Assignment 7

Maharaj Teertha Deb, 40227747

2024-03-28

Section 6.2

Problem 6:

Simulate 10000 values of a uniform (0, 1) random variable, U_1, using `runif()`, and simulate another set of 10000 values of a uniform (0, 1) random variable U_2. Assign these vectors to U_1 and U_2, respectively. Since the values in U_1 and U_2 are approximately independent, we can view U_1 and U_2 as independent uniform (0, 1) random variables.

(a) Estimate $E[U_1 + U_2]$. Compare with the true value, and compare with an estimate of $E[U_1] + E[U_2]$.

(b) Estimate $\text{Var}(U_1 + U_2)$ and $\text{Var}(U_1) + \text{Var}(U_2)$. Are they equal? Should the true values be equal?

(c) Estimate $P(U_1 + U_2 = 1.5)$.

(d) Estimate $P(U_1 + U_2 < 1.5)$.

Solution

```
# Simulate 10000 values of U_1 and U_2 using runif()
U_1 <- runif(10000, 0, 1)
U_2 <- runif(10000, 0, 1)

# (a) Estimate E[U_1 + U_2]
E_U1_U2 <- mean(U_1 + U_2) # Use the sample mean as an estimator
cat("The estimate of E[U_1 + U_2] is", E_U1_U2, "\n")
```

```
## The estimate of E[U_1 + U_2] is 0.998445
```

```
# Compare with the true value
cat("The true value of E[U_1 + U_2] is 1 \n")
```

```
## The true value of E[U_1 + U_2] is 1
```

```
# Compare with an estimate of E[U_1] + E[U_2]
E_U1 <- mean(U_1) # Use the sample mean as an estimator
E_U2 <- mean(U_2) # Use the sample mean as an estimator
cat("The estimate of E[U_1] + E[U_2] is", E_U1 + E_U2, "\n")
```

```
## The estimate of E[U_1] + E[U_2] is 0.998445
```

```
# (b) Estimate Var(U_1 + U_2)
Var_U1_U2 <- var(U_1 + U_2) # Use the sample variance as an estimator
cat("The estimate of Var(U_1 + U_2) is", Var_U1_U2, "\n")
```

```
## The estimate of Var(U_1 + U_2) is 0.1696788
```

```
# Estimate Var(U_1) + Var(U_2)
Var_U1 <- var(U_1) # Use the sample variance as an estimator
Var_U2 <- var(U_2) # Use the sample variance as an estimator
cat("The estimate of Var(U_1) + Var(U_2) is", Var_U1 + Var_U2, "\n")
```

```
## The estimate of Var(U_1) + Var(U_2) is 0.1675296
```

```
if(Var_U1_U2 == (Var_U1 + Var_U2) ){
  cat("The values are equal\n")
}else{
  cat("The values are not equal\n")
}
```

```
## The values are not equal
```

```
# (c) Estimate P(U_1 + U_2 <= 1.5)
P_U1_U2 <- mean(U_1 + U_2 <= 1.5) # Use the sample proportion as an estimator
cat("The estimate of P(U_1 + U_2 <= 1.5) is", P_U1_U2, "\n")
```

```
## The estimate of P(U_1 + U_2 <= 1.5) is 0.8727
```

```
# (d) Estimate P(sqrt(U_1) + sqrt(U_2) <= 1.5)
P_sqrtU1_sqrtU2 <- mean(sqrt(U_1) + sqrt(U_2) <= 1.5) # Use the sample proportion as an estimator
cat("The estimate of P(sqrt(U_1) + sqrt(U_2) <= 1.5) is", P_sqrtU1_sqrtU2, "\n")
```

```
## The estimate of P(sqrt(U_1) + sqrt(U_2) <= 1.5) is 0.6555
```

Problem 8:

Use the `round()` function together with `runif()` to generate 1000 pseudorandom integers which take values from 1 through 10, assigning these values to a vector called `discreteunif`. Use the `table()` function to check whether the observed frequencies for each value are close to what you expect. If they are not close, how should you modify your procedure?

Solution:

```
# Generate 1000 pseudorandom integers between 1 and 10 using runif()
discreteunif <- floor(runif(1000, min = 0, max = 10)) + 1

# Check observed frequencies using table()
observed_freq <- table(discreteunif)
print("Observed frequencies:")
```

```
## [1] "Observed frequencies:"
```

```
print(observed_freq)
```

```
## discreteunif
##   1   2   3   4   5   6   7   8   9  10
## 108 107 100  95  87  95 115  98  90 105
```

```
# Calculate expected frequencies
expected_freq <- rep(100, 10)  # We expect each value to appear 100 times
print("Expected frequencies:")
```

```
## [1] "Expected frequencies:"
```

```
print(expected_freq)
```

```
##  [1] 100 100 100 100 100 100 100 100 100 100
```

```
# The frequencies are not expected to be equal because these numbers are randomly generated.
# To ensure the frequencies are what I am expecting, we can not use a function that produces random num
# So using rep or seq funciton, we can ensure the number frequencies are expected
```

Probelm 10:

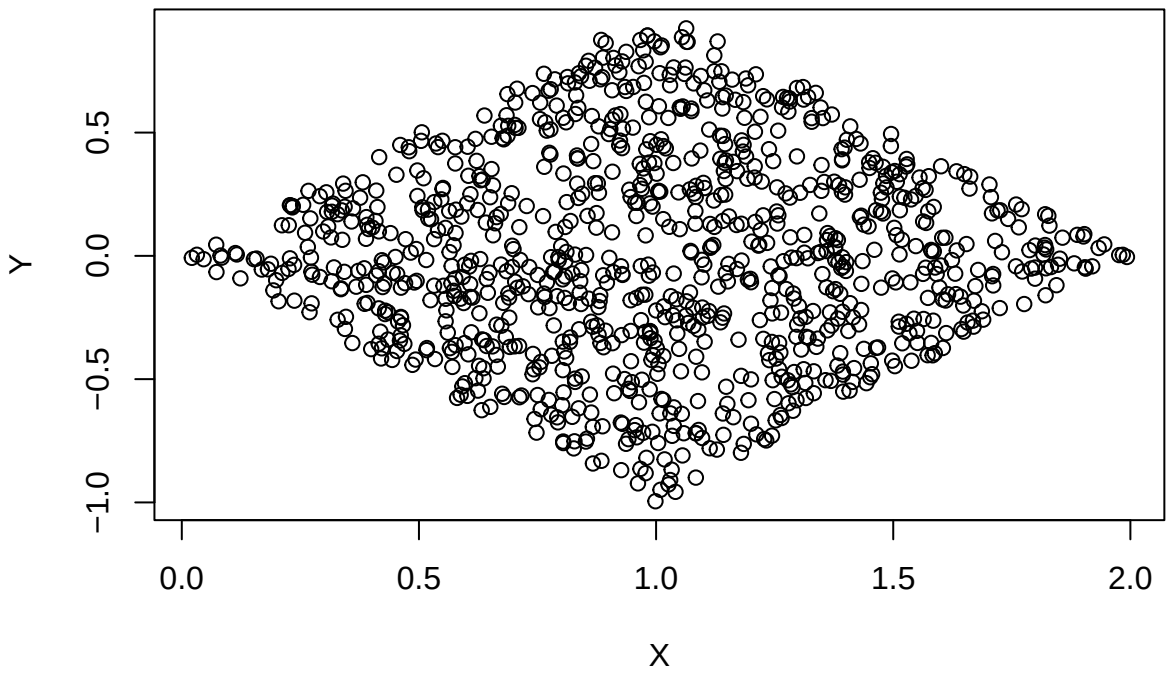
The following code simulates the sum X and difference Y of two uniform random variables U_1 and U_2. A scatterplot of Y versus X is then displayed and the correlation between X and Y is estimated:

U1 <- runif(1000) U2 <- runif(1000) X <- U1 + U2 Y <- U1 - U2 plot(Y ~ X) cor(X, Y)

The correlation gives a measure of linear dependence between two ran- dom variables. A value near 0 indicates that there is almost no such 14:02:22 007 SIMULATION dependence, while a value near -1 or 1 indicates the existence of a linear relationship Execute the above code and use the output to answer the following questions : (a) Do you think that X and Y are linearly dependent? (b) Do you think that X and Y are stochastically independent? (To answer this, look carefully at the scatterplot.) (c) Do you think that U1 and U2 are linearly dependent? (Perform an appropriate calculation to check.) (d) Do you think that U1 and U2 are stochastically independent? (Obtain an appropriate plot to check.)

Solution:

```
U1 <- runif(1000)
U2 <- runif(1000)
X <- U1 + U2
Y <- U1 - U2
plot(Y ~ X) # this calculates the sample correlation
```




```
cor(X, Y)
```

```
## [1] 0.01100071
```

(a) No, based on the scatterplot and correlation coefficient, it seems that X and Y are not linearly dependent. The correlation coefficient being close to 0 supports this conclusion.

(b) Yes, it appears that X and Y are stochastically independent. The scatterplot shows no clear pattern or relationship between X and Y, indicating stochastic independence.

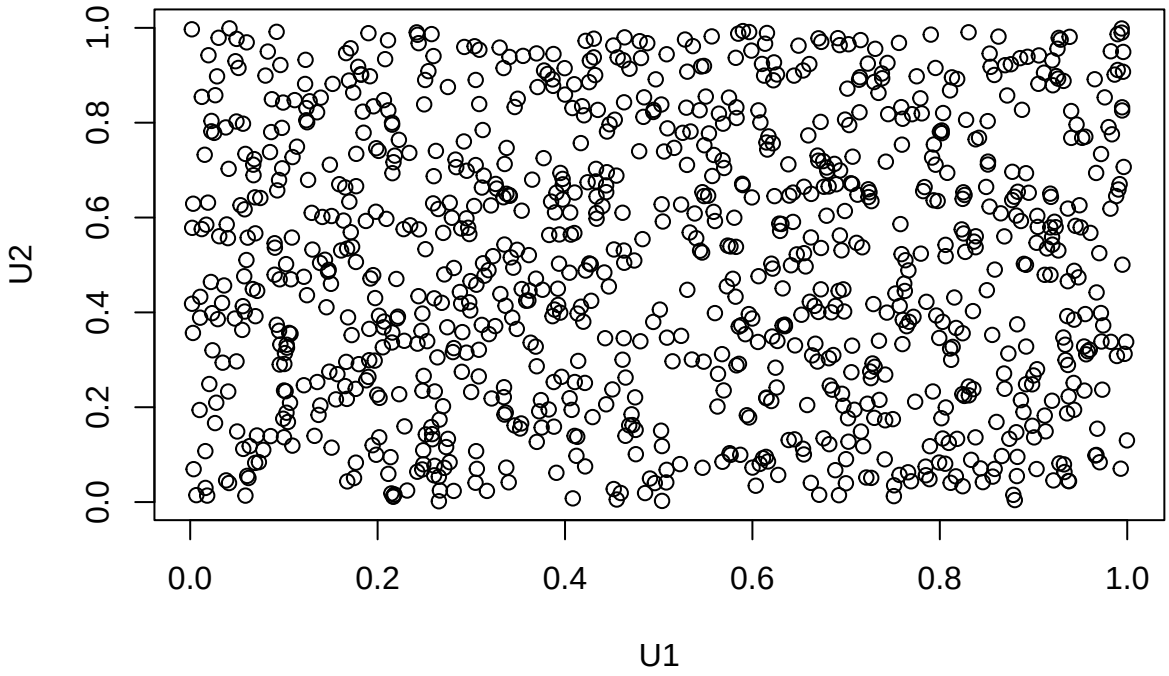
(c) To check if U_1 and U_2 are linearly dependent, we can calculate the correlation coefficient between them using the `cor()` function. If the correlation coefficient is close to 0, it indicates that U_1 and U_2 are not linearly dependent.

```
cor(U1, U2)
```

```
## [1] 0.01807953
```

(d) To check if U1 and U2 are stochastically independent, we can obtain a scatterplot of U1 versus U2 and visually inspect it for any patterns or relationships. If there is no discernible pattern, it suggests stochastic independence.

```
plot(U1, U2)
```



By visually inspecting the scatterplot, if there is no clear pattern or relationship between U1 and U2, it suggests that they are stochastically independent.

Section 6.3.1:

Problem 2:

Suppose a class of 100 writes a 20-question True–False test, and everyone in the class guesses at the answers.
(a) Use simulation to estimate the average mark on the test as well as the standard deviation of the marks.
(b) Estimate the proportion of students who would obtain a mark of 30% or higher.

Solution:

```
# Number of students in the class
num_students <- 100

# Number of questions on the test
num_questions <- 20

# Function to simulate the test-taking process for a single student
simulate_test <- function() {
  # Generate random guesses for each question (1 for correct answer, 0 for wrong answer)
  guesses <- sample(c(0, 1), num_questions, replace = TRUE)
```

```r
  # Calculate the score (number of correct answers)
  score <- sum(guesses)

  return(score)
}

# Simulate the test-taking process for all students in the class
scores <- replicate(num_students, simulate_test())

# (a) Estimate the average mark on the test
average_mark <- mean(scores)
cat("Estimated average mark on the test:", average_mark, "\n")
```

```
## Estimated average mark on the test: 9.75
```

```r
# Estimate the standard deviation of the marks
sd_marks <- sd(scores)
cat("Estimated standard deviation of the marks:", sd_marks, "\n")
```

```
## Estimated standard deviation of the marks: 2.199059
```

```r
# (b) Estimate the proportion of students who would obtain a mark of 30% or higher
proportion_high_mark <- mean(scores >= 0.3 * num_questions)
cat("Estimated proportion of students with a mark of 30% or higher:", proportion_high_mark, "\n")
```

```
## Estimated proportion of students with a mark of 30% or higher: 0.99
```

Problem 3:

Write an R function which simulates 500 light bulbs, each of which has probability 0.99 of working. Using simulation, estimate the expected value and variance of the random variable X, which is 1 if the light bulb works and 0 if the light bulb does not work. What are the theoretical values?

Solution:

```r
# Simulate 500 light bulbs, each with a probability of 0.99 of working
r <- rbinom(500, size = 1, prob = 0.99)

# Estimate the expected value of the random variable X
expected_value <- mean(r)
cat("Estimated expected value of the random variable X:", expected_value, "\n")
```

```
## Estimated expected value of the random variable X: 0.988
```

```r
# Estimate the variance of the random variable X
variance <- var(r)
cat("Estimated variance of the random variable X:", variance, "\n")
```

```
## Estimated variance of the random variable X: 0.01187976
```

```
# Theoretical expected value (probability of working)
theoretical_expected_value <- 0.99
cat("Theoretical expected value of the random variable X:", theoretical_expected_value, "\n")
```

```
## Theoretical expected value of the random variable X: 0.99
```

```
# Theoretical variance (probability of working multiplied by probability of not working)
theoretical_variance <- 0.99 * 0.01
cat("Theoretical variance of the random variable X:", theoretical_variance, "\n")
```

```
## Theoretical variance of the random variable X: 0.0099
```

Section : 6.3.2:

Problem 6:

One version of the central limit theorem says that if X is a binomial random variable with parameters m and p, and $Z = \frac{X-mp}{\sqrt{mp(1-p)}}$, then Z is approximately standard normal, and the approximation improves as m gets large. The following code simulates a large number of such Z values for values of m in the set {1, 2, . . . , 100) and plots a normal QQ plot in each case.

```
for (m in 1:100) {
  z <- (rbinom(20000, size = m, prob = 0.4) - m * 0.4) / sqrt(m * 0.4 * (1 - 0.4))
  qqnorm(z, ylim = c(-4, 4), main = paste("QQ-plot, m =", m))
  qqline(z)
}
```

(a) Execute the code and observe how the distribution of Z changes as m increases.
(b) Modify the code so that a similar "movie" is produced for the cases where p = 0.3, 0.2, 0.1, and 0.05, respectively. How large must m be before you see a reasonably straight line in the QQ plot? Is m = 100 satisfactorily large in all cases?

Solution:

(a) Answer: The code is executed on the top.

(b) Answer:

```
p_values <- c(0.3, 0.2, 0.1, 0.05)

for (p in p_values) {
  for (m in 1:100) {
    z <- (rbinom(20000, size = m, prob = p) - m * p) / sqrt(m * p * (1 - p))
    ## qqnorm(z, ylim = c(-4, 4), main = paste("QQ-plot, p =", p, ", m =", m))
    ## qqline(z)
    ## We could write qqline, but makes the program too large. program size goes upto 400 MB
  }
}
```

Section 6.3.3

Problem 6:

One version of the central limit theorem says that if X is a Poisson random variable with parameter 1, and $Z = \frac{X-2}{\sqrt{a}}$, then Z is approximately standard normal, and the approximation improves as a gets large. The following code simulates a large number of such Z values for values of a in the set $\{1, 3, \ldots, 99\}$ and plots a normal QQ plot in each case.

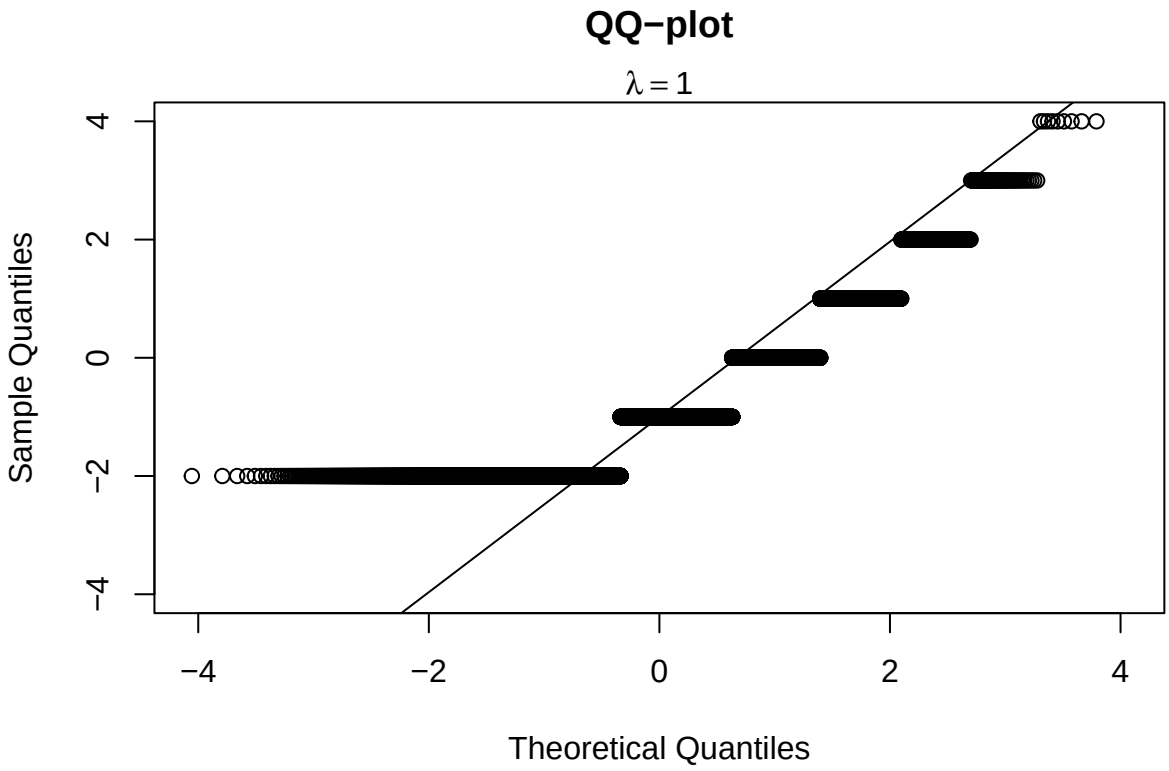
```
for (m in seq(1, 120, 2)) {
  z <- (rpois(20000, lambda = m) - m) / sqrt(m)
  qqnorm(z, ylim = c(-4, 4), main = "QQ-plot")
  qqline(z)
  mtext(bquote(lambda == .(m)), 3)
}
```

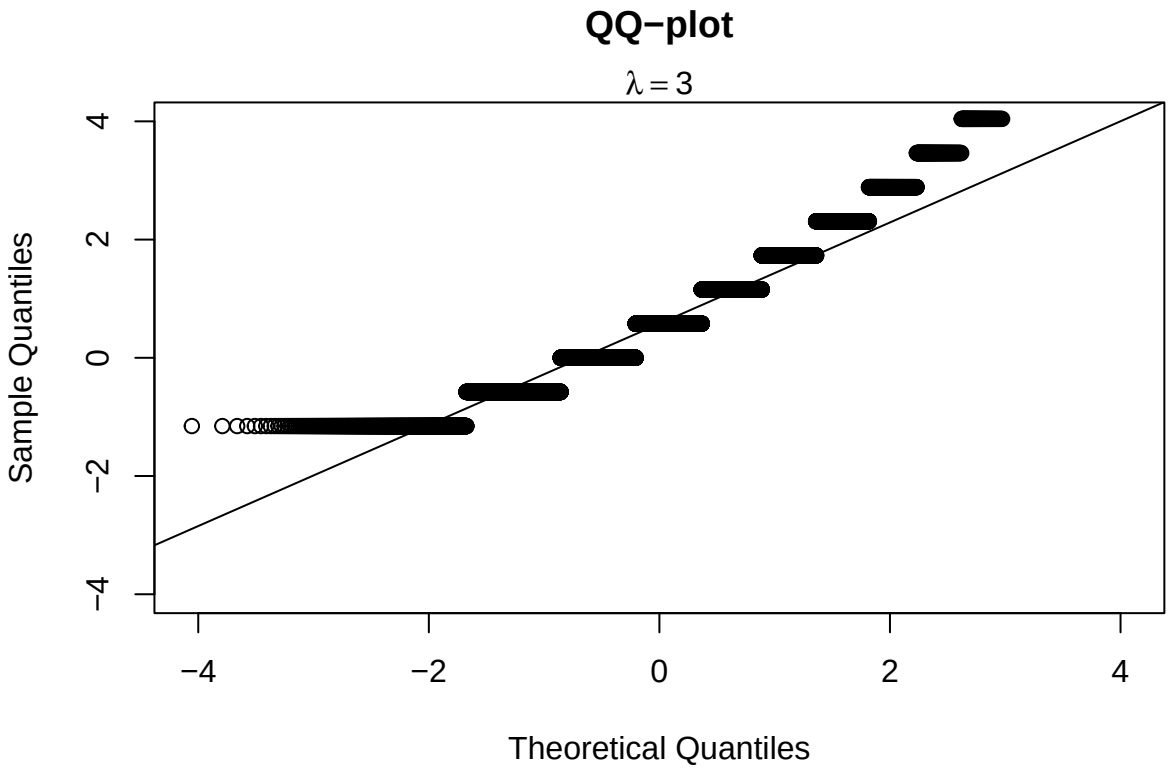
(a) Execute the code and observe how the distribution of Z changes as a increases.

