SML Assignment

Agbatan Fiacre Luc KOUDERIN

13/12/2024

EXERCICE 1 : Pratical SML on DNA Microarrays

Main libraries

```
library(class)
library(rpart)
library(rpart.plot)
library(ROCR)
```

1. Let's comment on the shape of this dataset

Let's load the data and print the first few rows

```
prostate = read.csv("prostate-cancer-1.csv")
head(prostate)
```

```
##      Y  X206212_at X207075_at X215872_at   X201876_at X211935_at X206788_s_at
## PG13 0 -0.21774839 -0.3399249 -0.3543970 0.3026506265  0.6148128   -0.1541378
## PG15 0 -0.13796955 -0.2659108 -0.2769138 0.0908803171  0.1133694   -0.1751358
## PG37 0 -0.14751340 -0.2624912 -0.2622395 0.0280230537  0.3519450   -0.1946007
## PG41 0 -0.19073798 -0.3259540 -0.3256833 0.3939189385  0.4115226   -0.1796192
## PG46 0 -0.09916344 -0.2076265 -0.2472927 0.0004993612  0.4157633   -0.1646695
## PG52 0 -0.17484954 -0.2837072 -0.2682238 0.1965511485  0.2685327   -0.2148018
##      X216441_at X209290_s_at X219877_at X220675_s_at X204229_at X216460_at
## PG13 -0.3563217   0.34631198 -0.3476009   -0.3519888 -0.3248372 -0.3496970
## PG15 -0.2833936   0.28588602 -0.2797931   -0.2804227 -0.2637505 -0.2729537
## PG37 -0.2608892   0.28523579 -0.2549100   -0.2597367 -0.2518594 -0.2511765
## PG41 -0.3264282   0.29499193 -0.3237691   -0.3246642 -0.3076725 -0.3224269
## PG46 -0.2417837   0.09958763 -0.2386715   -0.2477095 -0.2004705 -0.2435681
## PG52 -0.2883585   0.52926543 -0.2855168   -0.2815082 -0.2800032 -0.2831115
##      X215861_at X207287_at X211875_x_at  X205055_at X216887_s_at X213319_s_at
## PG13 -0.3141736 -0.3008636   -0.3483616  0.14709095   -0.3098196   -0.3305093
## PG15 -0.2298867 -0.2597878   -0.2751802  0.04645510   -0.1759998   -0.2523219
## PG37 -0.2083398 -0.2199262   -0.2549477  0.09122362   -0.2231047   -0.2471991
## PG41 -0.2773771 -0.2907855   -0.3206230  0.13114897   -0.2862452   -0.3152328
## PG46 -0.2250435 -0.2119762   -0.2475110  0.11626838   -0.2408154   -0.2193927
## PG52 -0.2428249 -0.2517216   -0.2842899 -0.05417943   -0.2189467   -0.2740198
##      X220709_at X204011_at X216174_at X219416_at X200793_s_at X216050_at
## PG13 -0.3491685 -0.2051633 -0.2837889 -0.2635437   0.04699453 -0.3565952
## PG15 -0.2830467 -0.1125013 -0.1984975 -0.1472836  -0.03377549 -0.2833850
## PG37 -0.2531432 -0.2093250 -0.2342251 -0.2027162  -0.01005613 -0.2627631
## PG41 -0.3229007 -0.2576061 -0.2736013 -0.2712478   0.37420779 -0.3295772
## PG46 -0.2436139 -0.1797184 -0.1748806 -0.2035605   0.02012378 -0.2400896
```

```
## PG52 -0.2867026 -0.2076603 -0.2729685 -0.2223393   0.32448345 -0.2842491
##      X216788_at   X201699_at X201610_at X221695_s_at X219889_at   X201632_at
## PG13 -0.3511941  0.274375228 -0.3533065   -0.1883429 -0.1902415  0.143634620
## PG15 -0.2786439 -0.003279215 -0.2821431   -0.1925493 -0.1566982  0.065649928
## PG37 -0.2544173 -0.042190422 -0.2597501   -0.1717068 -0.1516171  0.025082746
## PG41 -0.3230645  0.079266935 -0.3173522   -0.2544781 -0.2222404  0.156506903
## PG46 -0.2464061  0.036736976 -0.2155083   -0.1763517 -0.1158782 -0.009295204
## PG52 -0.2838473 -0.014966467 -0.2877685   -0.2021123 -0.1561414  0.049491295
##       X218164_at X214007_s_at  X215584_at X213817_at X207195_at  X203386_at
## PG13  0.24665688   0.02566953 -0.26663641 -0.3398453 -0.3497000 -0.11782749
## PG15  0.12050925  -0.16728662 -0.09010084 -0.2303066 -0.2739653  0.05463543
## PG37  0.10933394  -0.18170909 -0.14901594 -0.2302804 -0.2339273  0.24136939
## PG41  0.58022167  -0.14819300 -0.19938966 -0.3038371 -0.3271358  0.10054556
## PG46 -0.02976805  -0.18282802 -0.09841326 -0.2322822 -0.2449332  0.05039951
## PG52  0.37475956  -0.20123761 -0.16818855 -0.2873294 -0.2688461  0.16404847
##      X208861_s_at X206202_at X214767_s_at X209454_s_at X203216_s_at
## PG13   0.28605698 -0.3530564  -0.20355960  -0.20204732   0.56677425
## PG15   0.24604311 -0.2851579   0.02965385  -0.17008936  -0.14372274
## PG37   0.03530036 -0.2624785  -0.12742245  -0.11057217  -0.04748615
## PG41   0.22277863 -0.3114331  -0.18716924  -0.13606041   2.39057162
## PG46   0.14651692 -0.2521036  -0.11991976  -0.19713898   0.19202892
## PG52   0.23798872 -0.2902357  -0.17316664  -0.09186752   0.13973521
##      X222314_x_at X213009_s_at X208243_s_at X204742_s_at X214451_at
## PG13   -0.3485037   0.20249888   -0.3474458   -0.2894569 -0.3540157
## PG15   -0.2431937   0.06413411   -0.2802797   -0.2447921 -0.2815771
## PG37   -0.2552413   0.16274403   -0.2580054   -0.1966423 -0.2632765
## PG41   -0.3227867   0.20300829   -0.3264951   -0.2803777 -0.3224908
## PG46   -0.2453023  -0.01404611   -0.2434611   -0.2224046 -0.2372606
## PG52   -0.2846521   0.20182196   -0.2807054   -0.2373065 -0.2884649
##      X206296_x_at X221183_at X208087_s_at X212939_at X221662_s_at X212707_s_at
## PG13   -0.3469409 -0.3428994   -0.3464954 -0.3492401   -0.3278019   -0.2783247
## PG15   -0.2728619 -0.2138068   -0.2794345 -0.2610788   -0.2760298   -0.2369230
## PG37   -0.2562814 -0.2479551   -0.2596284 -0.2560260   -0.2328374   -0.2569849
## PG41   -0.3245838 -0.3170999   -0.3247783 -0.3225719   -0.3061316   -0.2217630
## PG46   -0.1859523 -0.2080768   -0.2291360 -0.2417867   -0.2448529   -0.1683261
## PG52   -0.2864390 -0.2621651   -0.2831963 -0.2859593   -0.2852696   -0.2437920
##      X220995_at X207780_at X204905_s_at X213631_x_at X205715_at X219849_at
## PG13 -0.2916161 -0.3237928  0.080660979   -0.2825929 -0.3510803 -0.2781870
## PG15 -0.2237092 -0.2406688  0.001412018   -0.2279035 -0.2773175 -0.2685818
## PG37 -0.2074726 -0.2327742  0.032502295   -0.2306954 -0.2627378 -0.2217287
## PG41 -0.2500546 -0.3140963  0.195706297   -0.2878709 -0.3262208 -0.2691937
## PG46 -0.1471656 -0.1840697  0.010091943   -0.1866634 -0.2202662 -0.1969716
## PG52 -0.2206902 -0.2293088 -0.027497919   -0.2146018 -0.2874858 -0.2090026
##      X216394_x_at X216274_s_at X216794_at X216782_at X222183_x_at  X204711_at
## PG13   -0.3501942    1.0197730 -0.3503652 -0.3510940   -0.3505852 -0.15997087
## PG15   -0.2657271    1.1165592 -0.2729689 -0.2794844   -0.2766975 -0.09160410
## PG37   -0.2329015    0.4967474 -0.2587489 -0.2608631   -0.2602964 -0.08257470
## PG41   -0.3171109    0.6546786 -0.3185732 -0.3244466   -0.3226753 -0.17855769
## PG46   -0.2305912    0.5464926 -0.2390467 -0.2464367   -0.2160703 -0.09655165
## PG52   -0.2817938    0.3797720 -0.2778642 -0.2861106   -0.2773892 -0.09804230
##      X211491_at X206023_at X201899_s_at X208531_at X211646_at  X218261_at
## PG13 -0.3192752 -0.3528606   0.30816313 -0.3498723 -0.3490394  0.19469810
## PG15 -0.2777301 -0.2840924   0.05857068 -0.2797999 -0.2848202 -0.04774288
## PG37 -0.2312931 -0.2632968   0.14874055 -0.2457789 -0.2636988  0.04495810
```

```
## PG41 -0.3235916 -0.3286206   0.23646930 -0.3264070 -0.3281047  0.10801128
## PG46 -0.2094264 -0.2497054   0.05620684 -0.2481823 -0.2441160 -0.11902461
## PG52 -0.2190111 -0.2875162   0.07850907 -0.2888247 -0.2870874  0.07915637
##      X200989_at X216665_s_at X209353_s_at X216820_at X215862_at X217122_s_at
## PG13  2.2725544   -0.3451688   -0.3505107 -0.3332113 -0.3369121    1.2187354
## PG15  0.7051982   -0.2571074   -0.2641723 -0.2804276 -0.2738264    0.5854046
## PG37  0.6883576   -0.2553882   -0.2616277 -0.2517107 -0.2629569    0.6893889
## PG41  1.2117159   -0.3157572   -0.3218633 -0.3261419 -0.3027613    1.4159559
## PG46  0.5556536   -0.2212628   -0.2366552 -0.2451220 -0.2440253    0.4486184
## PG52  0.7959165   -0.2889288   -0.2711612 -0.2691405 -0.2786511    0.8172576
##      X215180_at X208048_at X210808_s_at X215801_at X221209_s_at X215524_x_at
## PG13 -0.3035060 -0.3259374   -0.3376460 -0.3053783   -0.3316742   -0.3128176
## PG15 -0.2483377 -0.2537161   -0.2799988 -0.2302221   -0.2460974   -0.2474188
## PG37 -0.2319344 -0.2307367   -0.2531383 -0.1802830   -0.2420769   -0.2408566
## PG41 -0.3142675 -0.2780302   -0.3247376 -0.2699307   -0.3029098   -0.3170472
## PG46 -0.2099009 -0.2241434   -0.2467599 -0.1787282   -0.2057660   -0.1697024
## PG52 -0.2370987 -0.2448042   -0.2742996 -0.2298669   -0.2444991   -0.2617547
##      X208242_at X210565_at X216953_s_at X207461_at X216800_at X207285_x_at
## PG13 -0.3502010 -0.3402634   -0.3531664 -0.3431062 -0.3532580   -0.3518090
## PG15 -0.2794821 -0.2665681   -0.2787602 -0.2710486 -0.2744357   -0.2804772
## PG37 -0.2573100 -0.2598099   -0.2614862 -0.2518571 -0.2564151   -0.2592499
## PG41 -0.3216623 -0.3197441   -0.3243850 -0.3220370 -0.3247029   -0.3257883
## PG46 -0.2424216 -0.2485671   -0.2471954 -0.2317867 -0.2485303   -0.2468308
## PG52 -0.2801963 -0.2766467   -0.2865008 -0.2819556 -0.2816648   -0.2865017
##      X216057_at X217469_at X217919_s_at X215027_at X202359_s_at  X221761_at
## PG13 -0.3543631 -0.3529147   0.47186824 -0.3052142   -0.2209473  0.08195972
## PG15 -0.2831273 -0.2789323  -0.02005201 -0.2121251   -0.2309542 -0.12411490
## PG37 -0.2528117 -0.2605641   0.01474008 -0.2361953   -0.1577925 -0.05825734
## PG41 -0.3283234 -0.3200853   0.30252286 -0.2853954   -0.1091633  0.11143340
## PG46 -0.2492254 -0.2410963  -0.07285579 -0.2202662   -0.1971746 -0.01754071
## PG52 -0.2588766 -0.2859306   0.03928611 -0.2787183   -0.1881588  0.01768011
##      X221093_at X210493_s_at X202089_s_at X222124_at X210055_at  X204381_at
## PG13 -0.3537613   -0.3491707    1.1396754 -0.3381842 -0.3530894 -0.28218326
## PG15 -0.2742263   -0.2644679    0.3542234 -0.2619206 -0.2805212 -0.21781856
## PG37 -0.2610999   -0.2543404    0.7049999 -0.2456118 -0.2636537 -0.07763575
## PG41 -0.3276017   -0.3201635    0.8439846 -0.3098382 -0.3277427 -0.14456897
## PG46 -0.2431682   -0.2344731    0.2928078 -0.2261073 -0.2350626 -0.18674344
## PG52 -0.2833783   -0.2791309    0.6434007 -0.2739872 -0.2813896 -0.16764792
##      X215031_x_at X207848_at X220889_s_at X219829_at X208557_at X205082_s_at
## PG13   -0.2082292 -0.3460228   -0.3491542 -0.3520540 -0.3256170   -0.3099986
## PG15   -0.1483631 -0.2565421   -0.2800618 -0.2770471 -0.2561583   -0.1901084
## PG37   -0.1163792 -0.2562335   -0.2539561 -0.2417862 -0.2205938   -0.2316277
## PG41   -0.1511308 -0.3255852   -0.3174962 -0.3272458 -0.2782163   -0.2938350
## PG46   -0.1171073 -0.2077188   -0.2499496 -0.2180072 -0.1953350   -0.2073648
## PG52   -0.1359432 -0.2715531   -0.2844384 -0.2829338 -0.2377588   -0.2792357
##      X208017_s_at X213691_at X213810_s_at X206547_s_at X207853_s_at
## PG13   -0.3517722 -0.3322525   -0.3157025   -0.3164824   -0.3485811
## PG15   -0.2649026 -0.2642797   -0.2738590   -0.2802019   -0.2727152
## PG37   -0.2333280 -0.2225090   -0.2562715   -0.2420102   -0.2541374
## PG41   -0.3244336 -0.3020703   -0.3057400   -0.2863592   -0.3217910
## PG46   -0.2058026 -0.1910928   -0.2271869   -0.2105586   -0.2348736
## PG52   -0.2843664 -0.2610697   -0.2450642   -0.2654979   -0.2726780
##      X208374_s_at X211660_at X206338_at X220850_at X217283_at X215738_at
## PG13    0.9309330 -0.2576853 -0.3491219 -0.3536786 -0.3490346 -0.3220609
```

```
## PG15    0.3713111 -0.2301465 -0.2678917 -0.2823507 -0.2712795 -0.2350097
## PG37    0.3766132 -0.1811044 -0.2524169 -0.2435082 -0.2612498 -0.2171047
## PG41    0.5640255 -0.2549269 -0.3206483 -0.3239377 -0.3250322 -0.3041525
## PG46    0.6297305 -0.1882410 -0.2455914 -0.2405721 -0.2340557 -0.2162730
## PG52    0.4717318 -0.2038162 -0.2801612 -0.2774752 -0.2846459 -0.2336684
##       X218230_at X210244_at X214897_at X216634_at X220656_at X207100_s_at
## PG13  0.01199624 -0.3515182 -0.3507273 -0.3379717 -0.3555534   -0.3023835
## PG15 -0.16330251 -0.2801427 -0.2682960 -0.2844035 -0.2766945   -0.2278580
## PG37 -0.10859930 -0.2601154 -0.2580967 -0.2591543 -0.2643922   -0.2619455
## PG41 -0.01988028 -0.3237784 -0.3134936 -0.3286987 -0.3170609   -0.3082859
## PG46 -0.13405686 -0.2428713 -0.2439505 -0.2467230 -0.2504893   -0.1908157
## PG52 -0.06248865 -0.2850118 -0.2516126 -0.2892687 -0.2836101   -0.2621920
##      X216995_x_at X217844_at X218789_s_at X204153_s_at X205692_s_at
## PG13   -0.3530041  0.4022477   0.09126012   -0.2847932   -0.2975519
## PG15   -0.2788943  0.3688322  -0.16089658   -0.1743444   -0.1172514
## PG37   -0.2569621  0.3600720  -0.10526382   -0.2560637   -0.1793568
## PG41   -0.3151658  0.2100735  -0.06170316   -0.2889406    0.3532940
## PG46   -0.2361550  0.1164700  -0.17228148   -0.1122006    0.1691518
## PG52   -0.2859941  0.5831492  -0.11470501   -0.2157753   -0.2432652
##      X217215_s_at X217716_s_at X213873_at  X218732_at X220359_s_at X214471_x_at
## PG13   -0.3532257    1.3266009 -0.2423098  0.13192570   -0.3503187   -0.2592070
## PG15   -0.2781366    0.5711657 -0.2217502 -0.12518909   -0.2691128   -0.1473693
## PG37   -0.2588284    0.7775839 -0.2281468 -0.00423901   -0.2599069   -0.1537683
## PG41   -0.3250217    0.8870341 -0.2571788 -0.03053263   -0.3258042   -0.2279463
## PG46   -0.2411539    0.1633896 -0.1832044 -0.07151975   -0.2457785   -0.1632374
## PG52   -0.2841437    0.7869725 -0.2147727 -0.00542693   -0.2834995   -0.1717975
##      X205814_at X217663_at X213191_at X214267_s_at X207933_at X201521_s_at
## PG13 -0.3083388 -0.3478473 -0.2759976   -0.3512603 -0.3524861  -0.06529916
## PG15 -0.2447329 -0.2750102 -0.2312804   -0.2774280 -0.2830350  -0.13683617
## PG37 -0.2567169 -0.2550709 -0.2287710   -0.2538028 -0.2607902  -0.18089556
## PG41 -0.3010324 -0.3091663 -0.2521661   -0.3245239 -0.3214275  -0.16119273
## PG46 -0.2119171 -0.2264926 -0.1720927   -0.2490452 -0.2473300  -0.18510570
## PG52 -0.2719821 -0.2648383 -0.2274234   -0.2879391 -0.2884447  -0.18988562
##      X200996_at  X201975_at X200604_s_at X220553_s_at X221658_s_at X204424_s_at
## PG13  0.6037776  0.04291818   0.42170703   -0.2146048   -0.3478597  -0.19119885
## PG15  0.2754555 -0.09695608   0.18715027   -0.1818993   -0.2792853   0.05807041
## PG37  0.1317817 -0.07639411  -0.08278124   -0.1652296   -0.2554607  -0.11268871
## PG41  0.5198638 -0.12479432   0.09189116   -0.2038307   -0.3231346  -0.21641462
## PG46  0.5181335 -0.10502000   0.03204744   -0.1384256   -0.2047610  -0.08256872
## PG52  0.2029486 -0.03091405  -0.03792594   -0.1534677   -0.2844762  -0.13337613
##      X202132_at X204418_x_at X211094_s_at X217101_at X212099_at X215184_at
## PG13 -0.2500822  -0.05791861   -0.3060090 -0.3537361  1.6440992 -0.3308612
## PG15 -0.1156601   0.33559309   -0.2369096 -0.2805120  2.0534894 -0.2722844
## PG37 -0.2219047   0.10636125   -0.2180526 -0.2456637  0.9552120 -0.2609423
## PG41 -0.2436641  -0.05541325   -0.3225938 -0.3211921  1.7556314 -0.2748946
## PG46 -0.1689864   0.06853758   -0.2381880 -0.2285075  0.5361227 -0.2422152
## PG52 -0.2178263   0.02280216   -0.2685737 -0.2848896  1.6976300 -0.2379475
##      X213560_at X216423_at X205024_s_at X209916_at X215402_at X206532_at
## PG13 -0.3453982 -0.3539957   -0.2935757 -0.2213850 -0.3503974 -0.3542795
## PG15 -0.2102908 -0.2684109   -0.2781905 -0.2177247 -0.2820900 -0.2834920
## PG37 -0.2546809 -0.2013233   -0.1951926 -0.1027546 -0.2602986 -0.2500856
## PG41 -0.2779579 -0.2976870   -0.2821592 -0.2182229 -0.3110240 -0.3058091
## PG46 -0.2198493 -0.2442857   -0.2005561 -0.1712797 -0.2310763 -0.2461550
## PG52 -0.2789325 -0.2539124   -0.2673002 -0.1991627 -0.2863028 -0.2848272
```

```
