1.Please install "tidyverse" package and load it.

```
install.packages("tidyverse")

library(tidyverse)
```

Q1:

We will use the "diamonds" dataset that contains the prices and other attributes of almost 54,000 diamonds for mid-exam. This dataset is already included in "tidyverse" package.

1. How many rows are in "diamonds" dataset? How many columns?

   53940rows 10columns

```
library(tidyverse)

diamonds

 dim(diamonds)
```

2. Please find out what attributes contains in the "diamonds" dataset.

"carat" "cut" "color" "clarity" "depth" "table" "price" "x" "y" "z" names(diamonds)

用上述函式 names()得到 data 的 column name.

3. Please find out the high-quality diamonds via filter by color(D), cut(Ideal), and clarity(IF).

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.51	Ideal	D	IF	62.0	56	3446	5.14	5.18	3.20
2	0.51	Ideal	D	IF	62.1	55	3446	5.12	5.13	3.19
3	0.53	Ideal	D	IF	61.5	54	3517	5.27	5.21	3.22
4	0.53	Ideal	D	IF	62.2	55	3812	5.17	5.19	3.22
5	0.63	Ideal	D	IF	61.2	53	3832	5.55	5.60	3.41
6	0.59	Ideal	D	IF	60.7	58	4161	5.45	5.49	3.32
7	0.59	Ideal	D	IF	60.9	57	4208	5.40	5.43	3.30
8	0.56	Ideal	D	IF	62.4	56	4216	5.24	5.28	3.28
9	0.56	Ideal	D	IF	61.9	57	4293	5.28	5.31	3.28
10	0.56	Ideal	D	IF	60.8	58	4632	5.35	5.31	3.24
11	0.59	Ideal	D	IF	60.9	60	4916	5.41	5.39	3.29
12	0.63	Ideal	D	IF	62.5	55	6549	5.47	5.50	3.43
13	0.63	Ideal	D	IF	62.5	55	6607	5.50	5.47	3.43
14	1.04	Ideal	D	IF	61.8	57	14494	6.49	6.52	4.02
15	1.04	Ideal	D	IF	61.8	57	14626	6.52	6.49	4.02
16	1.02	Ideal	D	IF	63.0	57	15575	6.39	6.35	4.01
17	1.06	Ideal	D	IF	61.2	57	15813	6.57	6.61	4.03
18	1.00	Ideal	D	IF	60.7	57	16469	6.44	6.48	3.92
19	1.07	Ideal	D	IF	60.9	54	17042	6.66	6.73	4.08
20	1.03	Ideal	D	IF	62.0	56	17590	6.55	6.44	4.03
21	0.27	Ideal	D	IF	62.4	56	893	4.15	4.12	2.58
22	0.31	Ideal	D	IF	61.1	56	1251	4.39	4.42	2.69
23	0.31	Ideal	D	IF	61.1	56	1310	4.42	4.39	2.69
24	0.31	Ideal	D	IF	60.5	57	1917	4.39	4.41	2.66
25	0.34	Ideal	D	IF	62.1	57	2287	4.46	4.52	2.79
26	0.34	Ideal	D	IF	59.8	57	2287	4.57	4.59	2.74
27	0.34	Ideal	D	IF	62.1	57	2346	4.52	4.46	2.79
28	0.34	Ideal	D	IF	59.8	57	2346	4.59	4.57	2.74

```
A=subset(diamonds,cut=="Ideal")
B=subset(A,color=="D")
C=subset(B,clarity=="IF")
```

先篩出 cut 栏为“Ideal”的所有列作为子集合 A，再从 A 篩出 color 栏为“D”的子集合 B，最后从 B 中篩出同时符合题述条件的 diamonds

4. Following the previous question, which diamond is most expensive? Sort the data to find the answer.

如果是从第三题的结果里挑，最贵的 17590.

	carat	cut	color	clarity	depth	table	price	x	y	z
1	1.03	Ideal	D	IF	62.0	56	17590	6.55	6.44	4.03
2	1.07	Ideal	D	IF	60.9	54	17042	6.66	6.73	4.08
3	1.00	Ideal	D	IF	60.7	57	16469	6.44	6.48	3.92

```
data1 <- C[order(C$price,decreasing=TRUE),]
```

如果是从原始资料里挑，最贵的是 18823

```
data <- diamonds
data1 <- data[order(data$price,decreasing=TRUE),]
print (data1)
```

将 data 按照 price 栏按降序排列，得到 data1，第一列即为所求。

5.Please group the diamond by color, cut, and clarity.

Calculate the mean price and mean carat for each group.

