Getting and Cleaning Data

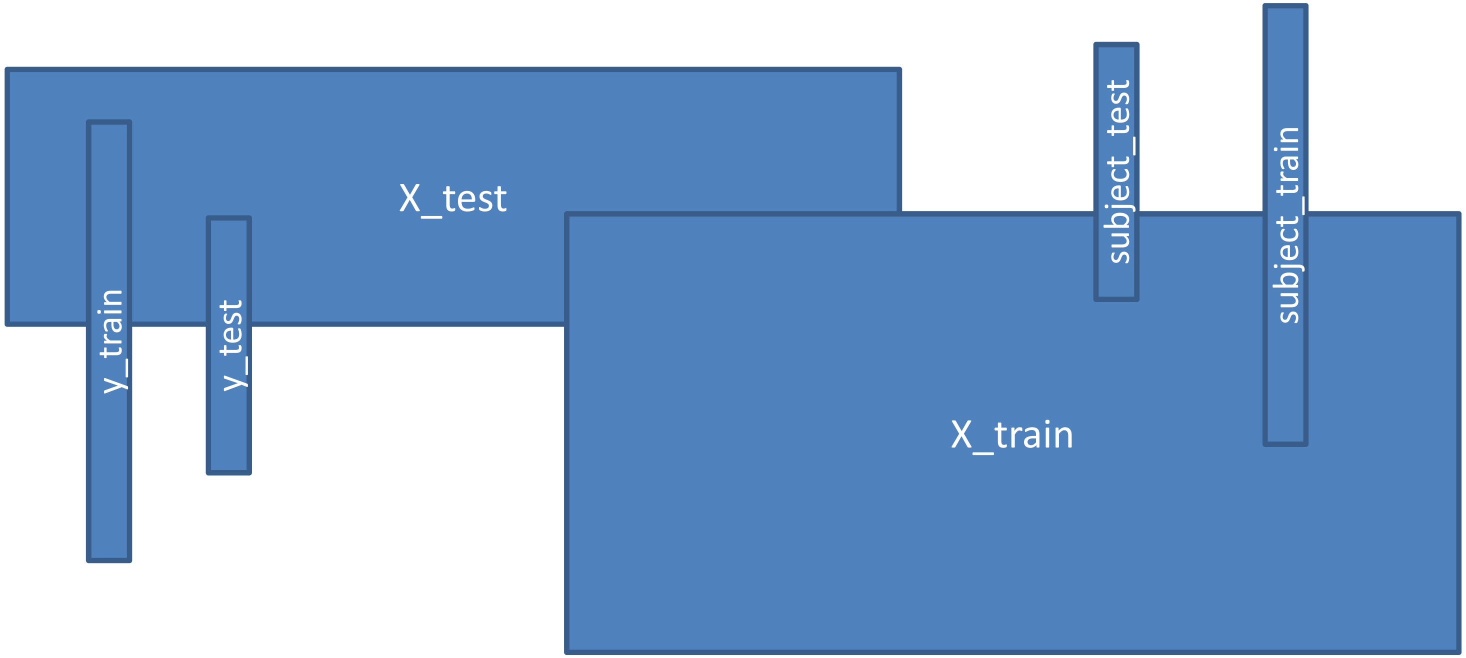
Course project help guide

0. Read data

- Always, open first a data .txt file in a notepad application to have an idea how the data was stored and what function to use to read the data
- You will see the data for this project was stored as a table of numbers separated with single spaces. There are several functions to read data, you can use whatever you like.
- **Tip:** In this case, the best function to use is ***read.table***

