

Data manipulation with Data.table Part 1 Solutions

Below are the solutions to [these](#) exercises on data.table.

```
library(data.table)
library(ggplot2)
#####
#                               #
# Exercise 1                     #
#                               #
#####
iris_dt <- as.data.table(iris)
iris_dt[,mean(Petal.Length),substr(Species,1,1)]
```

```
##      substr      V1
## 1:      s 1.462
## 2:      v 4.906
```

```
#####
#                               #
# Exercise 2                     #
#                               #
#####
dt <- as.data.table(diamonds)

dt[,("mean_price"= mean(price)),.(cut,color)]
```

```
##      cut color      V1
## 1:  Ideal   E 2597.550
## 2: Premium   E 3538.914
## 3:   Good   E 3423.644
## 4: Premium   I 5946.181
## 5:   Good   J 4574.173
## 6: Very Good J 5103.513
## 7: Very Good I 5255.880
## 8: Very Good H 4535.390
## 9:   Fair   E 3682.312
```

```

## 10:      Ideal      J 4918.186
## 11:     Premium     F 4324.890
## 12:      Ideal     I 4451.970
## 13:      Good      I 5078.533
## 14: Very Good     E 3214.652
## 15: Very Good     G 3872.754
## 16: Very Good     D 3470.467
## 17: Very Good     F 3778.820
## 18:      Good      F 3495.750
## 19:      Good      H 4276.255
## 20:      Good      D 3405.382
## 21:      Ideal     G 3720.706
## 22:     Premium     D 3631.293
## 23:     Premium     J 6294.592
## 24:      Ideal     D 2629.095
## 25:     Premium     G 4500.742
## 26:     Premium     H 5216.707
## 27:      Fair      F 3827.003
## 28:      Ideal     F 3374.939
## 29:      Fair      H 5135.683
## 30:      Ideal     H 3889.335
## 31:      Good      G 4123.482
## 32:      Fair      G 4239.255
## 33:      Fair      J 4975.655
## 34:      Fair      I 4685.446
## 35:      Fair      D 4291.061
##           cut color      V1

```

```
#####
```

```

#           #
# Exercise 3 #
#           #

```

```
#####
```

```
dt[, .N, .(price/carat)][order(-N)][1:5]
```

```

##      price  N
## 1: 2250.000 331
## 2: 1800.000 215
## 3: 4200.000 160
## 4: 2016.667 157
## 5: 2100.000 129

```

```
#####
#           #
# Exercise 4 #
#           #
#####
dt[,.(tail(carat,2)),cut]
```

```
##           cut    V1
##  1:      Ideal 0.72
##  2:      Ideal 0.75
##  3:   Premium 0.72
##  4:   Premium 0.86
##  5:       Good 0.79
##  6:       Good 0.72
##  7: Very Good 0.70
##  8: Very Good 0.70
##  9:       Fair 1.04
## 10:       Fair 0.71
```

```
#####
#           #
# Exercise 5 #
#           #
#####
```

```
dt[,lapply(.SD,median),cut,.SDcols=c("x","y","z")]
```

```
##           cut    x    y    z
##  1:      Ideal 5.250 5.26 3.23
##  2:   Premium 6.110 6.06 3.72
##  3:       Good 5.980 5.99 3.70
##  4: Very Good 5.740 5.77 3.56
##  5:       Fair 6.175 6.10 3.97
```

```
#####
#           #
# Exercise 6 #
#           #
#####
```

```
dt2 <- as.data.table(airquality)
```

```
dt2[,lapply(.SD,log10),by=.(by1=Month,by2=Day>15),.SDcols=c("Wind")][by2==TRUE]
```

