Diamonds dataset

Clear workspace

```
clear all; clc; close all;
```

Initialize variables

Load table

```
diamonds = readtable('diamonds.csv')
```

```
diamonds = 53940×11 table
```

...

	Var1	carat	cut	color	clarity	depth	table	price
1	1	0.2300	'Ideal'	'E'	'SI2'	61.5000	55	326
2	2	0.2100	'Premium'	'E'	'SI1'	59.8000	61	326
3	3	0.2300	'Good'	'E'	'VS1'	56.9000	65	327
4	4	0.2900	'Premium'	'I'	'VS2'	62.4000	58	334
5	5	0.3100	'Good'	'J'	'SI2'	63.3000	58	335
6	6	0.2400	'Very Good'	'J'	'VVS2'	62.8000	57	336
7	7	0.2400	'Very Good'	'I'	'VVS1'	62.3000	57	336
8	8	0.2600	'Very Good'	'H'	'SI1'	61.9000	55	337
9	9	0.2200	'Fair'	'E'	'VS2'	65.1000	61	337
10	10	0.2300	'Very Good'	'H'	'VS1'	59.4000	61	338
11	11	0.3000	'Good'	'J'	'SI1'	64	55	339
12	12	0.2300	'Ideal'	'J'	'VS1'	62.8000	56	340
13	13	0.2200	'Premium'	'F'	'SI1'	60.4000	61	342
14	14	0.3100	'Ideal'	'J'	'SI2'	62.2000	54	344
15	15	0.2000	'Premium'	'E'	'SI2'	60.2000	62	345
16	16	0.3200	'Premium'	'E'	'I1'	60.9000	58	345
17	17	0.3000	'Ideal'	'I'	'SI2'	62	54	348
18	18	0.3000	'Good'	'J'	'SI1'	63.4000	54	351
19	19	0.3000	'Good'	'J'	'SI1'	63.8000	56	351
20	20	0.3000	'Very Good'	'J'	'SI1'	62.7000	59	351
21	21	0.3000	'Good'	'I'	'SI2'	63.3000	56	351
22	22	0.2300	'Very Good'	'E'	'VS2'	63.8000	55	352
23	23	0.2300	'Very Good'	'H'	'VS1'	61	57	353
24	24	0.3100	'Very Good'	'J'	'SI1'	59.4000	62	353

	Var1	carat	cut	color	clarity	depth	table	price
25	25	0.3100	'Very Good'	'J'	'SI1'	58.1000	62	353
26	26	0.2300	'Very Good'	'G'	'VVS2'	60.4000	58	354
27	27	0.2400	'Premium'	'I'	'VS1'	62.5000	57	355
28	28	0.3000	'Very Good'	'J'	'VS2'	62.2000	57	357
29	29	0.2300	'Very Good'	'D'	'VS2'	60.5000	61	357
30	30	0.2300	'Very Good'	'F'	'VS1'	60.9000	57	357
31	31	0.2300	'Very Good'	'F'	'VS1'	60	57	402
32	32	0.2300	'Very Good'	'F'	'VS1'	59.8000	57	402
33	33	0.2300	'Very Good'	'E'	'VS1'	60.7000	59	402
34	34	0.2300	'Very Good'	'E'	'VS1'	59.5000	58	402
35	35	0.2300	'Very Good'	'D'	'VS1'	61.9000	58	402
36	36	0.2300	'Good'	'F'	'VS1'	58.2000	59	402
37	37	0.2300	'Good'	'E'	'VS1'	64.1000	59	402
38	38	0.3100	'Good'	'H'	'SI1'	64	54	402
39	39	0.2600	'Very Good'	'D'	'VS2'	60.8000	59	403
40	40	0.3300	'Ideal'	'I'	'SI2'	61.8000	55	403
41	41	0.3300	'Ideal'	'I'	'SI2'	61.2000	56	403
42	42	0.3300	'Ideal'	'J'	'SI1'	61.1000	56	403
43	43	0.2600	'Good'	'D'	'VS2'	65.2000	56	403
44	44	0.2600	'Good'	'D'	'VS1'	58.4000	63	403
45	45	0.3200	'Good'	'H'	'SI2'	63.1000	56	403
46	46	0.2900	'Premium'	'F'	'SI1'	62.4000	58	403
47	47	0.3200	'Very Good'	'H'	'SI2'	61.8000	55	403
48	48	0.3200	'Good'	'H'	'SI2'	63.8000	56	403
49	49	0.2500	'Very Good'	'E'	'VS2'	63.3000	60	404
50	50	0.2900	'Very Good'	'H'	'SI2'	60.7000	60	404
51	51	0.2400	'Very Good'	'F'	'SI1'	60.9000	61	404
52	52	0.2300	'Ideal'	'G'	'VS1'	61.9000	54	404
53	53	0.3200	'Ideal'	'I'	'SI1'	60.9000	55	404
54	54	0.2200	'Premium'	'E'	'VS2'	61.6000	58	404
55	55	0.2200	'Premium'	'D'	'VS2'	59.3000	62	404
56	56	0.3000	'Ideal'	'I'	'SI2'	61	59	405
57	57	0.3000	'Premium'	'J'	'SI2'	59.3000	61	405

	Var1	carat	cut	color	clarity	depth	table	price
58	58	0.3000	'Very Good'	'I'	'SI1'	62.6000	57	405
59	59	0.3000	'Very Good'	'I'	'SI1'	63	57	405
60	60	0.3000	'Good'	'I'	'SI1'	63.2000	55	405
61	61	0.3500	'Ideal'	'I'	'VS1'	60.9000	57	552
62	62	0.3000	'Premium'	'D'	'SI1'	62.6000	59	552
63	63	0.3000	'Ideal'	'D'	'SI1'	62.5000	57	552
64	64	0.3000	'Ideal'	'D'	'SI1'	62.1000	56	552
65	65	0.4200	'Premium'	'I'	'SI2'	61.5000	59	552
66	66	0.2800	'Ideal'	'G'	'VVS2'	61.4000	56	553
67	67	0.3200	'Ideal'	'I'	'VVS1'	62	55.3000	553
68	68	0.3100	'Very Good'	'G'	'SI1'	63.3000	57	553
69	69	0.3100	'Premium'	'G'	'SI1'	61.8000	58	553
70	70	0.2400	'Premium'	'E'	'VVS1'	60.7000	58	553
71	71	0.2400	'Very Good'	'D'	'VVS1'	61.5000	60	553
72	72	0.3000	'Very Good'	'H'	'SI1'	63.1000	56	554
73	73	0.3000	'Premium'	'H'	'SI1'	62.9000	59	554
74	74	0.3000	'Premium'	'H'	'SI1'	62.5000	57	554
75	75	0.3000	'Good'	'H'	'SI1'	63.7000	57	554
76	76	0.2600	'Very Good'	'F'	'VVS2'	59.2000	60	554
77	77	0.2600	'Very Good'	'E'	'VVS2'	59.9000	58	554
78	78	0.2600	'Very Good'	'D'	'VVS2'	62.4000	54	554
79	79	0.2600	'Very Good'	'D'	'VVS2'	62.8000	60	554
80	80	0.2600	'Very Good'	'E'	'VVS1'	62.6000	59	554
81	81	0.2600	'Very Good'	'E'	'VVS1'	63.4000	59	554
82	82	0.2600	'Very Good'	'D'	'VVS1'	62.1000	60	554
83	83	0.2600	'Ideal'	'E'	'VVS2'	62.9000	58	554
84	84	0.3800	'Ideal'	'I'	'SI2'	61.6000	56	554
85	85	0.2600	'Good'	'E'	'VVS1'	57.9000	60	554
86	86	0.2400	'Premium'	'G'	'VVS1'	62.3000	59	554
87	87	0.2400	'Premium'	'H'	'VVS1'	61.2000	58	554
88	88	0.2400	'Premium'	'H'	'VVS1'	60.8000	59	554
89	89	0.2400	'Premium'	'H'	'VVS2'	60.7000	58	554
90	90	0.3200	'Premium'	'I'	'SI1'	62.9000	58	554

	Var1	carat	cut	color	clarity	depth	table	price
91	91	0.7000	'Ideal'	'E'	'SI1'	62.5000	57	2757
92	92	0.8600	'Fair'	'E'	'SI2'	55.1000	69	2757
93	93	0.7000	'Ideal'	'G'	'VS2'	61.6000	56	2757
94	94	0.7100	'Very Good'	'E'	'VS2'	62.4000	57	2759
95	95	0.7800	'Very Good'	'G'	'SI2'	63.8000	56	2759
96	96	0.7000	'Good'	'E'	'VS2'	57.5000	58	2759
97	97	0.7000	'Good'	'F'	'VS1'	59.4000	62	2759
98	98	0.9600	'Fair'	'F'	'SI2'	66.3000	62	2759
99	99	0.7300	'Very Good'	'E'	'SI1'	61.6000	59	2760
100	100	0.8000	'Premium'	'H'	'SI1'	61.5000	58	2760

⋮

Transform categorical variables

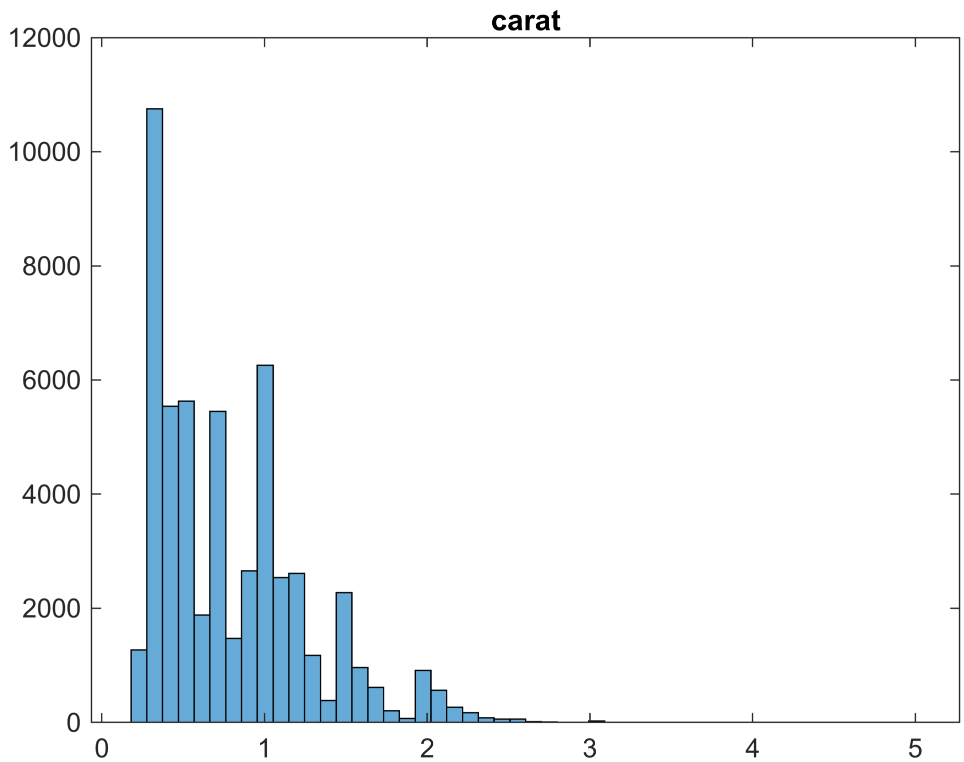
```
diamonds.cut = categorical(diamonds.cut);
diamonds.color = categorical(diamonds.color);
diamonds.clarity = categorical(diamonds.clarity);
```

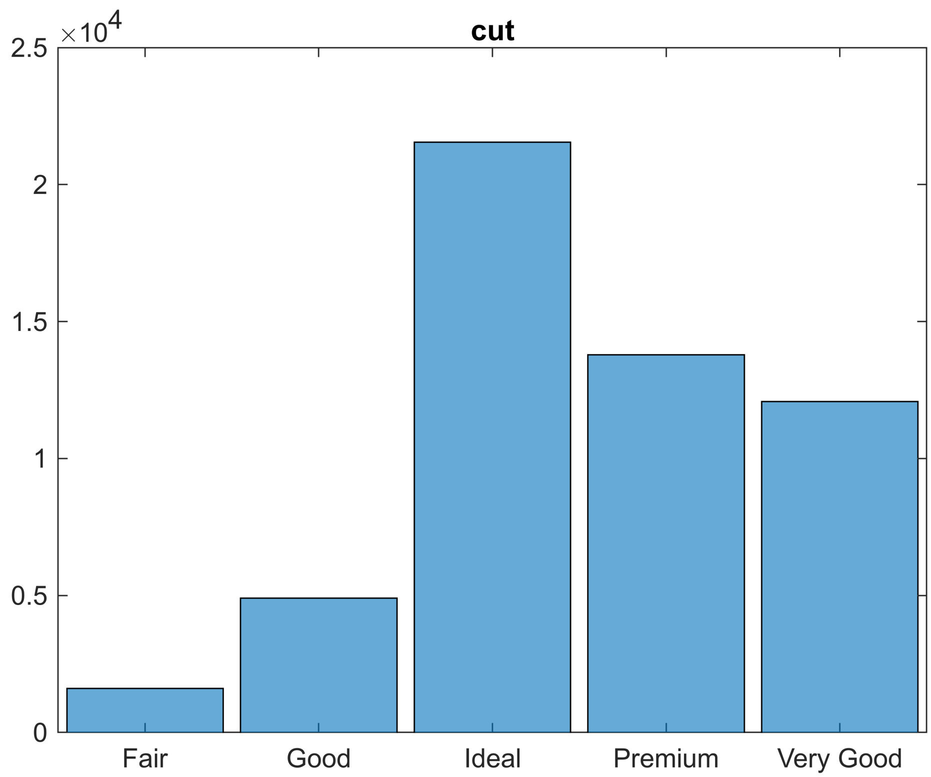
Some Histograms for better understanding of the data

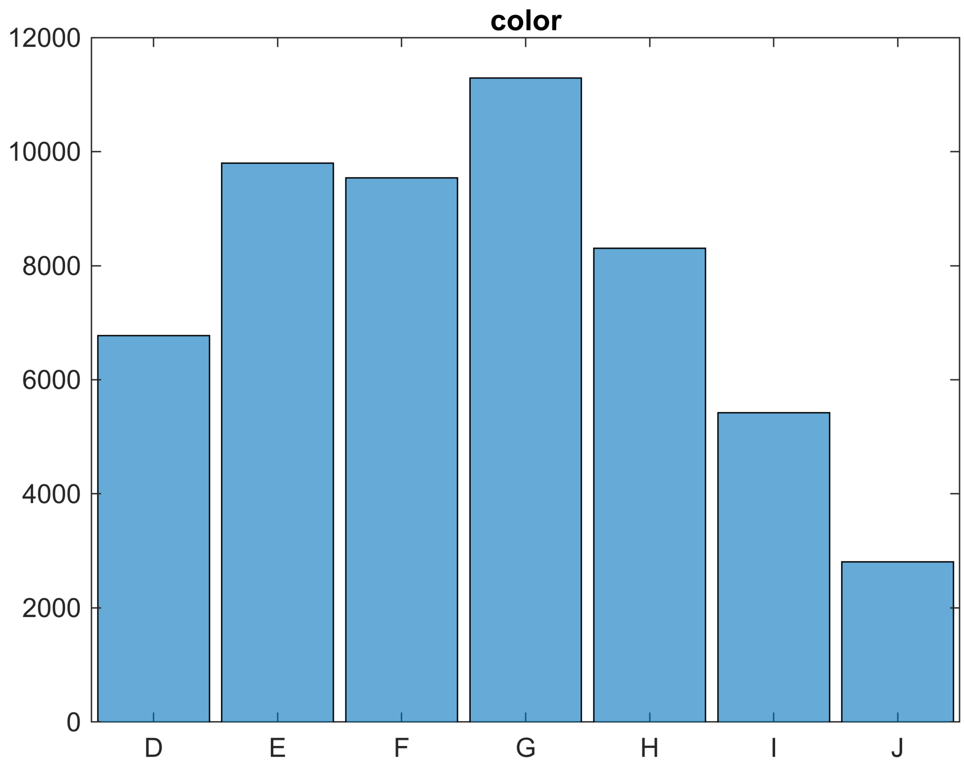
```
nbins = 50
```

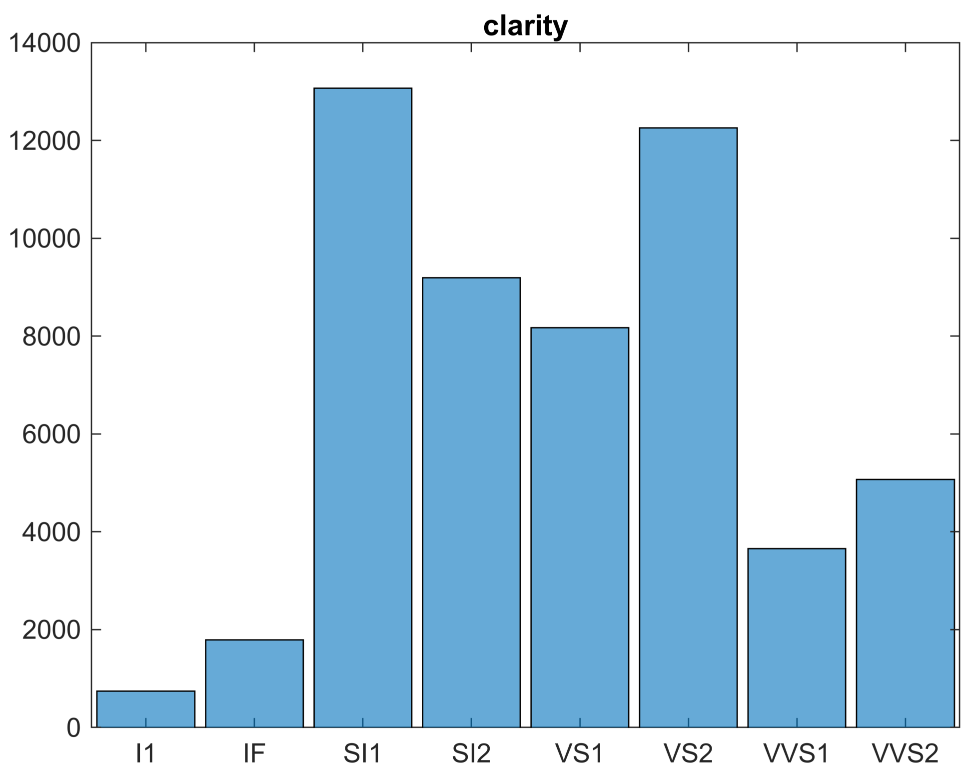
```
nbins = 50
```

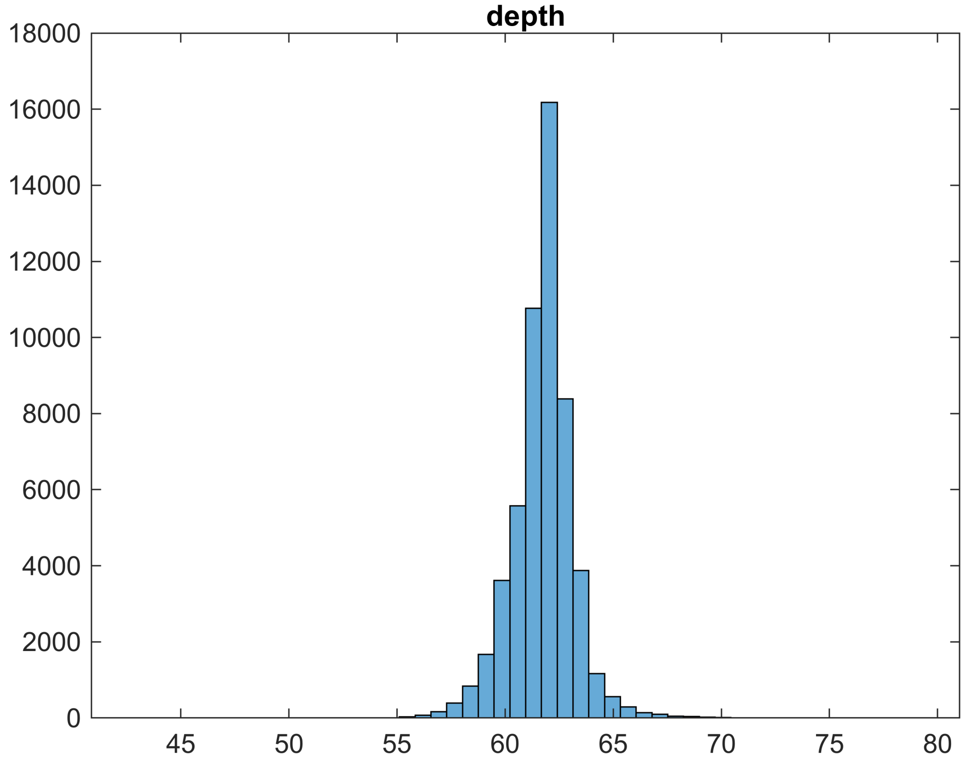
```

for col = 2: width(diamonds)
    figure
    if iscategorical(table2array(diamonds(:,col)))
        histogram(table2array(diamonds(:,col)))
    else
        histogram(table2array(diamonds(:,col)), nbins)
    end
    title(diamonds.Properties.VariableNames{col})
%     saveas(gcf,strcat('02_Diamonds/Hist_', diamonds.Properties.VariableNames{col}, '.png') )
end
```


