**Flow Analysis with Python**

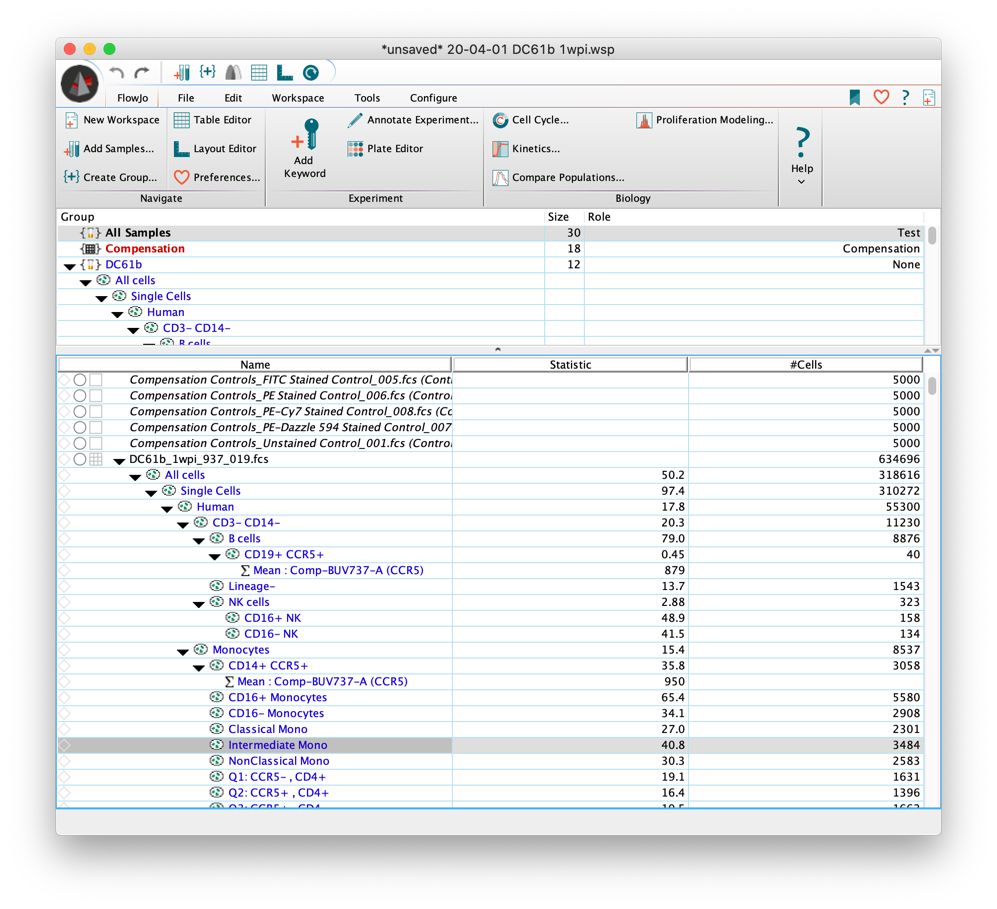
Goal: Export longitudinal data out of FlowJo and output csv files and plots

**Installing software and downloading code**

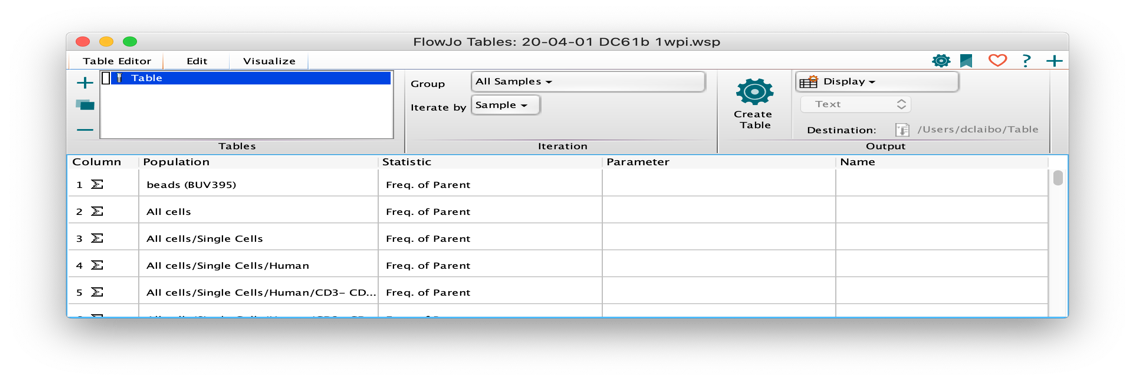
1. Download Anaconda and the latest version of python3 online
2. Create a folder anywhere and save the python files export-flowjo-figure and flowjo-mouse-clicker in them. The python files are available on LabArchives in SOPS>Data Analysis Code.

