Líkindareikningur og Tölfræði STÆ203G

Tölvuverkefni 3

Egill Ian Guðmundsson, 260693-2639

Verkefni 1

Sólarhringsúrkoma hversa dags er mæld kl 9:00 á morgnana. Mælingar á mestu sólarhringsúrkomu (mm) á Fagurhólsmýri innan hvers árs fyrir árin 1924 til 2007 (n = 84) er að finna í gagnaskránni `precipitation_fagurhm.txt`. Notið þessi gögn til að reikna stærðirnar sem eru gefnar fyrir neðan.

1.a. Reiknið meðaltal, miðgildi, dreifni, staðalfrávik, fyrsta fjórðungsmark, þriðja fjórðungsmark og fjórðungsbil fyrir úrtakið og setjið saman í eina töflu.

Svar: Við keyrum `precipitationStats.r` forritið sem fylgir með í viðauka til að fá eftirfarandi niðurstöðu:

```
Console C:/Users/egill/OneDrive/4Misseri/LikindiOgTolfraedi/R_verkefni/R_skjol/
> precipitationStats()
         Mean Median Variance Standard Deviation First Quartile Third Quartile Interquartile Range
Value 71.72024   69.7 355.3445           18.85058         56.775         82.525               25.75
Mean value is larger than median value.
Range with 50 percent of values with median as mid-point: [56.775 ; 82.525]
                      Mean  Median   Variance Standard Deviation First Quartile Third Quartile Interquartile Range
Logarithmic values 4.239824 4.24415 0.06630658          0.2575006       4.039089       4.413098            0.374009
> 
```

Þar sem öll gildin sem beðið var um koma fyrir.

1.b. Hvort er meðaltalið eða miðgildið stærra? Hvers vegna?

Svar: Eins og sést af töflunni að ofan er meðaltalið stærra en miðgildið. Þetta er væntanlega sökum þess að að eru nokkrir "útlagar"sem eru mun hærri en miðgildið en engir "útlagar"sem eru fyrir neðan miðgildið.

Áhrifin sem þetta hefur er að útlagarnir fyrir ofan hífa upp meðaltalið og skekkja það svo það verður hærra en miðgildið. Eins myndi miðgildið vera hærra en meðaltalið ef útlagarnir fyrir neðan miðgildið væru fleiri en útlagarnir fyrir ofan.

1.c. Á hvaða bili liggja 50% af mælingum sitt hvorum megin við miðgildið?

Svar: Svarið er einfaldlega bilið sem er milli fyrsta fjórðungsmarks og þriðja fjórðungsmarks. Eins og sést á mynd að ofan reiknast bilið [56.775 ; 82.525]

1.d. Reiknið logrann af hverju staki í gögnunum og reiknið sömu stærðir og hér að ofan í 1.a. en setjið gögnin í nýja töflu.

Svar: Aftur má athuga myndina til að sjá töfluna með gögnunum fyrir logragögnin:

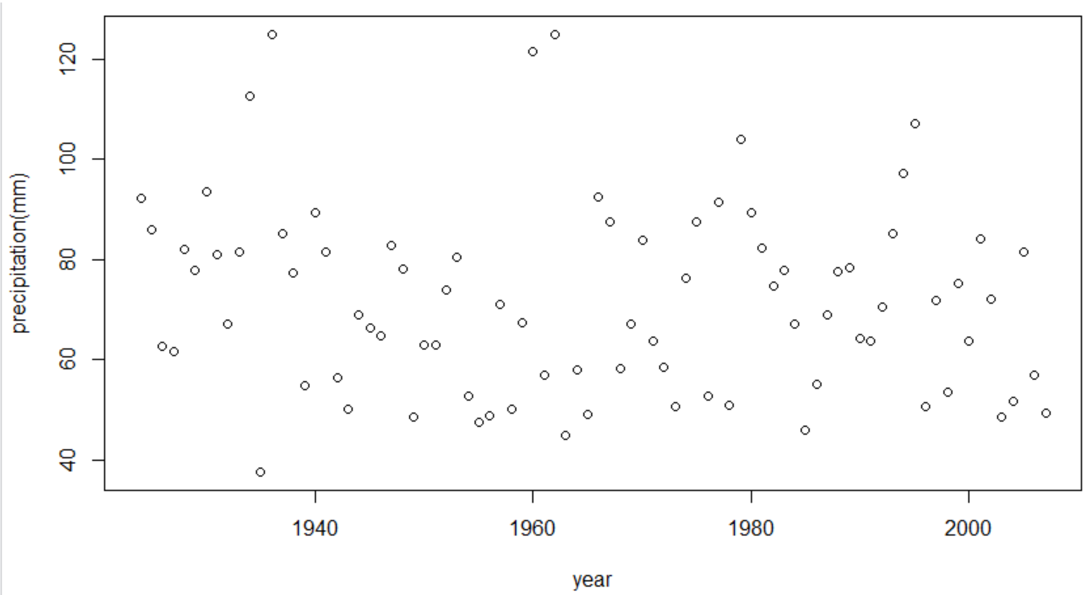

```
Console C:/Users/egill/OneDrive/4Misseri/LikindiOgTolfraedi/R_verkefni/R_skjol/
> precipitationStats()
         Mean Median Variance Standard Deviation First Quartile Third Quartile Interquartile Range
Value 71.72024   69.7 355.3445           18.85058         56.775         82.525               25.75
Mean value is larger than median value.
Range with 50 percent of values with median as mid-point: [56.775 ; 82.525]
                      Mean  Median   Variance Standard Deviation First Quartile Third Quartile Interquartile Range
Logarithmic values 4.239824 4.24415 0.06630658          0.2575006       4.039089       4.413098            0.374009
> 
```

Verkefni 2

Hér á að teikna myndir af gögnunum. Byggt á myndunum á að meta hvort að normaldreifingin lýsi gögnunum nægjanlega vel. Einnig á að meta hvort að normaldreifingin lýsi logranum af gögnunum nægjanlega vel. Teiknið eftirfarandi:

2.a. Teiknið árlega hámarkssólarhringsúrkomu á móti tíma. Á hvaða ári rigndi mest?

