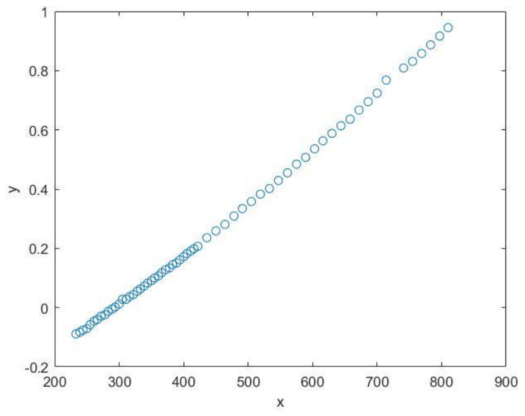
Q1:

a)

```
K = [233 239 244 250 255 261 266 272 278 283 289 294 300 305 ...
    311 316 322 328 333 339 344 350 355 361 366 372 378 383 389
394 400 ...
    405 411 416 422 436 450 464 478 491 505 519 533 547 561 575
589 603 ...
    616 630 644 658 672 686 700 714 741 755 769 783 797 810]';

percentChange = [-0.089 -0.083 -0.076 -0.071 -0.058 -0.046 -
0.040 ...
    -0.029 -0.024 -0.013 -0.005 0.002 0.012 0.028 0.028 0.037
0.044 ...
    0.055 0.063 0.073 0.083 0.091 0.100 0.107 0.118 0.128 0.134
0.145 0.151 0.161 0.172...
    0.182 0.191 0.199 0.207 0.236 0.259 0.281 0.309 0.334 0.358
0.383 0.402 0.429 0.455...
    0.484 0.507 0.536 0.563 0.588 0.614 0.636 0.667 0.695 0.724
0.768 0.809 0.831 0.858...
     0.887 0.917 0.945]';
```



b)

```
question b using the 'regress' function
b0 = -0.5296
b1 = 0.0018
question b using backslash operator
b0 = -0.5296
b1 = 0.0018
% By comparing, regress function and backslash give the same
answer.
```

c)

MSE = 1.7476e-04

Q2:

a)

```
    1. mpg     2. cylinders     3. displacement
    4. horsepower     5. weight     6. acceleration     7. model year     8. 
Origin
```

b)

Delete those rows with missing entries entirely and load('auto-mpg.data') successfully.

c)

b0 = 46.2643

b1 = -0.3979

b2 = -8.3130e-05

b3 = -0.0453

b4 = -0.0052

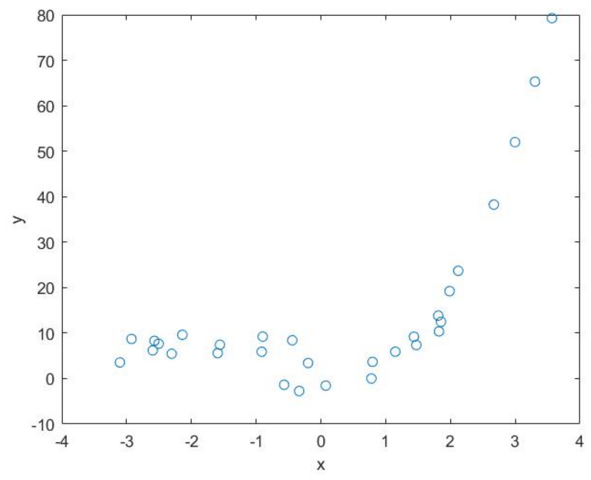
b5 = -0.0291

d)

