Lecture 1 supplement: January 12, 2018

The schema

_course_lib_r/_course_OF_xyz.r

_course_lib_r/__commands_Lecture-01.txt

on accuracy and precision

fg-asym-pi_darts_11005948

fg-asym-pi_BPP_numTerms

fg-asym-pi_BPP_tolR

fg_asym_pi_darts_cntProbe

fg_asym_pi_darts_needles_cntProbe

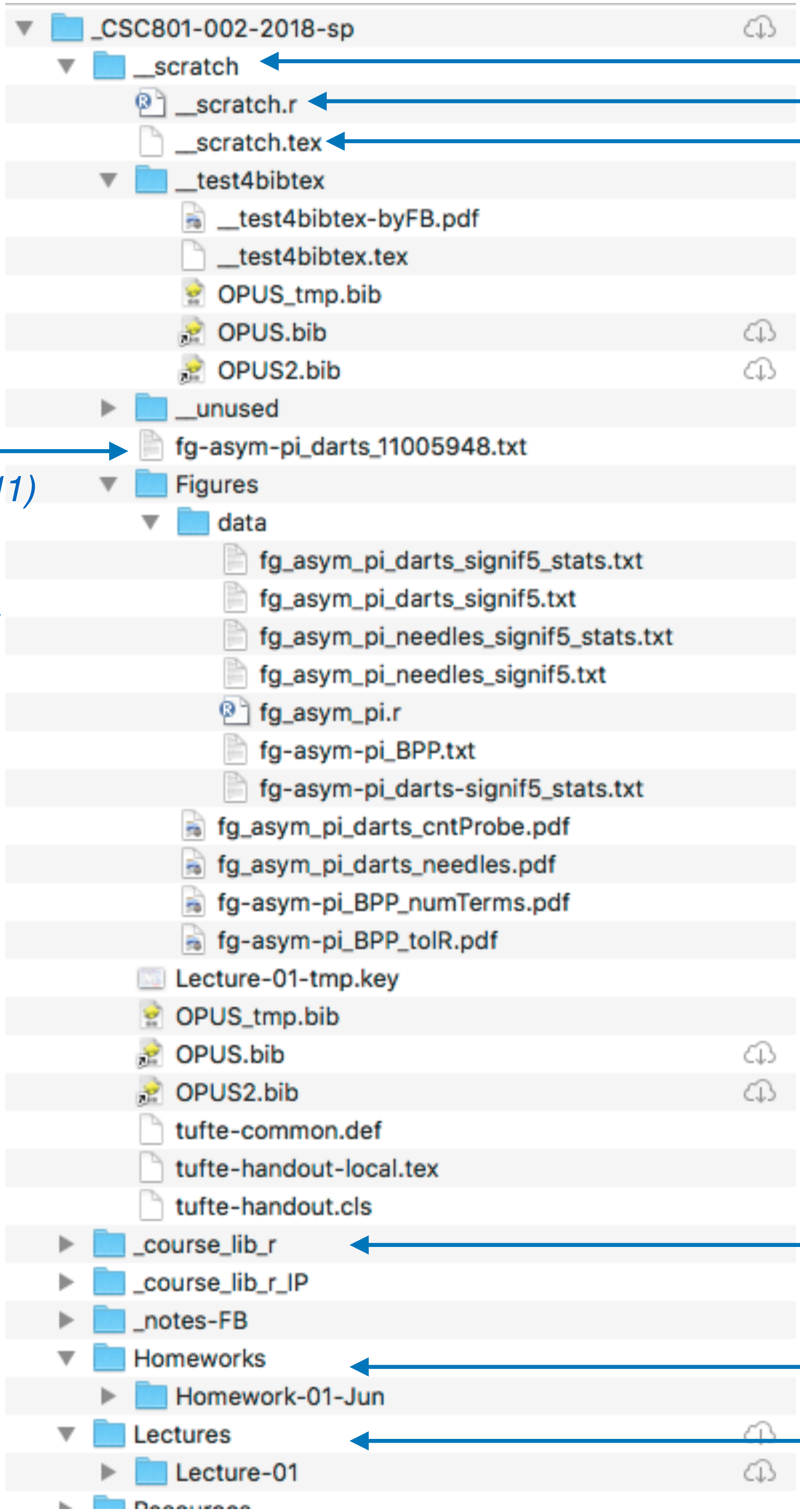
The function 'wild(x)' as part of the R-package DEoptim distribution!

Plotting results of the experiment: DEoptim-vs-walkX

../_course_lib_R/_course_wrap_xyz_tmp.R (read the code first) ←

Homework details about tabulating results in latex ←

The schema



```
▼ _CSC801-002-2018-sp
  ▼ __scratch
      __scratch.r
      __scratch.tex
    ▼ __test4bibtex
        __test4bibtex-byFB.pdf
        __test4bibtex.tex
        OPUS_tmp.bib
        OPUS.bib
        OPUS2.bib
    ▶ __unused
      fg-asym-pi_darts_11005948.txt
    ▼ Figures
      ▼ data
          fg_asym_pi_darts_signif5_stats.txt
          fg_asym_pi_darts_signif5.txt
          fg_asym_pi_needles_signif5_stats.txt
          fg_asym_pi_needles_signif5.txt
          fg_asym_pi.r
          fg-asym-pi_BPP.txt
          fg-asym-pi_darts-signif5_stats.txt
        fg_asym_pi_darts_cntProbe.pdf
        fg_asym_pi_darts_needles.pdf
        fg-asym-pi_BPP_numTerms.pdf
        fg-asym-pi_BPP_tolR.pdf
      Lecture-01-tmp.key
      OPUS_tmp.bib
      OPUS.bib
      OPUS2.bib
      tufte-common.def
      tufte-handout-local.tex
      tufte-handout.cls
  ▶ _course_lib_r
  ▶ _course_lib_r_IP
  ▶ _notes-FB
  ▼ Homeworks
    ▶ Homework-01-Jun
  ▼ Lectures
    ▶ Lecture-01
  ▶ Resources
```

All work begins here (__scratch)
On completions, items are moved to

local Figures/

_course_lib_r/

Homeworks/

Lectures/

and by April 30, 2018,
full content is jointly edited for

Manuscript/

Still running (since Jan 11) on server at home All results are visible on iCloud, we shall view latest results in the next slide!

The issue of file names, so they are compatible with rules under LaTex (no "." except preceding the extension string) and with names of functions in R (no "-"):

