

**Introductory Neuroscience and Neuro-Instrumentation**  
**Professor Rathin Joshi**  
**Indian Institute of Science, Bengaluru**  
**Lecture 46**

**Basics of BCI Experimentation: Signal Acquisition using MATLAB (EEGLAB)**

Hello everyone. Welcome to the final sub module of brainwave acquisition and recording system demonstration. So, in this module mainly we will talk about how MATLAB can be useful for EEG signal analysis or EEG signal processing. So, initially I will tell you one particular, I will give you an introduction about a tool box named EEGLAB, how it is being useful and what are the functions which you can do for different recorded EEG files.

Then, we will see that this EEG file is recorded in different formats across different system. So, we will see a couple of examples of how different files can be imported or loaded in MATLAB using scripting and as well as using a GUI of EEG lab. So, let us talk about EEGLAB a bit before I will demonstrate some of the very basic things about EEG lab.

(Refer Slide Time: 01:38)

### Basics of EEGLAB

- Pre-requisite for EEG/ERP Signal Processing
- Supports both: GUI and Scripting
- Scripting is advisable for batch file processing.
- Download Latest Version of EEGLAB and practice for sample dataset.  
(<https://sccn.ucsd.edu/eeglab/download.php>)
- Always keep yourself updated with different plugins.  
([https://sccn.ucsd.edu/eeglab/plugin\\_uploader/plugin\\_list\\_all.php](https://sccn.ucsd.edu/eeglab/plugin_uploader/plugin_list_all.php))

Ref:  
<https://sccn.ucsd.edu/eeglab/index.php>



So, EEGLAB is a prerequisite for any kind of EEG and ERP processing using MATLAB. There are some packages and functions which have been already made, so you can directly use that function and you can do your analysis, otherwise, it will take a lot of time again, to generate the same functions. As always, you can make use of different combinations of function and create a plugin for specific applications, but all the functions, documented functions or the included function in EEGLAB makes EEG signal processing very, very simpler. So, you can have a look.

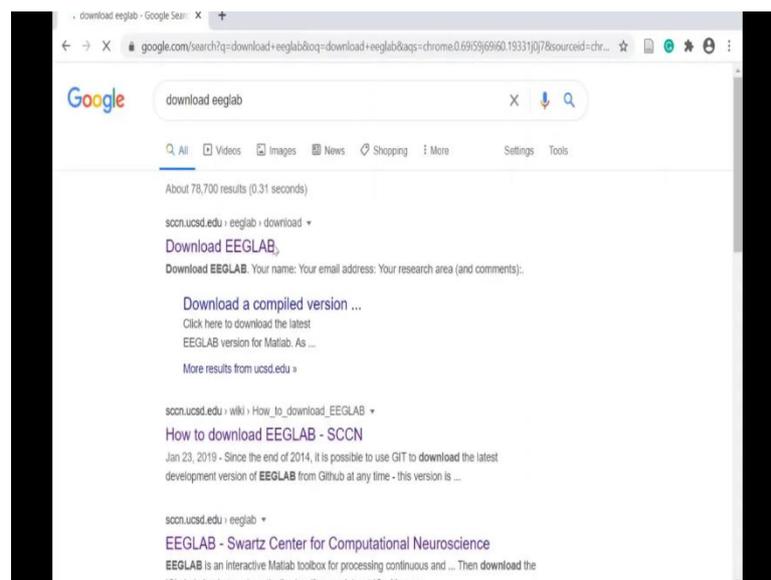
I will show you when I will demonstrate EEGLAB basic functions and plugins and all the things I will try to include. So, it is supposed both. Like in PsychoPy, we have seen that we were generated stimuli using GUI because at that time, it was like you have to generate stimuli mostly once for experimental twice.

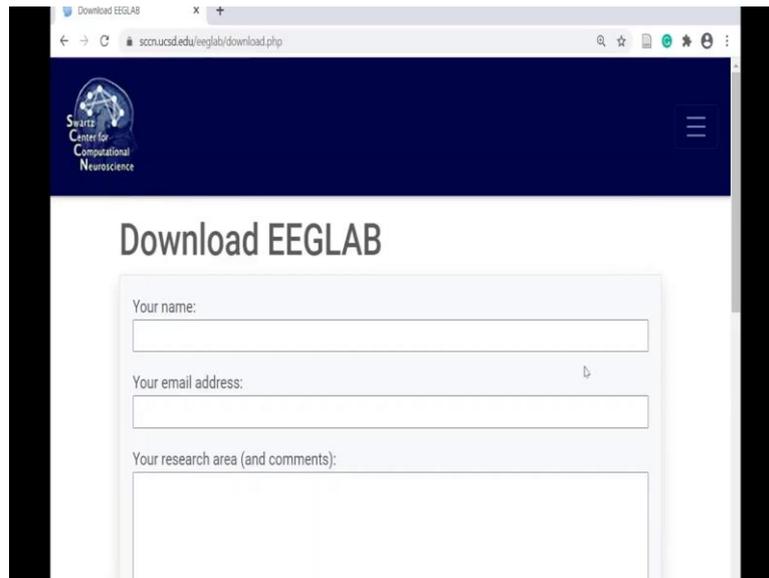
But whereas in this case, when you record across multiple subjects, instead of GUI a scripting is advisable because you do not have to repeat the same procedure again and again, you can act automate this stuff using scripting. So it supports both basically. How you can generate a particular script for GUI operation which you have made?

You know, and one particular scenario where the GUI is better or scripting, all this things I will try to cover. Further, we will see that scripting as I mentioned, it is advisable, one last important thing is plugins. So, in the demonstration, I will show you that there are many plugins which are already available with thousands of downloads and various applications, many of them are still in progress and getting updated with more and more features.

So first, we will see the demonstration or introduction to EEGLAB. I have included from scratch. So how from downloading EEGLAB to what are the basic things you can do with a sample data set. All this thing I have tried to cover so let us see first and let us see how EEGLAB works or basic introduction of EEGLAB.

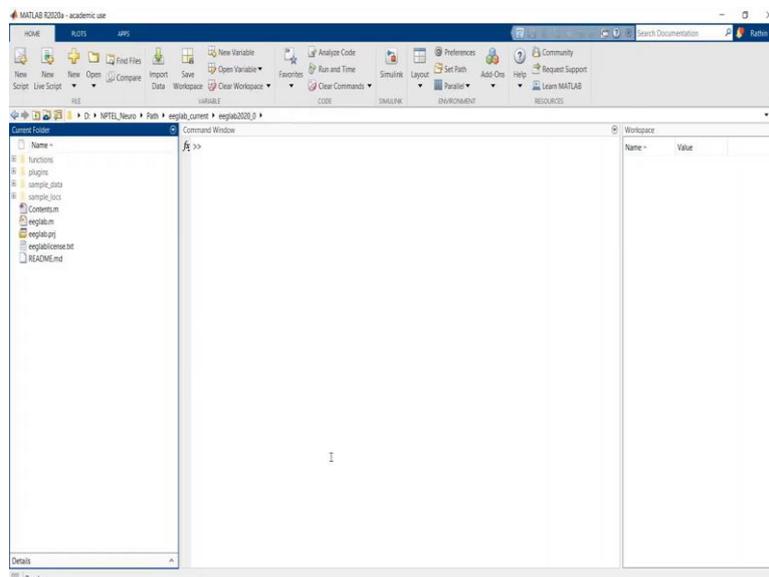
(Refer Slide Time: 04:14)





So, let us quickly see how to download EEGLAB. You can search it and it will redirect to Swartz Center for Computational Neuroscience site. Here you can add your basic details, your email id, your name, and your area of research and submit it. You should be able to download the latest version of EEGLAB.

(Refer Slide Time: 04:40)







neuro-scan is required to import some file which has been recorded in tune called Euroscep. So, all these different plugins and all are there.

(Refer Slide Time: 06:11)

The screenshot shows a web browser window displaying the 'List of plug-ins available for download in EEGLAB 2019.1 and later versions'. The page includes a title, an introductory paragraph, and a table of plugins. The table has columns for Plug-in name, Version, Plug-in description, Link, Tags, Contact, Ratings, and Number of downloads.

**List of plug-ins available for download in EEGLAB 2019.1 and later versions**

You may download the .zip file associated with the plug-ins below and unzip it in the "plugins" subfolder of your EEGLAB distribution. However, this list is primarily provided for informational purposes. To install plug-ins of interest, we recommend that you use the EEGLAB extension manager (in the EEGLAB GUI window, select menu item "File > Manage plug-ins"). Note: it is not possible to install new plug-ins in the EEGLAB compiled version. However, you may use a plug-in when running the compiled version by including the plug-in code base in the EEGLAB script you run.

Plug-in name	Version	Plug-in description	Link	Tags	Contact	Ratings	Number of downloads
AAR	131130	ICA-based Automatic Artifact Removal. Note that this extension overloads the "pca.m" Matlab	Download	artifact,ica	G. Gomez-Herrero	★★★★★	2886
NFT	2.2	EEG inverse problem solution	Download	source	Zeynep Akalin	☆☆☆☆	395
NihonKoden	1.12	Import Nihon Koden .m00 file. Now it supports data that has 'Trigger' in addition to EEG data (Thanks Elham Sherkat for sharing the test file.)	Download	import,	Makoto Miyakoshi	★★★★★	1034
NIMA	0.22	A plug-in for using Nima's Images from Measure-projection Analysis	Download	study	N. Bigdely	☆☆☆☆	126
nsgportal	1.0	Plug-in interfacing EEGLAB with NSG. The plug-in allows users to directly submit, manage and retrieve jobs running on the U.S XSEDE network of high-performance computing resources from within an	Download	erp, ica, source, time-freq, study, nsg	Ramon Martinez Cancino	☆☆☆☆	66

Use biutils to import BIOPAC .ACQ files.

Name	Version	Description	Download	Category	Author	Rating	Downloads
Biosig	3.3.0	Import multiple data files formats. EDF, EDF+, BDF, GDF, Neuroscan, CNT, BRK, DAT, RDF, HDF, RAW, TDMS, Biopac ACQ, BCI2000, EEProbe, etc...	Download	export,import	A. Schloegl	★★★★☆	40079
bva-io	1.6	Import Brain Vision Analyser data files	Download	export,import		★★★★☆	11331
CIAC	1.00	Cochlear Implant Artifact Correction	Download	artifact	S. Debener	☆☆☆☆☆	346
Cleanline	1.04	Removes sinusoidal artifacts (line noise)	Download	artifact	T. Mullen	★★★★★	5791
clean_rawdata	2.2	Cleans raw EEG data. Contains Artifact Subspace Reconstruction (ASR). Adopted from BCLILAB for offline	Download	artifact,		★★★★★	6290

Name	Version	Description	Download	Category	Author	Rating	Downloads
EMDLAB	0.1	Perform four types of EMD: plain EMD, ensemble EMD (EEMD), weighted sliding EMD (wSEMD) and multivariate EMD (MEMD) on EEG data	Download	other	Saad Al-Baddai	☆☆☆☆☆	387
envtopoForContinuous	0.10	Compute envelopes of scalp projections. Continuous data only.	Download	other	M.Miyakoshi	☆☆☆☆☆	275
ERPLAB	8.02	ERPLAB Toolbox is a free, open-source Matlab package for analyzing ERP data. It is tightly integrated with EEGLAB Toolbox, extending EEGLAB's capabilities to provide robust, industrial-strength tools for ERP processing, visualization, and	Download	erp, artifact,	Kayla Beavers	★★★★☆	2747

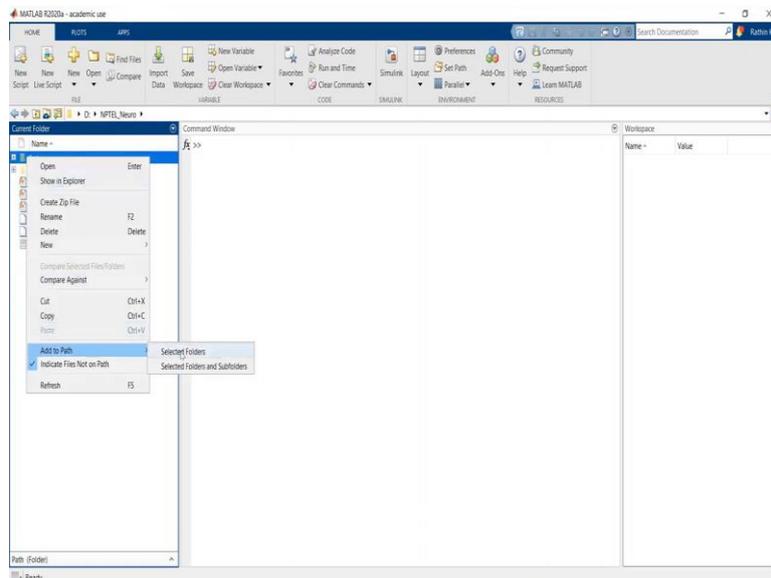
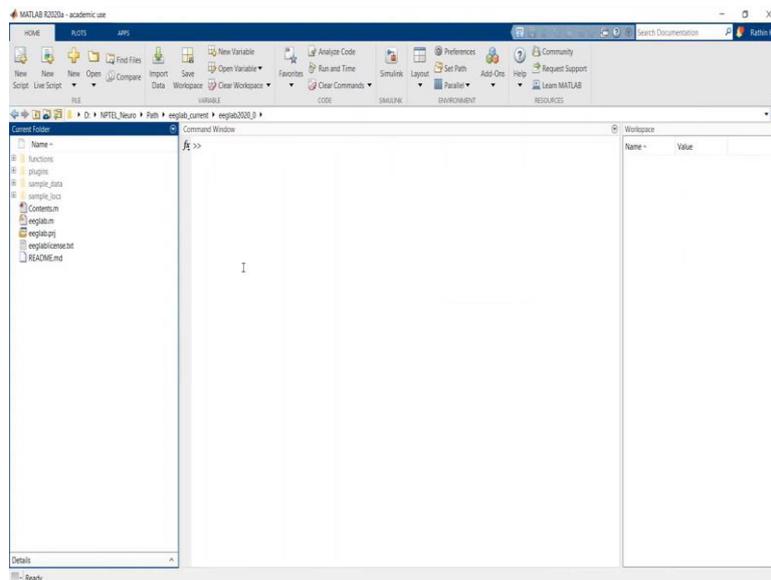
Name	Version	Description	Download	Category	Author	Rating	Downloads
fullRankAveRef	0.10	Apply average reference after adding back the original reference channel	Download	preprocessing	M. Miyakoshi	☆☆☆☆☆	327
get_chanlocs	2.32	Electrode position localization from 3-D model to EEG chanlocs	Download		Clement Lee	☆☆☆☆☆	427
GEVD	1.00	Generalized Eigenvalue Decomposition (GEVD)	Download	other	P. Sajda	☆☆☆☆☆	431
gtecimport	3.15.01	Import gRecorder HDF5 files and MATLAB files to EEGLAB including triggers/markers	Download	import	Patrick Reitner	★★★★☆	609
headModel	1.0.0	Encapsulates in a single interface common routines to allow for out-of-the-	Download	ica,source	Alejandro Ojeda	☆☆☆☆☆	330

Plugin Name	Version	Description	Download	Supported	Author	Rating	Downloads
MFFimport	2.3	Import EGI MFF files (legacy - see new MFFMatlabIO plugin)	Download	import	A. Chennu	★★★★★	1859
MFFMatlabIO	3.5	Import and export Philips MFF files (formerly known as EGI MFF data files)	Download	import, export,		★★★★☆	1775
MicrostateAnalysis	0.3	Model and quantify microstates in resting state data	Download	other	T. Koenig	☆☆☆☆☆	591
Mobilab	20200220	Mobilab is a multimodal data browser and motion capture data preprocessor for XDF	Download	import, export, erp, artifact,	Arnaud Delorme	☆☆☆☆☆	58

NFT	2.2	EEG inverse problem solution	Download	source	Zeynep Akalin	☆☆☆☆☆	395
NihonKoden	1.12	Import Nihon Koden .m00 file. Now it supports data that has 'Trigger' in addition to EEG data (Thanks Elham Sherkat for sharing the test file.)	Download	import,	Makoto Miyakoshi	★★★★★	1034
NIMA	0.22	A plug-in for using Nima's Images from Measure-projection Analysis	Download	study	N. Bigdely	☆☆☆☆☆	126
nsgportal	1.0	Plug-in interfacing EEGLAB with NSG. The plug-in allows users to directly submit, manage and retrieve jobs running on the U.S XSEDE network of high-performance computing resources from within an	Download	erp, ica, source, time-freq, study, nsg	Ramon Martinez Cancino	☆☆☆☆☆	66

One interesting thing I would like to recommend all of you is that you can go through the list of plugins of MATLAB. Very important, all these plugins, if you install and if you can study, it will make life easier for EEG signal processing. So, I would request all of you to at least take a look at the plugins which is useful for their analysis or their applications.

(Refer Slide Time: 06:43)



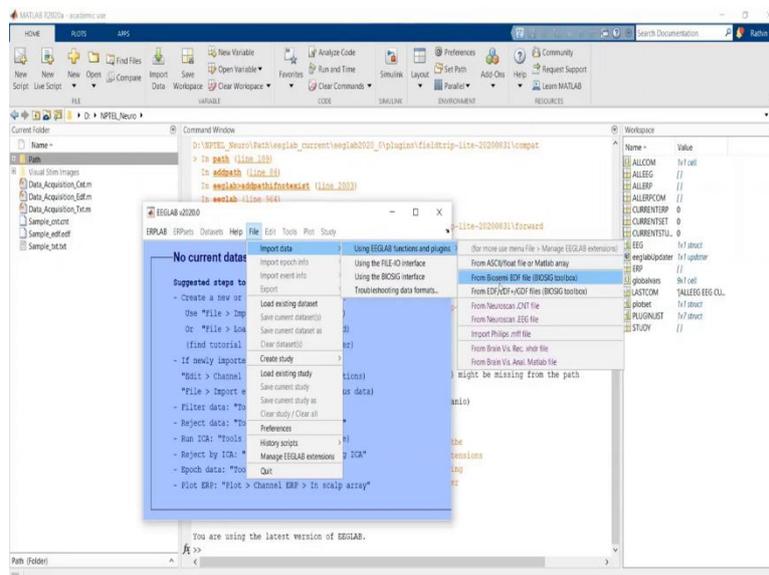


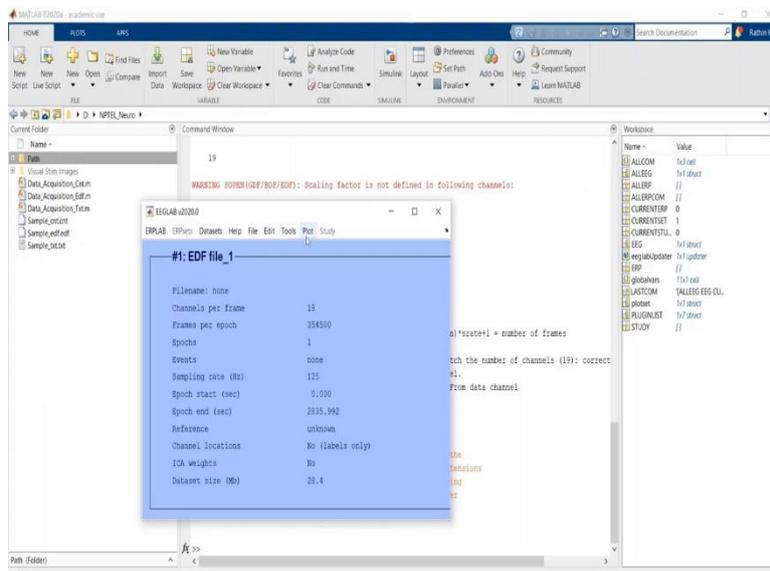
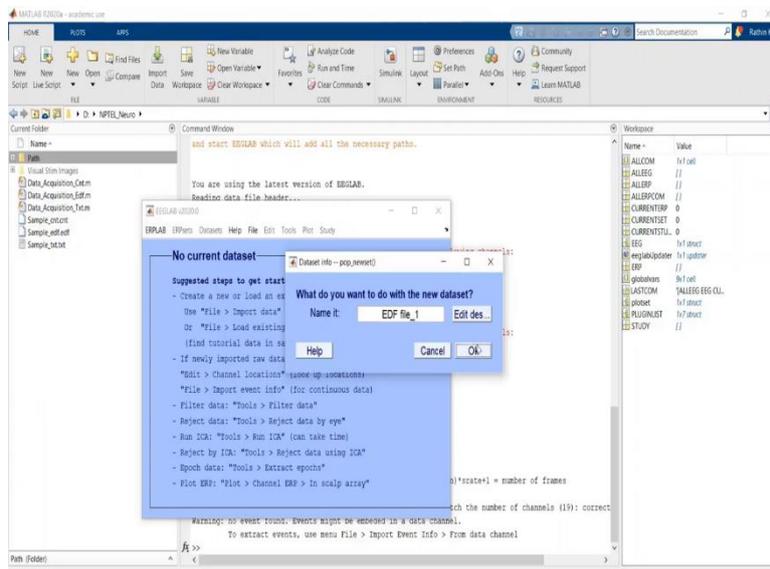
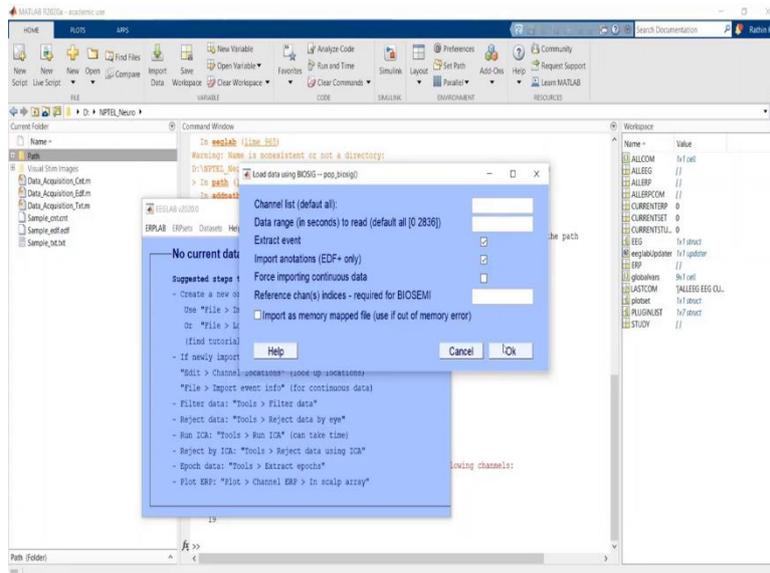
So, coming back to EEGLAB finally. EEGLAB is a toolbox which has been used specifically for EEG processing in MATLAB. So, I have already added this into my paths folder. So I will add this thing into path and see if I can run the EEGLAB. So once the downloaded EEGLAB package is in path, you have to write a command EEGLAB, it should open EEGLAB pursuit.