**Exporting FlowJo data**

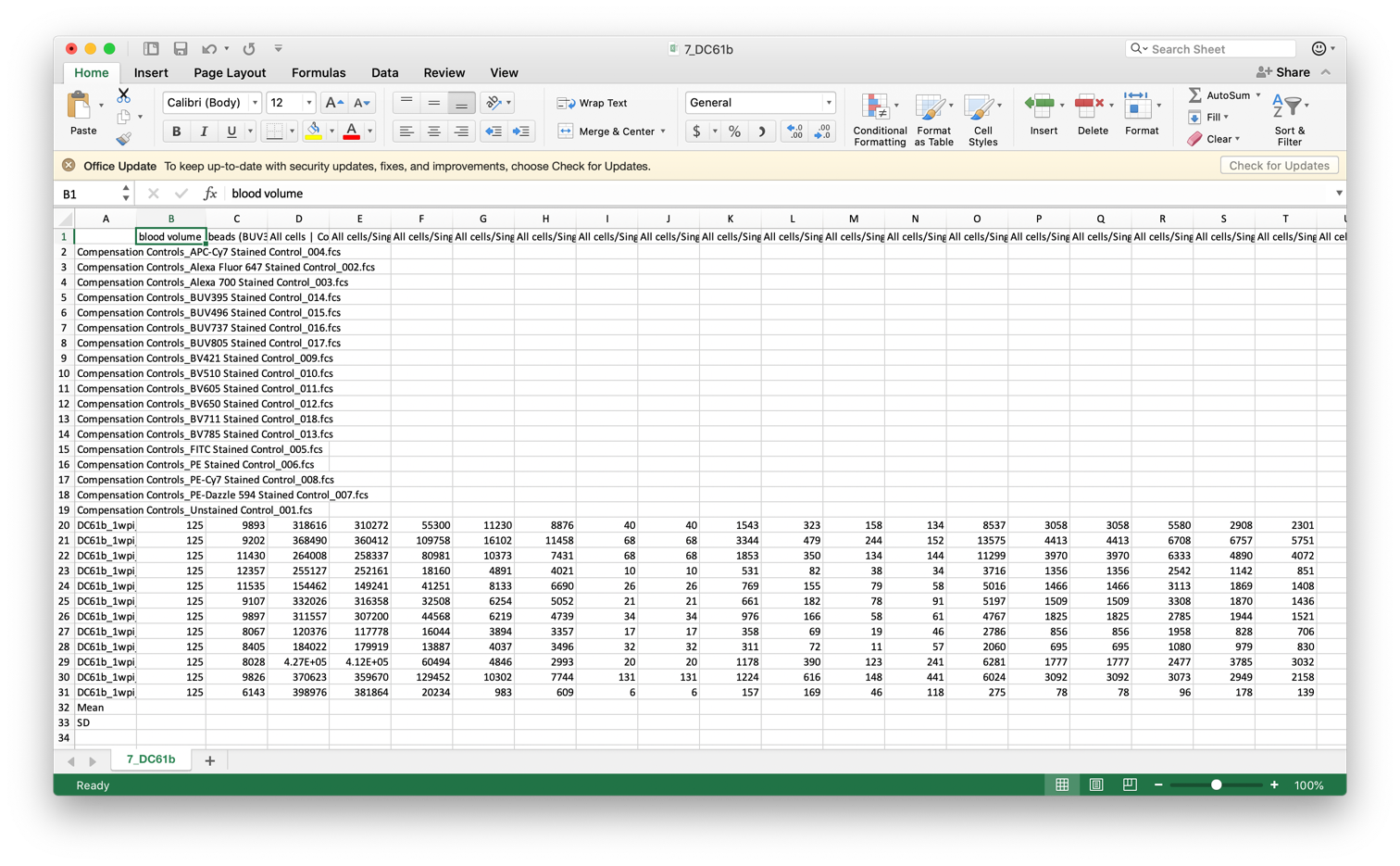
1. After gating your samples, your FlowJo window should look something like this:



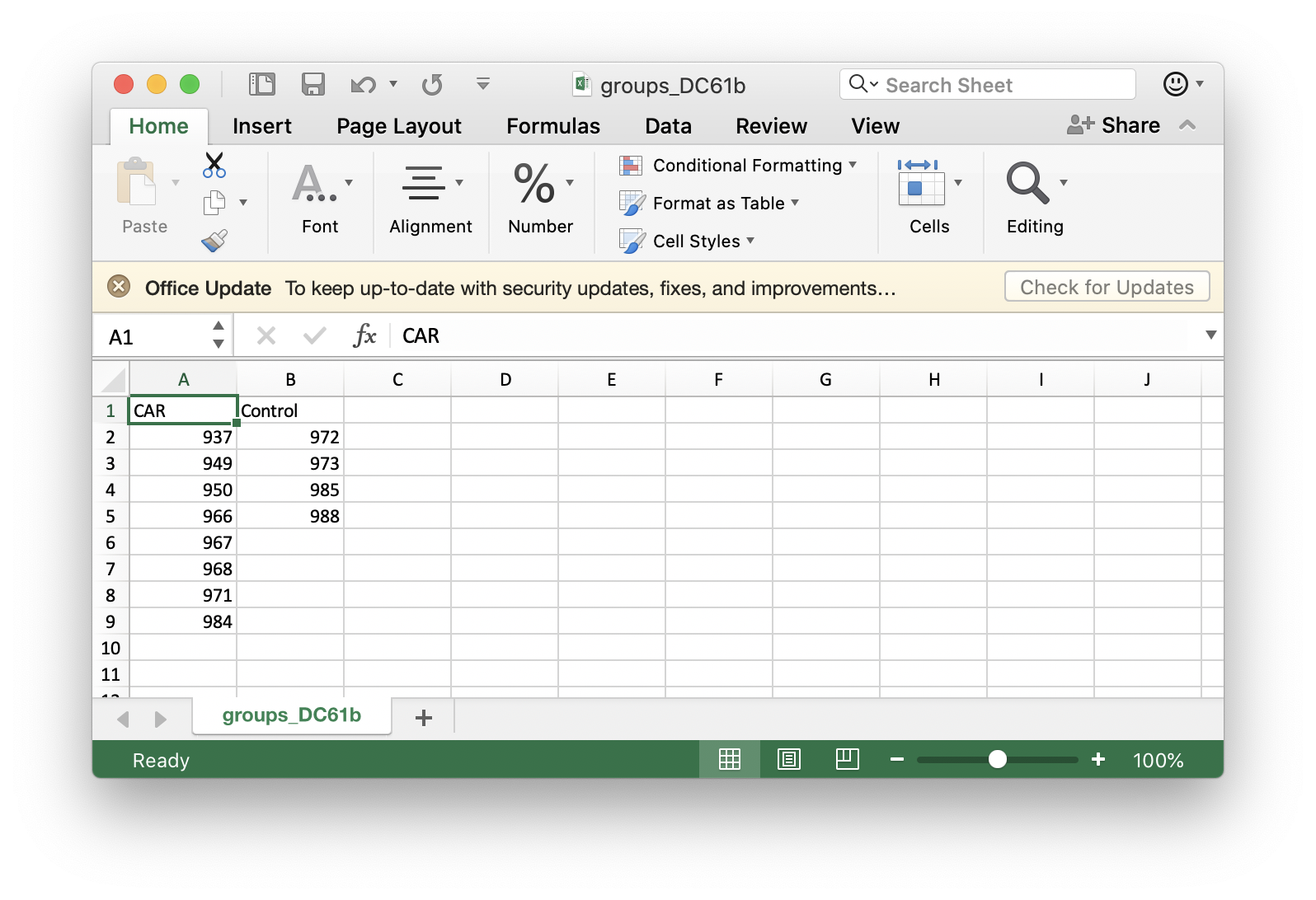
1. Open the table editor by clicking on the grid icon at the very top of the window or with shortcut (command-T)
2. Drag and drop countbright beads into table. It is important that it is in the first row.
3. Drag and drop all remaining gates in. To reduce error, drag and drop in every single gate you have made. Your window should look like this.