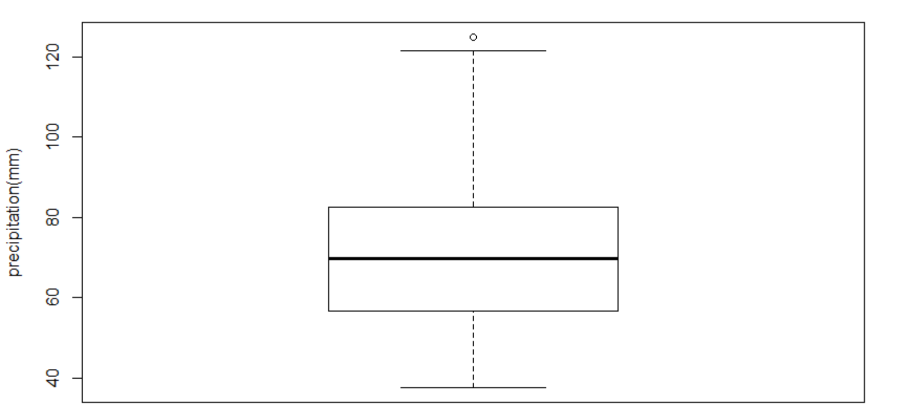
Svar: Við keyrum fall úr `precipitationGraphs.r` með skipun `precipitationGraphs(0,FALSE)` til að fá eftirfarandi mynd:



Forritið gefur síðan eftirfarandi úttak á skipanalínu:
`Maximum precipitation was 125 in the year 1936`

2.b. Kassarit. Sýnir kassaritið einhverja útlaga?

Svar: Við keyrum sama fall og áðan með skipun `precipitationGraphs(1,FALSE)` til að fá eftirfarandi mynd:

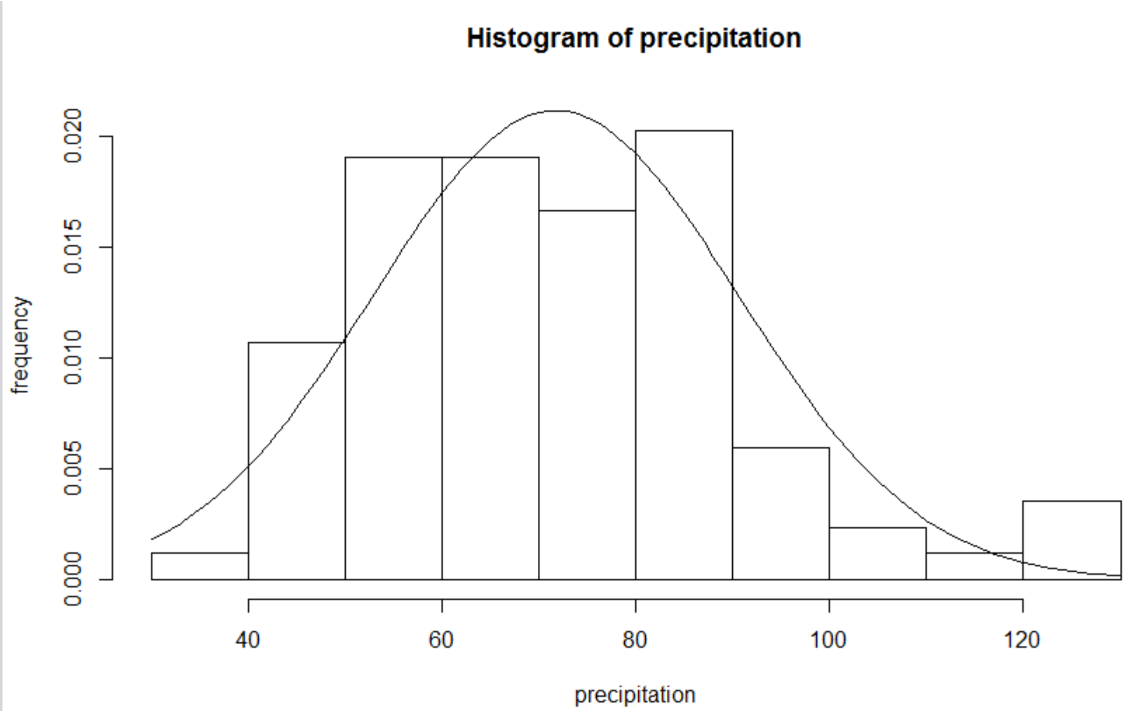


Einnig gefur forritið eftirfarandi úttak á skipanalínuna:

```
Number of outliers in boxplot: 2
```

