STAT 2450 Assignment 3 (40 points)

Tasneem Hoque

Banner: B00841761

Problem 1 (5 pts)

In this problem, we will manipulate unit conversion formulas. We will assume that:

1 USD = 1.28 CAD 1 american pint = 0.473 liters

An industry standard beer bottle or beer can is 12 U.S. fluid ounces. A U.S. pint is equal to 16 U.S. fluid ounces. In other words, the volume of liquid contained in a bottle of beer is equivalent to 3/4 U.S. pints.

Write a function, named ‘cpl’, that calculates the price of a litre of beer in canadian dollars (CAD), based on the price of a pack of 12 beers in U.S. dollars (USD).

Your function should take two arguments: the number of bottle in a pack of beers (nb) and the cost in USD of this pack of beer (usd). This way, you will be able to use packs of 6 or packs of 12 bottles of beer.

Your function should return the price, in canadian dollars (CAD) per liter of beer.

```
cpl=function(nb, usd){
  ppb = nb/usd
  vpb = 0.75*0.473
  usdpl = vpb/ppb
  cadpl= usdpl * 1.28
  return(cadpl)
}
```

Now use your function; assume that a pack of 12 beers costs 14.73 USD and let your function display the price in CAD of one liter of beer.

```
cpl(12,14.73)
```

```
## [1] 0.5573832
```

Problem 2 (5 points)

In this problem, we want to see if we find a value (root) of x such that:

$$x^3 - 10x + 1 = 0$$

We will implement a simple version of the so-called bisection method to find intervals that contain a root.

2.1 [2 pts]

write a code that creates a grid (vector) ‘x’ of regularly spaced values between $x = -5$ and $x = 5$ by steps of 0.01, and computes a vector ‘y’ of the values of the polynomial at the positions given by x.

Use x and y to plot (as a line, use type='l') the graph of this cubic function. Call the function 'abline' three times to overlay one horizontal line at $y = 0$, and two vertical lines: a red line at $x = -2$ and a blue line at $x = -4$.