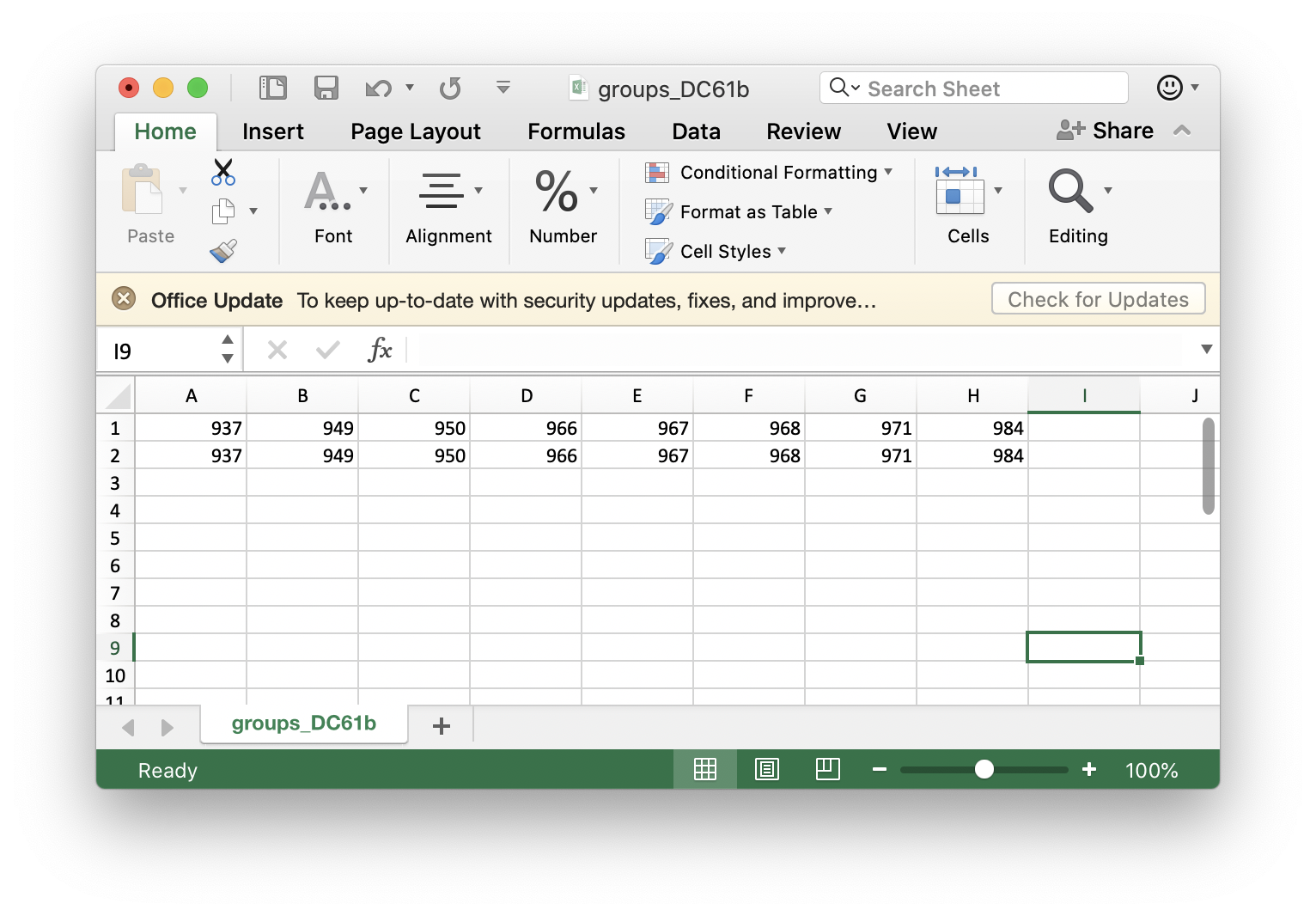


1. Change all statistics to Count. Do this manually or with flowjo-auto-clicker code (Protocol at end)
2. Export the table by selecting To File, file type as CSV, and name the file in this format: x\_studyname where x = days post infection and study name something like DC61b.
3. Save the file in the folder with the python code
4. Repeat for all timepoints. Make sure the gates are in the same order every time
5. Open the csv file for the first time point
6. Insert a column named blood volume and input the blood volumes