一共分了 276 组

```
Diamonds <- as_tibble(diamonds)
A=Diamonds%>%
  group_by(color,cut,clarity)%>%
  summarise(mean_price=mean(price),
            mean_carat=mean(carat))
```

Filter

	color	cut	clarity	mean_price	mean_carat
1	D	Fair	I1	7383.000	1.8775000
2	D	Fair	SI2	4355.143	1.0169643
3	D	Fair	SI1	4273.345	0.9137931
4	D	Fair	VS2	4512.880	0.8436000
5	D	Fair	VS1	2921.200	0.6300000
6	D	Fair	VVS2	3607.000	0.5911111
7	D	Fair	VVS1	4473.000	0.6066667
8	D	Fair	IF	1619.667	0.3800000
9	D	Good	I1	3490.750	1.0400000
10	D	Good	SI2	3595.296	0.8582511
11	D	Good	SI1	3021.173	0.7008017
12	D	Good	VS2	3588.462	0.7025000
13	D	Good	VS1	3556.581	0.6632558
14	D	Good	VVS2	2345.640	0.4812000
15	D	Good	VVS1	2586.231	0.4907692
16	D	Good	IF	10030.333	0.7866667
17	D	Very Good	I1	2622.800	0.9500000
18	D	Very Good	SI2	4425.459	0.9317197
19	D	Very Good	SI1	3234.931	0.7078340
20	D	Very Good	VS2	3145.194	0.6336570
21	D	Very Good	VS1	2955.480	0.5833714
22	D	Very Good	VVS2	2615.298	0.4657447
23	D	Very Good	VVS1	2987.731	0.4746154
24	D	Very Good	IF	10298.261	0.8030435
25	D	Premium	I1	3818.750	1.1550000
26	D	Premium	SI2	4351.086	0.9189074
27	D	Premium	SI1	3236.378	0.6916547
28	D	Premium	VS2	2919.357	0.5845723
29	D	Premium	VS1	4178.046	0.6870992
30	D	Premium	VVS2	3888.436	0.5805319
31	D	Premium	VVS1	3771.000	0.5382500
32	D	Premium	IF	9056.500	0.7080000
33	D	Ideal	I1	3526.923	0.9600000
34	D	Ideal	SI2	3142.048	0.7503090
35	D	Ideal	SI1	2490.459	0.5947967
36	D	Ideal	VS2	2111.927	0.4992935
37	D	Ideal	VS1	2576.040	0.5335043
38	D	Ideal	VVS2	3619.014	0.5447887
39	D	Ideal	VVS1	2705.778	0.4601389
40	D	Ideal	IF	6567.179	0.6157143
41	E	Fair	I1	2095.222	0.9688889
42	E	Fair	SI2	4172.385	1.0156410
43	E	Fair	SI1	3901.154	0.8670769
44	E	Fair	VS2	3041.714	0.6902381
45	E	Fair	VS1	3307.929	0.6328571
46	E	Fair	VVS2	3119.308	0.6007692
47	E	Fair	VVS1	4115.333	0.6400000
48	E	Good	I1	4398.130	1.3308696
49	E	Good	SI2	3785.490	0.8825743
50	E	Good	SI1	3162.132	0.7238592
51	E	Good	VS2	3772.019	0.7393750
52	E	Good	VS1	3712.775	0.6806742
53	E	Good	VVS2	3390.154	0.5601923
54	E	Good	VVS1	1905.953	0.4181395
55	E	Good	IF	1519.222	0.3733333
56	E	Very Good	I1	3443.545	1.0695455
57	E	Very Good	SI2	4279.447	0.9304045
58	E	Very Good	SI1	3228.176	0.7230831
59	E	Very Good	VS2	3329.497	0.6644135
60	E	Very Good	VS1	3089.358	0.6097952
61	E	Very Good	VVS2	2041.685	0.4267114
62	E	Very Good	VVS1	1997.447	0.4000588
63	E	Very Good	IF	4332.744	0.5793023
64	E	Premium	I1	3199.267	1.0430000
65	E	Premium	SI2	4489.931	0.9576686
66	E	Premium	SI1	3362.625	0.7262866
67	E	Premium	VS2	3070.394	0.6189348
68	E	Premium	VS1	3721.695	0.6431507
69	E	Premium	VVS2	2940.942	0.5115702
70	E	Premium	VVS1	2699.857	0.4622857
71	E	Premium	IF	4525.444	0.5762963
72	E	Ideal	I1	3559.389	1.0377778
73	E	Ideal	SI2	3891.303	0.8744136
74	E	Ideal	SI1	2883.808	0.6706266
75	E	Ideal	VS2	2163.324	0.5211356
76	E	Ideal	VS1	2175.798	0.5035919
77	E	Ideal	VVS2	2556.335	0.4839053
78	E	Ideal	VVS1	2205.519	0.4265075
79	E	Ideal	IF	3258.937	0.4577215
80	F	Fair	I1	2543.514	1.0234286
81	F	Fair	SI2	4520.112	1.0801124
82	F	Fair	SI1	3784.687	0.8640964
83	F	Fair	VS2	3400.472	0.7586792
84	F	Fair	VS1	4103.061	0.8048485
85	F	Fair	VVS2	4018.200	0.6270000
86	F	Fair	VVS1	4679.800	0.6680000
87	F	Fair	IF	2344.000	0.5550000
88	F	Good	I1	2569.526	0.9763158
89	F	Good	SI2	4426.786	1.0025373
90	F	Good	SI1	3261.454	0.7682784
91	F	Good	VS2	3790.543	0.7521196
92	F	Good	VS1	2787.508	0.6246212
93	F	Good	VVS2	3192.360	0.6076000
94	F	Good	VVS1	2189.514	0.4657143