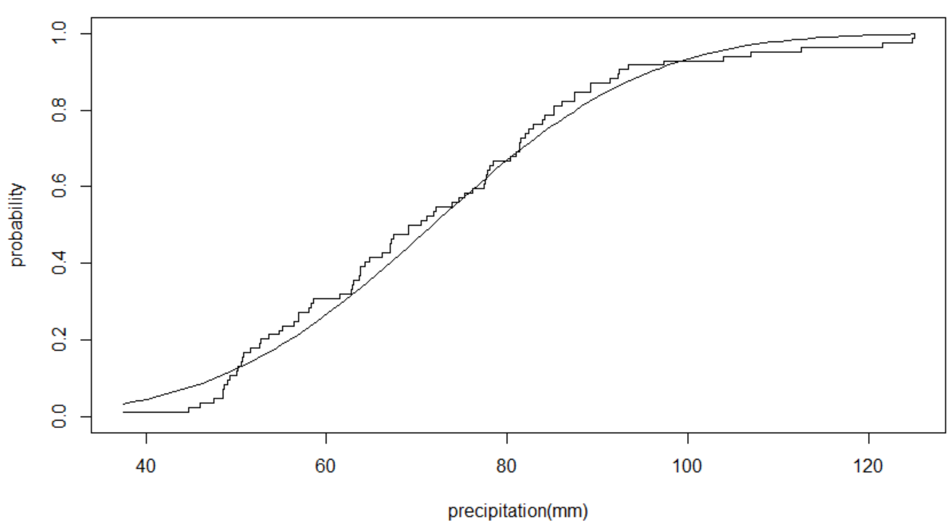
2.c. Tíðnirit með flatarmál sem er jafn einum. Teiknið ofan á það þéttifall normaldreifingar meðalgildi jafnt $\overline{x}$ og staðalfráviki s.

Svar: Við notum skipunina `precipitationGraphs(2,FALSE)` og fáum:



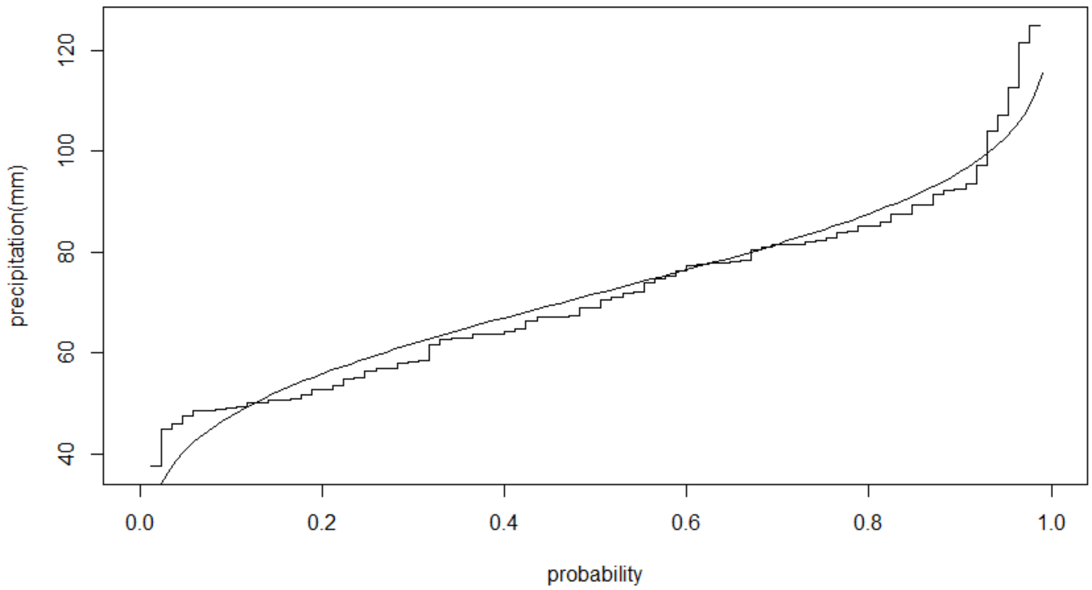
2.d. Dreififall úrtaks. Teiknið ofan á sætisfall úrtaksins sætisfall normaldreifingar með meðalgildi jafnt $\overline{x}$ og staðalfráviki s.

Svar: Keyrum fallið aftur með skipun `precipitationGraphs(3,FALSE)` og fáum:



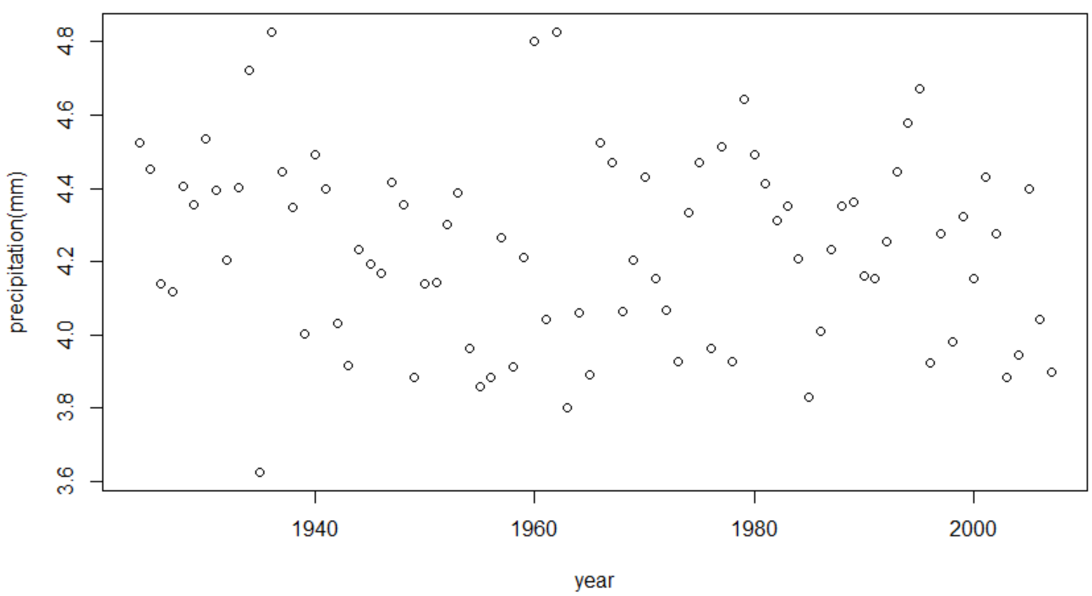
2.e. Sætisfall úrtaks. Teiknið ofan á það sætisfall normaldreifingar með meðalgildi jafnt $\overline{x}$ og staðalfráviki s.