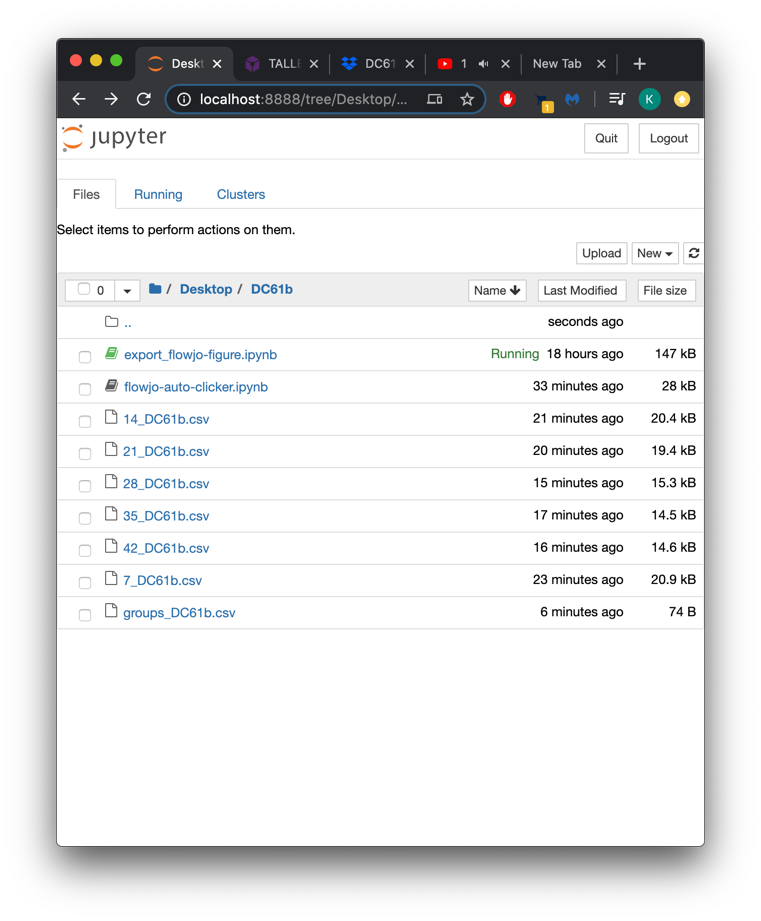


1. Confirm that blood volume is in column B and beads are in column C
2. Confirm that all sample names have the study name at the beginning. The script will not recognize it as a sample without the study name
3. Save as csv
4. Repeat for all other timepoints
5. OPTIONAL: If you want the script to group your samples into experimental groups, create a csv file with the group names as column names. If you want the script to plot out individual samples, only have one sample per group. Save this file as groups\_studyname. EX. groups\_DC61b. Make sure groups is all lowercase.

