Exploratory Data Analysis (EDA)

Statsomat.com

18 April 2021

Basic Information

Automatic statistics for the file:

File
HolzingerSwineford1939.csv

Your selection for the encoding: Auto
Your selection for the decimal character: Auto
Observations (rows with at least one non-missing value): 301
Variables (columns with at least one non-missing value): 16
Variables considered continuous: 11

Variables considered continuous
V1
id
x1
x2
x3
x4
x5
x6
x7
x8
x9

Variables considered categorical: 5

Variables considered categorical
sex
ageyr
school
grade
agemo

Results for Numerical Variables

Descriptive Statistics

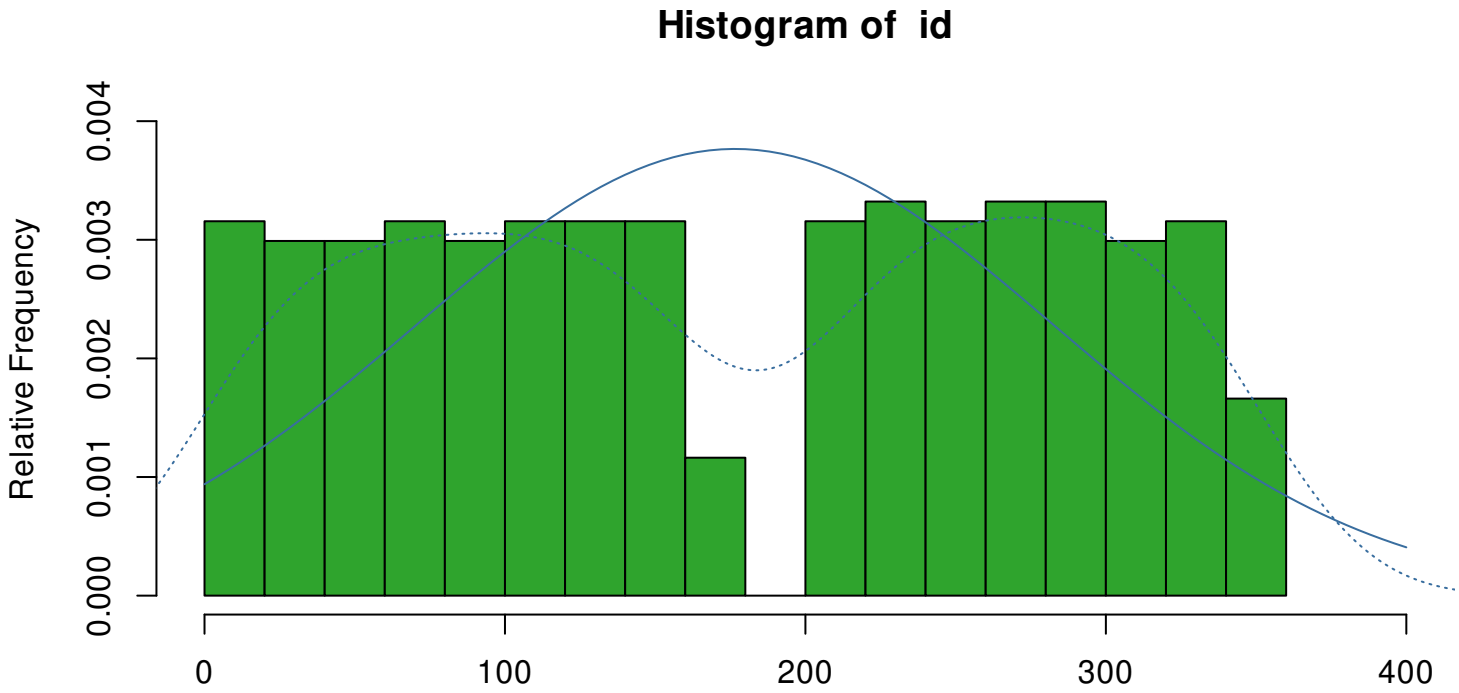
Variables are sorted alphabetically. Missings are omitted in the stats. CV only for positive variables.

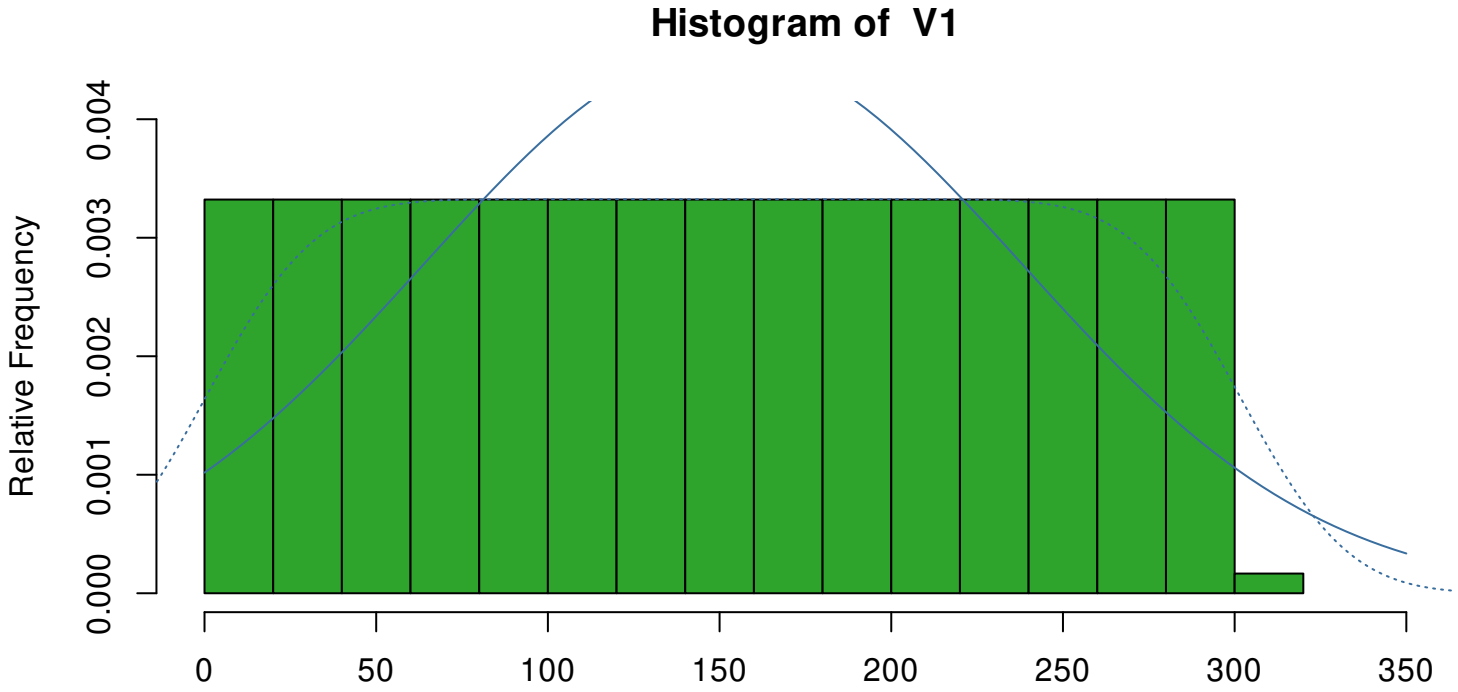
Variable	N Obs	N Missing	N Valid	% Complete	N Unique	Mean	SD	Median	MAD	MIN	MAX	Skewness	Kurtosis	CV
id	301	0	301	100	301	176.55	105.94	163.00	140.85	1.00	351.00	-0.01	-1.36	0.6
V1	301	0	301	100	301	151.00	87.04	151.00	111.19	1.00	301.00	0.00	-1.21	0.58
x1	301	0	301	100	35	4.94	1.17	5.00	1.24	0.67	8.50	-0.25	0.31	0.24
x2	301	0	301	100	25	6.09	1.18	6.00	1.11	2.25	9.25	0.47	0.33	0.19
x3	301	0	301	100	35	2.25	1.13	2.12	1.30	0.25	4.50	0.38	-0.91	0.5
x4	301	0	301	100	20	3.06	1.16	3.00	0.99	0.00	6.33	0.27	0.08	-
x5	301	0	301	100	25	4.34	1.29	4.50	1.48	1.00	7.00	-0.35	-0.55	0.3
x6	301	0	301	100	40	2.19	1.10	2.00	1.06	0.14	6.14	0.86	0.82	0.5
x7	301	0	301	100	97	4.19	1.09	4.09	1.10	1.30	7.43	0.25	-0.31	0.26
x8	301	0	301	100	84	5.53	1.01	5.50	0.96	3.05	10.00	0.53	1.17	0.18
x9	301	0	301	100	129	5.37	1.01	5.42	0.99	2.78	9.25	0.20	0.29	0.19

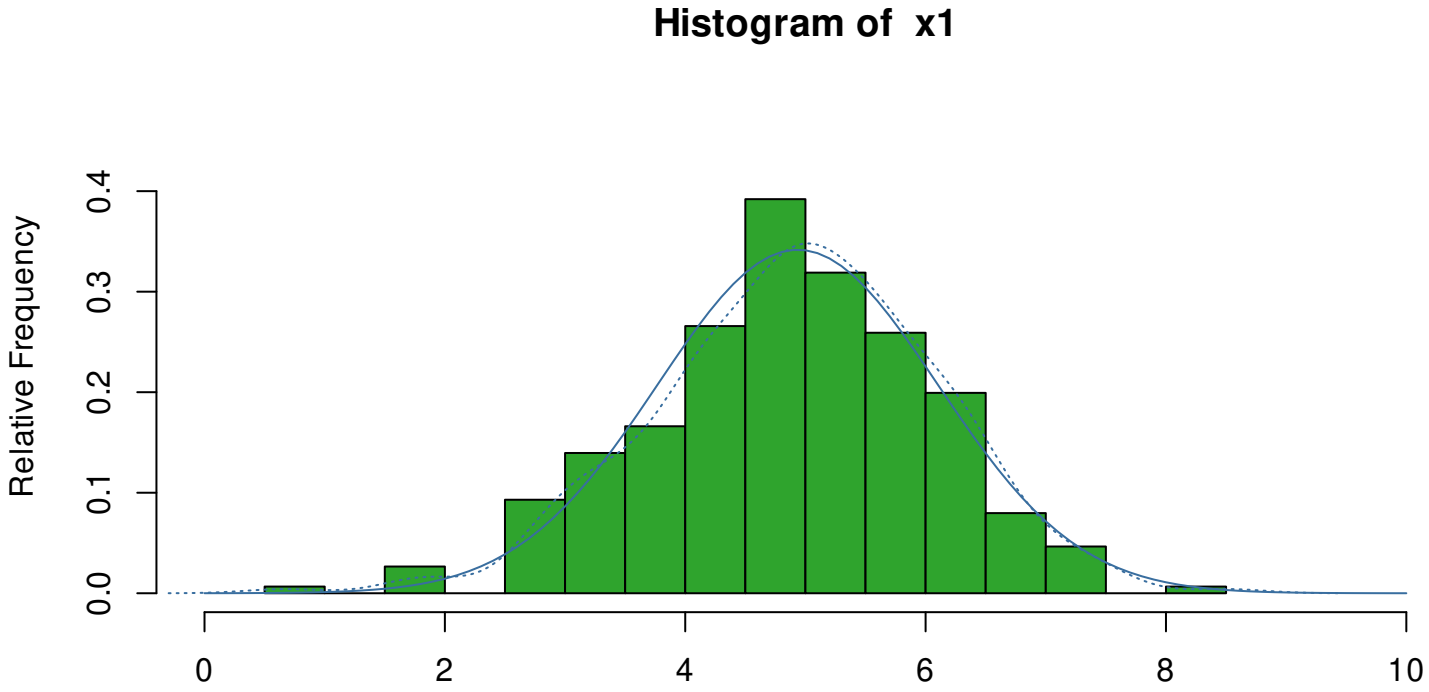
Graphics

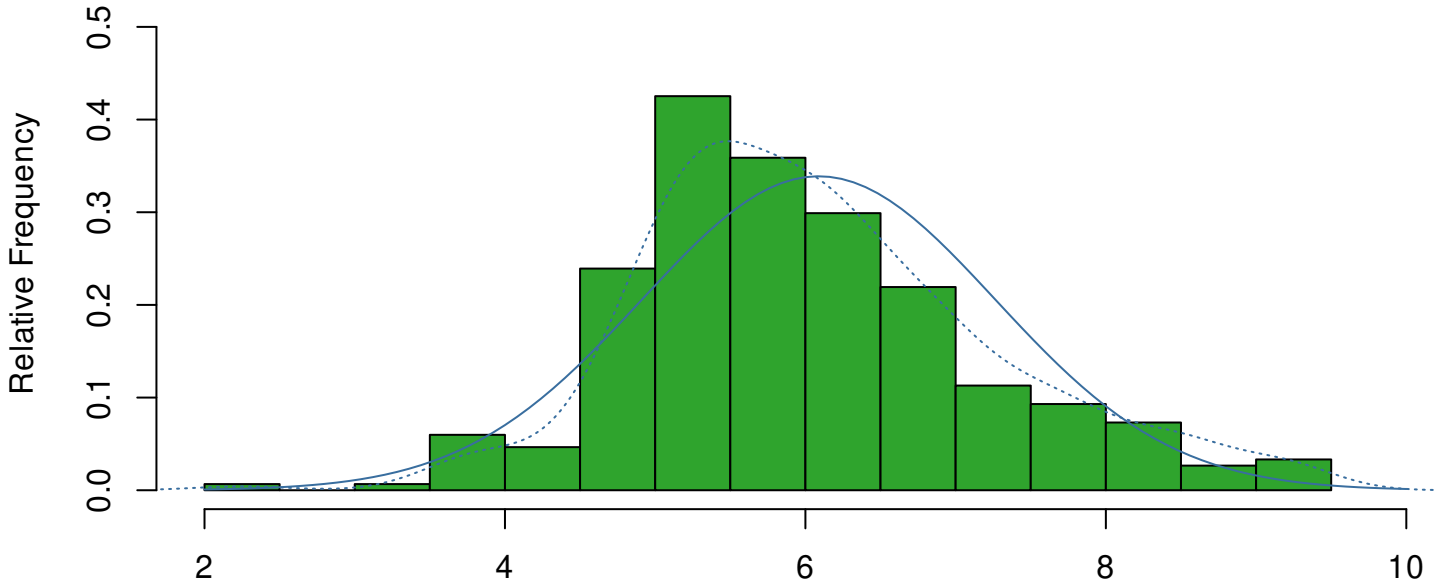
Histograms

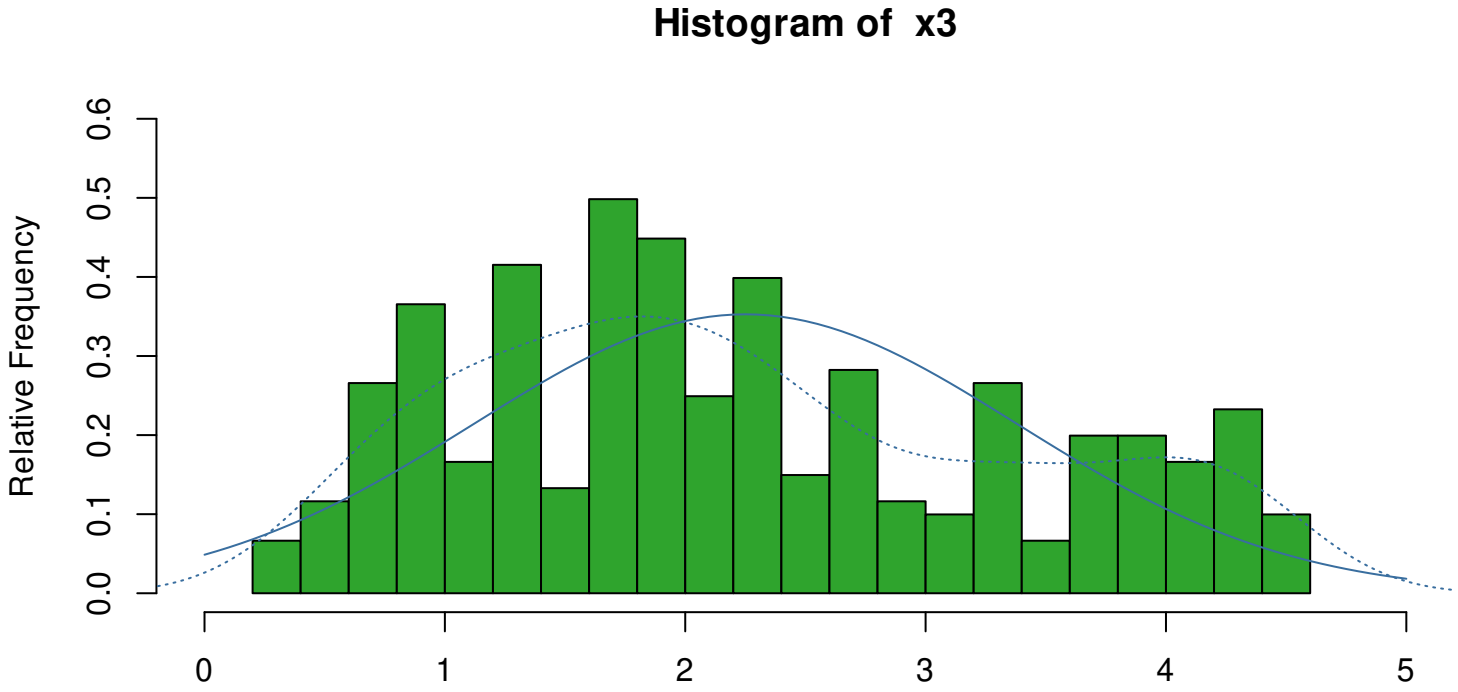
One Relative Frequency Histogram per page for each variable. Variables are sorted alphabetically. The blue line represents the normal density approximation. The blue dotted line represents a special kernel density approximation.

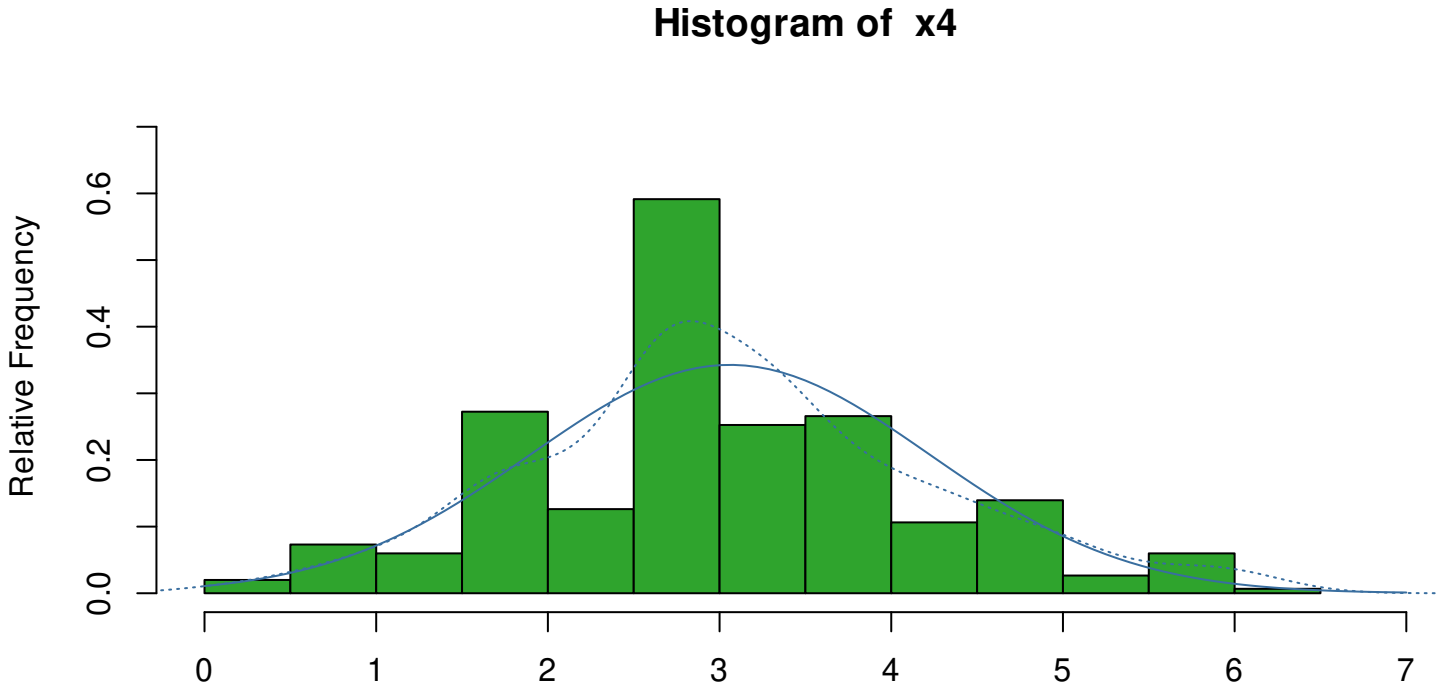


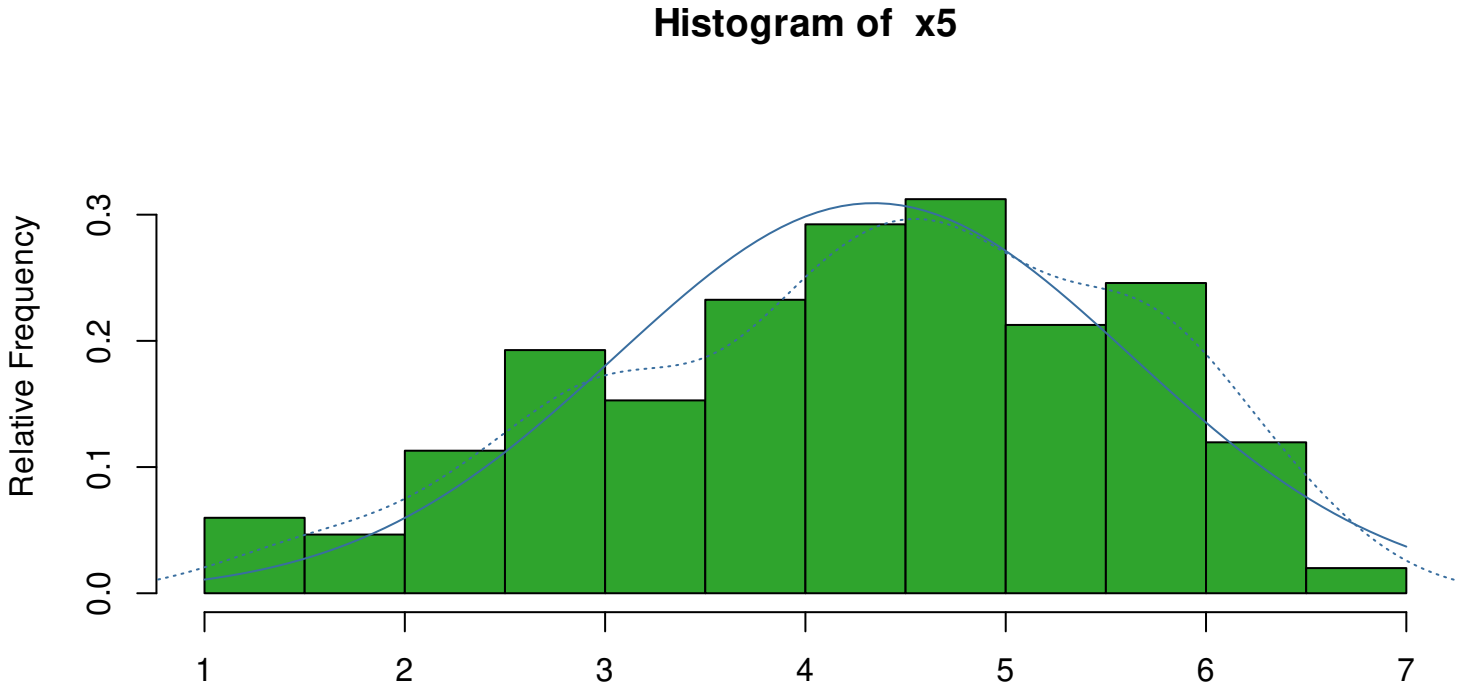

Histogram of V1
Relative Frequency
0.000
0.001
0.002
0.003
0.004
0
50
100
150
200
250
300
350

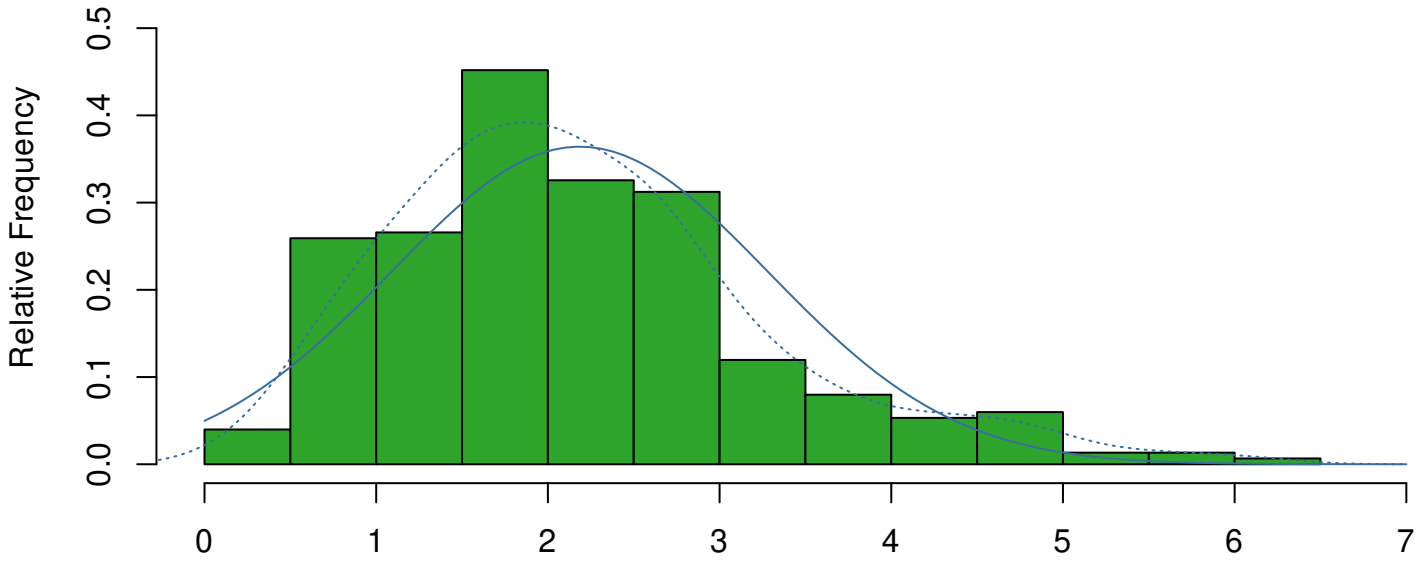

Histogram of x1
Relative Frequency
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10

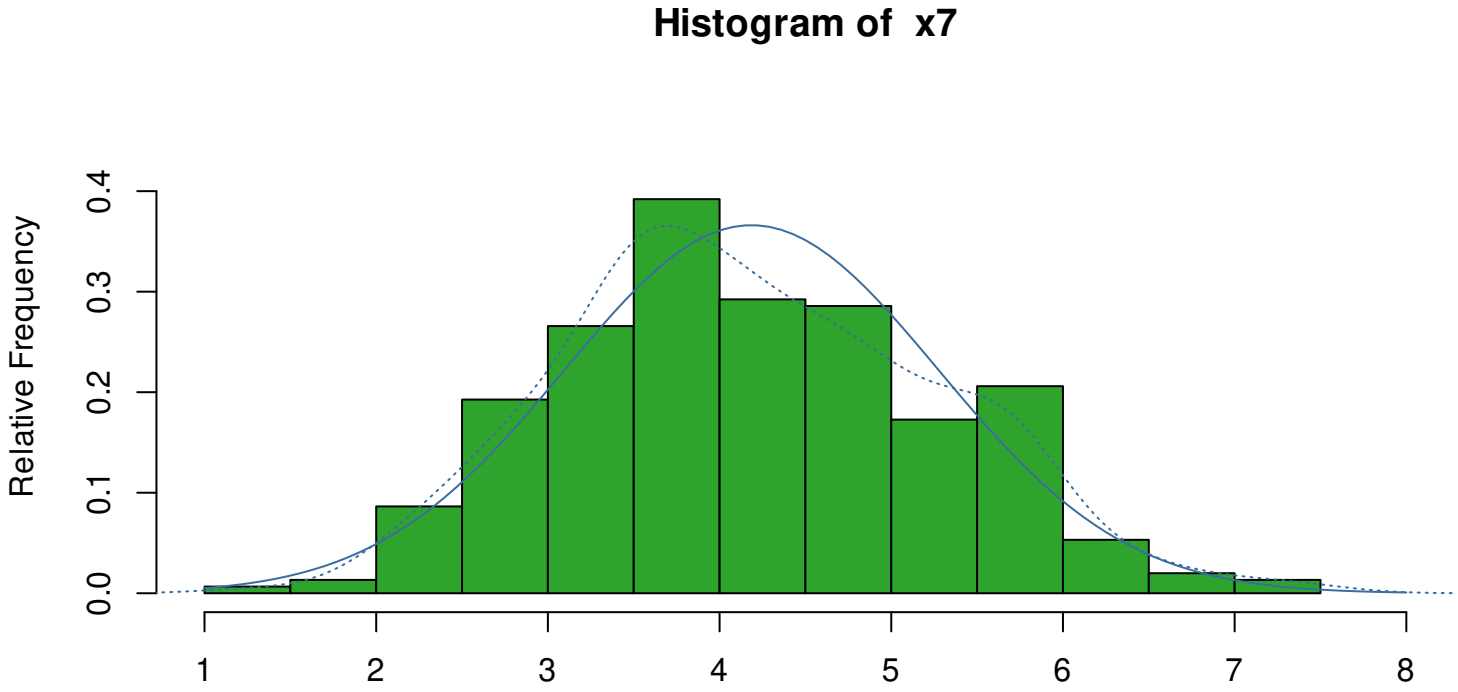

Histogram of x2
Relative Frequency
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10