carat
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5


cut
$\times 10^4$
2.5
2
1.5
1
0.5
0
Fair
Good
Ideal
Premium
Very Good


color
12000
10000
8000
6000
4000
2000
0
D
E
F
G
H
I
J


clarity
14000
12000
10000
8000
6000
4000
2000
0
I1
IF
SI1
SI2
VS1
VS2
VVS1
VVS2


depth
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
45
50
55
60
65
70
75
80

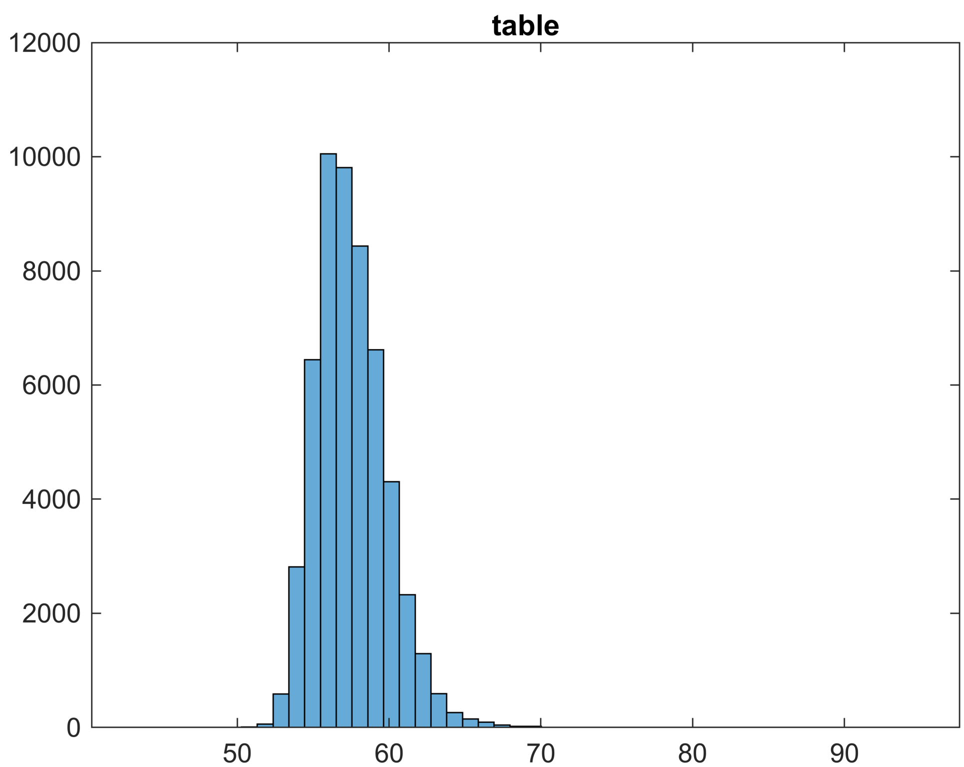
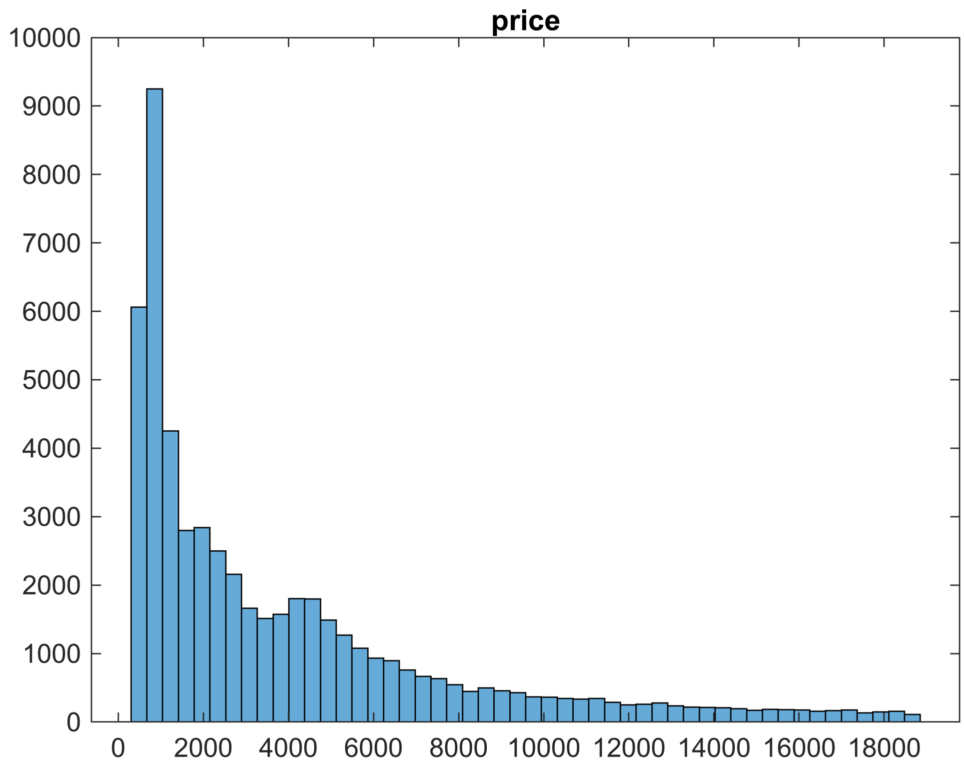
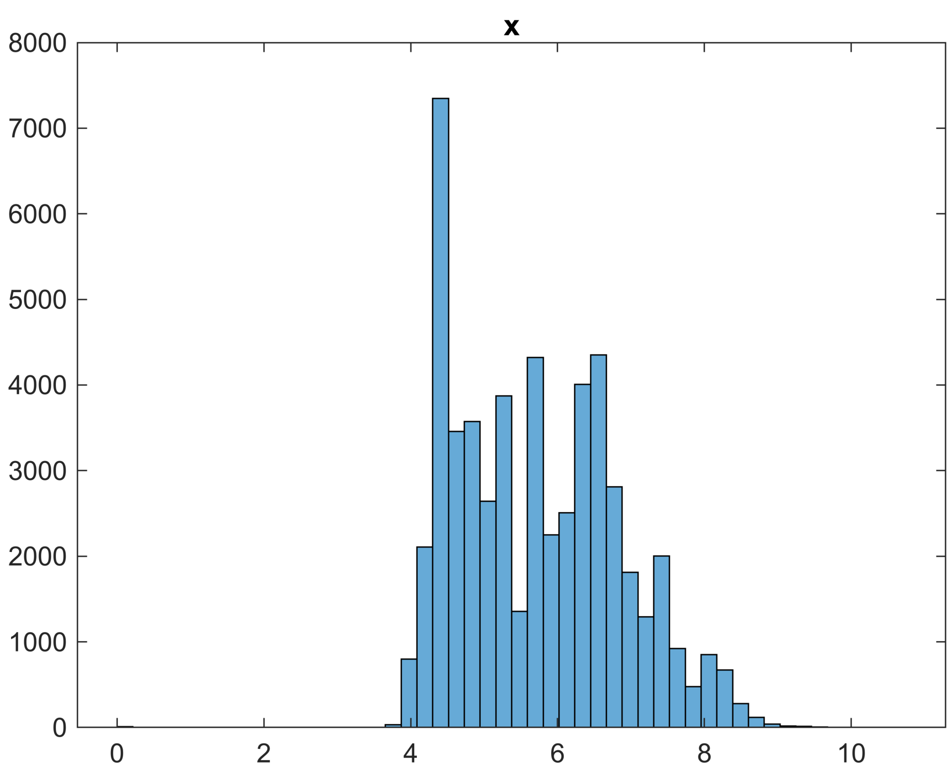
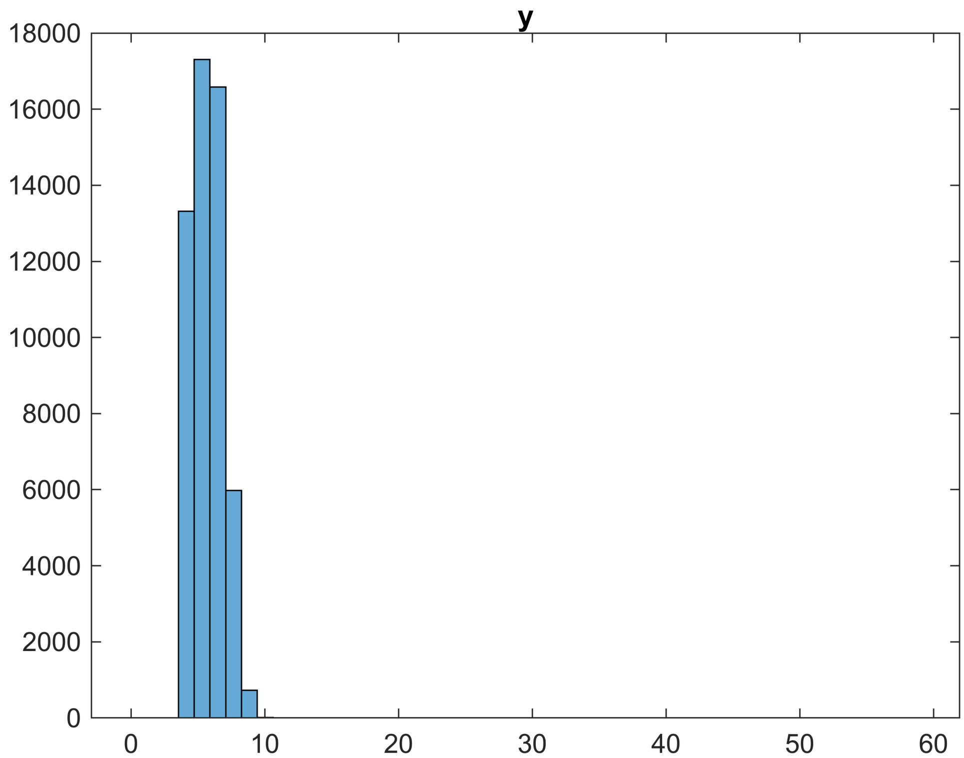


table
12000
10000
8000
6000
4000
2000
0
50
60
70
80
90

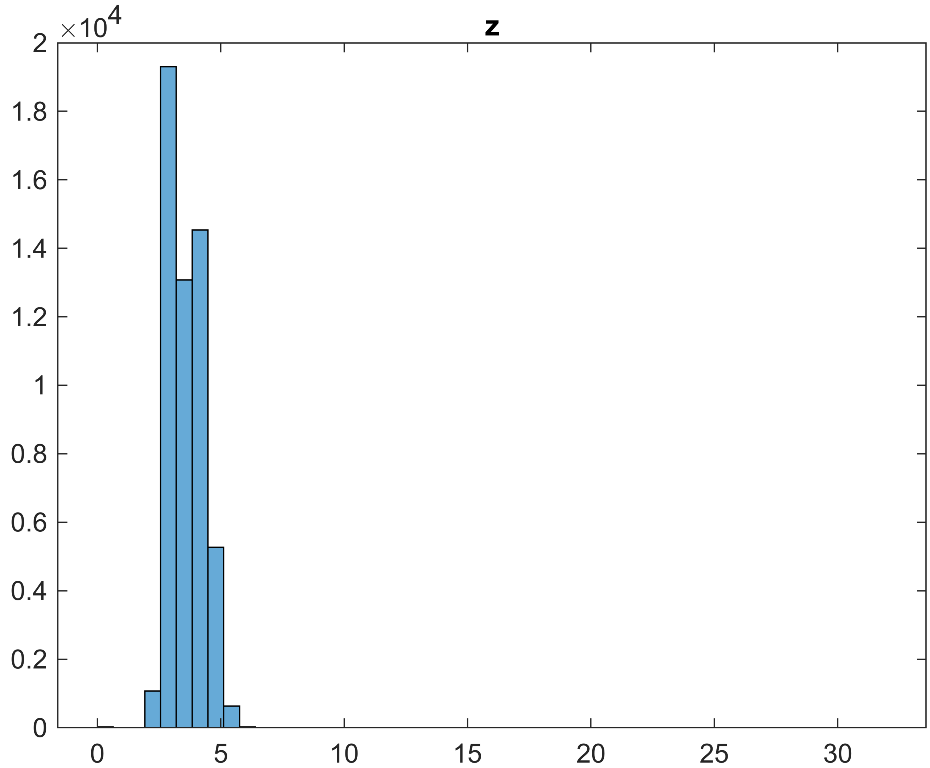

price
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000


x
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10



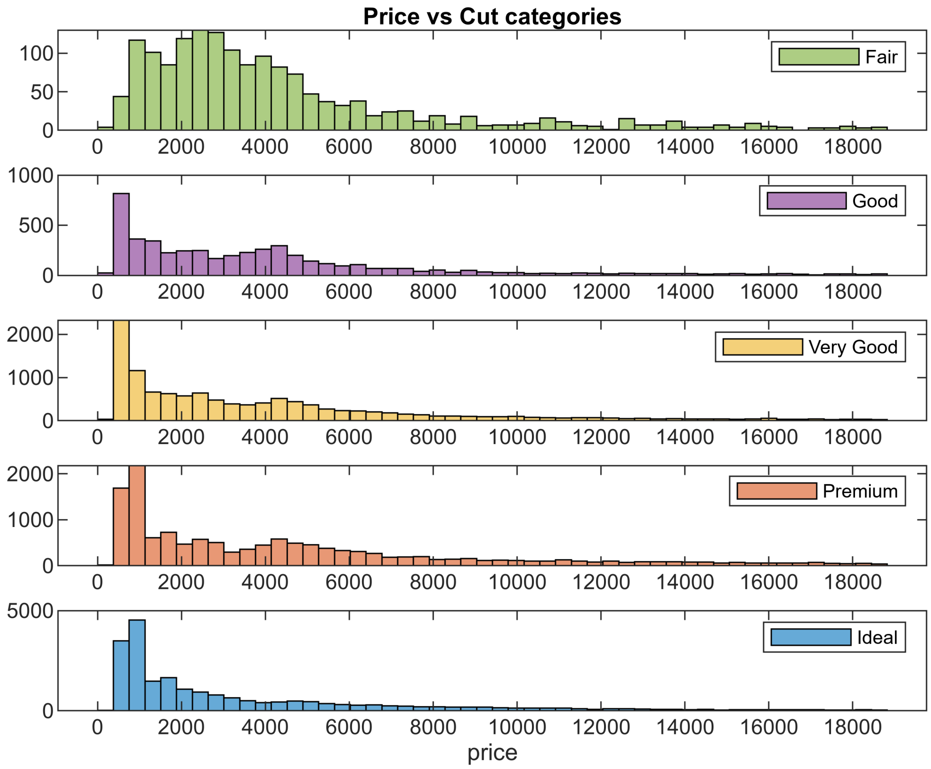
Prices for different categories

```
hist_edges = linspace(0, max(diamonds.price), nbins+1);
plot_edges = linspace(hist_edges(2)/2, (hist_edges(end-1)+hist_edges(end))/2, nbins);

colors = ["#0072BD", "#D95319", "#EDB120", "#7E2F8E", "#77AC30", "#4DBEEE", "#A2142F", "#0072BD
colororder(colors)
```