In EEGLAB you can process to it in two manners, one is using the scripting, like typing a command and get the desired work done, whereas in other way, it is using GUI. So I will show you both the ways. In this newer version as I have downloaded ERPLAB, it is showing ERPLAB, ERP sets everything.

But generally, it should start with, so the acquisition of your input data files generated from particular system should start with file menu. Let us import some of the data. Before importing, I would like to tell, I was telling two things, one is GUI, other one is scripting. So, we will start with GUI. We will go to scripting. We will see how there is a way from GUI to scripting and some tricks.

(Refer Slide Time: 8:20)



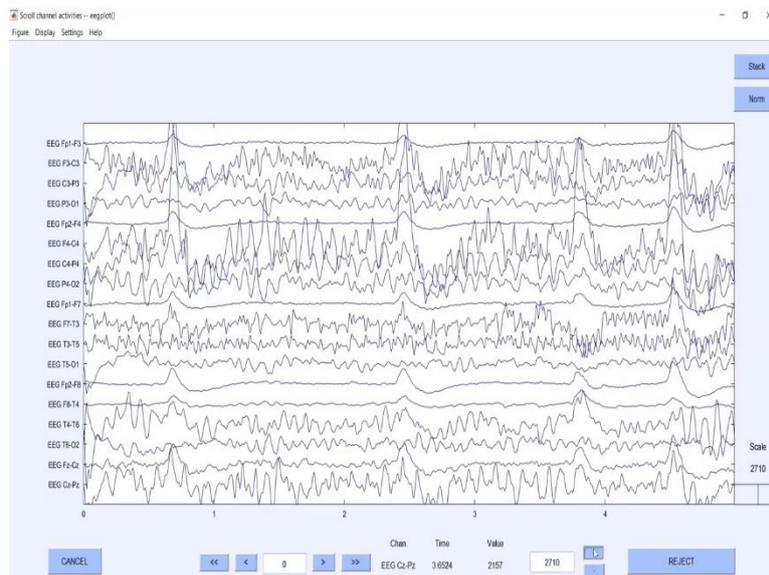


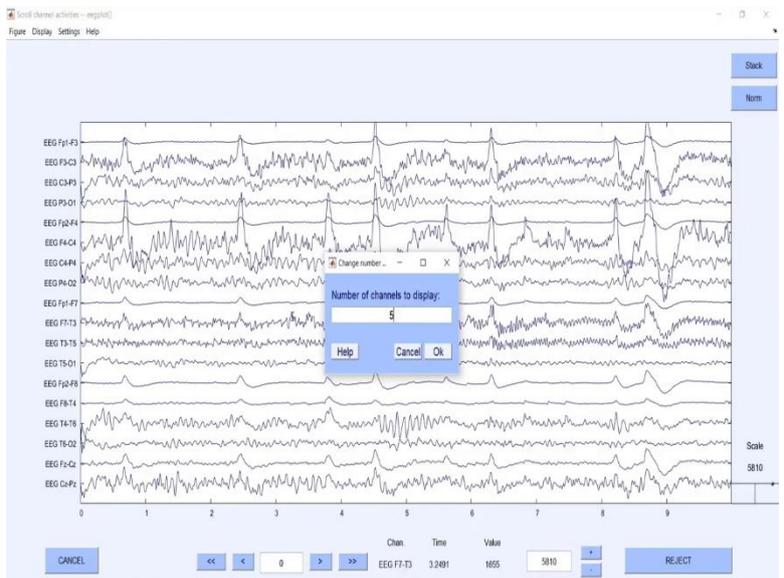
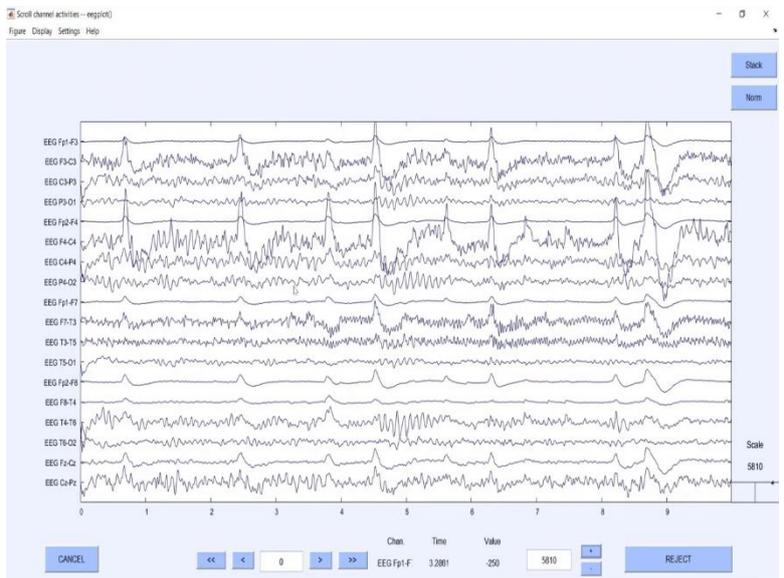
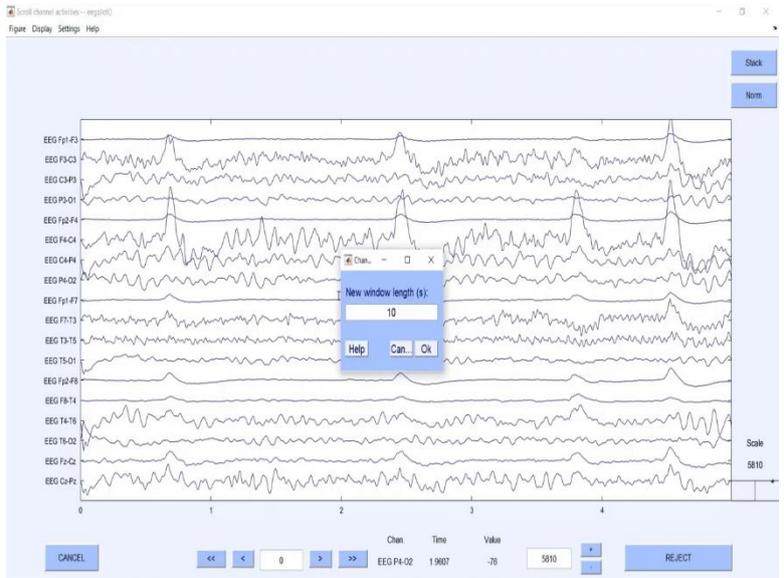
So first, let me import one of the sample data set which I have. Let us import EDF file. So, I will go to this from EDF file. This is a sample EDF file here. So I will view this file as an input, it will ask me a few questions, like if I want to see all the channels, if I want to see the entire data recorded throughout the complete data stream and all. I will for now, simply put, I would like to go ahead with default parameters, so I am not going to change anything.

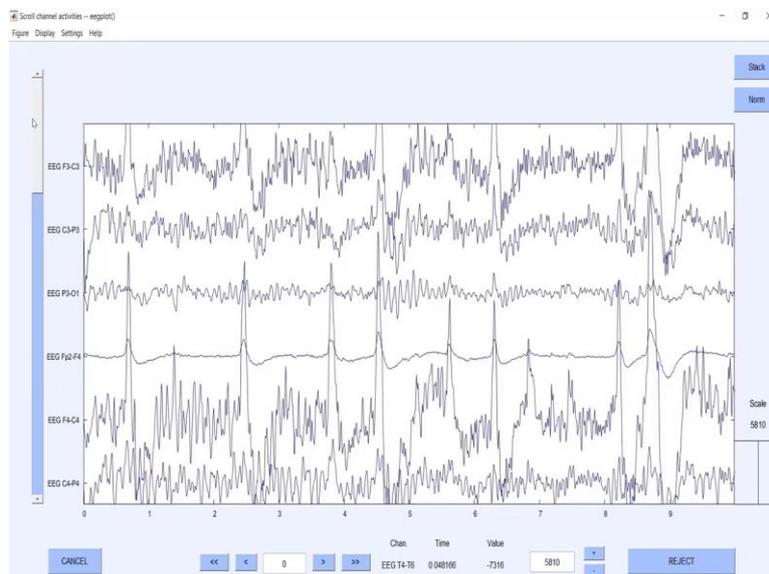
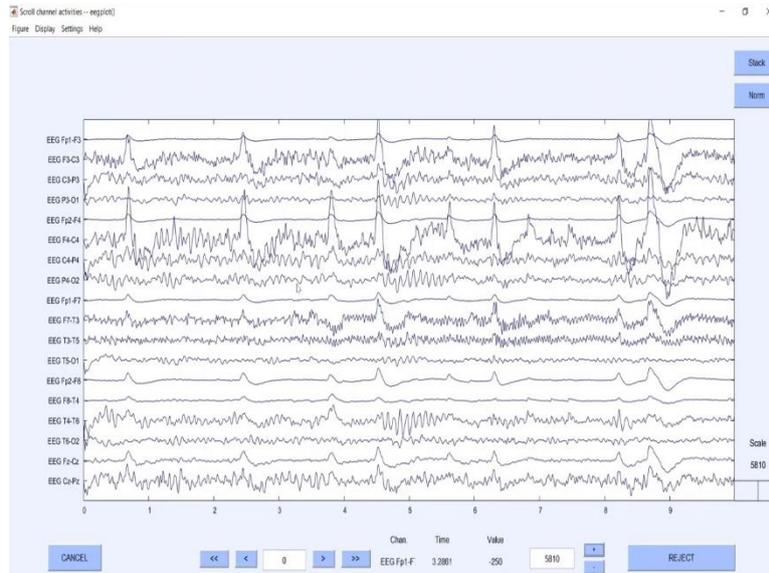
One thing to notice here is the function name is written. Like in GUI, whatever function you are executing is both BIOSIG. Based on my values here, a function with proper attributes and key value pairs can be used. So, for now, I will just simply say okay, further, it will process and it will ask me the name of the file.

I can simplify let us keep it as it is an EDF file. So, that will be the name of my new data set. Here if you can see the data set menu, so this data set will appear there. Or I will just simply write EDF file one and I will say okay. So, this has kind of imported the data on dot EDF file EDF is European Data Format dot EDF file into a MATLAB.

(Refer Slide Time: 10:04)







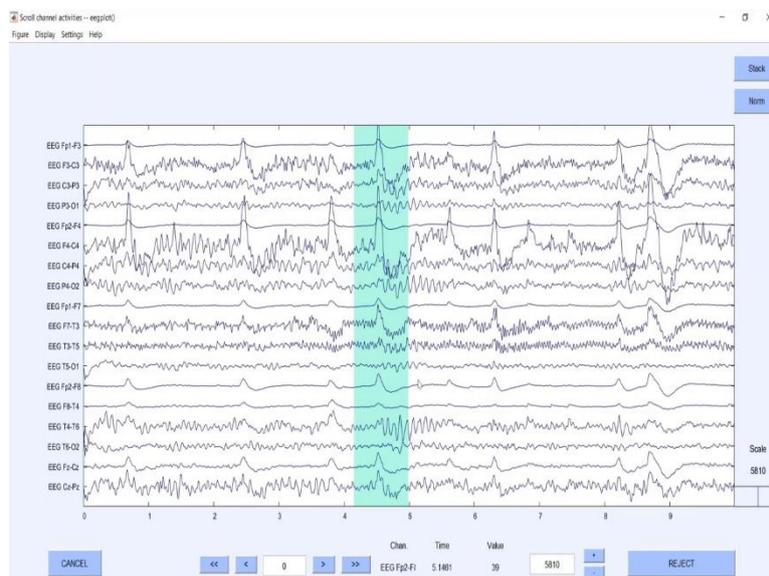
Further, if I want to see the data, I can go to plot and scroll data. So, I am able to see the data, if you can see it properly this all this y axis is different channels. And in the experimentation module I told you about different kinds of montages. So, if you notice it clearly that FP 1 minus F3, F3 minus C3, C3 minus P3, P3 minus O1.

So, this all this bipolar value bipolar channel representation is nothing but your double banana montage. Now, this is your data very difficult to generate an interpretation we can increase or decrease your amplitude scale using this function let us increase it. So, you know where we can see some channelise moment okay. So, now you can see we need activities which is being happening here.

And like we can notice that it is predoom like prominent in F4 and F3. So, something related to font although maybe some moment also but let say let say before we analyse it further I will tell you a few basic things. Using this you can select a particular time range you want to see for one in one display now I have changed it to 10 also you can see change the number of channels or do you want to display one display currently at General is there if I change it to five then I can only see those 5 images.

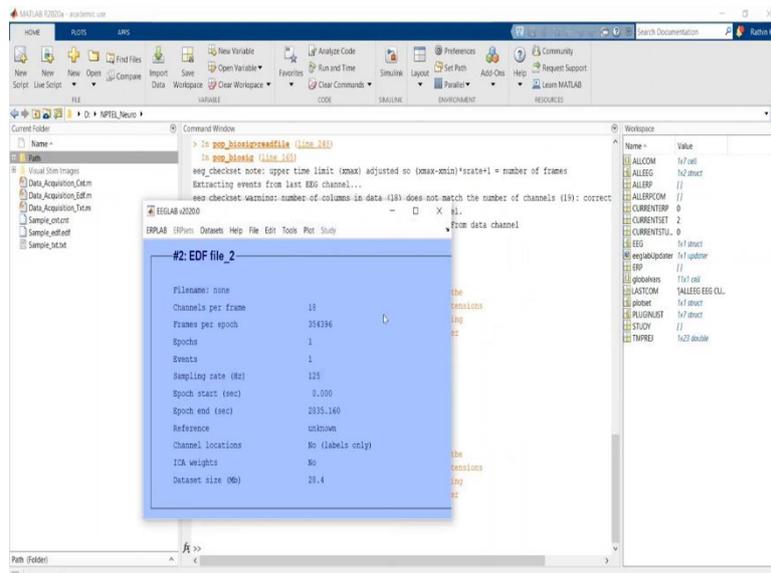
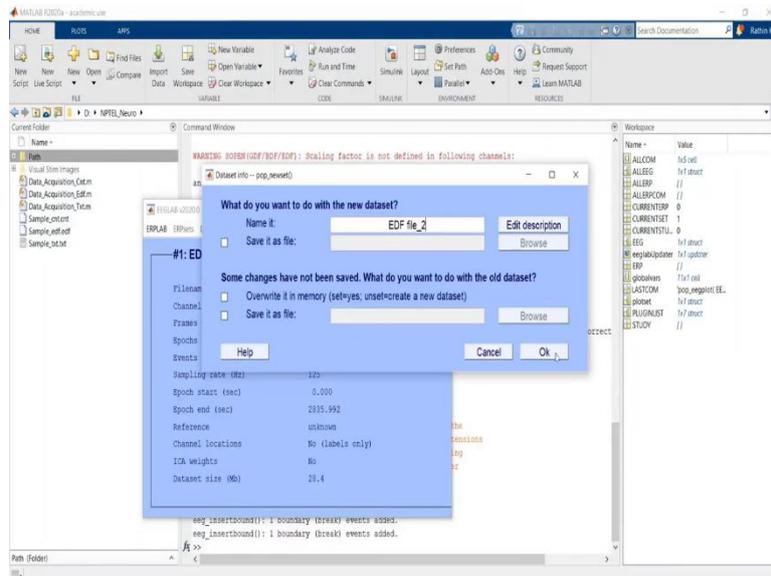
So, based on your analysis based on your criticality of problem, you can analyse the number of channels and you can accordingly adjust the time axis. You can move in the time axis using this push button, this will, second by second you move. If you use this button, it will shift the entire time interval. Currently it is 5 to 15, so it will shift to 15 to 25, it will shift to 25 to 35 and so on. So, now if I talk about Let us say, I will first shift it to original number of channels. Now, I will check. I want to, Let us say I want to, I do not, this portion is artefact, we will move to initial time, let us say this 5 in between 4 to 5 that is an artefact.

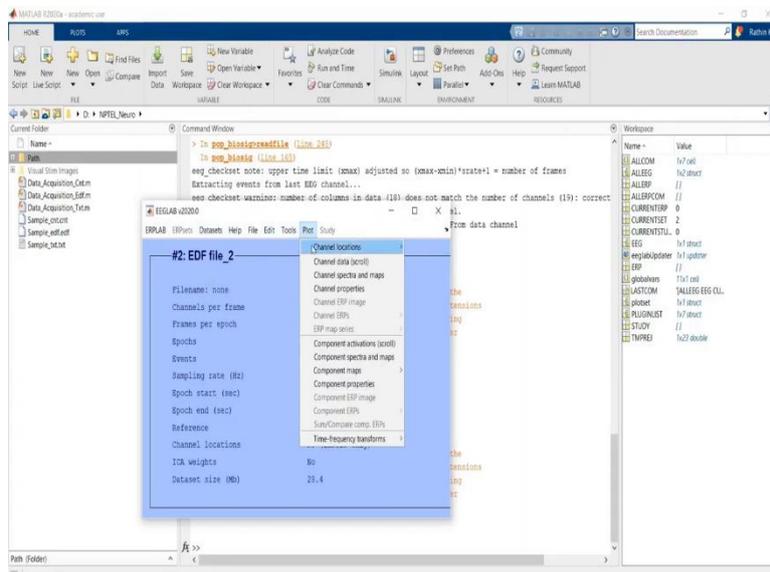
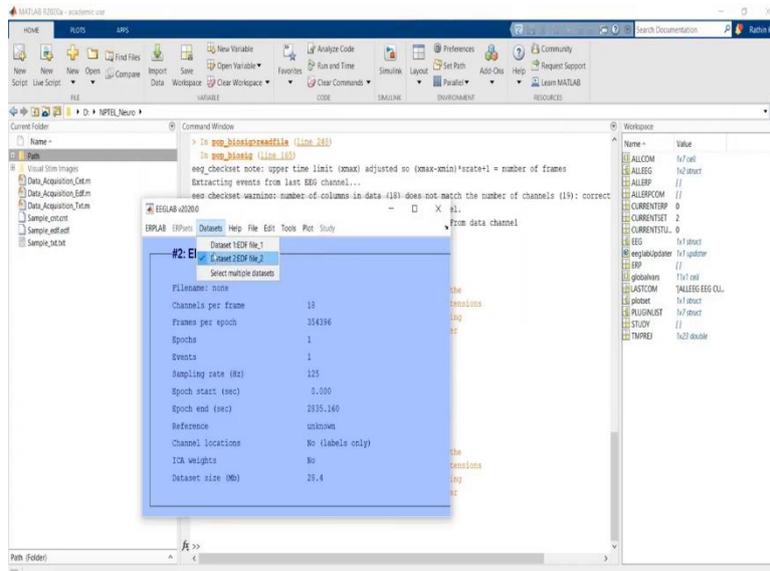
(Refer Slide Time: 12:50)



So, I will select this particular region and I will simply reject it.

