0    3rd Qu.:31.3   3rd Qu.:4.00
##   Max.   :122.0   Max.   :70.1   Max.   :4.00
##   NA's   :217     NA's   :20     NA's   :20
##
##   Obesity
##   Min.   :1.00

```

```

## 1st Qu.:1.00
## Median :1.00
## Mean   :1.31
## 3rd Qu.:2.00
## Max.   :2.00
## NA's    :20

##Change EmployeeID,Gender, Ethnic, and BMISatus from int to factor

EmployeeBiometric.df$EmployeeID <- as.factor(EmployeeBiometric.df$EmployeeID)

EmployeeBiometric.df$Gender <- factor(EmployeeBiometric.df$Gender,
                                         labels=c("Female","Male"))
levels(EmployeeBiometric.df$Gender)

## [1] "Female" "Male"

EmployeeBiometric.df$BMISatus<-factor(EmployeeBiometric.df$BMISatus,
                                         labels = c("Underweight","Normal Weight", "Overweight","Obese"))
levels(EmployeeBiometric.df$BMISatus)

## [1] "Underweight"    "Normal Weight"   "Overweight"
## [4] "Obese"

EmployeeBiometric.df$RaceEthnicity <-
  factor(EmployeeBiometric.df$RaceEthnicity,
         labels=c("Asian","Black","Hispanic","Other","White"))
levels(EmployeeBiometric.df$RaceEthnicity)

## [1] "Asian"      "Black"      "Hispanic"   "Other"      "White"

EmployeeBiometric.df$Obesity <-
  factor(EmployeeBiometric.df$Obesity,
         labels=c("Not Obese","Obese"))
levels(EmployeeBiometric.df$Obesity)

## [1] "Not Obese" "Obese"

Change AgeYears, SBPmmHg, DBPmmHg from int to numeric

EmployeeBiometric.df$AgeYears <- as.numeric(EmployeeBiometric.df$AgeYears)
EmployeeBiometric.df$SBPmmHg <- as.numeric(EmployeeBiometric.df$SBPmmHg)
EmployeeBiometric.df$DBPmmHg <- as.numeric(EmployeeBiometric.df$DBPmmHg)

Check data structure

str(EmployeeBiometric.df)

```

```

## 'data.frame':   3945 obs. of  10 variables:
## $ EmployeeID      : Factor w/ 3945 levels "ID10005","ID10010",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Gender          : Factor w/ 2 levels "Female","Male": 1 1 1 2 1 1 1 2 1 1 ...
## $ AgeYears         : num  34 67 41 57 35 40 40 34 40 50 ...
## $ RaceEthnicity    : Factor w/ 5 levels "Asian","Black",...: 5 5 5 1 5 3 2 5 2 2 ...
## $ TotalCholesterolmgdL: int  221 175 224 217 139 198 NA 205 87 149 ...
## $ SBPmmHg          : num  102 NA NA 90 86 114 NA 92 120 126 ...
## $ DBPmmHg          : num  44 NA NA 42 42 74 NA 48 80 72 ...
## $ BMIMetric        : num  18.4 18.7 21.6 20.6 18.7 ...
## $ BMISstatus       : Factor w/ 4 levels "Underweight",...: 1 2 2 2 2 2 2 2 1 2 ...
## $ Obesity          : Factor w/ 2 levels "Not Obese","Obese": 1 1 1 1 1 1 1 1 1 1 ...

getwd()

## [1] "C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics"

ls()

## [1] "EmployeeBiometric.df"

attach(EmployeeBiometric.df)
dim(EmployeeBiometric.df)

## [1] 3945   10

names(EmployeeBiometric.df)

##  [1] "EmployeeID"           "Gender"
##  [3] "AgeYears"             "RaceEthnicity"
##  [5] "TotalCholesterolmgdL" "SBPmmHg"
##  [7] "DBPmmHg"              "BMIMetric"
##  [9] "BMISstatus"           "Obesity"

colnames(EmployeeBiometric.df)

##  [1] "EmployeeID"           "Gender"
##  [3] "AgeYears"             "RaceEthnicity"
##  [5] "TotalCholesterolmgdL" "SBPmmHg"
##  [7] "DBPmmHg"              "BMIMetric"
##  [9] "BMISstatus"           "Obesity"

head(EmployeeBiometric.df)

##   EmployeeID Gender AgeYears RaceEthnicity
## 1 ID10005 Female    34     White
## 2 ID10010 Female    67     White
## 3 ID10016 Female    41     White
## 4 ID10028 Male      57     Asian
## 5 ID10041 Female    35     White

```

```

## 6 ID10051 Female 40 Hispanic
## TotalCholesterolmgdL SBPmmHg DBPmmHg BMIMetric
## 1 221 102 44 18.35
## 2 175 NA NA 18.66
## 3 224 NA NA 21.61
## 4 217 90 42 20.59
## 5 139 86 42 18.66
## 6 198 114 74 19.73
## BMIStatus Obesity
## 1 Underweight Not Obese
## 2 Normal Weight Not Obese
## 3 Normal Weight Not Obese
## 4 Normal Weight Not Obese
## 5 Normal Weight Not Obese
## 6 Normal Weight Not Obese

```

```
tail(EmployeeBiometric.df)
```

```

## EmployeeID Gender AgeYears RaceEthnicity
## 3940 ID50080 Male 57 White
## 3941 ID50087 Female 44 Asian
## 3942 ID50104 Male 45 Asian
## 3943 ID50108 Female 28 Hispanic
## 3944 ID50109 Female 57 White
## 3945 ID50126 Male 57 White
## TotalCholesterolmgdL SBPmmHg DBPmmHg BMIMetric
## 3940 152 98 62 NA
## 3941 NA 108 64 NA
## 3942 NA 138 74 NA
## 3943 186 156 88 NA
## 3944 139 NA NA NA
## 3945 170 NA NA NA
## BMIStatus Obesity
## 3940 <NA> <NA>
## 3941 <NA> <NA>
## 3942 <NA> <NA>
## 3943 <NA> <NA>
## 3944 <NA> <NA>
## 3945 <NA> <NA>

```

```
summary(EmployeeBiometric.df)
```

```

## EmployeeID Gender AgeYears
## ID10005: 1 Female:2078 Min. :21.0
## ID10010: 1 Male :1867 1st Qu.:34.0
## ID10016: 1 Median :47.0
## ID10028: 1 Mean :47.7
## ID10041: 1 3rd Qu.:61.0
## ID10051: 1 Max. :79.0
## (Other):3939
## RaceEthnicity TotalCholesterolmgdL SBPmmHg
## Asian : 482 Min. : 78 Min. : 74
## Black : 818 1st Qu.:151 1st Qu.:108

```

```

##   Hispanic: 904    Median :175           Median :118
##   Other     : 117    Mean   :180           Mean   :122
##   White     :1624    3rd Qu.:205         3rd Qu.:132
##                               Max.   :380           Max.   :222
##                               NA's    :894           NA's    :209
##      DBPmmHg        BMIMetric          BMIStatus
##   Min.    : 34.0    Min.   :13.4       Underweight  : 168
##   1st Qu.: 60.0    1st Qu.:22.6      Normal Weight:1437
##   Median  : 70.0    Median  :26.4       Overweight   :1114
##   Mean    : 69.3    Mean   :27.7       Obese       :1206
##   3rd Qu.: 78.0    3rd Qu.:31.3      NA's        :  20
##   Max.    :122.0    Max.   :70.1
##   NA's    :217      NA's   :20
##      Obesity
##   Not Obese:2719
##   Obese    :1206
##   NA's     :  20
##
##
##
##

```

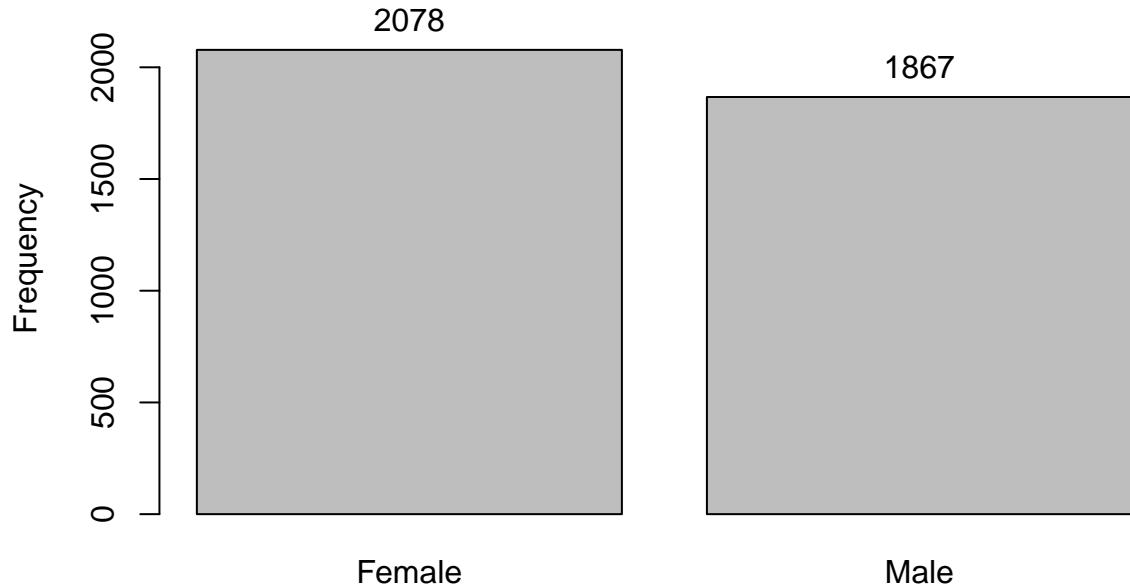
Conduct a Visual Data Check Using Graphics using epiDisplay

The epiDisplay::tab0.85() function provides syntax that generates an attractive barplot with frequency counts placed over each bar and a frequency distribution table printed to the screen. Also, for factor-type object variables with missing data, the epiDisplay::tab0.85() function incorporates the presentation of these missing data into the barplot and the frequency distribution table, providing a complete sense of the data

Create bar plots for factor variables : Gender, RaceEthnicity, BMIStatus, Obesity

```
epiDisplay::tab1(EmployeeBiometric.df$Gender)
```

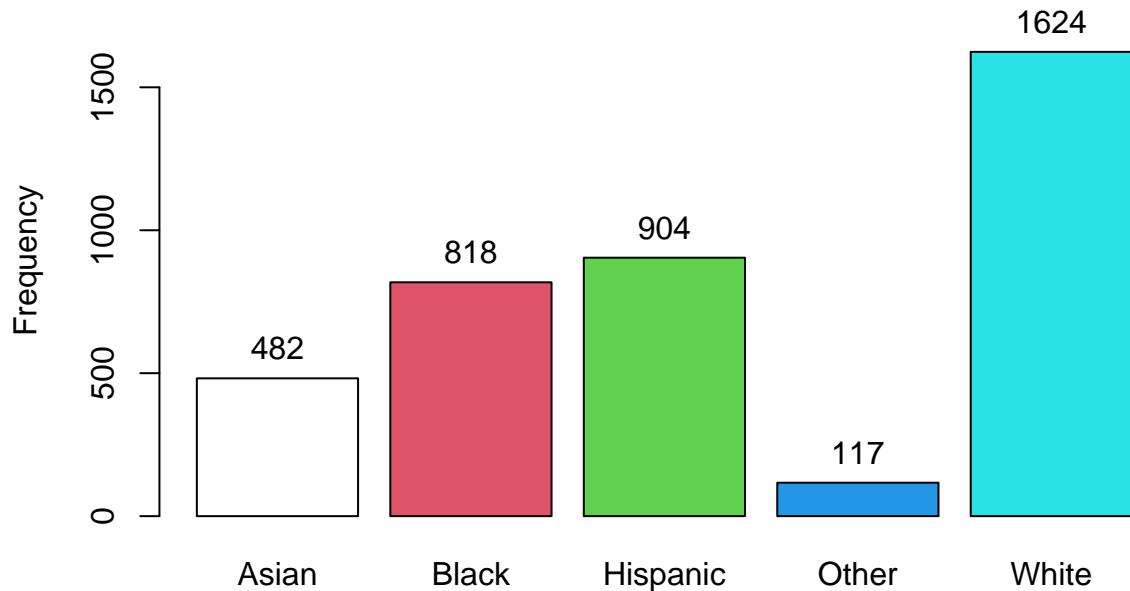
Distribution of EmployeeBiometric.df\$Gender



```
## EmployeeBiometric.df$Gender :  
##      Frequency Percent Cum. percent  
## Female      2078     52.7        52.7  
## Male        1867     47.3       100.0  
## Total       3945    100.0      100.0
```

```
epiDisplay::tab1(EmployeeBiometric.df$RaceEthnicity)
```

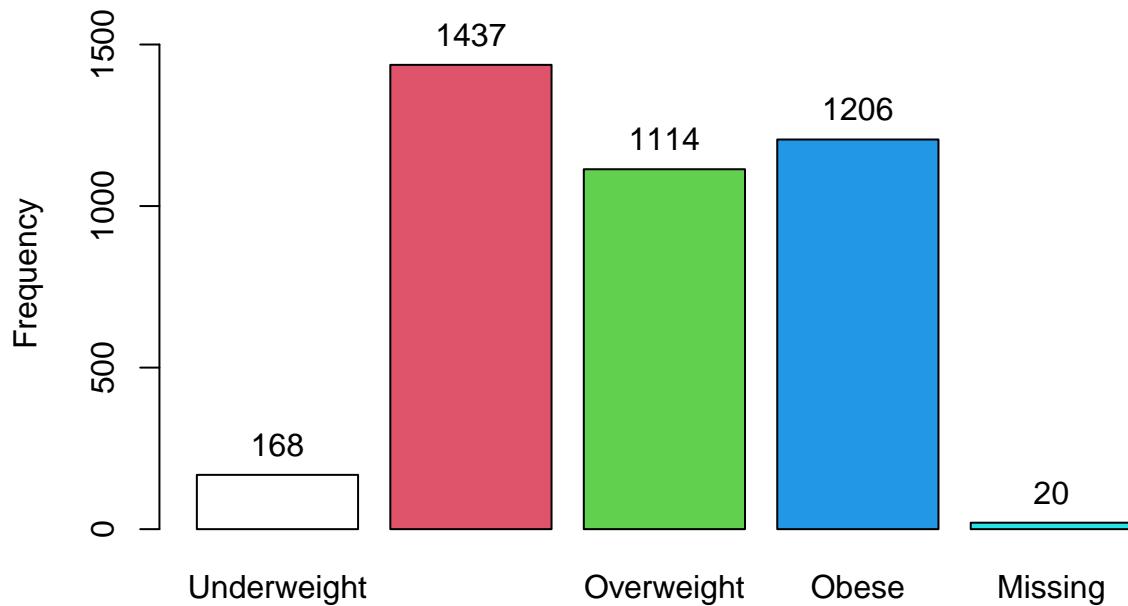
Distribution of EmployeeBiometric.df\$RaceEthnicity



```
## EmployeeBiometric.df$RaceEthnicity :  
##      Frequency Percent Cum. percent  
##  Asian        482    12.2       12.2  
##  Black        818    20.7       33.0  
##  Hispanic     904    22.9       55.9  
##  Other         117     3.0       58.8  
##  White       1624    41.2      100.0  
##  Total        3945   100.0      100.0
```

```
epiDisplay::tab1(EmployeeBiometric.df$BMISstatus)
```

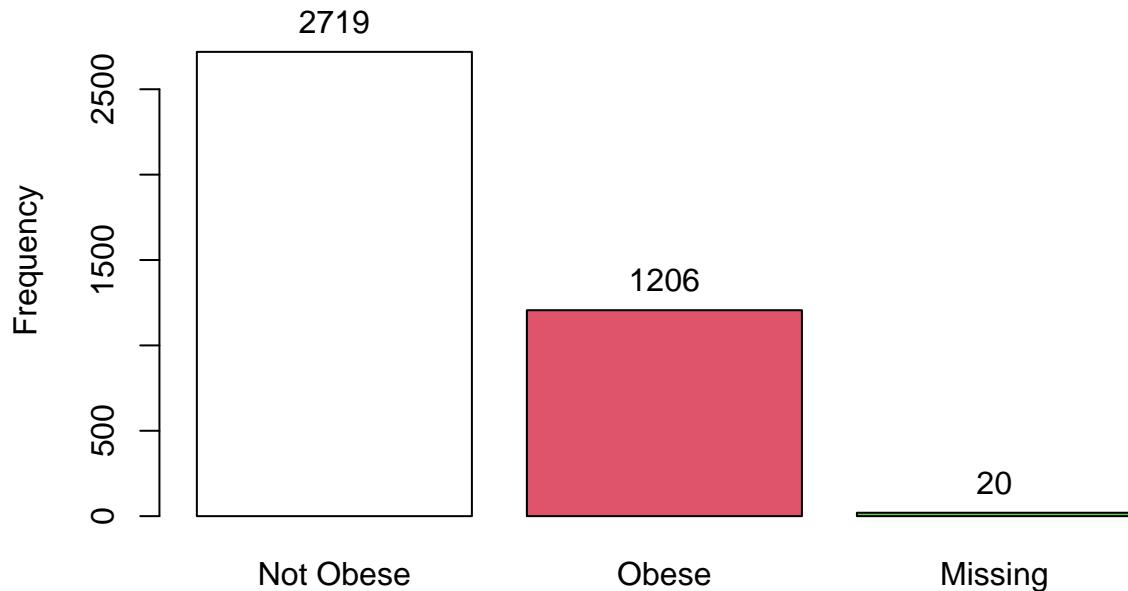
Distribution of EmployeeBiometric.df\$BMISatus



```
## EmployeeBiometric.df$BMISatus :  
##          Frequency  %(NA+)  %(NA-)  
## Underweight      168      4.3      4.3  
## Normal Weight    1437     36.4     36.6  
## Overweight       1114     28.2     28.4  
## Obese            1206     30.6     30.7  
## NA's              20      0.5      0.0  
## Total             3945    100.0    100.0
```

```
epiDisplay::tab1(EmployeeBiometric.df$Obesity)
```

Distribution of EmployeeBiometric.df\$Obesity



```
## EmployeeBiometric.df$Obesity :  
##      Frequency  % (NA+)  % (NA-)  
## Not Obese     2719     68.9     69.3  
## Obese        1206     30.6     30.7  
## NA's          20      0.5      0.0  
## Total        3945    100.0    100.0
```

Create bar plots for numeric variables : AgeYears, TotalCholesterolmgdL, SBPmmHg, DBPmmHg, and the calculated value for BMIMetric

Figures for the EmployeeBiometric.df object variables with numeric values (e.g., often called interval values or scale values) will be displayed in a 2 by 2 graphical grid. Each square in the grid uses a different R-based function to display data distribution:

- Histogram — Use the hist() function.
- Density Curve — The plot() function is wrapped around the density() function.
- Beanplot — Use the beanplot::beanplot() function, by loading the external package beanplot into the active R session.
- Quantile-Quantile Curve — Use the qqnorm() function.

Age Distribution

