Name: _____ Section: _____
Exam3 – Part2 Analysis

As usual, no credit given to something not taught. I prefer simplicity, efficiency, and least LOC.
NO LOOP. NO IF. Open notes, open labs, open doc, open solutions, etc… as usual.

In the end, submit a folder named Exam3. This folder should contain all the excel sheets, the 1 main code and the 2 function files. *CAUTION*: typically, an error will occur if you try to zip an excel sheet that is currently open. Close all Excel files before zipping.

It is time to analyze all the return values from part1. Specifications/Requirements for part 2 are as follows:

A. The main code

On top of the original requirements from part1, the main code will call 1 new function. Have semicolons everywhere.

B. Function #2

(5pts) has no (zero/0) return-values
(5pts) has nine (9) parameters: the 8 return-values from the function from part1 as well as the filename chosen
(3pts) has a valid name (that follows our usual conventions).
(6pts) must have a complete documentation.

Requirements/Specifications:

(5pts) Create a figure with 2 subplots. One plot will be on the top, the other on the bottom.

For the top plot: the state with my drive. Keep it as simple as possible and in order.

(5pts) Use `usamap();` to restrict the latitude/longitudes to the state picked. Check the doc, but the argument is the 2 letter state abbreviation.

(5pts) As indicated in the announcement, use the following LOC to show the state boundaries:

```
states = shaperead('usastatelo', 'UseGeoCoords', true,...
    'Selector',{@(name) ~any(strcmp(name,{'Alaska','Hawaii'})), 'Name'});
geoshow(ax, states, 'DisplayType', 'polygon');
```

(5pts) Create a dynamic title that shows the filename and the number of invalid data points.

(5pts) Using `plotm();`, add the latitude and longitudes of my drive.

For the bottom plot: velocity vs. distance, elevation vs. distance, and average velocity all on the same plot – in any order

(5pts) Although outdated, use `plotyy();` to plot both velocity vs. distance AND altitude vs. distance on the same plot. This will allow the y-scales to be different.

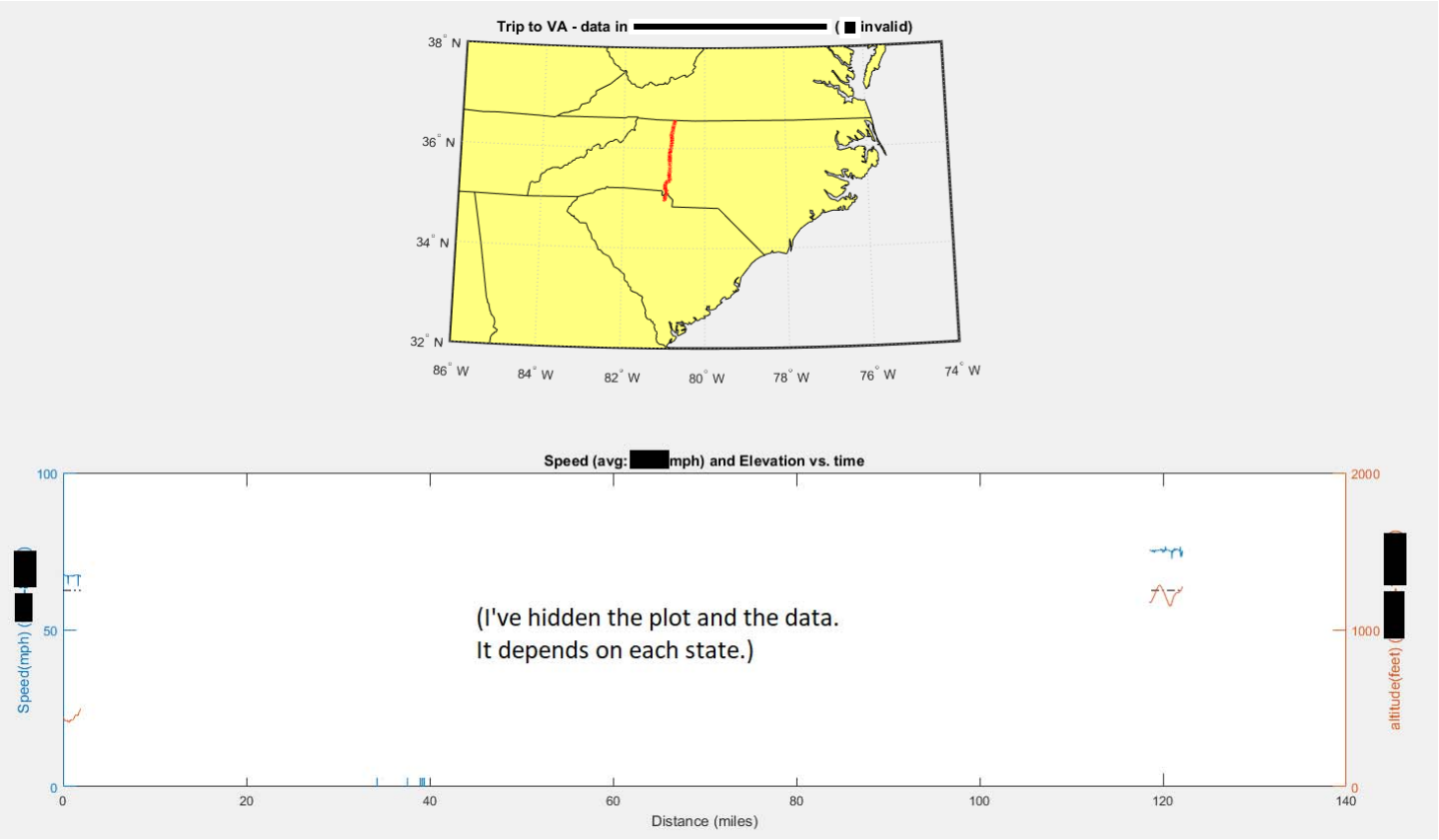
(5pts) Using a regular `plot();`, add a horizontal line to indicate the average speed that spans the entire width of the plot.