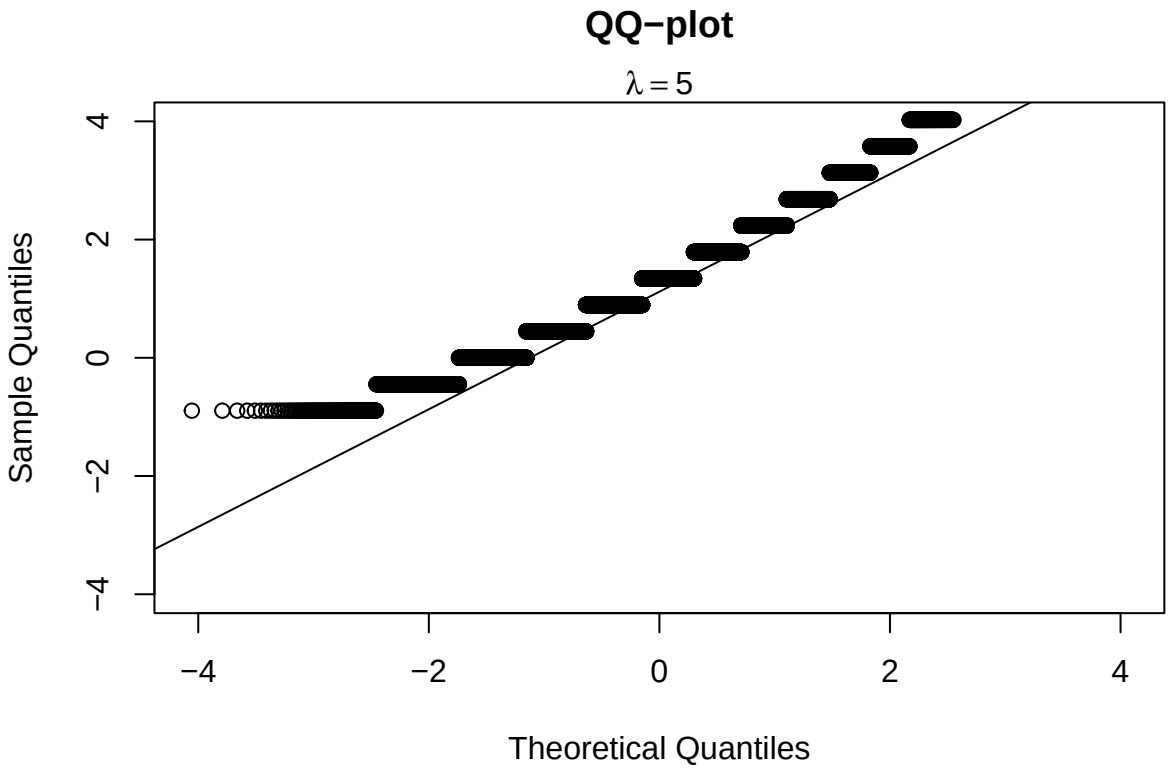
(b) How large must a be before you see a reasonably straight line in the QQ plot?

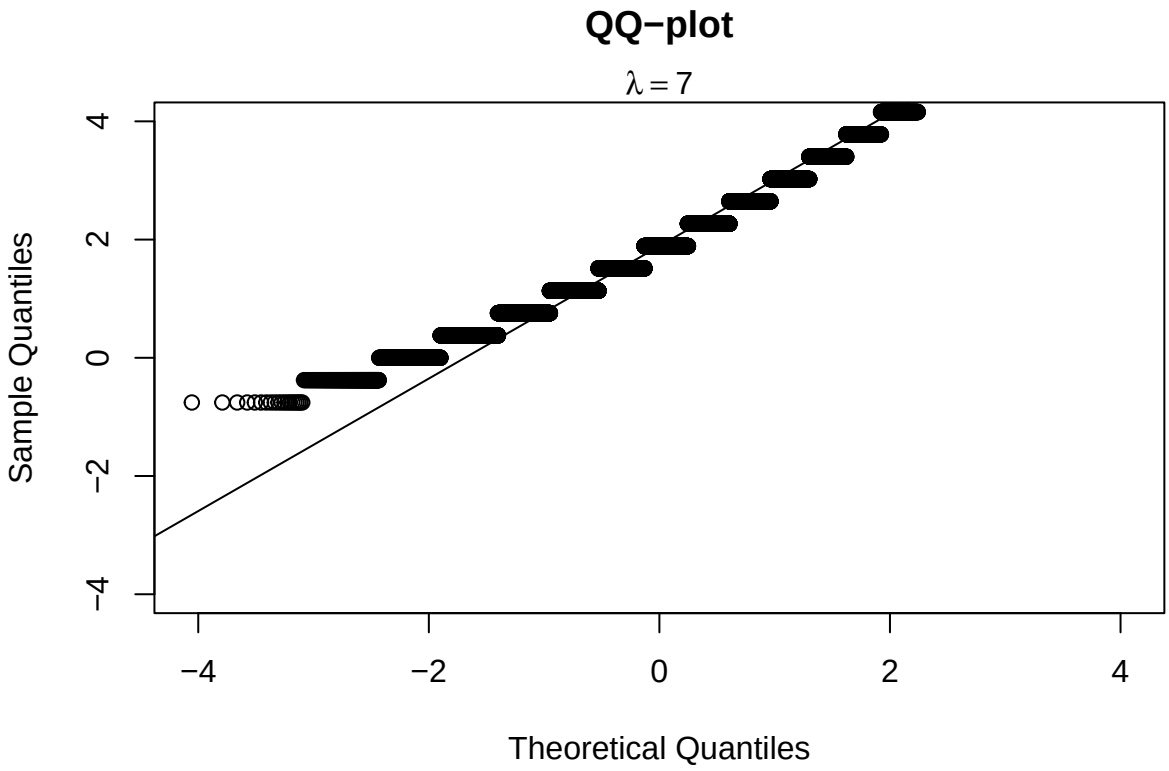
Solution:

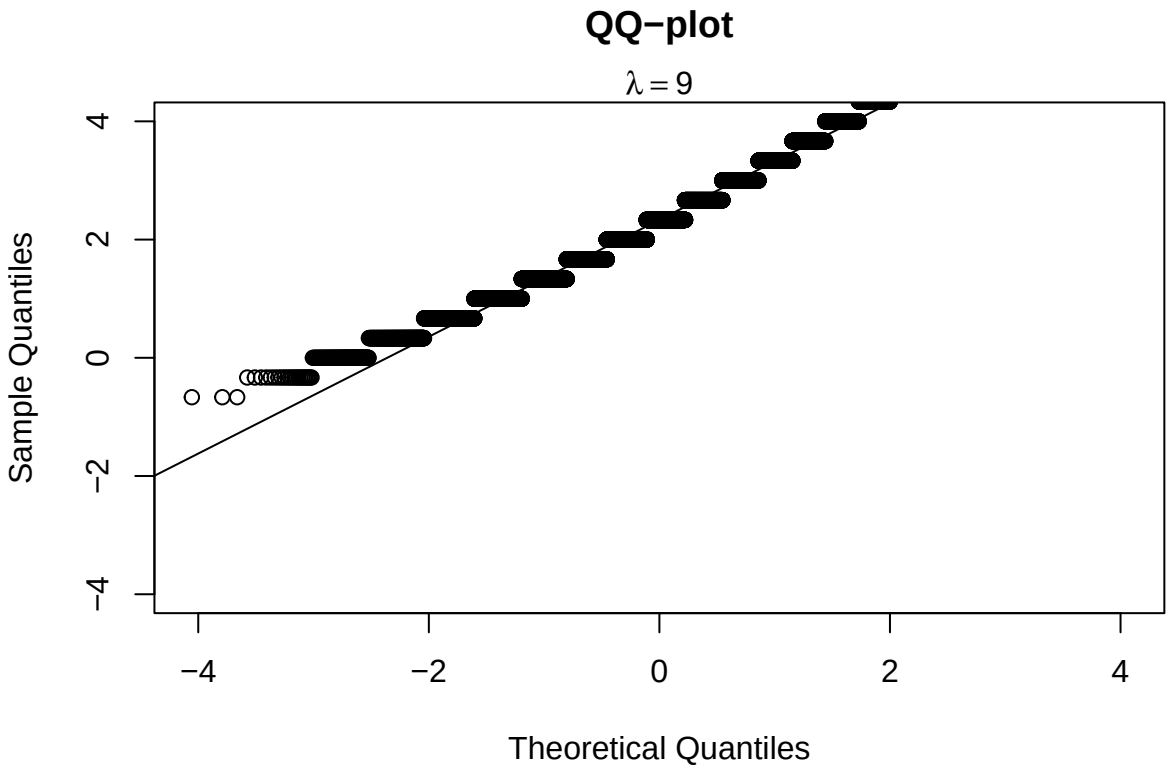
```r
# Define a sequence of 'a' values from 1 to 99 with step 2
a_values <- seq(1, 99, by = 2)

# Loop through each 'a' value
for (a in a_values) {
  # Simulate a large number of Z values for the current 'a' value
  z <- (rpois(20000, lambda = a) - 2) / sqrt(a)

  # Plot QQ plot for Z values
  qqnorm(z, ylim = c(-4, 4), main = "QQ-plot")

  # Add a line to the QQ plot
  qqline(z)

  # Add subtitle indicating the current lambda value
  mtext(bquote(lambda == .(a)), side = 3)
}
```