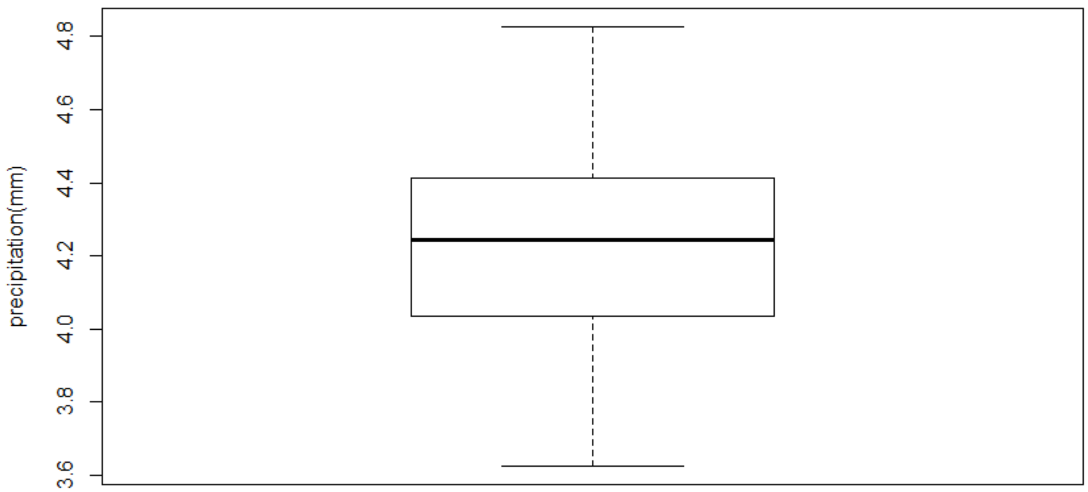
Svar: Keyrum þetta fall enn eina ferðina með skipun `precipitationGraphs(4,FALSE)` til að fá:

