I. R 이해하기

> - ## 4. 데이터 읽기
> - ## 5. 통계적 시각화

In [1]:

```
# working directory 확인 함수 : getwd()
getwd()
```

In [2]:

```
# setup working directory
setwd('R_Edu')
```

In [3]:

```
# read working directory files
dir()
```

4. 데이터 읽기

In [22]:

```
# 데이터 R파일로 저장하기
my_iris <- iris
save(my_iris, file = "my_iris.rdata")

# rm()함수로 변수 제거
rm(my_iris)
```

In [23]:

```
# R데이터파일 로드
load("my_iris.rdata")

head(my_iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

In [24]:

```
# 개별 변수들 저장
x <- c(1: 20)
y <- c( "a", "b")
z <- c("팀1", "팀2", "팀3")

save(x, y, z, file = "my_mult.rdata")
rm(x, y, z)
# x
# y
# z
```

In [25]:

```
# 개별 변수들 불러오기
load("my_mult.rdata")
x
y
z
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20

'a' 'b'

'팀1' '팀2' '팀3'

In [26]:

```
# 데이터 타입 확인하는 함수 class()
class(x)
```

'integer'

In [27]:

```
class(z)
```

'character'

In [28]:

```
# mtcars의 데이터를 RDS(R data file formats)로 저장(save)
saveRDS(mtcars, "my_mtcars.rds")
```

In [29]:

```
my_data <- readRDS("my_mtcars.rds")

# head() 함수를 활용하여 매개변수(argument) 기
# 본 디폴트값으로 6번째 행까지 데이터 확인
head(my_data)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3

In [30]:

```
# iris데이터 를 R데이터 파일형식으로 저장
save(iris, file =  "my_iris.RData")
```

In [31]:

```
# 저장된 R데이터 불러오기
load("my_iris.RData")
```

In [32]:

```
#  작업공간의 모든 데이터를 저장
save.image(file = "my_work_space.RData")
```

In [15]:

```
# 저장되어 있는 모든 데이터 불러오기
load("my_work_space.RData")
```

In [16]:

```
# 현재 R 버전 및 기본 사항, 인코딩 등의 내용 확인하는 함수
sessionInfo()
```

In [33]:

```
tomato <- read.csv("DATA/TomatoFirst.csv", header = TRUE)

head(tomato, 3)
```

Round	Tomato	Price	Source	Sweet	Acid	Color	Texture	Over
1	Simpson SM	3.99	Whole Foods	2.8	2.8	3.7	3.4	3
1	Tuttorosso (blue)	2.99	Pioneer	3.3	2.8	3.4	3.0	2
1	Tuttorosso (green)	0.99	Pioneer	2.8	2.6	3.3	2.8	2

In [34]:

```
class(tomato)
```

'data.frame'

In [35]:

```
library(readr)

tomato2 <- read_delim(file = "DATA/TomatoFirst.csv", delim = "
," )
```

```
Parsed with column specification:
cols(
  Round = col_double(),
  Tomato = col_character(),
  Price = col_double(),
  Source = col_character(),
  Sweet = col_double(),
  Acid = col_double(),
  Color = col_double(),
  Texture = col_double(),
  Overall = col_double(),
  Avg.of.Totals = col_double(),
  Total.of.Avg = col_double()
)
```

In [36]:

```
class(tomato2)
```

'spec_tbl_df' 'tbl_df' 'tbl' 'data.frame'

In [37]:

```
library(data.table)

tomato3 <- fread(input = "DATA/TomatoFirst.csv",
                 sep = "," , header = TRUE)

head(tomato3)
```

Round	Tomato	Price	Source	Sweet	Acid	Color	Texture	Ove
1	Simpson SM	3.99	Whole Foods	2.8	2.8	3.7	3.4	
1	Tuttorosso (blue)	2.99	Pioneer	3.3	2.8	3.4	3.0	
1	Tuttorosso (green)	0.99	Pioneer	2.8	2.6	3.3	2.8	
1	La Fede SM DOP	3.99	Shop Rite	2.6	2.8	3.0	2.3	
2	Cento SM DOP	5.49	D Agostino	3.3	3.1	2.9	2.8	
2	Cento Organic	4.99	D Agostino	3.2	2.9	2.9	3.1	

In [38]:

```
class(tomato3)
```

'data.table' 'data.frame'

other statistical Tools

function	format
read.spss	SPSS
read.dta	Stata
read.ssd	SAS
read.octave	Octave
read.mtp	Minitab
read.systat	Systat

5. 통계적 시각화 (Statistical Graphics)

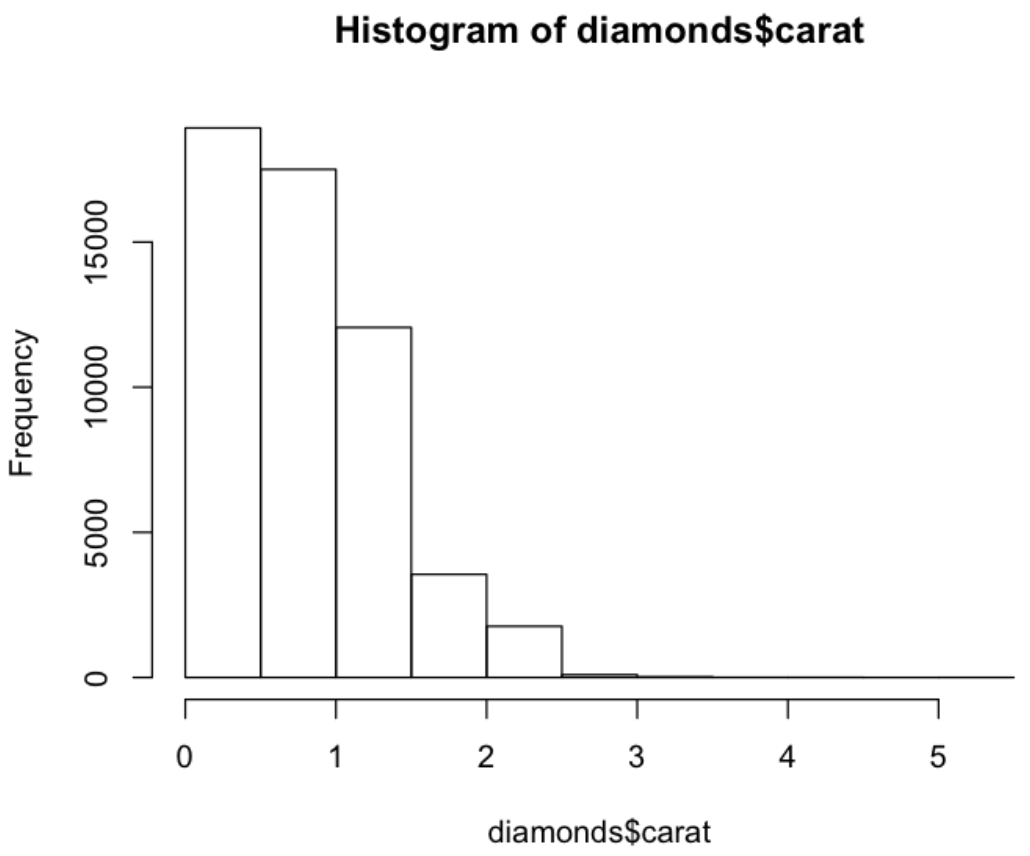
In [53]:

```
library(ggplot2)

data(diamonds)
head(diamonds)
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