(5pts) Create a dynamic title that shows the average velocity in mph.

(10pts) Create an x-label and two dynamic y-labels. Each y-label must show the range of values (min/max of velocity, and min/max of distances). Refer to the documentation like I had to in order to learn how to do ylabels with plotyy(). Caution: simplify and use only what you need though.

C. Overall

As always, the main code and function should have the usual basics. Use any of the labs as a checklist.



(5pts) In what state did I highly use the cruise-control, why?

In what state did I likely run into Thanksgiving traffic? How can you tell?

Name: _____ Section: _____

Exam3 – Part2 Analysis

As usual, no credit given to something not taught. I prefer simplicity, efficiency, and least LOC.

NO LOOP. NO IF. Open notes, open labs, open doc, open solutions, etc… as usual.

In the end, submit a folder named Exam3. This folder should contain all the excel sheets, the 1 main code and the 2 function files. *CAUTION*: typically, an error will occur if you try to zip an excel sheet that is currently open. Close all Excel files before zipping.

It is time to analyze all the return values from part1. Specifications/Requirements for part 2 are as follows:

A. The main code

On top of the original requirements from part1, the main code will call 1 new function. Have semicolons everywhere.

B. Function #2

(5pts) has no (zero/0) return-values
(5pts) has nine (9) parameters: the 8 return-values from the function from part1 as well as the filename chosen
(3pts) has a valid name (that follows our usual conventions).
(6pts) must have a complete documentation.

Requirements/Specifications:

(5pts) Create a figure with 2 subplots. One plot will be on the top, the other on the bottom.

For the top plot: velocity vs. distance, elevation vs. distance, and average velocity all on the same plot – in any order

(5pts) Although outdated, use `plotyy();` to plot both velocity vs. distance AND altitude vs. distance on the same plot. This will allow the y-scales to be different.

(5pts) Using a regular `plot();`, add a horizontal line to indicate the average speed that spans the entire width of the plot.

(5pts) Create a dynamic title that shows the average velocity in mph.

(10pts) Create an x-label and two dynamic y-labels. Each y-label must show the range of values (min/max of velocity, and min/max of distances). Refer to the documentation like I had to in order to learn how to do ylabels with plotyy(). Caution: simplify and use only what you need though.

For the bottom plot: the state with my drive. Keep it as simple as possible and in order.

(5pts) Use `usamap();` to restrict the latitude/longitudes to the state picked. Check the doc, but the argument is the 2 letter state abbreviation.