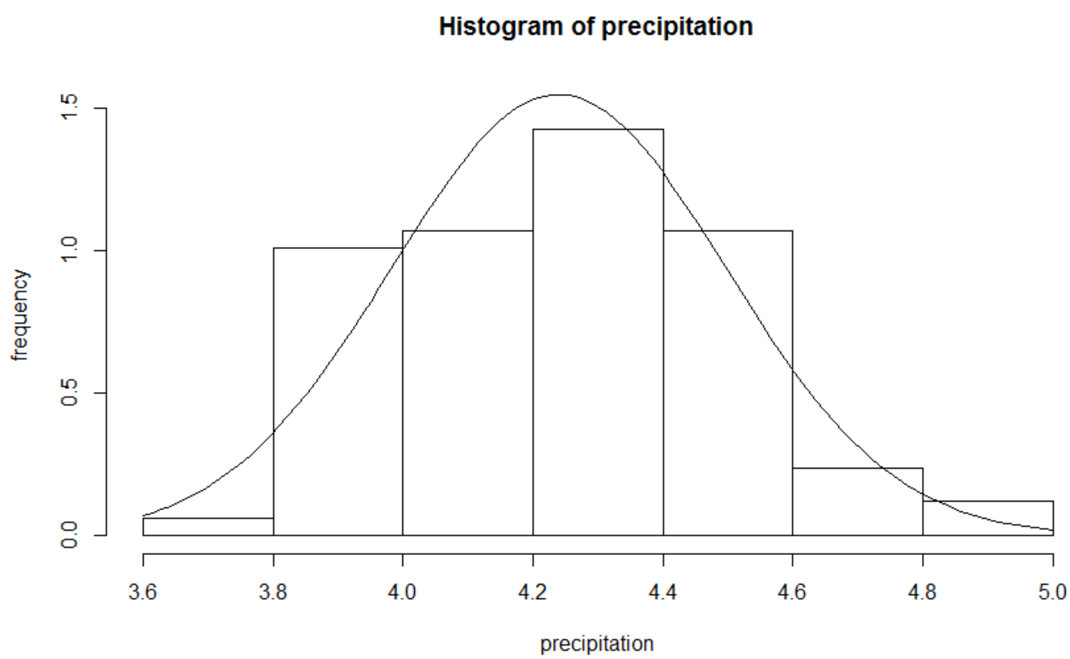


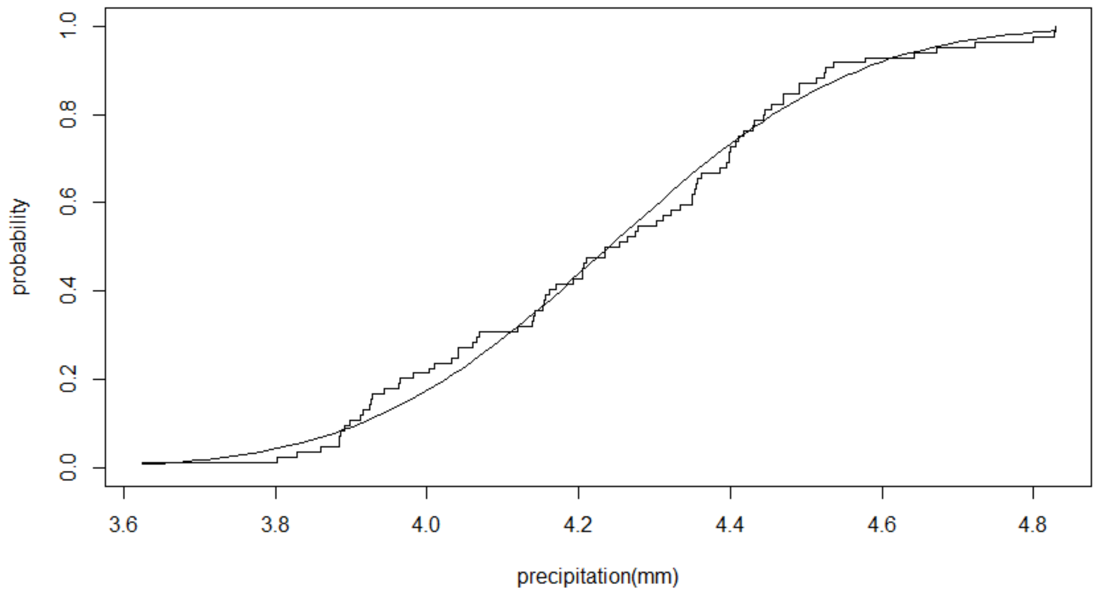
2.f. Teiknið sömu myndir og og í liðunum að ofan nema með logragildin af gögnunum.

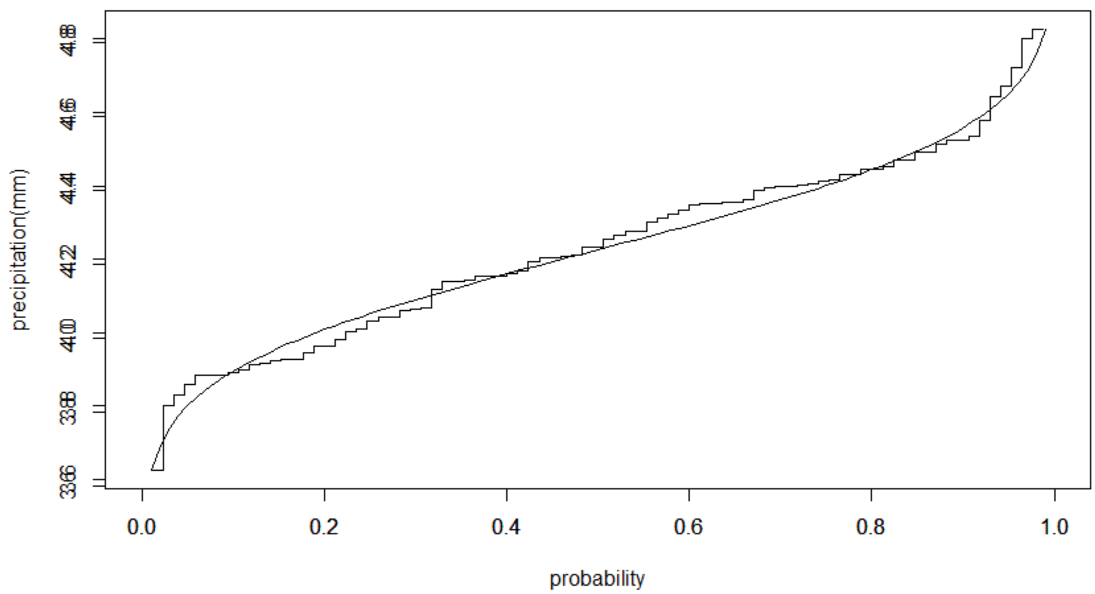
Svar: Notum þetta mjög svo skemmtilega forrit nokkrum sinnum í viðbót nema hvað að núna er seinna viðfangið `TRUE` í öllum tilfellum. Þá fást þessar undurfögru og mjög svo fróðlegu gröf:




precipitation(mm)
3.6
3.8
4.0
4.2
4.4
4.6
4.8


Histogram of precipitation
frequency
0.0
0.5
1.0
1.5
3.6
3.8
4.0
4.2
4.4
4.6
4.8
5.0
precipitation





2.g. Lýsir normaldreifing gögnunum nægjanlega vel? Lýsir lognormaldreifing gögnunum nægjanlega vel? Notið myndirnar úr liðunum hér fyrir ofan til að rökstyðja svör ykkar.

