Principal Components I

Mike Strube

September 12, 2018

1 Preliminaries

In this section, the RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Packages are loaded and any required data files are retrieved.

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

1.1 Packages

```
library(psych)
library(ggplot2)

##
## Attaching package:  'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha

library(factoextra)

## Warning:  package 'factoextra' was built under R version 3.5.1
## Welcome!  Related Books:  `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library(FactoMineR)

## Warning:  package 'FactoMineR' was built under R version 3.5.1
```

```r
library(reshape2)
library(GGally)
library(MASS)
library(parallel)
library(MVN)

## sROC 0.1-2 loaded

library(qqplotr)
```

2 Outlier Detection

Principal components analysis can be used to screen the data for outliers, especially cases that may not be univariate outliers but are unusual in the multivariate sense.

2.1 Data Without an Outlier

To provide a basis for comparison, we will start with a simulated data set containing no outliers, 250 cases, and 9 variables. The correlations among the variables designed to represent three underlying principal components.

$$R = \begin{bmatrix} 1.0 & 0.7 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.7 & 1.0 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.7 & 0.7 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.7 & 0.7 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7 & 1.0 & 0.7 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7 & 0.7 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.7 & 0.7 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7 & 1.0 & 0.7 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7 & 0.7 & 1.0 \end{bmatrix}$$

2.1.1 Data Generation

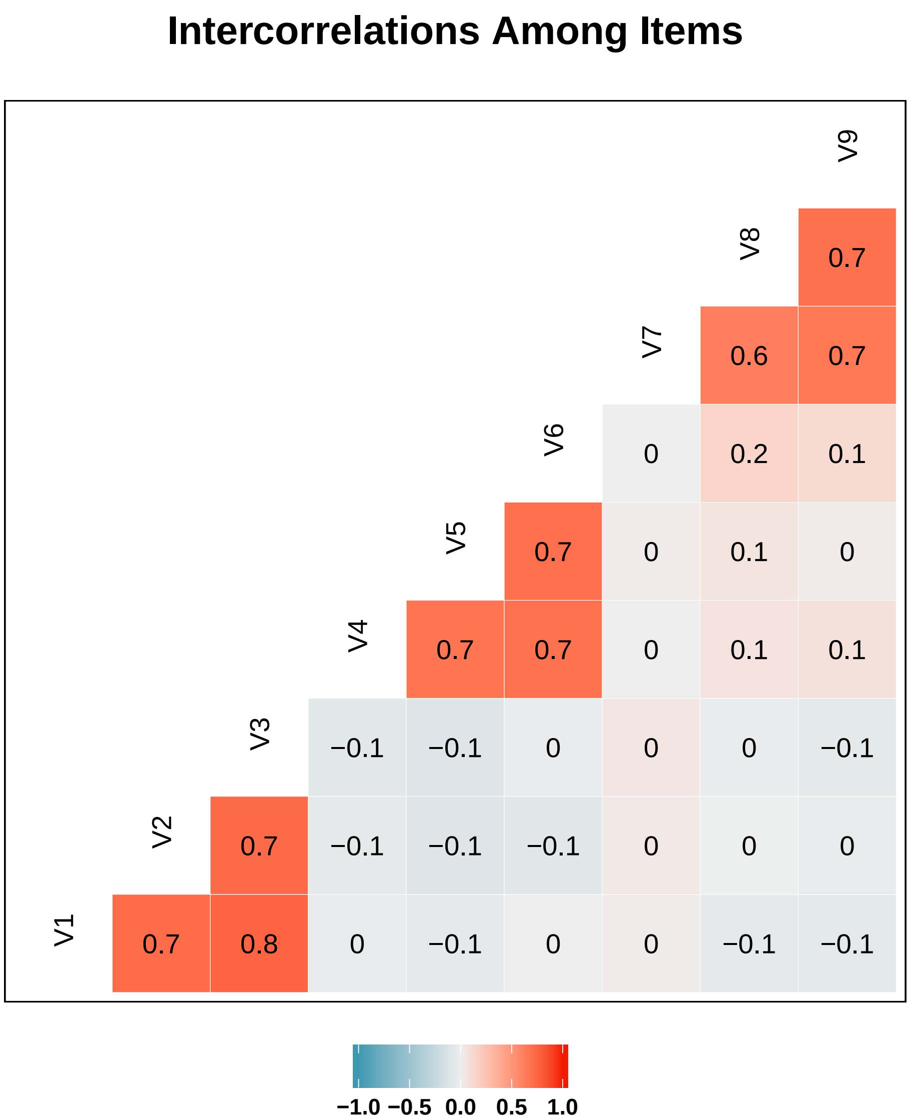
```r
means <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0))
sigma <- matrix(c(1, 0.7, 0.7, 0, 0, 0, 0, 0, 0, 0.7, 1, 0.7, 0, 0,
    0, 0, 0, 0, 0.7, 0.7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0.7, 0.7,
    0, 0, 0, 0, 0, 0, 0.7, 1, 0.7, 0, 0, 0, 0, 0, 0, 0.7, 0.7, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0.7, 0.7, 0, 0, 0, 0, 0, 0, 0.7,
    1, 0.7, 0, 0, 0, 0, 0, 0, 0.7, 0.7, 1), nrow = 9, ncol = 9)
Data <- mvrnorm(250, means, sigma)
Data <- as.data.frame(Data)
cor(Data)

##           V1        V2       V3        V4       V5        V6
## V1  1.000000  0.720410  0.75748 -0.039864 -0.05661 -0.006772
## V2  0.720410  1.000000  0.72777 -0.057630 -0.10176 -0.085436
## V3  0.757480  0.727774  1.00000 -0.073815 -0.10217 -0.033597
## V4 -0.039864 -0.057630 -0.07382  1.000000  0.67577  0.689880
```

```
## V5 -0.056611 -0.101759 -0.10217  0.675775  1.00000  0.701338
## V6 -0.006772 -0.085436 -0.03360  0.689880  0.70134  1.000000
## V7  0.018442  0.038930  0.04498 -0.006994  0.01802 -0.001917
## V8 -0.053959 -0.003951 -0.03068  0.068910  0.05636  0.153495
## V9 -0.068588 -0.037207 -0.06150  0.076454  0.01810  0.117135
##           V7        V8       V9
## V1  0.018442 -0.053959 -0.06859
## V2  0.038930 -0.003951 -0.03721
## V3  0.044980 -0.030676 -0.06150
## V4 -0.006994  0.068910  0.07645
## V5  0.018023  0.056360  0.01810
## V6 -0.001917  0.153495  0.11713
## V7  1.000000  0.626210  0.65968
## V8  0.626210  1.000000  0.69562
## V9  0.659678  0.695623  1.00000

Data_Original <- Data
```

2.1.2 Correlations

A heat map for the correlation matrix easily identifies the pattern of correlations in the simulated data.

```
ggcorr(Data, label = TRUE, angle = 90, hjust = 0.1, size = 4, digits = 2) +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Intercorrelations Among Items")
```



2.1.3 Should A PCA Be Conducted?

Two tests can be used to determine if a PCA should be conducted (generally a good idea if the approach is exploratory). The Kaiser-Meyer-Olkin (KMO) factor adequacy test can range from 0 to 1 and roughly indicates the proportion of variance in the data that might be common factor variance. The KMO test has the following cut-offs for sampling adequacy: .90 and above (undeniable evidence for factorability), .80 to .89 (very strong evidence), .70 to .79 (modest evidence), .60 to .69 (weak evidence), .50 to .59 (very weak evidence), and below .50 (unacceptable for factoring). The Bartlett test for sphericity (not the same as in repeated measures ANOVA) should be highly significant, indicating that the correlation matrix departs noticeably from an identity matrix.

```
R <- cor(Data)
KMO(R)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.72
## MSA for each item =
##   V1   V2   V3   V4   V5   V6   V7   V8   V9
## 0.74 0.76 0.73 0.75 0.71 0.69 0.71 0.72 0.69
```

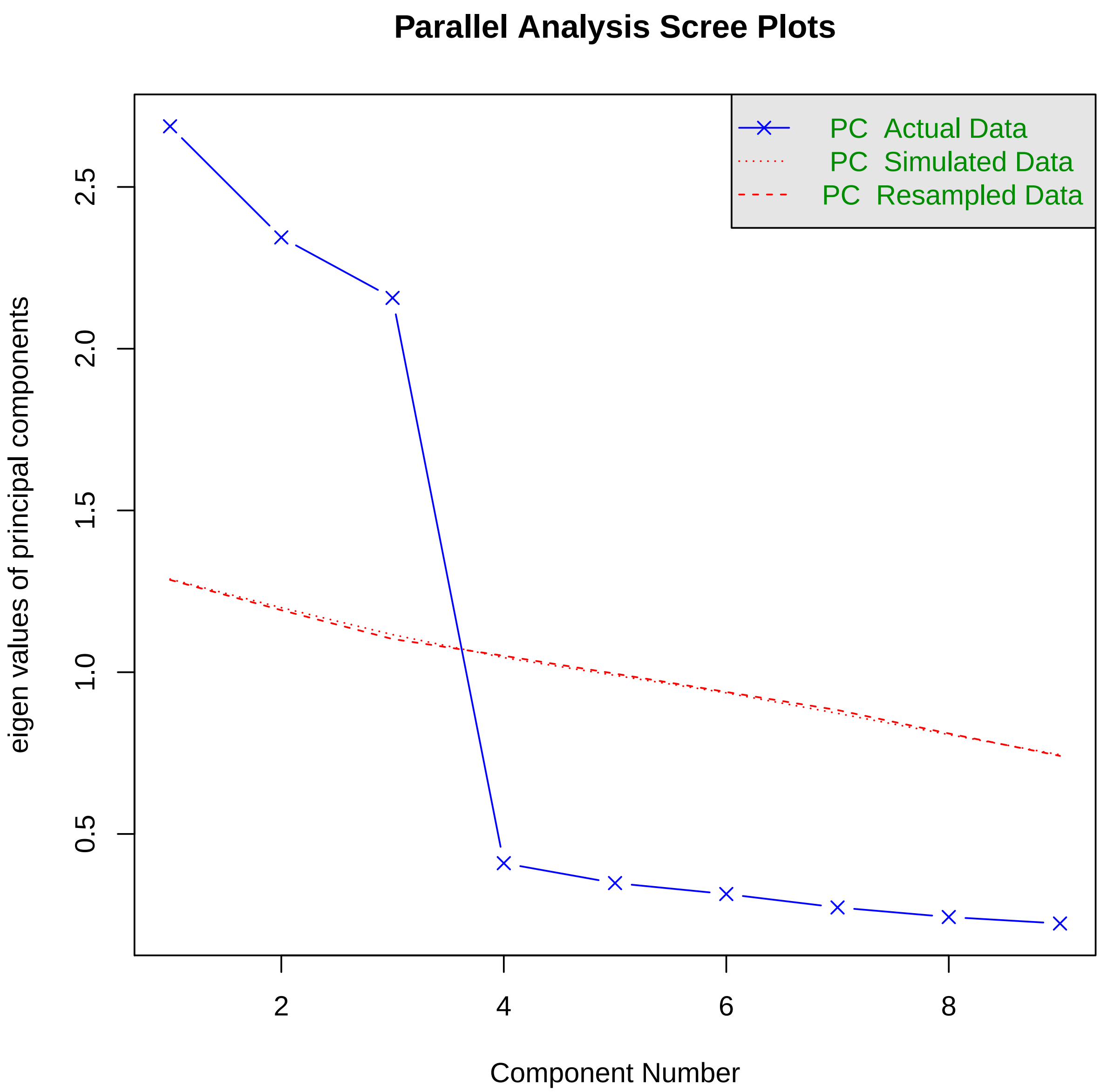
```
cortest.bartlett(R = R, n = length(Data[, 1]))

## $chisq
## [1] 1154
##
## $p.value
## [1] 6.233e-219
##
## $df
## [1] 36
```

These tests verify that the correlation matrix is more than an identity matrix, justifying a principal components analysis.

2.1.4 How Many Components?

If the correlation matrix is not singular, then as many components as there are variables or items can be extracted. But, only a few of them are likely to be meaningful or useful. The scree test is the most common way to determine how many components should be extracted. To make sure only meaningful departures from the scree are interpreted, a parallel analysis (Horn's procedure) or random selection of data points can be used.

```
scree <- fa.parallel(Data, fa = "pc")
```




```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  3
```

The scree test verifies the underlying three components designed into the data generation.

