|  |
| --- |
| [Type the company name] |
| STAT 445 Assignment 4 |
| Kun Yang 301178299 |
|  |
|  |
|  |

|  |
| --- |
|  |

***Problem 1***

*Construct a boxplot for the third principal component for the outlier‐contaminated data set in Assignment 3. Does it do a reasonable job of highlighting a potential outlier?*

> boxplot(data2, pch=20, main="Boxplots")

> data<-read.csv("Assignment 3a Data.csv",header=0)

> data

> library("MVA")

> cov.data<-cov(data)

> data\_PCA=eigen(cov.data)

> data\_PCA

$values

[1] 1.57273753 0.83284531 0.06292275

$vectors

[,1] [,2] [,3]

[1,] 0.0664854 0.99740273 -0.02770348

[2,] -0.7188168 0.02862205 -0.69461005

[3,] -0.6920130 0.06609516 0.71885283

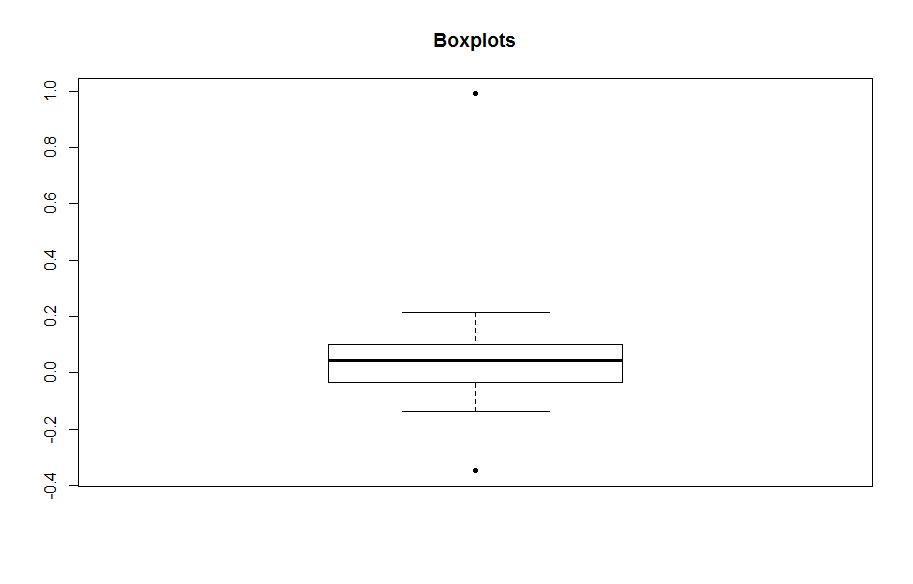
#### third principal component for the data set

> data2<-transform(data[0],newColumn=-0.02770348\*data[1]

-0.69461005\*data[,2]+0.71885283\*data[,3])

> data2

V1

1 -0.082123866

2 0.215288848

3 0.069833394

4 0.093817627

5 0.108783039

6 -0.025043634

7 0.079332266

8 0.045180274

9 -0.003479155

10 -0.045425139

11 -0.349058800

12 -0.025250178

13 0.172155319

**14 0.990143706**

15 0.048514012

16 0.181131800

17 -0.071962938

18 0.037852921

19 -0.138553196

20 0.028490516

> boxplot(data2, pch=20, main="Boxplots")

We can find a potential outlier at the 14th row. The outlier is (-0.4554475, -1.09029678, 0.30631461). The outlier is based on the third principal component.

***Problem 8.19***

> RFF<-read.csv("RecordsForFemales.csv")

> RFF<-transform(RFF[1],newColumn=100/RFF[2],newColumn=200/RFF[3],newColumn=400/RFF[4],

newColumn=800/(RFF[5]\*60),newColumn=1500/(RFF[6]\*60),

newColumn=3000/(RFF[7]\*60),newColumn=42195/(RFF[8]\*60) )

> RFF2<-RFF[,2:8]

> colnames(RFF)=c('100m/s','200m/s','400m/s','800m/s','1500m/s','3000m/s','Marm/s')

> RFF2 ###Data set in Appendix A

> cov.RFF<-cov(RFF2)

> cov.RFF

100m/s 200m/s 400m/s 800m/s 1500m/s 3000m/s Marm/s

100m/s 0.09053826 0.09560635 0.09667244 0.06506402 0.08221980 0.09214221 0.08109987

200m/s 0.09560635 0.11467144 0.11386990 0.07492487 0.09601895 0.10543645 0.09331033