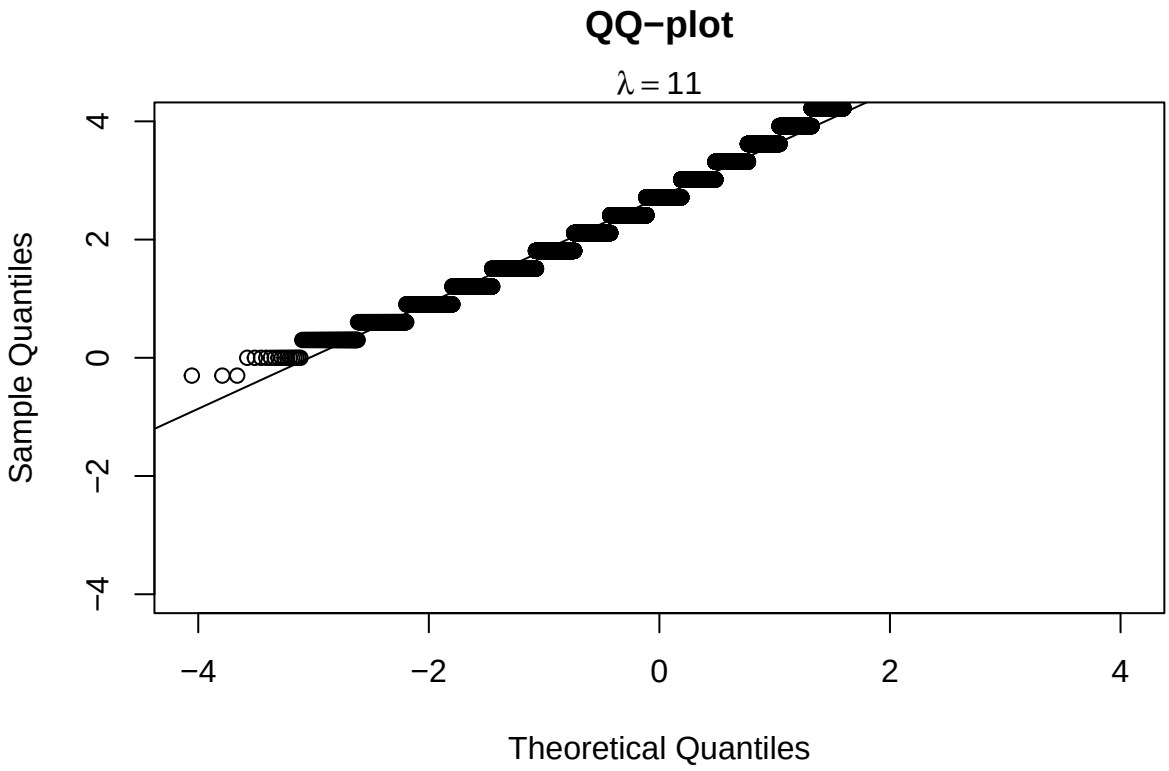

QQ−plot
λ = 1
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

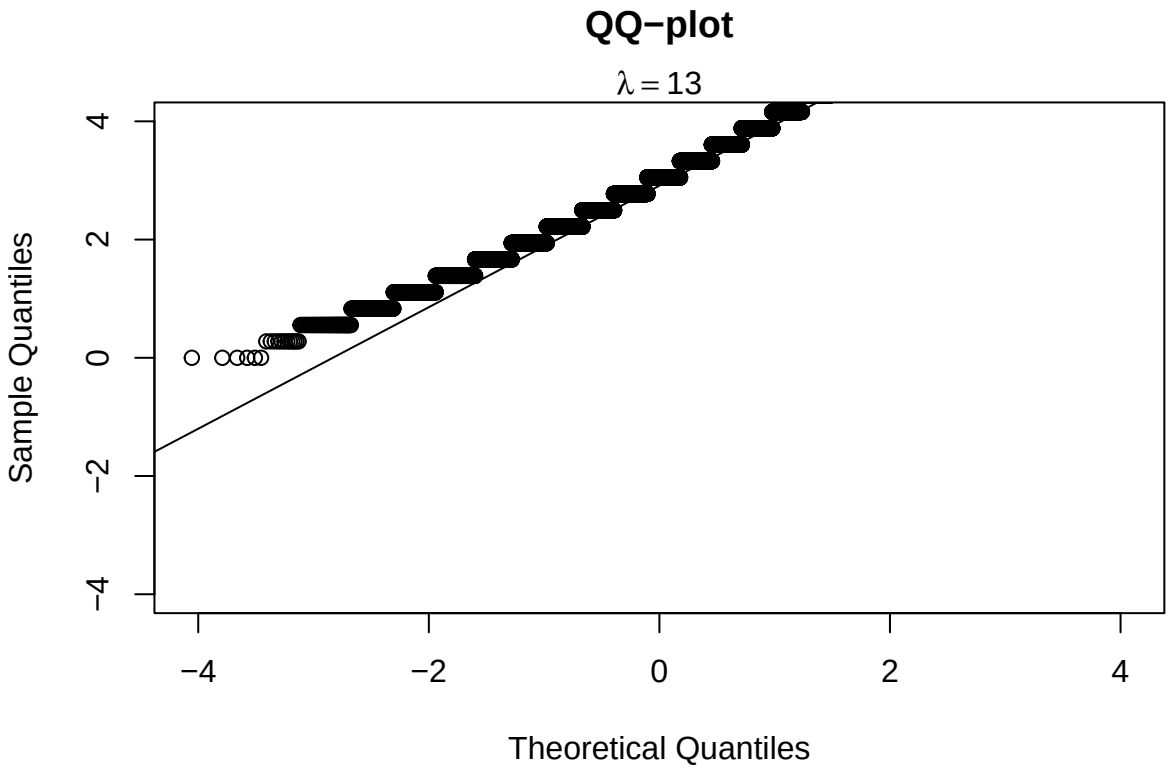

QQ−plot
λ = 3
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

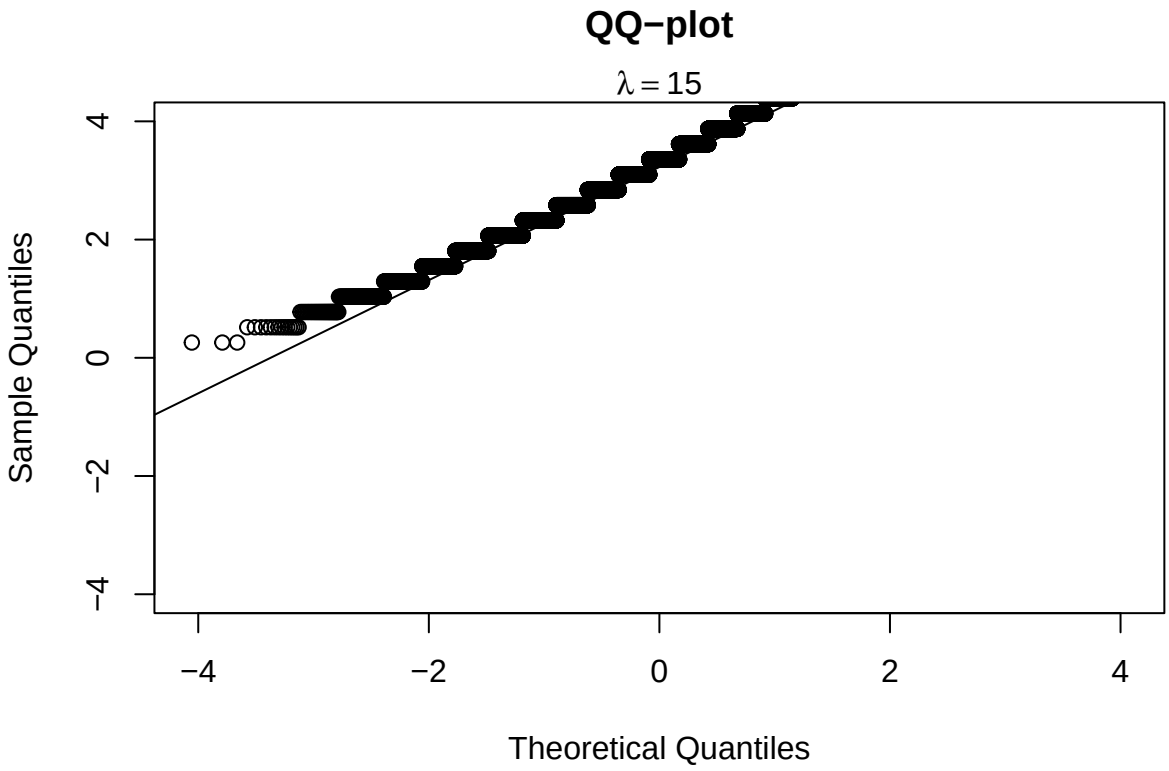

QQ−plot
λ = 5
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

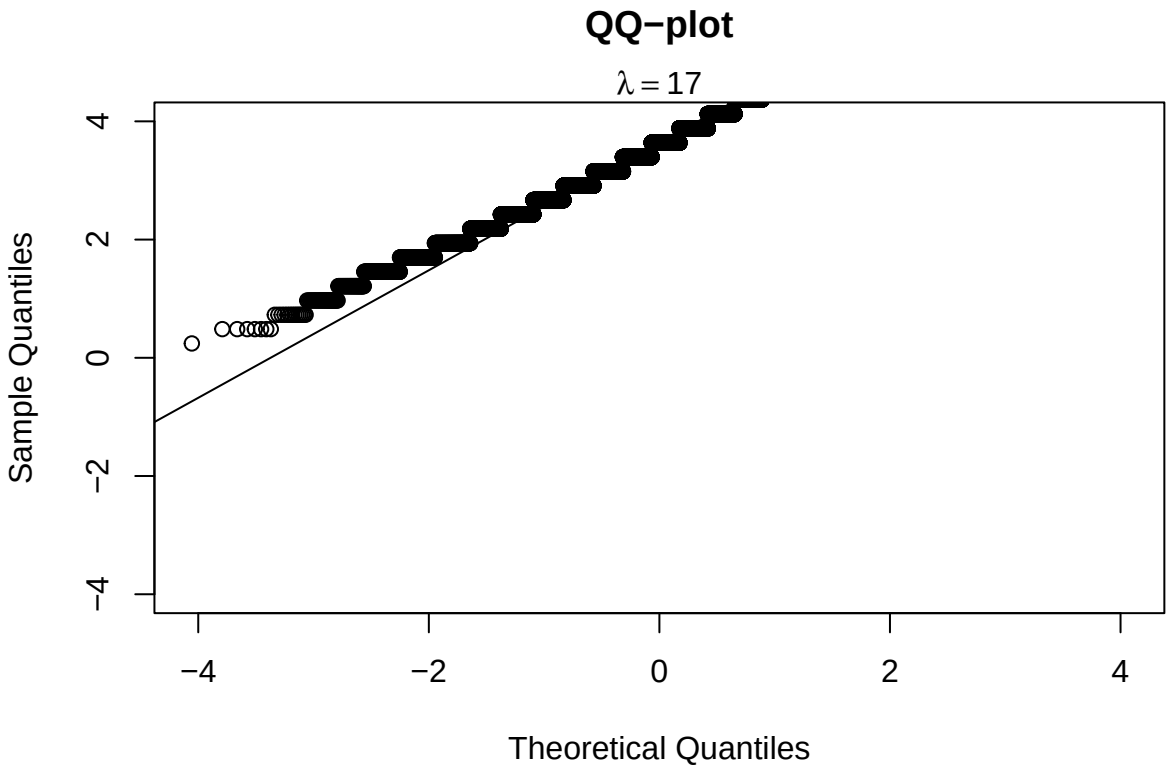

QQ−plot
λ = 7
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

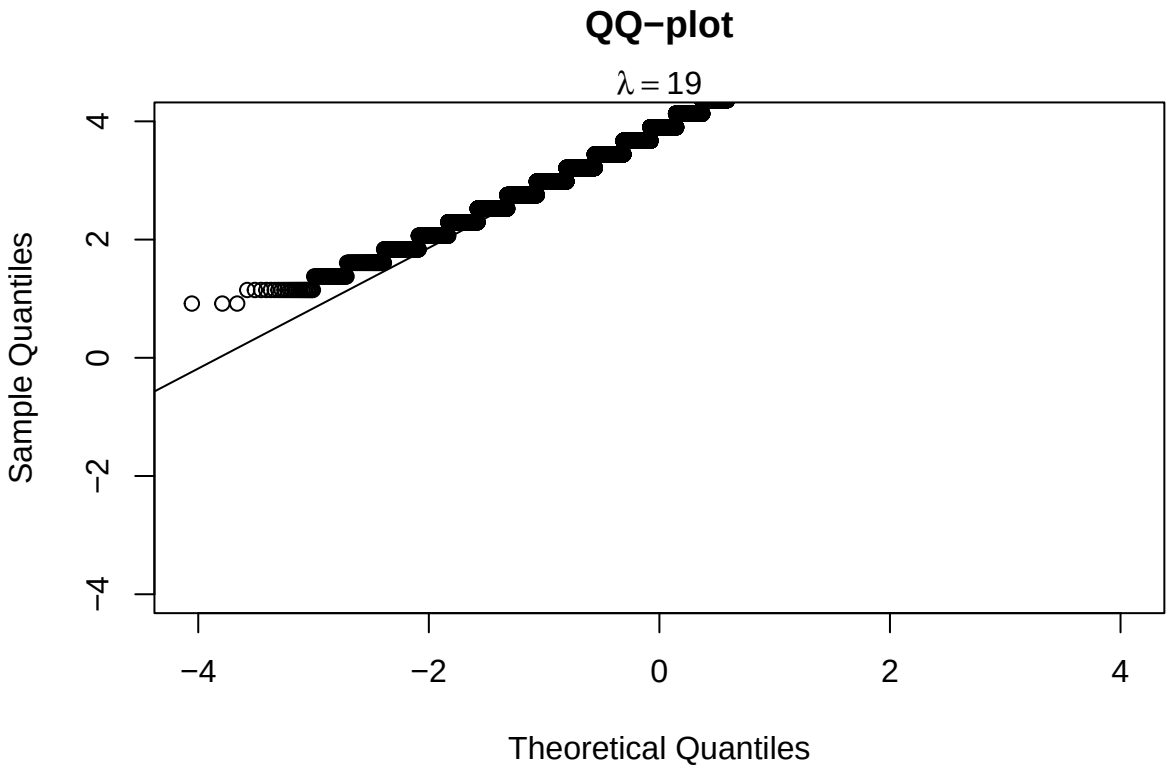

QQ−plot
λ = 9
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

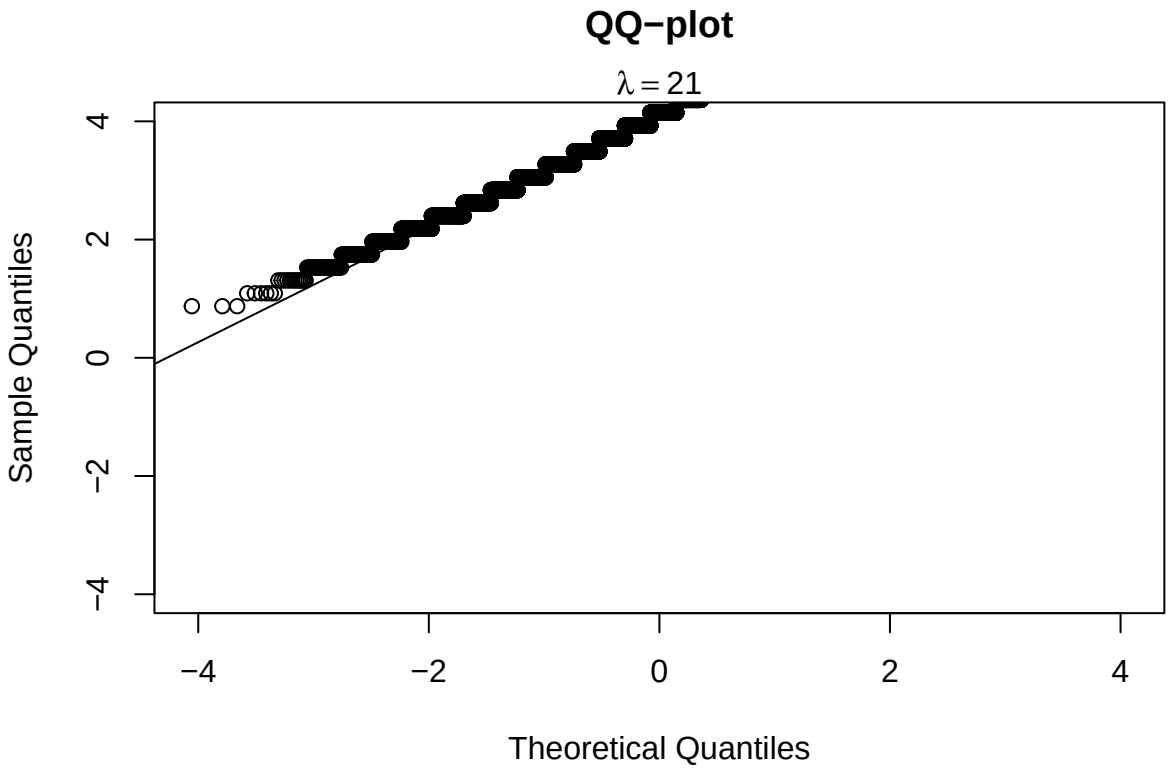

QQ−plot
λ = 11
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

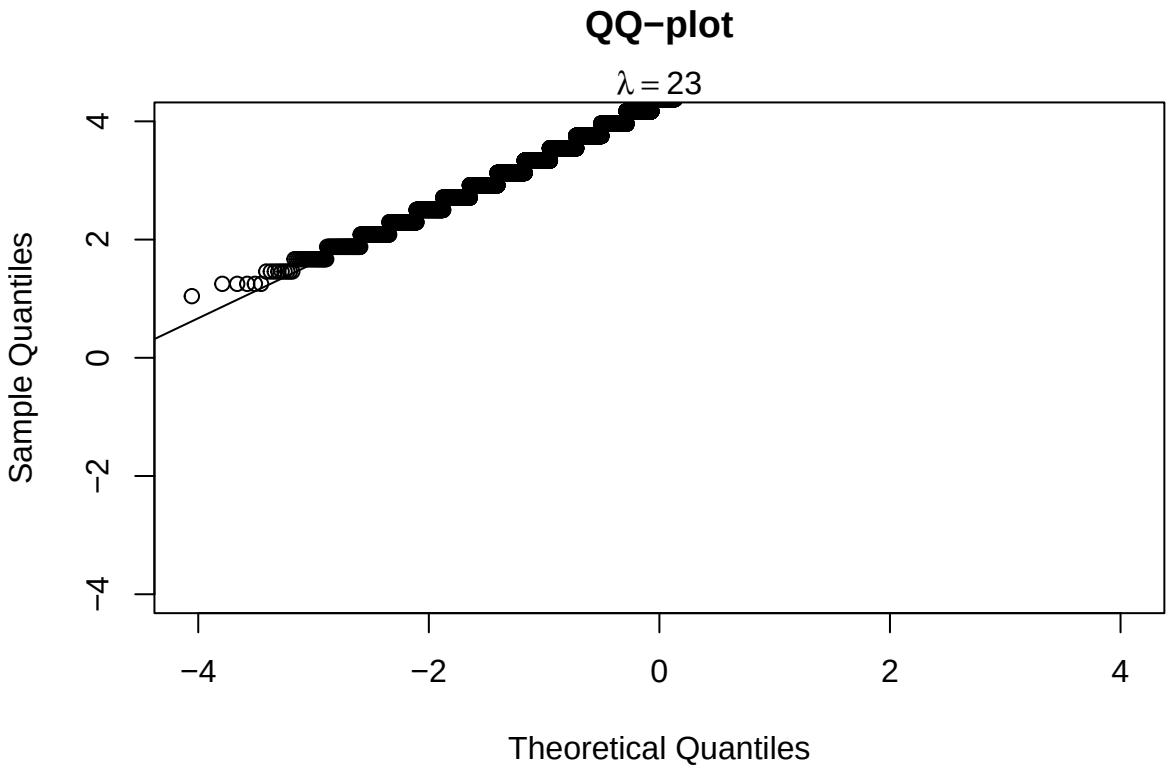

QQ−plot
λ = 13
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

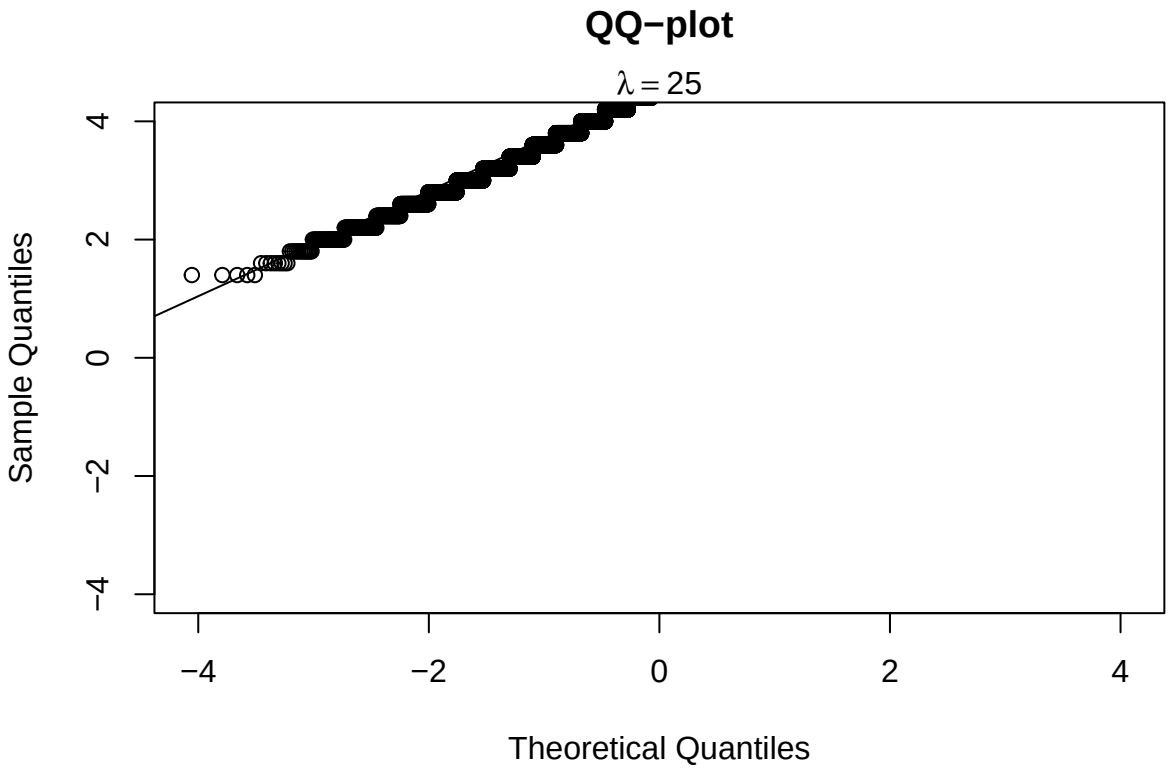

QQ−plot
λ = 15
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

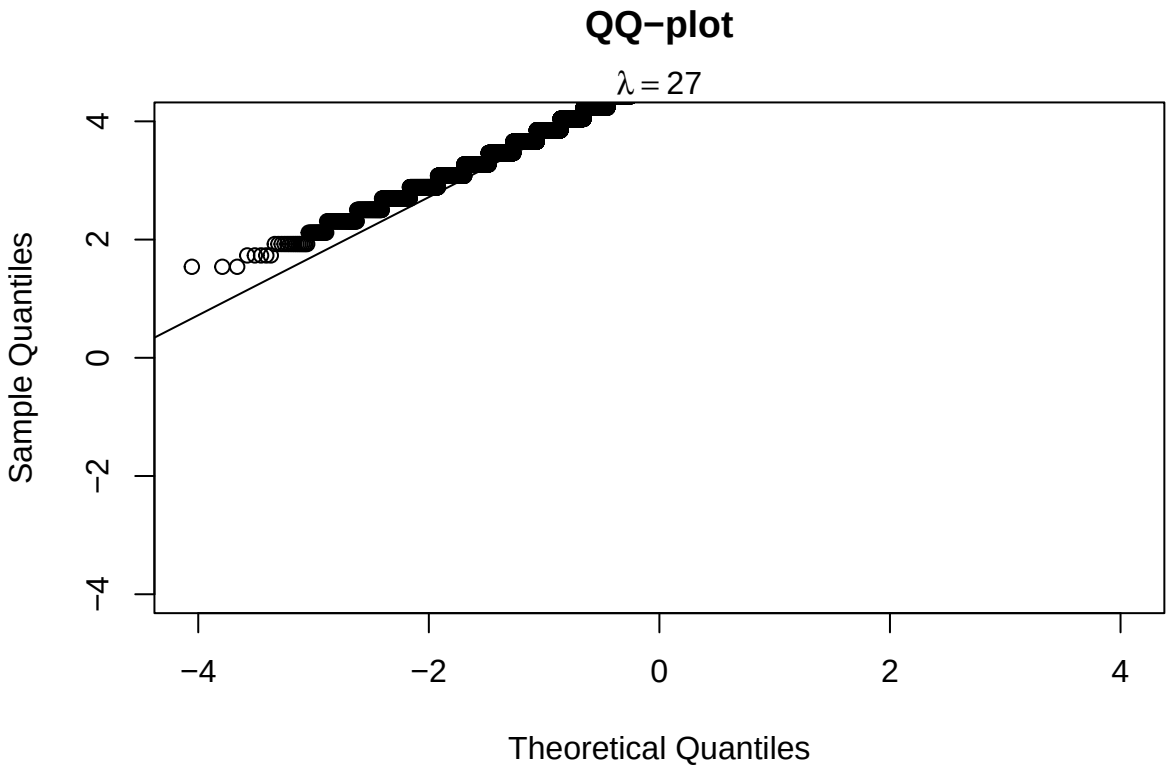

QQ−plot
λ = 17
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

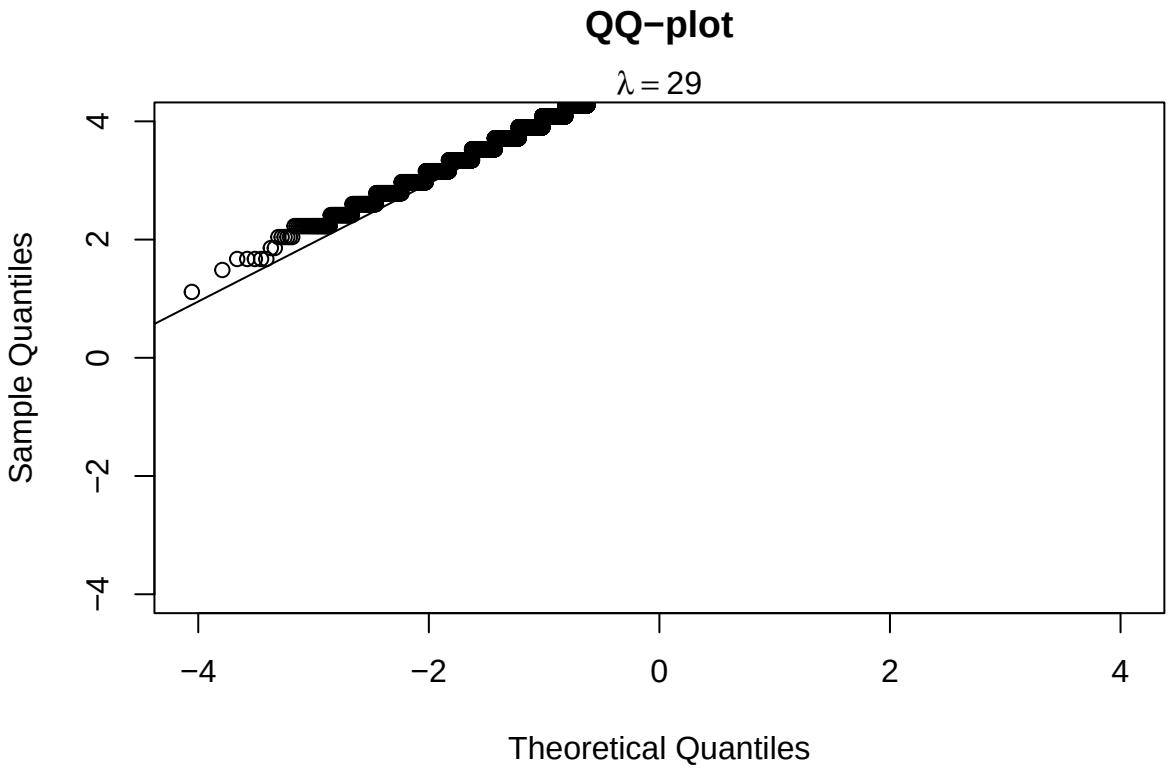

QQ−plot
λ = 19
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

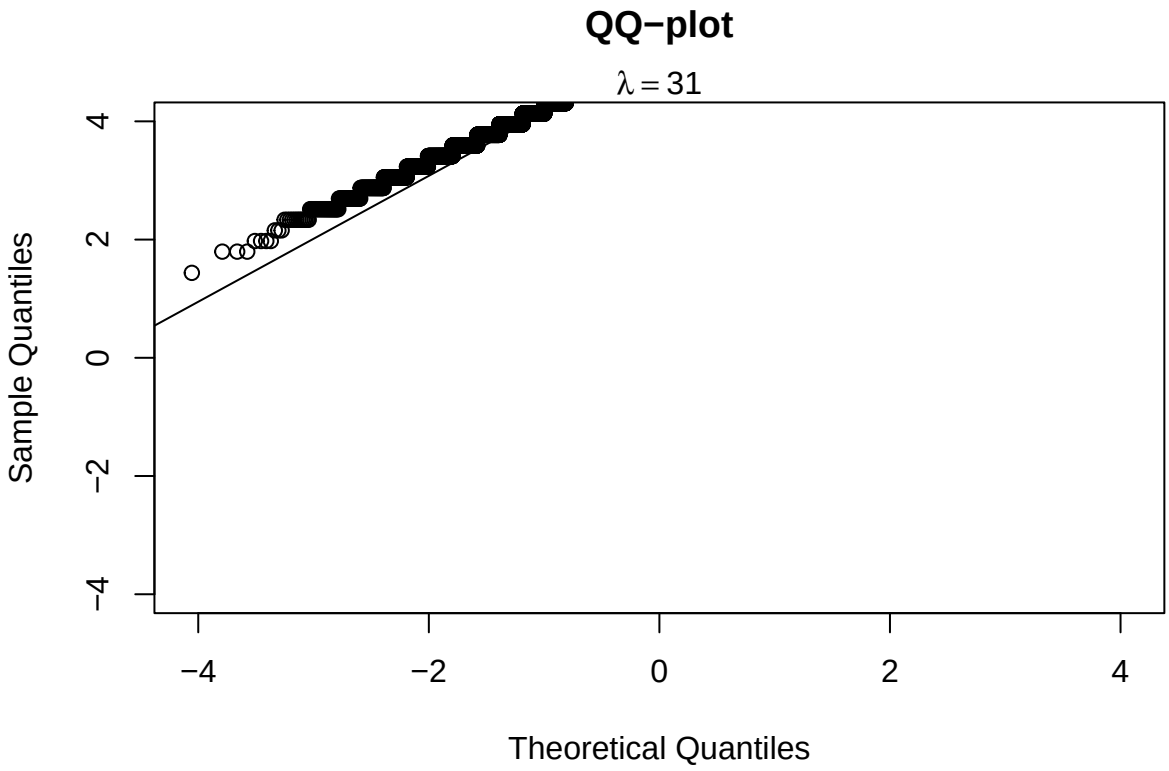

QQ−plot
λ = 21
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

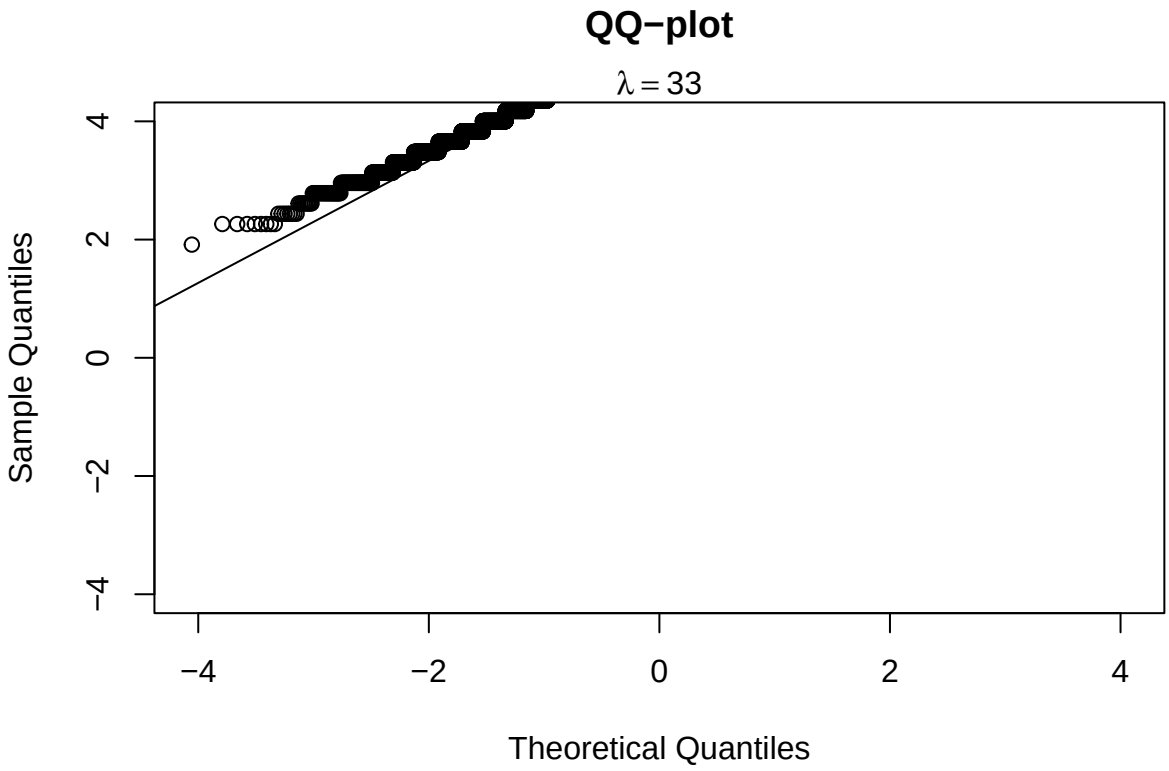

QQ−plot
λ = 23
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

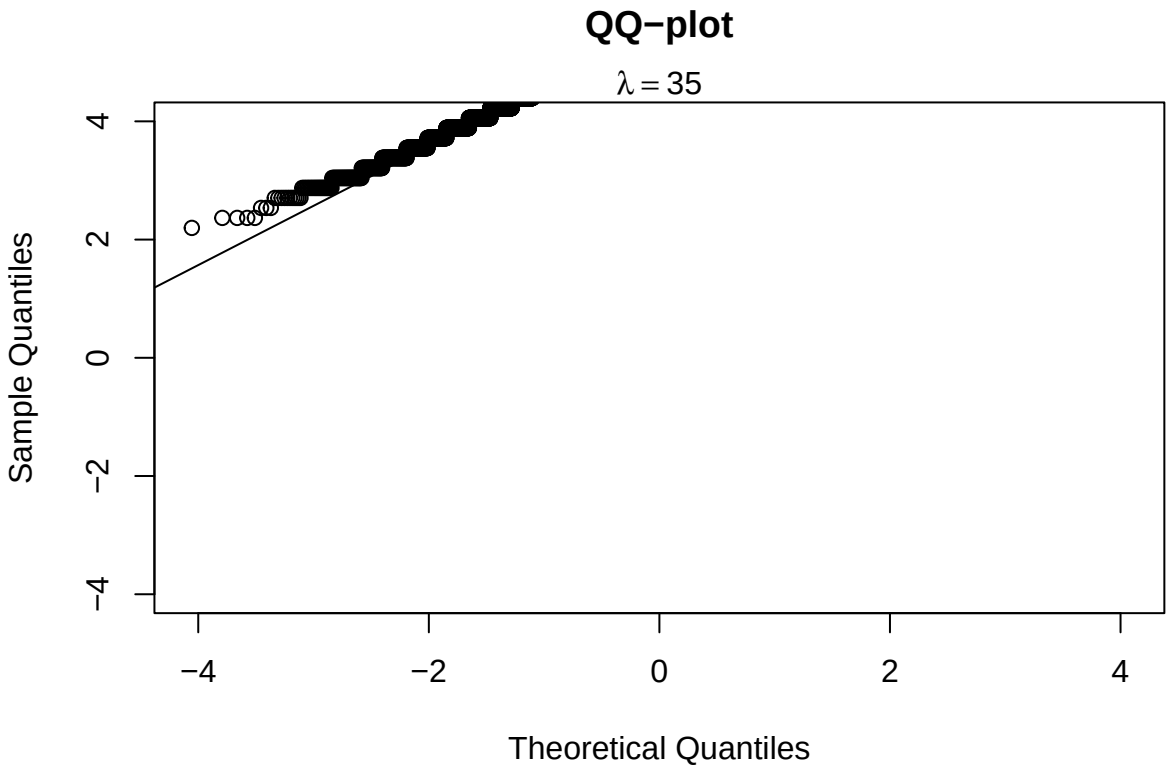

