Description_Examples_in_R.R

monca016

Tue Sep 11 15:10:45 2018

```
chooseCRANmirror(graphics=FALSE, ind=1)
knitr::opts_chunk$set(echo = TRUE)

library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.4.4
```

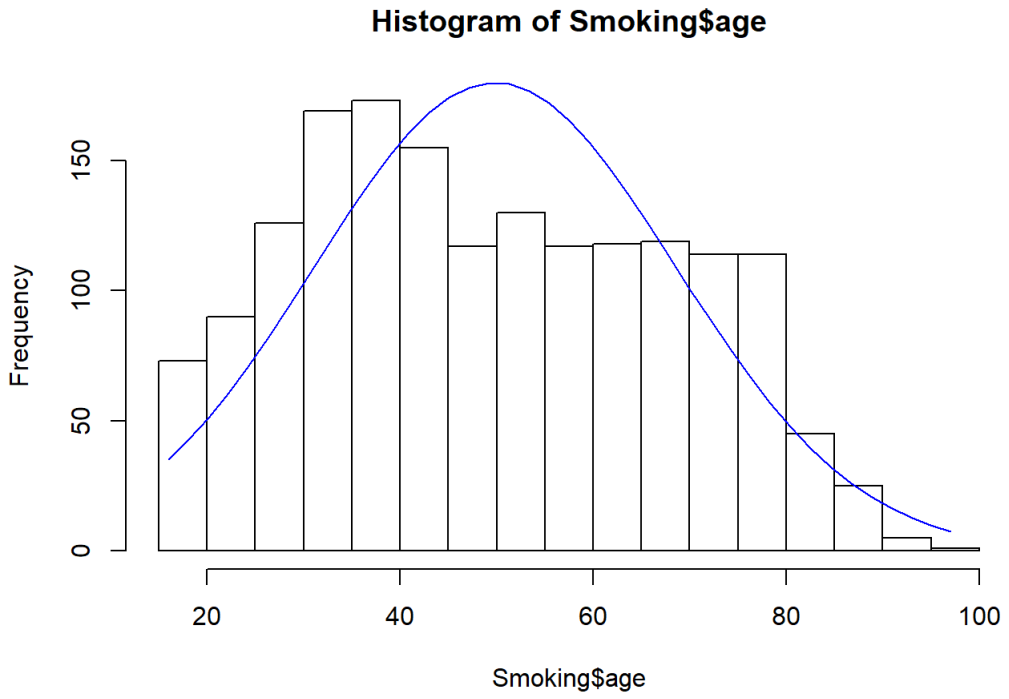
```
Smoking <- read_excel("Smoking.xlsx", na = "NA", col_names = TRUE)

# Some basic descriptive capabilities in the main R package
 # Numerical description
summary(Smoking)
```

```
##      record           sex                 age        maritalStatus
##  Min.   :   1.0   Length:1691        Min.   :16.00   Length:1691
##  1st Qu.: 423.5   Class :character   1st Qu.:34.00   Class :character
##  Median : 846.0   Mode  :character   Median :48.00   Mode  :character
##  Mean   : 846.0                      Mean   :49.84
##  3rd Qu.:1268.5                      3rd Qu.:65.50
##  Max.   :1691.0                      Max.   :97.00
##
##  grossIncome           region             smoke            amtWeekends
##  Length:1691        Length:1691        Length:1691        Min.   : 0.00
##  Class :character   Class :character   Class :character   1st Qu.:10.00
##  Mode  :character   Mode  :character   Mode  :character   Median :15.00
##                                                           Mean   :16.41
##                                                           3rd Qu.:20.00
##                                                           Max.   :60.00
##                                                           NA's   :1270
##   amtWeekdays
##  Min.   : 0.00
##  1st Qu.: 7.00
##  Median :12.00
##  Mean   :13.75
##  3rd Qu.:20.00
##  Max.   :55.00
##  NA's   :1270
```

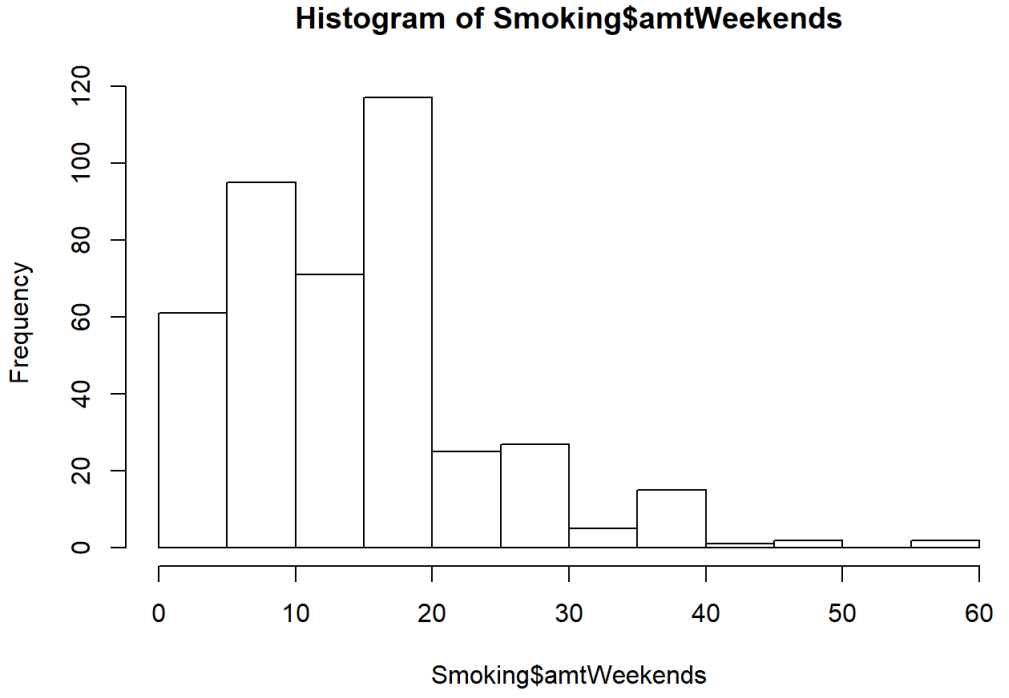
```
mean(Smoking$amtWeekends, na.rm = T) / 2
```

```
## [1] 8.205463
```