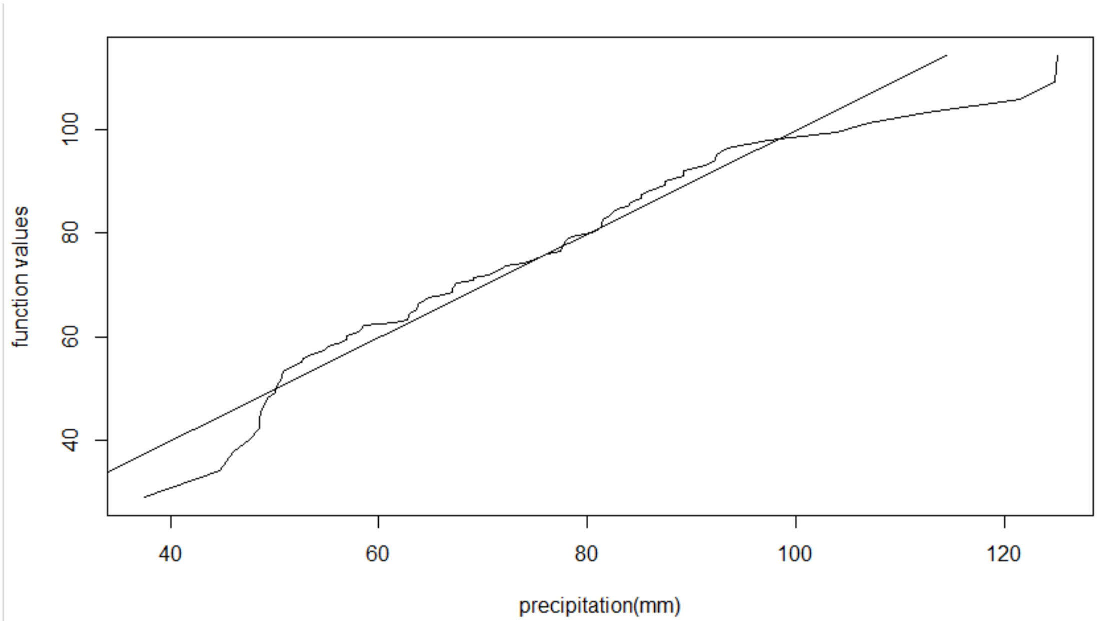
Svar: Bæði normaldreifing og lognormaldreifing lýsa gögnunum nokkuð vel eins og sést á ferlum að ofan (falla nokkuð vel að fræðilegum normaldrefingum). Hins vegar er lognormaldreifingin aðeins betri þar sem hún er nákvæmari og skekkjur eru minni.

Verkefni 3

Hér á að teikna tvær myndir, þá fyrri byggða á gögnunum og hina byggða á logranum af gögnunum. Látum $x_1 < x_2 < ... < x_i$ tákna röðuðu gögnin

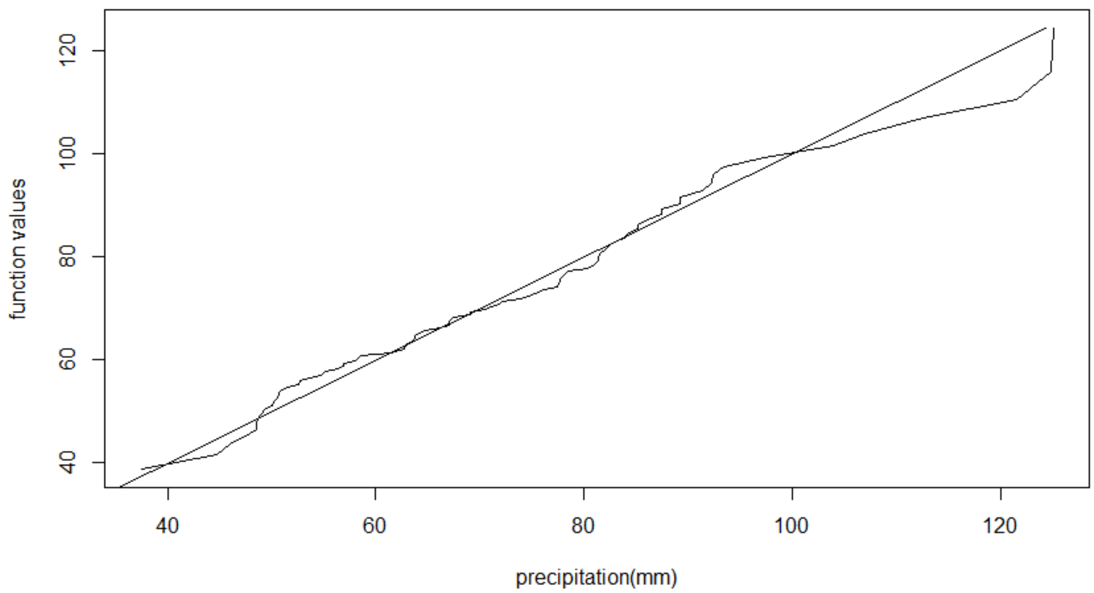
3.a. Teiknið x_i á x-ás og $\mu + \sigma\Phi^{-1}(i/(n+1))$. Teiknið á sömu mynd línuna $y = x$.

Svar: Keyrum fall með skipun `precipitationQnormGraphs(FALSE)` til að fá eftirfarandi graf:



3.b. Teiknið x_i á x-ás og $\mu_y + \sigma_y\Phi^{-1}(i/(n+1))$. Teiknið á sömu mynd línuna $y = x$.