	color	cut	clarity	mean_price	mean_carat
95	F	Good	IF	3132.867	0.5333333
96	F	Very Good	I1	4252.923	1.2107692
97	F	Very Good	SI2	4249.758	0.9511953
98	F	Very Good	SI1	3574.292	0.7964580
99	F	Very Good	VS2	3995.944	0.7420172
100	F	Very Good	VS1	3880.802	0.6880546
101	F	Very Good	VVS2	3461.912	0.5713655
102	F	Very Good	VVS1	2826.540	0.4937356
103	F	Very Good	IF	4677.075	0.6068657
104	F	Premium	I1	3554.559	1.1352941
105	F	Premium	SI2	4747.090	1.0352199
106	F	Premium	SI1	4040.467	0.8421217
107	F	Premium	VS2	4221.467	0.7309532
108	F	Premium	VS1	4758.038	0.7673448
109	F	Premium	VVS2	4099.466	0.6576027
110	F	Premium	VVS1	3969.325	0.6062500
111	F	Premium	IF	3617.581	0.5254839
112	F	Ideal	I1	3903.452	1.1078571
113	F	Ideal	SI2	4335.508	0.9321854
114	F	Ideal	SI1	3710.322	0.7696053
115	F	Ideal	VS2	3317.205	0.6322412
116	F	Ideal	VS1	3504.002	0.6440422
117	F	Ideal	VVS2	3323.629	0.5773077
118	F	Ideal	VVS1	2611.234	0.4761818
119	F	Ideal	IF	2153.709	0.4114925
120	G	Fair	I1	3187.472	1.2264151
121	G	Fair	SI2	5665.150	1.2620000
122	G	Fair	SI1	3579.362	0.9095652
123	G	Fair	VS2	5384.444	0.9777778
124	G	Fair	VS1	3497.622	0.7742222
125	G	Fair	VVS2	3099.059	0.6647059
126	G	Fair	VVS1	2216.333	0.5700000
127	G	Fair	IF	1488.000	0.4550000
128	G	Good	I1	3195.789	1.1742105
129	G	Good	SI2	4776.411	1.0869325
130	G	Good	SI1	4129.329	0.8838164
131	G	Good	VS2	4140.714	0.8156771
132	G	Good	VS1	4302.428	0.7792105
133	G	Good	VVS2	3310.467	0.6261333
134	G	Good	VVS1	2705.195	0.5470732
135	G	Good	IF	4060.136	0.6481818
136	G	Very Good	I1	3194.812	1.1237500
137	G	Very Good	SI2	4699.269	1.0327523
138	G	Very Good	SI1	3481.871	0.7857595
139	G	Very Good	VS2	4426.816	0.8102714
140	G	Very Good	VS1	3770.150	0.7013194
141	G	Very Good	VVS2	3711.785	0.6506954

	color	cut	clarity	mean_price	mean_carat
144	G	Premium	I1	4051.522	1.2913043
145	G	Premium	SI2	5617.205	1.1427033
146	G	Premium	SI1	4303.348	0.8827208
147	G	Premium	VS2	4556.255	0.8094591
148	G	Premium	VS1	4435.823	0.7501767
149	G	Premium	VVS2	4323.571	0.6926545
150	G	Premium	VVS1	2933.655	0.5351462
151	G	Premium	IF	3311.115	0.5640230
152	G	Ideal	I1	4044.438	1.1687500
153	G	Ideal	SI2	4612.086	0.9761111
154	G	Ideal	SI1	3441.108	0.7602576
155	G	Ideal	VS2	4310.035	0.7697033
156	G	Ideal	VS1	4116.918	0.7171459
157	G	Ideal	VVS2	3795.651	0.6460853
158	G	Ideal	VVS1	2909.199	0.5391077
159	G	Ideal	IF	2206.031	0.4547047
160	H	Fair	I1	4212.962	1.4986538
161	H	Fair	SI2	6022.407	1.3643956
162	H	Fair	SI1	5195.800	1.1122667
163	H	Fair	VS2	5110.927	1.0368293
164	H	Fair	VS1	4604.750	0.9759375
165	H	Fair	VVS2	3481.727	0.8409091
166	H	Fair	VVS1	4115.000	0.9100000
167	H	Good	I1	3849.714	1.2521429
168	H	Good	SI2	5529.778	1.1735443
169	H	Good	SI1	4179.285	0.9067234
170	H	Good	VS2	4433.043	0.8779710
171	H	Good	VS1	3819.117	0.7798701
172	H	Good	VVS2	2428.000	0.5884444
173	H	Good	VVS1	1719.710	0.4735484
174	H	Good	IF	5948.750	0.9350000
175	H	Very Good	I1	5258.833	1.6541667
176	H	Very Good	SI2	6112.414	1.2346356
177	H	Very Good	SI1	4933.945	0.9739671
178	H	Very Good	VS2	4620.221	0.8931649
179	H	Very Good	VS1	3750.198	0.7723346
180	H	Very Good	VVS2	2768.145	0.5937241
181	H	Very Good	VVS1	2042.191	0.5043478
182	H	Very Good	IF	2647.690	0.5582759
183	H	Premium	I1	3904.348	1.3397826
184	H	Premium	SI2	6718.946	1.3282917
185	H	Premium	SI1	5707.722	1.0840458
186	H	Premium	VS2	5553.876	1.0046429
187	H	Premium	VS1	3949.336	0.7796131
188	H	Premium	VVS2	2651.263	0.5774576
189	H	Premium	VVS1	1453.759	0.4162500
190	H	Premium	IF	3384.750	0.5980000