##      X221393_at X220384_at X218747_s_at X218133_s_at X219739_at X215756_at
## PG13 -0.3469799 -0.3549349  -0.20569851  -0.07856455 -0.2962822 -0.2560211
## PG15 -0.2744191 -0.2834871  -0.20613388  -0.03029964 -0.2212585 -0.2081705
## PG37 -0.2559768 -0.2609256  -0.19822896  -0.04494749 -0.2516793 -0.1847067
## PG41 -0.3223403 -0.3241072  -0.22452571  -0.12830703 -0.2850070 -0.2517087
## PG46 -0.2454276 -0.2492044  -0.05181231  -0.07623079 -0.1941451 -0.1829153
## PG52 -0.2836435 -0.2888172  -0.20887449  -0.05569367 -0.2275727 -0.1902460
##      X208462_s_at X208513_at X211233_x_at X216025_x_at X201450_s_at
## PG13   -0.2826487 -0.3474051   -0.3366792   -0.2499829   -0.1428424
## PG15   -0.2371765 -0.2801814   -0.2543384   -0.1800470   -0.1707084
## PG37   -0.2144339 -0.2630214   -0.2243328   -0.2050508   -0.1847376
## PG41   -0.2883941 -0.3235246   -0.2867244   -0.2460858   -0.1859829
## PG46   -0.1990745 -0.2509351   -0.2330330   -0.1510036   -0.1101731
## PG52   -0.2311489 -0.2859058   -0.2727051   -0.1837793   -0.1625465
##      X222297_x_at X217323_at X219185_at  X212864_at X215417_at  X52159_at
## PG13  -0.13350039 -0.3546060 -0.2099911 -0.04587124 -0.3479543 -0.2139086
## PG15  -0.05415811 -0.2769770 -0.1656947 -0.12405495 -0.2761821 -0.1413280
## PG37  -0.11270141 -0.2564414 -0.1523696 -0.10904925 -0.2477678 -0.1291448
## PG41  -0.13211823 -0.3224898 -0.2050478 -0.07380881 -0.2952573 -0.2019720
## PG46  -0.07677508 -0.2508834 -0.1524862 -0.11058260 -0.1909918 -0.1372086
## PG52  -0.05615322 -0.2848455 -0.1995305 -0.07709580 -0.2505280 -0.1159344
##      X220503_at X210676_x_at X221420_at X207964_x_at X207743_at X211910_at
## PG13 -0.3567283   0.08201616 -0.3563790   -0.3020466 -0.3413780 -0.3493706
## PG15 -0.2842373  -0.04261384 -0.2764088   -0.2108744 -0.2698114 -0.2803509
## PG37 -0.2606768  -0.03524903 -0.2638858   -0.2045321 -0.2524184 -0.2429536
## PG41 -0.3230846   0.18215982 -0.3293372   -0.2776221 -0.3205000 -0.3275129
## PG46 -0.2497007   0.05434597 -0.2419377   -0.1615011 -0.2386581 -0.2480701
## PG52 -0.2821287   0.01211988 -0.2899546   -0.1932716 -0.2801621 -0.2814872
##      X202093_s_at X210326_at X204708_at X214254_at X212455_at X214961_at
## PG13  -0.14419390 -0.3431369 -0.3384186 -0.3544713  0.9240327 -0.3405380
## PG15  -0.13561691 -0.2649506 -0.2676592 -0.2611734  0.6390162 -0.2827937
## PG37  -0.07327198 -0.2308992 -0.2471989 -0.2365488  0.4696356 -0.2603045
## PG41  -0.06152704 -0.3169512 -0.3164226 -0.3238795  0.7609564 -0.3015908
## PG46  -0.12671133 -0.2441366 -0.2460445 -0.2482800  0.6390327 -0.2433895
## PG52  -0.04652936 -0.2814946 -0.2877167 -0.2603637  0.8508722 -0.2724810
##        X204294_at X218833_at X207887_s_at X215816_at X217406_at X222037_at
## PG13 -0.024815498 -0.3067729   -0.3537327 -0.3092314 -0.2993552 -0.3091246
## PG15  0.048160587 -0.2108639   -0.2484297 -0.2471824 -0.2175899 -0.2618769
## PG37 -0.047107634 -0.2181548   -0.2075872 -0.2380005 -0.2266785 -0.2227982
## PG41 -0.004170073 -0.2784707   -0.2659511 -0.3065627 -0.2678796 -0.2957478
## PG46 -0.013264285 -0.2176139   -0.2498096 -0.1922008 -0.1962742 -0.2428650
## PG52  0.033849405 -0.2541936   -0.2742103 -0.2369095 -0.2082096 -0.2879086
##      X202735_at X209812_x_at X204443_at X220182_at X209048_s_at X205602_x_at
## PG13 -0.1278737   -0.3473577 -0.2167977 -0.3379630  -0.10306693   -0.3102091
## PG15 -0.1214101   -0.2746875 -0.2023814 -0.2515237  -0.06217449   -0.2272941
## PG37 -0.1121377   -0.2539280 -0.1226227 -0.2217197   0.03812874   -0.2544731
## PG41 -0.1985740   -0.3259090 -0.1903856 -0.2985500   0.13489866   -0.3186283
## PG46 -0.1649286   -0.2478408 -0.2164674 -0.2470471  -0.05646315   -0.1952431
## PG52 -0.1019818   -0.2872480 -0.2442245 -0.2733836  -0.04985871   -0.2746344
##      X215161_at X210532_s_at X215333_x_at X210525_x_at X205428_s_at X214008_at
## PG13 -0.3521534    1.7742752 -0.132489831   -0.3553861   -0.2950307 -0.3286241
## PG15 -0.2793098    1.0091744  0.172987078   -0.2854918   -0.2281887 -0.2815058
## PG37 -0.2625133    1.2112259 -0.082225304   -0.2626863   -0.2063014 -0.2597282
## PG41 -0.3256550    1.4725034 -0.133433945   -0.3269800   -0.2932836 -0.3040463
```

```
## PG46 -0.2473844    0.9786604 -0.005912159   -0.2424620   -0.1679625 -0.2471187
## PG52 -0.2864243    1.3522620 -0.072447707   -0.2852854   -0.2491420 -0.2872754
##      X204058_at X210766_s_at X209757_s_at X216584_at X204030_s_at X213611_at
## PG13 -0.3055459  0.008393222   -0.3462937 -0.3524905   -0.2054456 -0.3520035
## PG15 -0.2549280 -0.059509318   -0.2800831 -0.2796023   -0.1516894 -0.2790923
## PG37 -0.2481919  0.093363325   -0.2253105 -0.2603271   -0.2176429 -0.2615443
## PG41 -0.2158073 -0.017621780   -0.3263869 -0.3224590   -0.2466018 -0.3273345
## PG46 -0.2215104 -0.078023754   -0.2470728 -0.2474031   -0.1830331 -0.2475304
## PG52 -0.2635872  0.033640562   -0.2767866 -0.2855014   -0.1446491 -0.2854011
##          X823_at AFFX.ThrX.3_at X218675_at X215106_at X206317_s_at  X218984_at
## PG13 -0.15252434     -0.3529233 -0.2818548 -0.3532389   -0.3473619 -0.02501931
## PG15 -0.11099862     -0.2830795 -0.1343283 -0.2654663   -0.2666906 -0.07063538
## PG37 -0.16415722     -0.2566875 -0.1701557 -0.2613664   -0.2552531 -0.04155929
## PG41 -0.19800510     -0.3275218 -0.1041953 -0.3221501   -0.3141715 -0.02891484
## PG46 -0.01584755     -0.2451439 -0.1912854 -0.2451663   -0.2443784 -0.12985551
## PG52 -0.08582120     -0.2866020 -0.1318992 -0.2847882   -0.2864343 -0.07887534
##      X222112_at X206071_s_at X200047_s_at X208907_s_at X217000_at X214856_at
## PG13 -0.3547096   -0.3080788    0.6260850  -0.03204706 -0.3542095 -0.3559525
## PG15 -0.2830813   -0.1242571    0.3640370  -0.10069003 -0.2780672 -0.2703098
## PG37 -0.2628034   -0.2164419    0.2459493  -0.07057325 -0.2621588 -0.2627411
## PG41 -0.3275464   -0.2726141    0.5579900  -0.09017217 -0.3269013 -0.3287376
## PG46 -0.2468949   -0.2072941    0.3326458  -0.07083951 -0.2468389 -0.2483977
## PG52 -0.2868715   -0.2691596    0.3150429  -0.07581713 -0.2885451 -0.2887738
##      X211446_at X202610_s_at X207658_s_at X219597_s_at X220488_s_at
## PG13 -0.2917524  -0.06311598   -0.3479942   -0.3106046   -0.1768352
## PG15 -0.2576454  -0.22442492   -0.2769036   -0.2219610   -0.1318737
## PG37 -0.2301607  -0.20584504   -0.2556036   -0.2351629   -0.1666119
## PG41 -0.2771263  -0.09134140   -0.3033217   -0.2824539   -0.1833147
## PG46 -0.1805891  -0.14964316   -0.2421293   -0.2370520   -0.1780479
## PG52 -0.2514857  -0.10662208   -0.2829980   -0.2452147   -0.1355484
##      X207890_s_at X216437_at  X201626_at X217636_at X208102_s_at X213814_s_at
## PG13  -0.27082200 -0.3153345  0.07157226 -0.3434494   -0.3432379   -0.3055573
## PG15  -0.16066248 -0.2452674 -0.03316665 -0.2792848   -0.2551823   -0.2257438
## PG37  -0.09654151 -0.2356037 -0.05491894 -0.2524759   -0.2443638   -0.2533475
## PG41  -0.23507536 -0.2988759  0.53646043 -0.3274453   -0.3215575   -0.3152048
## PG46  -0.19120073 -0.1952043 -0.08648621 -0.2303972   -0.2384163   -0.2138721
## PG52  -0.22149537 -0.2421465 -0.12078485 -0.2812970   -0.2794252   -0.2598410
##      X215746_at X207276_at X215730_at  X212419_at X210103_s_at X212352_s_at
## PG13 -0.3377454 -0.3511056 -0.3375299  0.09710839   -0.3485609    1.9326396
## PG15 -0.2660068 -0.2837646 -0.2683426  0.26058904   -0.2749482    1.0627336
## PG37 -0.2445158 -0.2635393 -0.2390286  0.07126012   -0.2523878    1.2337182
## PG41 -0.3021970 -0.3296902 -0.3152069  0.13459931   -0.3253974    1.6523498
## PG46 -0.2273171 -0.2521929 -0.2207537 -0.15499391   -0.2396076    0.6685769
## PG52 -0.2817915 -0.2904313 -0.2636470  0.04536592   -0.2796246    1.5776704
##      X217034_at X209597_s_at X212433_x_at X205625_s_at X215987_at X219546_at
## PG13 -0.3501421   -0.3451130     8.483072   -0.3338012 -0.3349518 -0.2940655
## PG15 -0.2716538   -0.2780133     8.036509   -0.2453366 -0.2455755 -0.2778552
## PG37 -0.2593116   -0.2601835     8.767435   -0.2409847 -0.2212512 -0.2610087
## PG41 -0.2897808   -0.3259705     7.278126   -0.3009577 -0.3011032 -0.2918501
## PG46 -0.2313164   -0.2481317     7.447792   -0.2253733 -0.2307878 -0.2123241
## PG52 -0.2564765   -0.2752422     8.848964   -0.2717447 -0.2760888 -0.2857531
##      X216731_s_at X214509_at X220449_at X217974_at X215516_at X207629_s_at
## PG13   -0.3549837 -0.3516958 -0.2809285 -0.3035353 -0.3478492   -0.1919471
## PG15   -0.2813315 -0.2693004 -0.1512874 -0.2479129 -0.2637182   -0.1842073
```

```
## PG37   -0.2619056 -0.2275347 -0.1948893 -0.2355591 -0.2502840   -0.2140830
## PG41   -0.3248079 -0.3233223 -0.2820134 -0.3069617 -0.3180132   -0.1779774
## PG46   -0.2493241 -0.2372766 -0.1127293 -0.2314875 -0.2230174   -0.1021530
## PG52   -0.2827586 -0.2861436 -0.1887709 -0.2474862 -0.2727288   -0.1949033
##      X216233_at X205616_at X222181_at X221127_s_at X208245_at X217495_x_at
## PG13 -0.3362664 -0.3067980 -0.3346378  -0.27499542 -0.3571417   -0.3493274
## PG15 -0.2797409 -0.2677545 -0.2571102  -0.15910320 -0.2841475   -0.2787647
## PG37 -0.2621780 -0.2110477 -0.2431026  -0.08067347 -0.2517645   -0.2602273
## PG41 -0.3126860 -0.2877885 -0.3133678  -0.23816744 -0.3255473   -0.3261626
## PG46 -0.2122950 -0.1747175 -0.2111597  -0.16632584 -0.2317109   -0.2449997
## PG52 -0.2868481 -0.2636613 -0.2701956  -0.25017626 -0.2902252   -0.2849683
##      X205386_s_at X218161_s_at X216351_x_at X200023_s_at X204265_s_at
## PG13   -0.3306421   -0.3126659   -0.2484031    1.5645480 -0.217139465
## PG15   -0.2620902   -0.2634413   -0.2130424    1.8828951 -0.110300071
## PG37   -0.2613709   -0.2180048   -0.1825174    1.4559835 -0.151984215
## PG41   -0.3112286   -0.2908612   -0.2773654    1.3881390 -0.198360725
## PG46   -0.2085144   -0.2366460   -0.1606570    0.8773368  0.007712047
## PG52   -0.2878646   -0.2773552   -0.1754821    1.5576371 -0.184717729
##      X220377_at X214432_at X216436_at X219270_at X217458_at X200803_s_at
## PG13 -0.3542819 -0.3380781 -0.3544934 -0.3179893 -0.3454370    2.1041753
## PG15 -0.2820014 -0.2801457 -0.2827358 -0.2581062 -0.2500352    0.7146728
## PG37 -0.2613286 -0.2326966 -0.2534781 -0.2519880 -0.2553360    0.7982353
## PG41 -0.3261372 -0.3079284 -0.3258790 -0.2774129 -0.3214563    2.1846121
## PG46 -0.2467600 -0.2206223 -0.2456562 -0.2293634 -0.2242019    0.6301139
## PG52 -0.2876337 -0.2678045 -0.2872160 -0.2663631 -0.2653919    0.9540652
##      X208448_x_at X211986_at X217137_x_at X208522_s_at X210392_x_at X204664_at
## PG13   -0.2984936  1.4901553   -0.3483076   -0.3470509   -0.3531339 -0.3546174
## PG15   -0.2494067  2.4141417   -0.2780189   -0.2487497   -0.2758400 -0.2825286
## PG37   -0.2214188  0.9408286   -0.2070951   -0.2590115   -0.2604705 -0.2599290
## PG41   -0.2849955  1.9328427   -0.3181669   -0.3064016   -0.3156127 -0.3272910
## PG46   -0.2150893  0.7044364   -0.2409485   -0.2395174   -0.2226009 -0.2495580
## PG52   -0.2309313  1.8811244   -0.2755650   -0.2707974   -0.2769067 -0.2874880
##      X207245_at X216632_at X201290_at X221580_s_at  X203545_at X216646_at
## PG13 -0.3538624 -0.3537967  1.4153054  0.169174170 0.220390553 -0.3520648
## PG15 -0.2832395 -0.2809816  0.9373597  0.003840957 0.082630308 -0.2755177
## PG37 -0.2639932 -0.2285999  1.2244660  0.064327204 0.131452997 -0.2605139
## PG41 -0.3255588 -0.3265267  1.3582014  0.095801847 0.032991272 -0.3272762
## PG46 -0.2511288 -0.2487971  0.6820914 -0.024407692 0.005034262 -0.2484976
## PG52 -0.2888737 -0.2865627  0.8994881  0.039452184 0.111392195 -0.2859666
##      X208260_at X217211_at X220670_at X215126_at X207951_at   X213540_at
## PG13 -0.3338332 -0.2264103 -0.3381920 -0.2995734 -0.2584107 -0.080408111
## PG15 -0.2576411 -0.2313649 -0.2665183 -0.1854446 -0.1298503 -0.052559171
## PG37 -0.2407221 -0.1933850 -0.2387125 -0.1880690 -0.1512980 -0.050178409
## PG41 -0.3119360 -0.2886289 -0.3192979 -0.2355068 -0.1819051 -0.003779906
## PG46 -0.2168057 -0.1277957 -0.2395234 -0.1071174 -0.1765203 -0.029454378
## PG52 -0.2694670 -0.2591938 -0.2680862 -0.1990479 -0.2448457 -0.055487097
##      X203225_s_at X204390_at X217178_at X216358_at X214737_x_at X210659_at
## PG13   0.27499532 -0.3561609 -0.3343981 -0.3218351    1.8238286 -0.2848229
## PG15  -0.08047284 -0.2842249 -0.2737841 -0.2809027    0.8915641 -0.2122094
## PG37  -0.13480202 -0.2628310 -0.2587773 -0.2445890    0.7737953 -0.2018855
## PG41  -0.19460094 -0.3272181 -0.3070093 -0.3046328    1.1452924 -0.2999847
## PG46  -0.14408367 -0.2500116 -0.2465760 -0.2010307    1.0009710 -0.1340731
## PG52  -0.10185399 -0.2878914 -0.2783607 -0.2774035    1.2449251 -0.2309394
##      X218664_at X215303_at X205152_at X215939_at X213447_at X217758_s_at
```

```
## PG13 -0.1945122 -0.3261742 -0.3280151 -0.3406907 0.23604364    2.1554486
## PG15 -0.1642520 -0.2201957 -0.2611546 -0.2796218 0.01411552    0.7367146
## PG37 -0.1635955 -0.2326977 -0.2497704 -0.2447381 0.10474896    0.4791715
## PG41 -0.2111710 -0.2770043 -0.2926469 -0.3175387 0.08987781    0.9610685
## PG46 -0.2044346 -0.2338300 -0.2286381 -0.2484743 0.03898581    0.3601102
## PG52 -0.1646716 -0.2723364 -0.2837091 -0.2863655 0.16347971    1.0100330
##      X212760_at X210038_at X208117_s_at X215107_s_at X219735_s_at X213166_x_at
## PG13 0.19073497 -0.2785135  -0.16415504   -0.3562549  -0.13399196    0.6012778
## PG15 0.07923707 -0.1931963  -0.14908030   -0.2853322  -0.17632234    0.4324977
## PG37 0.01793932 -0.2226951  -0.14709310   -0.2638025  -0.10790740    0.2845685
## PG41 0.02883586 -0.2649518  -0.09394586   -0.3279854  -0.22137782    0.7291362
## PG46 0.05842193 -0.1368973  -0.17114614   -0.2520008  -0.09717793    0.4295648
## PG52 0.05924725 -0.2237803  -0.13017718   -0.2903061  -0.14945945    0.7910072
##      X207373_at X213813_x_at X205507_at X216760_at X200786_at X212545_s_at
## PG13 -0.3556416   -0.2372055 -0.2754383 -0.3499442  0.7732261   -0.3258405
## PG15 -0.2810934   -0.2361630 -0.1305594 -0.2655171  0.3076642   -0.2688463
## PG37 -0.2630463   -0.2273260 -0.1852097 -0.2560022  0.6091778   -0.2446702
## PG41 -0.3273478   -0.2470304 -0.2363379 -0.3213394  0.6036982   -0.3203088
## PG46 -0.2491710   -0.1816538 -0.1694850 -0.2427275  0.2071208   -0.2473313
## PG52 -0.2883694   -0.2276375 -0.1948220 -0.2782878  0.4940623   -0.2843976
##      X210197_at X208787_at X215225_s_at X216772_at X219577_s_at X215060_at
## PG13 -0.3095445  1.9908560   -0.3526997 -0.3352123   -0.2688281 -0.3091986
## PG15 -0.2554305  0.4262426   -0.2829585 -0.2531343   -0.2045169 -0.2325305
## PG37 -0.2035941  0.5977418   -0.2564554 -0.2323355   -0.2301167 -0.2196953
## PG41 -0.2986889  0.7968648   -0.3247025 -0.3248614   -0.2030850 -0.2855976
## PG46 -0.2120067  0.2088658   -0.2173765 -0.2230390   -0.1108058 -0.1999861
## PG52 -0.2075818  0.6891316   -0.2804325 -0.2788333   -0.2035700 -0.2180050
##      X213780_at X208369_s_at X218824_at X215346_at X210035_s_at X220561_at
## PG13 -0.3278115  -0.21643748 -0.2770462 -0.3099375   -0.3446430 -0.3516841
## PG15 -0.2508278  -0.19783164 -0.1211693 -0.2111148   -0.2717081 -0.2816999
## PG37 -0.2570558  -0.06967904 -0.2047598 -0.2268156   -0.2454638 -0.2568125
## PG41 -0.3226145  -0.22519054 -0.2713759 -0.2588343   -0.3291818 -0.3209579
## PG46 -0.2302657  -0.15349892 -0.1843077 -0.1292839   -0.2517478 -0.2483006
## PG52 -0.2842552  -0.13398355 -0.2355643 -0.2486655   -0.2876484 -0.2823966
##      X203770_s_at X207118_s_at X217332_at X206455_s_at X206881_s_at X215713_at
## PG13   -0.3491928   -0.3485666 -0.3182996   -0.3121988   -0.3489727 -0.3452429
## PG15   -0.2824451   -0.2189459 -0.2709774   -0.2376437   -0.2782110 -0.2847219
## PG37   -0.2626327   -0.2602149 -0.2304180   -0.2200645   -0.2541925 -0.2580911
## PG41   -0.3262374   -0.3248993 -0.3239338   -0.2489317   -0.3198320 -0.3262729
## PG46   -0.2502283   -0.2489090 -0.2212021   -0.2349339   -0.2426816 -0.2511922
## PG52   -0.2879277   -0.2876332 -0.2612620   -0.1977064   -0.2843642 -0.2889202
##      X212705_x_at X220767_at X221118_at X207736_s_at  X209529_at X221240_s_at
## PG13   -0.2169888 -0.3086577 -0.3554593   -0.3561291  0.07243748   -0.2809033
## PG15   -0.1703073 -0.2787071 -0.2817230   -0.2834901 -0.16473833   -0.2360534
## PG37   -0.1172640 -0.2546557 -0.2561849   -0.2609254 -0.16391695   -0.2060997
## PG41   -0.2378563 -0.2861049 -0.3220708   -0.3279499  0.14553749   -0.2620081
## PG46   -0.2042483 -0.2410928 -0.2506865   -0.2501708 -0.19334231   -0.1779420
## PG52   -0.1993066 -0.2723771 -0.2835265   -0.2892378  0.02224850   -0.2233329
##       X208838_at X208650_s_at X203096_s_at X206811_at X210940_s_at X216283_s_at
## PG13 -0.03562514  1.647144969  -0.20088340 -0.3522139   -0.3551079   -0.3223856
## PG15 -0.12244660 -0.199020867  -0.07557459 -0.2640261   -0.2840397   -0.2658797
## PG37 -0.12968334 -0.116381050  -0.14732692 -0.2125809   -0.2632861   -0.2332098
## PG41 -0.16203564  0.184005831  -0.14822203 -0.2932576   -0.3229889   -0.2878017
## PG46 -0.12676039  0.000218098  -0.16861731 -0.2252418   -0.2383510   -0.2271611
```