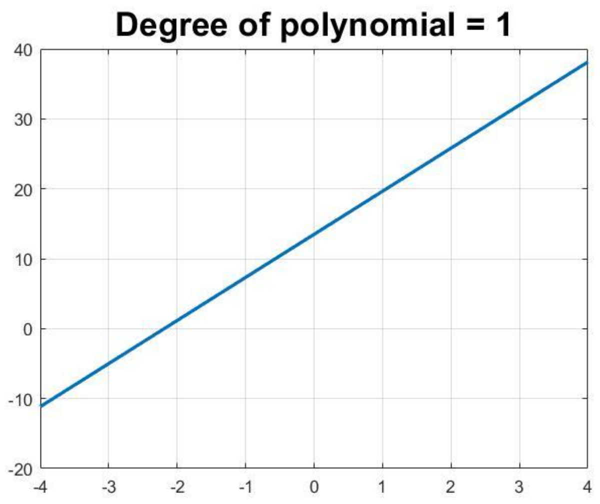
mpg will decrease when displacement, horsepower, weight (influence is small), cylinders and acceleration increase

Q3:

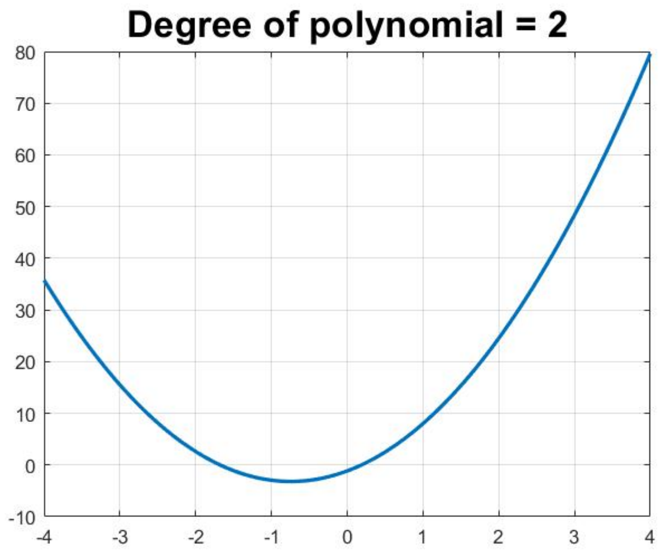
Origin:



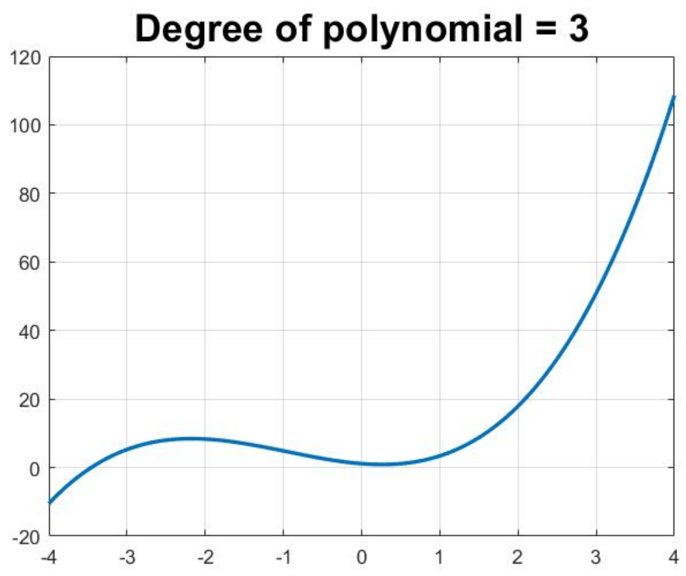
a)


Degree of polynomial = 1
40
30
20
10
0
-10
-20
-4
-3
-2
-1
0
1
2
3
4

b)

