ANOVA test of the data

H NIU

July 21, 2021

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Patient characteristics were compared by mortality outcome. For categorical variable, frequenceies and column percentages (%) were reported and p-values were calculated using 2 test or Fisher's exact test when smallest expected frequencies are lower than 5. Continuous variables were tested for normality using both visual inspections (QQ-plots with histograms) and a formal inspection (Shapiro-Wilk test). Normally distributed continuous variables were reported using means and standard deviations (SD) and p-values were calculated using two-sample t-tests and on-way ANOVAs. Non-normally distributed variables were reported as medians and first/third quartiles, with p-values calculated using Mann–Whitney–Wilcoxon and Kruskal -Wallis tests.

The analyses were done in R programming languages, version 3.6.3. The code has been uploaded on the Github as well https://github.com/niuneo/Mortality-factors-analysis-of-perioperative-cardiac-arrest.

```
library(tidyselect)
```

```
## Warning: package 'tidyselect' was built under R version 3.6.3
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 3.6.3
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.3
```

```
## Loading required package: carData
## Warning: package 'carData' was built under R version 3.6.3
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

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

```
## The following object is masked from 'package:car':
##
##     recode
```

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

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

```r
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.6.3
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```r
getwd()
```

```
## [1] "C:/Users/neo/Huijie/Code/Predicting CA Mortality/Data"
```

```
setwd("C:\\Users\\neo\\Huijie\\Code\\Predicting CA Mortality\\Data\\")

data<-read.xlsx("150CA_EN20210701.xlsx",1)

data$Outcome <- ifelse(data$Died ==1, "Died", "Survived")
```

Part I: Categorical variables: test of independence?

```
#H0 : the variables are independent, there is no relationship between the two categorical variables.
#H1 : the variables are dependent, there is a relationship between the two categorical variables
#if the p-value is less than the significance level, we can reject the null hypothesis

print('Chi-square test')
```

```
## [1] "Chi-square test"
```

```
my_seq0 <- c(1,4:20)
# empty legend text
#legend_text <- c(data)

for (i0 in my_seq0) { # variables to compare
  test <- chisq.test(table(data[, i0], data$Outcome))
  test$statistic # test statistic
  test$p.value # p-value
# add next legend text
  print(colnames(data)[i0])
  print('statistic=')
  print(test$statistic) #  test statistic
  print('p-value=')
  print(test$p.value) #  p-value
}
```

```
## [1] "Gender"
## [1] "statistic="
## X-squared
##  5.753873
## [1] "p-value="
## [1] 0.01645235
```

```
## Warning in chisq.test(table(data[, i0], data$Outcome)): Chi-squared
## approximation may be incorrect
```

```
## [1] "Comorbidity_diabetes"
## [1] "statistic="
## X-squared
##  1.672186
## [1] "p-value="
## [1] 0.195966
## [1] "Comorbidity_hypertension"
## [1] "statistic="
## X-squared
##   2.11186
## [1] "p-value="
## [1] 0.1461615
## [1] "Comorbidity_CAD"
## [1] "statistic="
## X-squared
##         0
## [1] "p-value="
## [1] 1
## [1] "Comorbidity_pulmonary"
## [1] "statistic="
## X-squared
## 0.4885246
## [1] "p-value="
## [1] 0.4845862
## [1] "Comorbidity_hepatic"
## [1] "statistic="
## X-squared
##  1.084079
## [1] "p-value="
## [1] 0.2977868
## [1] "Comorbidity_renal"
## [1] "statistic="
##   X-squared
## 3.40322e-30
## [1] "p-value="
## [1] 1
## [1] "Comorbidity_neurological"
## [1] "statistic="
##  X-squared
## 0.05023611
## [1] "p-value="
## [1] 0.8226529
## [1] "Comorbidity_tumor"
## [1] "statistic="
## X-squared
##         0
## [1] "p-value="
## [1] 1
## [1] "Surgical_type"
## [1] "statistic="
## X-squared
##  11.68151
## [1] "p-value="
## [1] 0.01988368
## [1] "Emergency"
## [1] "statistic="
## X-squared
##  7.141225
## [1] "p-value="
## [1] 0.007533168
## [1] "Trauma"
## [1] "statistic="
## X-squared
## 0.8655984
## [1] "p-value="
## [1] 0.3521767
```

```
## Warning in chisq.test(table(data[, i0], data$Outcome)): Chi-squared
## approximation may be incorrect
```

```
## [1] "Anaesthetic_type"
## [1] "statistic="
##    X-squared
## 6.366345e-31
## [1] "p-value="
## [1] 1
```

```
## Warning in chisq.test(table(data[, i0], data$Outcome)): Chi-squared
## approximation may be incorrect
```

```
## [1] "Operative_position"
## [1] "statistic="
## X-squared
##  12.56193
## [1] "p-value="
## [1] 0.01362702
```

```
## Warning in chisq.test(table(data[, i0], data$Outcome)): Chi-squared
## approximation may be incorrect
```

```
## [1] "ASA_PS"
## [1] "statistic="
## X-squared
##  26.24712
## [1] "p-value="
## [1] 2.821336e-05
## [1] "Timing_arrest"
## [1] "statistic="
## X-squared
## 0.4172873
## [1] "p-value="
## [1] 0.9366486
## [1] "Defibrillate"
## [1] "statistic="
## X-squared
## 0.4935731
## [1] "p-value="
## [1] 0.4823378
## [1] "Cause_arrest"
## [1] "statistic="
## X-squared
##  4.643279
## [1] "p-value="
## [1] 0.1998607
```

```
#H0 : the variables are independent, there is no relationship between the two categorical variables.
#H1 : the variables are dependent, there is a relationship between the two categorical variables
#if the p-value is less than the significance level, we can reject the null hypothesis

# The Fisher’s exact test does not require the assumption of a minimum of 5 expected counts.
print('Fishers exact test')
```

```
## [1] "Fishers exact test"
```

```
my_seq0 <- c(1,4:20)
# empty legend text
#legend_text <- c(data)

for (i0 in my_seq0) { # variables to compare
  test <-fisher.test(table(data[, i0], data$Outcome))
  test$statistic # test statistic
  test$p.value # p-value
# add next legend text
  print(colnames(data)[i0])
  print('statistic=')
  print(test$statistic) #  test statistic
  print('p-value=')
  print(test$p.value) #  p-value
}
```

```
## [1] "Gender"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.01116523
## [1] "Comorbidity_diabetes"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.1863895
## [1] "Comorbidity_hypertension"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.1075058
## [1] "Comorbidity_CAD"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 1
## [1] "Comorbidity_pulmonary"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.4440633
## [1] "Comorbidity_hepatic"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.2613506
## [1] "Comorbidity_renal"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 1
## [1] "Comorbidity_neurological"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.691224
## [1] "Comorbidity_tumor"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 1
## [1] "Surgical_type"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.018742
## [1] "Emergency"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.005396785
## [1] "Trauma"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.3245961
## [1] "Anaesthetic_type"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 1
## [1] "Operative_position"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.00866603
## [1] "ASA_PS"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 8.060649e-06
## [1] "Timing_arrest"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.9561173
## [1] "Defibrillate"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.4151752
## [1] "Cause_arrest"
## [1] "statistic="
## NULL
## [1] "p-value="
## [1] 0.1956162
```

```
# 3. Combination of plot and statistical test
#mosaic() function from the {vcd} package.
#This function has the advantage that it combines a mosaic plot (to visualize a contingency ta
ble)
#and the result of the Chi-square test of independence

# library(vcd)
#
# mosaic(~ Gender + Outcome,
#        direction = c("v", "h"),
#        data = data,
#        shade = TRUE)
```

Part II: Continuous variables: a normal distribution?

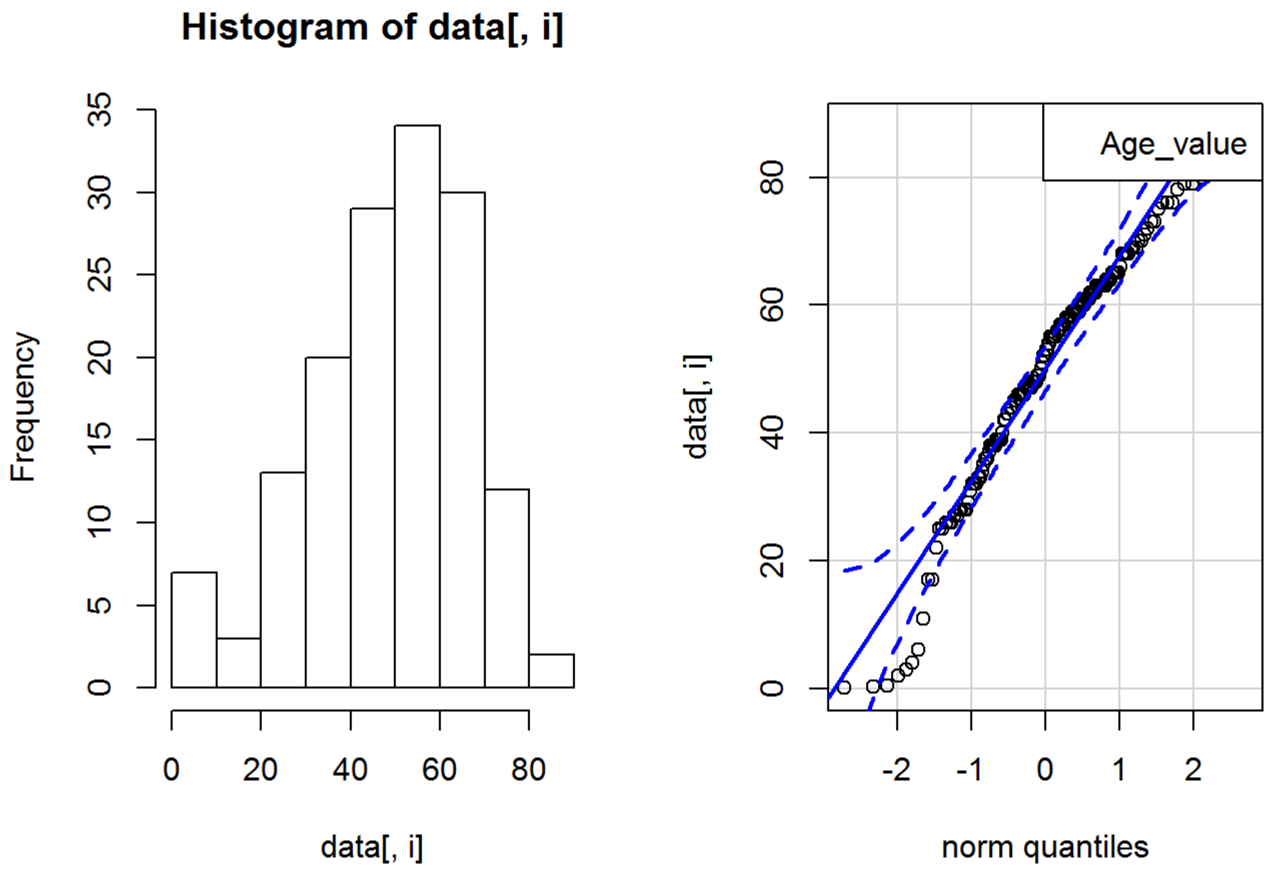
Inspection of the normality

```
#H0: the data follow a normal distribution; H1: the data do not follow a normal distribution
#the p-value>0.05 implys that we do not reject the null hypothesis
#In practice, normality tests are often considered as too conservative in the sense that for l
arge sample size (
#n> 50), a small deviation from the normality may cause the normality condition to be violated

print('in regard to total patients')
```

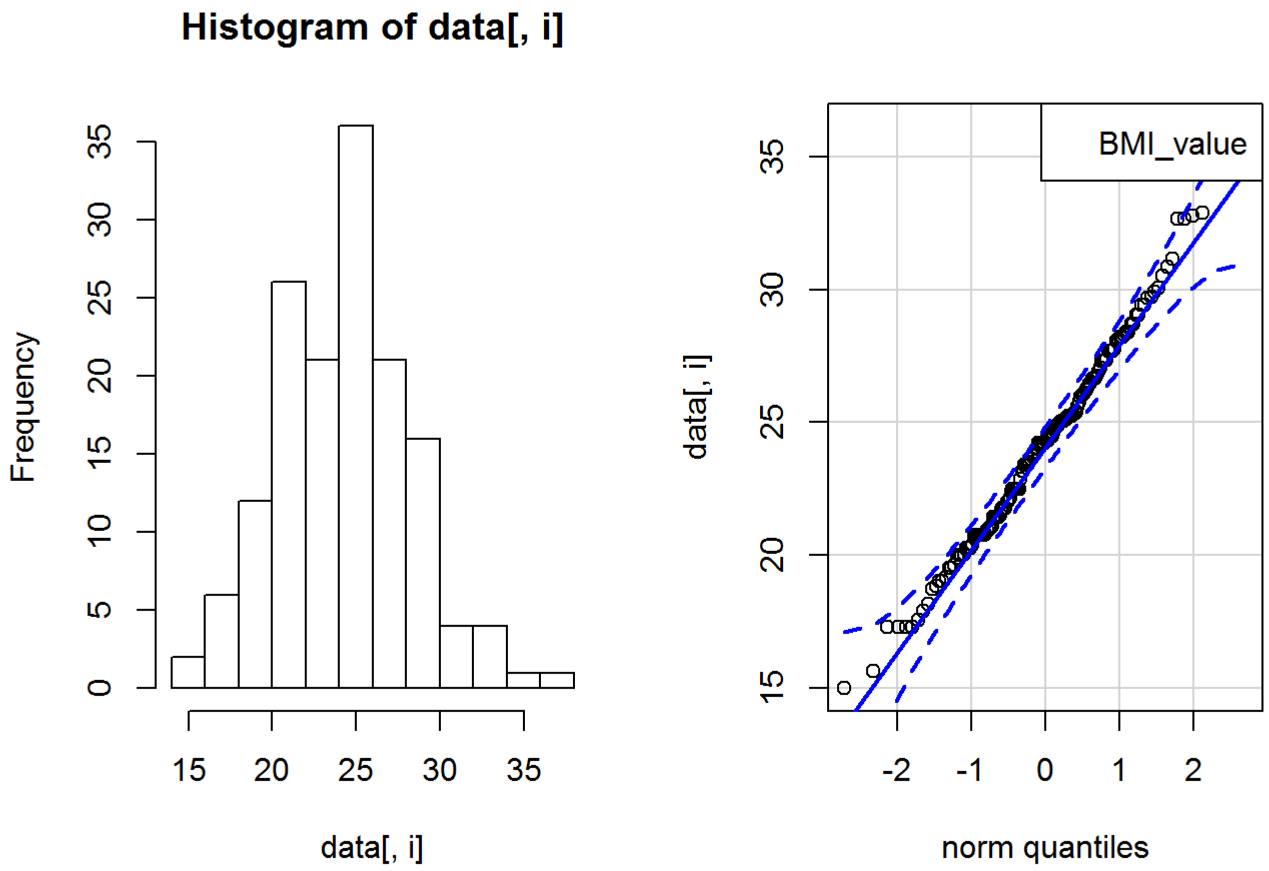
```
## [1] "in regard to total patients"
```

```
my_seq <- c(2,3,21:28)
# empty legend text
```

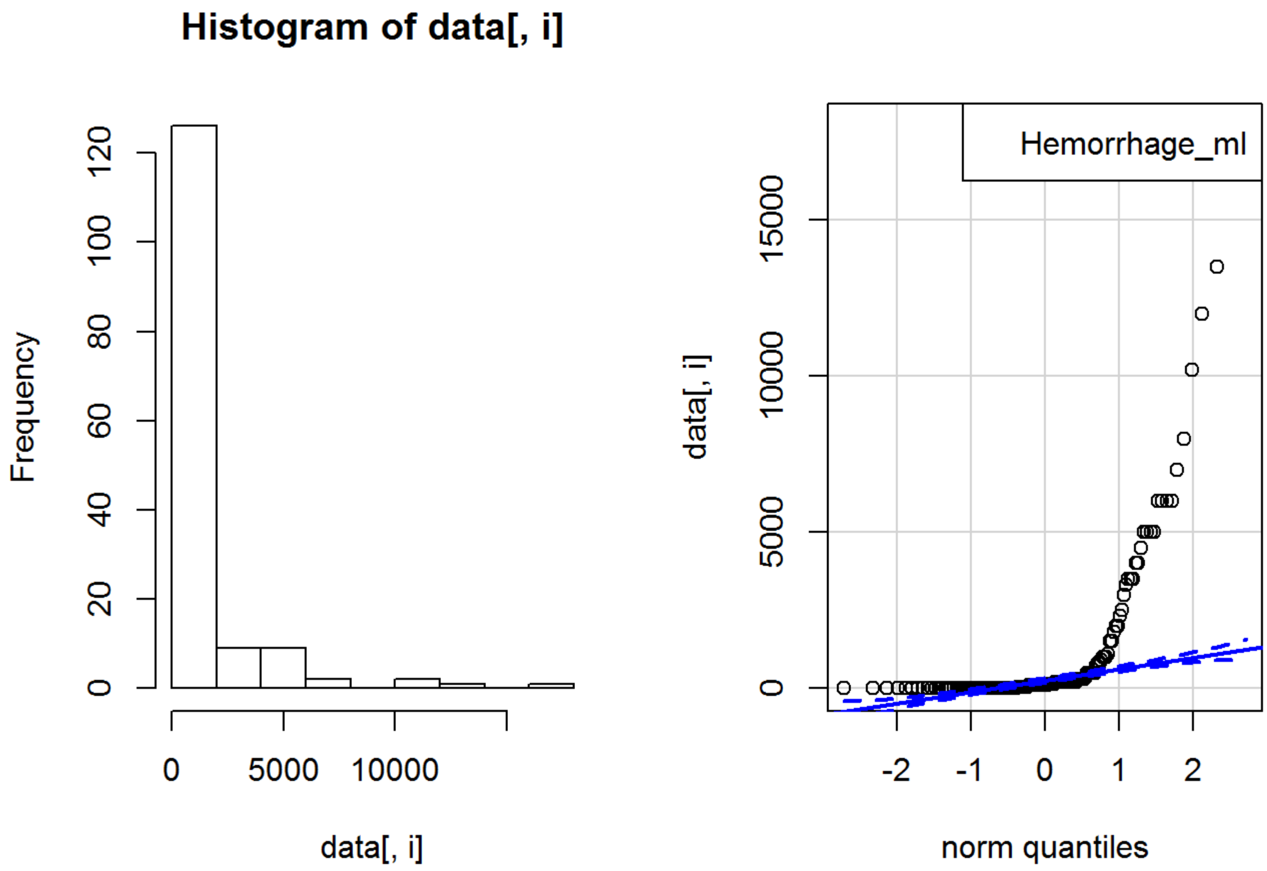
```
#legend_text <- c(data)