Your plot should clearly show that the graph of the cubic curve crosses the x-axis, $y = 0$, somewhere between $x = -4$ and $x = -2$. Observe that this cubic function takes a negative value at $x = -4$ and a positive value at $x = -2$.

```
x = seq(-5, 5, by = 0.01)
expand.grid(x)
```

```
##       Var1
## 1    -5.00
## 2    -4.99
## 3    -4.98
## 4    -4.97
## 5    -4.96
## 6    -4.95
## 7    -4.94
## 8    -4.93
## 9    -4.92
## 10   -4.91
## 11   -4.90
## 12   -4.89
## 13   -4.88
## 14   -4.87
## 15   -4.86
## 16   -4.85
## 17   -4.84
## 18   -4.83
## 19   -4.82
## 20   -4.81
## 21   -4.80
## 22   -4.79
## 23   -4.78
## 24   -4.77
## 25   -4.76
## 26   -4.75
## 27   -4.74
## 28   -4.73
## 29   -4.72
## 30   -4.71
## 31   -4.70
## 32   -4.69
## 33   -4.68
## 34   -4.67
## 35   -4.66
## 36   -4.65
## 37   -4.64
## 38   -4.63
## 39   -4.62
## 40   -4.61
## 41   -4.60
## 42   -4.59
## 43   -4.58
## 44   -4.57
## 45   -4.56
## 46   -4.55
## 47   -4.54
## 48   -4.53
## 49   -4.52
## 50   -4.51
## 51   -4.50
```

```
## 52   -4.49
## 53   -4.48
## 54   -4.47
## 55   -4.46
## 56   -4.45
## 57   -4.44
## 58   -4.43
## 59   -4.42
## 60   -4.41
## 61   -4.40
## 62   -4.39
## 63   -4.38
## 64   -4.37
## 65   -4.36
## 66   -4.35
## 67   -4.34
## 68   -4.33
## 69   -4.32
## 70   -4.31
## 71   -4.30
## 72   -4.29
## 73   -4.28
## 74   -4.27
## 75   -4.26
## 76   -4.25
## 77   -4.24
## 78   -4.23
## 79   -4.22
## 80   -4.21
## 81   -4.20
## 82   -4.19
## 83   -4.18
## 84   -4.17
## 85   -4.16
## 86   -4.15
## 87   -4.14
## 88   -4.13
## 89   -4.12
## 90   -4.11
## 91   -4.10
## 92   -4.09
## 93   -4.08
## 94   -4.07
## 95   -4.06
## 96   -4.05
## 97   -4.04
## 98   -4.03
## 99   -4.02
## 100  -4.01
## 101  -4.00
## 102  -3.99
## 103  -3.98
```

```
## 104  -3.97
## 105  -3.96
## 106  -3.95
## 107  -3.94
## 108  -3.93
## 109  -3.92
## 110  -3.91
## 111  -3.90
## 112  -3.89
## 113  -3.88
## 114  -3.87
## 115  -3.86
## 116  -3.85
## 117  -3.84
## 118  -3.83
## 119  -3.82
## 120  -3.81
## 121  -3.80
## 122  -3.79
## 123  -3.78
## 124  -3.77
## 125  -3.76
## 126  -3.75
## 127  -3.74
## 128  -3.73
## 129  -3.72
## 130  -3.71
## 131  -3.70
## 132  -3.69
## 133  -3.68
## 134  -3.67
## 135  -3.66
## 136  -3.65
## 137  -3.64
## 138  -3.63
## 139  -3.62
## 140  -3.61
## 141  -3.60
## 142  -3.59
## 143  -3.58
## 144  -3.57
## 145  -3.56
## 146  -3.55
## 147  -3.54
## 148  -3.53
## 149  -3.52
## 150  -3.51
## 151  -3.50
## 152  -3.49
## 153  -3.48
## 154  -3.47
## 155  -3.46
```

```
## 156  -3.45
## 157  -3.44
## 158  -3.43
## 159  -3.42
## 160  -3.41
## 161  -3.40
## 162  -3.39
## 163  -3.38
## 164  -3.37
## 165  -3.36
## 166  -3.35
## 167  -3.34
## 168  -3.33
## 169  -3.32
## 170  -3.31
## 171  -3.30
## 172  -3.29
## 173  -3.28
## 174  -3.27
## 175  -3.26
## 176  -3.25
## 177  -3.24
## 178  -3.23
## 179  -3.22
## 180  -3.21
## 181  -3.20
## 182  -3.19
## 183  -3.18
## 184  -3.17
## 185  -3.16
## 186  -3.15
## 187  -3.14
## 188  -3.13
## 189  -3.12
## 190  -3.11
## 191  -3.10
## 192  -3.09
## 193  -3.08
## 194  -3.07
## 195  -3.06
## 196  -3.05
## 197  -3.04
## 198  -3.03
## 199  -3.02
## 200  -3.01
## 201  -3.00
## 202  -2.99
## 203  -2.98
## 204  -2.97
## 205  -2.96
## 206  -2.95
## 207  -2.94
```

```
## 208  -2.93
## 209  -2.92
## 210  -2.91
## 211  -2.90
## 212  -2.89
## 213  -2.88
## 214  -2.87
## 215  -2.86
## 216  -2.85
## 217  -2.84
## 218  -2.83
## 219  -2.82
## 220  -2.81
## 221  -2.80
## 222  -2.79
## 223  -2.78
## 224  -2.77
## 225  -2.76
## 226  -2.75
## 227  -2.74
## 228  -2.73
## 229  -2.72
## 230  -2.71
## 231  -2.70
## 232  -2.69
## 233  -2.68
## 234  -2.67
## 235  -2.66
## 236  -2.65
## 237  -2.64
## 238  -2.63
## 239  -2.62
## 240  -2.61
## 241  -2.60
## 242  -2.59
## 243  -2.58
## 244  -2.57
## 245  -2.56
## 246  -2.55
## 247  -2.54
## 248  -2.53
## 249  -2.52
## 250  -2.51
## 251  -2.50
## 252  -2.49
## 253  -2.48
## 254  -2.47
## 255  -2.46
## 256  -2.45
## 257  -2.44
## 258  -2.43
## 259  -2.42
```

```
## 260  -2.41
## 261  -2.40
## 262  -2.39
## 263  -2.38
## 264  -2.37
## 265  -2.36
## 266  -2.35
## 267  -2.34
## 268  -2.33
## 269  -2.32
## 270  -2.31
## 271  -2.30
## 272  -2.29
## 273  -2.28
## 274  -2.27
## 275  -2.26
## 276  -2.25
## 277  -2.24
## 278  -2.23
## 279  -2.22
## 280  -2.21
## 281  -2.20
## 282  -2.19
## 283  -2.18
## 284  -2.17
## 285  -2.16
## 286  -2.15
## 287  -2.14
## 288  -2.13
## 289  -2.12
## 290  -2.11
## 291  -2.10
## 292  -2.09
## 293  -2.08
## 294  -2.07
## 295  -2.06
## 296  -2.05
## 297  -2.04
## 298  -2.03
## 299  -2.02
## 300  -2.01
## 301  -2.00
## 302  -1.99
## 303  -1.98
## 304  -1.97
## 305  -1.96
## 306  -1.95
## 307  -1.94
## 308  -1.93
## 309  -1.92
## 310  -1.91
## 311  -1.90
```

```
## 312  -1.89
## 313  -1.88
## 314  -1.87
## 315  -1.86
## 316  -1.85
## 317  -1.84
## 318  -1.83
## 319  -1.82
## 320  -1.81
## 321  -1.80
## 322  -1.79
## 323  -1.78
## 324  -1.77
## 325  -1.76
## 326  -1.75
## 327  -1.74
## 328  -1.73
## 329  -1.72
## 330  -1.71
## 331  -1.70
## 332  -1.69
## 333  -1.68
## 334  -1.67
## 335  -1.66
## 336  -1.65
## 337  -1.64
## 338  -1.63
## 339  -1.62
## 340  -1.61
## 341  -1.60
## 342  -1.59
## 343  -1.58
## 344  -1.57
## 345  -1.56
## 346  -1.55
## 347  -1.54
## 348  -1.53
## 349  -1.52
## 350  -1.51
## 351  -1.50
## 352  -1.49
## 353  -1.48
## 354  -1.47
## 355  -1.46
## 356  -1.45
## 357  -1.44
## 358  -1.43
## 359  -1.42
## 360  -1.41
## 361  -1.40
## 362  -1.39
## 363  -1.38
```

```
## 364  -1.37
## 365  -1.36
## 366  -1.35
## 367  -1.34
## 368  -1.33
## 369  -1.32
## 370  -1.31
## 371  -1.30
## 372  -1.29
## 373  -1.28
## 374  -1.27
## 375  -1.26
## 376  -1.25
## 377  -1.24
## 378  -1.23
## 379  -1.22
## 380  -1.21
## 381  -1.20
## 382  -1.19
## 383  -1.18
## 384  -1.17
## 385  -1.16
## 386  -1.15
## 387  -1.14
## 388  -1.13
## 389  -1.12
## 390  -1.11
## 391  -1.10
## 392  -1.09
## 393  -1.08
## 394  -1.07
## 395  -1.06
## 396  -1.05
## 397  -1.04
## 398  -1.03
## 399  -1.02
## 400  -1.01
## 401  -1.00
## 402  -0.99
## 403  -0.98
## 404  -0.97
## 405  -0.96
## 406  -0.95
## 407  -0.94
## 408  -0.93
## 409  -0.92
## 410  -0.91
## 411  -0.90
## 412  -0.89
## 413  -0.88
## 414  -0.87
## 415  -0.86
```

```
## 416  -0.85
## 417  -0.84
## 418  -0.83
## 419  -0.82
## 420  -0.81
## 421  -0.80
## 422  -0.79
## 423  -0.78
## 424  -0.77
## 425  -0.76
## 426  -0.75
## 427  -0.74
## 428  -0.73
## 429  -0.72
## 430  -0.71
## 431  -0.70
## 432  -0.69
## 433  -0.68
## 434  -0.67
## 435  -0.66
## 436  -0.65
## 437  -0.64
## 438  -0.63
## 439  -0.62
## 440  -0.61
## 441  -0.60
## 442  -0.59
## 443  -0.58
## 444  -0.57
## 445  -0.56
## 446  -0.55
## 447  -0.54
## 448  -0.53
## 449  -0.52
## 450  -0.51
## 451  -0.50
## 452  -0.49
## 453  -0.48
## 454  -0.47
## 455  -0.46
## 456  -0.45
## 457  -0.44
## 458  -0.43
## 459  -0.42
## 460  -0.41
## 461  -0.40
## 462  -0.39
## 463  -0.38
## 464  -0.37
## 465  -0.36
## 466  -0.35
## 467  -0.34
```

```
## 468  -0.33
## 469  -0.32
## 470  -0.31
## 471  -0.30
## 472  -0.29
## 473  -0.28
## 474  -0.27
## 475  -0.26
## 476  -0.25
## 477  -0.24
## 478  -0.23
## 479  -0.22
## 480  -0.21
## 481  -0.20
## 482  -0.19
## 483  -0.18
## 484  -0.17
## 485  -0.16
## 486  -0.15
## 487  -0.14
## 488  -0.13
## 489  -0.12
## 490  -0.11
## 491  -0.10
## 492  -0.09
## 493  -0.08
## 494  -0.07
## 495  -0.06
## 496  -0.05
## 497  -0.04
## 498  -0.03
## 499  -0.02
## 500  -0.01
## 501   0.00
## 502   0.01
## 503   0.02
## 504   0.03
## 505   0.04
## 506   0.05
## 507   0.06
## 508   0.07
## 509   0.08
## 510   0.09
## 511   0.10
## 512   0.11
## 513   0.12
## 514   0.13
## 515   0.14
## 516   0.15
## 517   0.16
## 518   0.17
## 519   0.18
```