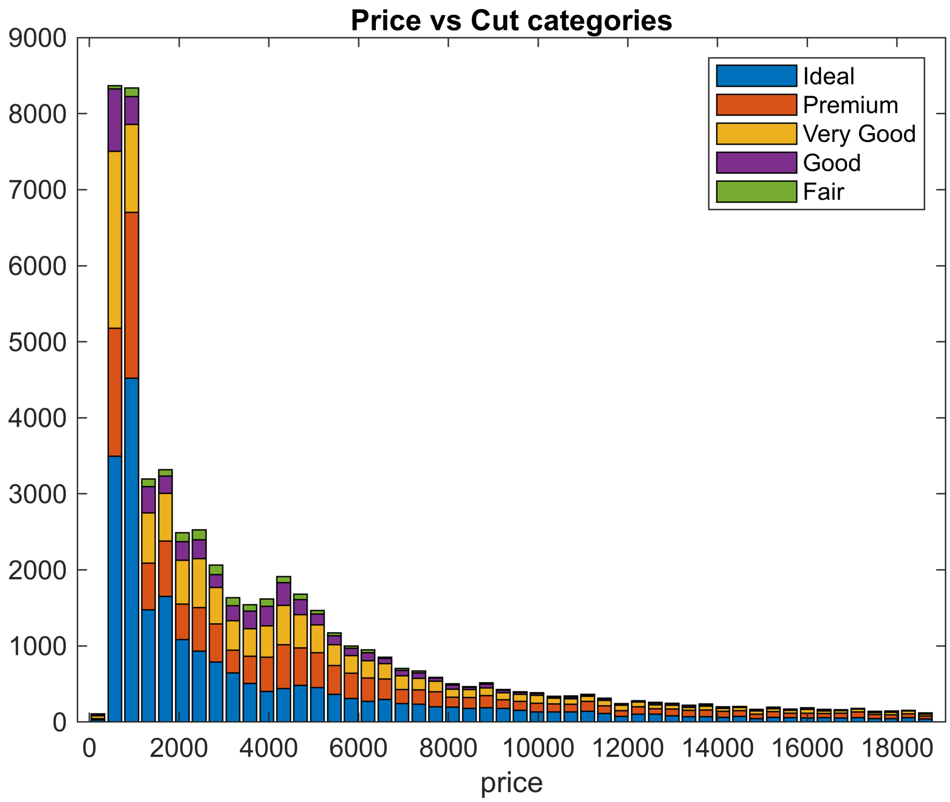


Cut

```
cut_categories = unique(diamonds.cut);
cut_categories = {'Ideal';'Premium';'Very Good';'Good';'Fair'};

figure
for ii = 1:size(cut_categories)
    X(ii,:) = histcounts( diamonds.price(diamonds.cut == cut_categories(ii)), hist_edges);
    subplot(size(cut_categories,1), 1, size(cut_categories,1)+1-ii)
    histogram( diamonds.price(diamonds.cut == cut_categories(ii)), hist_edges, FaceColor=colors
    legend(cut_categories(ii))
    if ii ==1
        xlabel('price')
    end
end
title('Price vs Cut categories')
```



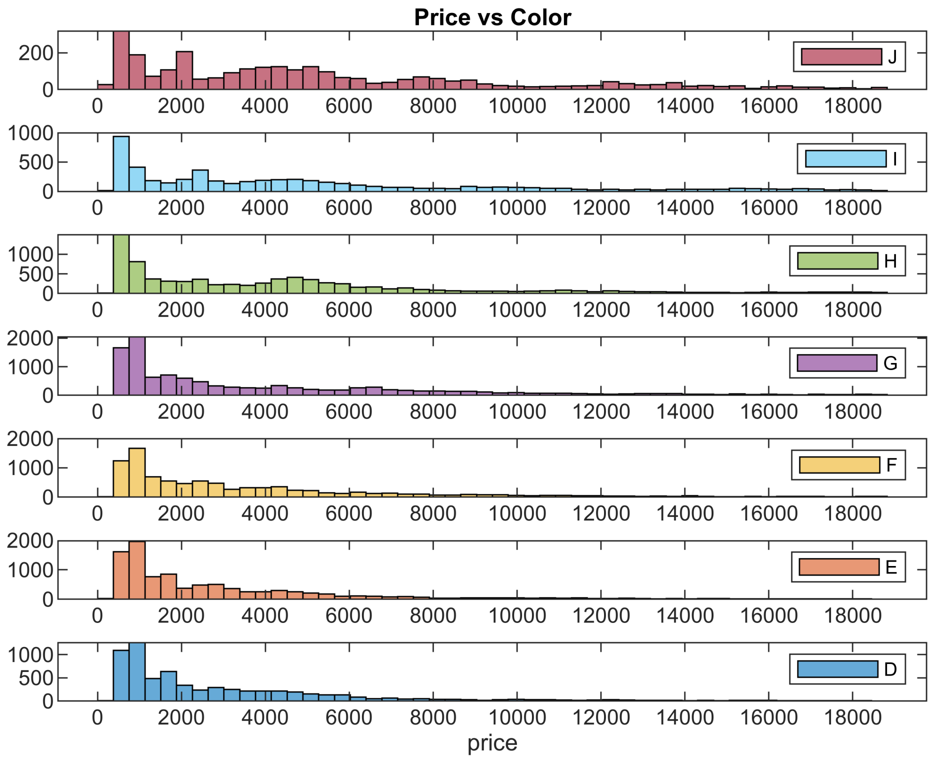

```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceCut.png') )

figure
bar(plot_edges, X, 'stacked')
xlabel('price')
legend(cut_categories)
title('Price vs Cut categories')
```



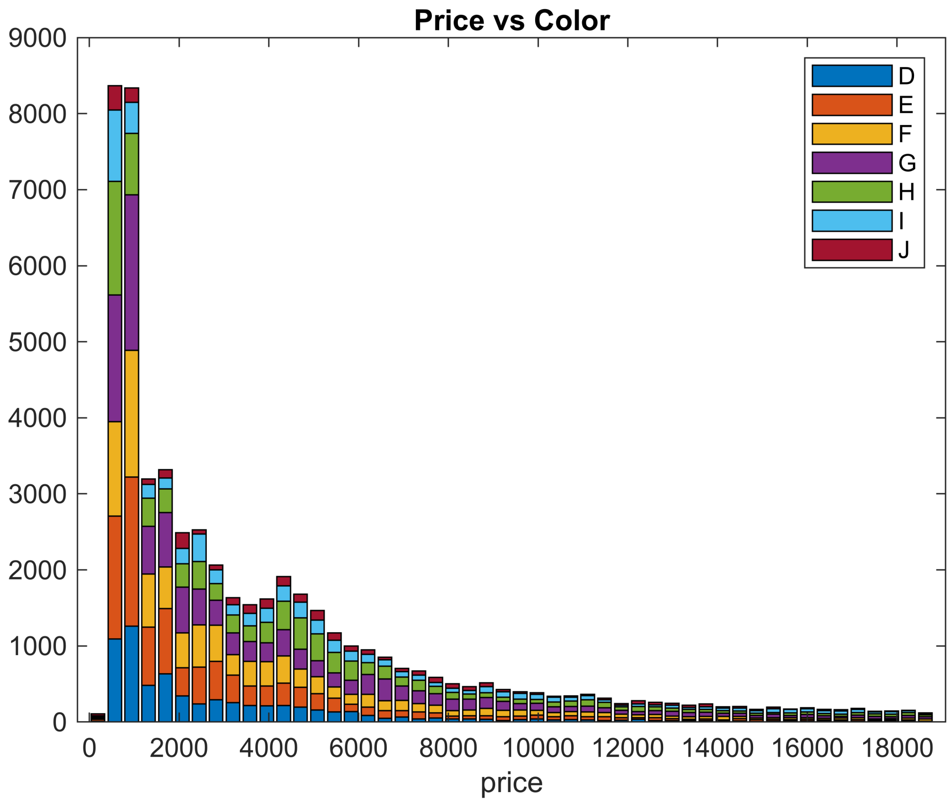

```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceCut_stacked.png') )
```

Color

```
color_categories = unique(diamonds.color);

figure
for ii = 1:size(color_categories)
    X(ii,:) = histcounts( diamonds.price(diamonds.color == color_categories(ii)), hist_edges);
    subplot(size(color_categories,1), 1, size(color_categories,1)+1-ii)
    histogram( diamonds.price(diamonds.color == color_categories(ii)), hist_edges, FaceColor=co
    legend(color_categories(ii))
    if ii ==1
        xlabel('price')
    end
end
title('Price vs Color')
```



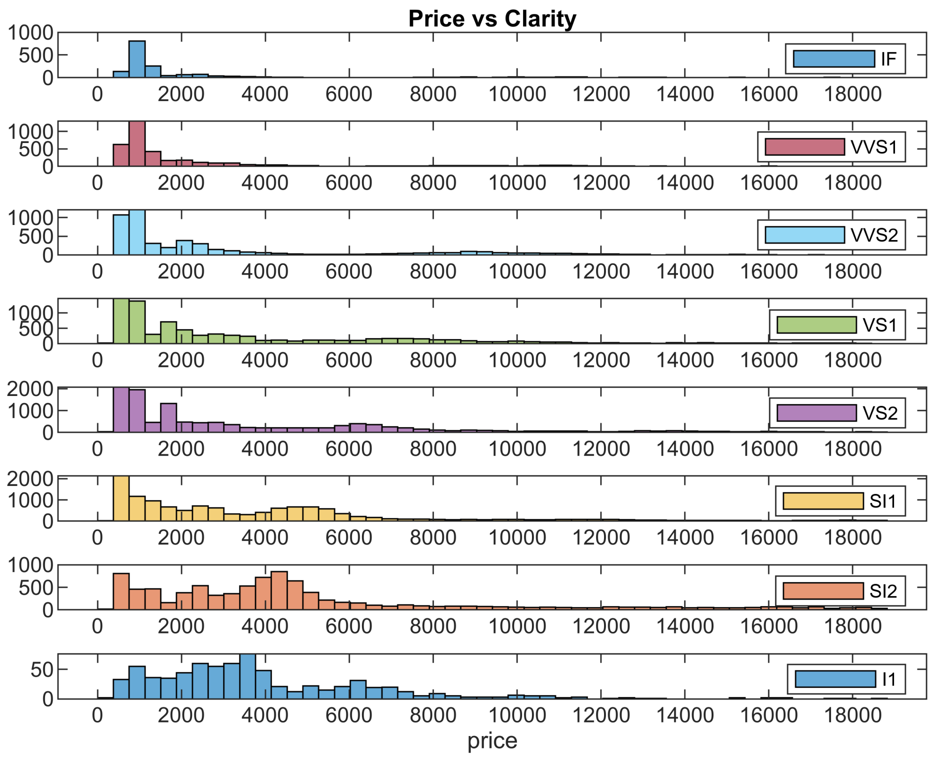

```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceColor.png') )

figure
bar(plot_edges, X, 'stacked')
xlabel('price')
legend(color_categories)
title('Price vs Color')
```



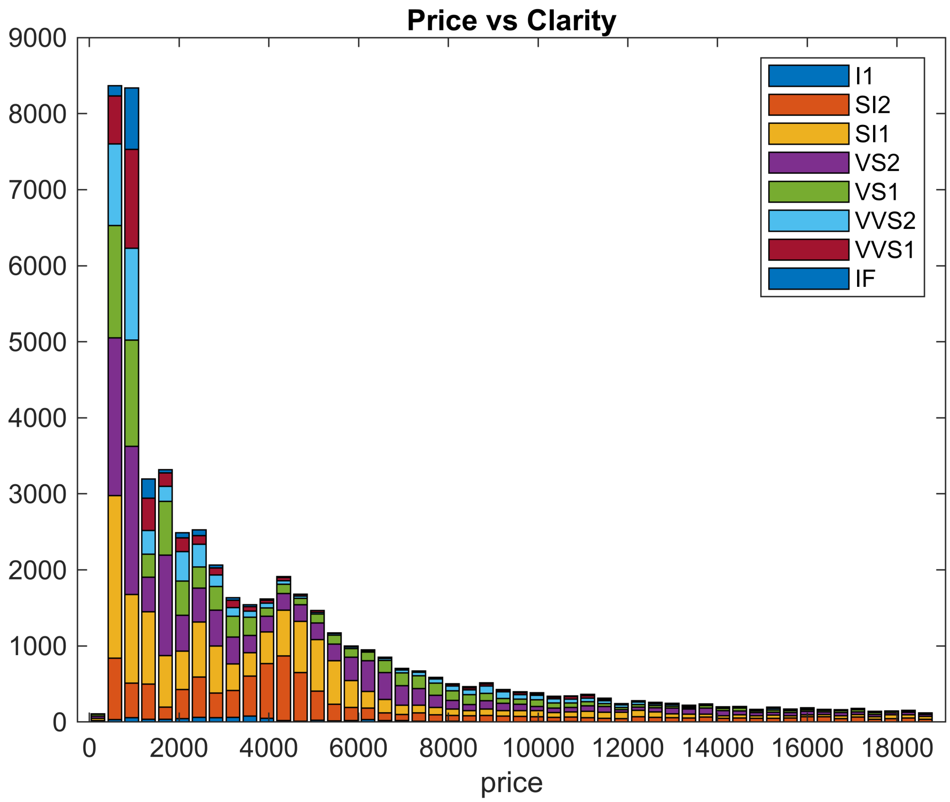

```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceColor_stacked.png') )
```

Clarity

```
clarity_categories = unique(diamonds.clarity);
clarity_categories = {'I1'; 'SI2'; 'SI1'; 'VS2'; 'VS1'; 'VVS2'; 'VVS1'; 'IF'};

figure
for ii = 1:size(clarity_categories)
    X(ii,:) = histcounts( diamonds.price(diamonds.clarity == clarity_categories(ii)), hist_edge
    subplot(size(clarity_categories,1), 1, size(clarity_categories,1)+1-ii)
    histogram( diamonds.price(diamonds.clarity == clarity_categories(ii)), hist_edges, FaceColo
    legend(clarity_categories(ii))
    if ii ==1
        xlabel('price')
    end
end
title('Price vs Clarity')
```



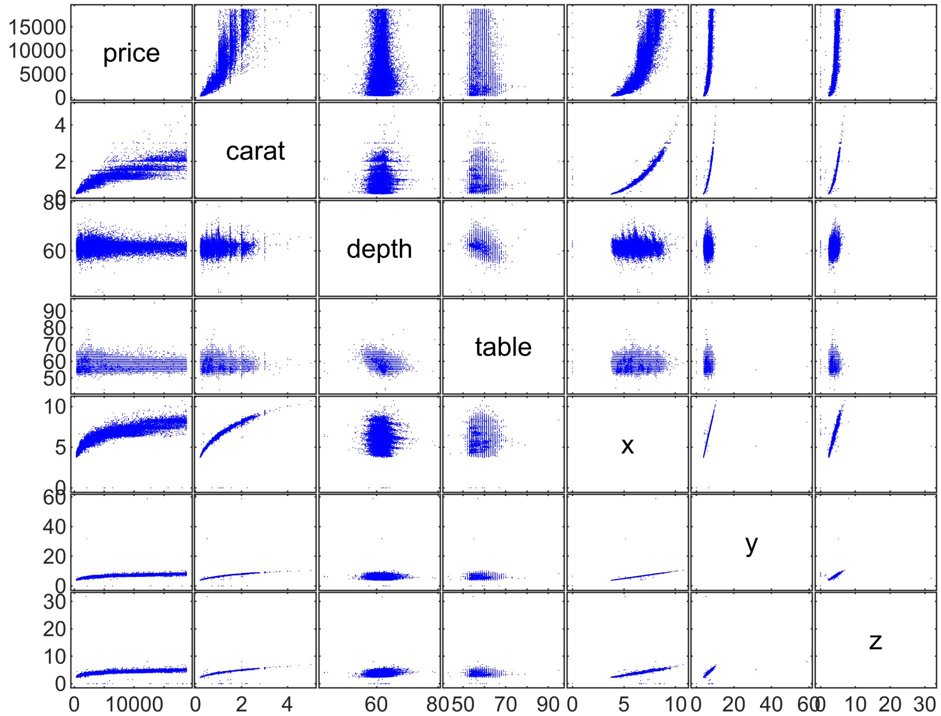

```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceClarity.png') )

figure
bar(plot_edges, X, 'stacked')
xlabel('price')
legend(clarity_categories)
title('Price vs Clarity')
```




```
% saveas(gcf,strcat('02_Diamonds/Hist_PriceClarity_stacked.png') )
```

Gplot of non-categorical variables (vs price)

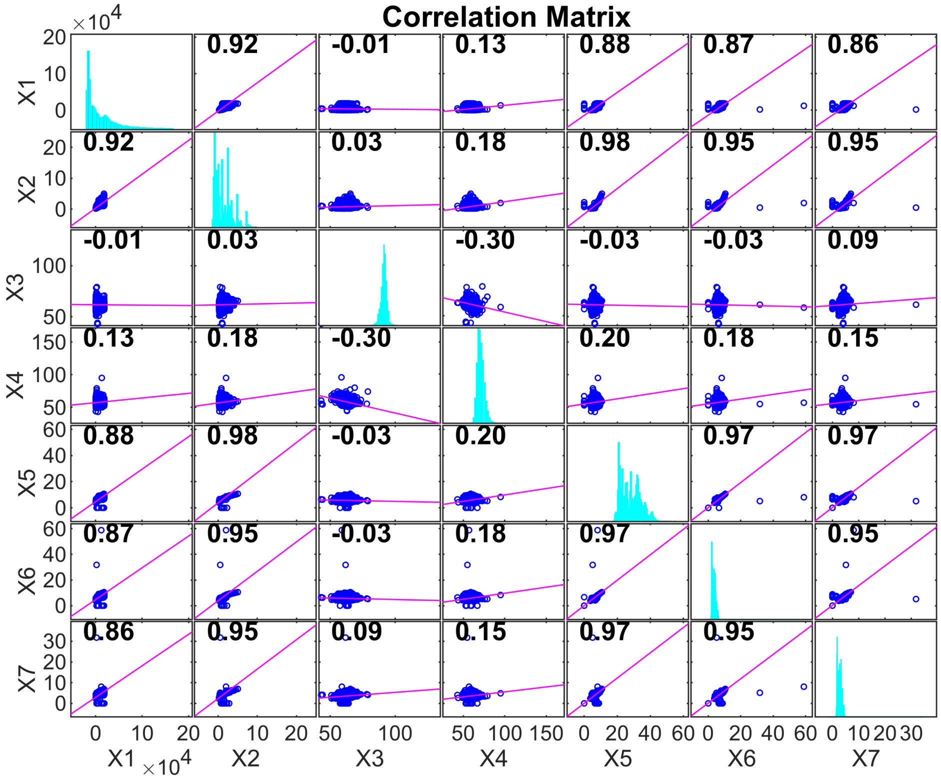
```
X = [diamonds.price diamonds.carat diamonds.depth diamonds.table diamonds.x diamonds.y diamonds

xnames = {'price', 'carat', 'depth', 'table', 'x', 'y', 'z'};

figure
[h, ax] = gplotmatrix( X,[],[],[],[],[],[],'variable',xnames);
```




```
% saveas(gcf,strcat('02_Diamonds/All_vars.png') )
```

Compute correlation

```
[R,PValue] = corrplot(array2table(X))
```



R = 7×7 table

	X1	X2	X3	X4	X5	X6	X7
1 X1	1	0.9216	-0.0106	0.1271	0.8844	0.8654	0.8612
2 X2	0.9216	1	0.0282	0.1816	0.9751	0.9517	0.9534
3 X3	-0.0106	0.0282	1	-0.2958	-0.0253	-0.0293	0.0949
4 X4	0.1271	0.1816	-0.2958	1	0.1953	0.1838	0.1509
5 X5	0.8844	0.9751	-0.0253	0.1953	1	0.9747	0.9708
6 X6	0.8654	0.9517	-0.0293	0.1838	0.9747	1	0.9520
7 X7	0.8612	0.9534	0.0949	0.1509	0.9708	0.9520	1