Svar: Keyrum fall með skipun `precipitationQnormGraphs(TRUE)` til að fá eftirfarandi graf:



3.c. Því nær sem að punktarnir eru beinu línunni því betur lýsir dreifingin sem er lögð til gögnunum. Notið þessar tvær myndir til að svara eftirfarandi spurningum. Lýsir normaldreifing gögnunum nægjanlega vel? Lýsir lognormaldreifing gögnunum nægjanlega vel?

Svar: Af þessum tveimur myndum að dæma er log-normaldreifing þó nokkuð nákvæmari en normaldreifing. Hvort log-normaldreifing eða normaldreifing er "nægjanleg"fer alfarið eftir því hvað er verið að gera hverju sinni.

Verkefni 4

Metið líkurnar á því að sólarhringsúrkoma á Fagurhólsmýri sé meiri en 115 mm, það er, metið $P(X > 115)$

4.a. með því að finna hlutfall mælinga stærri en 115 mm.

4.b. með því að gera ráð fyrir að normaldreifingin lýsi mælingunum nægjanlega vel.

4.c. með því að gera ráð fyrir að normaldreifingin lýsi logranum af mælingunum nægjanlega vel.

Svar: Við keyrum seinasta fallið með kallinu `precipitationEstimation(115)` og fáum:

```
> precipitationEstimation(115)
Probability of more than 115 mm of precipitation built
on proportions: 0.0357142857142857
Probability of more than 115 mm of precipitation built
on normal distribution: 0.0108397295634248
Probability of more than 115 mm of precipitation built
on log-normal distribution: 0.024905613055092
```

4.d. Ef þið ættuð að meta $P(X > 130)$, hvaða aðferð munduð þið nota? Aðferðina í (a), (b), (c) eða einhverja aðra aðferð? Útskýrið val ykkar.

Svar: Af fyrrum liðum að dæma væri skynsamlegast að velja log-normal dreifingu. Hlutfalls-ágiskun er afar ónákvæm aðferð og byggist bara á gögnum sem eru þegar komin. Skekkjur í því módeli magnast upp með svona ágiskunum. Log-normaldreifing virðist draga mun frekar úr skekkjum heldur en normaldreifing og gefa betri niðurstöður. Eflaust eru til flóknari og betri aðferðir til að smíða svona líkön en áætla má að log-normaldreifing gefi nógu nákvæma mynd fyrir hin ýmsu verk.

Viðauki - Forrit

Verkefni 1

```
precipitationStats <- function() {

  # Input of data for statistic processing
  max_urkoma <- read.table("precipitation_fagurhm.txt")
  urkoma <- max_urkoma[,2]
  # Create matrix for storing statistics and label columns/rows
  statMatrix <- matrix(rep(c(0),7),ncol=7,byrow=TRUE)
  colnames(statMatrix) <- c("Mean", "Median", "Variance", "Standard Deviation",
                            "First Quartile", "Third Quartile", "Interquartile Range")
  rownames(statMatrix) <- c("Value")

  # Calculate statistical values and input data
  statMatrix[1,1] <- mean(urkoma)
  statMatrix[1,2] <- median(urkoma)
  statMatrix[1,3] <- var(urkoma)
  statMatrix[1,4] <- sd(urkoma)
  statMatrix[1,5:6] <- quantile(urkoma,probs = c(0.25,0.75))
  statMatrix[1,7] <- statMatrix[1,6] - statMatrix[1,5]

  # Export table to global scope for debugging
  statMatrix <<- statMatrix
  print(statMatrix)

  # Check if mean or median value is bigger
  if(statMatrix[1,1] > statMatrix[1,2]){
    message("Mean value is larger than median value.")
  } else if(statMatrix[1,1] < statMatrix[1,2]){
    message("Median value is larger than mean value.")
  } else{
    message("Mean value and median value are equal.")
  }

  # Calculate range for 50 percent of values with median values as middle point
  message("Range with 50 percent of values with median as mid-point: [",
          statMatrix[1,5], " ; ", statMatrix[1,6], "]")


  #####################
  # LOGARITHM STATISTICS
  #####################


  # Calculate logarithmic values of data
  logUrkoma <- log(urkoma)
  # Create matrix for storing statistics and label columns/rows
```

```r
  logStatMatrix <- matrix(rep(c(0),7),ncol=7,byrow=TRUE)
  colnames(logStatMatrix) <- c("Mean", "Median", "Variance", "Standard Deviation",
                               "First Quartile", "Third Quartile", "Interquartile Range")
  rownames(logStatMatrix) <- c("Logarithmic Values")

  # Calculate statistical values and input logarithmic data
  logStatMatrix[1,1] <- mean(logUrkoma)
  logStatMatrix[1,2] <- median(logUrkoma)
  logStatMatrix[1,3] <- var(logUrkoma)
  logStatMatrix[1,4] <- sd(logUrkoma)
  logStatMatrix[1,5:6] <- quantile(logUrkoma,probs = c(0.25,0.75))
  logStatMatrix[1,7] <- logStatMatrix[1,6] - logStatMatrix[1,5]

  # Export table to global scope for debugging
  logStatMatrix <<- logStatMatrix
  print(logStatMatrix)
}
```