for (i in my_seq) { # variables to compare
  par(mfrow = c(1, 2)) # combine plots
  hist(data[, i])
  qqPlot(data[, i],id = FALSE # id = FALSE to remove point identification
       )
  # add next legend text
  legend_text <- colnames(data)[i]
  # plot legend once
  legend('topright', legend = legend_text)
  print(colnames(data)[i])
  print(shapiro.test(data[, i])) # print results of Shapiro-Wilk test
}
```



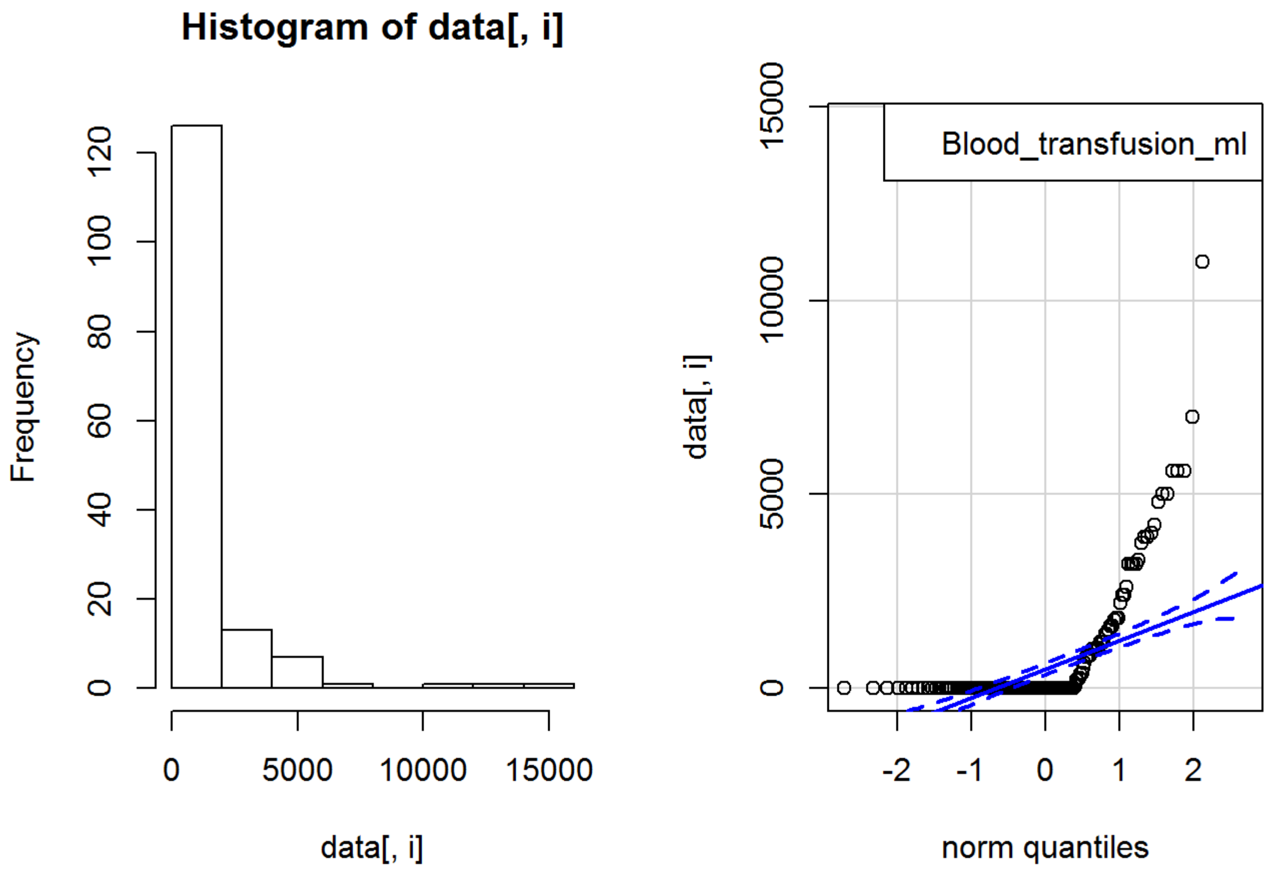
```
## [1] "Age_value"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.96325, p-value = 0.000494
```



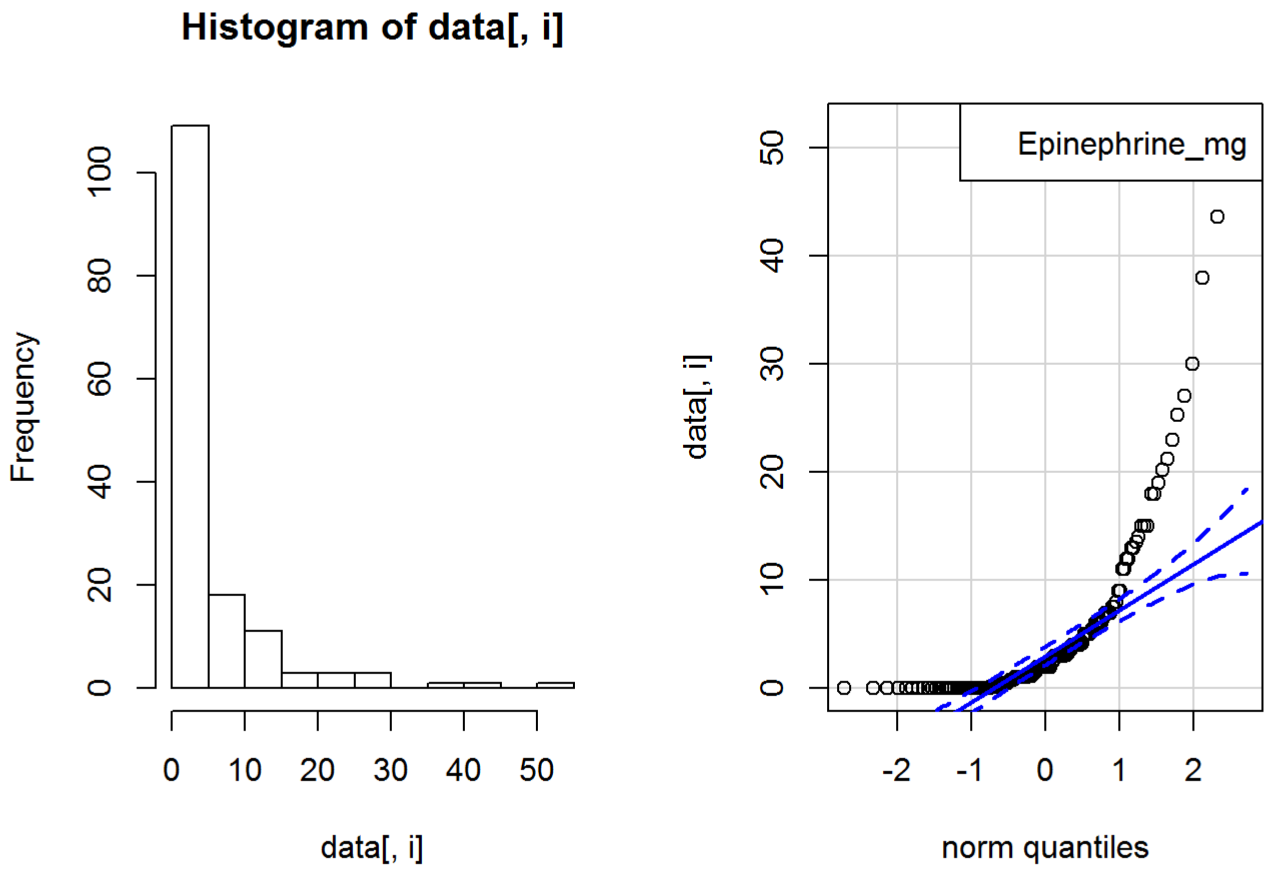
```
## [1] "BMI_value"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.99149, p-value = 0.5088
```



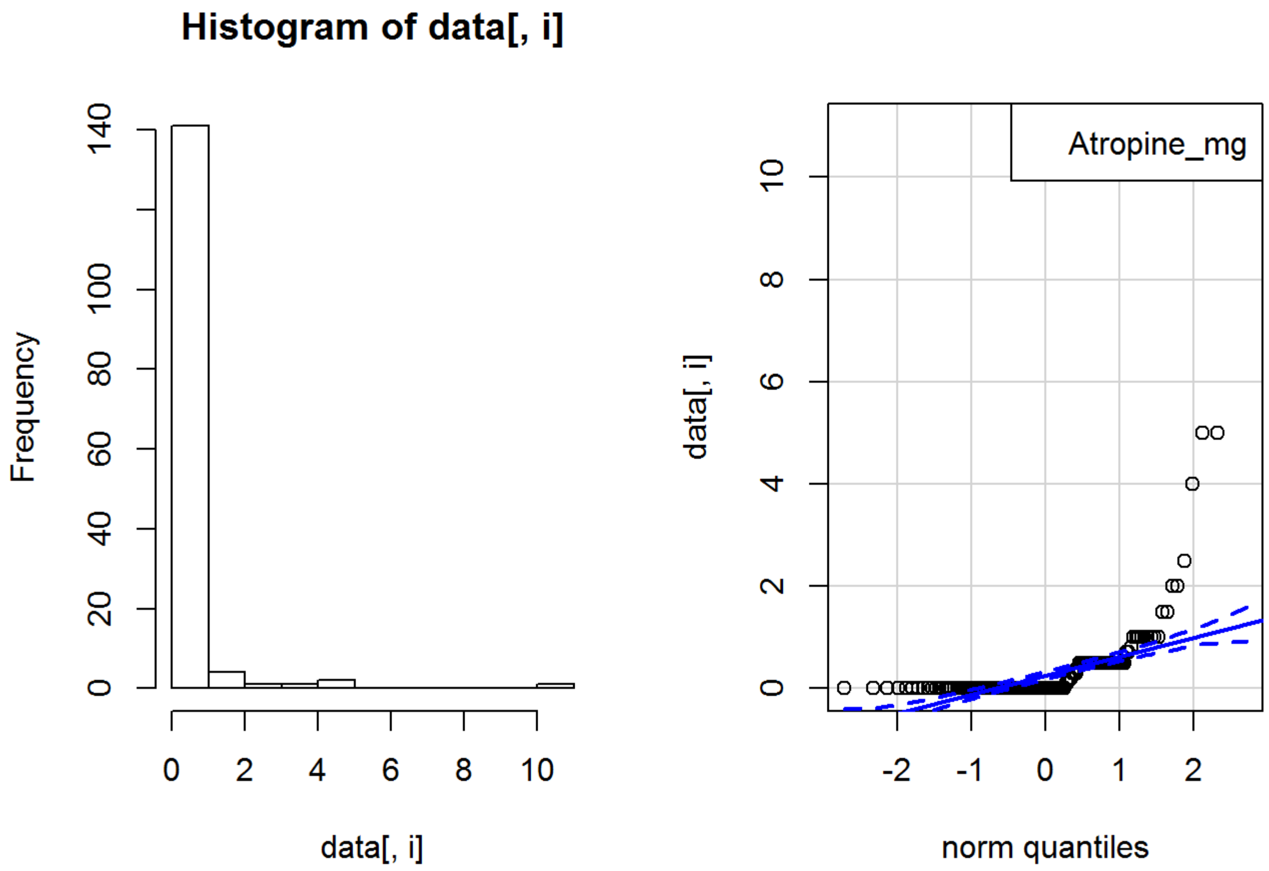
```
## [1] "Hemorrhage_ml"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.4961, p-value < 2.2e-16
```



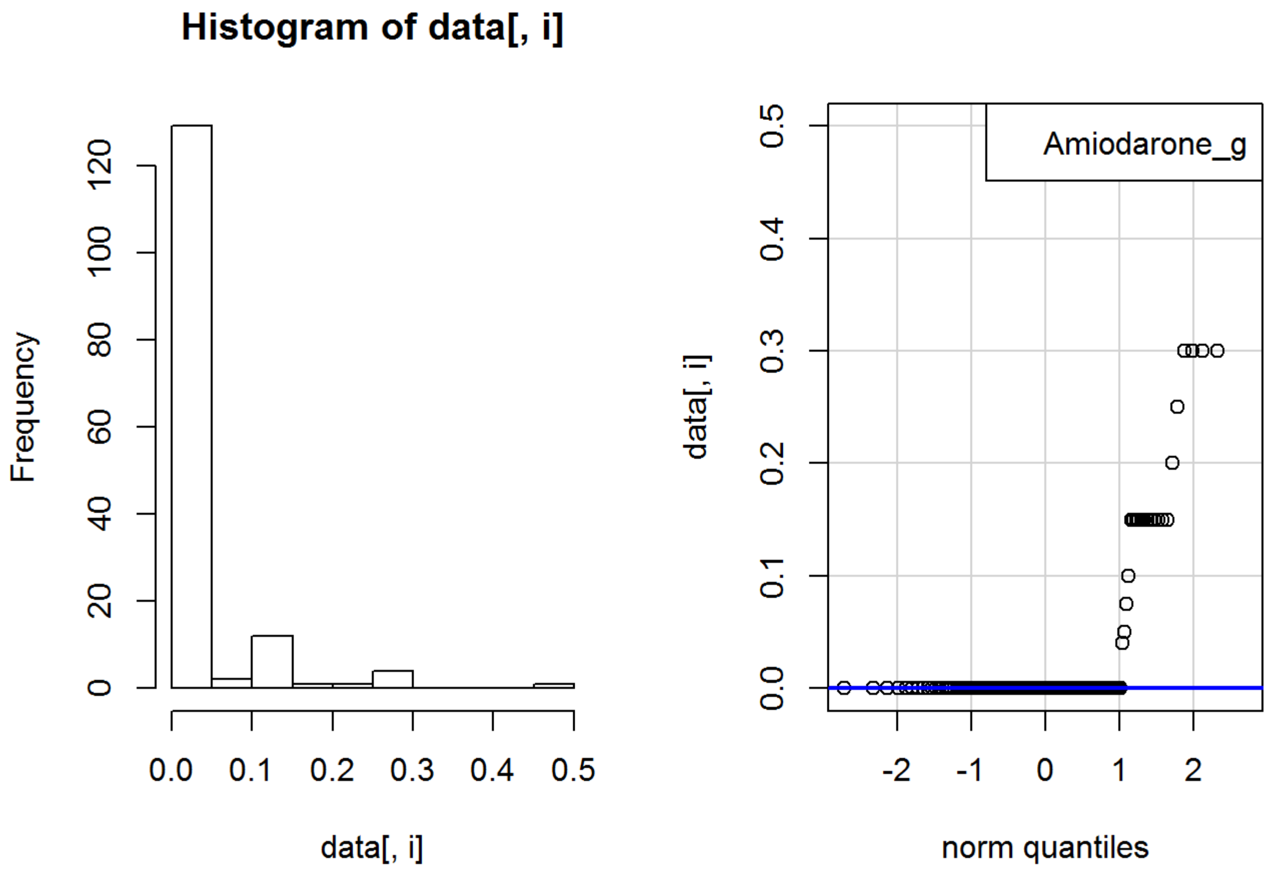
```
## [1] "Blood_transfusion_ml"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.50466, p-value < 2.2e-16
```



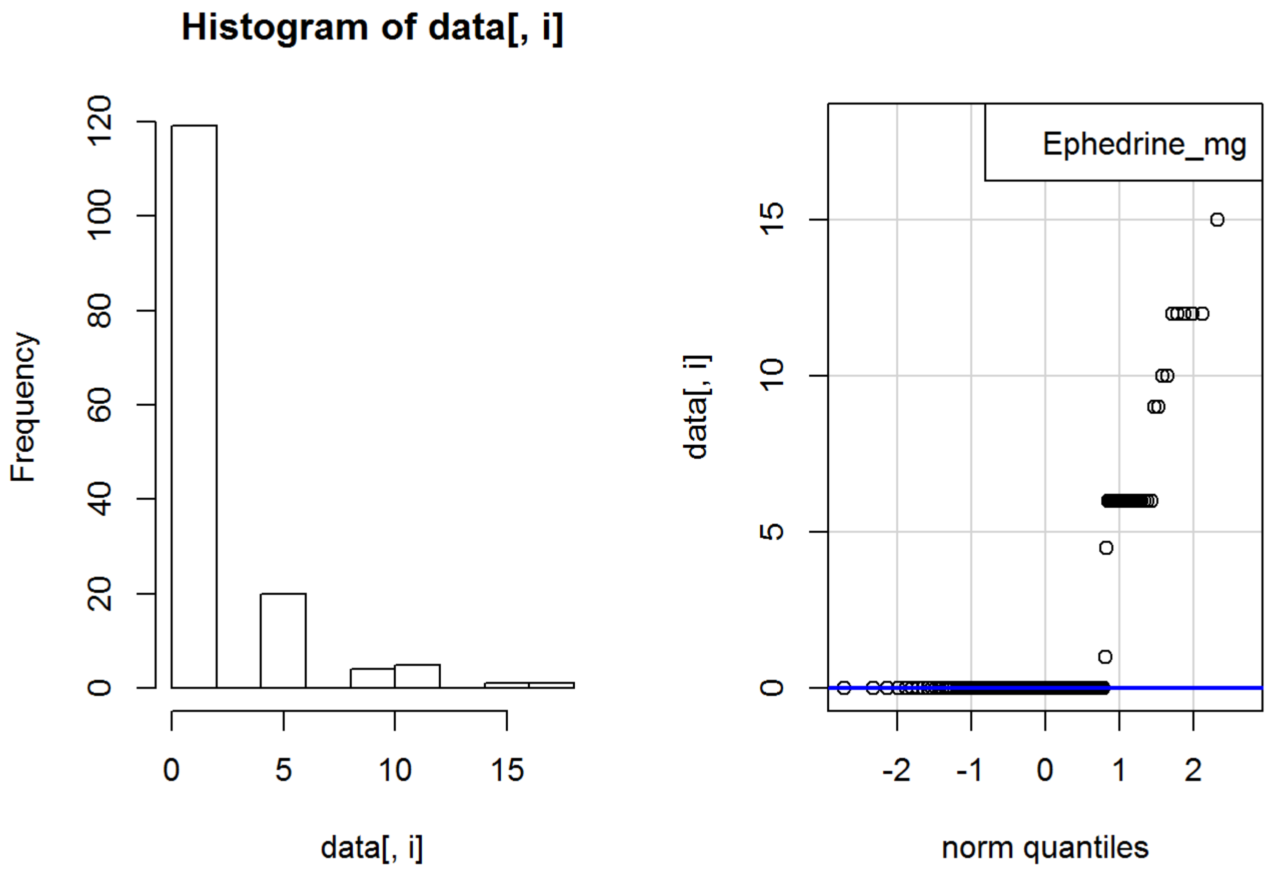
```
## [1] "Epinephrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.62859, p-value < 2.2e-16
```



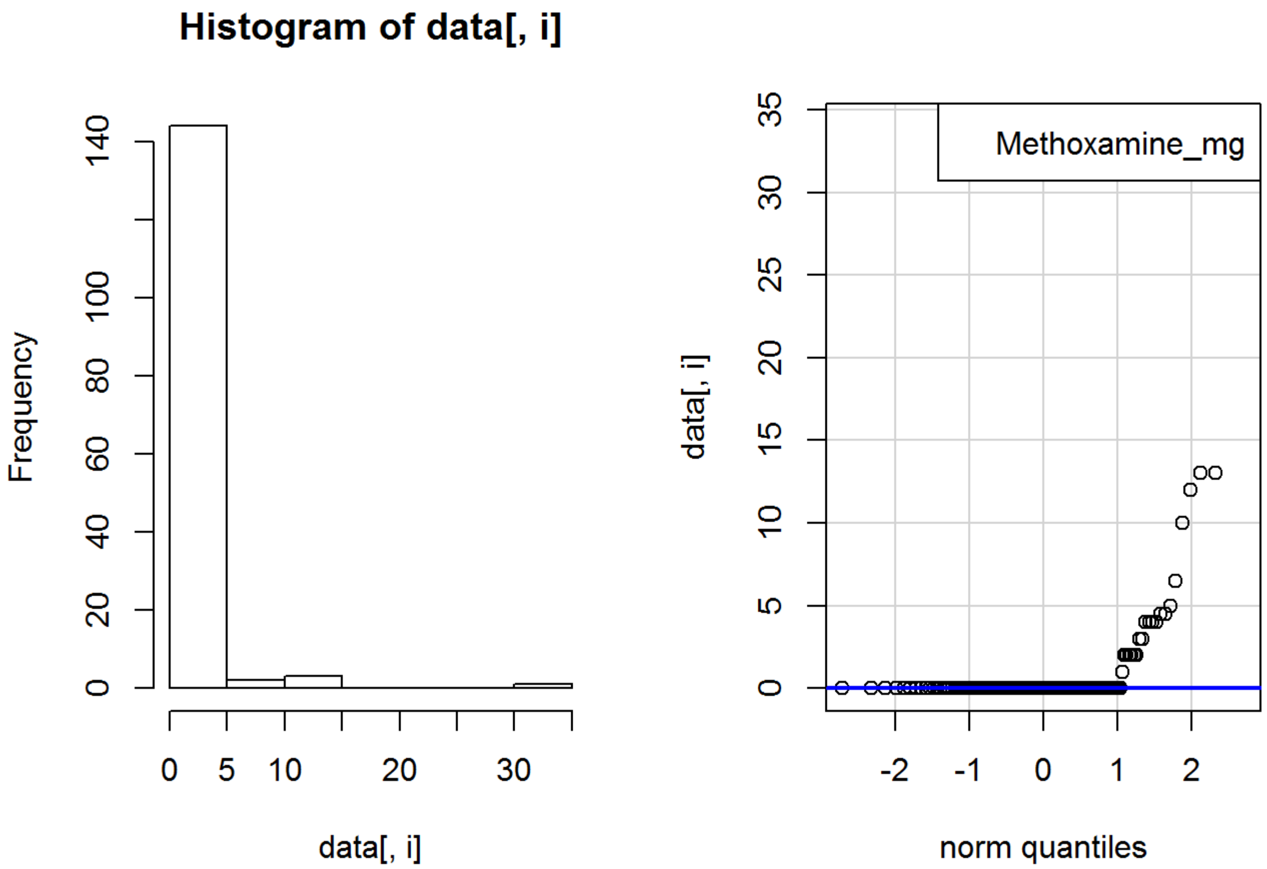
```
## [1] "Atropine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.36384, p-value < 2.2e-16
```



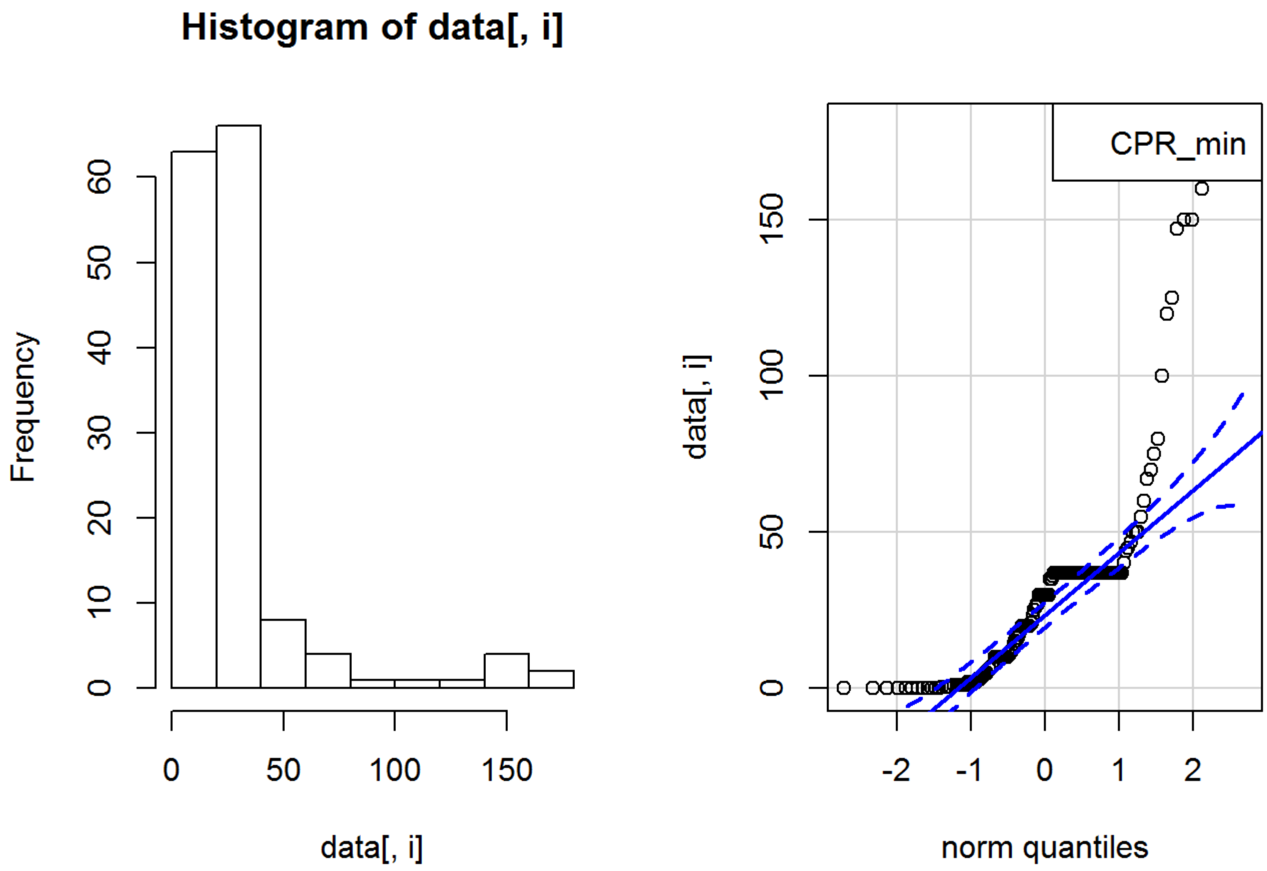
```
## [1] "Amiodarone_g"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.42146, p-value < 2.2e-16
```



```
## [1] "Ephedrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.52993, p-value < 2.2e-16
```



```
## [1] "Methoxamine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.28342, p-value < 2.2e-16
```

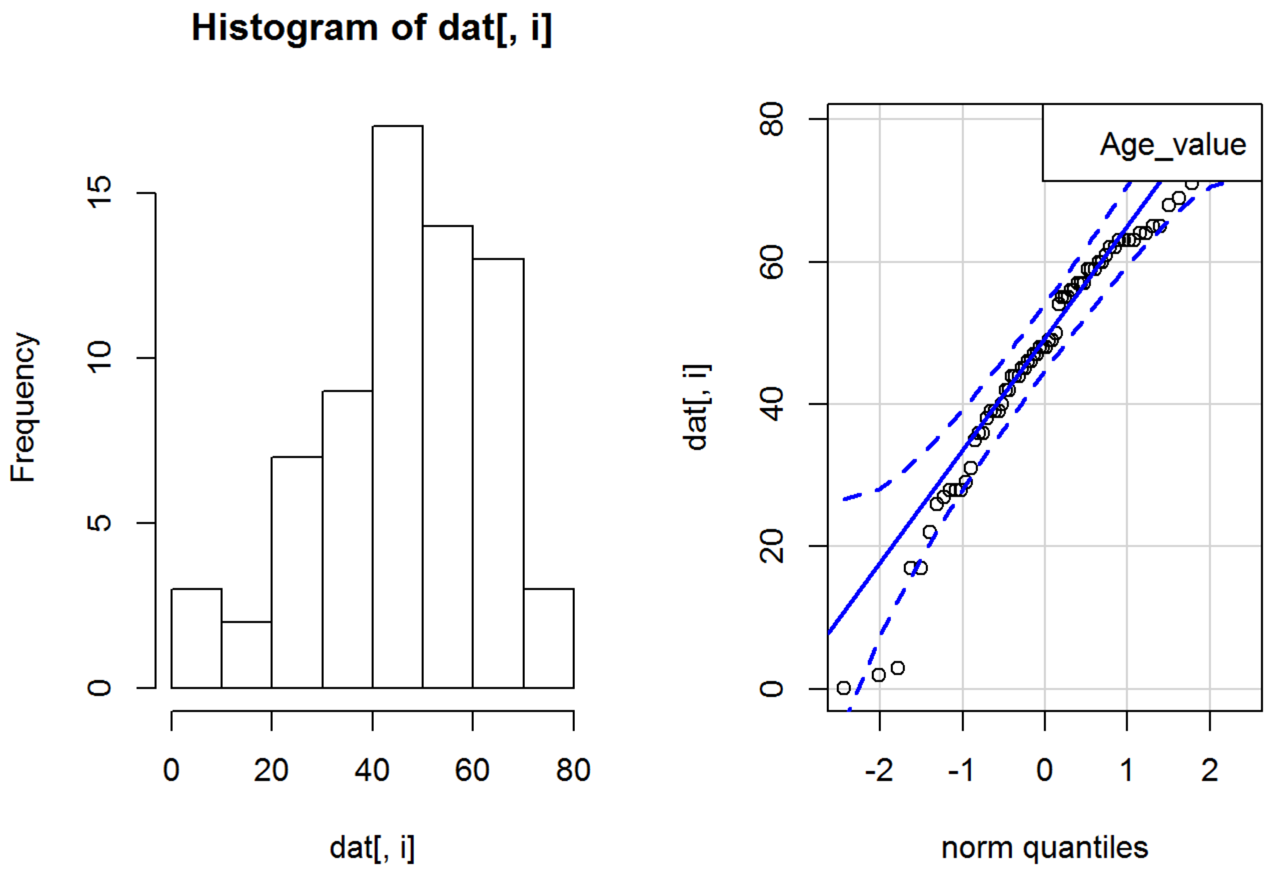


