W1005
Intro to CS and Programming in MATLAB

Project Lecture

Fall 2014

http://www.cs.columbia.edu/~vovsha/w1005

Fraud Analysis of 2000 Presidential Election Results in Florida using Matlab

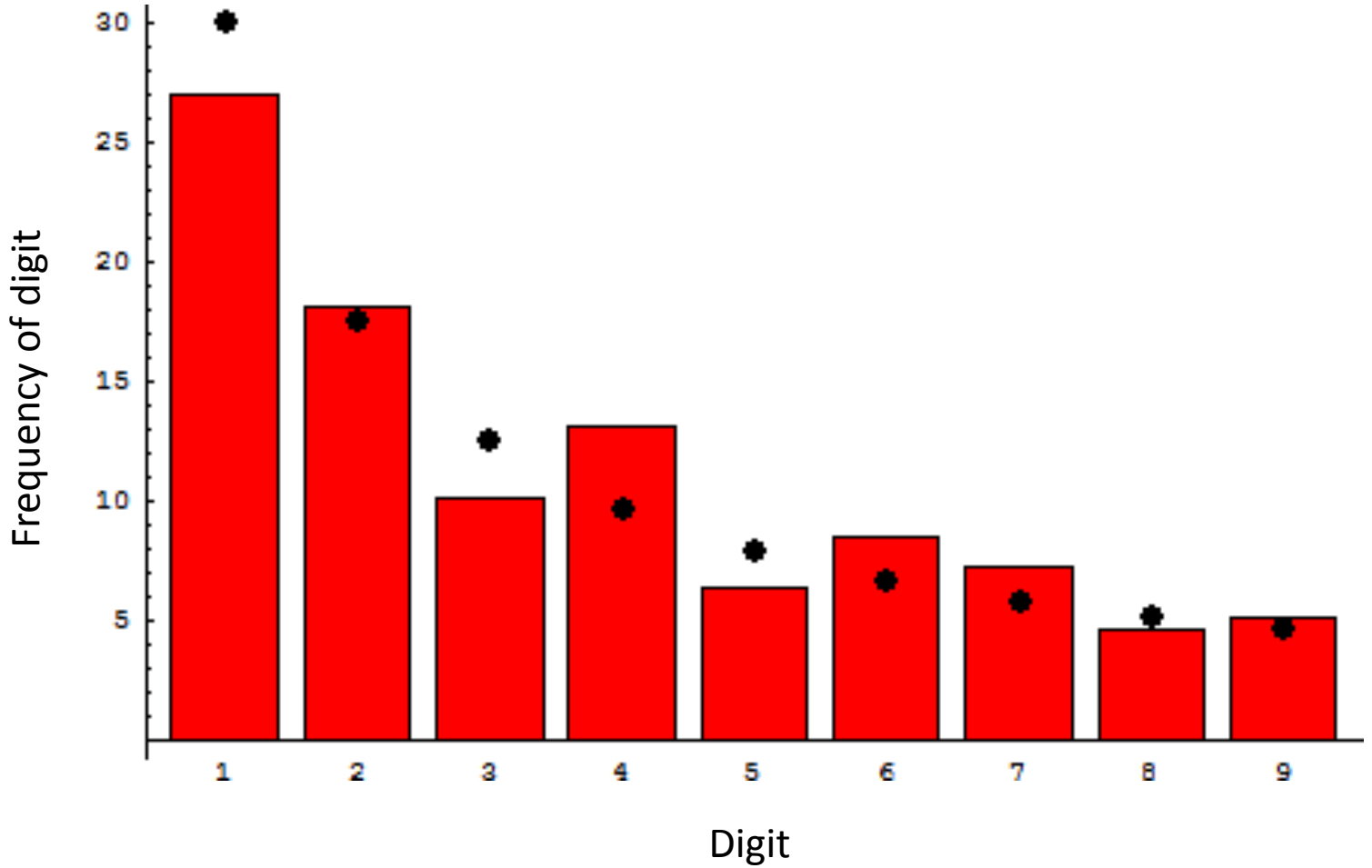
Background

- Election forensics: field of political science that uses statistical analysis to examine election data and detect irregular patterns that may be indicative of fraud
- Benford's Law: mathematical law that states that the frequency distribution digits in real-life sources of data is not uniformly distributed. Instead, smaller digits (1, 2, 3) appear more frequently than larger digits (7, 8, 9)

Application of Benford's Law

- Applied to leading digits
  - Distribution follows form P(d) = $\log_{10}(\frac{d+1}{d})$
- Applied to second digits
  - $r_j = \sum_{k=1}^{9} \log_{10}(1 + (10 * k + j)^{-1}$.