2.1.5 PCA

The following PCA is restricted to the three components that we believe underlie the data.

```
PCA_1 <- principal(Data, nfactors = 3, rotate = "none", residuals = TRUE)
PCA_1

## Principal Components Analysis
```

```
## Call: principal(r = Data, nfactors = 3, residuals = TRUE, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2   PC3   h2   u2 com
## V1 -0.62 0.58  0.32 0.83 0.17 2.5
## V2 -0.63 0.59  0.24 0.81 0.19 2.3
## V3 -0.64 0.59  0.27 0.84 0.16 2.4
## V4  0.59 0.24  0.61 0.78 0.22 2.3
## V5  0.61 0.21  0.62 0.79 0.21 2.2
## V6  0.60 0.30  0.59 0.81 0.19 2.5
## V7  0.28 0.62 -0.54 0.75 0.25 2.4
## V8  0.41 0.61 -0.49 0.78 0.22 2.7
## V9  0.41 0.59 -0.53 0.80 0.20 2.8
##
##                        PC1  PC2  PC3
## SS loadings           2.69 2.34 2.16
## Proportion Var        0.30 0.26 0.24
## Cumulative Var        0.30 0.56 0.80
## Proportion Explained  0.37 0.33 0.30
## Cumulative Proportion 0.37 0.70 1.00
##
## Mean item complexity =  2.4
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.05
##  with the empirical chi square  50.31  with prob <  0.0000012
##
## Fit based upon off diagonal values = 0.98
```

2.1.6 Examination of Residuals

A residual matrix gives the variances in the main diagonal and correlations in the off-diagonals. This can be converted to a correlation matrix, which can then be examined using the KMO and Bartlett tests to determine if additional components should be extracted.

```
# Create a correlation matrix of the residuals by replacing the
# main diagonal with ones.
R1 <- diag(PCA_1$residual)
R2 <- diag(R1)
R3 <- PCA_1$residual - R2
R4 <- diag(9) + R3

# Assess the factorability of the residual correlation matrix.
KMO(R4)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R4)
## Overall MSA =  0.46
## MSA for each item =
##   V1   V2   V3   V4   V5   V6   V7   V8   V9
## 0.46 0.47 0.46 0.46 0.47 0.46 0.47 0.45 0.44

cortest.bartlett(R = R4, n = length(Data[, 1]))
```

```
## $chisq
## [1] 26.29
##
## $p.value
## [1] 0.8822
##
## $df
## [1] 36
```

Once the three components are extracted, the residual correlation matrix shows no evidence of remaining components.

2.2 Data Modification: Addition of an Outlier

Now to simulate an outlier, we replace the last case with a profile of scores that is unusual, although not unlikely on a variable-by-variable basis:

3,-3,3,-3,3,-3,3,-3,3

```
Data[250, ] <- c(3, -3, 3, -3, 3, -3, 3, -3, 3)
```

2.2.1 Descriptive Statistics

There is nothing in the following descriptive statistics that indicates any particular problem. The case with the odd profile does not have the most extreme scores for some of the variables.

```
describe(Data)

##    vars   n  mean   sd median trimmed  mad   min  max range
## V1    1 250 -0.05 1.01  -0.04   -0.05 1.04 -2.78 3.00  5.78
## V2    2 250 -0.06 1.01  -0.04   -0.05 1.14 -3.00 2.68  5.68
## V3    3 250  0.02 1.01   0.08    0.03 0.95 -2.96 3.00  5.96
## V4    4 250  0.09 1.03   0.19    0.12 0.96 -3.00 2.96  5.96
## V5    5 250  0.07 1.01   0.01    0.05 1.00 -2.86 3.00  5.86
## V6    6 250  0.03 1.03   0.10    0.08 0.94 -3.32 3.01  6.33
## V7    7 250  0.03 0.94  -0.12   -0.03 0.91 -2.20 3.00  5.20
## V8    8 250 -0.04 0.99  -0.09   -0.06 1.03 -3.00 3.75  6.75
## V9    9 250 -0.01 1.00  -0.06   -0.03 1.00 -2.28 3.06  5.35
##     skew kurtosis   se
## V1  0.06    -0.02 0.06
## V2 -0.07    -0.25 0.06
## V3 -0.05    -0.05 0.06
## V4 -0.31    -0.01 0.07
## V5  0.16     0.01 0.06
## V6 -0.39     0.43 0.07
## V7  0.53     0.13 0.06
## V8  0.24     0.37 0.06
## V9  0.23    -0.16 0.06
```

2.2.2 Normality Tests

The distribution of each variable can be tested for its departure from normal, using either the Kolmogorov-Smirnoff test or the Shapiro-Wilk test. The latter is usually preferred, especially for small samples. There is no evidence of a problem from these tests.

```r
ks.test(Data$V1, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V1
## D = 0.045, p-value = 0.7
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V2, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V2
## D = 0.055, p-value = 0.4
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V3, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V3
## D = 0.045, p-value = 0.7
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V4, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V4
## D = 0.098, p-value = 0.02
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V5, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V5
## D = 0.043, p-value = 0.7
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V6, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V6
## D = 0.056, p-value = 0.4
## alternative hypothesis: two-sided
```

```r
ks.test(Data$V7, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V7
## D = 0.06, p-value = 0.3
## alternative hypothesis: two-sided

ks.test(Data$V8, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V8
## D = 0.064, p-value = 0.3
## alternative hypothesis: two-sided

ks.test(Data$V9, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data$V9
## D = 0.039, p-value = 0.8
## alternative hypothesis: two-sided
```

```r
shapiro.test(Data$V1)

##
##  Shapiro-Wilk normality test
##
## data:  Data$V1
## W = 1, p-value = 0.8

shapiro.test(Data$V2)

##
##  Shapiro-Wilk normality test
##
## data:  Data$V2
## W = 0.99, p-value = 0.6

shapiro.test(Data$V3)

##
##  Shapiro-Wilk normality test
##
## data:  Data$V3
## W = 1, p-value = 1

shapiro.test(Data$V4)

##
##  Shapiro-Wilk normality test
```

```
##
## data:  Data$V4
## W = 0.99, p-value = 0.08
```

```r
shapiro.test(Data$V5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data$V5
## W = 1, p-value = 0.6
```

```r
shapiro.test(Data$V6)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data$V6
## W = 0.99, p-value = 0.02
```

```r
shapiro.test(Data$V7)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data$V7
## W = 0.98, p-value = 0.001
```

```r
shapiro.test(Data$V8)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data$V8
## W = 0.99, p-value = 0.2
```

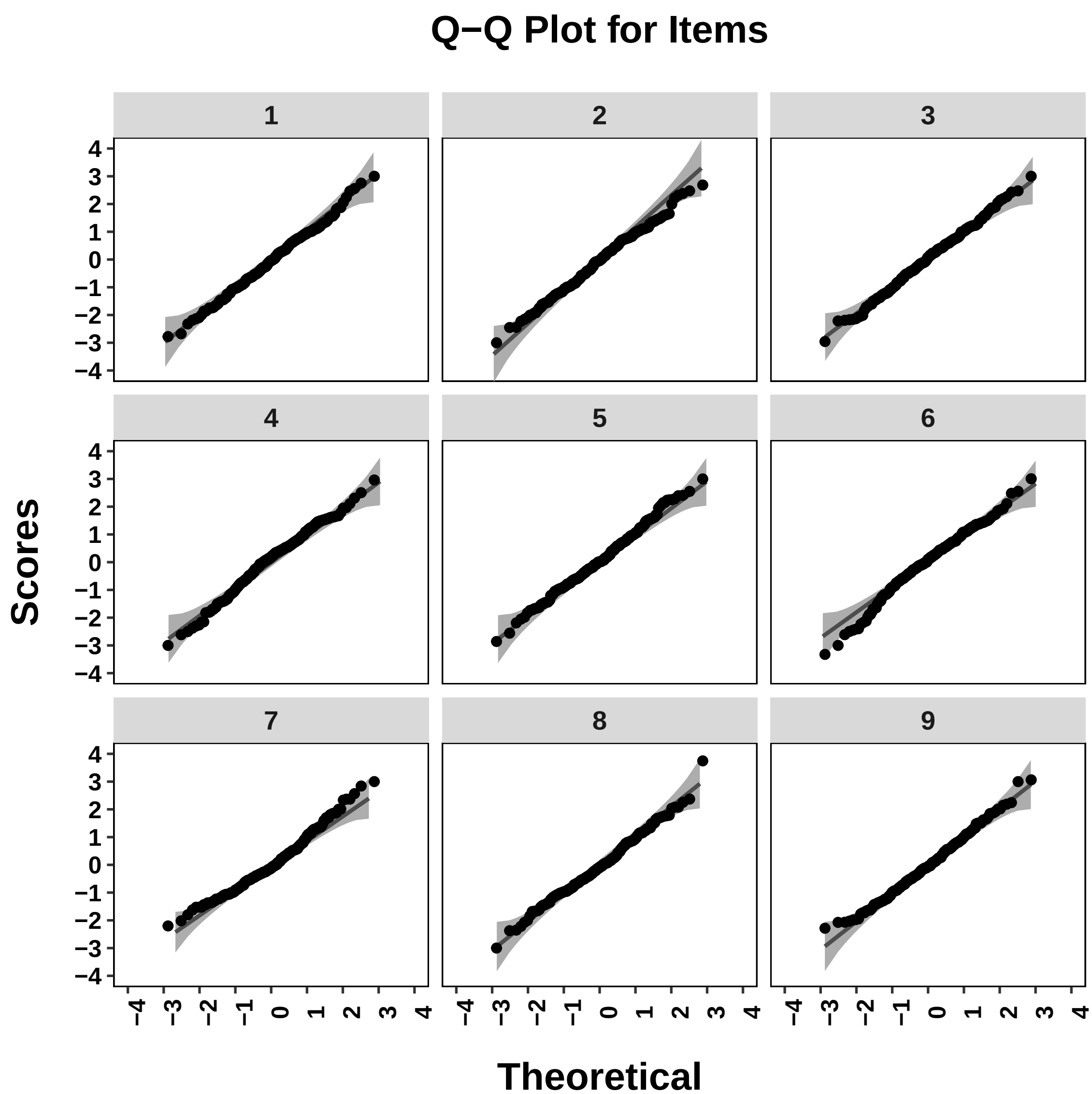
```r
shapiro.test(Data$V9)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data$V9
## W = 0.99, p-value = 0.4
```

We can also examine the QQ-plots. The following all verify overall normality. The outlier is not evident in the displays.

```r
Data_long <- melt(Data)

## No id variables; using all as measure variables

Data_long <- as.data.frame(Data_long)
Data_long$item <- factor(Data_long$variable, levels = c("V1", "V2",
    "V3", "V4", "V5", "V6", "V7", "V8", "V9"), labels = c("1", "2",
    "3", "4", "5", "6", "7", "8", "9"))
p <- ggplot(Data_long, aes(sample = value)) + stat_qq_band() + stat_qq_line() +
```

```
    stat_qq(distribution = qnorm) + scale_y_continuous(breaks = c(-4,
    -3, -2, -1, 0, 1, 2, 3, 4)) + scale_x_continuous(breaks = c(-4,
    -3, -2, -1, 0, 1, 2, 3, 4)) + coord_cartesian(xlim = c(-4, 4),
    ylim = c(-4, 4)) + xlab("Theoretical") + ylab("Scores") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 10, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 10, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Q-Q Plot for Items")
p + facet_wrap(~item)
```