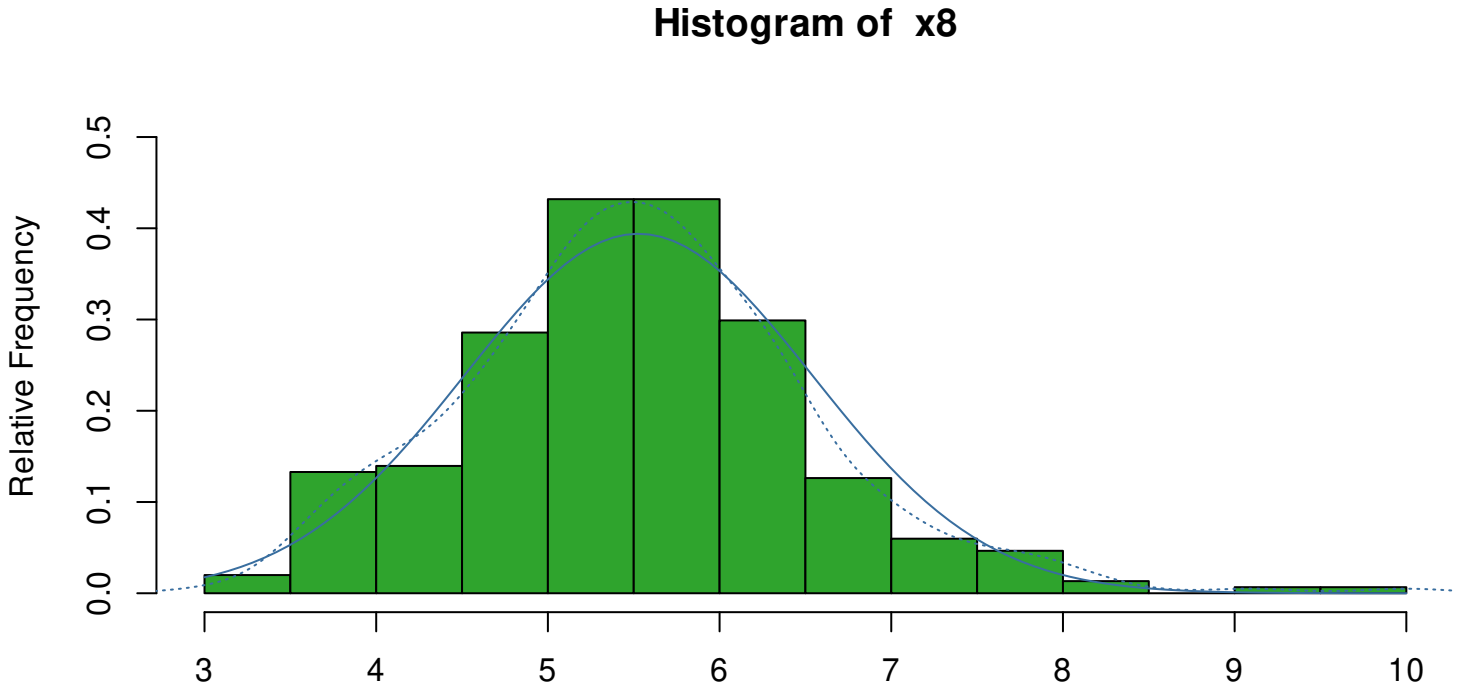

Histogram of x3
Relative Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
5

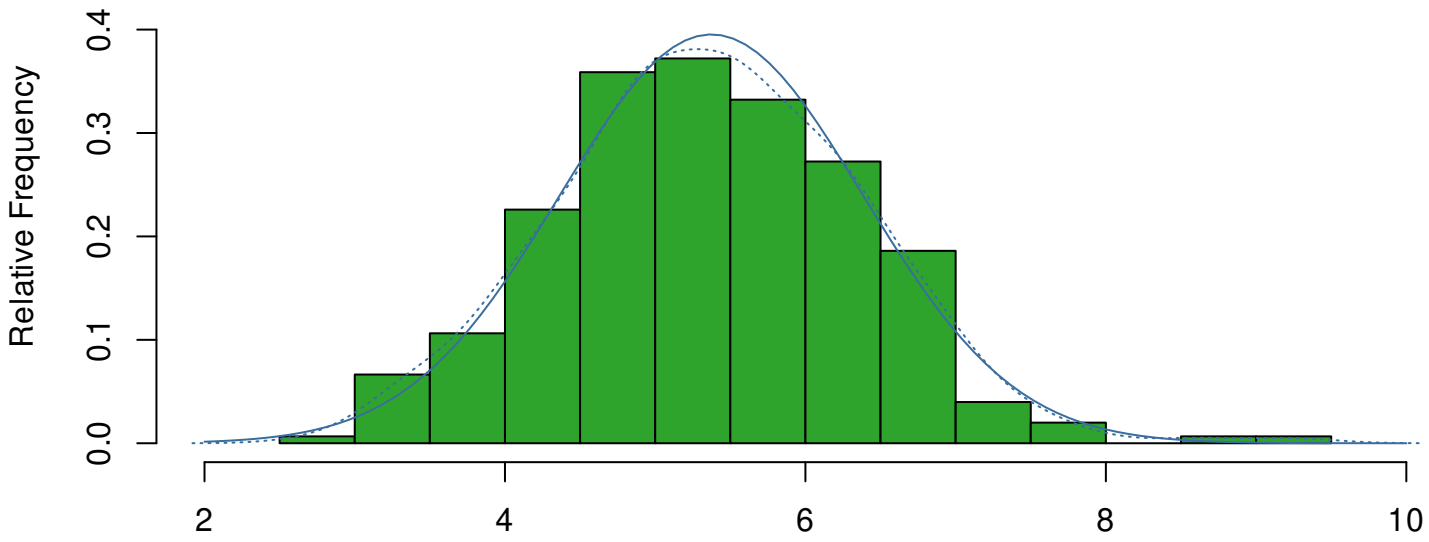

Histogram of x4
Relative Frequency
0.0
0.2
0.4
0.6
0
1
2
3
4
5
6
7


Histogram of x5
Relative Frequency
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7


Histogram of x6
Relative Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
7

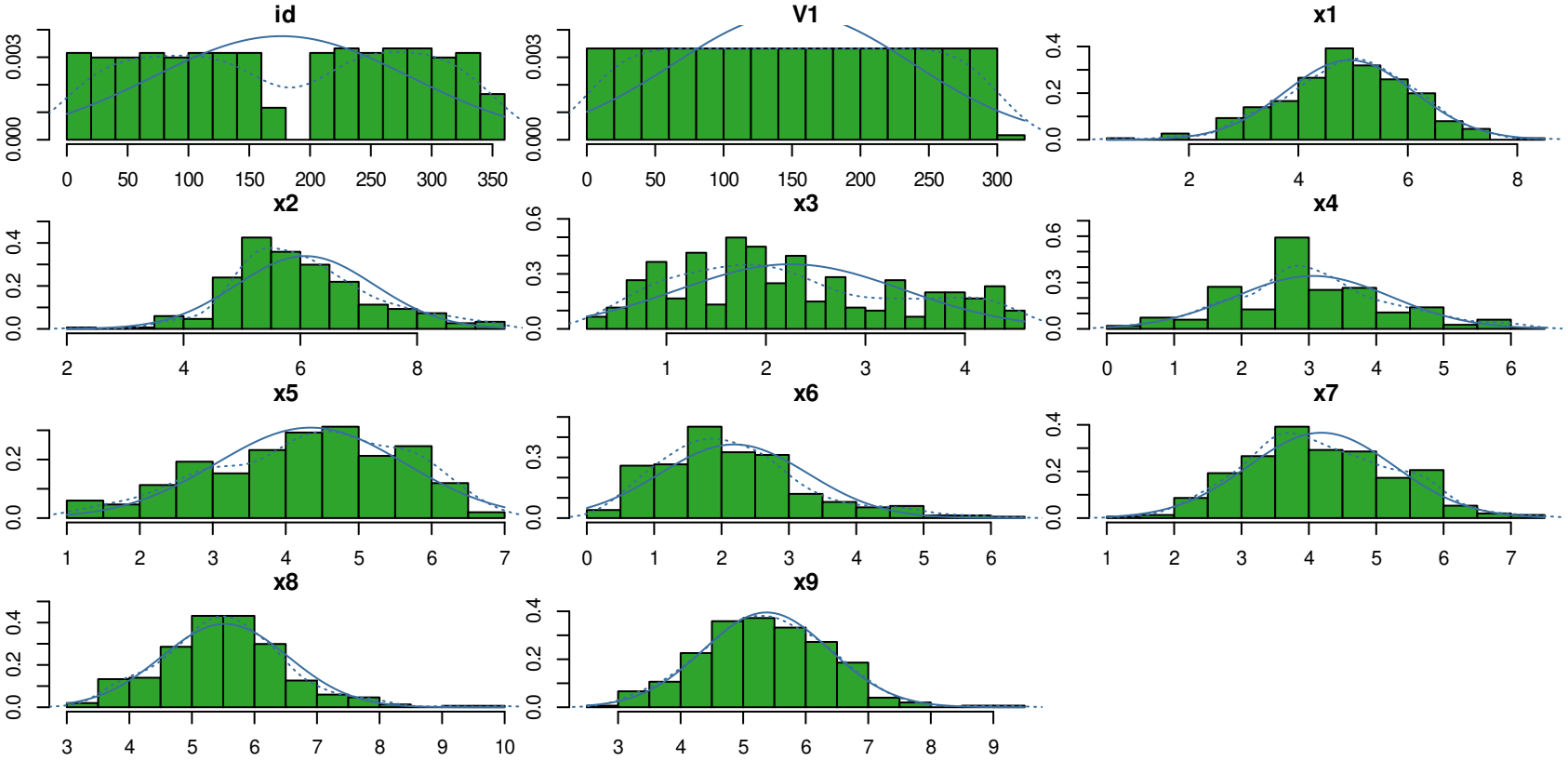

Histogram of x7
Relative Frequency
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
8


Histogram of x8
Relative Frequency
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8
9
10


Histogram of x9
Relative Frequency
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10

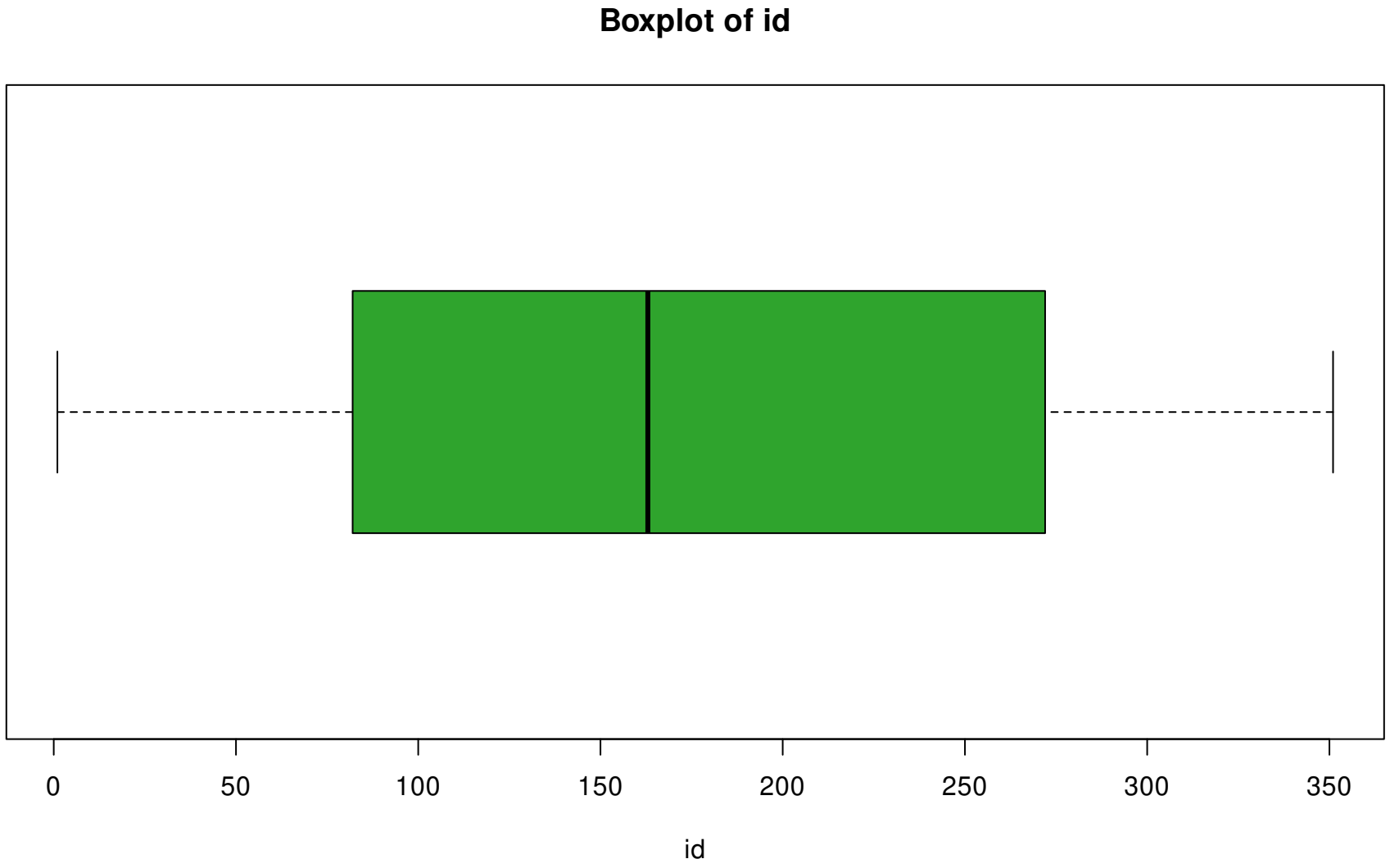
Histograms Summary

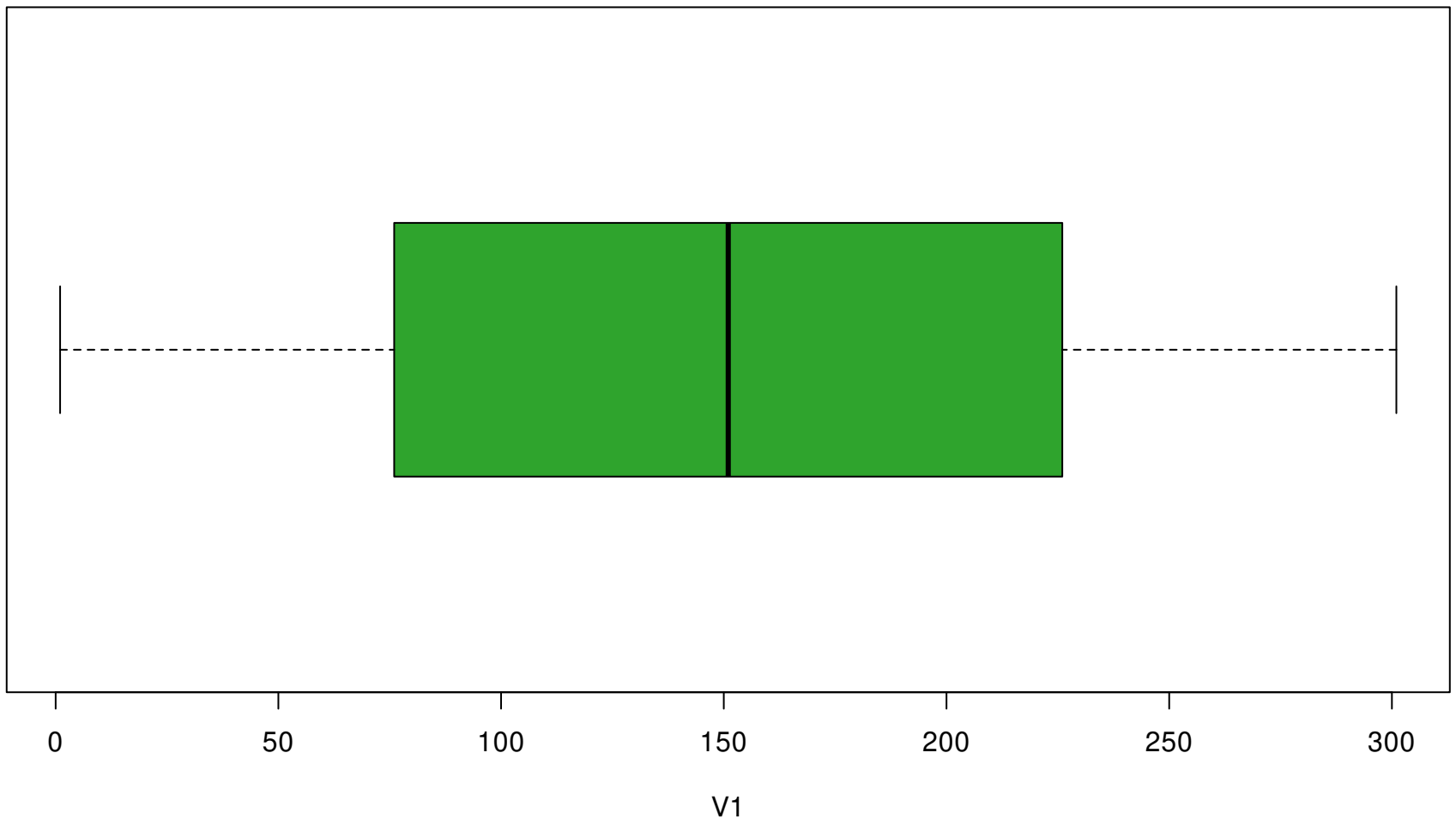
Multiple Relative Frequency Histogram in one figure. Variables are sorted alphabetically. The blue line represents the normal density approximation. The blue dotted line represents a special kernel density approximation.

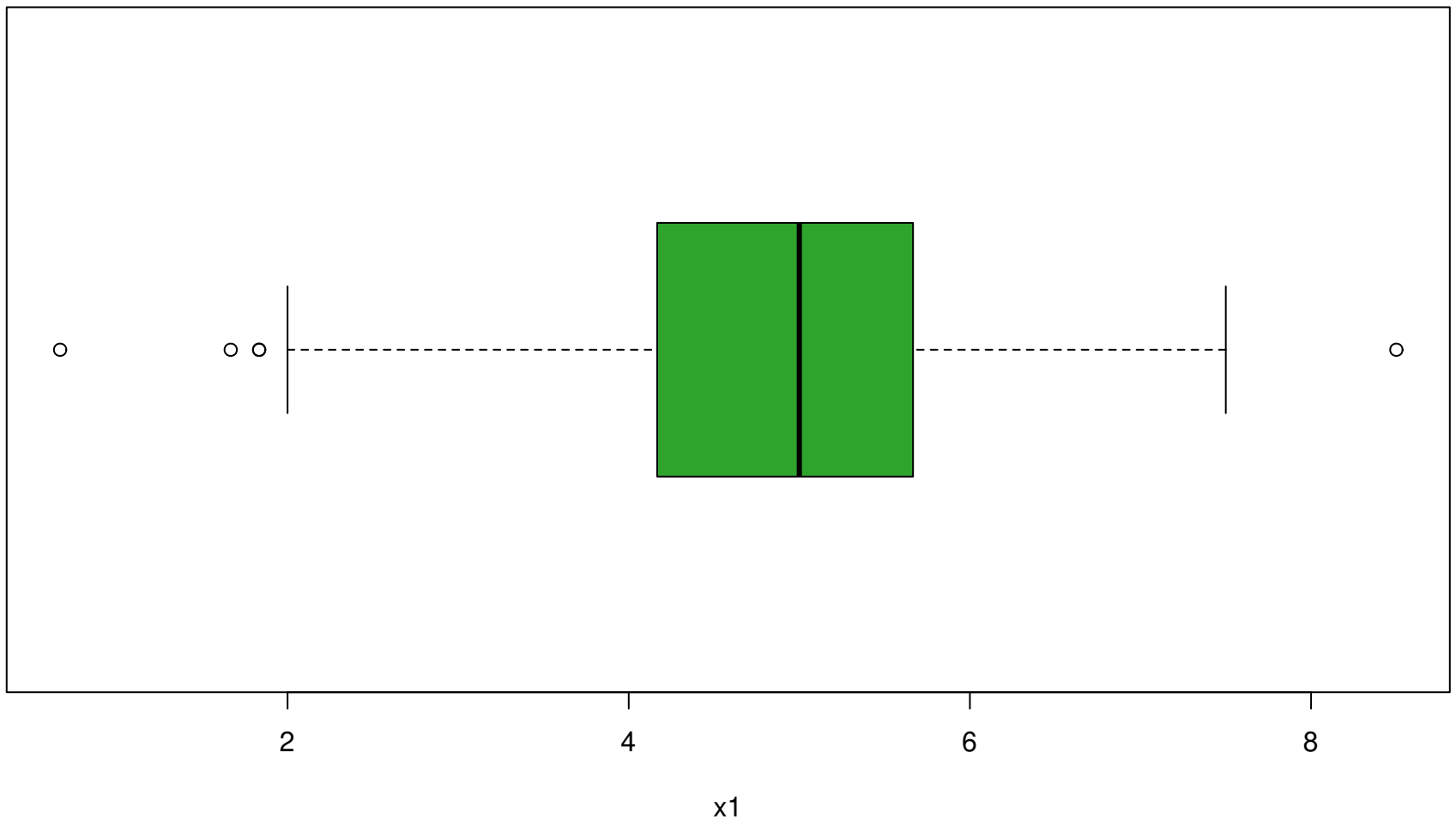


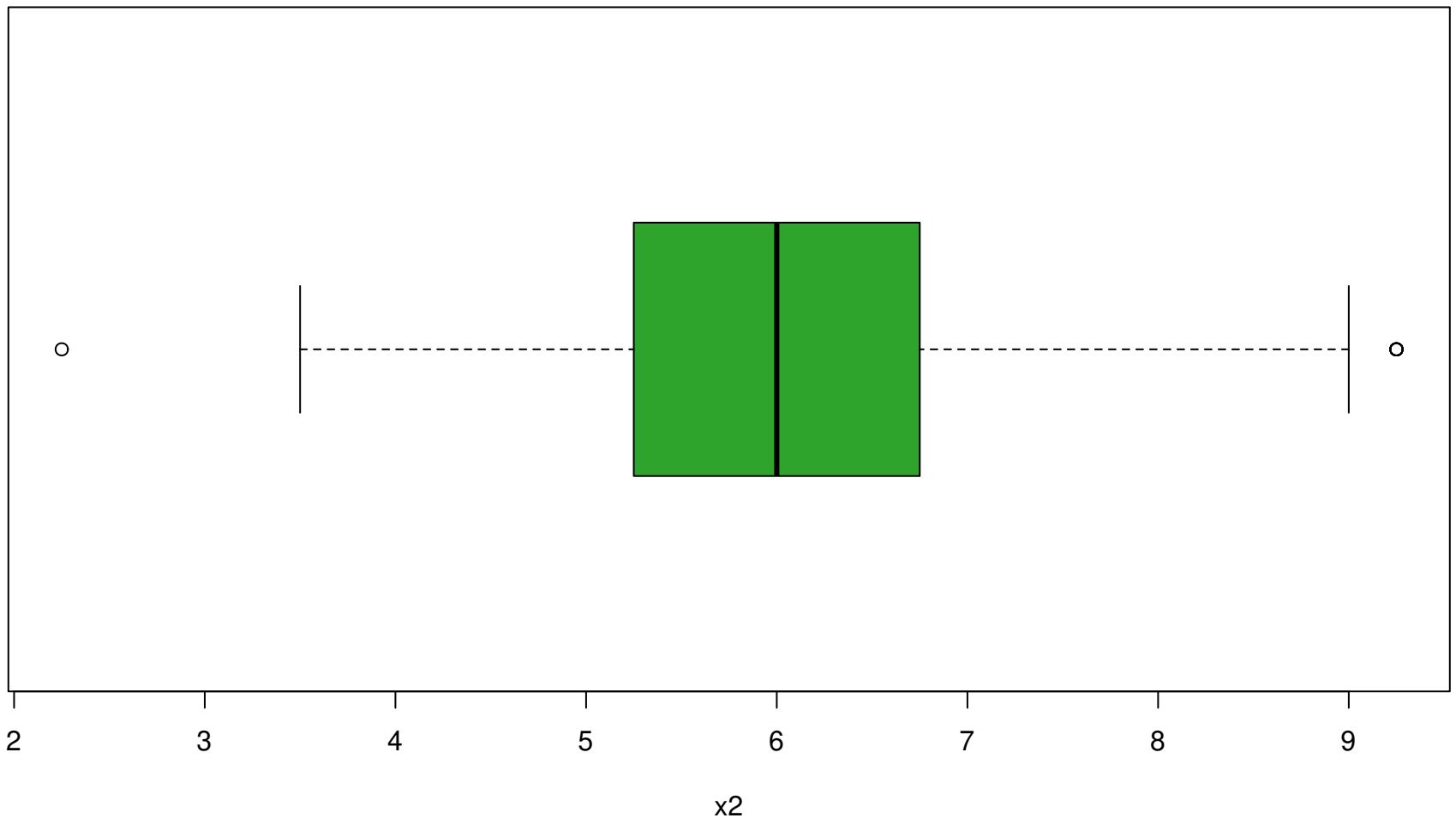
Box-Plots

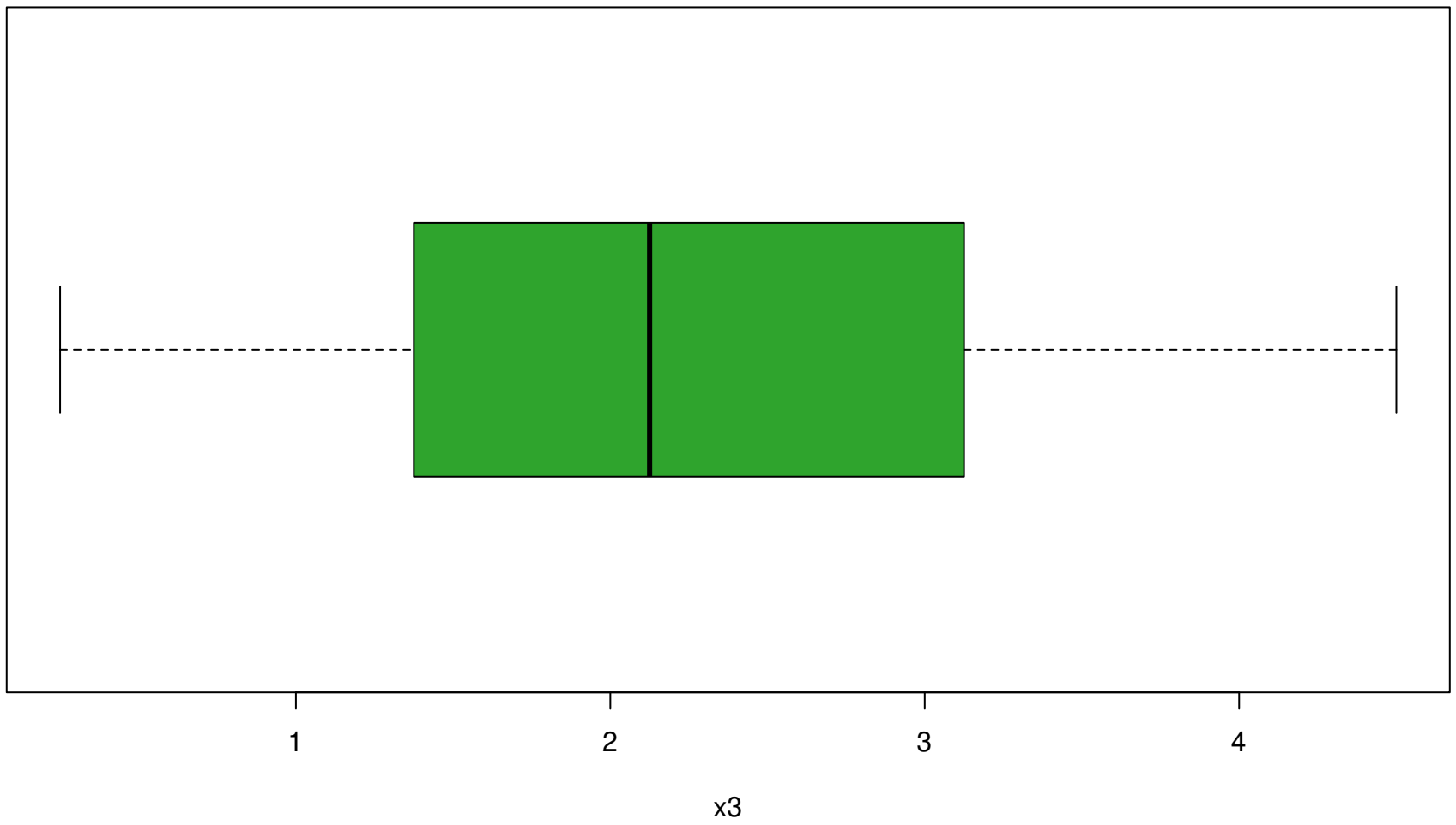
One Box-Plot per page for each variable. Variables are sorted alphabetically.