```
## PG52 -0.02215312  0.112225023  -0.21135328 -0.2424388   -0.2889531   -0.2863065
##      X217951_s_at X222100_at X207801_s_at X207846_at X210282_at X211207_s_at
## PG13   -0.3033349 -0.3505179   0.18866103 -0.3574931 -0.3139910   -0.3344417
## PG15   -0.2334851 -0.2786449   0.05540238 -0.2849154 -0.2501681   -0.2663107
## PG37   -0.2269370 -0.2618159   0.11575602 -0.2646271 -0.2288820   -0.2433846
## PG41   -0.3018138 -0.3188744   0.29481624 -0.3238765 -0.2994483   -0.2967934
## PG46   -0.2410937 -0.2488604  -0.02089041 -0.2493902 -0.2379900   -0.1971202
## PG52   -0.2732923 -0.2857538   0.13435052 -0.2901304 -0.2521643   -0.2586774
##      X206907_at X214676_x_at X210196_s_at X206815_at X215309_at  X203666_at
## PG13 -0.3206388   -0.1745441   -0.3573866 -0.3529445 -0.3570145 -0.24556521
## PG15 -0.1916295   -0.1672732   -0.2804057 -0.2758529 -0.2857702 -0.04918592
## PG37 -0.2083559   -0.1699205   -0.2615514 -0.2614363 -0.2643440 -0.09781014
## PG41 -0.2848381   -0.1883532   -0.3150583 -0.3263766 -0.3294890 -0.23926134
## PG46 -0.1650429   -0.1720229   -0.2491033 -0.2489812 -0.2519210 -0.06856966
## PG52 -0.2616176   -0.1488362   -0.2852473 -0.2870992 -0.2872101 -0.15906606
##      X215162_at X206406_at X204051_s_at X220882_at X208706_s_at X208225_at
## PG13 -0.2743656 -0.3504096    0.5490338 -0.3512514   0.36940410 -0.3526199
## PG15 -0.2122894 -0.2752066   -0.1719867 -0.2759928   0.16108342 -0.2819587
## PG37 -0.1539024 -0.2579486    0.1914794 -0.2591914   0.04870672 -0.2625770
## PG41 -0.2490609 -0.3235654    0.4283692 -0.3239996   0.15497927 -0.3249610
## PG46 -0.1714058 -0.2457514    0.2878889 -0.2470405   0.11141875 -0.2503404
## PG52 -0.2135082 -0.2814511    0.5068921 -0.2852363   0.20151020 -0.2885228
##      X216740_at X202366_at X214748_at X219839_x_at X218285_s_at X206598_at
## PG13 -0.3565270 -0.2490506 -0.1858073   -0.3539761  -0.06494081 -0.3553171
## PG15 -0.2832198 -0.1780471 -0.1903594   -0.2795449   0.04998412 -0.2824024
## PG37 -0.2632055 -0.1911517 -0.1684956   -0.2613001  -0.08194657 -0.2619878
## PG41 -0.3236014 -0.2238381 -0.1480672   -0.3265852  -0.15330759 -0.3249558
## PG46 -0.2183673 -0.1426109 -0.1249790   -0.2486050  -0.05795560 -0.2503656
## PG52 -0.2904711 -0.1805612 -0.1612783   -0.2841832  -0.07388242 -0.2875555
##      X217535_at X206423_at X200755_s_at X209327_s_at X206806_at AFFX.TrpnX.M_at
## PG13 -0.3322315 -0.3503940   0.08294589   -0.3542379 -0.3423452      -0.3563181
## PG15 -0.2776332 -0.2691389  -0.10537004   -0.2784196 -0.2618137      -0.2833549
## PG37 -0.2509607 -0.2550792  -0.01135052   -0.2584038 -0.2544506      -0.2608817
## PG41 -0.3200789 -0.3248503   0.06922442   -0.3216268 -0.3122069      -0.3257340
## PG46 -0.2355199 -0.2275537  -0.12340514   -0.2495809 -0.2266305      -0.2510829
## PG52 -0.2615324 -0.2839658  -0.04292003   -0.2883899 -0.2600330      -0.2872743
##      X210618_at X210107_at X205245_at X215296_at X208880_s_at X217128_s_at
## PG13 -0.3417171 -0.3107299 -0.2699169 -0.3005880 -0.138740951   -0.2835429
## PG15 -0.2697799 -0.2180872 -0.2055610 -0.2391534 -0.051305441   -0.2020245
## PG37 -0.2377274 -0.1881108 -0.1722960 -0.2098525 -0.047923811   -0.2346032
## PG41 -0.3019602 -0.2895980 -0.2086574 -0.2824584 -0.001195349   -0.2645892
## PG46 -0.2256155 -0.1666057 -0.1663735 -0.2107225 -0.116421110   -0.1692825
## PG52 -0.2466504 -0.2171498 -0.1854336 -0.2183378  0.018154650   -0.2409990
##      X212249_at X216363_at X214309_s_at X221223_x_at X214970_s_at X204110_at
## PG13 -0.2823038 -0.3265507   -0.3405827  -0.20097815   -0.3403919 -0.3067341
## PG15 -0.1985278 -0.2679275   -0.2711269  -0.10649845   -0.2745017 -0.2615930
## PG37 -0.1687962 -0.2209722   -0.2527849  -0.15315206   -0.2613990 -0.2523674
## PG41 -0.2520196 -0.3009558   -0.3212990  -0.13697561   -0.3118257 -0.2864059
## PG46 -0.1771416 -0.2076506   -0.2443684  -0.08768963   -0.2426404 -0.2150483
## PG52 -0.2273535 -0.2574995   -0.2834884  -0.16666993   -0.2884272 -0.2607401
##      X220705_s_at  X202813_at X212173_at X220627_at X215080_s_at X207493_x_at
## PG13   -0.2871695 -0.01866643 -0.3336511 -0.2731192   -0.3543141   -0.3543674
## PG15   -0.1774754 -0.11161414 -0.2610407 -0.2410532   -0.2827179   -0.2815568
## PG37   -0.1898792  0.14452164 -0.2335800 -0.2173967   -0.2363829   -0.2634866
```

```
## PG41   -0.2274527  0.11920945 -0.2945105 -0.2747521   -0.3109416   -0.3192116
## PG46   -0.1502243 -0.13125267 -0.2314178 -0.1800934   -0.2355182   -0.2431777
## PG52   -0.1972852  0.05492810 -0.2755850 -0.2562404   -0.2845203   -0.2839205
##      X219834_at   X218269_at X206820_at X214769_at X216771_at X218199_s_at
## PG13 -0.3019238 -0.004668724 -0.3516076 -0.3284610 -0.3554296   -0.1066698
## PG15 -0.2409736 -0.112093878 -0.2815824 -0.2830551 -0.2792810   -0.1649806
## PG37 -0.2295367 -0.076683788 -0.2603418 -0.2589044 -0.2466818   -0.0824832
## PG41 -0.2996941  0.047915936 -0.3219667 -0.3066457 -0.2982991   -0.1537098
## PG46 -0.2245953 -0.098704200 -0.2456295 -0.2487043 -0.2286703   -0.1022429
## PG52 -0.2530858  0.003344785 -0.2858033 -0.2839165 -0.2856053   -0.1237146
##      X222219_s_at X217004_s_at X217257_at X215778_x_at X206941_x_at X217464_at
## PG13   -0.3249089   -0.3293150 -0.3447863   -0.2218420   -0.3383407 -0.3555514
## PG15   -0.2560133   -0.2730026 -0.2847682   -0.2327626   -0.1981101 -0.2593665
## PG37   -0.2381135   -0.2377020 -0.2559419   -0.1537968   -0.2626661 -0.2457525
## PG41   -0.3020074   -0.3197948 -0.3270621   -0.1734175   -0.3080127 -0.3191252
## PG46   -0.2083982   -0.2275559 -0.2409269   -0.1280968   -0.2500059 -0.2500795
## PG52   -0.2753269   -0.2842271 -0.2878348   -0.1738872   -0.2885516 -0.2860270
##      X221546_at X208603_s_at X206859_s_at X215028_at  X219205_at X203599_s_at
## PG13 -0.3116580   -0.3460156   -0.3499622 -0.3064157 -0.18924507   -0.1809095
## PG15 -0.2752354   -0.2797047   -0.2767568 -0.2666041 -0.12562424   -0.1246956
## PG37 -0.2016762   -0.2557035   -0.2625944 -0.2413016 -0.11257985   -0.1780395
## PG41 -0.2857017   -0.3222628   -0.3265476 -0.2137592 -0.20922293   -0.1908231
## PG46 -0.2376673   -0.2443730   -0.2500028 -0.2208147 -0.11404422   -0.1054273
## PG52 -0.2752417   -0.2843721   -0.2849237 -0.2650409 -0.09566435   -0.1241673
##      X200059_s_at X221342_at X208084_at   X205000_at X214744_s_at X215880_at
## PG13     4.006275 -0.3389723 -0.3220361  0.070494265   -0.3542189 -0.3348892
## PG15     2.215048 -0.2640569 -0.2783539 -0.007626538   -0.2525955 -0.2785920
## PG37     2.081176 -0.2213476 -0.2583656 -0.052075276   -0.2263146 -0.2621291
## PG41     3.253453 -0.3205367 -0.3216135  0.140745639   -0.3008217 -0.3081431
## PG46     2.426416 -0.2231877 -0.2036180  0.105112779   -0.2391941 -0.2456021
## PG52     1.880288 -0.2611889 -0.2531546  0.049205099   -0.2716423 -0.2614968
##      X214065_s_at X220094_s_at X201104_x_at X219835_at X217192_s_at
## PG13   -0.3241601  -0.08059849    1.5433420 -0.3547697   -0.3278417
## PG15   -0.2696562  -0.11339213    0.9139746 -0.2809842   -0.2223613
## PG37   -0.2503474  -0.09591546    0.6335901 -0.2597674   -0.1971640
## PG41   -0.3129129  -0.03550418    1.3528936 -0.3246718   -0.2618981
## PG46   -0.2497112  -0.17231006    0.7530796 -0.2495263   -0.1122252
## PG52   -0.2758678  -0.12292457    1.2496697 -0.2877324   -0.2324491
##      X207447_s_at X217115_at X211618_s_at X215479_at X217904_s_at X215361_at
## PG13   -0.3432001 -0.3515206   -0.3535208 -0.2614024   -0.1502670 -0.3546118
## PG15   -0.2650671 -0.2667089   -0.2844113 -0.1731214   -0.2150122 -0.2808074
## PG37   -0.2585448 -0.2617283   -0.2633679 -0.1986532   -0.1757076 -0.2596814
## PG41   -0.3226984 -0.3261117   -0.3265536 -0.2209630   -0.2254006 -0.3226583
## PG46   -0.2158072 -0.2492319   -0.2502405 -0.1290159   -0.1642916 -0.2511120
## PG52   -0.2865591 -0.2877900   -0.2873044 -0.1806112   -0.1126091 -0.2889938
##      X206465_at X214503_x_at X205385_at X220703_at X219113_x_at X212640_at
## PG13 -0.3306340   -0.3526675 -0.3496162 -0.2779856   -0.3215054  1.3012757
## PG15 -0.2751090   -0.2806422 -0.2675927 -0.2523917   -0.2487308  0.2536109
## PG37 -0.2258492   -0.2601684 -0.2436508 -0.2210113   -0.1602855  0.8679013
## PG41 -0.3026936   -0.3263111 -0.3100751 -0.2603780   -0.1567456  0.1155969
## PG46 -0.2178680   -0.2504049 -0.2140810 -0.2041317   -0.1512688  0.2936065
## PG52 -0.2722489   -0.2818682 -0.2553410 -0.2369679   -0.1677487  0.6076278
##       X205083_at X214001_x_at X215423_at X206581_at X215753_at X217302_at
## PG13 -0.21363571   -0.1175050 -0.3447271 -0.3214744 -0.3442887 -0.3435915
```

```
## PG15 -0.09634837   -0.1238244 -0.2710681 -0.2684382 -0.2766273 -0.2803244
## PG37 -0.18448344   -0.1274948 -0.2526280 -0.2249123 -0.2583413 -0.2621525
## PG41 -0.24394580   -0.2077042 -0.3208882 -0.3024486 -0.3176514 -0.3206616
## PG46 -0.13721320   -0.1267747 -0.2025051 -0.2383227 -0.2427003 -0.2460225
## PG52 -0.21588656   -0.1565824 -0.2765581 -0.2810173 -0.2740338 -0.2605035
##      X219842_at X217925_s_at  X212250_at X212938_at X208294_x_at X201713_s_at
## PG13 -0.2825489   -0.1927286  0.09551878 -0.3087781   -0.3530079   0.25321735
## PG15 -0.2354443   -0.1104971 -0.07439333 -0.2137103   -0.2793189   0.02386146
## PG37 -0.2155043   -0.1070401 -0.07748732 -0.2166676   -0.2605560   0.01316489
## PG41 -0.2694657   -0.1572534 -0.09662879 -0.2597773   -0.3232518   0.16298097
## PG46 -0.2441334   -0.1456038  0.03887081 -0.2344026   -0.2480726   0.04566544
## PG52 -0.2410254   -0.1371882 -0.01559368 -0.2297809   -0.2847785   0.06552300
##      X204339_s_at X208253_at X201480_s_at X208989_s_at X214809_at X219930_at
## PG13   -0.3158140 -0.3277380  0.016482332   -0.2111509 -0.3354691 -0.3492880
## PG15   -0.2218346 -0.2782080  0.037206174   -0.1183806 -0.2786616 -0.2833372
## PG37   -0.2531938 -0.2380980  0.037881733   -0.1678374 -0.2259822 -0.2022442
## PG41   -0.2982710 -0.3013553  0.004063395   -0.2197293 -0.2848418 -0.3240538
## PG46   -0.2358029 -0.1762754  0.020632708   -0.1635732 -0.2189184 -0.2457761
## PG52   -0.2492235 -0.2850784  0.088783588   -0.1695316 -0.2812710 -0.2850559
##      X218140_x_at X211172_x_at X213013_at X201734_at X210102_at X221784_at
## PG13    1.4124118   -0.2981997 -0.2193286  1.1018798 -0.2560981 -0.3460602
## PG15    0.3554463   -0.2082995 -0.1781935  0.2863186 -0.1557305 -0.2734188
## PG37    0.6153734   -0.1786243 -0.1783323  0.7236994 -0.1949911 -0.2454163
## PG41    0.4229577   -0.2792965 -0.1193387  1.0061175 -0.2420748 -0.3178733
## PG46    0.2516321   -0.1863179 -0.1118728  0.3918230 -0.1482144 -0.2416312
## PG52    0.6524908   -0.2137649 -0.1467820  1.0339689 -0.2127307 -0.2796034
##      X220673_s_at X213695_at X214390_s_at X205387_s_at  X220465_at X201531_at
## PG13   -0.3561498 -0.3199788   -0.3485622   -0.3509817 -0.10143394  1.8293814
## PG15   -0.2812289 -0.2193253   -0.2749548   -0.2777989 -0.13155223  3.8113085
## PG37   -0.2629271 -0.2549160   -0.2556134   -0.2429445 -0.07115260  0.9318047
## PG41   -0.3284944 -0.3070089   -0.3200673   -0.3270386 -0.17804489  2.8472523
## PG46   -0.2514106 -0.2311541   -0.2405767   -0.2480992 -0.08139331  0.7349019
## PG52   -0.2874396 -0.2415414   -0.2804360   -0.2844894 -0.08937491  1.8721575
##      X215673_at X217522_at X213054_at X213466_at X216869_at X212812_at
## PG13 -0.3571809 -0.3375143 -0.3429953 -0.3515480 -0.2874733 3.48653825
## PG15 -0.2854881 -0.2841149 -0.2825360 -0.2811217 -0.2584200 1.26917887
## PG37 -0.2625827 -0.2518213 -0.2372932 -0.2592849 -0.2316445 1.34335384
## PG41 -0.3202053 -0.3043759 -0.2996193 -0.3249998 -0.3027217 3.76754594
## PG46 -0.2521754 -0.2296229 -0.2488691 -0.2480064 -0.2370215 0.05720001
## PG52 -0.2877732 -0.2711545 -0.2687123 -0.2855010 -0.2427783 1.58580650
##      X214639_s_at X217372_at X210122_at X216324_at   X220420_at X211313_s_at
## PG13   -0.3553138 -0.3549059 -0.3535183 -0.3547083 -0.007194473   -0.3455153
## PG15   -0.2791614 -0.2816012 -0.2819516 -0.2827002 -0.231596636   -0.2806988
## PG37   -0.2626775 -0.2286591 -0.2620233 -0.2622041 -0.209461539   -0.2574211
## PG41   -0.3262599 -0.3176058 -0.3231807 -0.3258630 -0.139196154   -0.3253773
## PG46   -0.2506056 -0.2481619 -0.2501774 -0.2502783 -0.238872382   -0.2188706
## PG52   -0.2867936 -0.2861072 -0.2880119 -0.2873021 -0.194274387   -0.2351500
##      X218575_at X205137_x_at X203036_s_at X208140_s_at X211926_s_at X205320_at
## PG13 -0.1301682   -0.3425205   -0.3521499   -0.2822669   0.26756649 -0.3296608
## PG15 -0.1811024   -0.2800705   -0.2778724   -0.2164190   0.16505058 -0.2702471
## PG37 -0.1820026   -0.2493444   -0.2612480   -0.2078572   0.02747855 -0.2015601
## PG41 -0.1529975   -0.3172597   -0.3226489   -0.3077099   0.34728053 -0.3205823
## PG46 -0.1303560   -0.2333625   -0.2388634   -0.2053473   0.04463743 -0.2180643
## PG52 -0.1496528   -0.2801089   -0.2869198   -0.2047218   0.05278543 -0.2752106
```

```
##      X221469_at X217060_at X220247_at  X203794_at X220249_at X208551_at
## PG13 -0.3495283 -0.3112365 -0.3523874 -0.08251750 -0.3500255 -0.3133575
## PG15 -0.2383077 -0.2535188 -0.2823284  0.02861305 -0.2787734 -0.2200214
## PG37 -0.2132843 -0.2265764 -0.2631561 -0.13327194 -0.2580473 -0.1841165
## PG41 -0.3039324 -0.2953744 -0.3259652 -0.04466422 -0.3246600 -0.2749428
## PG46 -0.2354920 -0.1912552 -0.2440536 -0.14261503 -0.2365675 -0.2035389
## PG52 -0.2835343 -0.2447495 -0.2754793 -0.03484775 -0.2834074 -0.2456107
##      X211788_s_at   X218379_at X210599_at X219524_s_at X202652_at X215486_at
## PG13   -0.3556365  0.263601947 -0.3264633   -0.3503525 -0.2801078 -0.3241247
## PG15   -0.2830879 -0.086368126 -0.2815842   -0.2805000 -0.1886584 -0.2573579
## PG37   -0.2629614 -0.143406228 -0.2325603   -0.2516777 -0.2495678 -0.2395160
## PG41   -0.3276935  0.028756984 -0.3103311   -0.3175157 -0.2251279 -0.3134517
## PG46   -0.2501262 -0.054423264 -0.2457907   -0.2478196 -0.1532688 -0.2044189
## PG52   -0.2886131 -0.004294509 -0.2629965   -0.2806833 -0.1469433 -0.2474236
##      X209207_s_at X217262_s_at X211756_at X218246_at  X202346_at X209717_at
## PG13  0.019120031   -0.3559509 -0.3431732 -0.2162278 -0.01017263 -0.2514204
## PG15  0.009837170   -0.2806520 -0.2810755 -0.1555485 -0.15717051 -0.2157599
## PG37 -0.102192464   -0.2629980 -0.2344726 -0.1172350 -0.12613165 -0.1885464
## PG41  0.007618203   -0.3269820 -0.3236117 -0.1619419 -0.15198220 -0.2863057
## PG46 -0.002683791   -0.2480171 -0.2495191 -0.1735834 -0.15031992 -0.2169784
## PG52 -0.034455853   -0.2893265 -0.2504119 -0.1382695 -0.09792595 -0.2520077
##      X206077_at X213692_s_at X214899_at X207068_at
## PG13 -0.3393309   -0.3161388 -0.3504453 -0.3324365
## PG15 -0.2732266   -0.2725610 -0.2781882 -0.2574738
## PG37 -0.2274245   -0.2453698 -0.2636068 -0.2210516
## PG41 -0.3102157   -0.3137273 -0.3175094 -0.3152171
## PG46 -0.2256879   -0.2371159 -0.2497966 -0.2309499
## PG52 -0.2608673   -0.2798694 -0.2904359 -0.2652237
```