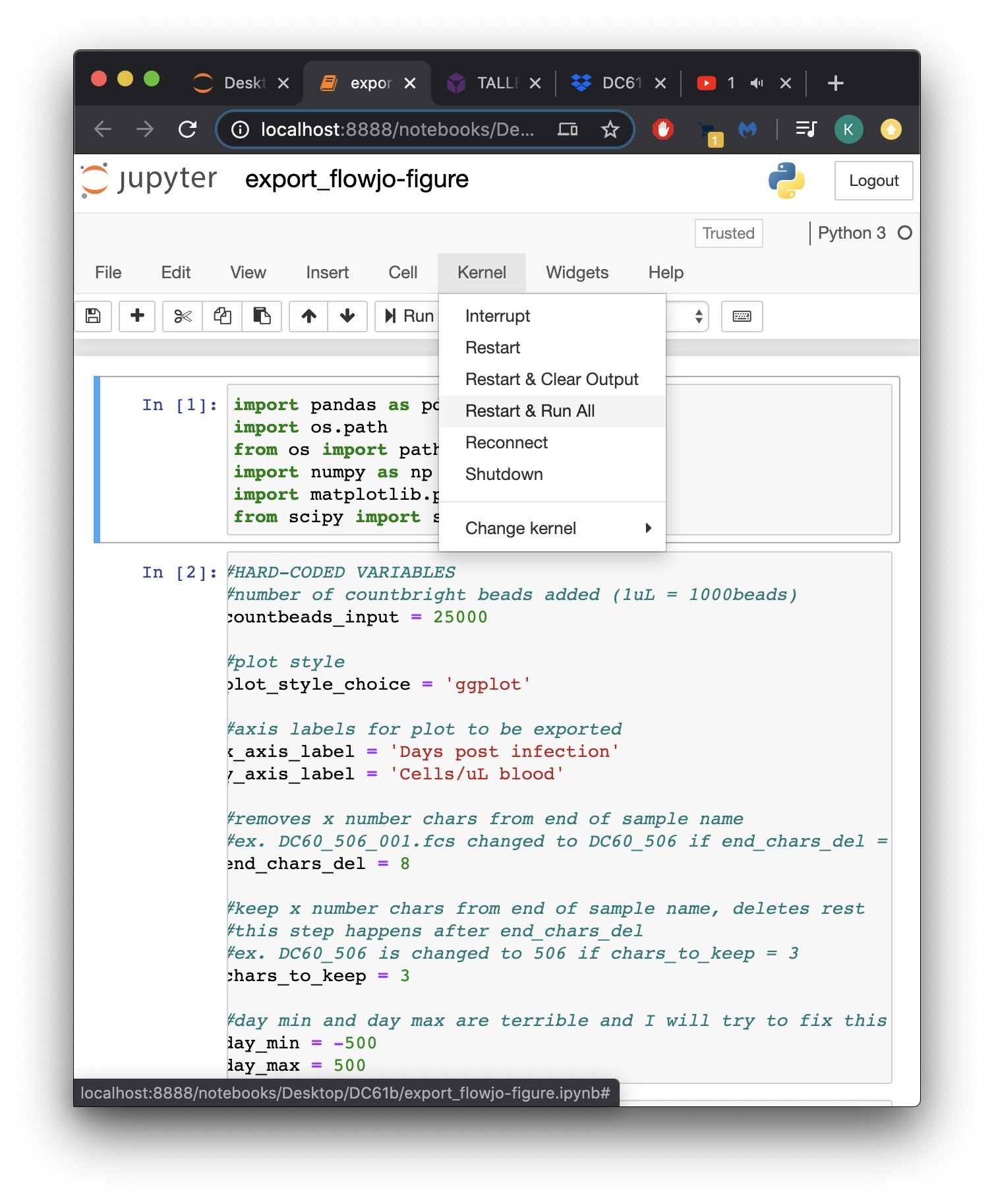




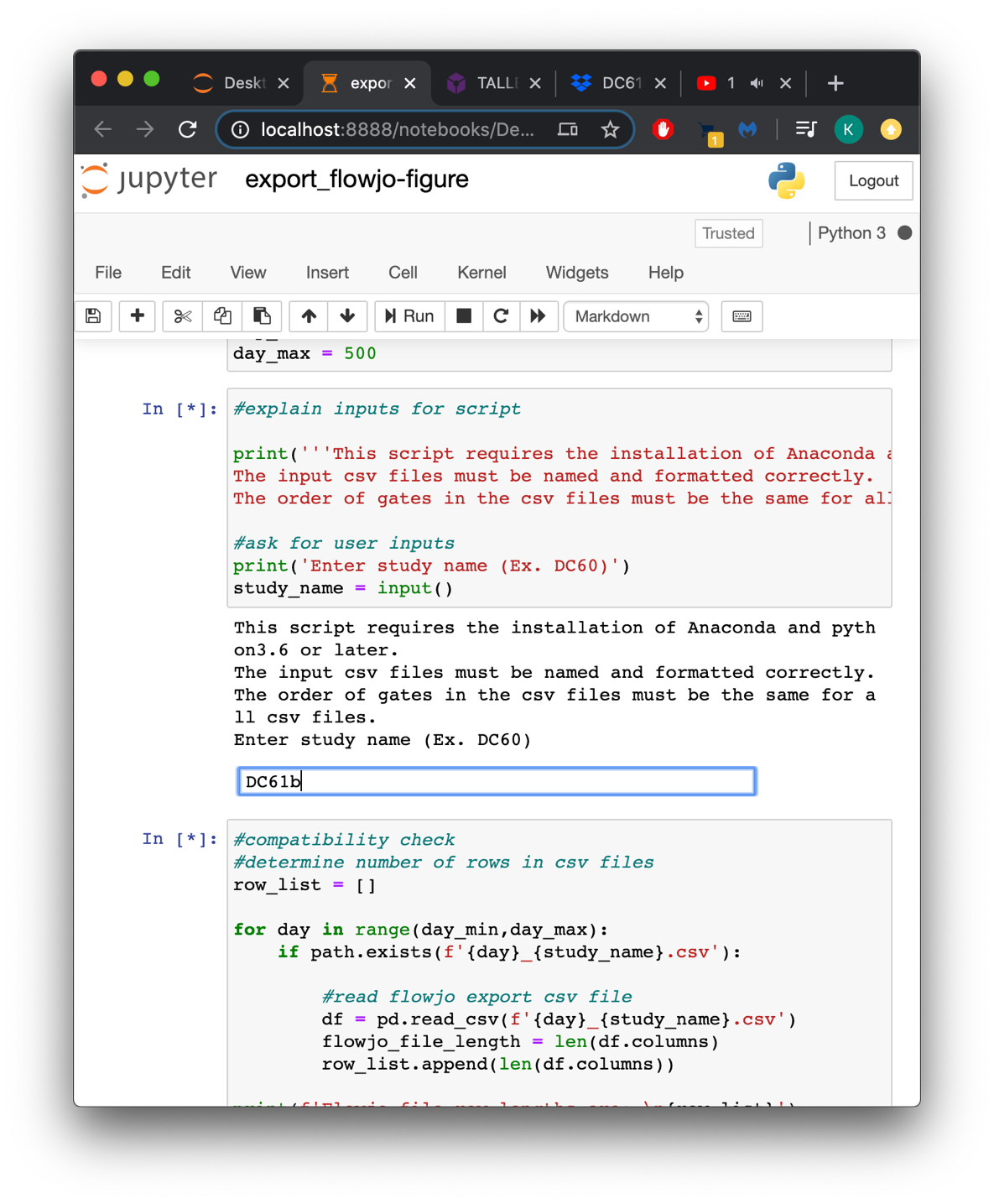
1. Open Anaconda and click Launch Jupyter Notebook. This will launch Jupyter Notebook in your browser. Navigate to the folder with the code and csv files.



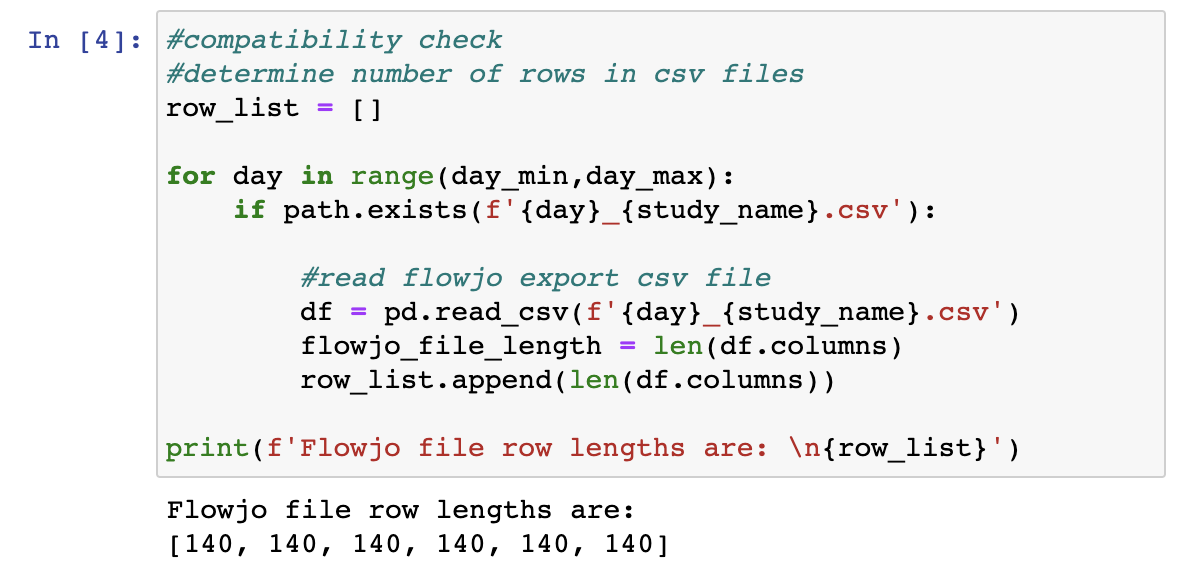
1. Click on export-flow-figure to open the script in a new window
2. Click on Kernel and hit Restart and Run