PValue = 7×7 table

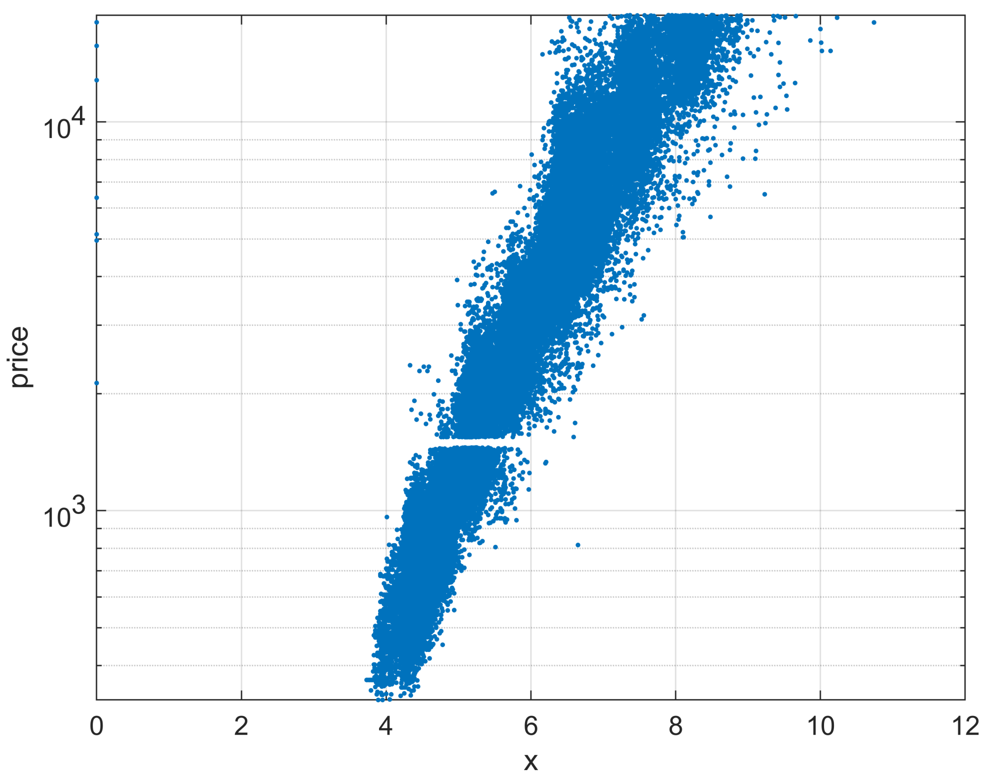
	X1	X2	X3	X4	X5	X6	X7
1 X1	1	0	0.0134	0	0	0	0
2 X2	0	1	0	0	0	0	0
3 X3	0.0134	0	1	0	0	0	0
4 X4	0	0	0	1	0	0	0
5 X5	0	0	0	0	1	0	0
6 X6	0	0	0	0	0	1	0
7 X7	0	0	0	0	0	0	1

```
% saveas(gcf,strcat('02_Diamonds/All_vars_corr.png') )
```

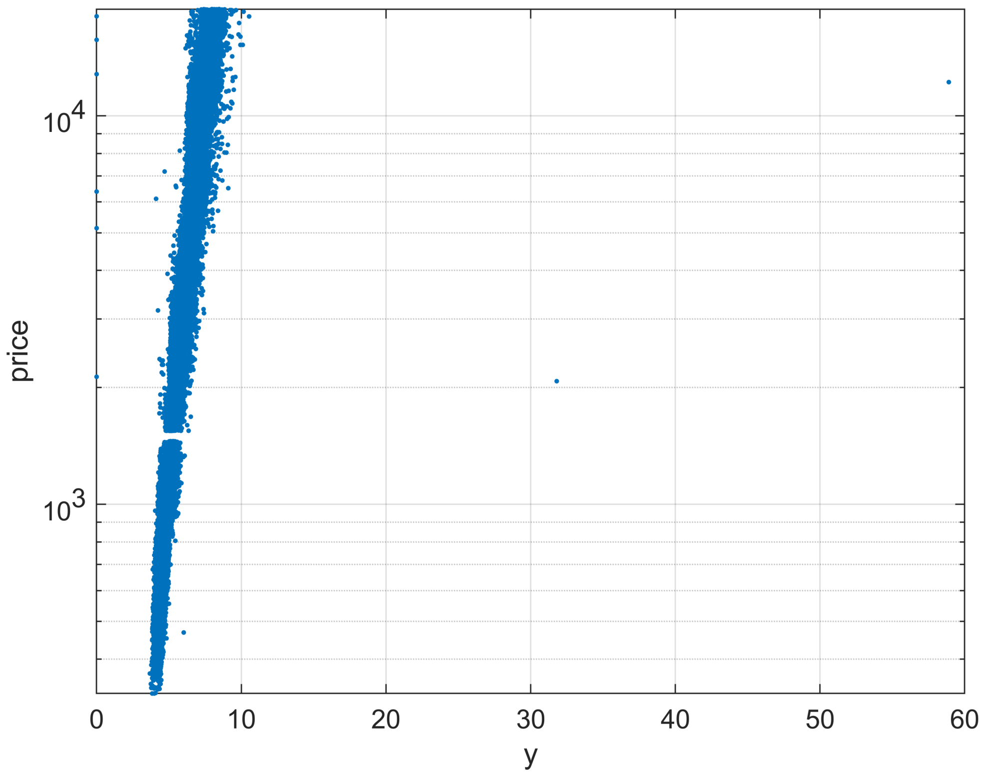
Depth and table don´t present a high influence on price, so these variables will be neglected from the model.

Focused at dimensions

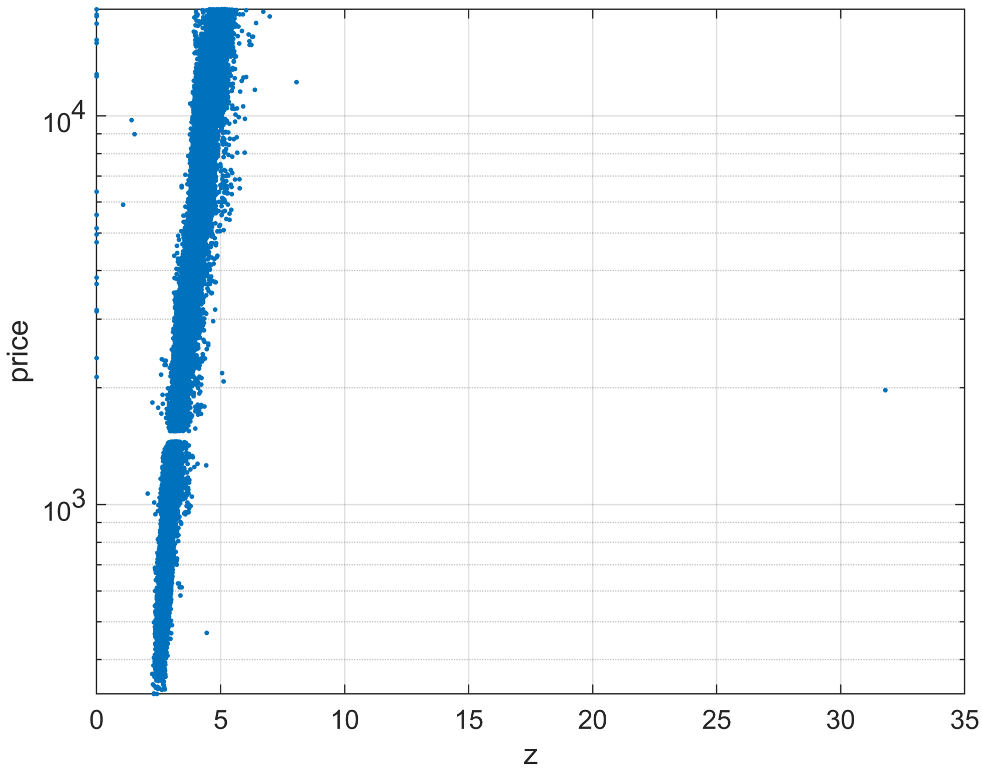
```
figure
semilogy(diamonds.x, diamonds.price, '.')
xlabel('x')
ylabel('price')
grid on
```




```
% saveas(gcf,strcat('02_Diamonds/x_unfilter.png') )

figure
semilogy(diamonds.y, diamonds.price, '.')
xlabel('y')
ylabel('price')
grid on
```




```
% saveas(gcf,strcat('02_Diamonds/y_unfilter.png') )

figure
semilogy(diamonds.z, diamonds.price, '.')
xlabel('z')
ylabel('price')
grid on
```




```
% saveas(gcf,strcat('02_Diamonds/z_unfilter.png') )
```

Apply filters

Remove zeroes from the table

```
idx = (diamonds.x > 0 & diamonds.y > 0 & diamonds.z > 0);
diamonds_filtered = diamonds(idx,:);
```

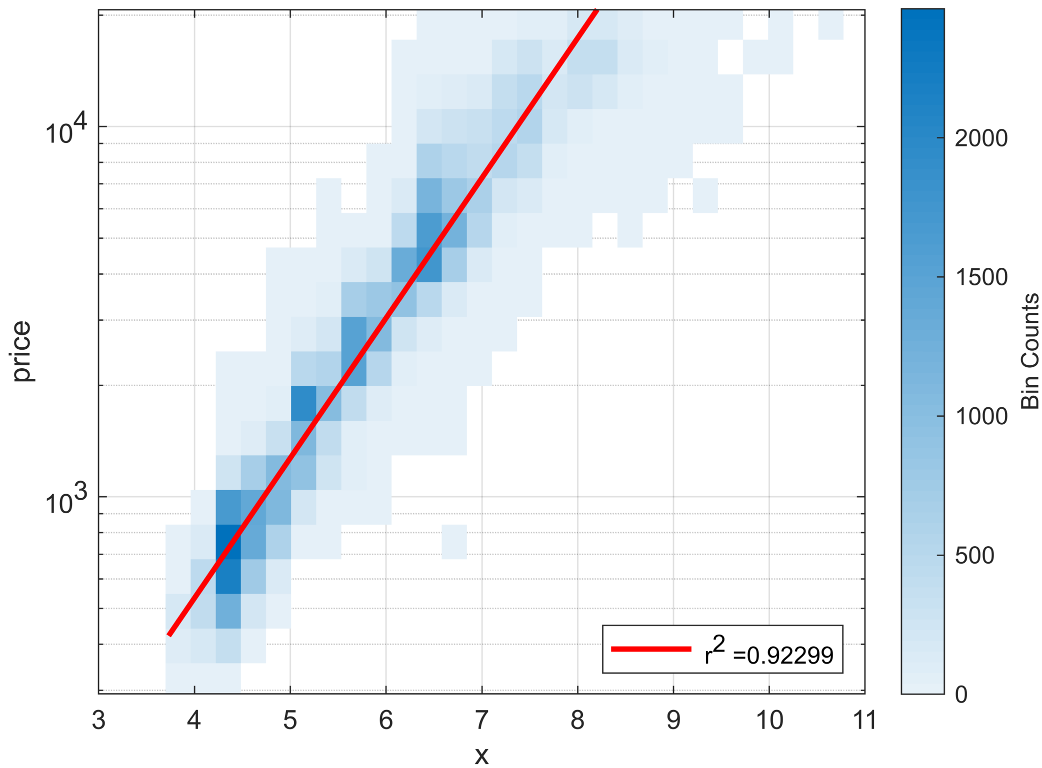
Remove clearly outliers from the data (y > 30 or z> 30)

```
idx = (diamonds_filtered.y < 30 & diamonds_filtered.z < 30);
diamonds_filtered = diamonds_filtered(idx,:);
```

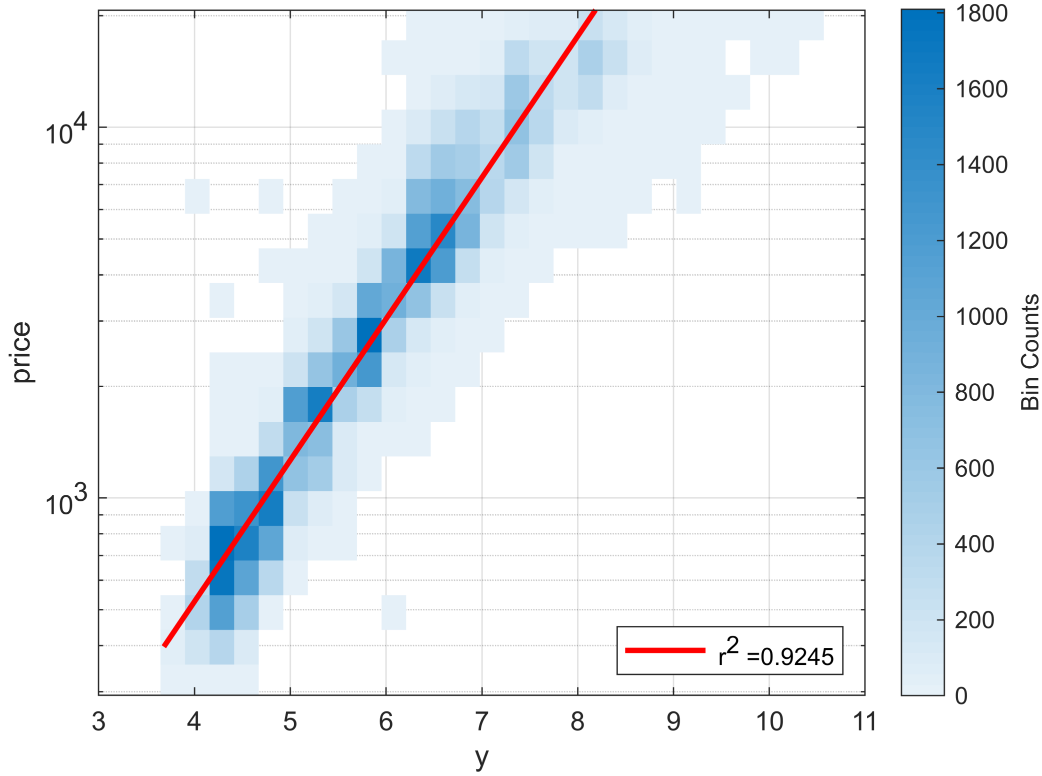
Plot filtered data

```
figure
% semilogy(diamonds_filtered.x, diamonds_filtered.price, '.')
binscatter(diamonds_filtered.x, diamonds_filtered.price)
set(gca,'YScale', 'log')
[f, gof] = fit(diamonds_filtered.x, log(diamonds_filtered.price), 'poly1');
x1 = linspace(min(diamonds_filtered.x), max(diamonds_filtered.x), 1001);
hold on
semilogy(x1, exp(f.p1*x1 + f.p2), 'r', LineWidth=2)
hold off
```

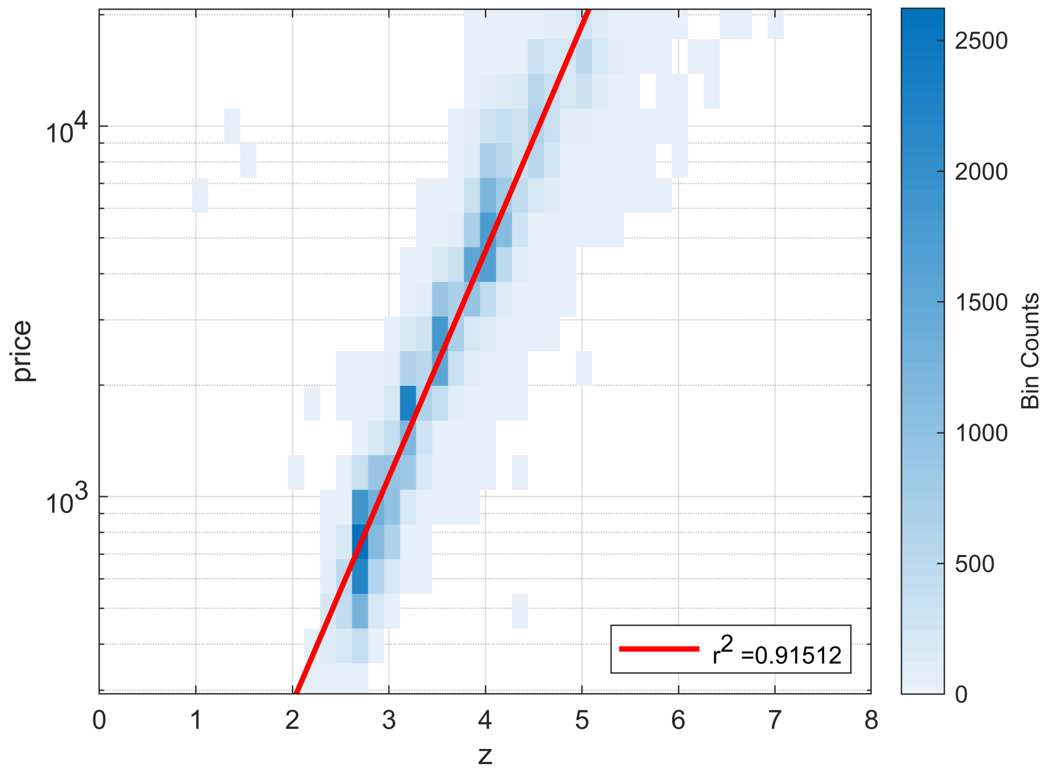
```
xlabel('x')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
```




```
% saveas(gcf,strcat('02_Diamonds/x_filter.png') )

figure
% semilogy(diamonds_filtered.y, diamonds_filtered.price, '.')
binscatter(diamonds_filtered.y, diamonds_filtered.price)
set(gca,'YScale', 'log')
[f, gof] = fit(diamonds_filtered.y, log(diamonds_filtered.price), 'poly1');
x1 = linspace(min(diamonds_filtered.y), max(diamonds_filtered.y), 1001);
hold on
semilogy(x1, exp(f.p1*x1 + f.p2), 'r', LineWidth=2)
hold off
xlabel('y')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
```



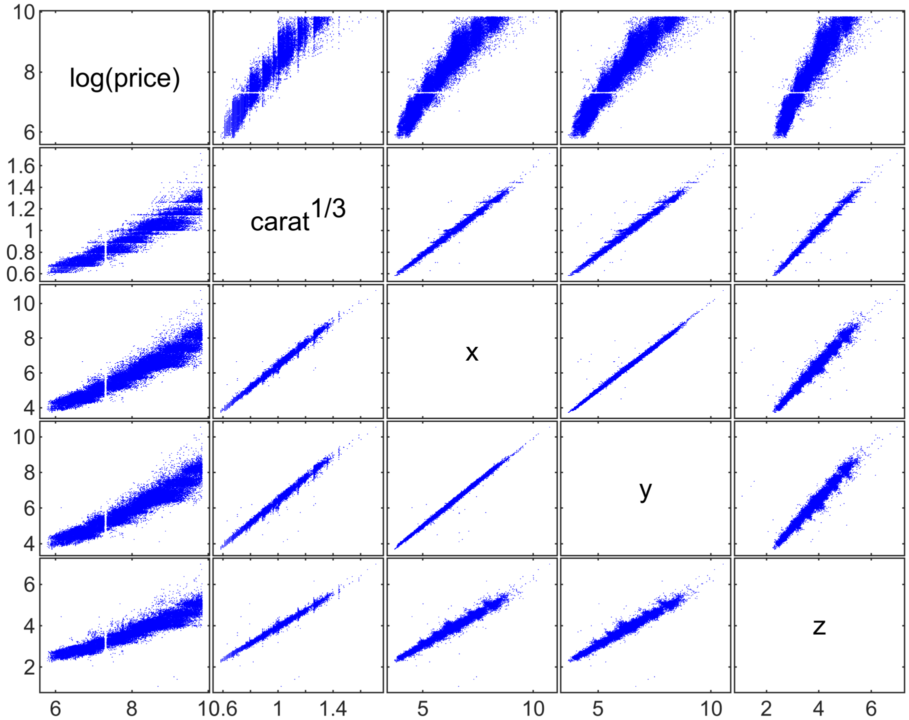

```
% saveas(gcf,strcat('02_Diamonds/y_filter.png') )

figure
% semilogy(diamonds_filtered.z, diamonds_filtered.price, '.')
binscatter(diamonds_filtered.z, diamonds_filtered.price)
set(gca,'YScale', 'log')
[f, gof] = fit(diamonds_filtered.z, log(diamonds_filtered.price), 'poly1');
x1 = linspace(min(diamonds_filtered.z), max(diamonds_filtered.z), 1001);
hold on
semilogy(x1, exp(f.p1*x1 + f.p2), 'r', LineWidth=2)
hold off
xlabel('z')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
```




```
% saveas(gcf,strcat('02_Diamonds/z_filter.png') )
```

Are the 3 variables independent? And carat?