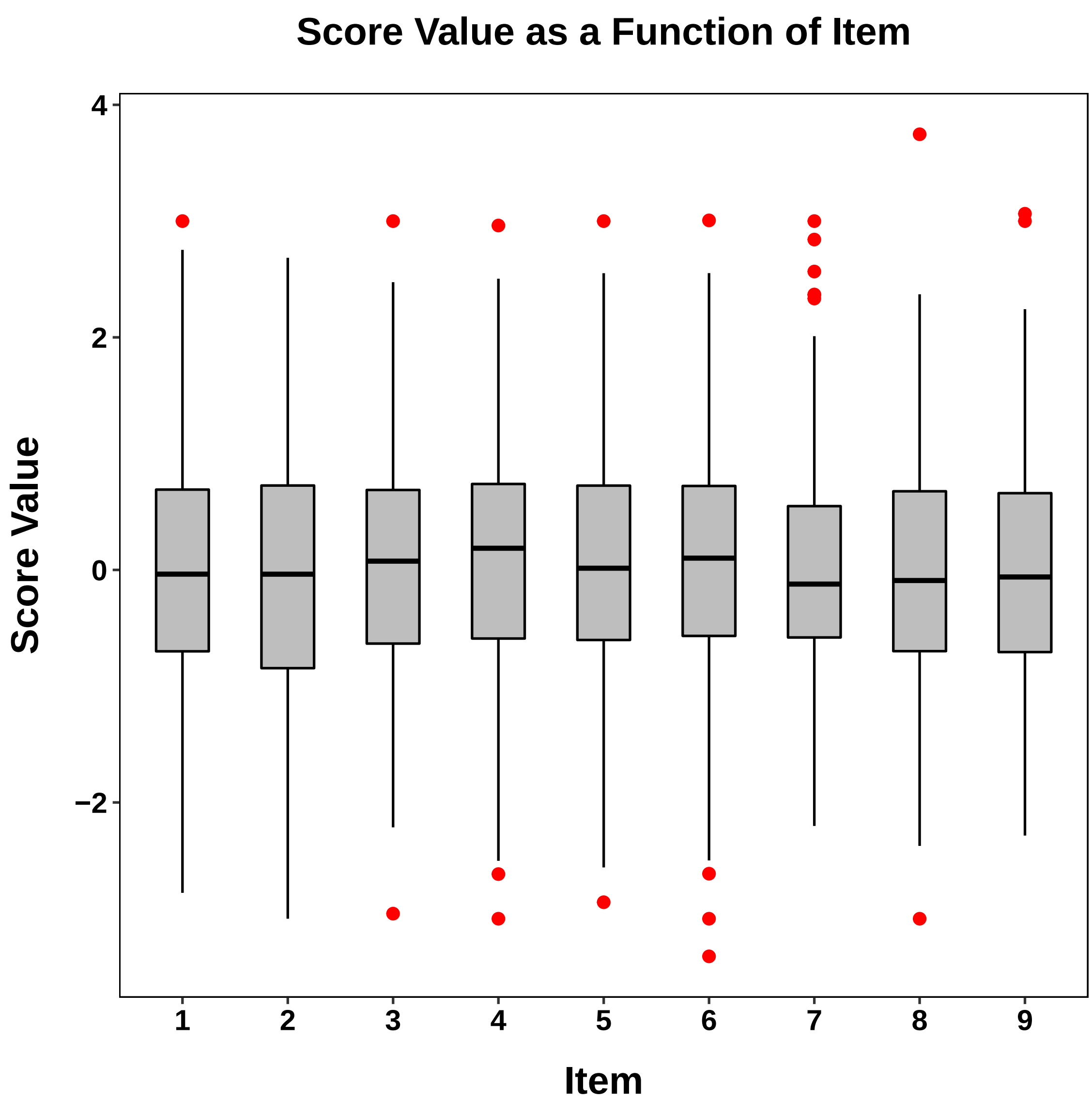


2.2.3 Boxplots

Boxplots are useful when searching for outliers, although the problematic case is not evidient here.

```
ggplot(Data_long, aes(y = value, x = item)) + geom_boxplot(aes(y = value,
    x = item), color = "black", size = 0.5, width = 0.5, fill = "grey",
    outlier.colour = "red", outlier.shape = 19, outlier.size = 2,
    notch = FALSE) + ylab("Score Value") + xlab("Item") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
```

```
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm")) + ggtitle("Score Value as a Function of Item")
```



2.2.4 PCA As Outlier Detector

A principal components analysis will seek linear combinations that capture the major sources of variance in the data. Most of these will be governed by the "well-behaved" data. But, once those data are captured, especially deviant multivariate cases may dominant the smaller components and emerge more readily. In this approach, all components are derived and component scores are produced. Then diagnostics are performed on the component scores.

```r
PCA_2 <- principal(Data, nfactors = 9, rotate = "none", residuals = TRUE,
    scores = TRUE)
PCA_2
```

```
## Principal Components Analysis
## Call: principal(r = Data, nfactors = 9, residuals = TRUE, rotate = "none",
##     scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9 h2
## V1 -0.64 0.53  0.36  0.16 -0.14 -0.06 -0.01  0.21 -0.29  1
## V2 -0.62 0.50  0.32 -0.33  0.25  0.07  0.22 -0.18 -0.03  1
## V3 -0.67 0.54  0.31  0.13 -0.12 -0.02 -0.17 -0.02  0.33  1
## V4  0.60 0.21  0.62 -0.16  0.29 -0.26 -0.12  0.17  0.04  1
## V5  0.58 0.22  0.58  0.39  0.02  0.24  0.25  0.06  0.08  1
## V6  0.61 0.27  0.60 -0.15 -0.27  0.02 -0.14 -0.26 -0.11  1
## V7  0.21 0.68 -0.51  0.31  0.26  0.11 -0.22 -0.12 -0.09  1
## V8  0.37 0.63 -0.42 -0.41 -0.12  0.25 -0.01  0.21  0.05  1
## V9  0.34 0.65 -0.50  0.08 -0.13 -0.35  0.23 -0.06  0.04  1
##          u2 com
## V1  1.1e-16 3.6
## V2 -8.9e-16 4.1
## V3  0.0e+00 3.3
## V4  2.2e-15 3.5
## V5  1.6e-15 3.9
## V6  2.2e-15 3.5
## V7 -1.6e-15 3.4
## V8 -1.3e-15 4.1
## V9 -1.6e-15 3.6
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8
## SS loadings           2.61 2.27 2.09 0.61 0.35 0.33 0.27 0.24
## Proportion Var        0.29 0.25 0.23 0.07 0.04 0.04 0.03 0.03
## Cumulative Var        0.29 0.54 0.77 0.84 0.88 0.92 0.95 0.97
## Proportion Explained  0.29 0.25 0.23 0.07 0.04 0.04 0.03 0.03
## Cumulative Proportion 0.29 0.54 0.77 0.84 0.88 0.92 0.95 0.97
##                        PC9
## SS loadings           0.23
## Proportion Var        0.03
## Cumulative Var        1.00
## Proportion Explained  0.03
## Cumulative Proportion 1.00
##
## Mean item complexity =  3.7
## Test of the hypothesis that 9 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

2.2.4.1 Extract All Principal Components

```r
Data_PC <- as.data.frame(PCA_2$scores)
```

2.2.4.2 Repeat Diagnostics on PC Scores

2.2.4.3 Descriptive Statistics

The descriptives now indicate a problem with the fourth principal component.

```r
describe(Data_PC)

##     vars   n mean sd median trimmed  mad   min   max range  skew
## PC1    1 250    0  1  -0.06   -0.02 0.96 -2.54  2.54  5.08  0.17
## PC2    2 250    0  1  -0.05   -0.02 0.91 -2.42  3.13  5.55  0.21
## PC3    3 250    0  1   0.06    0.02 1.11 -3.18  2.05  5.23 -0.25
## PC4    4 250    0  1  -0.01   -0.05 0.73 -2.17 10.26 12.43  4.34
## PC5    5 250    0  1   0.07    0.00 0.95 -3.25  2.99  6.24 -0.03
## PC6    6 250    0  1   0.01    0.03 1.01 -3.09  2.21  5.30 -0.27
## PC7    7 250    0  1   0.02    0.02 0.96 -2.38  2.15  4.53 -0.19
## PC8    8 250    0  1  -0.04    0.02 0.96 -3.18  2.73  5.92 -0.17
## PC9    9 250    0  1   0.09   -0.01 0.92 -2.50  3.10  5.61  0.21
##     kurtosis   se
## PC1    -0.40 0.06
## PC2     0.30 0.06
## PC3    -0.42 0.06
## PC4    42.39 0.06
## PC5     0.10 0.06
## PC6    -0.26 0.06
## PC7    -0.44 0.06
## PC8     0.15 0.06
## PC9     0.34 0.06
```

2.2.4.4 Normality Tests

The distribution of each variable can be tested for its departure from normal, using either the Kolmogorov-Smirnoff test or the Shapiro-Wilk test. The latter is usually preferred, especially for small samples. The fourth principal component is now quite clearly not normally distributed.

```r
ks.test(Data_PC$PC1, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC1
## D = 0.054, p-value = 0.5
## alternative hypothesis: two-sided

ks.test(Data_PC$PC2, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC2
## D = 0.048, p-value = 0.6
## alternative hypothesis: two-sided
```

```r
ks.test(Data_PC$PC3, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC3
## D = 0.043, p-value = 0.7
## alternative hypothesis: two-sided

ks.test(Data_PC$PC4, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC4
## D = 0.094, p-value = 0.02
## alternative hypothesis: two-sided

ks.test(Data_PC$PC5, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC5
## D = 0.034, p-value = 0.9
## alternative hypothesis: two-sided

ks.test(Data_PC$PC6, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC6
## D = 0.047, p-value = 0.6
## alternative hypothesis: two-sided

ks.test(Data_PC$PC7, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC7
## D = 0.046, p-value = 0.7
## alternative hypothesis: two-sided

ks.test(Data_PC$PC8, "pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC8
## D = 0.035, p-value = 0.9
## alternative hypothesis: two-sided

ks.test(Data_PC$PC9, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Data_PC$PC9
## D = 0.043, p-value = 0.7
## alternative hypothesis: two-sided
```

```r
shapiro.test(Data_PC$PC1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC1
## W = 0.99, p-value = 0.2
```

```r
shapiro.test(Data_PC$PC2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC2
## W = 0.99, p-value = 0.2
```

```r
shapiro.test(Data_PC$PC3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC3
## W = 0.99, p-value = 0.04
```

```r
shapiro.test(Data_PC$PC4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC4
## W = 0.75, p-value <2e-16
```

```r
shapiro.test(Data_PC$PC5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC5
## W = 1, p-value = 1
```

```r
shapiro.test(Data_PC$PC6)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC6
## W = 0.99, p-value = 0.1
```

```r
shapiro.test(Data_PC$PC7)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC7
## W = 0.99, p-value = 0.05
```

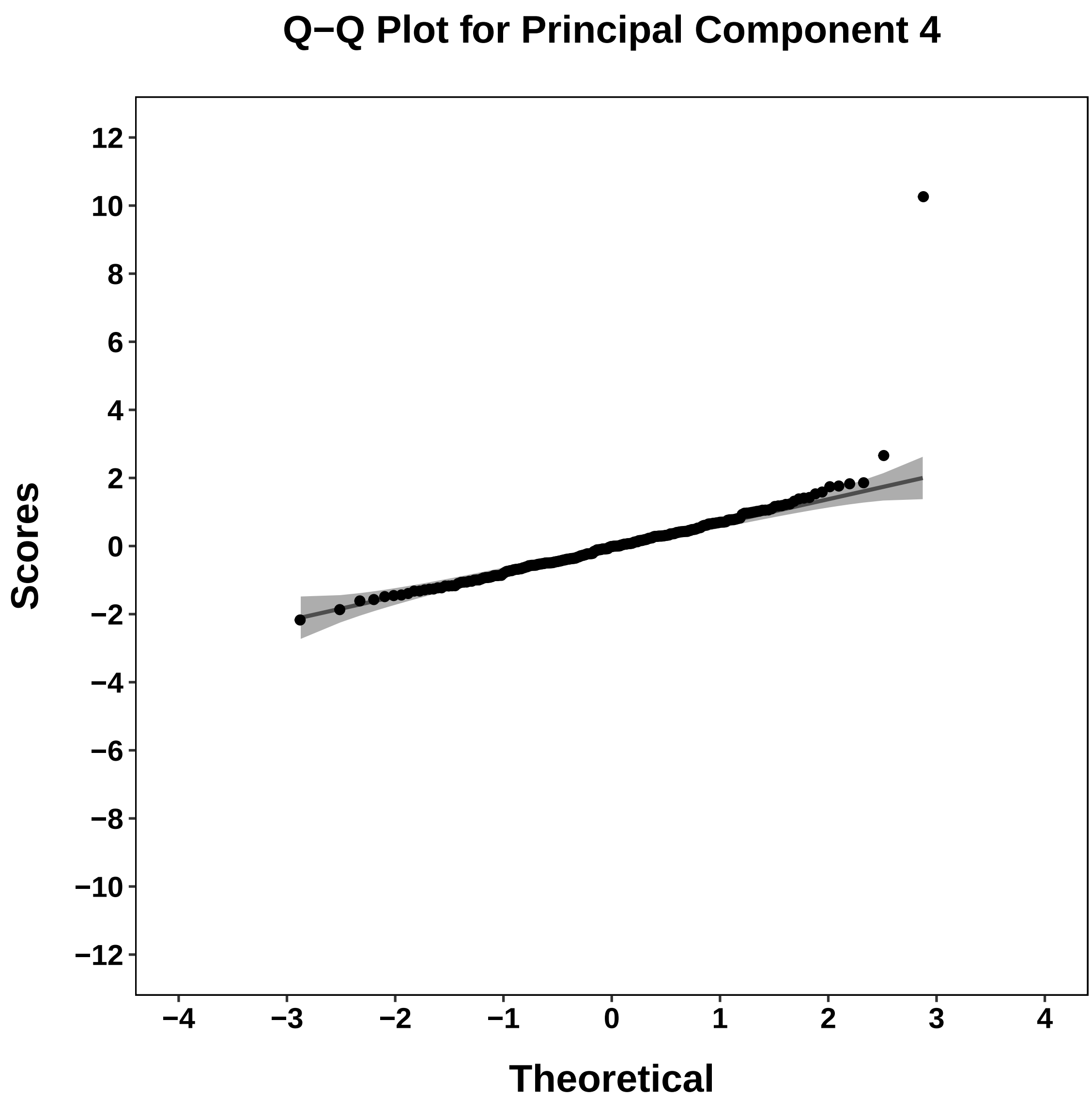
```
shapiro.test(Data_PC$PC8)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC8
## W = 1, p-value = 0.7
```