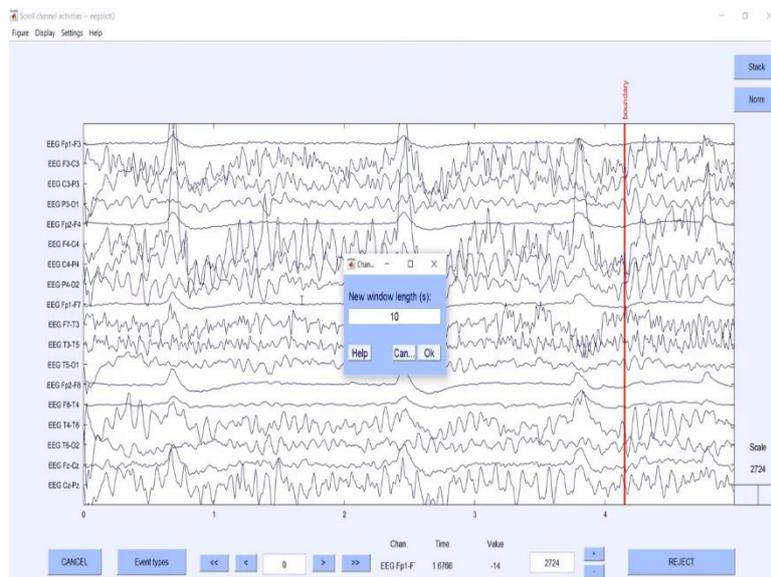
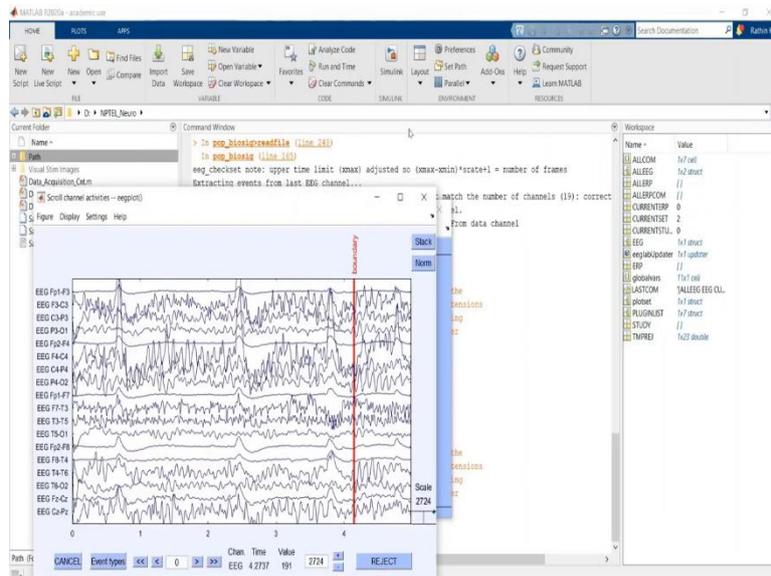
(Refer Slide Time: 12:58)

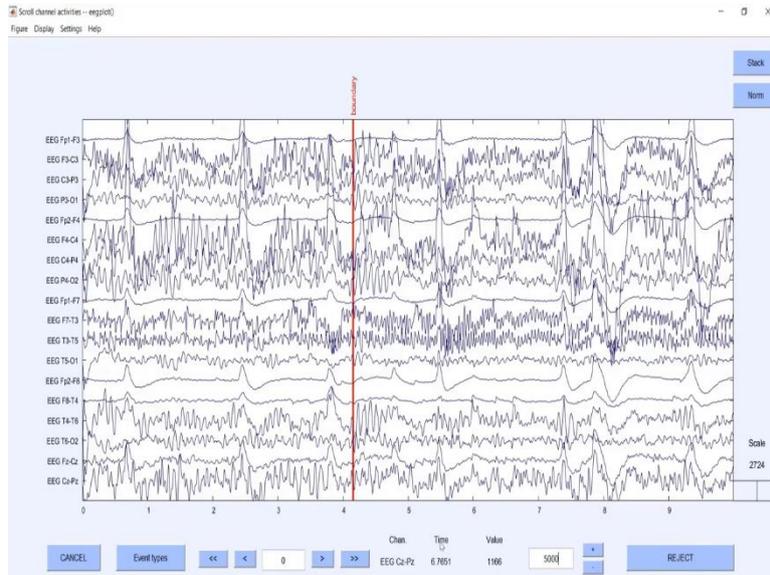




So, it will again ask, that the earlier EDF data set was 1, now I am going to change it as EDF data set 2. Once I do that, now I have a 2 data set here. So, I can process any of the data set by checking the data. Let us plot it together. Let us start it again like we plotted it before. It was for 10 seconds.

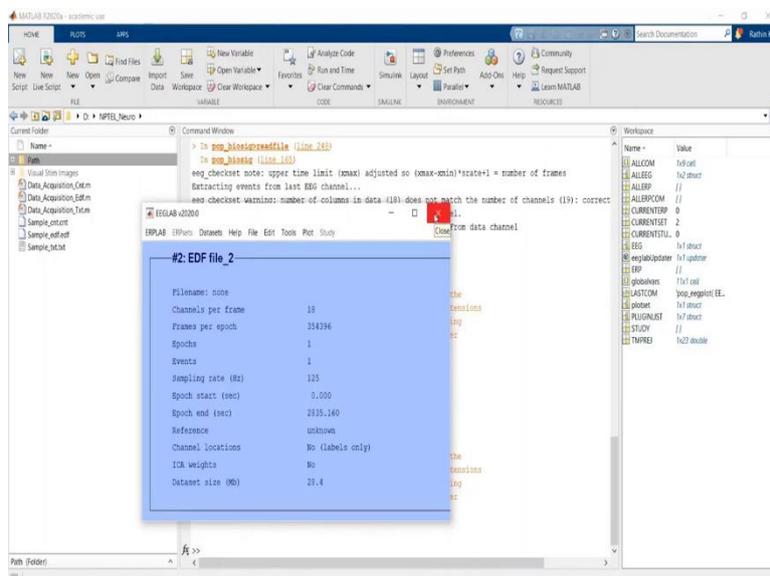
(Refer Slide Time: 12:21)

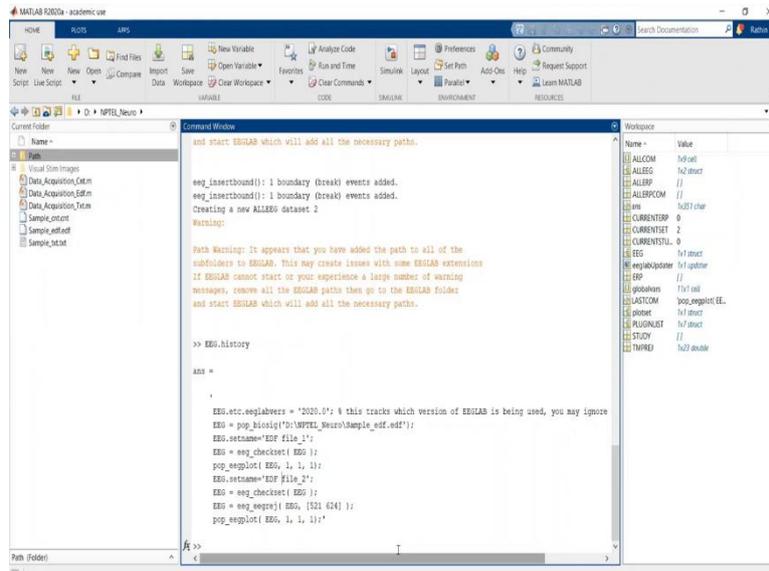




So, I will again change this time range to display. And if you can remember the time amplitude square was around 5,000, let us keep it 5000. So, in between there was a moment it will be in 4 to 5, here you can clearly see a boundary which gives an indication that some of the channel has been removed. So, this is kind of basic things which you can do. So, these are the basic operations using EEGLAB.

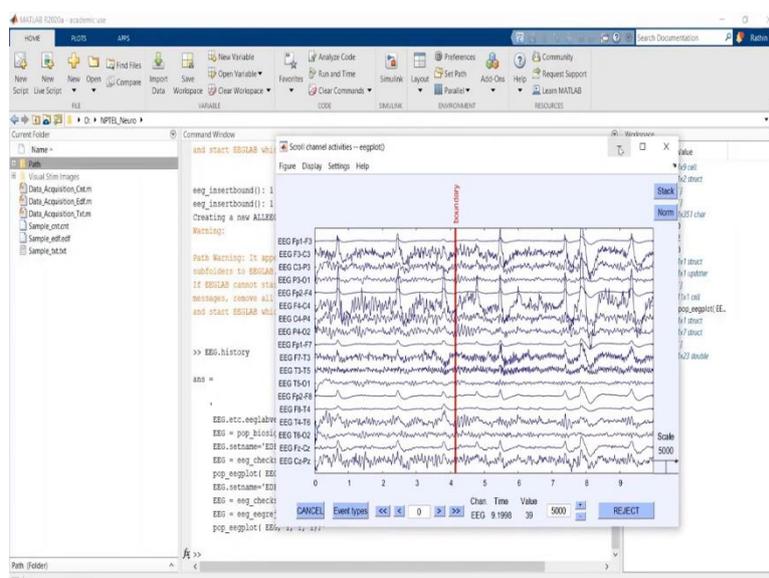
(Refer Slide Time: 14:03)





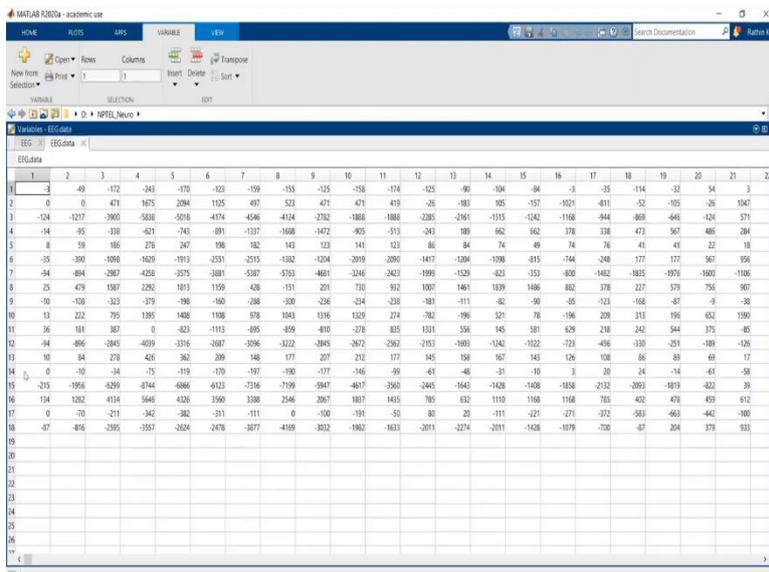
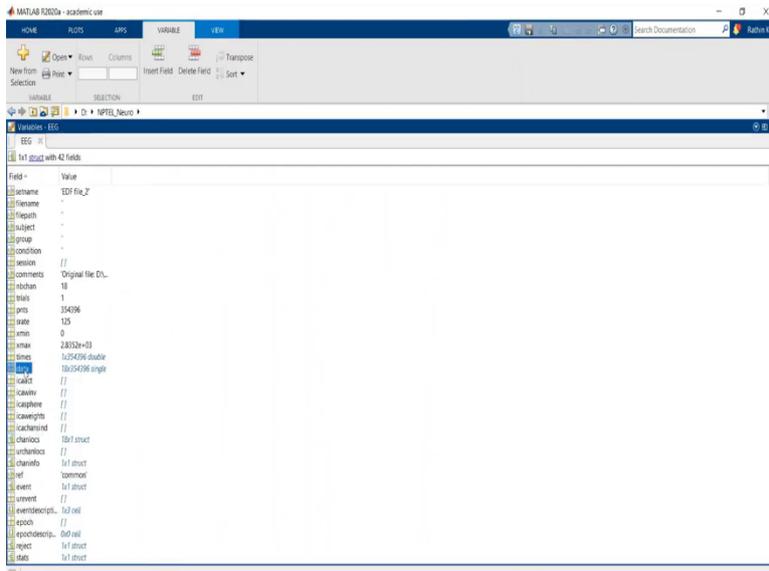
One more important thing about EEGLAB is that whatever processing you have done on your EEGLAB GUI, you can recall it using EEG dot history, EEG dot history. So, from the beginning, whatever you have done in the session, you can see it here. Open the EEGLAB, imported the file, name that set file as a EDF file 1, then you plotted it, again you name another file as EEG file 2 in which you have rejected some of the nuance, some of the indices data. So, this is how you will come to know for one particular GUI operation, which kind of scripting function should be used.

(Refer Slide Time: 15:01)



Now, GUI looks very easy to use and user friendly, then why we are going for scripting? We have seen using the PsychoPy we can, PsychoPy EEG where we can generate a multiple

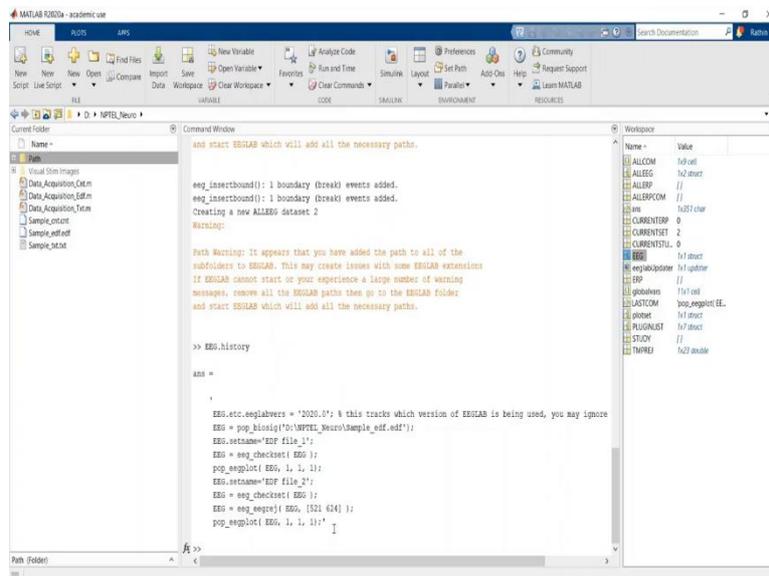




So if you successfully import your file, you will be getting all these fields under EEG structure which is the most important for further signal processing, it has all the channel locations, where it has been placed, whereas, it has the data as well as the venues most important thing for all channels.

If you can see that it is recorded for three lakh as almost 3.5 lakh examples at 125 rate, 125 sampling rates. So, if you want to know exactly how many minutes the recording has been done, you can use this data of number of samples and sampling rate, you should be able to find out that this is for which, how many number of, like what is the duration of the recording.

(Refer Slide Time: 17:07)



So, these are some of the basics of EEGLAB. I hope all of you have understood this thing. Similar way for different acquisition systems, we have different formats of the final acquire data set, and how we can obtain it using different scripting options that we will see in upcoming videos. So for now, I think I hope all of you understood the basic introduction of EEGLAB. We will shift our focus to data acquisition using EEGLAB for various formats from different EEG acquisition system.

(Refer Slide Time: 17:56)

### EEG System File Formats

EEG Recording System	File Format
Open BCI Boards (Ganglion, Cyton, Daisy etc. )	.txt
Compumedics Neuroscan	.cnt
Nicolet EEG Viewer (Natus)	.edf



So all of you have seen our demonstration of EEGLAB, so now most of you have some ideas about which kind of software EEGLAB is and how you can convert your GUI operation into scripting using EEG dot history function or EEG dot history variable, because EEG is a data structure. We will see in the upcoming data acquisition videos demonstration we will see that. It is basically a structure in which history is one of the field, which stores whatever operation which you do equivalent command.

So yeah, different systems stores your EEG information into different formats. Open BCI boards stores in dot txt file, the simplest form you know and you can see it in notepad. Neuroscan stores it into dot cnt file, whereas Nicolet EEG Viewer mostly used for a long run EEG and many number of electrodes are placed. Let us say for epilepsy and more, many number of electrodes are placed on your skull.

So, in such cases, that is EDF files. EDF is the standardized European Data Format for EEG recording. So, I have included a one sample data of each type, one see in txt file, one cnt and edf file and I have shown how you can acquire the data basically biopotentials and event using MATLAB.

So, and also, I have shown how you can see the events for triggers are like auditors stimuli, visual stimuli or whatever stimuli are generated that digital pins how you will fetch it from the acquired loaded EEG data. So, next would be three demonstration, quick short videos. So, in case in future any of you would come across such recorded files, will be able to fetch the data by your own.

So, next would be these three demonstration videos of acquiring EEG and events from recorded files and that will conclude this submodule, and as well as this four-part lecture series on brainwave recording, acquiring and processing. We have only seen loading brainwaves into MATLAB. Further if time permits, we will see how you can process it, like pre-processing is also important step, further extracting a particular biopotential, any time locked event.

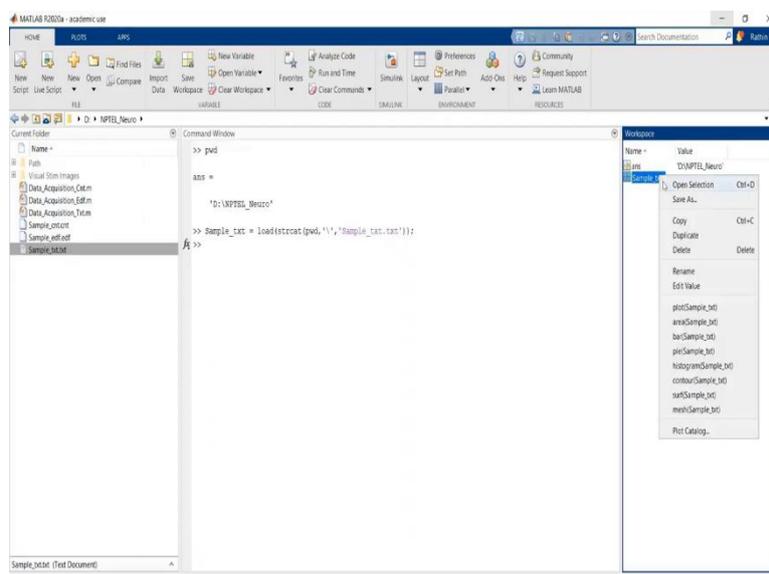
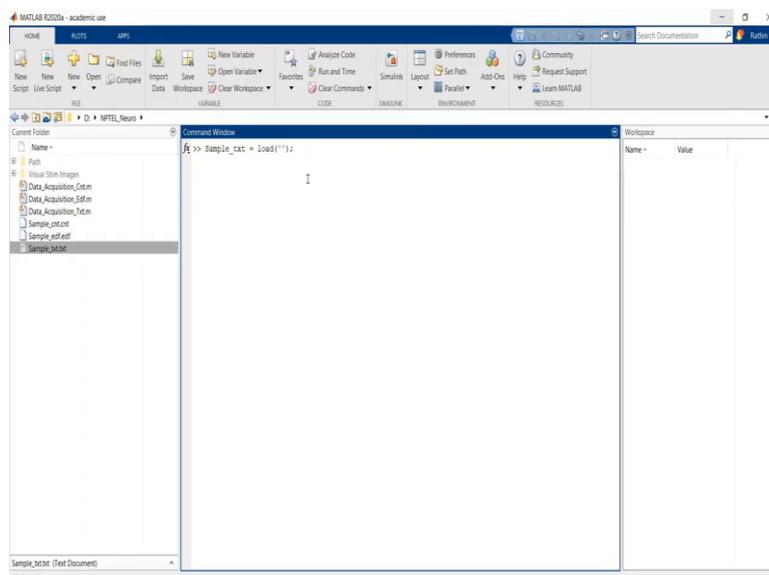
So, all this thing epochs and all, like if time permits, we will cover this topics as well. But for now flow-wise Let us see how acquisition happens and how different formats can be loaded into MATLAB, I will see you in some other module you can enjoy these three different demonstrations and then will meet in some other module for some other topic.



used as well as further this five 0s and 1 are nothing but your digital input outputs. So, we have inserted our trigger in pin number 11.

So, while extracting events, I will consider or I will take the value of pin number 11. I will show it to you, if you can see here, here the 11th column is constantly getting toggled between 0 and 1. We will observe the same in MATLAB as well once we successfully import the file.

(Refer Slide Time: 22:58)



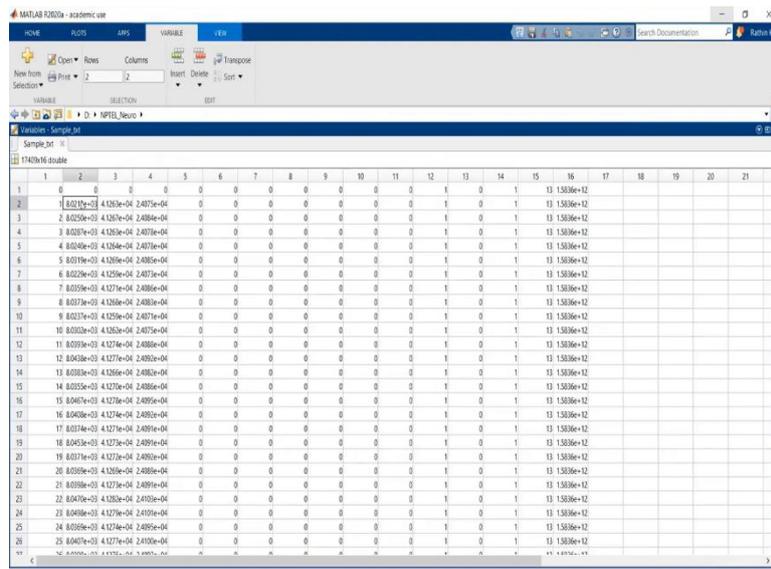
So, let us see how we can generate this file, how we can, also we will see how to write a script in case of multiple subjects scripting is advisable. So, how we can write the script in order to make life simpler. So, in order to import this test file, there is a function MATLAB

which you can use, I will name this variable or whatever ones get imported as sample txt, and let us load the file.