the root name

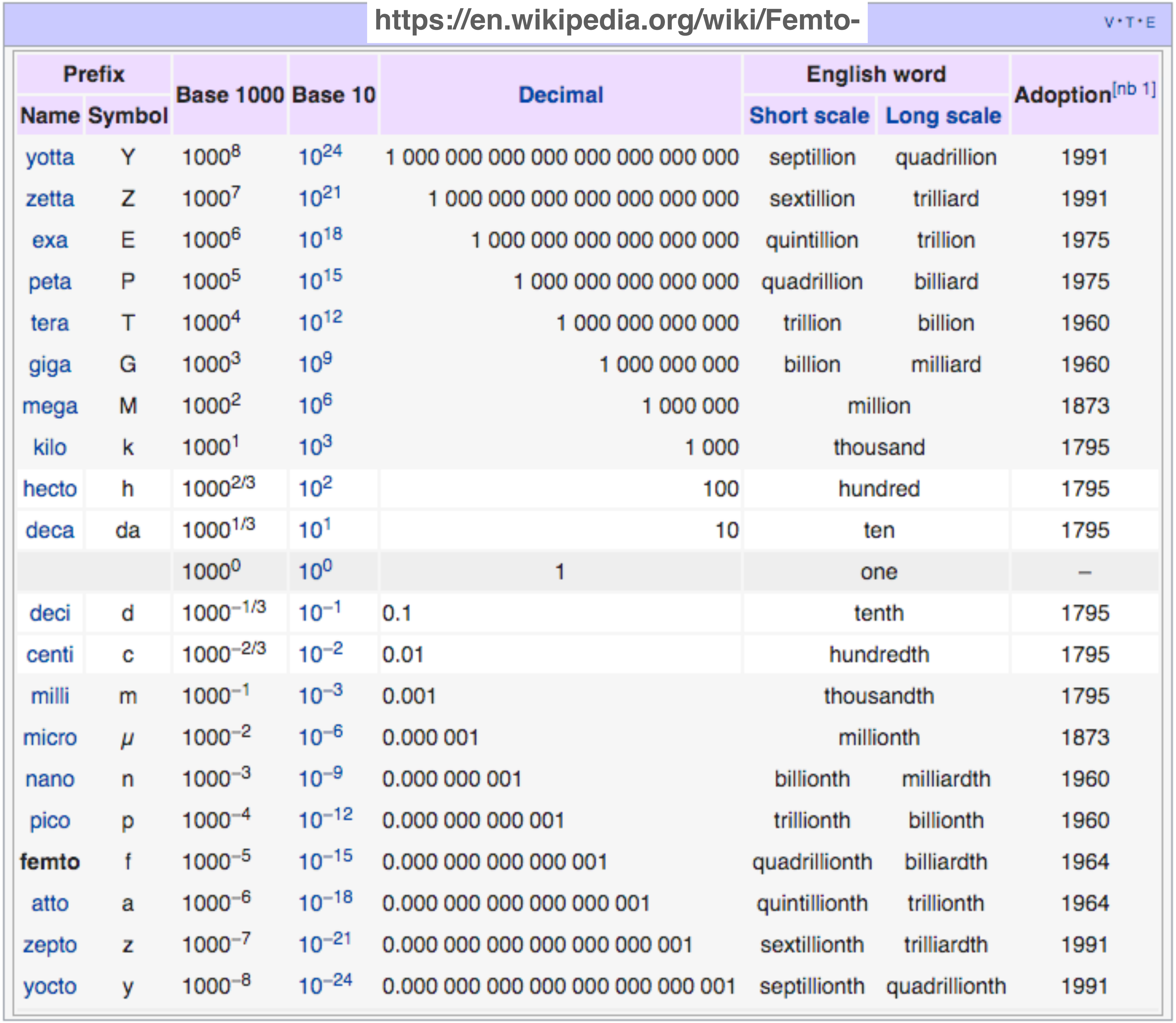
fg-asym-pi_darts_

will be changed to

fg_asym_pi_

fg_asym_pi_darts_11005948

```
# fileName  = fg-asym-pi_darts.txt
# userId    = brglez
# nodeName  = iMac-triangle.local
# sysName   = Darwin
# date      = 20180111
# timeStamp = 11005948
#
# command   = xPer_pi_darts(...)
#
sampleId     seedInit    signifDigits piMC  tolRadius    error  isCensored  cntProbe     runtime
1     1215   2     3.1   0.005  0.004 0     103   0.034
2     989979358    2     3.1    0.005 0.0013 0     14    0.001
3     276687189    2     3.1    0.005 0.0013 0     42    0.006
4     327447555    2     3.1    0.005 -0.0043      0     102    0.035
5     30741964     2     3.1    0.005 0.0013 0     196   0.12
6     577772815    2     3.1    0.005 0.0013 0     126   0.052
....
100   139169247    2     3.1    0.005 0.0013 0     14    0
101   69569211     3     3.14   5e-04 -0.000488    0     163    0.085
102   272477547    3     3.14   5e-04 -0.000488    0     163    0.084
103   532776928    3     3.14   5e-04 -4.01e-05    0     219    0.152
104   6342438      3     3.14   5e-04 0.000108     0     494    0.786
105   911482030    3     3.14   5e-04 -4.01e-05    0     219    0.15
106   786263289    3     3.14   5e-04 0.000469     0     718    1.663
107   785294913    3     3.14   5e-04 -0.000488    0     326    0.338
....
398   619462206    5     3.1416 5e-06 2.6676e-07   0     452    0.651
399   198947760    5     3.1416 5e-06 2.6676e-07   0     1356   5.893
400   420963582    5     3.1416 5e-06 -1.3015e-06  0     11286  402.3
401   587473055    6     3.14159      5e-07 2.66764e-07  0      452   0.653
402   139564609    6     3.14159      5e-07 -1.13518e-07 0      23271 1712.614
403   44112420     6     3.14159      5e-07 -3.17944e-07 0      30270 2898.621
404   314722844    6     3.14159      5e-07 -1.59002e-07 0      41570 5470.893
....
424   817224753    6     3.14159      5e-07 2.66764e-07  0      3164  31.719
425   720329768    6     3.14159      5e-07 2.66764e-07  0      6780  145.231
426   437070989    6     3.14159      5e-07 8.29002e-08  0      48131 7338.967
427   40503715     6     3.14159      5e-07 2.66764e-07  0      6328  126.374
428   287157421    6     3.14159      5e-07 -1.10084e-08 0      31859 3213.505
```



https://en.wikipedia.org/wiki/Femto-

Prefix		Base 1000	Base 10	Decimal	English word		Adoption[nb 1]
Name	Symbol				Short scale	Long scale	
yotta	Y	1000^8	10^{24}	1 000 000 000 000 000 000 000 000	septillion	quadrillion	1991
zetta	Z	1000^7	10^{21}	1 000 000 000 000 000 000 000	sextillion	trilliard	1991
exa	E	1000^6	10^{18}	1 000 000 000 000 000 000	quintillion	trillion	1975
peta	P	1000^5	10^{15}	1 000 000 000 000 000	quadrillion	billiard	1975
tera	T	1000^4	10^{12}	1 000 000 000 000	trillion	billion	1960
giga	G	1000^3	10^9	1 000 000 000	billion	milliard	1960
mega	M	1000^2	10^6	1 000 000	million		1873
kilo	k	1000^1	10^3	1 000	thousand		1795
hecto	h	$1000^{2/3}$	10^2	100	hundred		1795
deca	da	$1000^{1/3}$	10^1	10	ten		1795
		1000^0	10^0	1	one		–
deci	d	$1000^{-1/3}$	10^{-1}	0.1	tenth		1795
centi	c	$1000^{-2/3}$	10^{-2}	0.01	hundredth		1795
milli	m	1000^{-1}	10^{-3}	0.001	thousandth		1795
micro	μ	1000^{-2}	10^{-6}	0.000 001	millionth		1873
nano	n	1000^{-3}	10^{-9}	0.000 000 001	billionth	milliardth	1960
pico	p	1000^{-4}	10^{-12}	0.000 000 000 001	trillionth	billionth	1960
femto	f	1000^{-5}	10^{-15}	0.000 000 000 000 001	quadrillionth	billiardth	1964
atto	a	1000^{-6}	10^{-18}	0.000 000 000 000 000 001	quintillionth	trillionth	1964
zepto	z	1000^{-7}	10^{-21}	0.000 000 000 000 000 000 001	sextillionth	trilliardth	1991
yocto	y	1000^{-8}	10^{-24}	0.000 000 000 000 000 000 000 001	septillionth	quadrillionth	1991

1902:
Marie and Pierre Curie separated one-tenth of a gram of radium chloride from one ton of pitchblende:
1/(1000*1000/10)
10^(-5) = 0.00001

Computerized instruments today:

The HIV-1 virus weighs about 1 x 10^(-15) g (grams) or 1 fg

A proton has a diameter of about 1.6 to 1.7 femtometres

Computational limit of a 64-bit CPU:
9,223,372,036,854,775,807 (about 9.22*10^(18)) is equal to 2^(63) - 1, and as such is the largest number which can fit into a signed (two's complement) 64-bit integer on a computer.

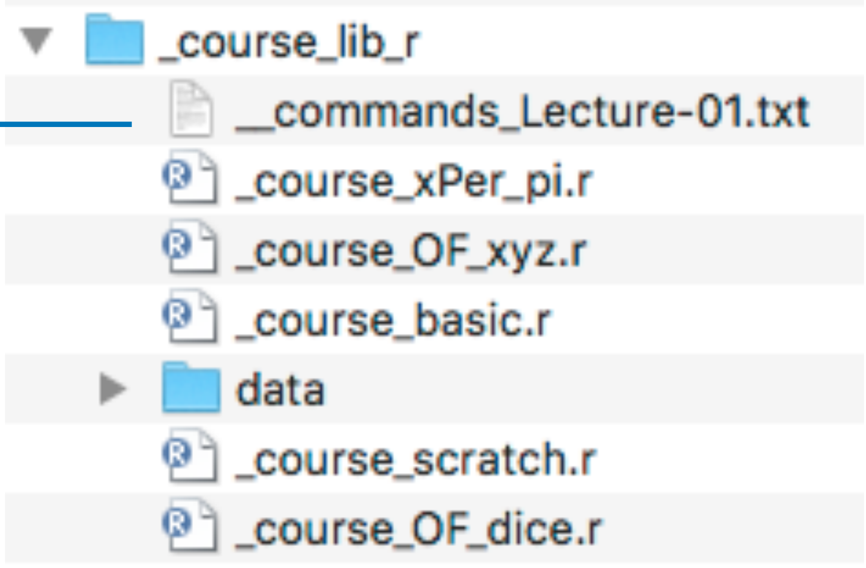
After centuries of increasingly precise measurements, in 1975 the speed of light was known to be 299792458 m/s with a measurement uncertainty of 4 parts per billion : 1 light-year = **9460730472580800 metres (exactly)**

signifDigits(9460730472580800) = 14

log2(9460730472580800) = 53.071

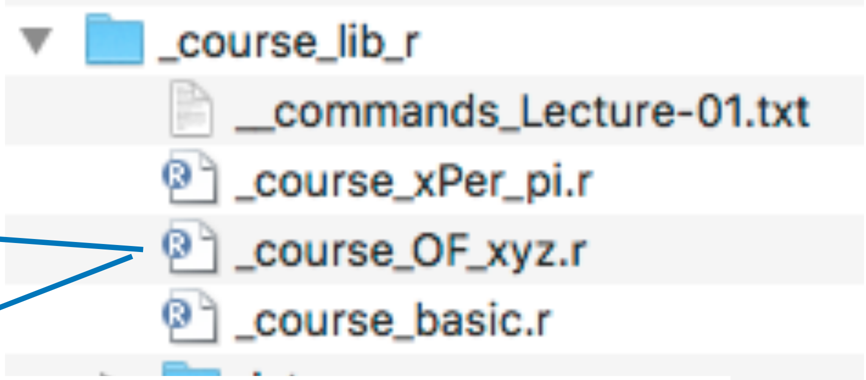
See also
https://en.wikipedia.org/wiki/Orders_of_magnitude_(mass)

_course_lib_r/__commands_Lecture-01.txt



```
> getwd()
[1] "/Users/brglez"

> setwd("/Users/brglez/Desktop/__work/_CSC801-002-2018-sp/__scratch")

> getwd()
[1] "/Users/brglez/Desktop/__work/_CSC801-002-2018-sp/__scratch"

######## testing the library _course_basic.r
> source("../_course_lib_r/_course_basic.r")  ; signifDigits(320.001) ; signifDigits(320) ;
signifDigits(3.142)

> source("../_course_lib_r/_course_basic.r")  ; file_header("fileTest.txt")

> source("../_course_lib_r/_course_basic.r")  ; file_asym_stats("../_course_lib_r/data/fg-asym-
pi_needles_test1.txt",  cntProbeLmt=1000, xName="signifDigits", yNames=c("cntProbe", "runtime"))

######## testing the library _course_OF_xyz.r(continuous Objective Functions)
> source("../_course_lib_r/_course_OF_xyz.r") ; pi_newton(2) ;  pi_newton(3) ;  pi_newton(4)  ;
pi_newton(5)

> source("../_course_lib_r/_course_OF_xyz.r") ; pi_BPP(3) ; pi_BPP(4) ; pi_BPP(5) ; pi_BPP(15)

> source("../_course_lib_r/_course_OF_xyz.r") ; pi_needles(1000)

> source("../_course_lib_r/_course_OF_xyz.r") ; pi_darts(1000)

######## testing the library _course_xPer_pi.r
> source("../_course_lib_r/_course_xPer_pi.r") ; xPer_pi_BPP(3:17)

> source("../_course_lib_r/_course_xPer_pi.r") ; xPer_pi_needles(10,1215,1000,1, 2:4)

> source("../_course_lib_r/_course_xPer_pi.r") ; xPer_pi_darts(10,1215,1000,2:4)


> source("fg_asym_pi.r") ; fg_asym_pi_darts("fg_asym_pi_darts_signif5.txt")

> source("fg_asym_pi.r") ; fg_asym_pi_darts("fg_asym_pi_needles_signif5.txt")

> source("fg_asym_pi.r") ; fg_asym_pi_darts_needles("fg_asym_pi_darts_signif5_stats.txt",
"fg_asym_pi_needles_signif5_stats.txt")
```