```
shapiro.test(Data_PC$PC9)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Data_PC$PC9
## W = 0.99, p-value = 0.05
```

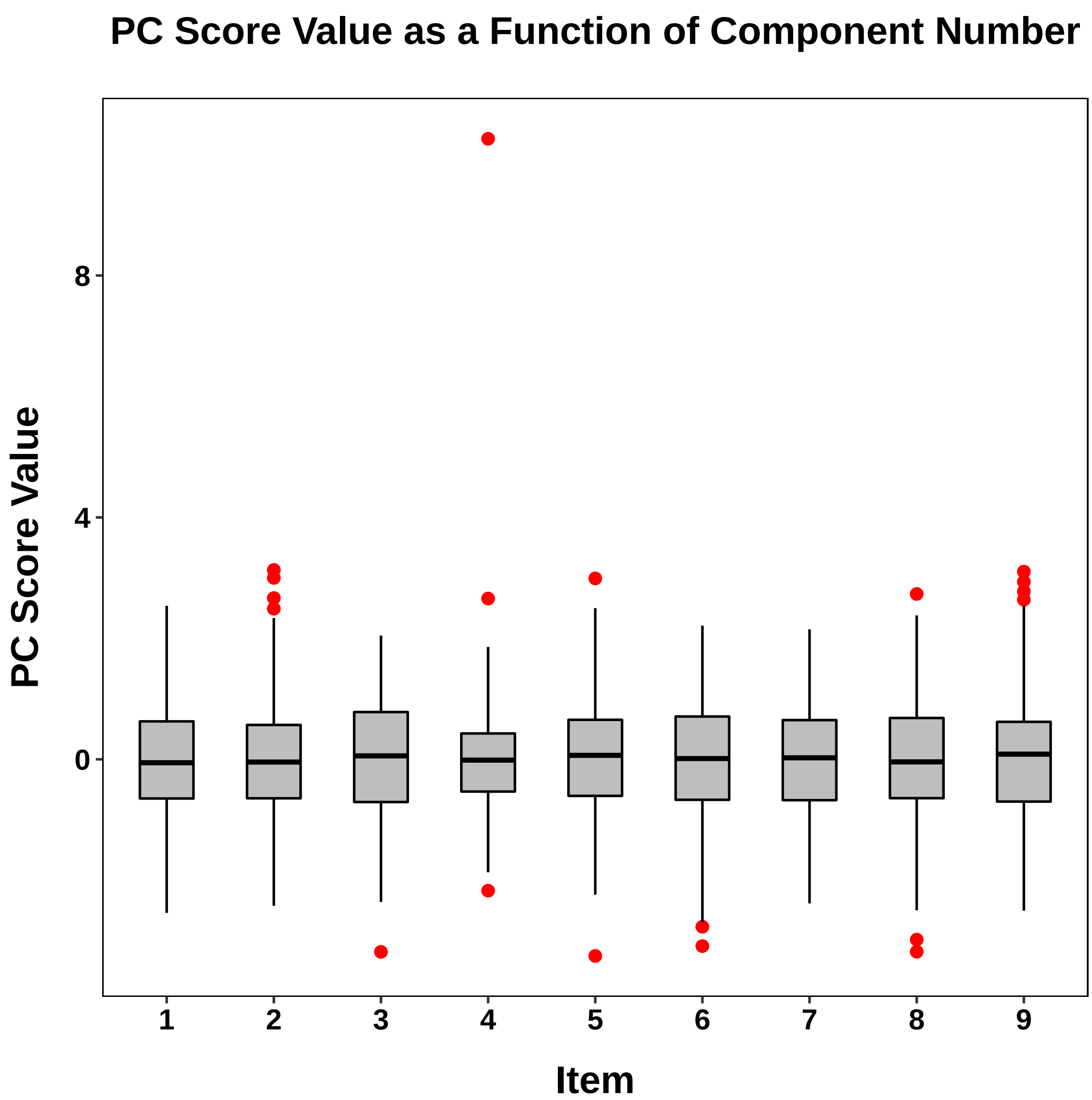
```
ggplot(Data_PC, aes(sample = PC4)) + stat_qq_band() + stat_qq_line() +
    stat_qq(distribution = qnorm) + scale_y_continuous(breaks = seq(-12,
    12, 2)) + scale_x_continuous(breaks = seq(-4, 4, 1)) + coord_cartesian(xlim = c(-4,
    4), ylim = c(-12, 12)) + xlab("Theoretical") + ylab("Scores") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Q-Q Plot for Principal Component 4")
```



2.2.4.5 Boxplots *The unusual case is not quite clearly identified in boxplots of the principal component scores.*

```
Data_PC_long <- melt(Data_PC)

## No id variables; using all as measure variables

Data_PC_long <- as.data.frame(Data_PC_long)

Data_PC_long$item <- factor(Data_PC_long$variable, levels = c("PC1",
    "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9"), labels = c("1",
    "2", "3", "4", "5", "6", "7", "8", "9"))
ggplot(Data_PC_long, aes(y = value, x = item)) + geom_boxplot(aes(y = value,
    x = item), color = "black", size = 0.5, width = 0.5, fill = "grey",
```

```
    outlier.colour = "red", outlier.shape = 19, outlier.size = 2,
    notch = FALSE) + ylab("PC Score Value") + xlab("Item") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm")) + ggtitle("PC Score Value as a Function of Component Number
```



3 Multivariate Normality

In univariate statistics, the normality assumption underlies significance testing. It is with reference to sampling from some theoretical distribution that we can make claims about the likelihood of results occurring "by chance" or "under the null hypothesis." Similarly, the establishment of confidence intervals depends on distributional assumptions.
Many multivariate procedures rely on maximum likelihood estimation. The normality assumption is important there as well. In maximum likelihood, the parameter estimates maximize the probability of the data, assuming a multivariate normal distribution.
Assessing multivariate normality is a bit tricky. When multivariate normality holds:

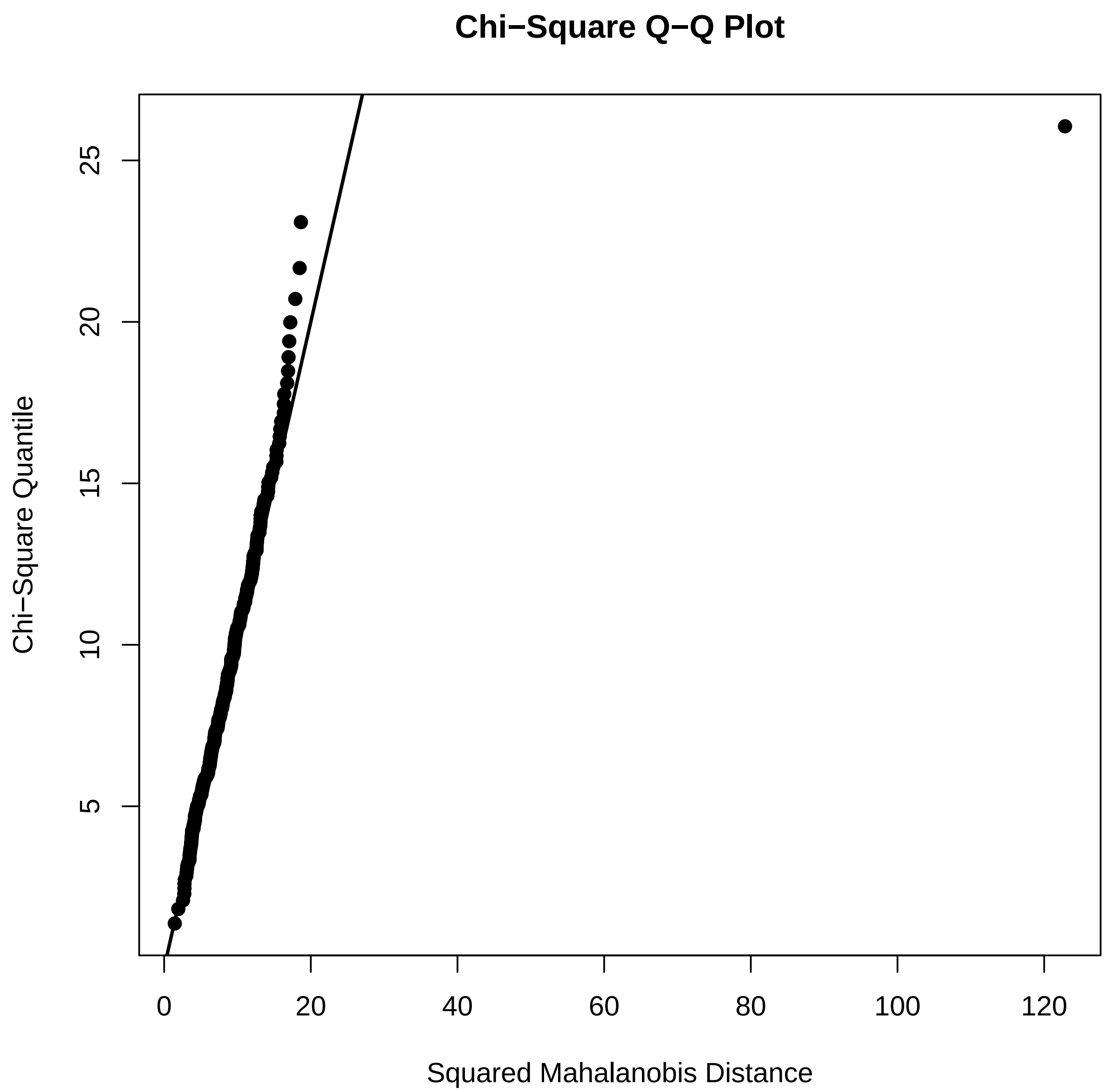
All marginal distributions will be normal.
All pairs of variables will be bivariate normal.
All linear combinations will be normal.
All pairs of linear combinations will be bivariate normal.
Squared distances from the population centroid will be χ^2 distributed with k (k = number of variables) degrees of freedom.

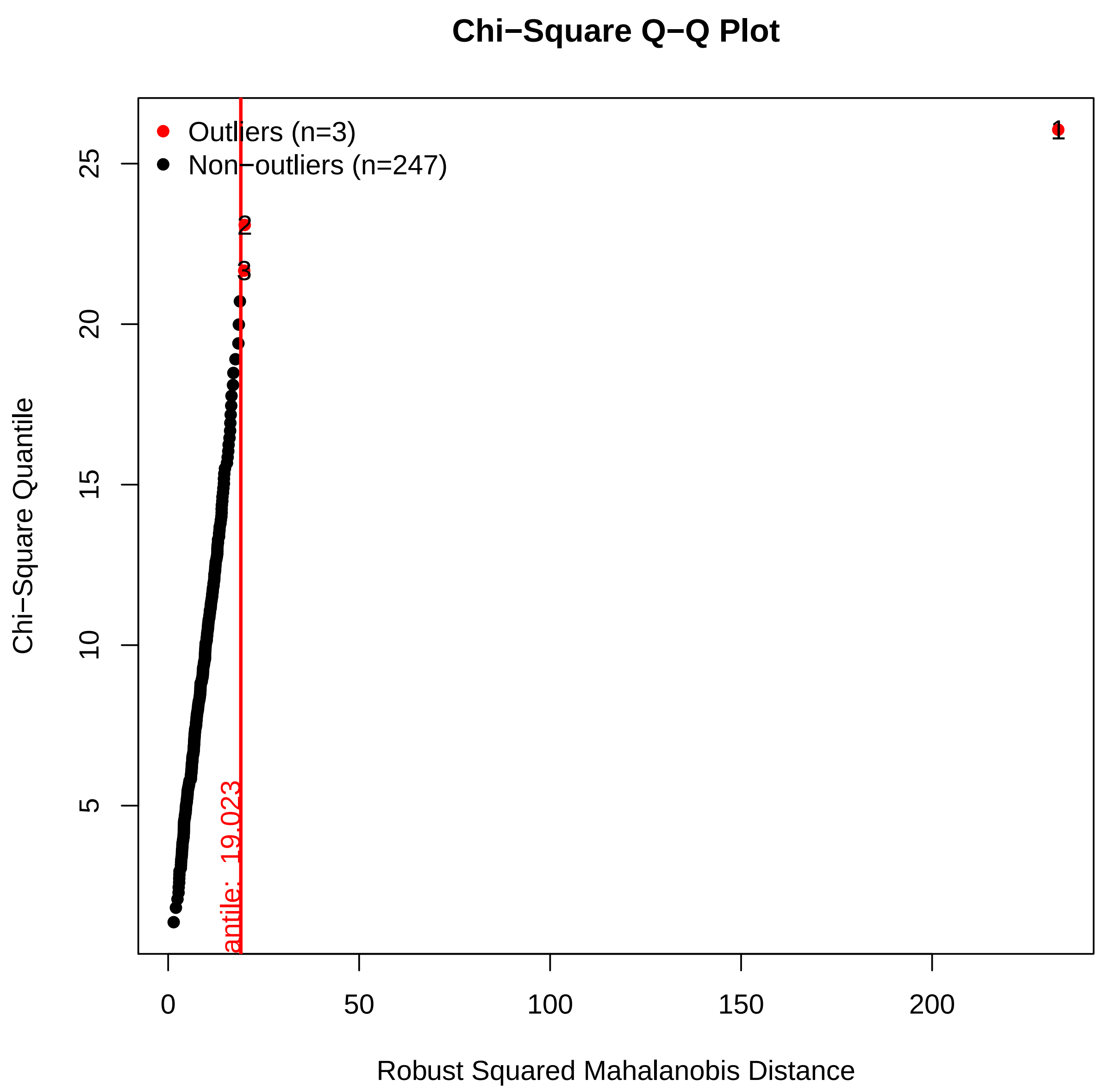
3.1 Mahalanobis Distance

Multivariate outliers are often revealed more easily using multivariate distance as assessed by Mahalanobis Distance. When multivariate normality holds, squared Mahalanobis distances will be χ^2 distributed with degrees of freedom equal to the number of measures.

3.1.1 Data With Outlier