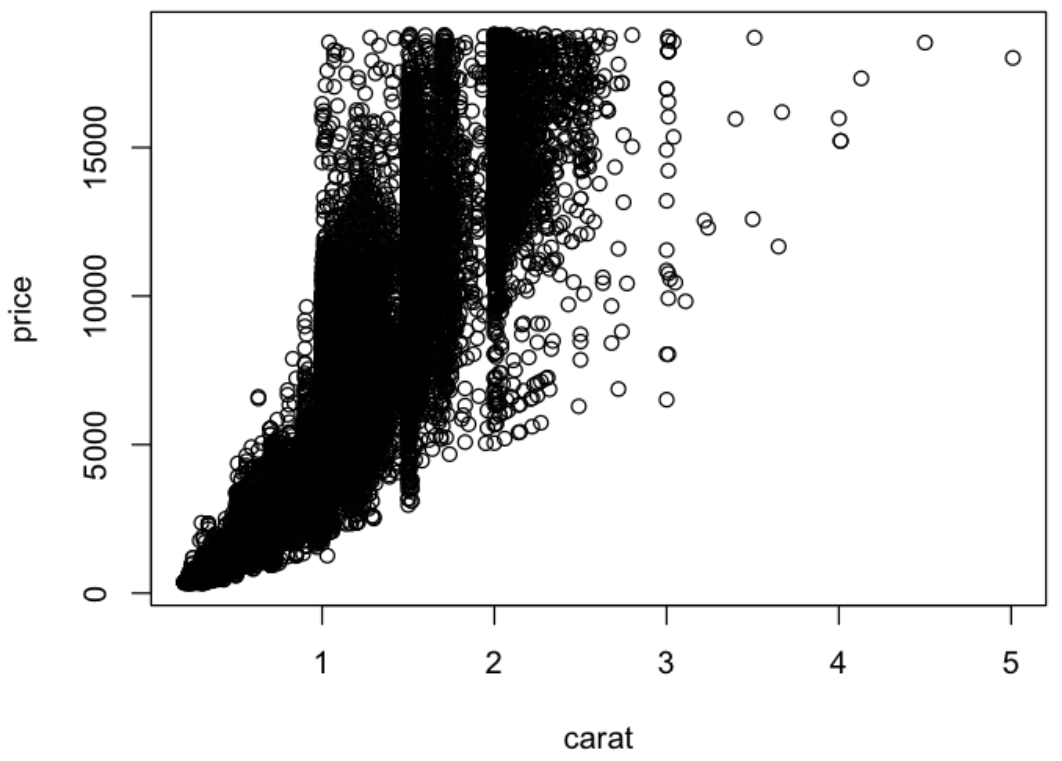
In [54]:

```
# 시각화 그림 사이즈 조정
options(repr.plot.width = 6, repr.plot.height = 5)

# 히스토그램(Histogram)
diamonds[, "carat"]
hist(diamonds$carat)
```



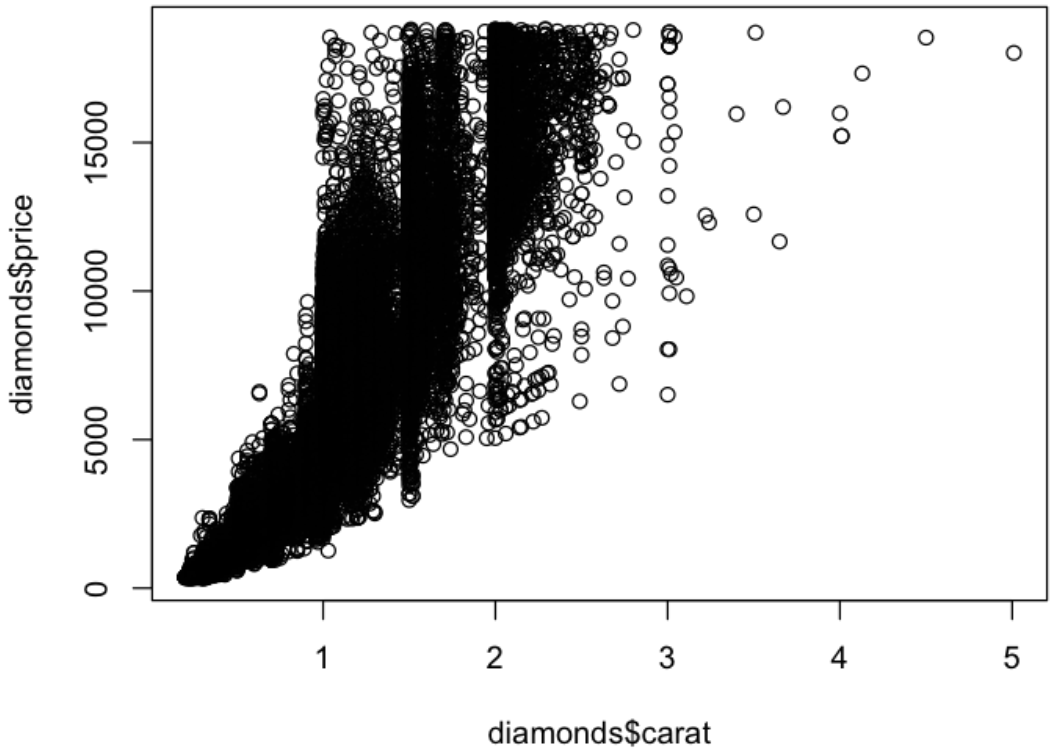
In [55]:

```
# 산포도(Scatter plot)

plot(price ~ carat, data = diamonds)
```