The dimension of the data set is given by :

```
dim(prostate)
```

```
## [1]  79 501
```

```
help(prostate)
```

```
## No documentation for 'prostate' in specified packages and libraries:
## you could try '??prostate'
```

- The dataset contains 79 observations, where each of them represent probably people which are tested to have prostate or not.
- The dataset has 501 variables : the response variable is called Y and its categorical with two possibles factors “0” and “1”. “0” probably represent a person which is don’t have the prostate and “1” represent someone who has prostate. This mean that we’ll deal with a classification, a binary classification task.
- **The dataset is highly high dimensional due to the fact that the number of variables far exceeds the number of observations**

2. Let’s comment from the statistical perspective on the type of data in the input space

```
summary(prostate) [,1:25]
```

```
##        Y             X206212_at         X207075_at        X215872_at
##  Min.   :0.0000   Min.   :-0.27572   Min.   :-0.3700   Min.   :-0.3745
##  1st Qu.:0.0000   1st Qu.:-0.22092   1st Qu.:-0.3189   1st Qu.:-0.3273
##  Median :1.0000   Median :-0.19108   Median :-0.2912   Median :-0.3019
```

```
##  Mean   :0.5316   Mean   :-0.18635   Mean   :-0.2945   Mean   :-0.3027
##  3rd Qu.:1.0000   3rd Qu.:-0.16036   3rd Qu.:-0.2733   3rd Qu.:-0.2783
##  Max.   :1.0000   Max.   :-0.07154   Max.   :-0.2076   Max.   :-0.1972
##    X201876_at        X211935_at       X206788_s_at        X216441_at
##  Min.   :-0.02151   Min.   :0.04825   Min.   :-0.26716   Min.   :-0.3771
##  1st Qu.: 0.12530   1st Qu.:0.22764   1st Qu.:-0.20338   1st Qu.:-0.3304
##  Median : 0.21209   Median :0.35430   Median :-0.18122   Median :-0.3080
##  Mean   : 0.23122   Mean   :0.37536   Mean   :-0.17665   Mean   :-0.3074
##  3rd Qu.: 0.26989   3rd Qu.:0.44529   3rd Qu.:-0.15582   3rd Qu.:-0.2827
##  Max.   : 1.03433   Max.   :1.42433   Max.   :-0.03853   Max.   :-0.2293
##   X209290_s_at       X219877_at       X220675_s_at       X204229_at
##  Min.   :0.00699   Min.   :-0.3699   Min.   :-0.3710   Min.   :-0.3545
##  1st Qu.:0.27076   1st Qu.:-0.3241   1st Qu.:-0.3241   1st Qu.:-0.3128
##  Median :0.40936   Median :-0.3029   Median :-0.3033   Median :-0.2844
##  Mean   :0.46183   Mean   :-0.3006   Mean   :-0.3031   Mean   :-0.2817
##  3rd Qu.:0.64654   3rd Qu.:-0.2760   3rd Qu.:-0.2803   3rd Qu.:-0.2595
##  Max.   :1.30156   Max.   :-0.2214   Max.   :-0.2269   Max.   :-0.1879
##    X216460_at        X215861_at        X207287_at       X211875_x_at
##  Min.   :-0.3731   Min.   :-0.3420   Min.   :-0.3109   Min.   :-0.3735
##  1st Qu.:-0.3241   1st Qu.:-0.2817   1st Qu.:-0.2717   1st Qu.:-0.3232
##  Median :-0.3004   Median :-0.2609   Median :-0.2468   Median :-0.3010
##  Mean   :-0.2994   Mean   :-0.2570   Mean   :-0.2449   Mean   :-0.3001
##  3rd Qu.:-0.2748   3rd Qu.:-0.2296   3rd Qu.:-0.2240   3rd Qu.:-0.2750
##  Max.   :-0.2201   Max.   :-0.1634   Max.   :-0.1624   Max.   :-0.2238
##    X205055_at        X216887_s_at      X213319_s_at       X220709_at
##  Min.   :-0.08091   Min.   :-0.3098   Min.   :-0.3705   Min.   :-0.3743
##  1st Qu.: 0.03499   1st Qu.:-0.2620   1st Qu.:-0.2998   1st Qu.:-0.3272
##  Median : 0.09122   Median :-0.2376   Median :-0.2759   Median :-0.3072
##  Mean   : 0.10307   Mean   :-0.2319   Mean   :-0.2744   Mean   :-0.3047
##  3rd Qu.: 0.14684   3rd Qu.:-0.2073   3rd Qu.:-0.2510   3rd Qu.:-0.2835
##  Max.   : 0.43603   Max.   :-0.1293   Max.   :-0.1640   Max.   :-0.2275
##    X204011_at         X216174_at        X219416_at       X200793_s_at
##  Min.   :-0.26095   Min.   :-0.3607   Min.   :-0.3180   Min.   :-0.109728
##  1st Qu.:-0.20582   1st Qu.:-0.2736   1st Qu.:-0.2579   1st Qu.: 0.007349
##  Median :-0.17972   Median :-0.2552   Median :-0.2237   Median : 0.088414
##  Mean   :-0.16953   Mean   :-0.2514   Mean   :-0.2201   Mean   : 0.136135
##  3rd Qu.:-0.14694   3rd Qu.:-0.2242   3rd Qu.:-0.1932   3rd Qu.: 0.227766
##  Max.   : 0.06377   Max.   :-0.1543   Max.   :-0.1081   Max.   : 0.665145
##    X216050_at
##  Min.   :-0.3738
##  1st Qu.:-0.3273
##  Median :-0.3040
##  Mean   :-0.3049
##  3rd Qu.:-0.2830
##  Max.   :-0.2248
```

- *Type of data in input space :* We have 500 variables which are all continuous.
- *Dimensionality challenges:* THe fact that we are dealing with a dataset which contain so many predictors can lead to some challenges such that : **collinearity** between some predictors, or between predictors and response variable and this can implies presence of redundant variables. Having more predictors than observations risks also **fitting noise rathen than true patterns**
- *Algorithm and computational costs :* Have 500 variables is a big number of parameters to estimate and this increases inevitably the computational cost. So, some dimensionnality reduction techniques might

be needed to reduce the number of variables.

