Djillali Liabes University of Sidi Bel Abbes
Faculty of Exact Sciences
Department of Computer Sciences

Module : Aide à la décision
1st year of Master's degeree in
Networks,Information Systems & Security(RSSI)
2021/2022

Confidence Intervals and Risk in R PART 2

Students:
Hadjazi M.Hisham
AMOUR Wassim Malik
Group: 01/RSSI

Instructors:
Pr. YOUSFATE
Abderrahmane
Dr. BENBEKRITI Soumia

A paper submitted in fulfilment of the requirements for the
Aide à la décision TP-06

May 11, 2022

Contents

Chapter 1

Confidence Intervals in R

1.1 Confidence Interval with runif

As requested we will using **runif** function to generates random deviates of the **uniform distribution** from size of 1000 and range between -2 and 3.

1.1.1 Generation of Random uniform Distribution Seed

```r
LO = -2 # Declaration of lower bownd
UP = 3 # Declaration of upper bownd
n = 1000 # Declaration of sample size

x <- runif(n, LO, UP) # Running runif function to create population
hist(x, freq = FALSE, xlab = 'x', density = 20) # Histogram of population
x # all generated values
```

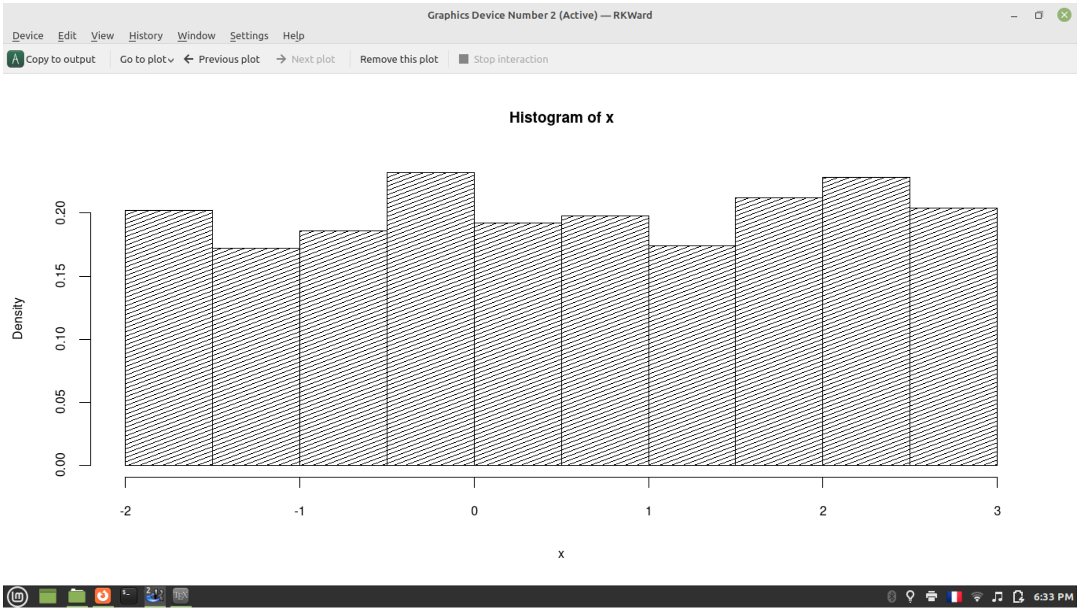


FIGURE 1.1: Seed Histogram.