```
mvn(Data, mvnTest = "mardia", multivariatePlot = "qq", multivariateOutlierMethod = "quan",
    showOutliers = TRUE)
```


Chi−Square Q−Q Plot
Chi−Square Quantile
5
10
15
20
25
0
20
40
60
80
100
120
Squared Mahalanobis Distance




```
## $multivariateNormality
##              Test        Statistic               p value Result
## 1 Mardia Skewness  1380.9251635341 7.25821195066503e-191     NO
## 2 Mardia Kurtosis 27.0433388155401                     0     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9961    0.7912    YES
## 2 Shapiro-Wilk    V2        0.9949    0.5658    YES
## 3 Shapiro-Wilk    V3        0.9974    0.9561    YES
## 4 Shapiro-Wilk    V4        0.9898    0.0760    YES
## 5 Shapiro-Wilk    V5        0.9952    0.6340    YES
## 6 Shapiro-Wilk    V6        0.9872    0.0248    NO
```

```
## 7 Shapiro-Wilk    V7        0.9799    0.0013    NO
## 8 Shapiro-Wilk    V8        0.9926    0.2431    YES
## 9 Shapiro-Wilk    V9        0.9937    0.3847    YES
##
## $Descriptives
##      n      Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.050533  1.0122 -0.03637 -2.777 3.000 -0.7001 0.6906
## V2 250 -0.057716  1.0090 -0.03678 -3.000 2.684 -0.8450 0.7260
## V3 250  0.024191  1.0086  0.07531 -2.956 3.000 -0.6336 0.6876
## V4 250  0.087338  1.0279  0.18634 -3.000 2.962 -0.5902 0.7393
## V5 250  0.070994  1.0120  0.01472 -2.858 3.000 -0.6027 0.7252
## V6 250  0.032343  1.0344  0.10124 -3.324 3.006 -0.5677 0.7219
## V7 250  0.027059  0.9399 -0.12191 -2.202 3.000 -0.5809 0.5485
## V8 250 -0.036554  0.9868 -0.09124 -3.000 3.747 -0.6988 0.6763
## V9 250 -0.006686  1.0006 -0.06066 -2.285 3.063 -0.7067 0.6601
##        Skew  Kurtosis
## V1  0.05594 -0.017509
## V2 -0.06706 -0.245770
## V3 -0.04505 -0.052613
## V4 -0.31408 -0.008075
## V5  0.15570  0.009624
## V6 -0.38932  0.425754
## V7  0.52965  0.134834
## V8  0.24031  0.373272
## V9  0.22770 -0.158336
##
## $multivariateOutliers
##   Observation Mahalanobis Distance Outlier
## 1           1               233.03    TRUE
## 2           2                20.05    TRUE
## 3           3                19.91    TRUE
```

```r
mvn(Data, mvnTest = "royston")
```

```
## $multivariateNormality
##      Test     H  p value MVN
## 1 Royston 21.65 0.008263  NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9961    0.7912    YES
## 2 Shapiro-Wilk    V2        0.9949    0.5658    YES
## 3 Shapiro-Wilk    V3        0.9974    0.9561    YES
## 4 Shapiro-Wilk    V4        0.9898    0.0760    YES
## 5 Shapiro-Wilk    V5        0.9952    0.6340    YES
## 6 Shapiro-Wilk    V6        0.9872    0.0248    NO
## 7 Shapiro-Wilk    V7        0.9799    0.0013    NO
## 8 Shapiro-Wilk    V8        0.9926    0.2431    YES
## 9 Shapiro-Wilk    V9        0.9937    0.3847    YES
##
## $Descriptives
##      n      Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.050533  1.0122 -0.03637 -2.777 3.000 -0.7001 0.6906
## V2 250 -0.057716  1.0090 -0.03678 -3.000 2.684 -0.8450 0.7260
```

```
## V3 250  0.024191  1.0086  0.07531 -2.956 3.000 -0.6336 0.6876
## V4 250  0.087338  1.0279  0.18634 -3.000 2.962 -0.5902 0.7393
## V5 250  0.070994  1.0120  0.01472 -2.858 3.000 -0.6027 0.7252
## V6 250  0.032343  1.0344  0.10124 -3.324 3.006 -0.5677 0.7219
## V7 250  0.027059  0.9399 -0.12191 -2.202 3.000 -0.5809 0.5485
## V8 250 -0.036554  0.9868 -0.09124 -3.000 3.747 -0.6988 0.6763
## V9 250 -0.006686  1.0006 -0.06066 -2.285 3.063 -0.7067 0.6601
##        Skew  Kurtosis
## V1  0.05594 -0.017509
## V2 -0.06706 -0.245770
## V3 -0.04505 -0.052613
## V4 -0.31408 -0.008075
## V5  0.15570  0.009624
## V6 -0.38932  0.425754
## V7  0.52965  0.134834
## V8  0.24031  0.373272
## V9  0.22770 -0.158336
```

```
mvn(Data, mvnTest = "hz")
```

```
## $multivariateNormality
##            Test     HZ p value MVN
## 1 Henze-Zirkler 0.9789  0.4521 YES
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9961    0.7912    YES
## 2 Shapiro-Wilk    V2        0.9949    0.5658    YES
## 3 Shapiro-Wilk    V3        0.9974    0.9561    YES
## 4 Shapiro-Wilk    V4        0.9898    0.0760    YES
## 5 Shapiro-Wilk    V5        0.9952    0.6340    YES
## 6 Shapiro-Wilk    V6        0.9872    0.0248    NO
## 7 Shapiro-Wilk    V7        0.9799    0.0013    NO
## 8 Shapiro-Wilk    V8        0.9926    0.2431    YES
## 9 Shapiro-Wilk    V9        0.9937    0.3847    YES
##
## $Descriptives
##      n      Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.050533  1.0122 -0.03637 -2.777 3.000 -0.7001 0.6906
## V2 250 -0.057716  1.0090 -0.03678 -3.000 2.684 -0.8450 0.7260
## V3 250  0.024191  1.0086  0.07531 -2.956 3.000 -0.6336 0.6876
## V4 250  0.087338  1.0279  0.18634 -3.000 2.962 -0.5902 0.7393
## V5 250  0.070994  1.0120  0.01472 -2.858 3.000 -0.6027 0.7252
## V6 250  0.032343  1.0344  0.10124 -3.324 3.006 -0.5677 0.7219
## V7 250  0.027059  0.9399 -0.12191 -2.202 3.000 -0.5809 0.5485
## V8 250 -0.036554  0.9868 -0.09124 -3.000 3.747 -0.6988 0.6763
## V9 250 -0.006686  1.0006 -0.06066 -2.285 3.063 -0.7067 0.6601
##        Skew  Kurtosis
## V1  0.05594 -0.017509
## V2 -0.06706 -0.245770
## V3 -0.04505 -0.052613
## V4 -0.31408 -0.008075
## V5  0.15570  0.009624
## V6 -0.38932  0.425754
```

```
## V7  0.52965  0.134834
## V8  0.24031  0.373272
## V9  0.22770 -0.158336
```

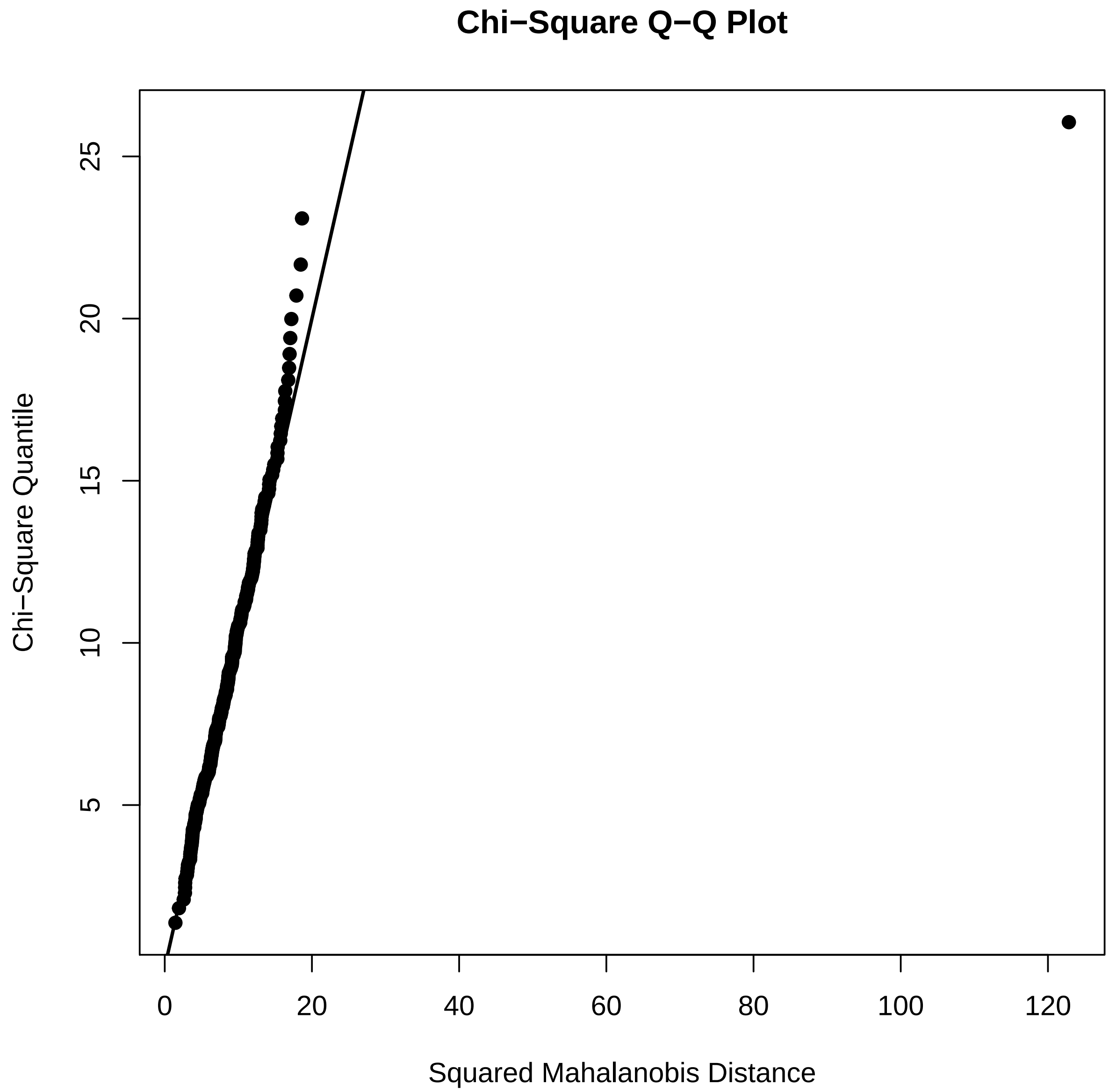
```r
mvn(Data, mvnTest = "dh")
```

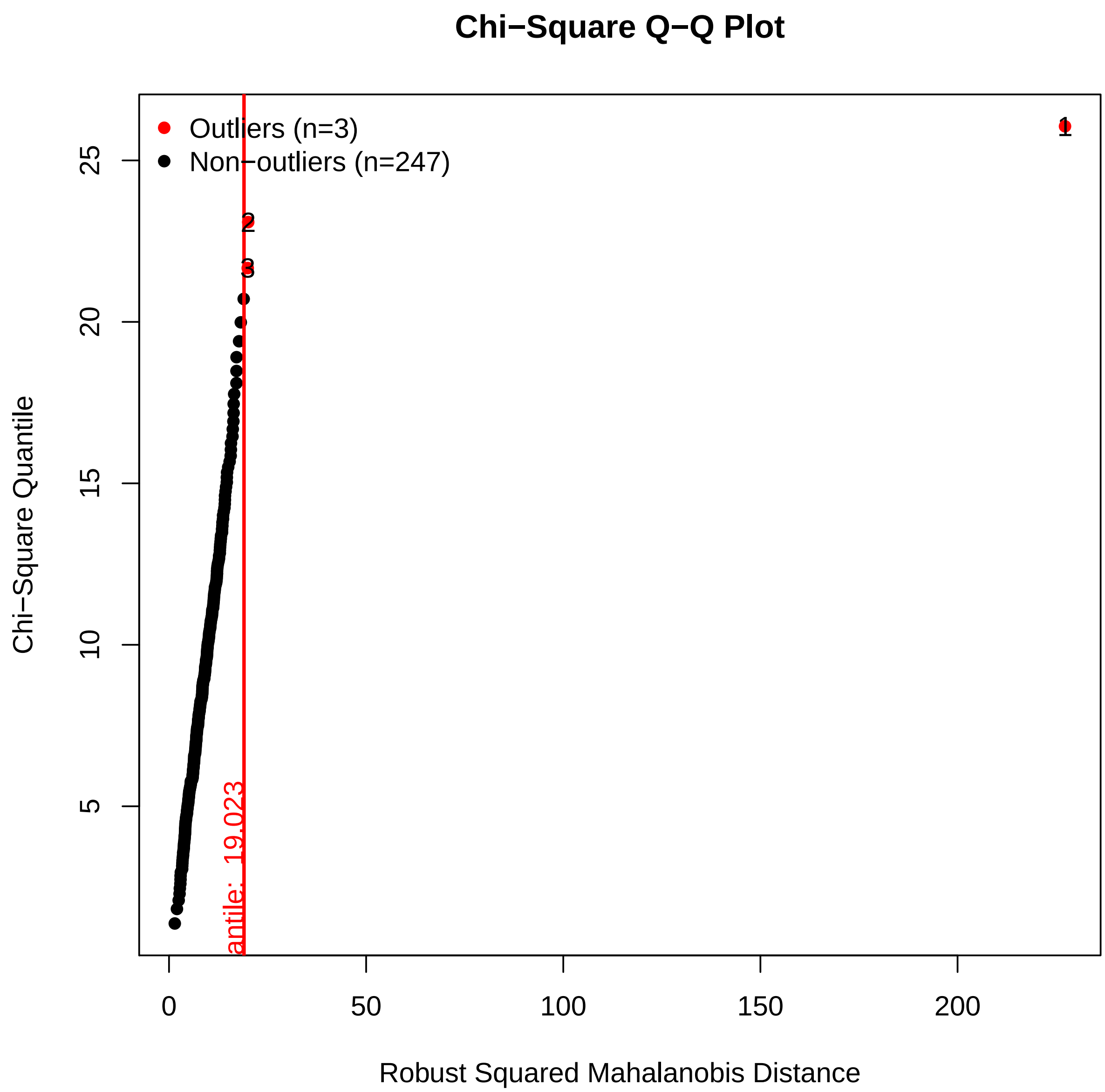
```
## $multivariateNormality
##             Test    E df     p value MVN
## 1 Doornik-Hansen 57.5 18 0.000005146  NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9961    0.7912    YES
## 2 Shapiro-Wilk    V2        0.9949    0.5658    YES
## 3 Shapiro-Wilk    V3        0.9974    0.9561    YES
## 4 Shapiro-Wilk    V4        0.9898    0.0760    YES
## 5 Shapiro-Wilk    V5        0.9952    0.6340    YES
## 6 Shapiro-Wilk    V6        0.9872    0.0248    NO
## 7 Shapiro-Wilk    V7        0.9799    0.0013    NO
## 8 Shapiro-Wilk    V8        0.9926    0.2431    YES
## 9 Shapiro-Wilk    V9        0.9937    0.3847    YES
##
## $Descriptives
##      n      Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.050533  1.0122 -0.03637 -2.777 3.000 -0.7001 0.6906
## V2 250 -0.057716  1.0090 -0.03678 -3.000 2.684 -0.8450 0.7260
## V3 250  0.024191  1.0086  0.07531 -2.956 3.000 -0.6336 0.6876
## V4 250  0.087338  1.0279  0.18634 -3.000 2.962 -0.5902 0.7393
## V5 250  0.070994  1.0120  0.01472 -2.858 3.000 -0.6027 0.7252
## V6 250  0.032343  1.0344  0.10124 -3.324 3.006 -0.5677 0.7219
## V7 250  0.027059  0.9399 -0.12191 -2.202 3.000 -0.5809 0.5485
## V8 250 -0.036554  0.9868 -0.09124 -3.000 3.747 -0.6988 0.6763
## V9 250 -0.006686  1.0006 -0.06066 -2.285 3.063 -0.7067 0.6601
##        Skew  Kurtosis
## V1  0.05594 -0.017509
## V2 -0.06706 -0.245770
## V3 -0.04505 -0.052613
## V4 -0.31408 -0.008075
## V5  0.15570  0.009624
## V6 -0.38932  0.425754
## V7  0.52965  0.134834
## V8  0.24031  0.373272
## V9  0.22770 -0.158336
```