3. Let's plot the distribution of the response variiable and comment

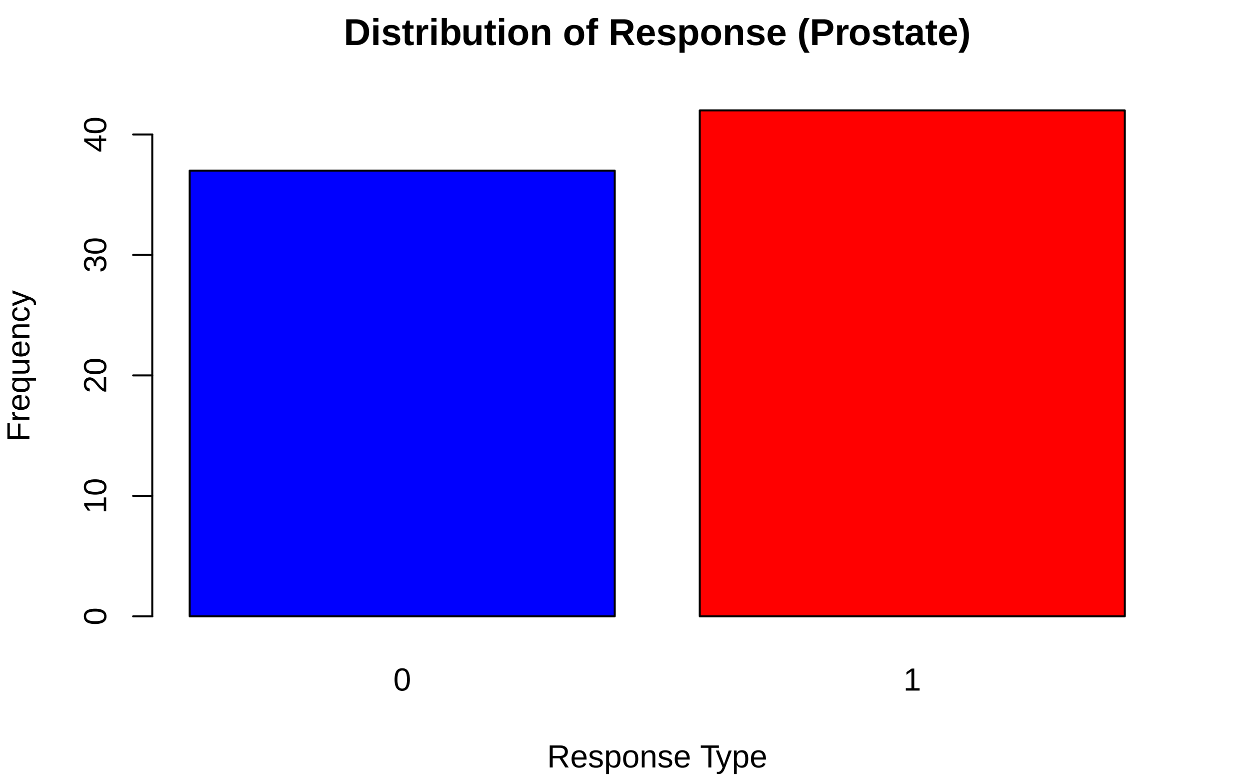
Let's first check the proportion of each factors in the response variable.

```
response_distribution <- table(prostate$Y) # Frequency table
response_distribution
```

```
##
##  0  1
## 37 42
```

Let's plot then the barplot of this response variable

```
# Plot the distribution of the response

barplot(response_distribution,
        main = "Distribution of Response (Prostate)",
        xlab = "Response Type",
        ylab = "Frequency",
        col = c("blue", "red"))
```



Comment : This response variable is called **Y**, it is a categorical variable with two factors : "0" and "1" which mean that we will deal with a binary classification problem.

Additionally, we have in our response variable, 37 people classified "0" and 42 classified "1". So, we are not in a case of imbalanced classes

4. Let's identify the 9 individually most powerful predictor variables with respect to the response variable.

Let's do the Krustal test and order the variables by using their p-value. The test statistic represents the degree of separation between two groups, so larger values indicate stronger differences. Thus, the variables which have the biggest statistic value are the most powerful. We'll also add the p-value in our table. In fact, the lowest p-value correspond to the most powerful predictors for our response variables.

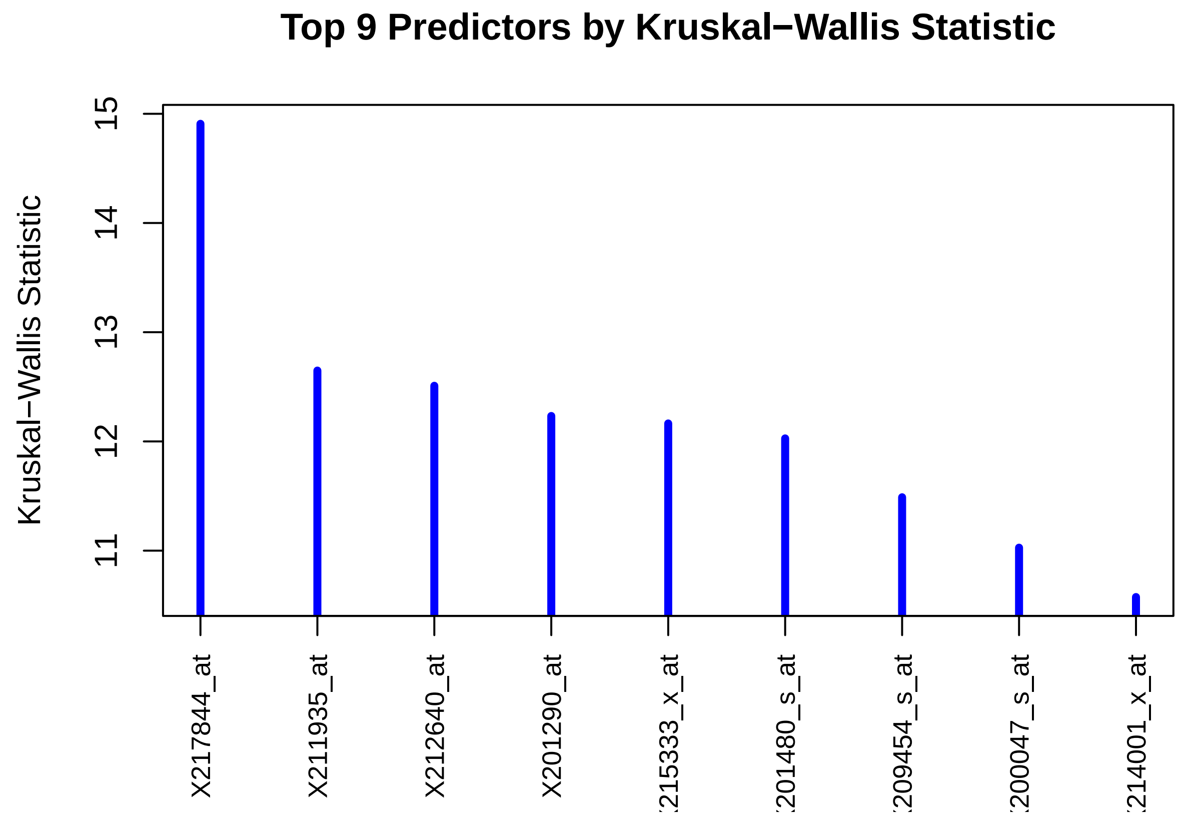
```r
# Initialize a dataframe to store p-values
kruskal_results <- data.frame(
  Predictor = colnames(prostate[, -1]),  # Exclude the first column (Y)
  P_Value = NA,
  Statistic = NA
)

# Perform Kruskal-Wallis test for each predictor
for (i in 2:ncol(prostate)) {  # Start from the second column since the first is Y
  kruskal_test <- kruskal.test(prostate[, i] ~ prostate[, 1])  # Use the first column as Y
  kruskal_results$Statistic[i - 1] <- kruskal_test$statistic  # Store the test statistic
  kruskal_results$P_Value[i - 1] <- kruskal_test$p.value      # Store the p-value
}

# Sort predictors by p-value (ascending order)
kruskal_results <- kruskal_results[order(kruskal_results$P_Value), ]

# Sort predictors by statistic (descending order)
kruskal_results <- kruskal_results[order(-kruskal_results$Statistic), ]

# Select the top 9 predictors
top_9_predictors <- head(kruskal_results, 9)

# Print the top 9 predictors
print(top_9_predictors)
```

```
##        Predictor      P_Value Statistic
## 125   X217844_at 0.0001128710  14.90820
## 5     X211935_at 0.0003757596  12.64903
## 430   X212640_at 0.0004048549  12.50965
## 278   X201290_at 0.0004694650  12.23320
## 202 X215333_x_at 0.0004870575  12.16458
## 445 X201480_s_at 0.0005241021  12.02790
## 40  X209454_s_at 0.0007001308  11.48890
## 220 X200047_s_at 0.0008977434  11.02741
## 432 X214001_x_at 0.0011460333  10.57539
```

5. Let's generate a type 'h' plot with the Krustal - Wallis test statistic

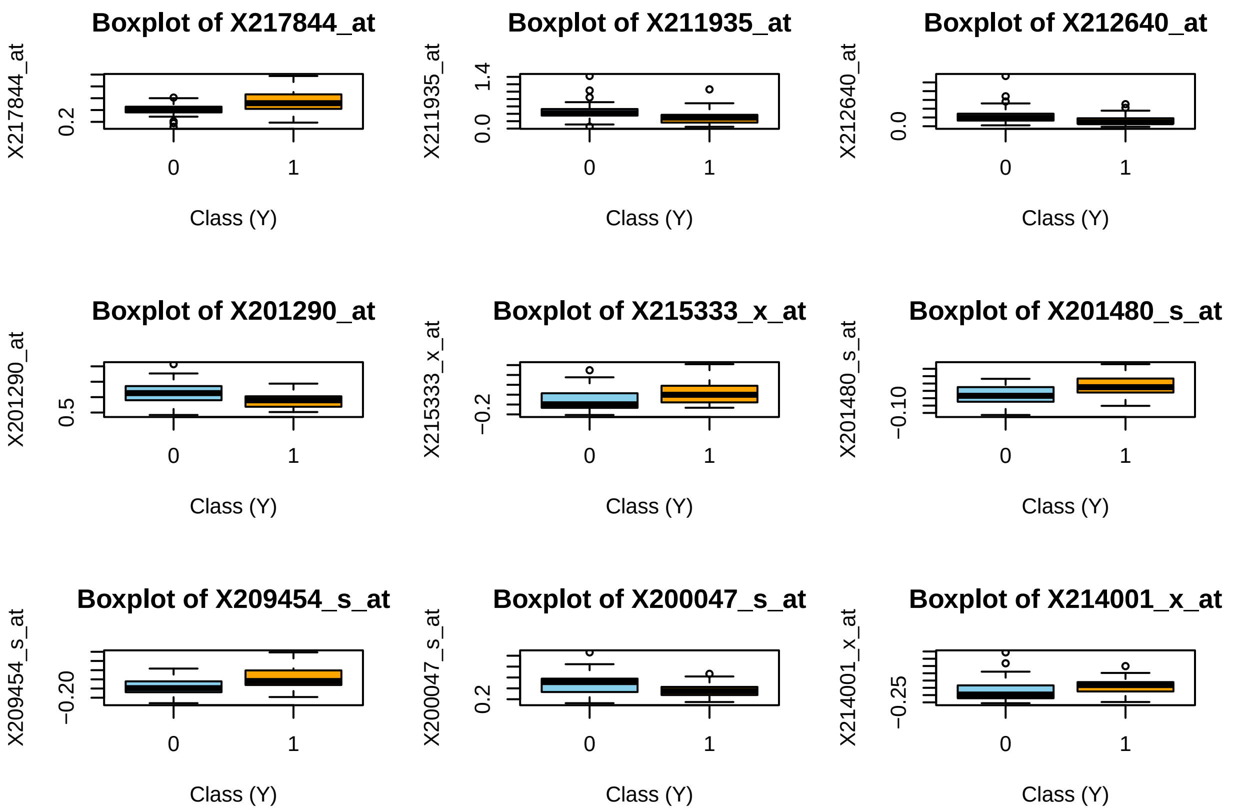
```r
plot(
  1:9,
  top_9_predictors$Statistic,
  type = "h",
  xaxt = "n",
  xlab = "",
  ylab = "Kruskal-Wallis Statistic",
  main = "Top 9 Predictors by Kruskal-Wallis Statistic",
```

```
  col = "blue",
  lwd = 4
)

# Add predictor names to the x-axis
axis(1, at = 1:9, labels = top_9_predictors$Predictor, las = 2, cex.axis = 0.8)
```



6. Let's generate the comparative boxplots of the 9 most powerful variable.

```
# Extract the top 9 predictor names
top_9 <- top_9_predictors$Predictor

# Generate boxplots for each of the top 9 predictors
par(mfrow = c(3, 3)) # Arrange plots in a 3x3 grid
for (predictor in top_9) {
  boxplot(
    prostate[[predictor]] ~ prostate$Y,
    main = paste("Boxplot of", predictor),
    xlab = "Class (Y)",
    ylab = predictor,
    col = c("skyblue", "orange"),
    border = "black"
  )
}
```

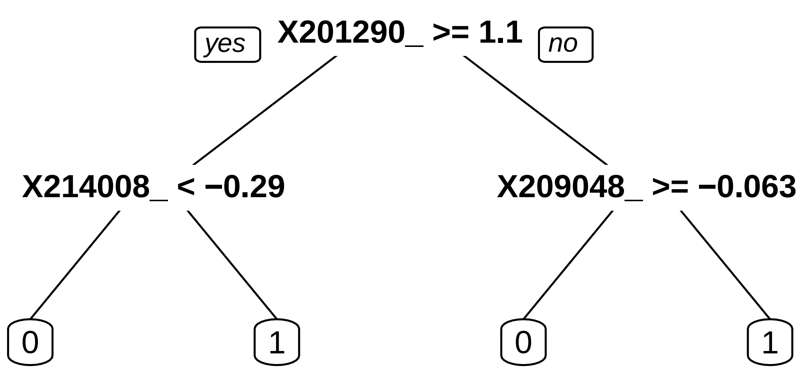


Comments : The boxplot present how each variable take alone can separate people who have the prostate and people who don't. Based on this plot, we said that : - All of the variable generally show a good separation between the two classes but some variables like *X201480_s_at*, *X201290_at* ,*X217844_at* have more good ability to differentiate the two classes. - The variable which have the least descriminative power is *X212640_at*. Some variables (like *X211935_at*) also have a number of outliers a little bit significant. This predictors may require further investigation to determine their reliability.

7. Let's build the classification tree with $cp = 0.01$

- **Plot the tree**

```
prostate$Y <- as.factor(prostate$Y)
tree.xy <- rpart(Y~., data=prostate, control = rpart.control(cp = 0.01))
prp(tree.xy)
```



- **Number of terminal nodes:** 4 - **Mathematical form of region 2 and region 4 :** The Region 2 is given by :

$$\mathcal{R}_{\infty} = \left\{ \mathbf{X} \in \mathbb{R}^{500} \text{ such that X201290 } \geq 1.1 \text{ and X214008} \geq -0.29 \right\}$$

And the Region 4 is given by :

$$\mathcal{R}_{\triangle} = \left\{ \mathbf{X} \in \mathbb{R}^{500} \text{ such that X201290 } < 1.1 \text{ and X209048} < -0.063 \right\}$$

- **Comment on the variable at the root :** At the root, we have the variable *X201290* which is among the top 9 powerful predictors according to what the Krustal Wallis test reveal to us.

8. Let's generate the boxplot of 9 weakest variables

This variables are :

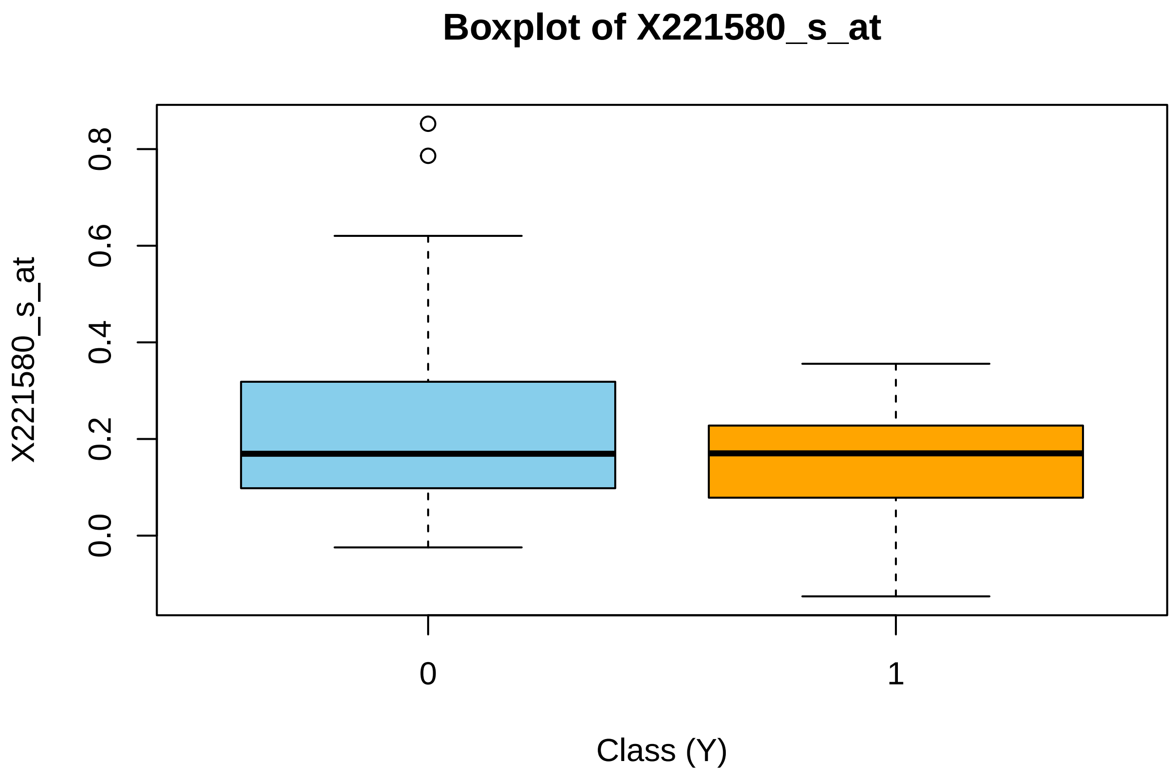
```
# Initialize a dataframe to store p-values
kruskal_results <- data.frame(
  Predictor = colnames(prostate[, -1]),  # Exclude the first column (Y)
  P_Value = NA,
  Statistic = NA
)

# Perform Kruskal-Wallis test for each predictor
for (i in 2:ncol(prostate)) {  # Start from the second column since the first is Y
  kruskal_test <- kruskal.test(prostate[, i] ~ prostate[, 1])  # Use the first column as Y
  kruskal_results$Statistic[i - 1] <- kruskal_test$statistic  # Store the test statistic
  kruskal_results$P_Value[i - 1] <- kruskal_test$p.value      # Store the p-value
}

# Sort predictors by p-value
kruskal_results <- kruskal_results[order(-kruskal_results$P_Value), ]

# Sort predictors by statistic
kruskal_results <- kruskal_results[order(kruskal_results$Statistic), ]

# Select the top 9 predictors
down_9_predictors <- head(kruskal_results, 9)

# Print the top 9 predictors
print(down_9_predictors)
```