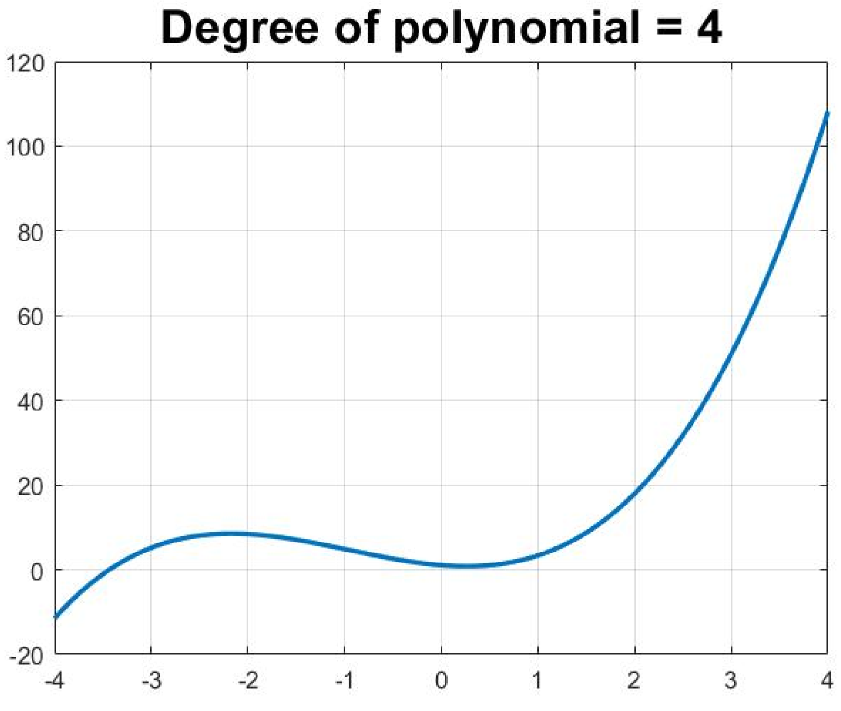

Degree of polynomial = 2
80
70
60
50
40
30
20
10
0
-10
-4
-3
-2
-1
0
1
2
3
4

c)



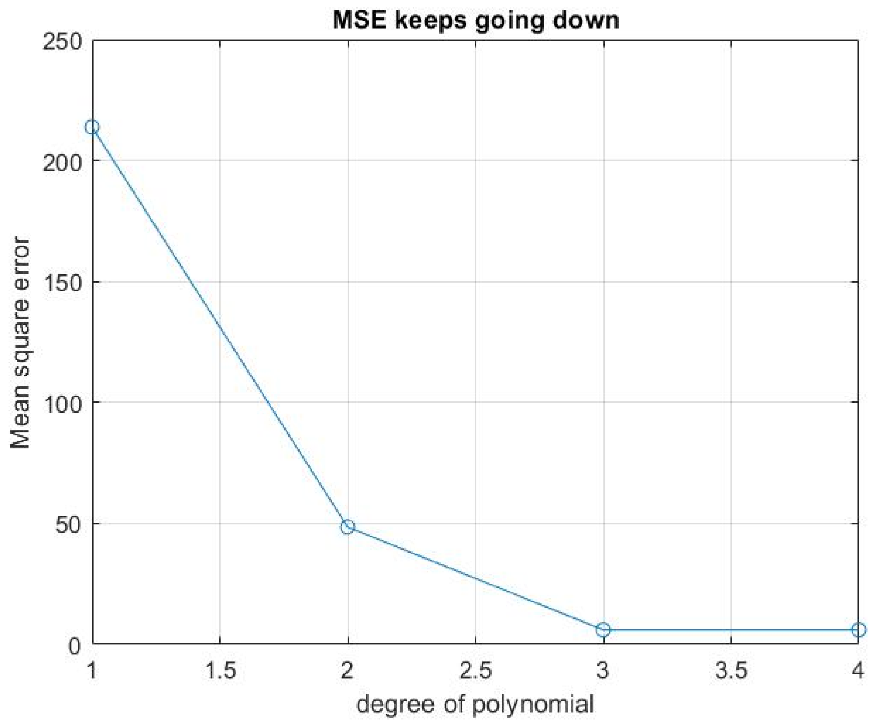
Degree of polynomial = 3
120
100
80
60
40
20
0
-20
-4
-3
-2
-1
0
1
2
3
4

d)



Degree of polynomial = 4
120
100
80
60
40
20
0
-20
-4
-3
-2
-1
0
1
2
3
4

e)



