**Assignment R scripts demonstrated with a simple example of 5 data sets for both training and testing datasets:**

# The intent was to mimic the real set up – there are two separate subfolders and test and train data are in these separate subfolders.

# only the first five data sets are taken to showcase the R script.

# Since test and train folders had the same activity code, changes were made to mimic the actual larger data bases better

# The datasets used are included in the zip folder uploade

# the .txt data files in the s\_test and s\_train folders (corresponding to test and train folders of the large database are labeled to be easy to recall:

#X\_test is now STestDataFeatures; y\_test is now STestActivityList; and subject\_test is now STestSubjectList. Similarly for the train folder .txt files.

# switch to project working directory and load x\_test data .

# Results of most steps are included. When long output, it has been shortened with “……” in the middle

# It is a zip upload since I was not sure how the .md file might reformat and chang the colors etc.

|  |
| --- |
| > directory <- "C:/Users/……/CleanData/Project/getdata\_projectfiles\_UCI HAR Dataset"  # complete path not shown. There may be other directories in the absolute path  > setwd(directory)  > getwd()  [1] "C:/Users/……../CleanData/Project/getdata\_projectfiles\_UCI HAR Dataset" |

> testdata = read.table(file=file.choose())

# This opens a browser window and you can choose the file to import

# Import X\_test of my demo set with 4 data sets (included in the zip folder – labeled as STestDataFeatures)

> testdata

V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11

1 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 -0.9525011 -0.9252487 -0.6743022 -0.8940875 -0.5545772

2 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 -0.9867988 -0.9684013 -0.9458234 -0.8940875 -0.5545772

3 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 -0.9944034 -0.9707350 -0.9634827 -0.9392603 -0.5685122

4 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 -0.9952743 -0.9744710 -0.9688974 -0.9386098 -0.5685122

V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22

1 -0.4662230 0.7172085 0.6355024 0.7894967 -0.8777642 -0.9977661 -0.9984138 -0.9343453 -0.9756690 -0.9498237 -0.8304778

2 -0.8060133 0.7680313 0.6836980 0.7967058 -0.9690965 -0.9995795 -0.9996456 -0.9977203 -0.9940063 -0.9736367 -0.9507199

3 -0.7991165 0.8483052 0.6678642 0.8224423 -0.9767823 -0.9999539 -0.9995698 -0.9987200 -0.9934559 -0.9742661 -0.9649758

4 -0.7991165 0.8483052 0.6678642 0.8224423 -0.9744184 -0.9999534 -0.9993302 -0.9989332 -0.9952355 -0.9787394 -0.9699818

……

V540 V541 V542 V543 V544 V545 V546 V547 V548 V549 V550

1 -0.46784901 -0.7613258 -0.8901655 -0.9073076 -0.8953006 -0.9178830 -0.9098288 -0.8901655 -0.9941054 -0.8980215 -0.2348153

2 -0.09065485 -0.4618011 -0.9519774 -0.9382124 -0.9380079 -0.9499222 -0.9869024 -0.9519774 -0.9982715 -0.9512549 -0.4310534

3 -0.66231704 -0.9253638 -0.9856888 -0.9832727 -0.9834216 -0.9868573 -0.9896086 -0.9856888 -0.9998306 -0.9861480 -0.8099757

4 -0.44647773 -0.7568553 -0.9855619 -0.9858429 -0.9842175 -0.9885656 -0.9920207 -0.9855619 -0.9998500 -0.9840150 -0.8490812

V551 V552 V553 V554 V555 V556 V557 V558 V559 V560

1 -1.0000000 0.07164545 -0.3303704 -0.7059739 0.006462403 0.16291982 -0.82588562 0.2711515 -0.7200093 0.2768010

2 -1.0000000 -0.40118872 -0.1218451 -0.5949439 -0.083494968 0.01749957 -0.43437455 0.9205932 -0.6980908 0.2813429

3 -0.9365079 0.06289131 -0.1904219 -0.6407357 -0.034956250 0.20230203 0.06410335 0.1450684 -0.7027715 0.2800830

4 -0.9365079 0.11669529 -0.3444180 -0.7361238 -0.017067021 0.15443783 0.34013408 0.2964071 -0.6989538 0.2841138

V561

1 -0.05797830

2 -0.08389801

3 -0.07934620

4 -0.07710800

# load X\_train data now

> traindata = read.table(file=file.choose())