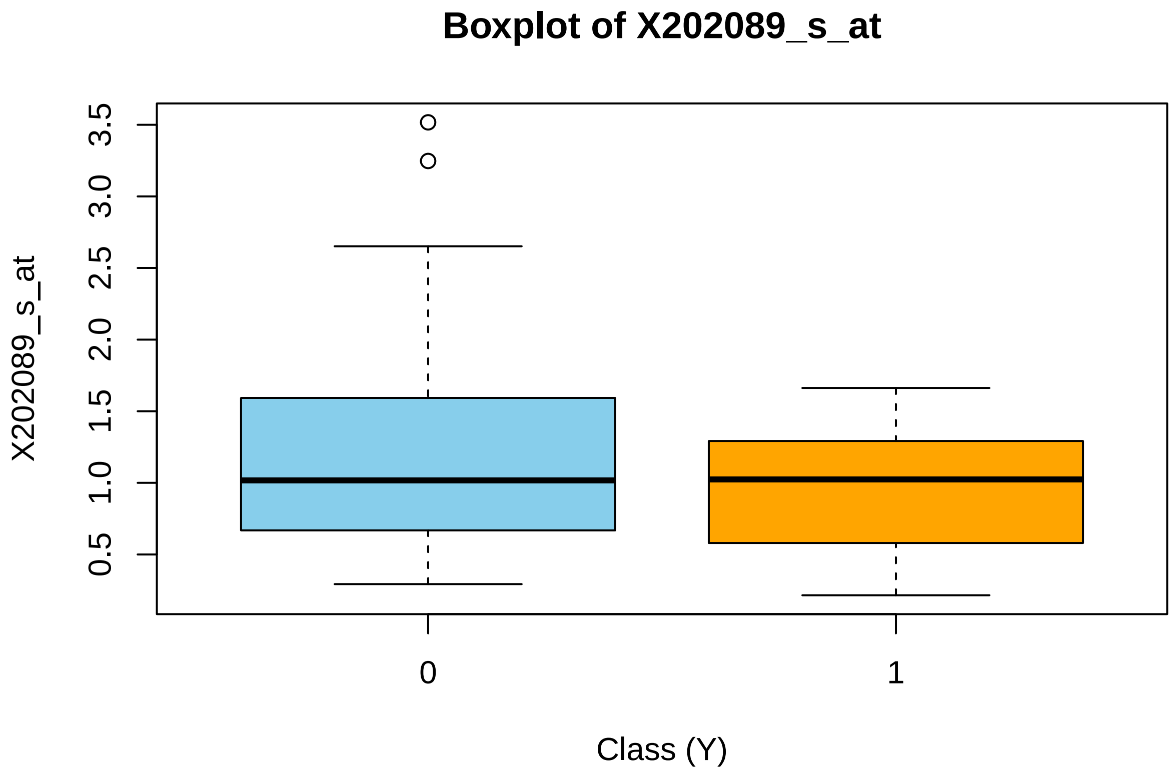
```
##        Predictor    P_Value Statistic
## 279 X221580_s_at 0.20501634  1.606274
## 97  X202089_s_at 0.15145560  2.057529
## 23  X200793_s_at 0.10085389  2.691988
## 55  X204905_s_at 0.09882994  2.724324
## 336   X208838_at 0.09487937  2.789575
## 142   X201975_at 0.09295208  2.822490
## 476 X208140_s_at 0.09295208  2.822490
## 206   X204058_at 0.08735818  2.922394
## 321   X215346_at 0.08378217  2.989961
```

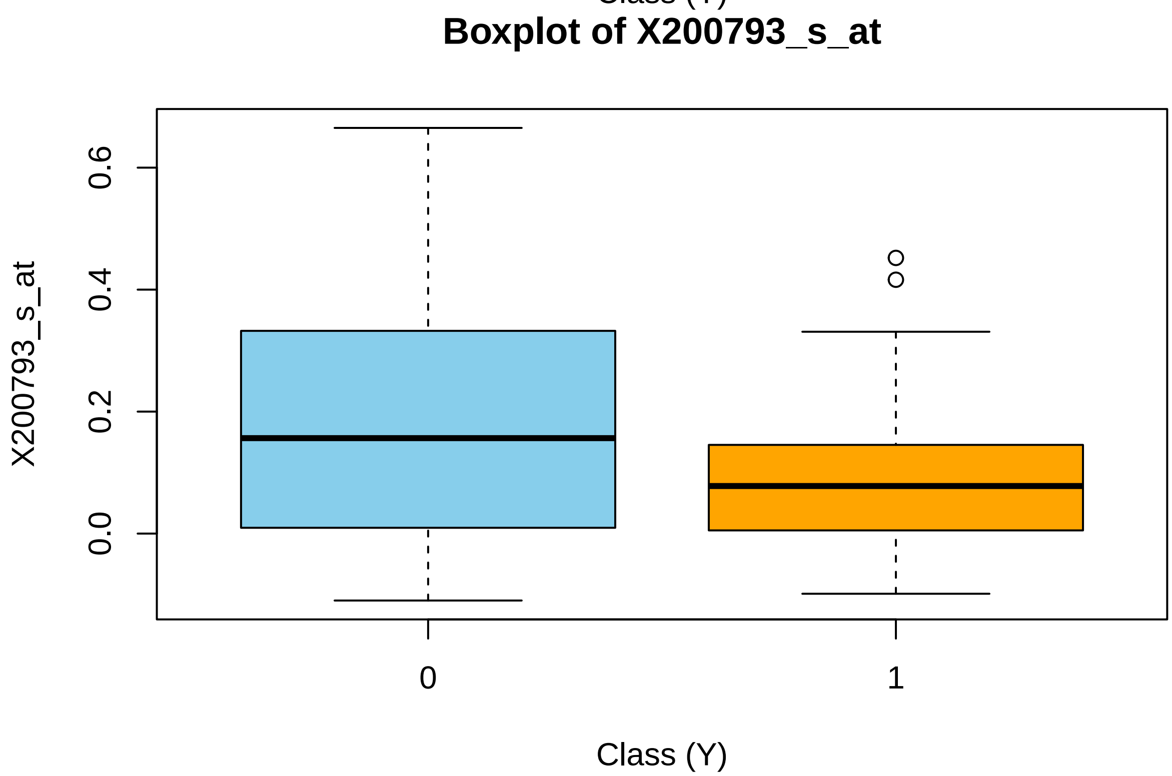
And the boxplots are :

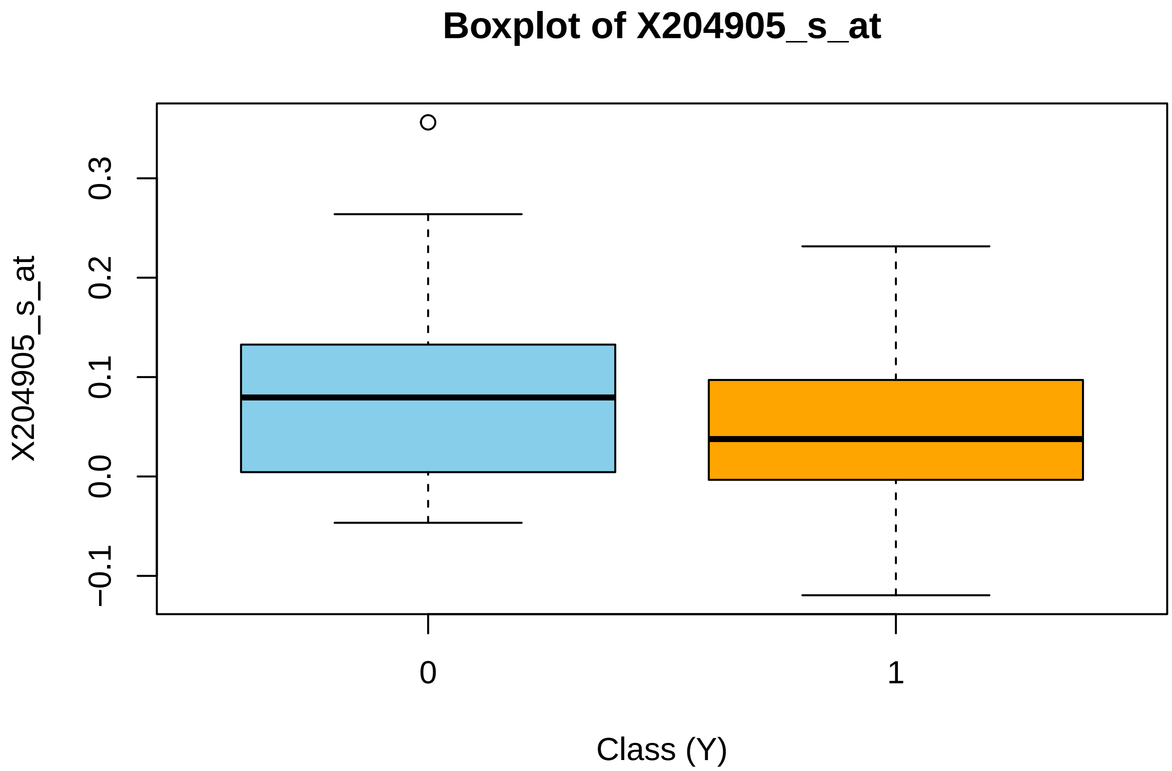
```
# Extract the top 9 predictor names
down_9 <- down_9_predictors$Predictor

# Generate boxplots for each of the top 9 predictors
#par(mfrow = c(3, 3)) # Arrange plots in a 3x3 grid
for (predictor in down_9) {
  boxplot(
```

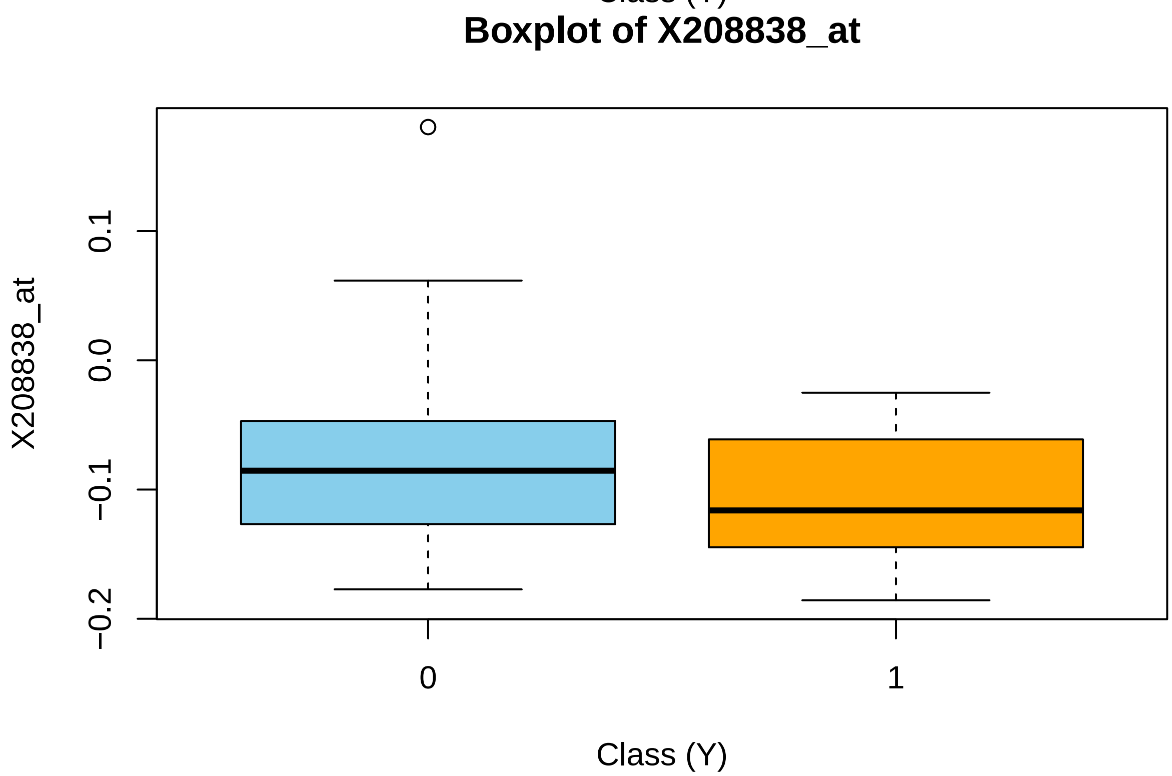
```
    prostate[[predictor]] ~ prostate$Y,
    main = paste("Boxplot of", predictor),
    xlab = "Class (Y)",
    ylab = predictor,
    col = c("skyblue", "orange"),
    border = "black"
  )
}
```



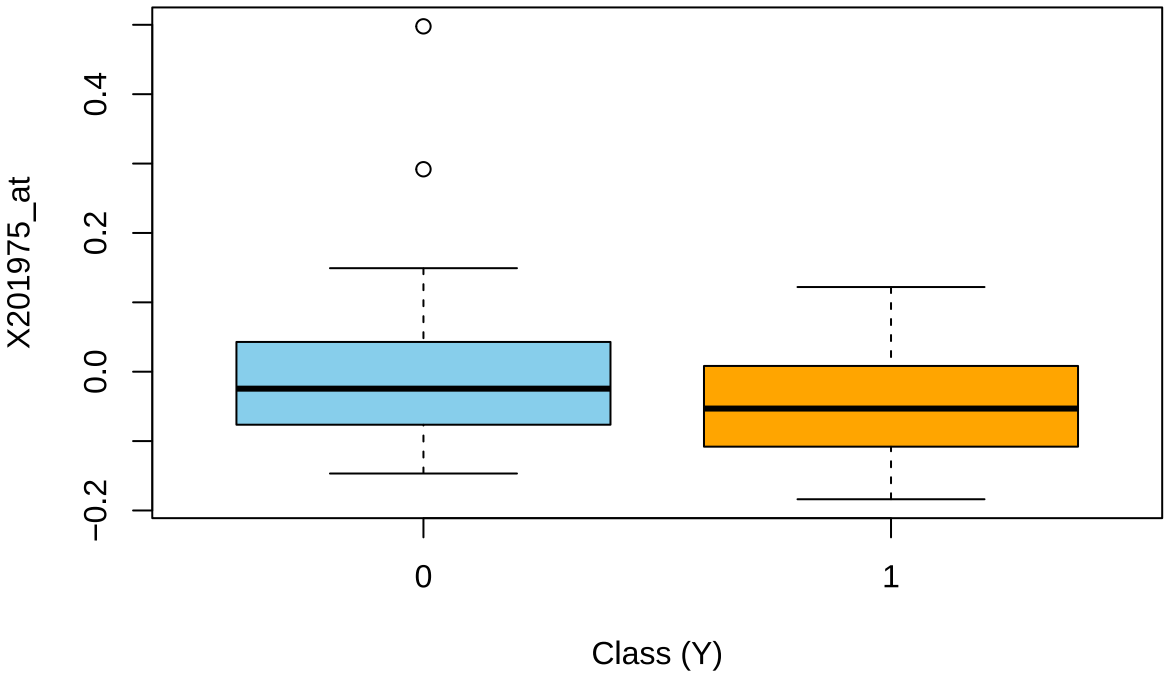

Boxplot of X202089_s_at
X202089_s_at
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
1
Class (Y)


Boxplot of X200793_s_at
X200793_s_at
0.0
0.2
0.4
0.6
0
1
Class (Y)

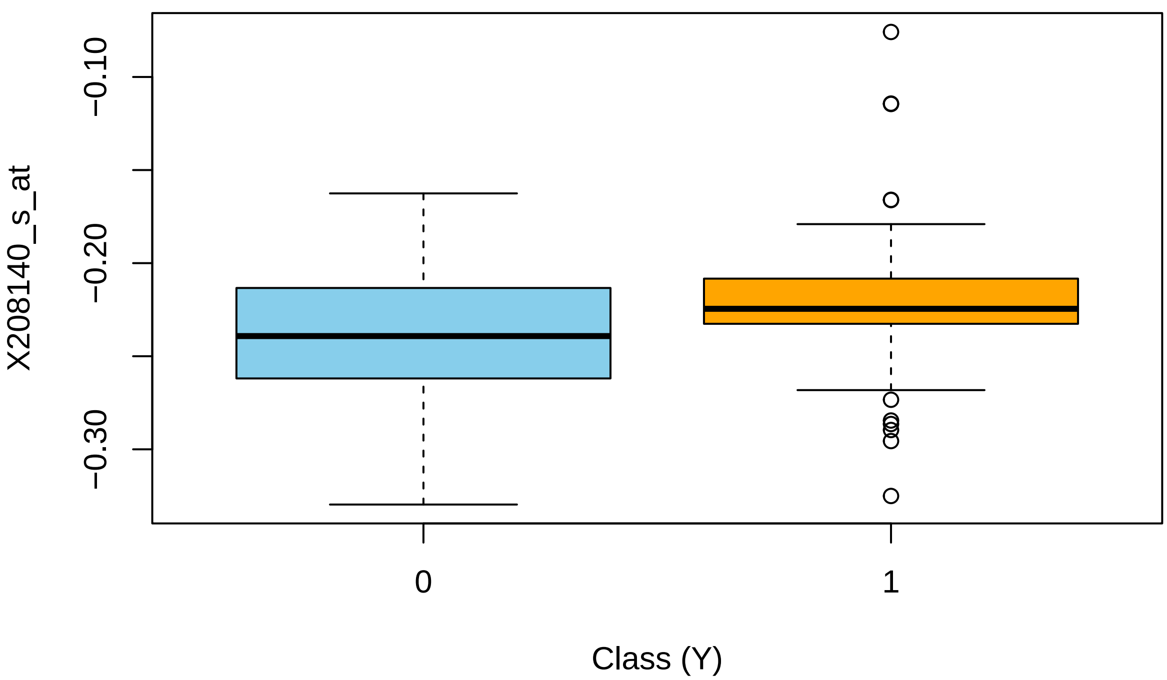

Boxplot of X204905_s_at
X204905_s_at
−0.1
0.0
0.1
0.2
0.3
0
1
Class (Y)