QQ−plot
λ = 25
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

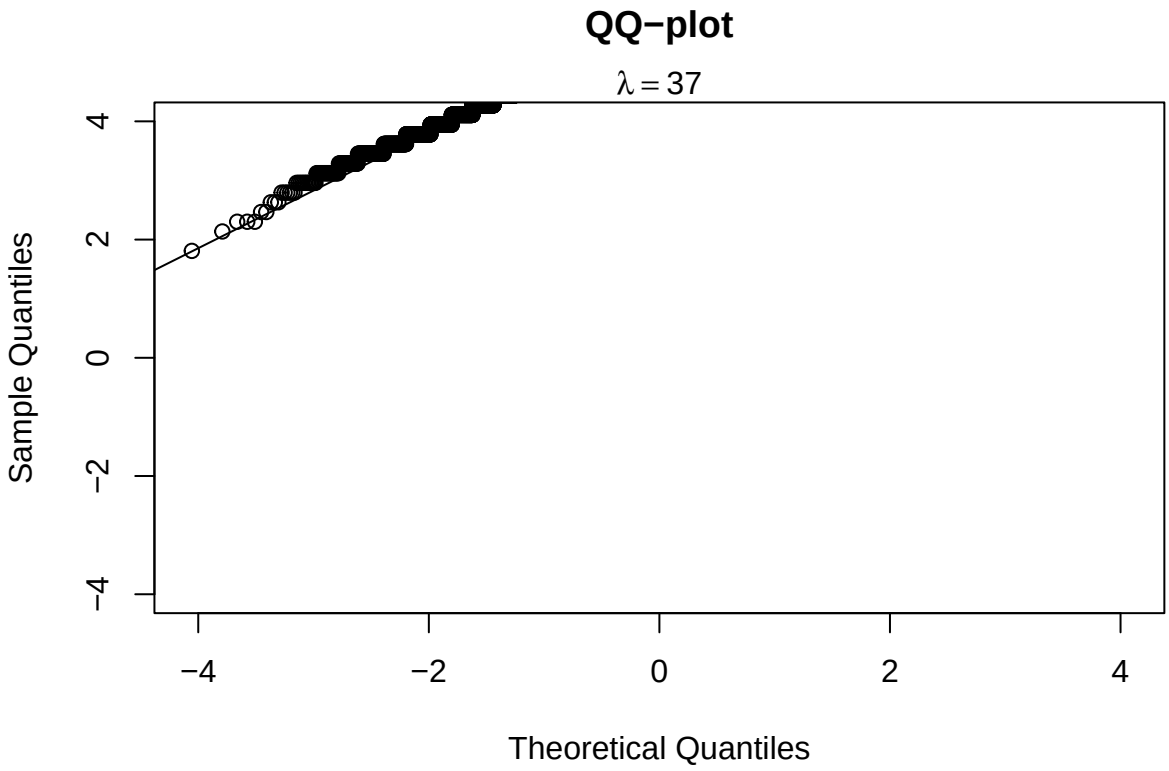

QQ−plot
λ = 27
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

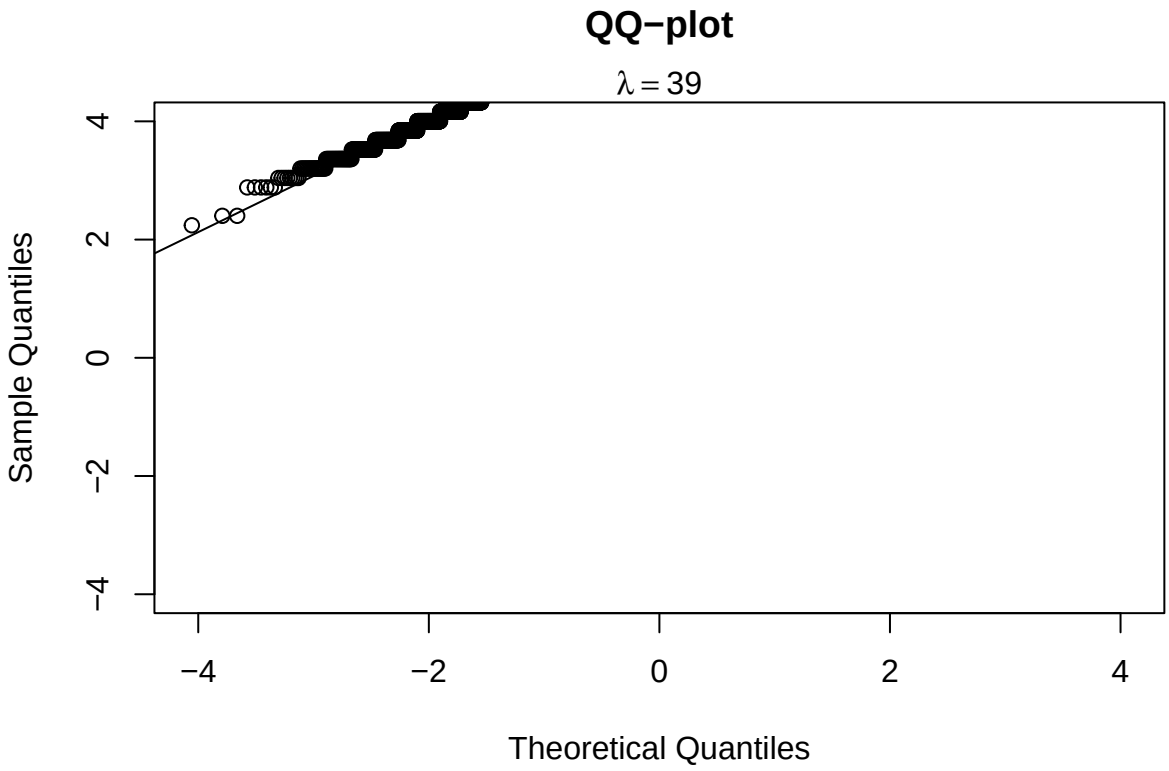

QQ−plot
λ = 29
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

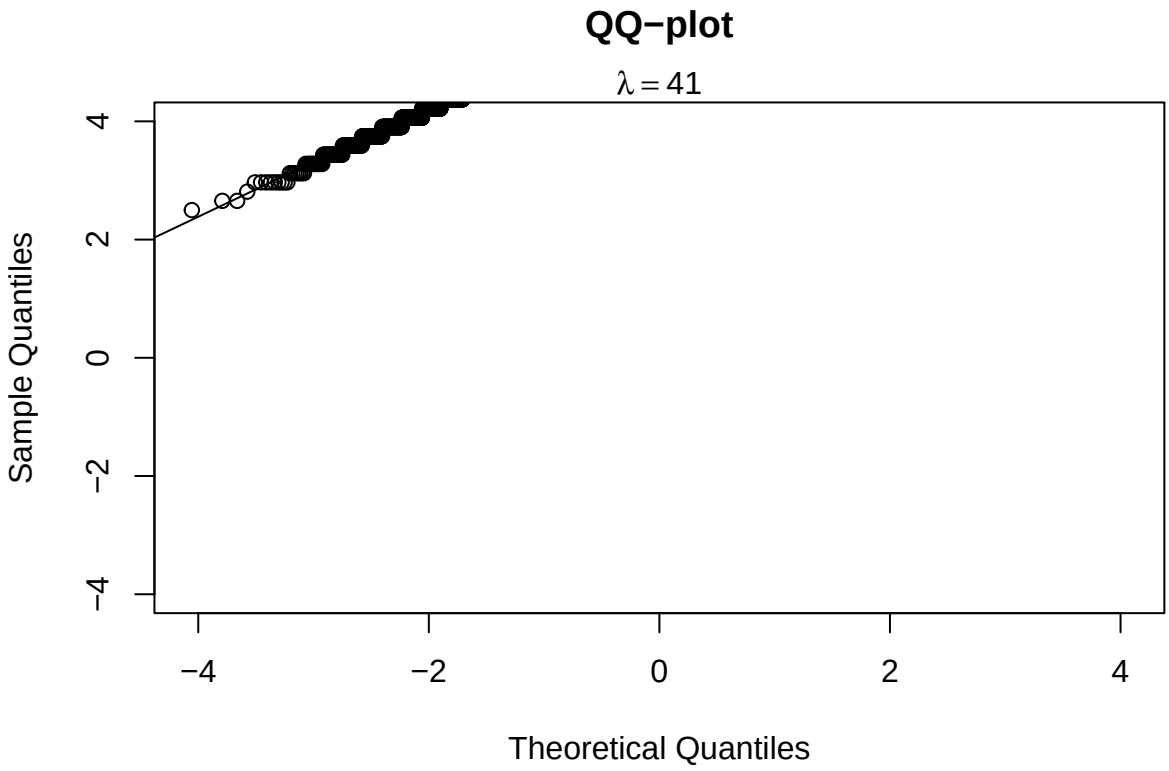

QQ−plot
λ = 31
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

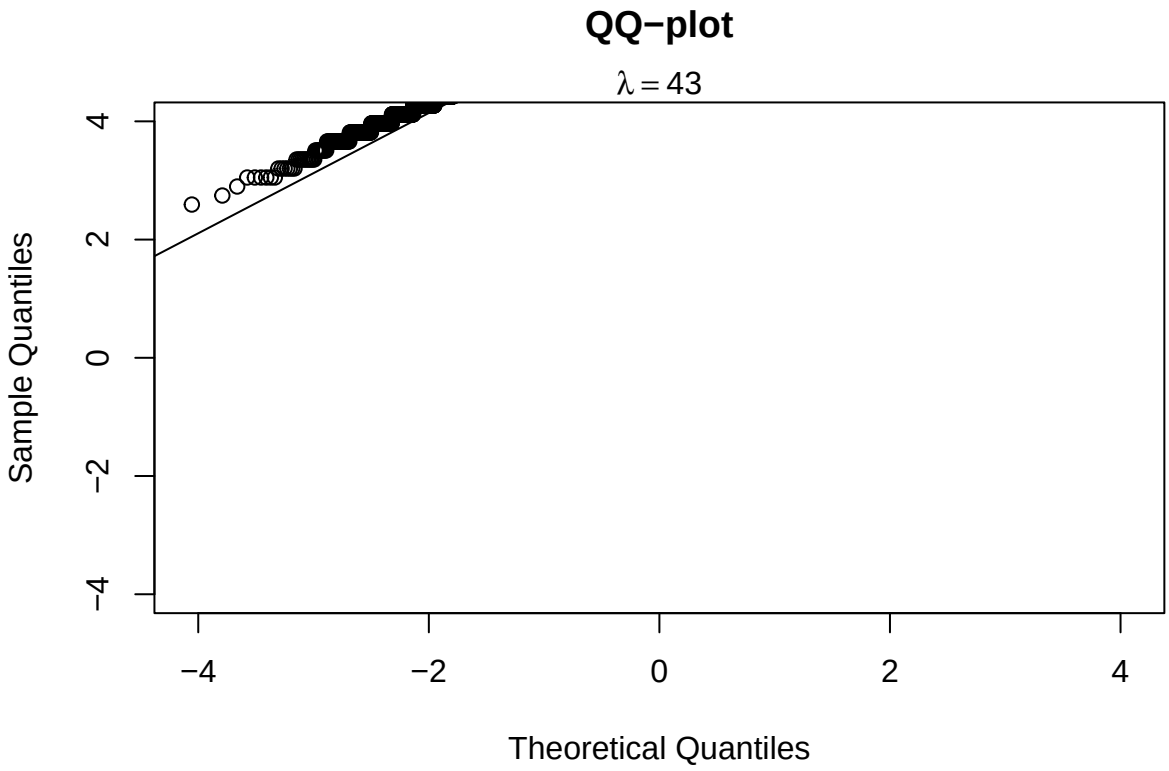

QQ−plot
λ = 33
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

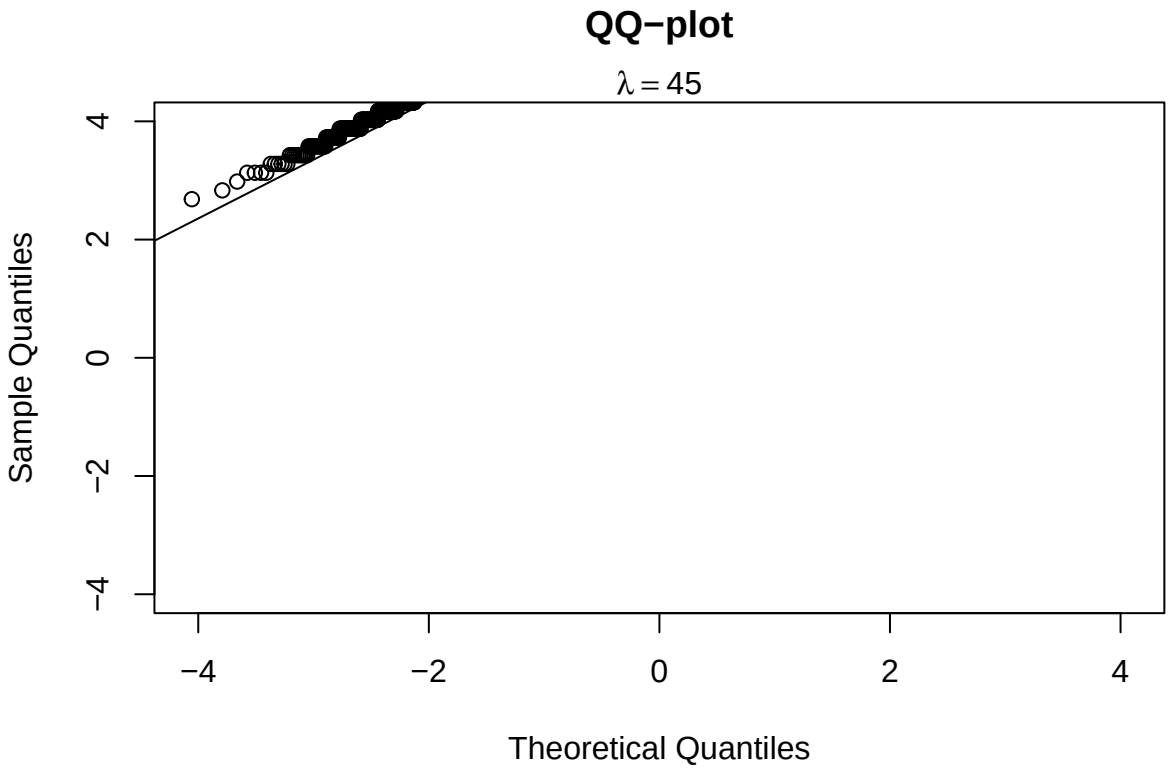

QQ−plot
λ = 35
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

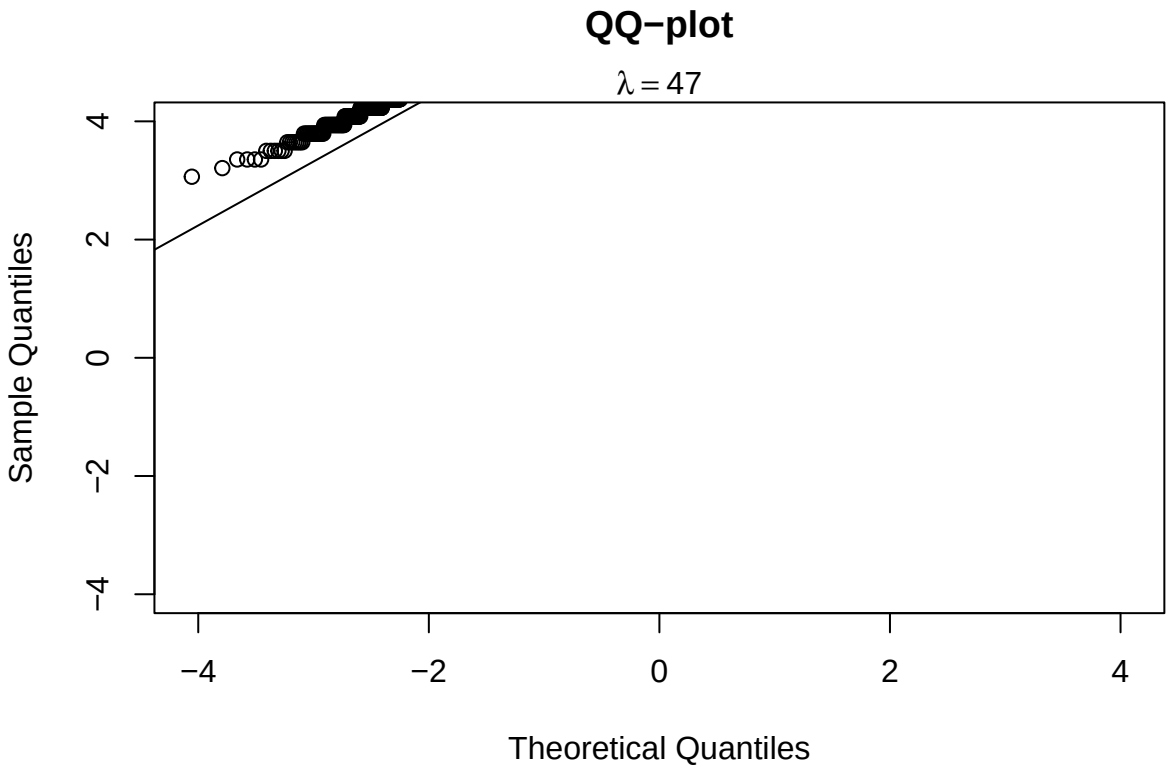

QQ−plot
λ = 37
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

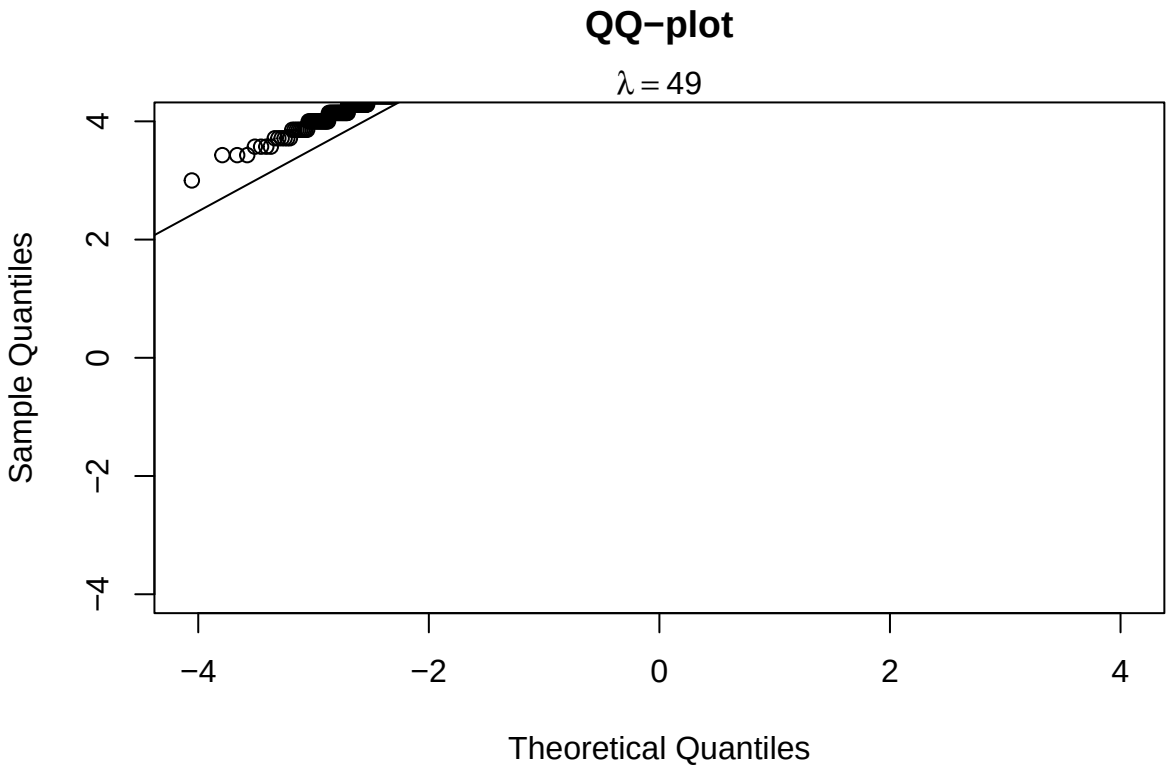

QQ−plot
λ = 39
Sample Quantiles
Theoretical Quantiles
4
2
0
−2
−4
−4
−2
0
2
4

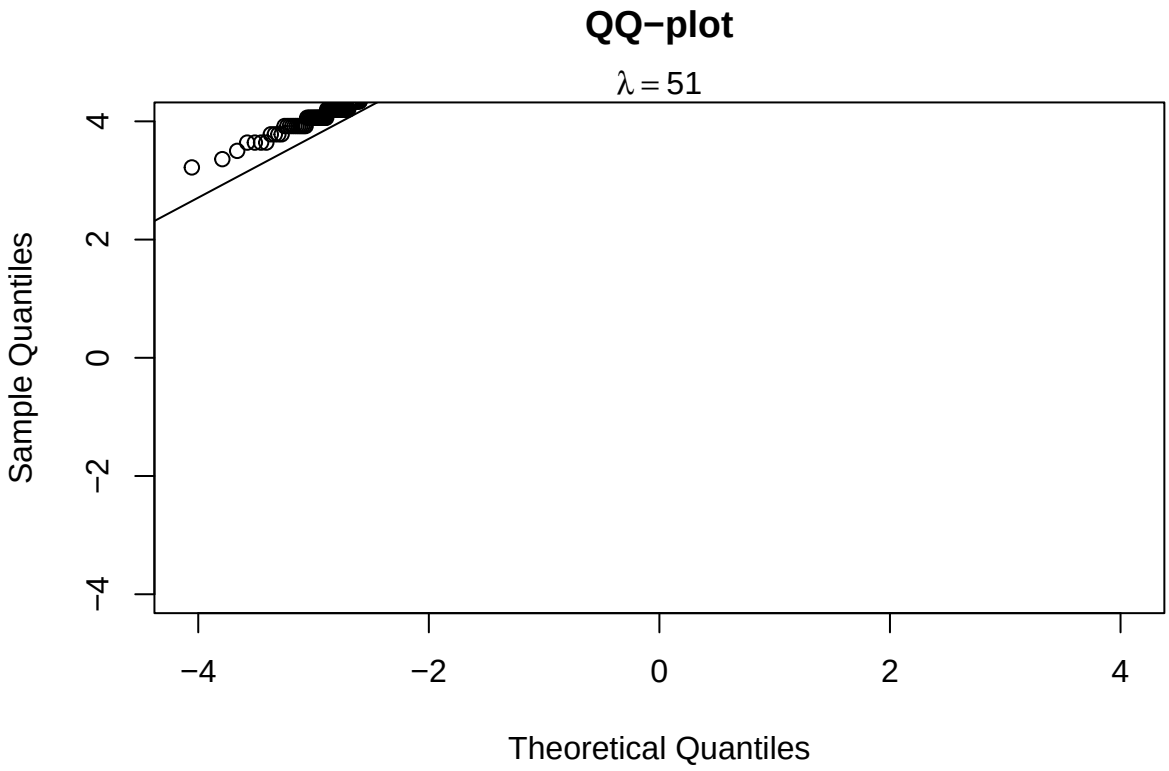

QQ−plot
λ = 41
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

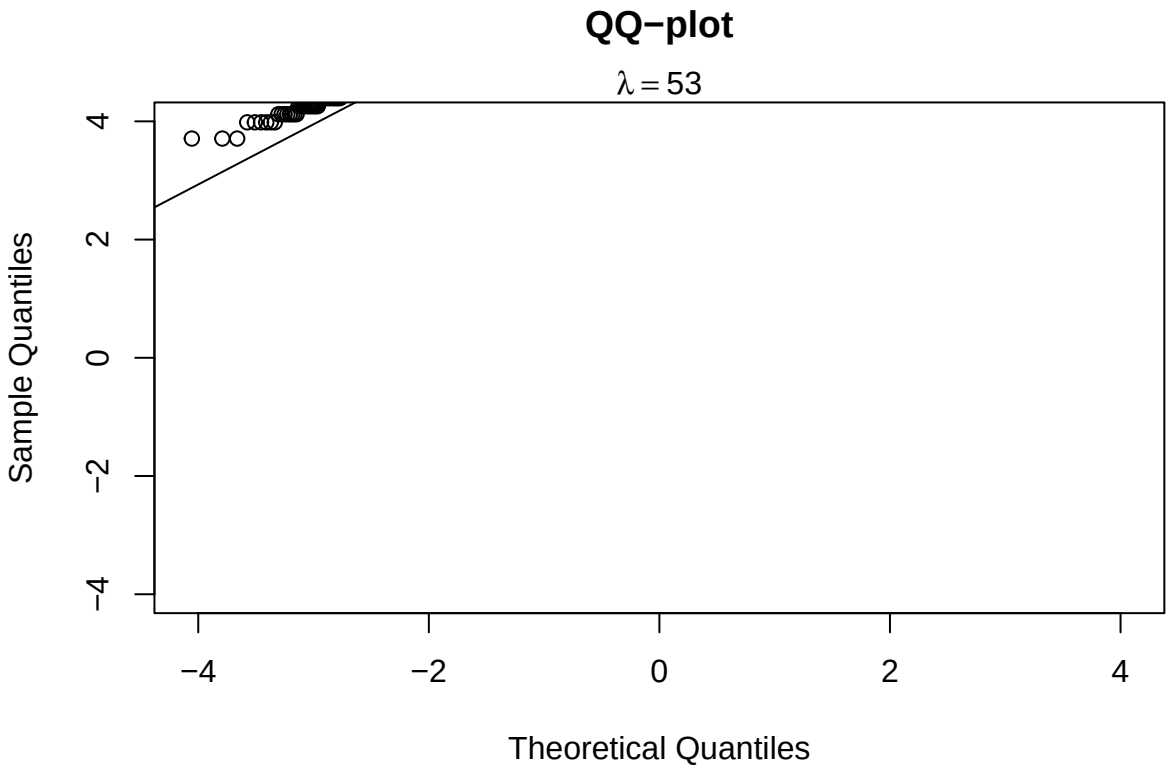

QQ−plot
λ = 43
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

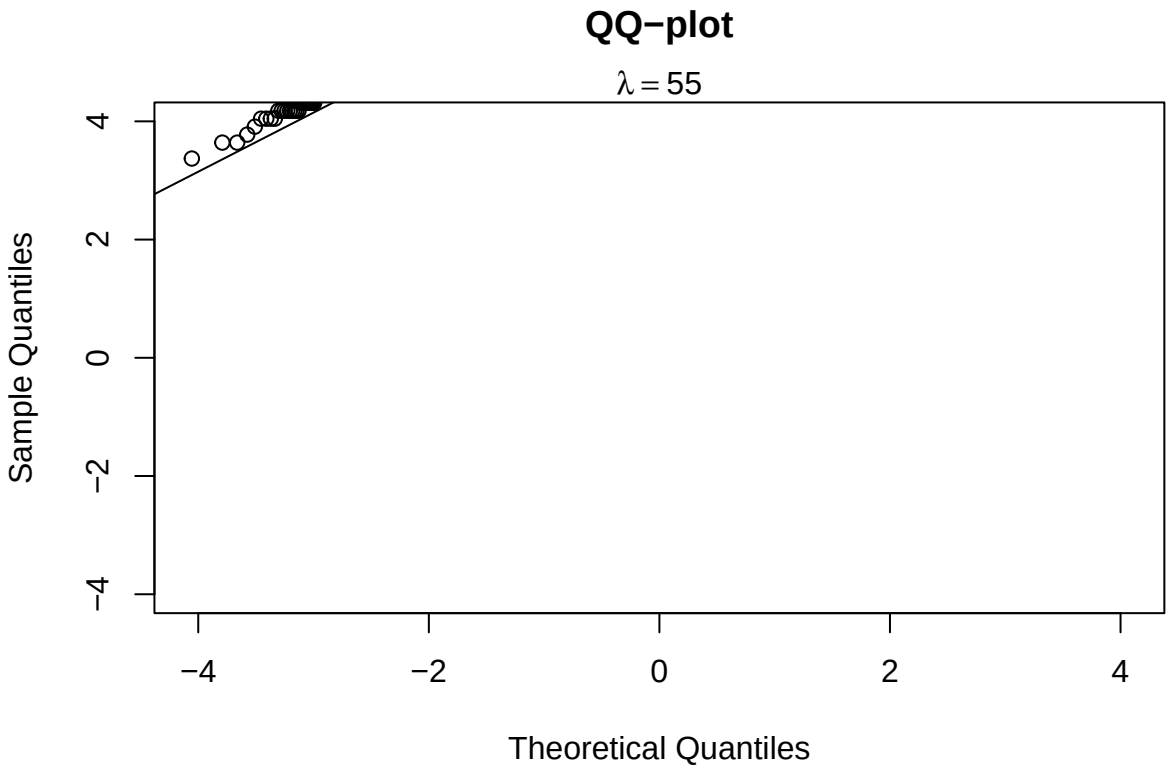

QQ−plot
λ = 45
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

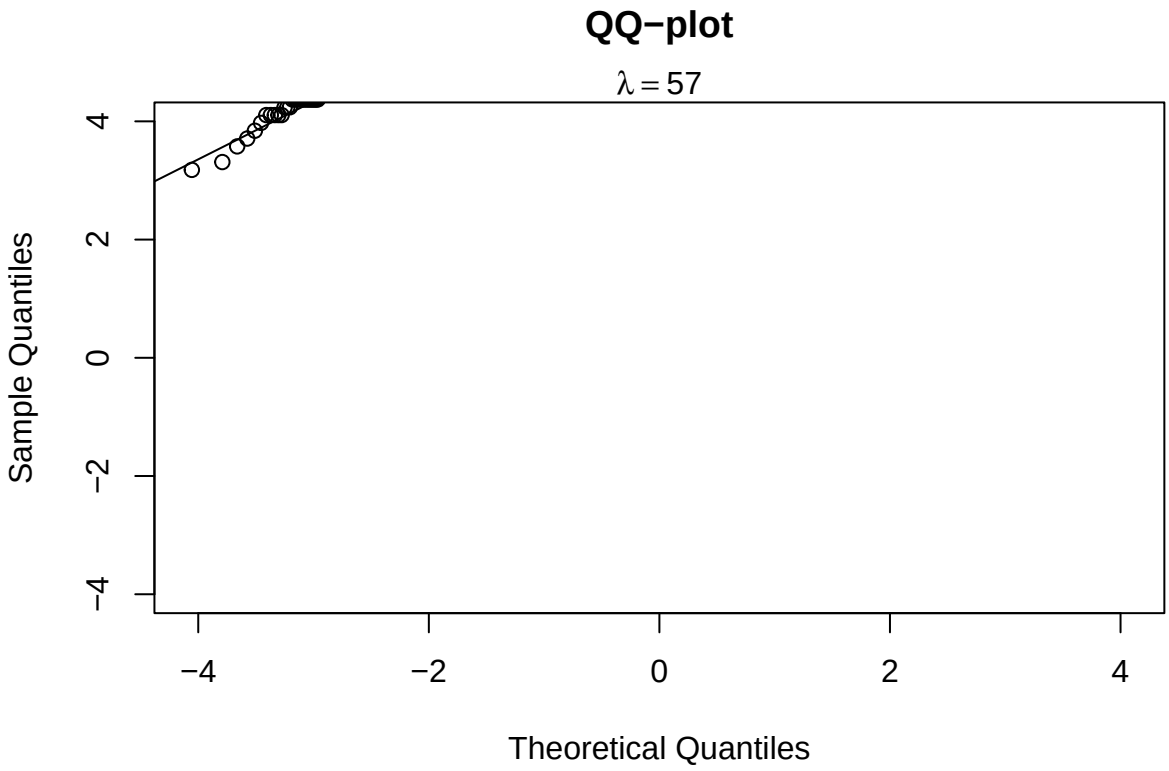

QQ−plot
λ = 47
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

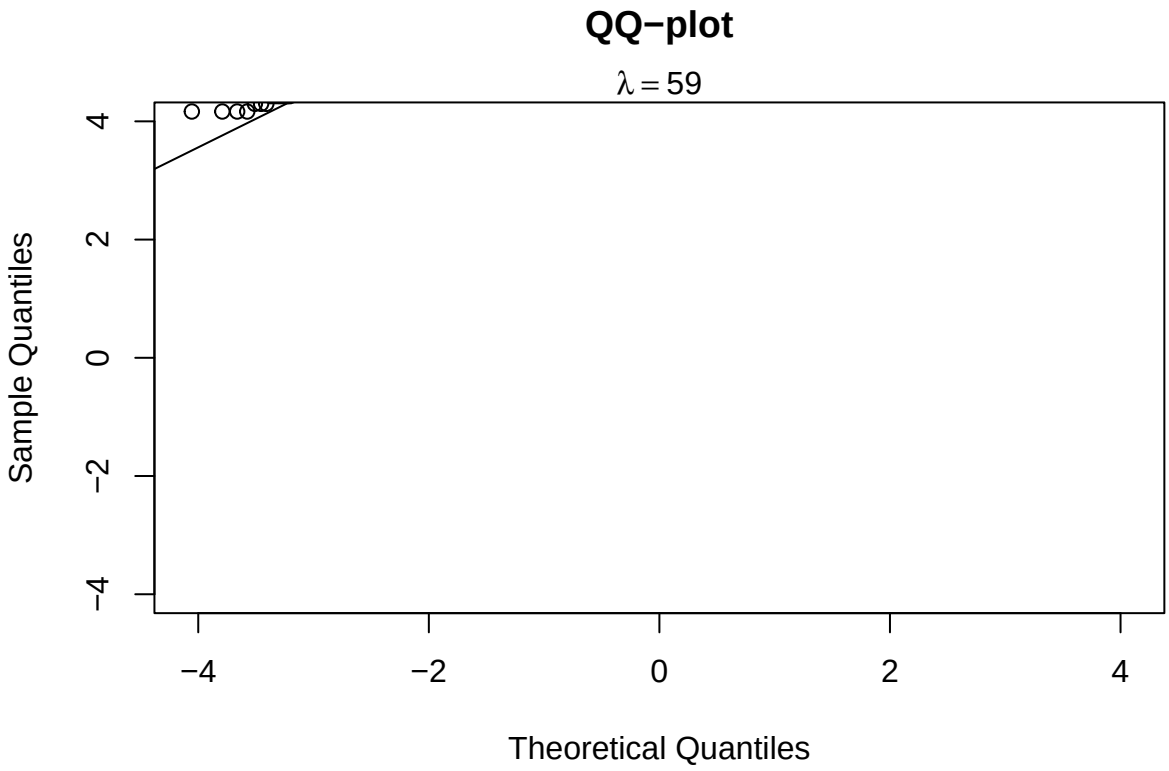