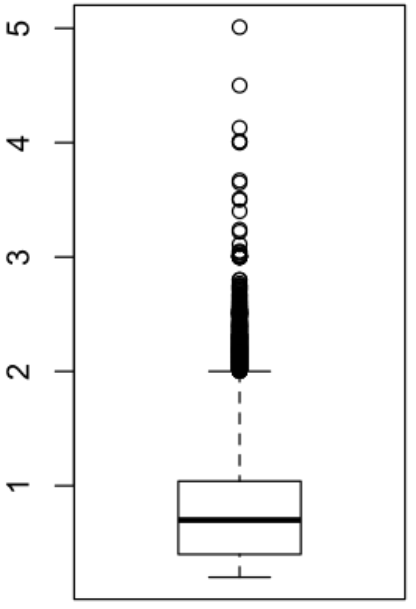
In [56]:

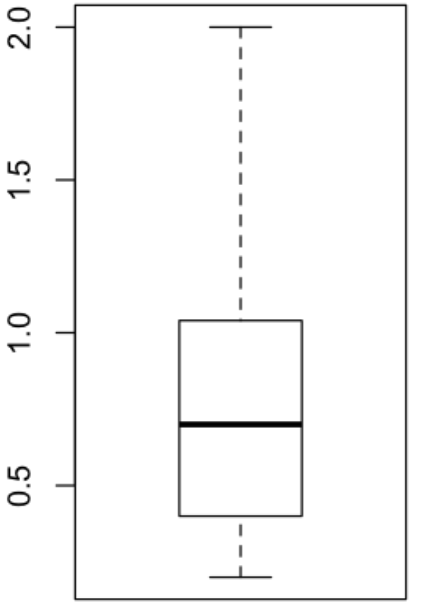
```
# 동일 산포도(Scatter plot)
# 매개변수에 데이터 타입에 따라 다른 입력
plot(diamonds$carat, diamonds$price)
```



In [57]:

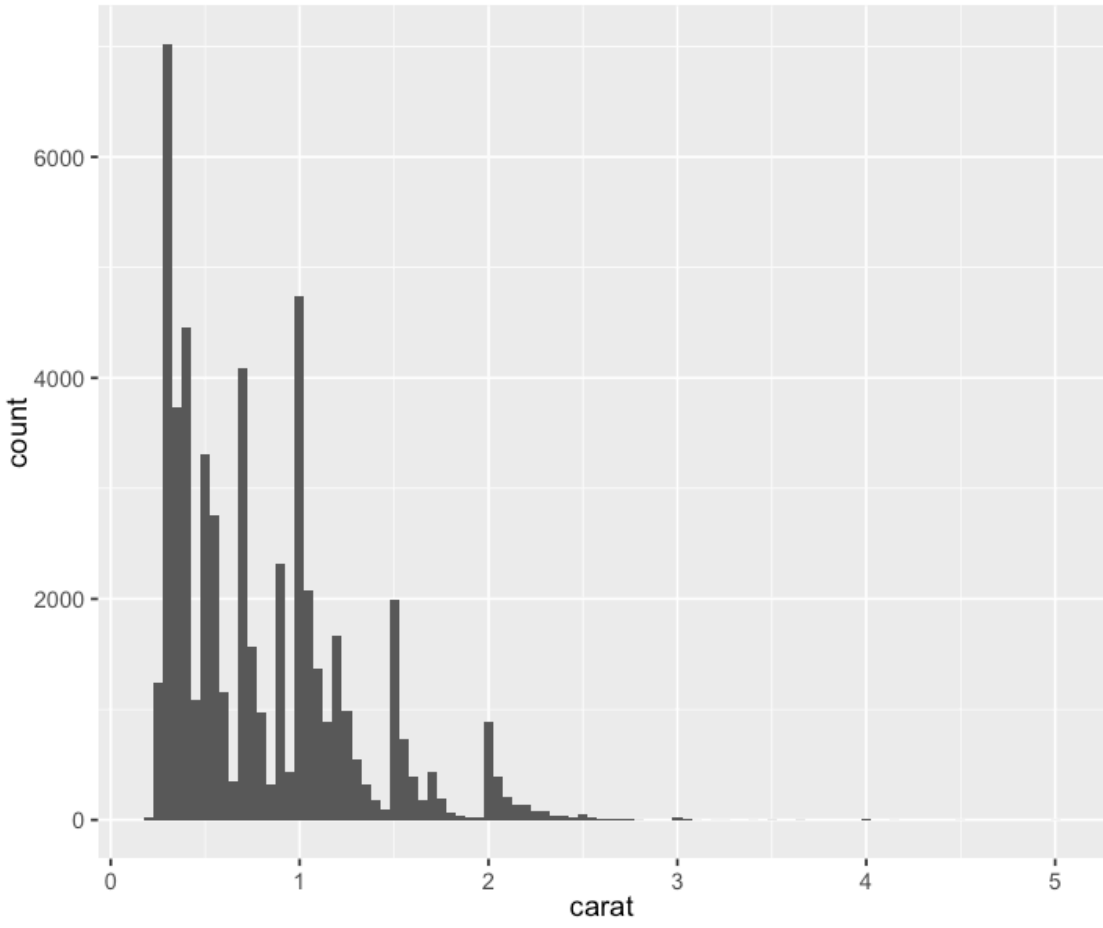
```
# 한화면에 두개 시각화 이미지를 주기 위한 파라미터 옵션
par(mfrow = c(1,2))

# 상자그림(Boxplots)
# 데이터 최소값, 1사분위수, 2사분위수,
# 3사 분위수, 최대값을 표현
boxplot(diamonds$carat)

# 이상치 값 제외하고 시가화
boxplot(diamonds$carat, outline = FALSE)
```