_course_lib_r/_course_OF_xyz.r

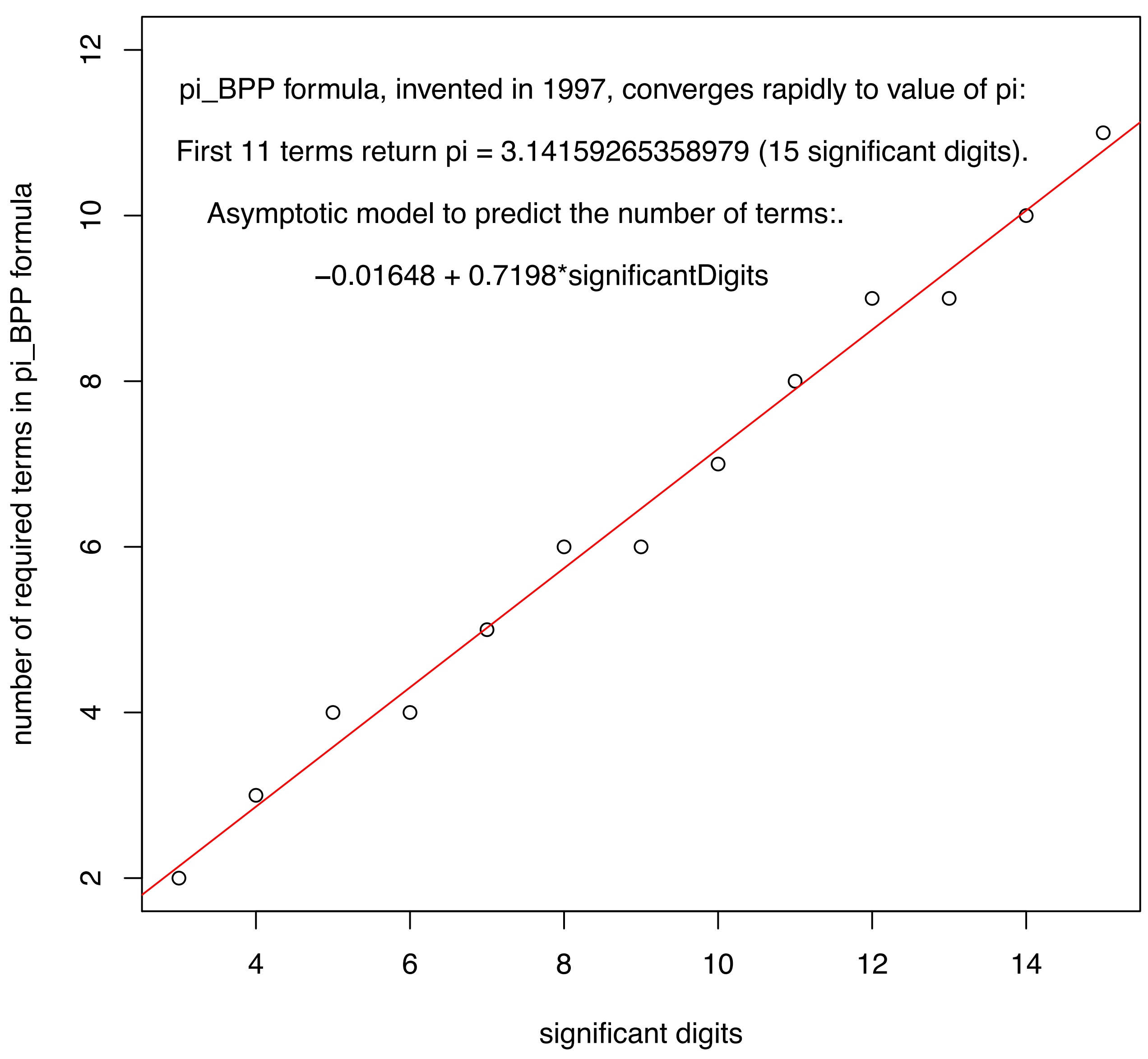


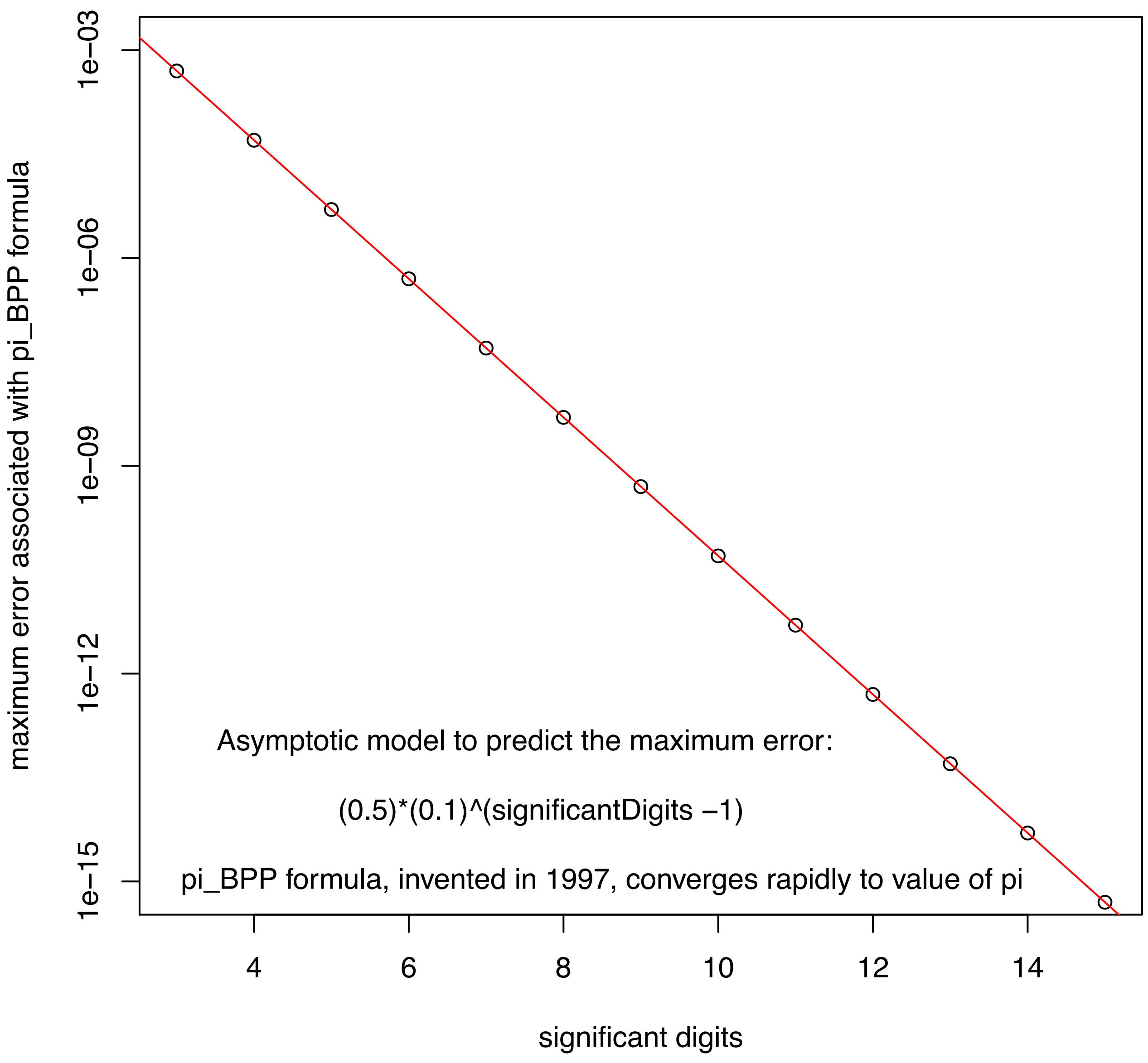
_course_lib_r
_commands_Lecture-01.txt
_course_xPer_pi.r
_course_OF_xyz.r
_course_basic.r

```
pi_BPP = function(signifDigits=3)
{
  # Copyright 2018, Franc Brglez
  # http://mathworld.wolfram.com/PiFormulas.html, formula (30)
  t = NULL ; tMax = 15
  for (i in seq_len(tMax)) {
    n = i - 1
    t[i] = (4/(8*n + 1) - 2/(8*n + 4) - 1/(8*n + 5) -
            1/(8*n + 6))*(1/16)^n
  }
  piHat = 0 ; numTerms = 0 ; isCensored = TRUE
  tolR = 5/10^(signifDigits+1)      # toleranceRadius
  LB = pi - tolR
  UB = pi + tolR
  for (numTerms in seq_len(length(t))) {
    piHat = piHat + t[numTerms]
    if (piHat > LB && piHat < UB) {
      isCensored = FALSE ; break
    }
  }
  error = piHat - pi
  options(digits=signifDigits)
  piHat =signif(piHat, signifDigits)
  return(list(numTerms=numTerms, piHat=piHat, tolR=tolR,
              error=error, isCensored=isCensored))

  # VIGNETTE
  # source("../_course_lib_r/_course_OF_xyz.r") ; pi_BPP(15)
  # $numTerms
  # [1] 11
  # $piHat
  # [1] 3.14159265358979
  # $tolR
  # [1] 5e-16
  # $error
  # [1] 0
  # $isCensored
  # [1] FALSE

} # pi_BPP
```