	color	cut	clarity	mean_price	mean_carat
191	H	Ideal	I1	5415.184	1.4755263
192	H	Ideal	SI2	5589.473	1.1438222
193	H	Ideal	SI1	4769.988	0.9372084
194	H	Ideal	VS2	4039.126	0.7960432
195	H	Ideal	VS1	3613.325	0.7065310
196	H	Ideal	VVS2	2591.156	0.5673010
197	H	Ideal	VVS1	1915.985	0.4934969
198	H	Ideal	IF	1982.765	0.4746018
199	I	Fair	I1	3501.000	1.3229412
200	I	Fair	SI2	6658.022	1.5115556
201	I	Fair	SI1	4574.967	1.1080000
202	I	Fair	VS2	3856.125	0.9531250
203	I	Fair	VS1	4500.480	1.0104000
204	I	Fair	VVS2	2994.625	0.8450000
205	I	Fair	VVS1	4194.000	0.9000000
206	I	Good	I1	4175.444	1.4100000
207	I	Good	SI2	6933.012	1.4248148
208	I	Good	SI1	4742.945	1.0152727
209	I	Good	VS2	5956.564	1.1296364
210	I	Good	VS1	4597.165	0.9266019
211	I	Good	VVS2	2758.000	0.7196154
212	I	Good	VVS1	2650.955	0.6663636
213	I	Good	IF	1749.333	0.5300000
214	I	Very Good	I1	6045.125	1.7662500
215	I	Very Good	SI2	6621.600	1.3343500
216	I	Very Good	SI1	5195.302	1.0658939
217	I	Very Good	VS2	5754.642	1.0663504
218	I	Very Good	VS1	5276.971	0.9854146
219	I	Very Good	VVS2	3059.887	0.7018310
220	I	Very Good	VVS1	2056.420	0.5708696
221	I	Very Good	IF	4093.895	0.7647368
222	I	Premium	I1	5044.625	1.6058333
223	I	Premium	SI2	7148.484	1.4240064
224	I	Premium	SI1	6092.093	1.1803542
225	I	Premium	VS2	7156.346	1.2359048
226	I	Premium	VS1	5339.367	0.9826244
227	I	Premium	VVS2	3190.768	0.6821951
228	I	Premium	VVS1	1831.083	0.5161905
229	I	Premium	IF	2358.565	0.5730435
230	I	Ideal	I1	4103.294	1.2982353
231	I	Ideal	SI2	7191.912	1.3784672
232	I	Ideal	SI1	5178.565	1.0276389
233	I	Ideal	VS2	4663.384	0.9278995
234	I	Ideal	VS1	3944.422	0.8061029
235	I	Ideal	VVS2	2858.680	0.6538202
236	I	Ideal	VVS1	2034.397	0.5513408
237	I	Ideal	IF	1502.621	0.4514737

Showing 191 to 237 of 276 entries, 5 total columns

237	I	Ideal	IF	1502.621	0.4514737
238	J	Fair	I1	5795.043	1.9934783
239	J	Fair	SI2	5131.815	1.3166667
240	J	Fair	SI1	4553.929	1.1810714
241	J	Fair	VS2	4067.826	1.0326087
242	J	Fair	VS1	5906.188	1.2293750
243	J	Fair	VVS2	2998.000	1.0100000
244	J	Fair	VVS1	1691.000	0.7000000
245	J	Good	I1	3794.500	1.3700000
246	J	Good	SI2	5306.113	1.3188679
247	J	Good	SI1	4627.625	1.1257955
248	J	Good	VS2	4803.167	1.1143333
249	J	Good	VS1	3662.827	0.8750000
250	J	Good	VVS2	4371.154	0.9369231
251	J	Good	VVS1	4633.000	1.0000000
252	J	Good	IF	2738.000	0.6900000
253	J	Very Good	I1	4478.375	1.4625000
254	J	Very Good	SI2	5992.898	1.3609375
255	J	Very Good	SI1	5026.544	1.1353297
256	J	Very Good	VS2	5325.549	1.1405435
257	J	Very Good	VS1	4339.592	0.9649167
258	J	Very Good	VVS2	5960.448	1.1020690
259	J	Very Good	VVS1	3175.526	0.7652632
260	J	Very Good	IF	1074.125	0.4550000
261	J	Premium	I1	4577.231	1.5784615
262	J	Premium	SI2	7550.286	1.5545342
263	J	Premium	SI1	5726.579	1.2577990
264	J	Premium	VS2	6175.559	1.2457426
265	J	Premium	VS1	5817.261	1.1362092
266	J	Premium	VVS2	6423.353	1.2520588
267	J	Premium	VVS1	7244.375	1.2245833
268	J	Premium	IF	7026.000	1.1416667
269	J	Ideal	I1	9454.000	1.9900000
270	J	Ideal	SI2	6555.173	1.3844545
271	J	Ideal	SI1	5115.675	1.1439095
272	J	Ideal	VS2	4867.134	1.0512500
273	J	Ideal	VS1	4734.428	0.9783582
274	J	Ideal	VVS2	4121.926	0.8705556
275	J	Ideal	VVS1	2000.172	0.5782759
276	J	Ideal	IF	2489.000	0.5768000

6. Following the previous question, explore the relationship between the carat and price for each quality of diamonds and plot by scatter plots.