Example of Benford's Law applied to world's countries population



Project Set Up

1. Read data from file (found online) to Matlab
2. Isolate only data that I want to analyze
   - First isolate vote counts, then 1^{st} and 2^{nd} digits
3. Perform statistical tests
4. Plot results

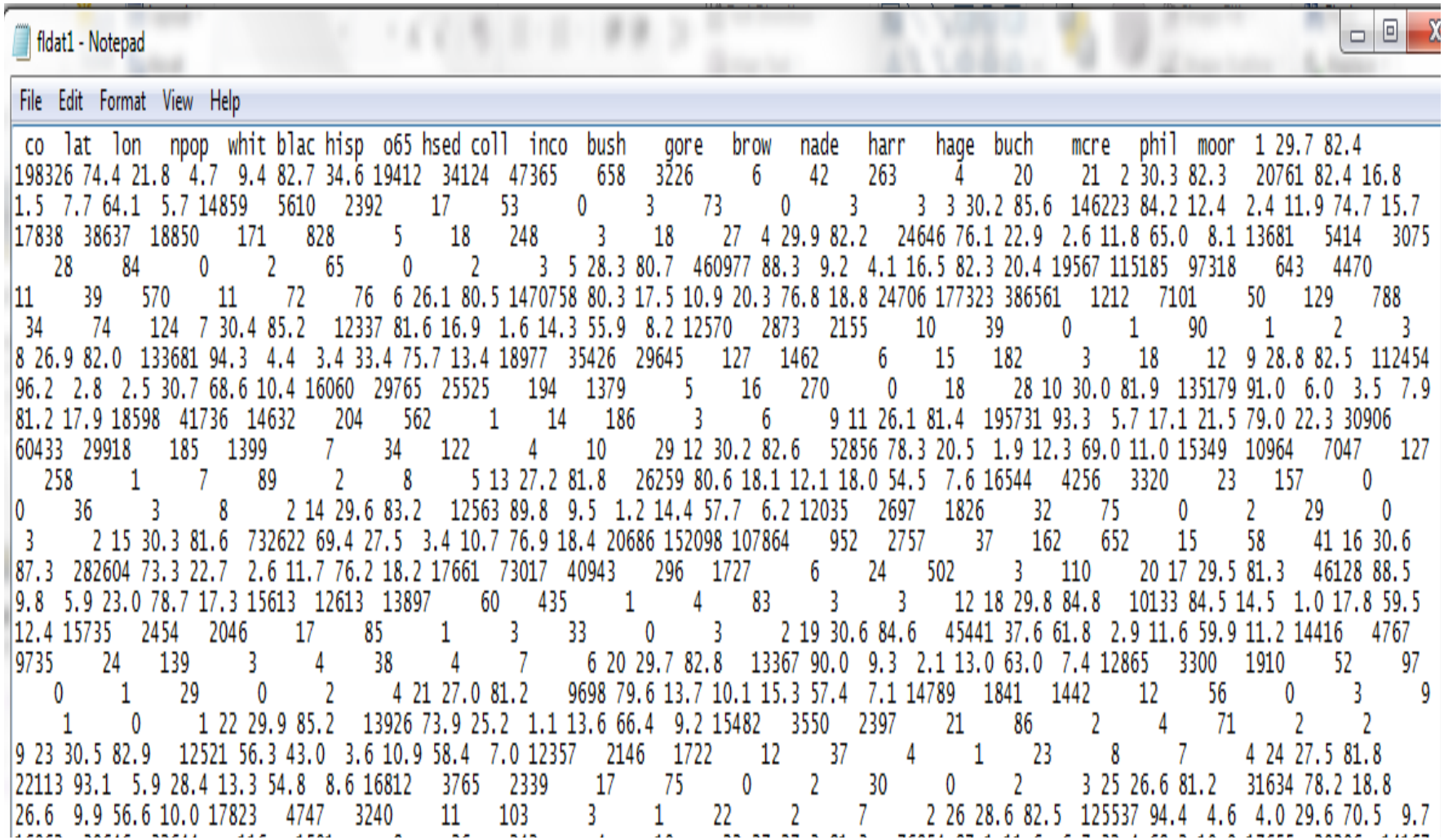
Implementation in Matlab

FILE EDIT NAVIGATE BREAKPOINTS RUN

benford_analysis.m | read_file.m

```
1   % Alex Pan UNI: aqp2000
2   % Final Project
3   function benford_analysis(fname)
4
5   [first,second] = read_file(fname);
6
7   a = input('Would you like to analyze first or second digits? First = 1; second = 2.\n');
8
9   if a == 1
10      [freq,exp] = create_dist(first,1);
11      chi2test(freq,exp,8);
12      ks_test(freq);
13      plot_cdf(freq,exp);
14  elseif a == 2
15      [freq,exp] = create_dist(second,2);
16      chi2test(freq,exp,9);
17      plot_cdf(freq,exp);
18  else
19      a = input('Invalid input. Please enter 1 or 2.\n');
20  end
21
```