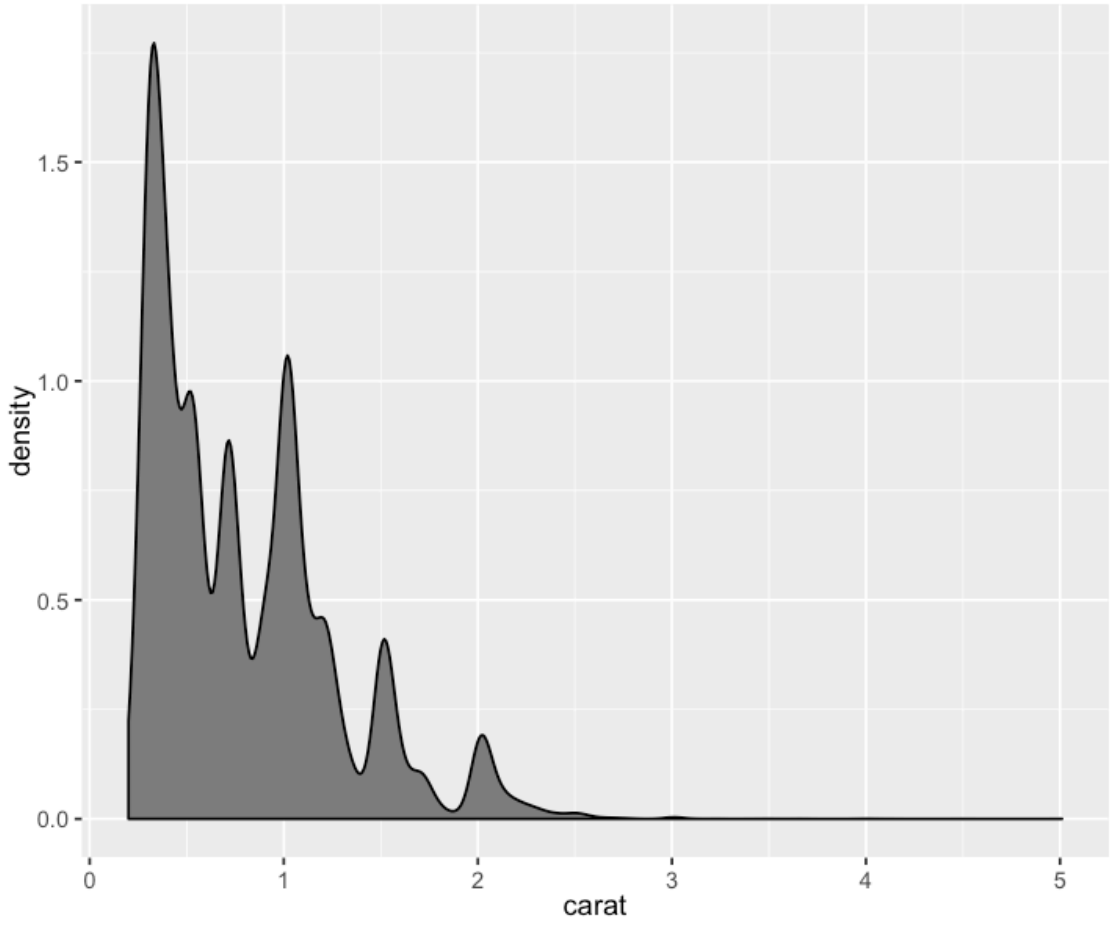
In [58]:

```
# 시각화 패키지 활용
# 히스토그램
library(ggplot2)

ggplot(data = diamonds) +
geom_histogram(aes(x = carat), binwidth = 0.05)
```