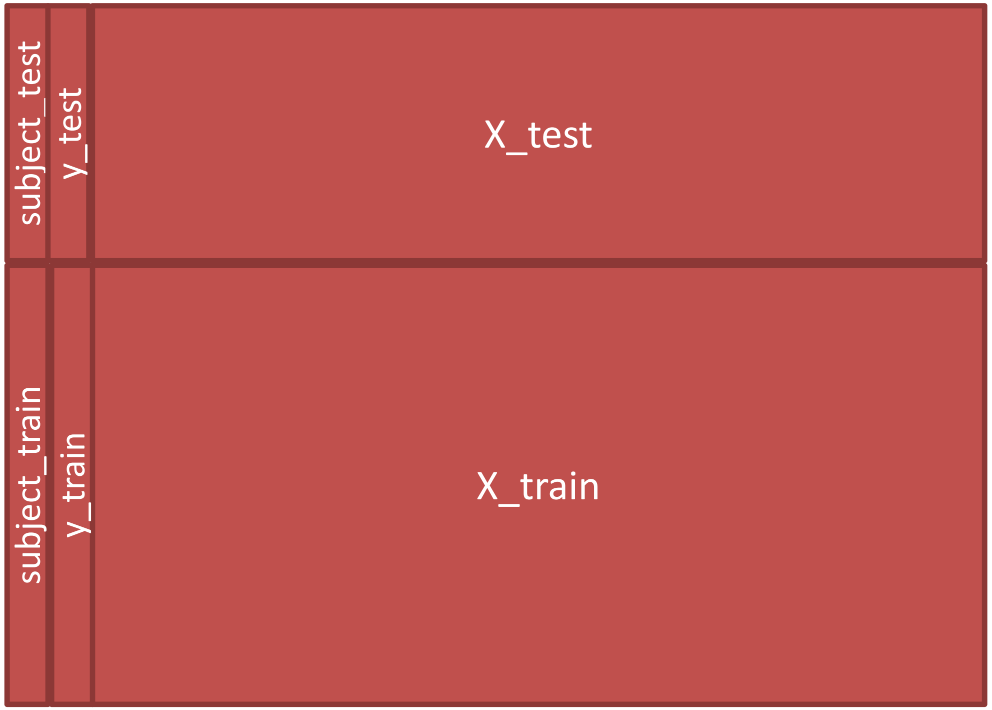
1. Merge the training and the test sets to create one data set

- Always, use functions like *str* to inspect the size and class of the data you've read
- After reading the data files you should end with some objects like this:



1. Merge the training and the test sets to create one data set

- Looking the size of the objects, you should have probably realized what they are asking for:



subject_test

y_test

X_test

subject_train

y_train

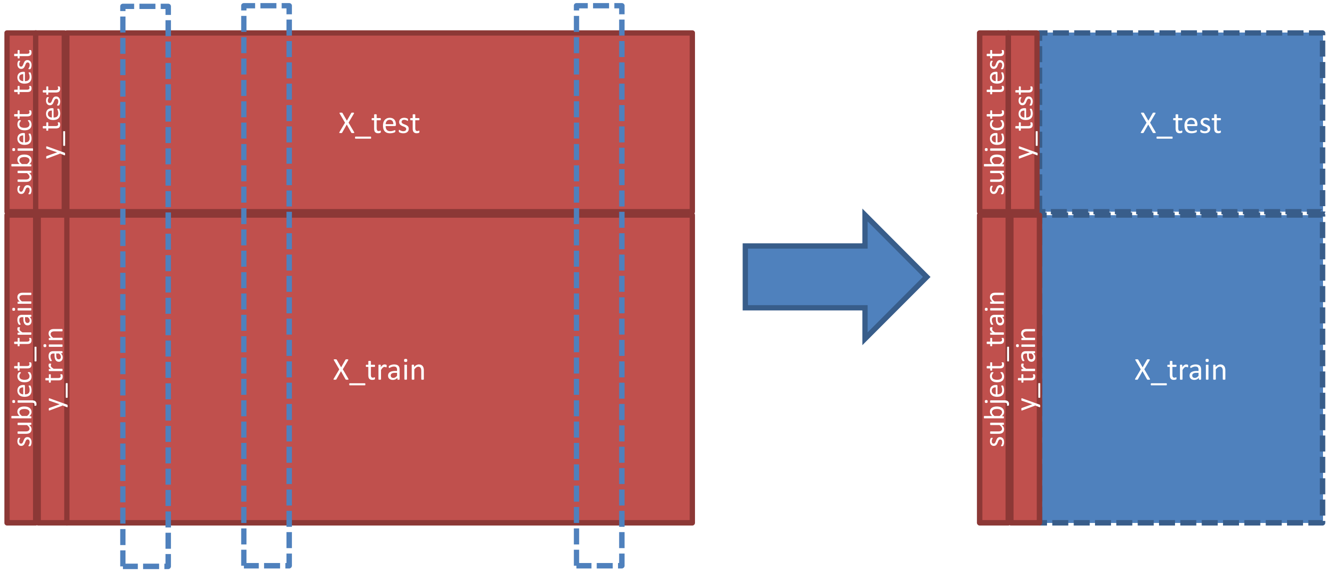
X_train

2. Extract only the measurements on the mean and standard deviation for each measurement

- This instruction can be a little confusing. Do you remember the tidy data principle? Observations (measurements) are rows and Variables measured are columns. You should have noticed that the columns in your new data set are named V1,…Vn. So what do you exactly have to extract ?
- ***"for each measurement"*** here means for each row, and "***the measurements on the mean and standard deviation"*** is telling you which columns to look for
- **Tip:** look at the file named ***features.txt**.* Here is a list with the variable names (columns) used originally in *X_train* and *X_test* tables. Remember you have added two columns (subject ids and activity names) to your new data set!

2. Extract only the measurements on the mean and standard deviation for each measurement

- You should have noticed they are asking for columns with the words ***mean*** and ***std***
- You will end with a different number of columns depending if you consider all variable names with ***mean*** and ***std*** or only those with ***mean()*** and ***std()***. There is no right or wrong answer here, it is open to your interpretation!



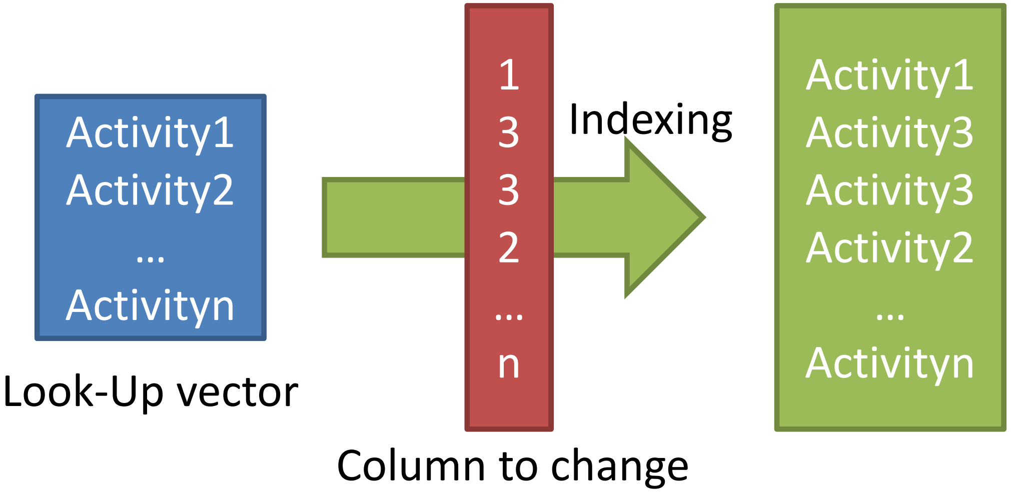
3. Uses descriptive activity names to name the activities in the data set

- Now you have to change activity IDs in the second column with activity names. Look at the file named ***activity_labels.txt****.* Here is a look-up table which links IDs with unique activity names.
- There are several ways to do this. You can use for loops, the sapply function, etc. However…
- *A good magician never reveals his secrets*… but a good programmer does!!!