```
## [1] "CPR_min"
##
##  Shapiro-Wilk normality test
##
## data:  data[, i]
## W = 0.71345, p-value = 9.317e-16
```

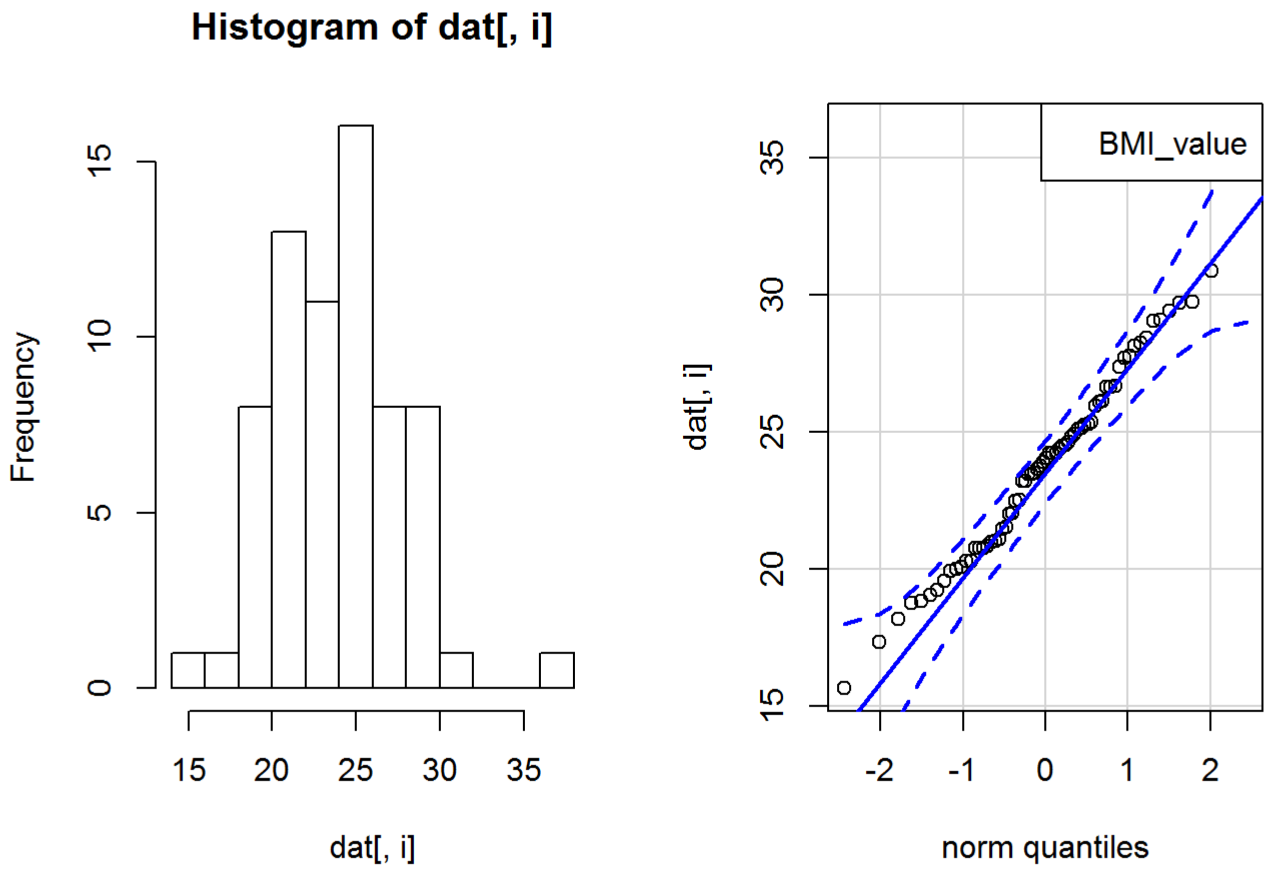
```
#H0: the data follow a normal distribution; H1: the data do not follow a normal distribution
#the p-value>0.05 implys that we do not reject the null hypothesis
#In practice, normality tests are often considered as too conservative in the sense that for l
arge sample size (
#n> 50), a small deviation from the normality may cause the normality condition to be violated

print('in regard to outcome status=Survived')
```

```
## [1] "in regard to outcome status=Survived"
```

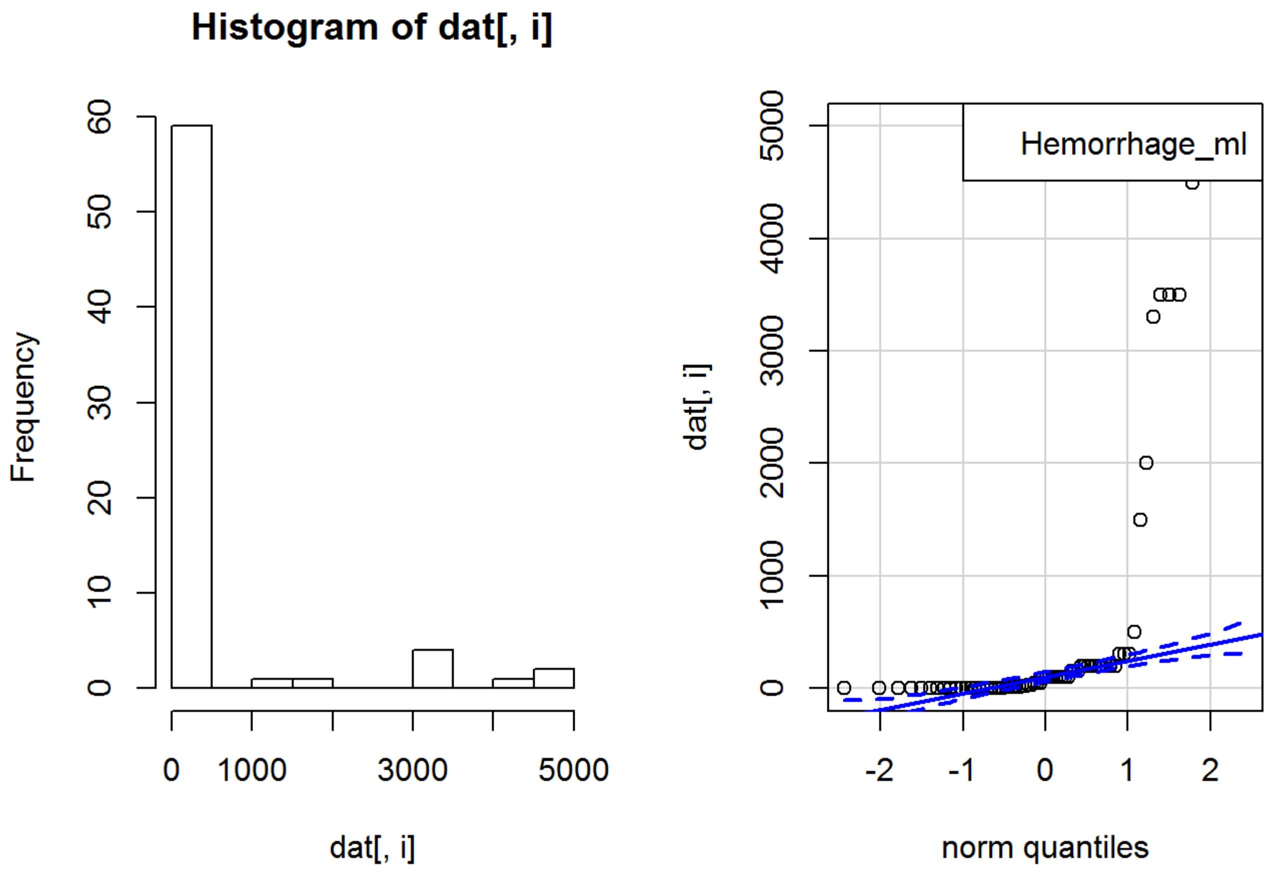
```
# empty legend text
#legend_text <- c(data)

for (i in my_seq) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Survived')
    par(mfrow = c(1, 2)) # combine plots
    hist(dat[, i])
    qqPlot(dat[, i],id = FALSE # id = FALSE to remove point identification
       ) # add next legend text
    legend_text <- colnames(dat)[i]  # plot legend once
      legend('topright', legend = legend_text)
      print(colnames(dat)[i])
      print(shapiro.test(dat[, i])) # print results of Shapiro-Wilk test
}
```



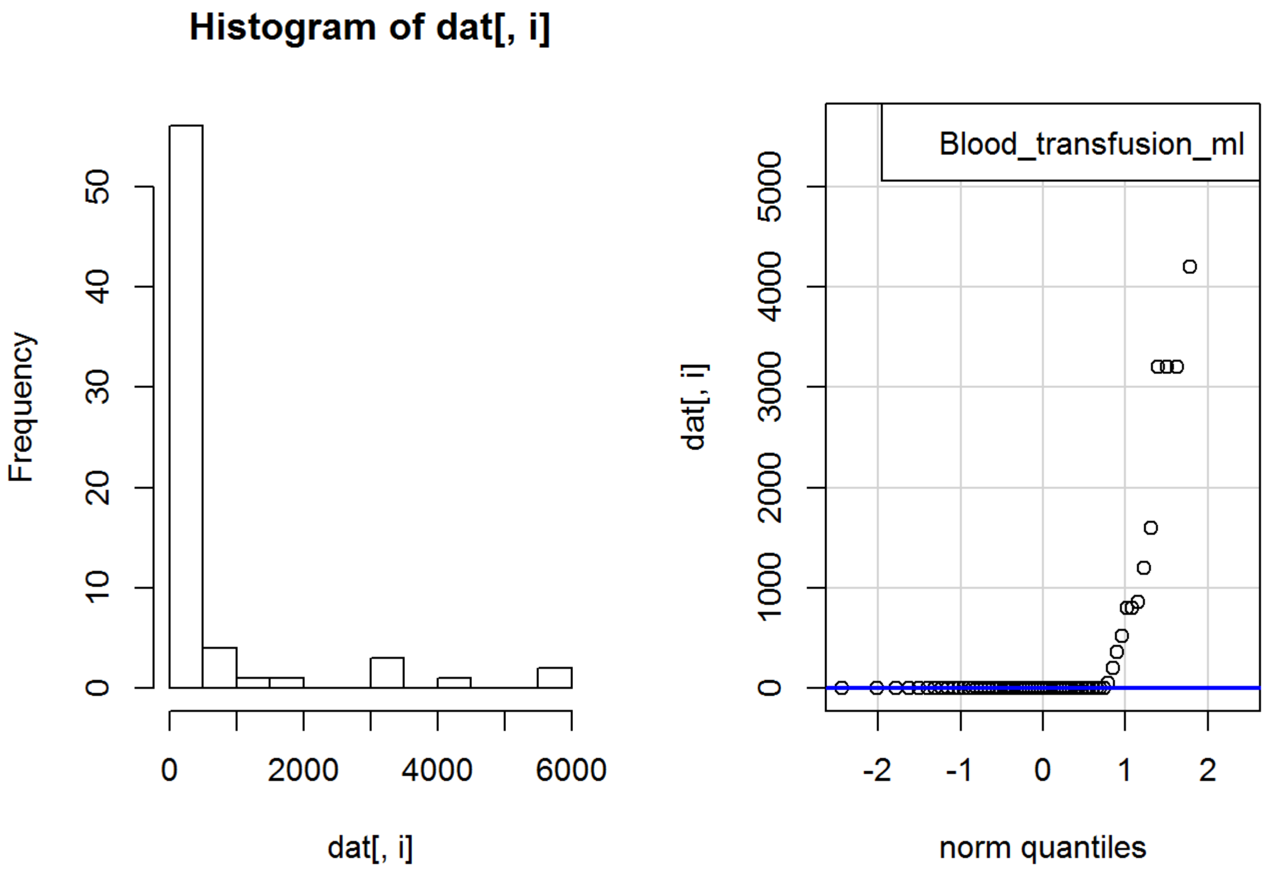
```
## [1] "Age_value"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.9522, p-value = 0.01095
```



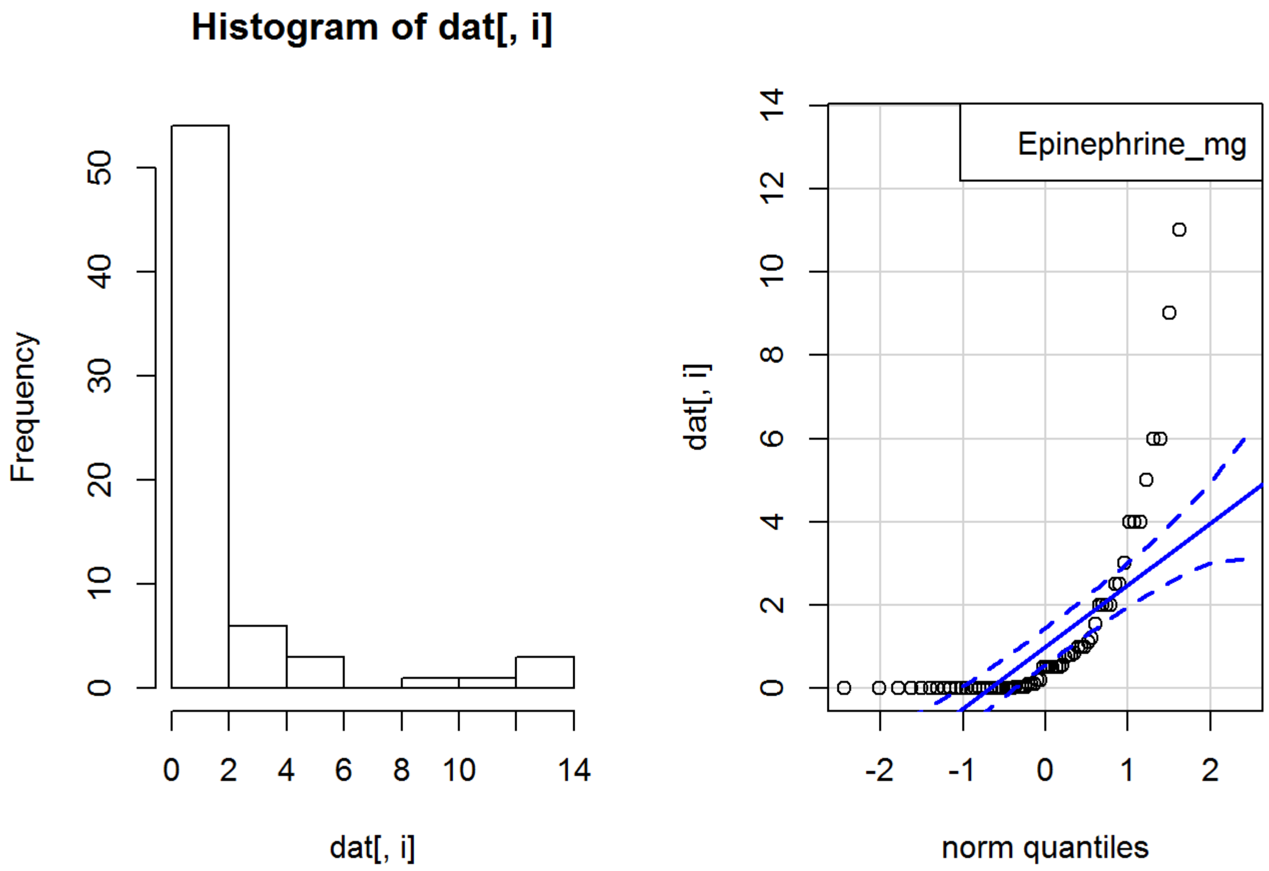
```
## [1] "BMI_value"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.98134, p-value = 0.4014
```



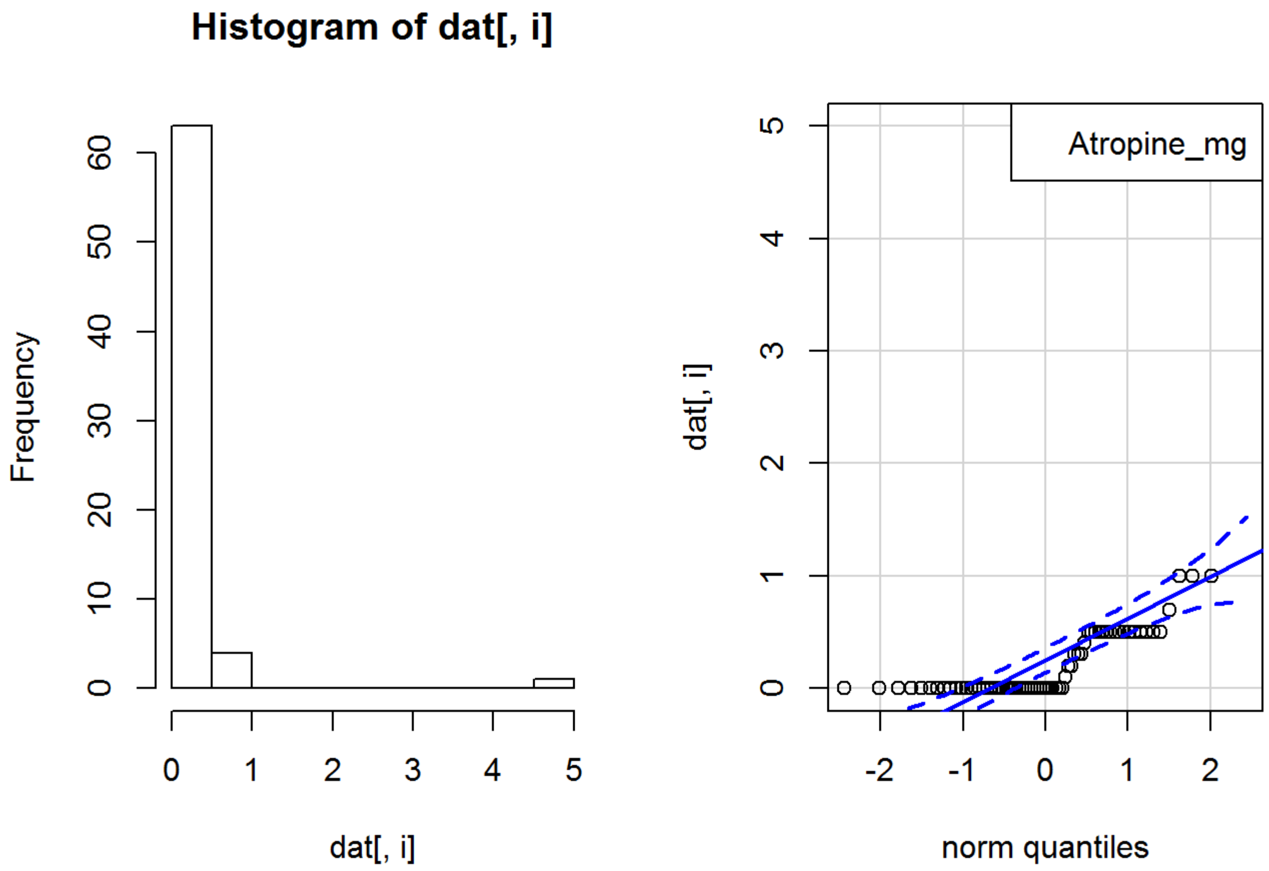
```
## [1] "Hemorrhage_ml"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.47292, p-value = 3.118e-14
```



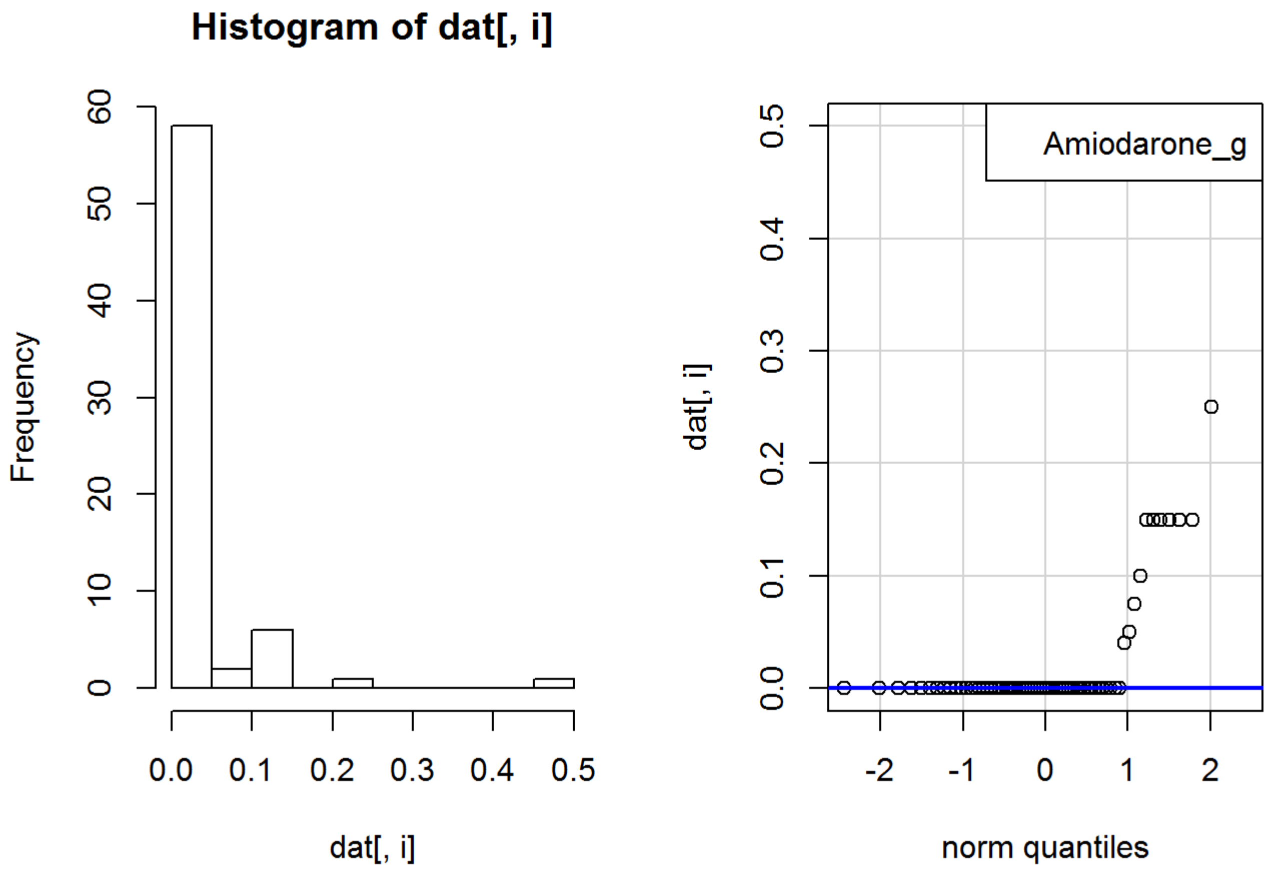
```
## [1] "Blood_transfusion_ml"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.43084, p-value = 8.614e-15
```



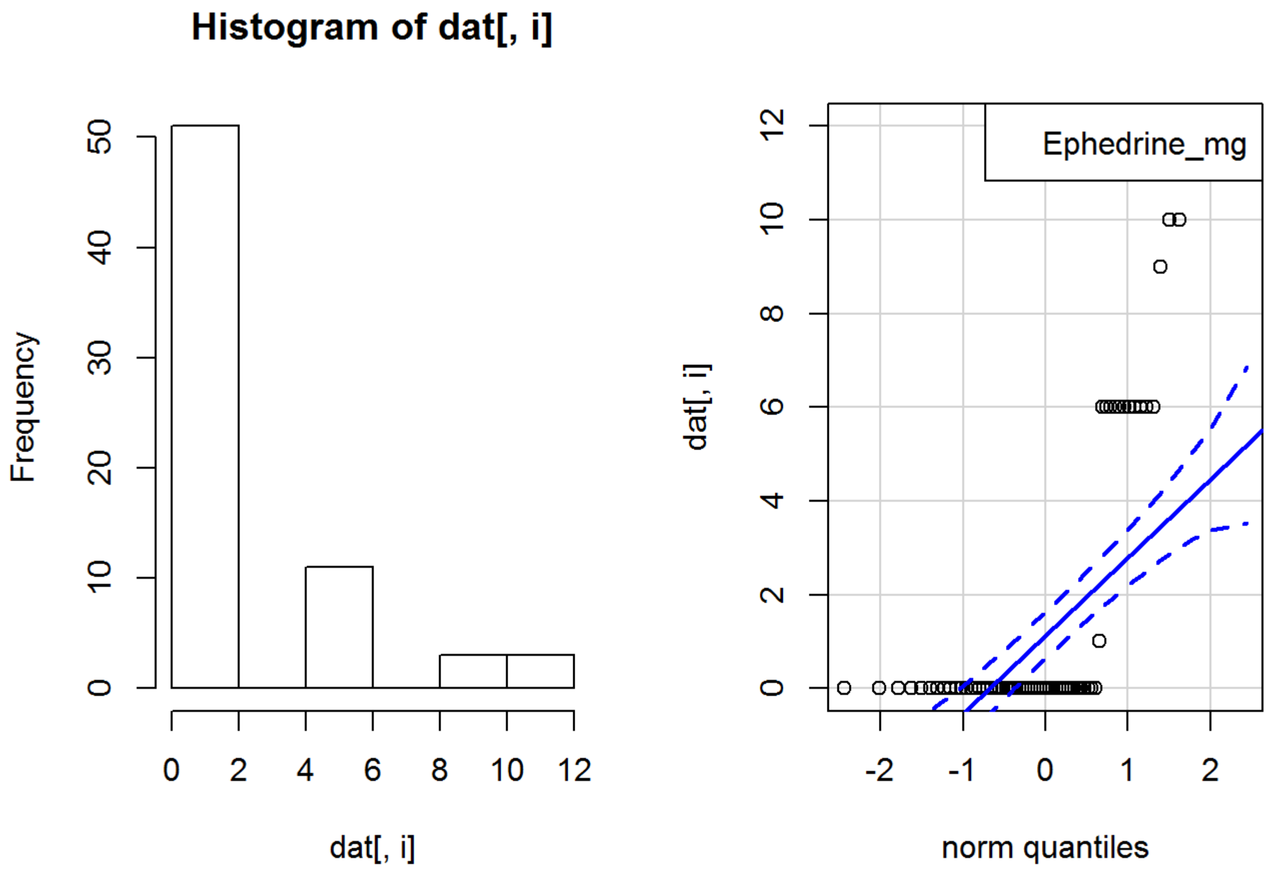
```
## [1] "Epinephrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.58914, p-value = 1.675e-12
```



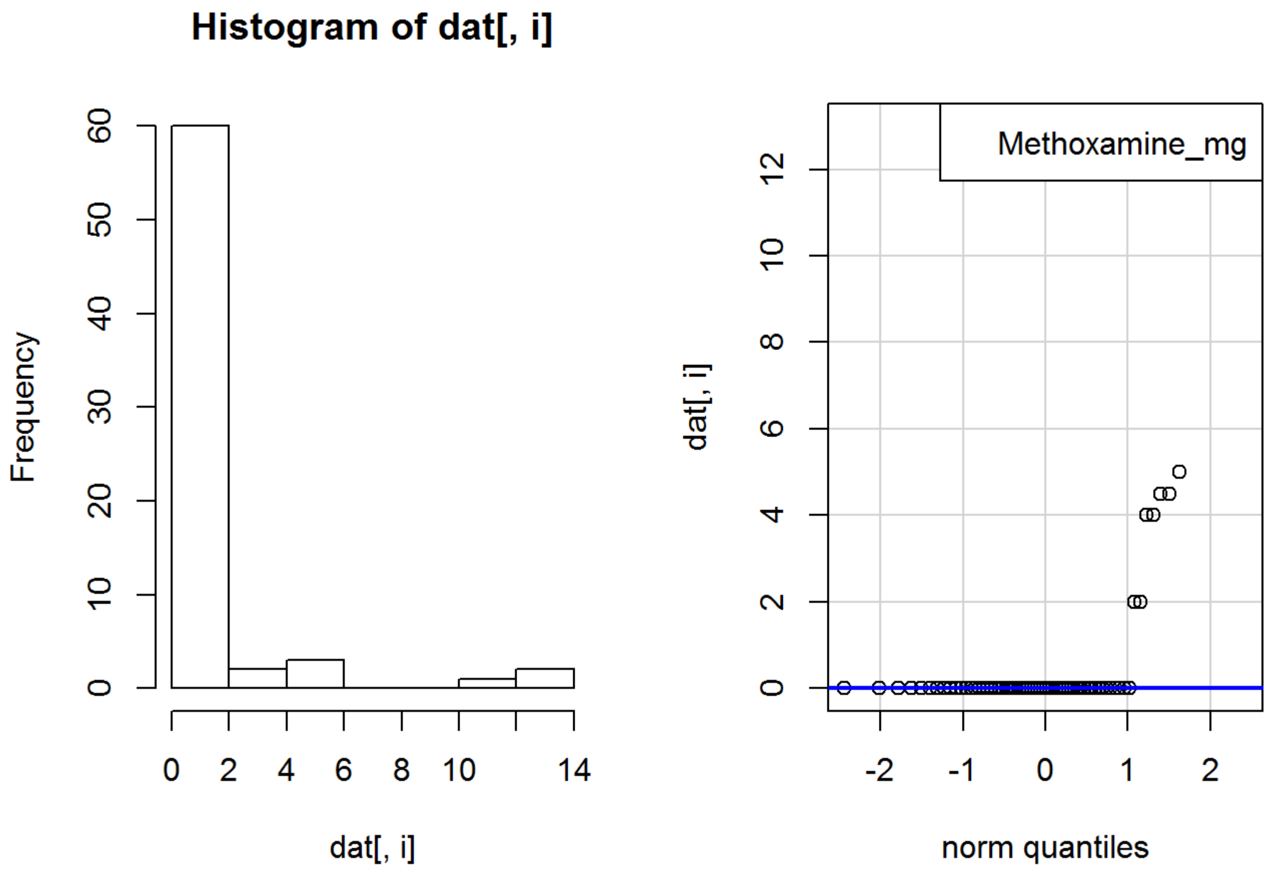
```
## [1] "Atropine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.39284, p-value = 2.857e-15
```



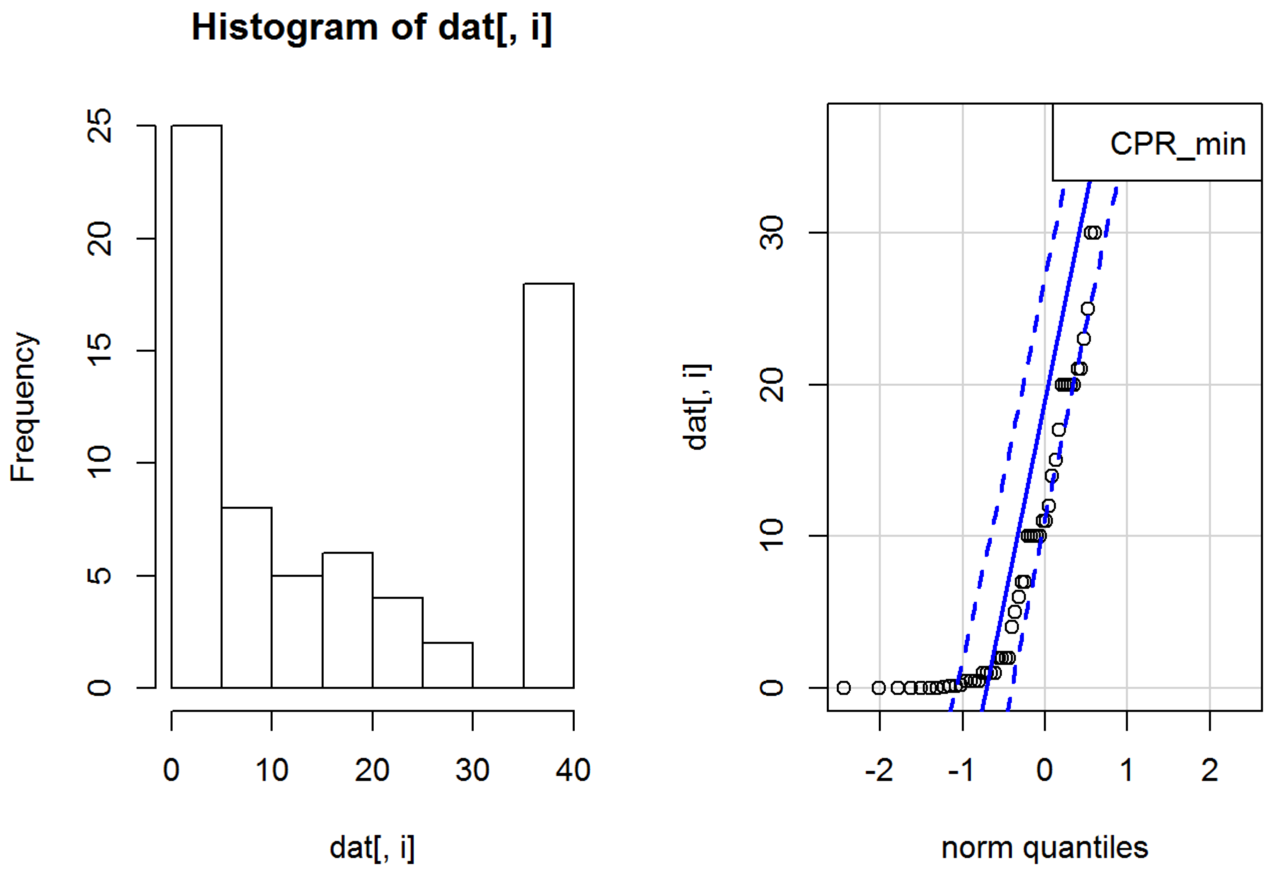
```
## [1] "Amiodarone_g"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.41307, p-value = 5.109e-15
```