A look at the data file


fldat1 - Notepad

File Edit Format View Help

```
 co  lat  lon   npop  whit blac hisp  o65 hsed coll  inco  bush    gore   brow   nade   harr   hage  buch    mcre   phil  moor  1 29.7 82.4
198326 74.4 21.8  4.7   9.4 82.7 34.6 19412  34124  47365    658   3226      6     42    263      4     20     21  2 30.3 82.3   20761 82.4 16.8
1.5  7.7 64.1  5.7 14859   5610   2392     17     53      0     3     73      0      3      3  3 30.2 85.6  146223 84.2 12.4  2.4 11.9 74.7 15.7
17838  38637  18850    171    828      5     18    248      3     18     27  4 29.9 82.2   24646 76.1 22.9  2.6 11.8 65.0  8.1 13681   5414   3075
    28     84      0      2     65      0      2      3  5 28.3 80.7  460977 88.3  9.2  4.1 16.5 82.3 20.4 19567 115185  97318    643   4470
11     39    570     11     72     76  6 26.1 80.5 1470758 80.3 17.5 10.9 20.3 76.8 18.8 24706 177323 386561   1212   7101     50    129    788
 34     74    124  7 30.4 85.2   12337 81.6 16.9  1.6 14.3 55.9  8.2 12570   2873   2155     10     39      0      1     90      1      2      3
8 26.9 82.0  133681 94.3  4.4  3.4 33.4 75.7 13.4 18977  35426  29645    127   1462      6     15    182      3     18     12  9 28.8 82.5  112454
96.2  2.8  2.5 30.7 68.6 10.4 16060  29765  25525    194   1379      5     16    270      0     18     28 10 30.0 81.9  135179 91.0  6.0  3.5  7.9
81.2 17.9 18598  41736  14632    204    562      1     14    186      3      6      9 11 26.1 81.4  195731 93.3  5.7 17.1 21.5 79.0 22.3 30906
60433  29918    185   1399      7     34    122      4     10     29 12 30.2 82.6   52856 78.3 20.5  1.9 12.3 69.0 11.0 15349  10964   7047    127
   258      1      7     89      2      8      5 13 27.2 81.8   26259 80.6 18.1 12.1 18.0 54.5  7.6 16544   4256   3320     23    157      0
0     36      3      8      2 14 29.6 83.2   12563 89.8  9.5  1.2 14.4 57.7  6.2 12035   2697   1826     32     75      0      2     29      0
 3      2 15 30.3 81.6  732622 69.4 27.5  3.4 10.7 76.9 18.4 20686 152098 107864    952   2757     37    162    652     15     58     41 16 30.6
87.3  282604 73.3 22.7  2.6 11.7 76.2 18.2 17661  73017  40943    296   1727      6     24    502      3    110     20 17 29.5 81.3   46128 88.5
9.8  5.9 23.0 78.7 17.3 15613  12613  13897     60    435      1      4     83      3      3     12 18 29.8 84.8   10133 84.5 14.5  1.0 17.8 59.5
12.4 15735   2454   2046     17     85      1      3     33      0      3      2 19 30.6 84.6   45441 37.6 61.8  2.9 11.6 59.9 11.2 14416   4767
9735     24    139      3      4     38      4      7      6 20 29.7 82.8   13367 90.0  9.3  2.1 13.0 63.0  7.4 12865   3300   1910     52     97
    0      1     29      0      2      4 21 27.0 81.2    9698 79.6 13.7 10.1 15.3 57.4  7.1 14789   1841   1442     12     56      0      3      9
     1      0      1 22 29.9 85.2   13926 73.9 25.2  1.1 13.6 66.4  9.2 15482   3550   2397     21     86      2      4     71      2      2
9 23 30.5 82.9   12521 56.3 43.0  3.6 10.9 58.4  7.0 12357   2146   1722     12     37      4      1     23      8      7      4 24 27.5 81.8
22113 93.1  5.9 28.4 13.3 54.8  8.6 16812   3765   2339     17     75      0      2     30      0      2      3 25 26.6 81.2   31634 78.2 18.8
26.6  9.9 56.6 10.0 17823   4747   3240     11    103      3      1     22      2      7      2 26 28.6 82.5  125537 94.4  4.6  4.0 29.6 70.5  9.7
```