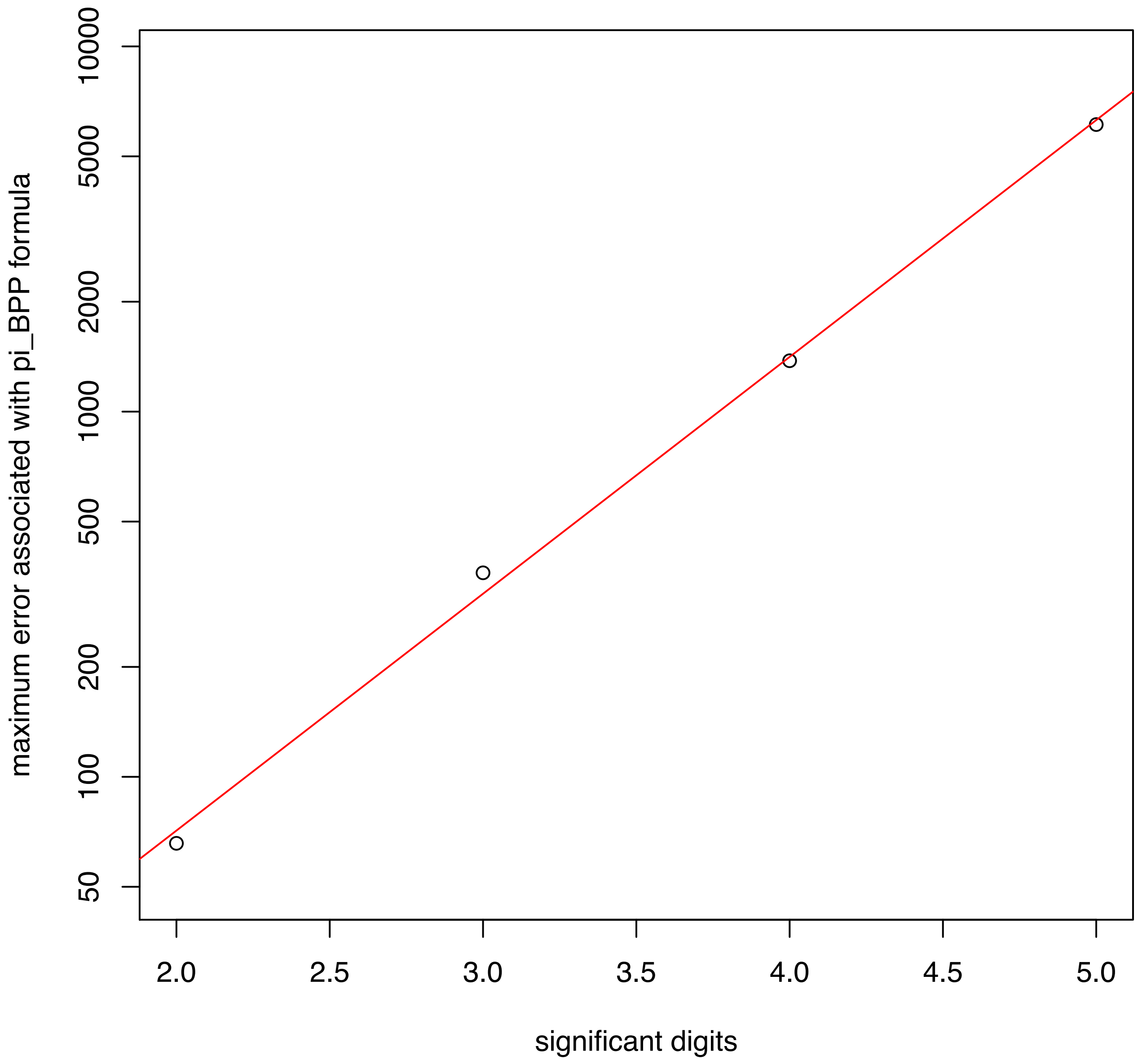
```
pi_darts = function(cntProbeLmt=10, signifDigits=3)
{
  # Copyright 2018, Franc Brglez
  # MC experiments with darts to approximate pi: a unit
diameter circle is embedded into a unit square
  tolR = 5/10^(signifDigits+1)
  piLB = pi - tolR
  piUB = pi + tolR
  cntProbe = 0
  isCensored  = TRUE

  while (cntProbe < cntProbeLmt) {
    cntProbe = cntProbe + 1
    hits = 0
    for (i in seq_len(cntProbe)) {
      x = runif(1,0,1)
      y = runif(1,0,1)
      if (x^2 + y^2 <= 1) {
        hits = hits + 1
      }
    }
    piMC = (4*hits)/cntProbe
    error = signif(piMC - pi, signifDigits)
    if ( (piMC >= piLB && piMC <= piUB) ) {
      isCensored  = FALSE
      break
    }
    if (cntProbe >= cntProbeLmt) {isCensored  = TRUE}
  }
  #piMC = signif(piMC, signifDigits)

  return(list(piMC=piMC, tolR=tolR,  error=error,
              isCensored=isCensored, cntProbe=cntProbe))
} # pi_darts
```

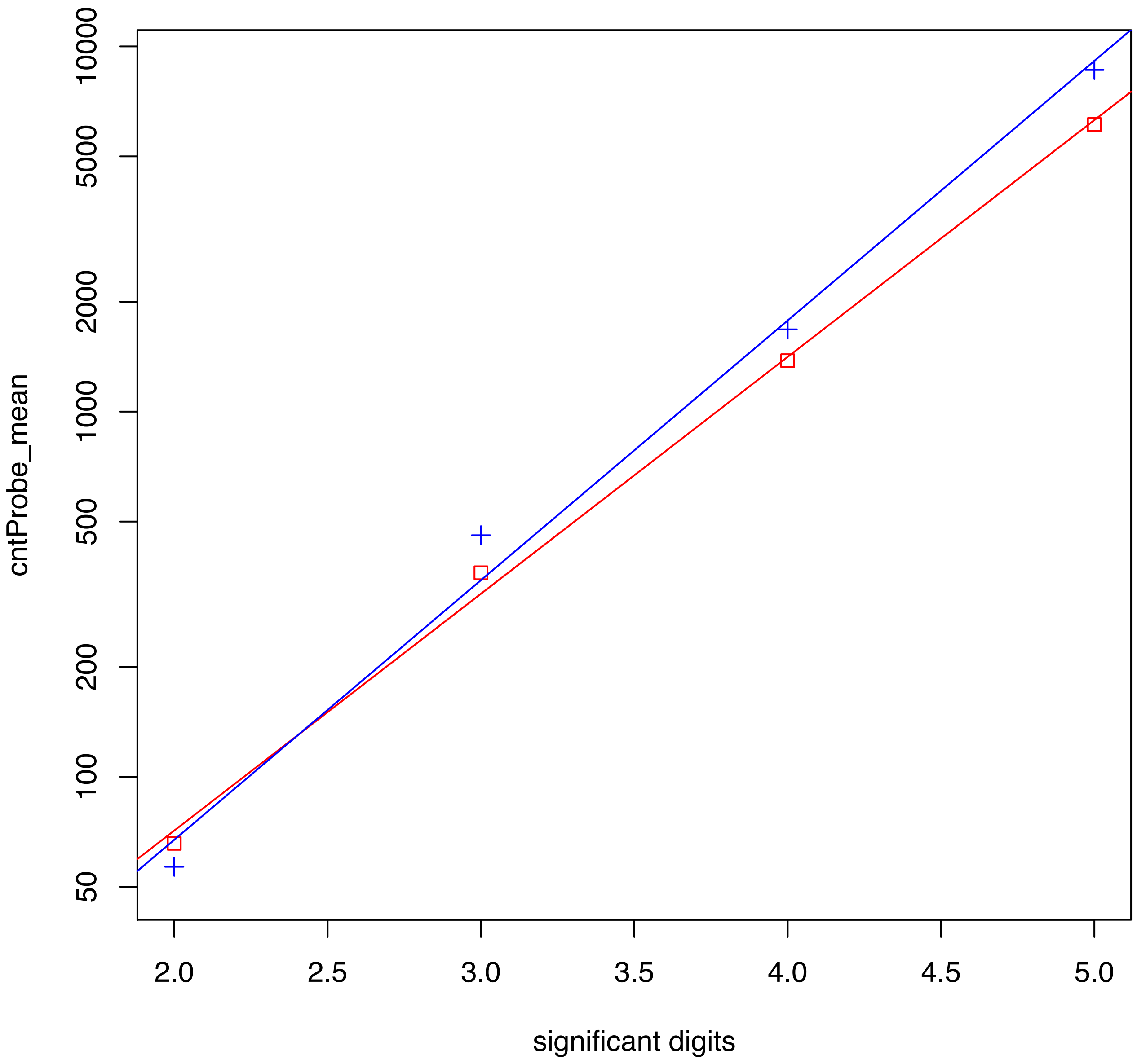
fg_asym_pi_BPP_numTerms and fg_asym_pi_BPP_tolR