```
par(ask=TRUE)  
par(mfrow=c(2,2))  
  
hist(EmployeeBiometric.df$AgeYears, main="Age", col="red", breaks=100)
```

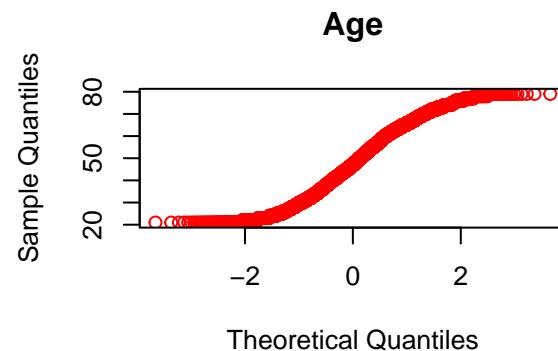
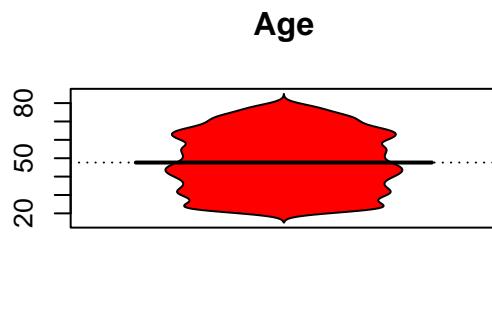
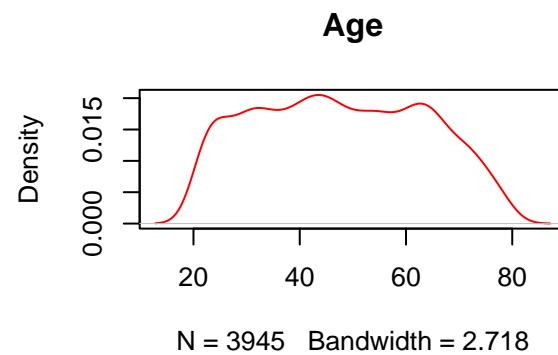
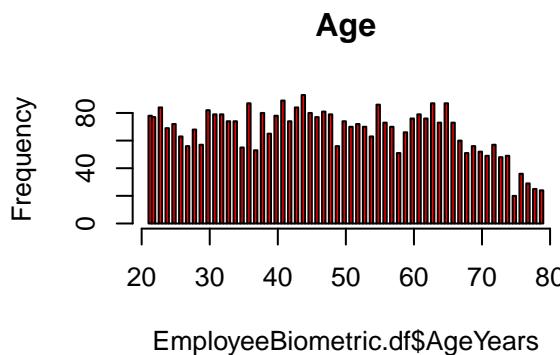
```

plot(density(EmployeeBiometric.df$AgeYears, na.rm=TRUE),
     main="Age", col="red")

beanplot::beanplot(EmployeeBiometric.df$AgeYears, main="Age", col="red", what=c(0.85,0.85,0.85,0), overall=TRUE)

qqnorm(EmployeeBiometric.df$AgeYears,main="Age",col="red")

```



Variable TotalCholesterolmgL

```

par(ask=TRUE)
par(mfrow=c(2,2))

hist(EmployeeBiometric.df$TotalCholesterolmgdL, main="Total Cholesterol mg/dL", col="blue", breaks=100)

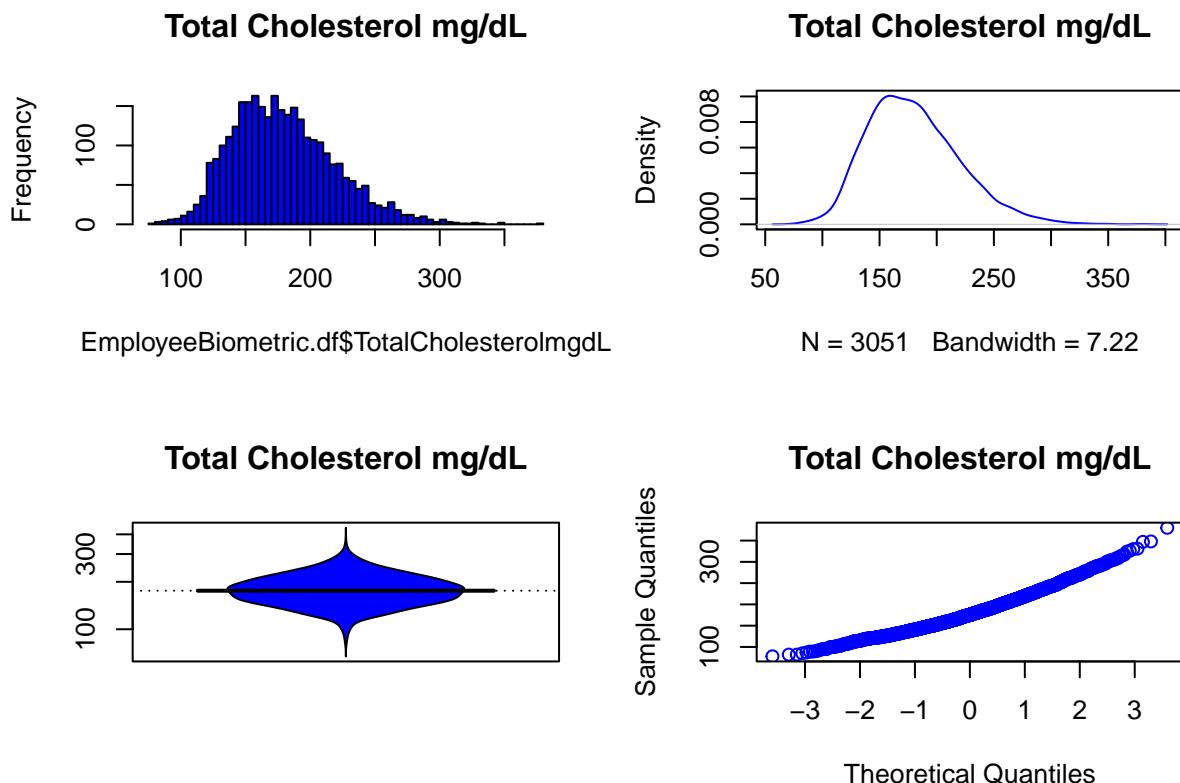
plot(density(EmployeeBiometric.df$TotalCholesterolmgdL, na.rm=TRUE),
     main="Total Cholesterol mg/dL", col="blue")

beanplot::beanplot(EmployeeBiometric.df$TotalCholesterolmgdL, main="Total Cholesterol mg/dL", col="blue")

## log="y" selected

qqnorm(EmployeeBiometric.df$TotalCholesterolmgdL, main="Total Cholesterol mg/dL", col="blue")

```



Variable SBPmmHg

```
par(ask=TRUE)
par(mfrow=c(2,2))

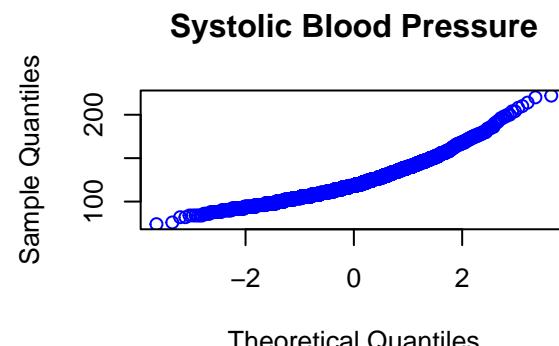
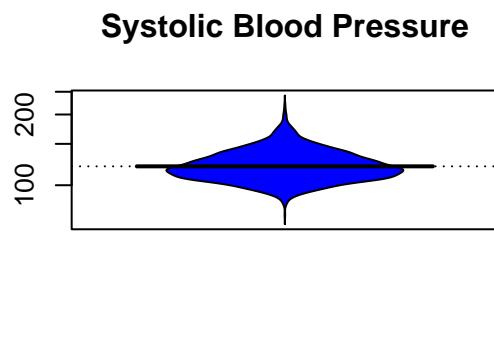
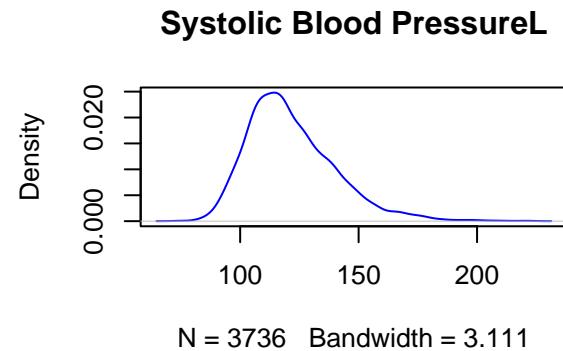
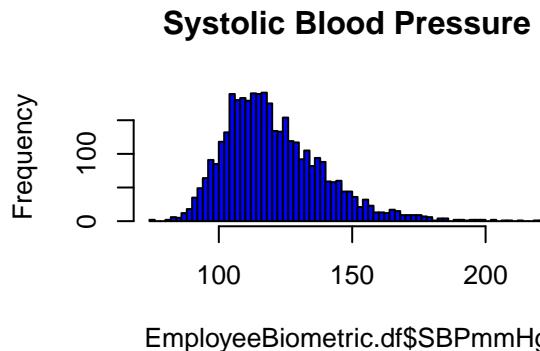
hist(EmployeeBiometric.df$SBPmmHg,
      main="Systolic Blood Pressure",
      col="blue",
      breaks=100)

plot(density(EmployeeBiometric.df$SBPmmHg,
             na.rm=TRUE),
     main="Systolic Blood PressureL",
     col="blue")

beanplot:::beanplot(EmployeeBiometric.df$SBPmmHg,
                    main="Systolic Blood Pressure",
                    col="blue",
                    what=c(1,1,1,0),
                    overallline="mean",
                    boxwex=0.75,
                    horizontal=FALSE)

## log="y" selected
```

```
qqnorm(EmployeeBiometric.df$SBPmmHg,
       main="Systolic Blood Pressure",
       col="blue")
```



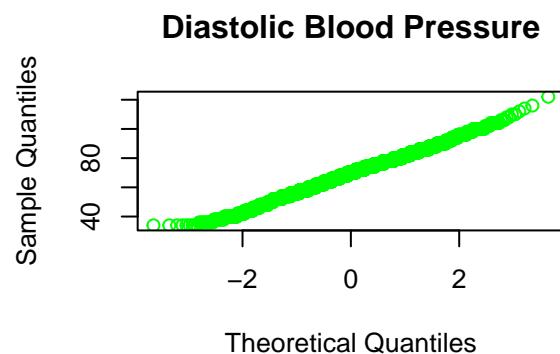
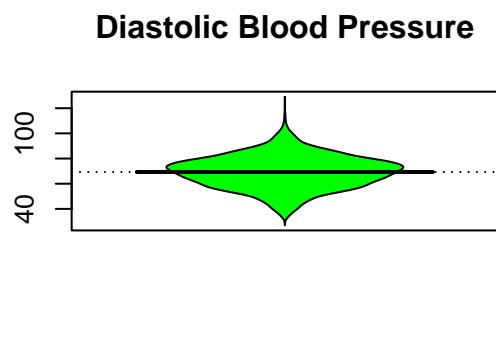
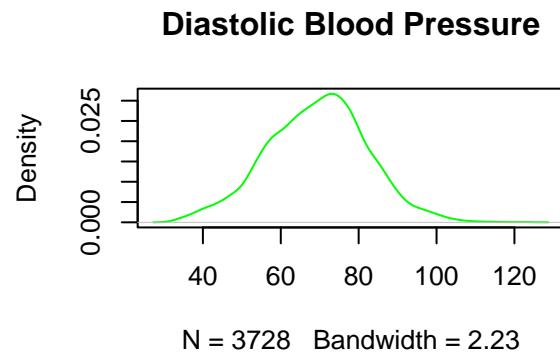
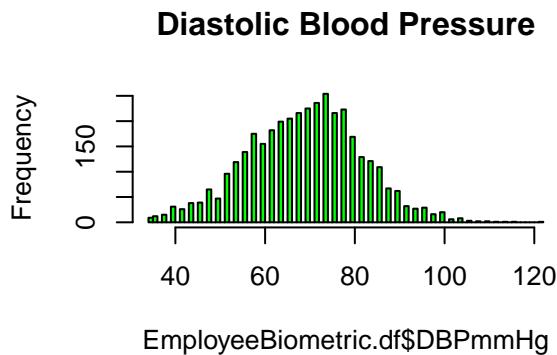
Variable DBPmmHg

```
par(ask=TRUE)
par(mfrow=c(2,2))

hist(EmployeeBiometric.df$DBPmmHg, main="Diastolic Blood Pressure", col="green", breaks=100)

plot(density(EmployeeBiometric.df$DBPmmHg, na.rm=TRUE),
     main="Diastolic Blood Pressure", col="green")

beanplot:::beanplot(EmployeeBiometric.df$DBPmmHg, main="Diastolic Blood Pressure", col="green",
                     what=c(0.85,0.85,0.85,0), overallline="mean", boxwex=0.75, horizontal=FALSE)
qqnorm(EmployeeBiometric.df$DBPmmHg, main="Diastolic Blood Pressure", col="green")
```



```
##Variable BMIMetric

par(ask=TRUE)
par(mfrow=c(2,2))

hist(EmployeeBiometric.df$BMIMetric, main="Body Mass Index: BMI Metric = kg/m^2", col="purple", breaks=20)

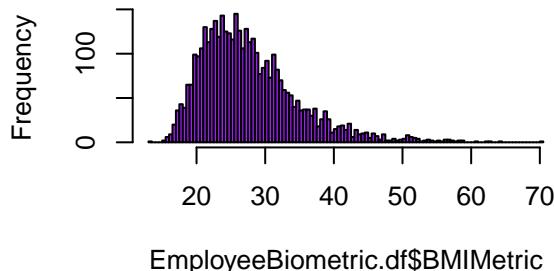
plot(density(EmployeeBiometric.df$BMIMetric, na.rm=TRUE),
     main="Body Mass Index: BMI Metric = kg/m^2", col="purple")

beanplot::beanplot(EmployeeBiometric.df$BMIMetric, main="Body Mass Index: BMI Metric = kg/m^2", col="purple")

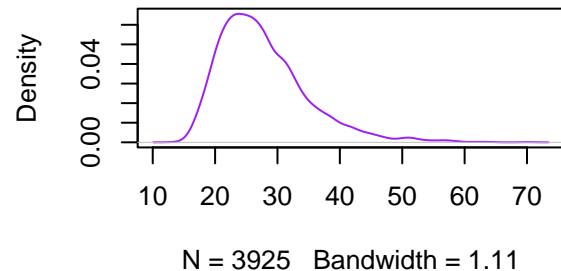
## log="y" selected

qqnorm(EmployeeBiometric.df$BMIMetric, main="Body Mass Index: BMI Metric = kg/m^2", col="purple")
```

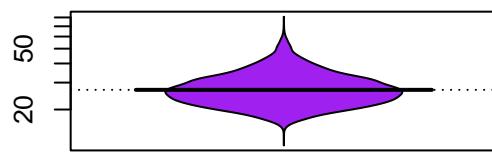
Body Mass Index: BMI Metric = kg/m²



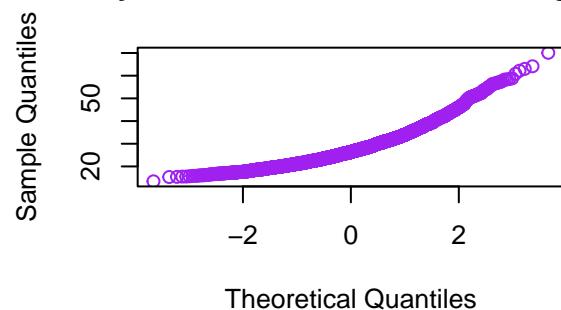
Body Mass Index: BMI Metric = kg/m²



Body Mass Index: BMI Metric = kg/m²



Body Mass Index: BMI Metric = kg/m²



```
## Summary for variable factors
```

```
summary(EmployeeBiometric.df[,c(2,4,9,10)])
```

```
##      Gender      RaceEthnicity      BMIStatus
## Female:2078   Asian    : 482   Underweight  : 168
## Male  :1867    Black    : 818   Normal Weight:1437
##           Hispanic: 904   Overweight   :1114
##           Other    : 117   Obese       :1206
##           White    :1624   NA's        :  20
##           Obesity
## Not Obese:2719
## Obese   :1206
## NA's    :  20
##
```

Summary for numeric variable

```
summary(EmployeeBiometric.df[,c(3,5,7,8)])
```

```
##      AgeYears      TotalCholesterolmgdL      DBPmmHg
## Min.   :21.0      Min.   : 78          Min.   : 34.0
## 1st Qu.:34.0     1st Qu.:151         1st Qu.: 60.0
```

```

## Median :47.0    Median :175          Median : 70.0
## Mean   :47.7    Mean   :180          Mean   : 69.3
## 3rd Qu.:61.0    3rd Qu.:205        3rd Qu.: 78.0
## Max.   :79.0    Max.   :380        Max.   :122.0
## NA's   :894     NA's   :217        NA's   :217

##      BMIMetric
## Min.   :13.4
## 1st Qu.:22.6
## Median :26.4
## Mean   :27.7
## 3rd Qu.:31.3
## Max.   :70.1
## NA's   :20

```

Provide percentage of factor variable

```
epiDisplay::tab1(EmployeeBiometric.df$Gender, graph=FALSE)
```

```

## EmployeeBiometric.df$Gender :
##           Frequency Percent Cum. percent
## Female       2078     52.7      52.7
## Male         1867     47.3      100.0
## Total        3945     100.0     100.0

```

```
epiDisplay::tab1(EmployeeBiometric.df$RaceEthnicity, graph = FALSE)
```

```

## EmployeeBiometric.df$RaceEthnicity :
##           Frequency Percent Cum. percent
## Asian        482      12.2      12.2
## Black        818      20.7      33.0
## Hispanic     904      22.9      55.9
## Other         117      3.0      58.8
## White        1624     41.2      100.0
## Total        3945     100.0     100.0

```

```
epiDisplay::tab1(EmployeeBiometric.df$BMIStatus, graph=FALSE)
```

```

## EmployeeBiometric.df$BMIStatus :
##           Frequency  %(NA+)  %(NA-)
## Underweight     168      4.3      4.3
## Normal Weight  1437     36.4     36.6
## Overweight     1114     28.2     28.4
## Obese          1206     30.6     30.7
## NA's            20       0.5      0.0
## Total          3945     100.0    100.0

```

```
epiDisplay::tab1(EmployeeBiometric.df$Obesity, graph = FALSE)
```

```

## EmployeeBiometric.df$Obesity :
##           Frequency  %(NA+)  %(NA-)
## Not Obese      2719     68.9     69.3
## Obese          1206     30.6     30.7
## NA's            20       0.5      0.0
## Total          3945     100.0    100.0

```

Measures of Central Tendency for Numeric-Type Object Variables

To address this concern and the need for attention to Standard Deviation as a vital descriptive statistic, consider use of the RcmdrMisc::numSummary() function, which by default provides Mean, Standard Deviation, IQR (Interquartile Range, or Q3 minus Q1) Quartiles (0% (Minimum), 25%, 50% (Median), 75%, 100% (Maximum)) N, and NAs (the number of missing values, if any).