```
## 520   0.19
## 521   0.20
## 522   0.21
## 523   0.22
## 524   0.23
## 525   0.24
## 526   0.25
## 527   0.26
## 528   0.27
## 529   0.28
## 530   0.29
## 531   0.30
## 532   0.31
## 533   0.32
## 534   0.33
## 535   0.34
## 536   0.35
## 537   0.36
## 538   0.37
## 539   0.38
## 540   0.39
## 541   0.40
## 542   0.41
## 543   0.42
## 544   0.43
## 545   0.44
## 546   0.45
## 547   0.46
## 548   0.47
## 549   0.48
## 550   0.49
## 551   0.50
## 552   0.51
## 553   0.52
## 554   0.53
## 555   0.54
## 556   0.55
## 557   0.56
## 558   0.57
## 559   0.58
## 560   0.59
## 561   0.60
## 562   0.61
## 563   0.62
## 564   0.63
## 565   0.64
## 566   0.65
## 567   0.66
## 568   0.67
## 569   0.68
## 570   0.69
## 571   0.70
```

```
## 572   0.71
## 573   0.72
## 574   0.73
## 575   0.74
## 576   0.75
## 577   0.76
## 578   0.77
## 579   0.78
## 580   0.79
## 581   0.80
## 582   0.81
## 583   0.82
## 584   0.83
## 585   0.84
## 586   0.85
## 587   0.86
## 588   0.87
## 589   0.88
## 590   0.89
## 591   0.90
## 592   0.91
## 593   0.92
## 594   0.93
## 595   0.94
## 596   0.95
## 597   0.96
## 598   0.97
## 599   0.98
## 600   0.99
## 601   1.00
## 602   1.01
## 603   1.02
## 604   1.03
## 605   1.04
## 606   1.05
## 607   1.06
## 608   1.07
## 609   1.08
## 610   1.09
## 611   1.10
## 612   1.11
## 613   1.12
## 614   1.13
## 615   1.14
## 616   1.15
## 617   1.16
## 618   1.17
## 619   1.18
## 620   1.19
## 621   1.20
## 622   1.21
## 623   1.22
```

```
## 624   1.23
## 625   1.24
## 626   1.25
## 627   1.26
## 628   1.27
## 629   1.28
## 630   1.29
## 631   1.30
## 632   1.31
## 633   1.32
## 634   1.33
## 635   1.34
## 636   1.35
## 637   1.36
## 638   1.37
## 639   1.38
## 640   1.39
## 641   1.40
## 642   1.41
## 643   1.42
## 644   1.43
## 645   1.44
## 646   1.45
## 647   1.46
## 648   1.47
## 649   1.48
## 650   1.49
## 651   1.50
## 652   1.51
## 653   1.52
## 654   1.53
## 655   1.54
## 656   1.55
## 657   1.56
## 658   1.57
## 659   1.58
## 660   1.59
## 661   1.60
## 662   1.61
## 663   1.62
## 664   1.63
## 665   1.64
## 666   1.65
## 667   1.66
## 668   1.67
## 669   1.68
## 670   1.69
## 671   1.70
## 672   1.71
## 673   1.72
## 674   1.73
## 675   1.74
```

```
## 676   1.75
## 677   1.76
## 678   1.77
## 679   1.78
## 680   1.79
## 681   1.80
## 682   1.81
## 683   1.82
## 684   1.83
## 685   1.84
## 686   1.85
## 687   1.86
## 688   1.87
## 689   1.88
## 690   1.89
## 691   1.90
## 692   1.91
## 693   1.92
## 694   1.93
## 695   1.94
## 696   1.95
## 697   1.96
## 698   1.97
## 699   1.98
## 700   1.99
## 701   2.00
## 702   2.01
## 703   2.02
## 704   2.03
## 705   2.04
## 706   2.05
## 707   2.06
## 708   2.07
## 709   2.08
## 710   2.09
## 711   2.10
## 712   2.11
## 713   2.12
## 714   2.13
## 715   2.14
## 716   2.15
## 717   2.16
## 718   2.17
## 719   2.18
## 720   2.19
## 721   2.20
## 722   2.21
## 723   2.22
## 724   2.23
## 725   2.24
## 726   2.25
## 727   2.26
```

```
## 728   2.27
## 729   2.28
## 730   2.29
## 731   2.30
## 732   2.31
## 733   2.32
## 734   2.33
## 735   2.34
## 736   2.35
## 737   2.36
## 738   2.37
## 739   2.38
## 740   2.39
## 741   2.40
## 742   2.41
## 743   2.42
## 744   2.43
## 745   2.44
## 746   2.45
## 747   2.46
## 748   2.47
## 749   2.48
## 750   2.49
## 751   2.50
## 752   2.51
## 753   2.52
## 754   2.53
## 755   2.54
## 756   2.55
## 757   2.56
## 758   2.57
## 759   2.58
## 760   2.59
## 761   2.60
## 762   2.61
## 763   2.62
## 764   2.63
## 765   2.64
## 766   2.65
## 767   2.66
## 768   2.67
## 769   2.68
## 770   2.69
## 771   2.70
## 772   2.71
## 773   2.72
## 774   2.73
## 775   2.74
## 776   2.75
## 777   2.76
## 778   2.77
## 779   2.78
```

```
## 780   2.79
## 781   2.80
## 782   2.81
## 783   2.82
## 784   2.83
## 785   2.84
## 786   2.85
## 787   2.86
## 788   2.87
## 789   2.88
## 790   2.89
## 791   2.90
## 792   2.91
## 793   2.92
## 794   2.93
## 795   2.94
## 796   2.95
## 797   2.96
## 798   2.97
## 799   2.98
## 800   2.99
## 801   3.00
## 802   3.01
## 803   3.02
## 804   3.03
## 805   3.04
## 806   3.05
## 807   3.06
## 808   3.07
## 809   3.08
## 810   3.09
## 811   3.10
## 812   3.11
## 813   3.12
## 814   3.13
## 815   3.14
## 816   3.15
## 817   3.16
## 818   3.17
## 819   3.18
## 820   3.19
## 821   3.20
## 822   3.21
## 823   3.22
## 824   3.23
## 825   3.24
## 826   3.25
## 827   3.26
## 828   3.27
## 829   3.28
## 830   3.29
## 831   3.30
```

```
## 832   3.31
## 833   3.32
## 834   3.33
## 835   3.34
## 836   3.35
## 837   3.36
## 838   3.37
## 839   3.38
## 840   3.39
## 841   3.40
## 842   3.41
## 843   3.42
## 844   3.43
## 845   3.44
## 846   3.45
## 847   3.46
## 848   3.47
## 849   3.48
## 850   3.49
## 851   3.50
## 852   3.51
## 853   3.52
## 854   3.53
## 855   3.54
## 856   3.55
## 857   3.56
## 858   3.57
## 859   3.58
## 860   3.59
## 861   3.60
## 862   3.61
## 863   3.62
## 864   3.63
## 865   3.64
## 866   3.65
## 867   3.66
## 868   3.67
## 869   3.68
## 870   3.69
## 871   3.70
## 872   3.71
## 873   3.72
## 874   3.73
## 875   3.74
## 876   3.75
## 877   3.76
## 878   3.77
## 879   3.78
## 880   3.79
## 881   3.80
## 882   3.81
## 883   3.82
```

```
## 884   3.83
## 885   3.84
## 886   3.85
## 887   3.86
## 888   3.87
## 889   3.88
## 890   3.89
## 891   3.90
## 892   3.91
## 893   3.92
## 894   3.93
## 895   3.94
## 896   3.95
## 897   3.96
## 898   3.97
## 899   3.98
## 900   3.99
## 901   4.00
## 902   4.01
## 903   4.02
## 904   4.03
## 905   4.04
## 906   4.05
## 907   4.06
## 908   4.07
## 909   4.08
## 910   4.09
## 911   4.10
## 912   4.11
## 913   4.12
## 914   4.13
## 915   4.14
## 916   4.15
## 917   4.16
## 918   4.17
## 919   4.18
## 920   4.19
## 921   4.20
## 922   4.21
## 923   4.22
## 924   4.23
## 925   4.24
## 926   4.25
## 927   4.26
## 928   4.27
## 929   4.28
## 930   4.29
## 931   4.30
## 932   4.31
## 933   4.32
## 934   4.33
## 935   4.34
```

```
## 936   4.35
## 937   4.36
## 938   4.37
## 939   4.38
## 940   4.39
## 941   4.40
## 942   4.41
## 943   4.42
## 944   4.43
## 945   4.44
## 946   4.45
## 947   4.46
## 948   4.47
## 949   4.48
## 950   4.49
## 951   4.50
## 952   4.51
## 953   4.52
## 954   4.53
## 955   4.54
## 956   4.55
## 957   4.56
## 958   4.57
## 959   4.58
## 960   4.59
## 961   4.60
## 962   4.61
## 963   4.62
## 964   4.63
## 965   4.64
## 966   4.65
## 967   4.66
## 968   4.67
## 969   4.68
## 970   4.69
## 971   4.70
## 972   4.71
## 973   4.72
## 974   4.73
## 975   4.74
## 976   4.75
## 977   4.76
## 978   4.77
## 979   4.78
## 980   4.79
## 981   4.80
## 982   4.81
## 983   4.82
## 984   4.83
## 985   4.84
## 986   4.85
## 987   4.86
```

```
## 988   4.87
## 989   4.88
## 990   4.89
## 991   4.90
## 992   4.91
## 993   4.92
## 994   4.93
## 995   4.94
## 996   4.95
## 997   4.96
## 998   4.97
## 999   4.98
## 1000  4.99
## 1001  5.00
```

```
y <- vector(length=length(x))

for (i in 1:length(x)) {
  y[i] = x[i]^3 - 10*x[i] + 1

}

expand.grid(y)
```

```
##           Var1
## 1    -74.000000
## 2    -73.351499
## 3    -72.705992
## 4    -72.063473
## 5    -71.423936
## 6    -70.787375
## 7    -70.153784
## 8    -69.523157
## 9    -68.895488
## 10   -68.270771
## 11   -67.649000
## 12   -67.030169
## 13   -66.414272
## 14   -65.801303
## 15   -65.191256
## 16   -64.584125
## 17   -63.979904
## 18   -63.378587
## 19   -62.780168
## 20   -62.184641
## 21   -61.592000
## 22   -61.002239
## 23   -60.415352
## 24   -59.831333
## 25   -59.250176
## 26   -58.671875
## 27   -58.096424
## 28   -57.523817
## 29   -56.954048
## 30   -56.387111
## 31   -55.823000
## 32   -55.261709
## 33   -54.703232
## 34   -54.147563
## 35   -53.594696
## 36   -53.044625
## 37   -52.497344
## 38   -51.952847
## 39   -51.411128
## 40   -50.872181
## 41   -50.336000
## 42   -49.802579
## 43   -49.271912
## 44   -48.743993
## 45   -48.218816
## 46   -47.696375
## 47   -47.176664
## 48   -46.659677
## 49   -46.145408
## 50   -45.633851
## 51   -45.125000
```