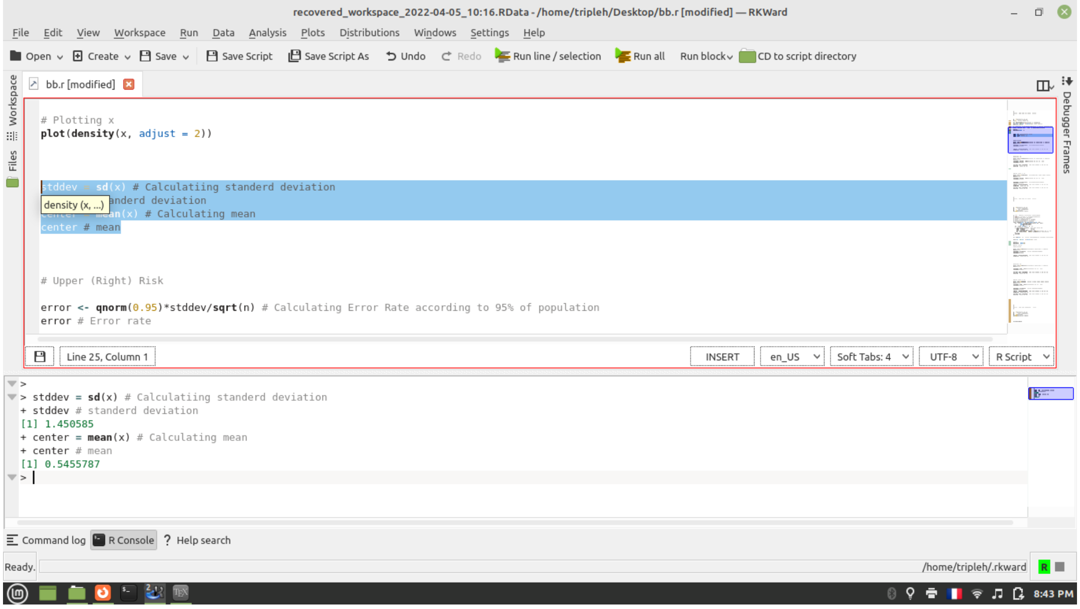
Sample of generated numbers :

```
[656]  0.3813996161  2.4257347495  2.9501461792 -0.3439974906  2.6721924189
```

```
[661] -1.7069758398 -1.9919105659 -0.5682839947 -0.1698572240  2.1766188550
[666]  2.1259460303 -0.4748134005  2.6861825290 -0.3083081152  0.8204065480
[671]  2.7893994430 -0.4446464770 -1.2376538937  1.7844204041 -0.4890021197
[676]  2.9545168958 -0.9832733071 -0.1086031161  1.9523365453  0.1649452916
[681] -0.8115781138 -1.5075847686  2.1916635458 -0.6813517509  1.1595426821
[686] -1.2972299247  1.1057026773 -1.0271868056  0.0378431329  1.4667883010
[691]  1.6694702739  2.1264238802  2.9148606462  2.6315843845  0.8007933067
[696]  2.0933396120  1.8568044163  1.5384381283  2.9681203486 -1.4027119374
[701]  0.5982736656 -0.4392616409  0.0532791491  1.1336477492 -0.4013634117
[706]  2.1036622517  1.1725118810  1.4840549517  1.4851128771 -0.9378550777
[711]  1.3588998194 -1.7498405632 -1.0858148329  2.7649365244 -1.6944917948
[716]  2.6904253052 -1.3718907470  1.4137118515 -0.3775374934 -0.8040759950
[721] -1.3621708534  0.0552953065 -1.6524381090  1.5260674357  2.5928918663
[726]  2.7491736128 -1.8523521284  0.2638581775  0.3966596590 -1.4307652919
[731] -1.5760219381 -1.1228304966 -1.8921418609  1.4406364267  0.5347665627
[736]  1.9234586596 -0.7565151707  2.0440741449  2.7295886686  1.1158512493
[741]  1.5015663505  0.5561288605  1.4875788400  2.3570341938  2.8440772563
[746] -0.3308353100  0.4111624272  1.3974237274  1.0447405553 -1.3868167617
[751]  1.6946422944  2.7661013680  1.5291836690  2.9929703807 -0.5640524689
[756] -0.8812302880 -0.0748715147 -1.1108034016  1.6513440299  1.8687401486
[761] -1.2423470458 -1.7383970383  1.5178913060 -1.1912305378  0.7003954467
[766]  2.4230878598 -0.9534112730  1.1830788974  1.1611230748 -1.1721150582
[771]  1.7582610291  0.0346237416  1.9112062566  1.4791723755 -1.7659081384
[776]  1.5747120762  2.1935196554 -0.2110054267  2.6586451575  0.6185839961
[781]  0.9111335962  2.9610665615  1.2534163748  0.7543342090  0.4418109108
[786]  1.7382183538  2.7631716216  0.3801716522 -0.3324272453 -1.6288395429
[791]  0.9629155551 -0.9031352359  0.0713478327 -1.9929349020  0.2839912050
[796] -0.2094068220  2.1137463006  1.1335198425 -1.3629527255  0.7328717150
[801] -0.8635372289 -1.2511274433  2.0804919973 -0.0102763081  2.0244089393
[806]  2.0593909181  1.4906496108  1.6627415335  2.8765612335  2.5833753119
[811]  2.7221900942 -0.2749454468  2.9613368977 -1.4997302629  2.2851109779
[816] -0.2907404716 -1.0059362201 -1.3930226308 -0.5002697073 -0.6187703651
[821]  2.3164683203  0.3184335779  2.9863529194  2.7773846728  0.4733544211
```

1.1.2 Finding Mean and Standard Deviation

```r
stddev = sd(x) # Calculatiing standerd deviation
stddev # standerd deviation
center = mean(x) # Calculating mean
center # mean
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Save Script Save Script As Undo Redo Run line / selection Run all Run block CD to script directory
Workspace
Files
bb.r [modified]
Debugger Frames
density (x, ...)
Line 25, Column 1
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:43 PM

```
# Plotting x
plot(density(x, adjust = 2))


stddev = sd(x) # Calculatiing standerd deviation
stddev # standerd deviation
center = mean(x) # Calculating mean
center # mean


# Upper (Right) Risk

error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate
```

```
>
> stddev = sd(x) # Calculatiing standerd deviation
+ stddev # standerd deviation
[1] 1.450585
+ center = mean(x) # Calculating mean
+ center # mean
[1] 0.5455787
>
```

FIGURE 1.2: Mean and SD.

1.1.3 Plotting Density