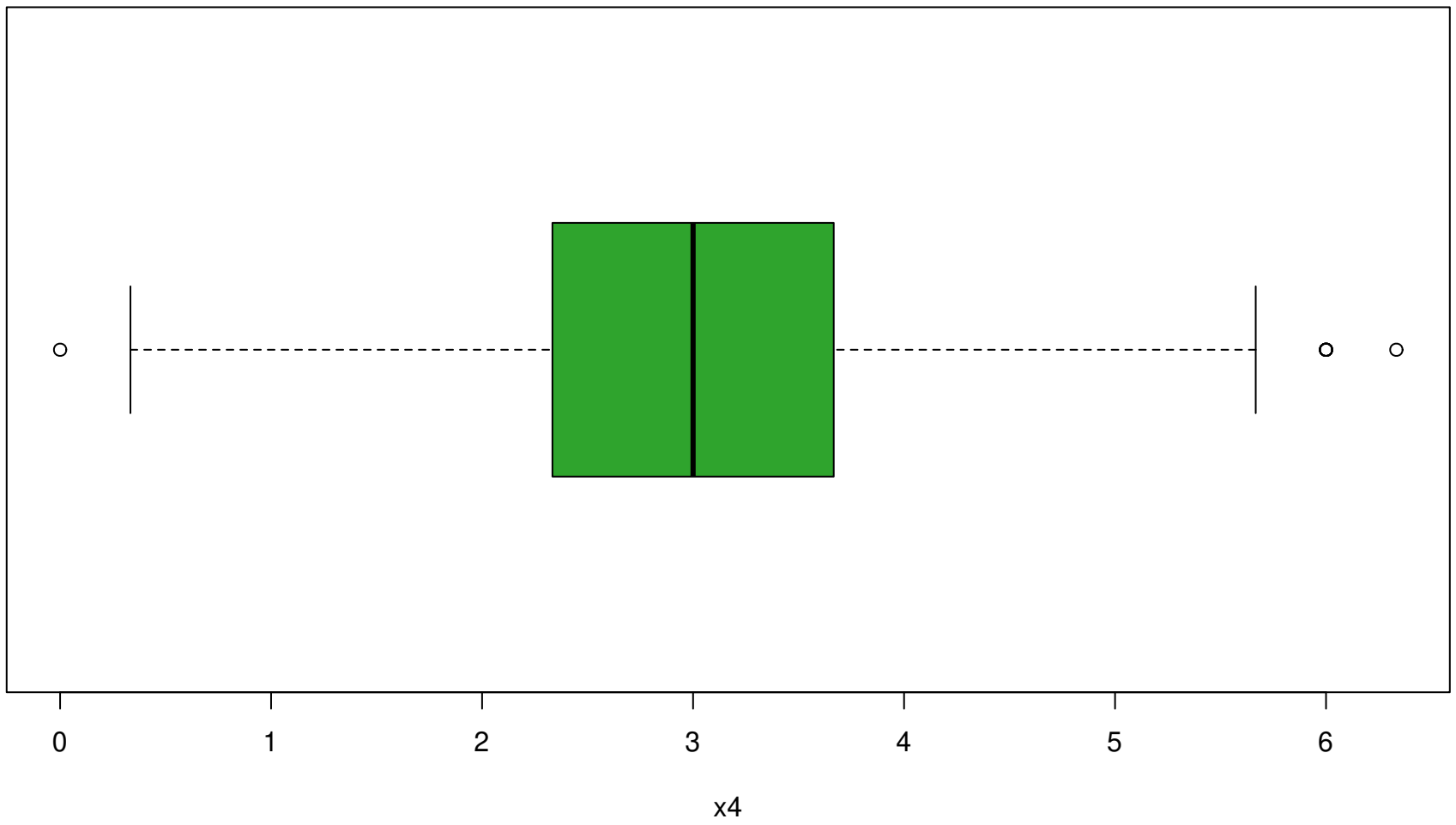


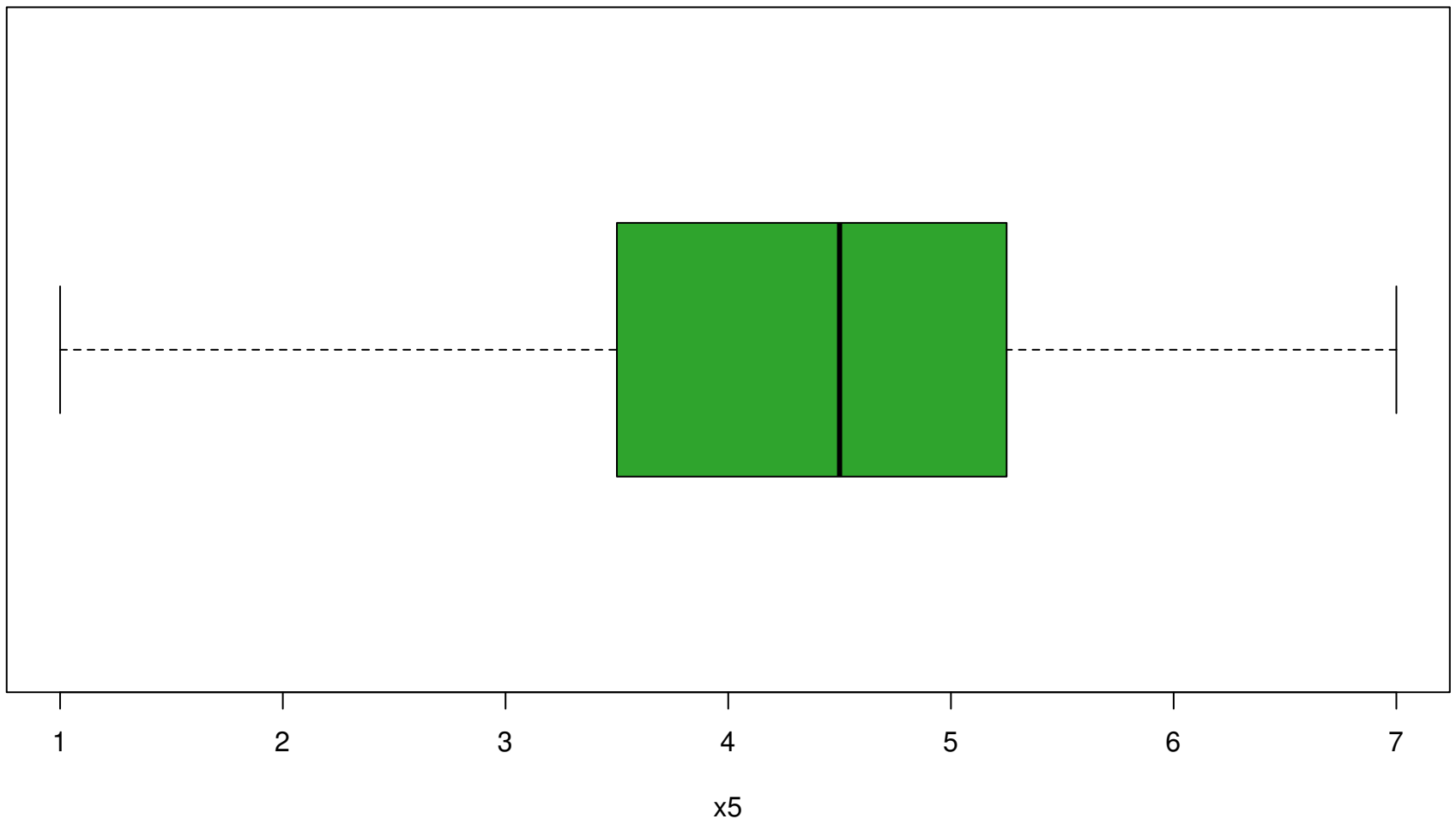

Boxplot of V1
0
50
100
150
200
250
300
V1

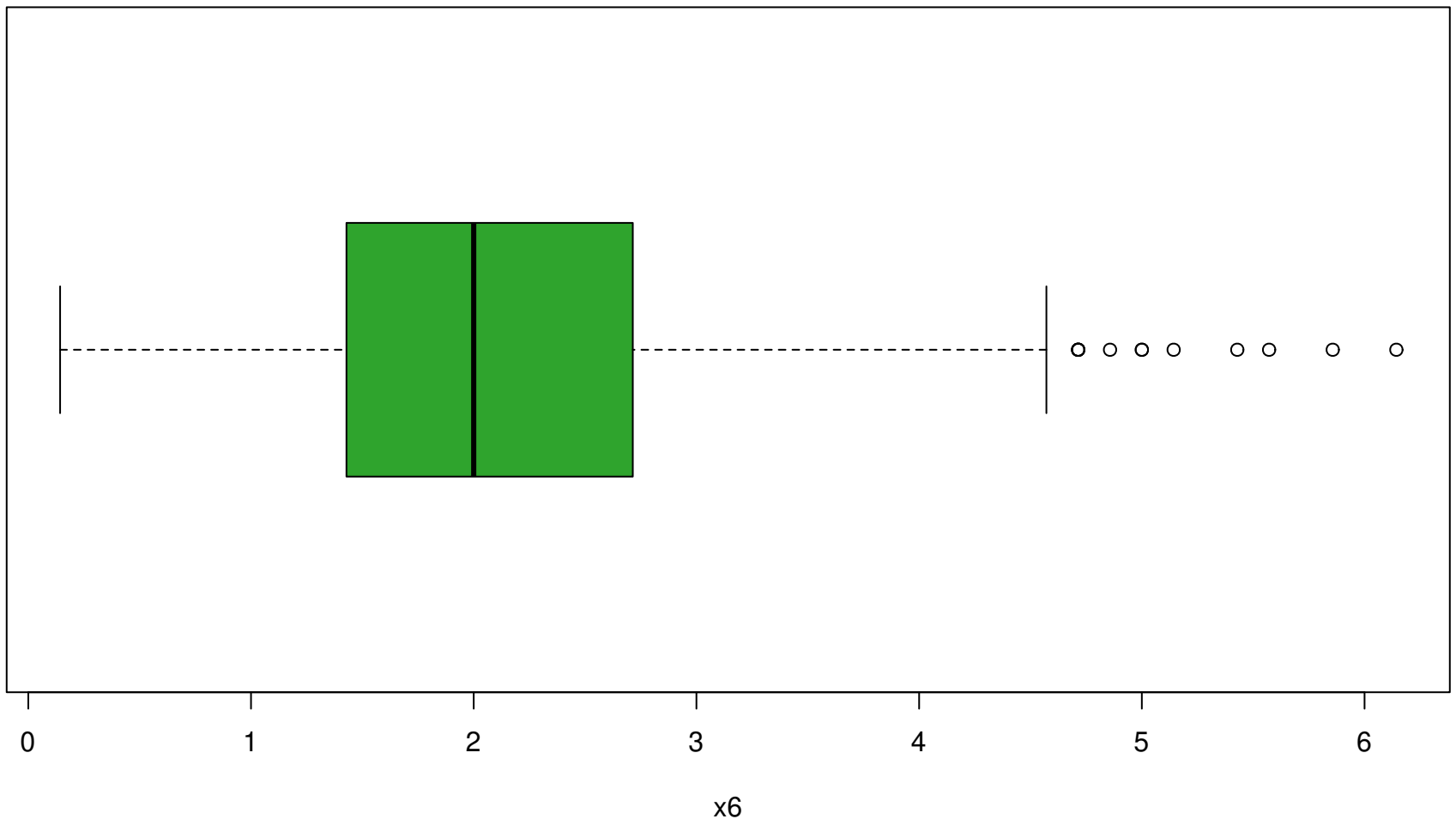

Boxplot of x1
2
4
6
8
x1

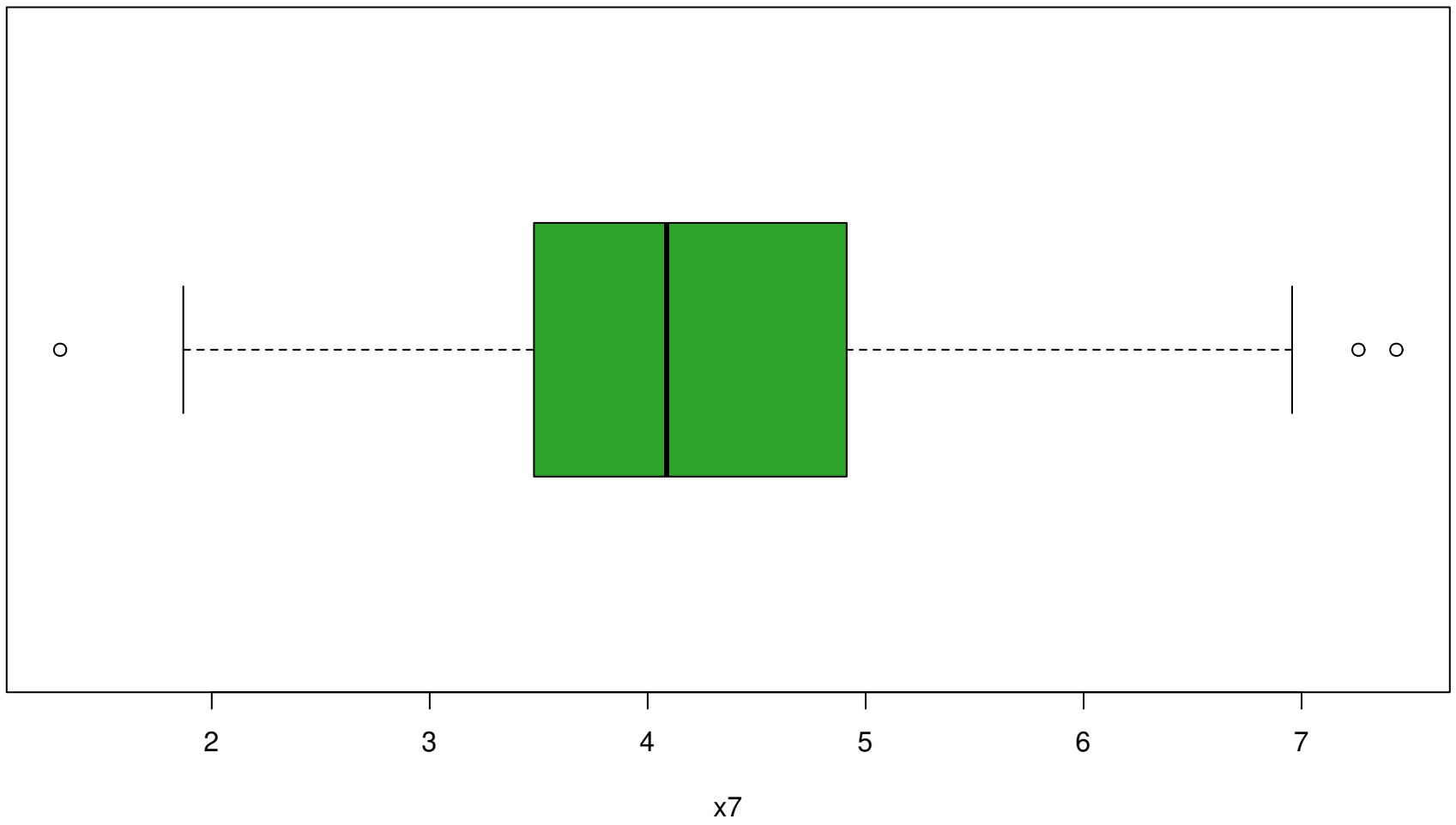

Boxplot of x2
2
3
4
5
6
7
8
9
x2

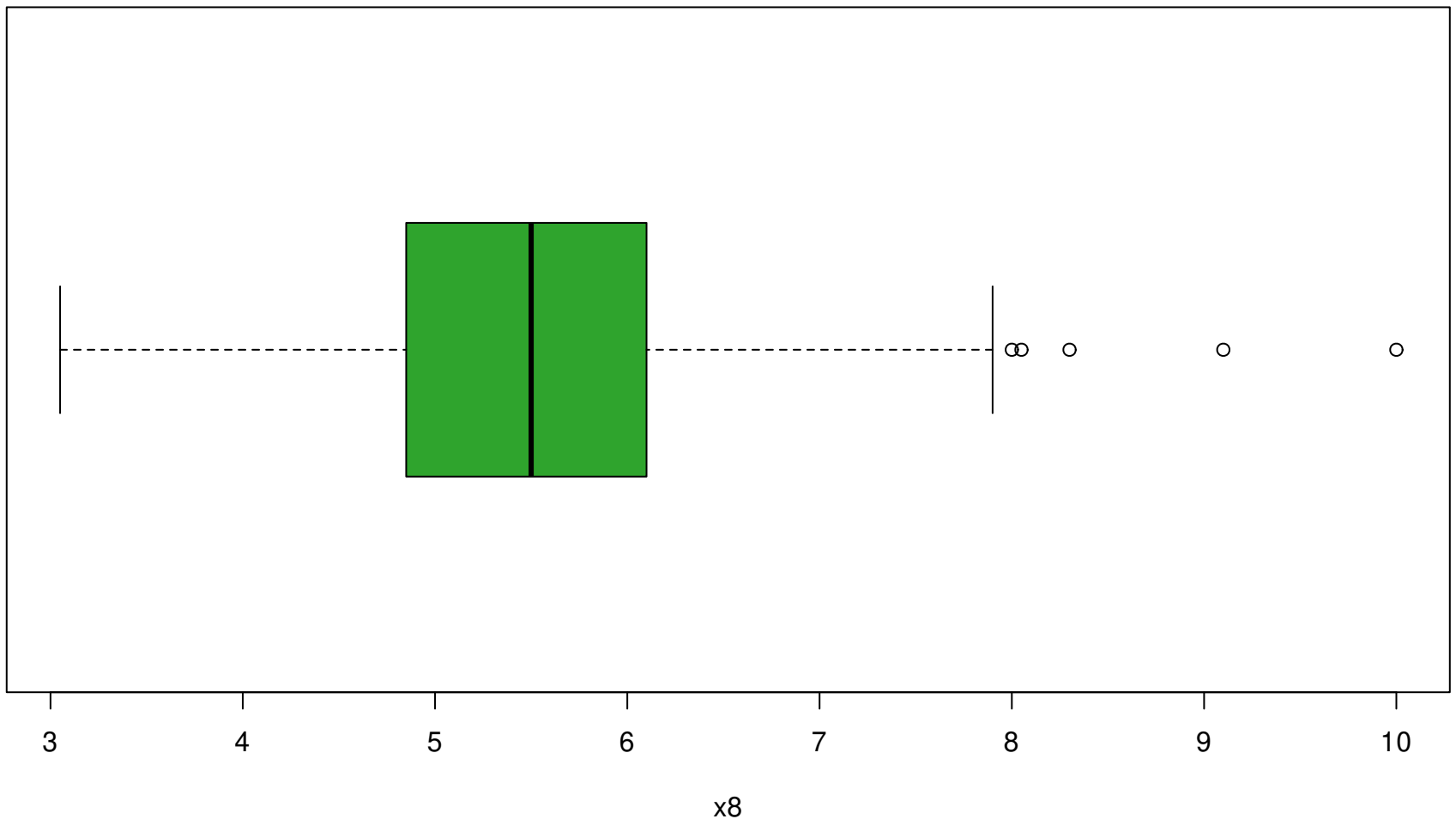

Boxplot of x3
1
2
3
4
x3

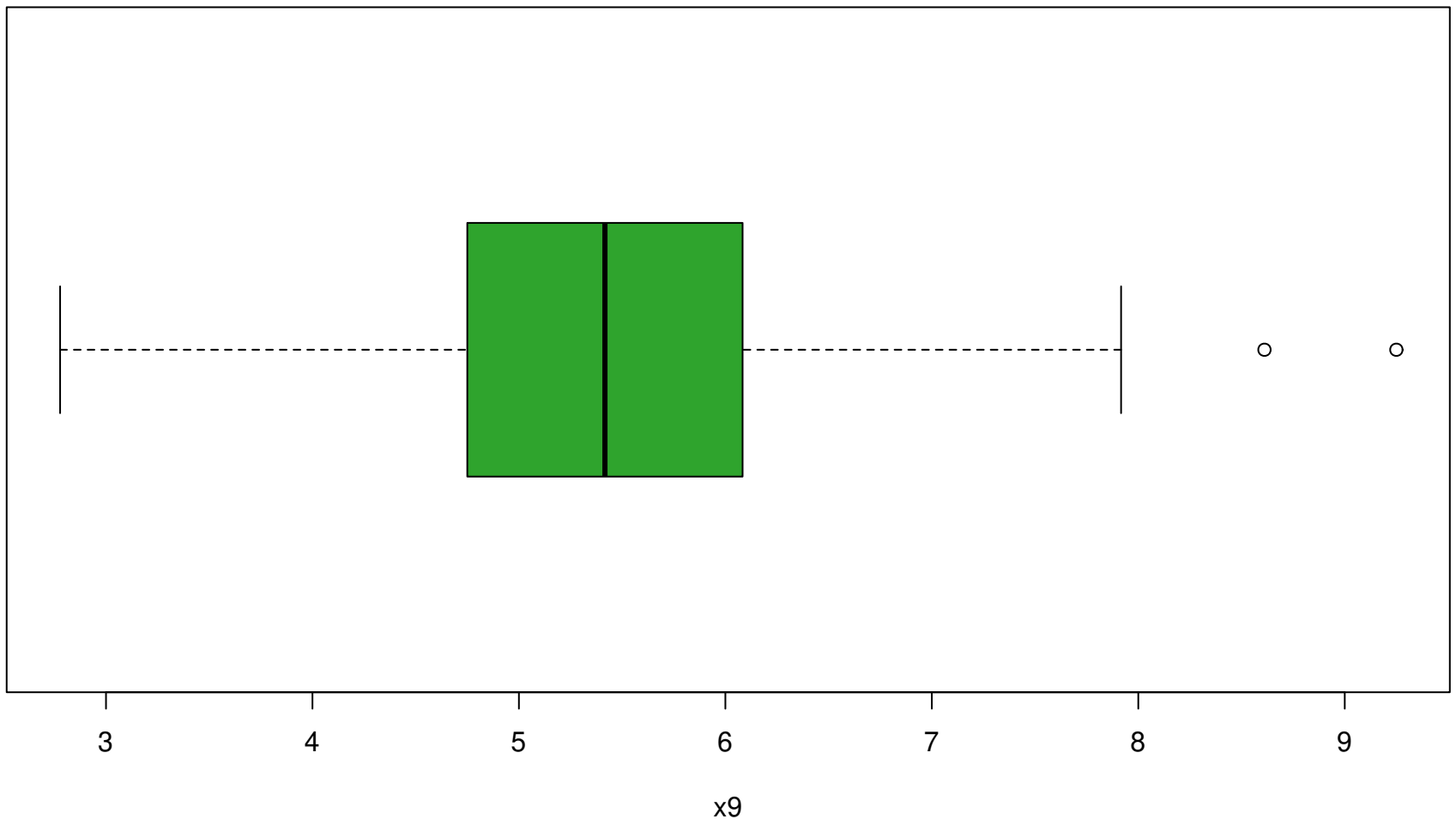

Boxplot of x4
0
1
2
3
4
5
6
x4


Boxplot of x5
1
2
3
4
5
6
7
x5


Boxplot of x6
0
1
2
3
4
5
6
x6

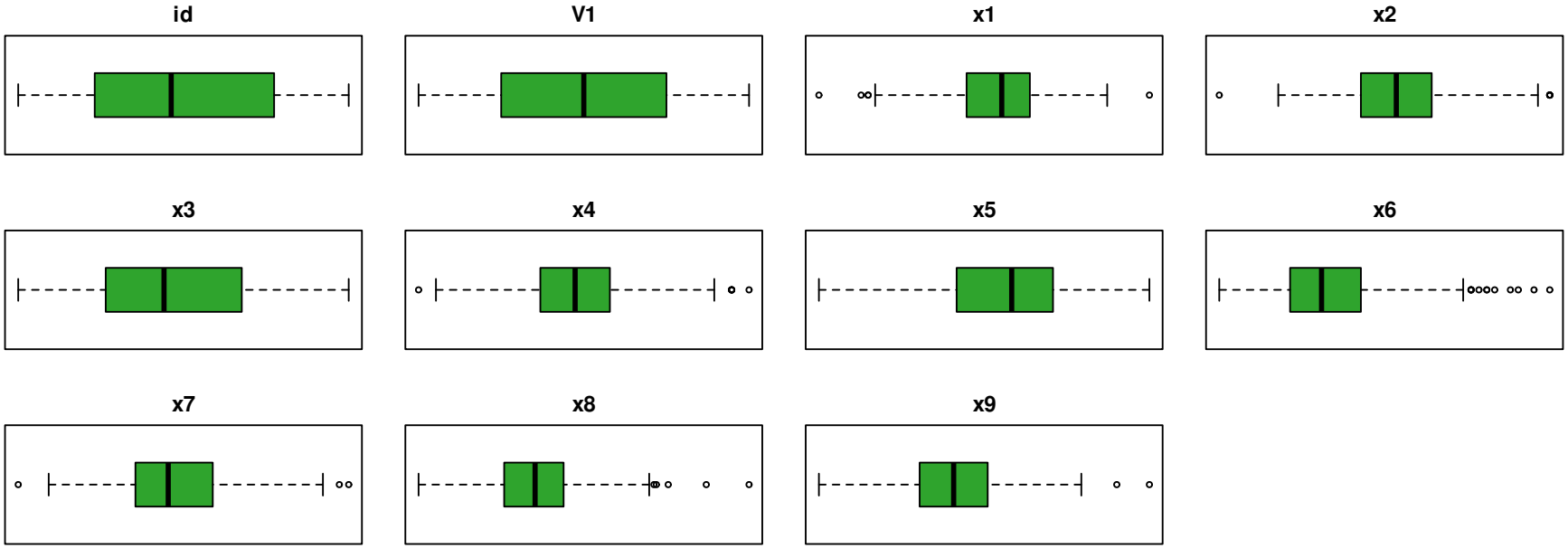

Boxplot of x7
2
3
4
5
6
7
x7


Boxplot of x8
3
4
5
6
7
8
9
10
x8


Boxplot of x9
3