```
RcmdrMisc::numSummary(EmployeeBiometric.df$AgeYears)
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n
##  47.6563 15.8221 27 21 34 47 61 79 3945
```

```
RcmdrMisc::numSummary(EmployeeBiometric.df$TotalCholesterolmgdL)
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
## 179.835 39.9184 54 78 151 175 205 380 3051 894
```

```
RcmdrMisc::numSummary(EmployeeBiometric.df$SBPmmHg)
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
## 121.646 18.6069 24 74 108 118 132 222 3736 209
```

```
RcmdrMisc::numSummary(EmployeeBiometric.df$DBPmmHg)
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
## 69.3192 12.836 18 34 60 70 78 122 3728 217
```

```
RcmdrMisc::numSummary(EmployeeBiometric.df$BMIMetric)
```

```
##      mean      sd  IQR    0% 25% 50% 75% 100%     n   NA
## 27.7057 7.12103 8.65 13.45 22.61 26.38 31.26 70.08 3925 20
```

Summary, group by gender

```
RcmdrMisc::numSummary(EmployeeBiometric.df[,c("AgeYears","TotalCholesterolmgdL","SBPmmHg","DBPmmHg","BMIMetric")])
```



```
##
```

```
## Variable: AgeYears
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
## Female 47.6925 15.7969 27 21 34 47 61 79 2078 0
## Male   47.6160 15.8542 27 21 34 47 61 79 1867 0
##
```

```
##
```

```
## Variable: TotalCholesterolmgdL
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
## Female 179.768 40.5027 56.0 78 150 176 206.0 380 1628 450
## Male   179.911 39.2533 52.5 85 151 175 203.5 348 1423 444
##
```

```
##
```

```
## Variable: SBPmmHg
```

```
##      mean      sd  IQR  0% 25% 50% 75% 100%     n   NA
```

```

## Female 119.390 18.0147 20 82 108 116 128 222 1976 102
## Male    124.178 18.9374 26 74 110 122 136 220 1760 107
##
## Variable: DBPmmHg
##          mean      sd IQR 0% 25% 50% 75% 100%     n NA
## Female 68.2222 12.604 16 34 60 68 76 122 1971 107
## Male   70.5498 12.985 18 34 62 72 80 114 1757 110
##
## Variable: BMIMetric
##          mean      sd IQR 0% 25% 50% 75% 100%
## Female 27.6231 7.17061 8.745 15.41 22.435 26.18 31.18 64.15
## Male   27.7976 7.06625 8.540 13.45 22.750 26.56 31.29 70.08
##          n NA
## Female 2067 11
## Male   1858  9

```

Summary, group by RaceEthnicity

```
RcmdrMisc::numSummary(EmployeeBiometric.df[,c("AgeYears","TotalCholesterolmgdL","SBPmmHg","DBPmmHg","BM...]
```

```

##
## Variable: AgeYears
##          mean      sd IQR 0% 25% 50% 75% 100%     n NA
## Asian   47.0353 14.7131 24 21 35 46 59 79 482 0
## Black   48.2433 16.1025 28 21 34 49 62 79 818 0
## Hispanic 46.8584 15.3455 26 21 35 46 61 79 904 0
## Other   38.6068 13.3177 20 21 28 36 48 78 117 0
## White   48.6410 16.2003 27 21 35 48 62 79 1624 0
##
## Variable: TotalCholesterolmgdL
##          mean      sd IQR 0% 25% 50% 75% 100%
## Asian   178.809 40.5190 52.00 95 150 175.0 202.00 324
## Black   181.892 40.7727 54.75 78 152 177.0 206.75 380
## Hispanic 178.557 38.1856 52.50 82 150 174.0 202.50 309
## Other   176.864 38.7806 59.00 109 147 171.5 206.00 292
## White   179.979 40.3126 54.00 82 151 176.0 205.00 347
##          n NA
## Asian   362 120
## Black   646 172
## Hispanic 687 217
## Other   88 29
## White   1268 356
##
## Variable: SBPmmHg
##          mean      sd IQR 0% 25% 50% 75% 100%     n NA
## Asian   119.062 18.0030 24 82 106 116 130 222 454 28
## Black   125.683 18.4081 24 90 112 122 136 200 769 49
## Hispanic 122.948 18.4694 22 88 110 120 132 214 863 41
## Other   122.000 18.0343 25 90 108 118 133 198 111  6
## White   119.635 18.6154 24 74 106 116 130 220 1539 85
##
## Variable: DBPmmHg
##          mean      sd IQR 0% 25% 50% 75% 100%     n NA

```

```

## Asian      67.6380 12.6124 18 38 58 68 76 104 453 29
## Black     72.5312 12.9211 16 34 64 74 80 112 768 50
## Hispanic  70.5128 13.0697 18 34 62 72 80 122 862 42
## Other     68.8649 12.2121 16 34 62 70 78 100 111 6
## White     67.5698 12.3939 18 34 58 68 76 116 1534 90
##
## Variable: BMIMetric
##          mean      sd   IQR    0%   25%   50%
## Asian     27.9289 7.45770 10.0150 13.45 22.2075 26.325
## Black     27.4193 6.73504 8.0600 15.55 22.8300 26.250
## Hispanic  27.6545 6.99525 8.6675 15.53 22.6600 26.375
## Other     28.5462 6.87024 7.7500 17.57 23.8900 26.690
## White     27.7517 7.29454 8.6200 15.34 22.5100 26.490
##          75% 100% n NA
## Asian     32.2225 58.34 476 6
## Black     30.8900 64.15 813 5
## Hispanic  31.3275 62.19 902 2
## Other     31.6400 51.24 117 0
## White     31.1300 70.08 1617 7

```

Summary, group by BMIstatus

```
RcmdrMisc::numSummary(EmployeeBiometric.df[,c("AgeYears", "TotalCholesterolmgdL", "SBPmmHg", "DBPmmHg", "BM...")])
```

```

##
## Variable: AgeYears
##          mean      sd   IQR    0%   25%   50%   75% 100%
## Underweight 45.9583 15.3289 27.25 21 32 46 59.25 77
## Normal Weight 47.7537 15.8668 27.00 21 34 47 61.00 79
## Overweight   47.9282 15.7987 26.75 21 35 48 61.75 79
## Obese        47.4677 15.8322 26.75 21 34 47 60.75 79
##          n NA
## Underweight 168 0
## Normal Weight 1437 0
## Overweight   1114 0
## Obese        1206 0
##
## Variable: TotalCholesterolmgdL
##          mean      sd   IQR    0%   25%   50%   75% 100%
## Underweight 179.354 39.8540 48 87 154.5 173.0 202.5 326
## Normal Weight 179.461 41.0036 54 78 150.0 174.0 204.0 380
## Overweight   181.183 40.0782 56 82 151.0 177.0 207.0 347
## Obese        179.254 38.6444 52 82 151.0 176.5 203.0 324
##          n NA
## Underweight 127 41
## Normal Weight 1088 349
## Overweight   871 243
## Obese        950 256
##
## Variable: SBPmmHg
##          mean      sd   IQR    0%   25%   50%   75% 100% n
## Underweight 113.045 18.6707 16 82 102 110 118 208 157
## Normal Weight 116.622 18.1826 20 76 104 114 124 220 1363

```

```

## Overweight    123.802 18.5569  24 74 110 120 134  222 1058
## Obese        126.915 17.1579  24 90 114 126 138  204 1145
##
## NA
## Underweight   11
## Normal Weight 74
## Overweight     56
## Obese          61
##
## Variable: DBPmmHg
##             mean      sd IQR 0% 25% 50% 75% 100%   n
## Underweight   63.3758 12.8100 18 36 54 64 72 116 157
## Normal Weight 65.9750 12.3119 16 34 58 66 74 114 1358
## Overweight    70.6199 12.5282 16 34 62 72 78 122 1055
## Obese         72.9817 12.4347 16 34 64 74 80 108 1145
##
## NA
## Underweight   11
## Normal Weight 79
## Overweight     59
## Obese          61
##
## Variable: BMIMetric
##             mean      sd IQR 0% 25% 50%
## Underweight   17.4923 0.824308 1.0325 13.45 17.0900 17.58
## Normal Weight 22.1205 1.747179 2.9300 18.60 20.7200 22.28
## Overweight    27.3041 1.395345 2.3650 25.00 26.0725 27.24
## Obese         36.1543 6.022411 6.9200 30.01 31.7400 34.19
##
##             75% 100%   n NA
## Underweight   18.1225 18.59 168 0
## Normal Weight 23.6500 25.00 1437 0
## Overweight    28.4375 30.00 1114 0
## Obese         38.6600 70.08 1206 0

```

Summary, group by Obesity

```
RcmdrMisc::numSummary(EmployeeBiometric.df[,c("AgeYears", "TotalCholesterolmgdL", "SBPmmHg", "DBPmmHg", "BM
```

```

##
## Variable: AgeYears
##             mean      sd IQR 0% 25% 50% 75% 100%   n
## Not Obese   47.7142 15.8071 27.00 21 34 47 61.00    79 2719
## Obese       47.4677 15.8322 26.75 21 34 47 60.75    79 1206
##
## NA
## Not Obese   0
## Obese       0
##
## Variable: TotalCholesterolmgdL
##             mean      sd IQR 0% 25% 50% 75% 100%   n
## Not Obese   180.174 40.5399 54 78 151 175.0 205 380 2086
## Obese       179.254 38.6444 52 82 151 176.5 203 324 950
##
## NA
## Not Obese   633
## Obese       256
##
```

```

## Variable: SBPmmHg
##           mean      sd IQR 0% 25% 50% 75% 100%     n  NA
## Not Obese 119.351 18.7503 22 74 106 116 128 222 2578 141
## Obese     126.915 17.1579 24 90 114 126 138 204 1145  61
##
## Variable: DBPmmHg
##           mean      sd IQR 0% 25% 50% 75% 100%     n  NA
## Not Obese 67.7230 12.6746 18 34 58 68 76 122 2570 149
## Obese     72.9817 12.4347 16 34 64 74 80 108 1145  61
##
## Variable: BMIMetric
##           mean      sd IQR 0% 25% 50% 75%
## Not Obese 23.9583 3.37783 5.38 13.45 21.365 24.01 26.745
## Obese     36.1543 6.02241 6.92 30.01 31.740 34.19 38.660
##           100%     n  NA
## Not Obese 30.00 2719  0
## Obese     70.08 1206  0

```

1. Distribution of age based on categorical variables :

```

par(ask=TRUE)
par(mfrow=c(2,2))

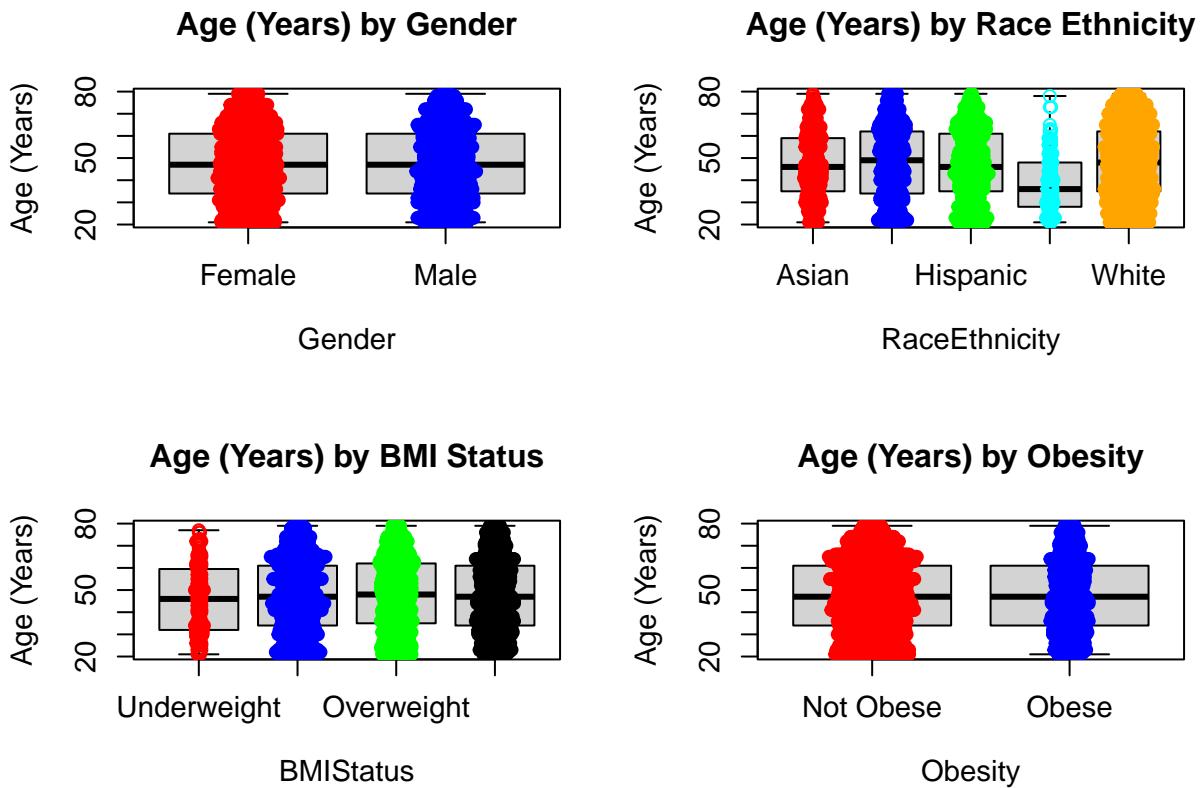
boxplot(AgeYears ~ Gender, data=EmployeeBiometric.df,
        main="Age (Years) by Gender", ylab="Age (Years)",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(AgeYears~Gender, data=EmployeeBiometric.df,
          col=c("red","blue"),add=TRUE,spacing=0.10)

boxplot(AgeYears~RaceEthnicity, data=EmployeeBiometric.df,
        main="Age (Years) by Race Ethnicity", ylab="Age (Years)",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(AgeYears~RaceEthnicity, data=EmployeeBiometric.df,
          col=c("red","blue","green","cyan","orange"),add=TRUE,spacing=0.10)

boxplot(AgeYears~BMISatus, data=EmployeeBiometric.df,
        main="Age (Years) by BMI Status", ylab="Age (Years)",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(AgeYears~BMISatus, data=EmployeeBiometric.df,
          col=c("red","blue","green","black"),add=TRUE,spacing=0.10)

boxplot(AgeYears~Obesity, data=EmployeeBiometric.df,
        main="Age (Years) by Obesity", ylab="Age (Years)",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(AgeYears~Obesity, data= EmployeeBiometric.df,
          col=c("red","blue"), add=TRUE, spacing=0.1)

```



2. Distribution of Total Cholesterol based on categorical variables :

```

par(ask=TRUE)
par(mfrow=c(2,2))

boxplot(TotalCholesterolmgdL ~ Gender, data=EmployeeBiometric.df,
        main="Total Cholesterol by Gender", ylab="Total Cholesterol",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(TotalCholesterolmgdL~Gender, data=EmployeeBiometric.df,
          col=c("red","blue"),add=TRUE,spacing=0.10)

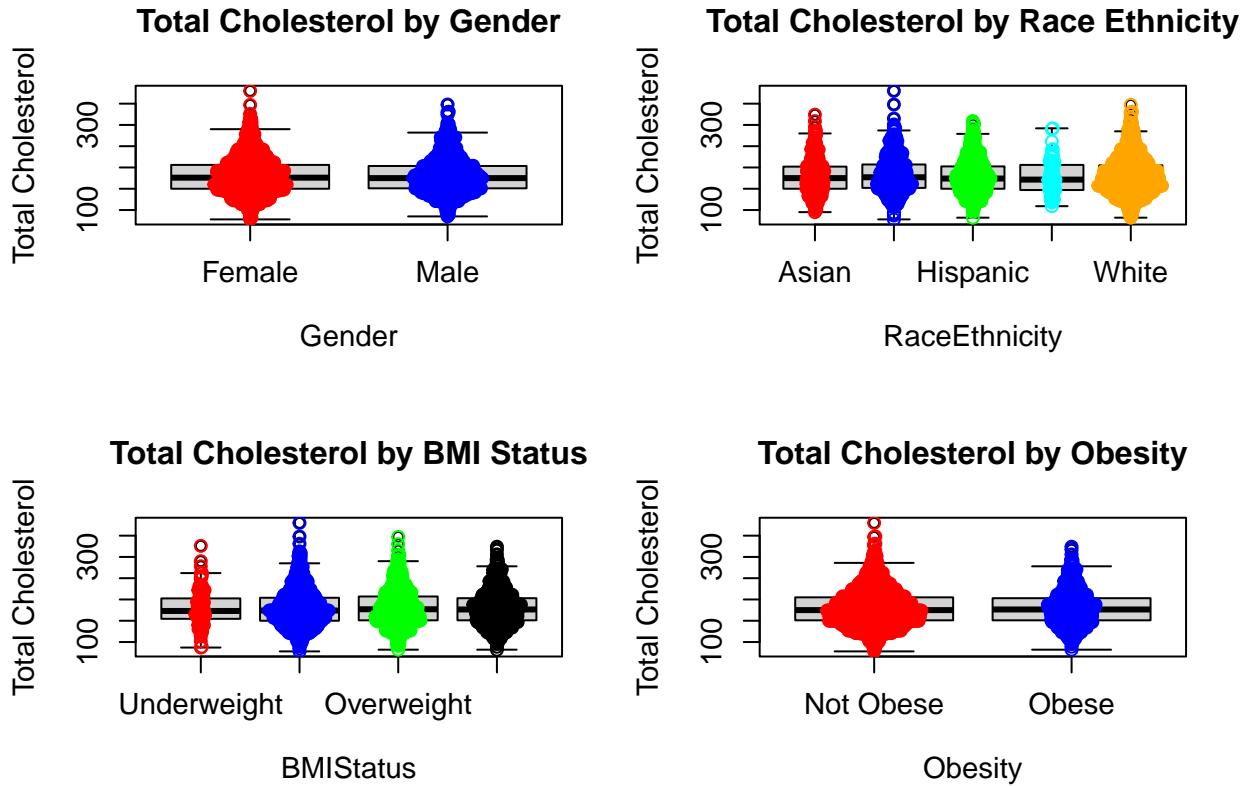
boxplot(TotalCholesterolmgdL~RaceEthnicity, data=EmployeeBiometric.df,
        main="Total Cholesterol by Race Ethnicity", ylab="Total Cholesterol",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(TotalCholesterolmgdL~RaceEthnicity, data=EmployeeBiometric.df,
          col=c("red","blue","green","cyan","orange"),add=TRUE,spacing=0.10)

boxplot(TotalCholesterolmgdL~BMISstatus, data=EmployeeBiometric.df,
        main="Total Cholesterol by BMI Status", ylab="Total Cholesterol",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(TotalCholesterolmgdL~BMISstatus, data=EmployeeBiometric.df,
          col=c("red","blue","green","black"),add=TRUE,spacing=0.10)

boxplot(TotalCholesterolmgdL~Obesity, data=EmployeeBiometric.df,
        main="Total Cholesterol by Obesity", ylab="Total Cholesterol",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(TotalCholesterolmgdL~Obesity, data= EmployeeBiometric.df,
          col=c("red","blue","green","black"),add=TRUE,spacing=0.10)

```

```
col=c("red","blue"), add=TRUE, spacing=0.1)
```



3. Distribution of SBP based on categorical variables :

```
par(ask=TRUE)
par(mfrow=c(2,2))
boxplot(SBPmmHg ~ Gender, data=EmployeeBiometric.df,
        main="Systolic Blood Pressure by Gender", ylab="Systolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(SBPmmHg~Gender, data=EmployeeBiometric.df,
          col=c("red","blue"),add=TRUE,spacing=0.10)