```
# Plotting
plot(density(x, adjust = 2))
```

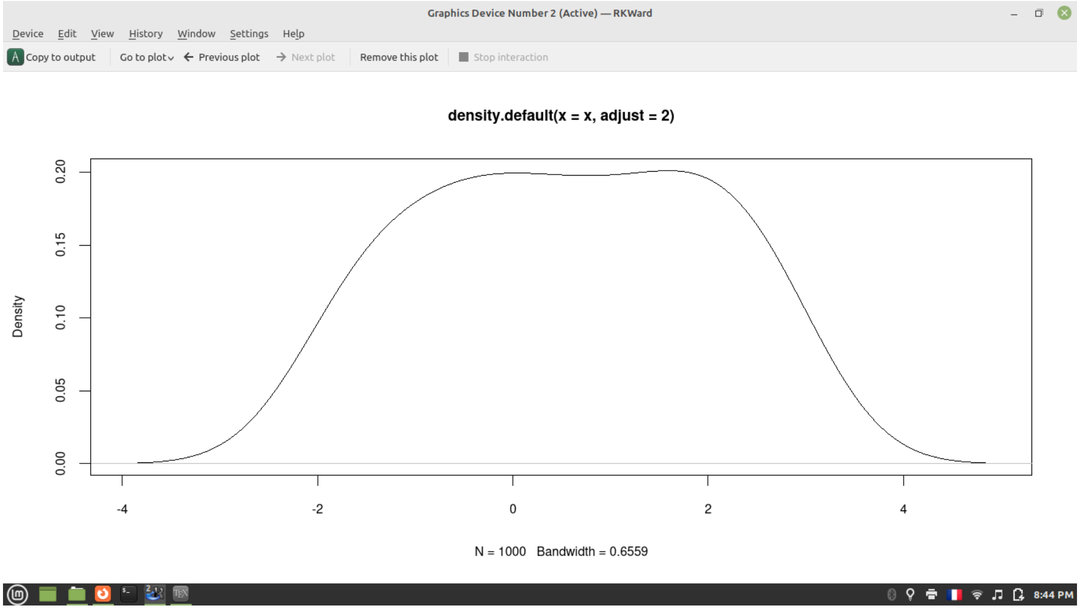


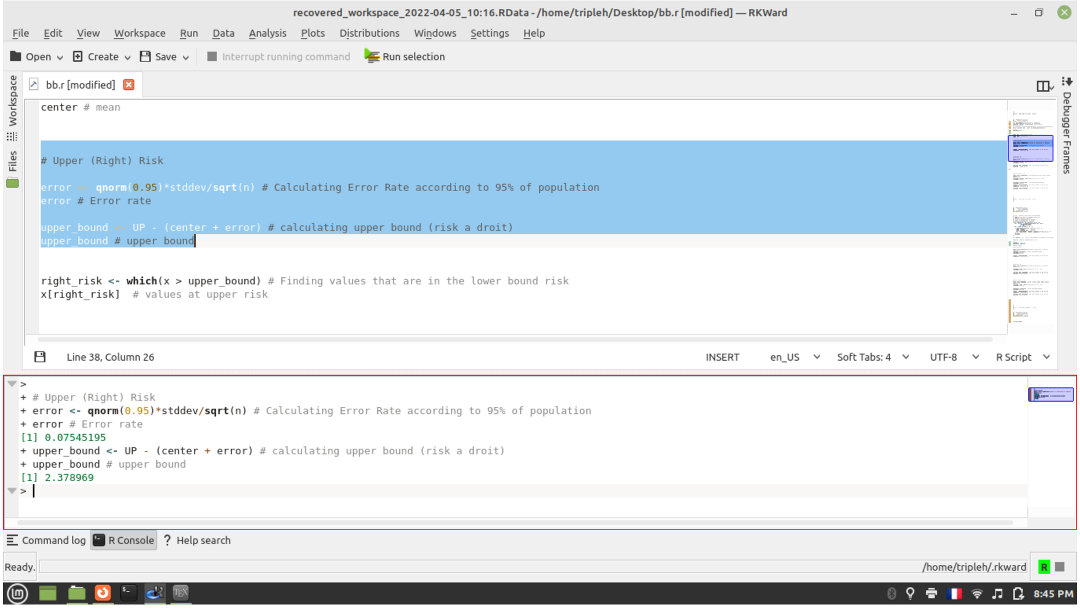
FIGURE 1.3: Plot Density.

1.1.4 Right Risk of 95%

calculating 95% error rate, upper limit

```
error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate
    according to 95% of population
```

```
error # Error rate

upper_bound <- UP - (center + error) # calculating upper bound (
    risk a droit)
upper_bound # upper bound
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Interrupt running command Run selection
Workspace
Files
bb.r [modified]
Debugger Frames
Line 38, Column 26
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:45 PM

```
center # mean


# Upper (Right) Risk

error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate

upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
upper_bound # upper bound


right_risk <- which(x > upper_bound) # Finding values that are in the lower bound risk
x[right_risk]  # values at upper risk
```

```
>
+ # Upper (Right) Risk
+ error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
+ error # Error rate
[1] 0.07545195
+ upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
+ upper_bound # upper bound
[1] 2.378969
>
```