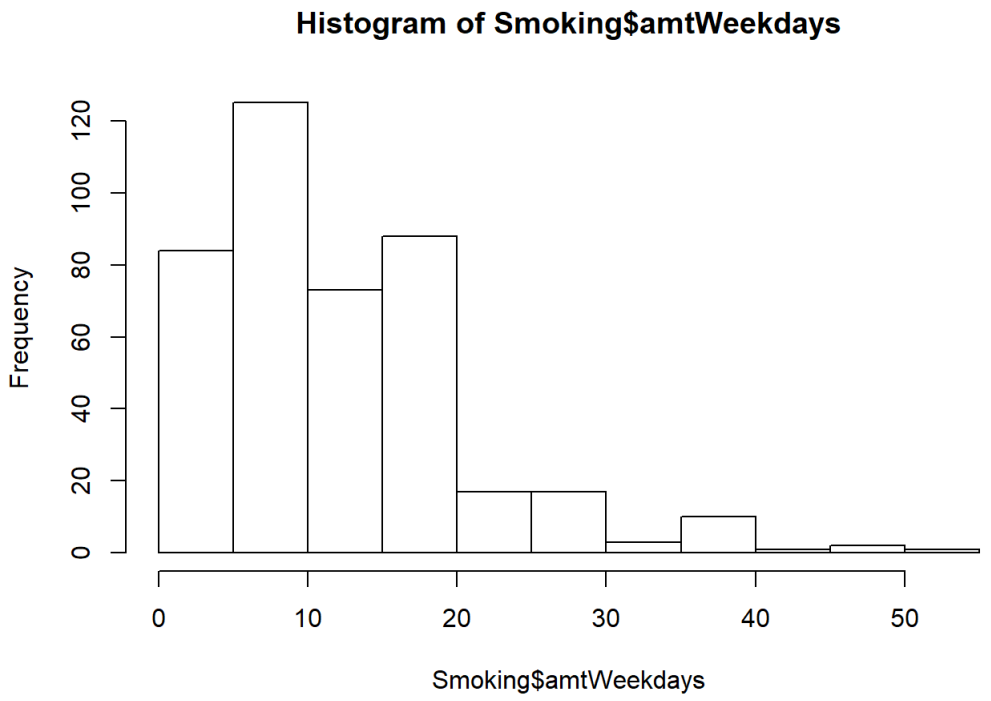
```
hist(Smoking$age)

h <- hist(Smoking$age)

x <- Smoking$age
xfit <- seq(min(x), max(x), length = 40)
yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
yfit <- yfit*diff(h$mids[1:2]*length(x))
lines(xfit, yfit, col = "blue")
```



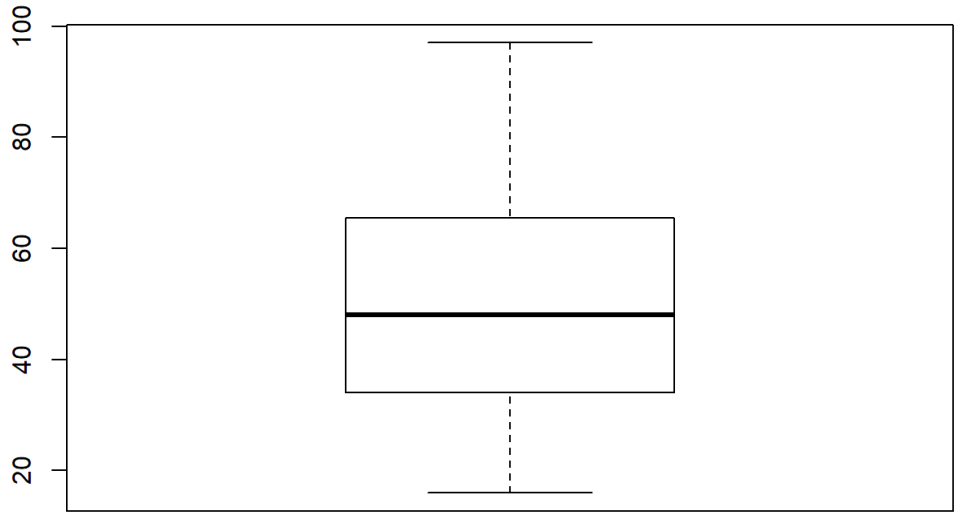

```
# other histograms
hist(Smoking$amtWeekends)
```