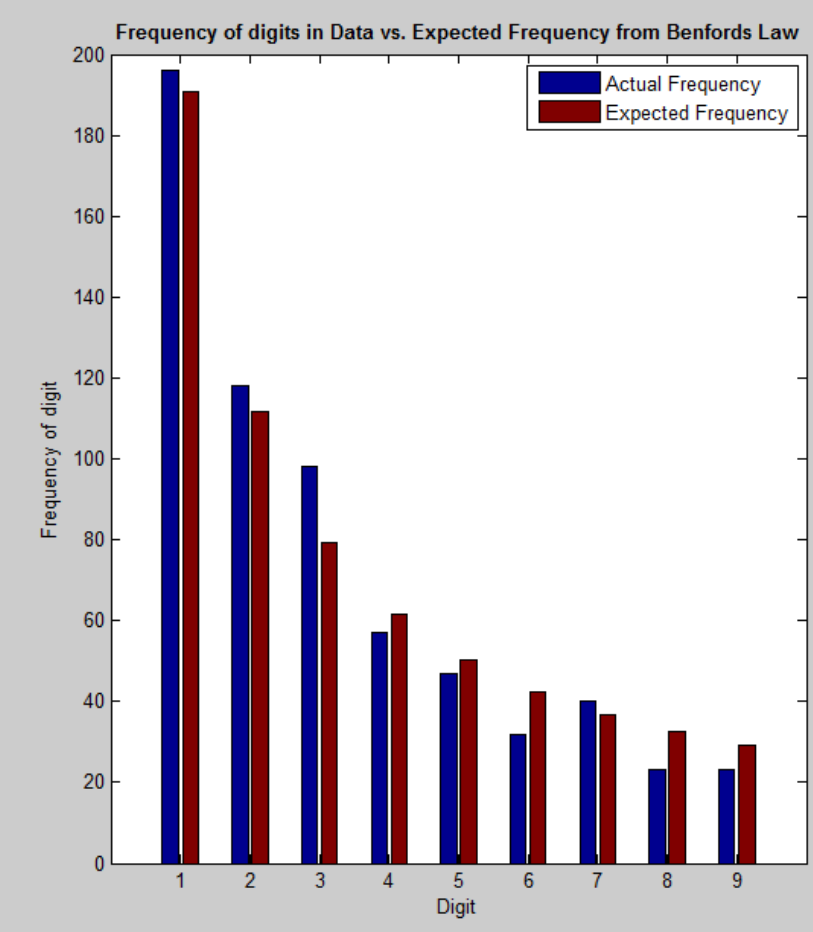
Reading Input File/storing data

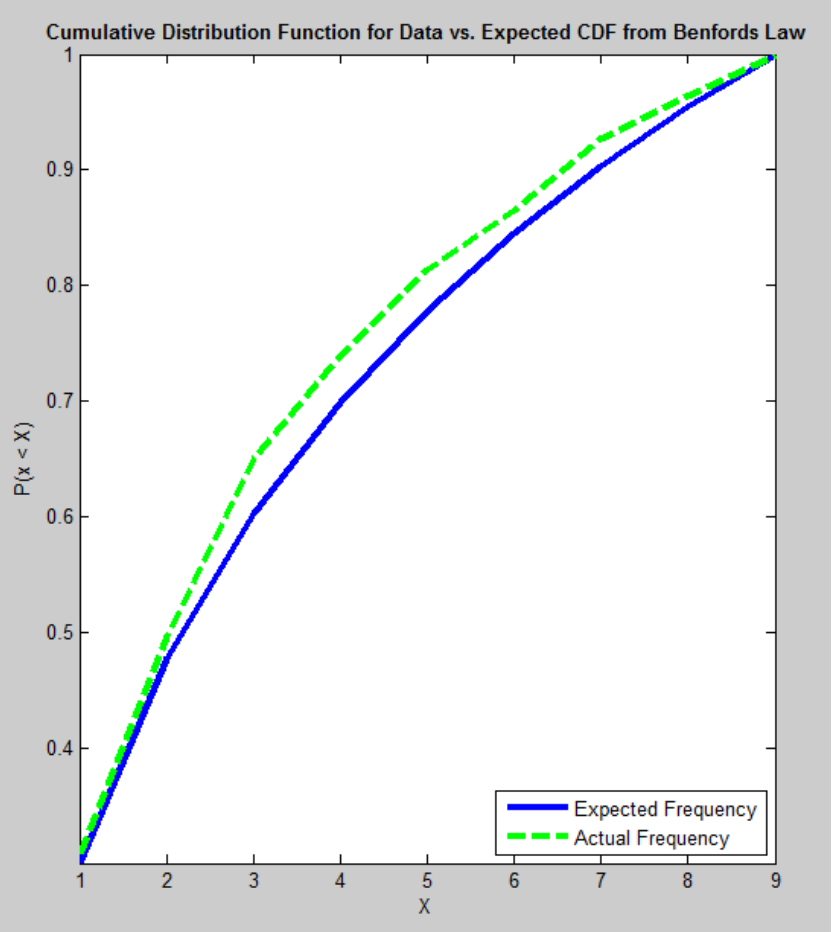
- Found election data online as a tab-delimited file
- Used textscan because of the file type that I used (.txt)
- Isolated the data that I wanted to analyze
  - Removed all 0s (ie if a candidate didn't receive any votes)
  - Removed all #s <10 for 2nd digit analysis
- This was the most difficult task for me

Statistical Analysis

- Used statistical toolbox, which can be downloaded from the Mathworks website for free as part of your student license
- Difficulty here is in choosing which statistical tests to use
  - Chi square test: compares observed values to expected values. Expected values generated from formulas found in literature
  - Kolmogorov-Smirnov Test: compares the theoretical cumulative distribution and expected cumulative distribution