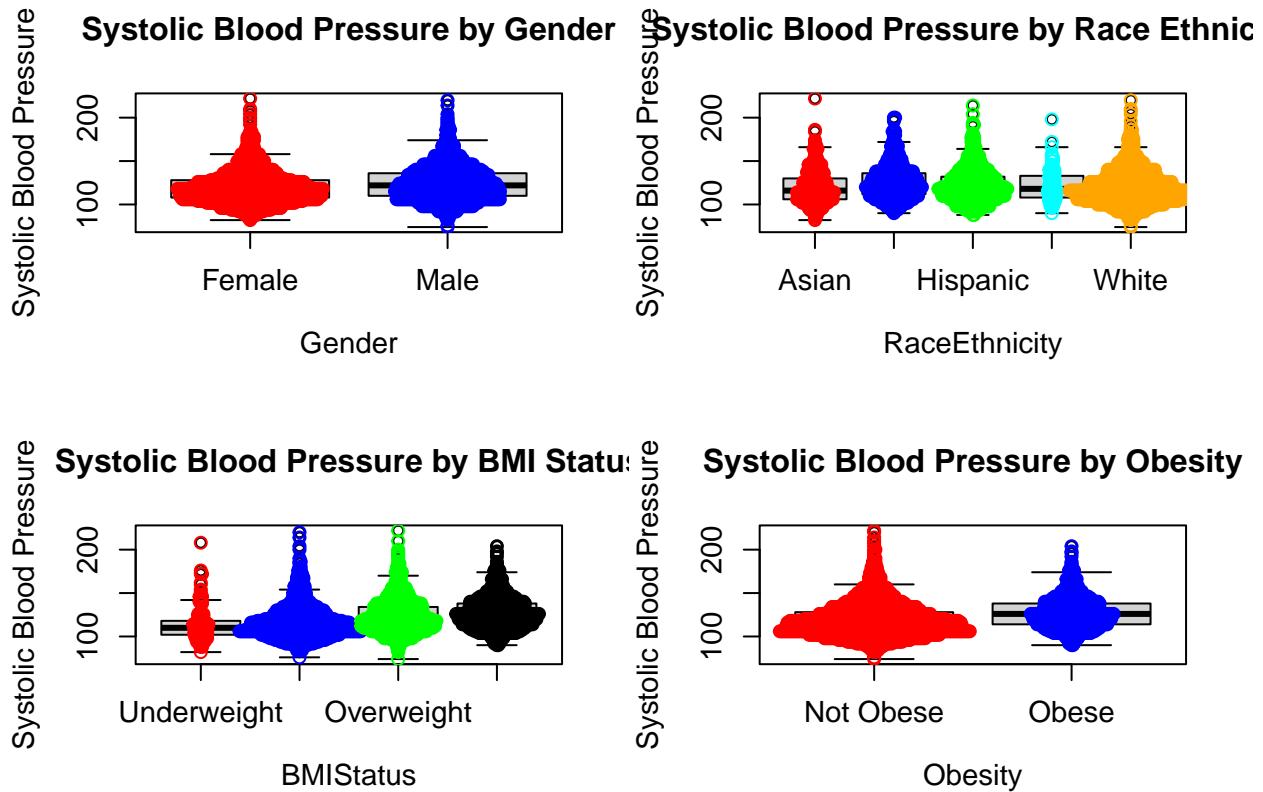
boxplot(SBPmmHg~RaceEthnicity, data=EmployeeBiometric.df,
        main="Systolic Blood Pressure by Race Ethnicity", ylab="Systolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(SBPmmHg~RaceEthnicity, data=EmployeeBiometric.df,
          col=c("red","blue","green","cyan","orange"),add=TRUE,spacing=0.10)

boxplot(SBPmmHg~BMIStatus, data=EmployeeBiometric.df,
        main="Systolic Blood Pressure by BMI Status", ylab="Systolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(SBPmmHg~BMIStatus, data=EmployeeBiometric.df,
          col=c("red","blue","green","black"),add=TRUE,spacing=0.10)
```

```

boxplot(SBPmmHg~Obesity, data=EmployeeBiometric.df,
        main="Systolic Blood Pressure by Obesity", ylab="Systolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(SBPmmHg~Obesity, data= EmployeeBiometric.df,
          col=c("red","blue"), add=TRUE, spacing=0.1)

```



4. Distribution of DBP based on categorical variables :

```

par(ask=TRUE)
par(mfrow=c(2,2))
boxplot(DBPmmHg ~ Gender, data=EmployeeBiometric.df,
        main="Diastolic Blood Pressure by Gender", ylab="Diastolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(DBPmmHg~Gender, data=EmployeeBiometric.df,
          col=c("red","blue"),add=TRUE,spacing=0.10)

boxplot(DBPmmHg~RaceEthnicity, data=EmployeeBiometric.df,
        main="Diastolic Blood Pressure by Race Ethnicity", ylab="Diastolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(DBPmmHg~RaceEthnicity, data=EmployeeBiometric.df,
          col=c("red","blue","green","cyan","orange"),add=TRUE,spacing=0.10)

boxplot(DBPmmHg~BMISatus, data=EmployeeBiometric.df,
        main="Diastolic Blood Pressure by BMI Status", ylab="Diastolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)

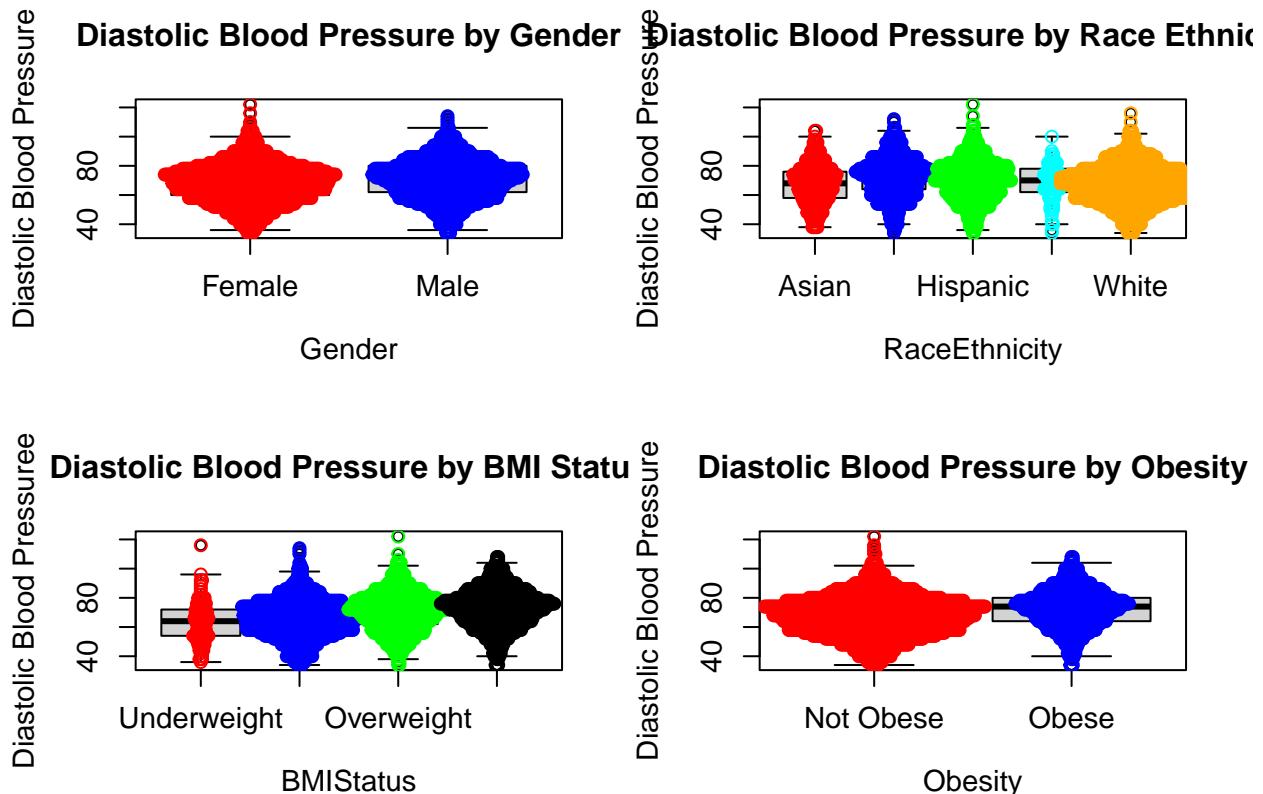
```

```

cex.axis=1.15, cex.lab=1.15)
beeswarm(DBPmmHg~BMIStatus, data=EmployeeBiometric.df,
          col=c("red","blue","green","black"),add=TRUE,spacing=0.10)

boxplot(DBPmmHg~Obesity, data=EmployeeBiometric.df,
        main="Diastolic Blood Pressure by Obesity", ylab="Diastolic Blood Pressure",
        cex.axis=1.15, cex.lab=1.15)
beeswarm(DBPmmHg~Obesity, data= EmployeeBiometric.df,
          col=c("red","blue"), add=TRUE, spacing=0.1)

```



5. Distribution of BMIMetric by categorical variables

```

par(ask=TRUE)
par(mfrow=c(2,2))

boxplot(BMIMetric~Gender, data=EmployeeBiometric.df,
        main = "BMIMetric by Gender",
        ylab = "BMI Metric", cex.axis=1.15, cex.lab=1.15)
beeswarm(BMIMetric~Gender, data=EmployeeBiometric.df,
          col = c("red","blue"), add = TRUE, spacing=0.10)

boxplot(BMIMetric~RaceEthnicity, data=EmployeeBiometric.df,
        main = "BMIMetric by RaceEthnicity",
        ylab = "BMI Metric", cex.axis=1.15, cex.lab=1.15)
beeswarm(BMIMetric~RaceEthnicity, data=EmployeeBiometric.df,
          col = c("red","blue","green","yellow","cyan"), add = TRUE, spacing=0.10)

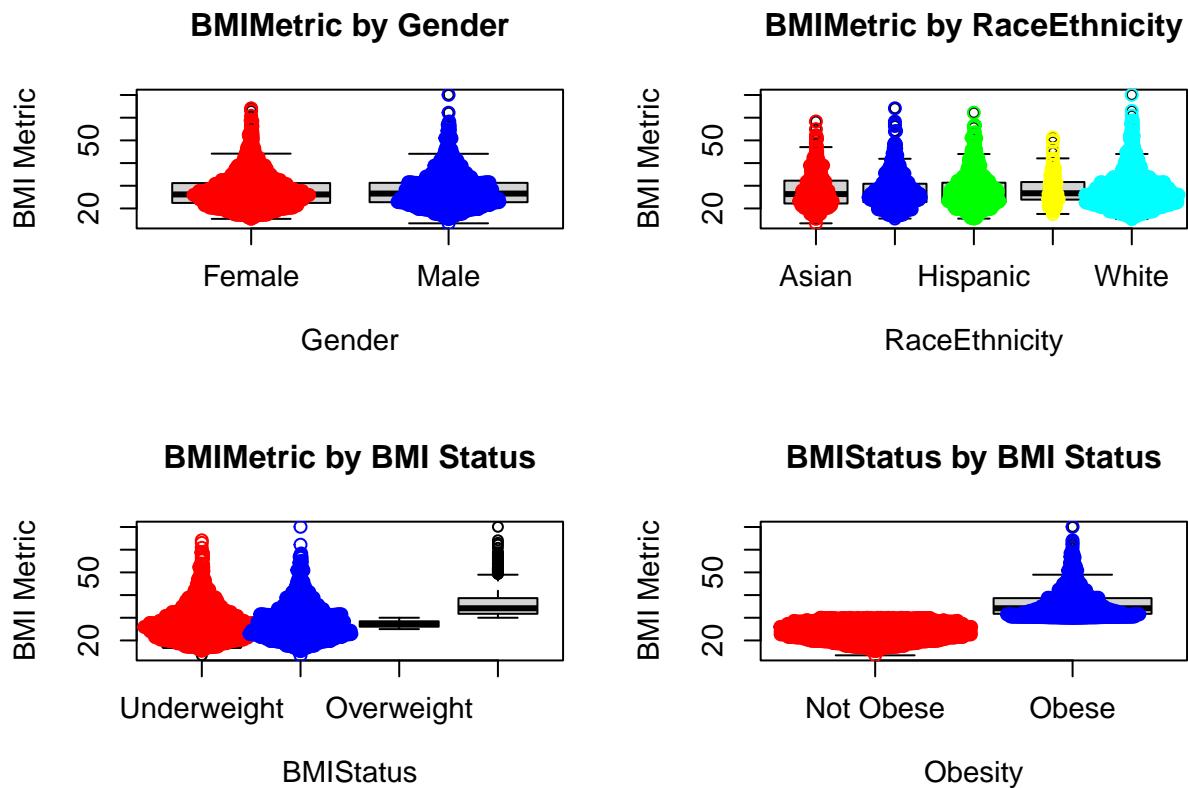
```

```

boxplot(BMIMetric~BMIStatus, data=EmployeeBiometric.df,
        main = "BMIMetric by BMI Status",
        ylab = "BMI Metric", cex.axis=1.15, cex.lab=1.15)
beeswarm(BMIMetric~Gender, data=EmployeeBiometric.df,
          col = c("red","blue","green","yellow"), add = TRUE, spacing=0.10)

boxplot(BMIMetric~Obesity, data=EmployeeBiometric.df,
        main = "BMIStatus by BMI Status",
        ylab = "BMI Metric", cex.axis=1.15, cex.lab=1.15)
beeswarm(BMIMetric~Obesity, data=EmployeeBiometric.df,
          col = c("red","blue"), add = TRUE, spacing=0.10)

```



Normality test

```

shapiro.test((EmployeeBiometric.df$AgeYears))

##
##  Shapiro-Wilk normality test
##
## data: (EmployeeBiometric.df$AgeYears)
## W = 0.9649, p-value <0.0000000000000002

```

```
shapiro.test(EmployeeBiometric.df$TotalCholesterolmgdL)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: EmployeeBiometric.df$TotalCholesterolmgdL  
## W = 0.9783, p-value <0.0000000000000002
```

```
shapiro.test((EmployeeBiometric.df$SBPmmHg))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: (EmployeeBiometric.df$SBPmmHg)  
## W = 0.95, p-value <0.0000000000000002
```

```
shapiro.test(EmployeeBiometric.df$DBPmmHg)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: EmployeeBiometric.df$DBPmmHg  
## W = 0.9962, p-value = 0.0000000321
```

```
shapiro.test(EmployeeBiometric.df$BMIMetric)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: EmployeeBiometric.df$BMIMetric  
## W = 0.9235, p-value <0.0000000000000002
```

Result of Shapiro test distribution : 1. Age = not normal 2. Total cholesterol = not normal 3. SBP = not normal 4. DBP = not normal 5. BMIMetrics = not normal

Distribution analysis using Q-Q Plot

```
par(aspect=TRUE)  
par(mfrow=c(2,3))  
qqnorm(EmployeeBiometric.df$AgeYears, col="red",  
       main = "Normality of AgeYears-Overall")  
qqline(EmployeeBiometric.df$AgeYears, col="blue")  
  
qqnorm(EmployeeBiometric.df$TotalCholesterolmgdL, col="red",  
       main="Normality of Total Cholesterol - Overall")  
qqline(EmployeeBiometric.df$TotalCholesterolmgdL, col="blue")  
  
qqnorm(EmployeeBiometric.df$SBPmmHg, col="red",
```

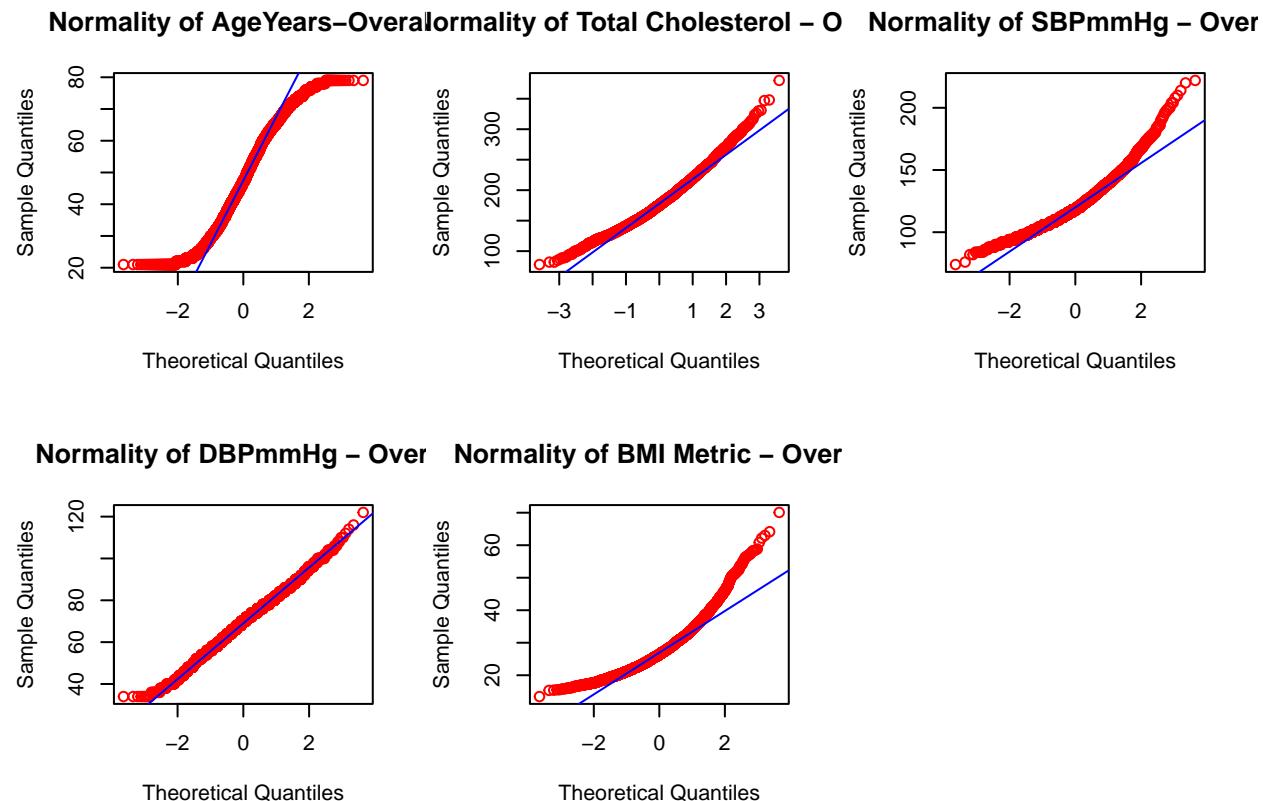
```

    main="Normality of SBPmmHg - Overall")
qqline(EmployeeBiometric.df$SBPmmHg, col="blue")

qqnorm(EmployeeBiometric.df$DBPmmHg, col="red",
       main="Normality of DBPmmHg - Overall")
qqline(EmployeeBiometric.df$DBPmmHg, col="blue")

qqnorm(EmployeeBiometric.df$BMIMetric, col="red",
       main="Normality of BMI Metric - Overall")
qqline(EmployeeBiometric.df$BMIMetric, col="blue")

```



As seen in the Q-Q plots provide ample evidence that the numeric object variables AgeYears, TotalCholesterolmgdL, SBPmmHg, DBPmmHg, and BMIMetric do not follow a normal distribution pattern.

The RVAideMemoire::byf.shapiro() function is an excellent choice to examine these questions on normality by breakouts. Observe below how the RVAideMemoire::byf.shapiro() function provides a simple and convenient summary of normality p-values by individual breakouts of the factor-type object variable.

1. Normality test of numeric object variables by Gender

```
RVAideMemoire:::byf.shapiro(AgeYears~Gender,
                           data=EmployeeBiometric.df)
```

```
##
##  Shapiro-Wilk normality tests
##
```

```

## data: AgeYears by Gender
##
##          W           p-value
## Female 0.9661 < 0.000000000000022 ***
## Male   0.9633 < 0.000000000000022 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

RVAideMemoire::byf.shapiro(TotalCholesterolmgdL~Gender,
                            data=EmployeeBiometric.df)

## 
## Shapiro-Wilk normality tests
##
## data: TotalCholesterolmgdL by Gender
##
##          W           p-value
## Female 0.9806 0.0000000000047433 ***
## Male   0.9739 0.000000000002233 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

RVAideMemoire::byf.shapiro(SBPmmHg ~ Gender,
                            data=EmployeeBiometric.df)