Q2:

Question 2 will use the "midwest" dataset includes demographic information of US midwest counties. This dataset is already included in the "tidyverse" package.

1. How many counties in Illinois?

   102 个

```
midwest
dim(midwest)
names(midwest)
data=midwest
A=subset(data,state=="IL")
dim(A)
```

   从 midwest 提出一个 state 为”IL”的子集合，该集合的列的数量即为 country 的个数

2. How many counties' populations are more than 10000 and less than 20000 in Illinois?

   31 个 打开 environment 查看 1.中建立的子集合 A，点选 poptotal 栏的上三角符号，可直接升序排列，有结果可知从第 14 列到第 44 列符合要求，共 31 个。Code 实现如下：

```
A=subset(data,state=="IL")
dim(A)
A <- A[order(A$poptotal,decreasing=FALSE),]
```

	PID	county	state	area	poptotal	popdensity	popwhite	popblack	popamerindian	popasian	popother	percwhite	percblack
13	593	HAMILTON	IL	0.025	8499	339.9600	8462	3	11	21	2	99.56465	0.03529
14	600	JASPER	IL	0.029	10609	365.8276	10574	1	11	17	6	99.67009	0.00942
15	562	ALEXANDER	IL	0.014	10626	759.0000	7054	3496	19	48	9	66.38434	32.90043
16	578	CUMBERLAND	IL	0.020	10670	533.5000	10627	5	6	26	6	99.59700	0.04686
17	625	MENARD	IL	0.018	11164	620.2222	11101	9	29	14	11	99.43569	0.08061
18	604	JOHNSON	IL	0.020	11347	567.3500	10230	1046	26	14	31	90.15599	9.21829
19	622	MARSHALL	IL	0.023	12846	558.5217	12752	17	30	28	19	99.26825	0.13233
20	653	WABASH	IL	0.012	13111	1092.5833	12955	40	11	80	25	98.81016	0.30508
21	569	CASS	IL	0.024	13437	559.8750	13384	16	8	23	6	99.60557	0.11907
22	630	MOULTRIE	IL	0.021	13930	663.3333	13884	8	22	13	3	99.66978	0.05743
23	587	FORD	IL	0.030	14275	475.8333	14157	43	14	40	21	99.17338	0.30122
24	573	CLAY	IL	0.028	14460	516.4286	14403	4	17	29	7	99.60581	0.02766
25	624	MASSAC	IL	0.014	14752	1053.7143	13804	870	37	31	10	93.57375	5.89750
26	655	WASHINGTON	IL	0.033	14965	453.4848	14856	46	31	26	6	99.27163	0.30738
27	563	BOND	IL	0.022	14991	681.4091	14477	429	35	16	34	96.57128	2.86171
28	591	GREENE	IL	0.033	15317	464.1515	15231	14	50	17	5	99.43853	0.09140
29	634	PIATT	IL	0.025	15548	621.9200	15508	8	16	11	5	99.74273	0.05145
30	572	CLARK	IL	0.030	15921	530.7000	15842	10	26	36	7	99.50380	0.06281
31	611	LAWRENCE	IL	0.022	15972	726.0000	15759	151	31	21	10	98.66642	0.94540
32	623	MASON	IL	0.033	16269	493.0000	16180	8	27	38	16	99.45295	0.04917
33	580	DE WITT	IL	0.023	16516	718.0870	16387	25	37	43	24	99.21894	0.15136
34	657	WHITE	IL	0.029	16522	569.7241	16397	41	37	35	12	99.24343	0.24815
35	640	RICHLAND	IL	0.022	16545	752.0455	16442	17	24	43	19	99.37746	0.10275
36	568	CARROLL	IL	0.027	16805	622.4074	16519	111	30	61	84	98.29813	0.66051
37	656	WAYNE	IL	0.042	17241	410.5000	17141	9	31	44	16	99.41999	0.05220
38	626	MERCER	IL	0.033	17290	523.9394	17155	30	33	35	37	99.21920	0.17351
39	635	PIKE	IL	0.049	17577	358.7143	17499	8	24	32	14	99.55624	0.04551
40	651	UNION	IL	0.024	17619	734.1250	17313	122	34	53	97	98.26324	0.69243
41	654	WARREN	IL	0.033	19181	581.2424	18630	356	20	70	105	97.12737	1.85600
42	577	CRAWFORD	IL	0.026	19464	748.6154	19300	63	34	48	19	99.15742	0.32367
43	581	DOUGLAS	IL	0.025	19464	778.5600	19280	16	19	41	108	99.05466	0.08220
44	583	EDGAR	IL	0.036	19595	544.3056	19469	68	24	24	10	99.35698	0.34702
45	602	JERSEY	IL	0.023	20539	893.0000	20346	96	43	32	22	99.06032	0.46740

3. Which county is the least population in Illinois? How much population is it? Sort the data to find the answer.

Pope country 4373 人

由 2.得到排序后的资料，第一列的 country 栏对应的为“POPE”，poptotal ==4373.