QQ−plot
λ = 49
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

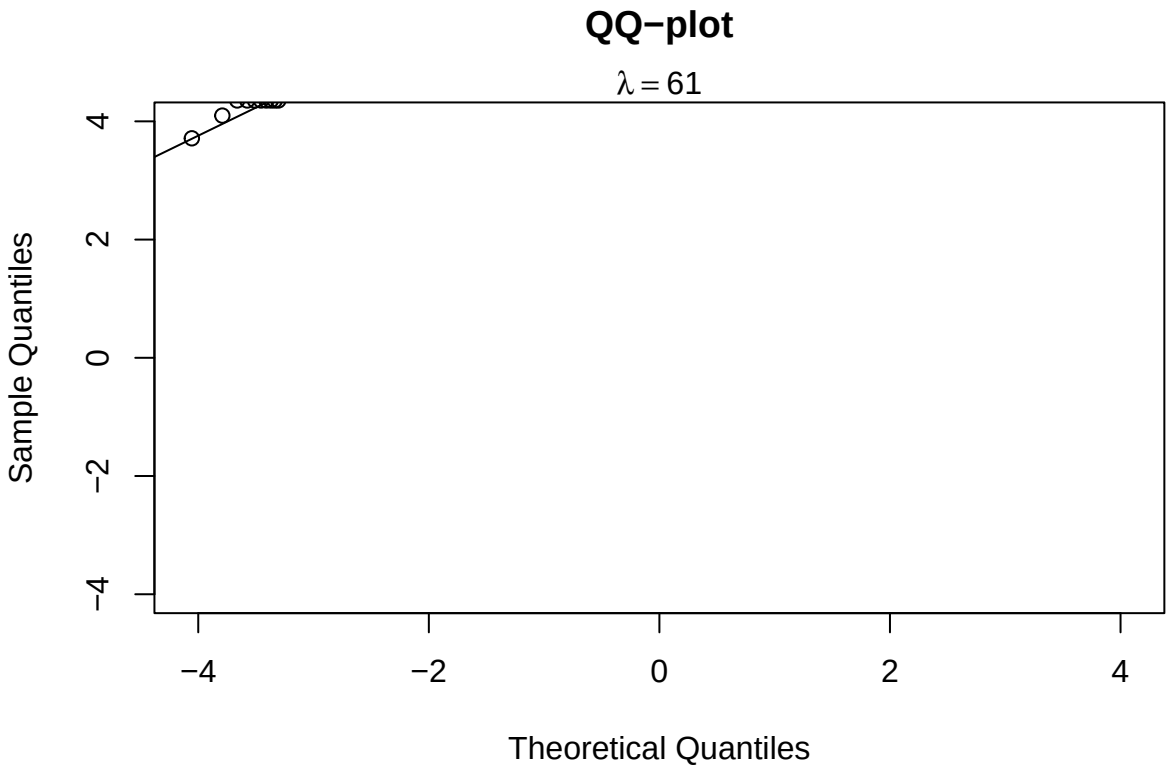

QQ−plot
λ = 51
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

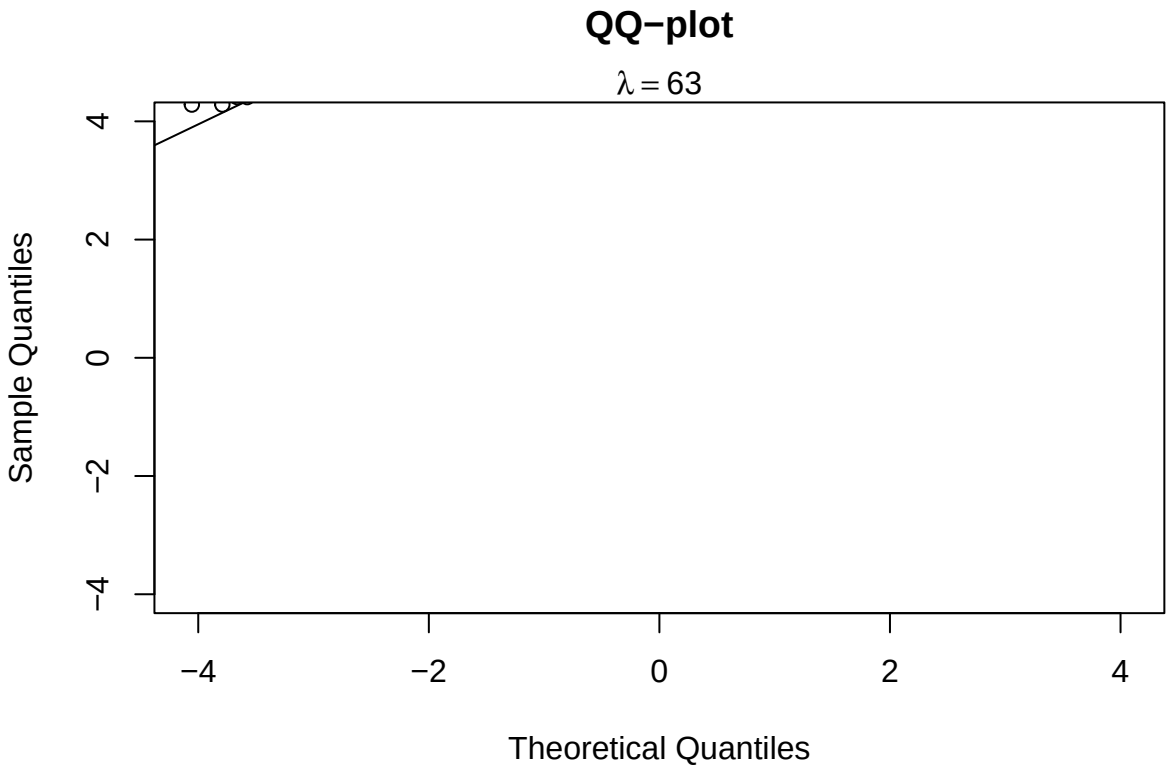

QQ−plot
λ = 53
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

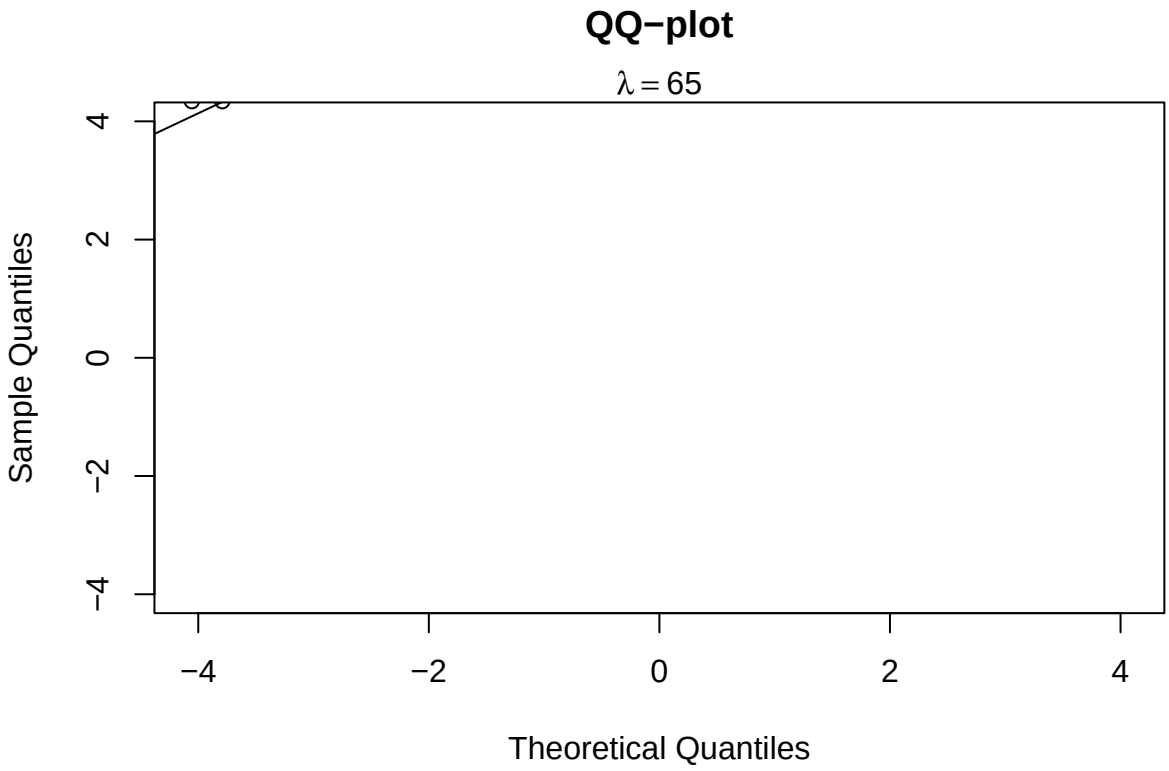

QQ−plot
λ = 55
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

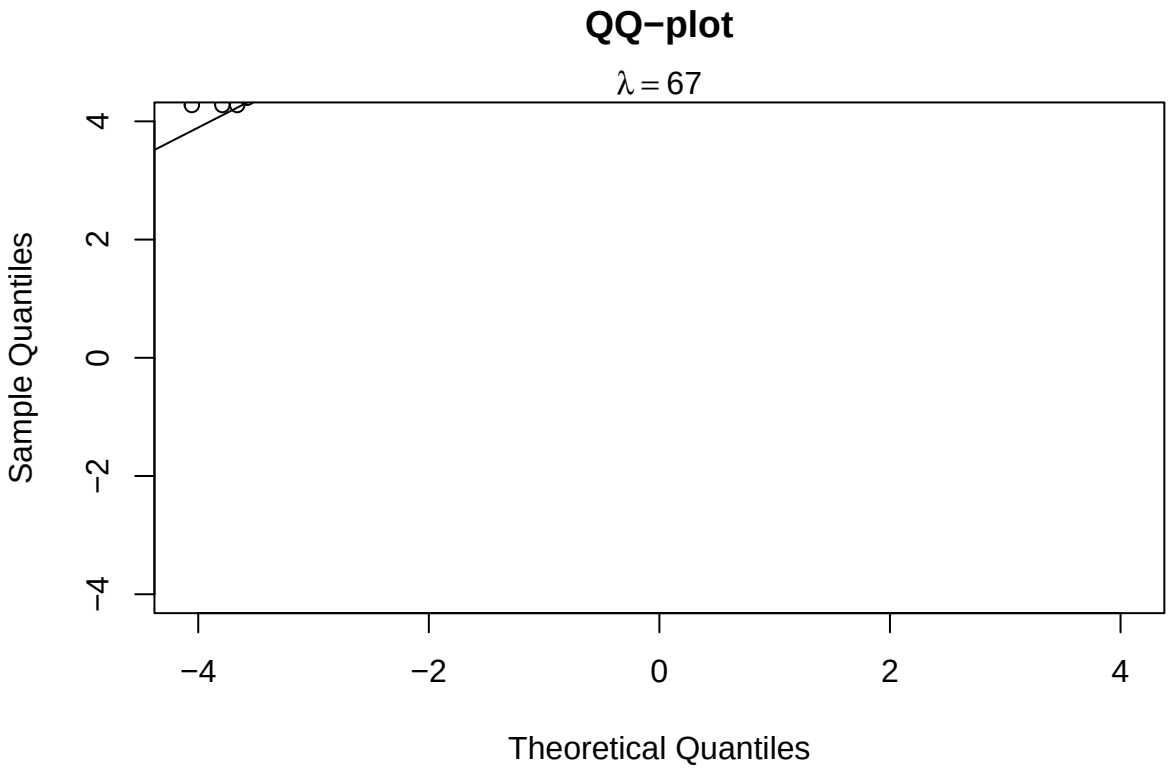

QQ−plot
λ = 57
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

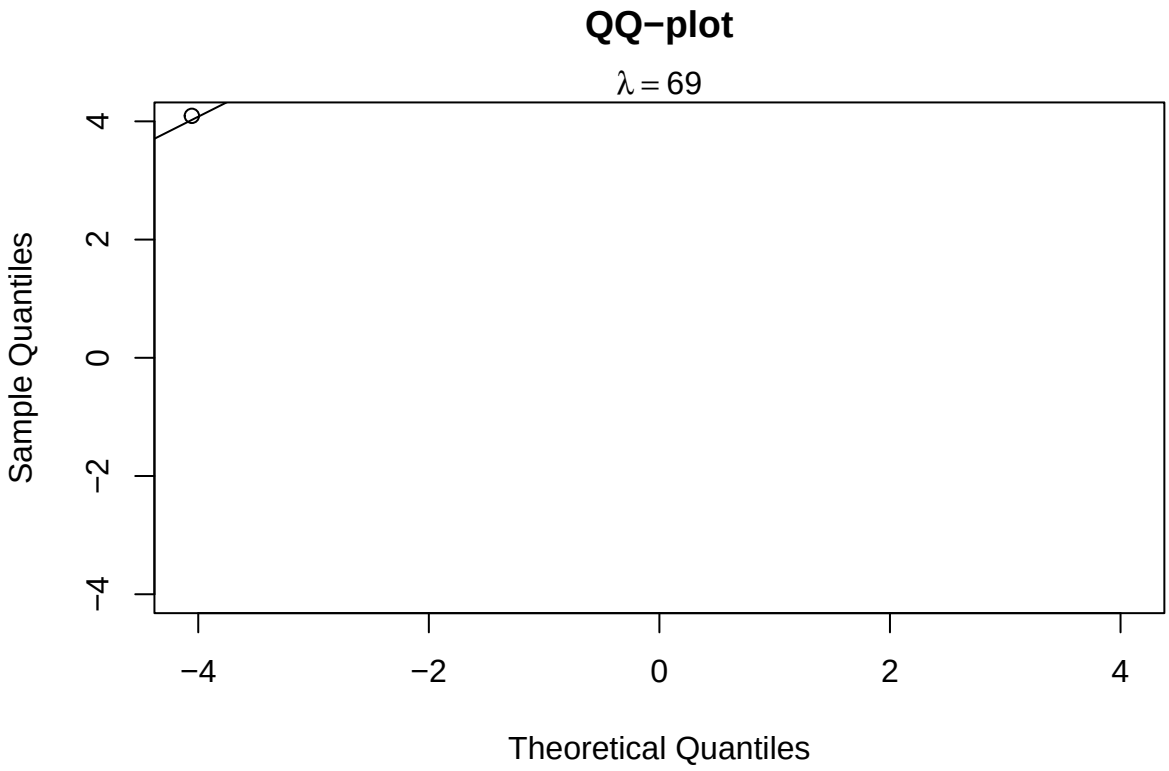

QQ−plot
λ = 59
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

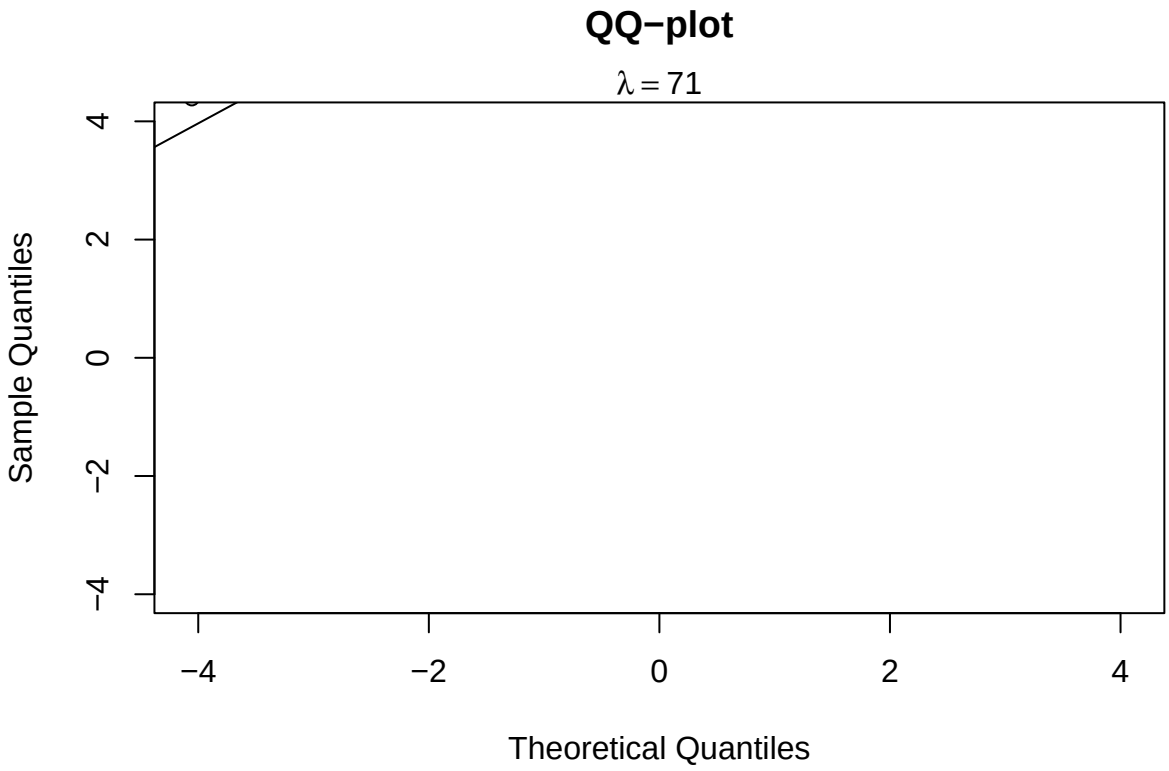

QQ−plot
λ = 61
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

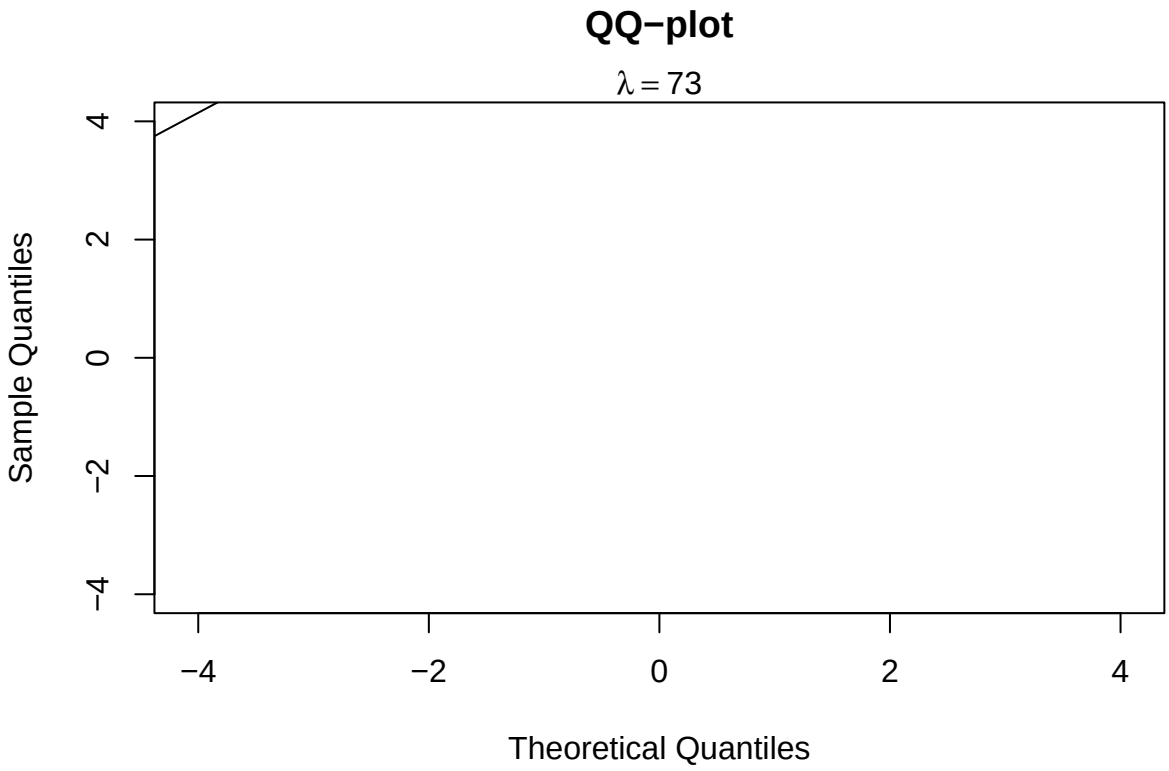

QQ−plot
λ = 63
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

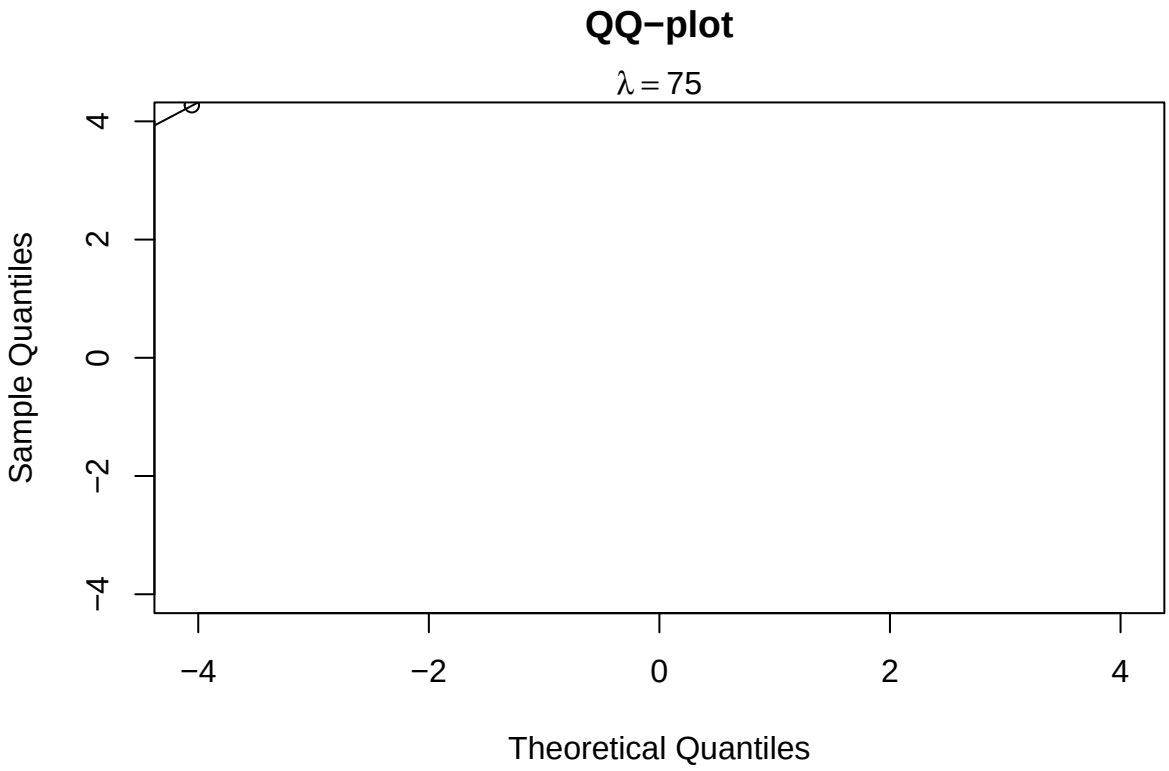

QQ−plot
λ = 65
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

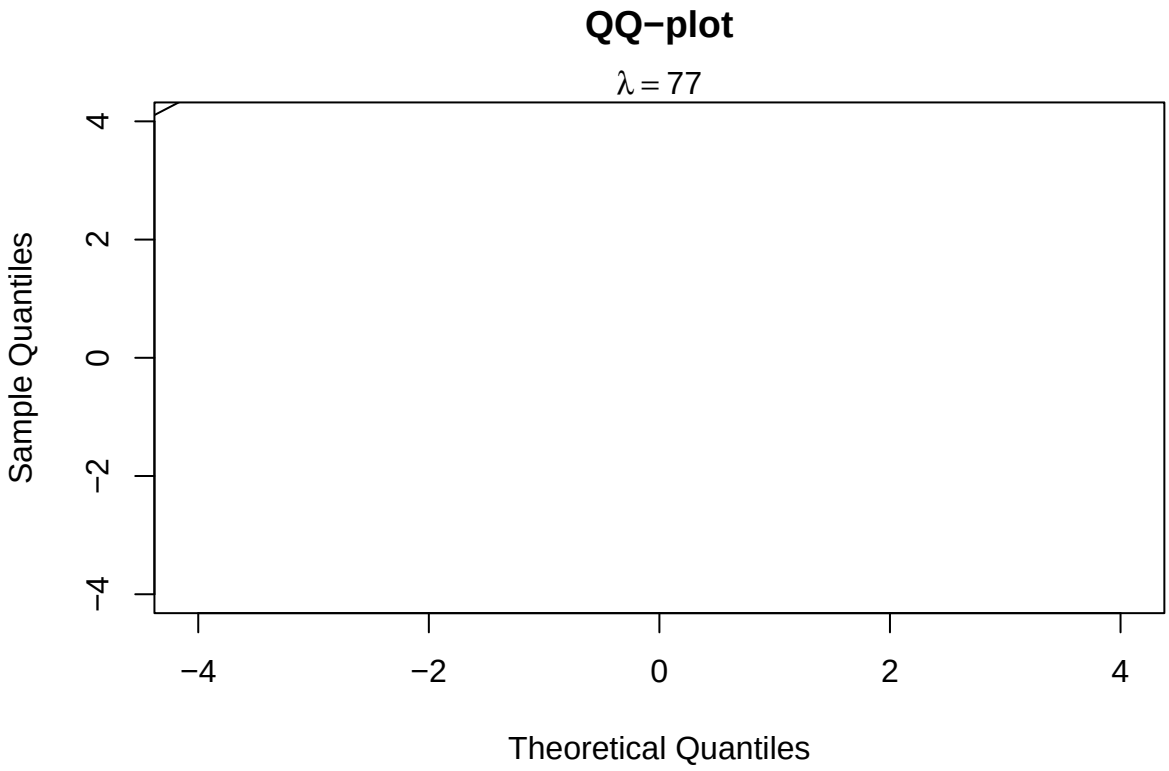

QQ−plot
λ = 67
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

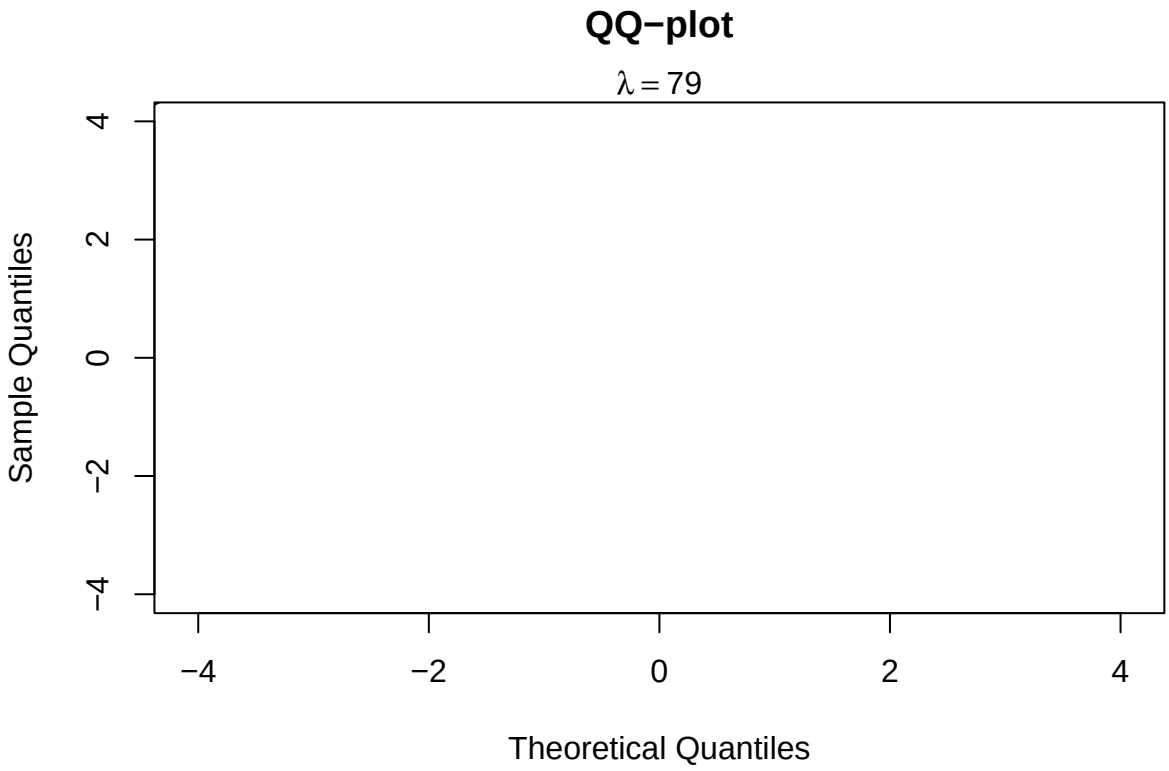

QQ−plot
λ = 69
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

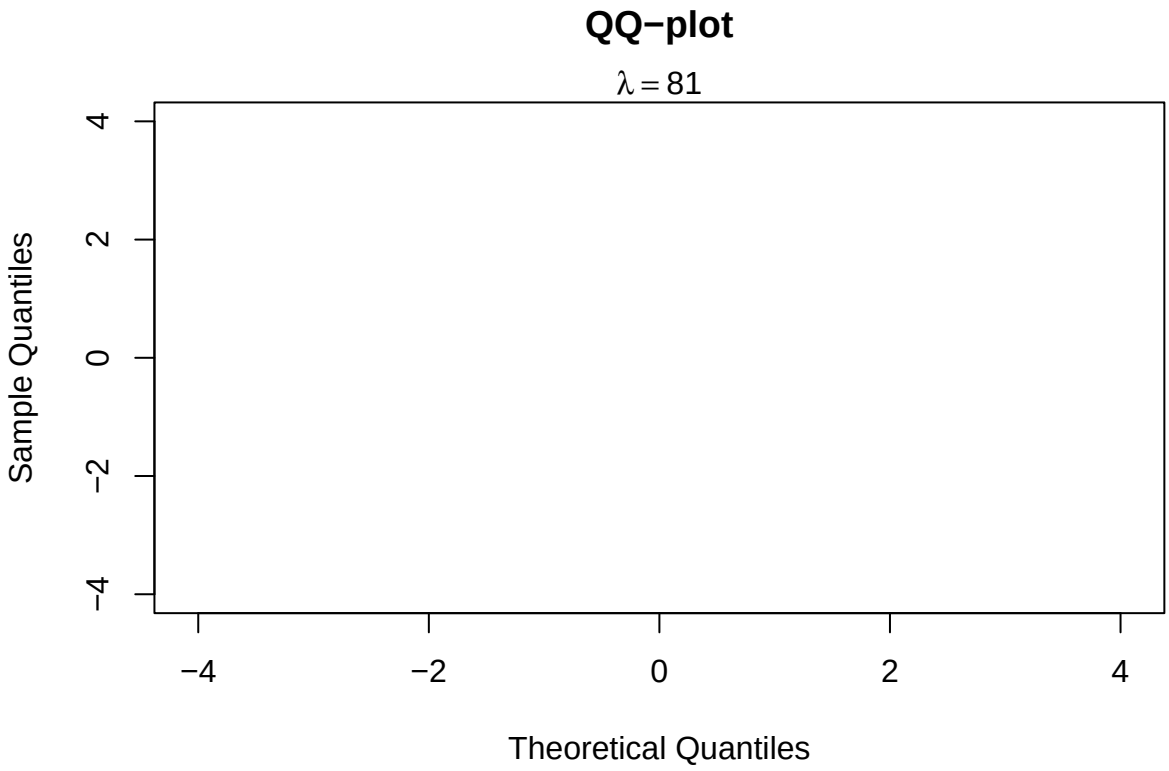

QQ−plot
λ = 71
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

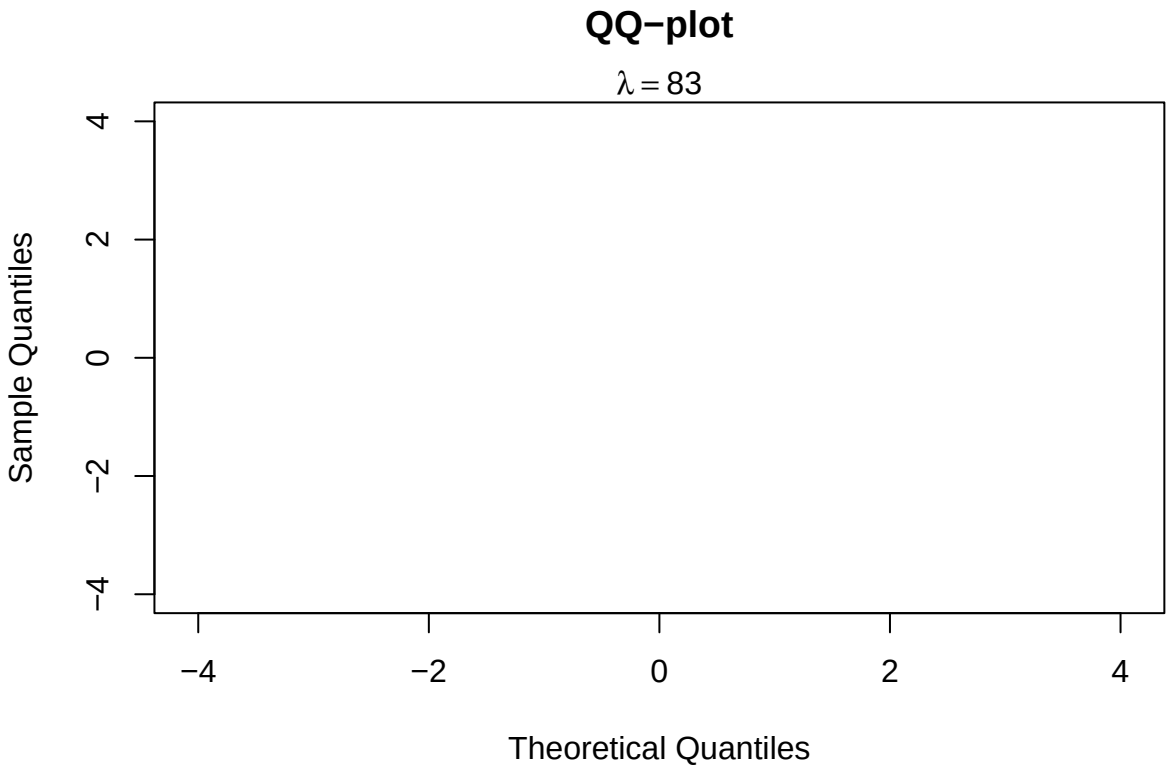

QQ−plot
λ = 73