```
## 52   -44.618849
## 53   -44.115392
## 54   -43.614623
## 55   -43.116536
## 56   -42.621125
## 57   -42.128384
## 58   -41.638307
## 59   -41.150888
## 60   -40.666121
## 61   -40.184000
## 62   -39.704519
## 63   -39.227672
## 64   -38.753453
## 65   -38.281856
## 66   -37.812875
## 67   -37.346504
## 68   -36.882737
## 69   -36.421568
## 70   -35.962991
## 71   -35.507000
## 72   -35.053589
## 73   -34.602752
## 74   -34.154483
## 75   -33.708776
## 76   -33.265625
## 77   -32.825024
## 78   -32.386967
## 79   -31.951448
## 80   -31.518461
## 81   -31.088000
## 82   -30.660059
## 83   -30.234632
## 84   -29.811713
## 85   -29.391296
## 86   -28.973375
## 87   -28.557944
## 88   -28.144997
## 89   -27.734528
## 90   -27.326531
## 91   -26.921000
## 92   -26.517929
## 93   -26.117312
## 94   -25.719143
## 95   -25.323416
## 96   -24.930125
## 97   -24.539264
## 98   -24.150827
## 99   -23.764808
## 100  -23.381201
## 101  -23.000000
## 102  -22.621199
## 103  -22.244792
```

```
## 104  -21.870773
## 105  -21.499136
## 106  -21.129875
## 107  -20.762984
## 108  -20.398457
## 109  -20.036288
## 110  -19.676471
## 111  -19.319000
## 112  -18.963869
## 113  -18.611072
## 114  -18.260603
## 115  -17.912456
## 116  -17.566625
## 117  -17.223104
## 118  -16.881887
## 119  -16.542968
## 120  -16.206341
## 121  -15.872000
## 122  -15.539939
## 123  -15.210152
## 124  -14.882633
## 125  -14.557376
## 126  -14.234375
## 127  -13.913624
## 128  -13.595117
## 129  -13.278848
## 130  -12.964811
## 131  -12.653000
## 132  -12.343409
## 133  -12.036032
## 134  -11.730863
## 135  -11.427896
## 136  -11.127125
## 137  -10.828544
## 138  -10.532147
## 139  -10.237928
## 140   -9.945881
## 141   -9.656000
## 142   -9.368279
## 143   -9.082712
## 144   -8.799293
## 145   -8.518016
## 146   -8.238875
## 147   -7.961864
## 148   -7.686977
## 149   -7.414208
## 150   -7.143551
## 151   -6.875000
## 152   -6.608549
## 153   -6.344192
## 154   -6.081923
## 155   -5.821736
```

```
## 156   -5.563625
## 157   -5.307584
## 158   -5.053607
## 159   -4.801688
## 160   -4.551821
## 161   -4.304000
## 162   -4.058219
## 163   -3.814472
## 164   -3.572753
## 165   -3.333056
## 166   -3.095375
## 167   -2.859704
## 168   -2.626037
## 169   -2.394368
## 170   -2.164691
## 171   -1.937000
## 172   -1.711289
## 173   -1.487552
## 174   -1.265783
## 175   -1.045976
## 176   -0.828125
## 177   -0.612224
## 178   -0.398267
## 179   -0.186248
## 180    0.023839
## 181    0.232000
## 182    0.438241
## 183    0.642568
## 184    0.844987
## 185    1.045504
## 186    1.244125
## 187    1.440856
## 188    1.635703
## 189    1.828672
## 190    2.019769
## 191    2.209000
## 192    2.396371
## 193    2.581888
## 194    2.765557
## 195    2.947384
## 196    3.127375
## 197    3.305536
## 198    3.481873
## 199    3.656392
## 200    3.829099
## 201    4.000000
## 202    4.169101
## 203    4.336408
## 204    4.501927
## 205    4.665664
## 206    4.827625
## 207    4.987816
```

```
## 208    5.146243
## 209    5.302912
## 210    5.457829
## 211    5.611000
## 212    5.762431
## 213    5.912128
## 214    6.060097
## 215    6.206344
## 216    6.350875
## 217    6.493696
## 218    6.634813
## 219    6.774232
## 220    6.911959
## 221    7.048000
## 222    7.182361
## 223    7.315048
## 224    7.446067
## 225    7.575424
## 226    7.703125
## 227    7.829176
## 228    7.953583
## 229    8.076352
## 230    8.197489
## 231    8.317000
## 232    8.434891
## 233    8.551168
## 234    8.665837
## 235    8.778904
## 236    8.890375
## 237    9.000256
## 238    9.108553
## 239    9.215272
## 240    9.320419
## 241    9.424000
## 242    9.526021
## 243    9.626488
## 244    9.725407
## 245    9.822784
## 246    9.918625
## 247   10.012936
## 248   10.105723
## 249   10.196992
## 250   10.286749
## 251   10.375000
## 252   10.461751
## 253   10.547008
## 254   10.630777
## 255   10.713064
## 256   10.793875
## 257   10.873216
## 258   10.951093
## 259   11.027512
```

```
## 260   11.102479
## 261   11.176000
## 262   11.248081
## 263   11.318728
## 264   11.387947
## 265   11.455744
## 266   11.522125
## 267   11.587096
## 268   11.650663
## 269   11.712832
## 270   11.773609
## 271   11.833000
## 272   11.891011
## 273   11.947648
## 274   12.002917
## 275   12.056824
## 276   12.109375
## 277   12.160576
## 278   12.210433
## 279   12.258952
## 280   12.306139
## 281   12.352000
## 282   12.396541
## 283   12.439768
## 284   12.481687
## 285   12.522304
## 286   12.561625
## 287   12.599656
## 288   12.636403
## 289   12.671872
## 290   12.706069
## 291   12.739000
## 292   12.770671
## 293   12.801088
## 294   12.830257
## 295   12.858184
## 296   12.884875
## 297   12.910336
## 298   12.934573
## 299   12.957592
## 300   12.979399
## 301   13.000000
## 302   13.019401
## 303   13.037608
## 304   13.054627
## 305   13.070464
## 306   13.085125
## 307   13.098616
## 308   13.110943
## 309   13.122112
## 310   13.132129
## 311   13.141000
```

```
## 312   13.148731
## 313   13.155328
## 314   13.160797
## 315   13.165144
## 316   13.168375
## 317   13.170496
## 318   13.171513
## 319   13.171432
## 320   13.170259
## 321   13.168000
## 322   13.164661
## 323   13.160248
## 324   13.154767
## 325   13.148224
## 326   13.140625
## 327   13.131976
## 328   13.122283
## 329   13.111552
## 330   13.099789
## 331   13.087000
## 332   13.073191
## 333   13.058368
## 334   13.042537
## 335   13.025704
## 336   13.007875
## 337   12.989056
## 338   12.969253
## 339   12.948472
## 340   12.926719
## 341   12.904000
## 342   12.880321
## 343   12.855688
## 344   12.830107
## 345   12.803584
## 346   12.776125
## 347   12.747736
## 348   12.718423
## 349   12.688192
## 350   12.657049
## 351   12.625000
## 352   12.592051
## 353   12.558208
## 354   12.523477
## 355   12.487864
## 356   12.451375
## 357   12.414016
## 358   12.375793
## 359   12.336712
## 360   12.296779
## 361   12.256000
## 362   12.214381
## 363   12.171928
```

```
## 364   12.128647
## 365   12.084544
## 366   12.039625
## 367   11.993896
## 368   11.947363
## 369   11.900032
## 370   11.851909
## 371   11.803000
## 372   11.753311
## 373   11.702848
## 374   11.651617
## 375   11.599624
## 376   11.546875
## 377   11.493376
## 378   11.439133
## 379   11.384152
## 380   11.328439
## 381   11.272000
## 382   11.214841
## 383   11.156968
## 384   11.098387
## 385   11.039104
## 386   10.979125
## 387   10.918456
## 388   10.857103
## 389   10.795072
## 390   10.732369
## 391   10.669000
## 392   10.604971
## 393   10.540288
## 394   10.474957
## 395   10.408984
## 396   10.342375
## 397   10.275136
## 398   10.207273
## 399   10.138792
## 400   10.069699
## 401   10.000000
## 402    9.929701
## 403    9.858808
## 404    9.787327
## 405    9.715264
## 406    9.642625
## 407    9.569416
## 408    9.495643
## 409    9.421312
## 410    9.346429
## 411    9.271000
## 412    9.195031
## 413    9.118528
## 414    9.041497
## 415    8.963944
```

```