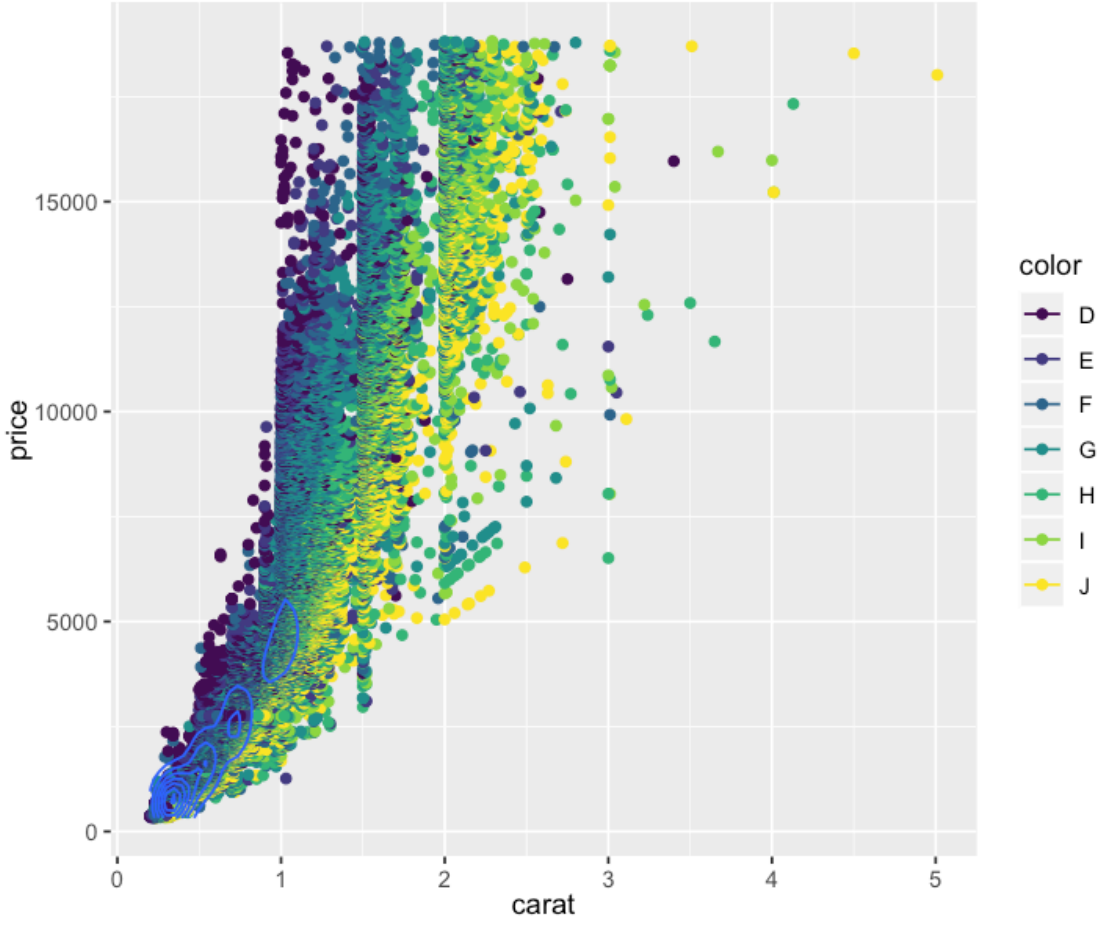
In [59]:

```
# 확률밀도로 표시
ggplot(data = diamonds) +
geom_density(aes(x = carat), fill = "grey50")
```