fg_asym_pi_darts_cntProbe and fg_asym_pi_darts_needles_cntProbe





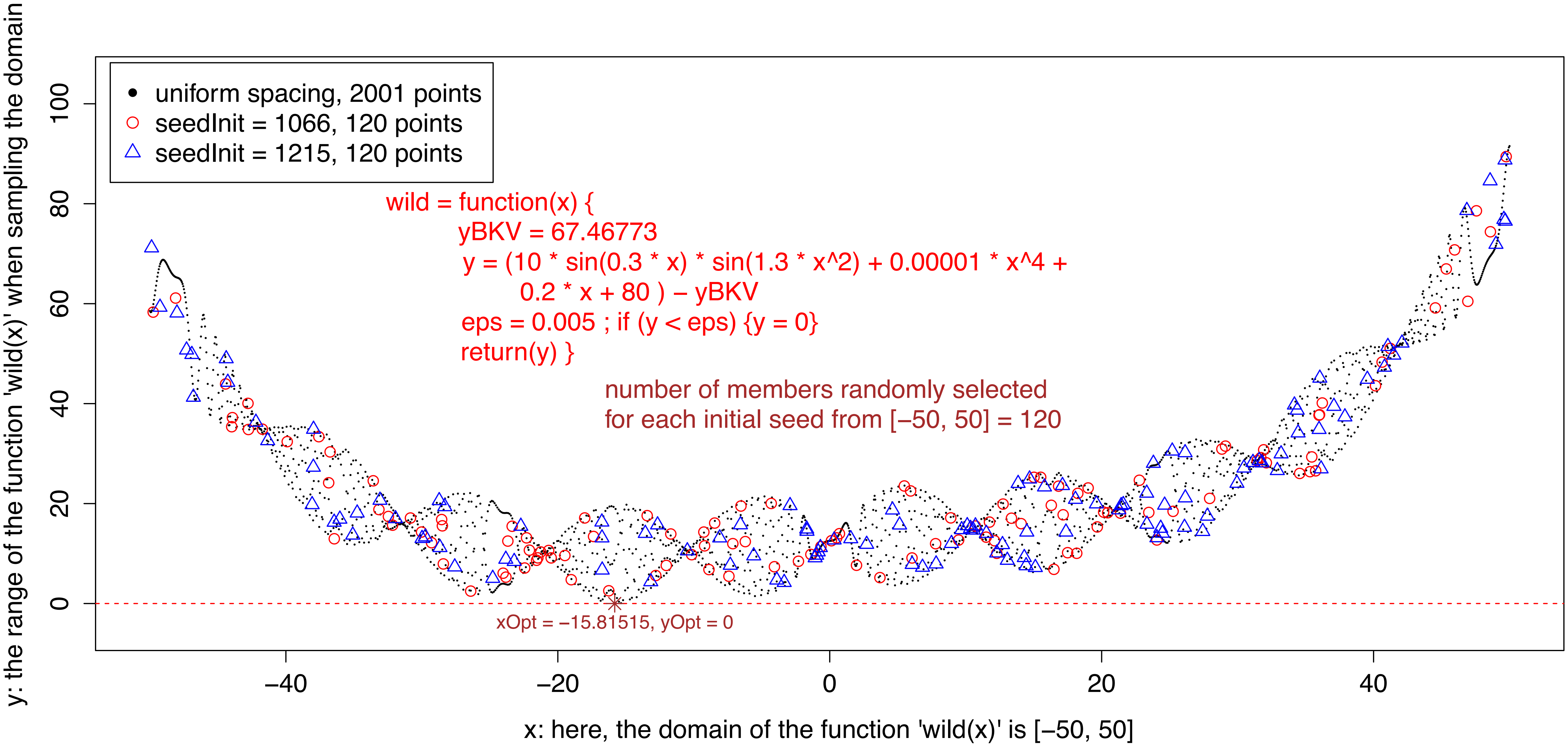
The function 'wild(x)' as part of the R-package DEoptim distribution!

Message:

finding the single optimum solution with 7 significant digits cannot be left to chance alone,

xOpt = -15.81515, yOpt = 0

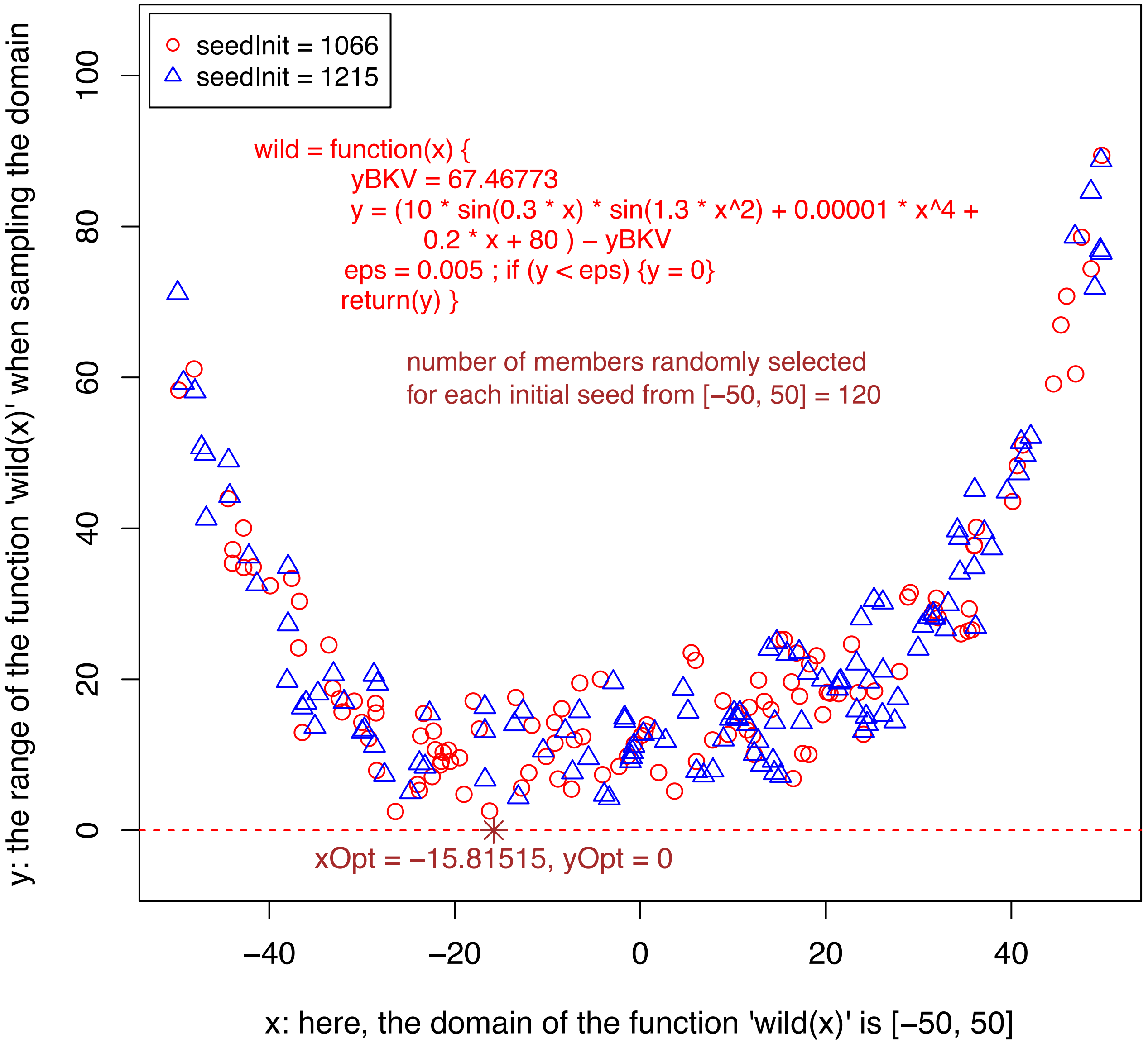
-- it would take too many random samples and too much runtime!



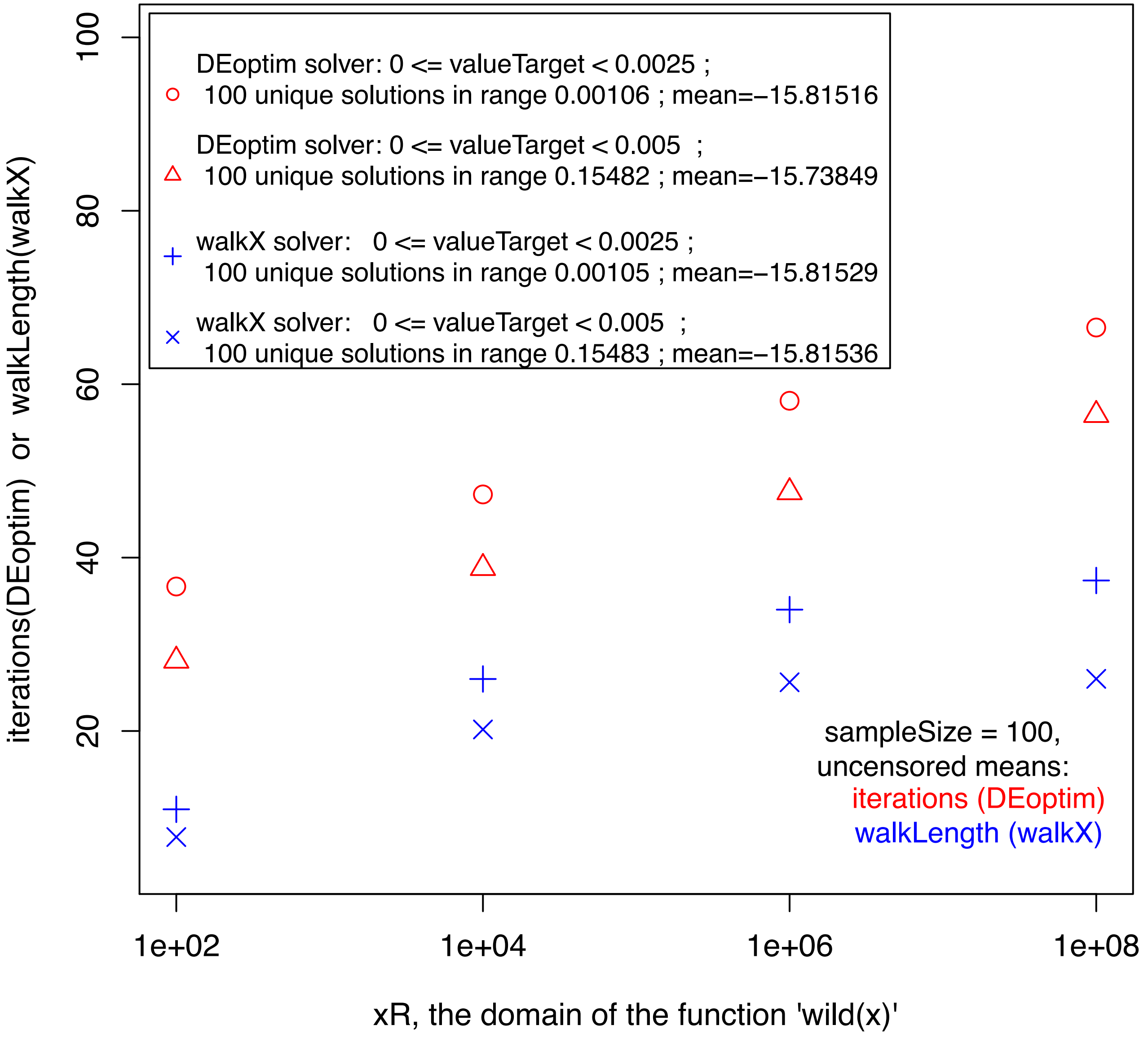
```
wild = function(x) {
        yBKV = 67.46773
        y = (10 * sin(0.3 * x) * sin(1.3 * x^2) + 0.00001 * x^4 +
               0.2 * x + 80 ) – yBKV
        eps = 0.005 ; if (y < eps) {y = 0}
        return(y) }
```