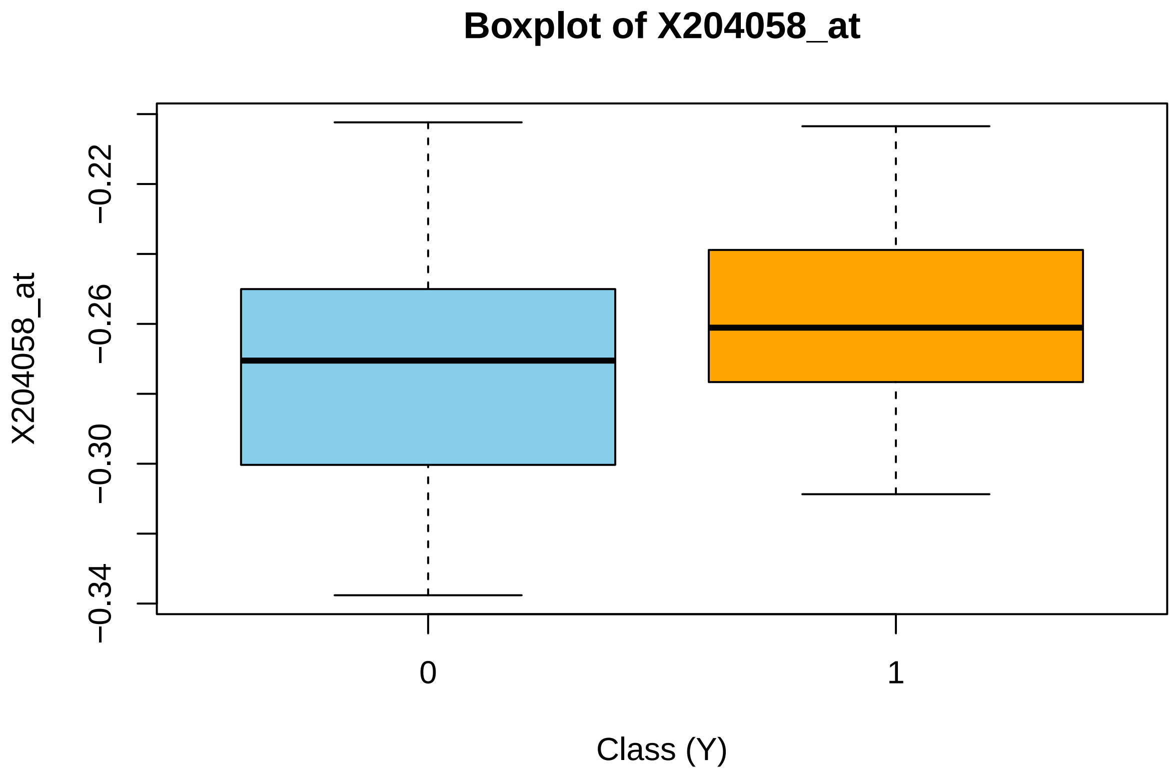

Boxplot of X208838_at
X208838_at
−0.2
−0.1
0.0
0.1
0
1
Class (Y)

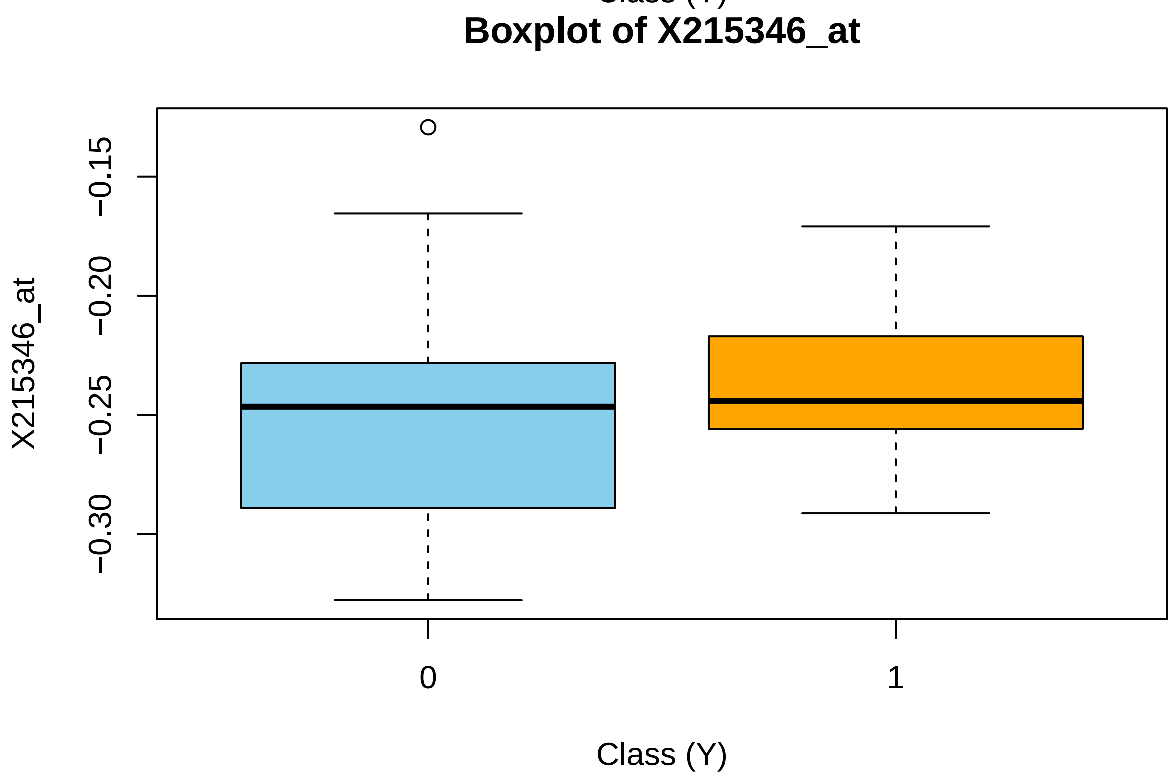
Boxplot of X201975_at



Boxplot of X208140_s_at







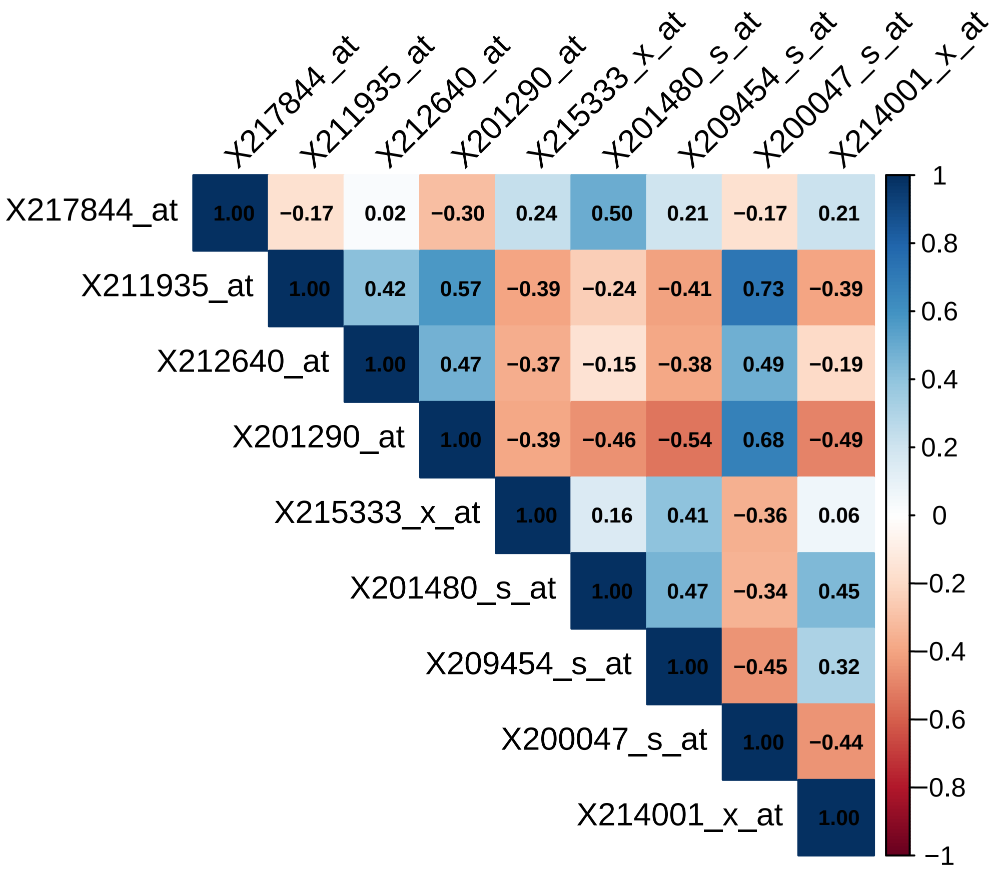
Comment : Even if most of this variables are not able to differenciate or sparate carefully the classes, we can remark that some variables like *X200793_s_at* , *X208838_at* and *X204905_s_at* show us a possible significant difference between the medians of the twoo classes, which could mean that theur discriminative power is not so negligible.

9. Let's generate the correlation plot of the top 9 best predictors

```r
# Subset the data to include only the 9 weakest variables
top_data <- prostate[, top_9]

# Compute the correlation matrix
cor_matrix <- cor(top_data)

# Generate the correlation plot
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7)
```



Comment : We can denote some important correlation such that : - The biggest positive correlation is between the variables *X211935_at* and *X200047_s_at* (0.73) - The biggest negative correlation is between the variables *X201290* and *X209454* (-0.54) - We can also identify another variables which are enough significantly correlated (positively or negatively) - We have also some variables with are weekly correlated such that *X215333* and *X214001* (0.06) - The predictors that are highly correlated can lead to redundancy in the model. - On the other side, the weakest correlation between some predictors means that this variables are likely independent. It means that they contribute by bringing distinct information to the model, which can improve the performance of the model.

```r
# Compute the correlation matrix of the dataset (including top and bottom 9 variables)
corr_matrix_full <- cor(prostate[, top_9_predictors$Predictor])
```

```r
# Perform eigendecomposition
eig_decomp <- eigen(corr_matrix_full)

# Extract eigenvalues
eigenvalues <- eig_decomp$values

# Calculate the ratio of max eigenvalue to min eigenvalue
lambda_max <- max(eigenvalues)
lambda_min <- min(eigenvalues)
lambda_ratio <- lambda_max / lambda_min

# Output the results
lambda_ratio
```

```
## [1] 17.02747
```

Comment : We observe a ratio equal to 17.02 which is significantly big. First, let's remember that the eigen decomposition of the correlation matrix provides us the insight into the variability structure of these variables. Then : - The high ratio mean that we are in presence of redundancy, a few number of variables contain the maximum of the total information while other provides little additional information. This is not suprising due to the fact that we observed a lot of significant correlations between some of this variables. - In context of tree classification, even if the redundancy doesn't harm tree accuracy, it can inflate the complexity of the tree because similar features may appear repeatedly. - In the context of kNN, which uses distance to classify observations, if several predictors are correlated, they may dominate the distance calculation and make the model overly sensitive to a subset of redundant variables

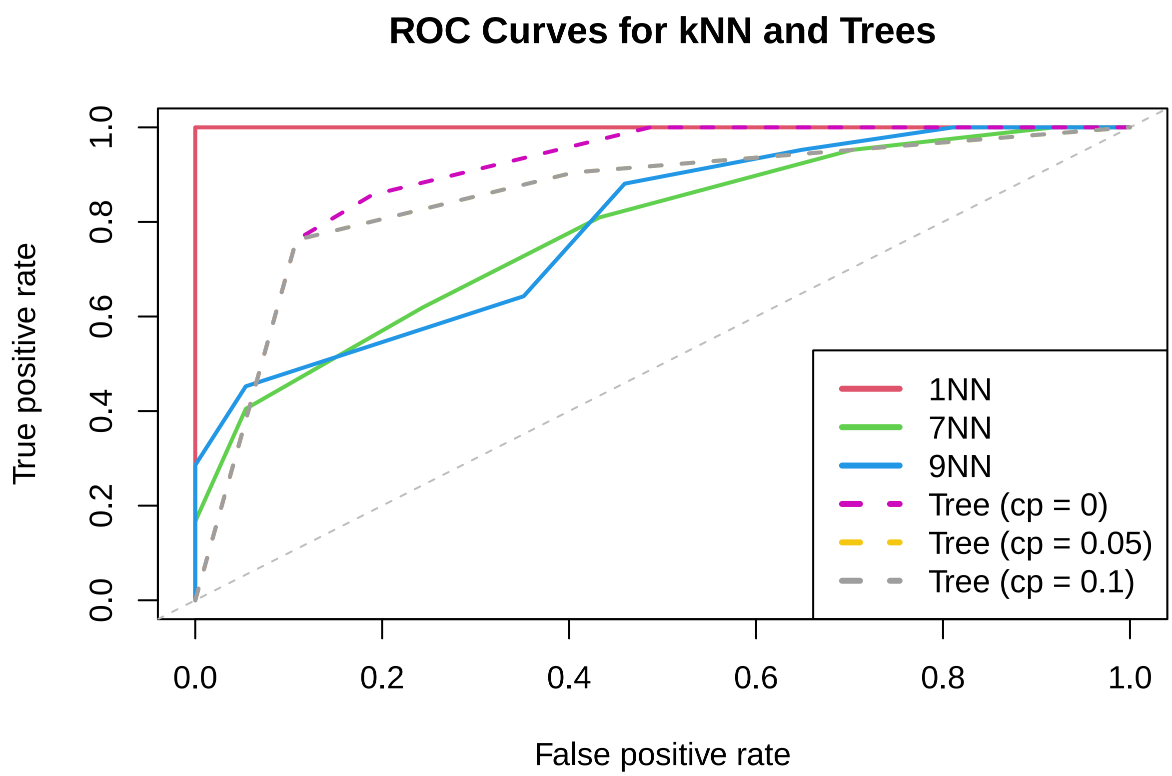
11. Let's use the whole data as training and test set and plot the ROC curves for the six machines

```r
# Resehape our variables
xy=prostate
  n   <- nrow(xy)       # Sample size
   p   <- ncol(xy) - 1   # Dimensionality of the input space
   pos <- 1              # Position of the response
   x   <- xy[,-pos]      # Data matrix: n x p matrix

   #xy[,pos] <- ifelse(xy[,pos]==1,1,0)

   y   <- as.factor(xy[, pos])      # Response vector
```

```r
library(ROCR)
  y.roc   <- as.factor(y)
  # 1NNN
kNN.mod_1 <- class::knn(x, x, y.roc, k = 1, prob = TRUE)
prob_1 <- attr(kNN.mod_1, 'prob')
prob_1 <- 2 * ifelse(kNN.mod_1 == "0", 1 - prob_1, prob_1) - 1
pred.knn_1 <- prediction(prob_1, y.roc)
perf.knn_1 <- performance(pred.knn_1, measure = 'tpr', x.measure = 'fpr')

# 7NN
kNN.mod_7 <- class::knn(x, x, y.roc, k = 7, prob = TRUE)
prob_7 <- attr(kNN.mod_7, 'prob')
prob_7 <- 2 * ifelse(kNN.mod_7 == "0", 1 - prob_7, prob_7) - 1
pred.knn_7 <- prediction(prob_7, y.roc)
```