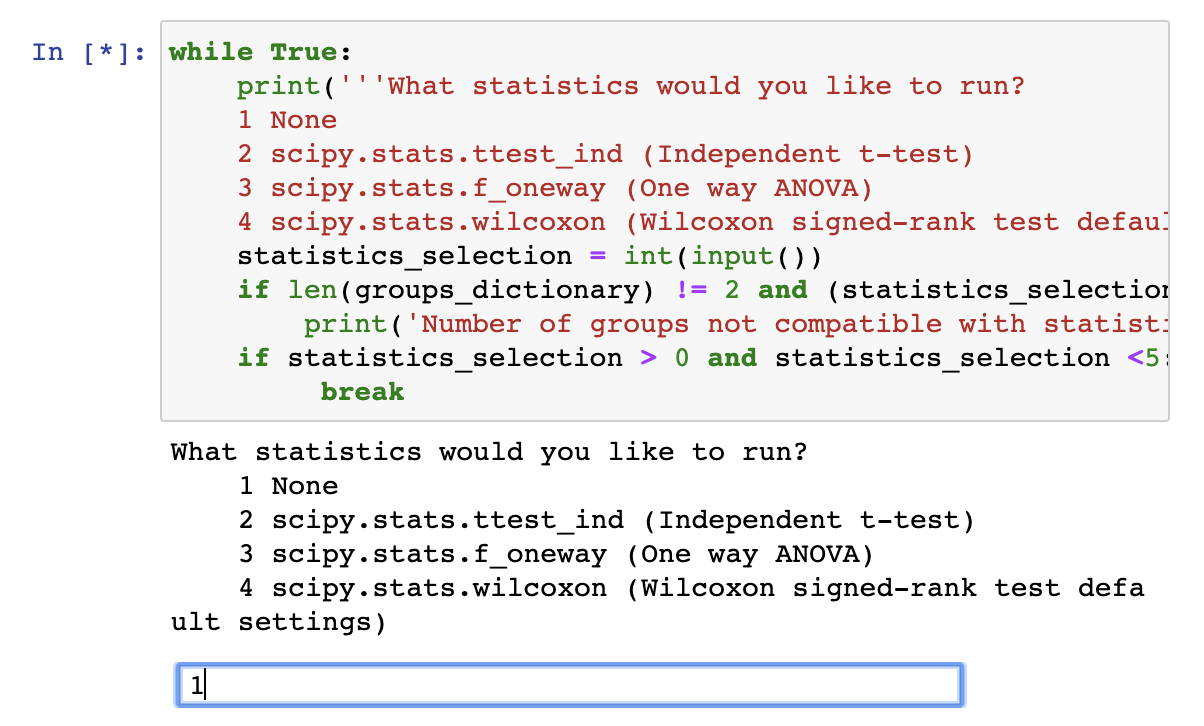
1. After hitting run, scroll down to the third cell (block of code) and input the study name ex. DC61b



1. Scroll down to the fourth cell which runs a compatibility check. This cell returns the number of gates in each csv file. Confirm that they are all equal



1. Scroll down to the sixth cell which asks if statistics should be run. All stats packages are run with scipy. This is the last input before the csv files and plots are generated.

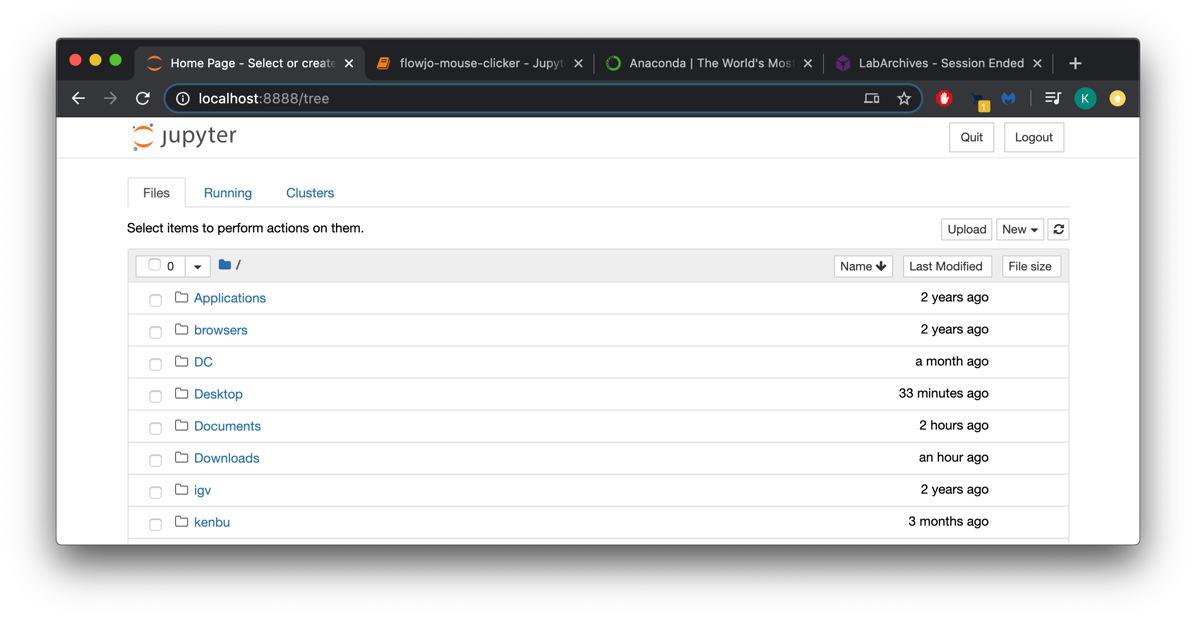


1. The gate names are used as the file names. This program uses the gate names from the very last time point to name the files.