400m/s 0.09667244 0.11386990 0.13778886 0.08094090 0.09544299 0.10831645 0.10188073

800m/s 0.06506402 0.07492487 0.08094090 0.07352284 0.08645423 0.09975466 0.09430563

1500m/s 0.08221980 0.09601895 0.09544299 0.08645423 0.12384050 0.14371481 0.11845777

3000m/s 0.09214221 0.10543645 0.10831645 0.09975466 0.14371481 0.17658433 0.14656043

Marm/s 0.08109987 0.09331033 0.10188073 0.09430563 0.11845777 0.14656043 0.16671409

> RFF\_PCA=eigen(cov.RFF)

> colnames(RFF\_PCA$vectors)=c('PC1','PC2','PC3','PC4','PC5','PC6','PC7')

> rownames(RFF\_PCA$vectors)=c('100m/s','200m/s','400m/s',

'800m/s','1500m/s','3000m/s','Marm/s')

> RFF\_PCA

**$values**

[1] 0.732146965 0.086071850 0.033380034 0.014977343 0.008851016 0.006167575 0.002065542

**$vectors**

PC1 PC2 PC3 PC4 PC5 PC6 PC7

100m/s -0.3102442 -0.37596510 -0.09755628 0.58479630 0.04613051 0.62433141 0.13775753

200m/s -0.3573948 -0.43376925 -0.08896099 0.32287531 0.02977941 -0.68870961 -0.31103524

400m/s -0.3787367 -0.51873227 0.27424547 -0.66667306 0.18727340 0.12377209 0.13198849

800m/s -0.2993405 0.05313551 0.05252266 -0.12808676 -0.89434367 0.13592439 -0.26472817

1500m/s -0.3912131 0.21084397 -0.43498609 -0.05510789 -0.12725405 -0.23626094 0.73364469

3000m/s -0.4595909 0.39557338 -0.42664455 -0.18388862 0.35674301 0.19925854 -0.49948755

Marm/s -0.4227291 0.44458346 0.73031571 0.23675670 0.13639673 -0.08106294 0.09516116

> PC.RFF.variance<-eigen(cov.RFF)$values

> PC.RFF.variables<-eigen(cov.RFF)$vectors

> PC.RFF.var.prop<-PC.RFF.variance/sum(PC.RFF.variance)

> PC.RFF.var.prop

[1] 0.828538913 0.097403774 0.037774734 0.016949208 0.010016310 0.006979577 0.002337484

The first principal component can be interpreted as “athlete excellence component”. All types of track competition contribute almost equally to the first component.

The second component can be called “distance component”. Because it contrasts different speed for short distances (100m, 200m, 400m) with the long distances (800m, l500m, 3000m and marathon).

> RANKF<-transform(RFF[1],newColumn=0.3102442\*RFF[2]

+0.3573948\*RFF[3]+0.3787367\*RFF[4]

+0.2993405\*RFF[5]+0.3912131\*RFF[6]

+0.4595909\*RFF[7]+0.4227291\*RFF[8])

> RANKF[order(RANKF[2],decreasing=T),]

Based on the “athlete excellence component”, the rank shows as following:

country (rank from the best to the worest)

9 CHN 18.74899

45 RUS 18.69661

18 GER 18.69560

19 GBR 18.55855

54 USA 18.77483

44 ROM 18.40235

42 POL 18.39643

29 KEN 18.04833

17 FRA 18.43057

13 CZE 18.38468

48 ESP 18.26327

2 AUS 18.30360

7 CAN 18.18658

25 IRL 18.06775

37 NED 18.11495

27 ITA 18.14123

43 POR 18.05117

35 MEX 17.94505

50 SUI 18.00629

39 NOR 17.88044

22 HUN 17.88852

28 JPN 17.81053

4 BEL 18.01584

38 NZL 17.83140

3 AUT 17.93961

16 FIN 18.02551

53 TUR 17.89332

20 GRE 17.90356

49 SWE 17.82510

6 BRA 17.86314

14 DEN 17.61793

31 KOR, N 17.10894

24 IND 17.60995

30 KOR, S 17.32747

10 COL 17.49167

8 CHI 17.23117

36 MYA 17.24067

1 ARG 17.40933

26 ISR 17.30626

32 LUX 16.98693

23 INA 16.91944

34 MRI 16.88265

51 TPE 17.20752

33 MAS 16.97362

52 THA 16.96924

12 CRC 16.72591

15 DOM 16.68779

5 BER 16.90767

41 PHI 16.81798

47 SIN 16.50441

21 GUA 16.38422

40 PNG 15.72947

11 COK 14.94404

46 SAM 14.85512

When we analyse the first and second components, I prefer the analysis based on speeds (m/s) instead of times (second), because the cumulative proportion is larger for ***Problem 8.19***(p=0.925) than ***Problem 8.18(p=0.919).*** But in general, both methods are good.