```
## [1] "Ephedrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.59119, p-value = 1.809e-12
```



```
## [1] "Methoxamine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.38111, p-value = 2.053e-15
```

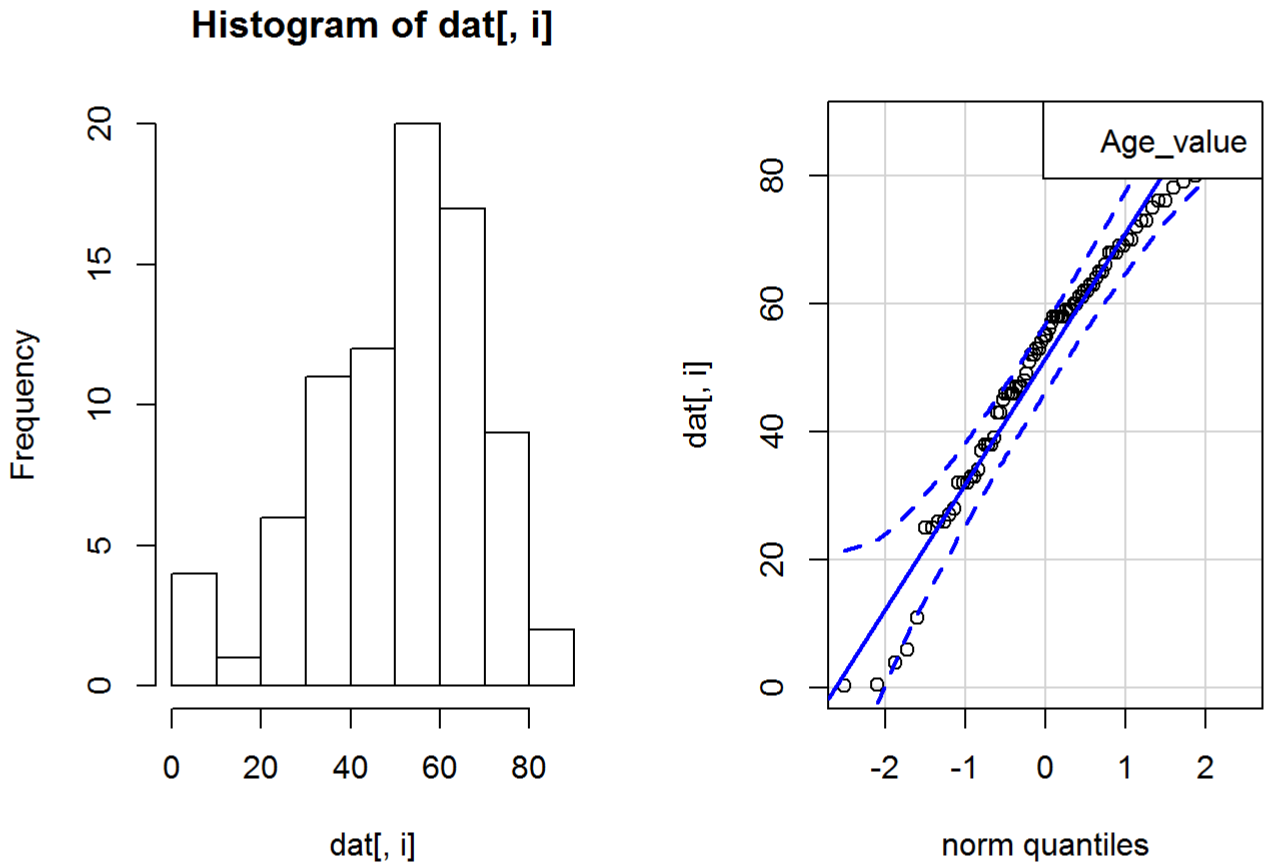


```
## [1] "CPR_min"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.82303, p-value = 1.421e-07
```

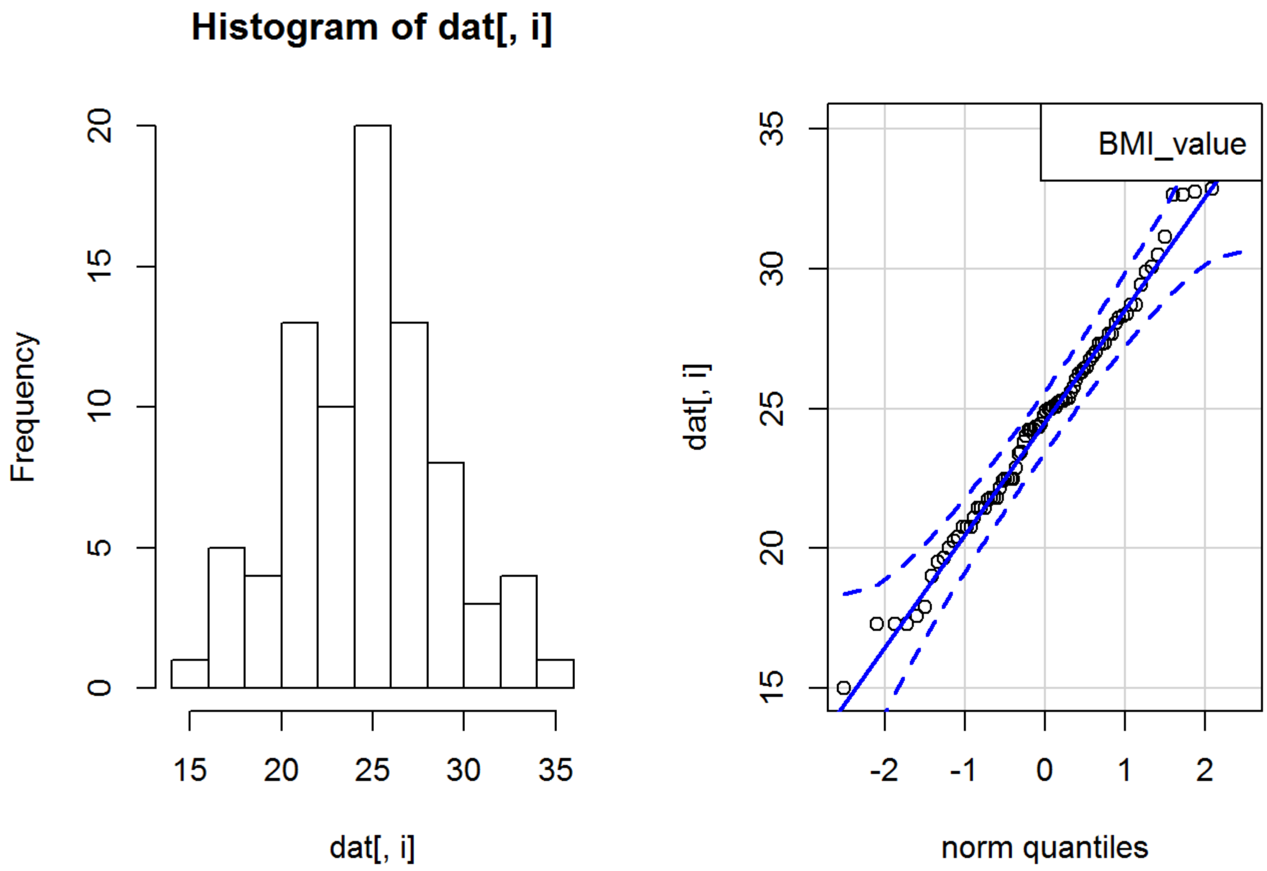
```
#H0: the data follow a normal distribution; H1: the data do not follow a normal distribution
#the p-value>0.05 implys that we do not reject the null hypothesis
#In practice, normality tests are often considered as too conservative in the sense that for l
arge sample size (
#n> 50), a small deviation from the normality may cause the normality condition to be violated


print('in regard to outcome status=Died')
```

```
## [1] "in regard to outcome status=Died"
```

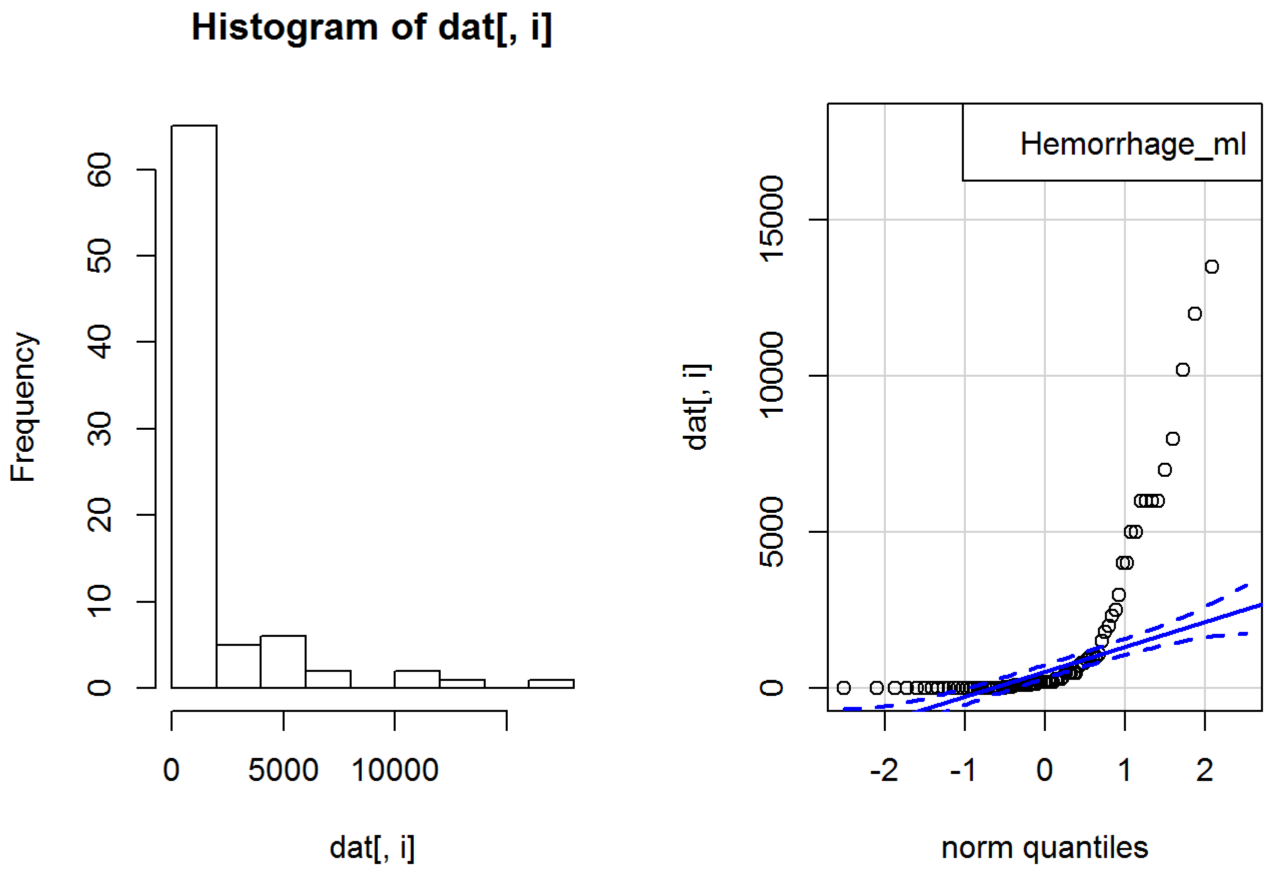
```
# empty legend text
#legend_text <- c(data)