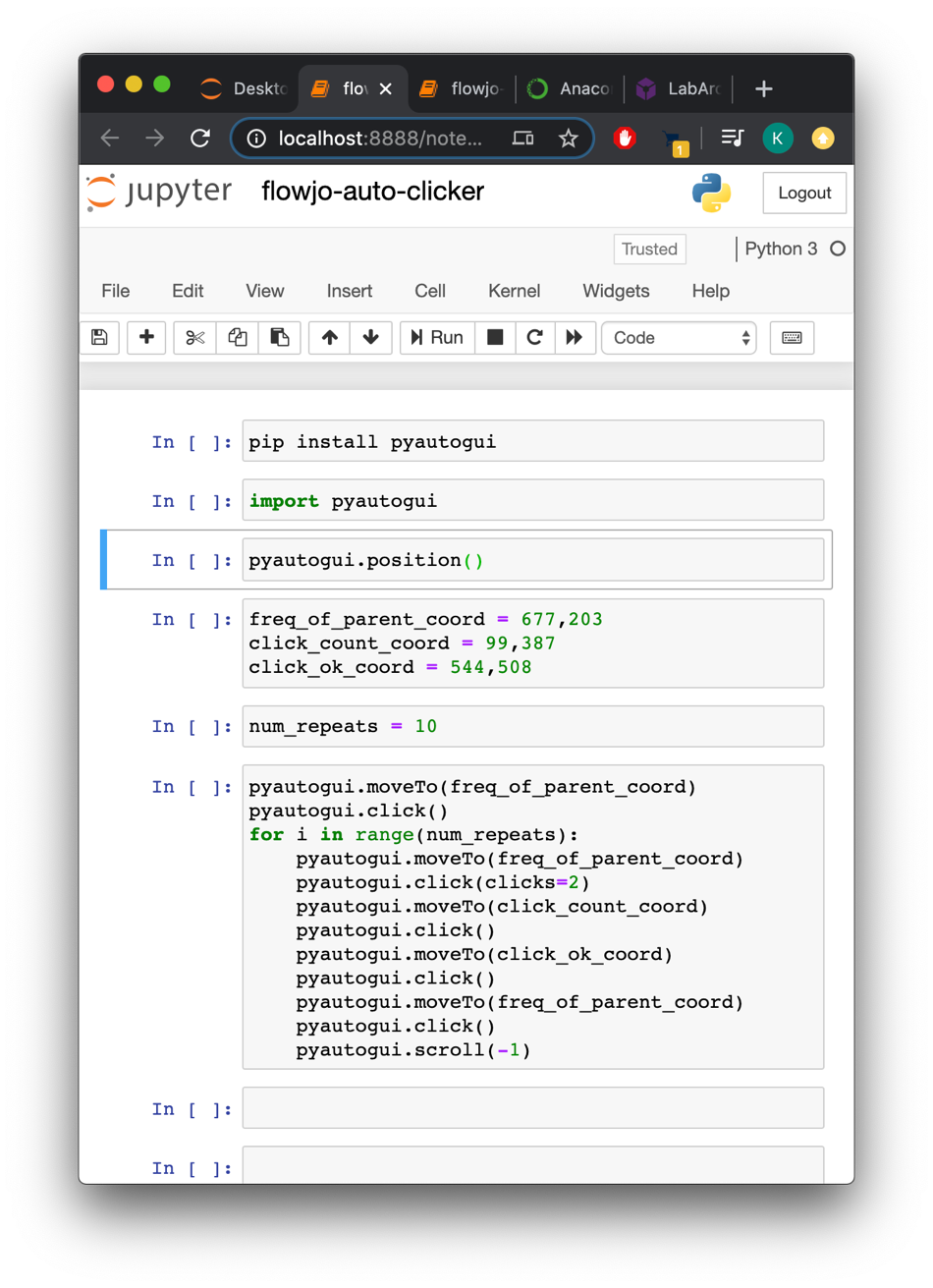
**Run flowjo-auto-clicker**

This is a script that controls the cursor. Its function is to double click freq. of parent, click count, click ok, scroll down the table viewer by 1, and repeat x number of times.

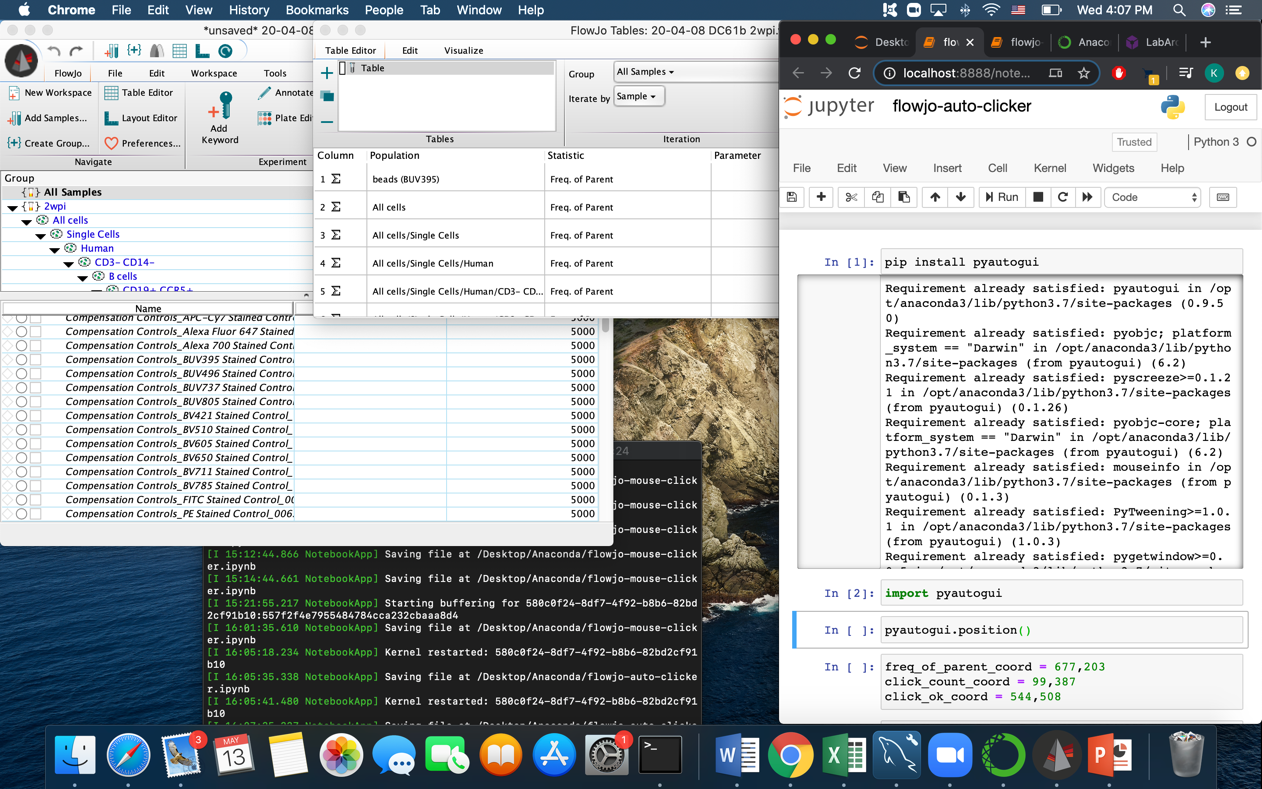
1. Open Anaconda and launch Jupyter Notebooks. It will open in your browser.