*Problem 8.20*

> RFM<-read.csv("Assignment 4a Data.csv")

> RFM<-transform(RFM[1],newColumn=100/RFM[2],newColumn=200/RFM[3],

newColumn=400/RFM[4],newColumn=800/(RFM[5]\*60),

newColumn=1500/(RFM[6]\*60),newColumn=5000/(RFM[7]\*60),

newColumn=10000/(RFM[8]\*60),newColumn=42195/(RFM[9]\*60) )

> RFM2<-RFM[,2:9]

> colnames(RFM2)=c('100m/s','200m/s','400m/s','800m/s',

'1500m/s','5000m/s','10000m/s','Marm/s')

> RFM2 ###Data set in Appendix B

> cov.RFM<-cov(RFM2)

> cov.RFM

100m/s 200m/s 400m/s 800m/s 1500m/s 5000m/s 10000m/s Marm/s

100m/s 0.04349790 0.04827718 0.04346323 0.03149513 0.04407274 0.04692523 0.04483253 0.04312562

200m/s 0.04827718 0.06484523 0.05586780 0.04323338 0.05669100 0.05877310 0.05725123 0.05629446

400m/s 0.04346323 0.05586780 0.06882169 0.04282214 0.05452718 0.06176643 0.05993536 0.05673423

800m/s 0.03149513 0.04323338 0.04282214 0.04688400 0.05357495 0.05715598 0.05539454 0.05419108

1500m/s 0.04407274 0.05669100 0.05452718 0.05357495 0.07917328 0.07846120 0.07656360 0.07552107

5000m/s 0.04692523 0.05877310 0.06176643 0.05715598 0.07846120 0.09593980 0.09373567 0.09058189

10000m/s 0.04483253 0.05725123 0.05993536 0.05539454 0.07656360 0.09373567 0.09428944 0.09099518

Marm/s 0.04312562 0.05629446 0.05673423 0.05419108 0.07552107 0.09058189 0.09099518 0.09792763

> RFM\_PCA=eigen(cov.RFM)

> colnames(RFM\_PCA$vectors)=c('PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8')

> rownames(RFM\_PCA$vectors)=c('100m/s','200m/s','400m/s','800m/s','1500m/s','5000m/s','10000m/s','Marm/s')

> RFM\_PCA

$values

[1] 0.498130294 0.046141358 0.015572444 0.012959543 0.007539391 0.006264317 0.003646366 0.001125256

$vectors

PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

100m/s -0.2431354 -0.43109847 -0.1082126 0.488066212 -0.3047838 -0.13089321 -0.61519090 -0.12897539

200m/s -0.3111771 -0.52028662 -0.1955516 0.311976868 0.3417179 -0.09967697 0.59890200 0.10728659

400m/s -0.3147271 -0.46837311 0.6287077 -0.460158940 -0.0347061 0.26393309 -0.05981611 -0.01051969

800m/s -0.2762299 -0.03282000 -0.3621606 -0.532289042 0.4210260 -0.47276921 -0.31724951 -0.08189675

1500m/s -0.3748871 0.04678827 -0.5827764 -0.243590130 -0.3527309 0.57100775 0.08384788 -0.02840245

5000m/s -0.4255004 0.26422566 0.1340482 0.004923083 -0.3485503 -0.35254265 0.06561602 0.69358630

10000m/s -0.4188931 0.31349483 0.1981696 0.075184225 -0.2171903 -0.27635878 0.28204650 -0.69154153

Marm/s -0.4142892 0.39079673 0.1635466 0.323064818 0.5646151 0.39121570 -0.26027903 0.07027692

> PC.RFM.variance<-eigen(cov.RFM)$values

> PC.RFM.variables<-eigen(cov.RFM)$vectors

> PC.RFM.var.prop<-PC.RFM.variance/sum(PC.RFM.variance)

> PC.RFM.var.prop

[1] 0.842319935 0.078023332 0.026332428 0.021914108 0.012748832 0.010592728 0.006165870 0.001902766

The first principal component can be interpreted as “athlete excellence component”. All types of track competition contribute almost equally to the first component.