## 
## Shapiro-Wilk normality tests
##
## data: SBPmmHg by Gender
##
##          W           p-value
## Female 0.9318 < 0.000000000000022 ***
## Male   0.9653 < 0.000000000000022 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

RVAideMemoire::byf.shapiro(DBPmmHg ~ Gender,
                            data=EmployeeBiometric.df)

## 
## Shapiro-Wilk normality tests
##
## data: DBPmmHg by Gender
##
##          W           p-value
## Female 0.9945 0.000001046 ***
## Male   0.9964  0.0004106 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
RVAideMemoire::byf.shapiro(BMIMetric ~ Gender,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: BMIMetric by Gender  
##  
##          W          p-value  
## Female  0.9206 < 0.0000000000000022 ***  
## Male    0.9259 < 0.0000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. Normality test of numeric object variables by RaceEthnicity

```
RVAideMemoire::byf.shapiro(AgeYears ~ RaceEthnicity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: AgeYears by RaceEthnicity  
##  
##          W          p-value  
## Asian   0.9702  0.00000002483267275 ***  
## Black   0.9616  0.00000000000008035 ***  
## Hispanic 0.9648  0.00000000000005845 ***  
## Other   0.9336  0.00002059768431550 ***  
## White   0.9625 < 0.0000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(TotalCholesterolmgdL~RaceEthnicity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: TotalCholesterolmgdL by RaceEthnicity  
##  
##          W          p-value  
## Asian   0.9784  0.0000299780295813 ***  
## Black   0.9757  0.0000000073145531 ***  
## Hispanic 0.9783  0.0000000149272617 ***  
## Other   0.9642      0.01574 *  
## White   0.9777  0.000000000004407 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(SBPmmHg~RaceEthnicity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: SBPmmHg by RaceEthnicity  
##  
##          W      p-value  
## Asian    0.9415  0.0000000000222558 ***  
## Black    0.9551  0.0000000000001435 ***  
## Hispanic 0.9462 < 0.0000000000000022 ***  
## Other    0.9320  0.00002641595847946 ***  
## White    0.9448 < 0.0000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
```

```
RVAideMemoire::byf.shapiro(DBPmmHg~RaceEthnicity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: DBPmmHg by RaceEthnicity  
##  
##          W      p-value  
## Asian    0.9939  0.0657746 .  
## Black    0.9935  0.0020140 **  
## Hispanic 0.9942  0.0020703 **  
## Other    0.9736  0.0263387 *  
## White    0.9961  0.0005367 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
```

```
RVAideMemoire::byf.shapiro(BMIMetric~RaceEthnicity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: BMIMetric by RaceEthnicity  
##  
##          W      p-value  
## Asian    0.9373  0.000000000002847 ***  
## Black    0.9295 < 0.0000000000000022 ***  
## Hispanic 0.9266 < 0.0000000000000022 ***  
## Other    0.9217  0.0000038873197156 ***  
## White    0.9118 < 0.0000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
```

3. Normality test of numeric object variables by BMIstatus

```
RVAideMemoire::byf.shapiro(AgeYears~BMIStatus,
                            data=EmployeeBiometric.df)

##  
##  Shapiro-Wilk normality tests  
##  
## data: AgeYears by BMIStatus  
##  
##          W          p-value  
## Underweight 0.9526 0.0000193201320965490 ***  
## Normal Weight 0.9651 < 0.00000000000000022 ***  
## Overweight   0.9643 0.0000000000000006834 ***  
## Obese        0.9645 < 0.00000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

RVAideMemoire::byf.shapiro(TotalCholesterolmgdL~BMIStatus,
                            data=EmployeeBiometric.df)

##  
##  Shapiro-Wilk normality tests  
##  
## data: TotalCholesterolmgdL by BMIStatus  
##  
##          W          p-value  
## Underweight 0.9780      0.03646 *  
## Normal Weight 0.9710 0.00000000006071 ***  
## Overweight   0.9785 0.0000000050969220 ***  
## Obese        0.9847 0.0000001972027389 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

RVAideMemoire::byf.shapiro(SBPmmHg~BMIStatus,
                            data=EmployeeBiometric.df)

##  
##  Shapiro-Wilk normality tests  
##  
## data: SBPmmHg by BMIStatus  
##  
##          W          p-value  
## Underweight 0.8420 0.0000000001006922 ***  
## Normal Weight 0.9122 < 0.00000000000000022 ***  
## Overweight   0.9447 < 0.00000000000000022 ***  
## Obese        0.9702 0.0000000000001384 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(DBPmmHg~BMIStatus,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: DBPmmHg by BMIStatus  
##  
##          W      p-value  
## Underweight   0.9792  0.0180324 *  
## Normal Weight 0.9943  0.00005119 ***  
## Overweight    0.9947  0.0009071 ***  
## Obese         0.9944  0.0002697 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(BMIMetric~BMIStatus,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: BMIMetric by BMIStatus  
##  
##          W      p-value  
## Underweight   0.9201      0.00000005606 ***  
## Normal Weight 0.9626 < 0.00000000000000022 ***  
## Overweight    0.9590 < 0.00000000000000022 ***  
## Obese         0.8329 < 0.00000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Normality test of numeric object variables by Obesity

```
RVAideMemoire::byf.shapiro(AgeYears~Obesity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: AgeYears by Obesity  
##  
##          W      p-value  
## Not Obese  0.9647 < 0.00000000000000022 ***  
## Obese     0.9645 < 0.00000000000000022 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(TotalCholesterolmgdL~Obesity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: TotalCholesterolmgdL by Obesity  
##  
##          W           p-value  
## Not Obese 0.9754 < 0.0000000000000002 ***  
## Obese     0.9847      0.00000001972 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(SBPmmHg~Obesity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: SBPmmHg by Obesity  
##  
##          W           p-value  
## Not Obese 0.9298 < 0.0000000000000002 ***  
## Obese     0.9702      0.0000000000001384 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(DBPmmHg~Obesity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##  
## data: DBPmmHg by Obesity  
##  
##          W           p-value  
## Not Obese 0.9953 0.000000308 ***  
## Obese     0.9944  0.0002697 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RVAideMemoire::byf.shapiro(BMIMetric~Obesity,  
                           data=EmployeeBiometric.df)
```

```
##  
##  Shapiro-Wilk normality tests  
##
```

```

## data: BMIMetric by Obesity
##
##              W          p-value
## Not Obese  0.9789 < 0.0000000000000022 ***
## Obese      0.8329 < 0.0000000000000022 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

For additional confirmation, use the `ggplot2::ggplot()` function to produce QQ plots of the numeric object variables (e.g., AgeYears, TotalCholesterolmgdL, SBPmmHg, DBPmmHg, and BMIMetric) by breakouts of each of the four factor-type object variables (e.g., Gender, RaceEthnicity, BMIStatus, and Obesity)

1.AgeYears

```

QQAgeYearsbyGender<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=AgeYears))+
  stat_qq(color="red")+
  stat_qq_line(color="blue", linewidth=1.75)+
  facet_grid(.~Gender)+
  ggtitle("QQ-AgeYears")

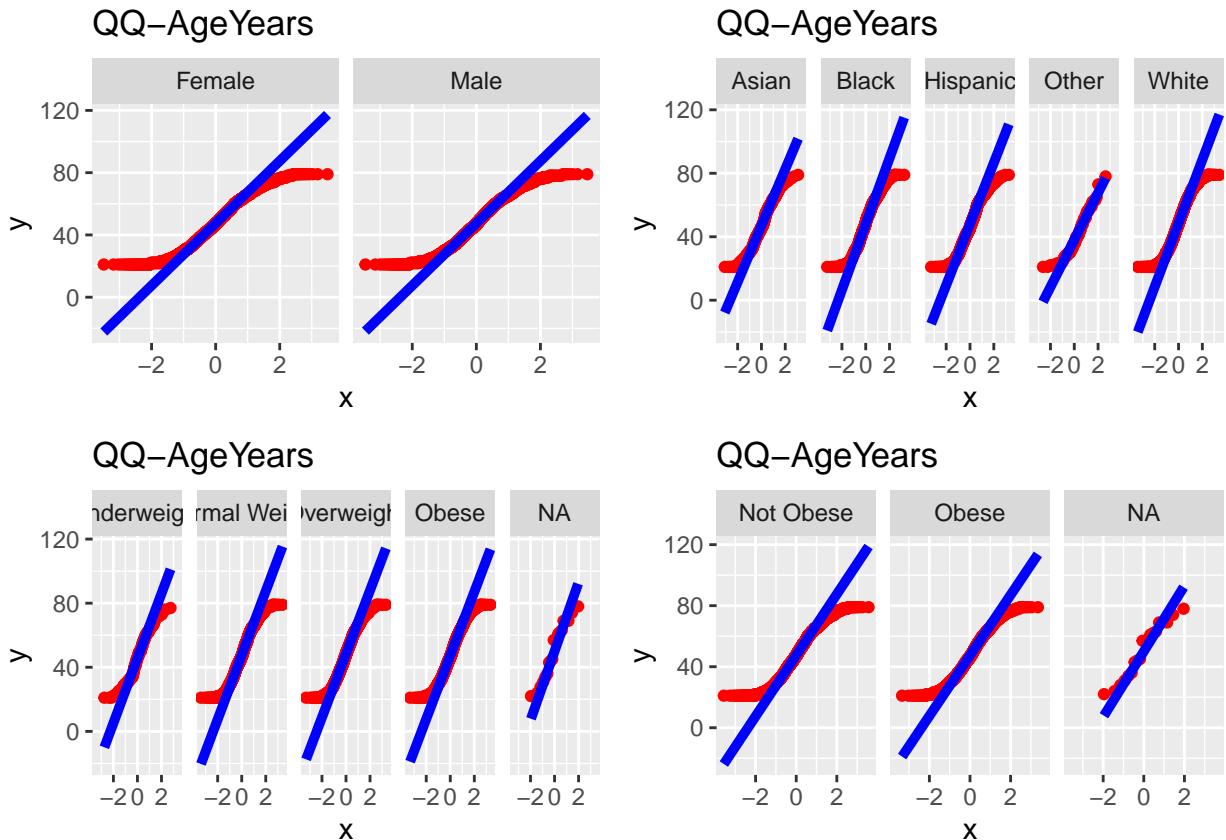
QQAgeYearsbyRaceEthnicity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=AgeYears))+
  stat_qq(color="red")+
  stat_qq_line(color="blue", linewidth=1.75)+
  facet_grid(.~RaceEthnicity)+
  ggtitle("QQ-AgeYears")

QQAgeYearsbyBMISatus<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=AgeYears))+
  stat_qq(color="red")+
  stat_qq_line(color="blue", linewidth=1.75)+
  facet_grid(.~BMISatus)+
  ggtitle("QQ-AgeYears")

QQAgeYearsbyObesity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=AgeYears))+
  stat_qq(color="red")+
  stat_qq_line(color="blue", linewidth=1.75)+
  facet_grid(.~Obesity)+
  ggtitle("QQ-AgeYears")

par(ask=TRUE); gridExtra::grid.arrange(
  QQAgeYearsbyGender,
  QQAgeYearsbyRaceEthnicity,
  QQAgeYearsbyBMISatus,
  QQAgeYearsbyObesity, ncol=2)

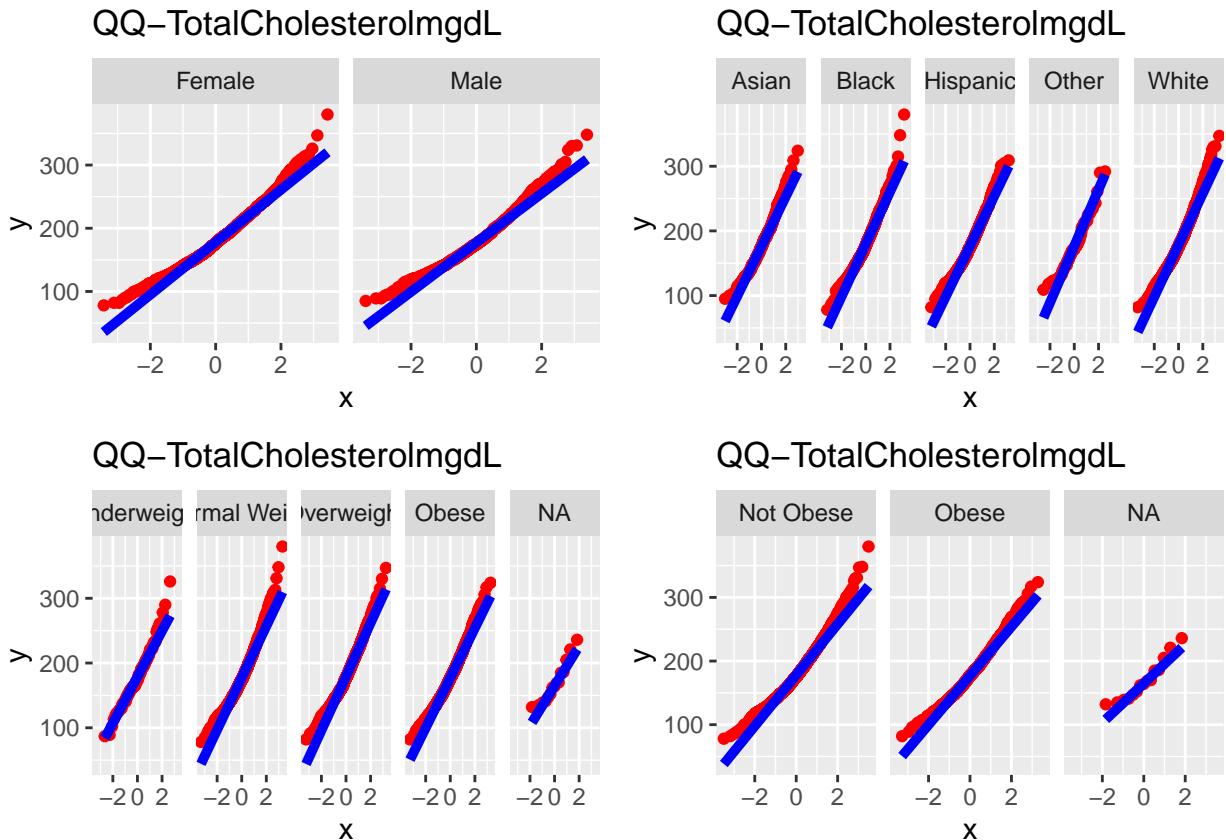
```



2. TotalCholesterol

```
QQTotalCholesterolbyGender<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=TotalCholesterolmgdL))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Gender) +  
  ggtitle("QQ-TotalCholesterolmgdL")  
  
QQTotalCholesterolbyRaceEthnicity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=TotalCholesterolmgdL))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~RaceEthnicity) +  
  ggtitle("QQ-TotalCholesterolmgdL")  
  
QQTotalCholesterolbyBMIStatus<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=TotalCholesterolmgdL))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~BMIStatus) +  
  ggtitle("QQ-TotalCholesterolmgdL")  
  
QQTotalCholesterolbyObesity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=TotalCholesterolmgdL))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Obesity) +  
  ggtitle("QQ-TotalCholesterolmgdL")  
  
par(ask=TRUE); gridExtra::grid.arrange(
```

```
QQTotalCholesterolbyGender,  
QQTotalCholesterolbyRaceEthnicity,  
QQTotalCholesterolbyBMIStatus,  
QQTotalCholesterolbyObesity, ncol=2)  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 894 rows containing non-finite outside the scale  
## range ('stat_qq()').
```



3. SBPmmHg

```

QQSBPbyGender<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=SBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Gender) +  
  ggttitle("QQ-SBPmmHg")

QQSBPbyRaceEthnicity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=SBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~RaceEthnicity) +  
  ggttitle("QQ-SBPmmHg")

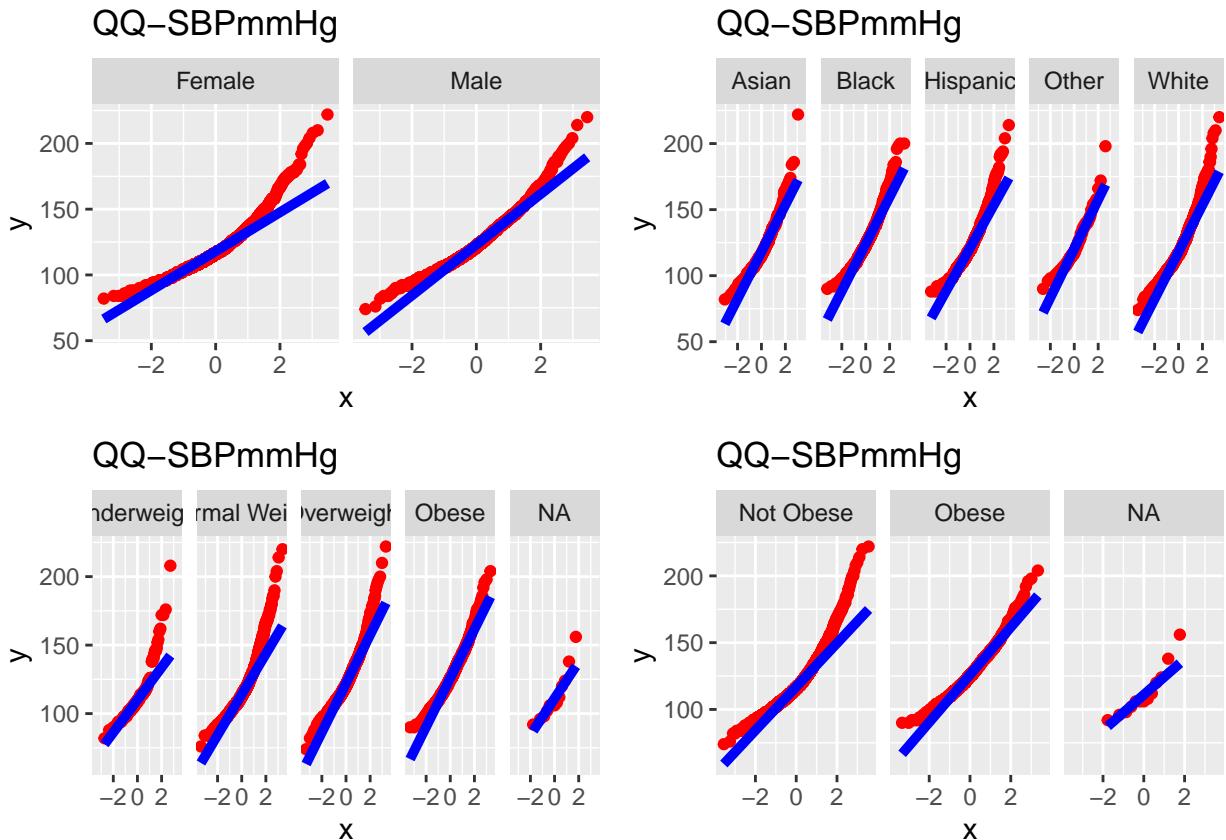
QQSBPbyBMIStatus<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=SBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~BMIStatus) +  
  ggttitle("QQ-SBPmmHg")

QQSBPbyObesity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=SBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Obesity) +  
  ggttitle("QQ-SBPmmHg")

par(ask=TRUE); gridExtra::grid.arrange(

```

```
QQSBPbyGender,  
QQSBPbyRaceEthnicity,  
QQSBPbyBMIStatus,  
QQSBPbyObesity, ncol=2)  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 209 rows containing non-finite outside the scale  
## range ('stat_qq()').
```