Q4:

a)

a0 =-0.5333

a1 = 0.1985

a2 = 0.7477

MSE1 = 0.0075

b0 = 2.8232

b1 = -1.3445

b2 = -0.6421

MSE2 = 0.0406

b)

a0 = 3.0234

a1 = -5.0748

a2 = -2.5197

a3 = 1.9098

a4 = 1.1691

a5 = 1.5591

MSE1b = 6.9914e-04

b0 = -4.6410

b1 = 9.7237

b2 = 6.8680

b3 = -4.0357

b4 = -2.7861

b5 = -3.4645

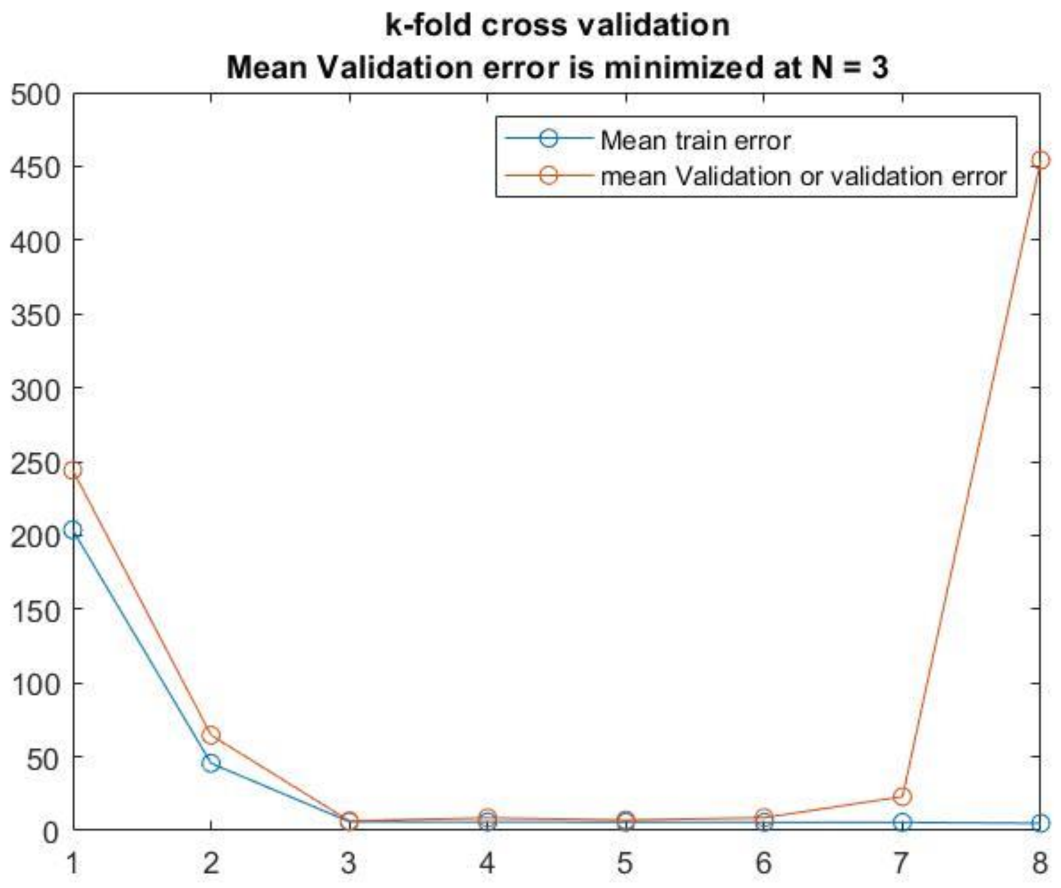
MSE2b = 0.0035

Q5:

a)

```
optimalDegree = find(mean(MSE_Matrix_Validation)==min(mean(MSE_Matrix_Validation)));
```

```
optimalDegree = 3
```



```
b)
noiseList =

   14.8747    7.0888    2.5029    2.5020    2.4526    2.4526
```

y has noise with standard deviation equal to 3, therefore polynomial degree is better if std(E)closer to 3. Thus, the polynomial degree should be 3