The second component can be called “distance component”. Because it contrasts different speed for short distances (100m, 200m, 400m) with the long distances (800m, 1500m, 5000m, 10000m and marathon).

> RANKM<-transform(RFM[1],newColumn=0.2431354\*RFM[2]

+0.3111771\*RFM[3]+0.3147271\*RFM[4]

+0.2762299\*RFM[5]+0.3748871\*RFM[6]

+0.4255004\*RFM[7]+0.4188931\*RFM[8]+0.4142892\*RFM[9])

> RANKM[order(RANKM[2],decreasing=F),]

Based on the “athlete excellence component”, the rank shows as following:

Country

29 Kenya 92.59109

54 U.S.A. 92.61796

17 France 93.88056

19 Great Britain 93.98399

28 Japan 94.04864

6 Brazil 94.10823

4 Belgium 94.14571

43 Portugal 94.26622

35 Mexico 94.27753

27 Italy 94.28057

2 Australia 94.28799

48 Spain 94.48163

18 Gef]Jlany 94.67200

37 Netherlands 95.22267

42 Poland 95.38918

45 Russia 95.40281

30 Korea, South 95.51870

38 New Zealand 95.62350

7 Canada 95.63595

50 Switzerland 95.72590

25 Ireland 95.77488

9 China 95.93891

39 Norway 96.11141

14 Denmark 96.13945

1 Argentina 96.16233

49 Sweden 96.37426

16 Finland 96.51148

13 Czech Republic 96.99710

22 Hungary 97.03080

20 Greece 97.03159

10 Columbia 97.06984

3 Austria 97.10809

53 Turkey 97.13556

44 Romania 97.20802

31 Korea, North 97.25901

8 Chile 97.26170

23 India 97.62182

12 Costa Rica 98.71187

26 Israel 98.97260

32 Luxembourg 99.31418

21 Guatemala 99.36665

51 Taiwan 99.41645

41 Philippines 100.97296

52 Thailand 101.54687

24 Indonesia 101.63295

34 Mauritius 102.49632

36 Myanmar(Burma) 102.77826

15 DominicanRepublic 104.24767

5 Bermuda 104.58086

47 Singapore 105.26994

33 Malaysia 105.72674

40 Papua New Guinea 106.60440

46 Samoa 115.56721

11 Cook Islands 120.83608

*Appendix A (Data set for Problem 8.19)*

100m/s 200m/s 400m/s 800m/s 1500m/s 3000m/s Marm/s

1 8.643042 8.718396 7.619048 6.504065 5.882353 5.440696 4.678353

2 8.992806 8.996851 8.225375 6.734007 6.218905 5.793743 4.900355

3 8.968610 8.810573 7.902015 6.872852 6.172840 5.694761 4.556203

4 8.976661 8.896797 7.774538 6.768190 6.127451 5.668934 4.916113

5 8.726003 8.676790 7.504690 6.441224 5.827506 5.096840 4.037490

6 8.952551 8.849558 7.902015 6.768190 5.995204 5.530973 4.770708

7 9.107468 8.841733 8.014426 6.768190 6.250000 5.854801 4.740159

8 8.583691 8.389262 7.451565 6.666667 5.924171 5.399568 4.619654

9 9.267841 9.086779 8.030516 6.908463 6.510417 6.172840 5.045197

10 8.841733 8.726003 8.058018 6.535948 5.760369 5.336179 4.531542

11 7.987220 7.719027 6.488240 5.847953 5.186722 4.504505 3.312061

12 8.532423 8.361204 7.608902 6.349206 5.530973 5.081301 4.279499

13 9.017133 9.103323 8.335070 7.054674 6.203474 5.636979 4.843653

14 8.756567 8.561644 7.558579 6.600660 6.067961 5.740528 4.709053

15 8.598452 8.364701 7.544323 6.379585 5.506608 5.055612 4.224739

16 8.984726 8.932559 7.977663 6.633499 6.097561 5.753740 4.751689

17 9.319664 9.095043 8.290155 6.872852 6.203474 5.787037 4.743036

18 9.250694 9.212345 8.403361 6.944444 6.313131 5.875441 4.971721

19 9.009009 9.049774 8.092252 6.872852 6.297229 5.973716 5.199630

20 9.233610 8.822232 7.911392 6.666667 6.112469 5.580357 4.584420

21 8.389262 8.163265 7.189073 6.201550 5.580357 5.149331 4.104652