4
5
6
7
8
9
x9

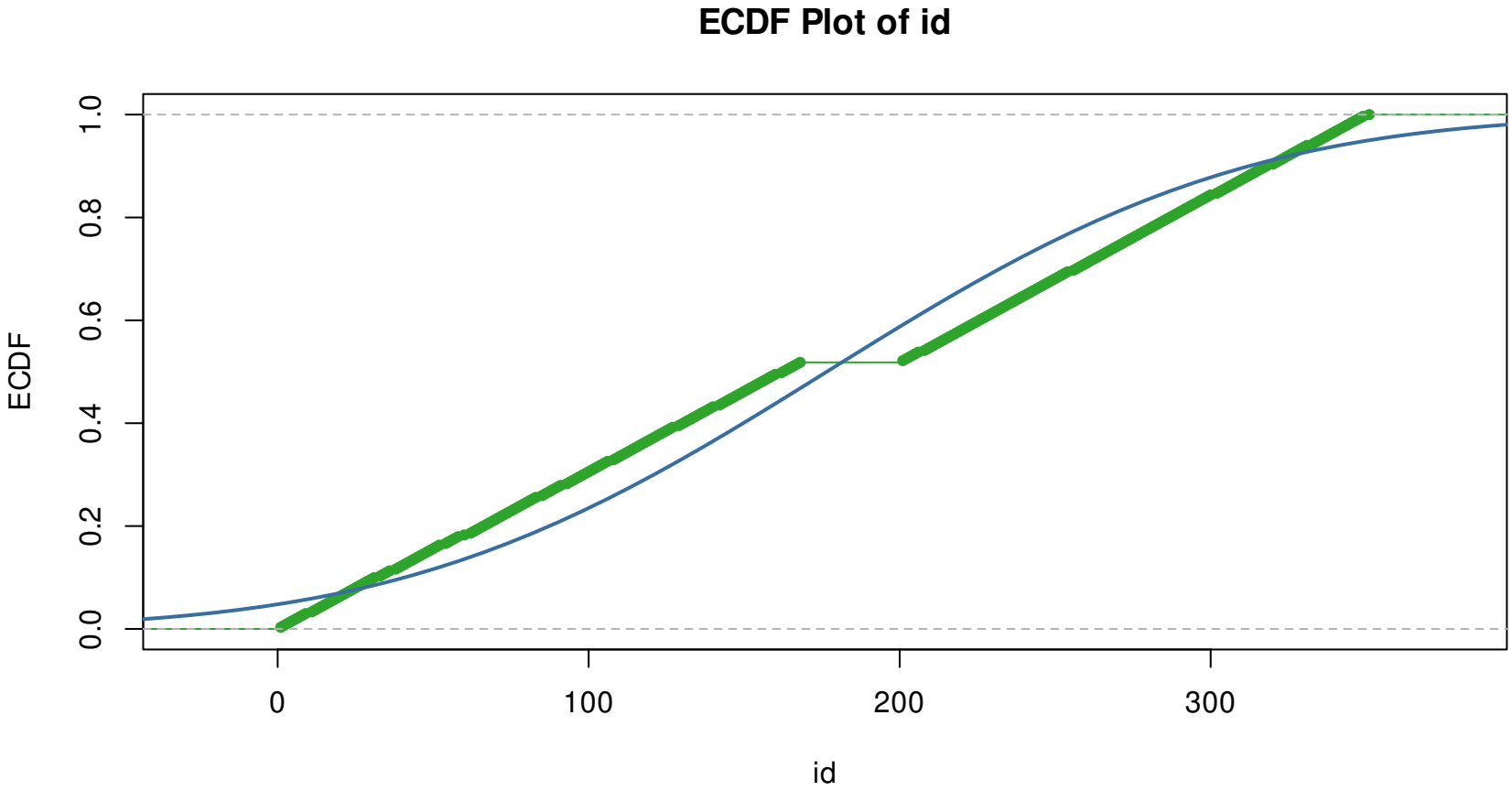
Box-Plots Summary

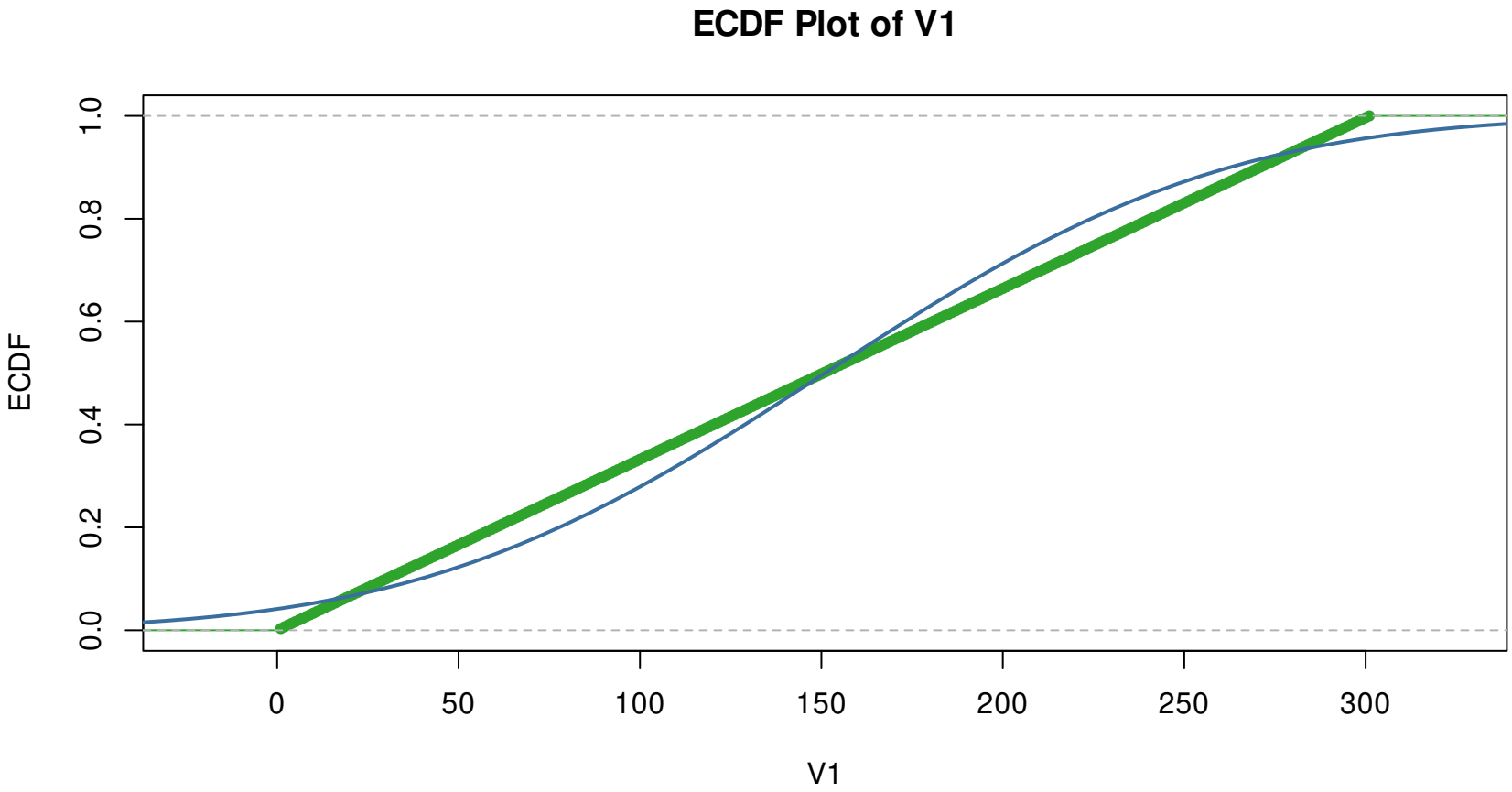
Multiple Box-Plots of variables in one figure. Variables are sorted alphabetically.

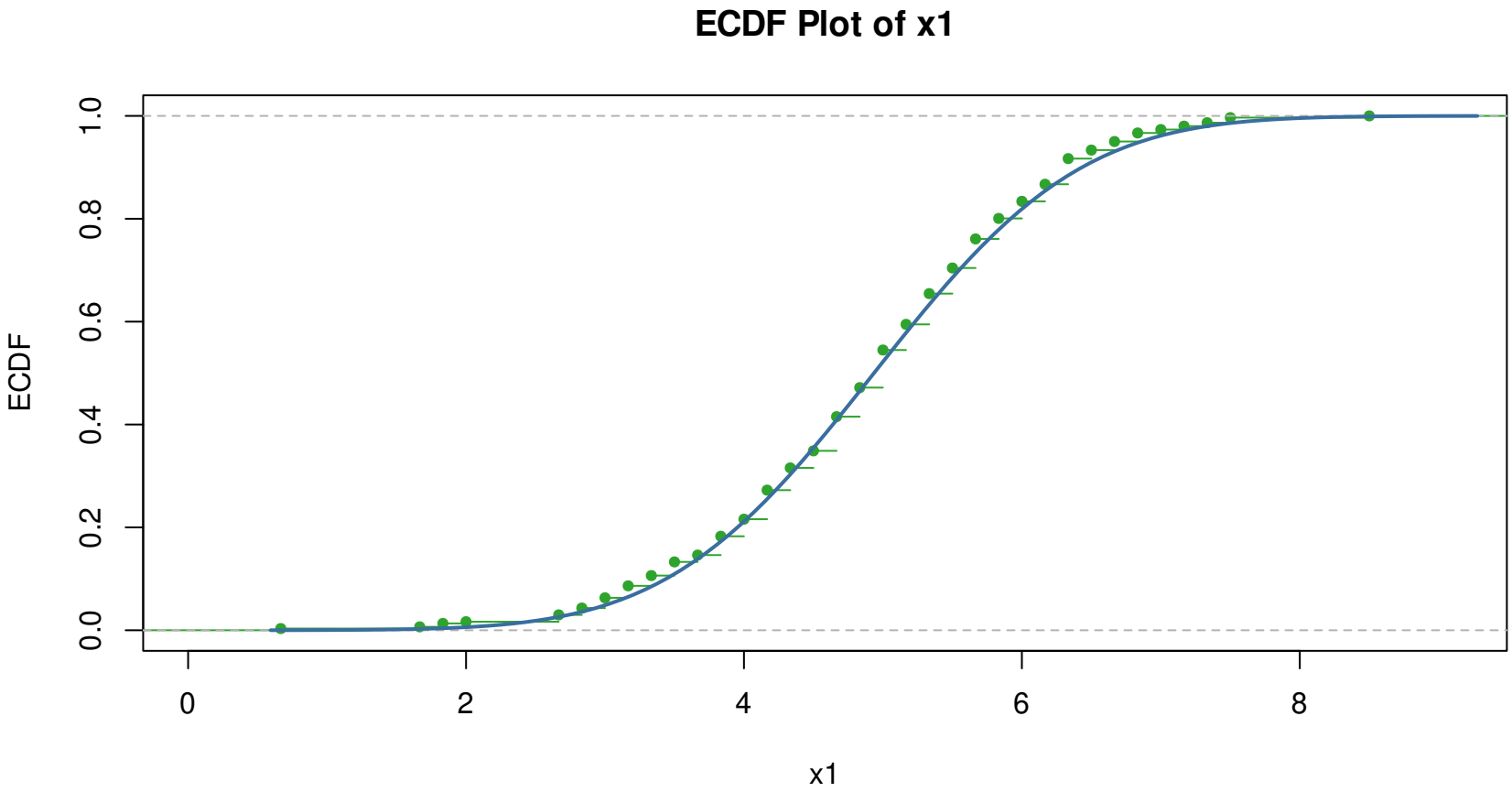


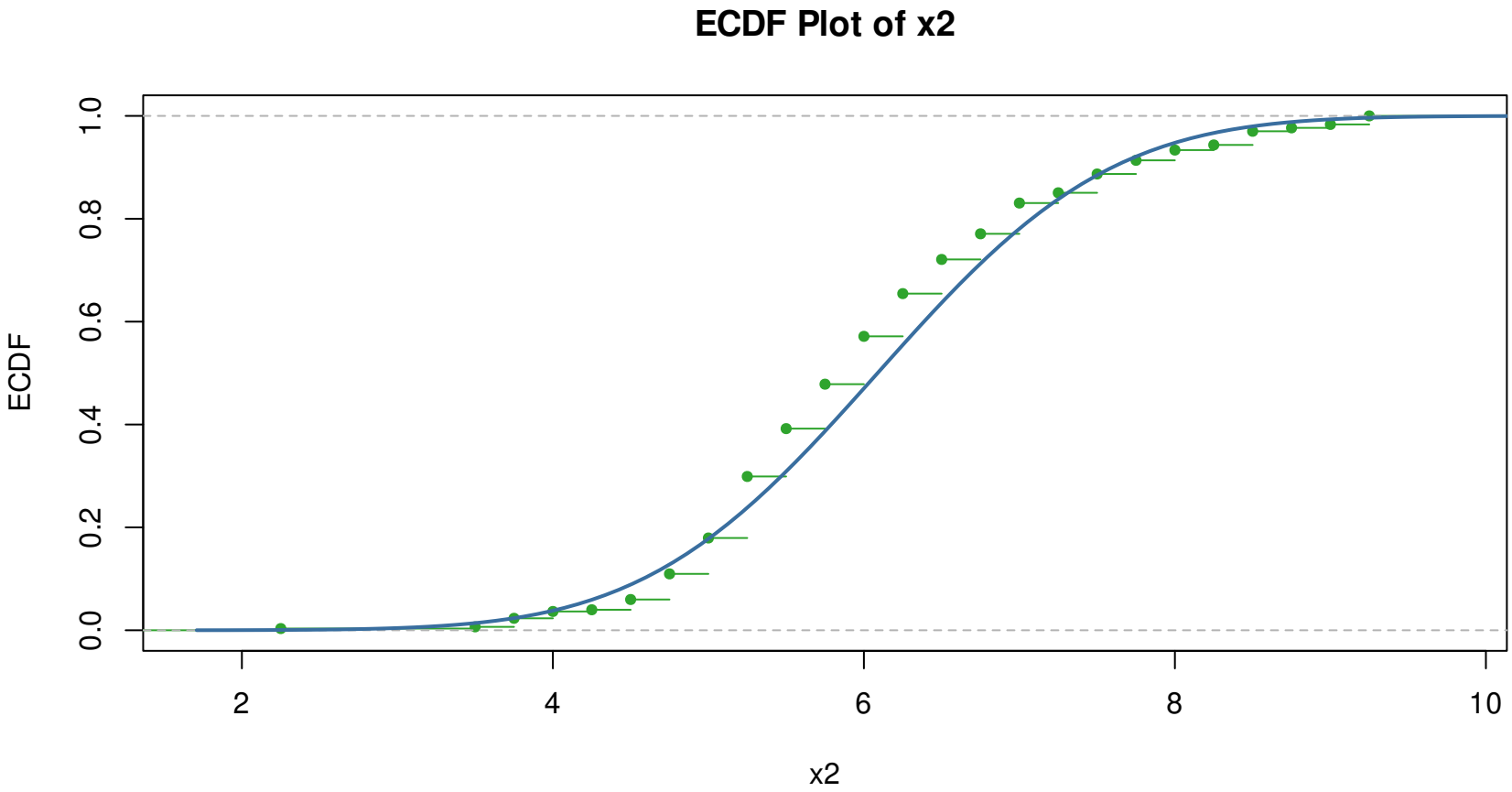
ECDF Plots

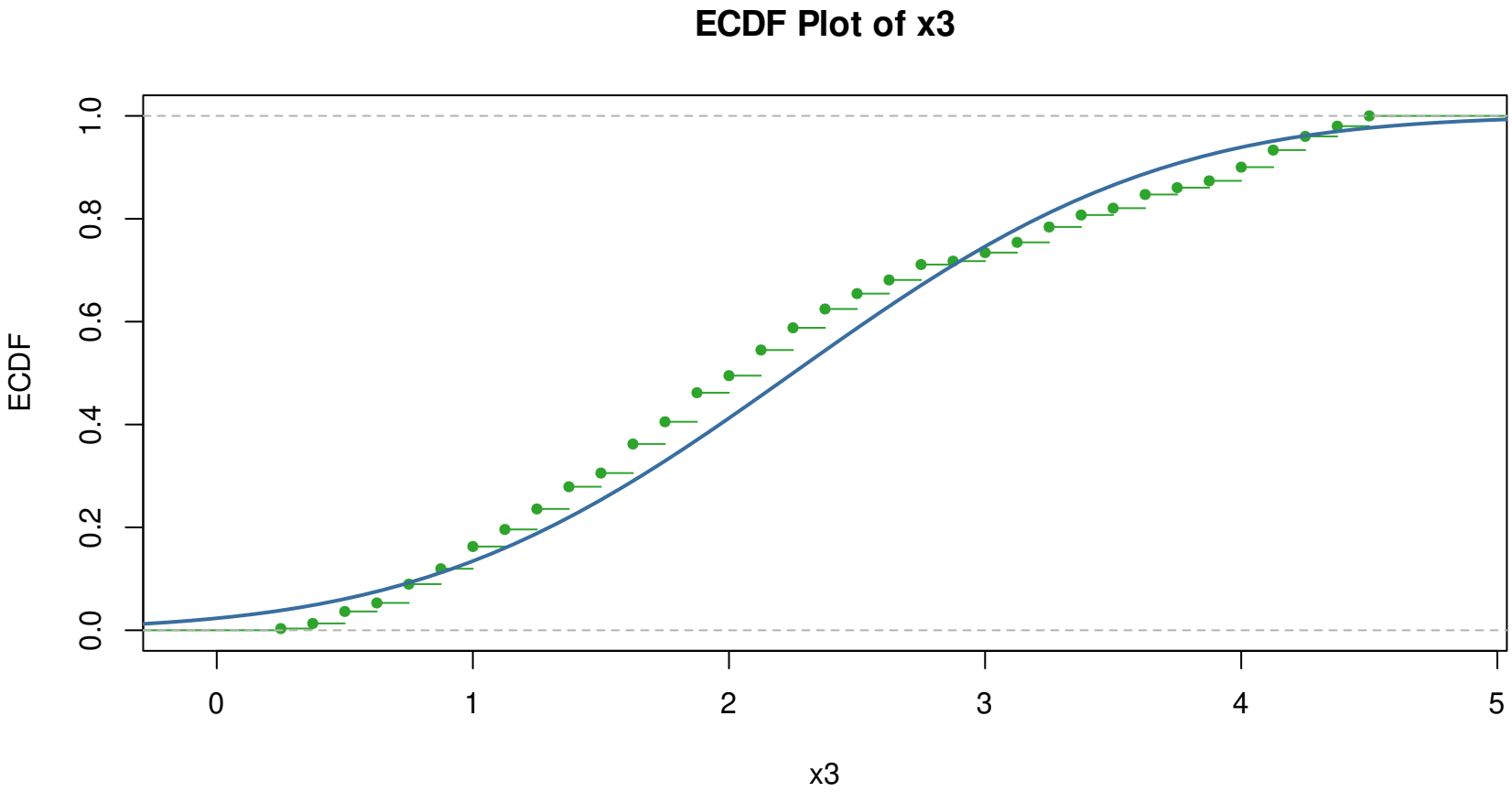
One ECDF (Empirical Cumulative Distribution Function) Plot per page for each variable. Variables are sorted alphabetically. The blue line represents the CDF of a normal distribution. If the variable is normally distributed, the blue line approximates well the ECDF.