4. DBPmmHg

```

QQDBPbyGender<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=DBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Gender) +  
  ggttitle("QQ-DBPmmHg")

QQDBPbyRaceEthnicity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=DBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~RaceEthnicity) +  
  ggttitle("QQ-DBPmmHg")

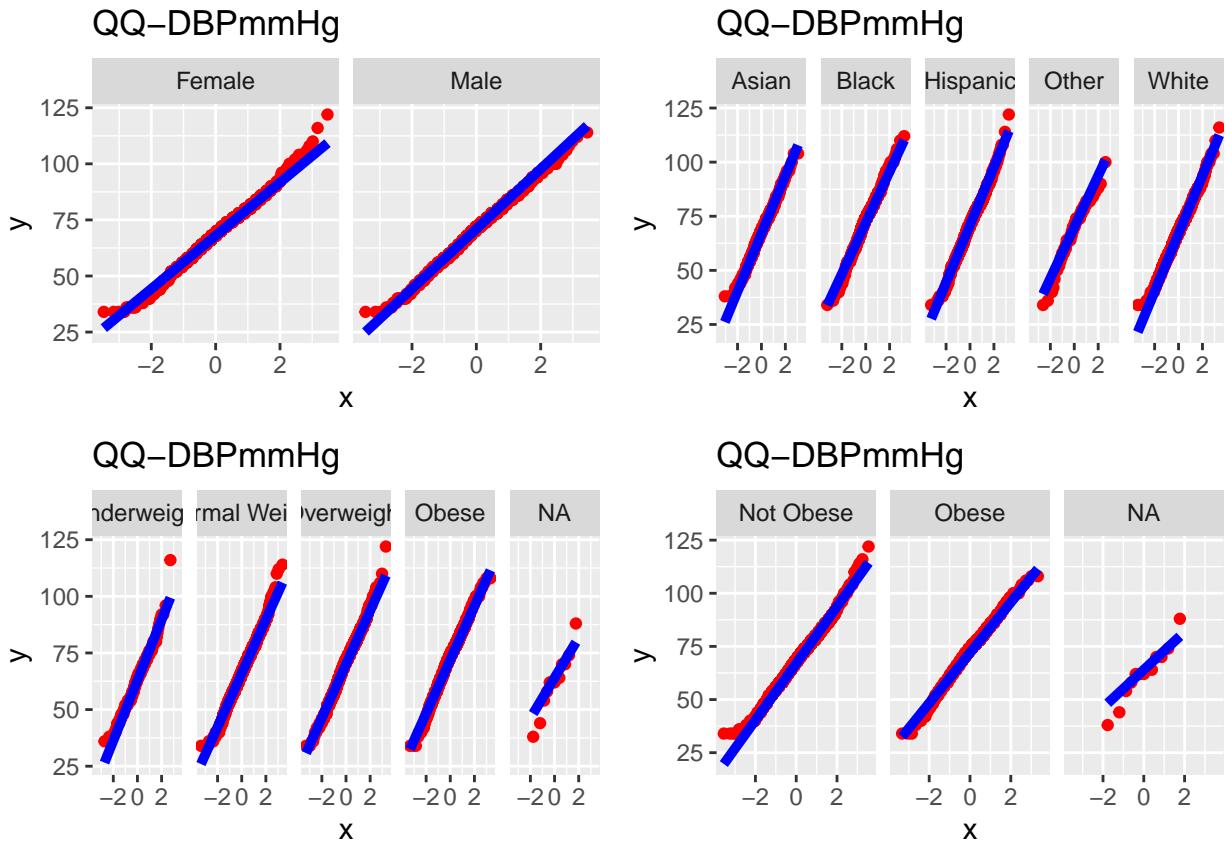
QQDBPbyBMISatus<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=DBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~BMISatus) +  
  ggttitle("QQ-DBPmmHg")

QQDBPbyObesity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=DBPmmHg))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Obesity) +  
  ggttitle("QQ-DBPmmHg")

par(ask=TRUE); gridExtra::grid.arrange(

```

```
QQDBPbyGender,  
QQDBPbyRaceEthnicity,  
QQDBPbyBMIStatus,  
QQDBPbyObesity, ncol=2)  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 217 rows containing non-finite outside the scale  
## range ('stat_qq()').
```



5. BMIMetric

```

QQBMIMetricbyGender<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=BMIMetric))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Gender) +  
  ggtitle("QQ-BMIMetric")

QQBMIMetricbyRaceEthnicity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=BMIMetric))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~RaceEthnicity) +  
  ggtitle("QQ-BMIMetric")

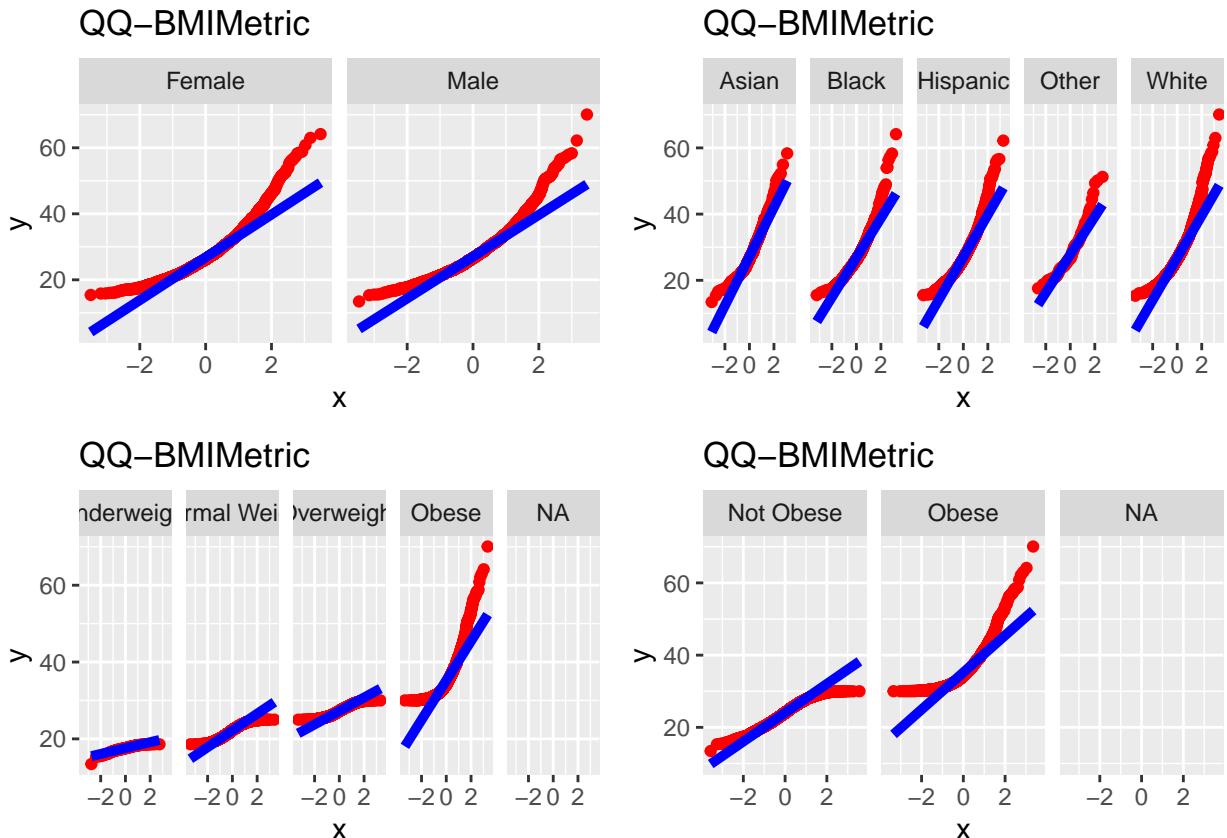
QQBMIMetricbyBMIStatus<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=BMIMetric))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~BMIStatus) +  
  ggtitle("QQ-BMIMetric")

QQBMIMetricbyObesity<-ggplot2::ggplot(EmployeeBiometric.df,aes(sample=BMIMetric))+  
  stat_qq(color="red") +  
  stat_qq_line(color="blue", linewidth=1.75) +  
  facet_grid(.~Obesity) +  
  ggtitle("QQ-BMIMetric")

par(ask=TRUE); gridExtra::grid.arrange(

```

```
QQBMIMetricbyGender,  
QQBMIMetricbyRaceEthnicity,  
QQBMIMetricbyBMIStatus,  
QQBMIMetricbyObesity, ncol=2)  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq_line()').  
  
## Warning: Removed 20 rows containing non-finite outside the scale  
## range ('stat_qq()').
```



```
lbsgen <- read.table(file="CPIIISectionLbsGender.csv", header=TRUE, dec=". ", sep = ",")  
library(asbio)
```

```
## Loading required package: tcltk  
  
##  
## Attaching package: 'asbio'  
  
## The following object is masked from 'package:lubridate':  
##  
##     pm  
  
library(epiDisplay)  
  
## Loading required package: foreign  
  
## Loading required package: survival  
  
## Loading required package: MASS  
  
##  
## Attaching package: 'MASS'
```

```

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

## Loading required package: nnet

##
## Attaching package: 'epiDisplay'

## The following object is masked from 'package:ggplot2':
##
##      alpha

## The following object is masked from 'package:gmodels':
##
##      ci

asbio::Mode(lbsgen$Lbs)

## [1] 114 122

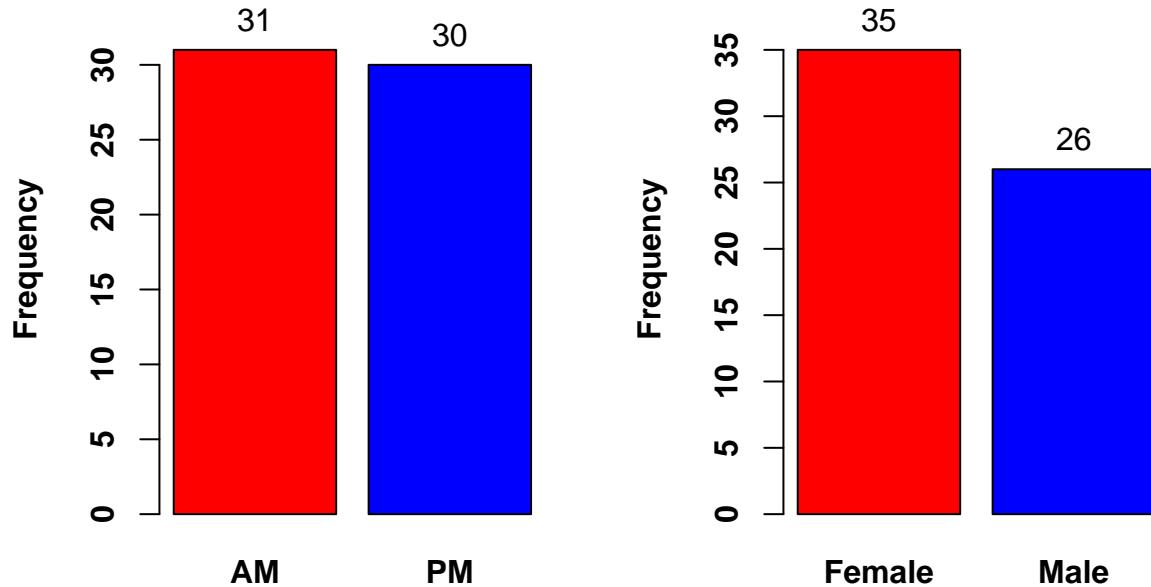
#Frequency Table
par(ask=TRUE)
par(mfrow=c(1,2))
epiDisplay::tab1(lbsgen$Section,
                 main="Frequency Distribution of Section",
                 col=c("red","blue"), font.lab=2, font.axis=2)

## lbsgen$Section :
##          Frequency Percent Cum. percent
## AM            31     50.8       50.8
## PM            30     49.2       100.0
## Total         61    100.0      100.0

epiDisplay::tab1(lbsgen$Gender, main="Frequency Distribution of Gender", col=c("red", "blue"), font.lab=

```

Frequency Distribution of Section by Gender



```
## lbsgen$Gender :
##          Frequency Percent Cum. percent
## Female      35     57.4      57.4
## Male        26     42.6     100.0
## Total       61    100.0     100.0
```

```
par(ask=TRUE)
par(mfrow=c(1,2))
epiDisplay:::tabpct(lbsgen$Section,
                    lbsgen$Gender, graph = TRUE, decimal=1,
                    main="Frequency Distribution of Section by Gender",
                    xlab="Section", ylab="Gender", cex.axis=1,
                    percent="both", las=1, col = c("red","blue"))
```

```
##
## Original table
##           lbsgen$Gender
## lbsgen$Section Female Male Total
##      AM      15   16   31
##      PM      20   10   30
##      Total    35   26   61
##
## Row percent
##           lbsgen$Gender
## lbsgen$Section Female Male Total
```

```

##          AM      15      16      31
##                (48.4)  (51.6)  (100)
##          PM      20      10      30
##                (66.7)  (33.3)  (100)
##
## Column percent
##           lbsgen$Gender
## lbsgen$Section  Female      %  Male      %
##          AM      15  (42.9)    16  (61.5)
##          PM      20  (57.1)    10  (38.5)
##          Total    35  (100)    26  (100)

epiDisplay::tabpct(lbsgen$Gender,
                   lbsgen$Section, graph = TRUE, decimal = 1,
                   main="Frequency Distribution of Gender by Section",
                   xlab="Gender", ylab="Section", cex.axis=1,
                   percent="both", las=1, col=c("red","blue"))

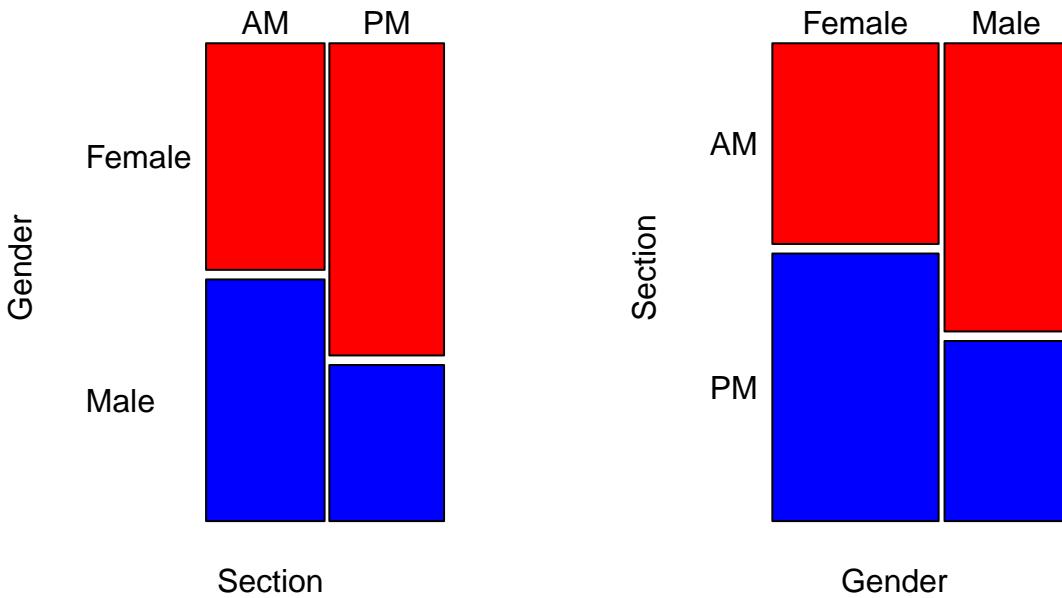
```

```

##
## Original table
##           lbsgen$Section
## lbsgen$Gender  AM  PM  Total
##          Female  15  20   35
##          Male    16  10   26
##          Total   31  30   61
##
## Row percent
##           lbsgen$Section
## lbsgen$Gender      AM      PM  Total
##          Female    15      20   35
##                (42.9)  (57.1) (100)
##          Male     16      10   26
##                (61.5)  (38.5) (100)
##
## Column percent
##           lbsgen$Section
## lbsgen$Gender  AM      %  PM      %
##          Female  15  (48.4)  20  (66.7)
##          Male    16  (51.6)  10  (33.3)
##          Total   31  (100)   30  (100)

```

Frequency Distribution of Section by Frequency Distribution of Gender by Section



```
par(ask=TRUE)
par(mfrow=c(1,2))
epiDisplay::tabpct(EmployeeBiometric.df$RaceEthnicity,
                    EmployeeBiometric.df$Gender, graph = TRUE, decimal=1,
                    main="Frequency Distribution of Race Ethnicity by Gender",
                    xlab="Race Ethnicity", ylab="Gender", cex.axis=1,
                    percent="both", las=1, col = c("cyan","green"))
```

```
##
## Original table
##                                     EmployeeBiometric.df$Gender
## EmployeeBiometric.df$RaceEthnicity   Female   Male  Total
##                               Asian     250    232   482
##                               Black    419    399   818
##                               Hispanic 484    420   904
##                               Other    71     46   117
##                               White   854    770  1624
##                               Total   2078   1867 3945
##
## Row percent
##                                     EmployeeBiometric.df$Gender
## EmployeeBiometric.df$RaceEthnicity   Female   Male  Total
##                               Asian     250    232   482
##                               (51.9)  (48.1) (100)
##                               Black    419    399   818
```

```

##          (51.2)  (48.8)  (100)
## Hispanic    484     420     904
##          (53.5)  (46.5)  (100)
## Other       71      46     117
##          (60.7)  (39.3)  (100)
## White      854     770    1624
##          (52.6)  (47.4)  (100)
##
## Column percent
##                                     EmployeeBiometric.df$Gender
## EmployeeBiometric.df$RaceEthnicity Female   % Male
##           Asian      250 (12.0) 232
##           Black      419 (20.2) 399
##           Hispanic   484 (23.3) 420
##           Other      71  (3.4)  46
##           White      854 (41.1) 770
##           Total      2078 (100) 1867
##                                     EmployeeBiometric.df$Gender
## EmployeeBiometric.df$RaceEthnicity %
##           Asian     (12.4)
##           Black     (21.4)
##           Hispanic (22.5)
##           Other     (2.5)
##           White     (41.2)
##           Total     (100)

epiDisplay::tabpct(EmployeeBiometric.df$Gender,
                    EmployeeBiometric.df$RaceEthnicity, graph = TRUE, decimal = 1,
                    main="Frequency Distribution of Gender by Race Ethnicity",
                    xlab="Gender", ylab="Section", cex.axis=1,
                    percent="both", las=1, col=c("cyan","green"))