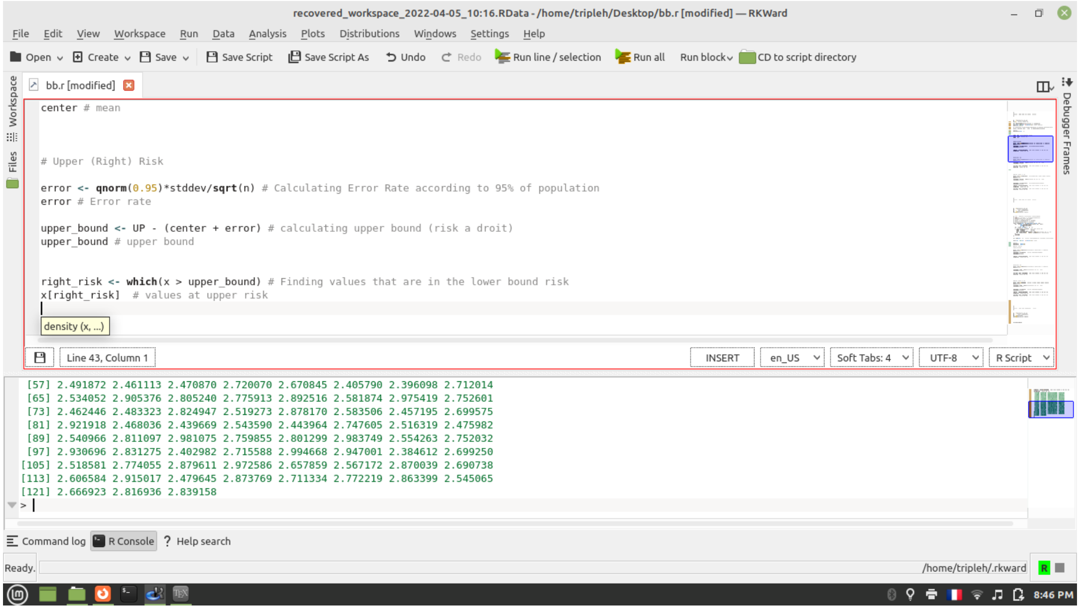
FIGURE 1.4: Error Rate Right.

Upper Limit

```
+ error # Error rate
[1] 0.07545195

+ upper_bound # upper bound
[1] 2.378969
```

1.1.5 Finding all Values at Risk (Right)

```
right_risk <- which(x > upper_bound) # Finding values that are in
    the lower bound risk
x[right_risk]  # values at upper risk
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Save Script Save Script As Undo Redo Run line / selection Run all Run block CD to script directory
bb.r [modified]
Workspace
Files
Debugger Frames
density (x, ...)
Line 43, Column 1
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:46 PM

```
center # mean



# Upper (Right) Risk

error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate

upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
upper_bound # upper bound


right_risk <- which(x > upper_bound) # Finding values that are in the lower bound risk
x[right_risk]  # values at upper risk
```

```
 [57] 2.491872 2.461113 2.470870 2.720070 2.670845 2.405790 2.396098 2.712014
 [65] 2.534052 2.905376 2.805240 2.775913 2.892516 2.581874 2.975419 2.752601
 [73] 2.462446 2.483323 2.824947 2.519273 2.878170 2.583506 2.457195 2.699575
 [81] 2.921918 2.468036 2.439669 2.543590 2.443964 2.747605 2.516319 2.475982
 [89] 2.540966 2.811097 2.981075 2.759855 2.801299 2.983749 2.554263 2.752032
 [97] 2.930696 2.831275 2.402982 2.715588 2.994668 2.947001 2.384612 2.699250
[105] 2.518581 2.774055 2.879611 2.972586 2.657859 2.567172 2.870039 2.690738
[113] 2.606584 2.915017 2.479645 2.873769 2.711334 2.772219 2.863399 2.545065
[121] 2.666923 2.816936 2.839158
> 
```

FIGURE 1.5: Finding numbers at Risk.

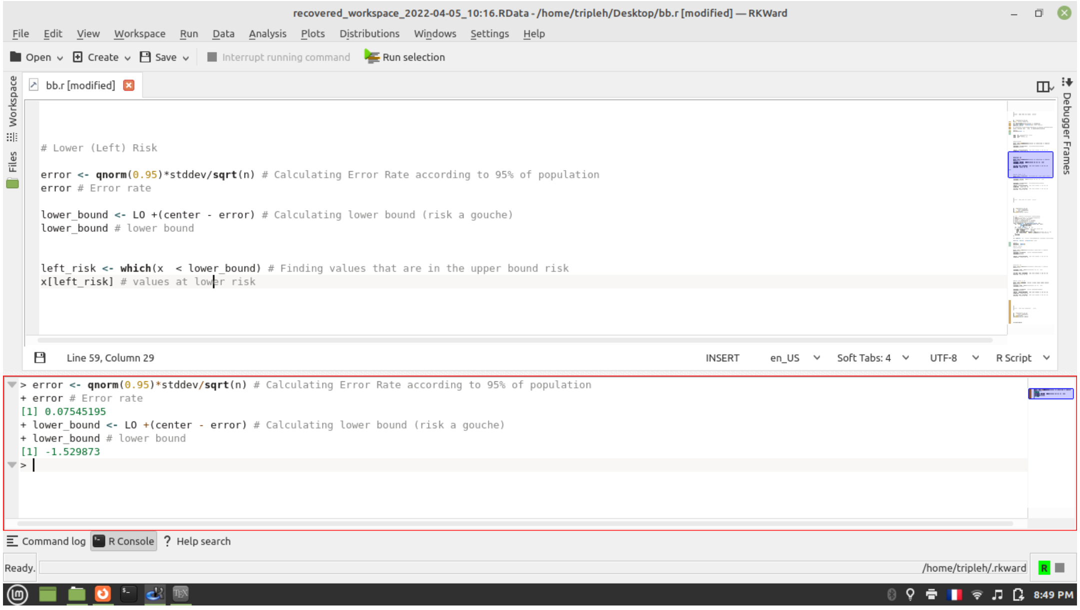
Values at Upper (Right) Risk :