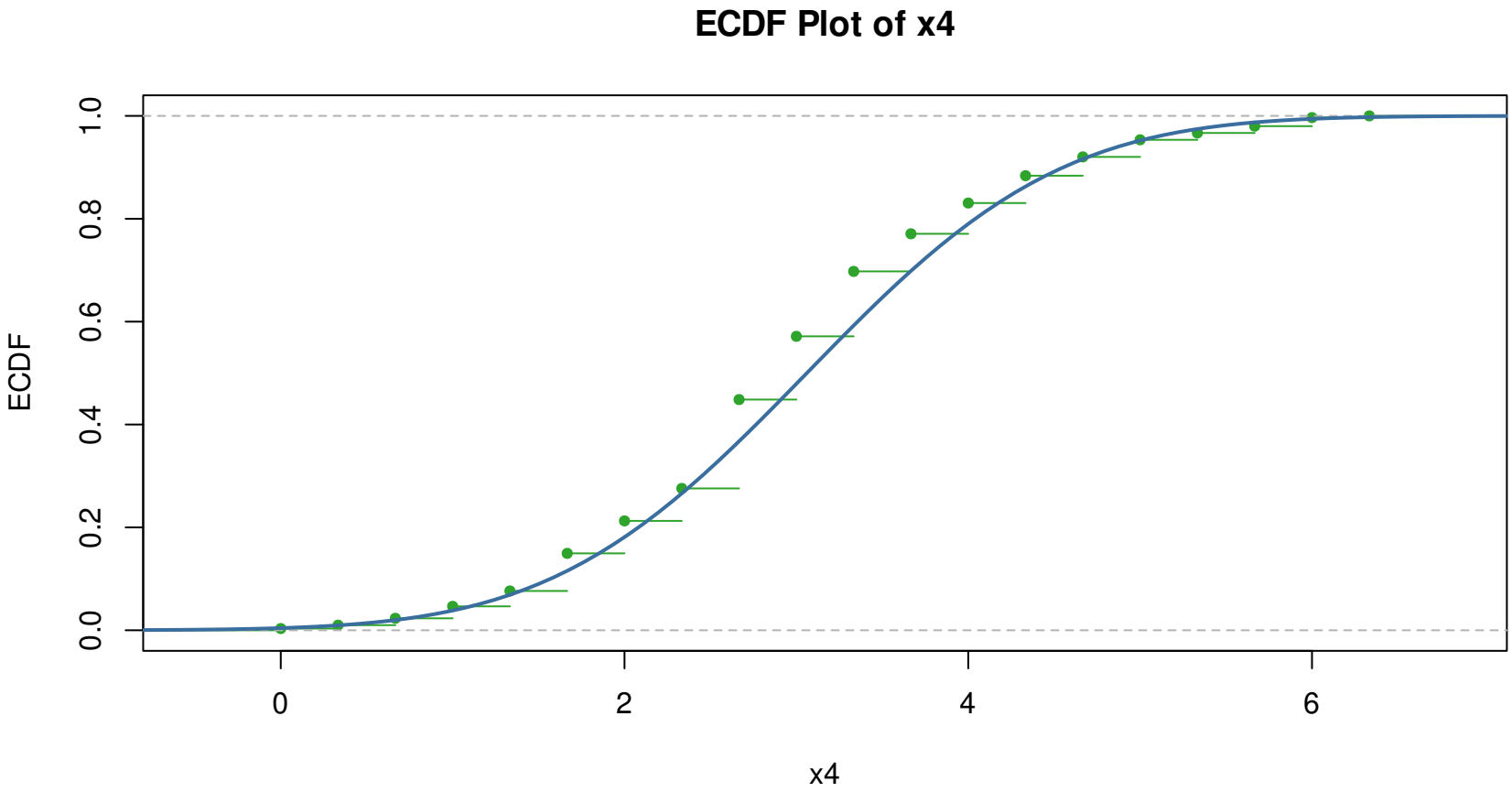


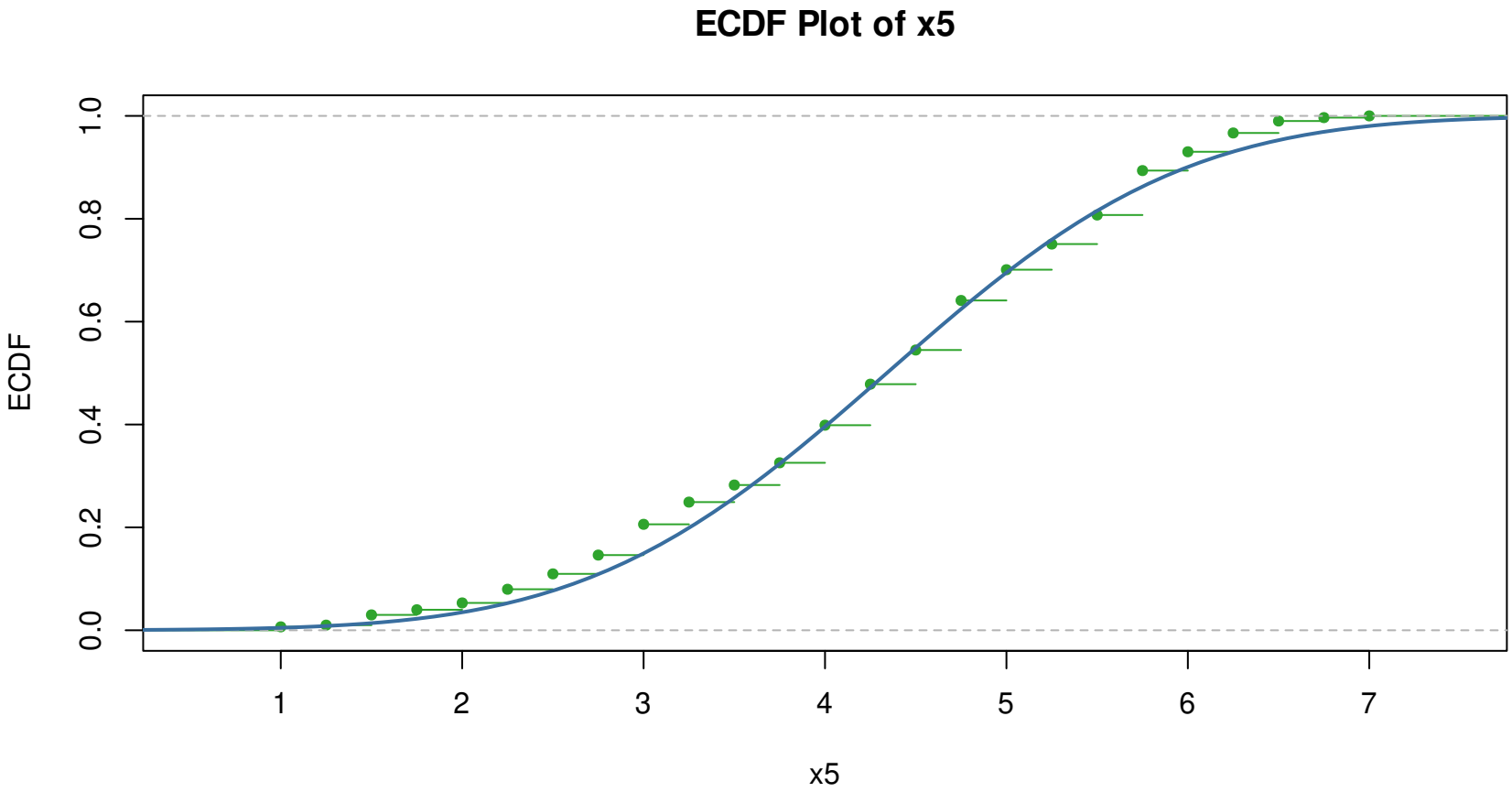

ECDF Plot of V1
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
V1

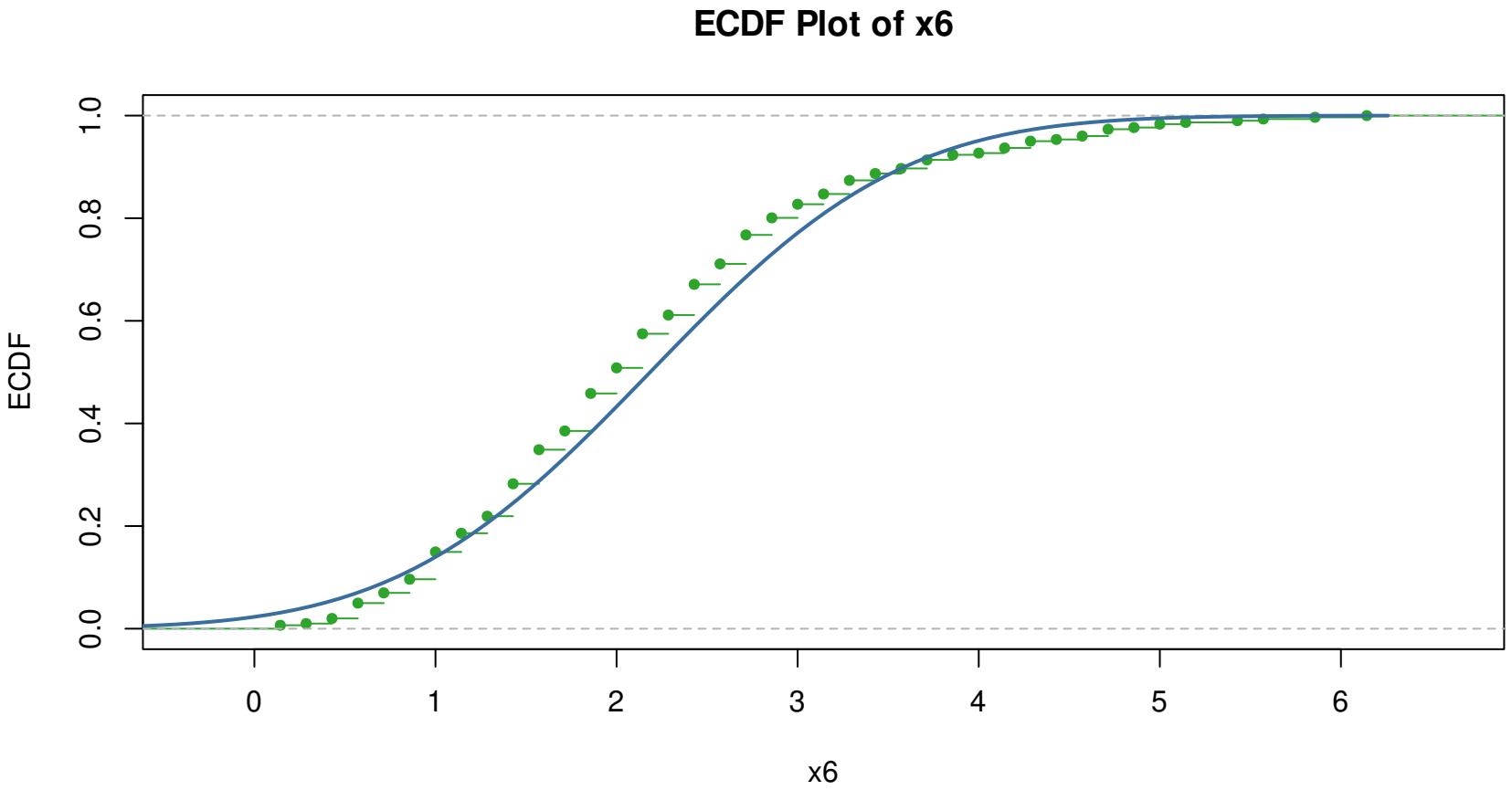

ECDF Plot of x1
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
x1

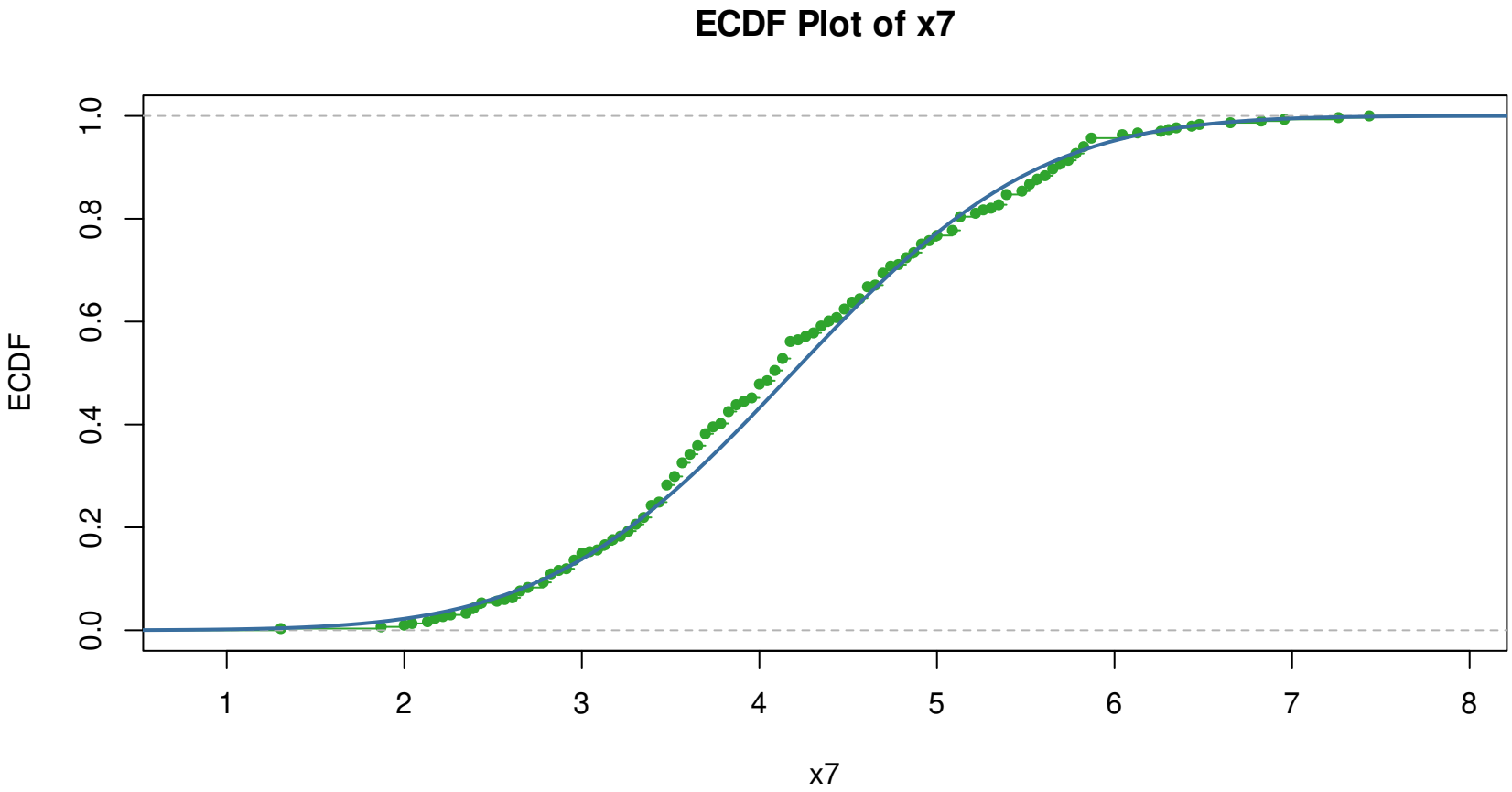

ECDF Plot of x2
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
x2

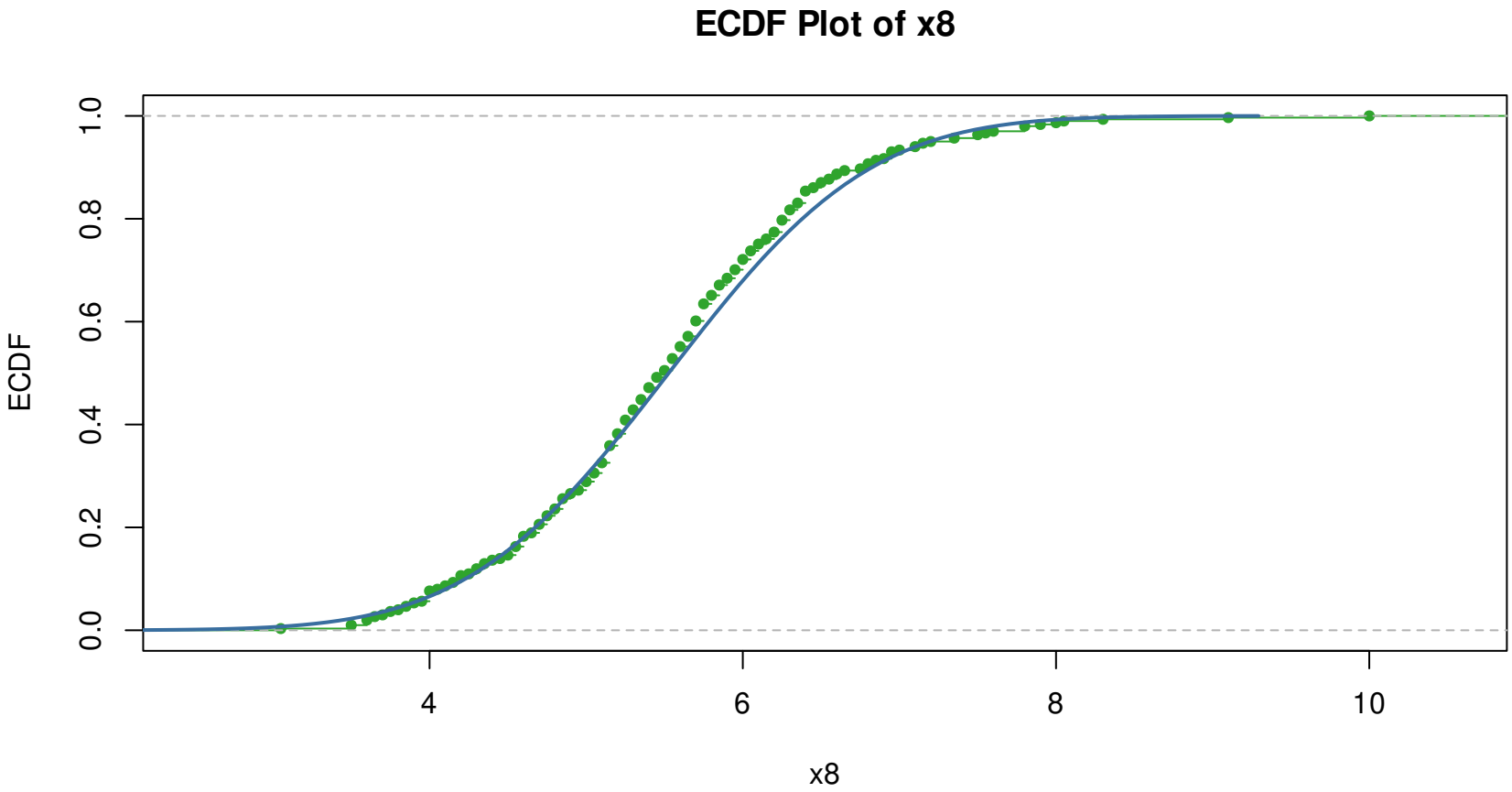

ECDF Plot of x3
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
x3

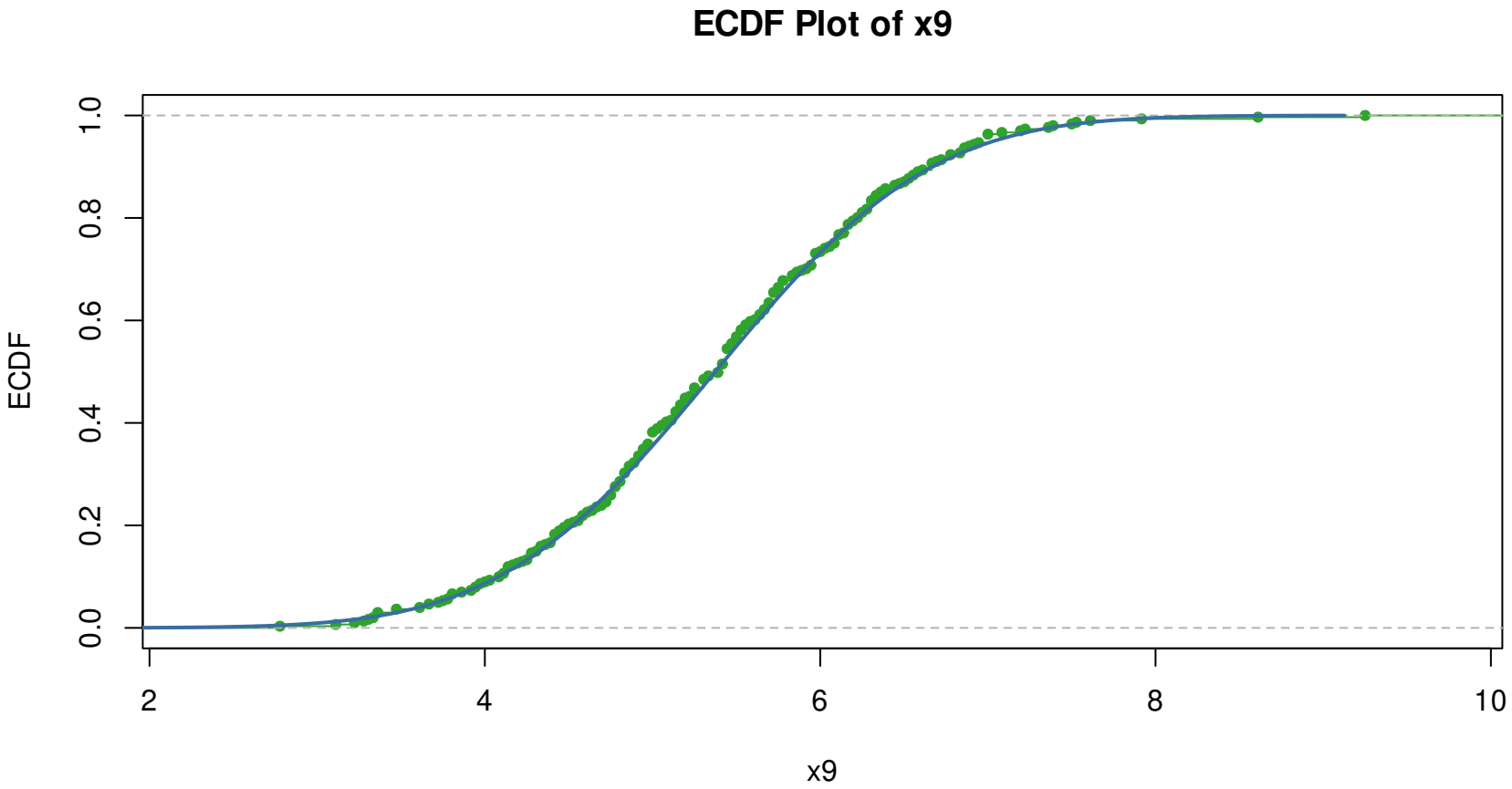

ECDF Plot of x4
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
x4


ECDF Plot of x5
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
x5


ECDF Plot of x6
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
x6

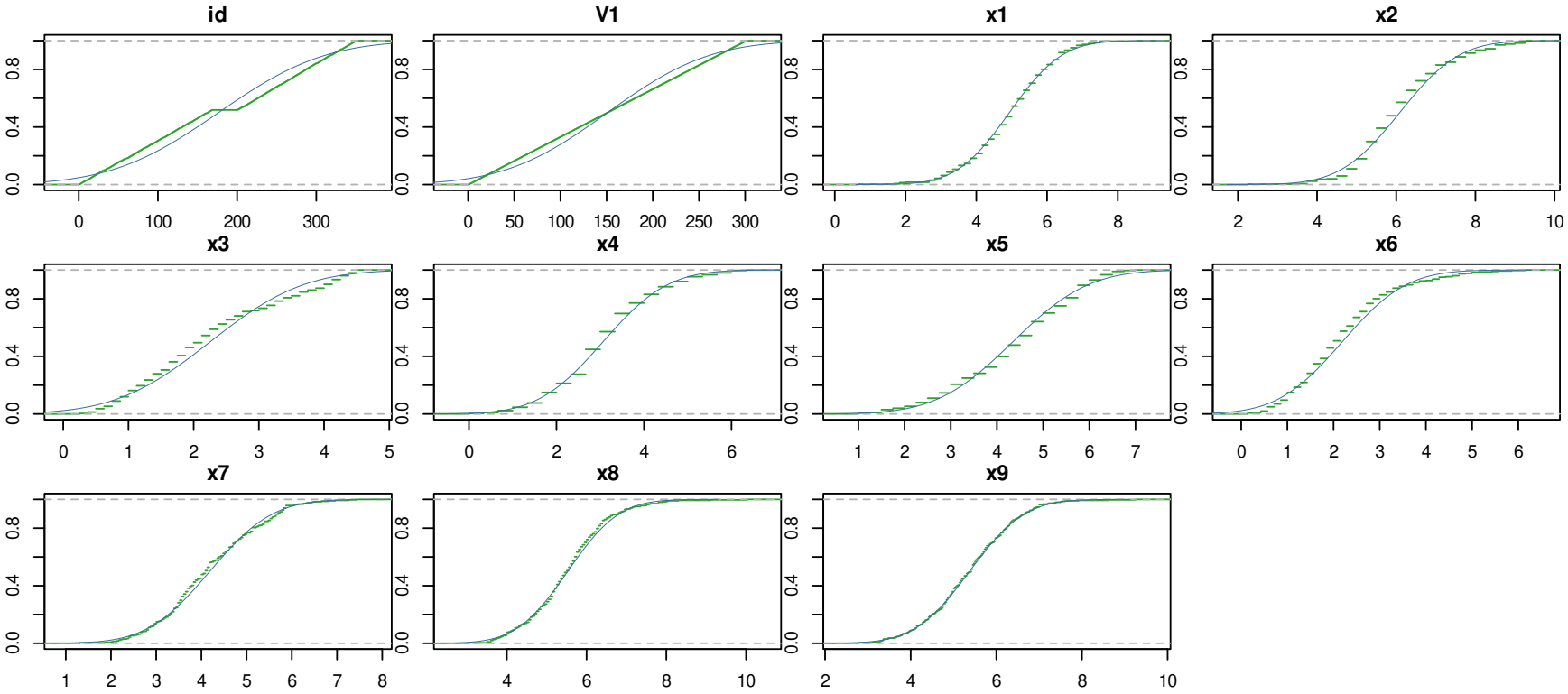

ECDF Plot of x7
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
x7


ECDF Plot of x8
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
4
6
8
10
x8


ECDF Plot of x9
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
x9

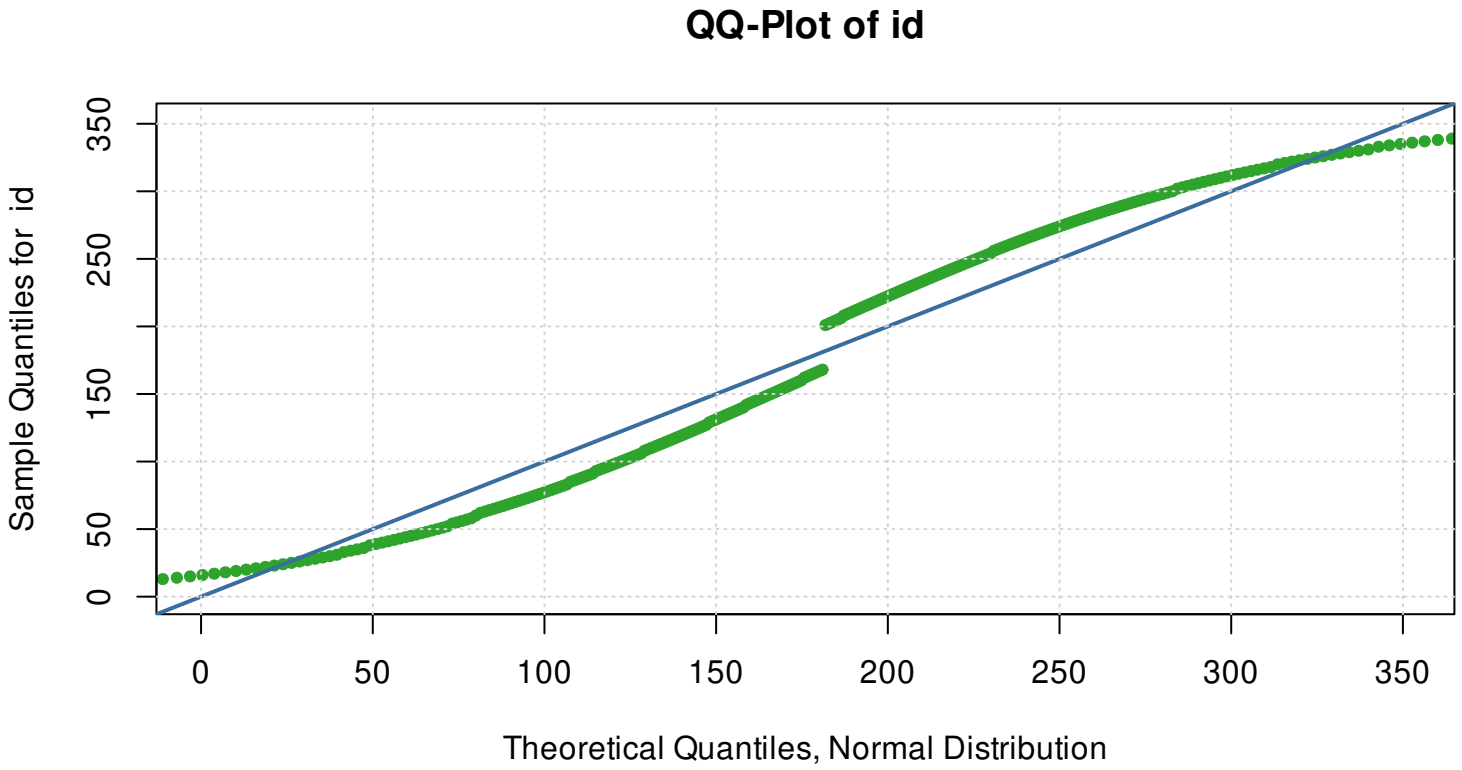
ECDF Plots Summary

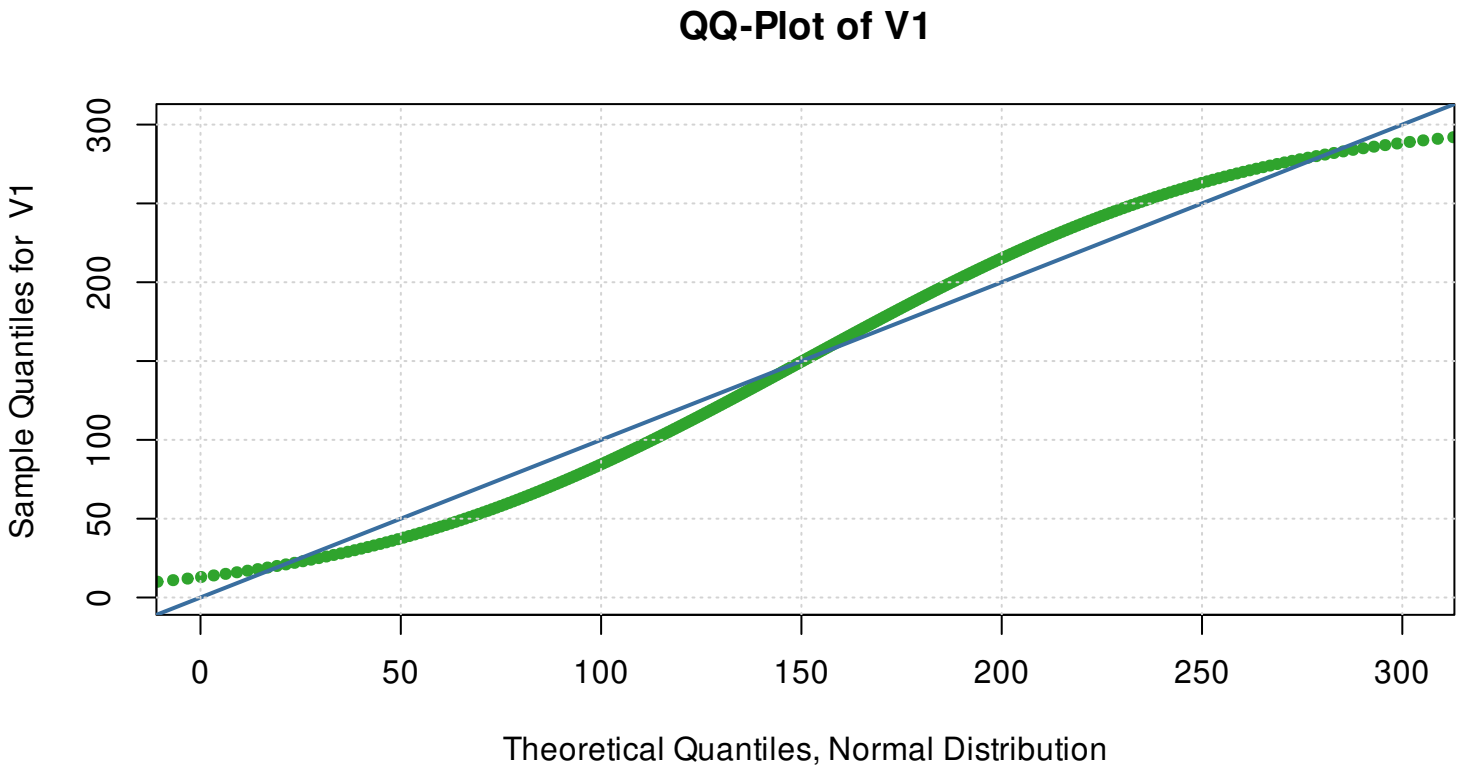
Multiple ECDF Plots of variables in one figure. Variables are sorted alphabetically.