So, while you load the particular file you should write everything like the path and the filename. Now, instead of writing path and file name, I will use a trick here, if I place pwd, it shows me present working directory, the, currently in which folder I am while running into MATLAB. So, what I will do, I will use a strcat function using this pwd as one of the string if you do not see here.

What I will do is, I will use the string concatenation function so it will merge all the inputs, after this output backslash once, and then I will write the filename, which is sample txt dot txt. Now, this also should be given as a string. So, should be something like this. So, ultimately it will merge all the things and it will load by text file into sample text variable. So let us see.

(Refer Slide Time: 24:42)



The image shows the MATLAB R2022a academic use interface. The main window displays a data table with 26 rows and 21 columns. The columns are labeled 1 through 21. The first column contains integers from 1 to 26. The second column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 8.0460e-03. The third column contains integers from 1 to 26. The fourth column contains floating-point numbers in scientific notation, ranging from 2.4875e-04 to 2.4935e-04. The fifth column contains integers from 0 to 1. The sixth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The seventh column contains integers from 0 to 1. The eighth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The ninth column contains integers from 0 to 1. The tenth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The eleventh column contains integers from 0 to 1. The twelfth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The thirteenth column contains integers from 0 to 1. The fourteenth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The fifteenth column contains integers from 0 to 1. The sixteenth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The seventeenth column contains integers from 0 to 1. The eighteenth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The nineteenth column contains integers from 0 to 1. The twentieth column contains floating-point numbers in scientific notation, ranging from 0.0000e+00 to 0.0000e+00. The twenty-first column contains integers from 0 to 1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
2	1	8.0217e-03	4.1262e-04	2.4875e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
3	2	8.0256e-03	4.1267e-04	2.4884e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
4	3	8.0287e-03	4.1265e-04	2.4877e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
5	4	8.0340e-03	4.1264e-04	2.4877e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
6	5	8.0319e-03	4.1265e-04	2.4885e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
7	6	8.0229e-03	4.1259e-04	2.4873e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
8	7	8.0259e-03	4.1271e-04	2.4886e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
9	8	8.0237e-03	4.1260e-04	2.4882e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
10	9	8.0237e-03	4.1259e-04	2.4877e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
11	10	8.0202e-03	4.1262e-04	2.4875e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
12	11	8.0293e-03	4.1274e-04	2.4886e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
13	12	8.0438e-03	4.1277e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
14	13	8.0383e-03	4.1266e-04	2.4882e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
15	14	8.0255e-03	4.1270e-04	2.4886e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
16	15	8.0461e-03	4.1278e-04	2.4895e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
17	16	8.0408e-03	4.1274e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
18	17	8.0214e-03	4.1273e-04	2.4891e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
19	18	8.0453e-03	4.1273e-04	2.4891e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
20	19	8.0371e-03	4.1272e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
21	20	8.0498e-03	4.1268e-04	2.4886e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
22	21	8.0398e-03	4.1273e-04	2.4891e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
23	22	8.0470e-03	4.1263e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
24	23	8.0498e-03	4.1278e-04	2.4894e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
25	24	8.0268e-03	4.1274e-04	2.4895e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
26	25	8.0461e-03	4.1277e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				
27	26	8.0461e-03	4.1277e-04	2.4892e-04	0	0	0	0	0	0	0	1	0	1	13	1.5836e+12				

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2253	204	8.090e-03	4.1202e-04	2.4104e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2254	205	8.0921e-03	4.1203e-04	2.4102e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2255	206	8.094e-03	4.1199e-04	2.4099e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2256	207	8.0738e-03	4.1199e-04	2.4097e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2257	208	8.0751e-03	4.1207e-04	2.4095e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2258	209	8.0754e-03	4.1203	2.4091e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2259	210	8.0851e-03	4.1206e-04	2.4095e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2260	211	8.0938e-03	4.1214e-04	2.4100e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2261	212	8.0793e-03	4.1205e-04	2.4094e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2262	213	8.0776e-03	4.1205e-04	2.4088e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2263	214	8.0754e-03	4.1205e-04	2.4091e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2264	215	8.0759e-03	4.1205e-04	2.4091e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2265	216	8.0773e-03	4.1203e-04	2.4085e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2266	217	8.0751e-03	4.1205e-04	2.4085e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2267	218	8.0751e-03	4.1207e-04	2.4085e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2268	219	8.0729e-03	4.1200e-04	2.4087e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2269	220	8.0700e-03	4.1200e-04	2.4088	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2270	221	8.0748e-03	4.1203e-04	2.4082e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2271	222	8.0674e-03	4.1200e-04	2.4079e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2272	223	8.0704e-03	4.1205e-04	2.4080e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2273	224	8.0803e-03	4.1216e-04	2.4091e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2274	225	8.0849e-03	4.1215e-04	2.4091	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2275	226	8.0807e-03	4.1211e-04	2.4085e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2276	227	8.0664e-03	4.1208e-04	2.4084e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2277	228	8.0658e-03	4.1214e-04	2.4085e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2278	229	8.0667e-03	4.1205e-04	2.4092e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

So, if you can see here sample text variable has been created and saved. The same thing what we have seen in the notepad you can observe here, as well as if I go to the 11th column and once I go further down, I should be able to see the change in triggers, I hope you can see here. If I scroll it slowly, you should be able to see, since zeros are there in the 11th column, again 1 for some period of time. So, this is how you can analyse the trigger, which is a trigger. Though it is same in both 11 and 12th column here, you can take any of the column as an event trigger.

(Refer Slide Time: 25:28)

```

>> pwd
ans =
'D:\MPTEL_Neuro'

>> Sample_txt = load(strcat(pwd, '\', 'Sample_txt.txt'));
  
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
2	1	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
3	2	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
4	3	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
5	4	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
6	5	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
7	6	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
8	7	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
9	8	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
10	9	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
11	10	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
12	11	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
13	12	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
14	13	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
15	14	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
16	15	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
17	16	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
18	17	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
19	18	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
20	19	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
21	20	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
22	21	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
23	22	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
24	23	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
25	24	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					
26	25	8.023e-03	4.1263e-04	2.4075e-04	0	0	0	0	0	0	0	1	0	1	13	1.5806e+12					

So, now how can we take the data out of this? As I already mentioned, this particular experiment is for three channels. So, column wise, it is column number 2 to column number 4. I will show you, show it to you one more time. It is column number 2, column number 3, column number 4. So, these are my biopotential values right now.

(Refer Slide Time: 25:48)

```

>> pu1
ans =
    'D:\NPEL_Neuro'

>> Sample_txt = load(fullfile(pu1, 'Sample_txt.txt'));
>> biopotentials=Sample_txt(:,2:4);
>> everr_chnl = Sample_txt(:,1);

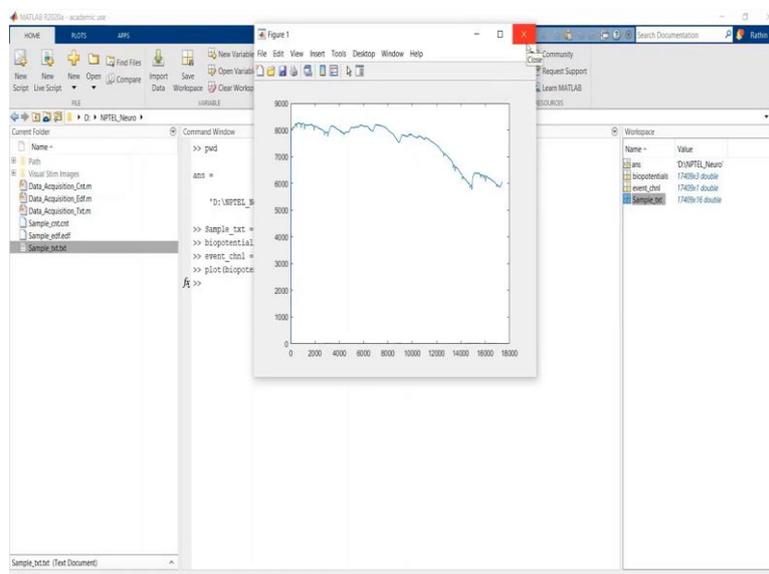
>> plot(biopotentials)
plot(tg,T)
plot(h,t,LineStyle)
plot(x1,Y1,...,Xn,Zn)
plot(x1,Y1,LineStyle,...,Xn,Yn,LineStyle)
plot(t)
plot(r,LineStyle)
plot(____,Value)
plot(as,____)
plot(____)
  
```

So, what I can do is and make sure it is a row value, you might say, that it is in the 10th to the power here or 10 to the power 4, but yes, its row value which is generated by the addition of so many noise that we have to remove gradually using proper pre-processing algorithm. So, what we can do here is, I will store it into a variable biopotential, the first three potential I will save it as a biopotential.

Now, I can take it from sample text and I will fetch first three, like second, third and fourth column. So, this is basic a MATLAB way to extract particular columns from any array, very simple. So, that three channels has been recorded here, you can check this thing here. Furthermore, Let us take a channel also event data or chan, which we say an event channel.

So, we can store in the similar fashion from sample dot txt and I need now 11th column because we have already seen that 11th column digit, of digital input output, keep get, was keep getting toggled. So, now I, we have both the data event channel as well as our biopotential, so, let us see, let us see just for visualization how it looks like plot biopotential. For now, we will plot only one channel just to give you guys an idea.

(Refer Slide Time: 27:40)

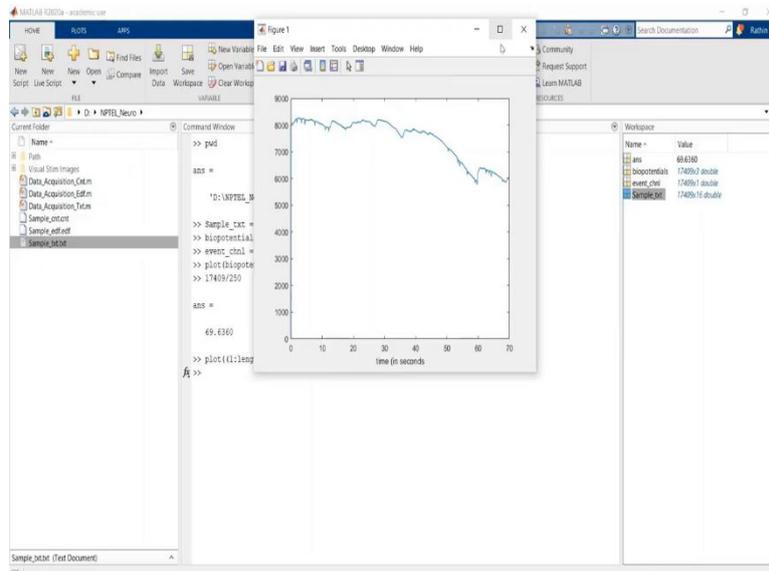


So, it is something looks like this. So, the first channel was having some starting value around 8,000 hours, so it is looking like this.



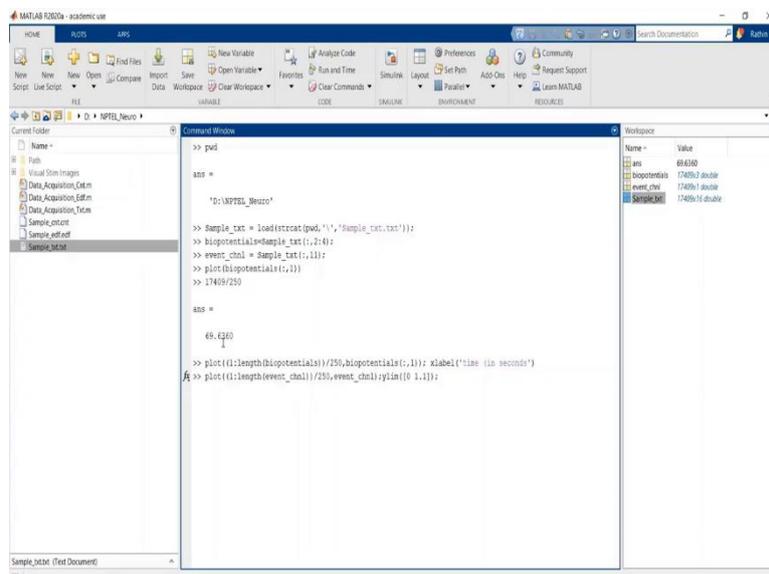
biopotential and you can divide it by 250, should give you the original timescale in seconds. Let us write that, so let us see.

(Refer Slide Time: 29:22)



So, this is nothing but I already mentioned so 16, 17 second video. So you can get it with respect to time, now you can analyse that and you can check it further.

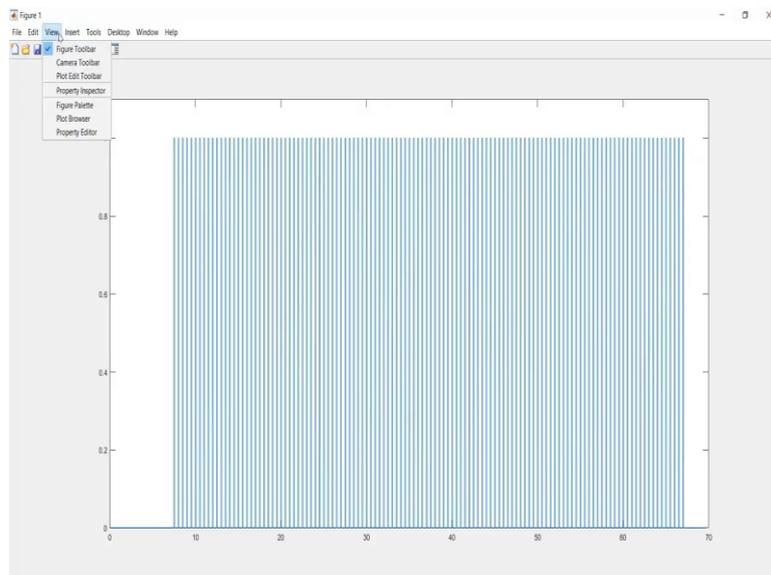
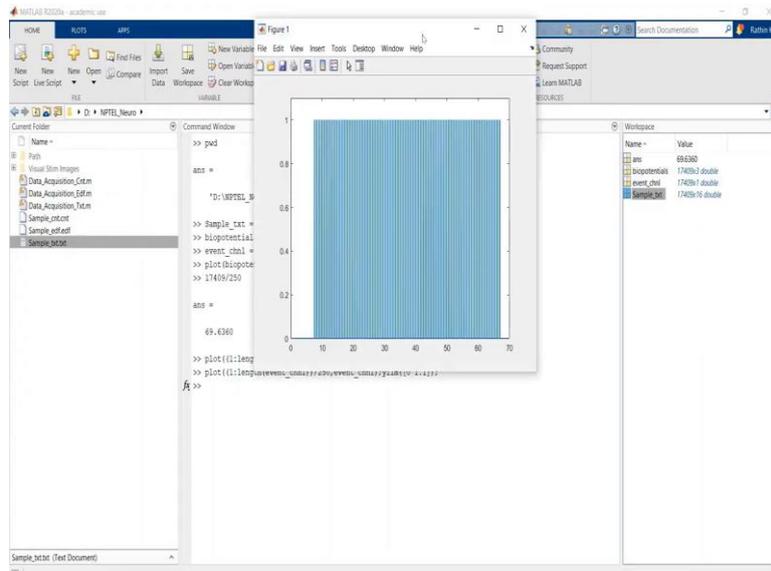
(Refer Slide Time: 29:37)



Let us see, Let us see the triggers as well. So, we can plot now event channel. I already mentioned that event channel is your digital input. So, it will be 0 and 1 in order to just look at, in order to just make it look fancy or, and good, I will just put a y limit to 0 to 1.1. Let say that x, and again, I will plot it with respect to a generated time.

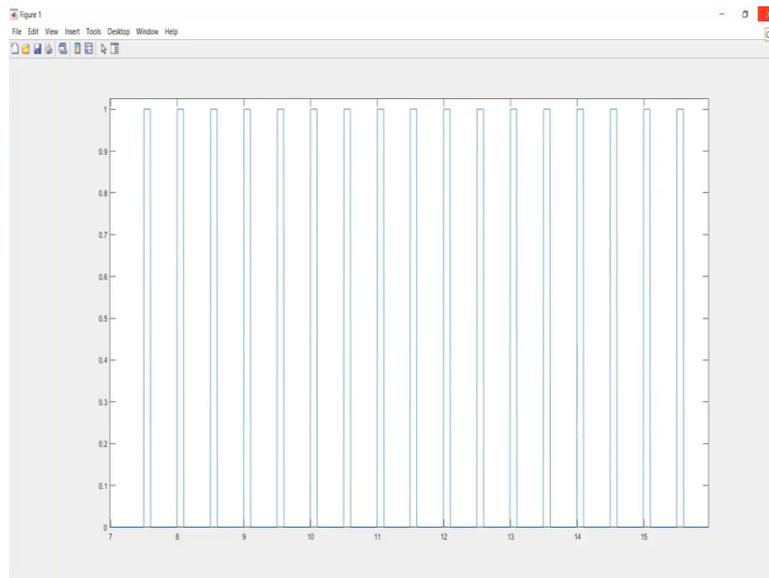
So, I will follow the same thing which we have done it, in the case of biopotential. I will give it indices, time indices of the same, you can use length of biopotential as well, that is fine, as long as you can plot the channel properly, that is completely okay. And I have to divide by the sampling rate as well. So, I guess now, we should be able to see the triggers as well.

(Refer Slide Time: 30:48)



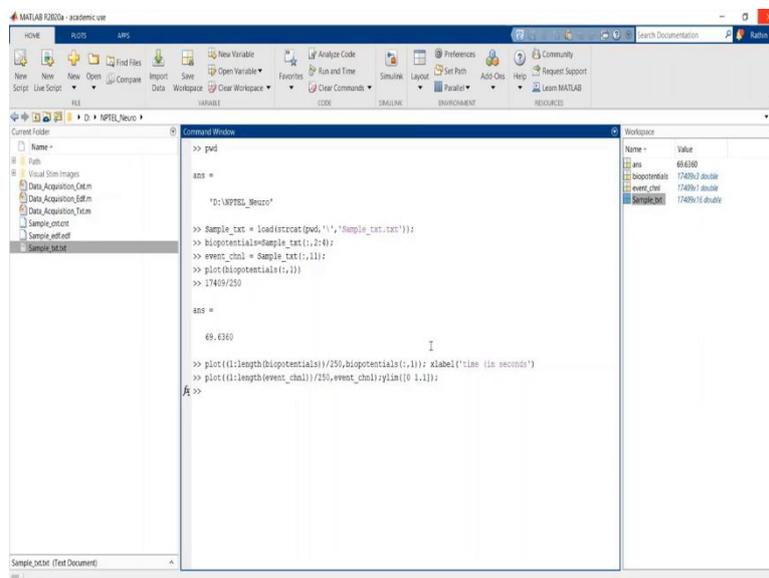
That is it. So these are the triggers in similar fashion you are getting from here. One thing to note down here is that experiment has started any after few seconds, the triggers have been inserted. So, you can further analyse it, you should consider the data corresponds to only the time when triggers have been given. If your experiment is audit like some evoked potential generated using this trigger, you can further check it, zoom it to verify whether the generated data is really accurate or popular, accurate or not.

(Refer Slide Time: 31:00)



So, I have shown you an image while explaining auditory stimuli, this is the output of the similar auditory stimuli, in which you can clearly see that it is getting repeated twice in a second, so two hertz auditory stimuli and the similar kind of stimuli has been presented here. So, I think this is a realisation of what I showed that we have made using PsychoPy.

(Refer Slide Time: 31:54)



```
>> pval
ans =
    49.4360

>> load('src/matlab/pval', 'Sample_txt.txt');
>> biopotentials=sample_txt(:,2:4);
>> event_chnl = Sample_txt(:,11);
>> plot(biopotentials(:,1))
>> 17409/250