```
  [1] 2.710283 2.764196 2.705285 2.444434 2.755979 2.878734 2.711472 2.722598
  [9] 2.850662 2.882673 2.563105 2.680986 2.776878 2.963367 2.644889 2.612474
 [17] 2.937783 2.577454 2.708086 2.786869 2.612106 2.468001 2.704758 2.639299
 [25] 2.572904 2.624294 2.791844 2.703460 2.924236 2.915694 2.769651 2.387021
 [33] 2.786046 2.777726 2.875089 2.828911 2.760081 2.577049 2.795525 2.654686
 [41] 2.916622 2.979905 2.681292 2.915171 2.885494 2.407985 2.924766 2.494888
 [49] 2.951954 2.717667 2.403572 2.706490 2.950434 2.801669 2.904595 2.647996
 [57] 2.491872 2.461113 2.470870 2.720070 2.670845 2.405790 2.396098 2.712014
 [65] 2.534052 2.905376 2.805240 2.775913 2.892516 2.581874 2.975419 2.752601
 [73] 2.462446 2.483323 2.824947 2.519273 2.878170 2.583506 2.457195 2.699575
 [81] 2.921918 2.468036 2.439669 2.543590 2.443964 2.747605 2.516319 2.475982
 [89] 2.540966 2.811097 2.981075 2.759855 2.801299 2.983749 2.554263 2.752032
 [97] 2.930696 2.831275 2.402982 2.715588 2.994668 2.947001 2.384612 2.699250
[105] 2.518581 2.774055 2.879611 2.972586 2.657859 2.567172 2.870039 2.690738
[113] 2.606584 2.915017 2.479645 2.873769 2.711334 2.772219 2.863399 2.545065
[121] 2.666923 2.816936 2.839158
```

1.1.6 Left Risk of 95%

calculating 95% error rate, lower limit

```
error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate
    according to 95% of population
error # Error rate

lower_bound <- LO +(center - error) # Calculating lower bound (risk
     a gouche)
lower_bound # lower bound
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Interrupt running command Run selection
Workspace Files
bb.r [modified]
Debugger Frames
Line 59, Column 29 INSERT en_US Soft Tabs: 4 UTF-8 R Script
Command log R Console Help search
Ready. /home/tripleh/.rkward
8:49 PM

```
# Lower (Left) Risk

error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate

lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
lower_bound # lower bound


left_risk <- which(x  < lower_bound) # Finding values that are in the upper bound risk
x[left_risk] # values at lower risk
```

```
> error <- qnorm(0.95)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
+ error # Error rate
[1] 0.07545195
+ lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
+ lower_bound # lower bound
[1] -1.529873
> 
```

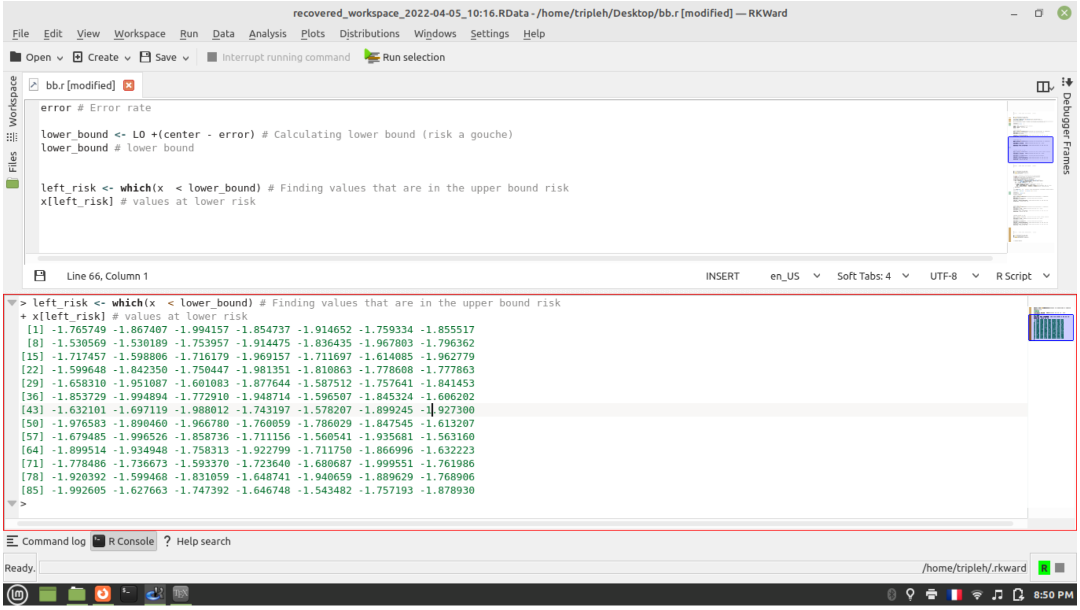
FIGURE 1.6: Error Rate Left.