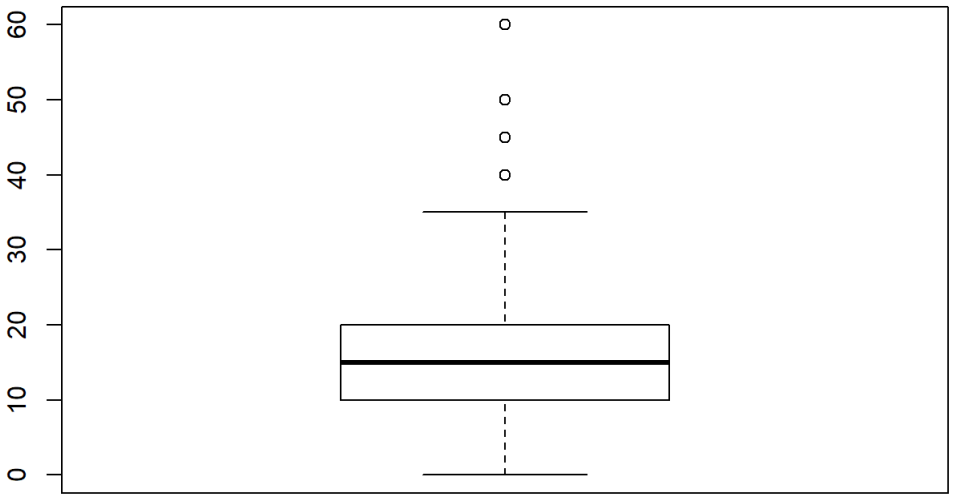

```
hist(Smoking$amtWeekdays)
```



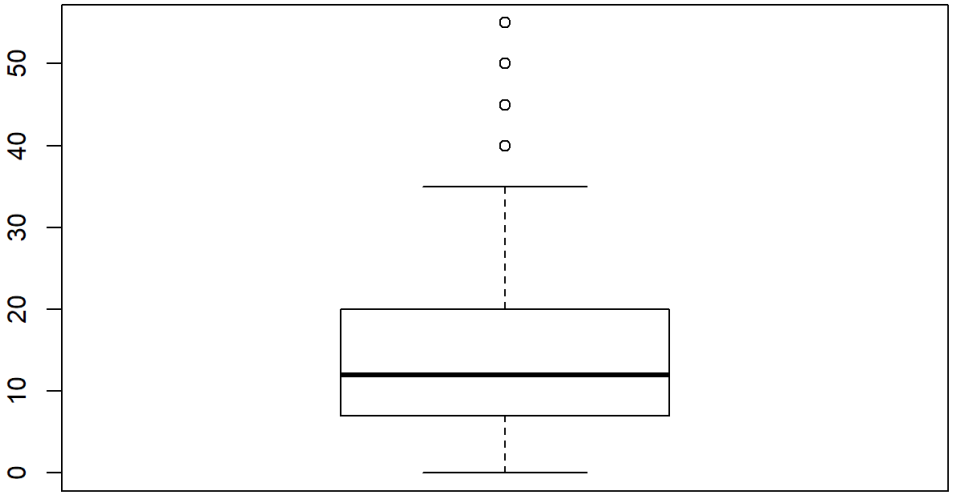

```
# box plots
boxplot(Smoking$age)
```



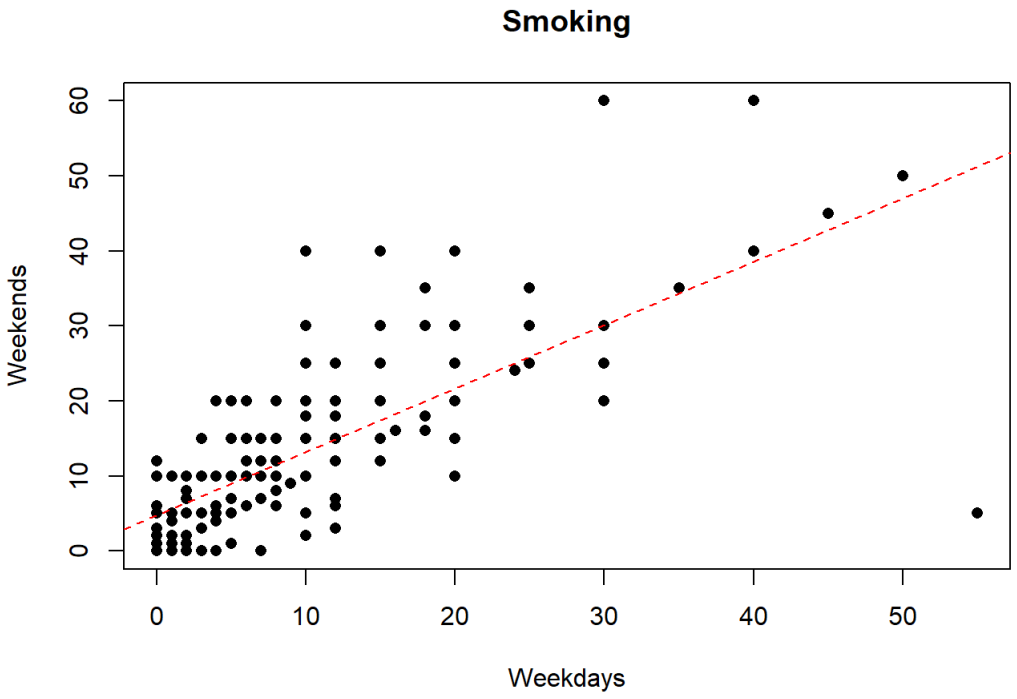

```
boxplot(Smoking$amtWeekends)
```




```
boxplot(Smoking$amtWeekdays)
```




```
# scatter plot
plot(Smoking$amtWeekdays, Smoking$amtWeekends, pch = 16, main = "Smoking",
xlab = "Weekdays", ylab ="Weekends")
abline(lm(Smoking$amtWeekends~Smoking$amtWeekdays), lty = 2, col = "red")
```




```
# bar charts for nominal and ordinal values
  # sex
sexCount <- table(Smoking$sex)
sexCount
```

```
##
## Female   Male
##    965    726
```

```
## Female  Male
##    965   726
barplot(sexCount, ylim = c(0, 1000), main = "Sex")
```

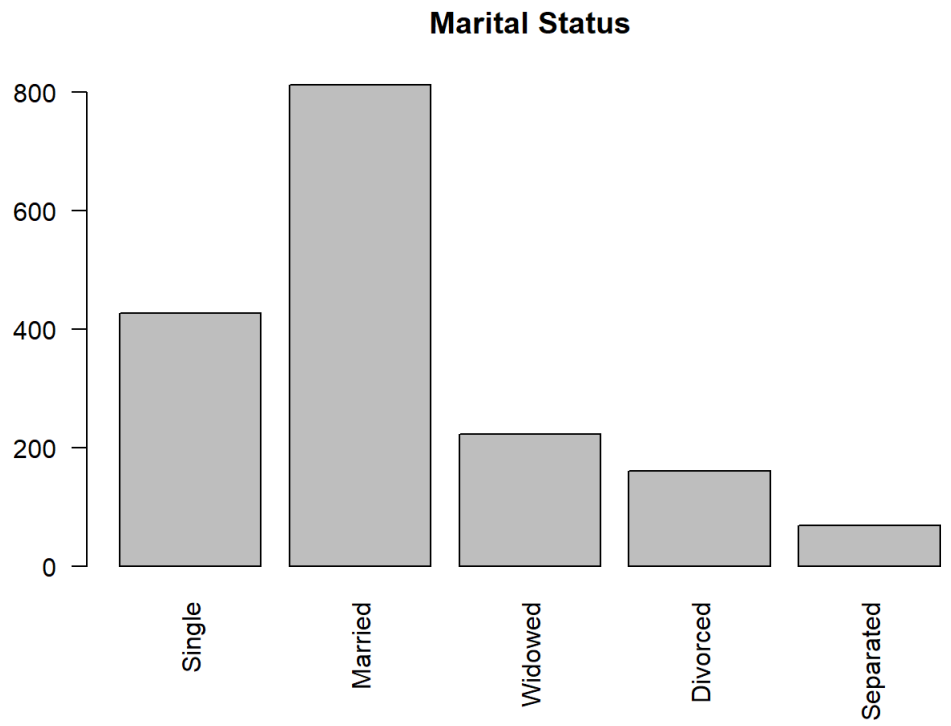

```
  # marital status
table(Smoking$maritalStatus)
```

```
##
##  Divorced   Married Separated    Single   Widowed
##       161       812        68       427       223
```

```
##
##  Divorced   Married Separated    Single   Widowed
##       161       812        68       427       223

    # factor function to reorder the categories
maritalSort <- factor(Smoking$maritalStatus, levels = c("Single", "Married",
"Widowed", "Divorced", "Separated"))
maritalCount <- table(maritalSort)
    # see the re-ordered categories
maritalCount
```

```
## maritalSort
##    Single   Married   Widowed  Divorced Separated
##       427       812       223       161        68
```