precipitationStats.r

Verkefni 2

```r

" Variable graphType is an integer and used for determining what method should be
implemented in creating the graph/plot (0 for scatter graph, 1 for box plot,
2 for histogram, 3 for cumulative distribution function or 4 for quantile function).
The variable logarithm is a boolean value to determine whether the logarithm of
the data should be used or not (TRUE means logarithmic values are used)."

precipitationGraphs <- function(graphType,logarithm){

  # Input of data for statistic processing
  max_urkoma <- read.table("precipitation_fagurhm.txt")

  # Use logarithmic data if requested
  if(logarithm){
    max_urkoma[2] <- log(max_urkoma[2])
    message("Notice: Logarithmic data being used")
  }

  # Often used statistics
  sortedData <- sort(max_urkoma[,2])
  meanOfData <<- mean(sortedData)
  stdDevOfData <<- sd(sortedData)
  n <- length(max_urkoma[,2])

  # Draw scatter graph if required (graphType = 0)
```

```
  if(graphType == 0){
    plot(max_urkoma, xlab="year", ylab="precipitation(mm)")
    # Check what year the precipitation was the most
    maxRow = 1
    for(g in 1:n){
      if(max_urkoma[g,2] > max_urkoma[maxRow,2]){
        maxRow <- g
      }
    }
    message("Maximum precipitation was ", max_urkoma[maxRow,2], " in the year ", max_urkoma[maxRow,1])
  }

  # Draw boxplot if required (graphType = 1)
  if(graphType == 1){
    boxplotInstance <- boxplot(max_urkoma[2], ylab="precipitation(mm)")
    message("Number of outliers in boxplot: ", length(boxplotInstance$out))
    }

  # Draw histogram if required (graphType = 2)
  if(graphType == 2){
    precipitation <- max_urkoma[,2]
    histogramInstance <- hist(precipitation, plot = F)
    yLimit <- range(0,histogramInstance$density, dnorm(0)/sd(precipitation))
    hist(precipitation, freq = F, ylim = yLimit, ylab = "frequency")
    curve(dnorm(x, mean = mean(max_urkoma[,2]), sd = sd(max_urkoma[,2])), add = T)
  }

  # Draw cumulative distribution function if required (graphType = 3)
  if(graphType == 3){
    # Cumulative distribution function for data
    plot(sortedData, (1:n)/n, type = "s", ylim = c(0,1), xlab = "precipitation(mm)", ylab = "probability")
    # Cumulative distribution function based on statistics from data
    curve(pnorm(x, mean = meanOfData, sd = stdDevOfData), add = T)
    }

  # Draw quantile function id required (graphType = 4)
  if(graphType == 4){
    # Quantile function for data
    sortedData <- sort(max_urkoma[,2])
    p <- (1:n)/(n + 1)
    plot(p,sortedData, type = "s", xlim = c(0,1), xlab = "probability", ylab = "precipitation(mm)")
    # Quantile function built on data statistics
    if(logarithm){
      # Workaround for logarithmic graph
      par(new = TRUE)
      curve(qnorm(x,mean = meanOfData, sd = stdDevOfData), xlab = ""
```

```
            , ylab = "")
    }else{
      curve(qnorm(x,mean = meanOfData, sd = stdDevOfData), add = T)
    }
  }
}
```

precipitationGraphs.r

Verkefni 3

```

# Parameter logarithm is a boolean variable to determine
# whether logarithmic data is used

precipitationQnormGraphs <- function(logarithm){

  # Input of data for statistic processing
  max_urkoma <- read.table("precipitation_fagurhm.txt")
  log_urkoma <- log(sort(max_urkoma[,2]))

  # Use logarithmic data if requested
  if(logarithm){
    message("Notice: Logarithmic data being used")
  }

  # Often used statistics
  sortedData <- sort(max_urkoma[,2])
  meanOfData <- mean(sortedData)
  stdDevOfData <- sd(sortedData)
  n <- length(sortedData)
  logMeanOfData <- mean(log_urkoma)
  logStdDevOfData <- sd(log_urkoma)

  # Produce values according to assignment and plot them
  if(logarithm){
    functionValues <<- exp(logMeanOfData + logStdDevOfData*qnorm((1:
        n)/(n+1)))
  }else{
    functionValues <<- meanOfData + stdDevOfData*qnorm((1:n)/(n+1))
  }
  plot(sortedData, functionValues, type = "n", xlab = "precipitation
      (mm)", ylab = "function values")
  lines(sortedData, functionValues)
  lines(c(0,max(functionValues)), c(0,max(functionValues)))
  }
```

precipitationQnormGraphs.r

Verkefni 4

```

# Various estimations of input data
```

```r

precipitationEstimation <- function(amount){

  # Input of data for statistic processing
  max_urkoma <- read.table("precipitation_fagurhm.txt")

  # Often used statistics
  sortedData <<- sort(max_urkoma[,2])
  meanOfData <- mean(sortedData)
  stdDevOfData <- sd(sortedData)
  n <- length(max_urkoma[,2])

  # Find place in data where values are less than amount
  place = 1
  for(g in 1:n){
    if(sortedData[g] > amount){
      break
    }
    place <- g
  }

  # Calculate probability built on proportions
  proportion <- (n - place)/n
  message("Probability of more than ", amount,
          " mm of precipitation built on proportions: ", proportion)

  # Calculate the probability built on normal distribution
  proportion <- 1 - pnorm(amount, mean = meanOfData, sd =
      stdDevOfData)
  message("Probability of more than ", amount,
          " mm of precipitation built on normal distribution: ",
              proportion)

  # Calculate the probability built on log-normal distribution
  proportion <- 1 - pnorm(log(amount), mean = mean(log(sortedData)),
      sd = sd(log(sortedData)))
  message("Probability of more than ", amount,
          " mm of precipitation built on log-normal distribution: ",
               proportion)

}
```

precipitationEstimation.r