for (i in my_seq) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Died')
    par(mfrow = c(1, 2)) # combine plots
    hist(dat[, i])
    qqPlot(dat[, i],id = FALSE # id = FALSE to remove point identification
       ) # add next legend text
    legend_text <- colnames(dat)[i]  # plot legend once
      legend('topright', legend = legend_text)
      print(colnames(dat)[i])
      print(shapiro.test(dat[, i])) # print results of Shapiro-Wilk test
}
```



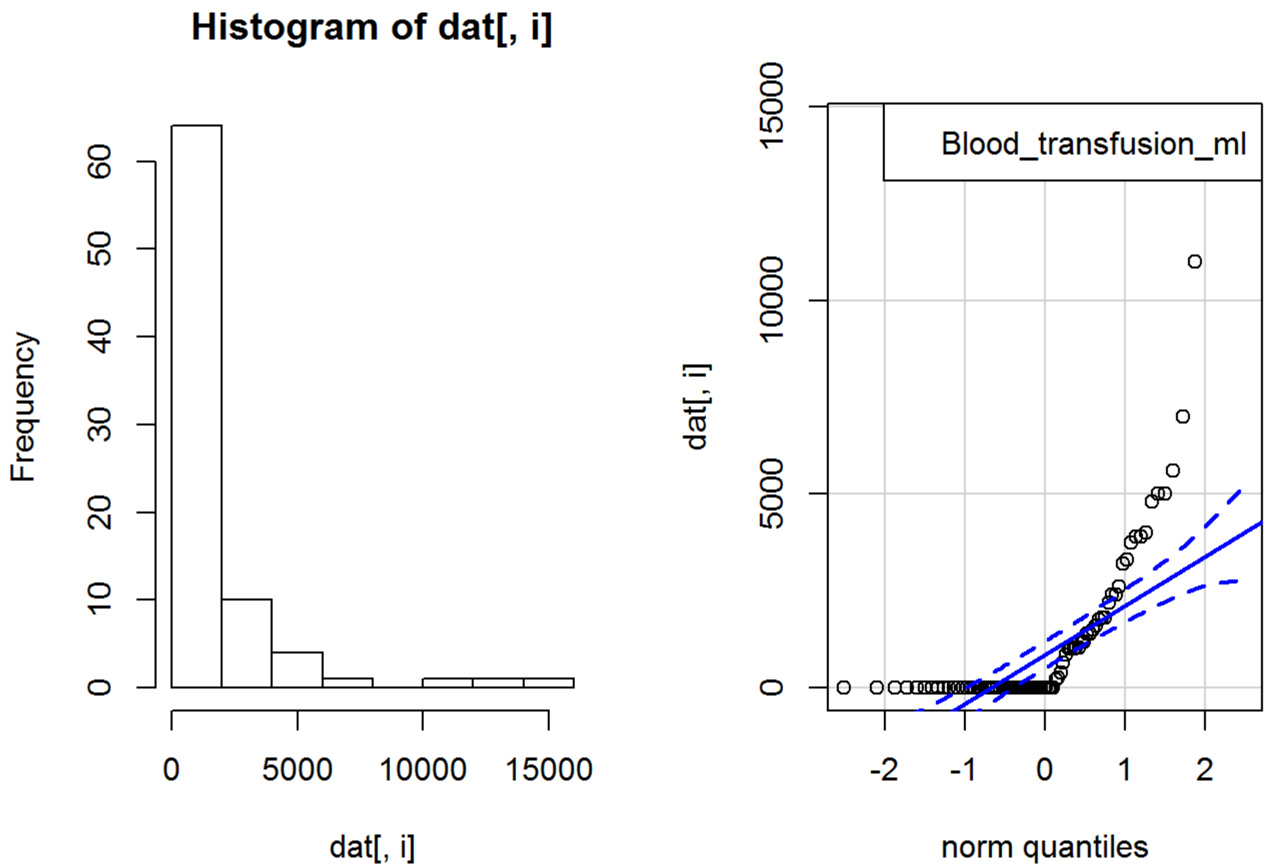
```
## [1] "Age_value"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.95984, p-value = 0.01162
```



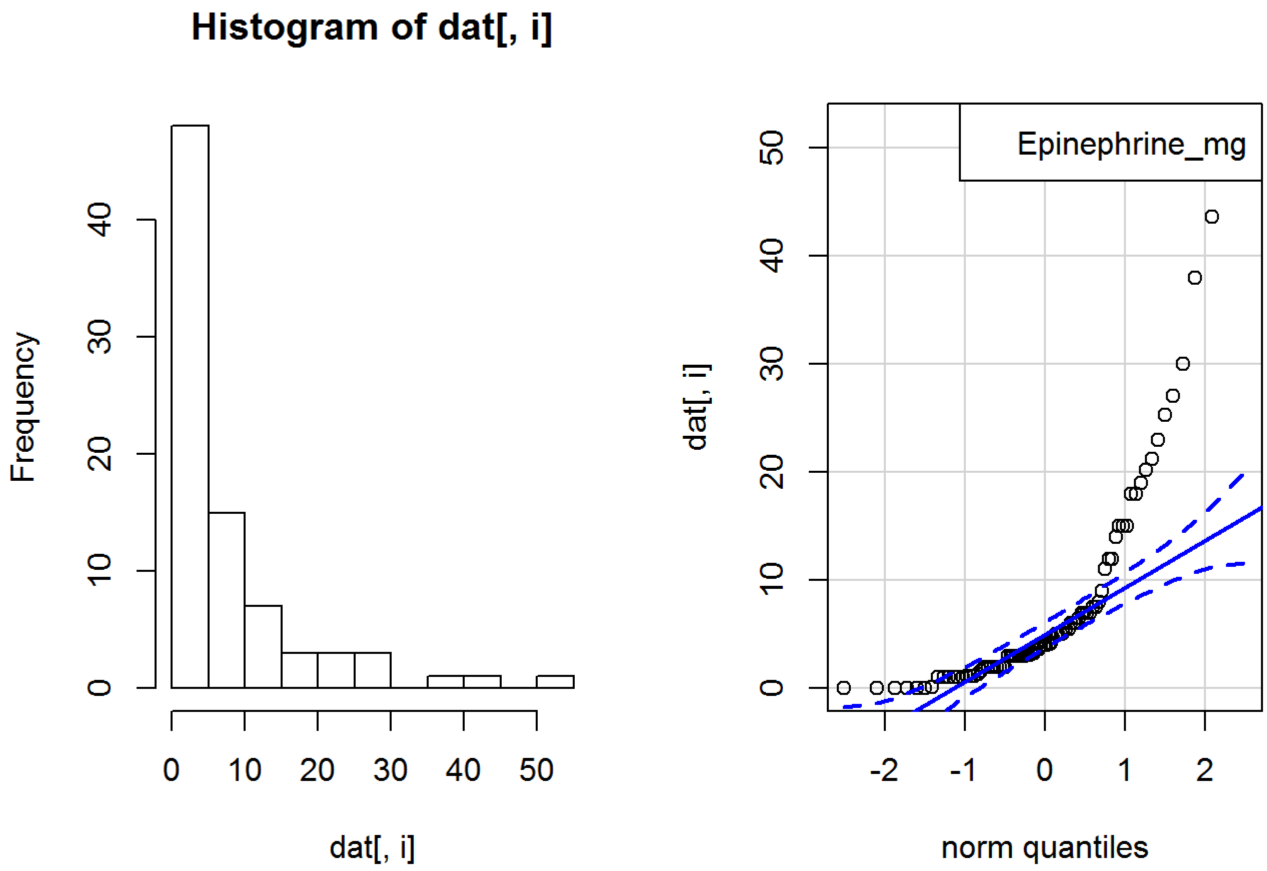
```
## [1] "BMI_value"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.99118, p-value = 0.8544
```



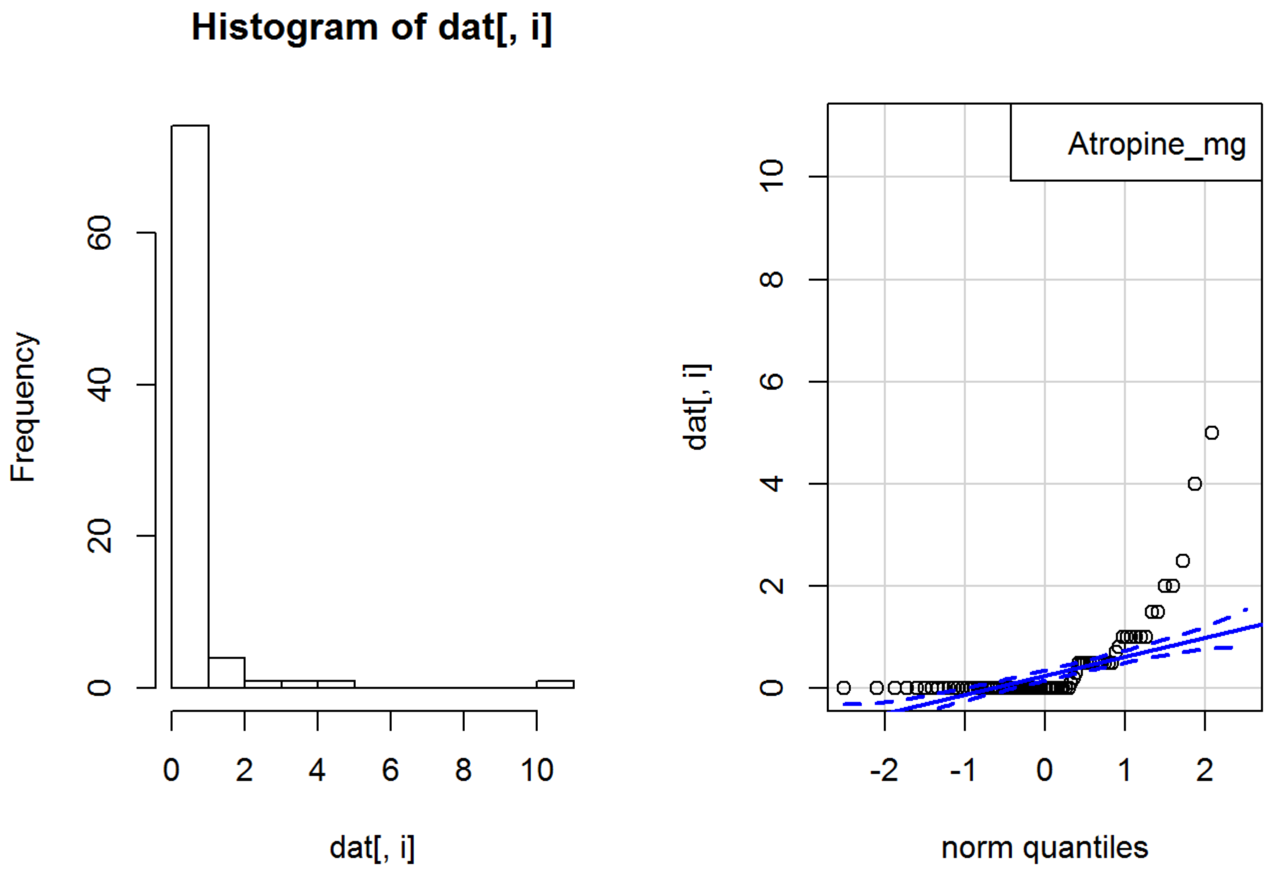
```
## [1] "Hemorrhage_ml"
## 
##  Shapiro-Wilk normality test
## 
## data:  dat[, i]
## W = 0.56845, p-value = 3.736e-14
```



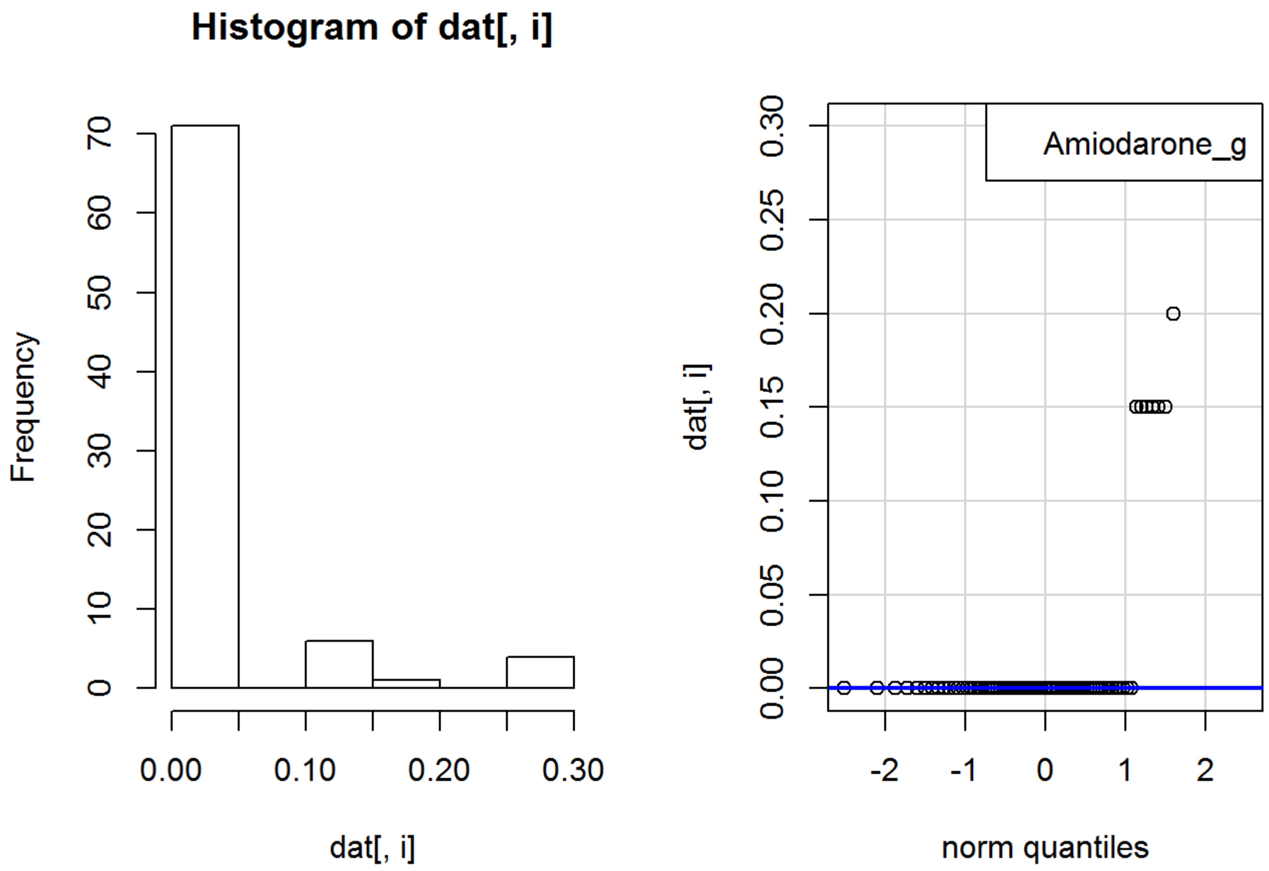
```
## [1] "Blood_transfusion_ml"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.57879, p-value = 5.592e-14
```



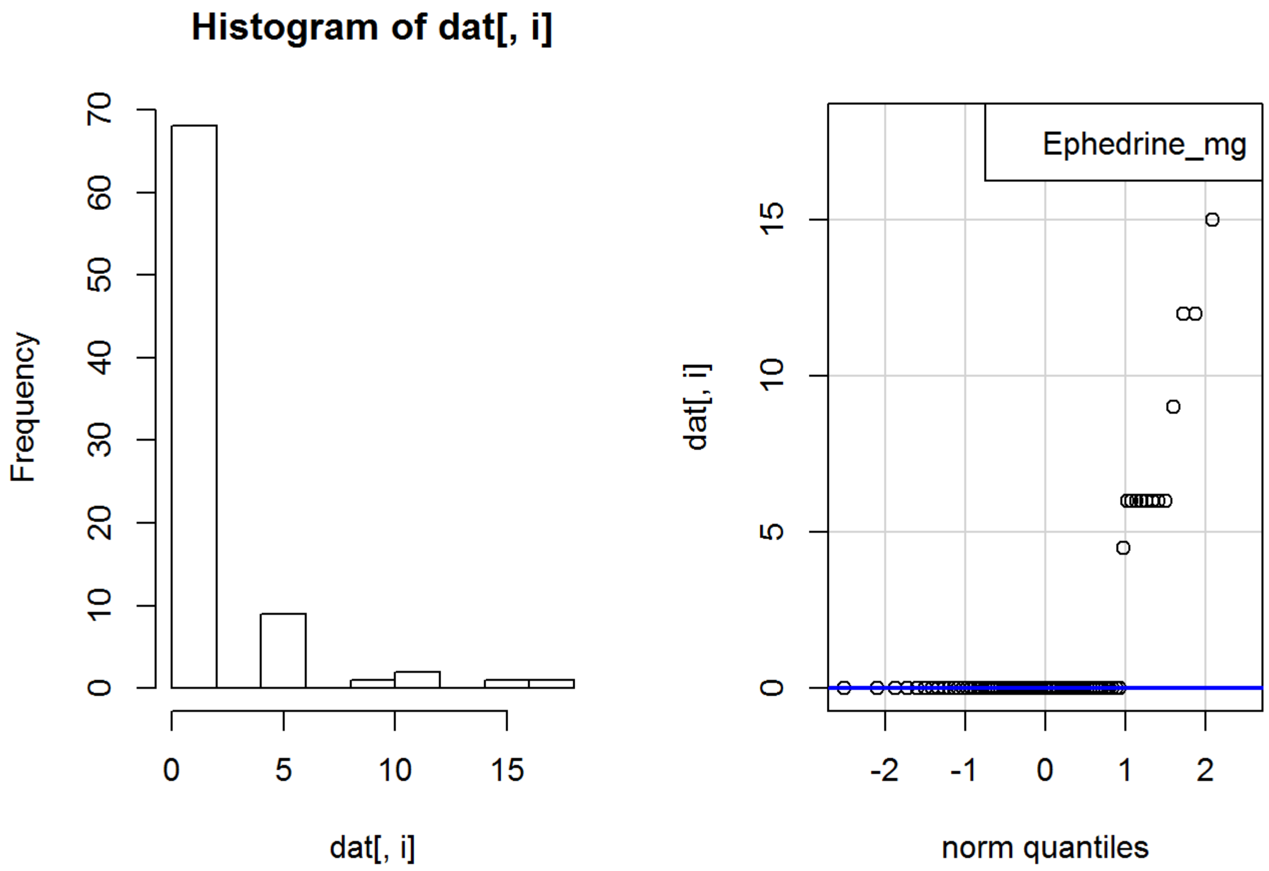
```
## [1] "Epinephrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.70795, p-value = 1.764e-11
```



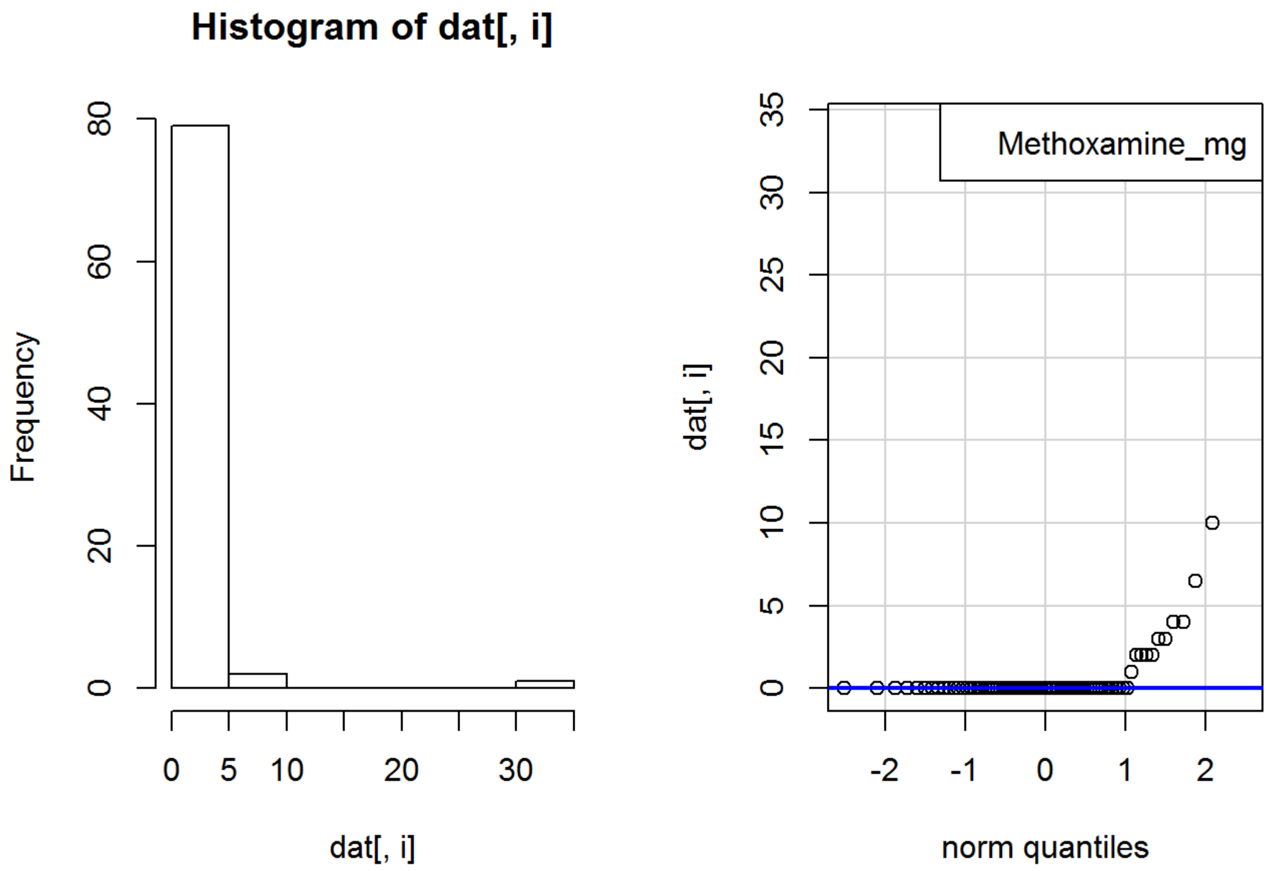
```
## [1] "Atropine_mg"
## 
##  Shapiro-Wilk normality test
## 
## data:  dat[, i]
## W = 0.39454, p-value < 2.2e-16
```



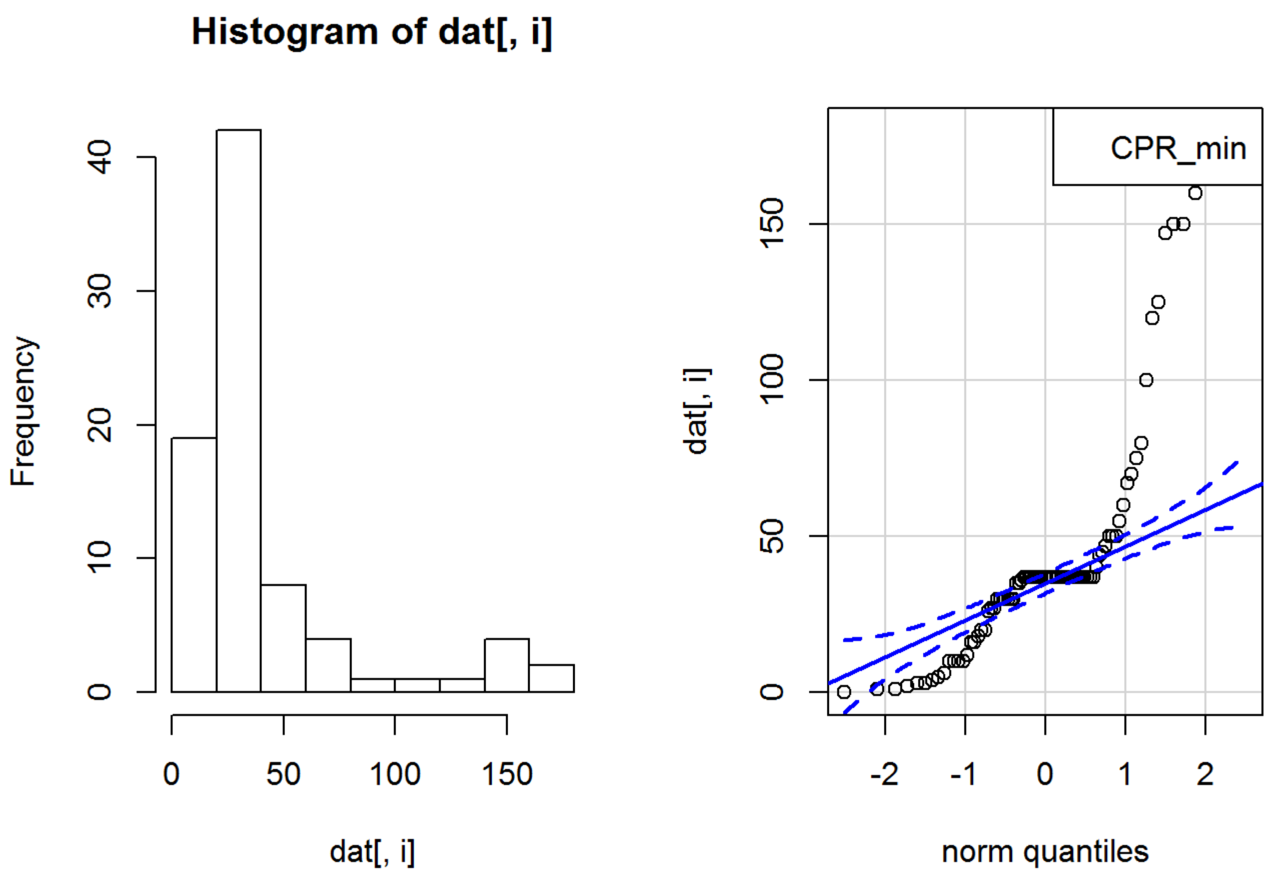
```
## [1] "Amiodarone_g"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.41214, p-value < 2.2e-16
```



```
## [1] "Ephedrine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.46391, p-value = 8.938e-16
```



```
## [1] "Methoxamine_mg"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.22913, p-value < 2.2e-16
```



```
## [1] "CPR_min"
##
##  Shapiro-Wilk normality test
##
## data:  dat[, i]
## W = 0.73704, p-value = 8.166e-11
```

###If the continuouse variable = normal distribution, then calcualte its Mean and Standar deviations with p-values calculated using two-sample t-tests and one-way ANOVAs

```
#H0: the mean values of 2 groups are same
#H1: the mean values of 2 groups are statistically significant different

my_seq2 <- c(2,3)
print('continuouse variable = normal distribution')
```

```
## [1] "continuouse variable = normal distribution"
```

```
print('in regard to total patients ')
```

```
## [1] "in regard to total patients "
```

```
# in regard to total patients
```

```
for (i2 in my_seq2) { # variables to compare
  mean = mean(data[, i2], na.rm = TRUE)
  sd = sd(data[, i2], na.rm = TRUE)
  # add next legend text
  legend_text <- colnames(data)[i2]
  print(colnames(data)[i2])
  print('mean=') # print results of t-test
  print(mean)
  print('sd=')
  print(sd)
}
```

```
## [1] "Age_value"
## [1] "mean="
## [1] 49.41287
## [1] "sd="
## [1] 18.44103
## [1] "BMI_value"
## [1] "mean="
## [1] 24.27697
## [1] "sd="
## [1] 3.908882
```

```
# in regard to outcome status
print('in regard to outcome status ')
```

```
## [1] "in regard to outcome status "
```

```
for (i2 in my_seq2) { # variables to compare
  anovas <- group_by(data, Outcome) %>%
    summarise(
    mean = mean(data[, i2], na.rm = TRUE),
    sd = sd(data[, i2], na.rm = TRUE)
  )
  # add next legend text
  legend_text <- colnames(data)[i2]
  print(colnames(data)[i2])
  print(anovas)
  #print('mean=') # print results of t-test
  #print(mean)
  #print('sd=')
  #print(sd)
}
```

```
## [1] "Age_value"
##       mean       sd
## 1 49.41287 18.44103
## [1] "BMI_value"
##       mean       sd
## 1 24.27697 3.908882
```

```r
# in regard to outcome status
print('in regard to outcome status Survived ')
```

```
## [1] "in regard to outcome status Survived "
```

```r
for (i2 in my_seq2) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Survived')
    mean = mean(dat[, i2], na.rm = TRUE)
    sd = sd(dat[, i2], na.rm = TRUE)
  # add next legend text
    legend_text <- colnames(data)[i2]
    print(colnames(data)[i2])
    print('mean=') # print results of t-test
    print(mean)
    print('sd=')
    print(sd)
}
```

```
## [1] "Age_value"
## [1] "mean="
## [1] 47.075
## [1] "sd="
## [1] 17.12083
## [1] "BMI_value"
## [1] "mean="
## [1] 23.86803
## [1] "sd="
## [1] 3.703178
```