## 416    8.885875
## 417    8.807296
## 418    8.728213
## 419    8.648632
## 420    8.568559
## 421    8.488000
## 422    8.406961
## 423    8.325448
## 424    8.243467
## 425    8.161024
## 426    8.078125
## 427    7.994776
## 428    7.910983
## 429    7.826752
## 430    7.742089
## 431    7.657000
## 432    7.571491
## 433    7.485568
## 434    7.399237
## 435    7.312504
## 436    7.225375
## 437    7.137856
## 438    7.049953
## 439    6.961672
## 440    6.873019
## 441    6.784000
## 442    6.694621
## 443    6.604888
## 444    6.514807
## 445    6.424384
## 446    6.333625
## 447    6.242536
## 448    6.151123
## 449    6.059392
## 450    5.967349
## 451    5.875000
## 452    5.782351
## 453    5.689408
## 454    5.596177
## 455    5.502664
## 456    5.408875
## 457    5.314816
## 458    5.220493
## 459    5.125912
## 460    5.031079
## 461    4.936000
## 462    4.840681
## 463    4.745128
## 464    4.649347
## 465    4.553344
## 466    4.457125
## 467    4.360696
```

```
## 468    4.264063
## 469    4.167232
## 470    4.070209
## 471    3.973000
## 472    3.875611
## 473    3.778048
## 474    3.680317
## 475    3.582424
## 476    3.484375
## 477    3.386176
## 478    3.287833
## 479    3.189352
## 480    3.090739
## 481    2.992000
## 482    2.893141
## 483    2.794168
## 484    2.695087
## 485    2.595904
## 486    2.496625
## 487    2.397256
## 488    2.297803
## 489    2.198272
## 490    2.098669
## 491    1.999000
## 492    1.899271
## 493    1.799488
## 494    1.699657
## 495    1.599784
## 496    1.499875
## 497    1.399936
## 498    1.299973
## 499    1.199992
## 500    1.099999
## 501    1.000000
## 502    0.900001
## 503    0.800008
## 504    0.700027
## 505    0.600064
## 506    0.500125
## 507    0.400216
## 508    0.300343
## 509    0.200512
## 510    0.100729
## 511    0.001000
## 512   -0.098669
## 513   -0.198272
## 514   -0.297803
## 515   -0.397256
## 516   -0.496625
## 517   -0.595904
## 518   -0.695087
## 519   -0.794168
```

```
## 520   -0.893141
## 521   -0.992000
## 522   -1.090739
## 523   -1.189352
## 524   -1.287833
## 525   -1.386176
## 526   -1.484375
## 527   -1.582424
## 528   -1.680317
## 529   -1.778048
## 530   -1.875611
## 531   -1.973000
## 532   -2.070209
## 533   -2.167232
## 534   -2.264063
## 535   -2.360696
## 536   -2.457125
## 537   -2.553344
## 538   -2.649347
## 539   -2.745128
## 540   -2.840681
## 541   -2.936000
## 542   -3.031079
## 543   -3.125912
## 544   -3.220493
## 545   -3.314816
## 546   -3.408875
## 547   -3.502664
## 548   -3.596177
## 549   -3.689408
## 550   -3.782351
## 551   -3.875000
## 552   -3.967349
## 553   -4.059392
## 554   -4.151123
## 555   -4.242536
## 556   -4.333625
## 557   -4.424384
## 558   -4.514807
## 559   -4.604888
## 560   -4.694621
## 561   -4.784000
## 562   -4.873019
## 563   -4.961672
## 564   -5.049953
## 565   -5.137856
## 566   -5.225375
## 567   -5.312504
## 568   -5.399237
## 569   -5.485568
## 570   -5.571491
## 571   -5.657000
```

```
## 572   -5.742089
## 573   -5.826752
## 574   -5.910983
## 575   -5.994776
## 576   -6.078125
## 577   -6.161024
## 578   -6.243467
## 579   -6.325448
## 580   -6.406961
## 581   -6.488000
## 582   -6.568559
## 583   -6.648632
## 584   -6.728213
## 585   -6.807296
## 586   -6.885875
## 587   -6.963944
## 588   -7.041497
## 589   -7.118528
## 590   -7.195031
## 591   -7.271000
## 592   -7.346429
## 593   -7.421312
## 594   -7.495643
## 595   -7.569416
## 596   -7.642625
## 597   -7.715264
## 598   -7.787327
## 599   -7.858808
## 600   -7.929701
## 601   -8.000000
## 602   -8.069699
## 603   -8.138792
## 604   -8.207273
## 605   -8.275136
## 606   -8.342375
## 607   -8.408984
## 608   -8.474957
## 609   -8.540288
## 610   -8.604971
## 611   -8.669000
## 612   -8.732369
## 613   -8.795072
## 614   -8.857103
## 615   -8.918456
## 616   -8.979125
## 617   -9.039104
## 618   -9.098387
## 619   -9.156968
## 620   -9.214841
## 621   -9.272000
## 622   -9.328439
## 623   -9.384152
```

```
## 624   -9.439133
## 625   -9.493376
## 626   -9.546875
## 627   -9.599624
## 628   -9.651617
## 629   -9.702848
## 630   -9.753311
## 631   -9.803000
## 632   -9.851909
## 633   -9.900032
## 634   -9.947363
## 635   -9.993896
## 636  -10.039625
## 637  -10.084544
## 638  -10.128647
## 639  -10.171928
## 640  -10.214381
## 641  -10.256000
## 642  -10.296779
## 643  -10.336712
## 644  -10.375793
## 645  -10.414016
## 646  -10.451375
## 647  -10.487864
## 648  -10.523477
## 649  -10.558208
## 650  -10.592051
## 651  -10.625000
## 652  -10.657049
## 653  -10.688192
## 654  -10.718423
## 655  -10.747736
## 656  -10.776125
## 657  -10.803584
## 658  -10.830107
## 659  -10.855688
## 660  -10.880321
## 661  -10.904000
## 662  -10.926719
## 663  -10.948472
## 664  -10.969253
## 665  -10.989056
## 666  -11.007875
## 667  -11.025704
## 668  -11.042537
## 669  -11.058368
## 670  -11.073191
## 671  -11.087000
## 672  -11.099789
## 673  -11.111552
## 674  -11.122283
## 675  -11.131976
```

```
## 676  -11.140625
## 677  -11.148224
## 678  -11.154767
## 679  -11.160248
## 680  -11.164661
## 681  -11.168000
## 682  -11.170259
## 683  -11.171432
## 684  -11.171513
## 685  -11.170496
## 686  -11.168375
## 687  -11.165144
## 688  -11.160797
## 689  -11.155328
## 690  -11.148731
## 691  -11.141000
## 692  -11.132129
## 693  -11.122112
## 694  -11.110943
## 695  -11.098616
## 696  -11.085125
## 697  -11.070464
## 698  -11.054627
## 699  -11.037608
## 700  -11.019401
## 701  -11.000000
## 702  -10.979399
## 703  -10.957592
## 704  -10.934573
## 705  -10.910336
## 706  -10.884875
## 707  -10.858184
## 708  -10.830257
## 709  -10.801088
## 710  -10.770671
## 711  -10.739000
## 712  -10.706069
## 713  -10.671872
## 714  -10.636403
## 715  -10.599656
## 716  -10.561625
## 717  -10.522304
## 718  -10.481687
## 719  -10.439768
## 720  -10.396541
## 721  -10.352000
## 722  -10.306139
## 723  -10.258952
## 724  -10.210433
## 725  -10.160576
## 726  -10.109375
## 727  -10.056824
```

```
## 728  -10.002917
## 729   -9.947648
## 730   -9.891011
## 731   -9.833000
## 732   -9.773609
## 733   -9.712832
## 734   -9.650663
## 735   -9.587096
## 736   -9.522125
## 737   -9.455744
## 738   -9.387947
## 739   -9.318728
## 740   -9.248081
## 741   -9.176000
## 742   -9.102479
## 743   -9.027512
## 744   -8.951093
## 745   -8.873216
## 746   -8.793875
## 747   -8.713064
## 748   -8.630777
## 749   -8.547008
## 750   -8.461751
## 751   -8.375000
## 752   -8.286749
## 753   -8.196992
## 754   -8.105723
## 755   -8.012936
## 756   -7.918625
## 757   -7.822784
## 758   -7.725407
## 759   -7.626488
## 760   -7.526021
## 761   -7.424000
## 762   -7.320419
## 763   -7.215272
## 764   -7.108553
## 765   -7.000256
## 766   -6.890375
## 767   -6.778904
## 768   -6.665837
## 769   -6.551168
## 770   -6.434891
## 771   -6.317000
## 772   -6.197489
## 773   -6.076352
## 774   -5.953583
## 775   -5.829176
## 776   -5.703125
## 777   -5.575424
## 778   -5.446067
## 779   -5.315048
```

```
## 780   -5.182361
## 781   -5.048000
## 782   -4.911959
## 783   -4.774232
## 784   -4.634813
## 785   -4.493696
## 786   -4.350875
## 787   -4.206344
## 788   -4.060097
## 789   -3.912128
## 790   -3.762431
## 791   -3.611000
## 792   -3.457829
## 793   -3.302912
## 794   -3.146243
## 795   -2.987816
## 796   -2.827625
## 797   -2.665664
## 798   -2.501927
## 799   -2.336408
## 800   -2.169101
## 801   -2.000000
## 802   -1.829099
## 803   -1.656392
## 804   -1.481873
## 805   -1.305536
## 806   -1.127375
## 807   -0.947384
## 808   -0.765557
## 809   -0.581888
## 810   -0.396371
## 811   -0.209000
## 812   -0.019769
## 813    0.171328
## 814    0.364297
## 815    0.559144
## 816    0.755875
## 817    0.954496
## 818    1.155013
## 819    1.357432
## 820    1.561759
## 821    1.768000
## 822    1.976161
## 823    2.186248
## 824    2.398267
## 825    2.612224
## 826    2.828125
## 827    3.045976
## 828    3.265783
## 829    3.487552
## 830    3.711289
## 831    3.937000
```