(5pts) As indicated in the announcement, use the following LOC to show the state boundaries:

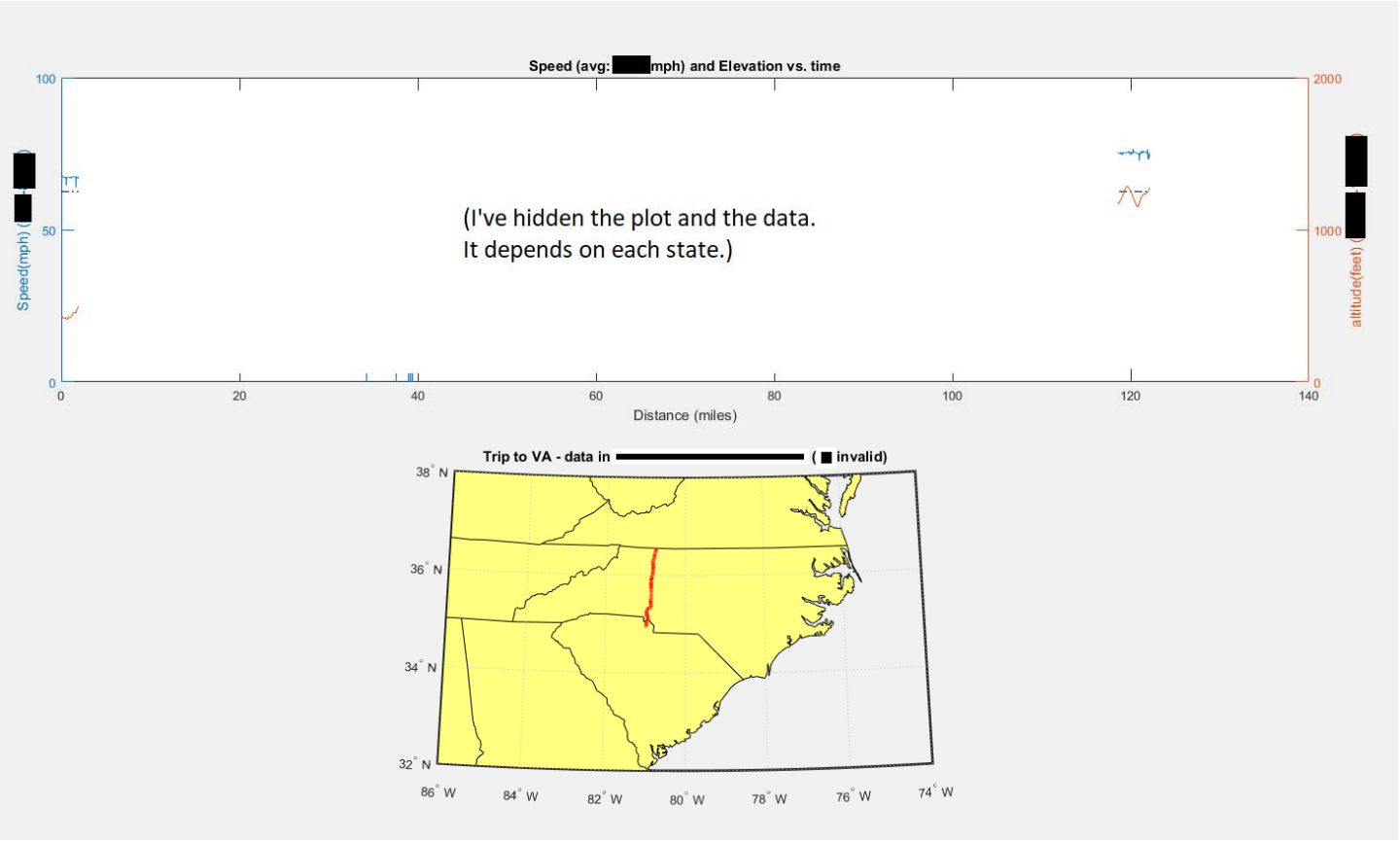
```
states = shaperead('usastatelo', 'UseGeoCoords', true,...
    'Selector',{@(name) ~any(strcmp(name,{'Alaska','Hawaii'})), 'Name'});
geoshow(ax, states, 'DisplayType', 'polygon');
```

(5pts) Create a dynamic title that shows the filename and the number of invalid data points.

(5pts) Using `plotm();`, add the latitude and longitudes of my drive.

C. Overall

As always, the main code and function should have the usual basics. Use any of the labs as a checklist.



(5pts) In what state did I highly use the cruise-control, why?

In what state did I likely run into Thanksgiving traffic? How can you tell?