number of members randomly selected
for each initial seed from [–50, 50] = 120

Plotting results of the experiment: DEoptim-vs-walkX



```
wild = function(x) {
          yBKV = 67.46773
          y = (10 * sin(0.3 * x) * sin(1.3 * x^2) + 0.00001 * x^4 +
                0.2 * x + 80 ) − yBKV
         eps = 0.005 ; if (y < eps) {y = 0}
         return(y) }
```



Needed:

comparisions of function evaluation counts

(currently, DEoptim reports these counts incorrectly)

../_course_lib_R/_course_wrap_xyz_tmp.R (read the code first!!!)

… copy and paste the commands from Lecture-01-b-commands.txt

```
################ within the temporary library ../_course_lib_R/_course_wrap_xyz_tmp.R

source("../_course_lib_R/_course_wrap_xyz_tmp.R") ; test_of_wild(c(-15.51009, -16.26599, -15.66054)) ; test_of_wild()

                                                    out = wrap.DEoptim(popSize=64,  iterLmt=100, OFpar=1) ; summary(out) ;# print(out)
                                                    ...
                                                    out = wrap.DEoptim(popSize=64,  iterLmt=200, OFpar=2) ; summary(out) ;# print(out)
                                                    ...
                                                    out = wrap.DEoptim(popSize=128, iterLmt=200, OFpar=2) ; summary(out) ;# print(out)
                                                    ...
                                                    out = wrap.DEoptim(popSize=128, iterLmt=200, OFpar=3); summary(out) ;# print(out)
                                                    ...
                                                    out = wrap.DEoptim(popSize=256, iterLmt=800, OFpar=3) ; summary(out) ;# print(out)
```

… implies runs under different random seeds (see the code)
tabulate results in a small latex table, using values from each summary:
values reported as “generations” are equivalent to “iterations”
values reported as “fn evaluated” should be ignored.
values reported as “best member” are equivalent to xBest in the table
values reported as “best value > 0” imply isCensored=TRUE and should be ignored

xAsym=OFpar	seedInit	iterLmt	popSize	isCensored	generations_min	xBest
1	xxx	xxx	xxx	FALSE	xxx	xxx
2	xxx	xxx	xxx	FALSE	xxx	xxx xxx
3	xxx	xxx	xxx	FALSE	xxx	xxx xxx xxx

In latex, write
(1) a paragraph of conclusions about these experiments and observations
(2) a paragraph that relates conclusions in (1) to recent “significant digit” experiments with pi

See also a few sample runs on the next page

../_course_lib_R/_course_wrap_xyz_tmp.R

```
> source("../_course_lib_R/_course_wrap_xyz_tmp.R") ; test_of_wild(c(-15.51009, -16.26599, -15.66054)) ;
test_of_wild()
<hash> containing 2 key-value pair(s).
  wild.BKV : 67.46774
  wild.tolR : 0.005

.. from test_of_wild: testing objective function with name = wild
<hash> containing 2 key-value pair(s).
  wild.BKV : 67.46774
  wild.tolR : 0.005
             x         y
[1,] -15.51009 0.2708414
[2,] -16.26599 0.2708414
[3,] -15.66054 0.2708414
[1] 0.2708414

.. from test_of_wild: testing objective function with name = wild
<hash> containing 2 key-value pair(s).
  wild.BKV : 67.46774
  wild.tolR : 0.005
             x y
[1,] -15.81559 0
[2,] -15.81491 0
[3,] -15.81529 0
[1] 0
>
```

../_course_lib_R/_course_wrap_xyz_tmp.R

```
> out = wrap.DEoptim(popSize=64,  iterLmt=100, OFpar=1) ;
summary(out) ;# print(out)

seedInit = 5456
lowerBnd = -50
upperBnd = 50
iterLmt  = 100
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.81521
best value    :  0
after         :  56 generations
fn evaluated  :  114 times
*************************************
```

```
> out = wrap.DEoptim(popSize=64,  iterLmt=100, OFpar=1) ;
summary(out) ;# print(out)

seedInit = 6466
lowerBnd = -50
upperBnd = 50
iterLmt  = 100
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.81488
best value    :  0
after         :  40 generations
fn evaluated  :  82 times
*************************************
```

```
> out = wrap.DEoptim(popSize=64,  iterLmt=100, OFpar=1) ;
summary(out) ;# print(out)

seedInit = 5515
lowerBnd = -50
upperBnd = 50
iterLmt  = 100
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.81489
best value    :  0
after         :  27 generations
fn evaluated  :  56 times
*************************************
>
```

```
> out = wrap.DEoptim(popSize=64,  iterLmt=200, OFpar=2) ;
summary(out) ;# print(out)

seedInit = 5802
lowerBnd = -50 -50
upperBnd = 50 50
iterLmt  = 200
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.81481 -15.81497
best value    :  0
after         :  183 generations
fn evaluated  :  368 times
*************************************
```

```
> out = wrap.DEoptim(popSize=64,  iterLmt=200, OFpar=2) ;
summary(out) ;# print(out)

seedInit = 39
lowerBnd = -50 -50
upperBnd = 50 50
iterLmt  = 200
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.81466 -15.81576
best value    :  0.00512
after         :  200 generations
fn evaluated  :  402 times
*************************************
```

```
> out = wrap.DEoptim(popSize=64,  iterLmt=200, OFpar=2) ;
summary(out) ;# print(out)

seedInit = 4334
lowerBnd = -50 -50
upperBnd = 50 50
iterLmt  = 200
popSize  = 64

***** summary of DEoptim object *****
best member   :  -15.66143 -15.66254
best value    :  0.01261
after         :  200 generations
fn evaluated  :  402 times
*************************************
>
```

```
> out = wrap.DEoptim(popSize=128,  iterLmt=400, OFpar=3) ;
summary(out) ;# print(out)

seedInit = 8993
lowerBnd = -50 -50 -50
upperBnd = 50 50 50
iterLmt  = 400
popSize  = 128

***** summary of DEoptim object *****
best member   :  -15.66179 -15.66173 -15.81558
best value    :  0.00708
after         :  400 generations
fn evaluated  :  802 times
*************************************
```

```
> out = wrap.DEoptim(popSize=128,  iterLmt=400, OFpar=3) ;
summary(out) ;# print(out)

seedInit = 4097
lowerBnd = -50 -50 -50
upperBnd = 50 50 50
iterLmt  = 400
popSize  = 128

***** summary of DEoptim object *****
best member   :  -15.6617 -15.81495 -15.66106
best value    :  0.00809
after         :  400 generations
fn evaluated  :  802 times
*************************************
```

```
> out = wrap.DEoptim(popSize=128,  iterLmt=600, OFpar=3) ;
summary(out) ;# print(out)

seedInit = 113
lowerBnd = -50 -50 -50
upperBnd = 50 50 50
iterLmt  = 600
popSize  = 128

***** summary of DEoptim object *****
best member   :  -15.81479 -15.81506 -15.81451
best value    :  0
after         :  435 generations
fn evaluated  :  872 times
*************************************
>
```

https://en.wikipedia.org/wiki/Electrometer

An **electrometer** is an electrical instrument for measuring electric charge or electrical potential difference. There are many different types, ranging from historical handmade mechanical instruments to high-precision electronic devices. Modern electrometers based on vacuum tube or solid-state technology can be used to make voltage and charge measurements with very low leakage currents, down to 1 femtoampere. A simpler but related instrument, the electroscope, works on similar principles but only indicates the relative magnitudes of voltages or charges.