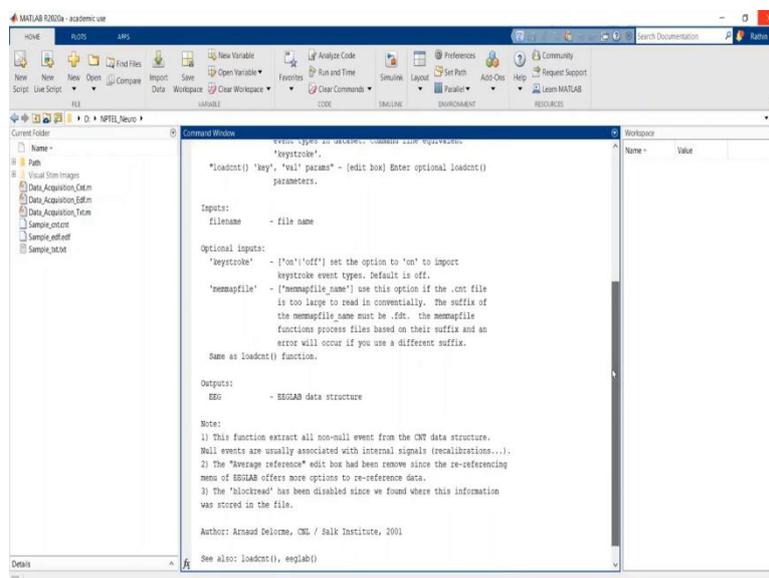
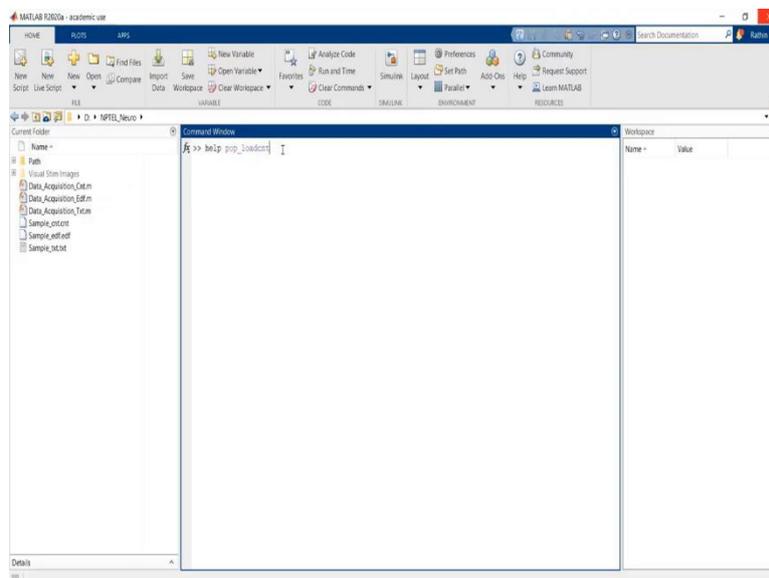
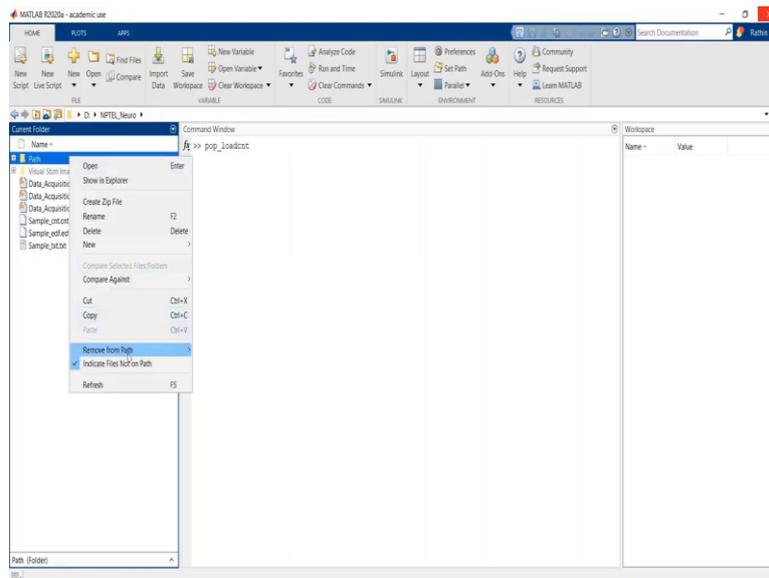
ans =
    49.4360