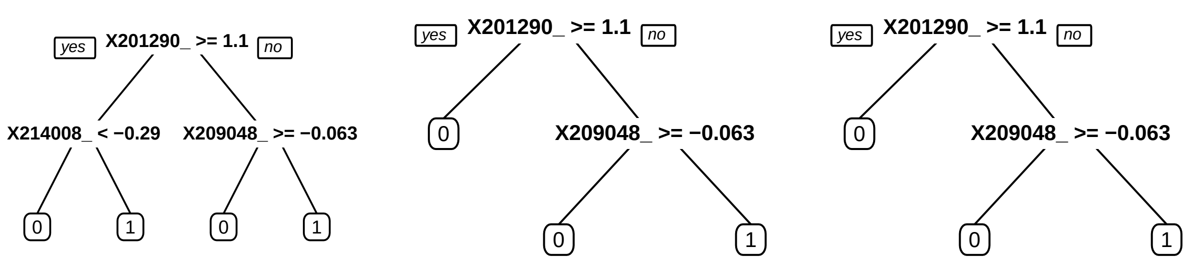
```r
perf.knn_7 <- performance(pred.knn_7, measure = 'tpr', x.measure = 'fpr')

#9NN
kNN.mod_9 <- class::knn(x, x, y.roc, k = 9, prob = TRUE)
prob_9 <- attr(kNN.mod_9, 'prob')
prob_9 <- 2 * ifelse(kNN.mod_9 == "0", 1 - prob_9, prob_9) - 1
pred.knn_9 <- prediction(prob_9, y.roc)
perf.knn_9 <- performance(pred.knn_9, measure = 'tpr', x.measure = 'fpr')


#Tree cp =0
tree_0 <- rpart(y.roc ~ ., data = x, control = rpart.control(cp = 0))
pred_tree_0_prob <- predict(tree_0, x, type = 'prob')[,2]
pred_tree_0 <- prediction(pred_tree_0_prob, y.roc)
perf_tree_0 <- performance(pred_tree_0, measure = 'tpr', x.measure = 'fpr')


#Tree cp =0.05
tree_1 <- rpart(y.roc ~ ., data = x, control = rpart.control(cp = 0.05))
pred_tree_1_prob <- predict(tree_1, x, type = 'prob')[,2]
pred_tree_1 <- prediction(pred_tree_1_prob, y.roc)
perf_tree_1 <- performance(pred_tree_1, measure = 'tpr', x.measure = 'fpr')


#Tree cp =0.1
tree_2 <- rpart(y.roc ~ ., data = x, control = rpart.control(cp = 0.1))
pred_tree_2_prob <- predict(tree_2, x, type = 'prob')[,2]
pred_tree_2 <- prediction(pred_tree_2_prob, y.roc)
perf_tree_2 <- performance(pred_tree_2, measure = 'tpr', x.measure = 'fpr')
```

```r
# Create an empty plot for the ROC curves
plot(perf.knn_1, col = 2, lwd = 2, lty = 1, main = "ROC Curves for kNN and Trees")
abline(a = 0, b = 1, col = "gray", lty = 2)  # Random classifier line

# Add the ROC curves for each model
plot(perf.knn_7, col = 3, lwd = 2, lty = 1, add = TRUE)  # 7NN
plot(perf.knn_9, col = 4, lwd = 2, lty = 1, add = TRUE)  # 9NN
plot(perf_tree_0, col = 6, lwd = 2, lty = 2, add = TRUE)  # Tree cp = 0
plot(perf_tree_1, col = 7, lwd = 2, lty = 2, add = TRUE)  # Tree cp = 0.05
plot(perf_tree_2, col = 8, lwd = 2, lty = 2, add = TRUE)  # Tree cp = 0.1

# Add a legend to explain the colors and line types
legend("bottomright",
       legend = c("1NN", "7NN", "9NN", "Tree (cp = 0)", "Tree (cp = 0.05)", "Tree (cp = 0.1)"),
       col = c(2, 3, 4, 6, 7, 8),
       lty = c(1, 1, 1, 2, 2, 2),
       lwd = 3)
```



12. Let's plot all of the tree grown.

```
par(mfrow=c(1,3))
prp(tree_0) #cp = 0
prp(tree_1) # cp = 0.05
prp(tree_2) #cp = 0.1
```




```
par(mfrow=c(1,1))
```

13. Comment : Based on the ROC curves, we can deduce that :

- The 1NN machine is clearly overfitting. There is no error in the predictions. This is expected because choose k=1 is the most complex kNN model and since we use the same data to train and test, its normal to not have any error in the predictions.
- The model 1NN take out, the best model based on this ROC curve is the tree with cp=0
- From the plot of the trees, we can see that the tree 2 (cp=0.05) and the tree 3 (cp=0.1) has give exatcly the same tree. It's means that by vari the cp parameter from 0.05 to 0.1, we don't it doesn't change the number of region given by the tree. And on the ROC curve of this two tree are confused and they are the best model after the tree with cp=0.
- Finally, the model 7NN and 9NN are the worse based on this ROC curves.

- Overall, the tree with cp=0 seem perform well than the other machines. However, these models are built using the same data as training and test set. SO, we cannot truth any conclusion on the predictive performance of the model. Our conclusion will be more useful when we'll use a test data different to the train data.

14. Let's :

- Plot the comparative boxplots :

```
#  Split the data

set.seed (19671210)          # Set seed for random number generation to be reproducible

epsilon <- 1/3               # Proportion of observations in the test set
nte     <- round(n*epsilon)  # Number of observations in the test set
ntr     <- n - nte

id.tr   <- sample(sample(sample(n)))[1:ntr]   # For a sample of ntr indices from {1,2,..,n}

#id .tr <- sample(1:n, ntr, replace=F)        # Another way to draw from {1,2,..n}
id.te   <- setdiff(1:n, id.tr)
```

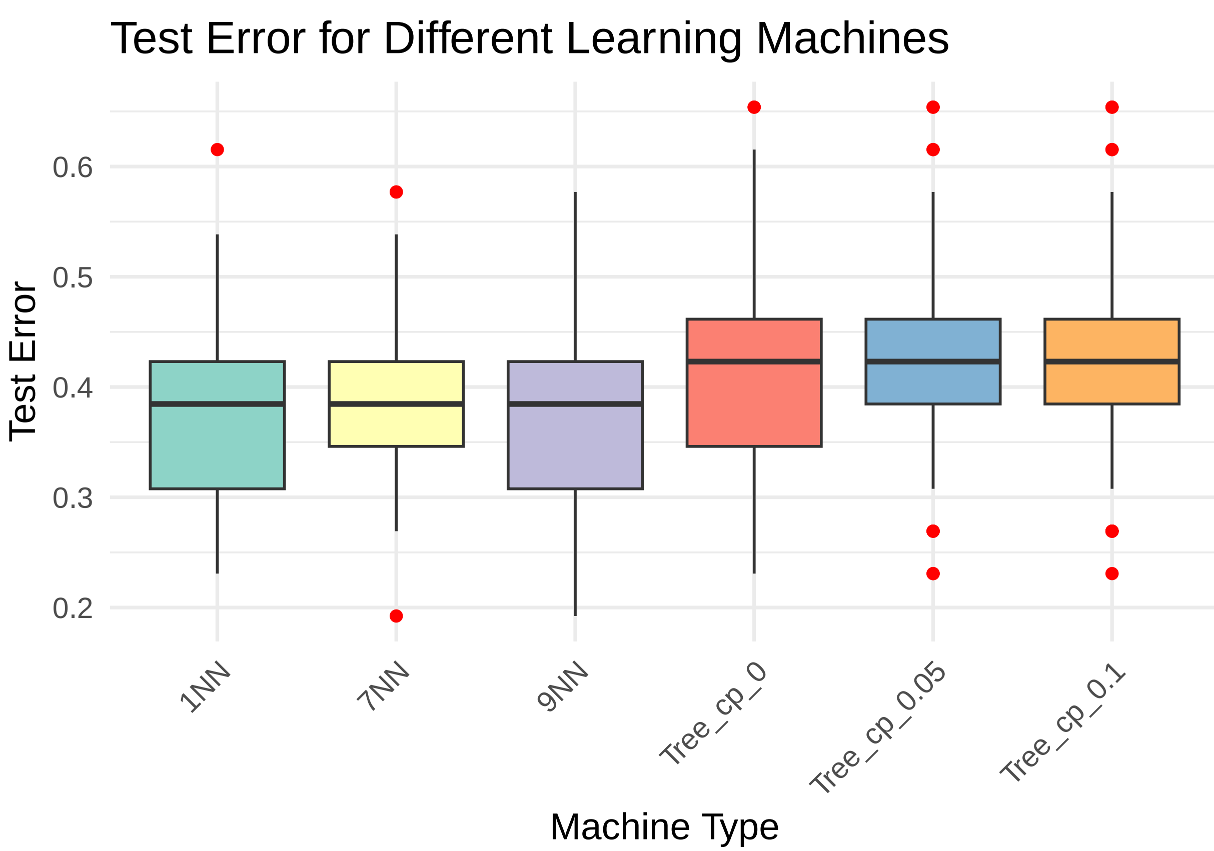
```
 stratified.holdout <- function(y, ptr)
 {
     n              <- length(y)
     labels         <- unique(y)       # Obtain classifiers
     id.tr <- id.te <- NULL

     # Loop once for each unique label value

     y <- sample(sample(sample(y)))

     for(j in 1:length(labels))
     {
       sj    <- which(y==labels[j])  # Grab all rows of label type j
       nj    <- length(sj)           # Count of label j rows to calc proportion below

       id.tr <- c(id.tr, (sample(sample(sample(sj))))[1:round(nj*ptr)])
     }                               # Concatenates each label type together 1 by 1

     id.te  <- (1:n) [-id.tr]          # Obtain and Shuffle test indices to randomize

 return(list(idx1=id.tr,idx2=id.te))

 }
```

```
epsilon <- 1/3               # Proportion of observations in the test set

 R <- 100   # Number of replications
 test.err <- matrix(0, nrow=R, ncol=6)

 library(class)

 for(r in 1:R)
 {
```

```r
# Split the data

hold <- stratified.holdout(as.factor(xy[,pos]), 1-epsilon)
id.tr <- hold$idx1
id.te <- hold$idx2
ntr   <- length(id.tr)
nte   <- length(id.te)

y.te         <- y[id.te]                           # True responses in test set


# 1-Nearest Neighbors Learning Machine

y.te.hat       <- knn(x[id.tr,], x[id.te,], y.roc[id.tr], k=1, prob=TRUE)
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
test.err[r,1]  <- mean(ind.err.te)


# 7-Nearest Neighbors Learning Machine

y.te.hat       <- knn(x[id.tr,], x[id.te,], y.roc[id.tr], k=7, prob=TRUE)
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
test.err[r,2]  <- mean(ind.err.te)


# 9-Nearest Neighbors Learning Machine

y.te.hat       <- knn(x[id.tr,], x[id.te,], y.roc[id.tr], k=9, prob=TRUE)
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
test.err[r,3]  <- mean(ind.err.te)


# Classification Trees cp = 0

tree.mod       <- rpart(as.factor(Y)~., data=xy[id.tr, ], control = rpart.control(cp=0))
y.te.hat       <- predict(tree.mod, x[id.te, ], type='class')
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
test.err[r,4]  <- mean(ind.err.te)


# Classification Trees cp = 0.05

tree.mod       <- rpart(as.factor(Y)~., data=xy[id.tr, ], control = rpart.control(cp=0.05))
y.te.hat       <- predict(tree.mod, x[id.te, ], type='class')
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
test.err[r,5]  <- mean(ind.err.te)


# Classification Trees cp = 0.1

tree.mod       <- rpart(as.factor(Y)~., data=xy[id.tr, ], control = rpart.control(cp=0.1))
y.te.hat       <- predict(tree.mod, x[id.te, ], type='class')
ind.err.te     <- ifelse(y.te!=y.te.hat,1,0)  # Random variable tracking error. Indicator
```

```
    test.err[r,6]  <- mean(ind.err.te)
  }
```

```
# Convert test errors into a tidy data frame
test <- data.frame(test.err)
Method <- c('1NN', '7NN', '9NN', 'Tree_cp_0', 'Tree_cp_0.05', 'Tree_cp_0.1')
colnames(test) <- Method

# Use stack() to reshape the data to long format
test_long <- stack(test)
colnames(test_long) <- c("TestError", "Method")

# Plot the boxplot using ggplot2
library(ggplot2)

ggplot(test_long, aes(x = Method, y = TestError, fill = Method)) +
  geom_boxplot(outlier.colour = "red", outlier.shape = 16, outlier.size = 2) +
  labs(title = "Test Error for Different Learning Machines",
       x = "Machine Type",
       y = "Test Error") +
  scale_fill_brewer(palette = "Set3") +  # Use a clean color palette
  theme_minimal(base_size = 14) +        # Minimal theme with large fonts
  theme(axis.text.x = element_text(angle = 45, hjust = 1),  # Rotate x-axis labels
        legend.position = "none")        # Remove the legend
```



-

Comment on the distribution of the errors in light of model complexity: - For the kNN model, we know that the least complex model is when k=9 (in our case). However, we can remark that there is no difference between between the predictive performance of the model when the complexity is increasing until k=1, the most complex model. - By the same way, we can remark the same thing on the tree classification models where the complexity in controled by the parameter cp. The least complex model in our case is when cp=0.1, however, as we reduce this parameter (then, increase the complexity), there is no significant difference between the other trees we get. - We can remark that even if the difference seem not too significant, the kNN are likely perform well on the test data than the tree classification.

- Perform ANOVA on the machines. We get the following confident intervals which can be interpreted like this :
- When zero belong to a interval, its mean the difference between these variables are not significant. A contrario, when the interval don't contain zero, its means that the difference between this two models is significant. Thus, we can conclude that :
- There is no difference in the predict performance of the all three kNN models. By the same way, all of the tree classification don't present any difference in their predictive performance.
- As we expected from the boxplot, there is a significant difference between the predictive performance of all of the tree compare to all of the kNN models.

```r
# Convert test errors into a tidy data frame
test <- data.frame(test.err)
Method <- c('1NN', '7NN', '9NN', 'Tree_cp_0', 'Tree_cp_0.05', 'Tree_cp_0.1')
colnames(test) <- Method

# Reshape the data into long format using stack()
test_long <- stack(test)
colnames(test_long) <- c("TestError", "Method")

# Perform the ANOVA test
anova_result <- aov(TestError ~ Method, data = test_long)

# Print ANOVA summary
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Method        5  0.238 0.04764   7.013 2.22e-06 ***
## Residuals   594  4.035 0.00679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

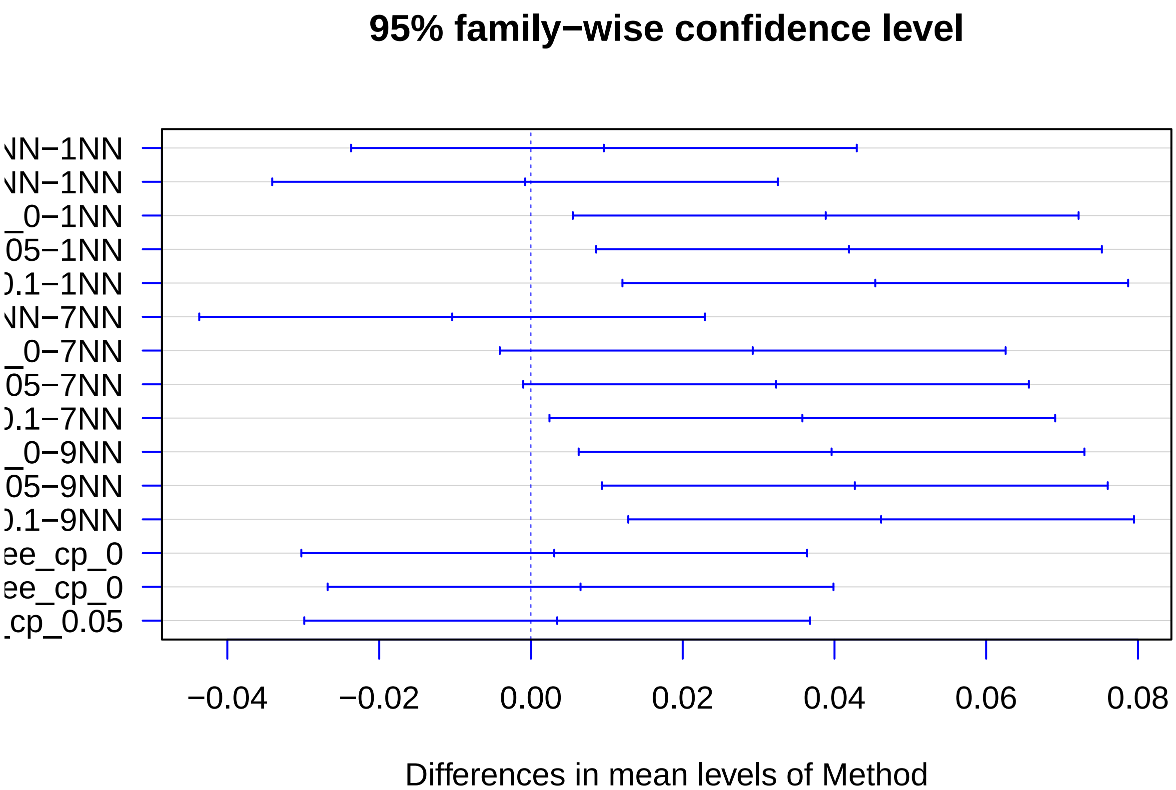
```r
# Post-hoc analysis: Tukey's Honest Significant Difference (HSD) test
tukey_result <- TukeyHSD(anova_result)

# Print Tukey HSD results
print(tukey_result)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = TestError ~ Method, data = test_long)
##
## $Method
##                                 diff          lwr        upr     p adj
## 7NN-1NN                 0.0096153846 -0.023709871 0.04294064 0.9628902
## 9NN-1NN                -0.0007692308 -0.034094486 0.03255602 0.9999998
## Tree_cp_0-1NN           0.0388461538  0.005520899 0.07217141 0.0117195
## Tree_cp_0.05-1NN        0.0419230769  0.008597822 0.07524833 0.0046855
## Tree_cp_0.1-1NN         0.0453846154  0.012059360 0.07870987 0.0015308
## 9NN-7NN                -0.0103846154 -0.043709871 0.02294064 0.9486533
```

```
## Tree_cp_0-7NN             0.0292307692 -0.004094486 0.06255602 0.1234126
## Tree_cp_0.05-7NN          0.0323076923 -0.001017563 0.06563295 0.0634832
## Tree_cp_0.1-7NN           0.0357692308  0.002443976 0.06909449 0.0271550
## Tree_cp_0-9NN             0.0396153846  0.006290129 0.07294064 0.0093848
## Tree_cp_0.05-9NN          0.0426923077  0.009367052 0.07601756 0.0036830
## Tree_cp_0.1-9NN           0.0461538462  0.012828591 0.07947910 0.0011795
## Tree_cp_0.05-Tree_cp_0    0.0030769231 -0.030248332 0.03640218 0.9998263
## Tree_cp_0.1-Tree_cp_0     0.0065384615 -0.026786794 0.03986372 0.9934255
## Tree_cp_0.1-Tree_cp_0.05  0.0034615385 -0.029863717 0.03678679 0.9996902
```

- The following figure confirm all of this conclusions.

```
plot(tukey_result, las = 1, col = "blue")
```



15. Comment on lesson :

From, this exercise, we observe and learn many things like : - How handle with a dataset which have more variables than observations. Some techniques (in our case the Krustal Wallis Test) can be used to find the best predictors variables.