Finding Lane Lines on the Road

1. **Describe the pipeline.**

My pipeline consisted of many steps. First I converted the image to grayscale, then I applied Gaussian smoothing. I then used the Canny algorithm to find the edges of the line. I created a mask to only look at lines at the region of interest. I used parameters from the image shape to dynamically create my vertices in case we use images with different dimensions. I put my image through the hough transformation to draw the lines on the image.

My draw lines function was the most interesting function in this project. The way I separated the left and right lines was detecting whether the 'x2' parameter was greater than the middle of the dimensions or not.

```
        if (x2> imshape[1]/2):
```

If it was greater than imshape[1]/2 then I pushed the line to the 'right' array, otherwise I pushed it to the 'left' one. Once I had two separated arrays, I sent them to a helper function called 'average_x'. The function takes in a side (left or right) as well as the dimensions of the image (dim). It appends all the x values to the x array and all the y values to the y array. Then it creates a line using 'numpy.polyfit'. numpy.polyfit creates a least squares fit line for all of the points in the x and y array. We set this to a variable 'z'. The linear equation is 'y = mx+b'. The ouputs of numpy.polyfit is z[0] = m and z[1] = b.

Since we want the top of the line to be the top of our area of interest, we can set Yf to 'dim[0]/2+70'. Since we want the bottom of the line to be the end of the screen, we can set Yo to dim[0]. Since we know the slope (m), y-intercept (b), and the two Y's, we can find the two X's to create our line. Since 'y=mx +b', we can reorder the equation to solve for x. 'x=(y-b)/m'.

Once I solved for this equation, I returned it to my 'draw lines' function and drew the lines on the canvas. There are other checks in case there are no lines found or if the slope was infinite, but it worked for the most part.

2. **Identify potential shortcomings with your current pipeline**

My current pipeline does not detect lines in the upper region of my region of interest. This leads to a huge skew in the lines that favors whatever is in the bottom region of my region of interest. This made this pipeline fail on the challenge video because the lines were curved. It also failed when there are horizontal lines on the bottom of the screen.

3. **Suggest possible improvements to the pipeline**

A possible improvement would be to edit the parameters of the Hough transform to see more of the lines at the top of the region of interest.