```r
mvn(Data, mvnTest = "energy")
```

```
## $multivariateNormality
##          Test Statistic p value MVN
## 1 E-statistic      7019       0  NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9961    0.7912    YES
## 2 Shapiro-Wilk    V2        0.9949    0.5658    YES
## 3 Shapiro-Wilk    V3        0.9974    0.9561    YES
```

```
## 4 Shapiro-Wilk    V4        0.9898    0.0760    YES
## 5 Shapiro-Wilk    V5        0.9952    0.6340    YES
## 6 Shapiro-Wilk    V6        0.9872    0.0248    NO
## 7 Shapiro-Wilk    V7        0.9799    0.0013    NO
## 8 Shapiro-Wilk    V8        0.9926    0.2431    YES
## 9 Shapiro-Wilk    V9        0.9937    0.3847    YES
##
## $Descriptives
##      n      Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.050533  1.0122 -0.03637 -2.777 3.000 -0.7001 0.6906
## V2 250 -0.057716  1.0090 -0.03678 -3.000 2.684 -0.8450 0.7260
## V3 250  0.024191  1.0086  0.07531 -2.956 3.000 -0.6336 0.6876
## V4 250  0.087338  1.0279  0.18634 -3.000 2.962 -0.5902 0.7393
## V5 250  0.070994  1.0120  0.01472 -2.858 3.000 -0.6027 0.7252
## V6 250  0.032343  1.0344  0.10124 -3.324 3.006 -0.5677 0.7219
## V7 250  0.027059  0.9399 -0.12191 -2.202 3.000 -0.5809 0.5485
## V8 250 -0.036554  0.9868 -0.09124 -3.000 3.747 -0.6988 0.6763
## V9 250 -0.006686  1.0006 -0.06066 -2.285 3.063 -0.7067 0.6601
##        Skew  Kurtosis
## V1  0.05594 -0.017509
## V2 -0.06706 -0.245770
## V3 -0.04505 -0.052613
## V4 -0.31408 -0.008075
## V5  0.15570  0.009624
## V6 -0.38932  0.425754
## V7  0.52965  0.134834
## V8  0.24031  0.373272
## V9  0.22770 -0.158336

# Mahalanobis distance is the same if calculated on the principal
# components.
mvn(Data_PC, mvnTest = "mardia", multivariatePlot = "qq", multivariateOutlierMethod = "quan",
    showOutliers = TRUE)
```


Chi−Square Q−Q Plot
Chi−Square Quantile
5
10
15
20
25
0
20
40
60
80
100
120
Squared Mahalanobis Distance

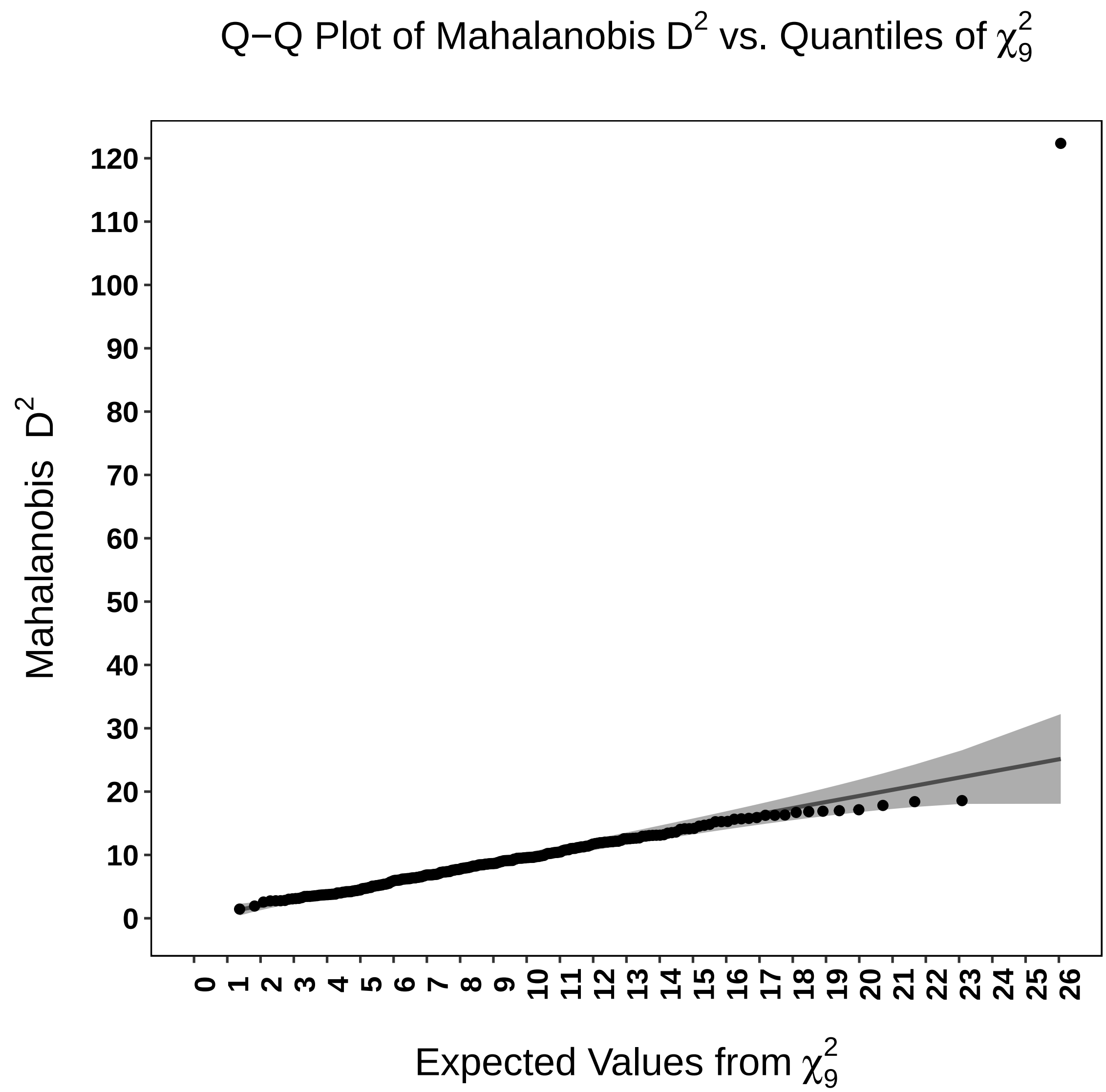



```
## $multivariateNormality
##              Test        Statistic               p value Result
## 1 Mardia Skewness  1380.9251635341 7.25821195066503e-191     NO
## 2 Mardia Kurtosis 27.0433388155401                     0     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    PC1       0.9924  0.2316      YES
## 2 Shapiro-Wilk    PC2       0.9922  0.2098      YES
## 3 Shapiro-Wilk    PC3       0.9883  0.0406      NO
## 4 Shapiro-Wilk    PC4       0.7511  <0.001      NO
## 5 Shapiro-Wilk    PC5       0.9975   0.964      YES
## 6 Shapiro-Wilk    PC6       0.9913  0.1426      YES
```

```
## 7 Shapiro-Wilk    PC7       0.9890  0.0549      YES
## 8 Shapiro-Wilk    PC8       0.9954  0.6699      YES
## 9 Shapiro-Wilk    PC9       0.9887  0.0478      NO
##
## $Descriptives
##       n       Mean Std.Dev   Median    Min    Max    25th   75th
## PC1 250 -6.044e-18       1 -0.05670 -2.539  2.539 -0.6477 0.6272
## PC2 250 -1.270e-17       1 -0.04557 -2.422  3.131 -0.6448 0.5682
## PC3 250 -6.472e-18       1  0.05715 -3.184  2.046 -0.7064 0.7819
## PC4 250  1.084e-17       1 -0.01361 -2.172 10.261 -0.5337 0.4279
## PC5 250 -1.966e-17       1  0.06536 -3.251  2.991 -0.6046 0.6537
## PC6 250  2.231e-17       1  0.01272 -3.088  2.211 -0.6693 0.7078
## PC7 250  2.564e-17       1  0.02463 -2.382  2.150 -0.6762 0.6484
## PC8 250  3.142e-17       1 -0.04238 -3.181  2.734 -0.6428 0.6836
## PC9 250 -1.577e-17       1  0.08547 -2.503  3.103 -0.6990 0.6196
##         Skew Kurtosis
## PC1  0.16996 -0.39703
## PC2  0.21311  0.30420
## PC3 -0.25192 -0.41617
## PC4  4.34404 42.38985
## PC5 -0.03383  0.09839
## PC6 -0.26771 -0.25603
## PC7 -0.19010 -0.43856
## PC8 -0.17270  0.14575
## PC9  0.21327  0.34171
##
## $multivariateOutliers
##   Observation Mahalanobis Distance Outlier
## 1           1               227.26    TRUE
## 2           2                20.09    TRUE
## 3           3                19.97    TRUE
```