PID	county	state	area	poptotal	popdensity	popwhite	popblack	popamerindian	popasian	popother	percwhite	percblack
636	POPE	IL	0.022	4373	198.7727	4072	266	15	6	14	93.11685	6.08278

4. Please find out the relationship between area, total population, and population density. Try to figure out how to calculate population density on this dataset.

$$\text{Density}=\frac{\text{poptotal}}{1000*\text{area}}$$

```
A <- A[order(A$poptotal,decreasing=FALSE),]
B=data[,5]/data[,4]
B
C=data[,6]
C=C/B
C
```

R-midexam.R* | A | data | data1 | B | C

Filter

	popdensity
1	0.001
2	0.001
3	0.001
4	0.001
5	0.001
6	0.001
7	0.001
8	0.001
9	0.001
10	0.001
11	0.001
12	0.001
13	0.001
14	0.001
15	0.001
16	0.001
17	0.001
18	0.001
19	0.001
20	0.001
21	0.001
22	0.001
23	0.001
24	0.001
25	0.001
26	0.001
27	0.001
28	0.001
29	0.001
30	0.001
31	0.001
32	0.001
33	0.001
34	0.001
35	0.001

用 poptotal 栏的值对应除 area 栏的值，再将商除 density=0.001，因此得到上述彼此的关系

5. Following the previous question, What is the population density of Illinois?

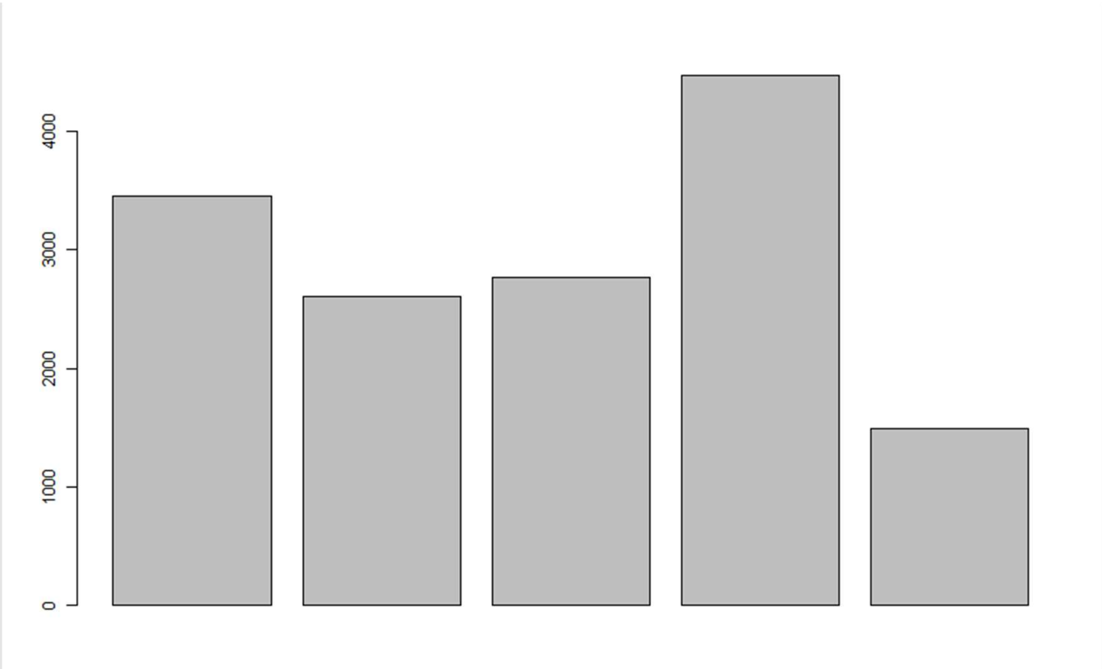
Density=3459.625

```
A=subset(data,state=="IL")
sumarea=sum(data$area[1:102])
sumpoptotal=sum(data$poptotal[1:102])
sumpoptotal
ILdensity=sumpoptotal/(1000*sumarea)
ILdensity
```

先得到 IL 的子集合 A，再将 A 的 area 和 poptotal 分别求和得到 sumarea and sumpoptotal ;最后根据 4.的公式得到 ILdensity

6. What is the population density of each state? Plot the bar plot to show it.

```
Midwest <- as_tibble(midwest)
C=Midwest %>%
  group_by(state)%>%
  summarise(sumpoptotal=sum(poptotal),
            sumarea=sum(area),
            sumdensity=sumpoptotal/(1000*sumarea))
C
barplot(C$sumdensity)
```



	state	sumpoptotal	sumarea	sumdensity
1	IL	11430602	3.304	3459.625
2	IN	5544159	2.127	2606.563
3	MI	9295297	3.357	2768.930
4	OH	10847115	2.421	4480.428
5	WI	4891769	3.286	1488.670

将资料按照 state 分组，然后求出每组的 poptotal，area 之和，再利用 4.的公式求出各组的 sumdensity,最后利用 barplot()函式将 sumdensity 可视化

Q3:

The "sleep" dataset shows the effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients. Please using data(sleep) to load the dataset.