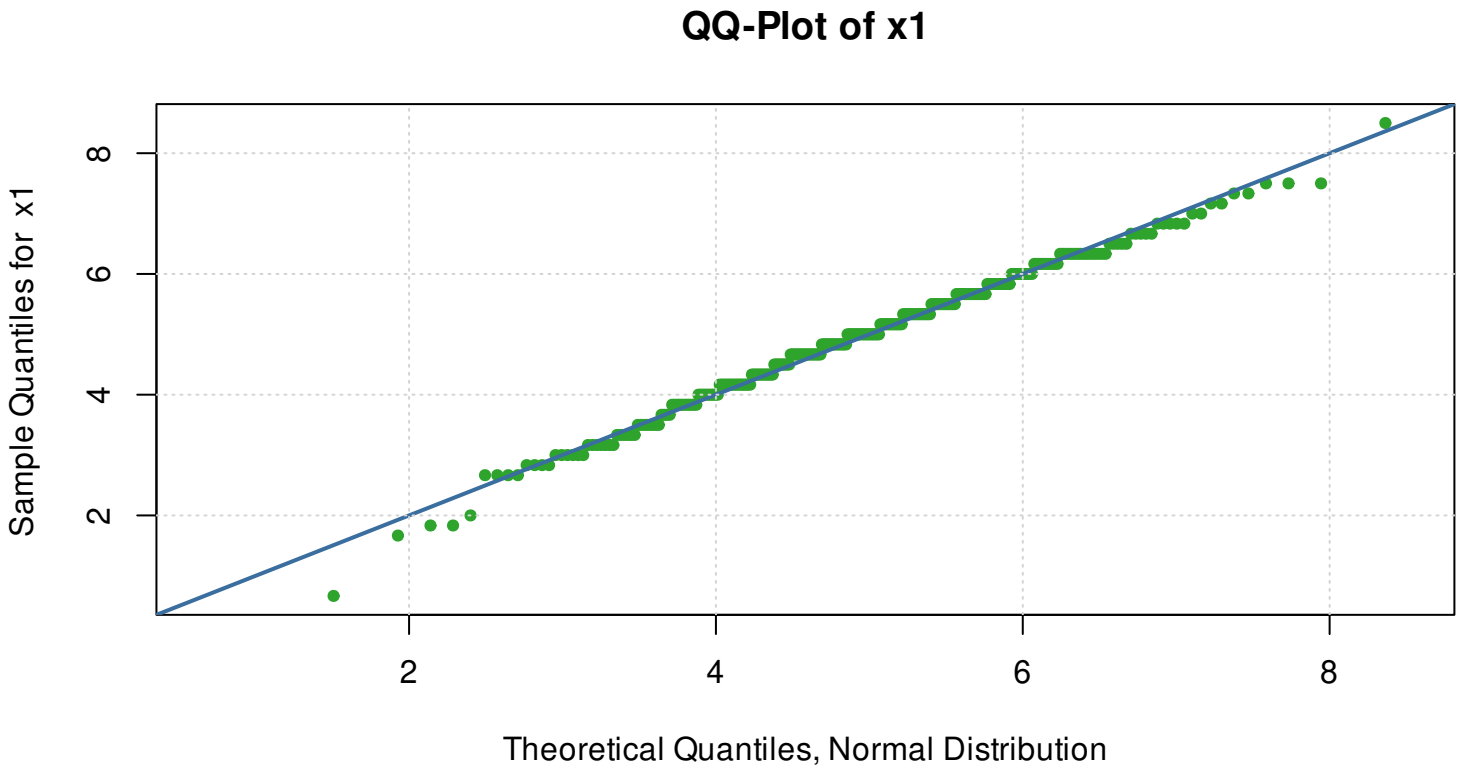


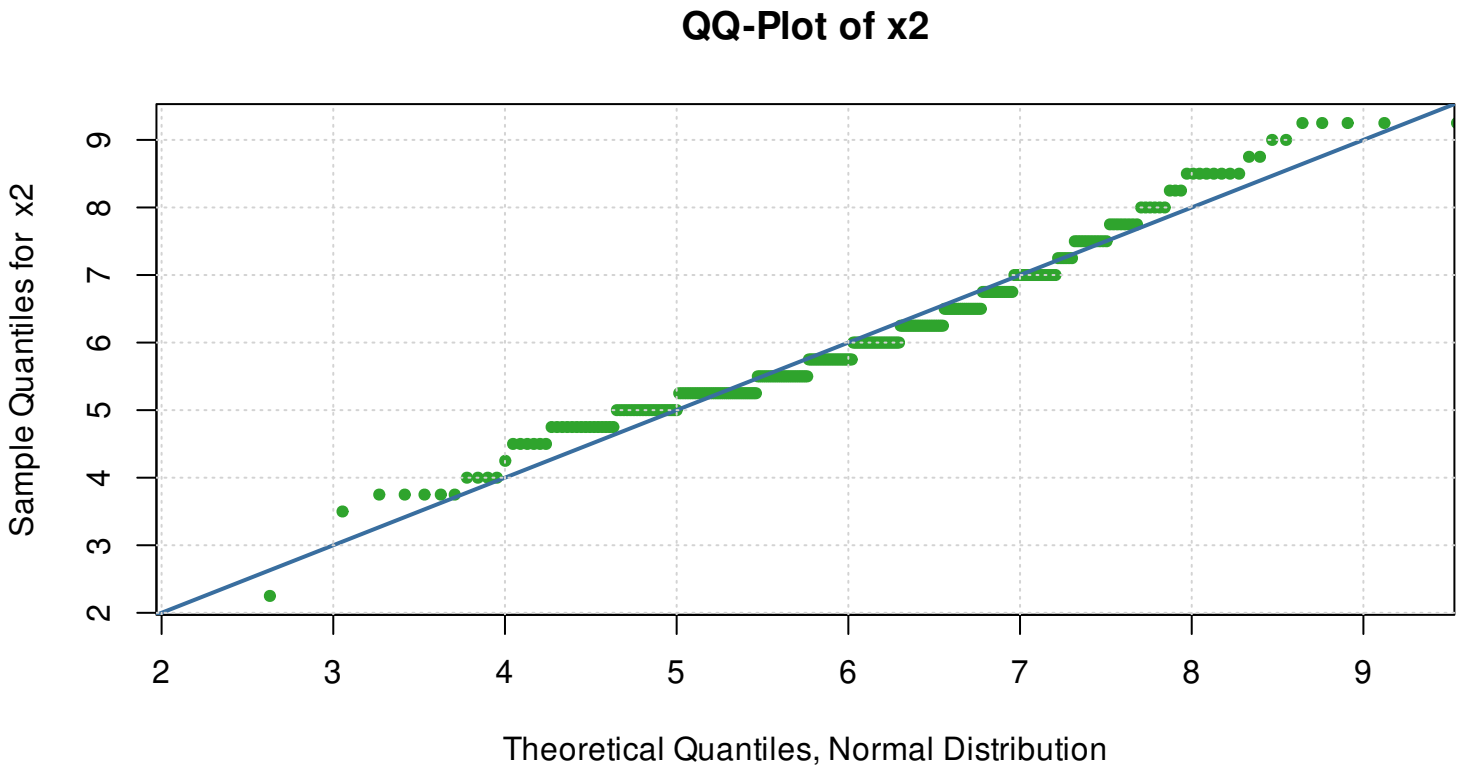
QQ-Plots

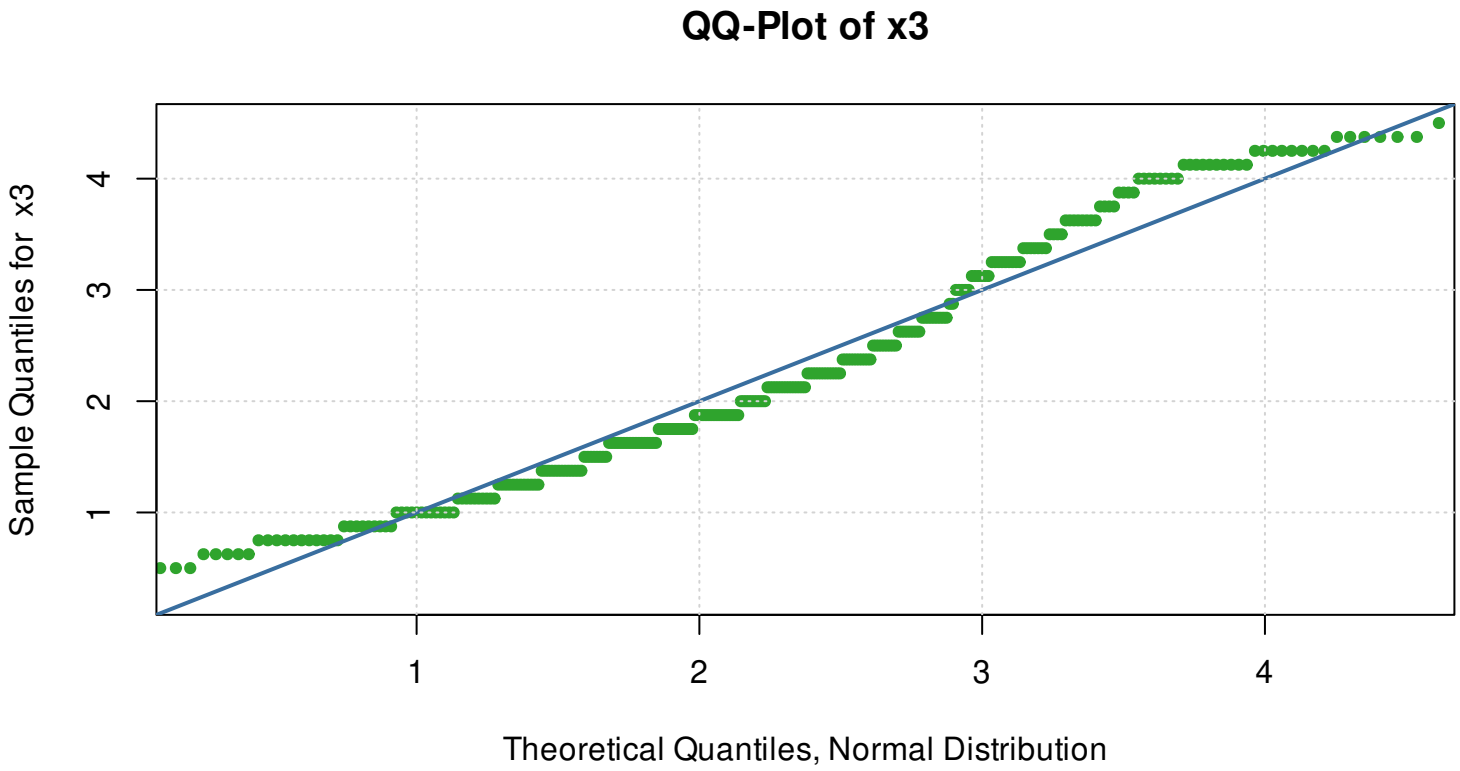
One QQ-Plot per page for each variable. Variables are sorted alphabetically.

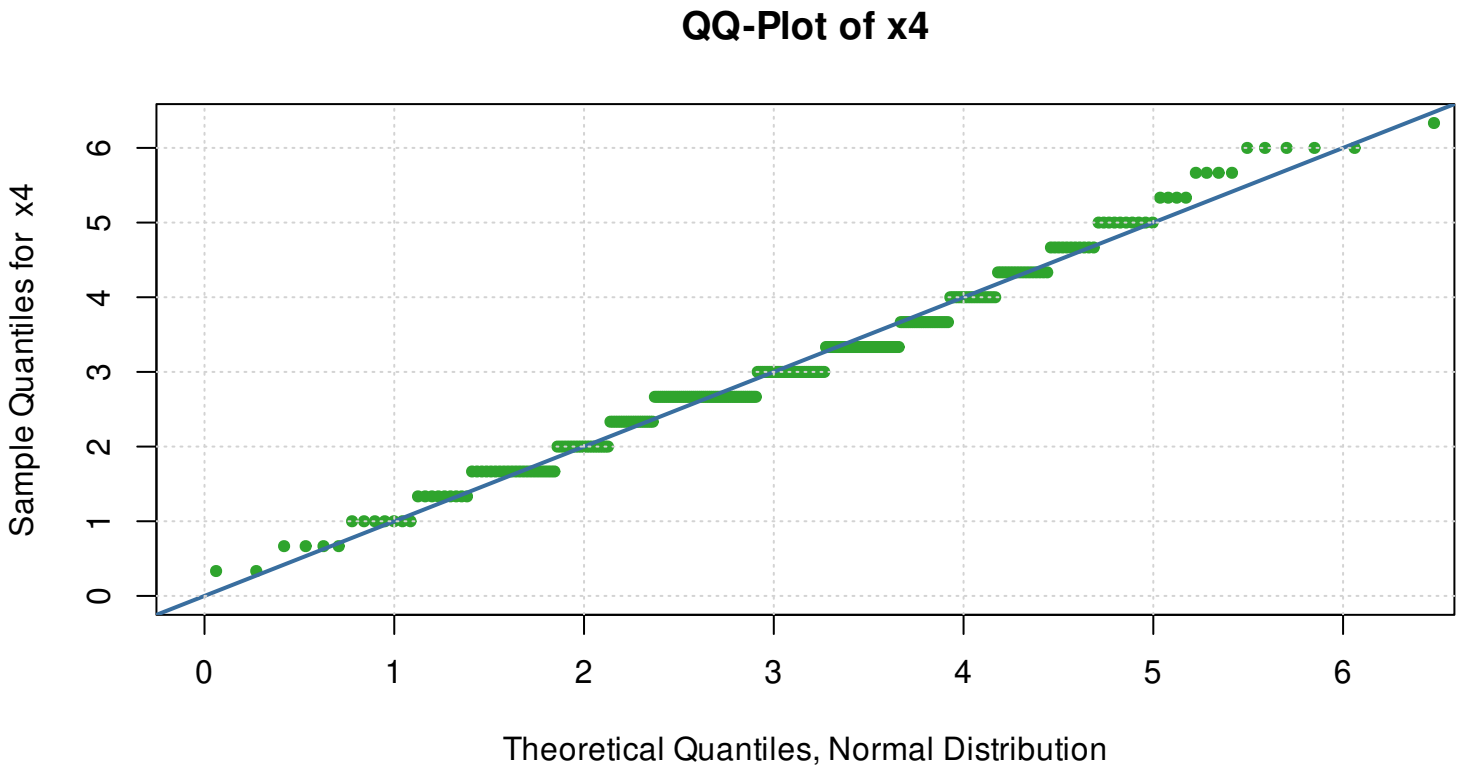


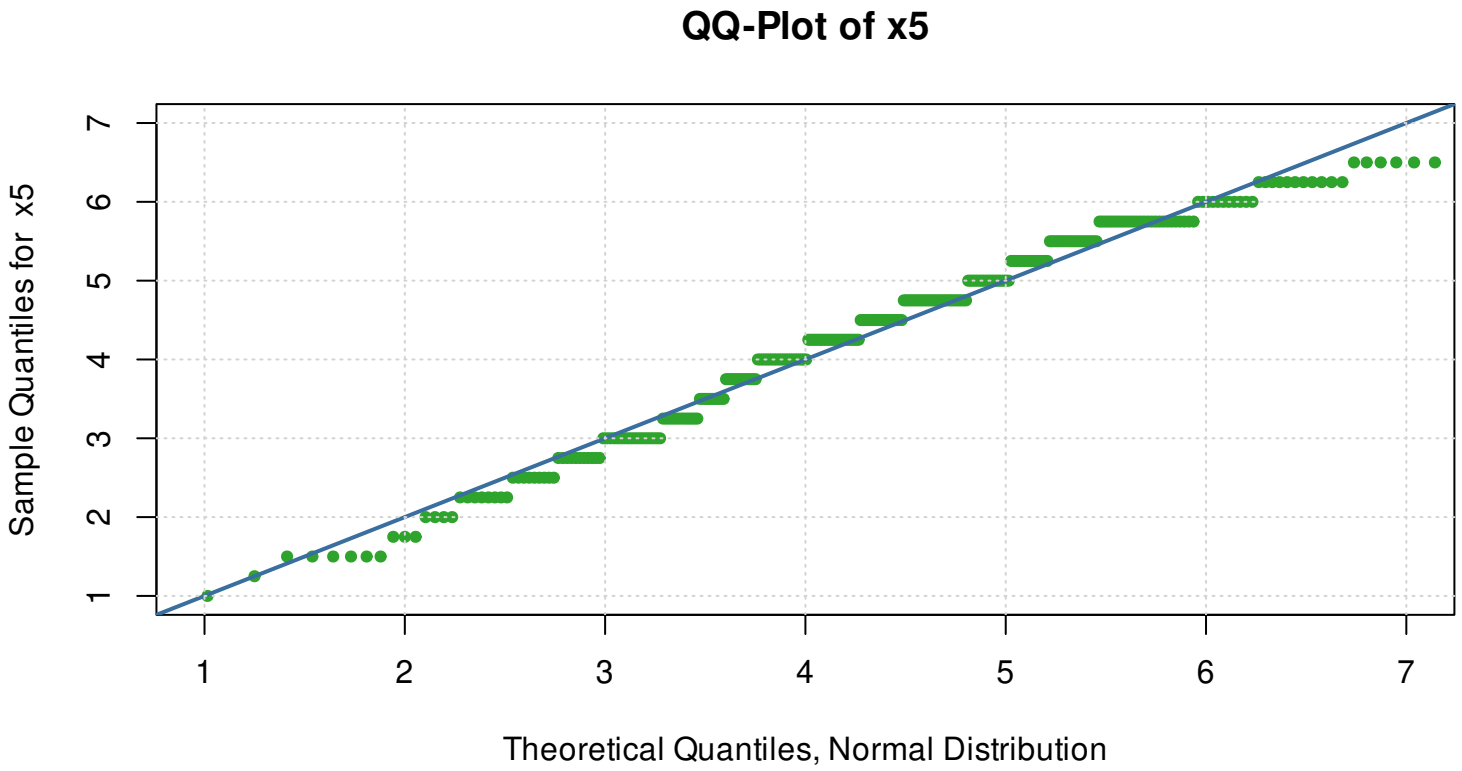

QQ-Plot of V1
Sample Quantiles for V1
0
50
100
200
300
0
50
100
150
200
250
300
Theoretical Quantiles, Normal Distribution

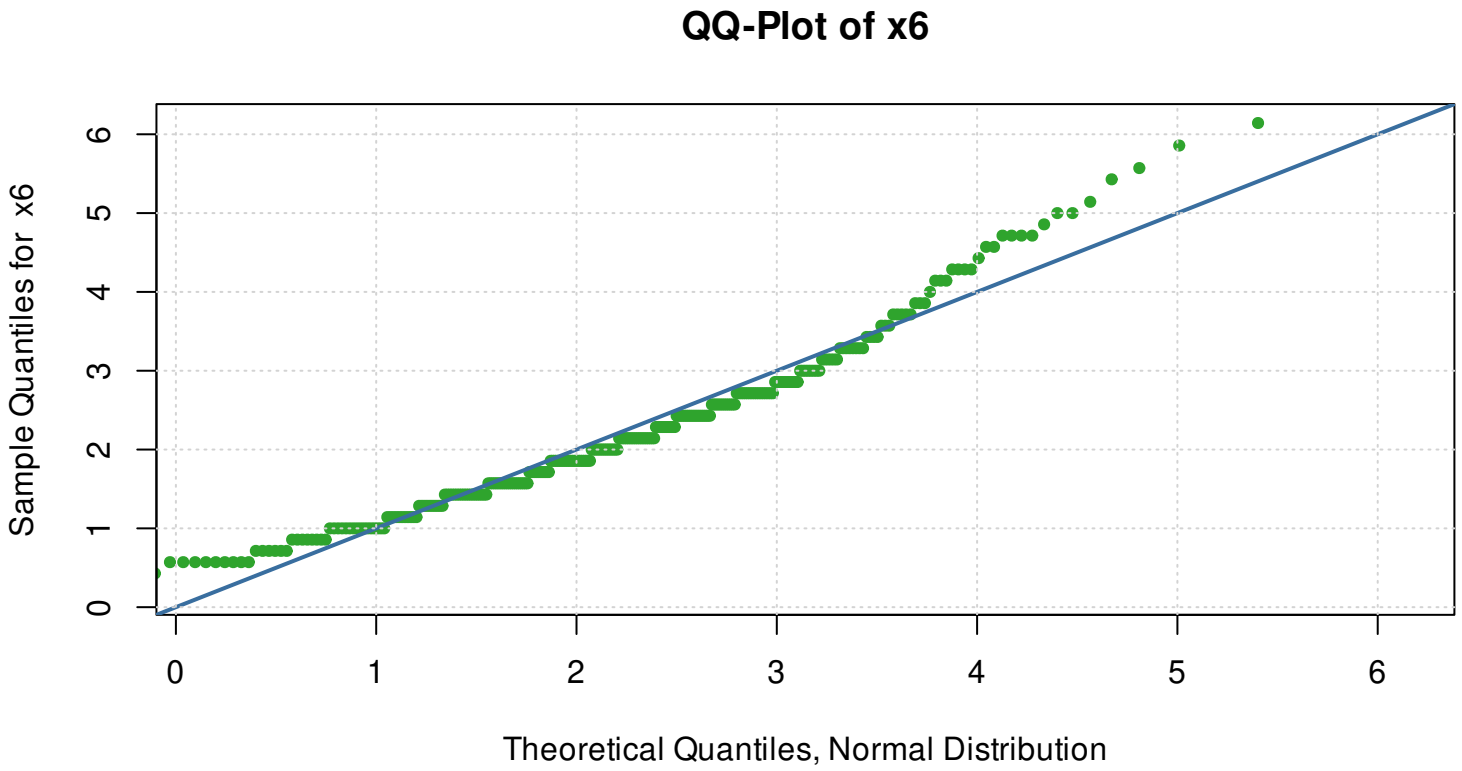

QQ-Plot of x1
Sample Quantiles for x1
2
4
6
8
Theoretical Quantiles, Normal Distribution
2
4
6
8

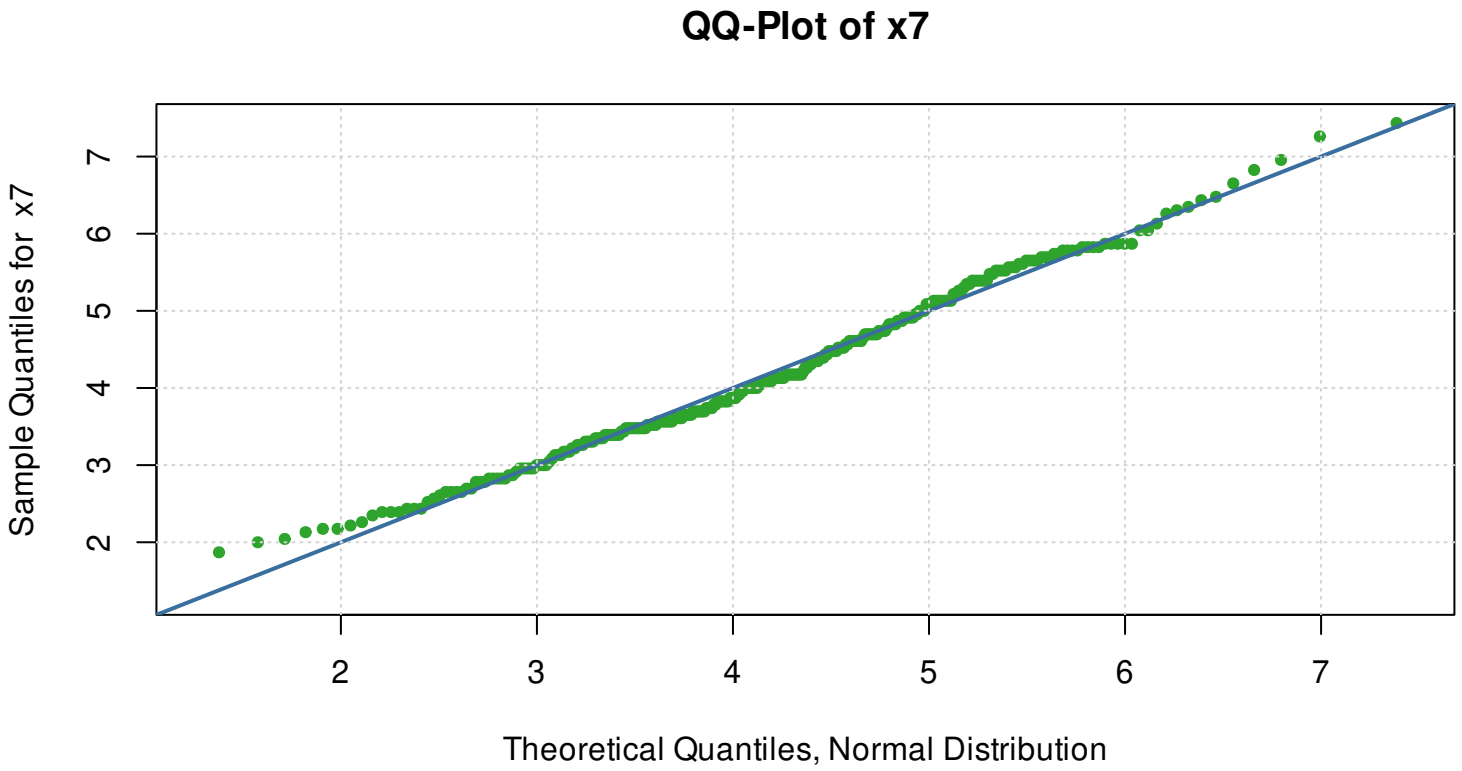

QQ-Plot of x2
Sample Quantiles for x2
Theoretical Quantiles, Normal Distribution
2
3
4
5
6
7
8
9

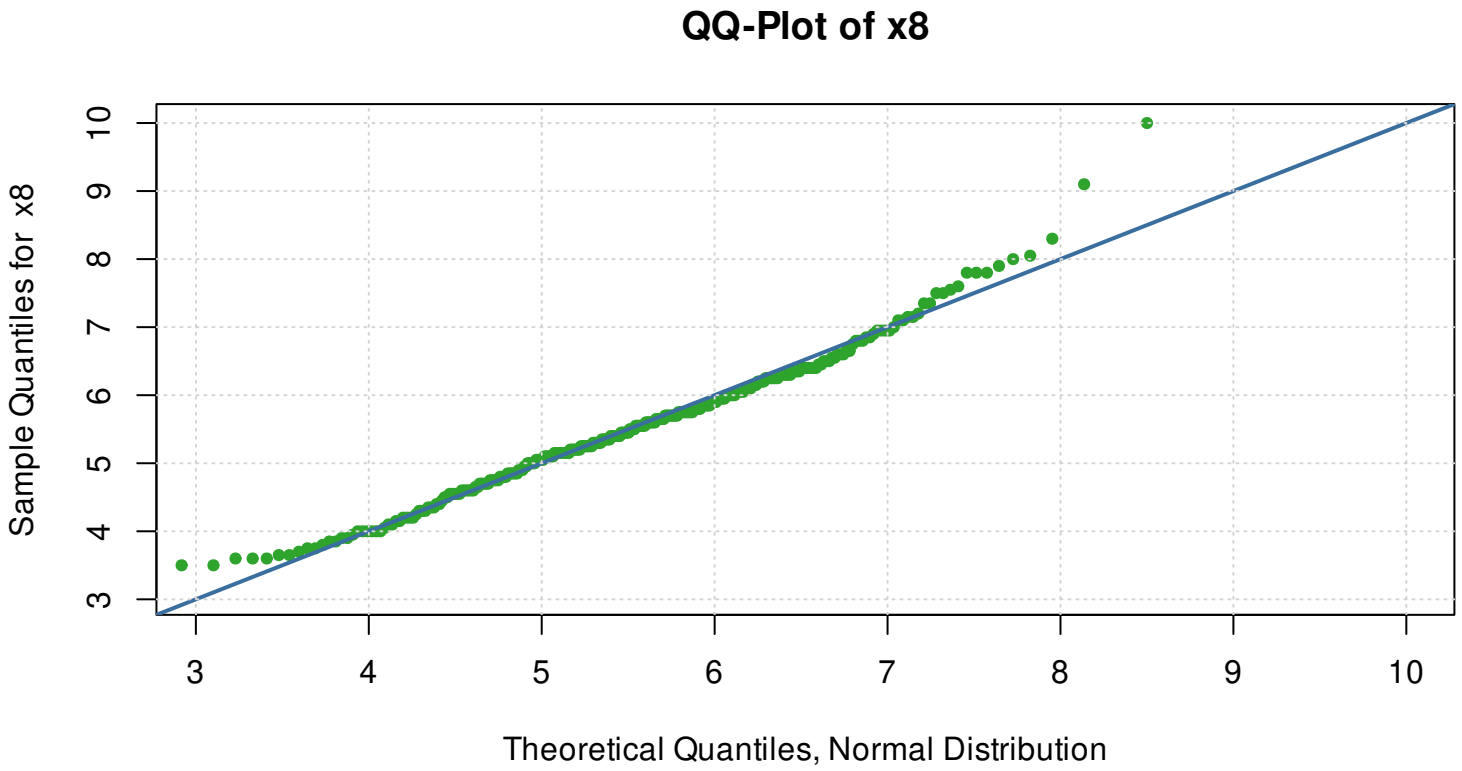

QQ-Plot of x3
Sample Quantiles for x3
1
2
3
4
1
2
3
4
Theoretical Quantiles, Normal Distribution

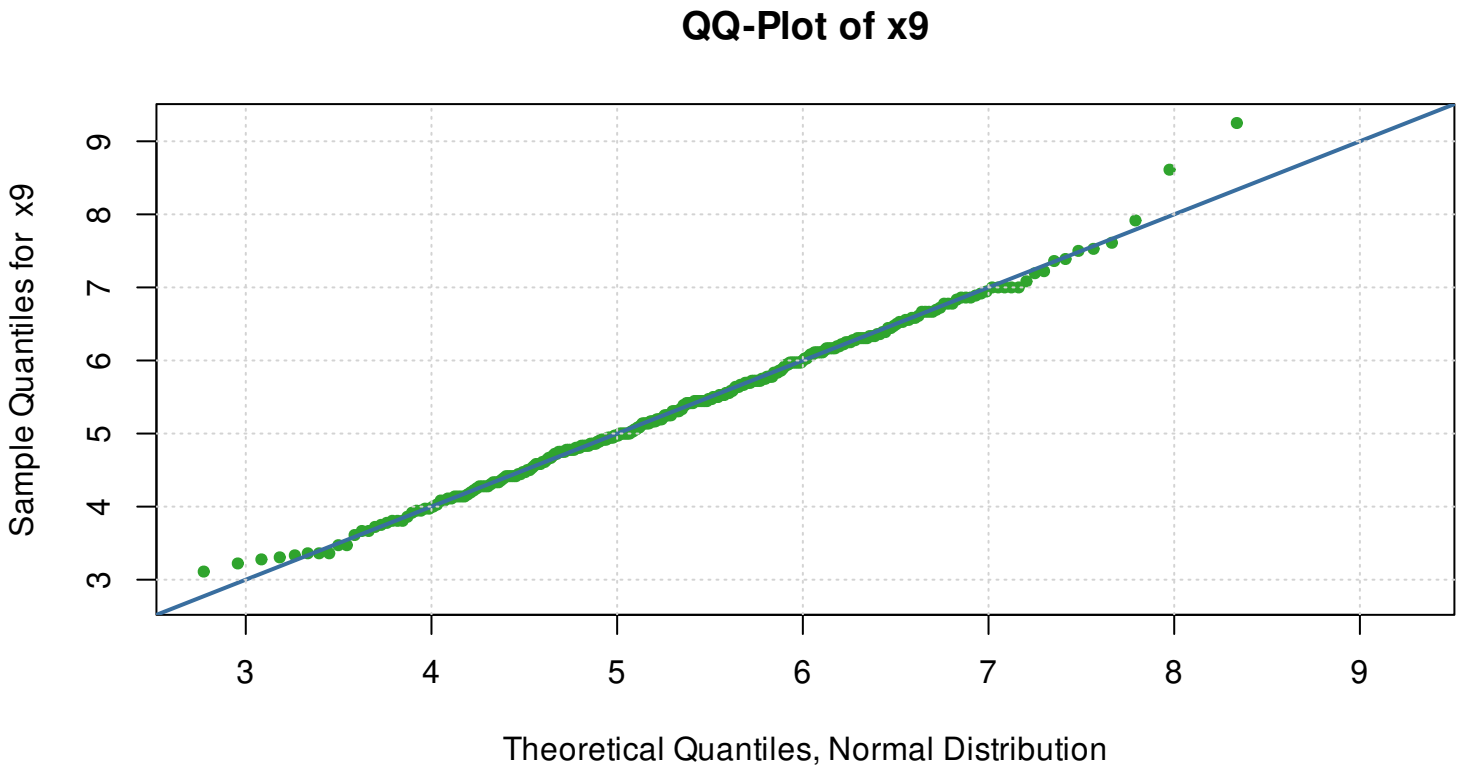

QQ-Plot of x4
Sample Quantiles for x4
Theoretical Quantiles, Normal Distribution
0
1
2
3
4
5
6