```
CV <- cov(Data)
D2_2 <- mahalanobis(Data, center = colMeans(Data), cov = CV)
D2_2 <- as.data.frame(D2_2)
ggplot(D2_2, aes(sample = D2_2)) + stat_qq_band(distribution = "chisq",
    dparams = list(df = 9)) + stat_qq_line(distribution = "chisq",
    dparams = list(df = 9)) + stat_qq(distribution = "qchisq", dparams = list(df = 9)) +
    scale_y_continuous(breaks = seq(0, 120, 10)) + scale_x_continuous(breaks = seq(0,
    26, 1)) + coord_cartesian(xlim = c(0, 26), ylim = c(0, 120)) +
    xlab(expression("Expected Values from" * ~chi[9]^2)) + ylab(expression("Mahalanobis " *
    ~D^2)) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle(expression("Q-Q Plot of Mahalanobis" *
```

```
    ~D^2 * " vs. Quantiles of" * ~chi[9]^2))
```

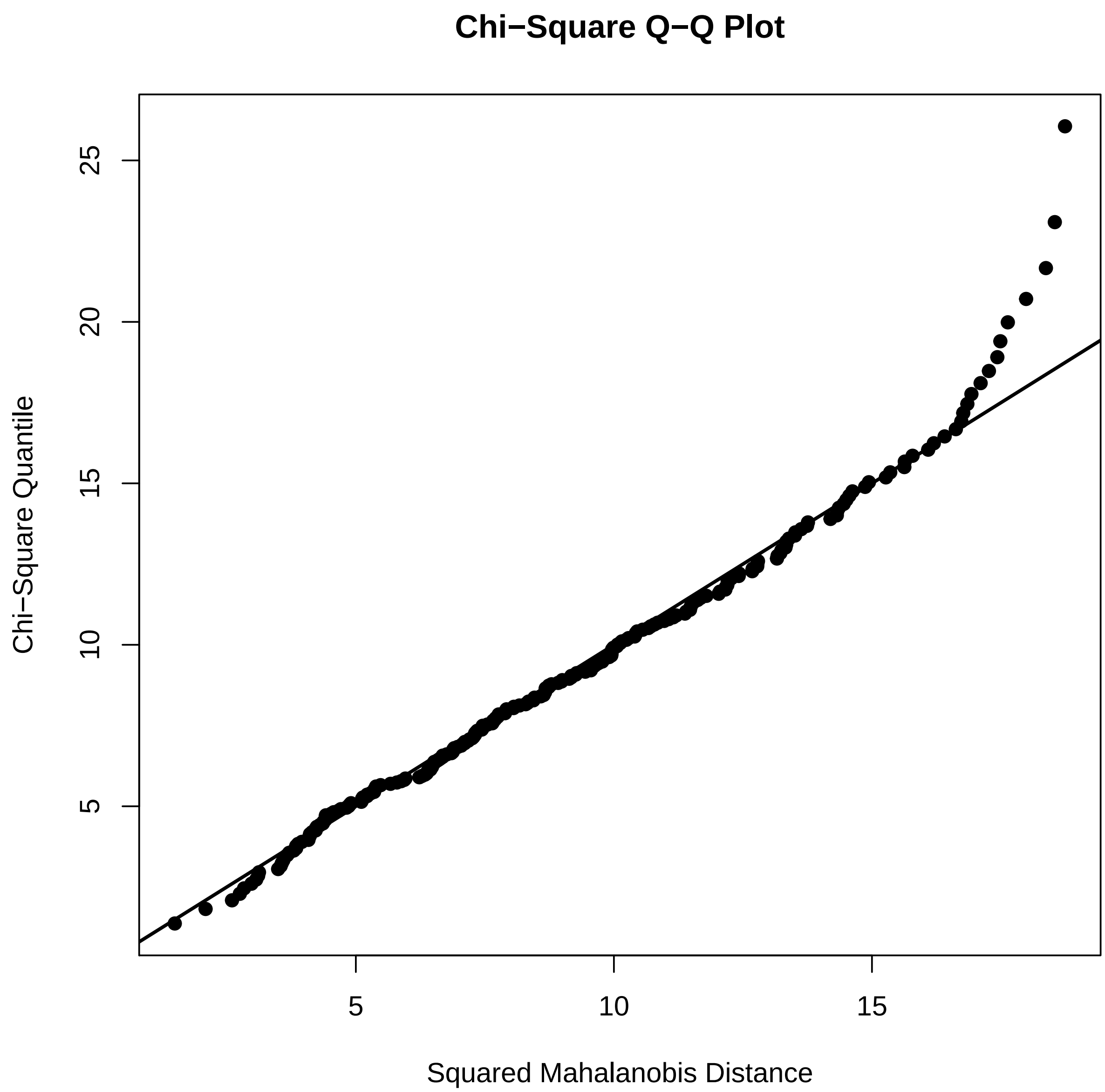


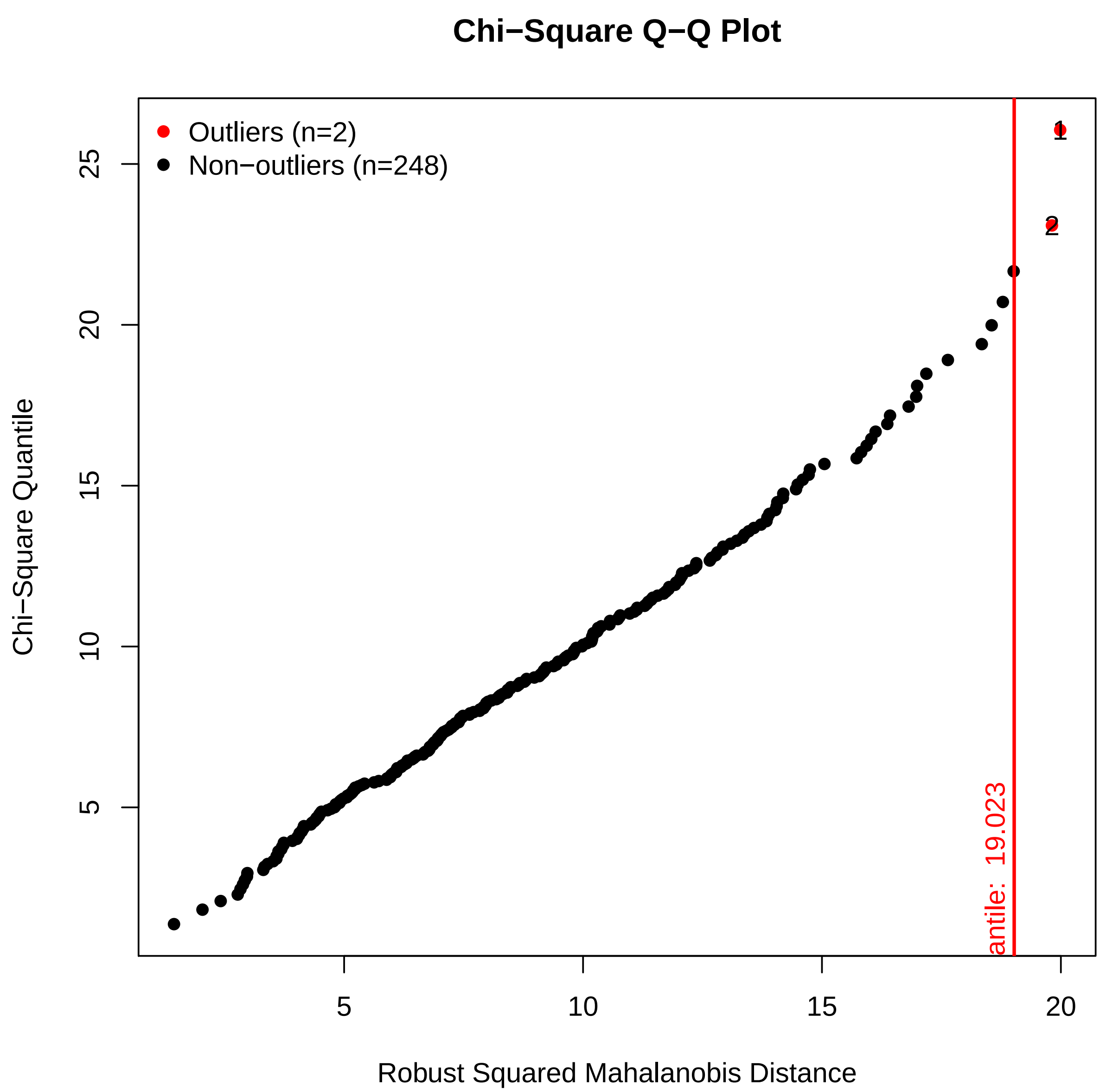
Mahalanobis distance is sensitive to all variables simultaneously, so it detects the unusual pattern.

3.1.2 Data Without Outlier

The original data, without the outlier, more closely approximate multivariate normality.

```
mvn(Data_Original, mvnTest = "mardia", multivariatePlot = "qq", multivariateOutlierMethod = "quan",
    showOutliers = TRUE)
```


Chi−Square Q−Q Plot
Chi−Square Quantile
5
10
15
20
25
5
10
15
Squared Mahalanobis Distance




```
## $multivariateNormality
##              Test        Statistic           p value Result
## 1 Mardia Skewness  146.823933463264 0.842021240357815    YES
## 2 Mardia Kurtosis -1.41439294192397 0.157246561287253    YES
## 3             MVN              <NA>              <NA>    YES
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9966    0.8698    YES
## 2 Shapiro-Wilk    V2        0.9933    0.3230    YES
## 3 Shapiro-Wilk    V3        0.9958    0.7456    YES
## 4 Shapiro-Wilk    V4        0.9902    0.0916    YES
## 5 Shapiro-Wilk    V5        0.9949    0.5662    YES
## 6 Shapiro-Wilk    V6        0.9891    0.0572    YES
```

```
## 7 Shapiro-Wilk    V7        0.9831    0.0046    NO
## 8 Shapiro-Wilk    V8        0.9905    0.1027    YES
## 9 Shapiro-Wilk    V9        0.9950    0.5879    YES
##
## $Descriptives
##      n     Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.06381  0.9937 -0.03989 -2.777 2.753 -0.7001 0.6870
## V2 250 -0.04388  0.9921 -0.02707 -2.450 2.684 -0.8138 0.7260
## V3 250  0.01566  0.9923  0.07531 -2.956 2.475 -0.6336 0.6876
## V4 250  0.10087  1.0092  0.20015 -2.616 2.962 -0.5678 0.7393
## V5 250  0.06034  0.9950  0.01472 -2.858 2.552 -0.6027 0.7191
## V6 250  0.04739  1.0173  0.11270 -3.324 3.006 -0.5420 0.7229
## V7 250  0.01707  0.9212 -0.12191 -2.202 2.841 -0.5809 0.5452
## V8 250 -0.02241  0.9693 -0.07922 -2.374 3.747 -0.6829 0.6763
## V9 250 -0.01763  0.9824 -0.06066 -2.285 3.063 -0.7067 0.6347
##         Skew Kurtosis
## V1 -0.016758 -0.14541
## V2 -0.007312 -0.36275
## V3 -0.127063 -0.17934
## V4 -0.257317 -0.11760
## V5  0.093887 -0.07536
## V6 -0.346035  0.36598
## V7  0.460968 -0.01794
## V8  0.325057  0.25375
## V9  0.159305 -0.28620
##
## $multivariateOutliers
##   Observation Mahalanobis Distance Outlier
## 1           1                19.98    TRUE
## 2           2                19.81    TRUE
```

```
mvn(Data_Original, mvnTest = "royston")
```

```
## $multivariateNormality
##      Test    H p value MVN
## 1 Royston 18.4 0.02342  NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9966    0.8698    YES
## 2 Shapiro-Wilk    V2        0.9933    0.3230    YES
## 3 Shapiro-Wilk    V3        0.9958    0.7456    YES
## 4 Shapiro-Wilk    V4        0.9902    0.0916    YES
## 5 Shapiro-Wilk    V5        0.9949    0.5662    YES
## 6 Shapiro-Wilk    V6        0.9891    0.0572    YES
## 7 Shapiro-Wilk    V7        0.9831    0.0046    NO
## 8 Shapiro-Wilk    V8        0.9905    0.1027    YES
## 9 Shapiro-Wilk    V9        0.9950    0.5879    YES
##
## $Descriptives
##      n     Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.06381  0.9937 -0.03989 -2.777 2.753 -0.7001 0.6870
## V2 250 -0.04388  0.9921 -0.02707 -2.450 2.684 -0.8138 0.7260
## V3 250  0.01566  0.9923  0.07531 -2.956 2.475 -0.6336 0.6876
```