```

```

##
## Original table
##                                     EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender Asian Black Hispanic Other
##           Female      250  419    484    71
##           Male       232  399    420    46
##           Total      482  818    904   117
##                                     EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender White Total
##           Female     854  2078
##           Male      770  1867
##           Total     1624 3945
##
## Row percent
##                                     EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender Asian Black Hispanic
##           Female      250  419    484
##           (12) (20.2) (23.3)
##           Male       232  399    420
##           (12.4) (21.4) (22.5)
##                                     EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender Other White Total

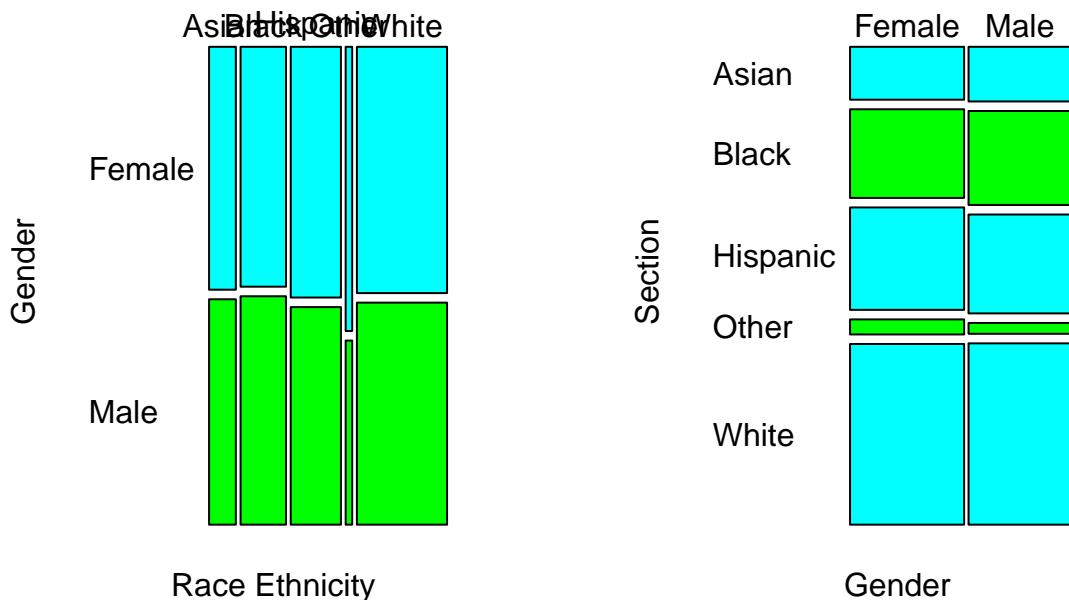
```

```

##          Female    71    854   2078
##                  (3.4)  (41.1) (100)
##          Male     46    770   1867
##                  (2.5)  (41.2) (100)
##
## Column percent
##           EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender Asian      % Black      %
##          Female    250  (51.9)  419  (51.2)
##          Male     232  (48.1)  399  (48.8)
##          Total    482  (100)   818  (100)
##           EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender Hispanic    % Other
##          Female    484  (53.5)   71
##          Male     420  (46.5)   46
##          Total    904  (100)  117
##           EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender      % White      %
##          Female  (60.7)  854  (52.6)
##          Male   (39.3)  770  (47.4)
##          Total   (100) 1624  (100)

```

Frequency Distribution of Race Ethnicity by Gender



```

epiDisplay::tabpct(EmployeeBiometric.df$Gender,
EmployeeBiometric.df$RaceEthnicity, graph = TRUE, decimal = 1,
main="Frequency Distribution of Gender by Race Ethnicity",

```

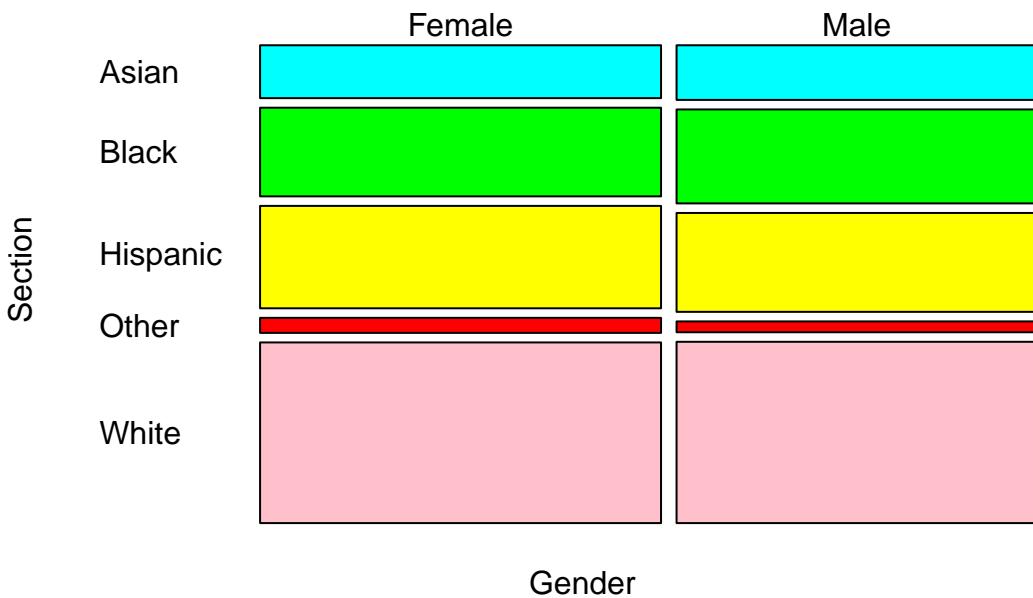
```

      xlab="Gender", ylab="Section", cex.axis=1,
      percent="both", las=1, col=c("cyan","green","yellow","red","pink"))

## Original table
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  Asian  Black  Hispanic  Other
##             Female    250     419     484      71
##             Male     232     399     420      46
##             Total    482     818     904     117
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  White  Total
##             Female    854    2078
##             Male     770    1867
##             Total   1624    3945
##
## Row percent
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  Asian  Black  Hispanic
##             Female    250     419     484
##                     (12)   (20.2)   (23.3)
##             Male     232     399     420
##                     (12.4)  (21.4)  (22.5)
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  Other  White  Total
##             Female    71     854    2078
##                     (3.4)  (41.1)  (100)
##             Male     46     770    1867
##                     (2.5)  (41.2)  (100)
##
## Column percent
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  Asian    %  Black    %
##             Female    250  (51.9)  419  (51.2)
##             Male     232  (48.1)  399  (48.8)
##             Total    482  (100)  818  (100)
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  Hispanic  %  Other
##             Female    484  (53.5)  71
##             Male     420  (46.5)  46
##             Total    904  (100)  117
## EmployeeBiometric.df$RaceEthnicity
## EmployeeBiometric.df$Gender  %  White    %
##             Female  (60.7)  854  (52.6)
##             Male   (39.3)  770  (47.4)
##             Total  (100)  1624  (100)

```

Frequency Distribution of Gender by Race Ethnicity



```

par(ask=TRUE)
par(mfrow=c(1,3))
epiDisplay::summ(lbsgen$Lbs, by=NULL,
                 graph=TRUE, box=TRUE, pch=20, ylab="auto", main="Sorted Dotplot of Weight (Lbs), Overall",
                 cex.X.axis=1.25,
                 cex.Y.axis=1.25,
                 font.lab=2, dot.col="auto")

##   obs. mean   median   s.d.   min.   max.
##   60    131.733 127      17.589 99      192

epiDisplay::summ(lbsgen$Lbs, by=lbsgen$Section,
                 graph=TRUE,
                 pch=20, ylab="auto",
                 main="Sorted Dotplot of Weight (Lbs) by Section",
                 cex.X.axis=1.25,
                 cex.Y.axis=1.25,
                 font.lab=2, dot.col="auto")

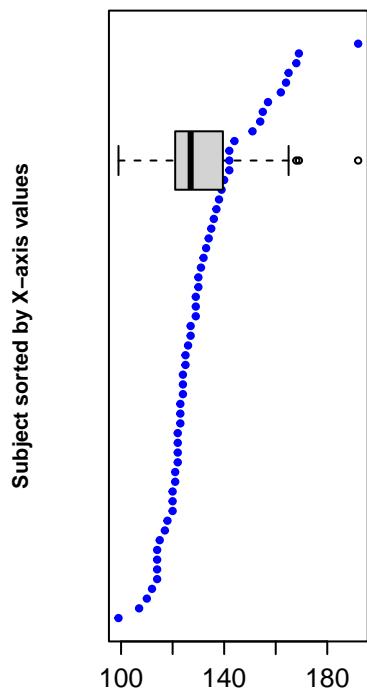
## For lbsgen$Section = AM
##   obs. mean   median   s.d.   min.   max.
##   30    128.3 126      13.152 107      157
##
## For lbsgen$Section = PM
##   obs. mean   median   s.d.   min.   max.

```

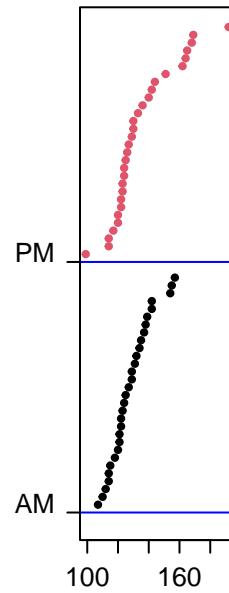
```
##   30    135.167 128      20.786 99      192
```

```
epiDisplay::summ(lbsgen$Lbs, by=lbsgen$Gender,
                 graph=TRUE,
                 pch=20, ylab="auto", main="Sorted Dotplot of Weight (Lbs)
                 by Gender",
                 cex.X.axis=1.25,
                 cex.Y.axis=1.25,
                 font.lab=2,
                 dot.col="auto")
```

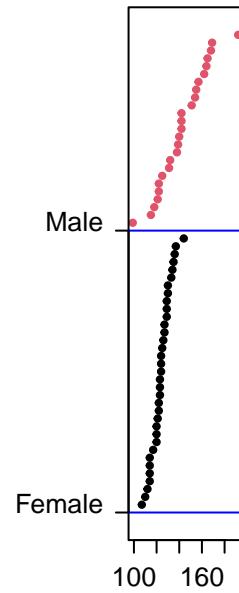
Sorted Dotplot of Weight (Lbs), Overall



Sorted Dotplot of Weight (Lbs) by Gender



Sorted Dotplot of Weight (Lbs) by Gender



```
## For lbsgen$Gender = Female
##   obs. mean   median   s.d.   min.   max.
##   35    123.971 124     8.28    107    144
##
## For lbsgen$Gender = Male
##   obs. mean   median   s.d.   min.   max.
##   25    142.6   142     21.274 99     192

library(s20x)
#Create descriptive summary for variable Lbs
s20x::summaryStats(lbsgen$Lbs, na.rm=TRUE)
```

```
## Minimum value: 99
```

```

## Maximum value:          192
## Mean value:            131.73
## Median:                127
## Upper quartile:        139.25
## Lower quartile:        121
## Variance:              309.39
## Standard deviation:   17.59
## Midspread (IQR):      18.25
## Skewness:               1.08
## Number of data values: 61
## Number of missing values: 1

#Summary stat of Lbs by gender
s20x::summaryStats(Lbs~Gender, lbsgen,
                    na.rm=TRUE)

##           Sample Size No. Miss.    Mean Median Std Dev
## Female       35          0 123.971   124 8.27997
## Male         25          1 142.600   142 21.27401
##           Midspread
## Female       9
## Male         32

#Summary stat of Lbs by Section
s20x::summaryStats(Lbs~Section, lbsgen,
                    na.rm=TRUE)

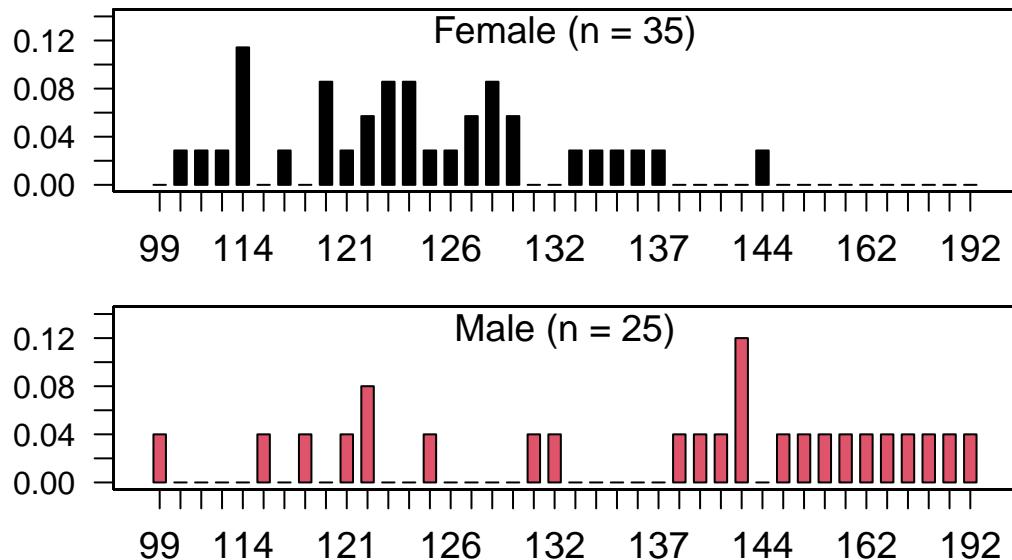
##           Sample Size No. Miss.    Mean Median Std Dev Midspread
## AM           30          1 128.300   126 13.1520    16.25
## PM           30          0 135.167   128 20.7864    21.25

#Creating distribution list of Lbs by Gender
par(ask=TRUE);
s20x::rowdistr(crosstablist = xtabs(~Gender + Lbs, data = lbsgen),
                plot = TRUE,
                suppressText = FALSE,
                comp = 'basic')

## Row Proportions
##          99 107 110 112 114 115 117 118 120 121
## Female  0.00 0.03 0.03 0.03 0.11 0.00 0.03 0.00 0.09 0.03
## Male    0.04 0.00 0.00 0.00 0.00 0.04 0.00 0.04 0.00 0.04
##          122 123 124 125 126 127 129 130 131 132
## Female  0.06 0.09 0.09 0.03 0.03 0.06 0.09 0.06 0.00 0.00
## Male    0.08 0.00 0.00 0.04 0.00 0.00 0.00 0.00 0.04 0.04
##          133 134 135 136 137 138 139 140 142 144
## Female  0.03 0.03 0.03 0.03 0.03 0.00 0.00 0.00 0.00 0.03
## Male    0.00 0.00 0.00 0.00 0.00 0.04 0.04 0.04 0.12 0.00
##          151 154 155 157 162 164 165 168 169 192
## Female  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## Male    0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04
##          Totals n
## Female   1 35
## Male     1 25

```

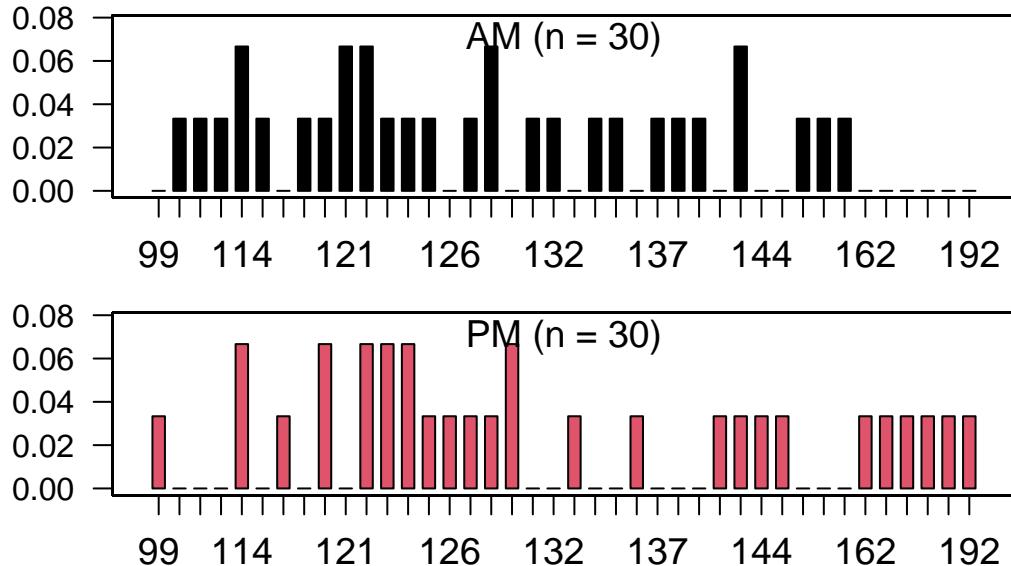
**fac2 distribution for each
level of fac1 (row proportions)**



```
#Creating distribution list of Lbs by Section
par(ask=TRUE); s20x::rowdistr(crosstablist = xtabs(~Section + Lbs, data = lbsgen),
  plot = TRUE,
  suppressText = FALSE,
  comp = 'basic')
```

```
## Row Proportions
##    99 107 110 112 114 115 117 118 120 121 122
## AM 0.00 0.03 0.03 0.03 0.07 0.03 0.00 0.03 0.03 0.07 0.07
## PM 0.03 0.00 0.00 0.00 0.07 0.00 0.03 0.00 0.07 0.00 0.07
##    123 124 125 126 127 129 130 131 132 133 134
## AM 0.03 0.03 0.03 0.00 0.03 0.07 0.00 0.03 0.03 0.00 0.03
## PM 0.07 0.07 0.03 0.03 0.03 0.03 0.07 0.00 0.00 0.03 0.00
##    135 136 137 138 139 140 142 144 151 154 155
## AM 0.03 0.00 0.03 0.03 0.03 0.00 0.07 0.00 0.00 0.03 0.03
## PM 0.00 0.03 0.00 0.00 0.00 0.03 0.03 0.03 0.03 0.00 0.00
##    157 162 164 165 168 169 192 Totals n
## AM 0.03 0.00 0.00 0.00 0.00 0.00 1 30
## PM 0.00 0.03 0.03 0.03 0.03 0.03 1 30
```

fac2 distribution for each level of fac1 (row proportions)



```

library(arsenal)

##
## Attaching package: 'arsenal'

## The following object is masked from 'package:lubridate':
##       is.Date

## The following object is masked from 'package:Hmisc':
##       %nin%

```

#Summary of descriptive statistics

```

summary(arsenal::tableby(list(Section, Gender)~Lbs,
                         data=lbsgen), text=TRUE, total=TRUE)

```

```

##
##          |      AM (N=31)      |      PM (N=30)      |      Total (N=61)      | p value|
## |:-----|:-----|:-----|:-----|:-----|:-----|
## |Lbs    |           |           |           |           | 0.132 |
## |- N-Miss |       1       |       0       |       1       |       |
## |- Mean (SD) | 128.300 (13.152) | 135.167 (20.786) | 131.733 (17.589) |       |

```

```

## | - Range      | 107.000 - 157.000 | 99.000 - 192.000 | 99.000 - 192.000 |           |
## 
## 
## |           | Female (N=35)   | Male (N=26)    | Total (N=61)   | p value|
## |:-----|:-----|:-----|:-----|:-----|:-----|
## |Lbs    |           |           |           |           | < 0.001|
## |- N-Miss | 0          | 1          | 1          |           |           |
## |- Mean (SD) | 123.971 (8.280) | 142.600 (21.274) | 131.733 (17.589) |           |
## |- Range    | 107.000 - 144.000 | 99.000 - 192.000 | 99.000 - 192.000 |           |

```

#Pivot table for Gender (rows) and Section (column)

```
library(pivottabler)
```

```
pivottabler::qpvt(lbsgen, "Gender", "Section",
  c("Mean Lbs"="mean(Lbs, na.rm=TRUE)",
    "SD Lbs"="sd(Lbs, na.rm=TRUE)",  

  formats=list("%.0f", "%.1f"))
```

	AM			PM			Total				
	Mean	Lbs	SD	Mean	Lbs	SD	Lbs	Mean	Lbs	SD	Lbs
## Female	122	9.5		125	7.3			124	8.3		
## Male	134	13.9		155	24.7			143	21.3		
## Total	128	13.2		135	20.8			132	17.6		

```
pivottabler::qpvt(lbsgen, "Section", "Gender",
  c("Mean Lbs"="mean(Lbs, na.rm=TRUE)",
    "SD Lbs"="sd(Lbs, na.rm=TRUE)",  

  formats=list("%.0f", "%.1f"))
```

	Female			Male			Total				
	Mean	Lbs	SD	Mean	Lbs	SD	Lbs	Mean	Lbs	SD	Lbs
## AM	122	9.5		134	13.9			128	13.2		
## PM	125	7.3		155	24.7			135	20.8		
## Total	124	8.3		143	21.3			132	17.6		

```
pivottabler::qpvt(EmployeeBiometric.df, "RaceEthnicity", "Gender",
  c("Mean SBP"="mean(SBPmmHg, na.rm=TRUE)",
    "SD SBP"="sd(SBPmmHg, na.rm=TRUE)",  

  formats=list("%.0f", "%.1f"))
```