Results from 1st digit analysis





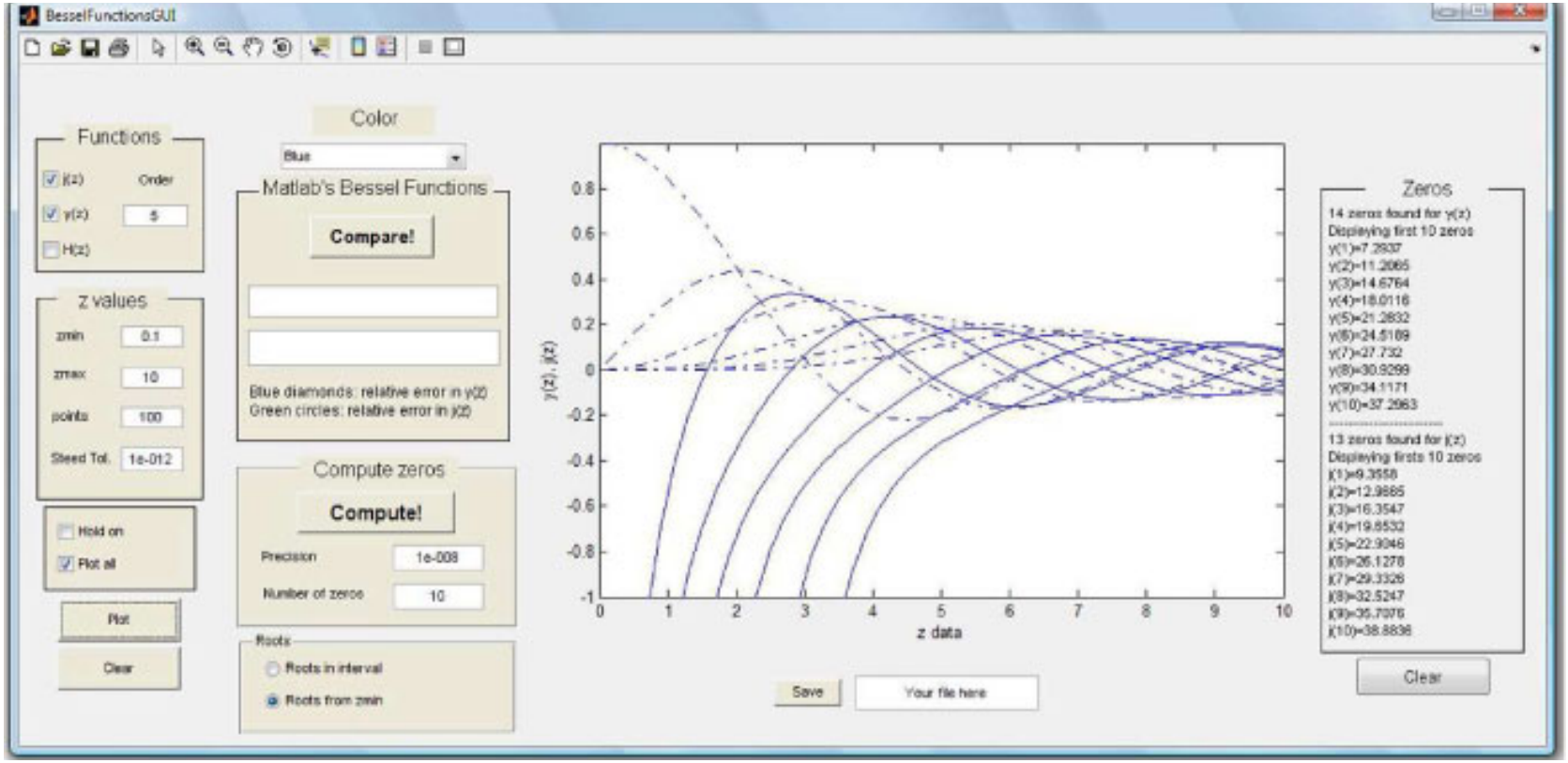
Findings

- In analysis of both leading and second digits, there was insufficient evidence to show that the data did not come from a Benford distribution

How to Make a GUI

What is a GUI?

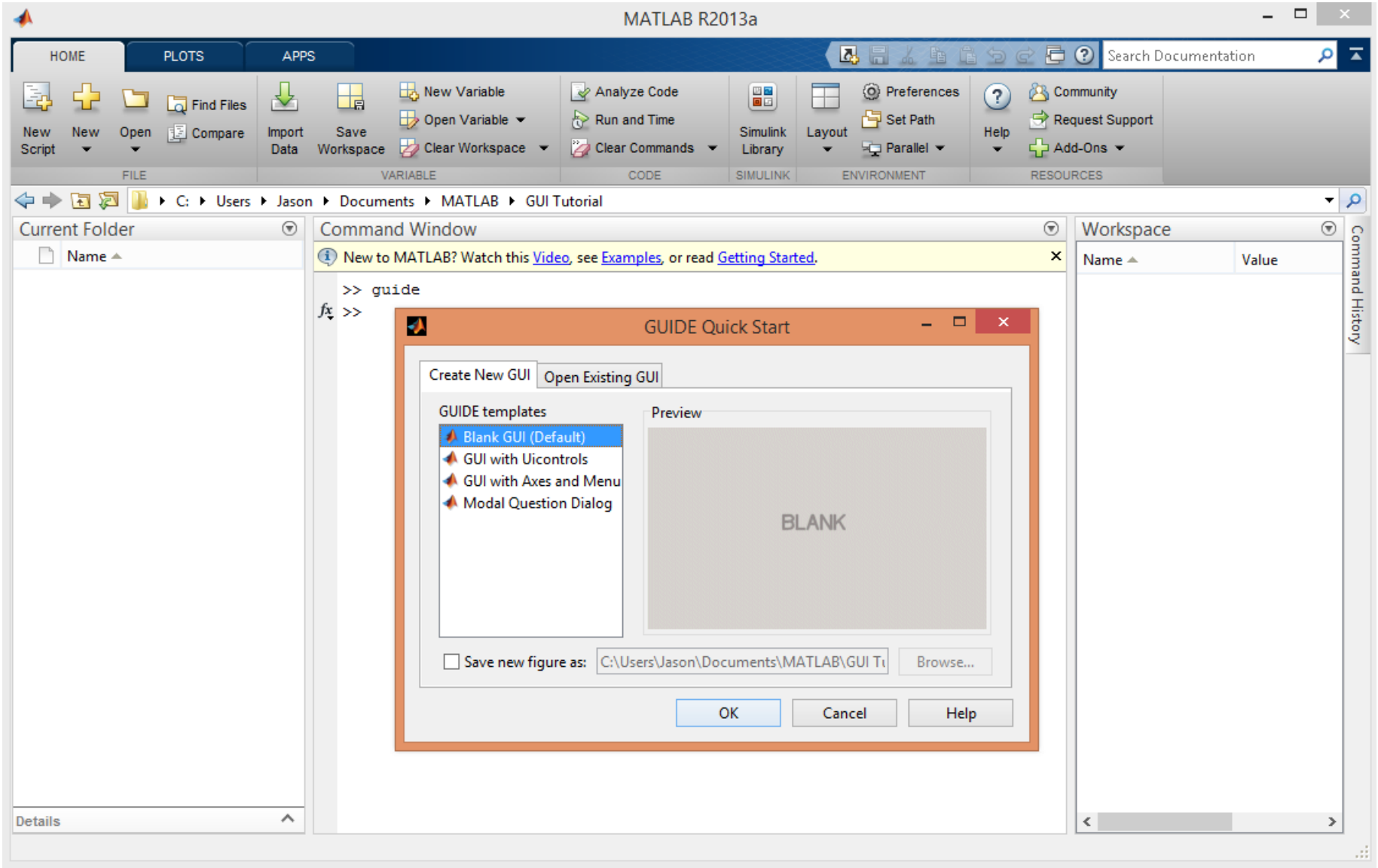
- Graphical user interface
- Point-and-click interface that allows users to run a program
- Allows others to use your code without writing code