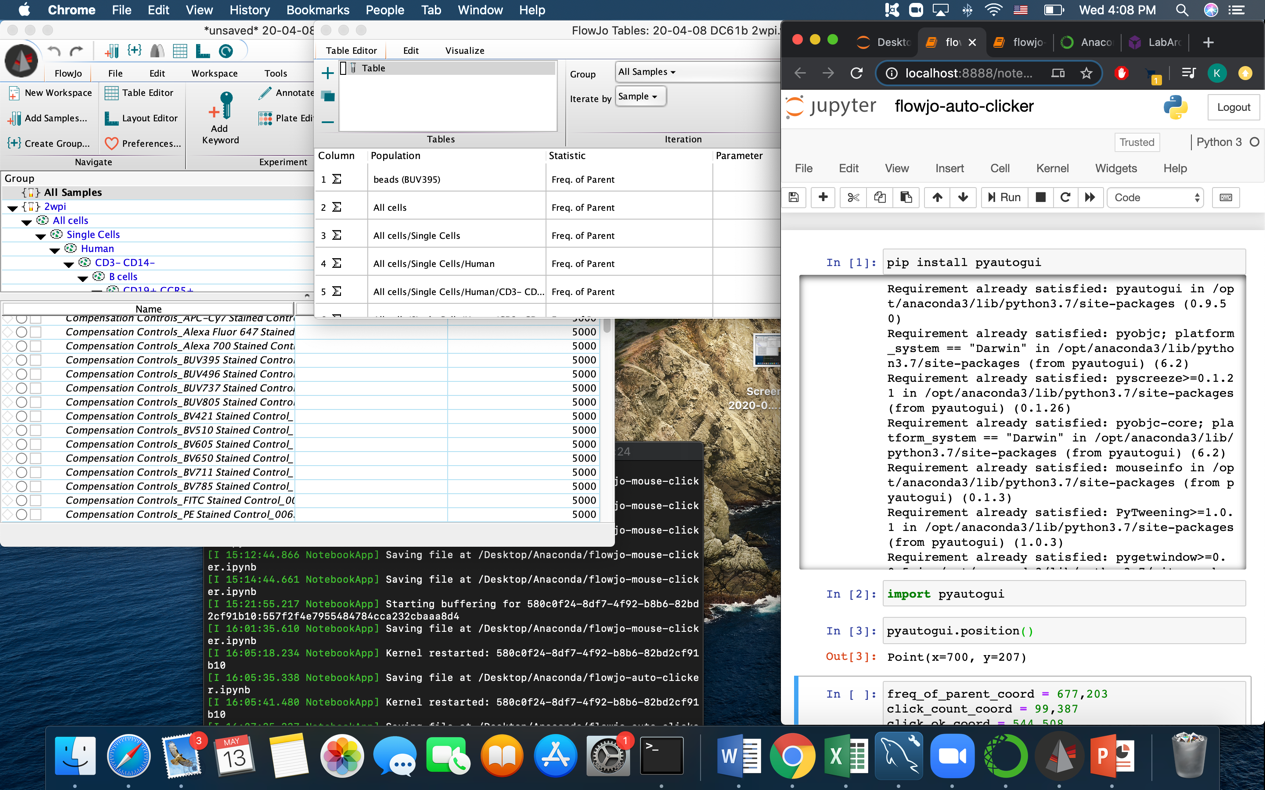
1. Navigate to the folder you saved flowjo-auto-clicker in and open file



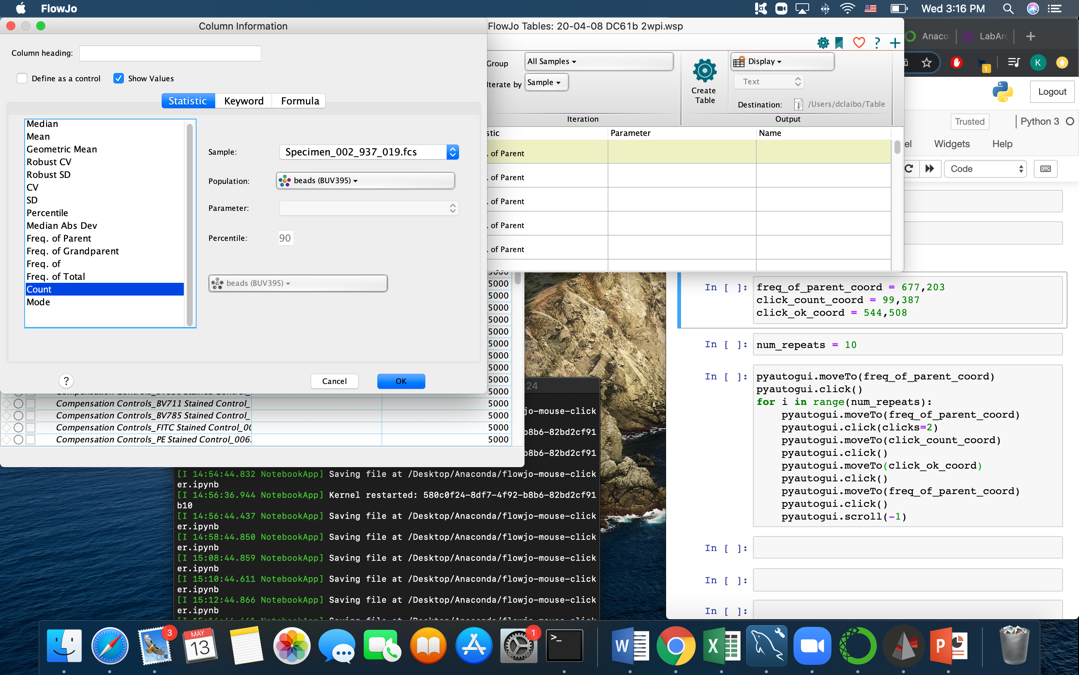
1. Click on the first grey box of code. Jupyter calls this a cell. When the box is selected, it should be highlighted. Run the first cell by clicking run or with shift+enter
2. Run the second cell. Your screen should look like this.



1. Click on the third cell which contains the code pyautogui.position()
2. Move mouse to freq of parent
3. Hit shift+enter to run third cell



1. The code will return the cursor x,y coordinates of freq of parent
2. Type in the x,y coordinates of freq of parent into freq\_of\_parent\_coord
3. Repeat steps 6-10 to get and save the x,y coordinates of count and ok.



1. After updating all of the coordinates, run that cell to save
2. Adjust num\_repeats, run that kernel to save (suggest 2 for test run)
3. BEFORE RUNNING LAST CELL: If you want to interrupt the program, move your cursor to the top left corner of the screen
4. Run last cell