1. How many rows are in "sleep" dataset? How many columns?

```
data(sleep)
sleep
dim(sleep)
```

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10
11	1.9	2	1
12	0.8	2	2
13	1.1	2	3
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6
17	5.5	2	7
18	1.6	2	8
19	4.6	2	9
20	3.4	2	10

20rows 3columns

Group1 有 10 列，Group2 有 10 列，column：extra/group/ID

2. Please find out what attributes contains in the "sleep" dataset and explain it.

Extra：与对照组相比，该药物对病人的影响为多少（睡眠时间的增加），正值为正相关，值越大催眠效果越好，负值为负相关，值越小催眠效果越差

Group:为区分两种药物的组别　ID：10 位病人的 ID

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10
11	1.9	2	1
12	0.8	2	2
13	1.1	2	3
14	0.1	2	4
15	-0.1	2	5
16	4.4	2	6
17	5.5	2	7
18	1.6	2	8
19	4.6	2	9
20	3.4	2	10

3. Please group the two soporific drugs and calculate the mean of increase in hours of sleep and median of increase in hours of sleep for each soporific drug.

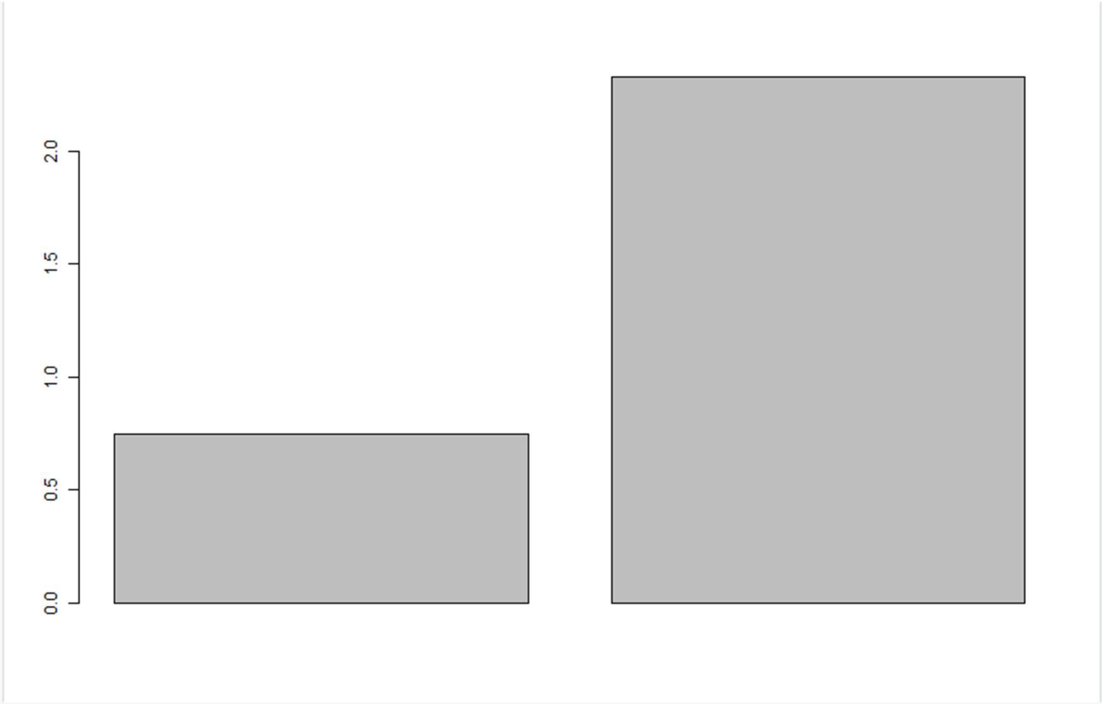
```
# A tibble: 2 x 4
  group sum_extra median_extra mean_extra
  <fct>     <dbl>        <dbl>      <dbl>
1 1           7.5         0.35       0.75
2 2          23.3         1.75       2.33
```

group1:mean:0.75 median:0.35

group2:mean:2.33 median:1.75

```
sleep
dim(sleep)
Sleep<-as_tibble(sleep)
Sleep %>%
  group_by(group)%>%
  summarise(sum_extra=sum(extra),
            median_extra=median(extra),
            mean_extra=mean(extra))
```

4. Using any data visualization skill to present Which soporific drug can increase more sleep time? Please submit the figure and your code.




```
C=Sleep %>%
  group_by(group)%>%
  summarise(sum_extra=sum(extra),
            median_extra=median(extra),
            mean_extra=mean(extra))
res1 <- C$mean_extra
barplot(res1)
```

将分组后的 mean_extra 做柱形图，可知 group2 的平均睡眠增加时长远大于 group1，所以 group2 的药效要远好于 group1.

Q4:

Question 4 will use the "quakes" dataset includes the locations of 1000 seismic events of MB > 4.0. The events occurred in a cube near Fiji since 1964. Please using data(quakes) to load the dataset.