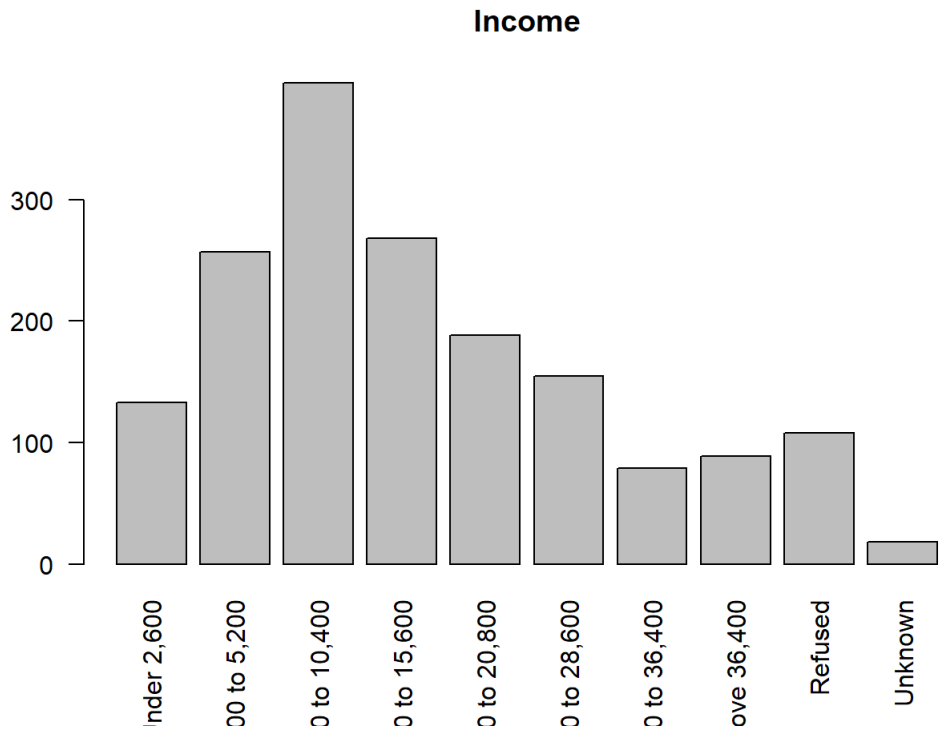
```
## maritalSort
##  Single   Married   Widowed  Divorced Separated
##     427       812       223       161        68
    # bar chart, las = 2 to make x-axis labels vertical
barplot(maritalCount, main = "Marital Status", las = 2)
```




```
#  income
incomeSort <- factor(Smoking$grossIncome, levels = c("Under 2,600", "2,600 to 5,200", "5,200 to 10,400", "10,400 to 15,60
0", "15,600 to 20,800", "20,800 to 28,600", "28,600 to 36,400", "Above 36,400", "Refused", "Unknown"))
incomeCount <- table(incomeSort)
incomeCount
```

```
## incomeSort
##      Under 2,600   2,600 to 5,200  5,200 to 10,400 10,400 to 15,600
##              133              257              396              268
## 15,600 to 20,800 20,800 to 28,600 28,600 to 36,400     Above 36,400
##              188              155               79               89
##          Refused          Unknown
##              108               18
```

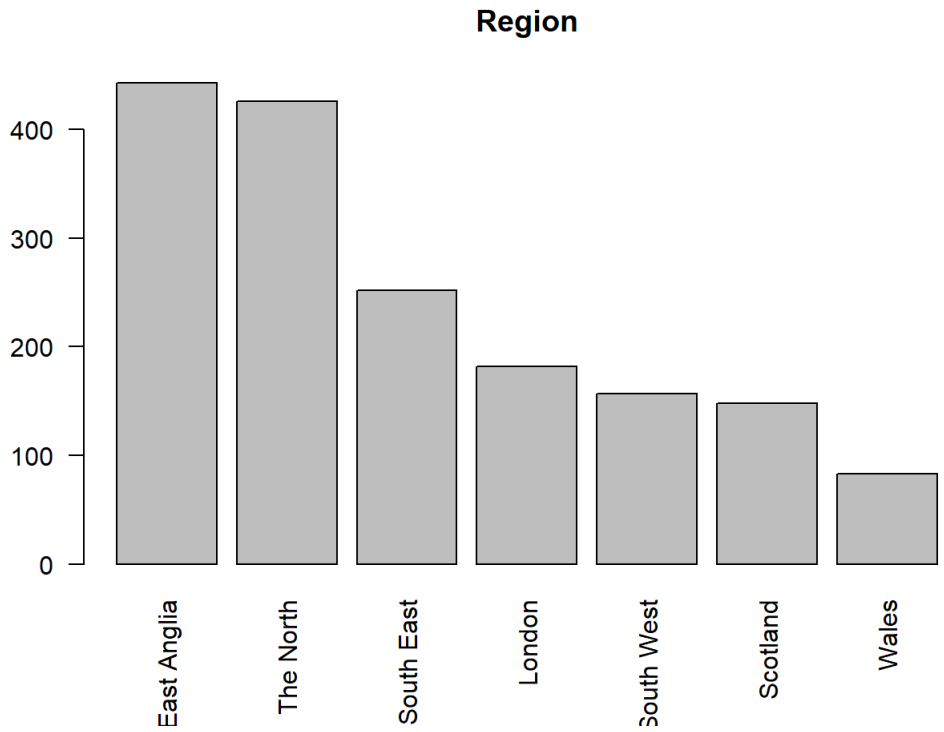
```
## incomeSort
##      Under 2,600   2,600 to 5,200  5,200 to 10,400 10,400 to 15,600 15,600 to 20,800 20,800 to 28,600
##              133              257              396              268              188              155
##      28,600 to 36,400     Above 36,400          Refused          Unknown
##                    79               89              108               18

barplot(incomeCount, main = "Income", las = 2)
```




```
    # region
regionSort <- factor(Smoking$region, levels = c("Midlands & East Anglia", "The North", "South East", "London", "South Wes
t", "Scotland", "Wales"))
regionCount <- table(regionSort)
regionCount
```

```
## regionSort
## Midlands & East Anglia             The North            South East
##                    443                   426                   252
##                 London            South West              Scotland
##                    182                   157                   148
##                  Wales
##                     83
```