```r
# in regard to outcome status
print('in regard to outcome status Died ')
```

```
## [1] "in regard to outcome status Died "
```

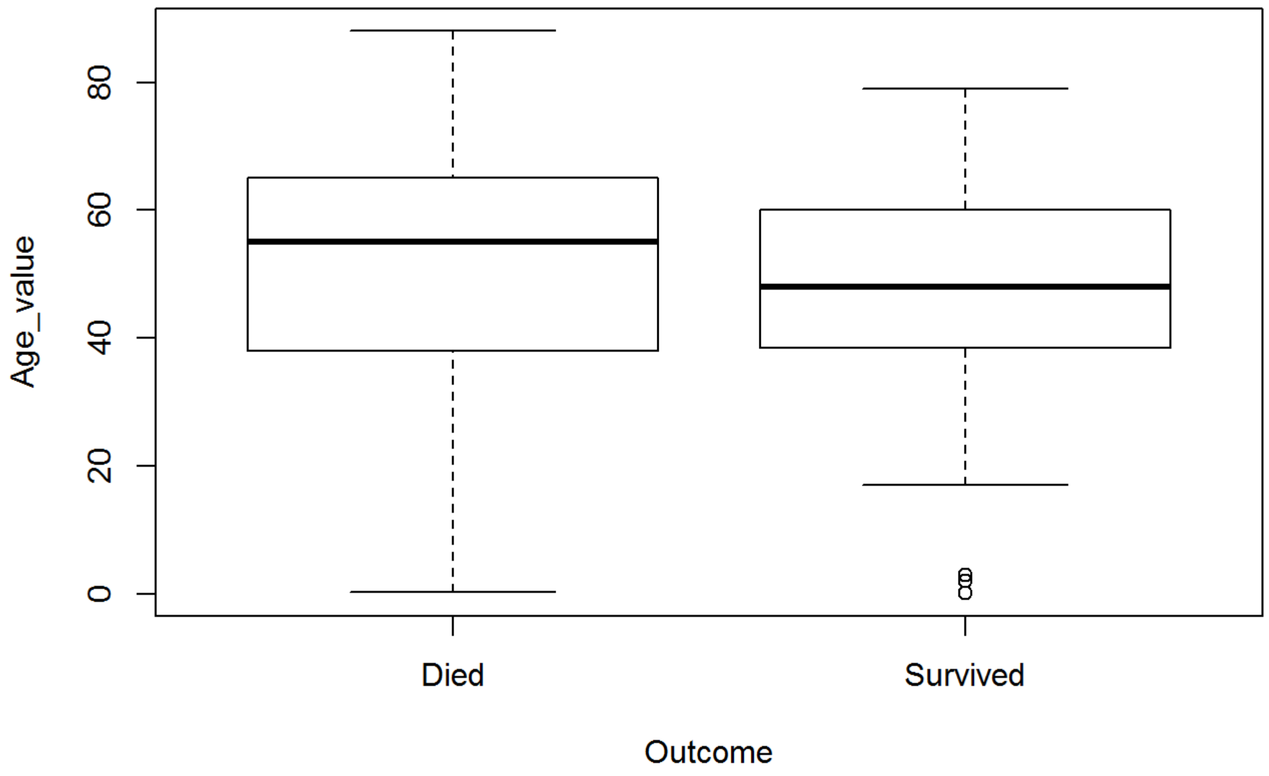
```r
for (i2 in my_seq2) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Died')
    mean = mean(dat[, i2], na.rm = TRUE)
    sd = sd(dat[, i2], na.rm = TRUE)
  # add next legend text
    legend_text <- colnames(data)[i2]
    print(colnames(data)[i2])
    print('mean=') # print results of t-test
    print(mean)
    print('sd=')
    print(sd)
}
```

```
## [1] "Age_value"
## [1] "mean="
## [1] 51.35159
## [1] "sd="
## [1] 19.35744
## [1] "BMI_value"
## [1] "mean="
## [1] 24.6161
## [1] "sd="
## [1] 4.062802
```

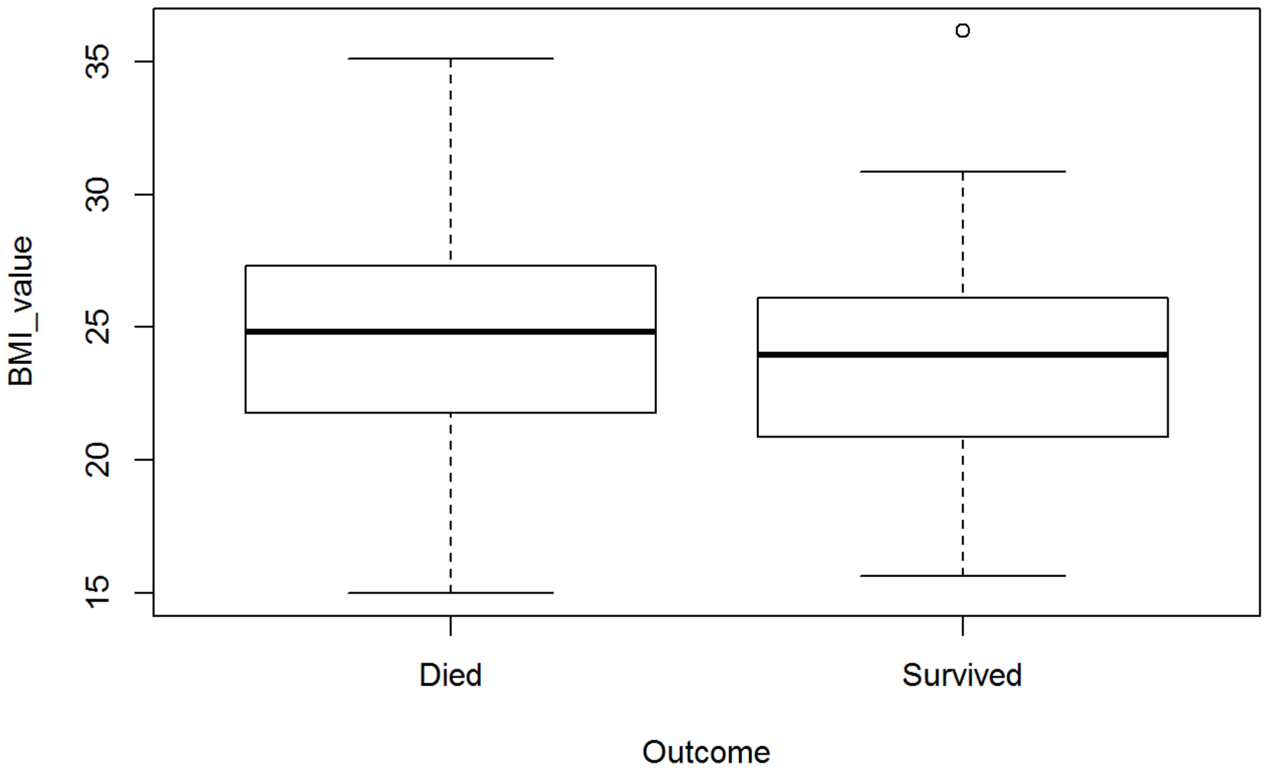
```
# Draw boxplots illustrating the distributions by group (with the boxplot() function
#with p-values calculated using two-sample t-tests and one-way ANOVAs



# 1st method: two sample Student’s t-test
print('1st method: two sample Student’s t-test')
```

```
## [1] "1st method: two sample Student’s t-test"
```

```
for (i2 in my_seq2) { # variables to compare
  boxplot(data[, i2] ~ data$Outcome, # draw boxplots by group
          ylab = names(data[i2]), # rename y-axis with variable's name
          xlab = "Outcome"
  )
  # add next legend text
  legend_text <- colnames(data)[i2]
  print(colnames(data)[i2])
  print(t.test(data[, i2] ~ data$Outcome)) # print results of t-test
}
```



```
## [1] "Age_value"
##
##  Welch Two Sample t-test
##
## data:  data[, i2] by data$Outcome
## t = 1.4351, df = 147.36, p-value = 0.1534
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.612427 10.165597
## sample estimates:
##     mean in group Died mean in group Survived
##               51.35159               47.07500
```

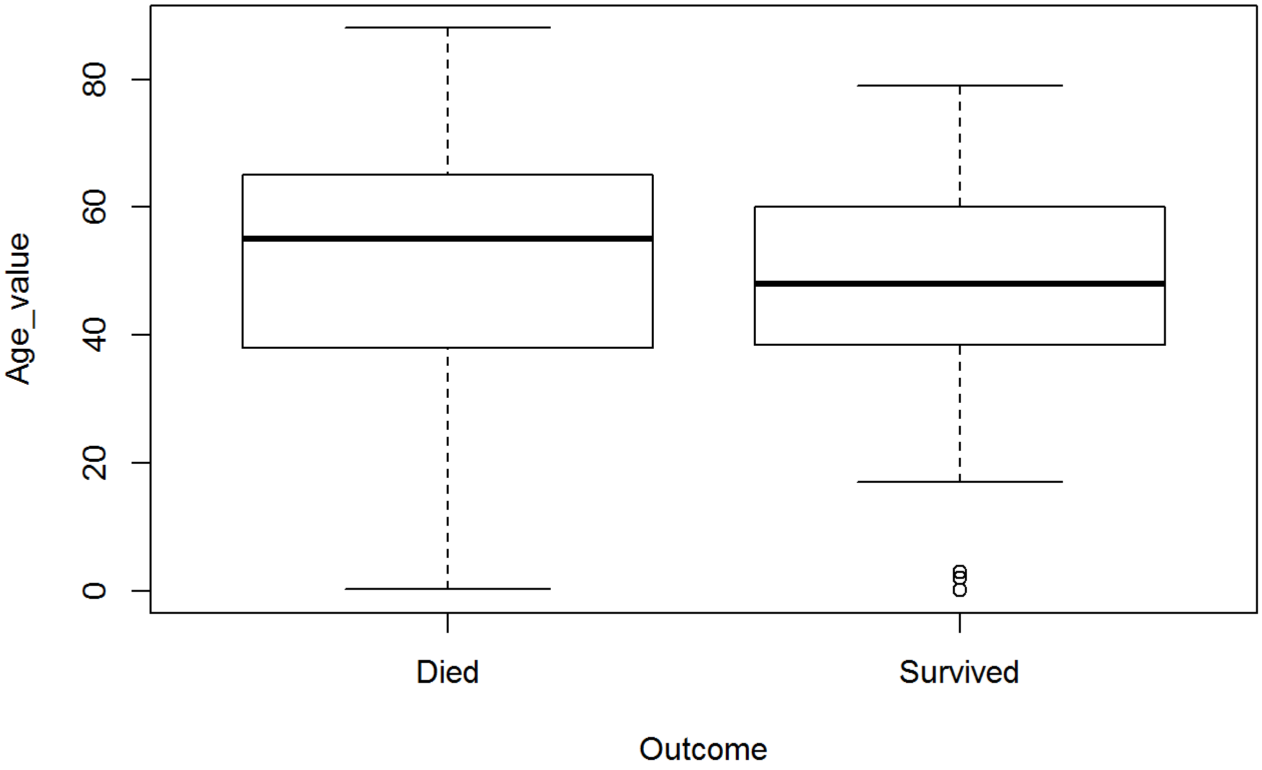


```
## [1] "BMI_value"
##
##  Welch Two Sample t-test
##
## data:  data[, i2] by data$Outcome
## t = 1.1784, df = 146.65, p-value = 0.2405
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5064603  2.0026016
## sample estimates:
##     mean in group Died mean in group Survived
##               24.61610               23.86803
```

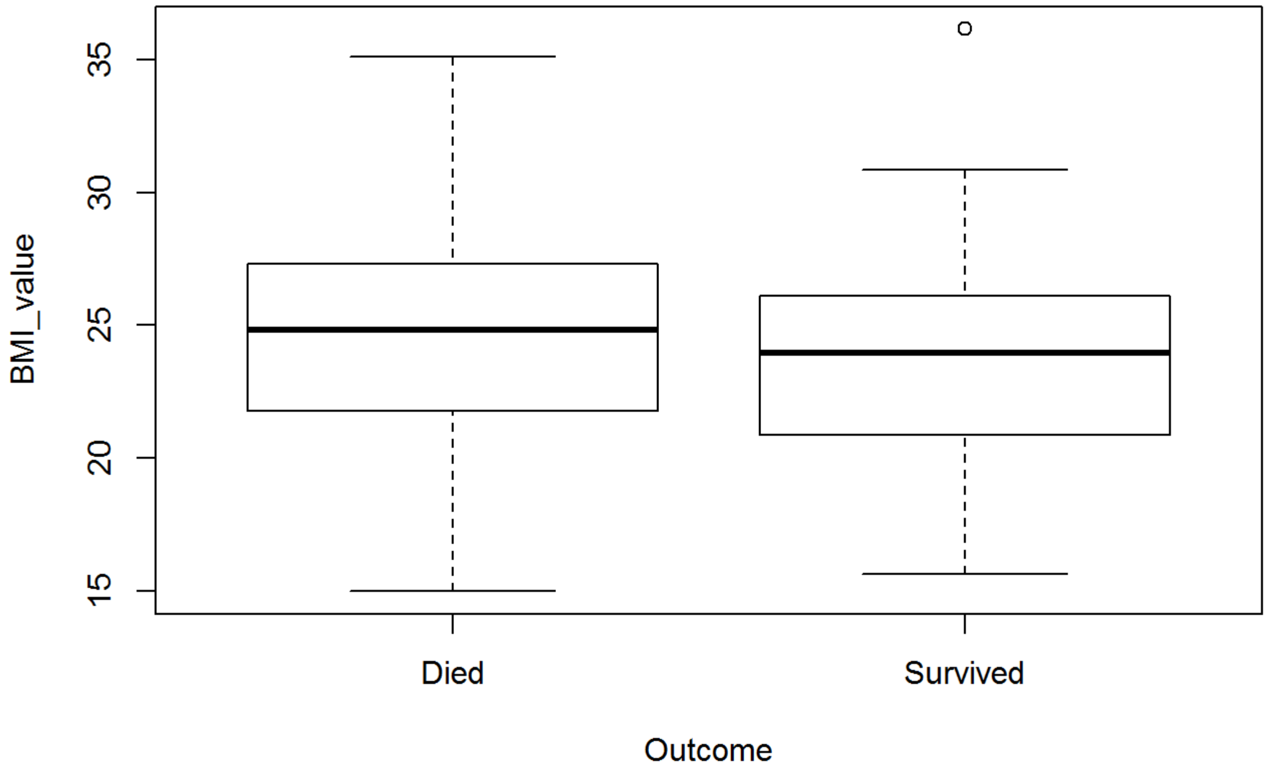
```
# Draw boxplots illustrating the distributions by group (with the boxplot() function
#with p-values calculated using two-sample t-tests and one-way ANOVAs


# 2nd method: One-way ANOVA
print('2nd method: One-way ANOVA')
```

```
## [1] "2nd method: One-way ANOVA"
```

```
for (i2 in my_seq2) { # variables to compare
  boxplot(data[, i2] ~ data$Outcome, # draw boxplots by group
          ylab = names(data[i2]), # rename y-axis with variable's name
          xlab = "Outcome"
  )
  # add next legend text
  legend_text <- colnames(data)[i2]
  print(colnames(data)[i2])
  #print(summary(aov(data[, i] ~ data$Outcome))) # print results of One-way ANOVA
  print(oneway.test(data[, i2] ~ data$Outcome))
}
```



```
## [1] "Age_value"
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  data[, i2] and data$Outcome
## F = 2.0595, num df = 1.00, denom df = 147.36, p-value = 0.1534
```



```
## [1] "BMI_value"
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  data[, i2] and data$Outcome
## F = 1.3887, num df = 1.00, denom df = 146.65, p-value = 0.2405
```

###If the continuouse variable =/= normal distribution, then calcualte its Medians and 1st/3rd quartiles with p-values calcualted using Mann -Whitney U and Kruskal -Wallis tests

```
my_seq3 <- c(21:28)
print('continuouse variable =/= normal distribution ')
```

```
## [1] "continuouse variable =/= normal distribution "
```

```
# in regard to total patients
for (i3 in my_seq3) { # variables to compare
  summary <- summary(data[, i3])
  # add next legend text
  print(colnames(data)[i3])
  print(summary)
}
```

```
## [1] "Hemorrhage_ml"
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##      0.00     2.25   100.00  1162.95   500.00 18000.00 
## [1] "Blood_transfusion_ml"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##        0       0       0    1006    1000   14500 
## [1] "Epinephrine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.000   0.125   2.000   5.052   5.875  52.000 
## [1] "Atropine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.000   0.000   0.000   0.414   0.500  11.000 
## [1] "Amiodarone_g"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.0000  0.0000  0.0000  0.0281  0.0000  0.5000 
## [1] "Ephedrine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.00    0.00    0.00    1.67    0.00   18.00 
## [1] "Methoxamine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.0000  0.0000  0.0000  0.9167  0.0000 34.0000 
## [1] "CPR_min"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.00   10.00   30.00   31.56   37.00  180.00 
```

```
print('continuouse variable =/= normal distribution to outcome status Survived ')
```

```
## [1] "continuouse variable =/= normal distribution to outcome status Survived "
```

```
# in regard to total patients
for (i3 in my_seq3) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Survived')
    summary <- summary(dat[, i3])
  # add next legend text
    print(colnames(dat)[i3])
    print(summary)
}
```

```
## [1] "Hemorrhage_ml"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.00    4.25   95.00  542.13  200.00 5000.00 
## [1] "Blood_transfusion_ml"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      0.0     0.0     0.0   461.5     0.0  5600.0 
## [1] "Epinephrine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.000   0.000   0.500   1.749   2.000  13.500 
## [1] "Atropine_mg"
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.0000  0.0000  0.0000  0.2721  0.5000  5.0000 
## [1] "Amiodarone_g"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.00000 0.00000 0.02816 0.00000 0.50000 
## [1] "Ephedrine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   0.000   1.941   2.250  12.000 
## [1] "Methoxamine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.9412  0.0000 13.0000 
## [1] "CPR_min"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    1.00   11.00   16.02   37.00   37.00
```

```
print('continuouse variable =/= normal distribution to outcome status Died ')
```

```
## [1] "continuouse variable =/= normal distribution to outcome status Died "
```

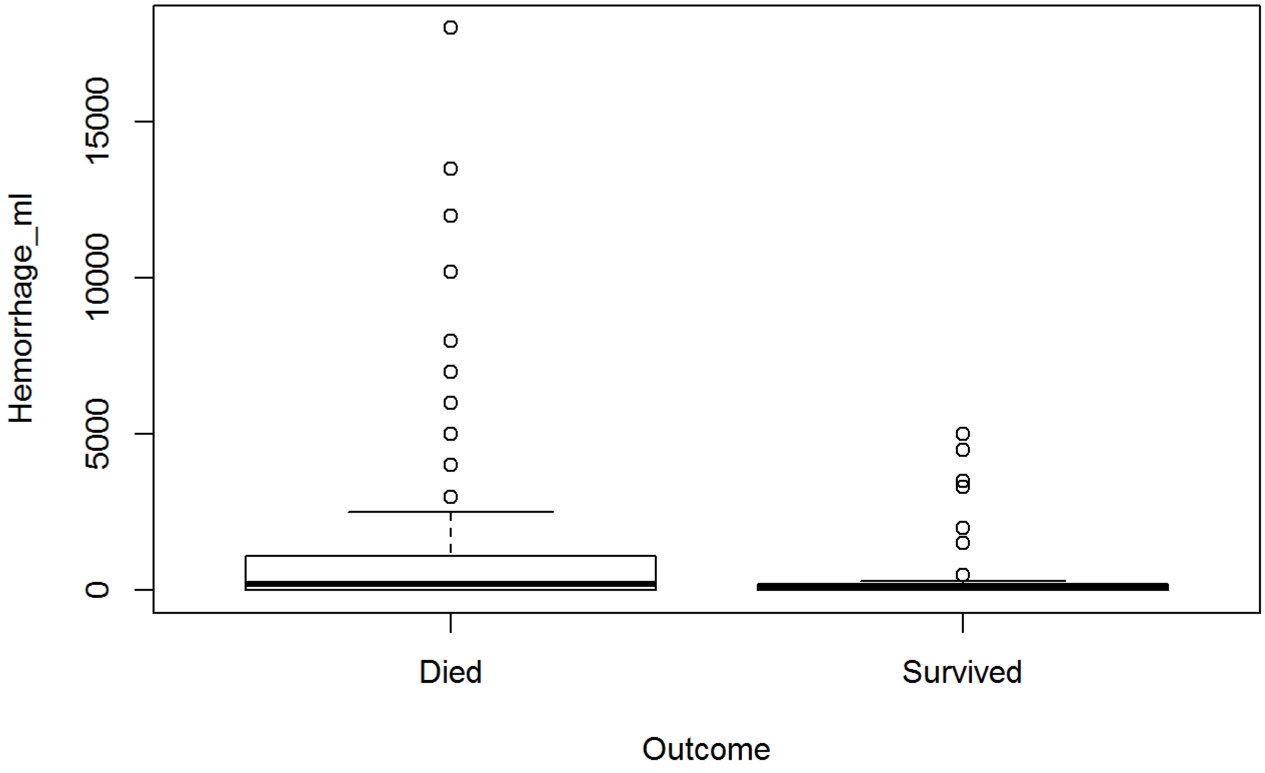
```
# in regard to total patients
for (i3 in my_seq3) { # variables to compare
  dat <- data %>%
    filter(Outcome=='Died')
    summary <- summary(dat[, i3])
  # add next legend text
    print(colnames(dat)[i3])
    print(summary)
}
```

```
## [1] "Hemorrhage_ml"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##     0.00     0.75   200.00  1677.78  1075.00 18000.00 
## [1] "Blood_transfusion_ml"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0       0       0    1457    1712   14500 
## [1] "Epinephrine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.000   4.000   7.791   7.875  52.000 
## [1] "Atropine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.5317  0.5000 11.0000 
## [1] "Amiodarone_g"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.00000 0.00000 0.02805 0.00000 0.30000 
## [1] "Ephedrine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   0.000   1.445   0.000  18.000 
## [1] "Methoxamine_mg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.8963  0.0000 34.0000 
## [1] "CPR_min"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   27.00   37.00   44.44   43.00  180.00
```

```
#H0: the 2 groups are similar
#H1: the 2 groups are different

# Draw boxplots illustrating the distributions by group (with the boxplot() function
#with p-values calculated using two-sample t-tests and one-way ANOVAs

#1st method:  Mann -Whitney U
print('Mann -Whitney U test')
```

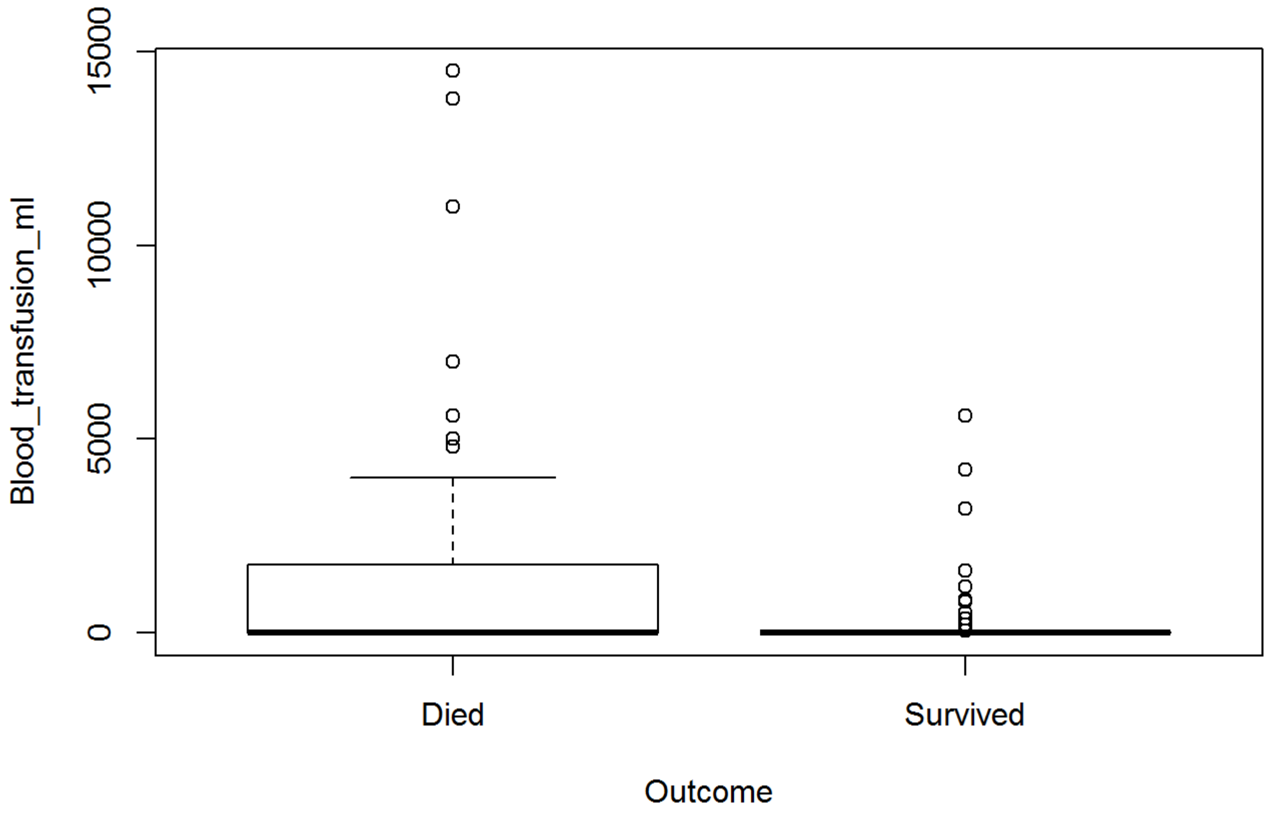
```
## [1] "Mann -Whitney U test"
```