##		by1	by2	Wind
##	1:	5	TRUE	1.0606978
##	2:	5	TRUE	1.0791812
##	3:	5	TRUE	1.2648178
##	4:	5	TRUE	1.0606978
##	5:	5	TRUE	0.9867717
##	6:	5	TRUE	0.9867717
##	7:	5	TRUE	1.2201081
##	8:	5	TRUE	0.9867717
##	9:	5	TRUE	1.0791812
##	10:	5	TRUE	1.2201081
##	11:	5	TRUE	1.1731863
##	12:	5	TRUE	0.9030900
##	13:	5	TRUE	1.0791812
##	14:	5	TRUE	1.1731863
##	15:	5	TRUE	0.7558749
##	16:	5	TRUE	0.8692317
##	17:	6	TRUE	1.1731863
##	18:	6	TRUE	1.3159703
##	19:	6	TRUE	0.9637878
##	20:	6	TRUE	1.0606978
##	21:	6	TRUE	1.0128372
##	22:	6	TRUE	0.7993405
##	23:	6	TRUE	0.2304489
##	24:	6	TRUE	0.6627578
##	25:	6	TRUE	0.7993405
##	26:	6	TRUE	0.9030900
##	27:	6	TRUE	0.9030900
##	28:	6	TRUE	1.0128372
##	29:	6	TRUE	1.0606978
##	30:	6	TRUE	1.1731863
##	31:	6	TRUE	0.9030900
##	32:	7	TRUE	0.8388491
##	33:	7	TRUE	1.0128372
##	34:	7	TRUE	0.7993405
##	35:	7	TRUE	0.7075702
##	36:	7	TRUE	1.0606978
##	37:	7	TRUE	0.8388491

38: 7 TRUE 0.9867717
39: 7 TRUE 1.0606978
40: 7 TRUE 0.9344985
41: 7 TRUE 0.9030900
42: 7 TRUE 0.9344985
43: 7 TRUE 1.0791812
44: 7 TRUE 0.8692317
45: 7 TRUE 0.8692317
46: 7 TRUE 0.8692317
47: 7 TRUE 0.9637878
48: 8 TRUE 1.0128372
49: 8 TRUE 0.7993405
50: 8 TRUE 0.8692317
51: 8 TRUE 1.0374265
52: 8 TRUE 1.0128372
53: 8 TRUE 1.1903317
54: 8 TRUE 1.1553360
55: 8 TRUE 1.1003705
56: 8 TRUE 0.9867717
57: 8 TRUE 0.5314789
58: 8 TRUE 0.9030900
59: 8 TRUE 0.7558749
60: 8 TRUE 0.9867717
61: 8 TRUE 0.3617278
62: 8 TRUE 0.7993405
63: 8 TRUE 0.7993405
64: 9 TRUE 0.8388491
65: 9 TRUE 1.1398791
66: 9 TRUE 1.0128372
67: 9 TRUE 1.0128372
68: 9 TRUE 0.9030900
69: 9 TRUE 1.1003705
70: 9 TRUE 0.9637878
71: 9 TRUE 1.0128372
72: 9 TRUE 1.0128372
73: 9 TRUE 1.2201081
74: 9 TRUE 0.8388491
75: 9 TRUE 1.1205739
76: 9 TRUE 1.1553360
77: 9 TRUE 0.9030900
78: 9 TRUE 1.0606978

```

##      by1  by2      Wind

#####
#              #
# Exercise 7   #
#              #
#####
#dt2[c(TRUE,FALSE),Temp:=Temp+10L]
dt2[rep(c(TRUE,FALSE),length = .N),Temp:=Temp+10L]
##P.S Following Hyunwoo's comment ,it turns out that
dt2[c(TRUE,FALSE),Temp:=Temp+10L] does not work in data
##.table version '1.10.4' while this works for 1.9.6 .So I
have included the solution which should work for ##the
newer/current version and should be the preffered way .

#####
#              #
# Exercise 8   #
#              #
#####
dt2[, `:=`(Solar.R = Solar.R+ 10,Wind=Wind+10)]

#####
#              #
# Exercise 9   #
#              #
#####

dt2[,c("Solar.R", "Wind", "Temp"):= NULL]

#####
#              #
# Exercise 10  #
#              #
#####
dt2 <- as.data.table(airquality)
dt2[,c("a", "b"):= list(celcius <- (Temp-32)*5/9, kelvin <-
celcius+273.15)]

```