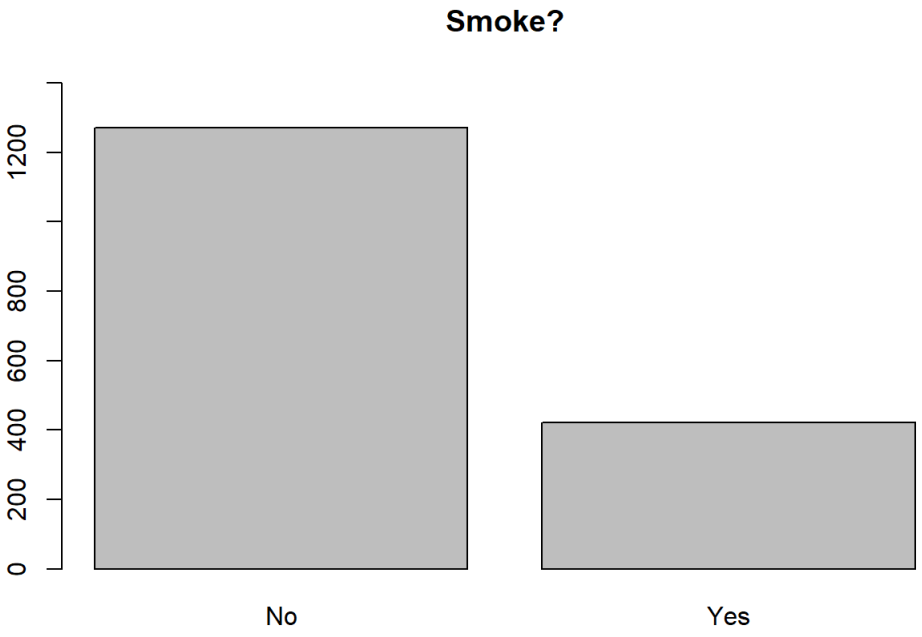
```
## regionSort
## Midlands & East Anglia             The North            South East                London
##                    443                   426                   252                   182
##             South West              Scotland                 Wales
##                    157                   148                    83

barplot(regionCount, main = "Region", las = 2)
```




```
    # smoke: yes or no
smokeCount <- table(Smoking$smoke)
smokeCount
```

```
##
##   No  Yes
## 1270  421
```

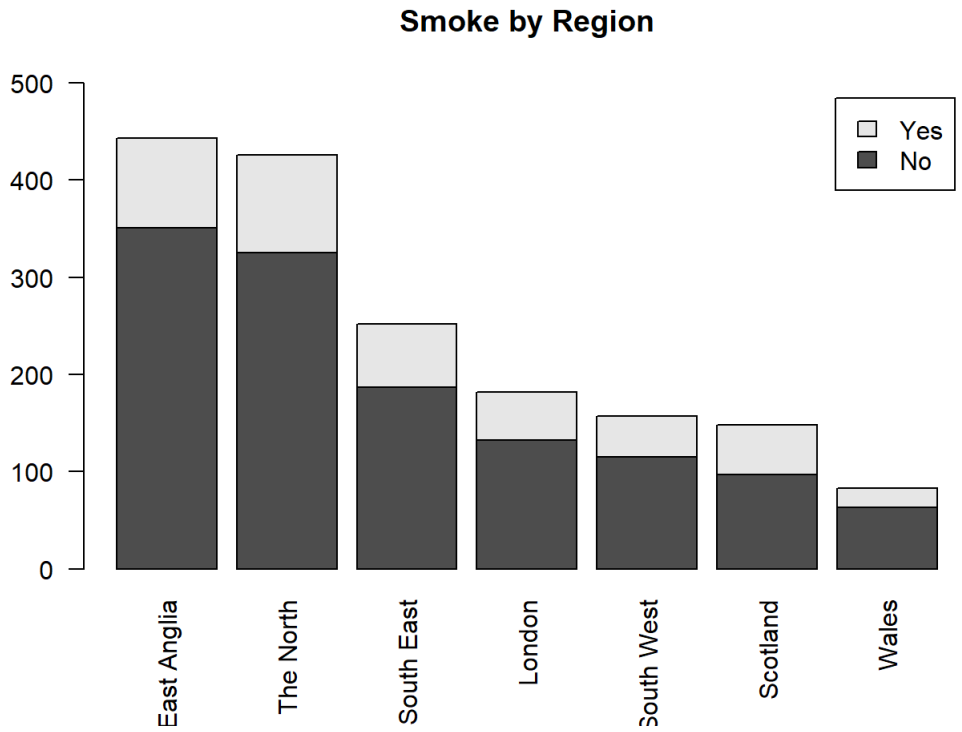
```
##
##   No  Yes
## 1270  421
barplot(smokeCount, ylim = c(0, 1400), main = "Smoke?")
```