Making the GUI

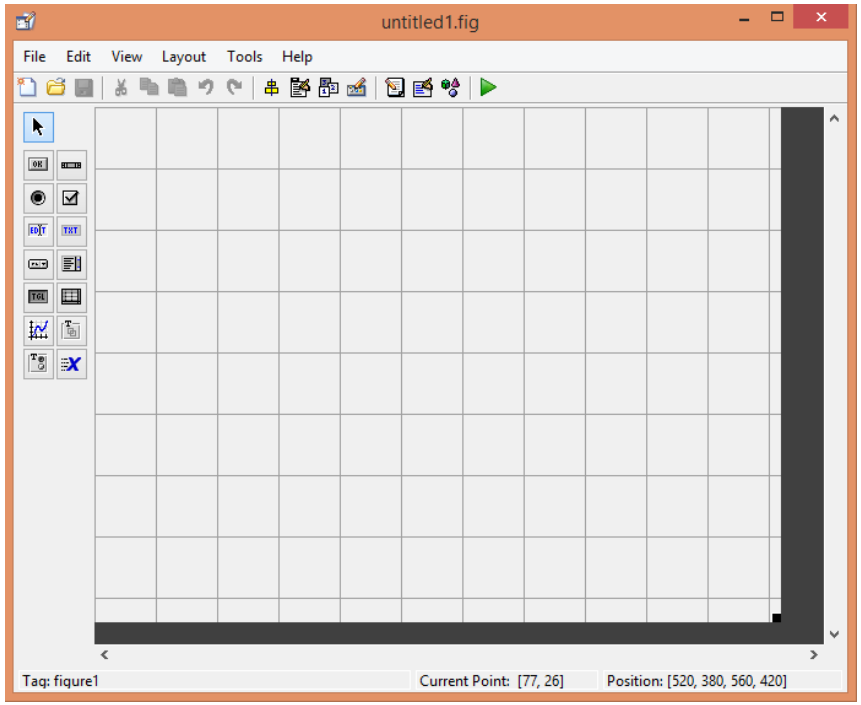
- Write .m file directly
- Through GUIDE