QQ-Plot of x5
Sample Quantiles for x5
Theoretical Quantiles, Normal Distribution
1
2
3
4
5
6
7


QQ-Plot of x6
Sample Quantiles for x6
Theoretical Quantiles, Normal Distribution
0
1
2
3
4
5
6

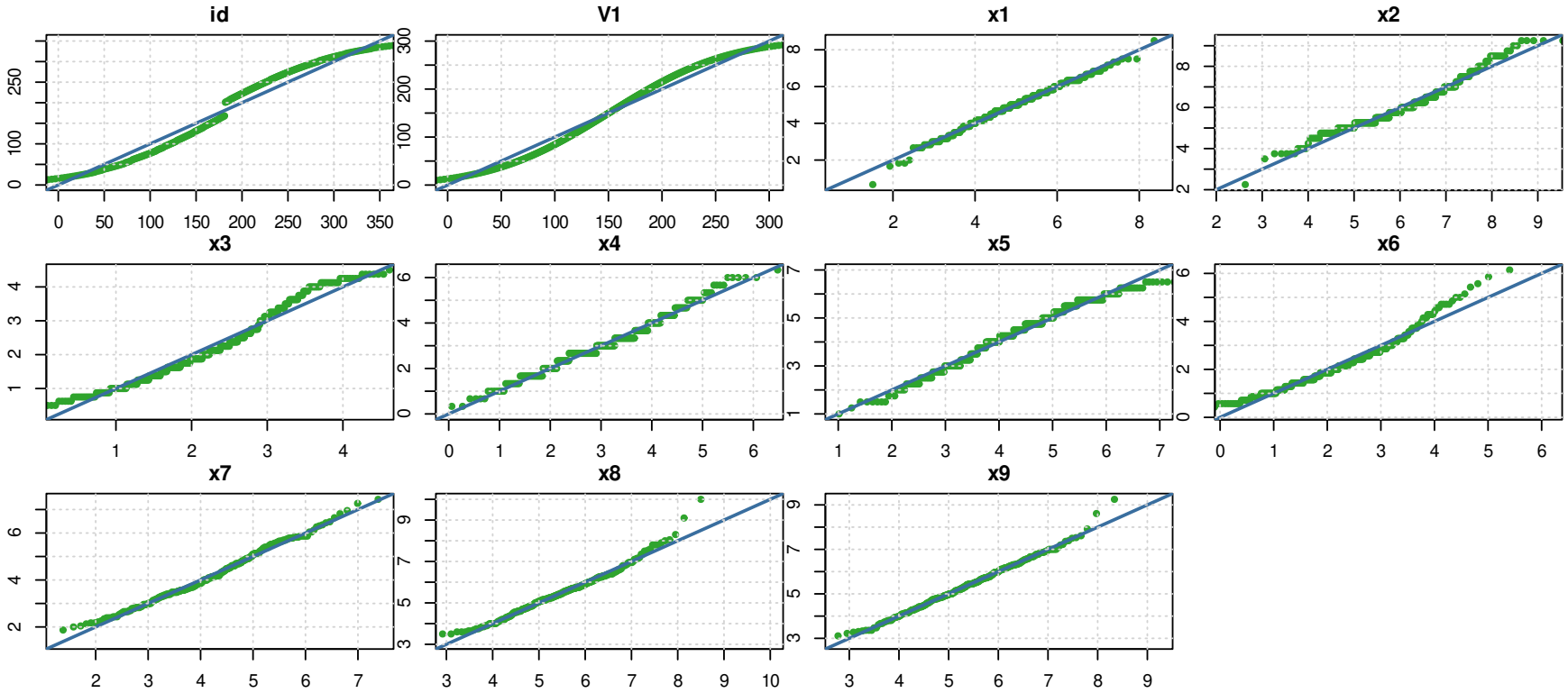

QQ-Plot of x7
Sample Quantiles for x7
2
3
4
5
6
7
Theoretical Quantiles, Normal Distribution


QQ-Plot of x8
Sample Quantiles for x8
3
4
5
6
7
8
9
10
Theoretical Quantiles, Normal Distribution


QQ-Plot of x9
Sample Quantiles for x9
Theoretical Quantiles, Normal Distribution
3
4
5
6
7
8
9

QQ-Plots Summary

QQ-Plots of variables in one figure. Theoretical Quantiles of the Normal Distribution.



Results for Discrete Variables

Descriptive Statistics

Totals

The table is sorted by the variable name. If any, N Unique contains the missing category.

Variable	N Obs	N Missing	N Valid	% Complete	N Unique
agemo	301	0	301	100.00	12
ageyr	301	0	301	100.00	6
grade	301	1	300	99.67	3
school	301	0	301	100.00	2
sex	301	0	301	100.00	2

Frequencies

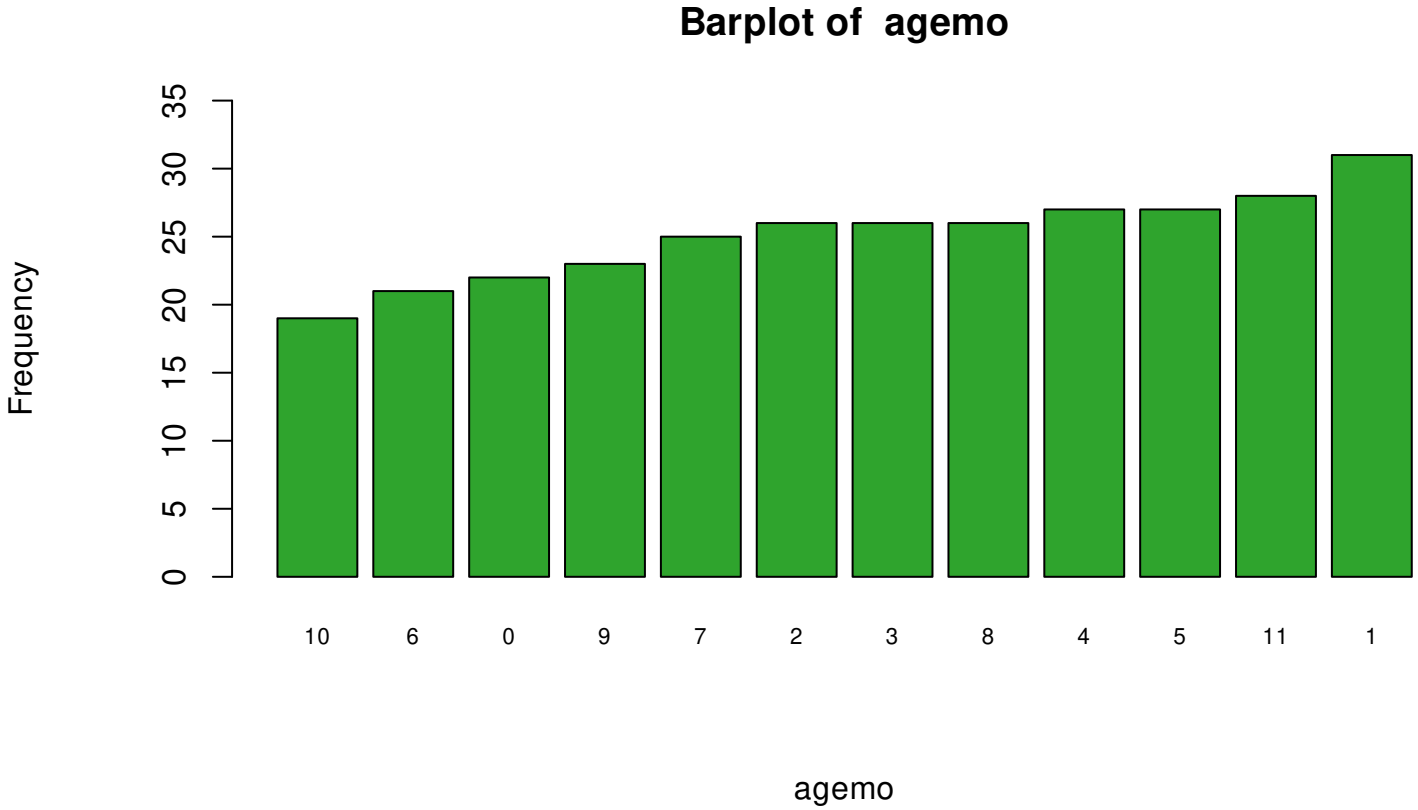
The table is sorted by the variable name. For each variable, a maximum of 20 unique values are considered, sorted in decreasing order of their frequency. If any, missings are counted as a category.

Variable	Category	Frequency	Percent
agemo	1	31	10.30
agemo	11	28	9.30
agemo	4	27	8.97
agemo	5	27	8.97
agemo	2	26	8.64
agemo	3	26	8.64
agemo	8	26	8.64
agemo	7	25	8.31
agemo	9	23	7.64
agemo	0	22	7.31
agemo	6	21	6.98
agemo	10	19	6.31
ageyr	13	110	36.54
ageyr	12	101	33.55
ageyr	14	55	18.27
ageyr	15	20	6.64
ageyr	11	8	2.66
ageyr	16	7	2.33
grade	7	157	52.16
grade	8	143	47.51
grade	Missing	1	0.33
school	Pasteur	156	51.83
school	Grant-White	145	48.17
sex	2	155	51.50
sex	1	146	48.50

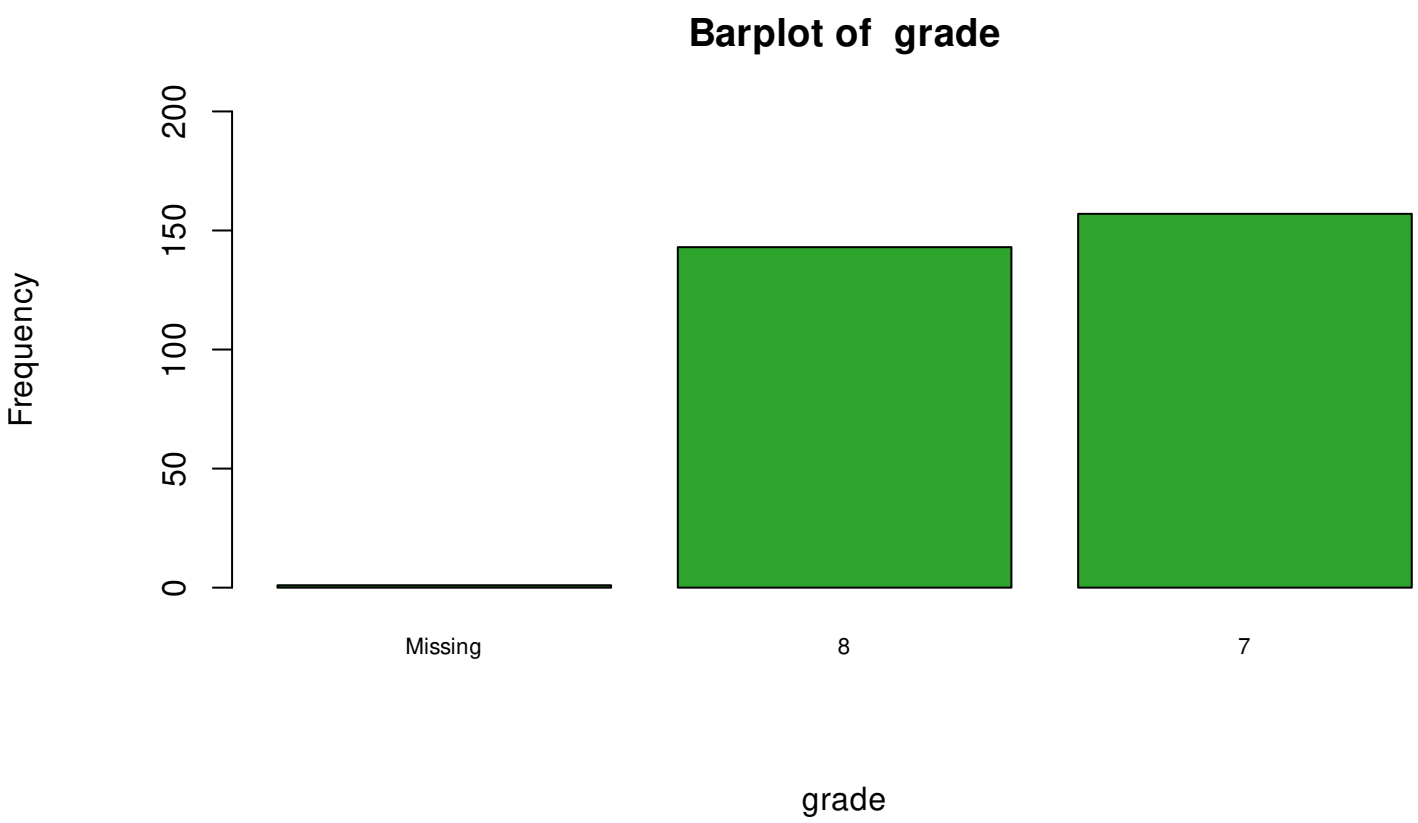
Graphics

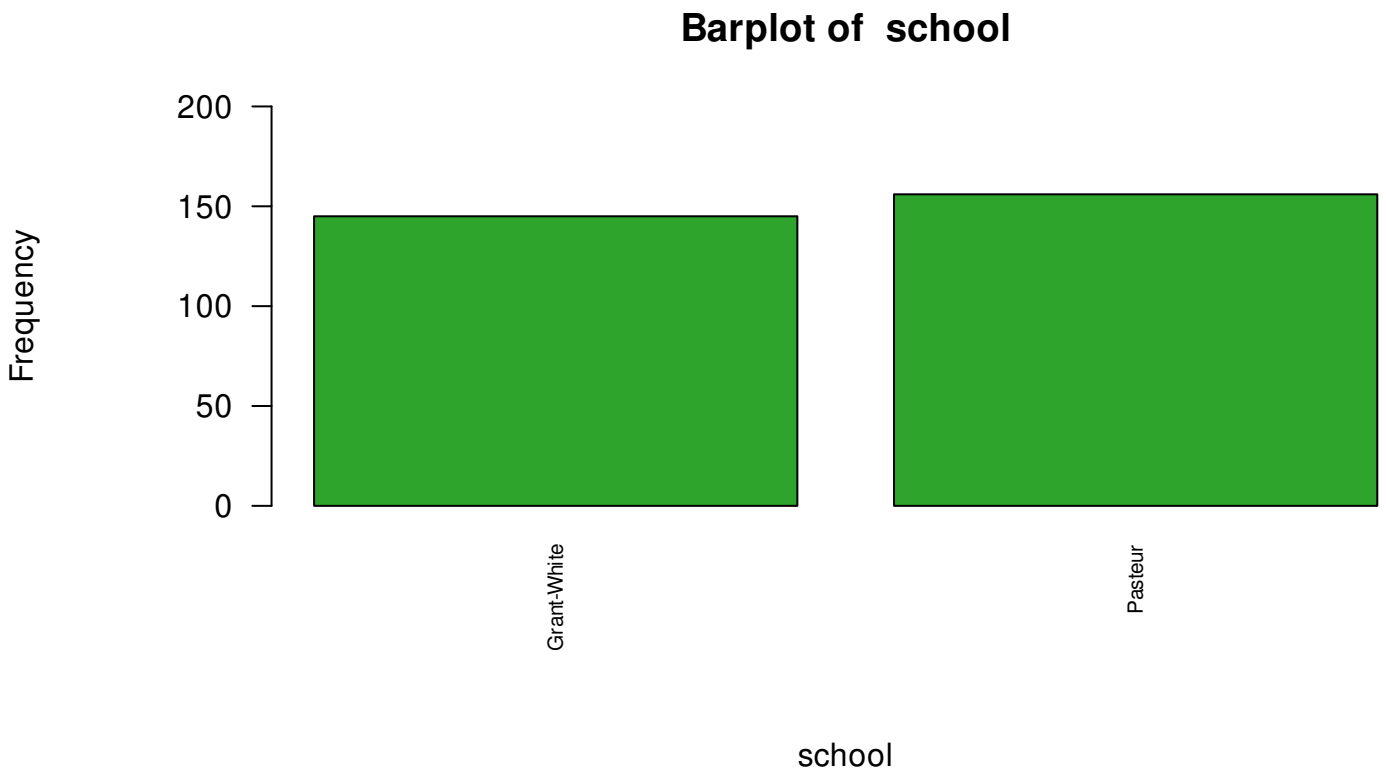
Bar-Plots

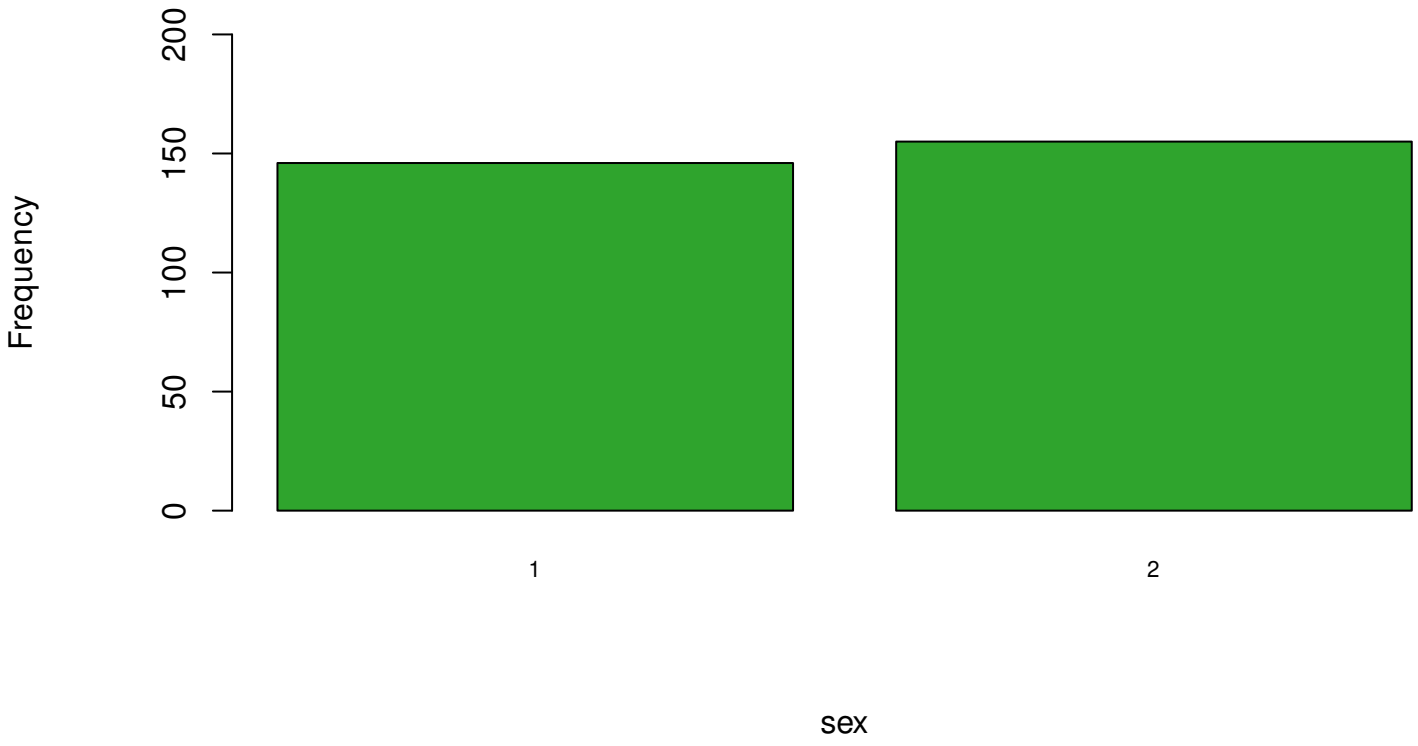
One Bar-Plot per page for each variable. Variables are sorted alphabetically.




Barplot of ageyr
Frequency
0
20
40
60
80
100
16
11
15
14
12
13
ageyr

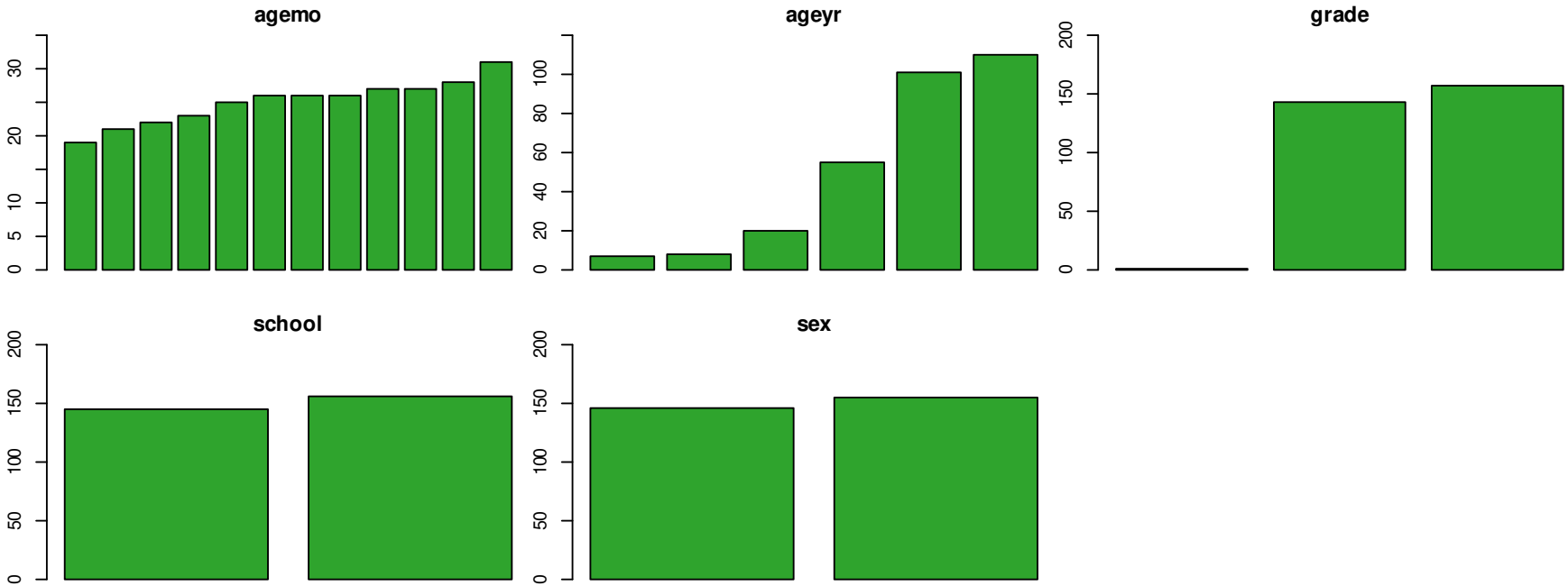

Barplot of grade
Frequency
0
50
100
150
200
Missing
8
7
grade


Barplot of school
Frequency
200
150
100
50
0
Grant-White
Pasteur
school


Barplot of sex
Frequency
0
50
100
150
200
1
2
sex

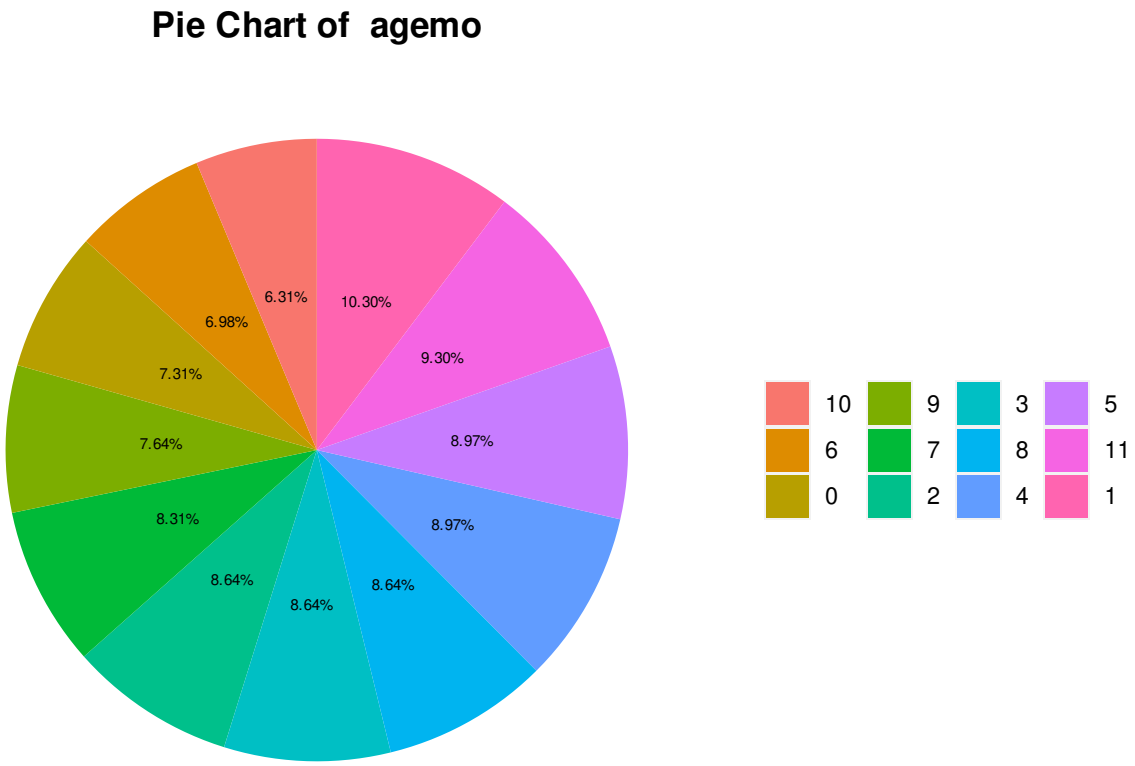
Bar-Plots Summary

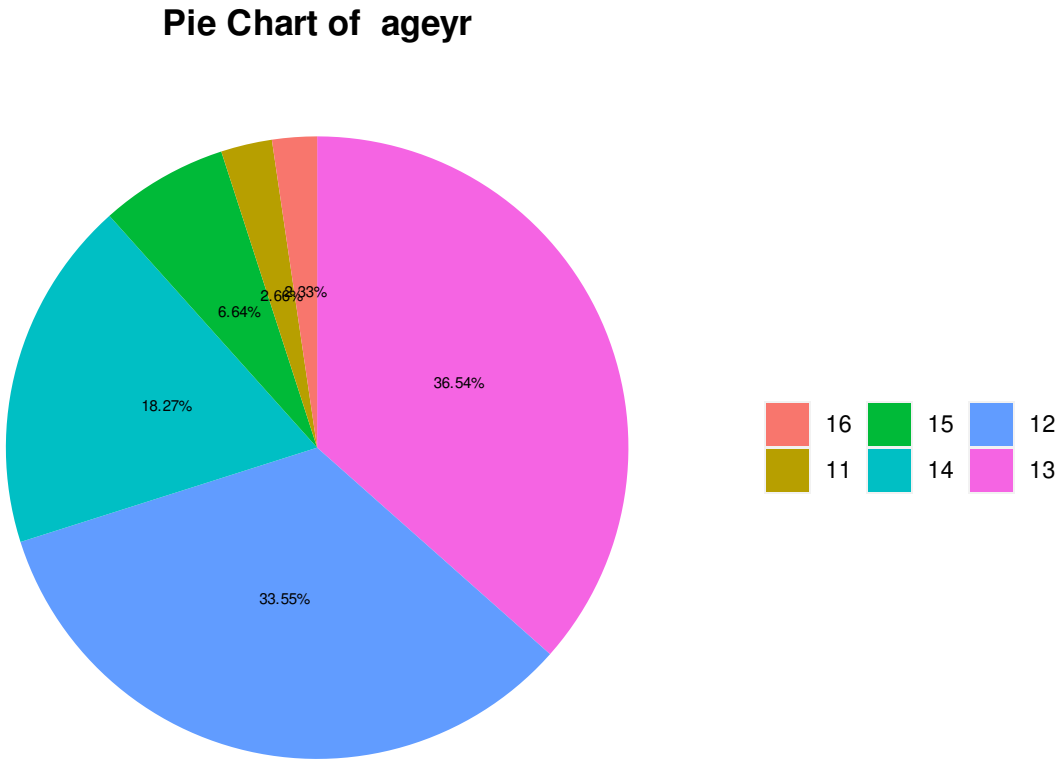
Multiple Bar-Plots of variables in one figure. Variables are sorted alphabetically.