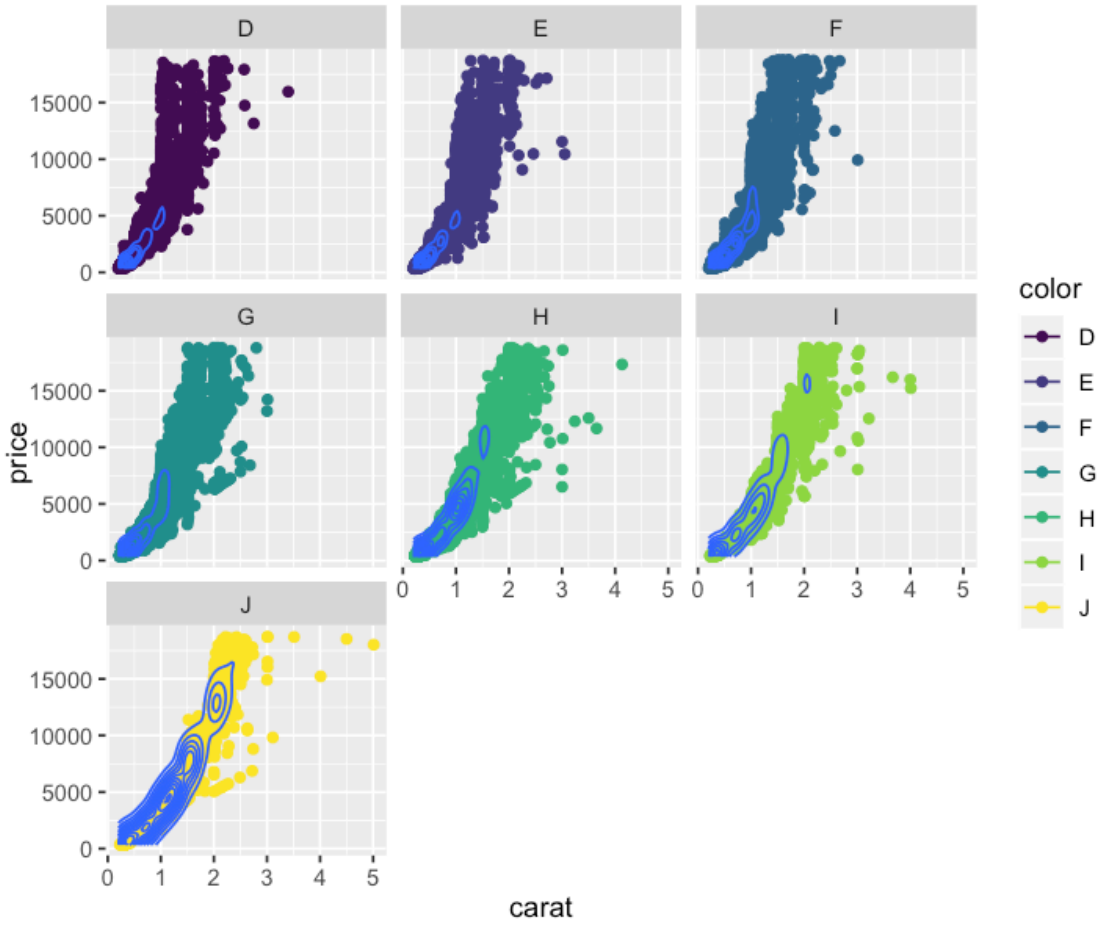
In [61]:

```
# 산포도에  추가 정보포함 시각화
ggplot(data = diamonds, aes( x = carat, y = price)) +
geom_point(aes(color = color)) +
geom_density_2d()
```



In [62]:

```
# 개별 특성에 따른 산포도와 확률밀도 시각화
ggplot(data = diamonds, aes( x = carat, y = price)) +
geom_point(aes(color = color)) + facet_wrap( ~color) +
geom_density2d()
```



In [63]:

```
# 5가지 데이터 특성활용 시각화
ggplot(data = diamonds, aes( x = carat, y = price)) +
geom_point(aes(color = color)) + facet_wrap(cut ~ clarity)
```