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

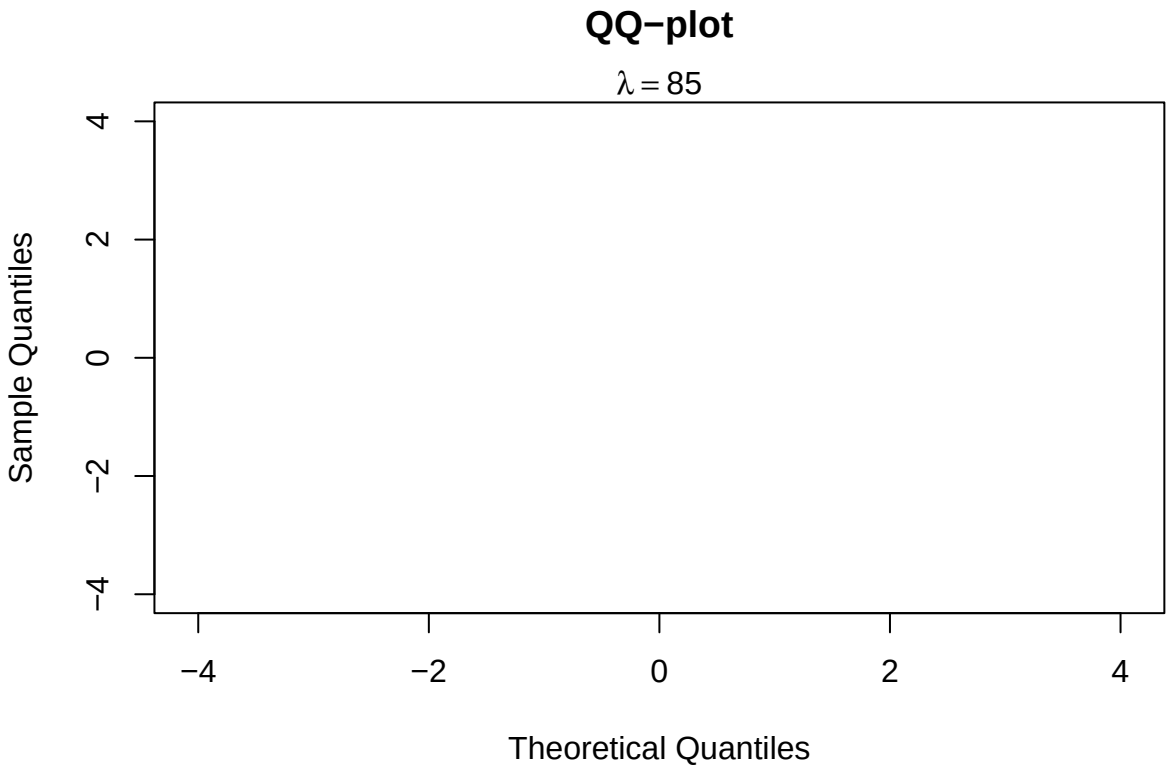

QQ−plot
λ = 75
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

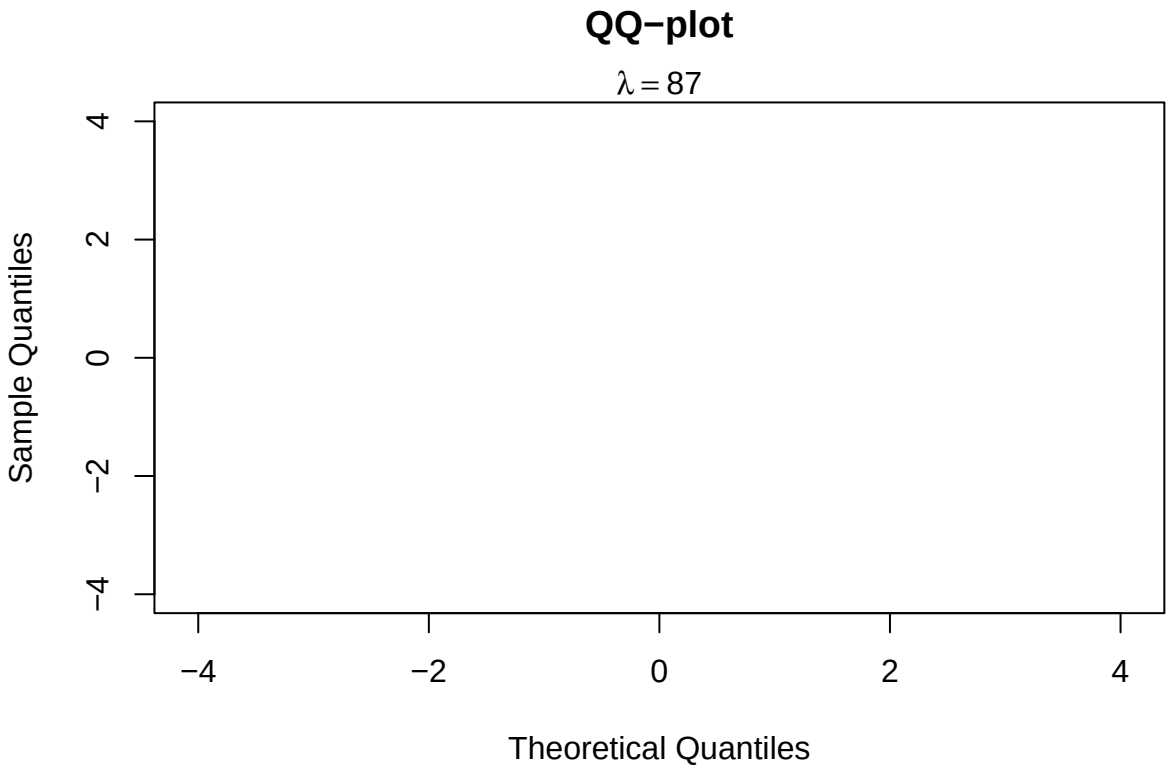

QQ−plot
λ = 77
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

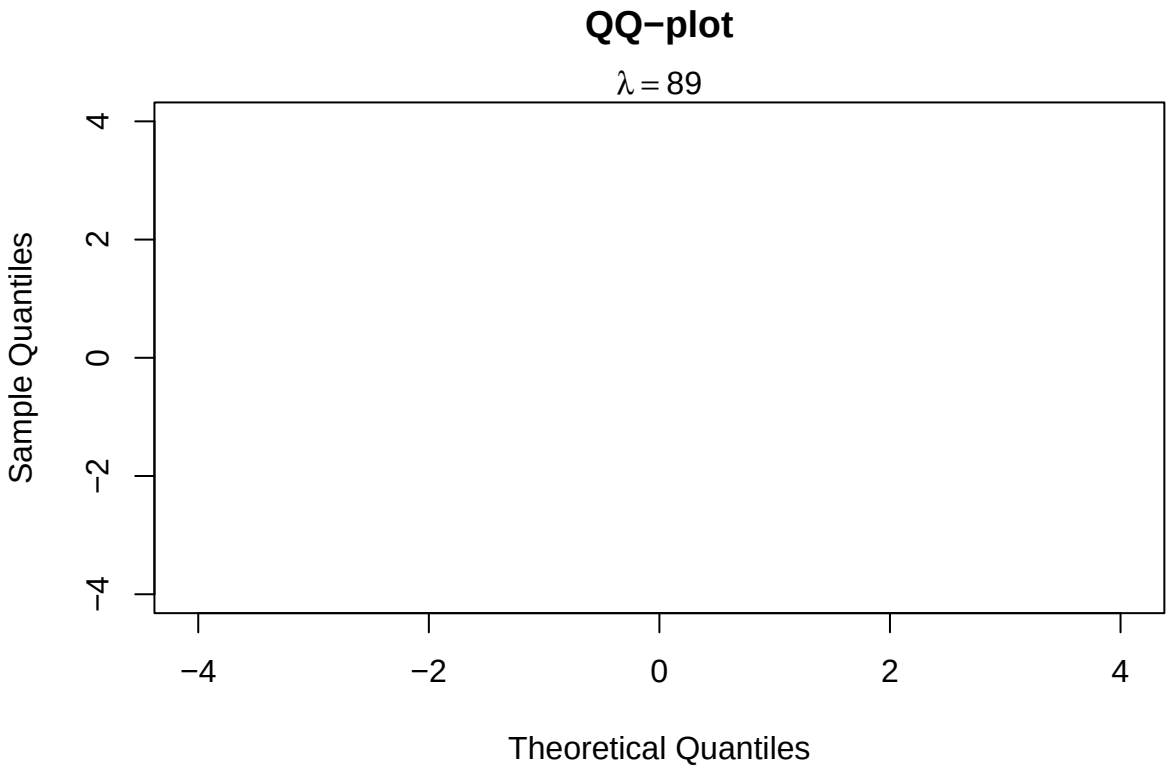

QQ−plot
λ = 79
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

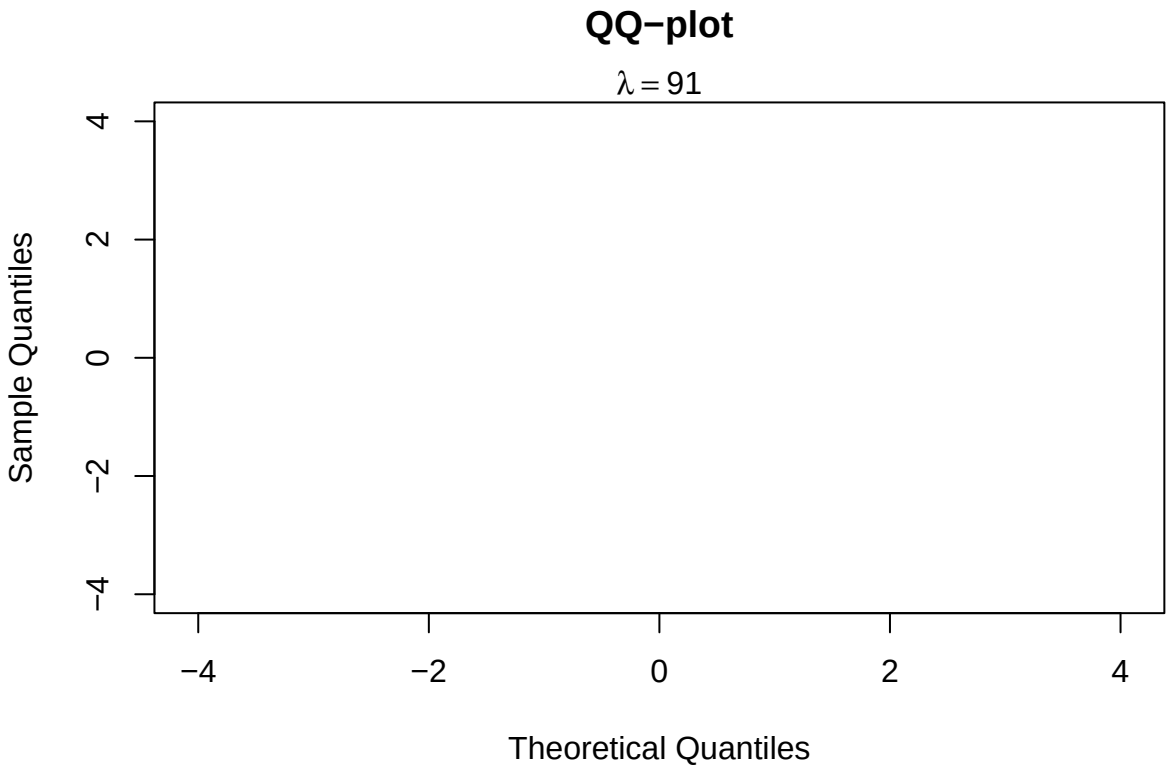

QQ−plot
λ = 81
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

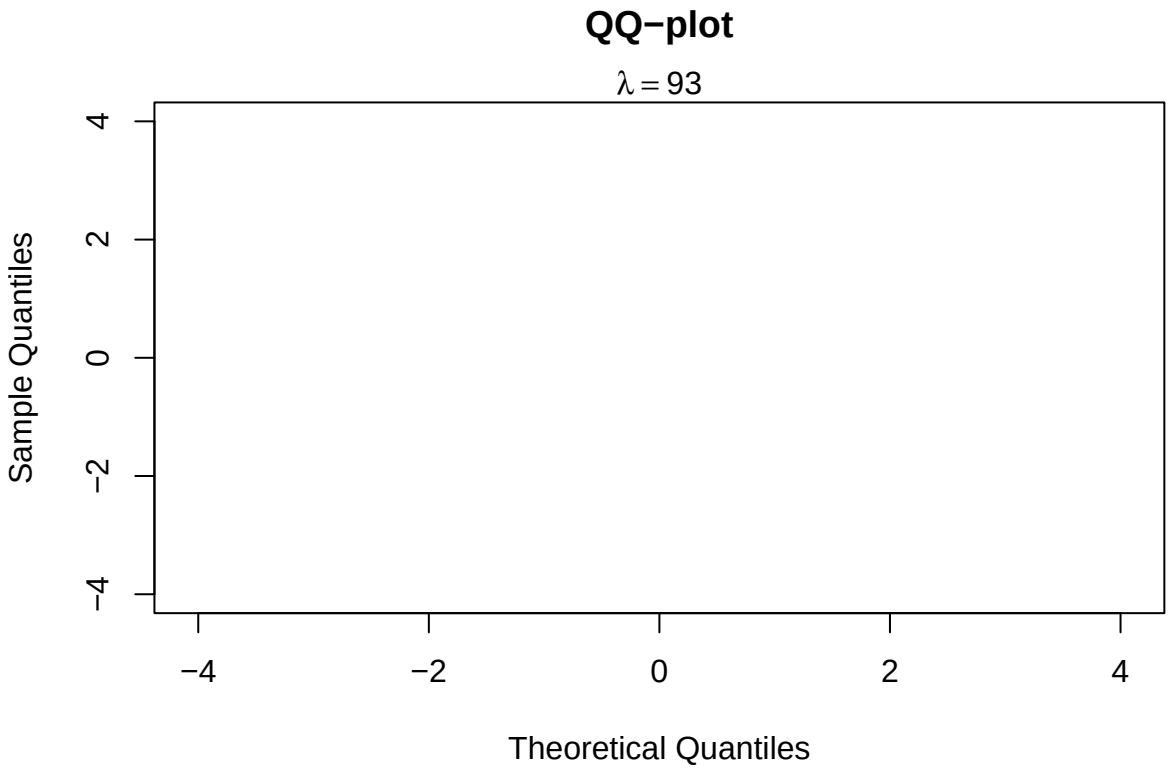

QQ−plot
λ = 83
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

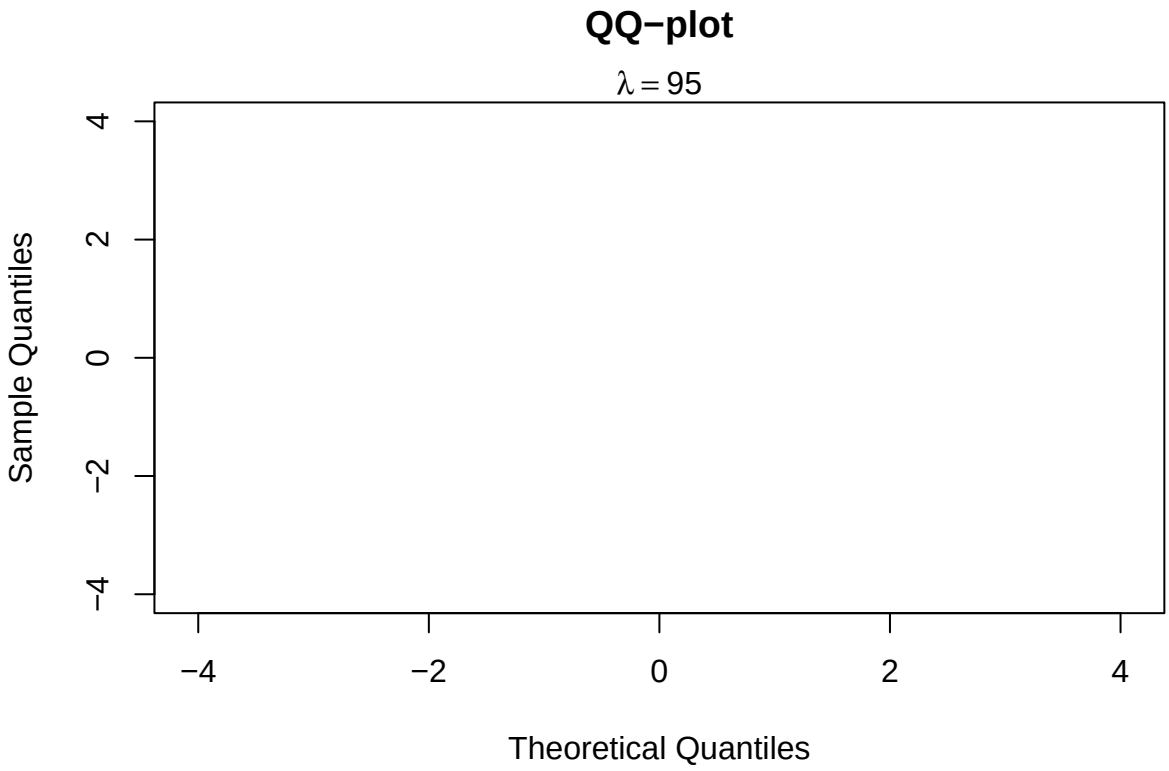

QQ−plot
λ = 85
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

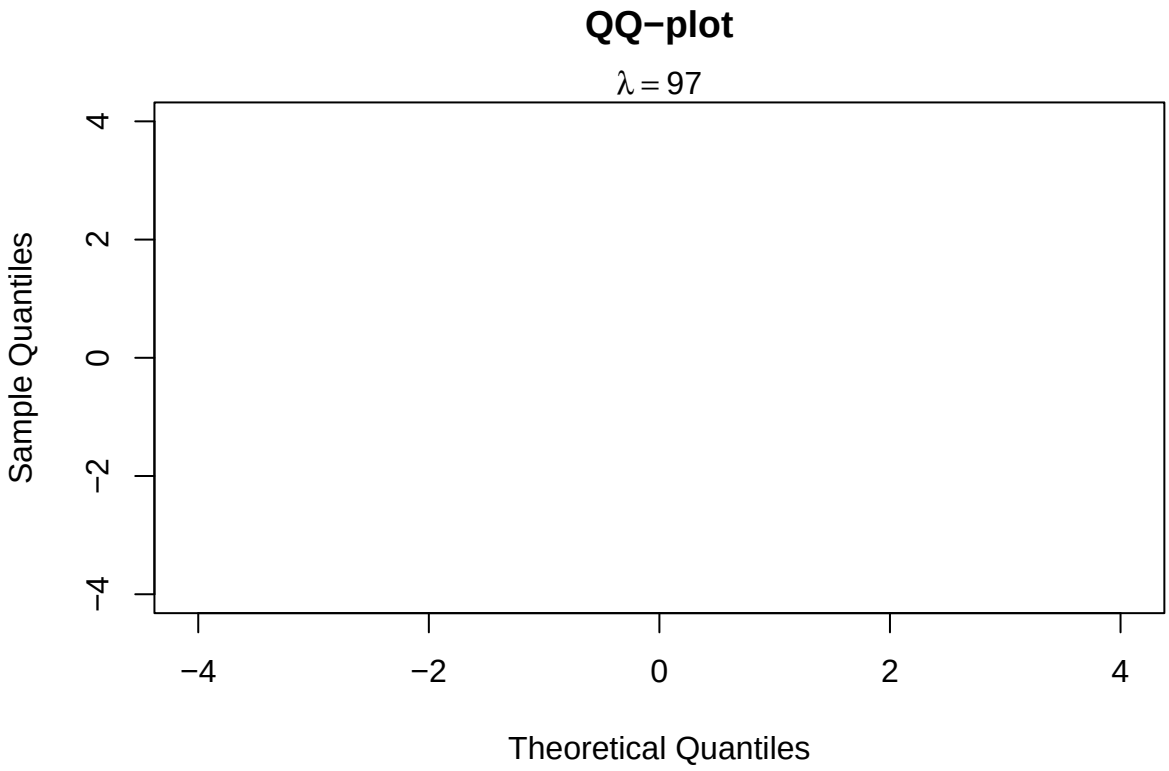

QQ−plot
λ = 87
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4

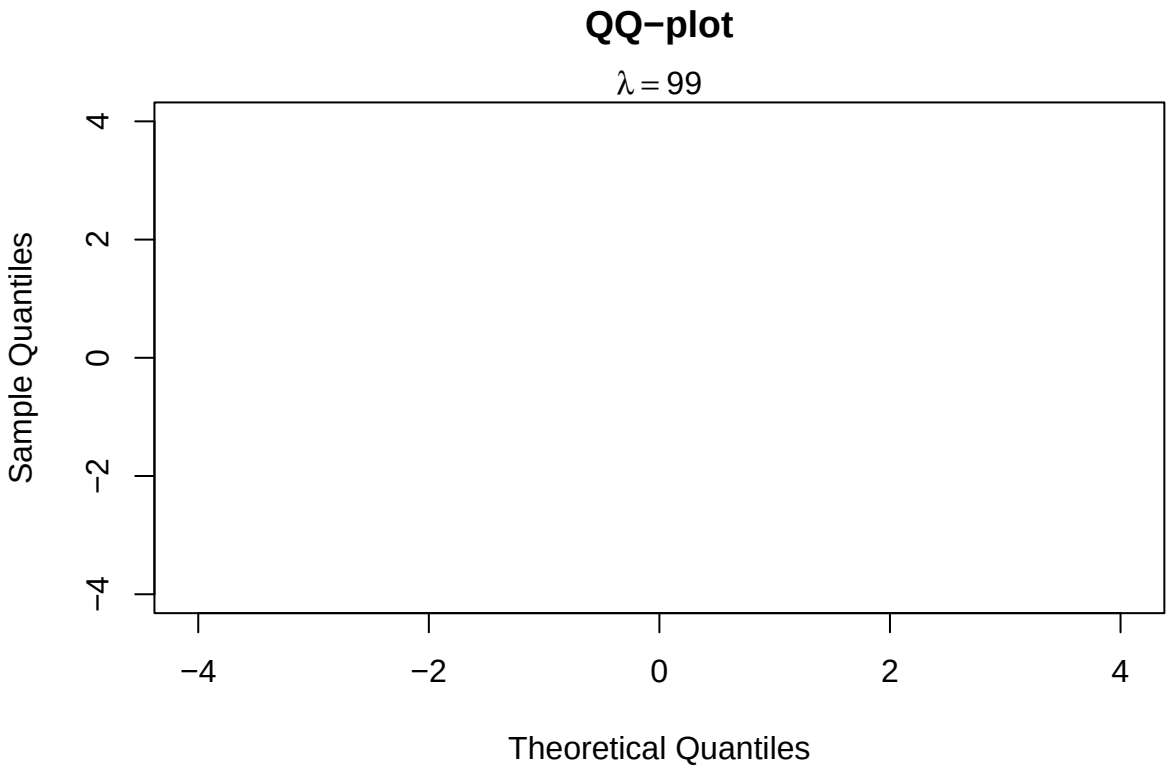

QQ−plot
λ = 89
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4


QQ−plot
λ = 91
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4


QQ−plot
λ = 93
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4


QQ−plot
λ = 95
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4


QQ−plot
λ = 97
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4