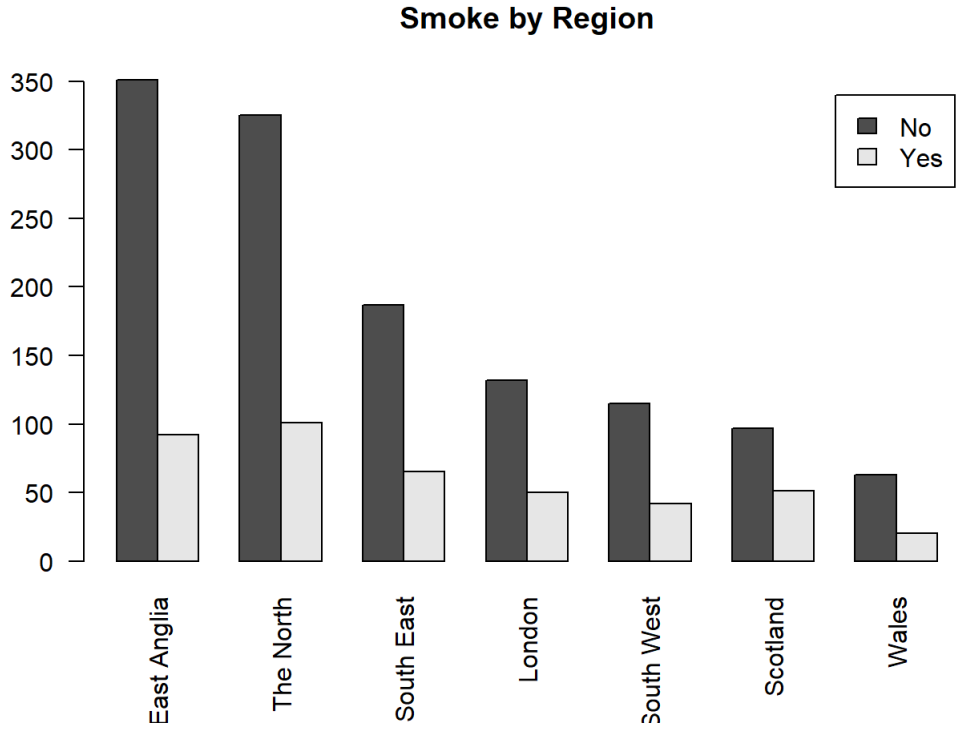

```
#  stacked bar chart: smoke by region
smoke_regionCount <- table(Smoking$smoke, regionSort)
smoke_regionCount
```

```
##      regionSort
##       Midlands & East Anglia The North South East London South West
##   No                     351       325        187    132        115
##   Yes                     92       101         65     50         42
##      regionSort
##       Scotland Wales
##   No        97    63
##   Yes       51    20
```

```
##      regionSort
##        Midlands & East Anglia The North South East London South West Scotland Wales
##    No                    351       325        187    132        115       97    63
##    Yes                    92       101         65     50         42       51    20

barplot(smoke_regionCount, main = "Smoke by Region", las = 2, ylim = c(0, 500), legend = rownames(smoke_regionCount))
```




```
    # grouped bar chart: smoke by region
barplot(smoke_regionCount, main = "Smoke by Region", las = 2, legend = row.names(smoke_regionCount), beside = T)
```




```
marathon <- read.table("marathon.csv", header = TRUE, sep = ",", strip.white = TRUE)
hist(marathon$Time, breaks = 10)

# Some other descriptive capabilites in other packages
  # Numerical description
    #install pastecs package first
install.packages("pastecs")
```

```
## package 'pastecs' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\monca016\AppData\Local\Temp\RtmpCiM7KM\downloaded_packages
```

```
library(pastecs)
```

```
## Warning: package 'pastecs' was built under R version 3.4.4
```

```
  # useful function within pastecs package
stat.desc(Smoking[, c('age', 'amtWeekends', 'amtWeekdays')])
```

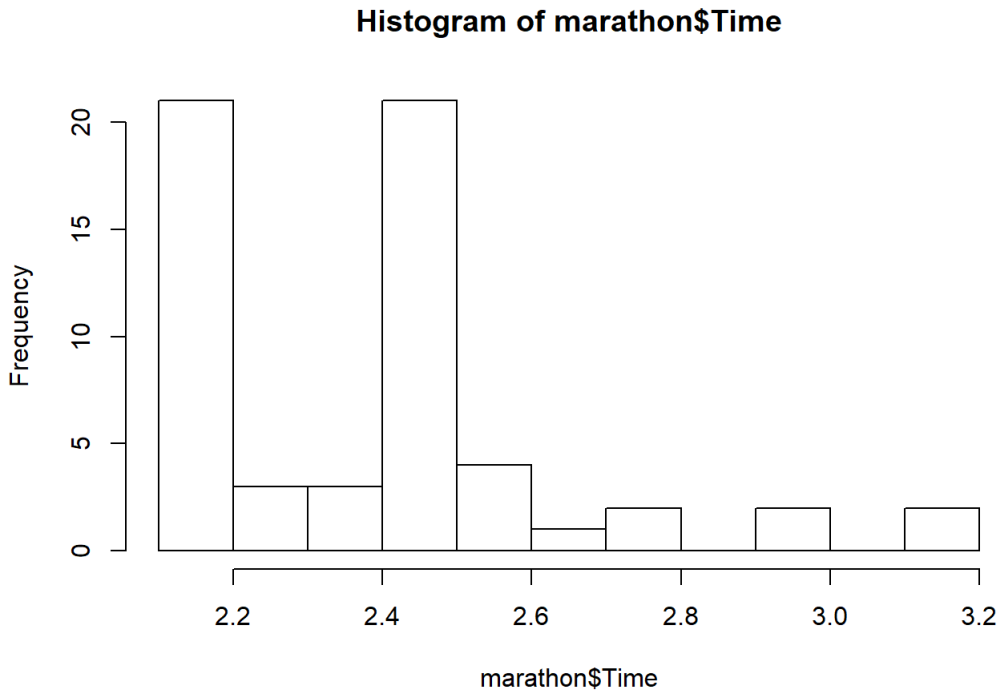
```
##                      age  amtWeekends  amtWeekdays
## nbr.val      1.691000e+03  421.0000000  421.0000000
## nbr.null     0.000000e+00    6.0000000   16.0000000
## nbr.na       0.000000e+00 1270.0000000 1270.0000000
## min          1.600000e+01    0.0000000    0.0000000
## max          9.700000e+01   60.0000000   55.0000000
## range        8.100000e+01   60.0000000   55.0000000
## sum          8.427300e+04 6909.0000000 5789.0000000
## median       4.800000e+01   15.0000000   12.0000000
## mean         4.983619e+01   16.4109264   13.7505938
## SE.mean      4.556431e-01    0.4821547    0.4575574
## CI.mean.0.95 8.936841e-01    0.9477370    0.8993877
## var          3.510696e+02   97.8712137   88.1400294
## std.dev      1.873685e+01    9.8929881    9.3882921
## coef.var     3.759688e-01    0.6028294    0.6827554
```

```
##                      age  amtWeekends  amtWeekdays
## nbr.val      1.691000e+03  421.0000000  421.0000000
## nbr.null     0.000000e+00    6.0000000   16.0000000
## nbr.na       0.000000e+00 1270.0000000 1270.0000000
## min          1.600000e+01    0.0000000    0.0000000
## max          9.700000e+01   60.0000000   55.0000000
## range        8.100000e+01   60.0000000   55.0000000
## sum          8.427300e+04 6909.0000000 5789.0000000
## median       4.800000e+01   15.0000000   12.0000000
## mean         4.983619e+01   16.4109264   13.7505938
## SE.mean      4.556431e-01    0.4821547    0.4575574
## CI.mean.0.95 8.936841e-01    0.9477370    0.8993877
## var          3.510696e+02   97.8712137   88.1400294
## std.dev      1.873685e+01    9.8929881    9.3882921
## coef.var     3.759688e-01    0.6028294    0.6827554

  # Graphical description
    # install ggplot2 package first
    # ggplot2 is a popular package with a lot of capabilities for creating better looking graphics
install.packages("ggplot2")
```