22 8.764242 8.673027 7.766990 6.700168 6.218905 5.847953 4.735690

23 8.650519 8.382230 7.262164 6.349206 5.733945 5.263158 4.557975

24 8.787346 8.764242 7.835455 6.666667 6.097561 5.488474 4.448134

25 8.748906 8.688097 7.832387 6.633499 6.281407 5.980861 4.944456

26 8.733624 8.639309 7.683442 6.441224 5.896226 5.359057 4.497634

27 8.976661 8.849558 7.795751 6.802721 6.281407 5.820722 4.901722

28 8.802817 8.572653 7.702677 6.633499 6.009615 5.720824 5.044473

29 8.605852 8.557980 7.757952 6.768190 6.313131 5.959476 5.078717

30 8.703220 8.403361 7.452953 6.379585 5.896226 5.549390 4.812825

31 8.474576 7.968127 7.113640 6.768190 5.882353 5.580357 4.839653

32 8.503401 8.347245 7.133940 6.441224 5.747126 5.428882 4.712524

33 8.695652 8.557980 7.610350 6.289308 5.694761 5.370569 4.154360

34 8.532423 8.392782 7.323325 6.472492 5.773672 5.411255 4.208810

35 9.017133 8.646779 8.181632 6.600660 5.966587 5.624297 4.881647

36 8.576329 8.442381 7.552870 6.568144 5.952381 5.506608 4.439149

37 9.025271 8.768084 7.789679 6.908463 6.157635 5.834306 4.903089

38 8.833922 8.646779 7.751938 6.768190 6.097561 5.707763 4.801652

39 8.764242 8.580009 7.626311 6.568144 6.234414 5.861665 4.985467

40 8.361204 8.103728 7.249003 5.952381 5.411255 4.897160 3.180112

41 8.865248 8.565310 7.305936 6.289308 5.668934 5.096840 4.249758

42 9.149131 9.037506 8.116883 6.837607 6.265664 5.861665 4.877584

43 8.849558 8.741259 7.704160 6.734007 6.313131 5.882353 4.907879

44 8.849558 8.948546 8.019246 6.944444 6.410256 5.980861 4.935088

45 9.285051 9.144947 8.144981 6.980803 6.459948 5.966587 4.976647

46 8.077544 7.858546 7.102273 5.822416 4.612546 3.810976 3.670790

47 8.244023 8.149959 7.262164 6.289308 5.530973 5.030181 4.554433

48 9.041591 8.936550 8.053151 6.802721 6.234414 5.896226 4.800014

49 8.960573 8.764242 7.738441 6.700168 6.112469 5.675369 4.676175

50 8.818342 8.741259 7.794232 6.734007 6.297229 5.813953 4.833001

51 8.912656 8.865248 7.584376 6.410256 5.707763 5.192108 4.408262

52 8.826125 8.583691 7.604563 6.472492 5.707763 4.965243 4.330624

53 8.888889 8.806693 7.525870 6.633499 6.377551 5.861665 4.644060

54 9.532888 9.372071 8.191685 6.872852 6.329114 5.931198 4.981935

*Appendix B (Data set for Problem 8.20)*

100m/s 200m/s 400m/s 800m/s 1500m/s 5000m/s 10000m/s Marm/s

1 9.775171 9.818360 8.661758 7.532957 6.793478 6.251563 6.027728 5.427568

2 10.070493 9.970090 9.013069 7.662835 7.082153 6.444960 6.054002 5.515254

3 9.852217 9.779951 8.733624 7.532957 6.983240 6.284565 6.012506 5.318787

4 9.861933 9.905894 8.884940 7.707129 7.002801 6.495194 6.202704 5.528695

5 9.737098 9.852217 8.837826 7.448790 6.756757 5.692168 5.466273 4.804605

6 10.000000 10.055304 9.031384 7.843137 7.002801 6.181998 5.924873 5.579135

7 10.162602 9.915716 8.944544 7.619048 7.082153 6.298816 6.038647 5.405873

8 9.900990 9.925558 8.710801 7.575758 6.849315 6.223550 5.933310 5.319994

9 9.832842 9.794319 8.839779 7.532957 6.925208 6.209637 5.916460 5.443954

10 9.718173 9.592326 8.726003 7.407407 6.720430 6.177415 5.978001 5.361363

11 9.115770 8.904720 7.782101 6.872852 5.896226 4.990020 4.710759 4.106330

12 9.689922 9.541985 8.616975 7.130125 6.510417 6.060606 5.785028 5.278466