```
for (i3 in my_seq3) { # variables to compare
  boxplot(data[, i3] ~ data$Outcome, # draw boxplots by group
          ylab = names(data[i3]), # rename y-axis with variable's name
          xlab = "Outcome"
  )
  # add next legend text
  legend_text <- colnames(data)[i3]
  print(colnames(data)[i3])
  print(wilcox.test(data[, i3] ~ data$Outcome)) # print results of t-test
}
```

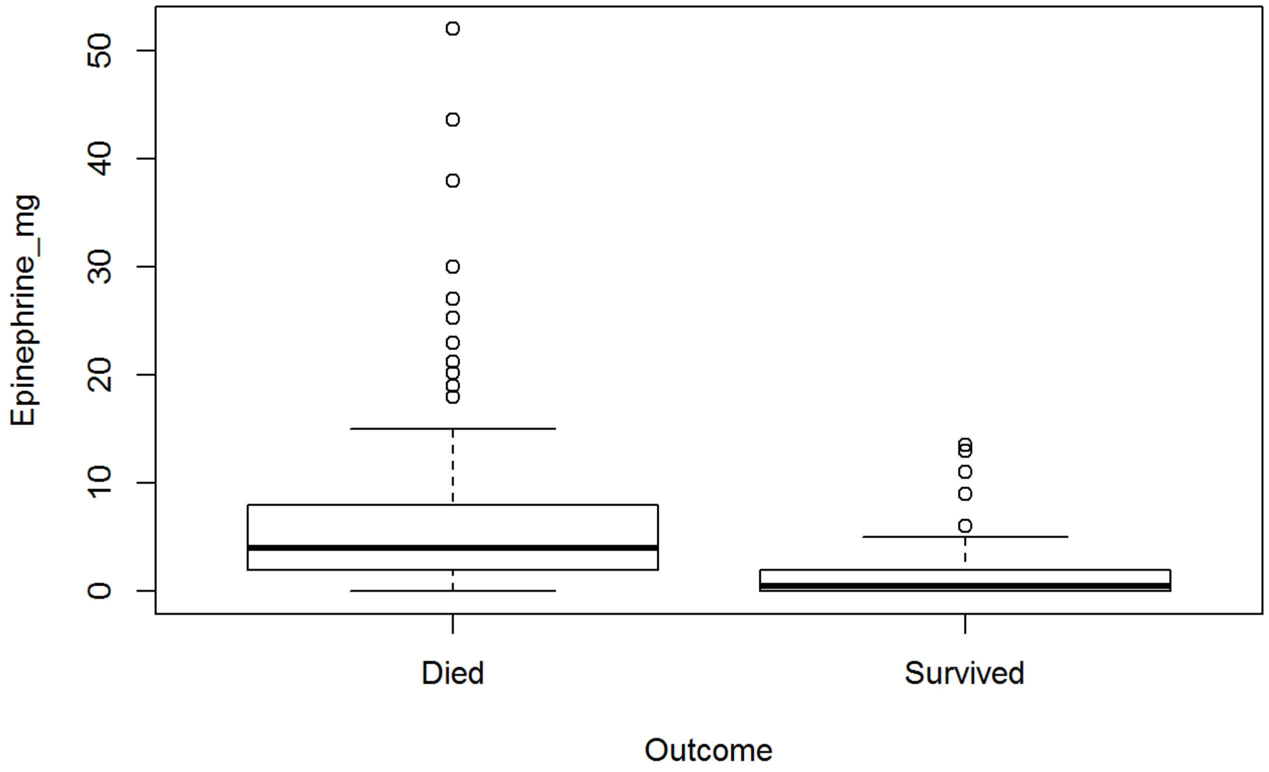


```
## [1] "Hemorrhage_ml"
```

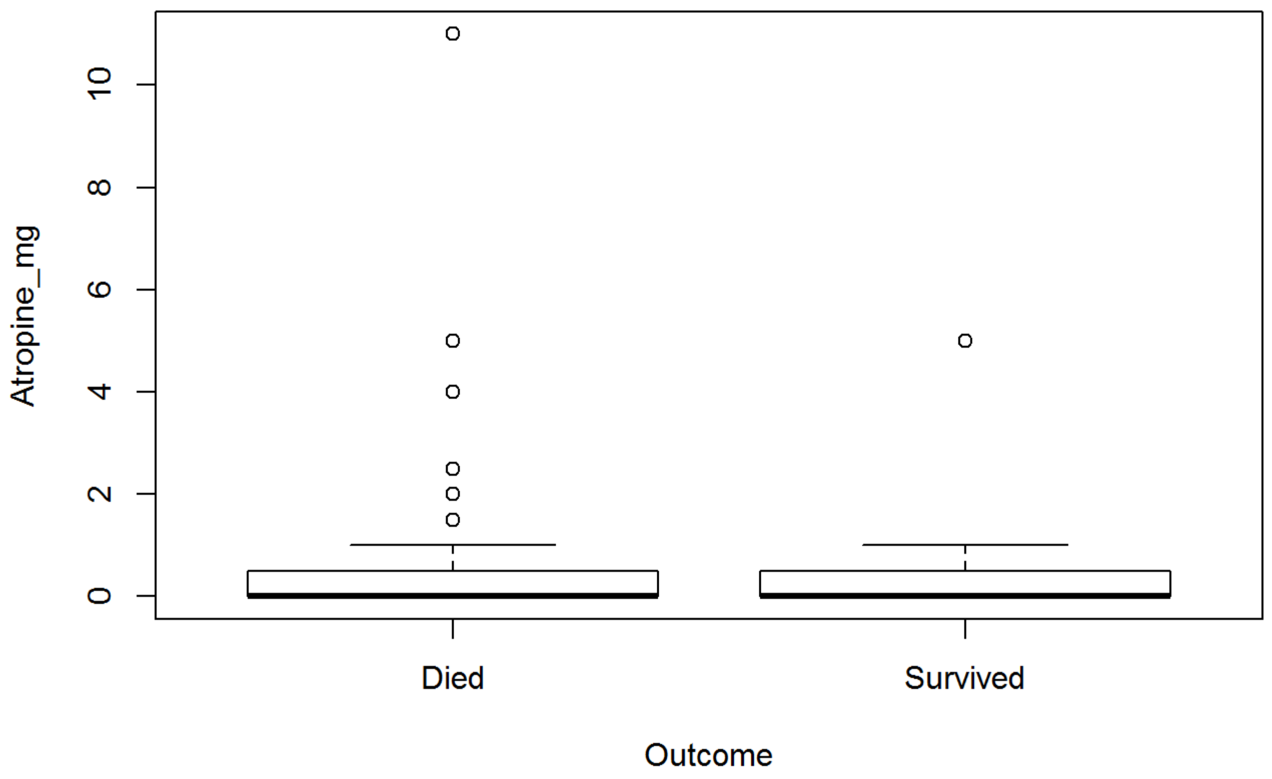
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 3372, p-value = 0.02642
## alternative hypothesis: true location shift is not equal to 0
```



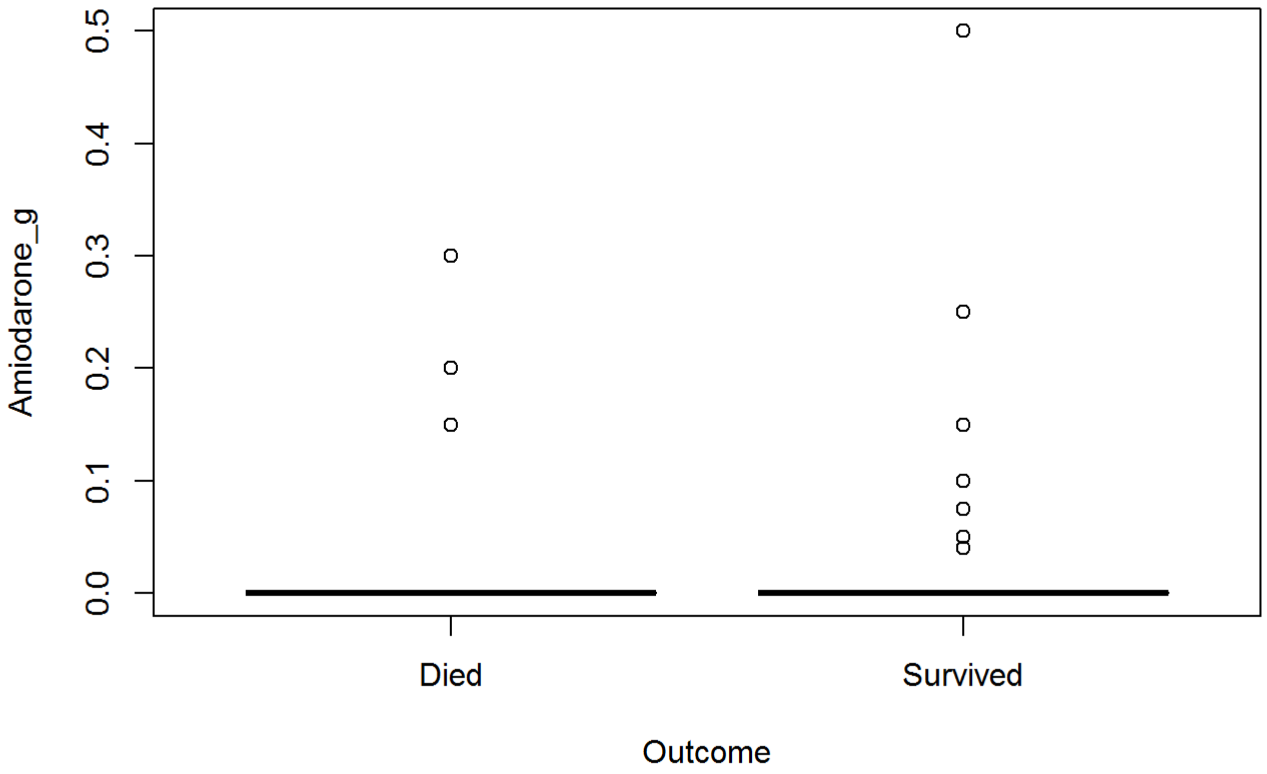
```
## [1] "Blood_transfusion_ml"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 3499.5, p-value = 0.001573
## alternative hypothesis: true location shift is not equal to 0
```



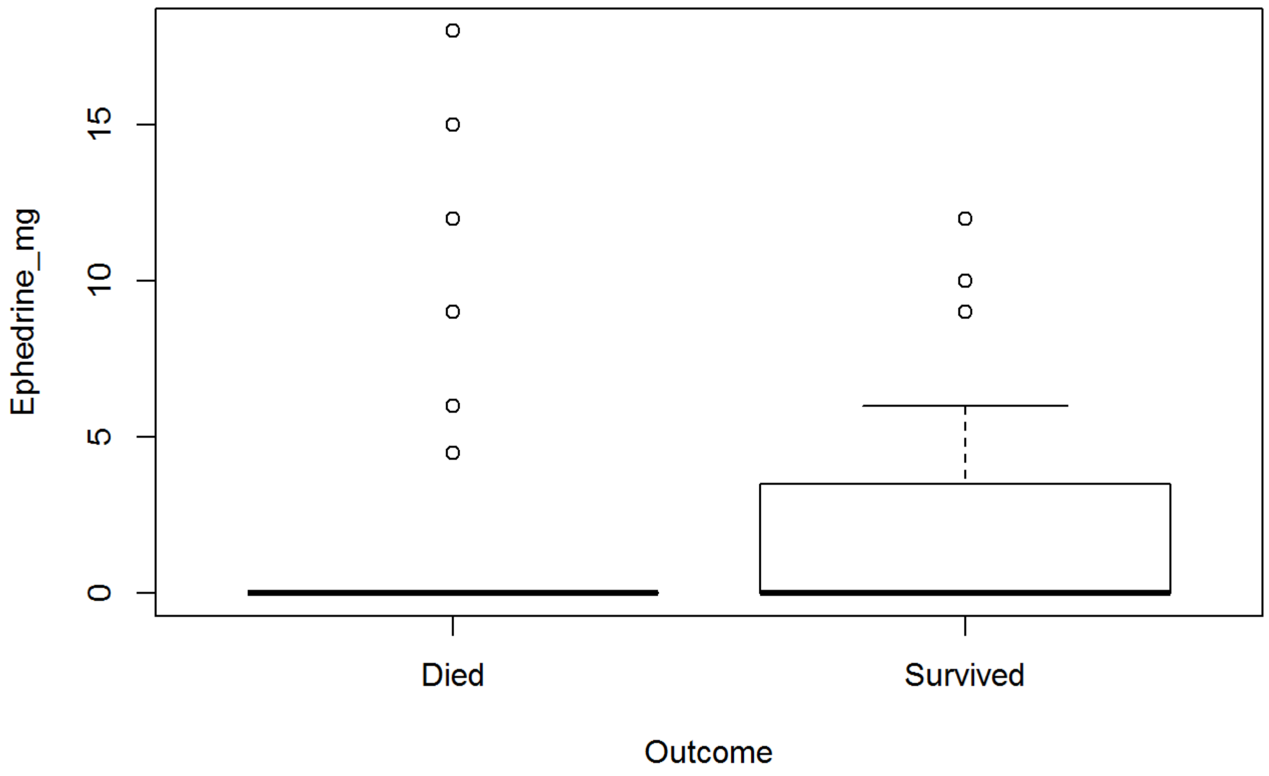
```
## [1] "Epinephrine_mg"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 4525, p-value = 4.526e-11
## alternative hypothesis: true location shift is not equal to 0
```



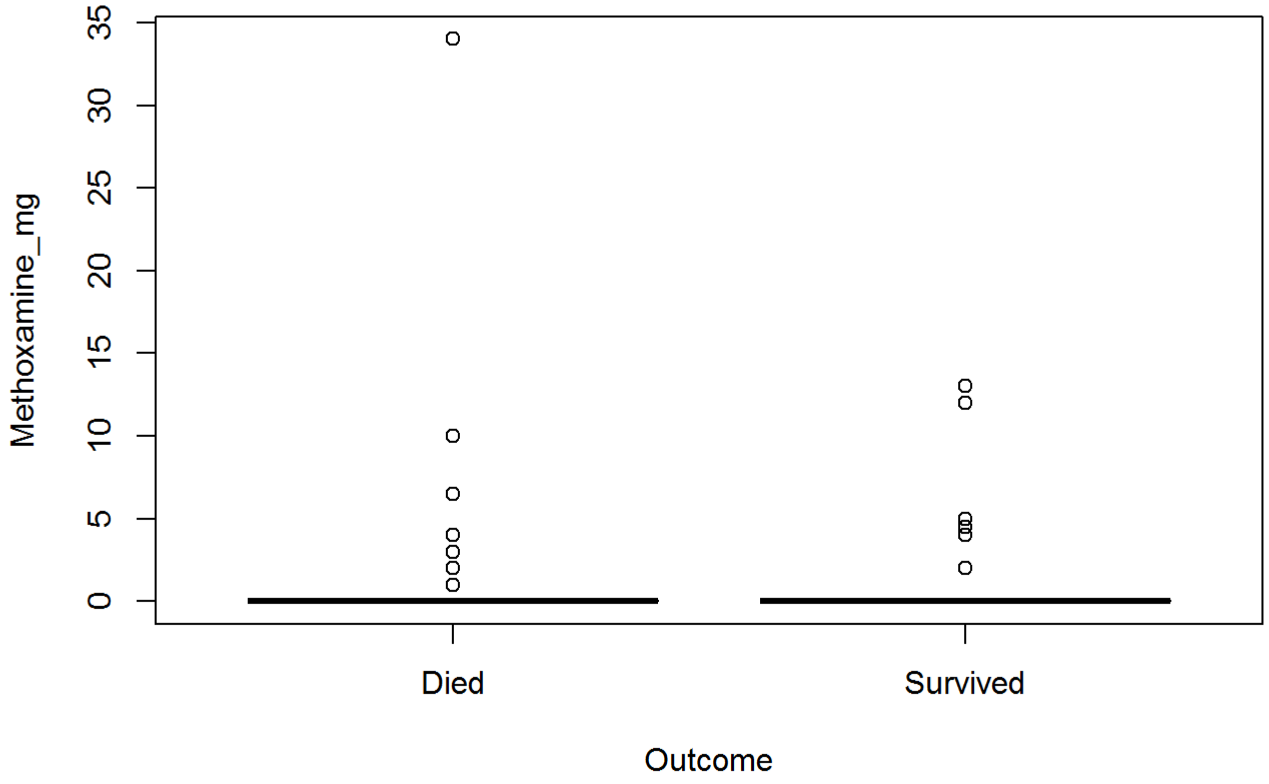
```
## [1] "Atropine_mg"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 2865, p-value = 0.7421
## alternative hypothesis: true location shift is not equal to 0
```



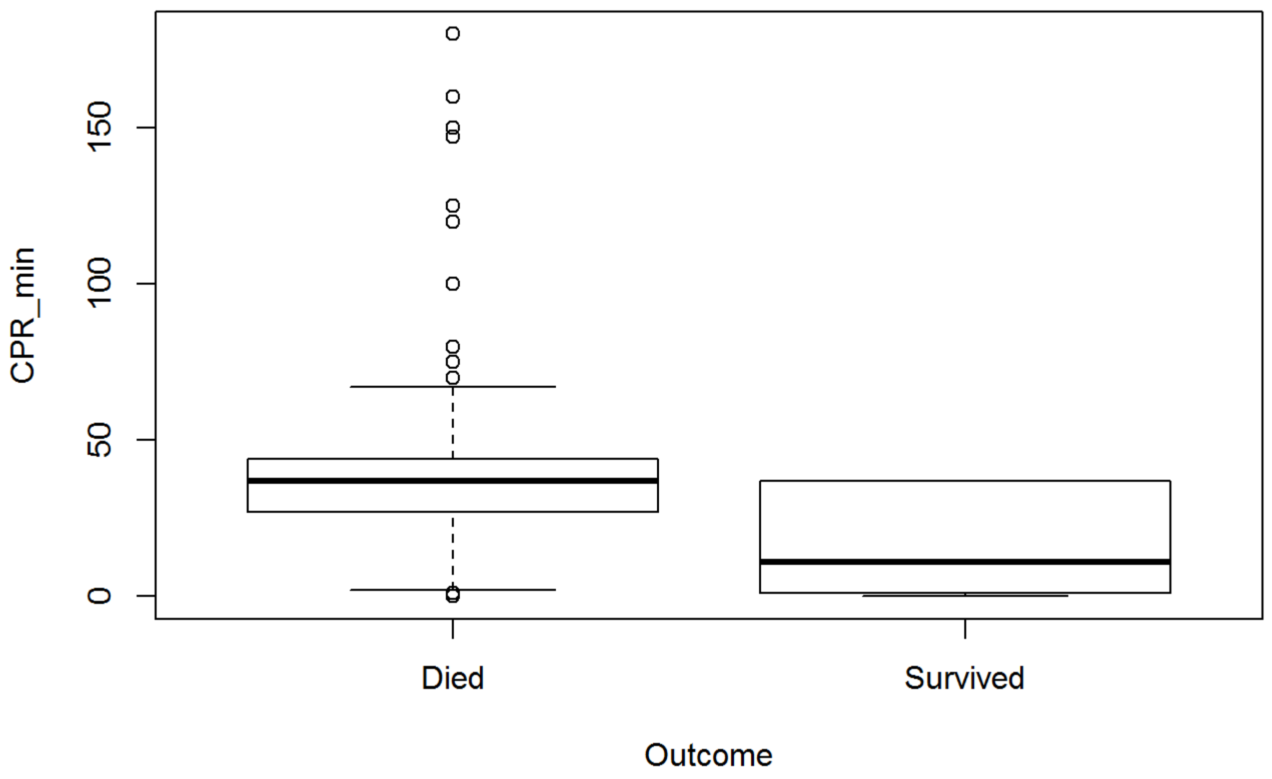
```
## [1] "Amiodarone_g"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 2700, p-value = 0.598
## alternative hypothesis: true location shift is not equal to 0
```



```
## [1] "Ephedrine_mg"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 2534.5, p-value = 0.1816
## alternative hypothesis: true location shift is not equal to 0
```



```
## [1] "Methoxamine_mg"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 2764, p-value = 0.8853
## alternative hypothesis: true location shift is not equal to 0
```

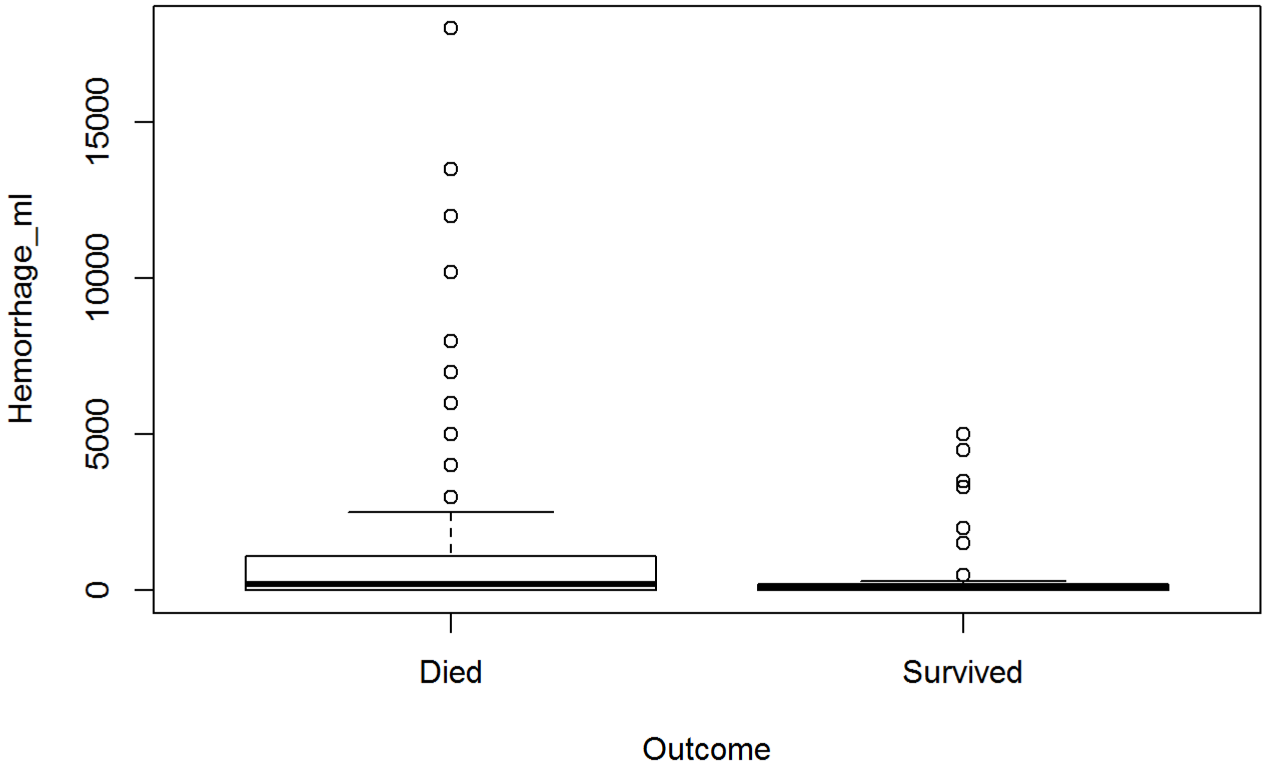


```
## [1] "CPR_min"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data[, i3] by data$Outcome
## W = 4318.5, p-value = 4.528e-09
## alternative hypothesis: true location shift is not equal to 0
```