	Female			Male			Total				
	Mean	SBP	SD	Mean	SBP	SD	SBP	Mean	SBP	SD	SBP
## Asian	119	18.2		119	17.8			119	18.0		
## Black	120	17.5		132	17.2			126	18.4		
## Hispanic	120	18.2		126	18.4			123	18.5		
## Other	123	19.1		121	16.3			122	18.0		
## White	119	18.0		121	19.2			120	18.6		
## Total	119	18.0		124	18.9			122	18.6		

```
# Set the seed
```

```
base::set.seed(8)
```

```
# Data with Normal Distribution Patterns
```

```

SBPNormal <- stats::rnorm(100000, mean=120, sd=06)

# Data That Do Not Exhibit Normal Distribution Patterns
SBPNotNormal <- stats::runif(100000, min=102, max=138)

```

Exploratory Data Analysis

```

base::getwd()

## [1] "C:/Users/linan/Documents/GitHub/project/R-project/using-r-for-biostatistics"

base::ls()

##  [1] "EmployeeBiometric.df"
##  [2] "lbsgen"
##  [3] "QQAgeYearsbyBMISatus"
##  [4] "QQAgeYearsbyGender"
##  [5] "QQAgeYearsbyObesity"
##  [6] "QQAgeYearsbyRaceEthnicity"
##  [7] "QQBMIMetricbyBMISatus"
##  [8] "QQBMIMetricbyGender"
##  [9] "QQBMIMetricbyObesity"
## [10] "QQBMIMetricbyRaceEthnicity"
## [11] "QQDBPbyBMISatus"
## [12] "QQDBPbyGender"
## [13] "QQDBPbyObesity"
## [14] "QQDBPbyRaceEthnicity"
## [15] "QQSBPbyBMISatus"
## [16] "QQSBPbyGender"
## [17] "QQSBPbyObesity"
## [18] "QQSBPbyRaceEthnicity"
## [19] "QQTotalCholesterolbyBMISatus"
## [20] "QQTotalCholesterolbyGender"
## [21] "QQTotalCholesterolbyObesity"
## [22] "QQTotalCholesterolbyRaceEthnicity"
## [23] "SBPNormal"
## [24] "SBPNotNormal"

utils::str(SBPNormal)

## num [1:100000] 119 125 117 117 124 ...
##  [1] 119.492 125.042 117.219 116.695 124.416 119.353 118.978
##  [8] 113.470 101.934 116.441

```

```

utils::tail(SBPNormal, n=10) # Show the tail, last 10 cases

## [1] 123.309 116.203 120.248 122.950 115.576 122.086 114.251
## [8] 113.029 128.144 130.011

base::summary(SBPNormal)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      94.6   116.0   120.0   120.0   124.1   143.6

# Summary statistics
utils::str(SBPNotNormal) # Identify structure

## num [1:100000] 114 111 120 137 129 ...

```

```

utils::head(SBPNotNormal, n=10) # Show the head, first 10 cases

## [1] 113.562 110.940 120.247 136.714 128.528 110.294 118.809
## [8] 104.221 135.889 120.197

utils::tail(SBPNotNormal, n=10) # Show the tail, last 10 cases

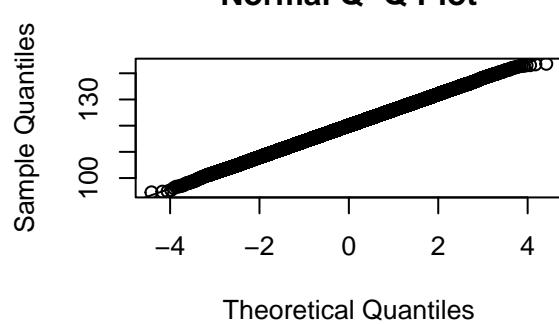
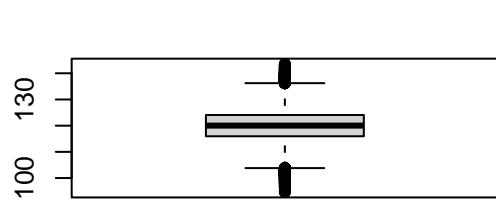
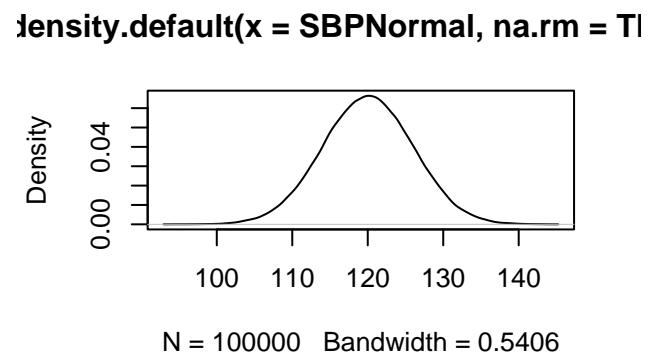
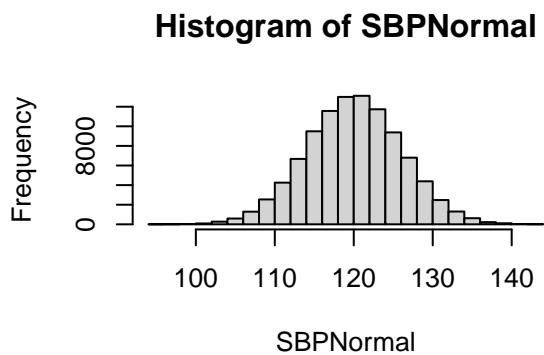
## [1] 109.066 116.185 114.395 127.216 121.895 137.712 103.724
## [8] 105.628 110.198 118.988

base::summary(SBPNotNormal) # Summary statistics

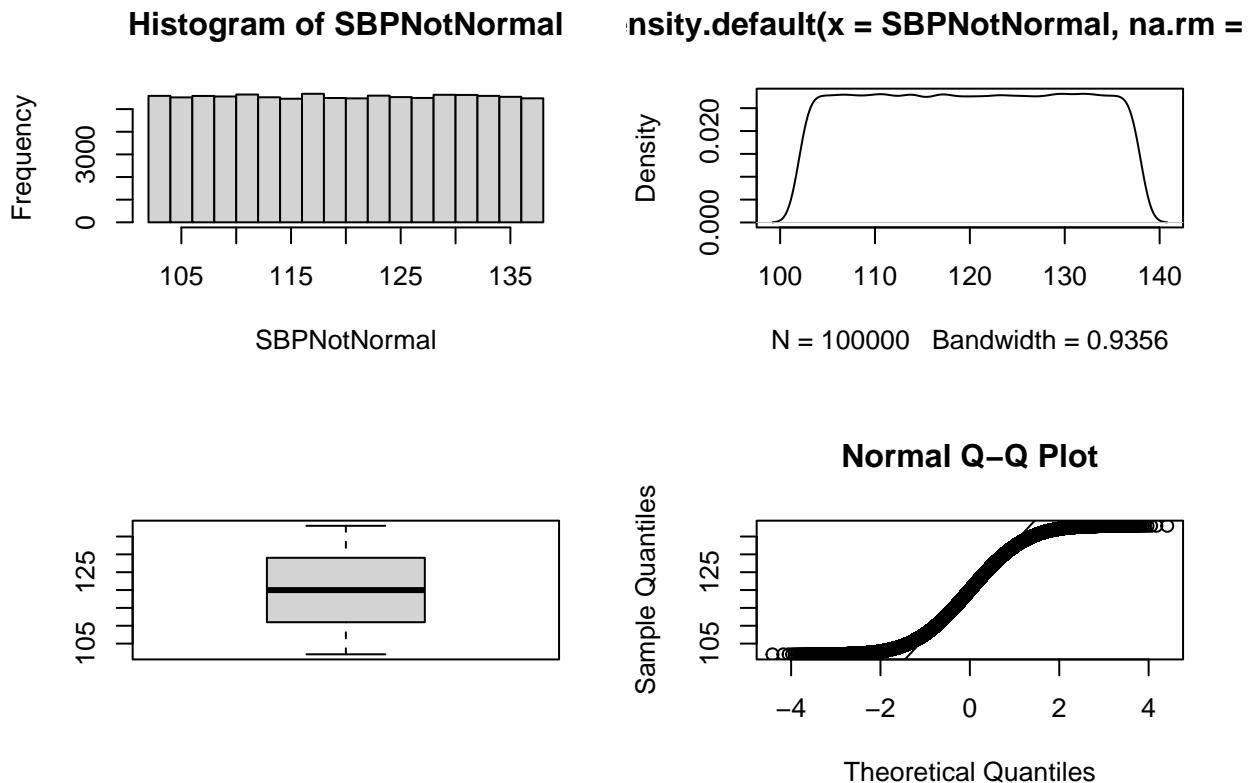
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      102     111     120     120     129     138

par.ask=TRUE)
par(mfrow=c(2,2)) # Pause # 4 figures - 2 rows by 2 column grid
graphics::hist(SBPNormal)
graphics::plot(stats::density(SBPNormal, na.rm=TRUE))
graphics::boxplot(SBPNormal)
stats::qqnorm(SBPNormal);
stats::qqline(SBPNormal)

```

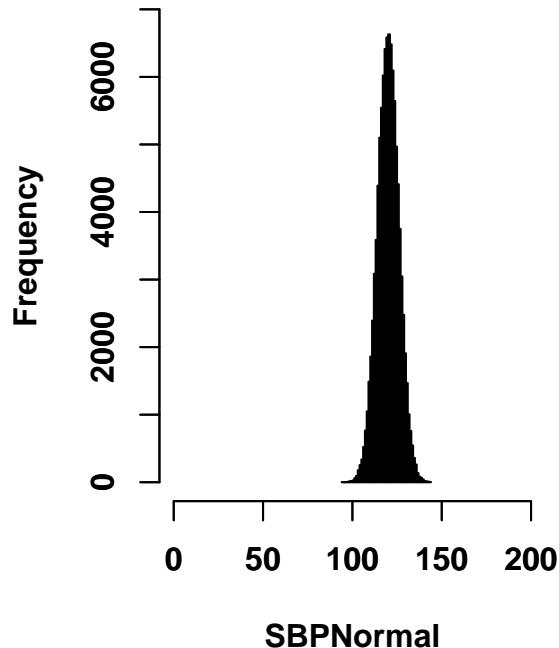
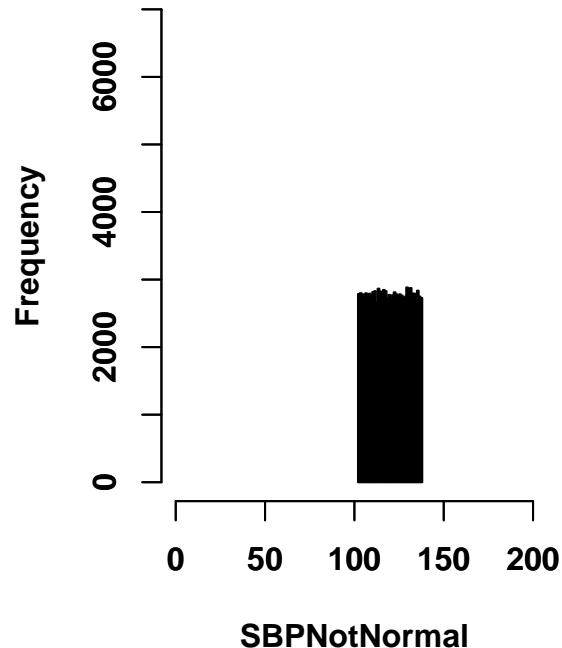


```
par(ask=TRUE)
par(mfrow=c(2,2))
graphics::hist(SBPNotNormal)
graphics::plot(stats::density(SBPNotNormal, na.rm=TRUE))
graphics::boxplot(SBPNotNormal)
stats::qqnorm(SBPNotNormal); stats::qqline(SBPNotNormal)
```



Histogram check

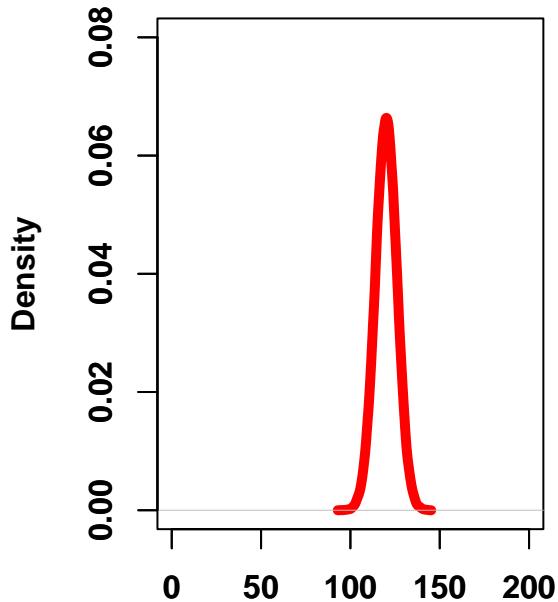
```
par.ask=TRUE)
par(mfrow=c(1,2))
graphics::hist(SBPNormal,
               main="SBP-Normal Distribution",
               col="red",
               breaks=50,
               font.lab=2,
               xlim=c(0,200),
               ylim=c(0,7000))
axis(side=1, font=2)
axis(side = 2, font=2)
graphics::hist(SBPNotNormal,
               main="SBP - Not Normal Distribution",
               col="red",# Add color
               breaks=50,
               font.lab=2,# Increase granularity of histogram # Bold labels
               xlim=c(0,200), # X axis scale
               ylim=c(0,7000)) # Y axis scale
axis(side=1, font=2) # X axis bold
axis(side=2, font=2) # Y axis bold
```

SBP–Normal Distribution**SBP – Not Normal Distribution**

Notice how both histograms have the same X axis scale # and Y axis scale, allowing meaningful side-by

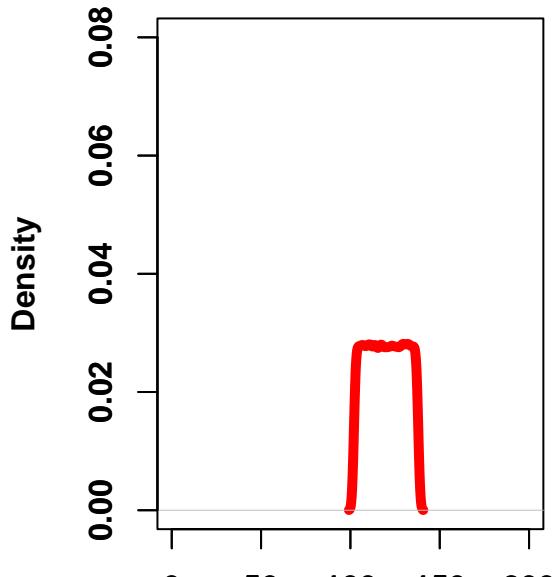
```
# Density Plot par(ask=TRUE)
par(mfrow=c(1,2)) # 2 figures - 1 row by 2 column grid
graphics::plot(stats::density(SBPNormal, na.rm=TRUE),
              main="SBP - Normal Distribution",
              col="red", # Add color
              lwd=5, # Thick line
              font.lab=2,
              xlim=c(0,200),
              ylim=c(0,0.08))
              axis(side=1, font=2)
              axis(side=2, font=2)
graphics::plot(stats::density(SBPNotNormal, na.rm=TRUE),
              main="SBP - Not Normal Distribution", col="red",# Add color
              lwd=5,
              font.lab=2,
              xlim=c(0,200),
              ylim=c(0,0.08))
axis(side=1, font=2) # X axis bold
axis(side=2, font=2)
```

SBP – Normal Distribution



N = 100000 Bandwidth = 0.5406

SBP – Not Normal Distribution

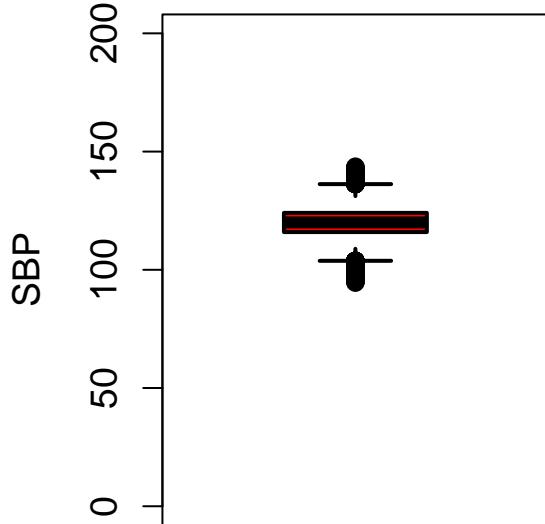


N = 100000 Bandwidth = 0.9356

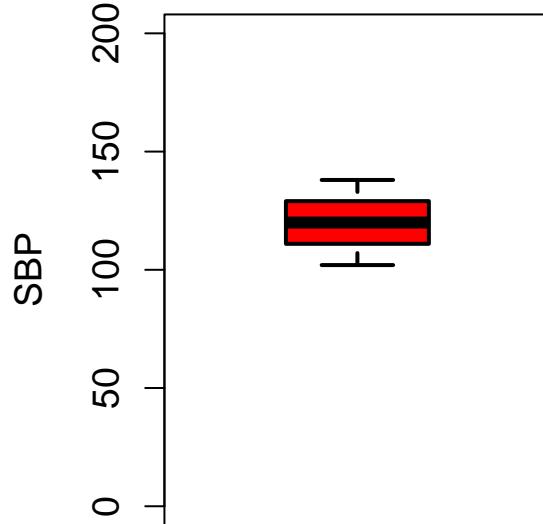
```
# Boxplot
par(ask=TRUE) # Pause
par(mfrow=c(1,2)) # 2 figures - 1 row by 2 column grid
graphics::boxplot(SBPNormal,
                   main="SBP - Normal Distribution",
                   xlab="Boxplot",
                   ylab="SBP",cex.axis=1.15,
                   cex.lab=1.15,
                   col="red",
                   lwd=2,
                   font.lab=2,
                   font=2,
                   ylim=c(0,200))

graphics::boxplot(SBPNotNormal,
                   main="SBP - Not Normal Distribution",
                   xlab="Boxplot",
                   ylab="SBP",
                   cex.axis=1.15,
                   cex.lab=1.15,
                   col="red",
                   lwd=2,
                   font.lab=2, font=2,
                   ylim=c(0,200))
```

SBP – Normal Distribution



SBP – Not Normal Distribution



Boxplot

Boxplot

QQ Plot

```
# Q-Q Plot
par(ask=TRUE) # Pause
par(mfrow=c(1,2)) # 2 figures - 1 row by 2 column grid
stats::qqnorm(SBPNormal,
              main="Q-Q Plot (Blue) and Q-Q Line (Red) of SBP Normal Distribution",
              col="blue",
              xlim=c(-4,4),
              ylim=c(0,200),
              font.axis=2,
              font.lab=2)
stats::qqline(SBPNormal, # Add a Q-Q Line to the Q-Q Plot
              col="red",
              lwd=4,
              lty=2)
stats::qqnorm(SBPNotNormal, main="Q-Q Plot (Blue) and Q-Q Line (Red) of SBP Not Normal Distribution",
              col="blue",
              xlim=c(-4,4),
              ylim=c(0,200),
              font.axis=2,
              font.lab=2)
stats::qqline(SBPNotNormal,#
              col="red",
              lwd=4,
              lty=2)
```

lue) and Q-Q Line (Red) of SBP Noe) and Q-Q Line (Red) of SBP Not N