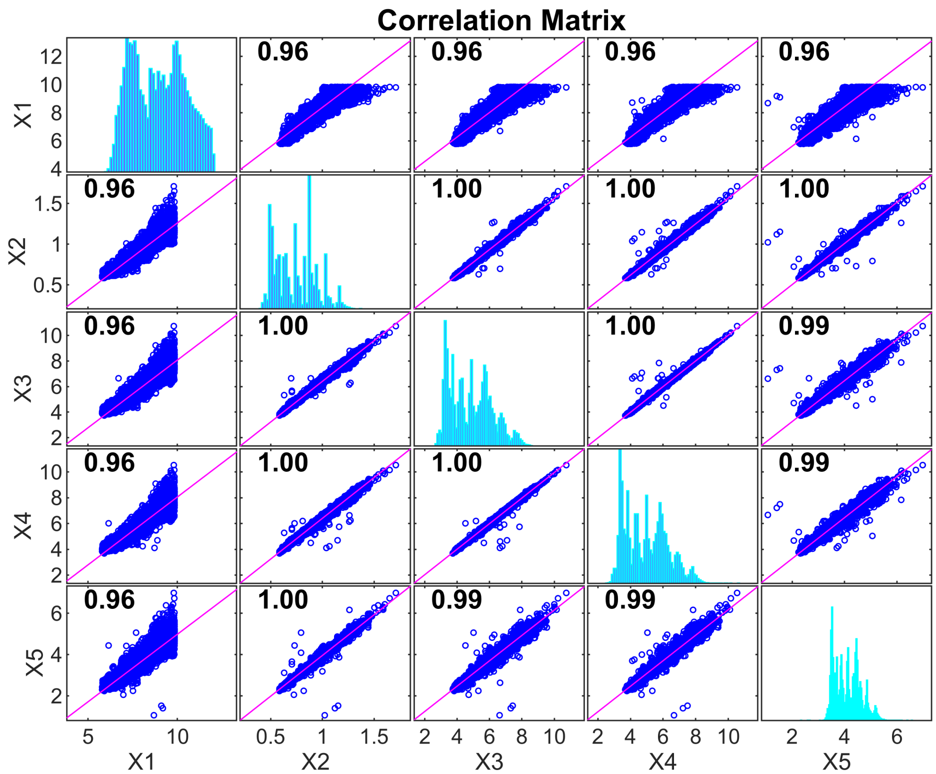
```
X = [log(diamonds_filtered.price) diamonds_filtered.carat.^(1/3) diamonds_filtered.x diamonds_f

xnames = {'log(price)', 'carat^{1/3}', 'x', 'y', 'z'};

[h, ax] = gplotmatrix( X,[],[],[],[],[],[],'variable',xnames);
```




```
% saveas(gcf,strcat('02_Diamonds/Linear_vars.png') )
```

Compute correlation

```
[R,PValue] = corrplot(array2table(X))
```



R = 5×5 table

	X1	X2	X3	X4	X5
1 X1	1	0.9611	0.9607	0.9615	0.9566
2 X2	0.9611	1	0.9976	0.9971	0.9964
3 X3	0.9607	0.9976	1	0.9987	0.9911
4 X4	0.9615	0.9971	0.9987	1	0.9907
5 X5	0.9566	0.9964	0.9911	0.9907	1

PValue = 5×5 table

	X1	X2	X3	X4	X5
1 X1	1	0	0	0	0
2 X2	0	1	0	0	0
3 X3	0	0	1	0	0
4 X4	0	0	0	1	0
5 X5	0	0	0	0	1

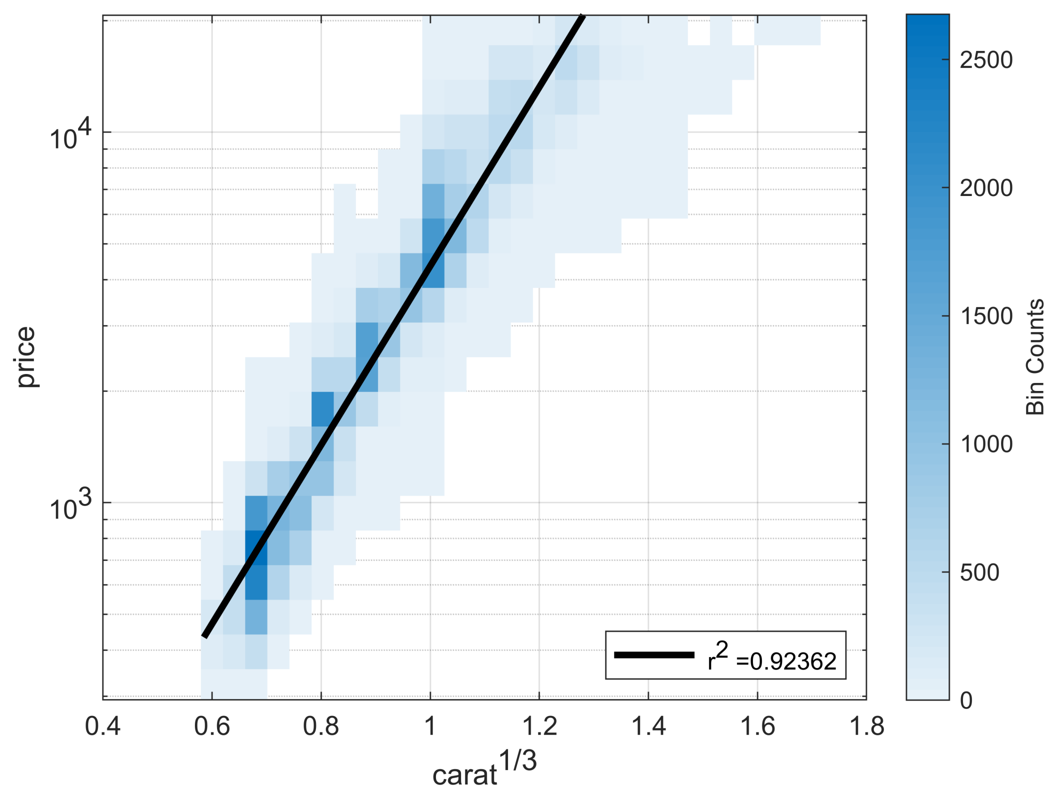
```
% saveas(gcf,strcat('02_Diamonds/Linear_vars_corr.png') )
```

As diamond shape seems to be standard, x, y & z are linearly related, while carat (which is a measure of their mass) scales with x^3

This means that among all the numerical variables

Regression with carat

```
figure
% semilogy(diamonds_filtered.carat, diamonds_filtered.price, '.')
binscatter(diamonds_filtered.carat.^(1/3), diamonds_filtered.price)
set(gca,'YScale', 'log')
[f, gof] = fit(diamonds_filtered.carat.^(1/3), log(diamonds_filtered.price), 'poly1');
x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 1001);
hold on
semilogy(x1, exp(f.p1*x1 + f.p2), 'k', LineWidth=2.5)
hold off
xlabel('carat^{1/3}')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
```

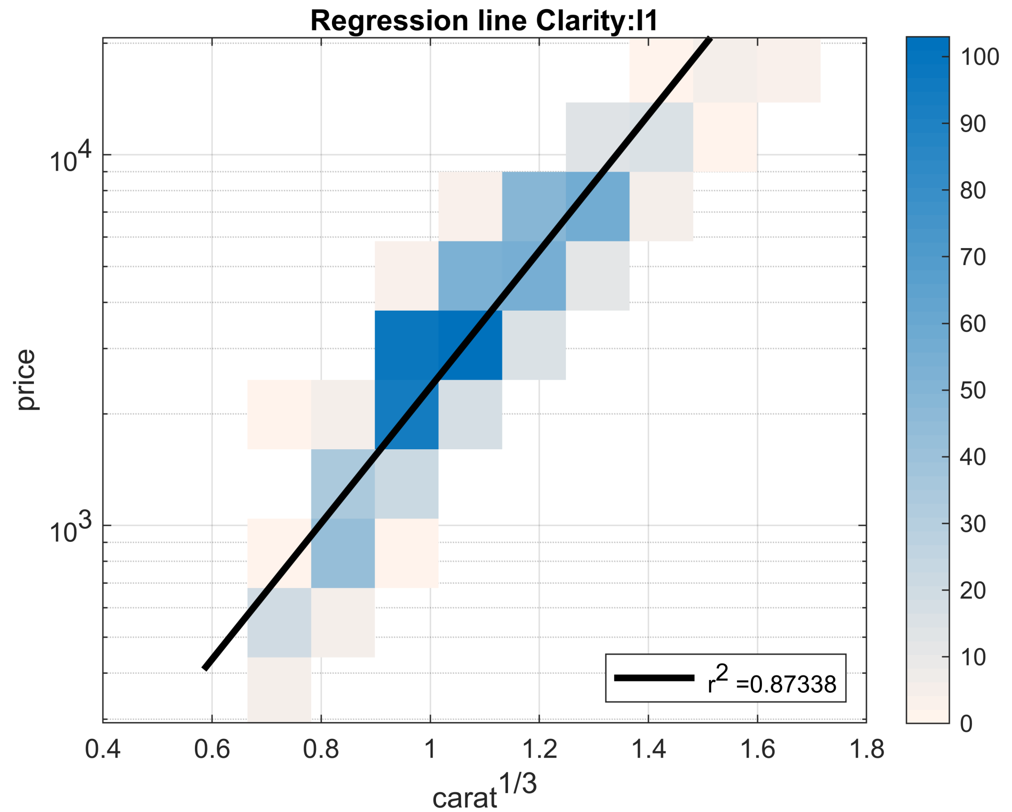


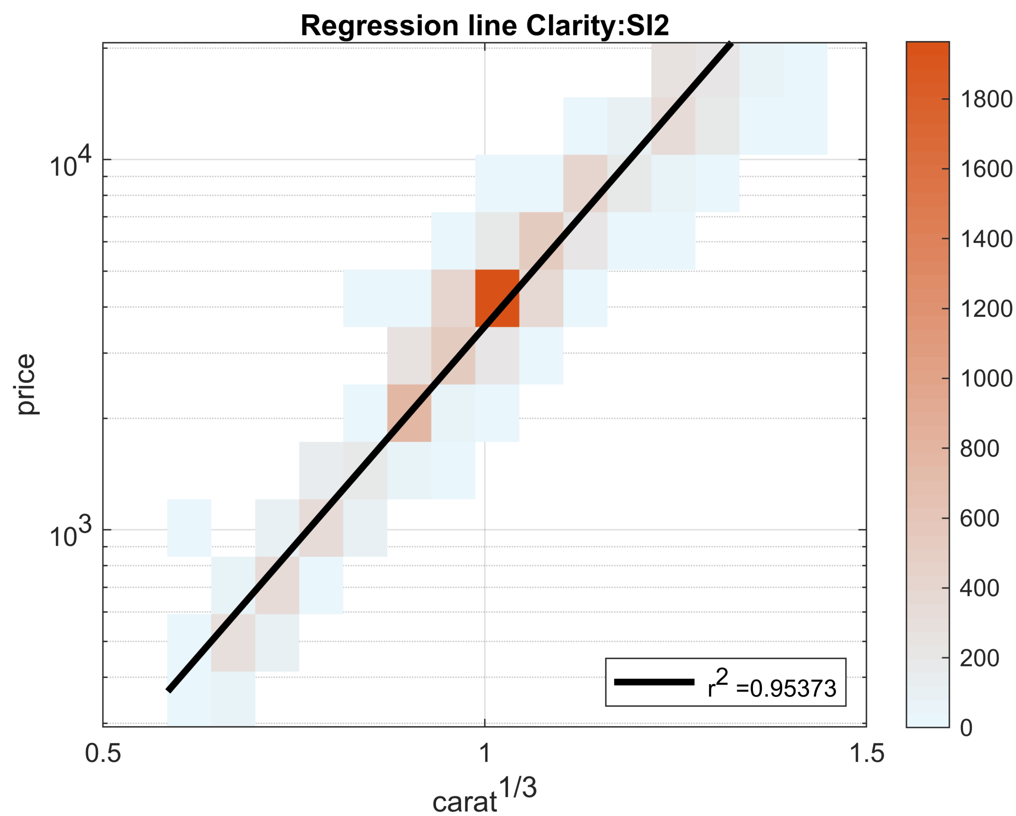

```
% saveas(gcf,strcat('02_Diamonds/carat_regression.png') )
```

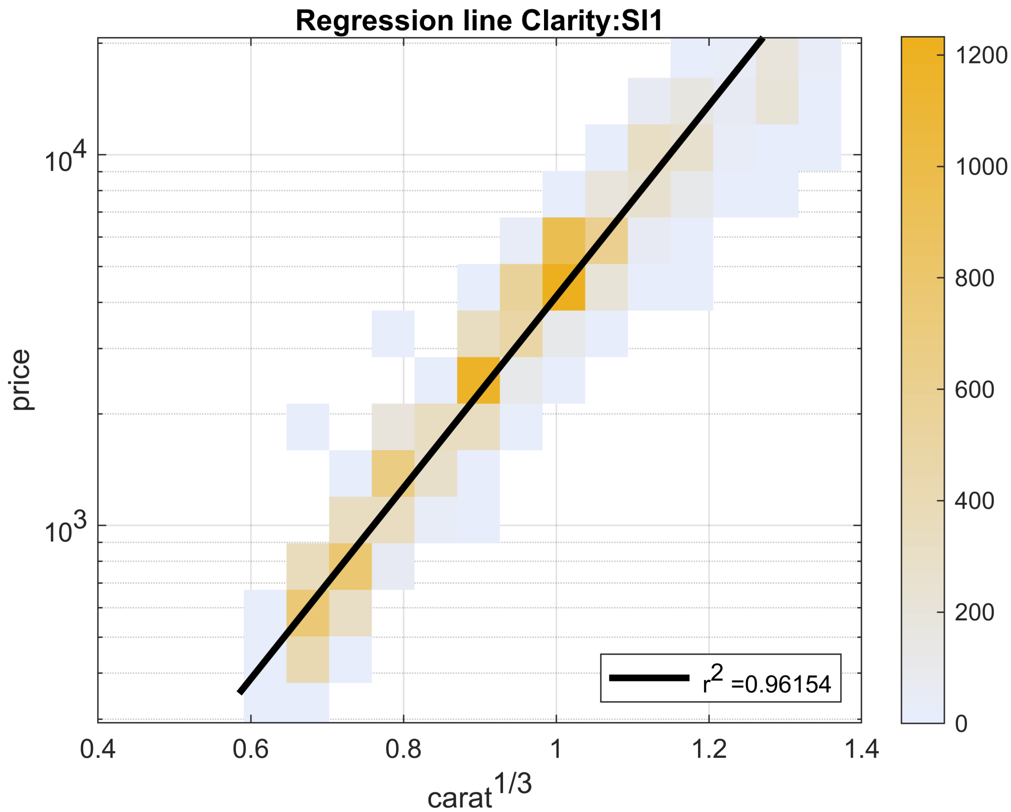
Include clarity in the equation

```
clarity_categories = unique(diamonds_filtered.clarity);
clarity_categories = {'I1'; 'SI2'; 'SI1'; 'VS2'; 'VS1'; 'VVS2'; 'VVS1'; 'IF'};
colors = [[0, 0.4470, 0.7410],
```

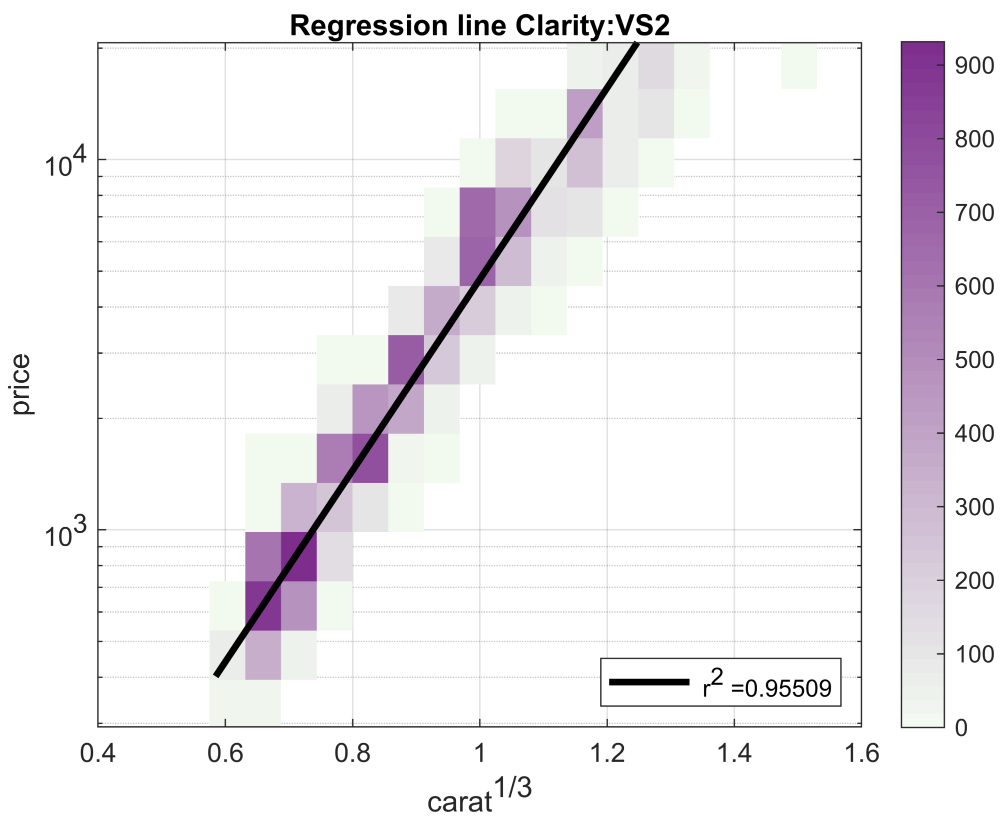
```
          [0.8500, 0.3250, 0.0980],
          [0.9290, 0.6940, 0.1250],
          [0.4940, 0.1840, 0.5560],
          [0.4660, 0.6740, 0.1880],
          [0.3010, 0.7450, 0.9330],
          [0.6350, 0.0780, 0.1840],
          [0, 0.4470, 0.7410]];

for ii = 1:size(clarity_categories)
    idx = diamonds_filtered.clarity==clarity_categories(ii);

    figure
    binscatter(diamonds_filtered.carat(idx).^(1/3), diamonds_filtered.price(idx))
    set(gca,'YScale', 'log')
    colormap_graded(colors(ii,:));
    colorbar
    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)), 'pol
    x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 100
    hold on
    semilogy(x1, exp(f.p1*x1 + f.p2), 'k', LineWidth=2.5)
    hold off
    xlabel('carat^{1/3}')
    ylabel('price')
    grid on
    ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
    legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
    title(strcat('Regression line Clarity: ',string(clarity_categories(ii))))
%     saveas(gcf,strcat('02_Diamonds/carat_regression_',string(clarity_categories(ii)),'.png')
end
```


