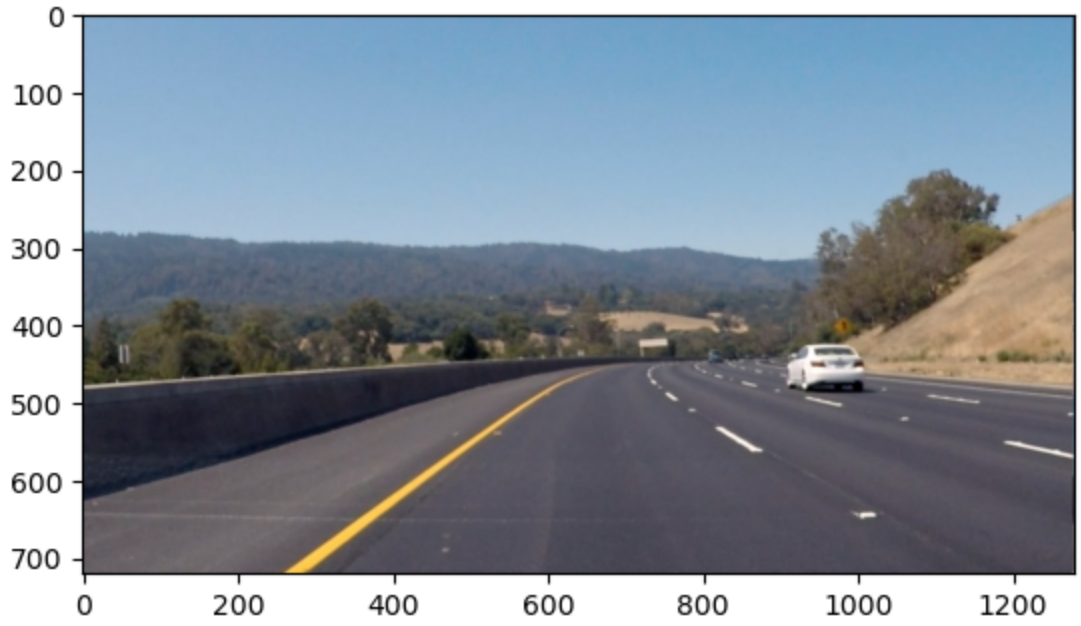
Creating a Filter, Edge Detection

In [6]:
```
# Import resources and display image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

import cv2
import numpy as np

%matplotlib inline

# Read in the image
image = mpimg.imread('D:/Udacity/Jupyter/curved_lane.jpg')

plt.imshow(image)
```

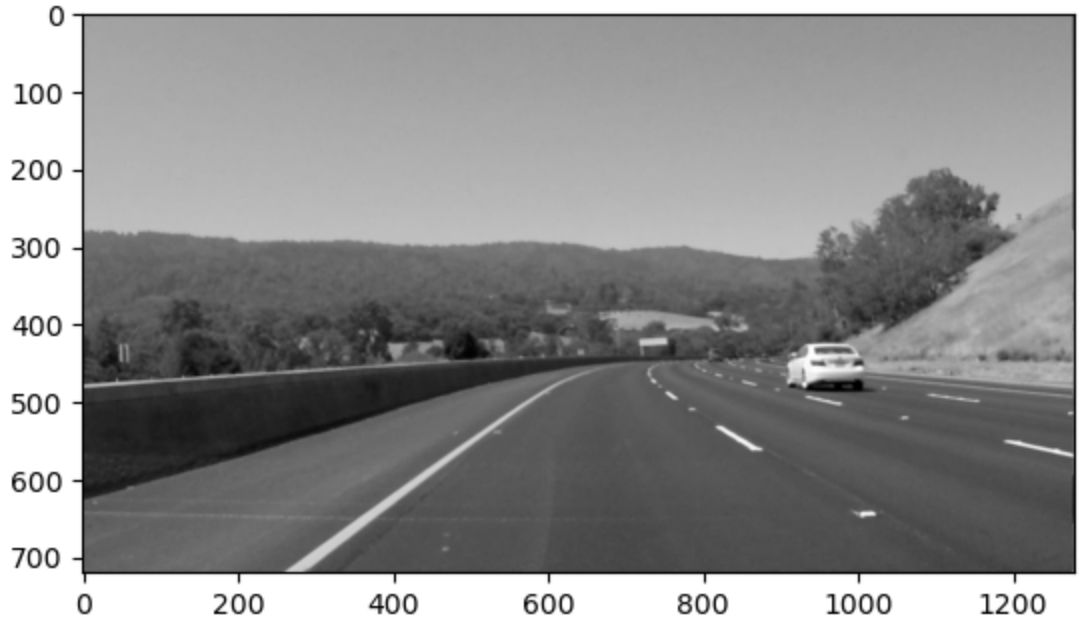
Out[6]:
```
<matplotlib.image.AxesImage at 0x1f90c225850>
```



Converting the image to grayscale

In [7]:
```
# Converting to grayscale for filtering
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
plt.imshow(gray, cmap='gray')
```

Out[7]:
```
<matplotlib.image.AxesImage at 0x1f90be54210>
```



In [5]:
```
# Creating a custom kernel

# 3x3 array for edge detection
sobel_y = np.array([[ -1, -2, -1],
                    [ 0, 0, 0],
                    [ 1, 2, 1]])

# Filter the image using filter2D, which has inputs: (grayscale image, bit-depth, kernel
filtered_image = cv2.filter2D(gray, -1, sobel_y)
plt.imshow(filtered_image, cmap='gray')
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x1f90c17a150>
```