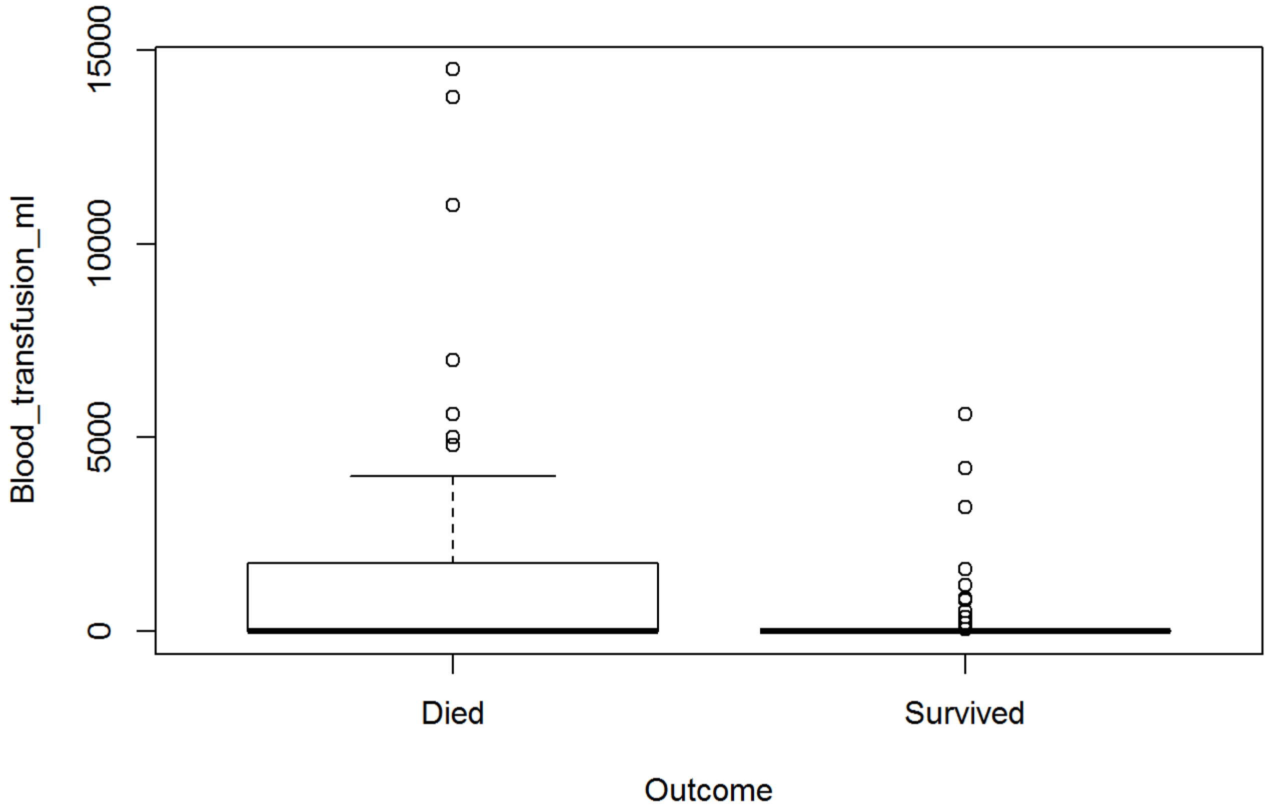
```
# Draw boxplots illustrating the distributions by group (with the boxplot() function
#with p-values calculated using two-sample t-tests and one-way ANOVAs

#H0: the 2 groups are similar
#H1: the 2 groups are different

#2nd method: Kruskal -Wallis test
print('Kruskal -Wallis test')
```

```
## [1] "Kruskal -Wallis test"
```

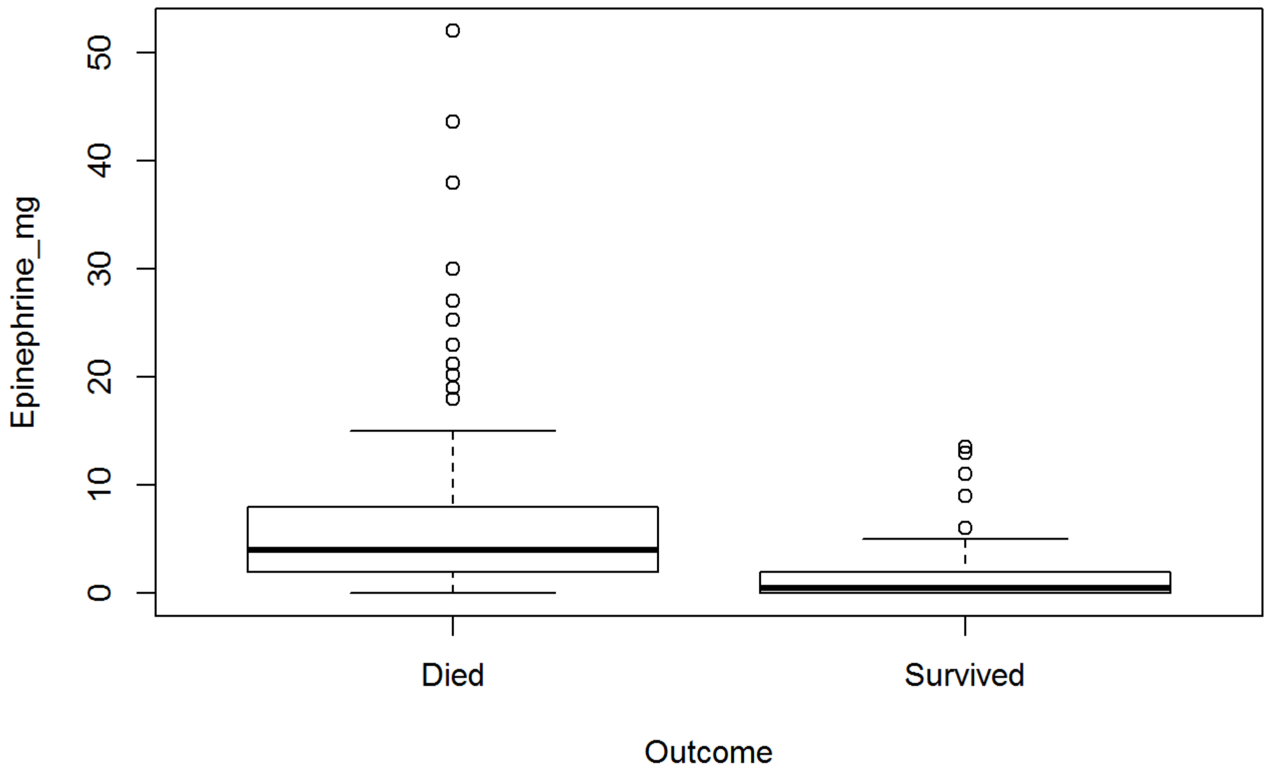
```
for (i3 in my_seq3) { # variables to compare
  boxplot(data[, i3] ~ data$Outcome, # draw boxplots by group
          ylab = names(data[i3]), # rename y-axis with variable's name
          xlab = "Outcome"
  )
  # add next legend text
  legend_text <- colnames(data)[i3]
  print(colnames(data)[i3])
  print(kruskal.test(data[, i3] ~ data$Outcome)) # print results of t-test
}
```



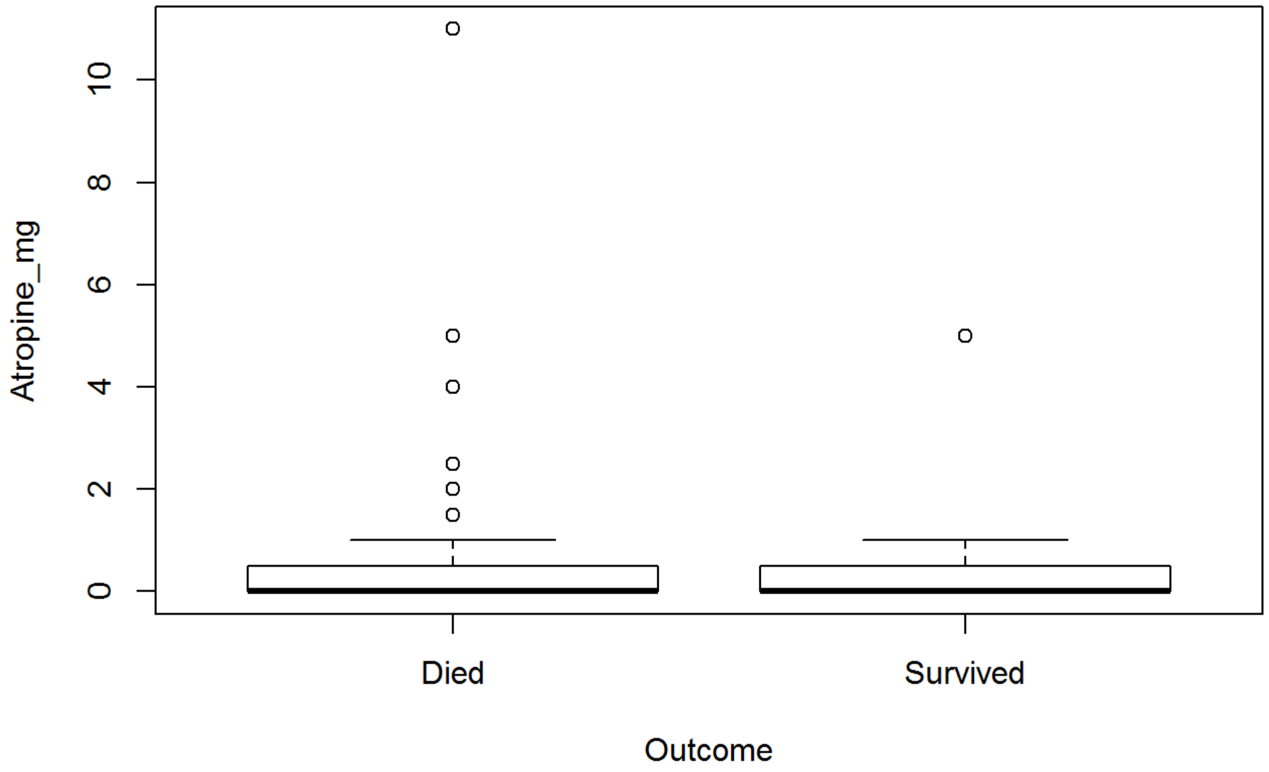
```
## [1] "Hemorrhage_ml"
## 
##  Kruskal-Wallis rank sum test
## 
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 4.9366, df = 1, p-value = 0.02629
```



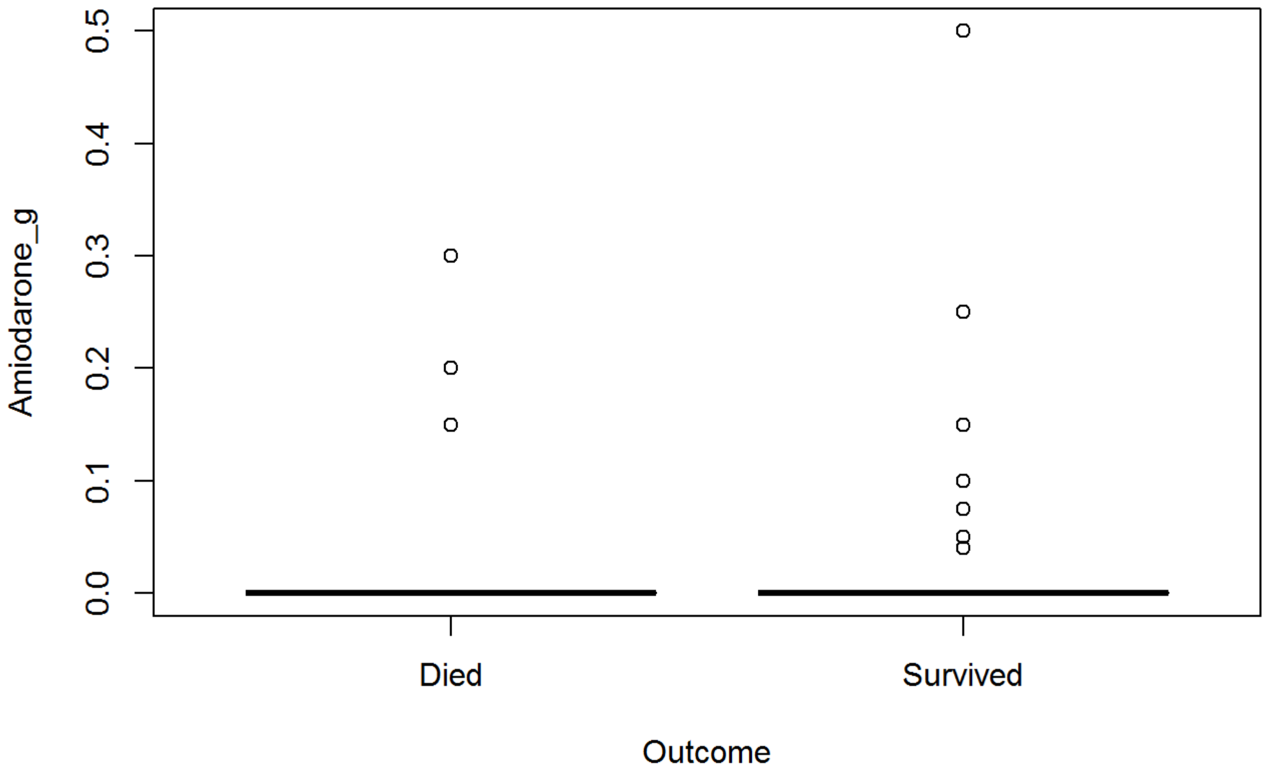
```
## [1] "Blood_transfusion_ml"
##
##  Kruskal-Wallis rank sum test
##
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 10.006, df = 1, p-value = 0.001561
```



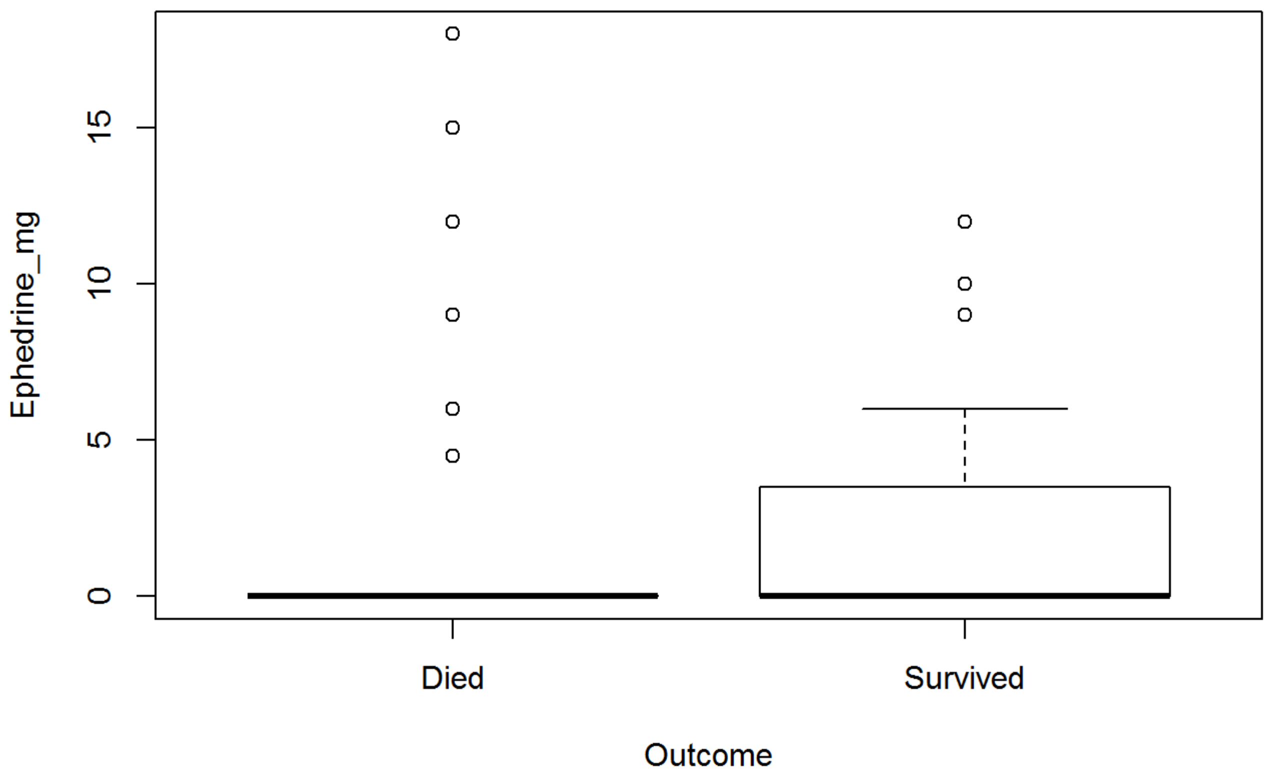
```
## [1] "Epinephrine_mg"
##
##  Kruskal-Wallis rank sum test
##
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 43.397, df = 1, p-value = 4.468e-11
```



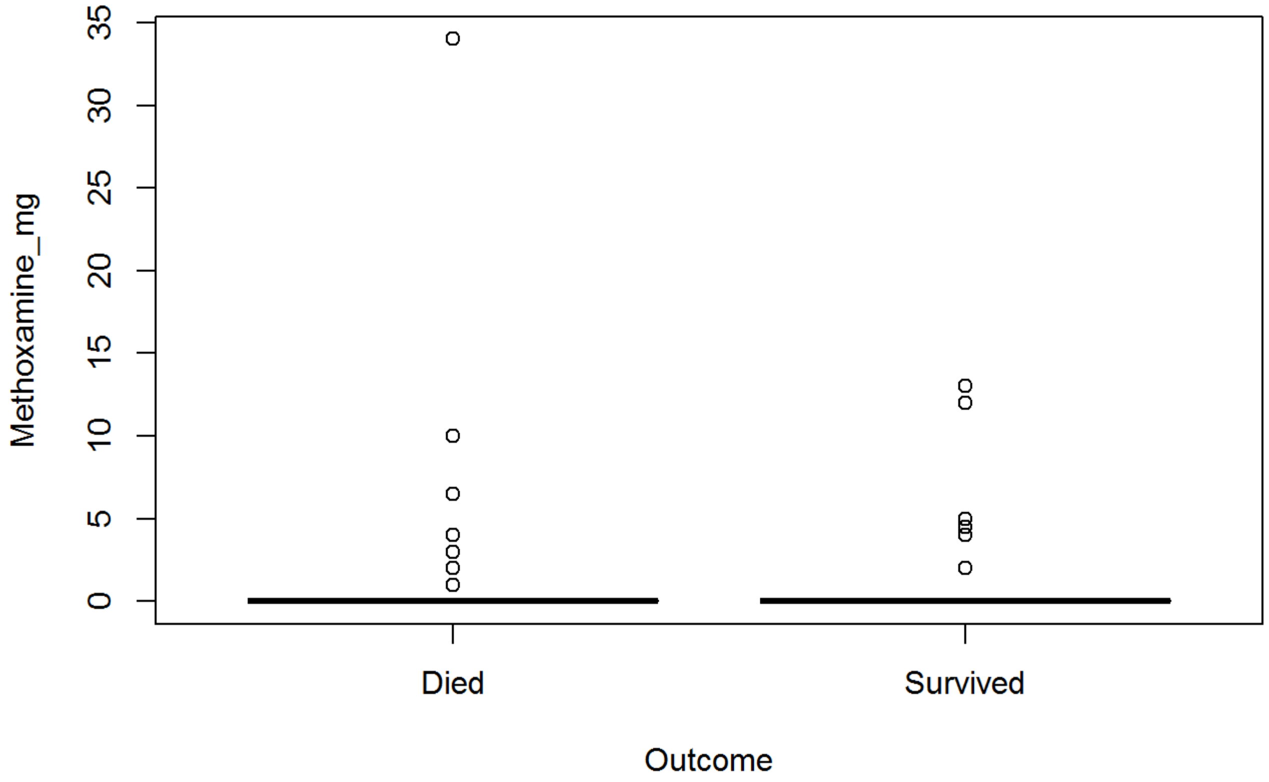
```
## [1] "Atropine_mg"
##
##  Kruskal-Wallis rank sum test
##
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 0.10974, df = 1, p-value = 0.7404
```



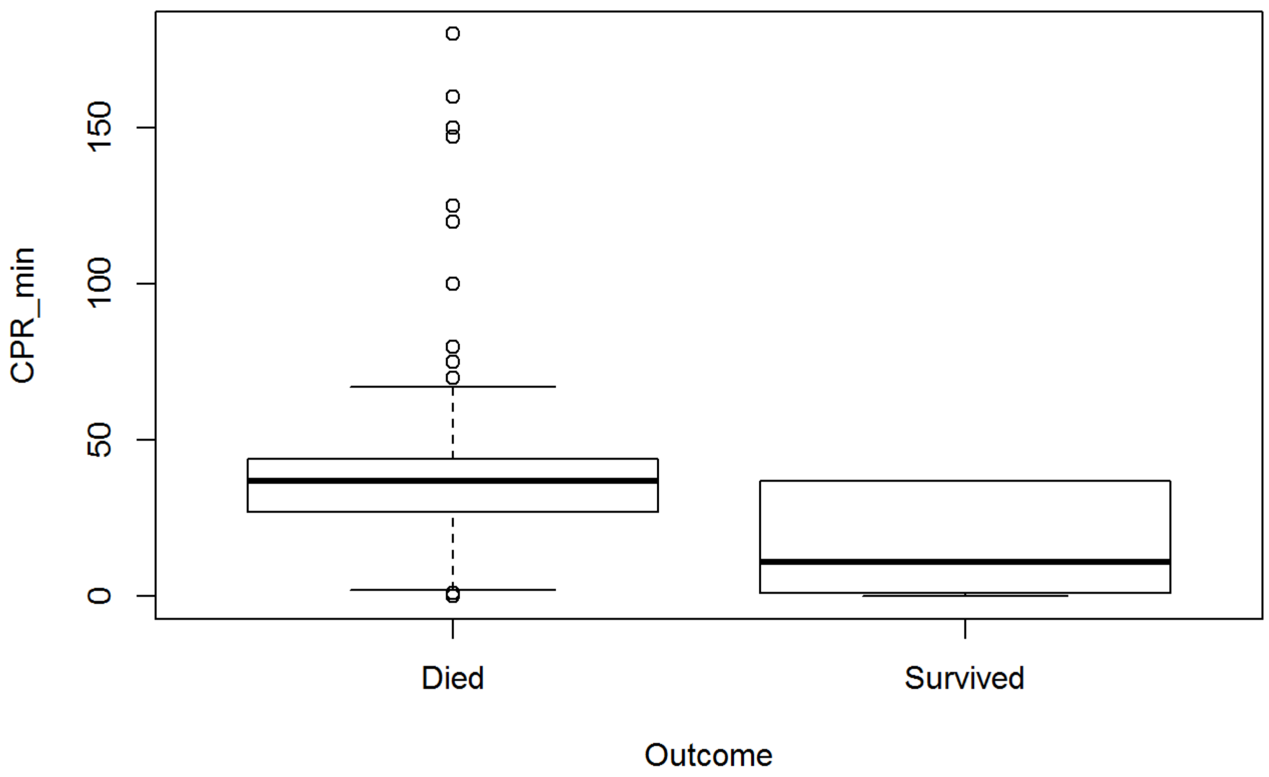
```
## [1] "Amiodarone_g"
##
##  Kruskal-Wallis rank sum test
##
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 0.28115, df = 1, p-value = 0.5959
```



```
## [1] "Ephedrine_mg"
## 
##  Kruskal-Wallis rank sum test
## 
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 1.7919, df = 1, p-value = 0.1807
```



```
## [1] "Methoxamine_mg"
## 
##  Kruskal-Wallis rank sum test
## 
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 0.021686, df = 1, p-value = 0.8829
```



```
## [1] "CPR_min"
## 
##  Kruskal-Wallis rank sum test
## 
## data:  data[, i3] by data$Outcome
## Kruskal-Wallis chi-squared = 34.405, df = 1, p-value = 4.476e-09
```

Export results in csv format

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.