Femto- (symbol f) is a unit prefix in the metric system denoting a factor of 10^{-15} or 0.000000000000001. Adopted by the 11th General Conference on Weights and Measures, it was added in 1964 to the SI. It is derived from the Danish word femten, meaning “fifteen”.

https://en.wikipedia.org/wiki/Femto-

https://en.wikipedia.org/wiki/Orders_of_magnitude_(numbers)

Marie Courie:
One-thousand’s of 1%:
0.01/1000 = 1e-05 needed to discover “radium”

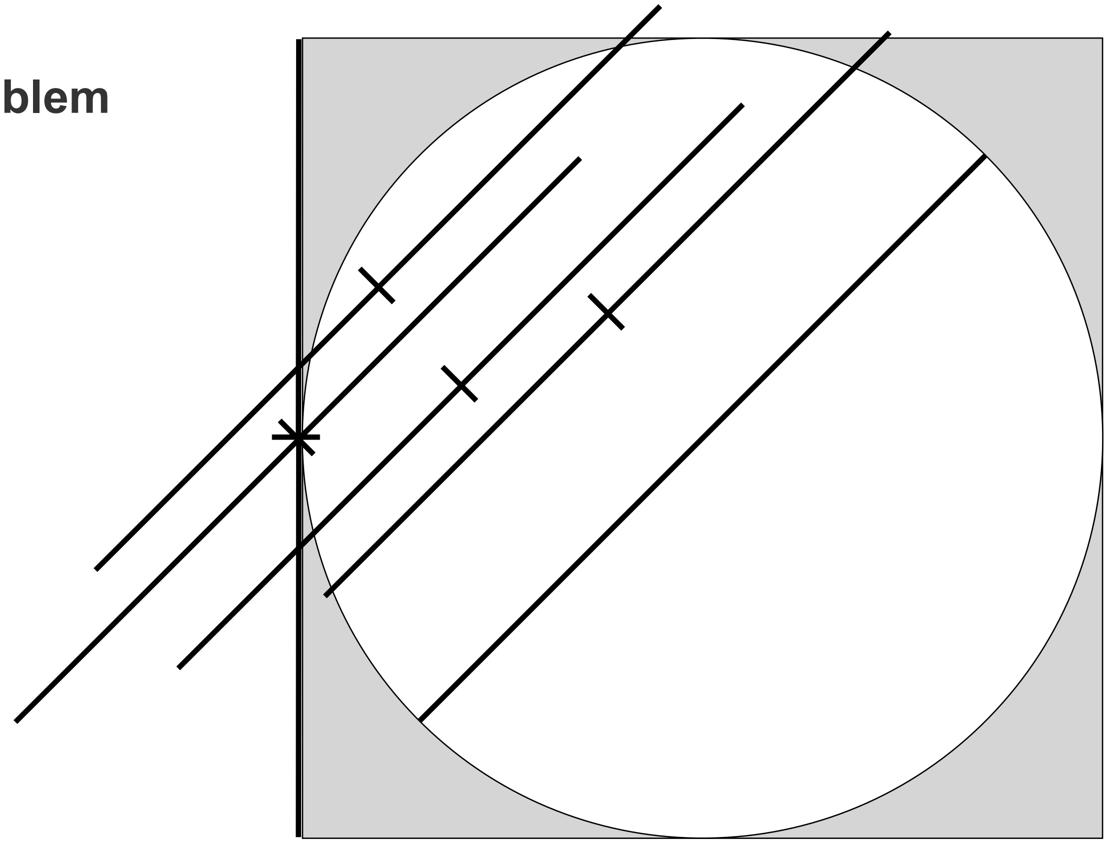
https://en.wikipedia.org/wiki/Marie_Curie
From a ton of pitchblende, one-tenth of a gram of radium chloride was separated in 1902

```
> 1/(1000*1000/10)
[1] 1e-05
>
```

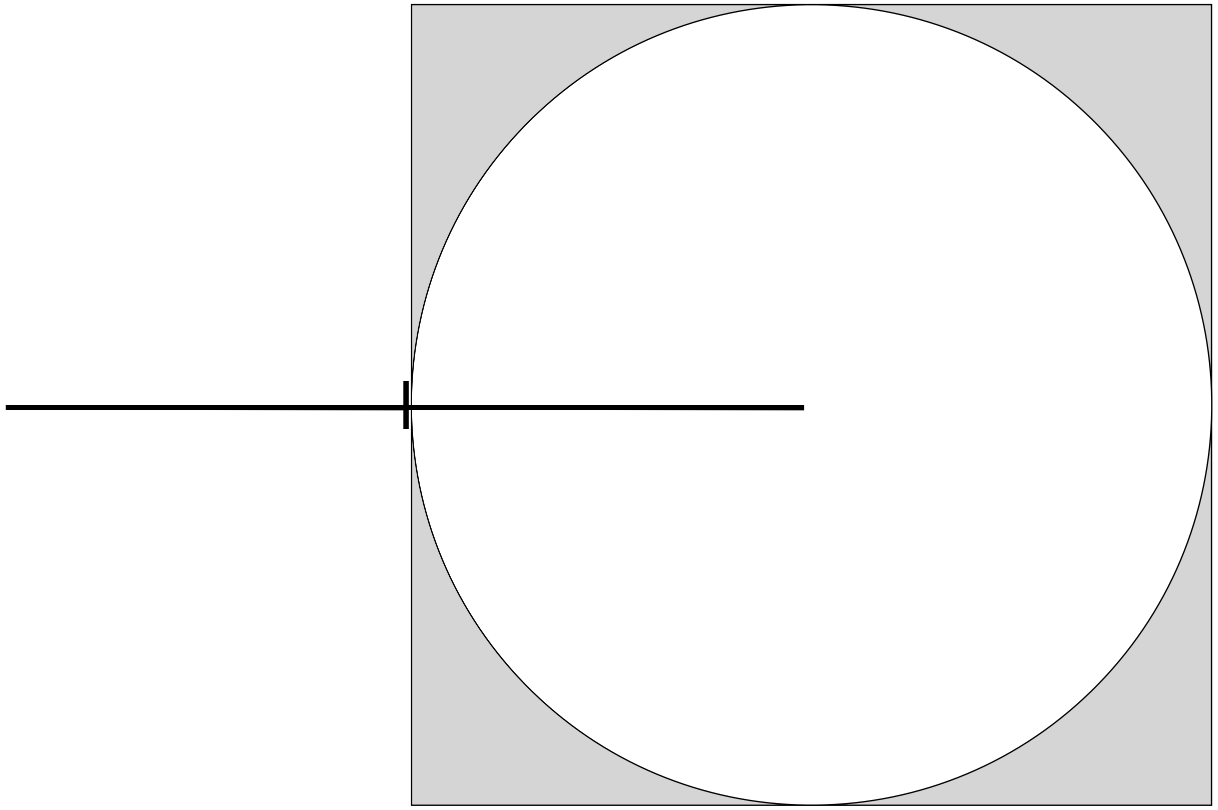
JOURNAL ARTICLE

On Laplace's Extension of the Buffon Needle Problem

Barry J. Arnow
The College Mathematics Journal
Vol. 25, No. 1 (Jan., 1994), pp. 40-43

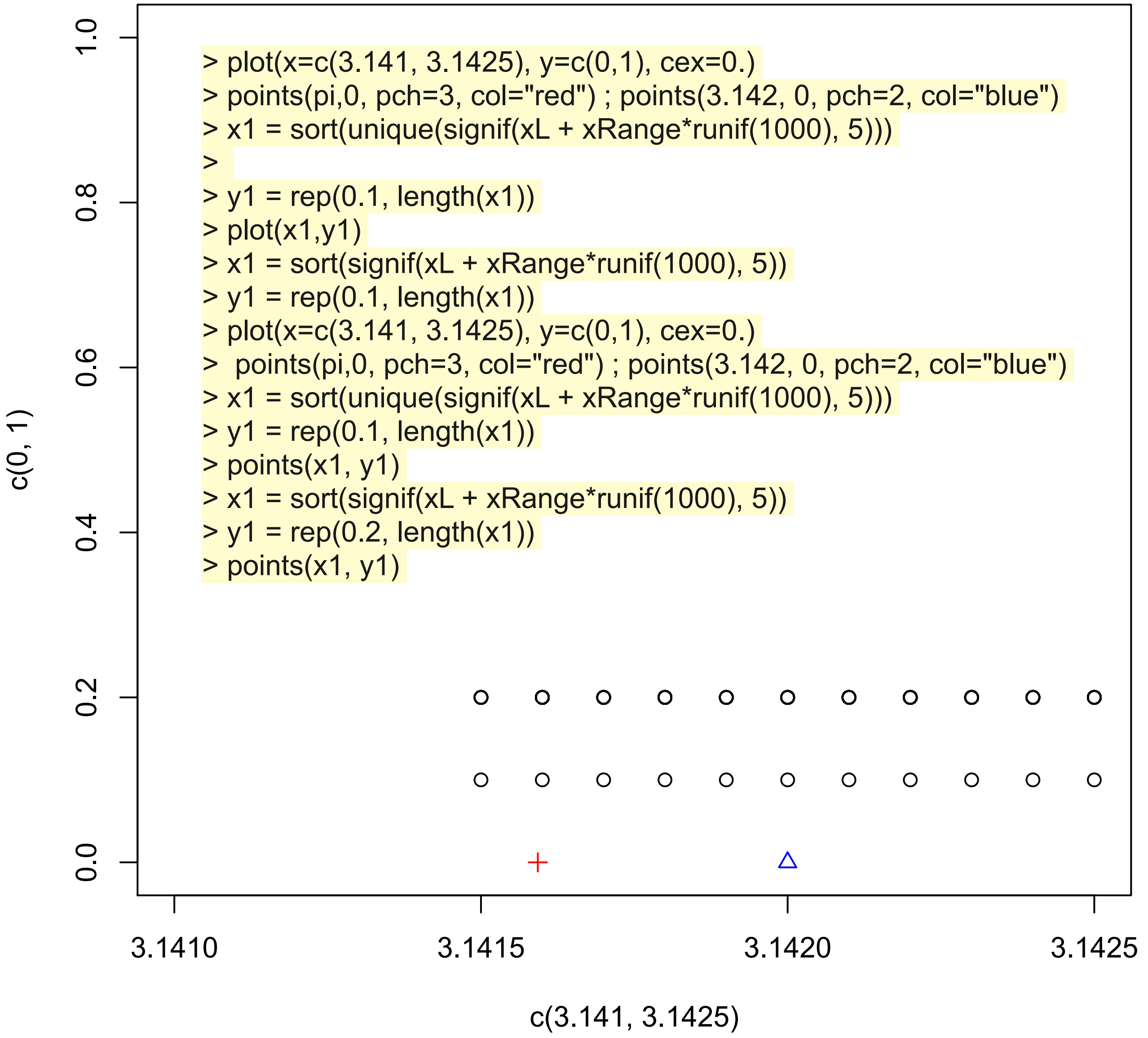


```
 [1] 3.141593
> signif(pi, 3)
[1] 3.14
> signif(pi, 4)
[1] 3.142
> signif(pi, 5)
[1] 3.1416
> signif(pi + 5e-6, 5)
[1] 3.1416
> signif(pi - 5e-6, 5)
[1] 3.1416
>

> signif(355/113 + 5*10^-6, 5)
[1] 3.1416
> signif(355/113 - 5*10^-6, 5)
[1] 3.1416
> 355/113
[1] 3.141593
> 22/7
[1] 3.142857
>  signif(22/7 + 5e-4, 3)
[1] 3.14
> signif(22/7  - 5e-4, 3)
[1] 3.14
>
```