Lower Limit

```
[1] 0.07545195

+ lower_bound # lower bound
[1] -1.529873
```

1.1.7 Finding all Values at Risk (Left)

```
left_risk <- which(x  < lower_bound) # Finding values that are in the upper bound risk
x[left_risk] # values at lower risk
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Interrupt running command Run selection
Workspace
Files
bb.r [modified]
Debugger Frames
Line 66, Column 1
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:50 PM

```
error # Error rate

lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
lower_bound # lower bound


left_risk <- which(x  < lower_bound) # Finding values that are in the upper bound risk
x[left_risk] # values at lower risk
```

```
> left_risk <- which(x  < lower_bound) # Finding values that are in the upper bound risk
+ x[left_risk] # values at lower risk
 [1] -1.765749 -1.867407 -1.994157 -1.854737 -1.914652 -1.759334 -1.855517
 [8] -1.530569 -1.530189 -1.753957 -1.914475 -1.836435 -1.967803 -1.796362
[15] -1.717457 -1.598806 -1.716179 -1.969157 -1.711697 -1.614085 -1.962779
[22] -1.599648 -1.842350 -1.750447 -1.981351 -1.810863 -1.778608 -1.777863
[29] -1.658310 -1.951087 -1.601083 -1.877644 -1.587512 -1.757641 -1.841453
[36] -1.853729 -1.994894 -1.772910 -1.948714 -1.596507 -1.845324 -1.606202
[43] -1.632101 -1.697119 -1.988012 -1.743197 -1.578207 -1.899245 -1.927300
[50] -1.976583 -1.890460 -1.966780 -1.760059 -1.786029 -1.847545 -1.613207
[57] -1.679485 -1.996526 -1.858736 -1.711156 -1.560541 -1.935681 -1.563160
[64] -1.899514 -1.934948 -1.758313 -1.922799 -1.711750 -1.866996 -1.632223
[71] -1.778486 -1.736673 -1.593370 -1.723640 -1.680687 -1.999551 -1.761986
[78] -1.920392 -1.599468 -1.831059 -1.648741 -1.940659 -1.889629 -1.768906
[85] -1.992605 -1.627663 -1.747392 -1.646748 -1.543482 -1.757193 -1.878930
>
```

FIGURE 1.7: Finding numbers at Left.

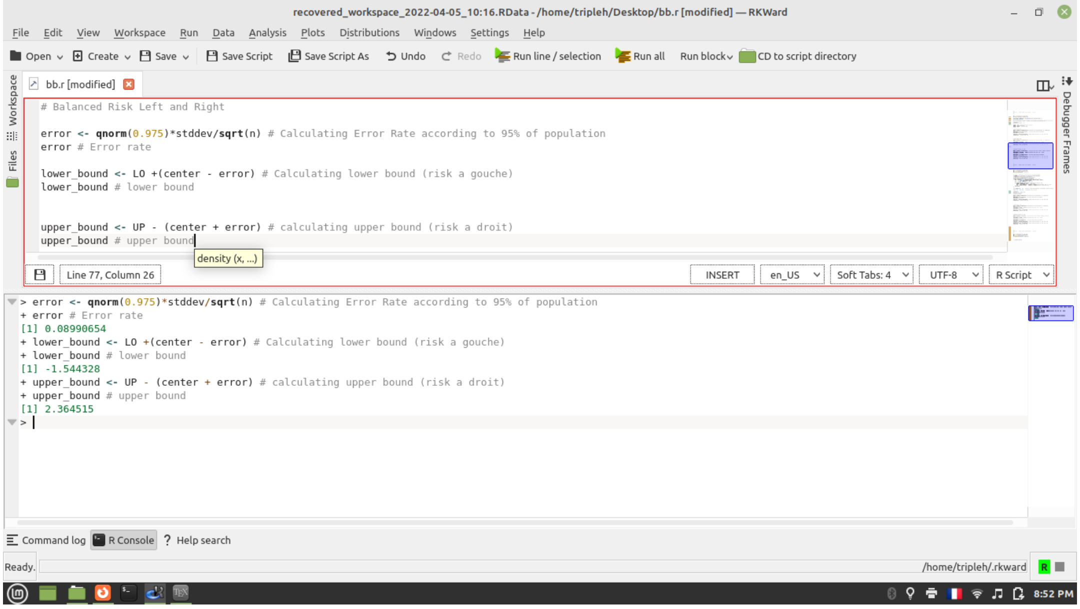
Values at Lower (Left) Risk :

```
 [1] -1.765749 -1.867407 -1.994157 -1.854737 -1.914652 -1.759334 -1.855517
 [8] -1.530569 -1.530189 -1.753957 -1.914475 -1.836435 -1.967803 -1.796362
[15] -1.717457 -1.598806 -1.716179 -1.969157 -1.711697 -1.614085 -1.962779
[22] -1.599648 -1.842350 -1.750447 -1.981351 -1.810863 -1.778608 -1.777863
[29] -1.658310 -1.951087 -1.601083 -1.877644 -1.587512 -1.757641 -1.841453
[36] -1.853729 -1.994894 -1.772910 -1.948714 -1.596507 -1.845324 -1.606202
[43] -1.632101 -1.697119 -1.988012 -1.743197 -1.578207 -1.899245 -1.927300
[50] -1.976583 -1.890460 -1.966780 -1.760059 -1.786029 -1.847545 -1.613207
[57] -1.679485 -1.996526 -1.858736 -1.711156 -1.560541 -1.935681 -1.563160
[64] -1.899514 -1.934948 -1.758313 -1.922799 -1.711750 -1.866996 -1.632223
[71] -1.778486 -1.736673 -1.593370 -1.723640 -1.680687 -1.999551 -1.761986
[78] -1.920392 -1.599468 -1.831059 -1.648741 -1.940659 -1.889629 -1.768906
[85] -1.992605 -1.627663 -1.747392 -1.646748 -1.543482 -1.757193 -1.878930
```

1.1.8 Balanced Risk of 95%

calculating 95% error rate, upper limit and lower limit