```
## V4 250  0.10087  1.0092  0.20015 -2.616 2.962 -0.5678 0.7393
## V5 250  0.06034  0.9950  0.01472 -2.858 2.552 -0.6027 0.7191
## V6 250  0.04739  1.0173  0.11270 -3.324 3.006 -0.5420 0.7229
## V7 250  0.01707  0.9212 -0.12191 -2.202 2.841 -0.5809 0.5452
## V8 250 -0.02241  0.9693 -0.07922 -2.374 3.747 -0.6829 0.6763
## V9 250 -0.01763  0.9824 -0.06066 -2.285 3.063 -0.7067 0.6347
##         Skew Kurtosis
## V1 -0.016758 -0.14541
## V2 -0.007312 -0.36275
## V3 -0.127063 -0.17934
## V4 -0.257317 -0.11760
## V5  0.093887 -0.07536
## V6 -0.346035  0.36598
## V7  0.460968 -0.01794
## V8  0.325057  0.25375
## V9  0.159305 -0.28620
```

```
mvn(Data_Original, mvnTest = "hz")
```

```
## $multivariateNormality
##            Test     HZ p value MVN
## 1 Henze-Zirkler 0.9649  0.7945 YES
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9966    0.8698    YES
## 2 Shapiro-Wilk    V2        0.9933    0.3230    YES
## 3 Shapiro-Wilk    V3        0.9958    0.7456    YES
## 4 Shapiro-Wilk    V4        0.9902    0.0916    YES
## 5 Shapiro-Wilk    V5        0.9949    0.5662    YES
## 6 Shapiro-Wilk    V6        0.9891    0.0572    YES
## 7 Shapiro-Wilk    V7        0.9831    0.0046    NO
## 8 Shapiro-Wilk    V8        0.9905    0.1027    YES
## 9 Shapiro-Wilk    V9        0.9950    0.5879    YES
##
## $Descriptives
##      n     Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.06381  0.9937 -0.03989 -2.777 2.753 -0.7001 0.6870
## V2 250 -0.04388  0.9921 -0.02707 -2.450 2.684 -0.8138 0.7260
## V3 250  0.01566  0.9923  0.07531 -2.956 2.475 -0.6336 0.6876
## V4 250  0.10087  1.0092  0.20015 -2.616 2.962 -0.5678 0.7393
## V5 250  0.06034  0.9950  0.01472 -2.858 2.552 -0.6027 0.7191
## V6 250  0.04739  1.0173  0.11270 -3.324 3.006 -0.5420 0.7229
## V7 250  0.01707  0.9212 -0.12191 -2.202 2.841 -0.5809 0.5452
## V8 250 -0.02241  0.9693 -0.07922 -2.374 3.747 -0.6829 0.6763
## V9 250 -0.01763  0.9824 -0.06066 -2.285 3.063 -0.7067 0.6347
##         Skew Kurtosis
## V1 -0.016758 -0.14541
## V2 -0.007312 -0.36275
## V3 -0.127063 -0.17934
## V4 -0.257317 -0.11760
## V5  0.093887 -0.07536
## V6 -0.346035  0.36598
## V7  0.460968 -0.01794
```

```
## V8  0.325057  0.25375
## V9  0.159305 -0.28620
```

```r
mvn(Data_Original, mvnTest = "dh")
```

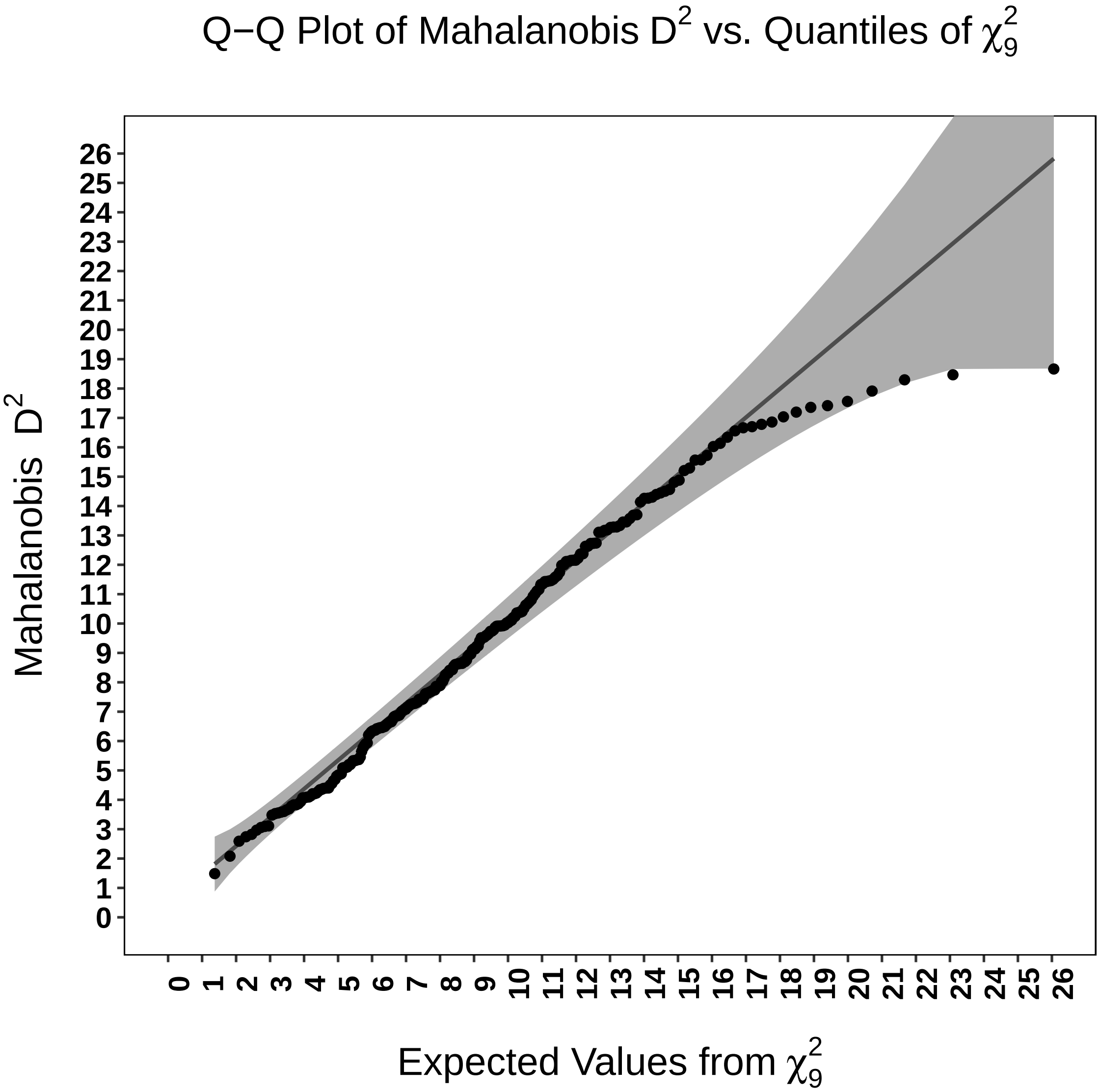
```
## $multivariateNormality
##              Test     E df p value MVN
## 1 Doornik-Hansen 15.39 18  0.6348 YES
##
## $univariateNormality
##            Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9966    0.8698    YES
## 2 Shapiro-Wilk    V2        0.9933    0.3230    YES
## 3 Shapiro-Wilk    V3        0.9958    0.7456    YES
## 4 Shapiro-Wilk    V4        0.9902    0.0916    YES
## 5 Shapiro-Wilk    V5        0.9949    0.5662    YES
## 6 Shapiro-Wilk    V6        0.9891    0.0572    YES
## 7 Shapiro-Wilk    V7        0.9831    0.0046    NO
## 8 Shapiro-Wilk    V8        0.9905    0.1027    YES
## 9 Shapiro-Wilk    V9        0.9950    0.5879    YES
##
## $Descriptives
##      n     Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.06381  0.9937 -0.03989 -2.777 2.753 -0.7001 0.6870
## V2 250 -0.04388  0.9921 -0.02707 -2.450 2.684 -0.8138 0.7260
## V3 250  0.01566  0.9923  0.07531 -2.956 2.475 -0.6336 0.6876
## V4 250  0.10087  1.0092  0.20015 -2.616 2.962 -0.5678 0.7393
## V5 250  0.06034  0.9950  0.01472 -2.858 2.552 -0.6027 0.7191
## V6 250  0.04739  1.0173  0.11270 -3.324 3.006 -0.5420 0.7229
## V7 250  0.01707  0.9212 -0.12191 -2.202 2.841 -0.5809 0.5452
## V8 250 -0.02241  0.9693 -0.07922 -2.374 3.747 -0.6829 0.6763
## V9 250 -0.01763  0.9824 -0.06066 -2.285 3.063 -0.7067 0.6347
##         Skew Kurtosis
## V1 -0.016758 -0.14541
## V2 -0.007312 -0.36275
## V3 -0.127063 -0.17934
## V4 -0.257317 -0.11760
## V5  0.093887 -0.07536
## V6 -0.346035  0.36598
## V7  0.460968 -0.01794
## V8  0.325057  0.25375
## V9  0.159305 -0.28620
```

```r
mvn(Data_Original, mvnTest = "energy")
```

```
## $multivariateNormality
##          Test Statistic p value MVN
## 1 E-statistic     1.472   0.925 YES
##
## $univariateNormality
##            Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk    V1        0.9966    0.8698    YES
## 2 Shapiro-Wilk    V2        0.9933    0.3230    YES
## 3 Shapiro-Wilk    V3        0.9958    0.7456    YES
## 4 Shapiro-Wilk    V4        0.9902    0.0916    YES
```

```
## 5 Shapiro-Wilk    V5        0.9949    0.5662    YES
## 6 Shapiro-Wilk    V6        0.9891    0.0572    YES
## 7 Shapiro-Wilk    V7        0.9831    0.0046    NO
## 8 Shapiro-Wilk    V8        0.9905    0.1027    YES
## 9 Shapiro-Wilk    V9        0.9950    0.5879    YES
##
## $Descriptives
##      n     Mean Std.Dev   Median    Min   Max    25th   75th
## V1 250 -0.06381  0.9937 -0.03989 -2.777 2.753 -0.7001 0.6870
## V2 250 -0.04388  0.9921 -0.02707 -2.450 2.684 -0.8138 0.7260
## V3 250  0.01566  0.9923  0.07531 -2.956 2.475 -0.6336 0.6876
## V4 250  0.10087  1.0092  0.20015 -2.616 2.962 -0.5678 0.7393
## V5 250  0.06034  0.9950  0.01472 -2.858 2.552 -0.6027 0.7191
## V6 250  0.04739  1.0173  0.11270 -3.324 3.006 -0.5420 0.7229
## V7 250  0.01707  0.9212 -0.12191 -2.202 2.841 -0.5809 0.5452
## V8 250 -0.02241  0.9693 -0.07922 -2.374 3.747 -0.6829 0.6763
## V9 250 -0.01763  0.9824 -0.06066 -2.285 3.063 -0.7067 0.6347
##         Skew Kurtosis
## V1 -0.016758 -0.14541
## V2 -0.007312 -0.36275
## V3 -0.127063 -0.17934
## V4 -0.257317 -0.11760
## V5  0.093887 -0.07536
## V6 -0.346035  0.36598
## V7  0.460968 -0.01794
## V8  0.325057  0.25375
## V9  0.159305 -0.28620
```

```r
# Get the Mahalanobis distances for later use.
CV <- cov(Data_Original)
D2_1 <- mahalanobis(Data_Original, center = colMeans(Data_Original),
    cov = CV)
D2_1 <- as.data.frame(D2_1)
ggplot(D2_1, aes(sample = D2_1)) + stat_qq_band(distribution = "chisq",
    dparams = list(df = 9)) + stat_qq_line(distribution = "chisq",
    dparams = list(df = 9)) + stat_qq(distribution = "qchisq", dparams = list(df = 9)) +
    scale_y_continuous(breaks = seq(0, 26, 1)) + scale_x_continuous(breaks = seq(0,
    26, 1)) + coord_cartesian(xlim = c(0, 26), ylim = c(0, 26)) +
    xlab(expression("Expected Values from" * ~chi[9]^2)) + ylab(expression("Mahalanobis " *
    ~D^2)) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle(expression("Q-Q Plot of Mahalanobis" *
    ~D^2 * " vs. Quantiles of" * ~chi[9]^2))
```



Q−Q Plot of Mahalanobis D^2 vs. Quantiles of χ^2_9
Mahalanobis D^2
Expected Values from χ^2_9
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26