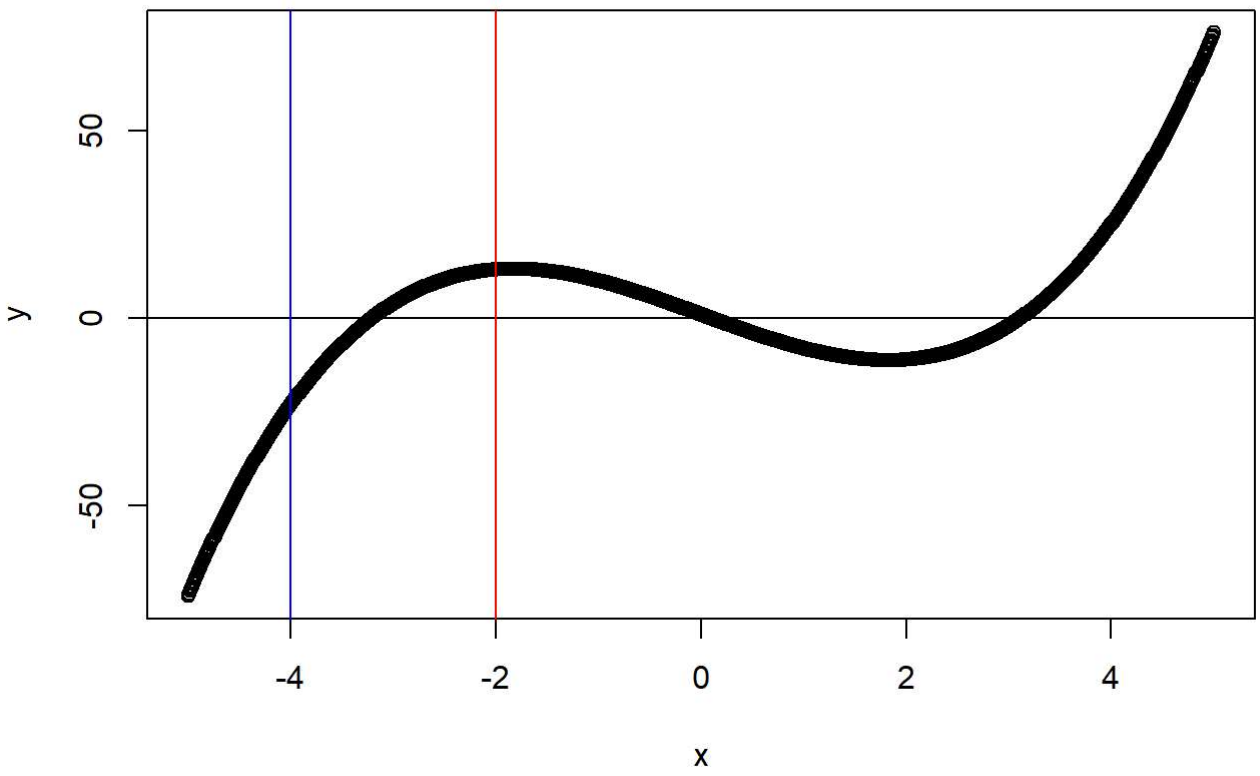
```
## 832    4.164691
## 833    4.394368
## 834    4.626037
## 835    4.859704
## 836    5.095375
## 837    5.333056
## 838    5.572753
## 839    5.814472
## 840    6.058219
## 841    6.304000
## 842    6.551821
## 843    6.801688
## 844    7.053607
## 845    7.307584
## 846    7.563625
## 847    7.821736
## 848    8.081923
## 849    8.344192
## 850    8.608549
## 851    8.875000
## 852    9.143551
## 853    9.414208
## 854    9.686977
## 855    9.961864
## 856   10.238875
## 857   10.518016
## 858   10.799293
## 859   11.082712
## 860   11.368279
## 861   11.656000
## 862   11.945881
## 863   12.237928
## 864   12.532147
## 865   12.828544
## 866   13.127125
## 867   13.427896
## 868   13.730863
## 869   14.036032
## 870   14.343409
## 871   14.653000
## 872   14.964811
## 873   15.278848
## 874   15.595117
## 875   15.913624
## 876   16.234375
## 877   16.557376
## 878   16.882633
## 879   17.210152
## 880   17.539939
## 881   17.872000
## 882   18.206341
## 883   18.542968
```

```
## 884   18.881887
## 885   19.223104
## 886   19.566625
## 887   19.912456
## 888   20.260603
## 889   20.611072
## 890   20.963869
## 891   21.319000
## 892   21.676471
## 893   22.036288
## 894   22.398457
## 895   22.762984
## 896   23.129875
## 897   23.499136
## 898   23.870773
## 899   24.244792
## 900   24.621199
## 901   25.000000
## 902   25.381201
## 903   25.764808
## 904   26.150827
## 905   26.539264
## 906   26.930125
## 907   27.323416
## 908   27.719143
## 909   28.117312
## 910   28.517929
## 911   28.921000
## 912   29.326531
## 913   29.734528
## 914   30.144997
## 915   30.557944
## 916   30.973375
## 917   31.391296
## 918   31.811713
## 919   32.234632
## 920   32.660059
## 921   33.088000
## 922   33.518461
## 923   33.951448
## 924   34.386967
## 925   34.825024
## 926   35.265625
## 927   35.708776
## 928   36.154483
## 929   36.602752
## 930   37.053589
## 931   37.507000
## 932   37.962991
## 933   38.421568
## 934   38.882737
## 935   39.346504
```

```
## 936   39.812875
## 937   40.281856
## 938   40.753453
## 939   41.227672
## 940   41.704519
## 941   42.184000
## 942   42.666121
## 943   43.150888
## 944   43.638307
## 945   44.128384
## 946   44.621125
## 947   45.116536
## 948   45.614623
## 949   46.115392
## 950   46.618849
## 951   47.125000
## 952   47.633851
## 953   48.145408
## 954   48.659677
## 955   49.176664
## 956   49.696375
## 957   50.218816
## 958   50.743993
## 959   51.271912
## 960   51.802579
## 961   52.336000
## 962   52.872181
## 963   53.411128
## 964   53.952847
## 965   54.497344
## 966   55.044625
## 967   55.594696
## 968   56.147563
## 969   56.703232
## 970   57.261709
## 971   57.823000
## 972   58.387111
## 973   58.954048
## 974   59.523817
## 975   60.096424
## 976   60.671875
## 977   61.250176
## 978   61.831333
## 979   62.415352
## 980   63.002239
## 981   63.592000
## 982   64.184641
## 983   64.780168
## 984   65.378587
## 985   65.979904
## 986   66.584125
## 987   67.191256
```

```
## 988   67.801303
## 989   68.414272
## 990   69.030169
## 991   69.649000
## 992   70.270771
## 993   70.895488
## 994   71.523157
## 995   72.153784
## 996   72.787375
## 997   73.423936
## 998   74.063473
## 999   74.705992
## 1000  75.351499
## 1001  76.000000
```

```
plot(x,y)
abline(h=0)
abline(v=-2, col='red')
abline(v=-4, col='blue')
```