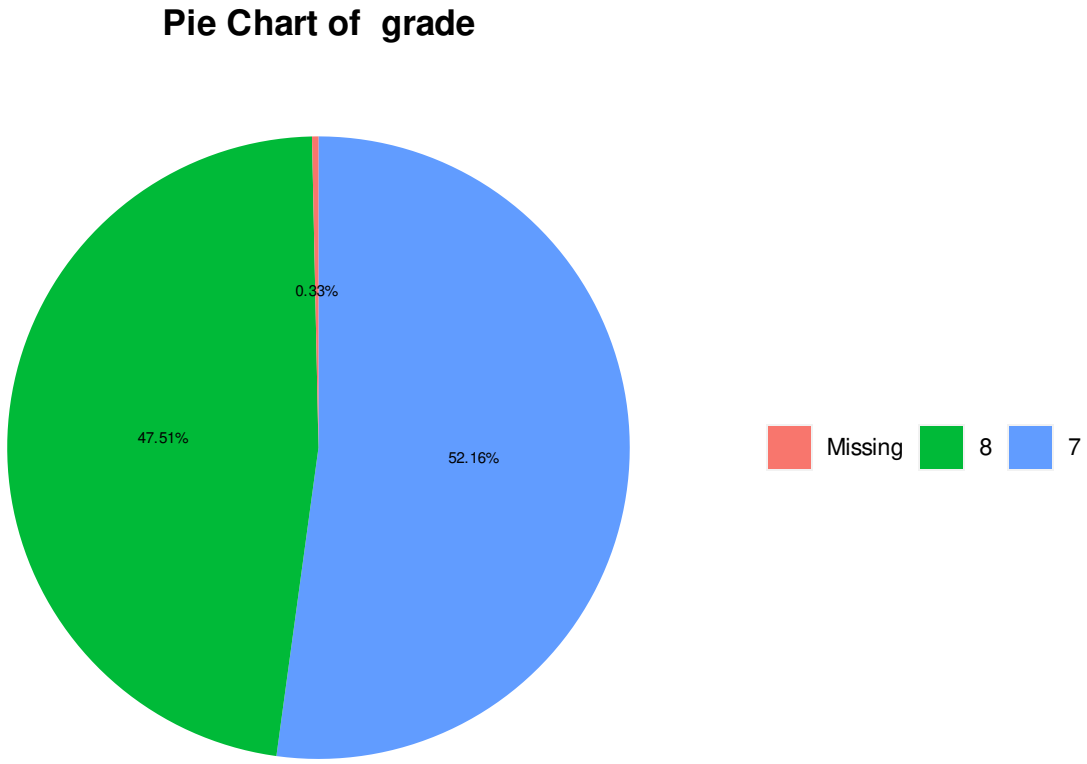


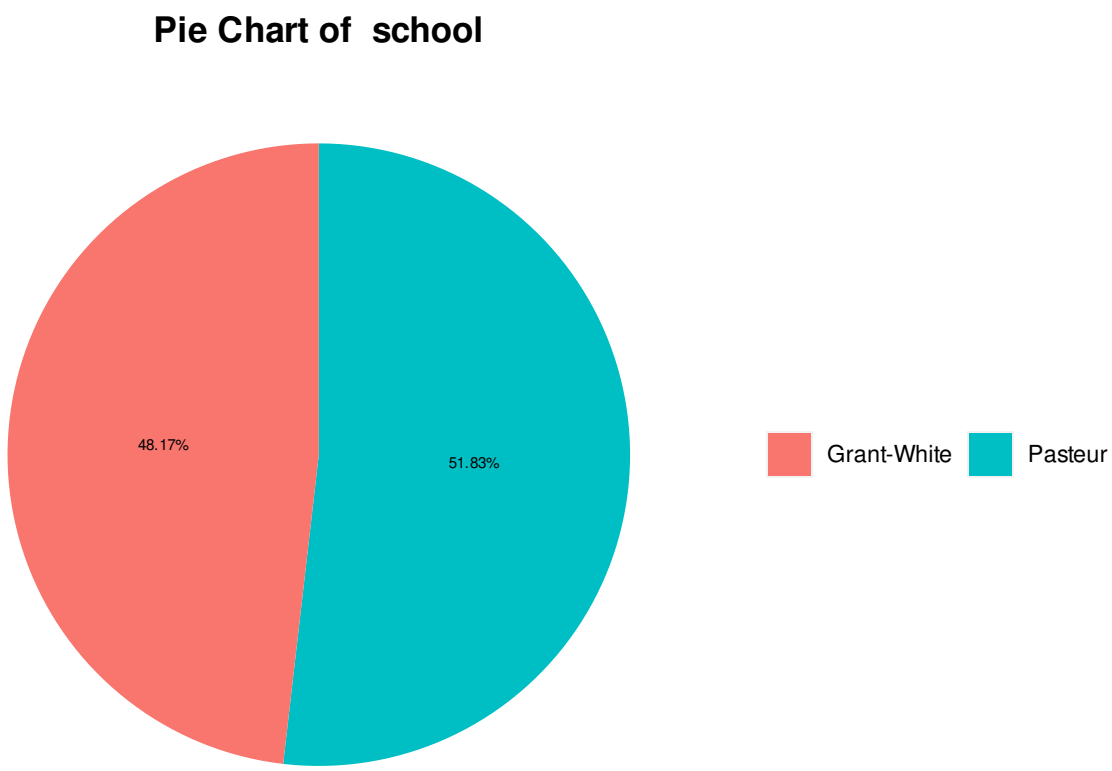
Pie Plots

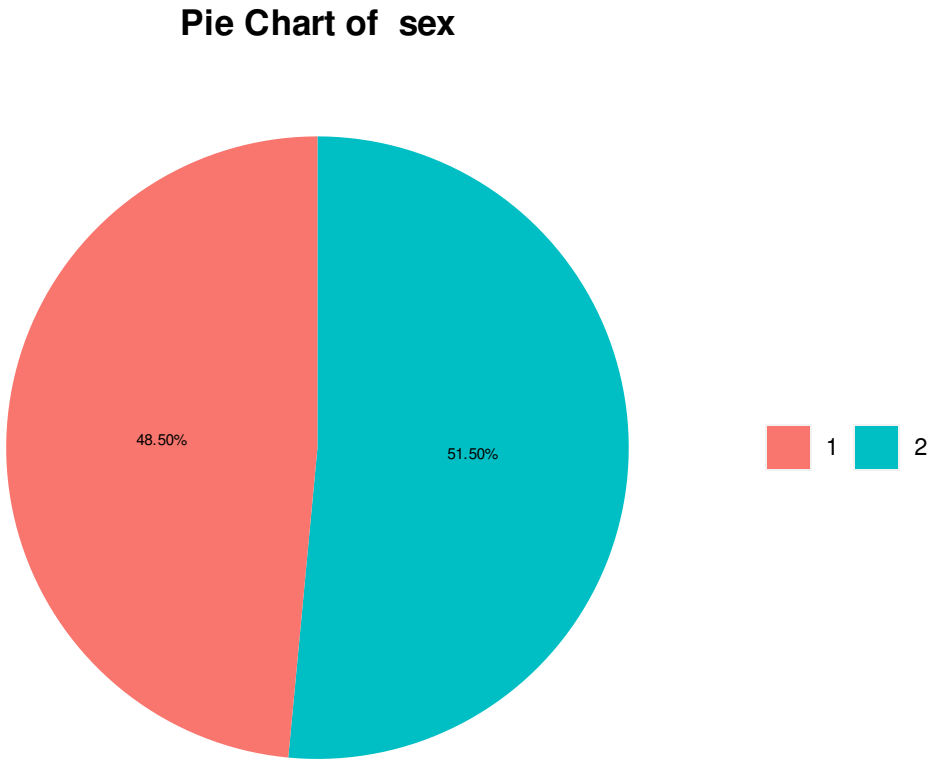
One Pie Plot per page for each variable. Variables are sorted alphabetically.




Pie Chart of ageyr
2.66%
6.64%
18.27%
36.54%
33.55%
16
15
12
11
14
13


Pie Chart of grade
0.33%
47.51%
52.16%
Missing
8
7

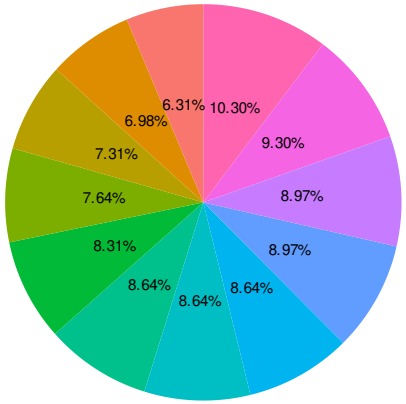

Pie Chart of school
48.17%
51.83%
Grant-White
Pasteur


Pie Chart of sex
48.50%
51.50%
1
2

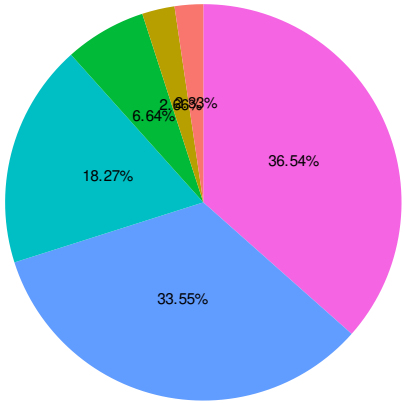
Pie Plots Summary

Multiple Pie Plots of variables in one figure. Variables are sorted alphabetically.

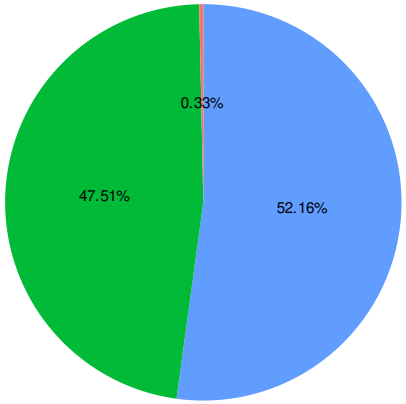
agemo



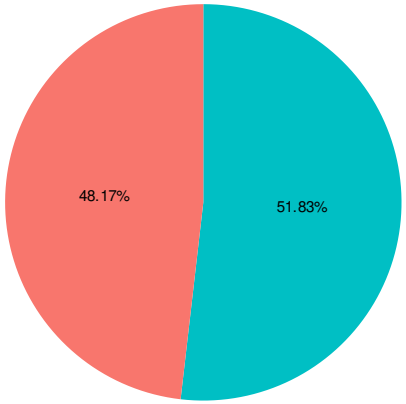
ageyr



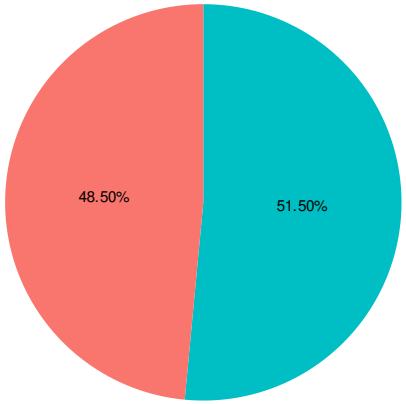
grade



school



sex



R Packages

To run the code you need to install following R packages:

R version: 4.0.3
Package car, version: 3.0.10
Package data.table, version: 1.12.8
Package ggplot2, version: 3.3.3
Package gridExtra, version: 2.3
Package Hmisc, version: 4.4.2
Package knitr, version: 1.31
Package PerformanceAnalytics, version: 2.0.4
Package psych, version: 2.0.12
Package reshape2, version: 1.4.4

R Code

Instructions
If not already available, please install R, RStudio and the requiered packages listed on the previous page. Copy the code below, paste it carefully in a new R Script within RStudio. For a seamless copy & paste process, open the PDF report in a browser. Change the path to your data in the line `filepath <- ...` For Windows users, avoid using paths containing spaces. Run the code. Explore the results (numerical results in the Console, plots in the Plots tab).

```
# Import required libraries
suppressPackageStartupMessages(library(data.table))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(psych))
suppressPackageStartupMessages(library(Hmisc))
suppressPackageStartupMessages(library(reshape2))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(PerformanceAnalytics))
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(car))

# Make a copy of current graphical settings
opar <- par(no.readonly = TRUE)

# Define the path to your data (please remark the forward slash)
filepath <-"C:/HolzingerSwineford1939.csv"

# Upload the data
df <- fread(filepath, header ="auto", sep ="auto", dec =".", encoding ="unknown", data.table = FALSE, na.strings = "")

# Convert characters to UTF-8 encoding
## Depending on your local R settings
## you could try to ignore and skip the next 4 lines
colnames(df) <- iconv(colnames(df),"ASCII","UTF-8")
col_names <- sapply(df, is.character)
df[ ,col_names] <- sapply(df[, col_names], function(col) iconv(col, "ASCII", "UTF-8"))

# Column names of selected continuous variables
colnames_continuous = c(1,2,8,9,10,11,12,13,14,15,16)

# Data frame of the continuous variables
df_num <- df[ ,colnames_continuous, drop=FALSE]

# Column names of selected categorical variables
colnames_categorical = c(3,4,5,6,7)

# Data frame of the categorical variables
df_factor <- df[ ,colnames_categorical, drop=FALSE]
```

```
# Continuous variables
## Descriptive statistics

### Take over summary from psych package and add new stats
stats_new <- psych::describe(df_num)

### Drop some stats which we do not need
stats_new <- as.data.frame(stats_new)
stats_new <- stats_new[c(-1,-6,-10,-13)]

### Add new stats
stats_new$Variable <- colnames(df_num)
stats_new$ntotal <- nrow(df_num)

### Missings
stats_new$miss <- sapply(df_num, function(col) sum(is.na(col)))

### Complete rate
stats_new$complete <- sapply(df_num, function(col) (1-(sum(is.na(col)) / nrow(df_num)))*100)

### N Unique
stats_new$N_Unique <- sapply(df_num, function(col) length(unique(na.omit(col))))

### CV
stats_new$CV <- sapply(df_num, function(col) {
  ifelse(any(col <= 0, na.rm=TRUE), "-", round((sd(col, na.rm=TRUE) / mean(col, na.rm=TRUE)),2))
  })

### Reorder columns
stats_new <- stats_new[,c(10,11,12,1,13,14,2:9,15)]

### Column names
colnames(stats_new) <- c("Variable", "N Obs", "N Missing", "N Valid", "% Complete", "N Unique", "Mean",
                         "SD", "Median", "MAD", "MIN", "MAX", "Skewness", "Kurtosis", "CV")

### Order by variable name
stats_new <- stats_new[order(stats_new$Variable),]

### Output
knitr::kable(stats_new, digits=2, row.names = FALSE, format="simple")
```

```
# Continuous variables
## Descriptive graphics: Histograms One Per Page
```

```
### Order by variable name
df_num_order <- df_num[,order(colnames(df_num)),drop=FALSE]

### Function to plot histogram for each variable
single_hist <- function(x, main = "Histogram",
                    ylab="Relative Frequency", xlab=NULL, freq=FALSE, bcol="#2fa42d",
                    dcol=c("#396e9f","#396e9f"), dlty=c("dotted", "solid"),
                    breaks=21) {

  h <-  hist(x, plot=FALSE, breaks=breaks)
  m <- mean(x, na.rm=TRUE)
  s <- sd(x, na.rm=TRUE)
  d <- density(x, na.rm=TRUE)

  # Set nice x and y axis limits
  xlims <- pretty(c(floor(h$breaks[1]),ceiling(last(h$breaks))))
  ymax <- max(h$density)
  dmax <- max(d$y)
  ymax <- max(ymax,dmax)

  # Plots
  plot(h, freq=freq, ylim=c(0, ymax*1.2), ylab=ylab, xlab=xlab,
       main=main, col=bcol,  xlim = c(min(xlims), max(xlims)))
  lines(d, lty=dlty[1], col=dcol[1])
  curve(dnorm(x,m,s), add=TRUE, lty=dlty[2], col=dcol[2])

}


### Loop over variables
for (i in 1:ncol(df_num)){
  single_hist(df_num_order[,i], main = paste("Histogram of ", colnames(df_num_order[i])))
}
# Continuous variables
## Descriptive graphics: Histograms Summary
k <- ceiling(ncol(df_num)/20)-1
for (i in 0:k){
  m <- 20*i+1
  n <- min(20*(i+1),ncol(df_num))
  multi.hist(df_num_order[,m:n], dcol=c("#396e9f","#396e9f"),
             bcol= "#2fa42d",
```

```
            dlty=c("dotted", "solid"),
            main = colnames(df_num_order[,m:n]))
}
```

```
# Continuous variables
## Descriptive graphics: Box-Plot One Per Page

### Loop over variables
for (i in 1:ncol(df_num)){
  boxplot(df_num_order[,c(i)], col = "#2fa42d",
       main = paste("Boxplot of",colnames(df_num_order[i])),
       xlab=paste(colnames(df_num_order[i])), horizontal = TRUE)
}
```

```
# Continuous variables
## Descriptive graphics: Box-Plots Summary

### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))), ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)){
  boxplot(df_num_order[,c(i)], col = "#2fa42d", main = colnames(df_num_order[i]),
          xlab=paste(colnames(df_num_order[i])), xaxt="n", horizontal = TRUE)
}

### Restore original graphical settings
par(opar)
```

```
# Continuous variables
## Descriptive graphics: ECDF Plots One Per Page

### Loop over variables
for (i in 1:ncol(df_num)){

  data <- as.data.frame(df_num_order[,c(i)])
  colnames(data) <- "variable"

  # Plot ECDF
  step_function <- ecdf(data$variable)
  plot(step_function,
      main=paste("ECDF Plot of", colnames(df_num_order[i])),
      xlab=colnames(df_num_order[i]), ylab="ECDF",
```

```
      cex=0.7, col="#2fa42d", do.points=TRUE)

  # Plot CDF of normal distribution
  data_mean<- mean(data$variable, na.rm=TRUE)
  data_sd<- sd(data$variable, na.rm=TRUE)
  curve(pnorm(x, data_mean,data_sd),
        from=qnorm(0.0001, mean=data_mean, sd=data_sd),
        to=qnorm(0.9999, mean=data_mean, sd=data_sd),
        add=TRUE, col="#396e9f", lwd=2)
}
```

```
# Continuous variables
## Graphics: ECDF Plots Summary

### ECDF function
ecdf_plot <- function(i){

  data <- as.data.frame(df_num_order[,c(i)])
  colnames(data)<-"variable"

  # Plot ECDF
  step_function <- ecdf(data$variable)
  ecdf_plot <- plot(step_function,
                    main = colnames(df_num_order[i]),
                    xlab = colnames(df_num_order[i]), ylab = "ECDF",
                    cex = 0.7, col="#2fa42d", do.points = FALSE)

  # Plot CDF of normal distribution
  data_mean <- mean(data$variable, na.rm=TRUE)
  data_sd <- sd(data$variable, na.rm=TRUE)
  curve(pnorm(x, data_mean,data_sd),
        from = qnorm(0.0001, mean = data_mean, sd = data_sd),
        to = qnorm(0.9999, mean = data_mean, sd = data_sd),
        add = TRUE, col="#396e9f", lwd=0.5,pch=1)
}


### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))), ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)) ecdf_plot(i)
```

```
### Restore original graphical settings
par(opar)
```

```
# Continuous variables
## Graphics: QQ Plots One Per Page

### Define function for the QQ-Plot
qq_plot <- function(i, main, xlab, ylab){
    var <- df_num_order[,i]
    qqplot(x = qnorm(ppoints(var), mean = mean(var, na.rm = TRUE),
                     sd = sd(var, na.rm = TRUE)),
         y = var,
         xlim = c(min(var, na.rm = TRUE), max(var, na.rm = TRUE)),
         ylim = c(min(var, na.rm = TRUE), max(var, na.rm = TRUE)),
         main = main,
         xlab = xlab,
         ylab = ylab,
         col = "#2fa42d", cex=0.7, pch=19
         )
    abline(a = 0, b = 1, col = "#396e9f", lwd = 2)
    grid()
}

### Loop over variables
for (i in 1:ncol(df_num)){
  qq_plot(i, main = paste("QQ-Plot of", colnames(df_num_order[i])),
            xlab = "Theoretical Quantiles, Normal Distribution",
            ylab = paste("Sample Quantiles for ", colnames(df_num_order[i]))
          )
}
```

```
# Continuous variables
## Graphics: QQ Plots Summary
### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))),
            ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)){
  qq_plot(i, colnames(df_num_order[i]), "", "")
}
```

```
### Restore original graphical settings
par(opar)

# Categorical variables
## Descriptive statistics: Totals

### Totals statistics
miss <- sapply(df_factor, function(col) sum(is.na(col)))
complete <- sapply(df_factor, function(col) (1-(sum(is.na(col)) / nrow(df_factor)))*100)
complete <- round(complete,3)
totals <- data.frame(miss, complete)
totals$Variable <- rownames(totals)
totals$ntotal <- nrow(df_factor)
totals$valid <- totals$ntotal - totals$miss
totals$N_Unique <- sapply(df_factor, function(col) length(unique(col)))
totals <- totals[,c(3,4,1,5,2,6)]
totals <- totals[order(totals$Variable),]
colnames(totals) <- c("Variable", "N Obs", "N Missing", "N Valid", "% Complete","N Unique")

### Output
kable(totals, digits=2, row.names = FALSE, format="simple")

# Categorical variables
## Descriptive statistics: Frequencies

### Function stats per variable
discrete <- function(i){

  # Calculate individual statistics
  count <- table(df_factor[,i], useNA="always")
  perc <- as.data.frame(prop.table(count))
  perc$Percent <- perc$Freq*100
  perc$Freq <- NULL

  # Merge to one dataframe
  freq <- merge(count, perc, by="Var1")
  freq$Variable <- rep(colnames(df_factor)[i],nrow(freq))
  freq <- freq[,c(4,1,2,3)]
  colnames(freq) <- c("Variable", "Category","Frequency", "Percent")

  # Rename missing category
  if(length(is.na(freq$Category))>0){
```

```
    levels(freq$Category) <- c(levels(freq$Category),"Missing")
    freq$Category[is.na(freq$Category)] <- "Missing"
  }

  # Sort
  freq_order <- freq[order(-freq[,4],freq[,2]),]

  # Add category "All other values" in case of more than 20 categories
  min <- min(20, length(unique(df_factor[,i])))
  if(min==20){
    freq_order$Category <- as.character(freq_order$Category)
    freq_order <- rbind(freq_order[1:20,],
                        c(colnames(df_factor)[i], as.character("****All Other Values****"),
                          sum(freq_order$Frequency[-c(1:20)]), sum(freq_order$Percent[-c(1:20)])))
  } else {
    freq_order <- freq_order[1:min,]
  }
  return(freq_order)
}

### Loop over variables
cat_table <- discrete(1)
for (i in 1:ncol(df_factor)){
  if (i>1){
    cat_i <- discrete(i)
    cat_table <- rbind(cat_table, cat_i)
  }
}

### Sort by variable name
cat_table <- cat_table[order(cat_table$Variable),]
cat_table$Percent <- round(as.numeric(cat_table$Percent),2)

### Output
kable(cat_table, digits=2, row.names = FALSE, format="simple")
```

```
# Categorical variables
## Descriptive graphics: Bar-Plots One Per Page

### Data frame sorted by column name
df_factor_order <- df_factor[,order(colnames(df_factor)), drop=FALSE]

### Loop over variables
```

```
for (i in 1:ncol(df_factor)){
  counts <- table(df_factor_order[i], useNA = "ifany")
  names(counts)[is.na(names(counts))] <- "Missing"
  counts <- counts[order(counts)]

  # Plot by case (e.g. category names length)
  if (any(nchar(names(counts), type = "chars") >= 11) || length(counts) > 12){

     if(length(counts) > 40){
        # Bar-Plot with suppressed category names
        par(mar = c(6,6,4.1, 2.1), mgp = c(3, 1, 0))
        barplot(counts, col = "#2fa42d", main = paste("Barplot of ", colnames(df_factor_order[i])), xaxt="n",
                ylab = "Frequency", cex.names = 0.6, las = 2, xlab = colnames(df_factor_order[i]),
                ylim = range(pretty(c(0,counts))))


      } else {
     # Bar-Plot with shortened category names
       par(mar = c(8, 8, 4.1, 2.1), mgp = c(6, 1, 0))
       names(counts) <- substr(names(counts), 1, 15)
       barplot(counts, col = "#2fa42d", main = paste("Barplot of ", colnames(df_factor_order[i])), ylab = "Frequency",
               cex.names = 0.65, xlab= colnames(df_factor_order[i]), las=2, ylim=range(pretty(c(0,counts))))

      }

  } else {
   # Bar-Plot with full-length names
    par(mar = c(6,6, 4.1, 2.1), mgp = c(5, 1, 0))
    barplot(counts, col = "#2fa42d", main = paste("Barplot of ", colnames(df_factor_order[i])), ylab = "Frequency",
            cex.names = 0.7, ylim=range(pretty(c(0,counts))), xlab= colnames(df_factor_order[i]))
   }
}
```

```
# Categorical variables
## Descriptive graphics: Bar-Plots Summary

### Function for Bar-Plot per variable
plot_bar <- function(i){
  counts <- table(df_factor_order[i], useNA = "ifany")
  names(counts)[is.na(names(counts))] <- "Missing"
  names(counts)[names(counts)=="NA"] <- "Missing"
  counts <- counts[order(counts)]
  barplot(counts, col = "#2fa42d", main = colnames(df_factor_order[i]),
```

```
          ylab = "Frequency",  xaxt="n", ylim=range(pretty(c(0,counts))))
}

### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_factor))), ceiling(sqrt(length(df_factor)))),
    mar=c(1.5,1,2,1), oma= c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_factor)) plot_bar(i)

### Restore original graphical settings
par(opar)
```

```
# Categorical variables
## Descriptive graphics: Pie-Plots One Per Page

### Function to create frequency table for each variable
freqtable <- function(col){

  # Replace NA with "Missing"
  col[is.na(col)] <- "Missing"

  # Create table with frequencies
  pie_table_unsorted <- as.data.frame(table(col))
  pie_table_sorted <- pie_table_unsorted[order(pie_table_unsorted$Freq, decreasing=TRUE),]
  colnames(pie_table_sorted) <- c("Category", "Frequency")

  # If more than 20 categories: summarize the smallest categories to one category
  if (nrow(pie_table_sorted)>20){
    pie_table_sorted$Category <- as.character(pie_table_sorted$Category)
    pie_table_summarized <- rbind(pie_table_sorted[c(1:20),],
                                  c(as.character("All Other Values"),
                                    sum(pie_table_sorted$Frequency[-c(1:20)])))
    pie_table_sorted <- pie_table_summarized
  }
  pie_table_sorted$RelFreq <- as.numeric(pie_table_sorted$Frequency) / length(col)
  return(pie_table_sorted)
}

### Plot function
plot_pie <- function(table, title, title_size, legend_pos){
  # Direction of the legend
  if (max(nchar(as.character(table[,1])))>15){
```

```
    legend = "vertical"
  } else {
    legend = "horizontal"
  }
  plot <-
    ggplot(table, aes(x = "", y = RelFreq,
                                fill = reorder(Category, RelFreq))) +
    guides(fill = guide_legend(title="", reverse = FALSE, direction = legend)) +
    ggtitle(title) +
    geom_col() +
    geom_text(aes(label = scales::percent(RelFreq,accuracy = 0.01)),
              position = position_stack(vjust = 0.5), size = 2) +
    coord_polar("y", start = 0) +
    theme(axis.title.x = element_blank(),axis.title.y = element_blank(),
          axis.ticks = element_blank(), panel.grid = element_blank(),
          axis.text = element_blank(), legend.position = legend_pos,
          panel.background = element_blank(), plot.title = title_size)
  return(plot)
}

### Loop over variables
for(i in 1:ncol(df_factor)){
  table <- freqtable(df_factor_order[,i])
  title <- paste("Pie Chart of ", colnames(df_factor_order[i]))
  title_size <- element_text(hjust = 0.5, face = "bold")
  legend_pos <- "right"
  print(plot_pie(table=table, title=title, title_size=title_size, legend_pos=legend_pos))
}
```

```
# Categorical variables
## Descriptive graphics: Pie-Plots Summary

### Save variable plots in a list
plots <- list()
for (i in 1:ncol(df_factor)){
  title <- substr(colnames(df_factor_order[i]), 1, 19)
  title_size <- element_text(size = 30 / min(25, ceiling(sqrt(ncol(df_factor)))),
                             face = "bold", hjust = 0.5)
  legend_pos <- "none"
  table <- freqtable(df_factor_order[,i])
  plots[[i]] <- plot_pie(table=table, title=title, title_size=title_size, legend_pos=legend_pos)
}
```

```
### Summary Plot
grid.arrange(grobs = plots, ncol = ceiling(sqrt(ncol(df_factor_order))))
```

R Code License