3. Uses descriptive activity names to name the activities in the data set

- Here is a little trick for free: index the look-up vector (vector with activity names ordered according to IDs) with the column containing the IDs you want to change and, Voila! The magic of R:

  Column.with.names <- lookUpVect[Column.with.numbers]



4. Appropriately label the data set with descriptive variable names

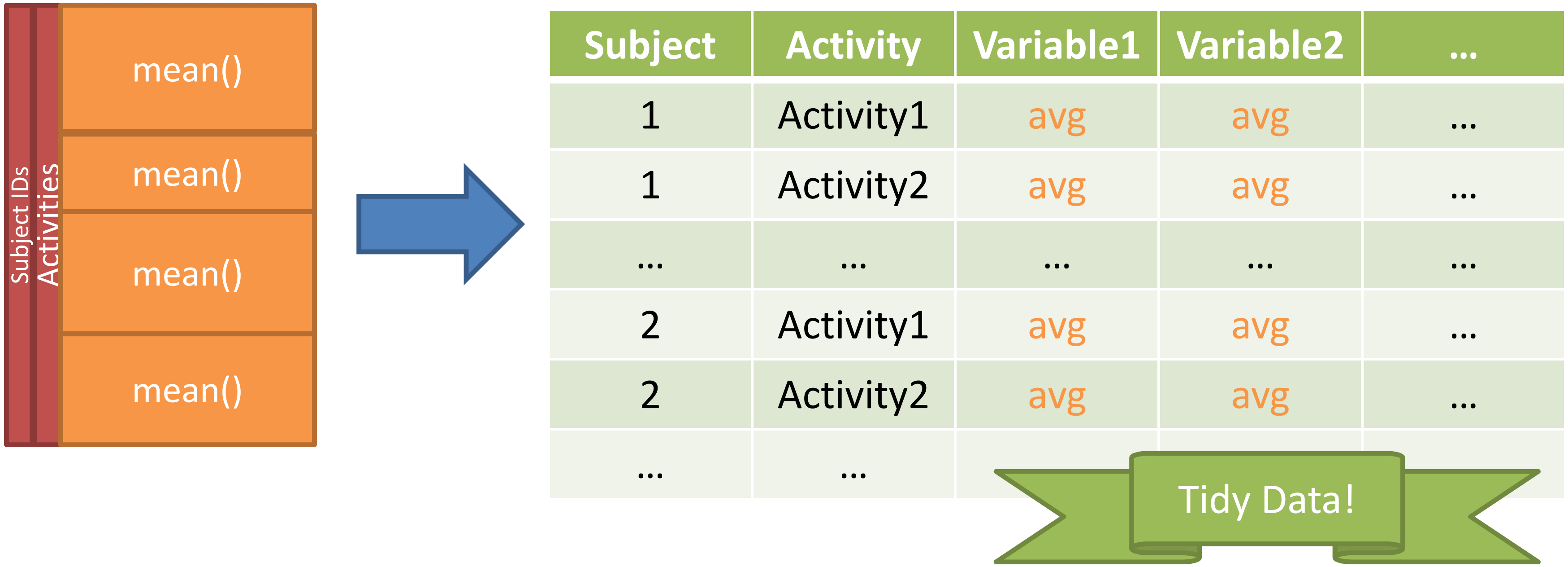
- Do you remember columns were named V1,...Vn? Now is time to change names using the previous variable names list
- You may want to change a little the existing names using the *sub* function. Keep in mind It is a good practice to use only letters, numbers and points (.). Avoid especial characters like - , _ , (,), etc…
- To be more descriptive, you can change "t" to "time" or "f" to "freq", for example.
- **IMPORTANT**: Don't use a new row with text strings to name the variables!!! Use *colnames* function instead or you will change the columns class from *numeric* to *factor* and your variables will still with V1,…,Vn names

5. From the data set in step 4, create a second, independent tidy data set with the average of each variable for each activity and each subject

- Finally, you should have realized for each variable (column) in your new data set, there are several observations for the same subject-activity pair.
- Remember the tidy data principle! One observation per row and one variable per column

5. From the data set in step 4, create a second, independent tidy data set with the average of each variable for each activity and each subject

- **Tip:** You can group your data by Subjects and then by Activities and then use the *summarize_each* function to average the columns by group to end with only one observation (the mean in this case) for each subject-activity pair (30 subjects * 6 activities = 180 observations in total)



Subject IDs

Activities

mean()

mean()

mean()

mean()

Subject	Activity	Variable1	Variable2	…
1	Activity1	avg	avg	…
1	Activity2	avg	avg	…
…	…	…	…	…
2	Activity1	avg	avg	…
2	Activity2	avg	avg	…
…	…			

Tidy Data!