2.2 (2 pts)

Write a function called ‘findroot’ that obeys the following prescriptions:

1. the function has 4 arguments: p,q,a,b

2. p and q are the coefficients of a cubic polynomial

$$f(x) = x^3 + px + q$$

   of which we want to find a root (real value of x that makes this expression equal to zero).
3. We want the function to always return the midpoint of the interval $[a, b]$.
4. We want the function to print the message 'there is a root between a and b', when the signs that the polynomial function takes at $x = a$ and $x = b$ differ (i.e. $f(a)$ and $f(b)$ have different signs).

Therefore, all you have to do is to calculate the values $f(a)$ and $f(b)$ of the polynomial at each end (a or b) of the interval $[a, b]$, and compute the product of these values. If the product is negative, it means that the signs of the polynomial function differ when calculated at a and at b, which implies that the graph of the polynomial is crossing the x axis (i.e. there is a zero or a root inside the interval [a,b]). If this condition is satisfied, then print the message 'there is a root between a and b'. otherwise, print no message.

Make sure that your function always returns the midpoint of the interval $[a, b]$, even if no message is printed.

write the function 'findroot' here:

```
findroot=function(p,q,a,b){
  fa = (a)^3+p*(a)+q
  fb = (b)^3+p*(b)+q
  num = fa*fb

  if(num < 0) {
    cat("There is a zero or root inside the interval [",a,",",b,"]\n")
  }
  return((fa+fb)/2)
}
```

2.3 (1 pt)

Now, let's apply your function to the example above. Use your function to answer these questions:

1. is there is a root between -4 and -2 ?

```
findroot(-10,1,-4,-2)
```

```
## There is a zero or root inside the interval [ -4 , -2 ]
```

```
## [1] -5
```

2. Is there a root between -3.25 and -3.125 ?

```
findroot(-10,1,-3.25,-3.125)
```

```
## There is a zero or root inside the interval [ -3.25 , -3.125 ]
```

```
## [1] 0.4521484
```

Note: the bisection method is a robust method that can be used to find the roots of any continuous fonction on a compact (finite and closed) interval $[a, b]$. Feel free (THIS IS NOT a question for THIS assignment) to fully implement the bisection algorithm as a fun exercise.

Problem 3 (5 pts)

When $x_1, x_2, \ldots, x_n$ is a sample from a normal distribution with unknown mean μ and unknown variance σ^2, the level $100(1-\alpha)\%$ confidence interval for μ is given by the following formula:

$$\bar{x} \pm t_{1-\alpha/2,n-1} \frac{s}{\sqrt{n}}$$

where $\bar{x}$ and s are the sample mean and sample standard deviation of the data, and $t_{1-\alpha/2,n-1}$ cuts off an area $1-\alpha/2$ to its left under the t curve with $n-1$ degrees of freedom.

Write a function which has two arguments, a vector of data x, and alpha, which takes any value between 0 and 1, but should have a default value of .05 (hence you must learn how to program functions with default values for their arguments).

The function should return a vector of length 2, which contains the endpoints of the confidence interval.

The percentiles of the t-distribution can be calculated as follows. Suppose that you want the 97.5'th percentile of the t-distribution with 11 degrees of freedom. This can be calculated in R as

```
qt(.975,11)
```

```
## [1] 2.200985
```

```
# write your function
conf = function(x,a=0.05) {
    n= length(x)
    y = vector(length=2)

  meanResult = mean(x)
   stdDev = sd(x)

   tnx <- qt(1-a/2,n-1)
   print(tnx)

   y[1]<-meanResult - tnx*stdDev/sqrt(n)
   y[2]<-meanResult + tnx*stdDev/sqrt(n)

   return(y)
}
```

Test your function by calculating the 98% confidence interval using the following data

```
set.seed(222)
data=rnorm(25,mean=2.5,sd=.45)
```

When putting your two endpoints together, you may find something similar to the following to be useful.

```
1+c(-1,1)*.25
```

```
## [1] 0.75 1.25
```

```
# apply your function to the data sample 'data'

conf(data,0.02)
```

```
## [1] 2.492159
```

```
## [1] 2.248660 2.711667
```

Now check that your calculation is correct, by using:

```
t.test(data,conf.level=.98)
```

```
##
##  One Sample t-test
##
## data:  data
## t = 26.699, df = 24, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 98 percent confidence interval:
##  2.248660 2.711667
## sample estimates:
## mean of x
##  2.480163
```

Problem 4 (5 pts)

The derivative of a function $f(x)$ at x can be approximated by the Newton's quotient

$$\frac{f(x+h)-f(x-h)}{2*h}$$

where h is a small number.

Write a function named 'slope' to calculate the Newton's quotient for the function $f(x)=exp(x)$.

The function should take two scalar arguments, x and h. Make sure that your function has a default value of $h=1.e-6$.

Test your function at the point $x=1$ using the default value of h, and write a line of code to compare approximate value of the derivative calculated by your function to the true (theoretical) value of the derivative of $f(x)=e^x$ at $x=1$, which is $f'(1)=e^1$.

```
slope = function(x, h = 1.e-6){
   quotient = (exp(x+h) - exp(x-h))/(2*h)
   return(quotient)
}

# call function to find slope
slope(1)
```

```
## [1] 2.718282
```

```
# test function
x=1
deriv(exp(x), 'x')
```

```
## expression({
##     .value <- 2.71828182845905
##     .grad <- array(0, c(length(.value), 1L), list(NULL, c("x")))
##     .grad[, "x"] <- 0
##     attr(.value, "gradient") <- .grad
##     .value
## })
```

Problem 5 (10 pts)

A very useful feature in R is the ability to pass a function name as an argument.

Here is an example, where we pass three different function names: $exp(x)$, $log(x)$, and $sin(x)$, to the function 'test' to do calculations that depends on the function choice:

```
test=function(x,a,f){
 output=f(x+a)+f(x-a)
 return(output)
}
test(0,2,exp) # this will calculate exp(0+2)+exp(0-2)
```

```
## [1] 7.524391
```

```
test(20,5,log) # this will calculate log(20+5)+log(20-5)
```

```
## [1] 5.926926
```

```
test(0,1,sin) # this will calculate sin(0+1)+sin(0-1)
```

```
## [1] 0
```

4.1 (4 pts)

Now, modify the function you wrote for problem 4 so that your new function, called 'slopef', accepts a new argument that is the name of a function of which you want to approximate the derivative.

The function 'slopef' should also accept an argument named h for the 'infinitesimal' step size. Use the same default value as in problem 4 for this argument.

```
# implement your 'slopef' function here
slopef = function(x,f, h  = 1e-06) {
  output=(f(x+h)-f(x-h))/(2*h)
  return(output)
}
```

4.2 (3 pts)

When your function is written, use it to approximate the derivative of sin(x) at $x = \pi/4$, of $log(x)$ at $x = 3$, and of $exp(x)$ at $x = 2$.

```
slopef(pi/4,sin)
```

```
## [1] 0.7071068
```

```
slopef(3,log)
```

```
## [1] 0.3333333
```

```
slopef(2,exp)
```

```
## [1] 7.389056
```

4.3 (3 pts)

Note that you can easily check that your results are right, because we know the analytical expression of the derivatives of the sin, the log and the exp function.

$$\frac{d}{dx}(\sin x) = \cos x$$
$$\frac{d}{dx}(\log x) = \frac{1}{x}$$
$$\frac{d}{dx}(e^x) = e^x$$

Write a code to compute (and print) the 3 values of the derivatives, computed from these analytical expressions

```
cos(pi/4)
```

```
## [1] 0.7071068
```

```
1/3
```

```
## [1] 0.3333333
```

```
exp(2)
```

```
## [1] 7.389056
```

Problem 6 (10 points)

In this problem, you will need to write a code that estimates the empirical coverage of the student t confidence intervals. To do so, proceed exactly as seen in lecture 1 of Module 3. Feel free to copy, paste and adapt the code contained in the Rmd file of this lecture.

Please write your code so as to conform to the following prescriptions:

Use 20 simulations.