```
error <- qnorm(0.975)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate

lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
lower_bound # lower bound


upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
upper_bound # upper bound
```


recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Save Script Save Script As Undo Redo Run line / selection Run all Run block CD to script directory
bb.r [modified]
Workspace
Files
Debugger Frames
density (x, ...)
Line 77, Column 26
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:52 PM

```
# Balanced Risk Left and Right

error <- qnorm(0.975)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
error # Error rate

lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
lower_bound # lower bound


upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
upper_bound # upper bound
```

```
> error <- qnorm(0.975)*stddev/sqrt(n) # Calculating Error Rate according to 95% of population
+ error # Error rate
[1] 0.08990654
+ lower_bound <- LO +(center - error) # Calculating lower bound (risk a gouche)
+ lower_bound # lower bound
[1] -1.544328
+ upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
+ upper_bound # upper bound
[1] 2.364515
>
```

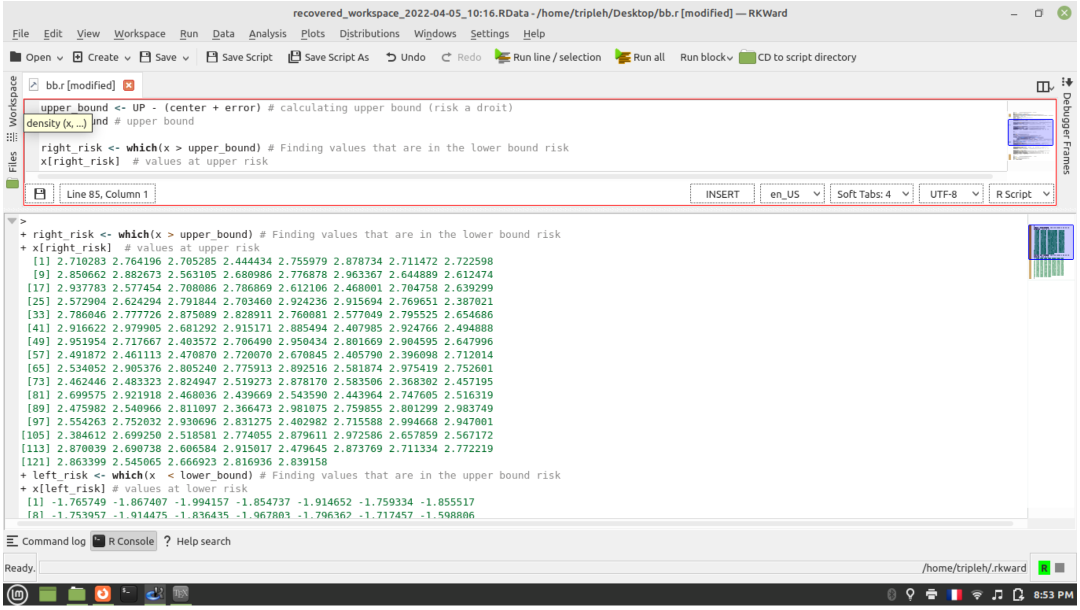
FIGURE 1.8: Error Rate.

Upper Limit and Lower Limit

```
+ error # Error rate
[1] 0.08990654

+ lower_bound # lower bound
[1] -1.544328

+ upper_bound # upper bound
[1] 2.364515
```

1.1.9 Finding all Values at Risk (Right and Left)

```
right_risk <- which(x > upper_bound) # Finding values that are in
    the lower bound risk
x[right_risk]  # values at upper risk

left_risk <- which(x  < lower_bound) # Finding values that are in
    the upper bound risk
x[left_risk] # values at lower risk
```



recovered_workspace_2022-04-05_10:16.RData - /home/tripleh/Desktop/bb.r [modified] — RKWard
File Edit View Workspace Run Data Analysis Plots Distributions Windows Settings Help
Open Create Save Save Script Save Script As Undo Redo Run line / selection Run all Run block CD to script directory
bb.r [modified]
Workspace
Files
Debugger Frames
density (x, ...)
Line 85, Column 1
INSERT
en_US
Soft Tabs: 4
UTF-8
R Script
Command log
R Console
Help search
Ready.
/home/tripleh/.rkward
8:53 PM

```
upper_bound <- UP - (center + error) # calculating upper bound (risk a droit)
          und # upper bound

right_risk <- which(x > upper_bound) # Finding values that are in the lower bound risk
x[right_risk]  # values at upper risk
```