13 9.765625 9.704027 8.739349 7.619048 6.983240 6.209637 5.995204 5.345063

14 9.718173 9.746589 8.716496 7.889546 7.102273 6.209637 5.971575 5.433439

15 9.842520 9.685230 8.908686 7.366483 6.702413 5.823433 5.477051 4.816781

16 9.794319 9.770396 8.793141 7.662835 6.925208 6.279829 6.056202 5.362181

17 9.980040 9.920635 8.960573 7.751938 7.183908 6.420134 6.087168 5.565448

18 9.940358 9.886307 9.023235 7.707129 7.082153 6.454944 6.091618 5.474041

19 10.131712 10.030090 9.017133 7.843137 7.163324 6.405329 6.105006 5.531739

20 9.891197 10.075567 8.777705 7.619048 6.925208 6.181998 5.926980 5.326038

21 9.689922 9.483167 8.257638 7.326007 6.684492 5.960897 5.680527 5.306346

22 9.920635 9.945301 8.804755 7.575758 6.963788 6.195787 5.946010 5.323618

23 9.680542 9.647853 8.795075 7.575758 6.887052 6.172840 5.785028 5.327652

24 9.803922 9.555662 8.626267 7.285974 6.631300 5.864415 5.621135 5.052809

25 9.661836 9.737098 8.775779 7.619048 7.022472 6.375925 5.999520 5.445219

26 9.803922 9.573959 8.585533 7.407407 6.756757 6.100537 5.803157 5.239923

27 9.990010 10.141988 8.837826 7.707129 7.462687 6.366183 6.109482 5.524786

28 10.000000 9.985022 8.932559 7.532957 6.906077 6.303580 6.043026 5.574271

29 9.727626 9.789525 9.053871 7.843137 7.267442 6.582412 6.298816 5.646327

30 9.671180 9.799118 8.816399 7.662835 6.868132 6.021195 5.845902 5.528695

31 9.433962 9.420631 8.519702 7.326007 6.631300 5.995204 5.858231 5.440585

32 9.606148 9.629273 8.350731 7.575758 6.811989 6.109482 5.793071 5.246960

33 9.708738 9.560229 8.618832 7.448790 6.648936 5.905977 5.649718 4.711261

34 9.871668 9.970090 8.950548 7.407407 6.527415 5.889282 5.585344 4.915426

35 9.794319 9.803922 9.027308 7.490637 6.887052 6.346789 6.140997 5.529130

36 9.398496 9.293680 8.225375 7.407407 6.578947 5.872680 5.626829 5.038690

37 9.813543 9.905894 8.756567 7.707129 7.042254 6.303580 6.073858 5.480867

38 9.891197 9.794319 8.678672 7.662835 7.062147 6.308352 6.016847 5.468932

39 9.920635 9.915716 8.674908 7.797271 6.906077 6.356471 6.051803 5.402551

40 9.615385 9.442871 8.552491 7.407407 6.250000 5.661232 5.314626 4.747519

41 9.460738 9.332711 8.777705 7.407407 6.544503 5.965163 5.739210 5.079818

42 10.000000 10.010010 8.964590 7.751938 6.963788 6.270379 5.975858 5.441848

43 10.141988 9.940358 8.674908 7.619048 7.142857 6.385696 6.125199 5.565448

44 9.794319 9.638554 8.739349 7.575758 7.002801 6.289308 6.023371 5.315571

45 9.891197 9.886307 8.968610 7.797271 7.062147 6.313131 5.973716 5.444797

46 9.276438 9.149131 8.003201 6.872852 6.234414 5.118755 4.801690 4.354489

47 9.643202 9.460738 8.403361 7.246377 6.476684 5.570410 5.321413 4.876231

48 9.832842 9.713453 8.896797 7.707129 7.183908 6.390593 6.118453 5.527391

49 9.823183 9.789525 8.783487 7.575758 6.925208 6.270379 5.967299 5.393849

50 9.842520 9.799118 8.890865 7.797271 7.082153 6.346789 5.973716 5.427987

51 9.652510 9.610764 8.561644 7.448790 6.631300 5.990894 5.707763 5.234462

52 9.775171 9.666506 8.686211 7.366483 6.631300 5.847953 5.617346 5.047370

53 9.633911 9.505703 8.578169 7.490637 6.963788 6.195787 5.883045 5.399232

54 10.224949 10.351967 9.263548 7.797271 7.225434 6.425084 6.120700 5.608949