Regression line Clarity:I1
price
carat$^{1/3}$
r^2 =0.87338
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
10^3
10^4
0
10
20
30
40
50
60
70
80
90
100

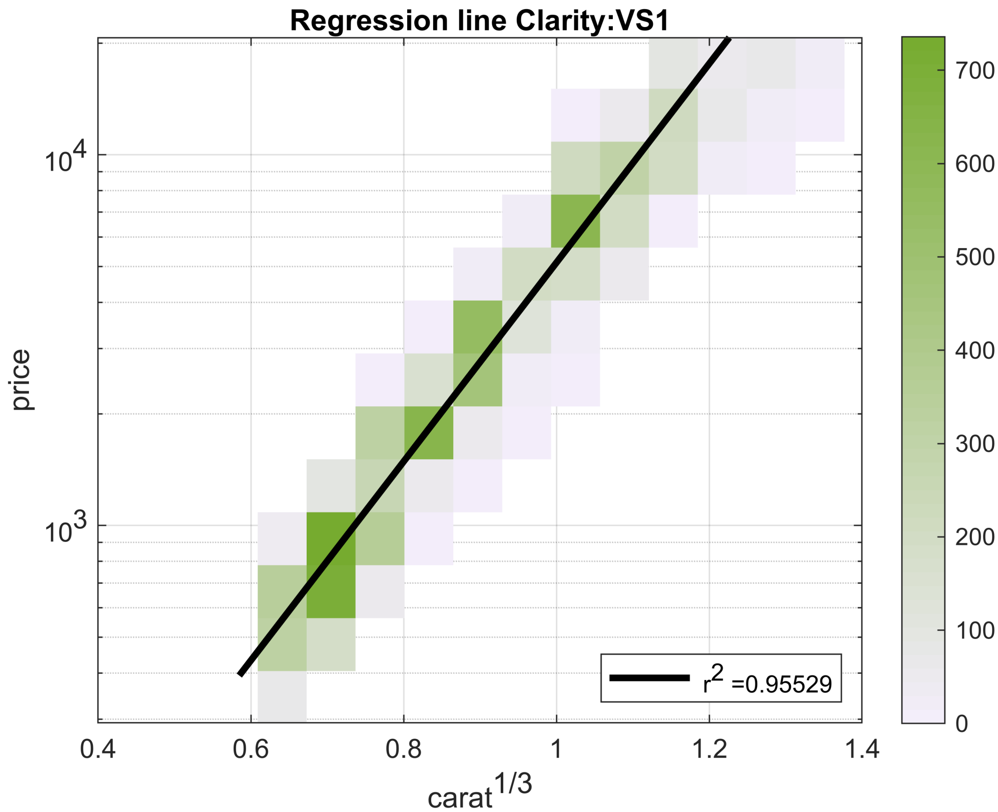

Regression line Clarity:SI2
price
carat$^{1/3}$
r^2 =0.95373
0.5
1
1.5
10^3
10^4
0
200
400
600
800
1000
1200
1400
1600
1800



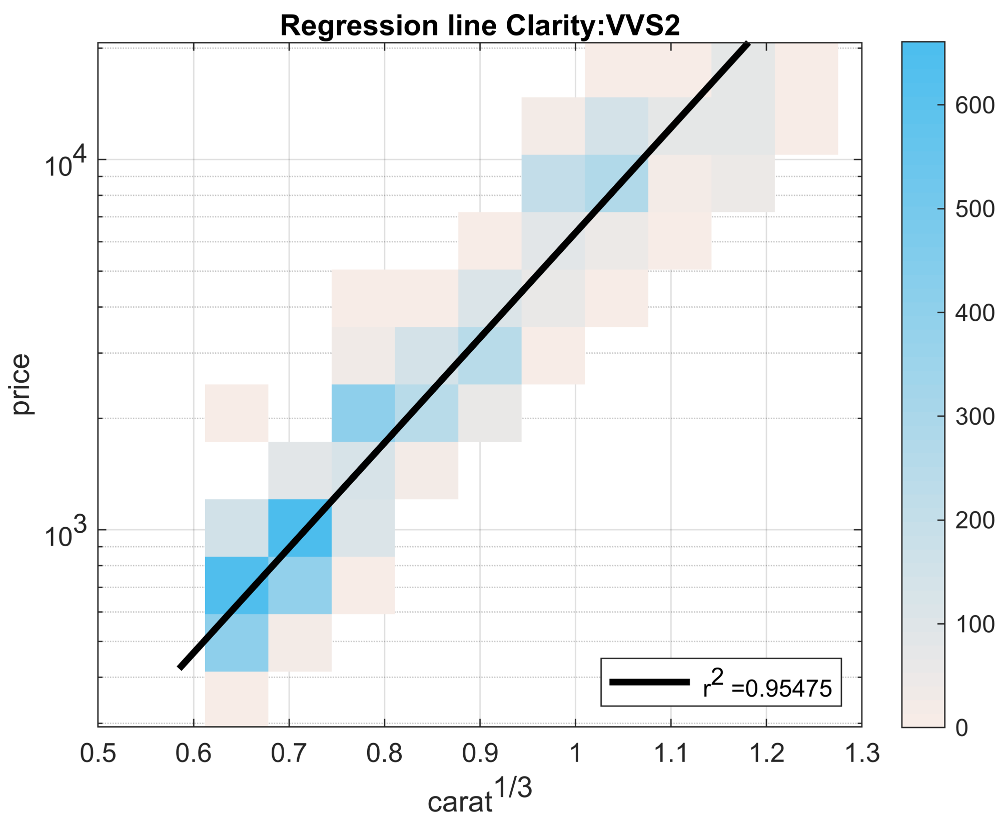
Regression line Clarity:SI1
price
carat$^{1/3}$
r^2 =0.96154
0.4
0.6
0.8
1
1.2
1.4
10^3
10^4
0
200
400
600
800
1000
1200



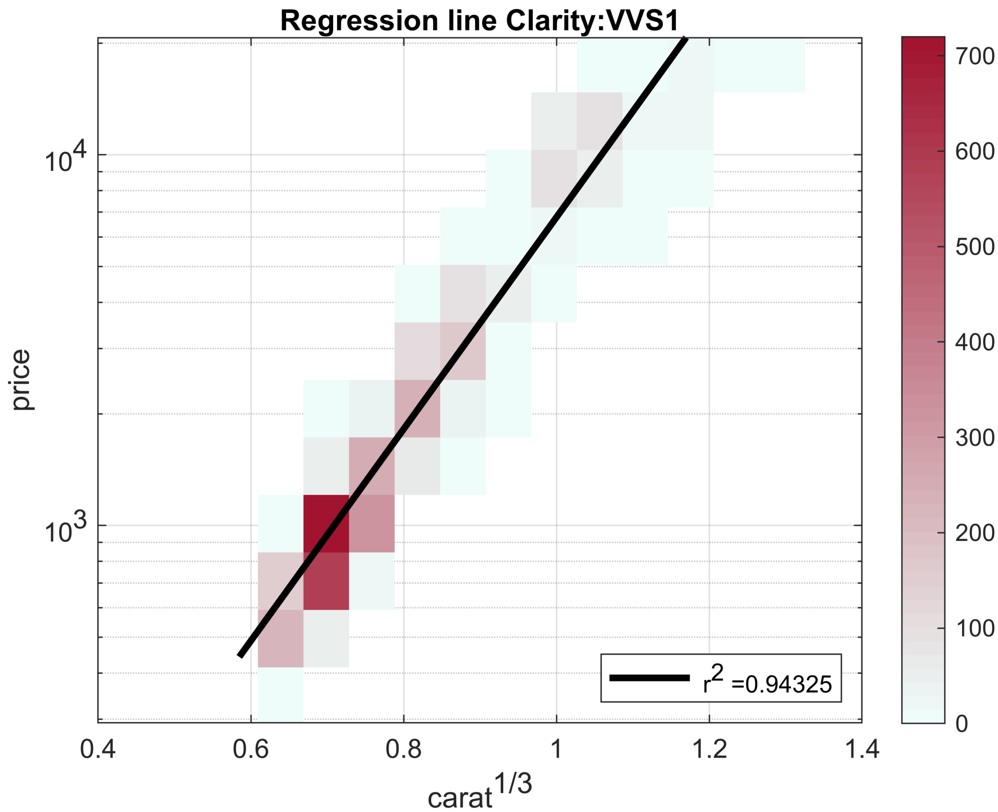
Regression line Clarity:VS2
price
carat$^{1/3}$
r^2 =0.95509
0.4
0.6
0.8
1
1.2
1.4
1.6
10^3
10^4
0
100
200
300
400
500
600
700
800
900

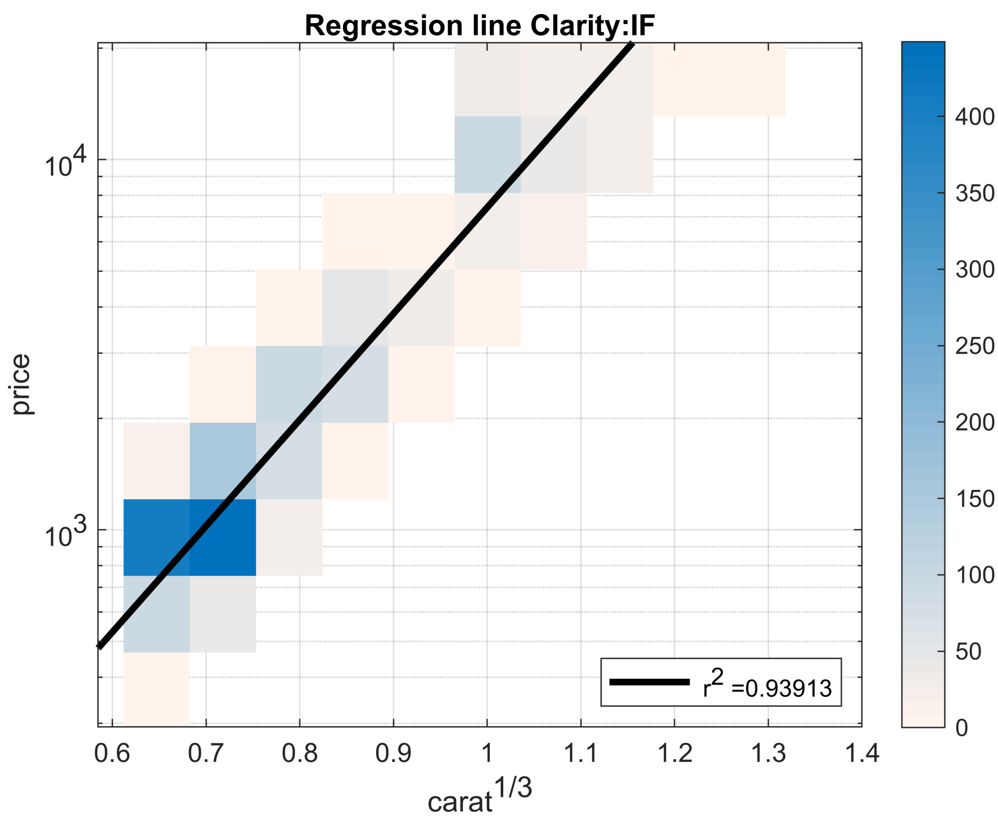


Regression line Clarity:VS1
price
10^4
10^3
0.4
0.6
0.8
1
1.2
1.4
$carat^{1/3}$
r^2 =0.95529
700
600
500
400
300
200
100
0



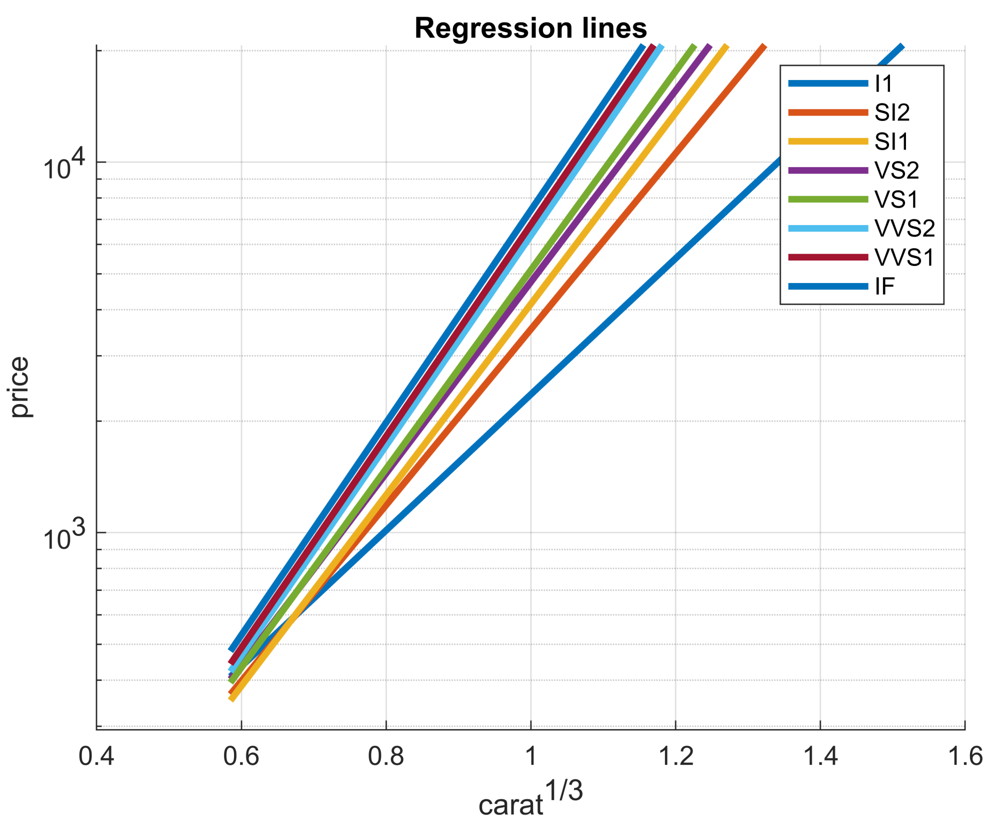
Regression line Clarity:VVS2
price
10^4
10^3
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
$carat^{1/3}$
r^2 =0.95475
600
500
400
300
200
100
0





Now gather all the regression lines

```
x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 1001);

figure
hold on
for ii = 1:size(clarity_categories)
    idx = diamonds_filtered.clarity==clarity_categories(ii);

    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)), 'po
    semilogy(x1, exp(f.p1*x1 + f.p2), color=colors(ii,:), LineWidth=2.5)
end
hold off
set(gca,'YScale', 'log')
xlabel('carat^{1/3}')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend(clarity_categories)
title(strcat('Regression lines'))
```

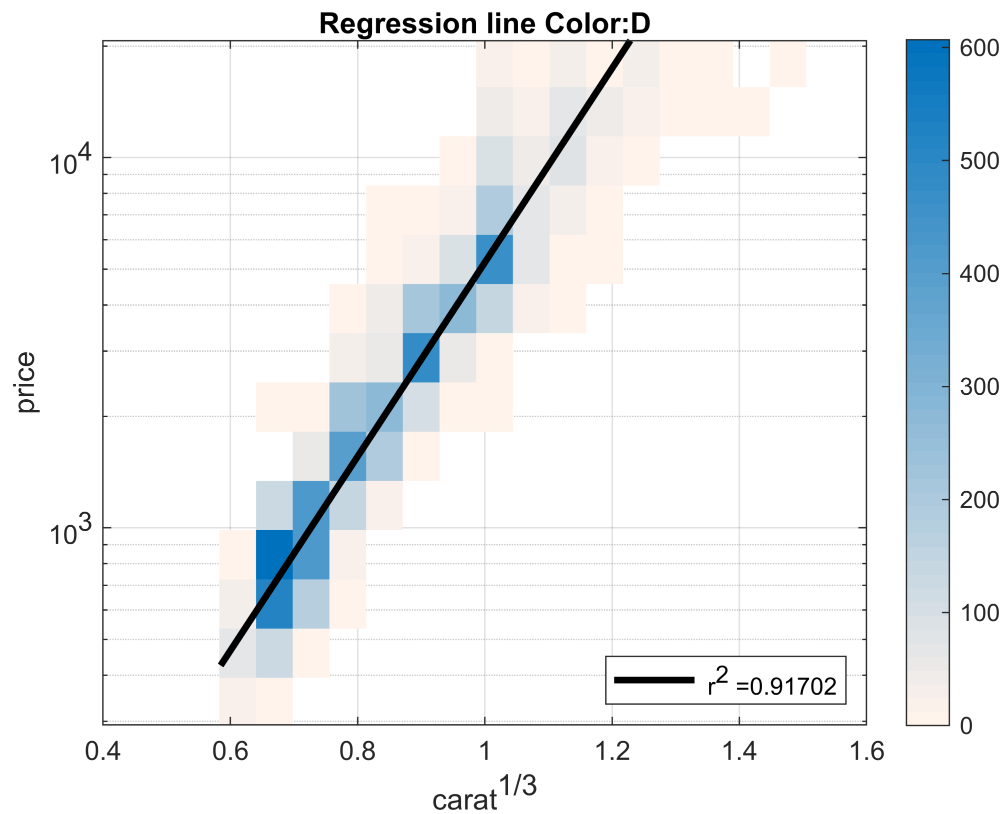


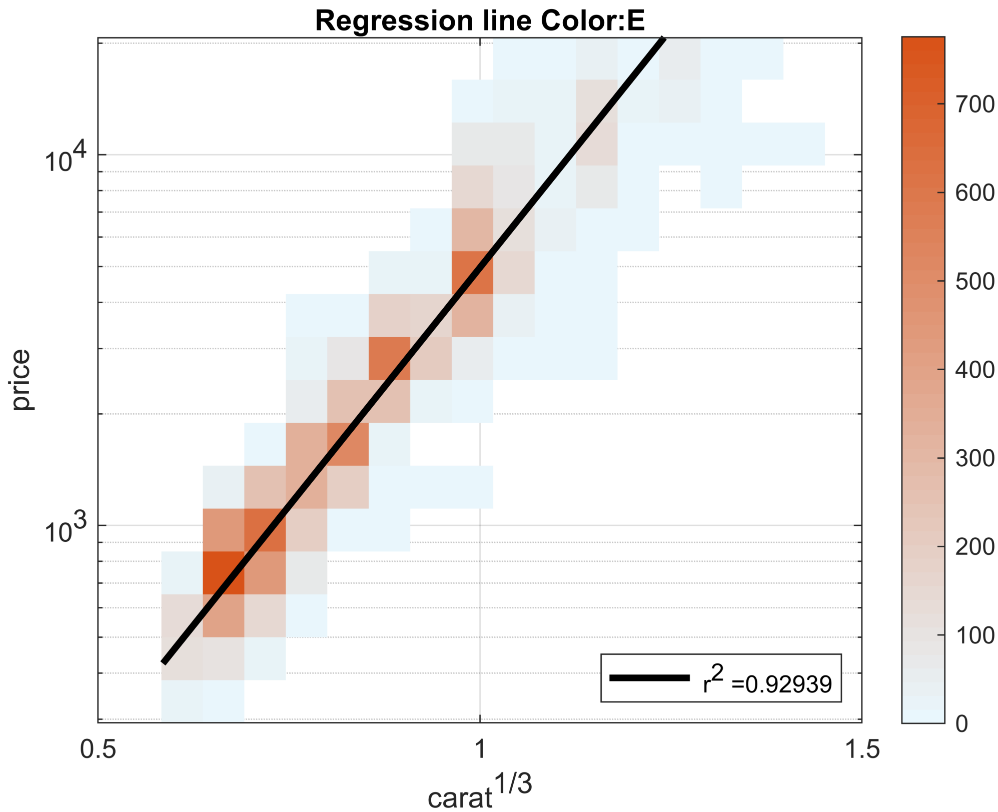

```
% saveas(gcf,strcat('02_Diamonds/carat_regression_all.png') )
```