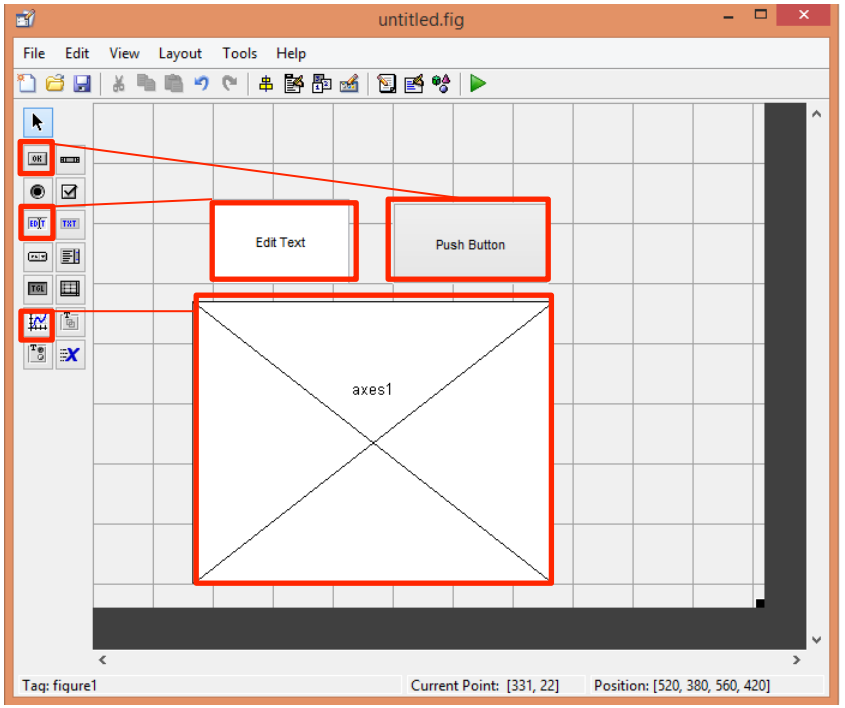
GUIDE – Getting Started



Use Open Existing GUI to change your .fig file

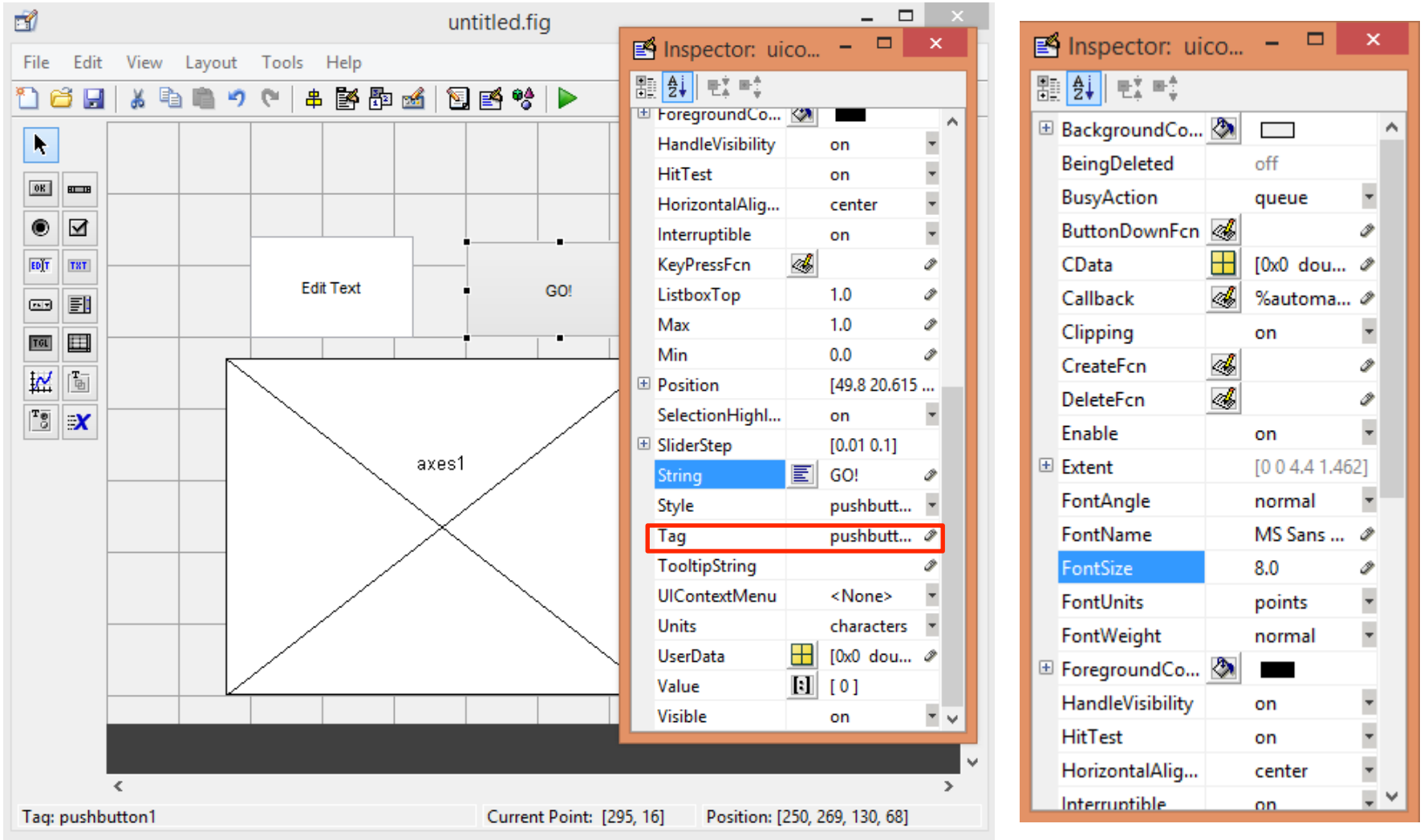
GUIDE – Making Objects





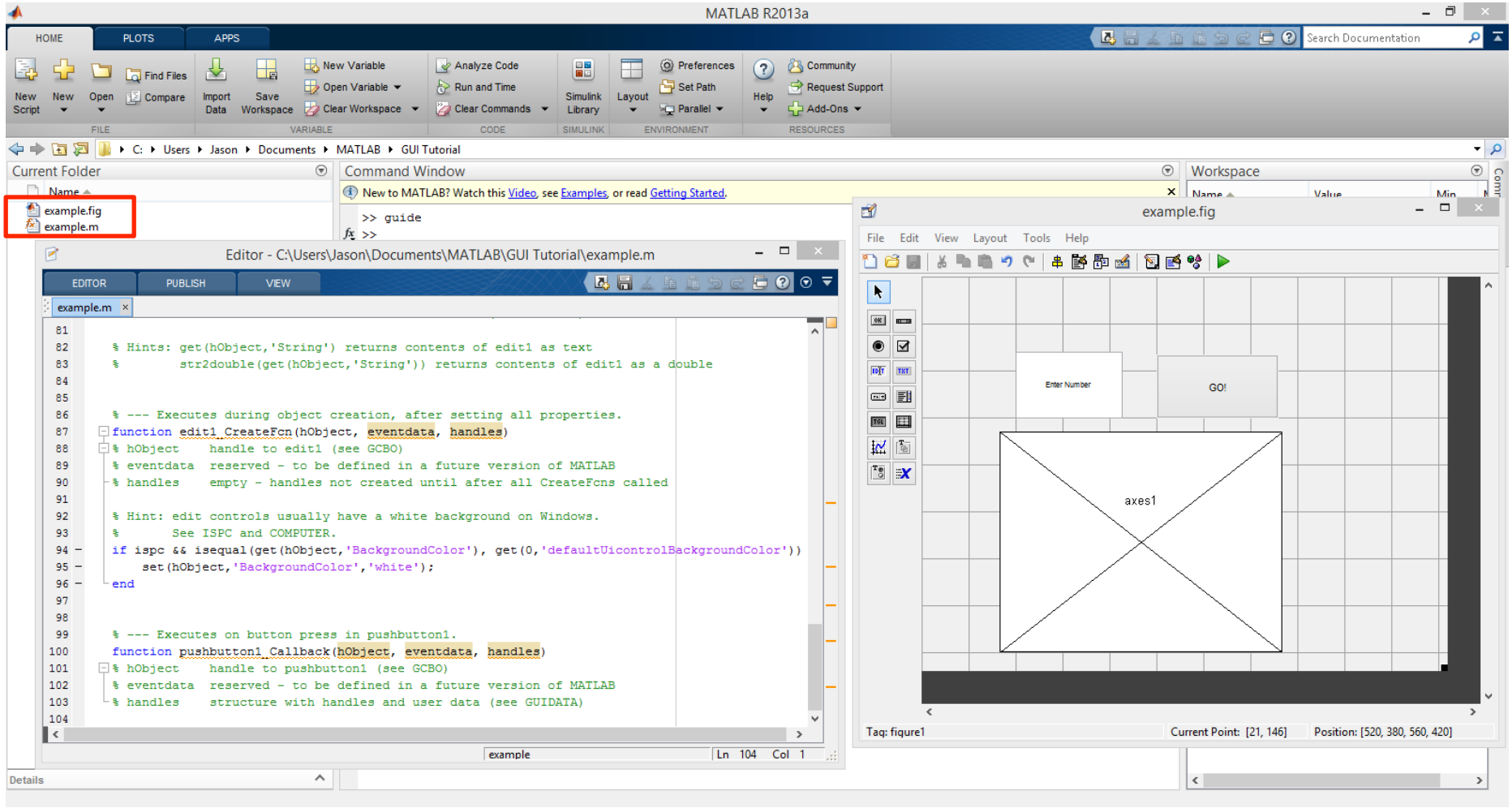
Click and drag
Adding lines of code

GUIDE – Changing Object Properties



Double click on object, and type in values in the Inspector

GUIDE - Results



.fig file is what users see

.m file contains instructions for GUI

“Handling” the .m file

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% comments
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% comments
```

Depending on object, your objects will create Callback and CreateFcn functions
-Callback function is what reacts to someone pushing a button, dragging slider, etc.
-Mainly implementing your code in Callback function

“Handling” the .m file

```
% --- Executes just before example is made visible.
function example_OpeningFcn(hObject, eventdata, handles, varargin)
% comments
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

guidata() stores all of the GUI’s data in one place

This allows you to use hObject to represent each object, and the handles struct to store all of your variables

Putting it All Together

```
function edit1_Callback(hObject, eventdata, handles)
% comments
number = str2double(get(hObject,'String'));
if isnan(number)