1. Please inspect data to find out what attributes contains in the "quakes" dataset and explain it.

   lat: numeric Latitude of event

   long： numeric longitude

   depth： numeric depth(KM)

   mag: Richer Magnitude

station： numeric Number of stations reporting

```
> names(quakes)
[1] "lat"      "long"     "depth"    "mag"      "stations"
```

```
library('datasets')
quakes
names(quakes)
```

2. How many observations magnitude > 5.0 and depth > 500 in this dataset?

   38 个

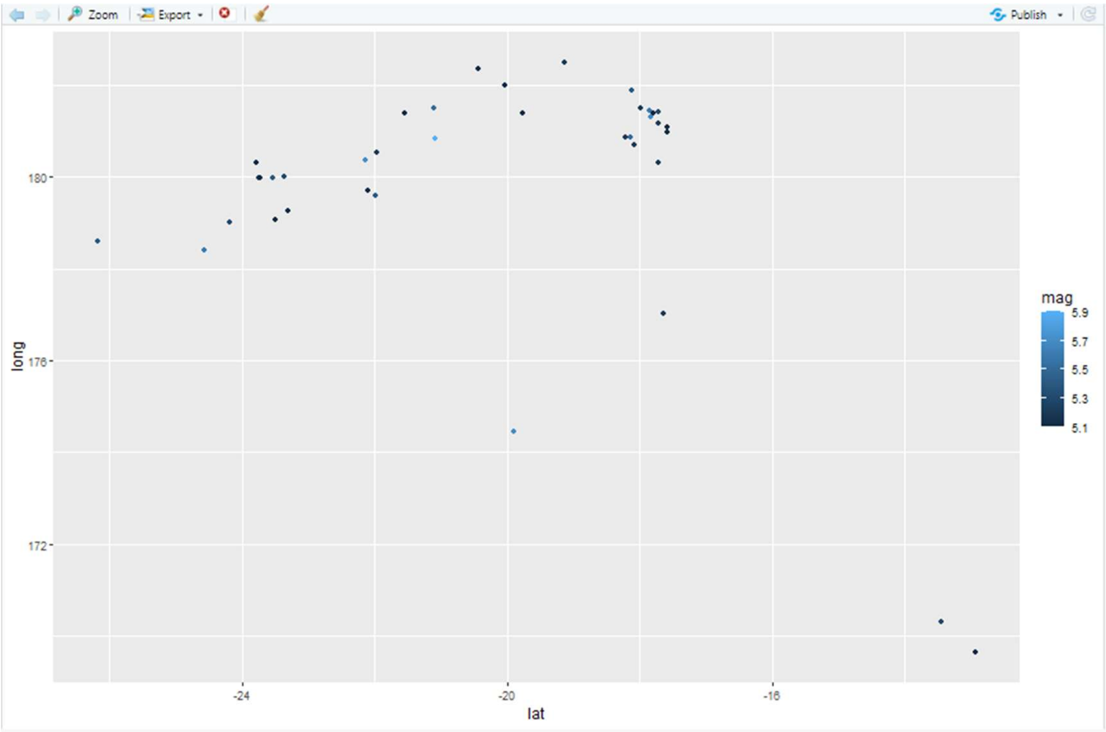
	lat	long	depth	mag	stations
1	-23.74	179.99	506	5.2	75
2	-17.72	180.30	595	5.2	74
3	-21.96	180.54	603	5.2	66
4	-23.36	180.01	553	5.3	61
5	-17.80	181.38	587	5.1	47
6	-22.13	180.38	577	5.7	104
7	-19.13	182.51	579	5.2	56
8	-24.57	178.40	562	5.6	80
9	-12.93	169.63	641	5.1	57
10	-23.49	179.07	544	5.1	58
11	-21.98	179.60	583	5.4	67
12	-20.43	182.37	502	5.1	48
13	-23.73	179.99	527	5.1	49
14	-17.59	181.09	536	5.1	61
15	-19.77	181.40	630	5.1	54
16	-20.04	182.01	605	5.1	49
17	-17.72	181.42	565	5.3	89
18	-17.84	181.30	535	5.7	112
19	-13.45	170.30	641	5.3	93
20	-26.18	178.59	548	5.4	65
21	-22.10	179.71	579	5.1	58
22	-21.11	181.50	538	5.5	104
23	-23.53	179.99	538	5.4	87
24	-18.08	180.70	628	5.2	72
25	-17.71	181.18	574	5.2	67
26	-23.31	179.27	566	5.1	49
27	-19.89	174.46	546	5.7	99
28	-24.18	179.02	550	5.3	86
29	-23.78	180.31	518	5.1	71
30	-18.12	181.88	649	5.4	88
31	-17.59	180.98	548	5.1	79
32	-18.14	180.87	624	5.5	105
33	-18.21	180.87	631	5.2	69
34	-17.64	177.01	545	5.2	91
35	-17.98	181.51	586	5.2	68
36	-21.08	180.85	627	5.9	119
37	-21.55	181.39	513	5.1	81
38	-17.85	181.44	589	5.6	115

```
library('datasets')
quakes
names(quakes)
A=quakes
library(data.table)
B=data.table(A)
B[depth>500]
C=B[depth>500]
C=C[mag>5]
```

将 quakes data 转为 table，然后逐步筛出 depth>500 的资料 C，最后再从 C 里面筛出 mag>5.0 的资料集。

3. Following the previous question(magnitude > 5.0 and depth > 500), please plot the scatter plot by longitude(long) and latitude(lat) and color by the magnitude

```
A=quakes
library(data.table)
B=data.table(A)
B[depth>500]
C=B[depth>500]
C=C[mag>5]
library(ggplot2)
ggplot(C,aes(x=lat,y=long,color=mag))
```



4. Please plot the box plot by magnitude to find out how many outliers in the figure? Tips: x = factor(0)

5. Please plot the histogram by magnitude frequency to find out the distribution. Tips: Please note the width of the bin when you plot.

6. Please plot the histogram by depth frequency to find out the distribution. Tips: Please note the width of the bin when you plot.