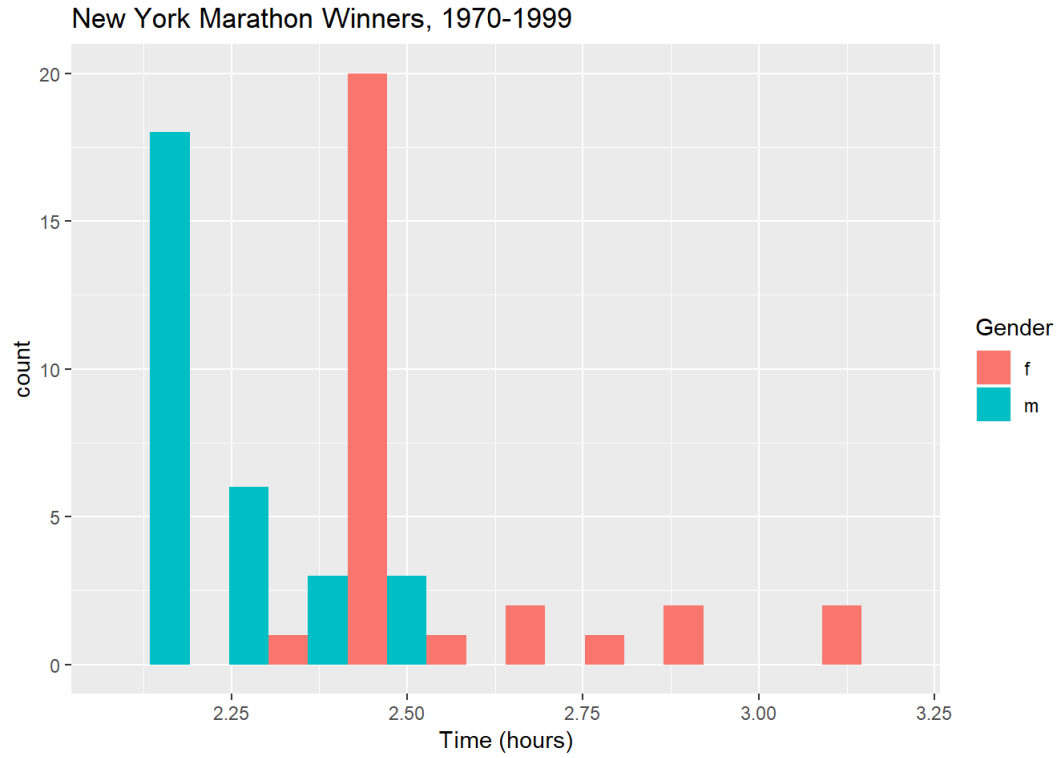
```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\monca016\AppData\Local\Temp\RtmpCiM7KM\downloaded_packages
```

```
library(ggplot2)
```

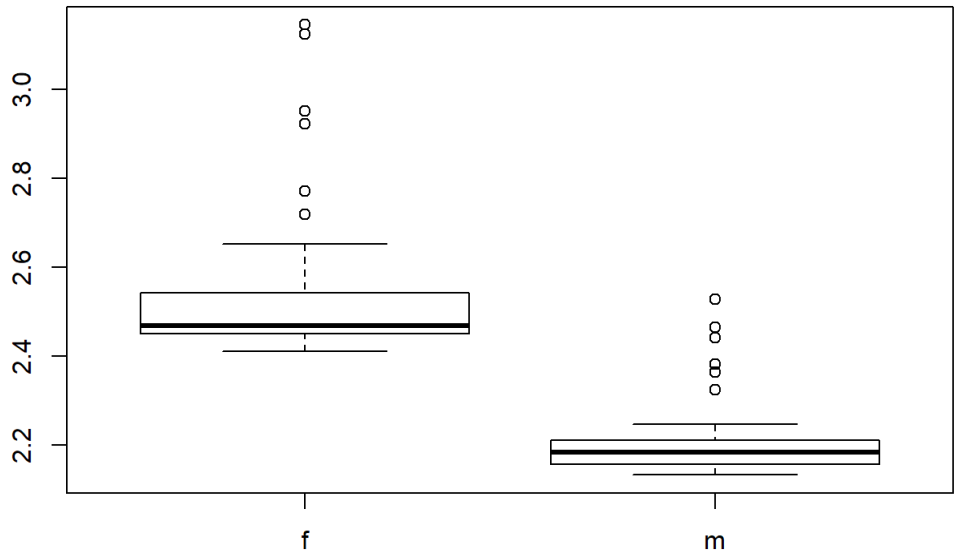
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```