```
>
+ right_risk <- which(x > upper_bound) # Finding values that are in the lower bound risk
+ x[right_risk]  # values at upper risk
  [1] 2.710283 2.764196 2.705285 2.444434 2.755979 2.878734 2.711472 2.722598
  [9] 2.850662 2.882673 2.563105 2.680986 2.776878 2.963367 2.644889 2.612474
 [17] 2.937783 2.577454 2.708086 2.786869 2.612106 2.468001 2.704758 2.639299
 [25] 2.572904 2.624294 2.791844 2.703460 2.924236 2.915694 2.769651 2.387021
 [33] 2.786046 2.777726 2.875089 2.828911 2.760081 2.577049 2.795525 2.654686
 [41] 2.916622 2.979905 2.681292 2.915171 2.885494 2.407985 2.924766 2.494888
 [49] 2.951954 2.717667 2.403572 2.706490 2.950434 2.801669 2.904595 2.647996
 [57] 2.491872 2.461113 2.470870 2.720070 2.670845 2.405790 2.396098 2.712014
 [65] 2.534052 2.905376 2.805240 2.775913 2.892516 2.581874 2.975419 2.752601
 [73] 2.462446 2.483323 2.824947 2.519273 2.878170 2.583506 2.368302 2.457195
 [81] 2.699575 2.921918 2.468036 2.439669 2.543590 2.443964 2.747605 2.516319
 [89] 2.475982 2.540966 2.811097 2.366473 2.981075 2.759855 2.801299 2.983749
 [97] 2.554263 2.752032 2.930696 2.831275 2.402982 2.715588 2.994668 2.947001
[105] 2.384612 2.699250 2.518581 2.774055 2.879611 2.972586 2.657859 2.567172
[113] 2.870039 2.690738 2.606584 2.915017 2.479645 2.873769 2.711334 2.772219
[121] 2.863399 2.545065 2.666923 2.816936 2.839158
+ left_risk <- which(x  < lower_bound) # Finding values that are in the upper bound risk
+ x[left_risk] # values at lower risk
 [1] -1.765749 -1.867407 -1.994157 -1.854737 -1.914652 -1.759334 -1.855517
 [8] -1.753957 -1.914475 -1.836435 -1.967803 -1.796362 -1.717457 -1.598806
```

FIGURE 1.9: Finding numbers at Risk.

Values at Upper (Right) Risk :

```
  [1] 2.710283 2.764196 2.705285 2.444434 2.755979 2.878734 2.711472 2.722598
  [9] 2.850662 2.882673 2.563105 2.680986 2.776878 2.963367 2.644889 2.612474
 [17] 2.937783 2.577454 2.708086 2.786869 2.612106 2.468001 2.704758 2.639299
 [25] 2.572904 2.624294 2.791844 2.703460 2.924236 2.915694 2.769651 2.387021
 [33] 2.786046 2.777726 2.875089 2.828911 2.760081 2.577049 2.795525 2.654686
 [41] 2.916622 2.979905 2.681292 2.915171 2.885494 2.407985 2.924766 2.494888
 [49] 2.951954 2.717667 2.403572 2.706490 2.950434 2.801669 2.904595 2.647996
 [57] 2.491872 2.461113 2.470870 2.720070 2.670845 2.405790 2.396098 2.712014
 [65] 2.534052 2.905376 2.805240 2.775913 2.892516 2.581874 2.975419 2.752601
 [73] 2.462446 2.483323 2.824947 2.519273 2.878170 2.583506 2.368302 2.457195
 [81] 2.699575 2.921918 2.468036 2.439669 2.543590 2.443964 2.747605 2.516319
 [89] 2.475982 2.540966 2.811097 2.366473 2.981075 2.759855 2.801299 2.983749
 [97] 2.554263 2.752032 2.930696 2.831275 2.402982 2.715588 2.994668 2.947001
[105] 2.384612 2.699250 2.518581 2.774055 2.879611 2.972586 2.657859 2.567172
[113] 2.870039 2.690738 2.606584 2.915017 2.479645 2.873769 2.711334 2.772219
[121] 2.863399 2.545065 2.666923 2.816936 2.839158
```

Values at Lower (Left) Risk :

```
 [1] -1.765749 -1.867407 -1.994157 -1.854737 -1.914652 -1.759334 -1.855517
 [8] -1.753957 -1.914475 -1.836435 -1.967803 -1.796362 -1.717457 -1.598806
[15] -1.716179 -1.969157 -1.711697 -1.614085 -1.962779 -1.599648 -1.842350
[22] -1.750447 -1.981351 -1.810863 -1.778608 -1.777863 -1.658310 -1.951087
[29] -1.601083 -1.877644 -1.587512 -1.757641 -1.841453 -1.853729 -1.994894
[36] -1.772910 -1.948714 -1.596507 -1.845324 -1.606202 -1.632101 -1.697119
[43] -1.988012 -1.743197 -1.578207 -1.899245 -1.927300 -1.976583 -1.890460
[50] -1.966780 -1.760059 -1.786029 -1.847545 -1.613207 -1.679485 -1.996526
[57] -1.858736 -1.711156 -1.560541 -1.935681 -1.563160 -1.899514 -1.934948
[64] -1.758313 -1.922799 -1.711750 -1.866996 -1.632223 -1.778486 -1.736673
[71] -1.593370 -1.723640 -1.680687 -1.999551 -1.761986 -1.920392 -1.599468
[78] -1.831059 -1.648741 -1.940659 -1.889629 -1.768906 -1.992605 -1.627663
[85] -1.747392 -1.646748 -1.757193 -1.878930
```