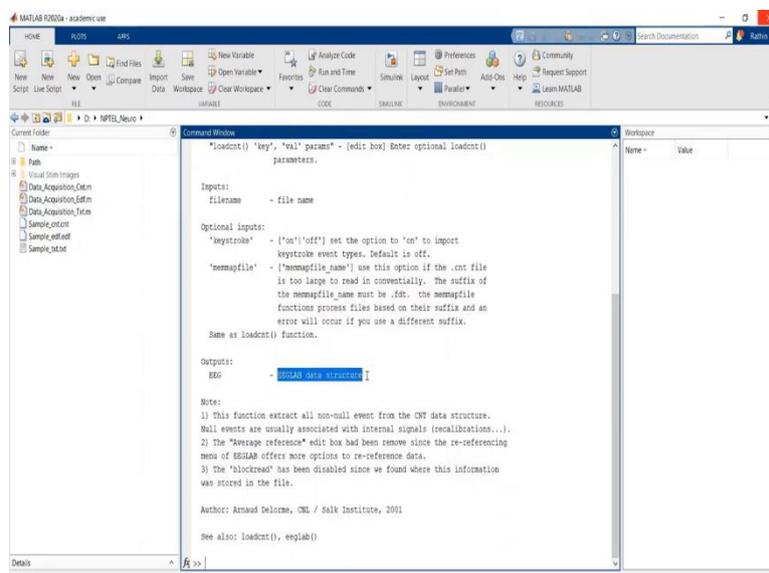
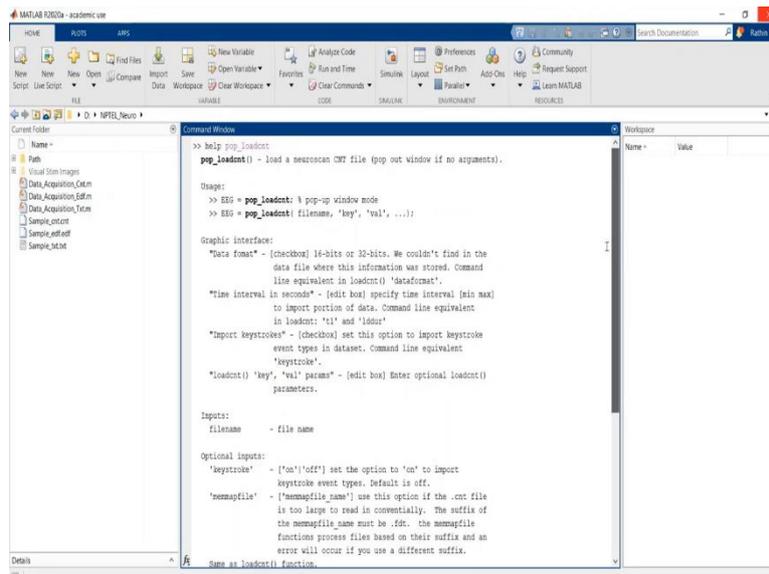
>> plot((1:length(biopotentials))/250,biopotentials(:,1)); xlabel('time (in seconds)')
>> plot((1:length(event_chnl))/250,event_chnl(:,1)); ylim([0 1.1]);
>>
```

Name	Value
ans	49.4360
biopotentials	17409/2 double
event_chnl	17409/1 double
Sample_txt	17409/11 double

So, similarly, you can try out your own triggers and record your potentials with some of the biopotential acquisition device. In our case, we have used OpenBCI Cyton board, which I showed you in the previous module. So, I hope this MATLAB acquisition of your data and events is clear and you can now import the data and further process it.

(Refer Slide Time: 32:21)





Let us see how we can load different EEG file formats recording from various devices. We have already seen one instance of that. In one of the videos I have shown you how we can import OpenBCI Cyton board recorded file which was the sample txt dot txt file, how you can record, how you can rather import that file into MATLAB for further processing.

In a similar fashion, we will see we have two more instances, one is sample cnt dot cnt file, which is a file recorded for our Neuro-scan EEG system. So, how you can load the particular file. In OpenBCI case, there was a text function, a lobe function for the text file. In this case, we have a specific function which is popular cnt. This is a function of our neuro scanner plugin, it is also a part of EEGLAB function.





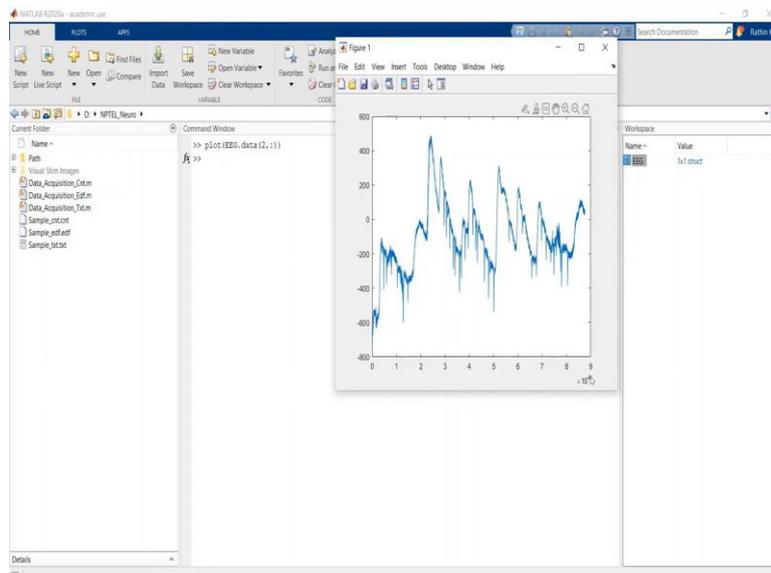
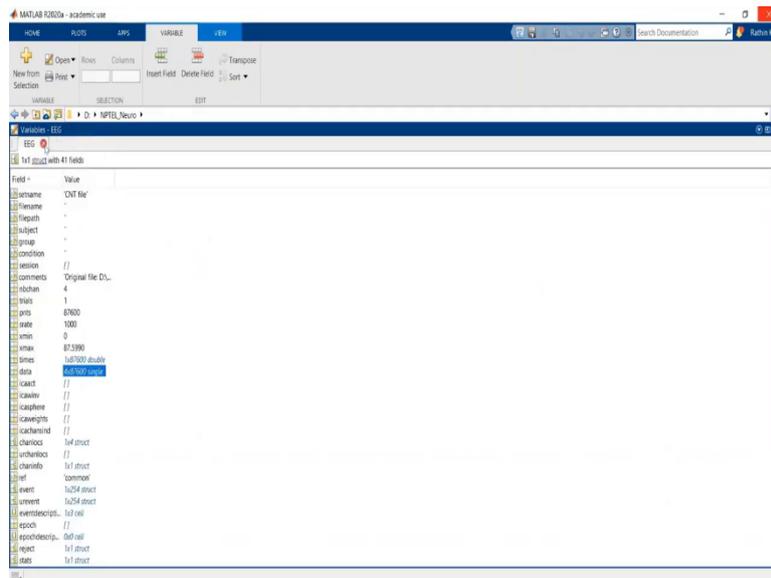


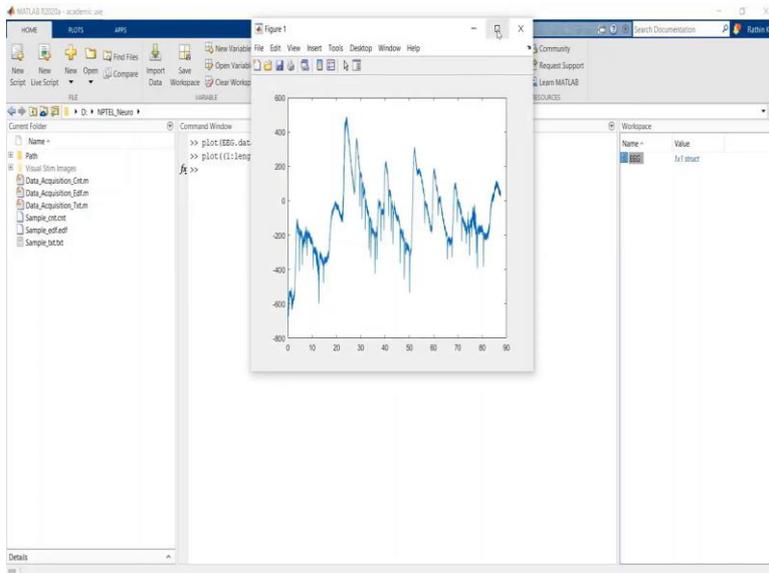
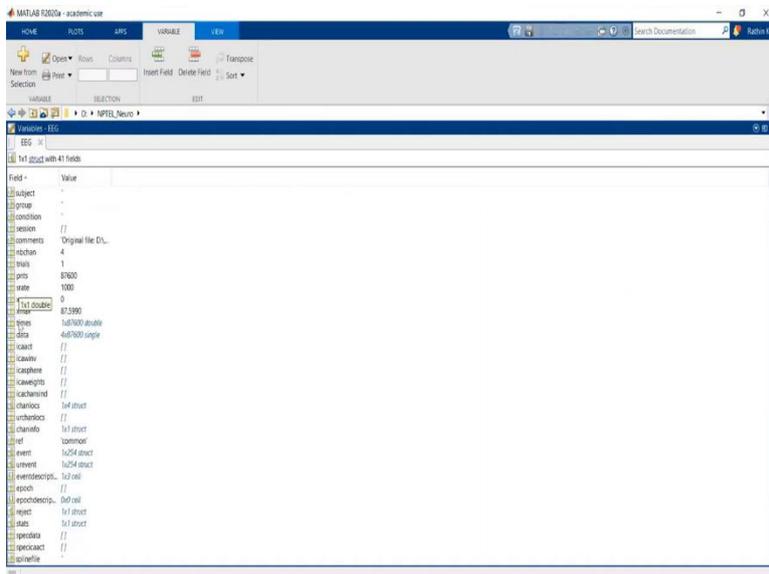
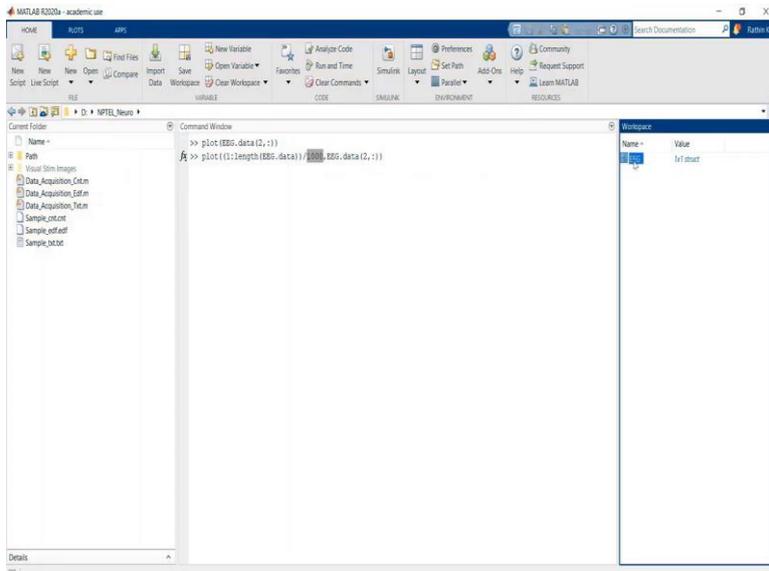
	9693	9694	9695	9696	9697	9698	9699	9700	9701	9702	9703	9704	9705	9706	9707	9708	9709	9710	9711	9712	9713	
1	12.7847	-425.5433	-424.8475	-423.4817	-428.8890	-431.5425	-424.8602	-420.1342	-422.7896	-428.1168	-418.6768	-414.5604	-413.2209	-419.1407	-427.9348	-427.7821	-422.0724	-425.0263	-427.7249	-426.0566	-422.6884	-421.6182
2	16.2058	-227.8446	-227.7353	-226.3633	-232.1987	-235.8738	-238.5335	-234.2885	-225.8330	-220.5161	-222.0499	-217.7393	-216.9648	-222.3489	-231.8195	-234.4893	-227.9623	-232.3525	-234.4018	-232.4041	-226.3505	-224.2089
3	14.4064	-425.4492	-425.5218	-423.4991	-428.9919	-432.7862	-438.4431	-425.5937	-426.8023	-428.9136	-420.7021	-416.2055	-416.4278	-423.8414	-432.8962	-435.2081	-424.8275	-431.6700	-427.0300	-434.7012	-425.5470	-423.2638
4	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096	4096
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						
14																						
15																						
16																						
17																						
18																						
19																						
20																						
21																						
22																						
23																						
24																						
25																						
26																						
27																						
28																						
29																						
30																						
31																						
32																						

If you observe the data, it is very evident and clear that first three rows are, has the values which varies with respect to time. And if I just talk about the nature of the values, they are analog in nature, whereas the values in the fourth column you can observe here it is zero. Further if I move ahead, still it is 0. So, after a few seconds, it will get toggled. So, let us see, let us see and scroll it slowly. Yeah, here if you see it has changed to 4096.

Now, why 4096? Because 4096 is 10 to the power, 2 to the power 13 or 14. So, a particular bit in the data stream will get toggled from 0 to 1. When a specific stimuli is generated. So other way, it is to one particular bit out of that stream is changed from 0 to 1. So, you can see that event is generated, I just want you people to realise that this three, this three rows are your biopotential, analog in nature, at the same time this last row is your data, sorry, your events, event data, which is instored.

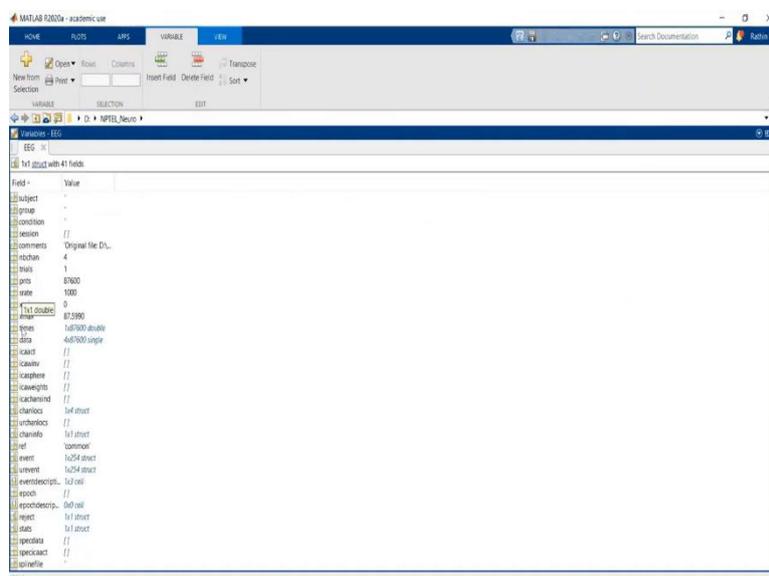
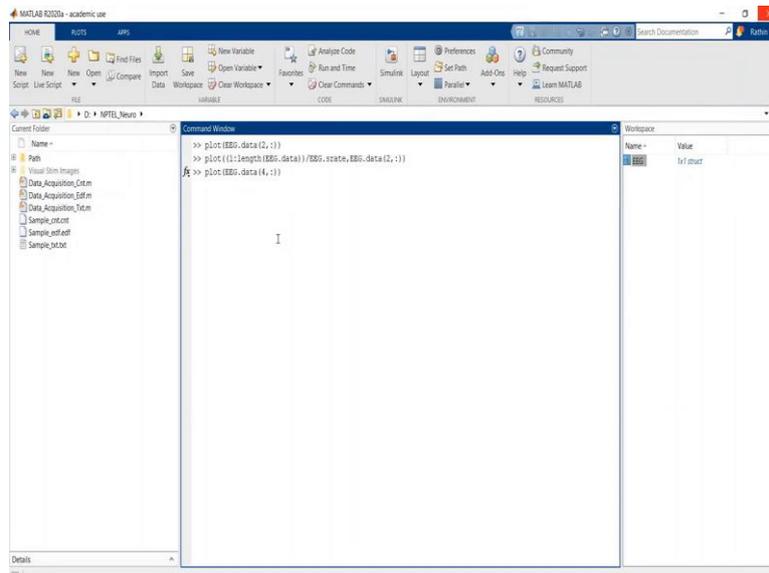
(Refer Slide Time: 37:41)





Now let us quickly visualise what I said. So, just clear it. We can plot one channel from EEG dot data. Let us plot second row and you can see the graph. Again, like you did in OpenBCI, if you want to see this, if you want to see in the time domain, if you are comfortable not in terms of samples, if you want to see it in seconds, what you can do is,

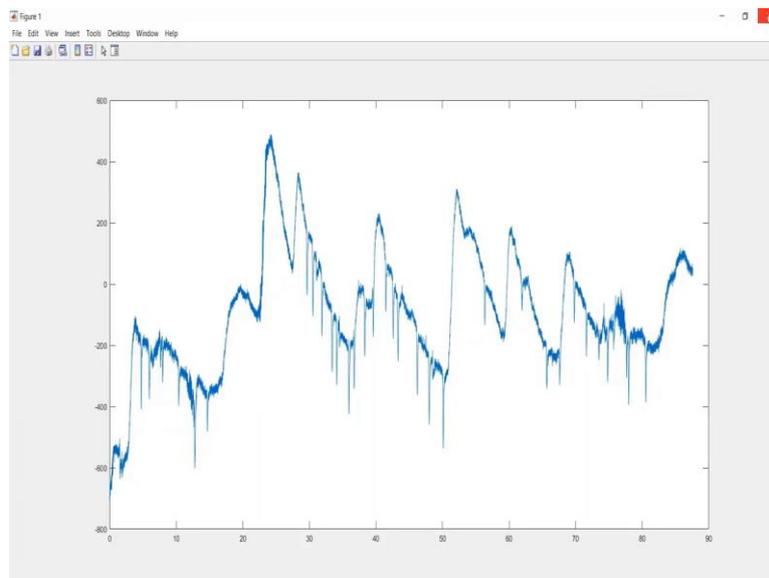
(Refer Slide Time: 38:12)



You can plot the same thing for one colon length of EEG dot data divide by 1000. This should be, why 1000, because we have seen that sampling rate is 1000. Instead, if you want to write you can write EEG dot s rate also, that is completely fine. It is just a basic way of addressing a particular variable from your structure.

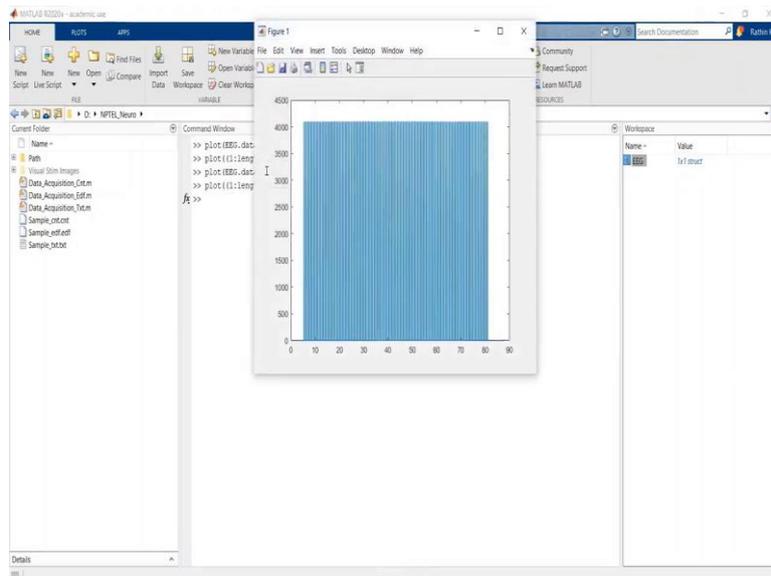
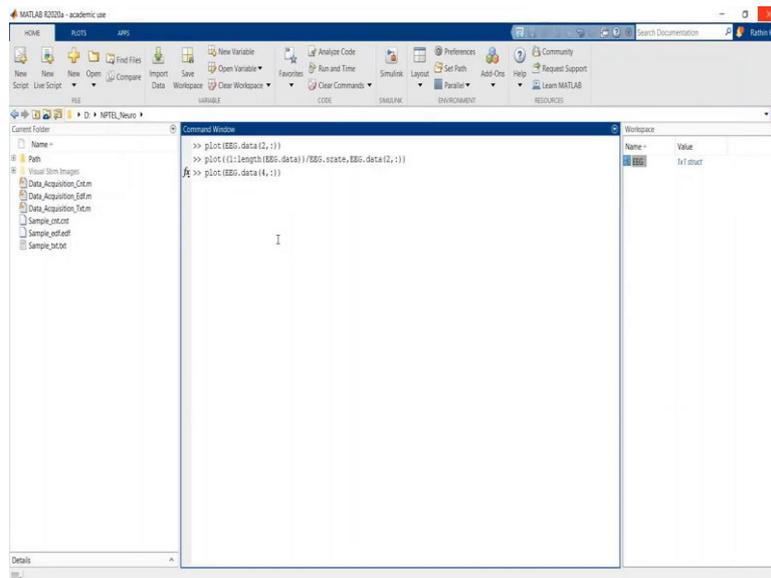
Structure for all those who are not aware about what is structure, it is a kind of composite data type in which you can store multiple variables. Like if you know variables, there are different type of variables, right, some of them are integer, float, string and all. Structure is composite data type, which can have more than one type of data. Like if this asset is let us say double whereas if I go to the sat name is character array, so like that, you can consider structure as a composite data type. And finally, once I am done with this, here it should be dot first.

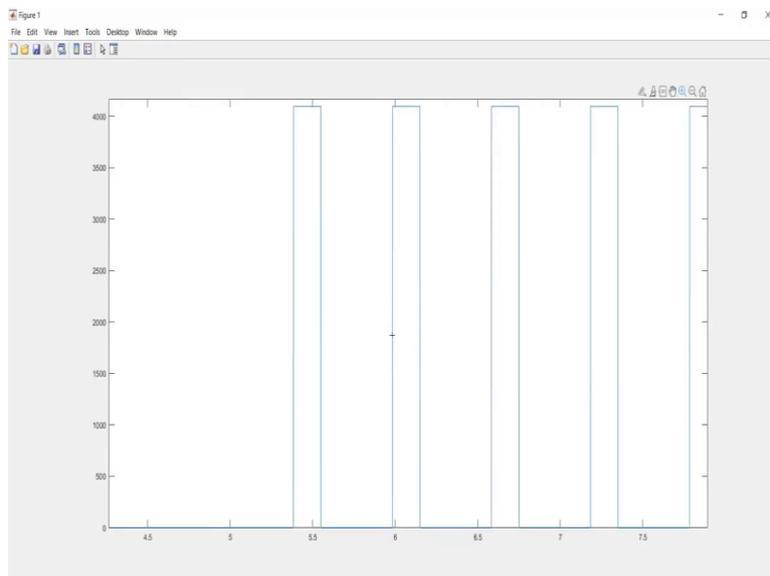
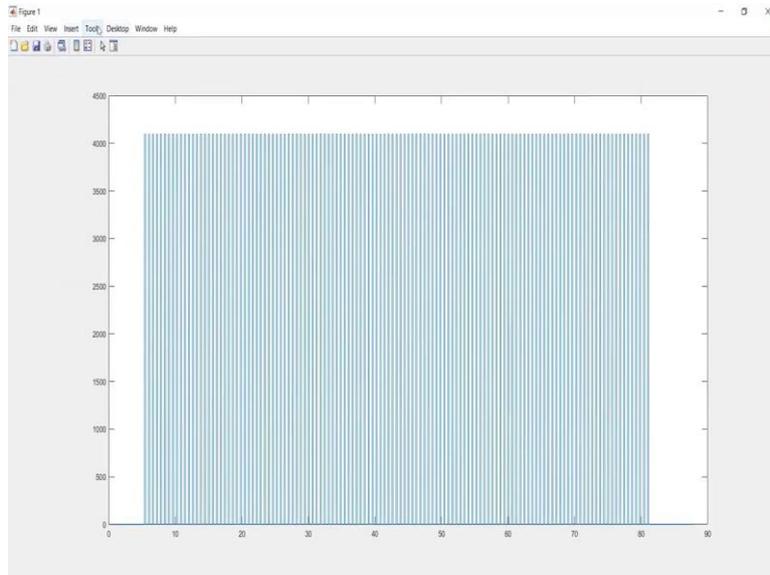
(Refer Slide Time: 39:35)



So, if I run that, it is showing me this kind of graph, exactly in relation or in agreement with what we have seen the total number of time duration is around 89 seconds. And you can see this kind of EEG movement.

(Refer Slide Time: 39:50)



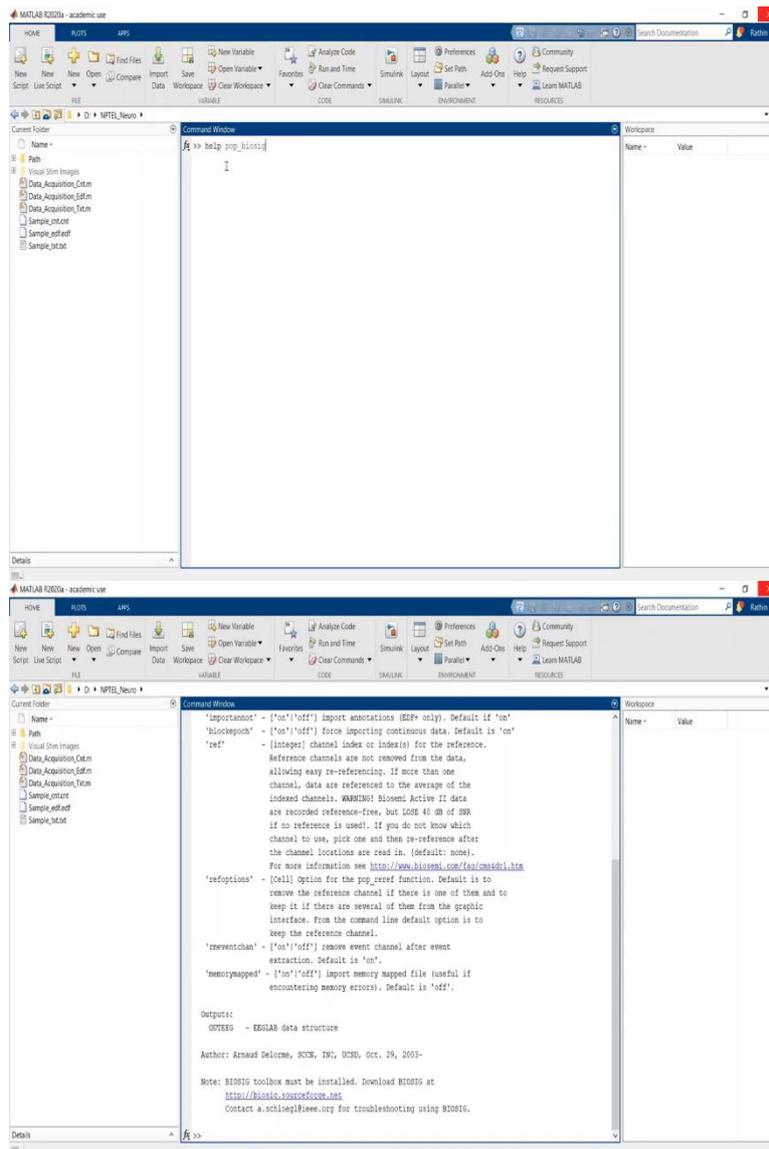


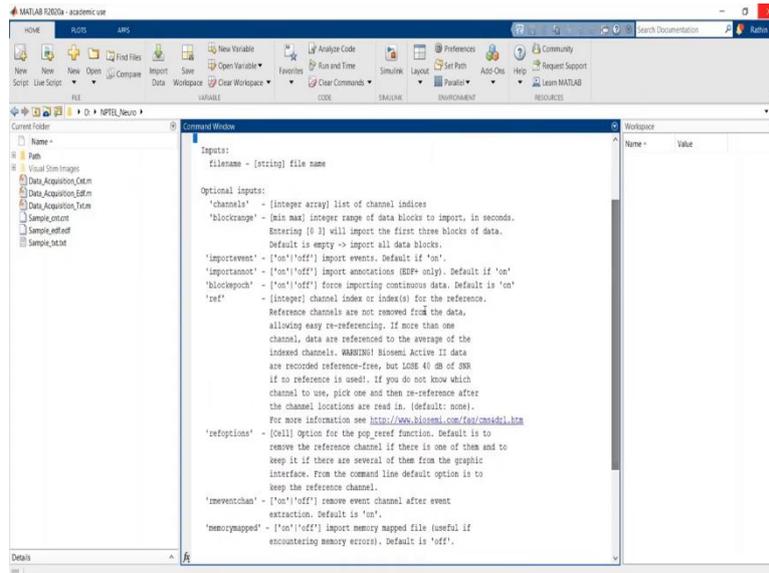
Let us quickly have a look at the events. So, events was stored in the fourth column. So, we will see the events first. So, you need to zoom in to see exactly how this event looks like and what is the time duration between consecutive events. Before that, let us convert the samples into the actual time, like we did for biopotential.

So, in this case, you should use the fourth number of the row. Similarly, you can plot it, it should show you now from 0 to 89 seconds, and if I further zoom in, Let us take only 4 to 5 iteration. By the way, this one stimulus from here to here interval is your one iteration. You should perform at least 100 or 120 iteration based on your experiment, because this brain response is highly variable.

So, you need to average out some of the noise and so, it will also average out some of the common noise and all. But yeah, if you see here, if you focus on this, this stimulus onset is around 6 seconds and it lasts till 6.6, 6.5 seconds around 650, 600 milliseconds for one period. So yeah, this now you can analyse it, your events and your biopotential, further you can process it based on your requirement.

(Refer Slide Time: 41:26)

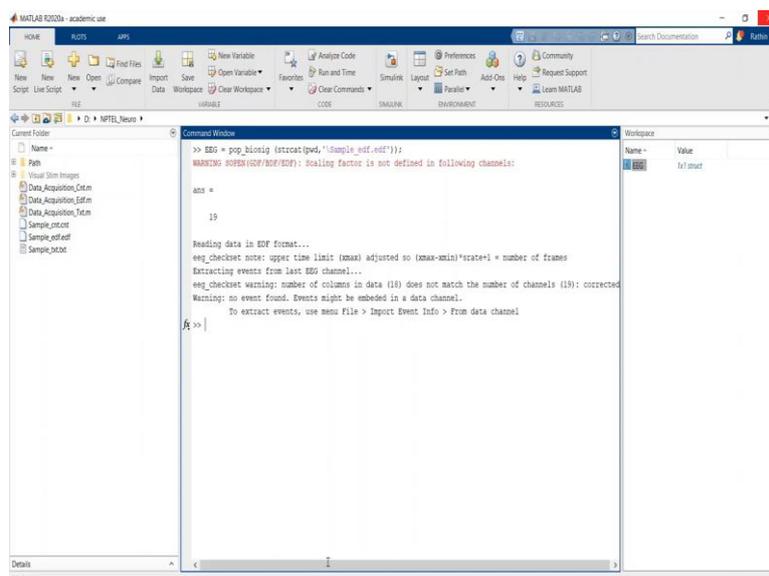




Let us quickly see how we can import EEG biopotentials and channel from EDF file. So, if you have seen the introduction to MATLAB, I have shown you how we can import EDF file using GUI. At that time I have mentioned that pop BIOSIG function is used to import a EDF file. Now what is BIOSIG function? So, Let us just take the help of EEGLAB.

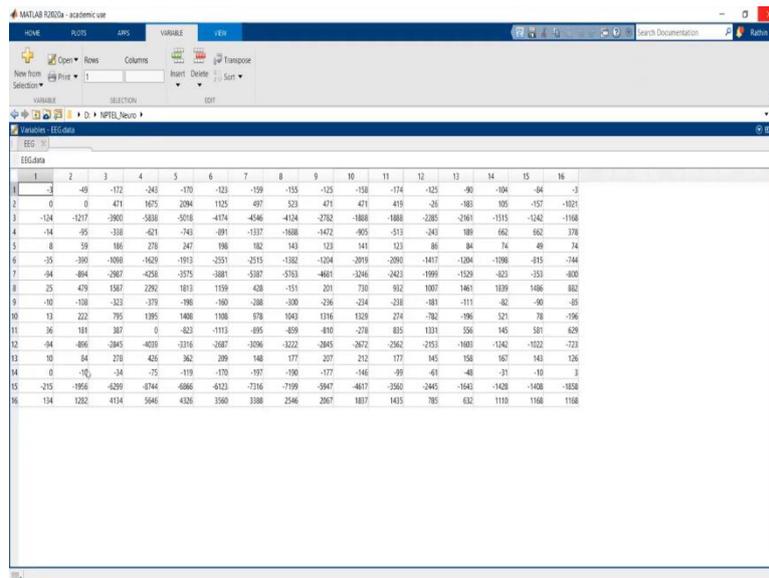
One more thing to mention that, I am using EEGLABs function. So, you must add EEGLAB to your path. I hope all of you know how to add EEGLAB to a path and basics of you know your default path and how you can restore it. These are the different ways in which you can implement your purpose and function for loading a data from dot EDF file these are different optional inputs, you can have a look.

(Refer Slide Time: 42:28)





(Refer Slide Time: 43:34)

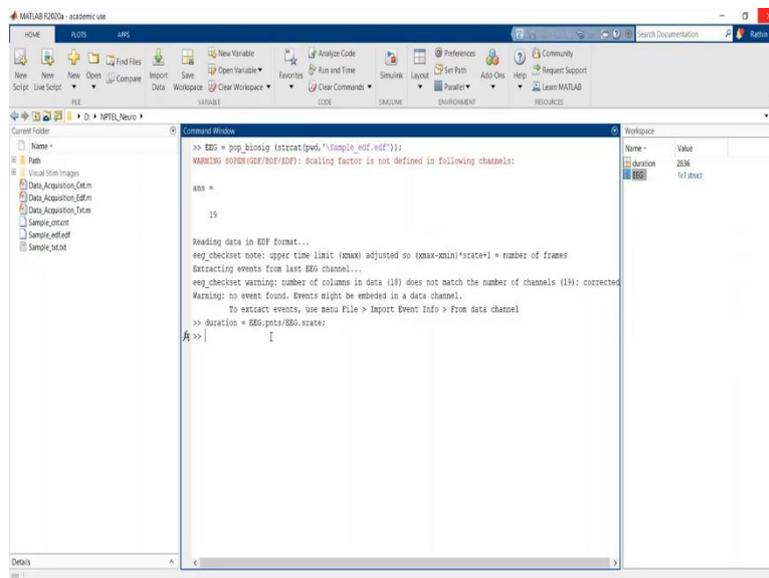


The image shows the MATLAB R2020a academic use interface. The main window displays a table of data with 16 columns and 16 rows. The columns are numbered 1 through 16, and the rows are numbered 1 through 16. The data values are numerical, ranging from approximately -3 to 1168. The interface includes a menu bar with options like HOME, PLOTS, APPS, VARIABLE, and VIEW. Below the menu bar, there are various toolbars and a workspace area showing the current data table.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-3	-49	-172	-243	-170	-123	-159	-155	-125	-158	-174	-125	-90	-104	-84	-3
2	0	0	471	1675	2084	1125	497	523	471	471	419	-26	-183	105	-157	-1021
3	-114	-1277	-2900	-5030	-5018	-4174	-4046	-4174	-2762	-1888	-1888	-2205	-2761	-1515	-1242	-1168
4	-14	-95	-338	-421	-743	-891	-1037	-1608	-1472	-905	-913	-242	189	662	882	378
5	8	59	186	278	247	198	182	143	123	141	123	86	84	74	49	14
6	-35	-390	-1068	-1420	-1913	-2551	-2515	-1382	-1204	-2019	-2090	-1417	-1304	-1098	-815	-744
7	-94	-804	-2987	-4258	-3575	-3881	-5387	-5763	-4681	-3246	-2423	-1999	-1529	-823	-353	-800
8	25	479	1587	2292	1813	1159	428	-151	201	730	932	1007	1461	1839	1486	882
9	-10	-108	-323	-379	-190	-190	-208	-300	-236	-234	-238	-181	-111	-80	-90	-85
10	13	222	795	1895	1408	1108	878	1043	1316	1329	274	-782	-196	521	78	-196
11	36	181	387	0	-823	-1113	-895	-859	-810	-278	835	1311	526	145	381	829
12	-14	-886	-3845	-4039	-3316	-2687	-3096	-3232	-2845	-2672	-2362	-2155	-1803	-1242	-1022	-723
13	10	84	278	426	362	289	148	177	207	212	177	145	158	167	143	126
14 <td>0</td> <td>-10</td> <td>-14</td> <td>-75</td> <td>-119</td> <td>-170</td> <td>-197</td> <td>-190</td> <td>-177</td> <td>-146</td> <td>-99</td> <td>-61</td> <td>-40</td> <td>-31</td> <td>-10</td> <td>3</td>	0	-10	-14	-75	-119	-170	-197	-190	-177	-146	-99	-61	-40	-31	-10	3
15	-215	-1956	-6299	-8744	-6866	-6523	-7316	-7199	-5947	-4637	-3560	-2445	-1643	-1428	-1408	-1858
16	134	1282	4134	5646	4326	3360	3188	2546	2657	1837	1435	785	632	1110	1168	1168

Here you can see the data. It is a bit longer duration, which you can see here. We can see how many sampling points are there and how many what is the sampling rate. So, let us just try to find out the duration.

(Refer Slide Time: 43:45)



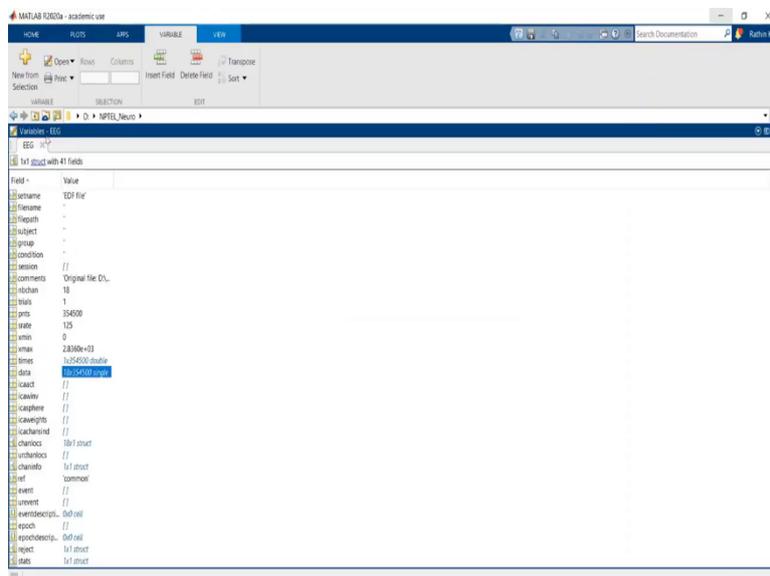
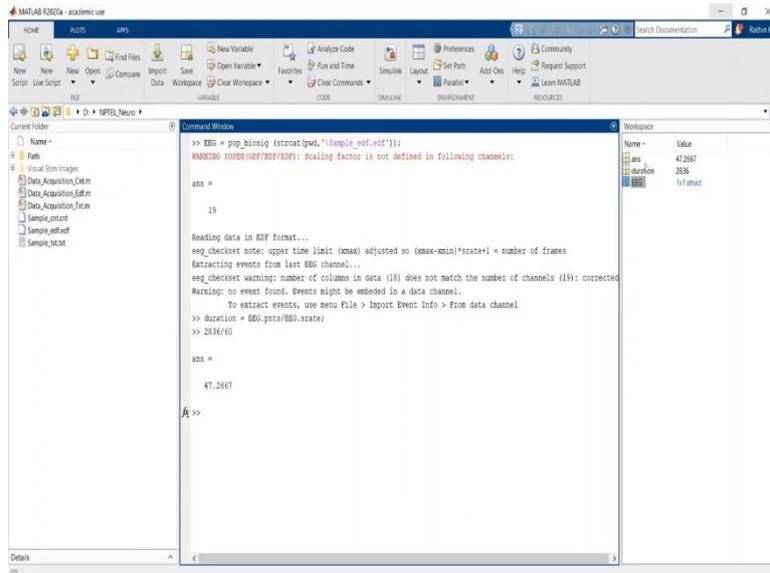
The image shows the MATLAB R2020a academic use interface. The Command Window is open, displaying the following code and output:

```
>> EEG = pop_biosig('stratipool','sample_rate');
WARNING: pop_biosig('stratipool','sample_rate'): scaling factor is not defined in following channels:
EEG =
    15

Reading data in EDF format...
eeg_checklist note: upper time limit (kmax) adjusted so (kmax-min)*rate+1 = number of frames
Extracting events from last EEG channel...
eeg_checklist warning: number of columns in data (18) does not match the number of channels (19): corrected
Warning: no event found. Events might be embedded in a data channel.
To extract events, use menu File > Import Event Info > From data channel
>> duration = EEG.pnts/EEG.rate;
duration =
     1
```

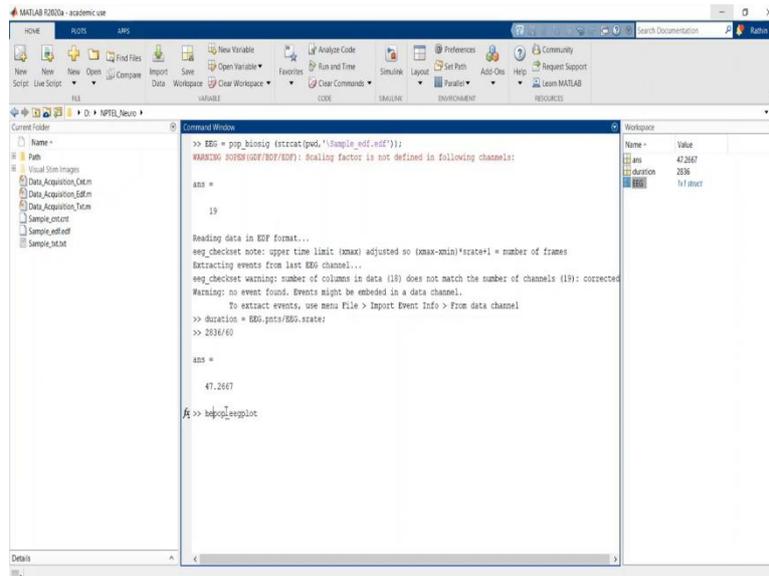
The Workspace window shows the following variables:

Name	Value
duration	2036
EEG	1x1 struct



So, you can use the same EEG dot points variable divided by EEG dot srate. This should give you the total duration in seconds. So, it is about 2836 seconds. So, if I convert it into around 2836 divided by 60. So, 47 minute recorded data. And one more thing if you observed it was recorded for 18 channels. So, it should like, channels would have covered your entire score, entire head, let us see what are there in the channel and where it has been placed using the channel name or channel label.

(Refer Slide Time: 44:50)



The screenshot shows the MATLAB R2020a academic use interface. The Command Window contains the following text:

```
>> EEG = pop_biosig(struct(pwd,'\Sample_001.eeg'));
WARNING: POPSIG(SDF/SDF/EEG): scaling factor is not defined in following channels:

ans =

    19

Reading data in EDF format...
seg_checker note: upper time limit (tmax) adjusted so (tmax-min)*rate+1 = number of frames
Extracting events from last EEG channel...
seg_checker warning: number of columns in data (18) does not match the number of channels (19): corrected
Warning: no event found. Events might be embedded in a data channel.
To extract events, use menu File > Import Event Info > From data channel
>> duration = EEG.pnts/EEG.rate;
>> 2836/60

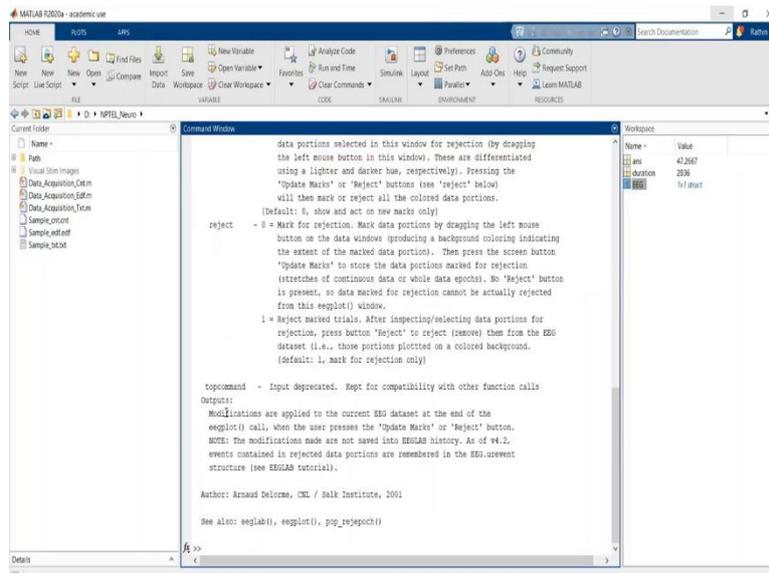
ans =

    47.2667

f1 >> biosig_eeplot
```

The Workspace window shows the following variables:

Name	Value
ans	47.2667
duration	2836
EEG	1x1 struct



The screenshot shows the MATLAB R2020a academic use interface. The Command Window contains the following text:

```
data portions selected in this window for rejection (by dragging
the left mouse button in this window). These are differentiated
using a lighter and darker hue, respectively. Pressing the
"Update Marks" or "Reject" buttons (see "reject" below)
will then mark or reject all the colored data portions.
(Default: 0, show and act on new marks only)
reject - 0 = Mark for rejection. Mark data portions by dragging the left mouse
button on the data windows (producing a background coloring indicating
the extent of the marked data portion). Then press the screen button
"Update Marks" to store the data portions marked for rejection
(stretches of continuous data or whole data epochs). No "Reject" button
is present, so data marked for rejection cannot be actually rejected
from this eegplot() window.
1 = Reject marked trials. After inspecting/selecting data portions for
rejection, press button "Reject" to reject (removes) them from the EEG
dataset (i.e., those portions plotted on a colored background.
(default: 1, mark for rejection only)

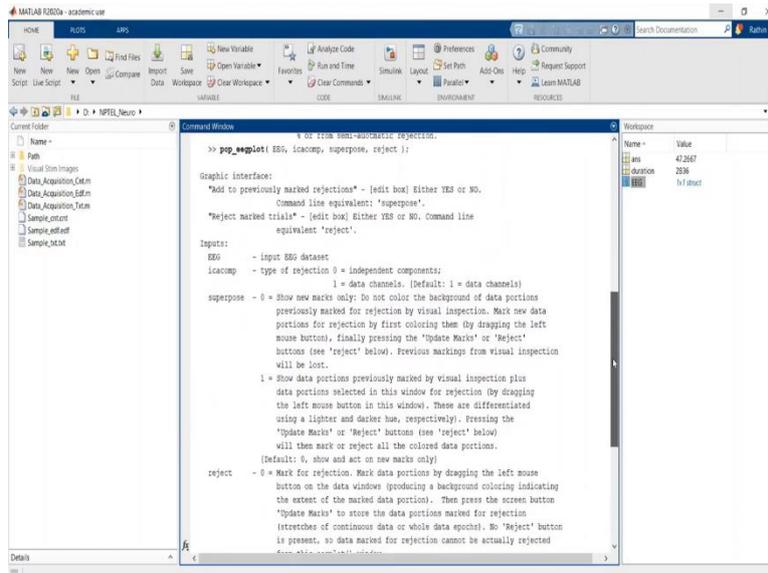
topcommand - Input deprecated. Kept for compatibility with other function calls
Outputs:
Modifications are applied to the current EEG dataset at the end of the
eegplot() call, when the user presses the "Update Marks" or "Reject" button.
NOTE: the modifications made are not saved into EEGlab history. As of v4.2,
events contained in rejected data portions are remembered in the EEG.urevent
structure (see EEGlab tutorial).

Author: Arnand Delorme, CNRS / Salk Institute, 2001

See also: eeglab(), eegplot(), pop_rejepoch()
```

The Workspace window shows the following variables:

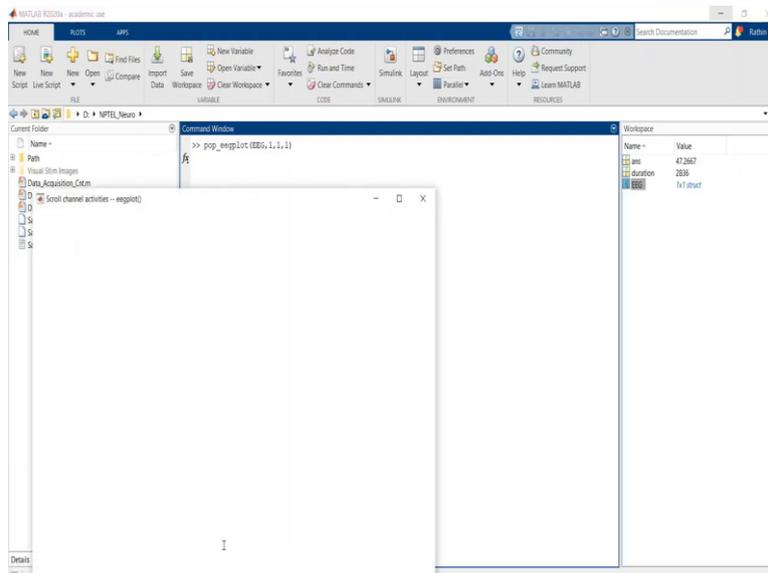
Name	Value
ans	47.2667
duration	2836
EEG	1x1 struct

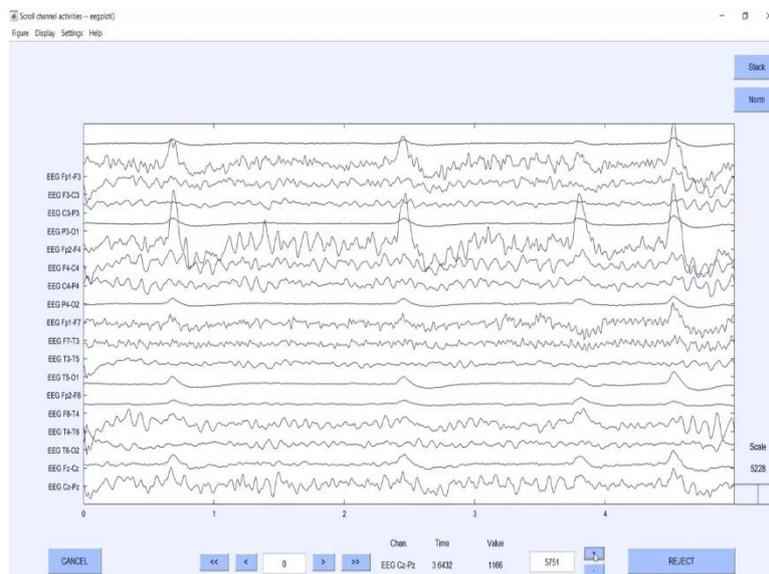
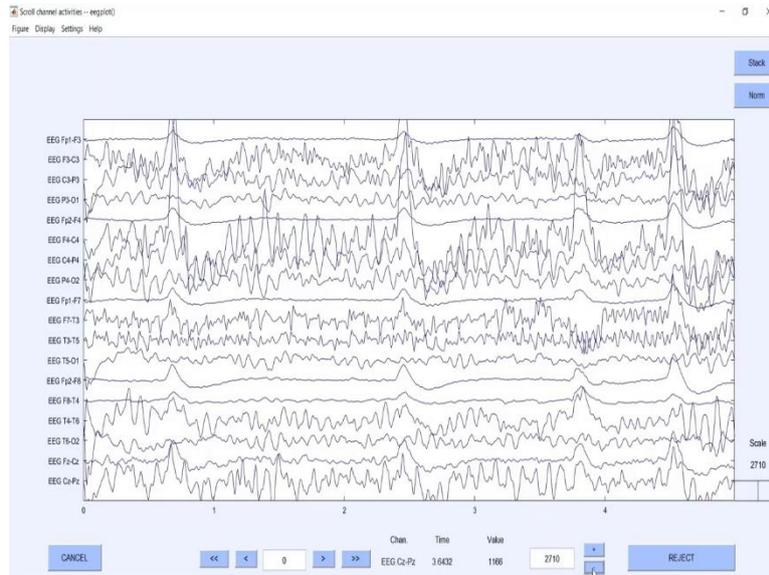


So, how you can plot this data? So, two ways. First, pop eegplot. If you do not know what is pop eegplot, go to the help of pop eegplot and see the different optional input require input

So, here I see a component, independent component analysis is a separate signal processing method to show any multivariate data as a addition of independent variables. So, you can have a look in detail if you are interested. Further you can see all this inputs. Generally, we will keep the inputs after EEG data structure as 1.

(Refer Slide Time: 45:29)

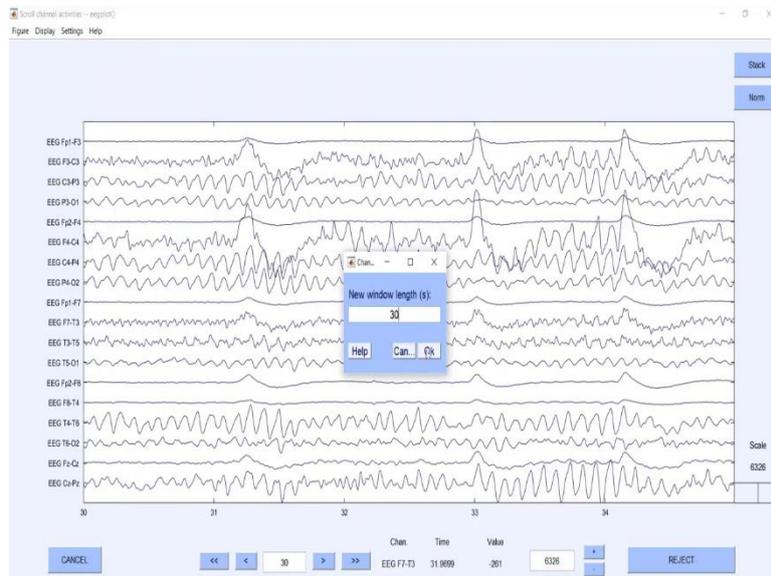
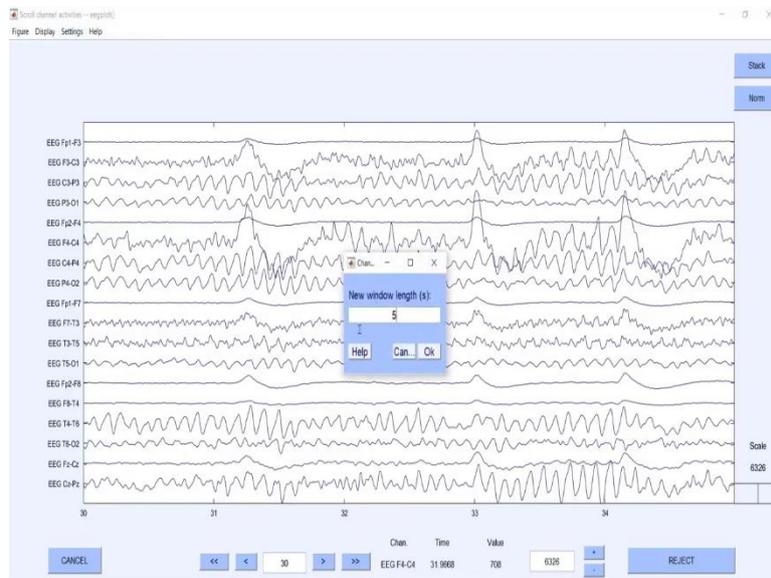


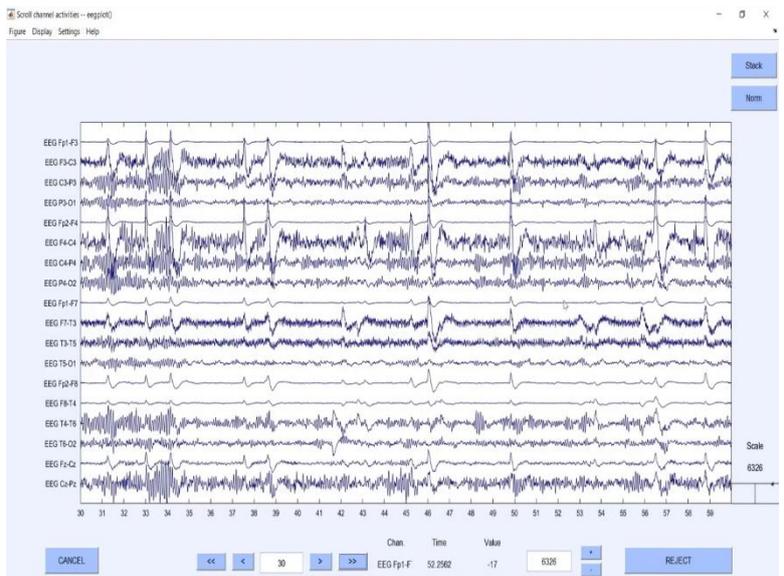
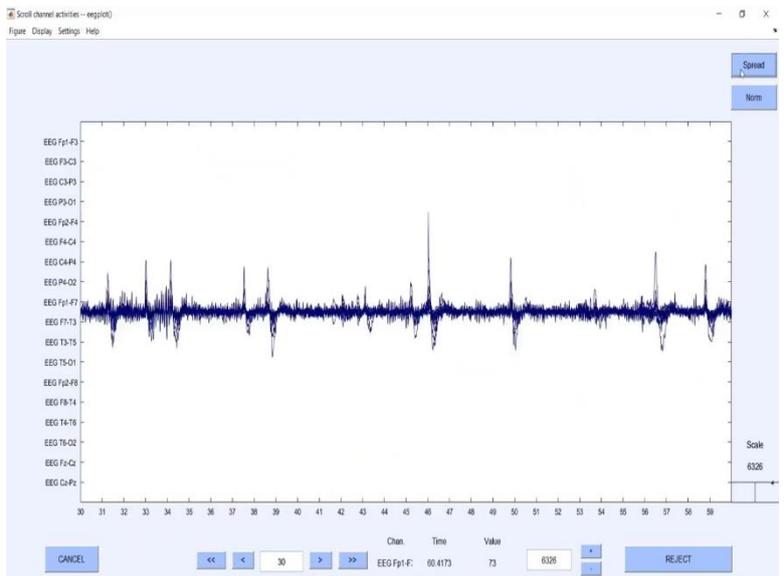
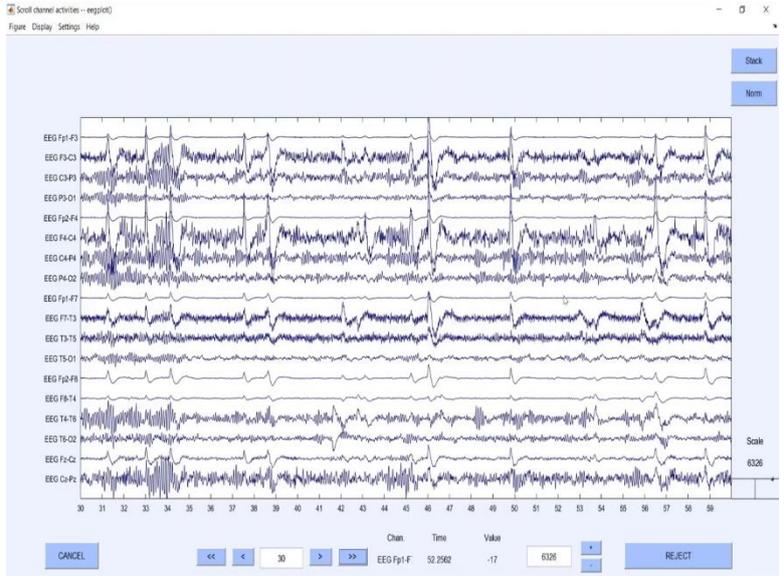


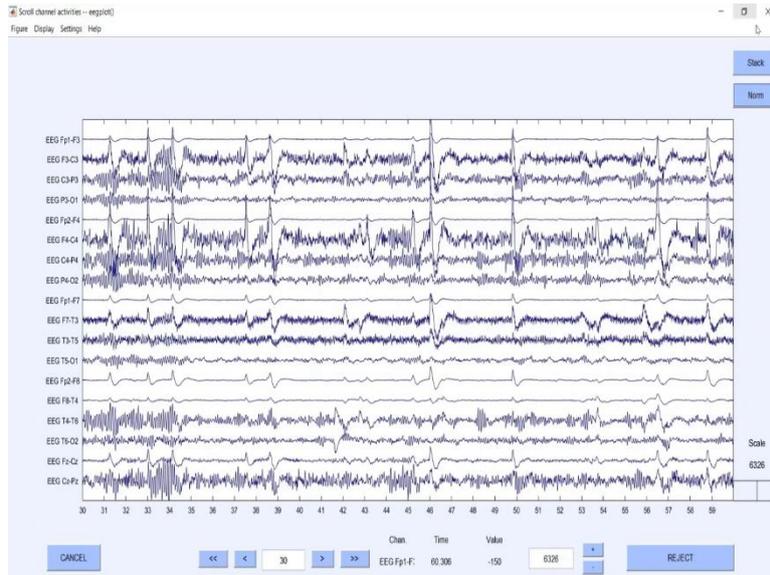
So, let us plot that pop eegplot. EEG all the other three inputs I will keep that one. I would encourage you to go to documentation and check, if you want to change you can change and play with the sample data set available online.

So, let us just see how it looks like. Yeah, the similar kind of graph you have seen when I have demonstrated EEGLAB using GUI and all this is nothing but a double banana which I explained in the experimental setup. You can, I hope you remember otherwise you can play with the y axis scale using this push button you can go travel scroll through the data.

(Refer Slide Time: 46:17)







If you want you can change your time axis from here. Let us say I want to see 30 seconds at a time, so you can do that. You can observe this based on your need. You can spread it, you can stack all the channels, you can spread all the channels, you can normalize all the channels, denormalized based on your requirement. This is one way to look at the images, one way to look at the biopotential graph rather.

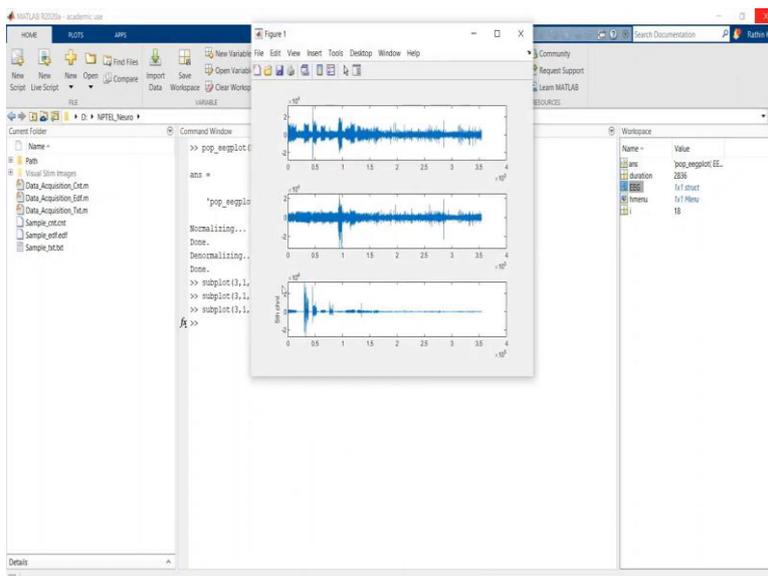
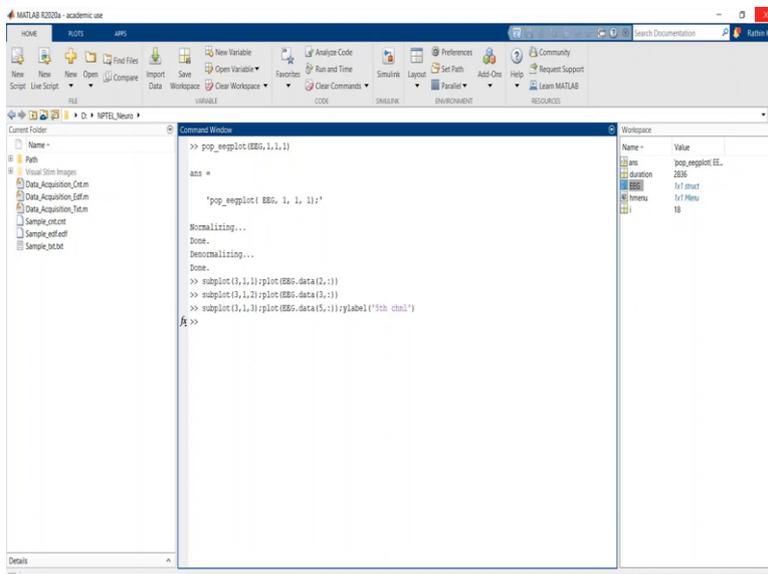
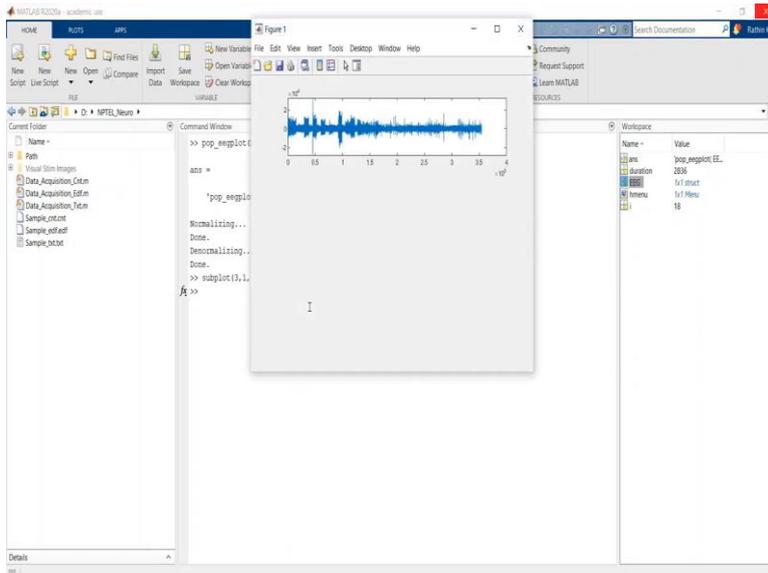
(Refer Slide Time: 46:45)