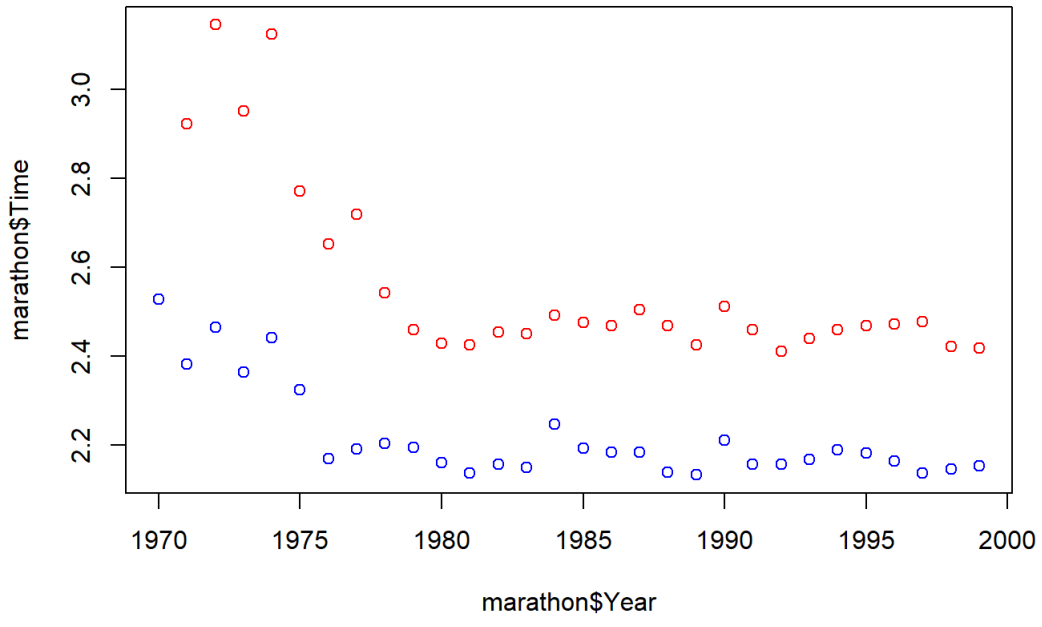

```
  # ggplot is a function within the popular ggplot2 package
  # ggplot() is used to construct a plot incrementally, using the + operator to add layers to the existing ggplot object
    # Histogram of Times by Gender
Gender <- marathon$Gender
ggplot(marathon, aes(x = marathon$Time, fill = Gender)) +
  geom_histogram(position = "dodge", bins = 10) + xlab("Time (hours)") +
  ggtitle("New York Marathon Winners, 1970-1999")
```




```
  # Returning to the techniques in the main R package
    # Scatter Plot of Times by Gender, Year
boxplot(marathon$Time~marathon$Gender)
```




```
plot(marathon$Year, marathon$Time, col = c("red", "blue")[marathon$Gender])
```




```
  # line chart
    # reorder Marathon data frame by year
marathon <- marathon[order(marathon$Year), ]
marathon
```

```
##    Year Gender    Time
## 41 1970      m 2.52722
## 42 1971      m 2.38167
## 51 1971      f 2.92278
## 43 1972      m 2.46444
## 52 1972      f 3.14472
## 44 1973      m 2.36500
## 53 1973      f 2.95194
## 45 1974      m 2.44167
## 54 1974      f 3.12472
## 46 1975      m 2.32417
## 55 1975      f 2.77056
## 47 1976      m 2.16944
## 56 1976      f 2.65306
## 48 1977      m 2.19111
## 57 1977      f 2.71944
## 49 1978      m 2.20333
## 58 1978      f 2.54167
## 50 1979      m 2.19500
## 59 1979      f 2.45917
## 1  1980      m 2.16139
## 21 1980      f 2.42833
## 2  1981      m 2.13694
## 22 1981      f 2.42472
## 3  1982      m 2.15806
## 23 1982      f 2.45389
## 4  1983      m 2.14972
## 24 1983      f 2.45000
## 5  1984      m 2.24806
## 25 1984      f 2.49167
## 6  1985      m 2.19278
## 26 1985      f 2.47611
## 7  1986      m 2.18500
## 27 1986      f 2.46833
## 8  1987      m 2.18361
## 28 1987      f 2.50472
## 9  1988      m 2.13889
## 29 1988      f 2.46861
## 10 1989      m 2.13361
## 30 1989      f 2.42500
## 11 1990      m 2.21083
## 31 1990      f 2.51250
## 12 1991      m 2.15778
## 32 1991      f 2.45889
## 13 1992      m 2.15806
## 33 1992      f 2.41111
## 14 1993      m 2.16778
## 34 1993      f 2.44000
## 15 1994      m 2.18917
## 35 1994      f 2.46028
## 16 1995      m 2.18333
## 36 1995      f 2.46833
## 17 1996      m 2.16500
## 37 1996      f 2.47167
## 18 1997      m 2.13667
## 38 1997      f 2.47833
## 19 1998      m 2.14583
## 39 1998      f 2.42139
## 20 1999      m 2.15389
## 40 1999      f 2.41833
```

```
##    Year Gender    Time
## 41 1970      m 2.52722
## 42 1971      m 2.38167
## 51 1971      f 2.92278
## 43 1972      m 2.46444
## 52 1972      f 3.14472
```

```
  # plot set up
plot(marathon$Year, marathon$Time, type ="n", col = c("red", "blue") [marathon$Gender], xlab = "Year", ylab = "Running Tim
e (hours)")
  # add lines and points
LineF <- subset(marathon, marathon$Gender == "f")
LineM <- subset(marathon, marathon$Gender == "m")
lines(LineF$Year, LineF$Time, type = "b", col = "red", pch = 22)
lines(LineM$Year, LineM$Time, type = "b", col = "red", pch = 21, lty = 2)
  # add legend and tile
title("Marathon Times")
legend(1990, 3, c("Male", "Female"), cex = .8, col = c("blue", "red"), pch = 21:22, lty = 2:1, title = "Gender")
```