(a) Answer: Executing the code and observing the distribution of Z as a increases allows us to see how the approximation to a standard normal distribution improves. Initially, for smaller a values, the QQ plot may deviate more from a straight line, indicating less conformity to a normal distribution. However, as a increases, the points on the QQ plot tend to align more closely with the diagonal line, suggesting better approximation to a standard normal distribution.

(b) Answer: To determine how large a must be before seeing a reasonably straight line in the QQ plot, one needs to visually inspect the plots. Typically, as a increases, the distribution of Z becomes closer to a standard normal distribution, resulting in a straighter line on the QQ plot. However, the exact threshold for observing a reasonably straight line may vary depending on the specific data and the desired level of approximation.

Problem 7:

Simulate 10000 realizations of a Poisson process with rate 2.5 on the interval [0,2]

(a) In each case, count the number of points in the subintervals [0,1] and [1,2].

(b) Are the counts in part (a) reasonably approximated by Poisson distributions with rate 2.5?

(c) Using an appropriate scatterplot, make a judgment as to whether it is reasonable to assume that the number of points in the interval [0,1] is independent of the number in [1,2]. (In order for the scatterplot to be useful, it will be necessary to use the jitter() function.)

```r
# Simulate 10000 realizations of a Poisson process with rate 2.5 on the interval [0,2]
N <- rpois(10000, lambda = 2.5 * 2)
pts <- list()

# Generate random points within the interval [0,2]
for (i in 1:10000)
  pts[[i]] <- runif(N[i], 0, 2)

# (a) Count the number of points in the subintervals [0,1] and [1,2]
count1 <- numeric(10000)
count2 <- numeric(10000)

for (i in 1:10000) {
  count1[i] <- sum(pts[[i]] < 1)
  count2[i] <- sum(pts[[i]] > 1)
}

# (b) Check if the counts are reasonably approximated by Poisson distributions with rate 2.5
# Calculate the observed counts in each subinterval
table(count1)
```

```
## count1
##    0    1    2    3    4    5    6    7    8    9   10
##  858 2038 2548 2100 1312  688  321   97   28    6    4
```

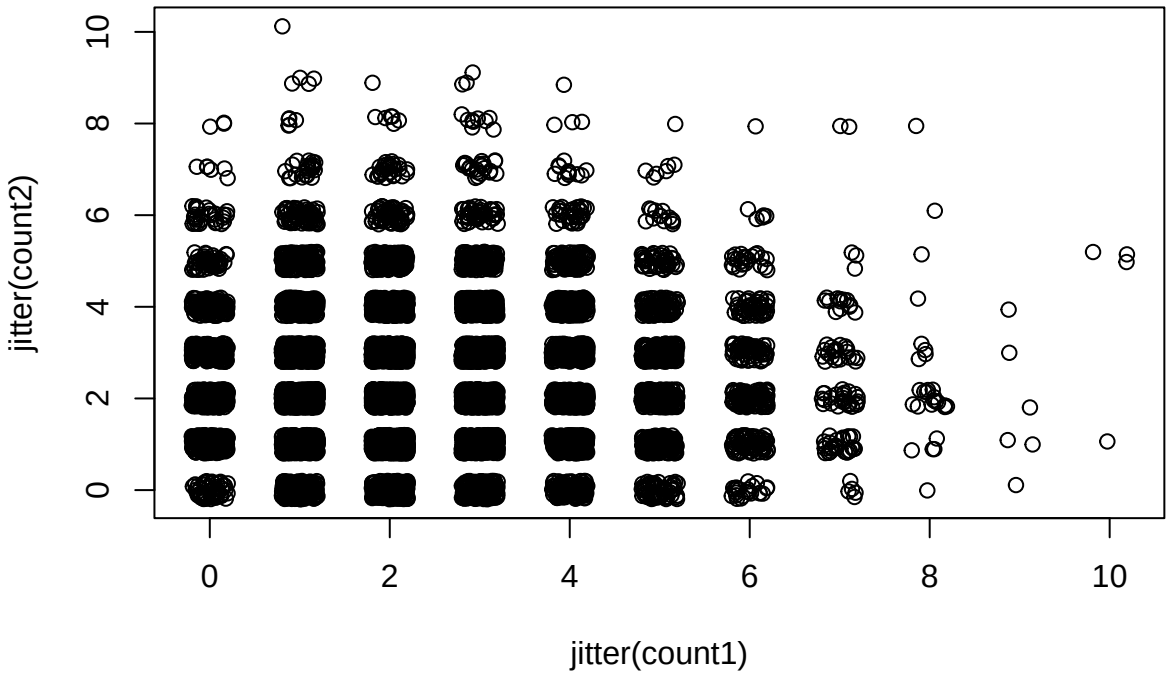
```r
table(count2)
```

```
## count2
##    0    1    2    3    4    5    6    7    8    9   10
##  818 2080 2598 2119 1326  680  243   95   31    9    1
```

```r
# Calculate the expected counts from a Poisson distribution with rate 2.5
round(10000 * dpois(0:12, 2.5))
```

```
##  [1]  821 2052 2565 2138 1336  668  278   99   31    9    2    0    0
```

```r
# (c) Visualize the independence assumption between counts in [0,1] and [1,2]
# Create a scatterplot with jitter for visualization
plot(jitter(count1), jitter(count2))
```


jitter(count2)
0
2
4
6
8
10
0
2
4
6
8
10
jitter(count1)