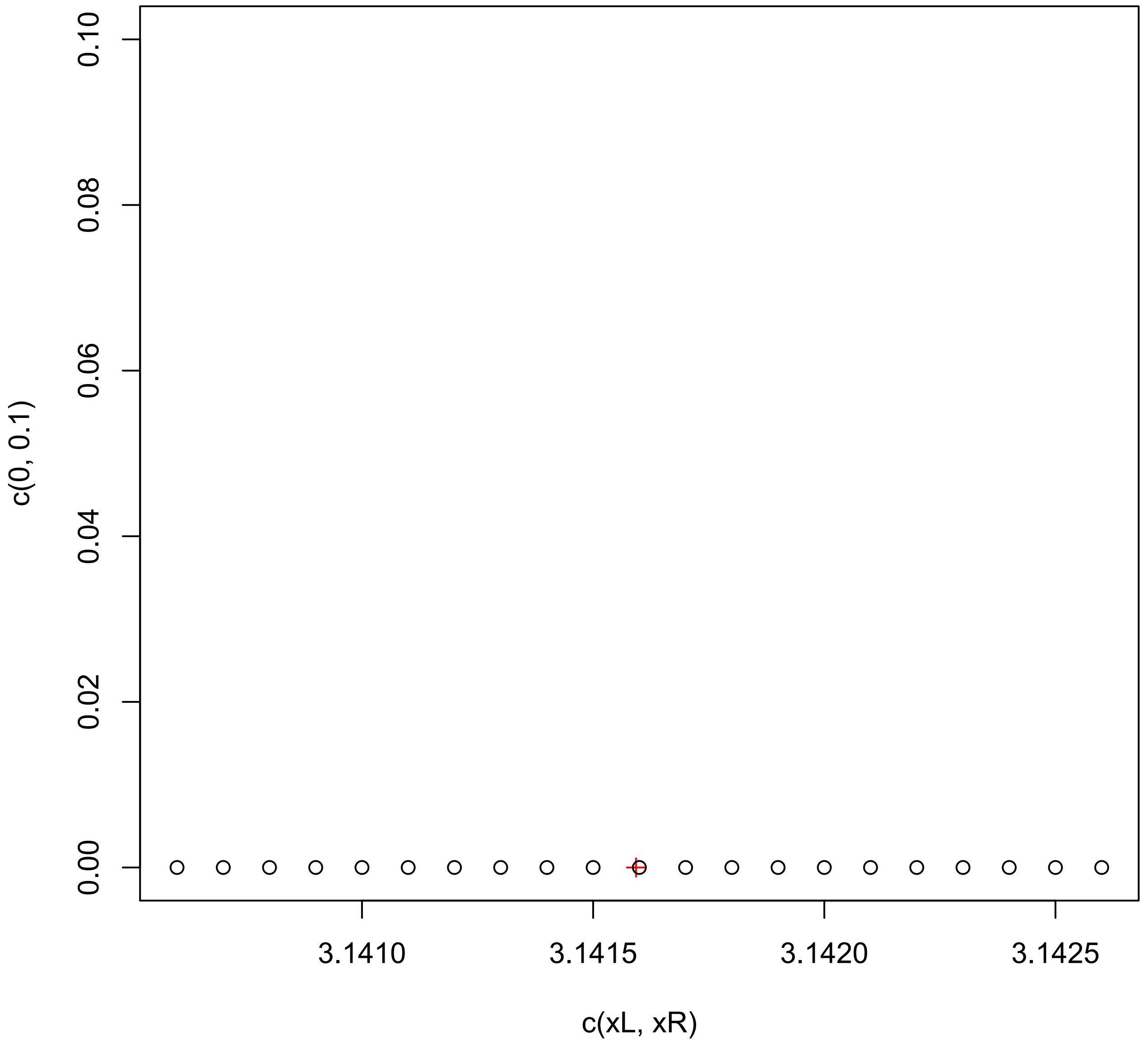
a story without words:

```
> pi ; options(digits=16) ; pi ; signif(pi, 7) ; options(digits=7) ; pi ; signif(pi, 4)
[1] 3.141593
[1] 3.141592653589793
[1] 3.141593
[1] 3.141593
[1] 3.142

> signif(pi- 0.000093, 4) ; signif(pi- 0.000092, 4)   ; dxL = 0.000092  ; xL = pi - dxL ; dxL ; xL
[1] 3.141
[1] 3.142
[1] 9.2e-05
[1] 3.141501

> signif(pi + 0.000908, 4) ; signif(pi + 0.000907, 4) ; dxR = 0.000907  ; xR = pi + dxR ; dxR ; xR
[1] 3.143
[1] 3.142
[1] 0.000907
[1] 3.1425

> xRange = xR - xL ; xRange
[1] 0.000999

> table(sort(signif(xL + xRange*runif(1000), 5)))
3.1415 3.1416 3.1417 3.1418 3.1419  3.142 3.1421 3.1422 3.1423 3.1424 3.1425
    63    102     92    109    104     93    110     88    100    102     37

> table(sort(signif(xL + xRange*runif(1000), 5)))
3.1415 3.1416 3.1417 3.1418 3.1419  3.142 3.1421 3.1422 3.1423 3.1424 3.1425
    52    106    108     88     78    113    118    104     91     96     46

> table(sort(signif(xL + xRange*runif(1000), 5)))
3.1415 3.1416 3.1417 3.1418 3.1419  3.142 3.1421 3.1422 3.1423 3.1424 3.1425
    51    107    102     95     90    101    103    106    112     92     41

> table(sort(signif(xL + xRange*runif(1000), 4)))
3.142
 1000
```



```
> plot(x=c(3.141, 3.1425), y=c(0,1), cex=0.)
> points(pi,0, pch=3, col="red") ; points(3.142, 0, pch=2, col="blue")
> x1 = sort(unique(signif(xL + xRange*runif(1000), 5)))
>
> y1 = rep(0.1, length(x1))
> plot(x1,y1)
> x1 = sort(signif(xL + xRange*runif(1000), 5))
> y1 = rep(0.1, length(x1))
> plot(x=c(3.141, 3.1425), y=c(0,1), cex=0.)
>  points(pi,0, pch=3, col="red") ; points(3.142, 0, pch=2, col="blue")
> x1 = sort(unique(signif(xL + xRange*runif(1000), 5)))
> y1 = rep(0.1, length(x1))
> points(x1, y1)
> x1 = sort(signif(xL + xRange*runif(1000), 5))
> y1 = rep(0.2, length(x1))
> points(x1, y1)
> xL = pi - 0.001 ; xR = pi + 0.001; xRange = xR - xL ; xL ; xR ; xRange
[1] 3.140593
[1] 3.142593
[1] 0.002
> 
> table(sort(signif(xL + xRange*runif(1000), 5)))

3.1406 3.1407 3.1408 3.1409  3.141 3.1411 3.1412 3.1413 3.1414 3.1415 
3.1416 3.1417 3.1418 3.1419  3.142 3.1421 
    26     49     48     48     53     51     48     43     48     58 
57     58     53     43     40     58 
3.1422 3.1423 3.1424 3.1425 3.1426 
    49     54     46     47     23 
> table(sort(signif(xL + xRange*runif(1000), 5)))

3.1406 3.1407 3.1408 3.1409  3.141 3.1411 3.1412 3.1413 3.1414 3.1415 
3.1416 3.1417 3.1418 3.1419  3.142 3.1421 
    27     49     49     51     45     56     47     39     51     45 
59     50     49     50     44     59 
3.1422 3.1423 3.1424 3.1425 3.1426 
    54     54     56     48     18 
> table(sort(signif(xL + xRange*runif(1000), 5)))

3.1406 3.1407 3.1408 3.1409  3.141 3.1411 3.1412 3.1413 3.1414 3.1415 
3.1416 3.1417 3.1418 3.1419  3.142 3.1421 
    28     59     52     60     51     49     51     33     40     54 
56     48     45     50     52     51 
3.1422 3.1423 3.1424 3.1425 3.1426 
    46     50     45     56     24 
> table(sort(signif(xL + xRange*runif(1000), 4)))

3.141 3.142 3.143 
  465   494    41 
> 
> plot(x=c(xL, xR), y=c(0,0.1), cex=0.)
> points(pi,0., pch=3, col="red")
> x1 = sort(unique(signif(xL + xRange*runif(1000), 5)))
> y1 = rep(0., length(x1))
> points(x1,y1)
> 

> (3.14 + 3.15)/2 ; (3.14 + 3.15)/2 -pi ; signif((3.14 + 3.15)/2 -pi, 4)
[1] 3.145
[1] 0.003407346
[1] 0.003407

> (3.141 + 3.142)/2 ; (3.141 + 3.142)/2 -pi ; signif((3.141 + 3.142)/2 -pi, 4)
[1] 3.1415
[1] -9.265359e-05
[1] -9.265e-05
```

```
xL = pi - 0.001 ; xR = pi + 0.001; xRange = xR - xL ; xL ; xR ; xRange

table(sort(signif(xL + xRange*runif(1000), 5)))
table(sort(signif(xL + xRange*runif(1000), 5)))
table(sort(signif(xL + xRange*runif(1000), 5)))
table(sort(signif(xL + xRange*runif(1000), 4)))

plot(x=c(xL, xR), y=c(0,0.1), cex=0.)
points(pi,0., pch=3, col="red")
x1 = sort(unique(signif(xL + xRange*runif(1000), 5)))
y1 = rep(0., length(x1))
points(x1,y1)
```


c(0, 0.1)
c(xL, xR)