    errordlg('Input must be a number.','Error');
end
handles.number = number;
guidata(hObject,handles);
```

For instance, my textbox (tag = edit1) will take in some sort of number input
-I retrieve the number by using get() on hObject, which represents the textbox
-Then I store it in the GUI by using a field in the handles struct
-Note that I re-update guidata()

Putting it All Together

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
hold on;
axes(handles.axes1);
axis([-30 30 -30 30]);
set(gca,'XTick',-30:5:30);
set(gca,'YTick',-30:5:30);
xlabel('x');
ylabel('y');
title('Randomize Me');
grid on;
number = handles.number;
for i=number:-1:1
    plot(10*randn(1),10*randn(1),'r.','Markersize',20);
end
```

Note that:

-Objects from .fig file are already stored in handles (e.g. handles.axes1) and can be used

-Fields I created are accessible in other functions as long as part of same GUI (e.g. handles.number)

-set() is useful for objects with no input (e.g. set(handles.text1,'String','Hello'))

Done!

- Now let’s see what it can do…

Main Takeaways

- GUIs are really easy
- GUIDE is useful if you don't want to write from scratch
- Clean way to ask for input and display data (if done well)
  - Example: mobile devices
- Do **not** make them the meat of your project
  - Built-in technical work, but not very conceptual