# This opens a browser window in which you can choose the data file to import

# Import X\_train of my demo set with 4 data sets (included in the zip folder – labeled as STrainDataFeatures)

> dim(traindata)

[1] 4 561

> dim(testdata)

[1] 4 561

#combine or merge the two data sets

|  |
| --- |
| > alldata <- rbind (testdata, traindata)  > dim(alldata)  [1] 8 561 |
|  |
| # extract parameters of interest  > alldatapart1<- alldata[,1:6]  > alldatapart2<- alldata[, 41:46]  > alldatapart3<- alldata[, 121:126]  > alldatapart1  V1 V2 V3 V4 V5 V6  1 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833  2 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582  3 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480  4 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907  5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264  6 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220  7 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440  8 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751  > alldatapart2  V41 V42 V43 V44 V45 V46  1 0.9364893 -0.2827192 0.11528825 -0.9254273 -0.9370141 -0.5642884  2 0.9274036 -0.2892151 0.15256831 -0.9890571 -0.9838872 -0.9647811  3 0.9299150 -0.2875128 0.14608559 -0.9959365 -0.9882505 -0.9815796  4 0.9288814 -0.2933958 0.14292592 -0.9931392 -0.9704192 -0.9915917  5 0.9633961 -0.1408397 0.11537494 -0.9852497 -0.9817084 -0.8776250  6 0.9665611 -0.1415513 0.10937881 -0.9974113 -0.9894474 -0.9316387  7 0.9668781 -0.1420098 0.10188392 -0.9995740 -0.9928658 -0.9929172  8 0.9676152 -0.1439765 0.09985014 -0.9966456 -0.9813928 -0.9784764  # Combine them into one common database.  #**Note**: delete the original databases loaded to conserver memory space – not done here  > alldataasg<- cbind(alldatapart1, alldatapart2, alldatapart3)  V1 V2 V3 V4 V5 V6 V41 V42 V43  1 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 0.9364893 -0.2827192 0.11528825  2 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 0.9274036 -0.2892151 0.15256831  3 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 0.9299150 -0.2875128 0.14608559  4 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 0.9288814 -0.2933958 0.14292592  5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264 0.9633961 -0.1408397 0.11537494  6 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220 0.9665611 -0.1415513 0.10937881  7 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440 0.9668781 -0.1420098 0.10188392  8 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751 0.9676152 -0.1439765 0.09985014  V44 V45 V46 V121 V122 V123 V124 V125  1 -0.9254273 -0.9370141 -0.5642884 0.119976160 -0.09179234 0.18962854 -0.8830891 -0.8161636  2 -0.9890571 -0.9838872 -0.9647811 -0.001552463 -0.18729119 0.18070522 -0.9255665 -0.9295992  3 -0.9959365 -0.9882505 -0.9815796 -0.048212671 -0.16627974 0.15417437 -0.9729882 -0.9785106  4 -0.9931392 -0.9704192 -0.9915917 -0.056641577 -0.12601846 0.11833718 -0.9677905 -0.9751338  5 -0.9852497 -0.9817084 -0.8776250 -0.006100849 -0.03136479 0.10772540 -0.9853103 -0.9766234  6 -0.9974113 -0.9894474 -0.9316387 -0.016111620 -0.08389378 0.10058429 -0.9831200 -0.9890458  7 -0.9995740 -0.9928658 -0.9929172 -0.031698294 -0.10233542 0.09612688 -0.9762921 -0.9935518  8 -0.9966456 -0.9813928 -0.9784764 -0.043409983 -0.09138618 0.08553770 -0.9913848 -0.9924073  V126  1 -0.9408812  2 -0.9675810  3 -0.9756483  4 -0.9632327  5 -0.9922053  6 -0.9891212  7 -0.9863787  8 -0.9875542  # Load y\_test data  > testact <- read.table ("C:/Users/……/CleanData/Project/s\_test/STestActivityList.txt")  #Full path not shown above You can also use the browser method shown above – look for STestActivityList.txt file in s\_test  # activity type data in test data set  > testact  V1  1 5  2 5  3 5  4 5  # Load subject\_test data  > testsubj <- read.table ("C:/Users/……/CleanData/Project/s\_test/STestSubjectList.txt")  # Full path not shown above. You can also use the browser method shown above – look for STestSubjectList.txt file in s\_test  # subject id data in test data set  > testsubj  V1  1 2  2 2  3 2  4 2  # Similarly, Load train data for both  > trainact <- read.table ("C:/Users/……../CleanData/Project/s\_train/STrainActivityList.txt")  #Full path not shown above. You can also use the browser method shown above – look for STrainActivityList.txt file in s\_train  # activity type data in training data set  > trainact  V1  1 5  2 5  3 5  4 5  > trainsubj <- read.table ("C:/Users/……./CleanData/Project/s\_train/STrainSubjectList.txt")  #Full path not shown above. You can also use the browser method shown above – look for STrainSubjectList.txt file in s\_train  #subjecvt id data in train data set  > trainsubj  V1  1 1  2 1  3 1  4 1  # combine the two datasets  > allact<- rbind(testact, trainact)  > allsubj<- rbind(testsubj, trainsubj)  > allact  V1  1 5  2 5  3 5  4 5  5 5  6 5  7 5  8 5  > allsubj  V1  1 2  2 2  3 2  4 2  5 1  6 1  7 1  8 1    # change column names to be understandable  > colnames(allact)<- "act"  > colnames(allsubj) <- "id"  # combine all the three sets of data  > allasg <- cbind (allsubj, allact, alldataasg)  > allasg  id act V1 V2 V3 V4 V5 V6 V41 V42  1 2 5 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 0.9364893 -0.2827192  2 2 5 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 0.9274036 -0.2892151  3 2 5 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 0.9299150 -0.2875128  4 2 5 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 0.9288814 -0.2933958  5 1 5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264 0.9633961 -0.1408397  6 1 5 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220 0.9665611 -0.1415513  7 1 5 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440 0.9668781 -0.1420098  8 1 5 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751 0.9676152 -0.1439765  V43 V44 V45 V46 V121 V122 V123 V124  1 0.11528825 -0.9254273 -0.9370141 -0.5642884 0.119976160 -0.09179234 0.18962854 -0.8830891  2 0.15256831 -0.9890571 -0.9838872 -0.9647811 -0.001552463 -0.18729119 0.18070522 -0.9255665  3 0.14608559 -0.9959365 -0.9882505 -0.9815796 -0.048212671 -0.16627974 0.15417437 -0.9729882  4 0.14292592 -0.9931392 -0.9704192 -0.9915917 -0.056641577 -0.12601846 0.11833718 -0.9677905  5 0.11537494 -0.9852497 -0.9817084 -0.8776250 -0.006100849 -0.03136479 0.10772540 -0.9853103  6 0.10937881 -0.9974113 -0.9894474 -0.9316387 -0.016111620 -0.08389378 0.10058429 -0.9831200  7 0.10188392 -0.9995740 -0.9928658 -0.9929172 -0.031698294 -0.10233542 0.09612688 -0.9762921  8 0.09985014 -0.9966456 -0.9813928 -0.9784764 -0.043409983 -0.09138618 0.08553770 -0.9913848  V125 V126  1 -0.8161636 -0.9408812  2 -0.9295992 -0.9675810  3 -0.9785106 -0.9756483  4 -0.9751338 -0.9632327  5 -0.9766234 -0.9922053  6 -0.9890458 -0.9891212  7 -0.9935518 -0.9863787  8 -0.9924073 -0.9875542  # Note the subject IDs (my.id) are different between test and train sets (sanity check).  # However, by coincidence, the activity label is the same for both (5 = “STANDING”).  # The latter makes sense – the subjects need to be standing before they can start any of the other activities.  # Now, work on finding averages for various combinations of subject id and type of activity  # Create another column, to separate the rows by.   |  | | --- | | > allasg$id\_act = paste (allasg$id, allasg$act, sep = "\_")  > allasg  id act V1 V2 V3 V4 V5 V6 V41 V42  1 2 5 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 0.9364893 -0.2827192  2 2 5 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 0.9274036 -0.2892151  3 2 5 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 0.9299150 -0.2875128  4 2 5 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 0.9288814 -0.2933958  5 1 5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264 0.9633961 -0.1408397  6 1 5 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220 0.9665611 -0.1415513  7 1 5 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440 0.9668781 -0.1420098  8 1 5 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751 0.9676152 -0.1439765  V43 V44 V45 V46 V121 V122 V123 V124  1 0.11528825 -0.9254273 -0.9370141 -0.5642884 0.119976160 -0.09179234 0.18962854 -0.8830891  2 0.15256831 -0.9890571 -0.9838872 -0.9647811 -0.001552463 -0.18729119 0.18070522 -0.9255665  3 0.14608559 -0.9959365 -0.9882505 -0.9815796 -0.048212671 -0.16627974 0.15417437 -0.9729882  4 0.14292592 -0.9931392 -0.9704192 -0.9915917 -0.056641577 -0.12601846 0.11833718 -0.9677905  5 0.11537494 -0.9852497 -0.9817084 -0.8776250 -0.006100849 -0.03136479 0.10772540 -0.9853103  6 0.10937881 -0.9974113 -0.9894474 -0.9316387 -0.016111620 -0.08389378 0.10058429 -0.9831200  7 0.10188392 -0.9995740 -0.9928658 -0.9929172 -0.031698294 -0.10233542 0.09612688 -0.9762921  8 0.09985014 -0.9966456 -0.9813928 -0.9784764 -0.043409983 -0.09138618 0.08553770 -0.9913848  V125 V126 id\_act  1 -0.8161636 -0.9408812 2\_5  2 -0.9295992 -0.9675810 2\_5  3 -0.9785106 -0.9756483 2\_5  4 -0.9751338 -0.9632327 2\_5  5 -0.9766234 -0.9922053 1\_5  6 -0.9890458 -0.9891212 1\_5  7 -0.9935518 -0.9863787 1\_5  8 -0.9924073 -0.9875542 1\_5 | |  |   > factorc<-factor(allasg$id\_act)  > factorc  [1] 2\_5 2\_5 2\_5 2\_5 1\_5 1\_5 1\_5 1\_5  Levels: 1\_5 2\_5  > library(dplyr)  # loading details of dplyr omitted here   |  | | --- | | > myasg <- tbl\_df(allasg)  > myasg  > myasg <- tbl\_df(allasg)  > myasg  Source: local data frame [8 x 21]  id act V1 V2 V3 V4 V5 V6 V41 V42  1 2 5 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 0.9364893 -0.2827192  2 2 5 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 0.9274036 -0.2892151  3 2 5 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 0.9299150 -0.2875128  4 2 5 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 0.9288814 -0.2933958  5 1 5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264 0.9633961 -0.1408397  6 1 5 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220 0.9665611 -0.1415513  7 1 5 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440 0.9668781 -0.1420098  8 1 5 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751 0.9676152 -0.1439765  Variables not shown: V43 (dbl), V44 (dbl), V45 (dbl), V46 (dbl), V121 (dbl), V122 (dbl), V123 (dbl),  V124 (dbl), V125 (dbl), V126 (dbl), id\_act (chr)  > myasg$id\_act  [1] "2\_5" "2\_5" "2\_5" "2\_5" "1\_5" "1\_5" "1\_5" "1\_5" | | #use grouping and averaging (with ‘mean’) in side of dplyr. Note the column names do not have to be preceded by df name and $ | | |  | | --- | |  | |   > by\_id\_act <- group\_by (myasg, id\_act)  > by\_id\_act  Source: local data frame [8 x 21]  Groups: id\_act  id act V1 V2 V3 V4 V5 V6 V41 V42  1 2 5 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833 0.9364893 -0.2827192  2 2 5 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582 0.9274036 -0.2892151  3 2 5 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480 0.9299150 -0.2875128  4 2 5 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907 0.9288814 -0.2933958  5 1 5 0.2885845 -0.02029417 -0.13290514 -0.9952786 -0.9831106 -0.9135264 0.9633961 -0.1408397  6 1 5 0.2784188 -0.01641057 -0.12352019 -0.9982453 -0.9753002 -0.9603220 0.9665611 -0.1415513  7 1 5 0.2796531 -0.01946716 -0.11346169 -0.9953796 -0.9671870 -0.9789440 0.9668781 -0.1420098  8 1 5 0.2791739 -0.02620065 -0.12328257 -0.9960915 -0.9834027 -0.9906751 0.9676152 -0.1439765  Variables not shown: V43 (dbl), V44 (dbl), V45 (dbl), V46 (dbl), V121 (dbl), V122 (dbl), V123 (dbl),  V124 (dbl), V125 (dbl), V126 (dbl), id\_act (chr)  # add meaningful column names also. ‘a’ is used to imply body acceleration; ‘g’ implies gravitational acceleration; and ‘r’ refers to rotational or gyroscope data  #’av’ refers to average; ‘sd’ refers to standard deviation; and x, y, and z refer to the three dimensions.  > myavg <- summarize(by\_id\_act, id = mean(id), act = mean(act), ax.av= mean(V1), ay.av = mean(V2), az.av=mean(V3), ax.sd =mean(V4), ay.sd=mean(V5), az.sd = mean(V6), gx.av= mean(V41), gy.av = mean(V42), gz.av=mean(V43), gx.sd =mean(V44), gy.sd=mean(V45), gz.sd = mean(V46),rx.av= mean(V121), ry.av = mean(V122), rz.av=mean(V123), rx.sd =mean(V124), ry.sd=mean(V125), rz.sd = mean(V126) )  > myavg  Source: local data frame [2 x 21]  id\_act id act ax.av ay.av az.av ax.sd ay.sd az.sd gx.av  1 1\_5 1 5 0.2814576 -0.02059314 -0.12329240 -0.9962487 -0.9772501 -0.9608669 0.9661126  2 2\_5 2 5 0.2722469 -0.02377822 -0.09235203 -0.9755951 -0.9576855 -0.8856200 0.9306723  Variables not shown: gy.av (dbl), gz.av (dbl), gx.sd (dbl), gy.sd (dbl), gz.sd (dbl), rx.av (dbl),  ry.av (dbl), rz.av (dbl), rx.sd (dbl), ry.sd (dbl), rz.sd (dbl)  # substitute activity with a character string that is descriptive  # such 5 more gsubs will have to be included for the complete database  > myavg$act  [1] 5 5  > myavg [, 3]<- gsub ("5", "STAND", myavg$act)  > myavg  Source: local data frame [2 x 21]  id\_act id act ax.av ay.av az.av ax.sd ay.sd az.sd gx.av  1 1\_5 1 STAND 0.2814576 -0.02059314 -0.12329240 -0.9962487 -0.9772501 -0.9608669 0.9661126  2 2\_5 2 STAND 0.2722469 -0.02377822 -0.09235203 -0.9755951 -0.9576855 -0.8856200 0.9306723  Variables not shown: gy.av (dbl), gz.av (dbl), gx.sd (dbl), gy.sd (dbl), gz.sd (dbl), rx.av (dbl),  ry.av (dbl), rz.av (dbl), rx.sd (dbl), ry.sd (dbl), rz.sd (dbl)  #remove the act\_id. It is no longer needed  > myavg <- myavg[, -1]  > myavg  Source: local data frame [2 x 20]  id act ax.av ay.av az.av ax.sd ay.sd az.sd gx.av gy.av  1 1 STAND 0.2814576 -0.02059314 -0.12329240 -0.9962487 -0.9772501 -0.9608669 0.9661126 -0.1420943  2 2 STAND 0.2722469 -0.02377822 -0.09235203 -0.9755951 -0.9576855 -0.8856200 0.9306723 -0.2882107  Variables not shown: gz.av (dbl), gx.sd (dbl), gy.sd (dbl), gz.sd (dbl), rx.av (dbl), ry.av (dbl),  rz.av (dbl), rx.sd (dbl), ry.sd (dbl), rz.sd (dbl)  # save as a text file  > fname <- "C:/Users/……./CleanData/Project/getdata\_projectfiles\_UCI HAR Dataset/myavg.txt"  > write.table (myavg, sep=" \t", file= fname, row.names=FALSE, col.names=TRUE)  # For the assigned database, there would be 180 rows( 30 subject ids x 6 types of activities) with 20 columns as given here. This comes to 3200 data entries.  # There are other ways of organizing it too – depending upon one’s interpretation of one variable/column. This could also mean acceleration and sd for acceleration. If #so, there would be an additional column to identify the type of acceleration/sd – 3 (3 dimensions) x 3 types (body, gravitational, and gyroscope). The number of rows will increase by the number of possibilities here – that is 9. Thus, the number of rows will increase to 180 x 9 = 1620, with columns for id, act, acc, sd, and type of acc/sd, that is 5. This leads to 1620 x 5 = 8100 data entries, with a bit of redundancy. There is an intermediate soluation too.     |  | | --- | |  | |