```

>> pop_eegplot(EEG,1,1,1);
ans =
'pop_eegplot(EEG, 1, 1, 1);'

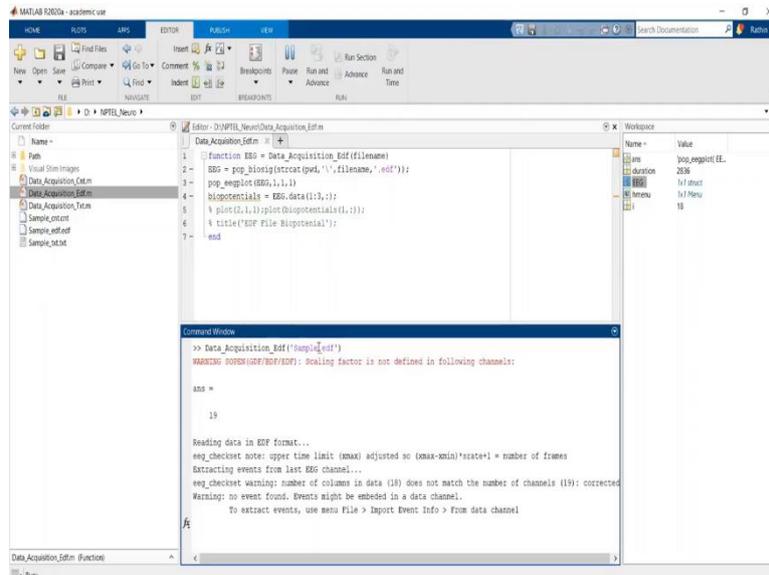
Normalizing...
Done.
Denormalizing...
Done.
>> subplot(3,1,1);plot(EEG.data(:,2:));

```





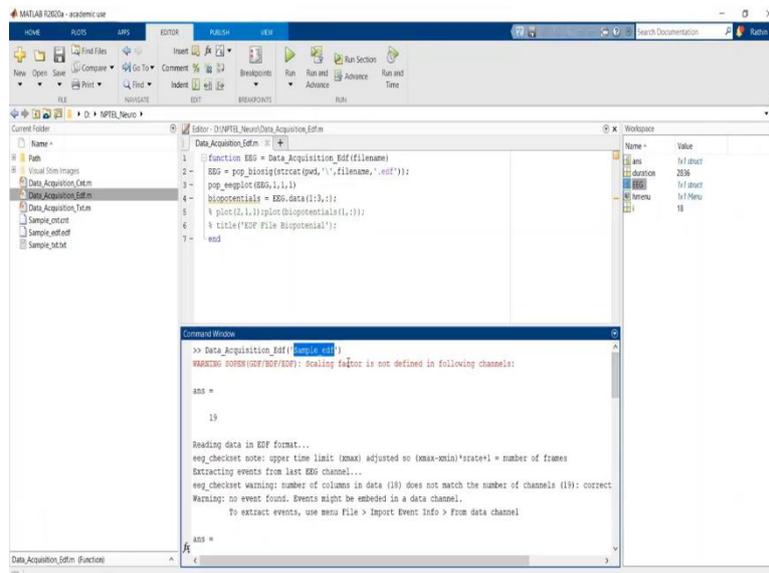
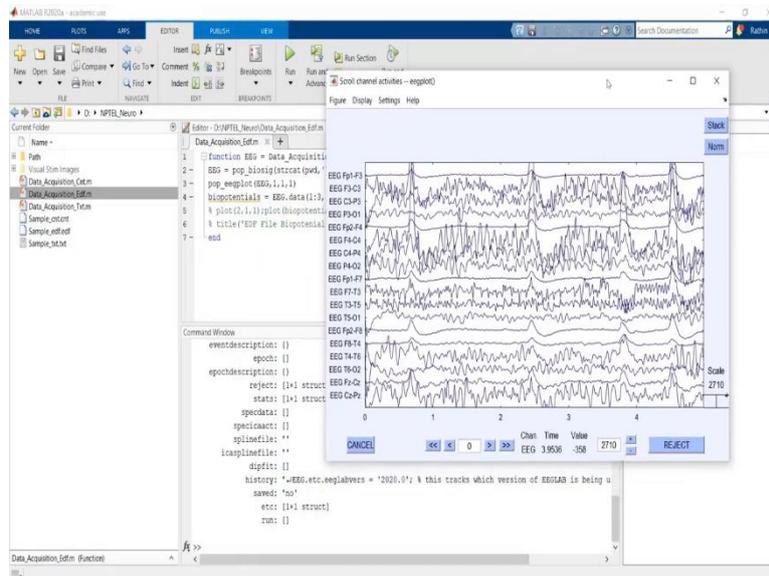




One more important thing I would like to tell all of you is that you have seen separately by writing a command window for each file, how you can generate your, how you can load your values and further process it, but it is advisable for batch processing you should have a script. So, like I said, I have made a script for almost each kind of data inputs.

So, whatever I am doing, I just showed you pop BIOSIG function, I will pass this file name as an input and I can further process it. I just only have to give this file name and it should me the final graph. Just for example, I will show you this one example here. We are talking about EDF. Here I am already concatenating it with an extension EDF. So, no need to write EDF. I just have to write, it should show me the final plot.

(Refer Slide Time: 49:43)



So, exactly the same plot whatever you have seen that has been generated using this function. So similarly, you can make your own script. This is a very simple and small example how you can generate your script.

(Refer Slide Time: 50:01)

```
>> Data_Acquisition_Edf('Sample.edf')
WARNING: SPPEN(SDF/BDF/EDF): Scaling factor is not defined in following channels:
ans =
    19

Reading data in EDF format...
seg_checker note: upper time limit (kmax) adjusted so (kmax-min)'rate' = number of frames
Extracting events from last EEG channel...
seg_checker warning: number of columns in data (18) does not match the number of channels (19): correct
Warning: no event found. Events might be embedded in a data channel.
To extract events, use menu File > Import Event Info > From data channel

ans =

'pop_event([ EEG; 1, 1, 1]);'

ans =

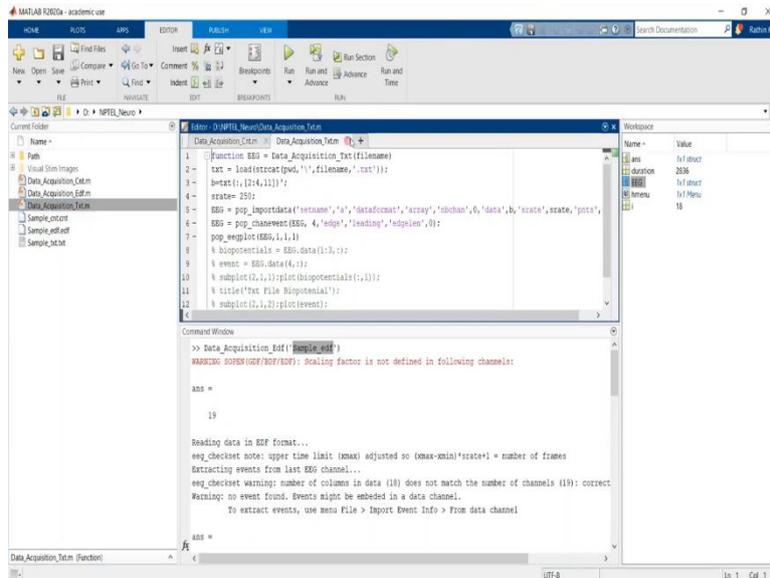
struct with fields:
    setname: 'EDF file'
    filename: ''
    filepath: ''
    subject: ''
    group: ''
    condition: ''
    session: []
    comments: 'Original file: D:\NPTEL_Neuro\Sample.edf'
```

```
function EEG = Data_Acquisition_Ctm(filename)
1  EEG = pop_loadset(struct('ed', filename, '.cm'));
2  EEG = pop_loadset(struct('ed', filename, '.cm'));
3  biopotential = EEG.data(1:3,:);
4  event = EEG.data(4,:);
5  pop_event(EEG,1,1);
6  end

>> Data_Acquisition_Edf('Sample.edf')
WARNING: SPPEN(SDF/BDF/EDF): Scaling factor is not defined in following channels:
ans =
    19

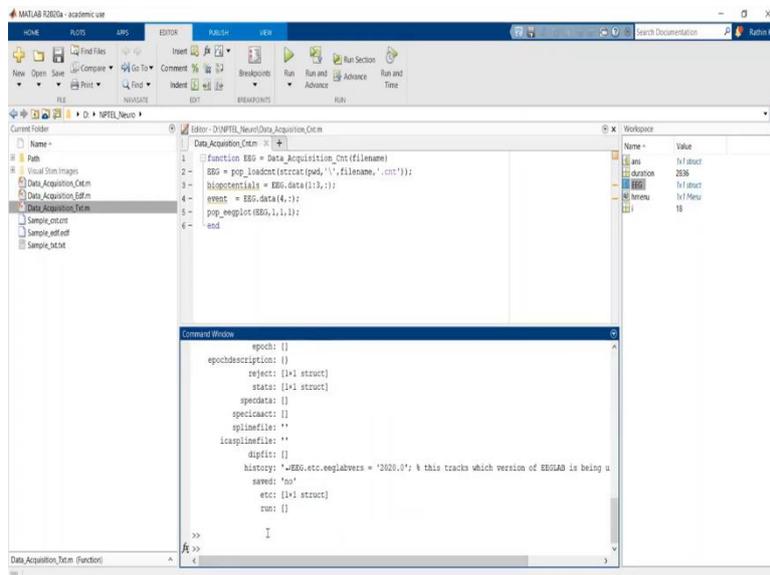
Reading data in EDF format...
seg_checker note: upper time limit (kmax) adjusted so (kmax-min)'rate' = number of frames
Extracting events from last EEG channel...
seg_checker warning: number of columns in data (18) does not match the number of channels (19): correct
Warning: no event found. Events might be embedded in a data channel.
To extract events, use menu File > Import Event Info > From data channel

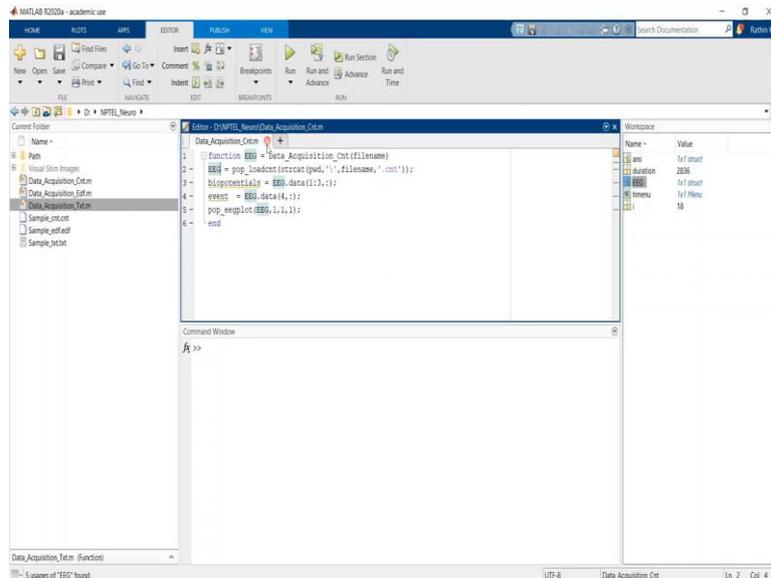
ans =
```



I have generated in all three cases, for cnt file as well as txt file. If I talk little more about that, then if I have a folder where I have a particular set of cnt files, then you can use, you just, you can make a provision in MATLAB, that you, have to just have to keep your input directory as an input and it will search for all the cnt files and all the cnt file gets loaded in a particular manner.

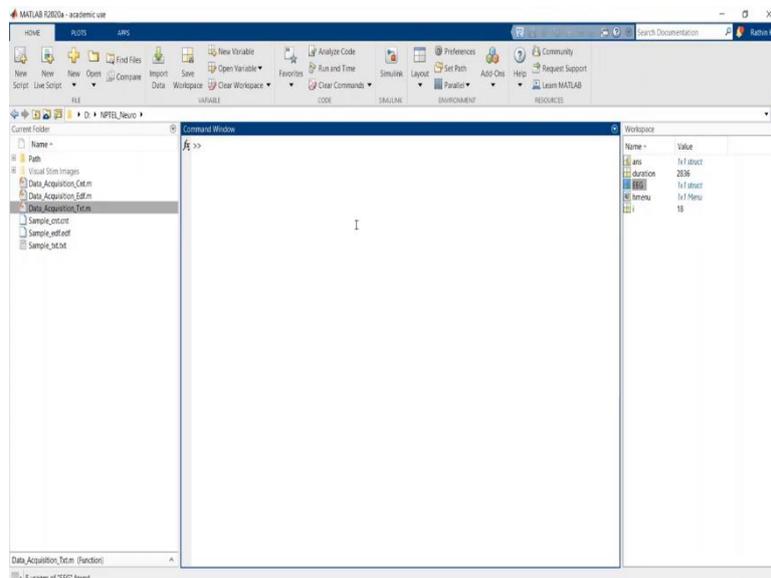
(Refer Slide Time: 50:32)





At that time, your output variable should be a function of your number of cnt file, otherwise everything gets overwrite in your EEG function. So, as in when you see and implement this, you will get more ideas of how you can process the multiple subject and batch files.

(Refer Slide Time: 50:54)



So, this is just a small illustration of how you can load the data, recorded using different EEG systems into MATLAB for further processing. In some other module, we will have a look at the basic steps of signal processing, what are the pre-processing step filtration, filtering and further steps related to event related potential.

The objective behind this demonstration is to give you idea or provide you a platform using which you can get actively involved and try to solve or try to process EEG related data

analysis by your own. If you have any questions related to this entire module, you are free to ask us in the forum. I wish all of you would have understood a bit and it will help you for further EEG related us signal processing which will be taught in the subsequent lectures. Till then, all of you please take care stay safe.