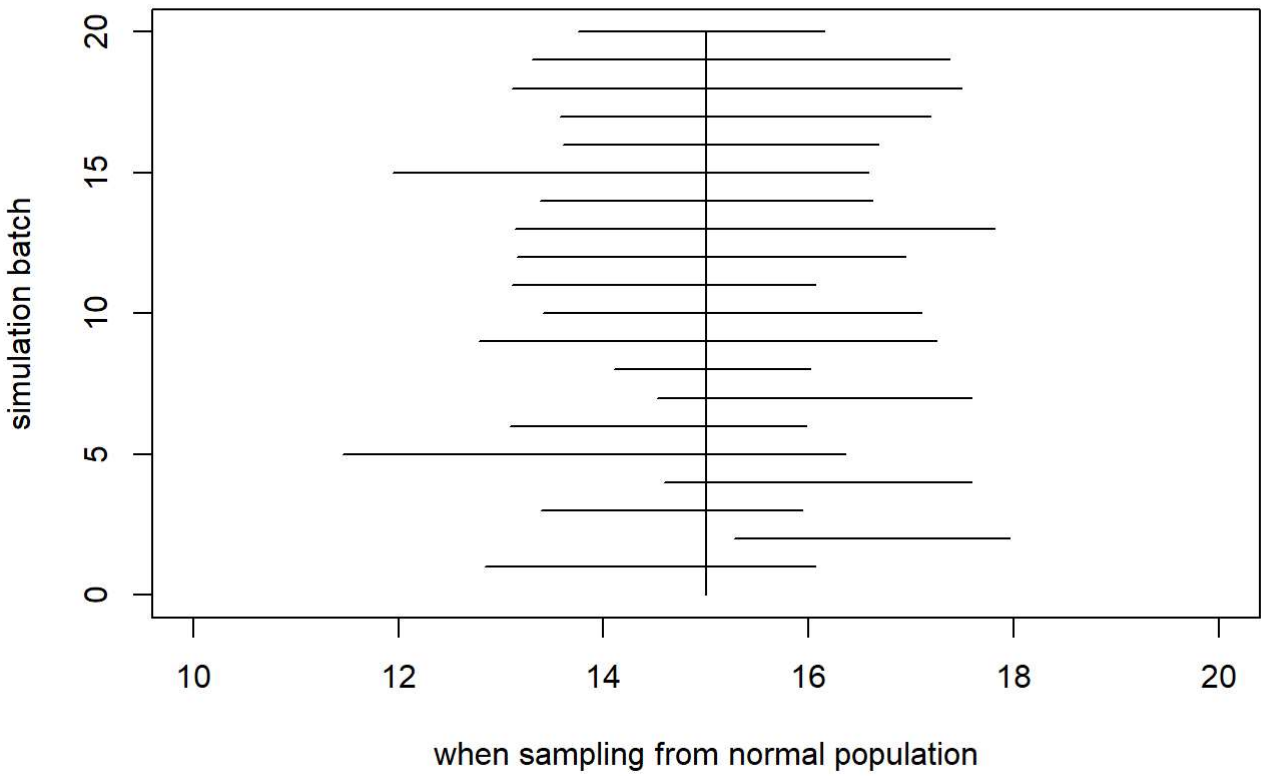
In each simulation, use a new sample of size n=10 Draw your sample from a normal distribution with ('true') mean 15 and ('true') standard deviation 3.

For your plot, which should show each t-confidence interval as a horizontal segment, use xlim =c(10,20) (for Nbatch simulations, one should see Nbatch confidence interval segments in the plot).

For each simulation, you will use the formula of student to compute a 90% CI, and increment a counter to estimate the empirical coverage.

Write your simulation code:

```
Nbatch=20
plot(0,0,xlim=c(10,20),ylim=c(0,Nbatch),type="n",  ylab="simulation batch", xlab="when sampling
 from normal population")


 n=10
 truemean=15;
 truesd=3;

 alpha=.1

 cover=0

for (i in 1:Nbatch) {

   data=rnorm(n,truemean,truesd)
   estmean = mean(data)
   estsd   =   sd(data)

   tnx <- qt(1-alpha/2,n-1)

   tint=estmean + c(-1,1)*tnx*estsd/sqrt(n)

   count1=ifelse(tint[1]<truemean&&truemean<tint[2],1,0)

   cover=cover+count1

   lines(tint,c(i,i))

 }

lines(c(truemean,truemean),c(0,Nbatch))
```




```
empirical=cover/Nbatch
```

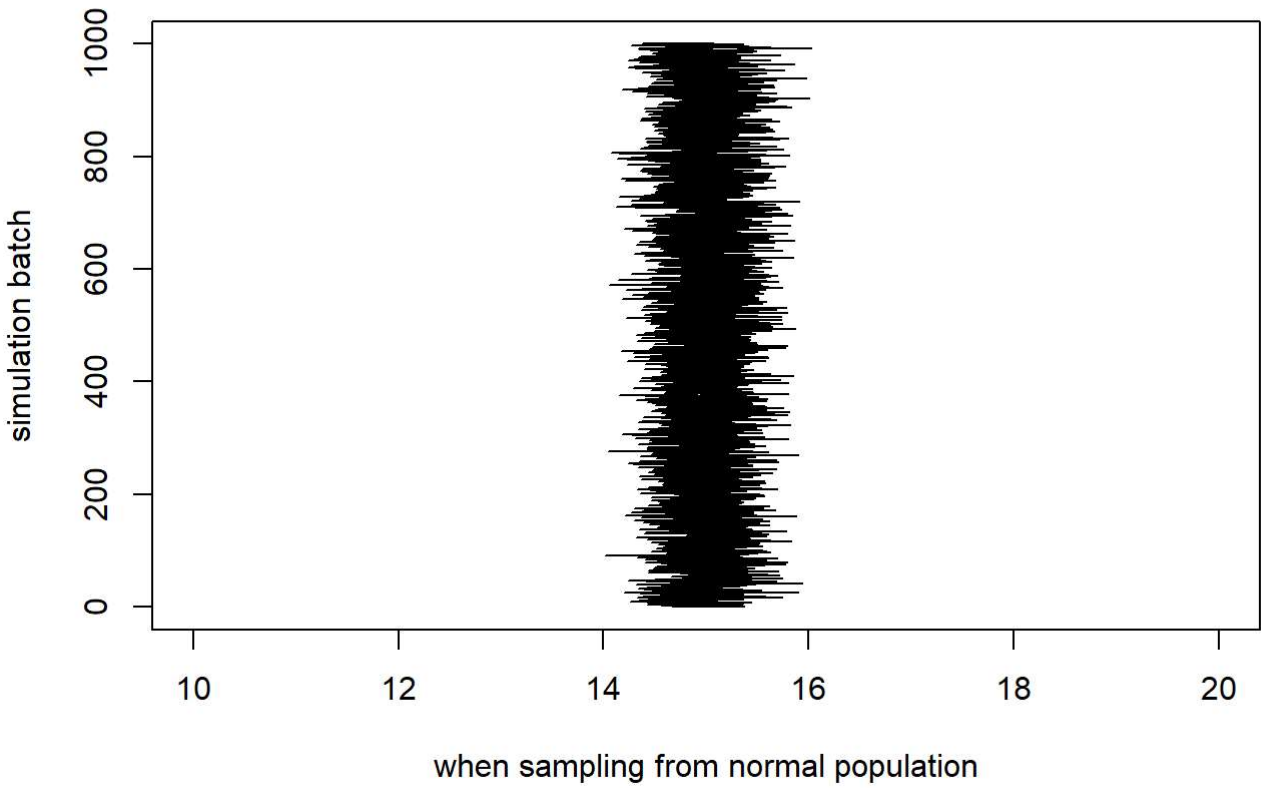
Print the estimate empirical coverage:

```
print(paste("empirical coverage is ",100*empirical,"%"))
```

```
## [1] "empirical coverage is  95 %"
```

What does the empirical coverage become when you repeat this exercise, but replace the value n=10 by n=200 AND Nbatch=20 by Nbatch=1000?

```
# copy your code above, but replace n=20 by n=200
# and rerun the code
# print the new estimation of the empirical coverage

Nbatch=1000
plot(0,0,xlim=c(10,20),ylim=c(0,Nbatch),type="n",  ylab="simulation batch", xlab="when sampling
 from normal population")

 n=200

 truemean=15;
 truesd=3;

 alpha=.1

 cover=0

for (i in 1:Nbatch){

   data=rnorm(n,truemean,truesd)

   estmean = mean(data)
   estsd   =   sd(data)

   tnx <- qt(1-alpha/2,n-1)

   tint=estmean + c(-1,1)*tnx*estsd/sqrt(n)

   count1=ifelse(tint[1]<truemean&&truemean<tint[2],1,0)

   cover=cover+count1

   lines(tint,c(i,i))

 }


lines(c(truemean,truemean),c(0,Nbatch))
```




```
    empirical=cover/Nbatch

print(paste("empirical coverage is ",100*empirical,"%"))
```

```
## [1] "empirical coverage is  88.2 %"
```

Empirical coverage seems to get closer to nominal when we increase these parameters (fill with the right word).