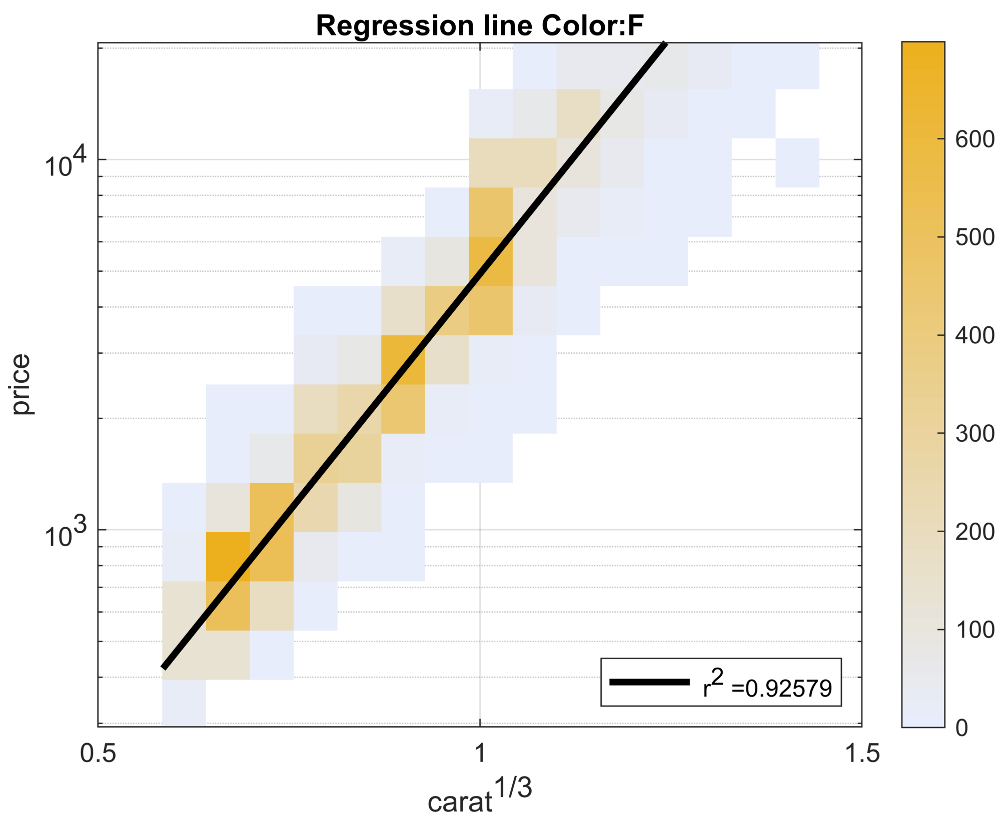
Test for Color

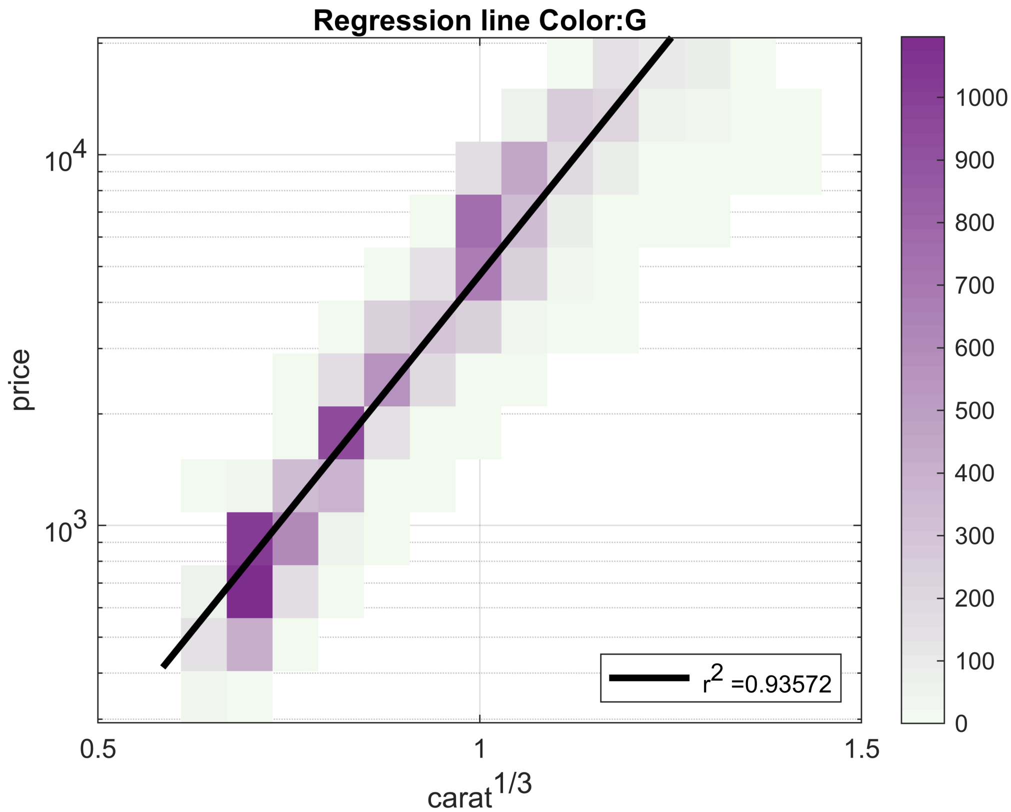
```
color_categories = unique(diamonds_filtered.color);
```

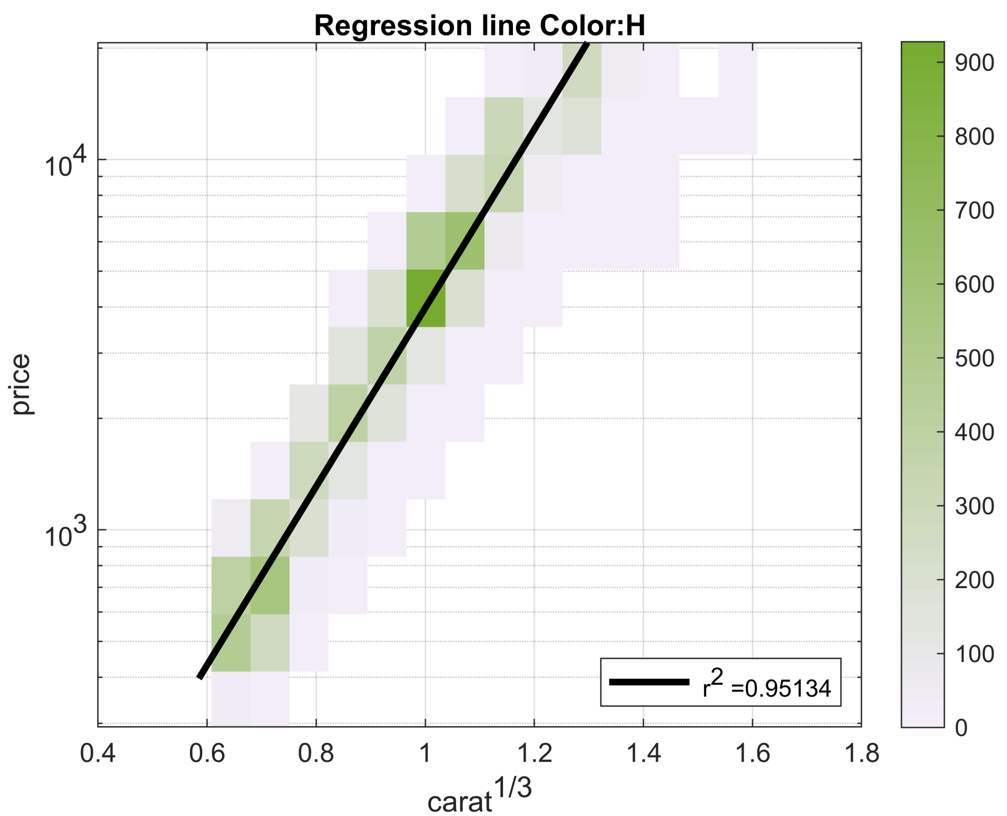
```
for ii = 1:size(color_categories)
    idx = diamonds_filtered.color==color_categories(ii);

    figure
    binscatter(diamonds_filtered.carat(idx).^(1/3), diamonds_filtered.price(idx))
    set(gca,'YScale', 'log')
    colormap_graded(colors(ii,:));
    colorbar
    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)), 'pol
    x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 100
    hold on
    semilogy(x1, exp(f.p1*x1 + f.p2), 'k', LineWidth=2.5)
    hold off
    xlabel('carat^{1/3}')
    ylabel('price')
    grid on
    ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
    legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
    title(strcat('Regression line Color: ',string(color_categories(ii))))
%     saveas(gcf,strcat('02_Diamonds/carat_regression_color',string(color_categories(ii)),'.png
end
```

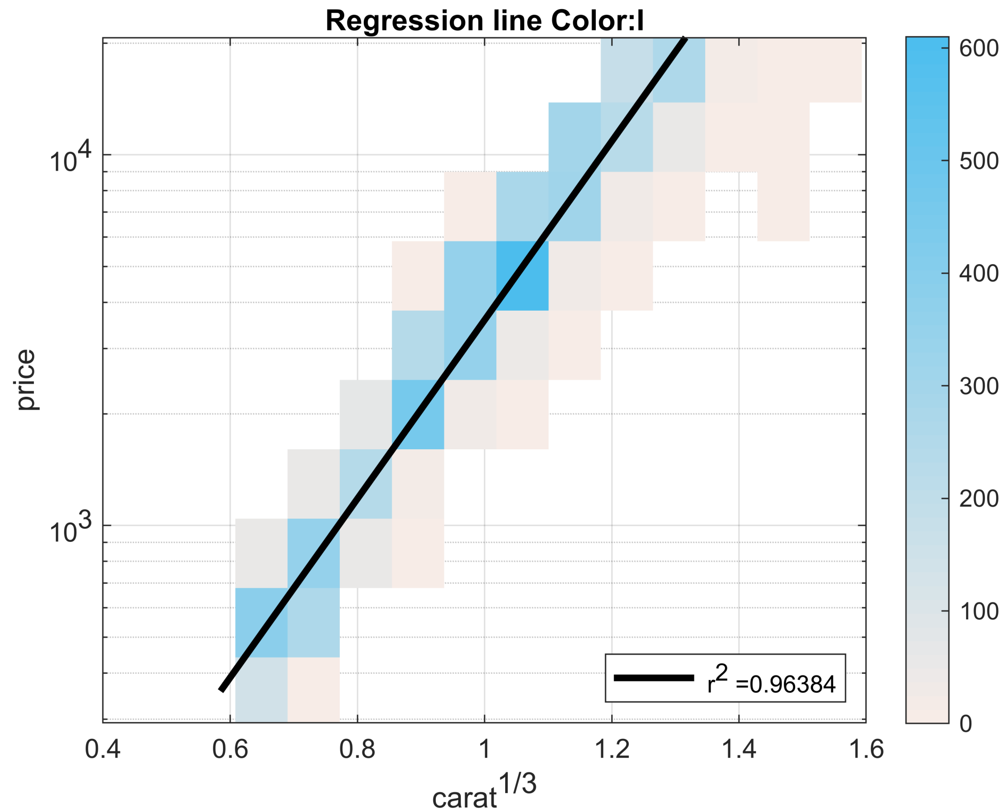



Regression line Color:E
price
carat^1/3
r^2 =0.92939
0.5
1
1.5
700
600
500
400
300
200
100
0

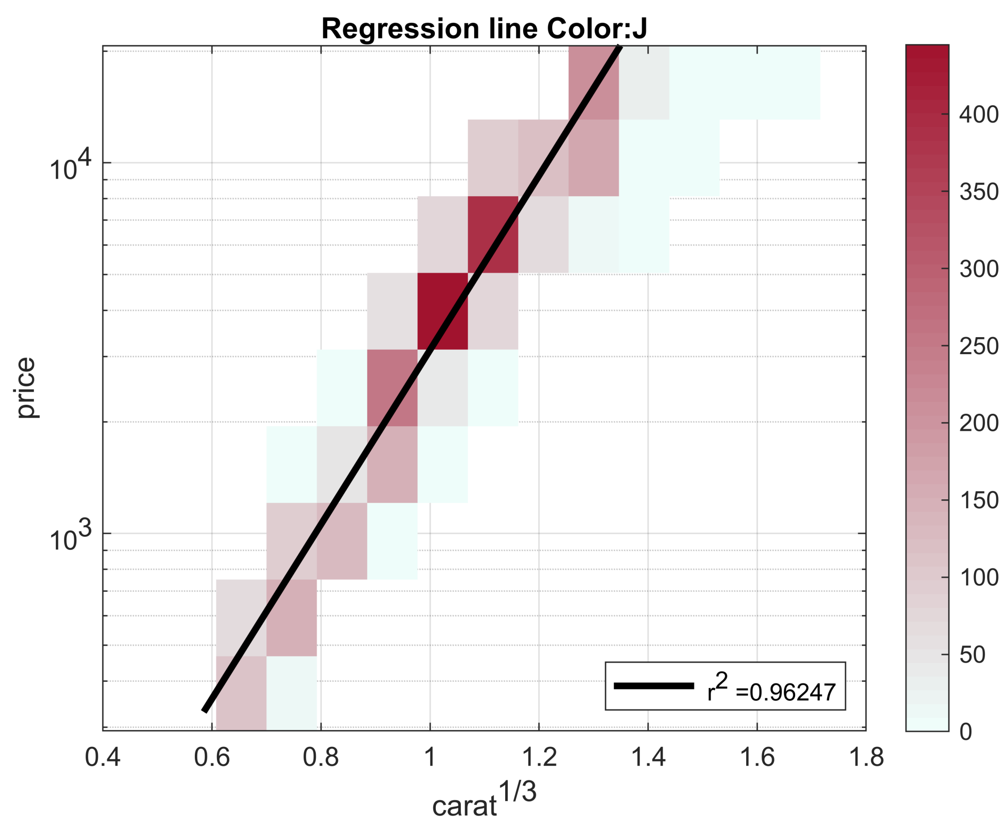

Regression line Color:F
price
carat^1/3
r^2 =0.92579
0.5
1
1.5
600
500
400
300
200
100
0


Regression line Color:G
price
carat^1/3
r^2 =0.93572


Regression line Color:H
price
carat^1/3
r^2 =0.95134

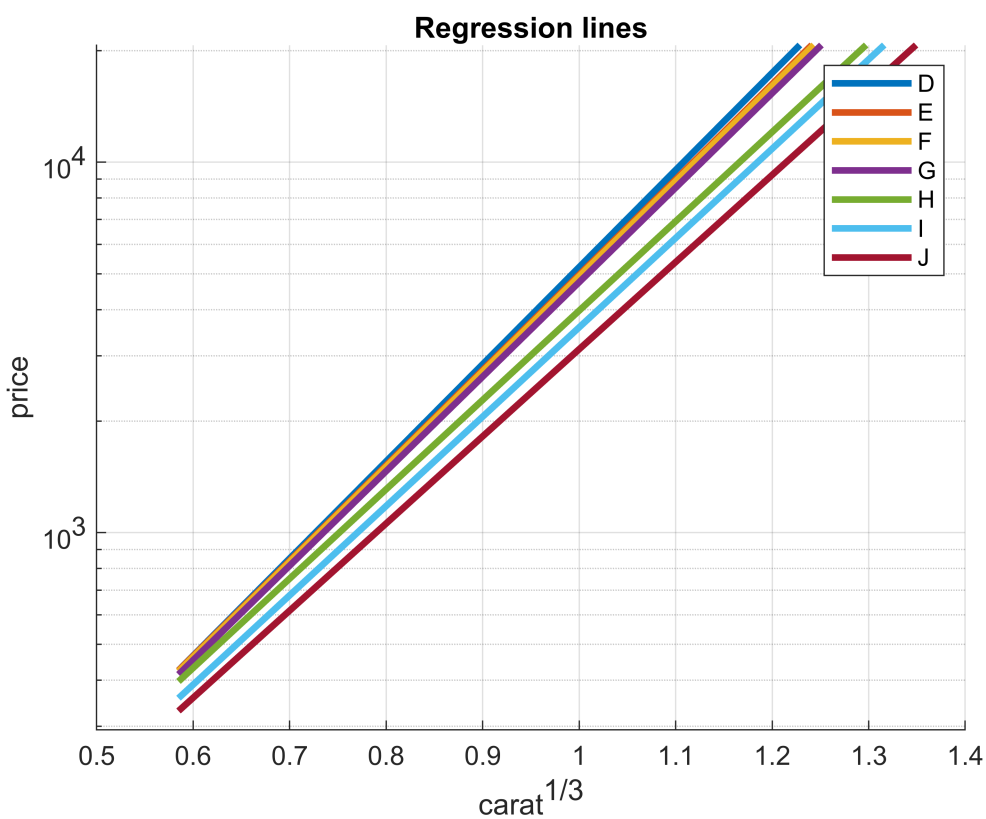


Regression line Color:I
price
carat1/3
r2 =0.96384



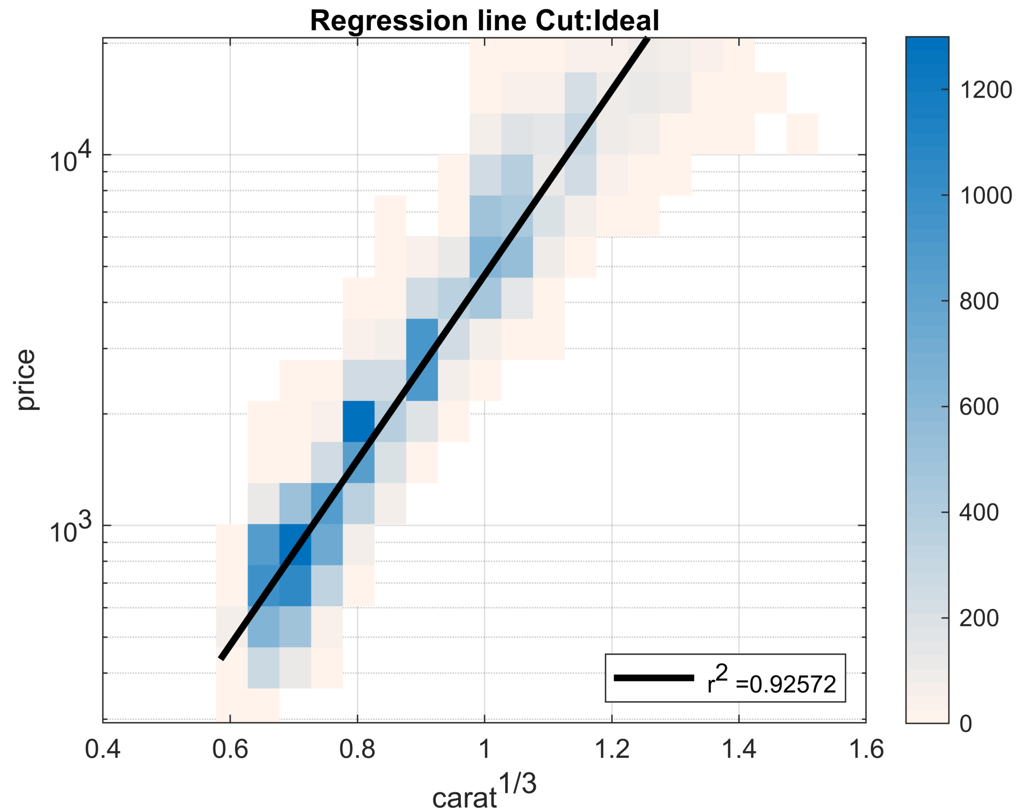
Regression line Color:J
price
carat1/3
r2 =0.96247

```
x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 1001);

figure
hold on
for ii = 1:size(color_categories)
    idx = diamonds_filtered.color==color_categories(ii);

    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)), 'po
    semilogy(x1, exp(f.p1*x1 + f.p2), color=colors(ii,:), LineWidth=2.5)
end
hold off
set(gca,'YScale', 'log')
xlabel('carat^{1/3}')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend(color_categories)
title(strcat('Regression lines'))
% saveas(gcf,strcat('02_Diamonds/carat_regression_color_all.png') )
```

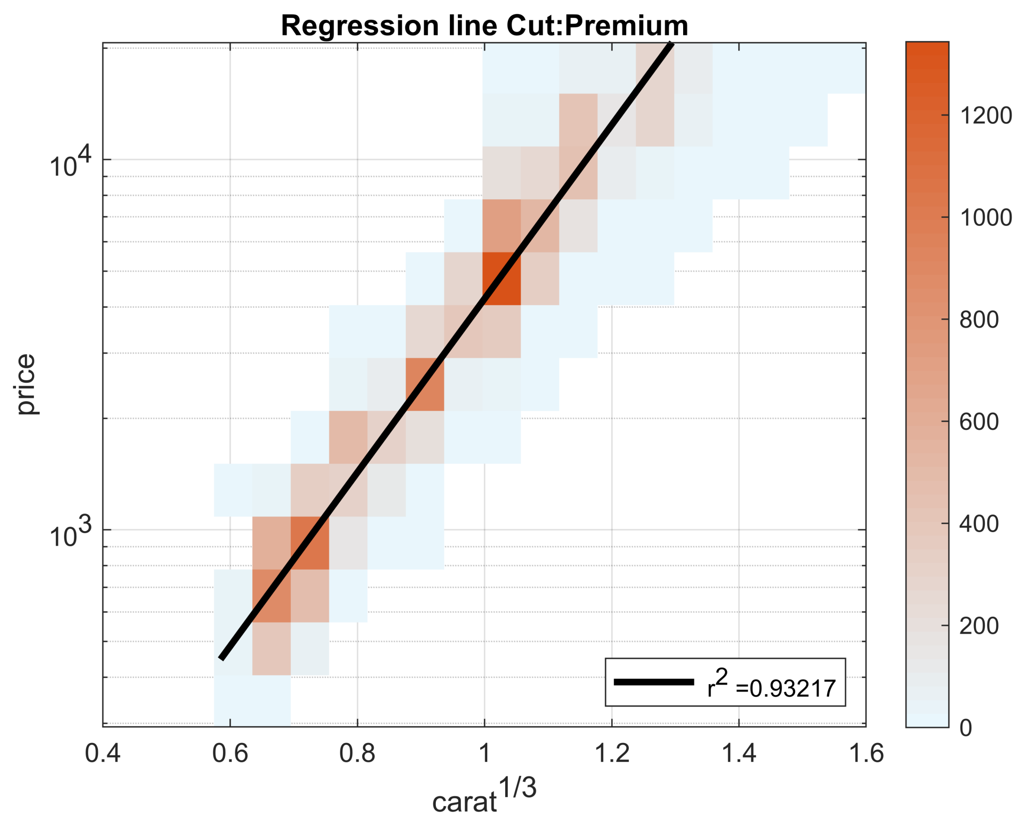


Test for Cut

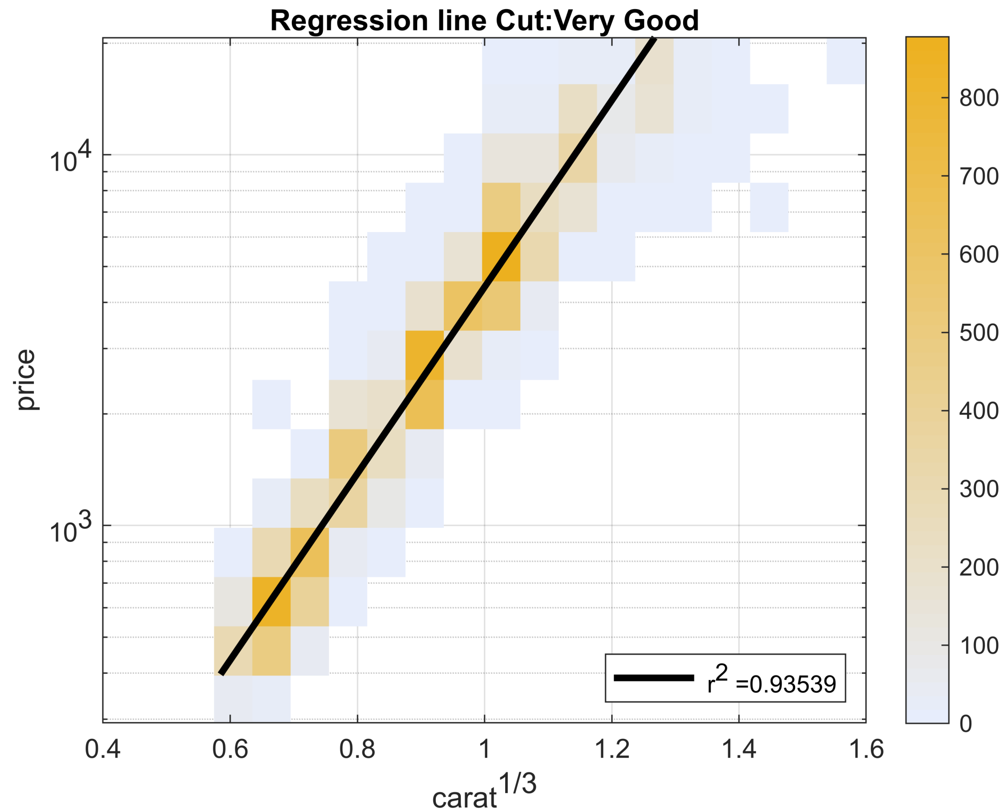
```
cut_categories = unique(diamonds_filtered.cut);
cut_categories = {'Ideal';'Premium';'Very Good';'Good';'Fair'};

color_categories = unique(diamonds_filtered.color);

for ii = 1:size(cut_categories)
    idx = diamonds_filtered.cut==cut_categories(ii);

    figure
    binscatter(diamonds_filtered.carat(idx).^(1/3), diamonds_filtered.price(idx))
    set(gca,'YScale', 'log')
    colormap_graded(colors(ii,:));
    colorbar
    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)), 'pol
    x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 10(
    hold on
    semilogy(x1, exp(f.p1*x1 + f.p2), 'k', LineWidth=2.5)
    hold off
    xlabel('carat^{1/3}')
    ylabel('price')
    grid on
    ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
    legend('',strcat(' r^2 = ',num2str(gof.rsquare)), 'Location','southeast')
    title(strcat('Regression line Cut: ',string(cut_categories(ii))))
%     saveas(gcf,strcat('02_Diamonds/carat_regression_cut_',string(cut_categories(ii)),'.png')
end
```

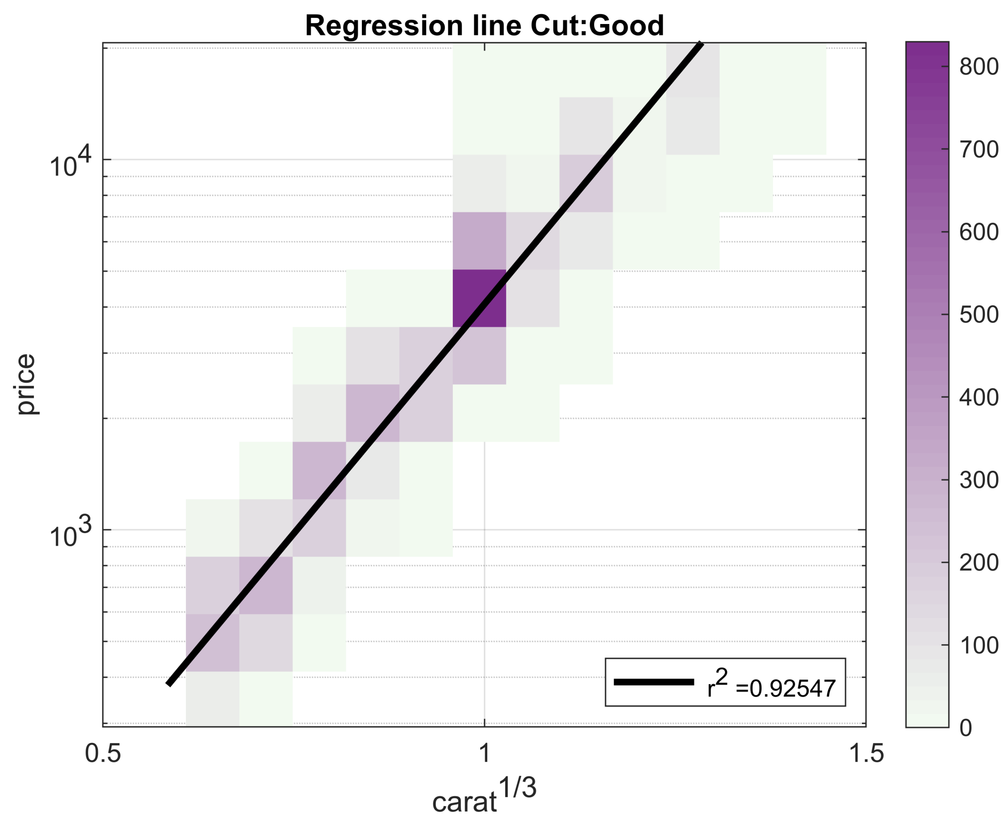


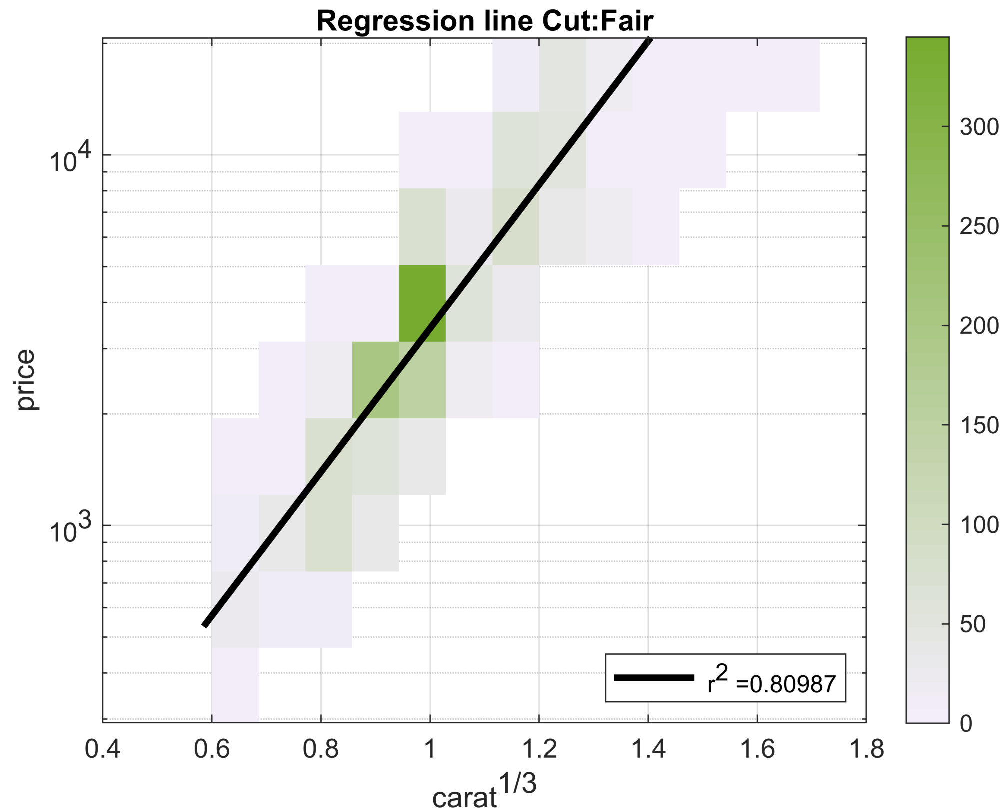
Regression line Cut:Ideal
price
carat^1/3
r^2 =0.92572



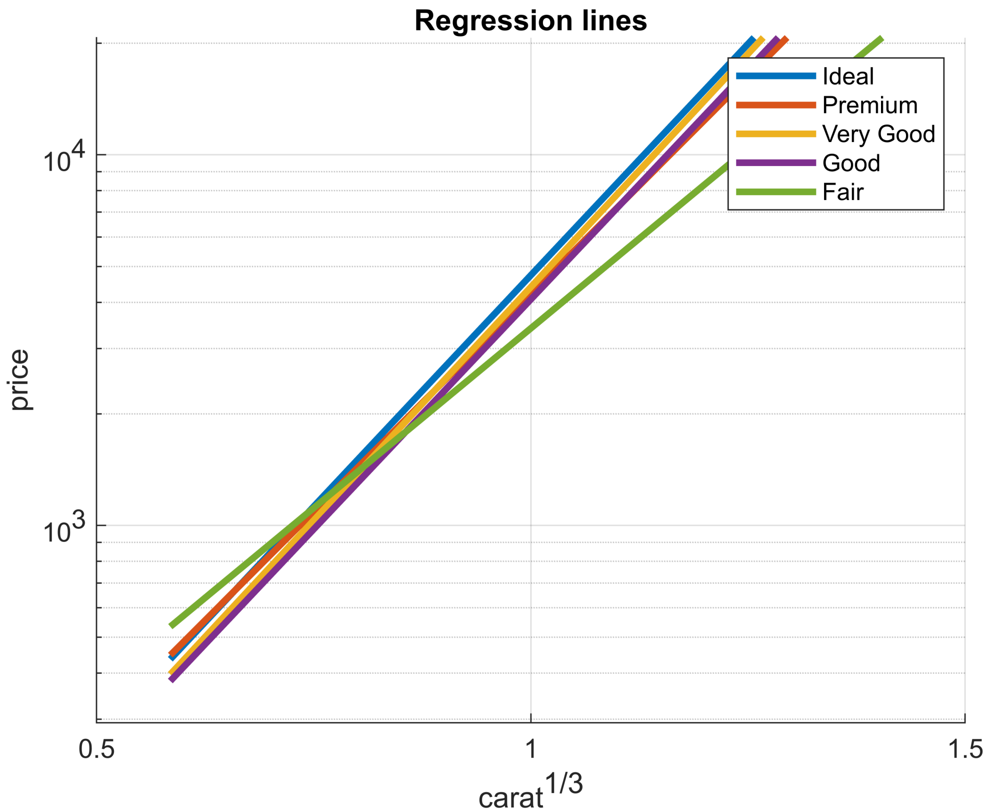
Regression line Cut:Premium
price
carat^1/3
r^2 =0.93217


Regression line Cut:Very Good
price
10^4
10^3
0.4
0.6
0.8
1
1.2
1.4
1.6
$carat^{1/3}$
r^2 =0.93539
800
700
600
500
400
300
200
100
0


Regression line Cut:Good
price
10^4
10^3
0.5
1
1.5
$carat^{1/3}$
r^2 =0.92547
800
700
600
500
400
300
200
100
0




```
x1 = linspace(min(diamonds_filtered.carat.^(1/3)), max(diamonds_filtered.carat.^(1/3)), 1001);

figure
hold on
for ii = 1:size(cut_categories)
    idx = diamonds_filtered.cut==cut_categories(ii);

    [f, gof] = fit(diamonds_filtered.carat(idx).^(1/3), log(diamonds_filtered.price(idx)),  'po
    semilogy(x1, exp(f.p1*x1 + f.p2), color=colors(ii,:), LineWidth=2.5)
end
hold off
set(gca,'YScale', 'log')
xlabel('carat^{1/3}')
ylabel('price')
grid on
ylim([min(diamonds_filtered.price)*0.9, max(diamonds_filtered.price)*1.1])
legend(cut_categories)
title(strcat('Regression lines'))
% saveas(gcf,strcat('02_Diamonds/carat_regression_cut_all.png') )
```


Regression lines
Ideal
Premium
Very Good
Good
Fair
10^4
10^3
price
0.5
1
1.5
carat$^{1/3}$