

**data.table**

# **Advanced 1hr Tutorial**

**Matthew Dowle**

**R/Finance, Chicago  
May 2013**

# Overview

- **Practice tomorrow's 6 min lightning talk as introduction now**
- **What would you like covered by this tutorial? (6 min)**
- **Deep dive (48 mins)**

**Deep dive ...**

.I

```
if (length(err <- allocation[,  
      if(length(unique(Price))>1) .I,  
      by=stock ]$V1 )) {  
  warning("Fills allocated to different  
accounts at different prices! Investigate.")  
  print(allocation[err])  
} else {  
  cat("Ok    All fills allocated to each  
account at same price\n")  
}
```

# .SD

```
stocks[, head(.SD, 2), by=sector]
```

```
stocks[, lapply(.SD, sum), by=sector]
```

```
stocks[, lapply(.SD, sum), by=sector,  
.SDcols=c("mcap", paste0("revenueFQ", 1:8))]
```

# All symbols

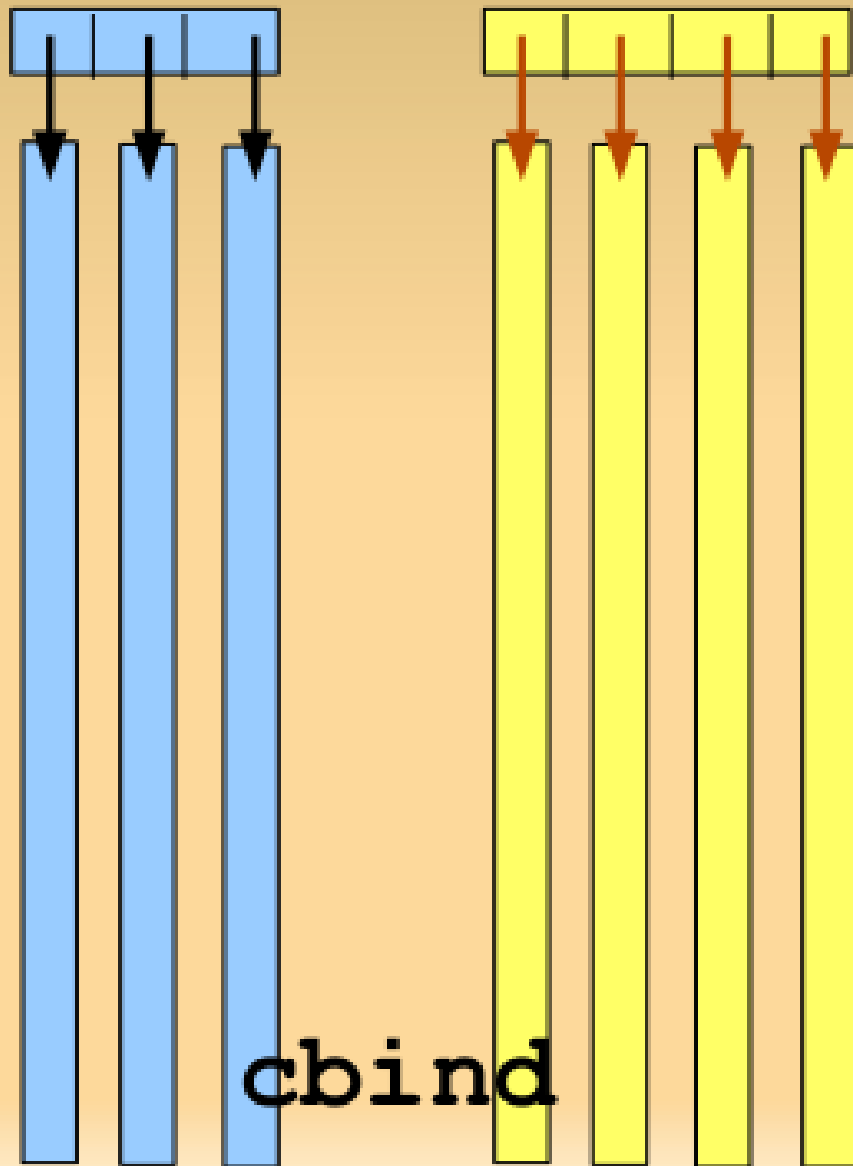
- **.N**
- **.SD**
- **.I**
- **.BY**
- **.GRP**

# All options

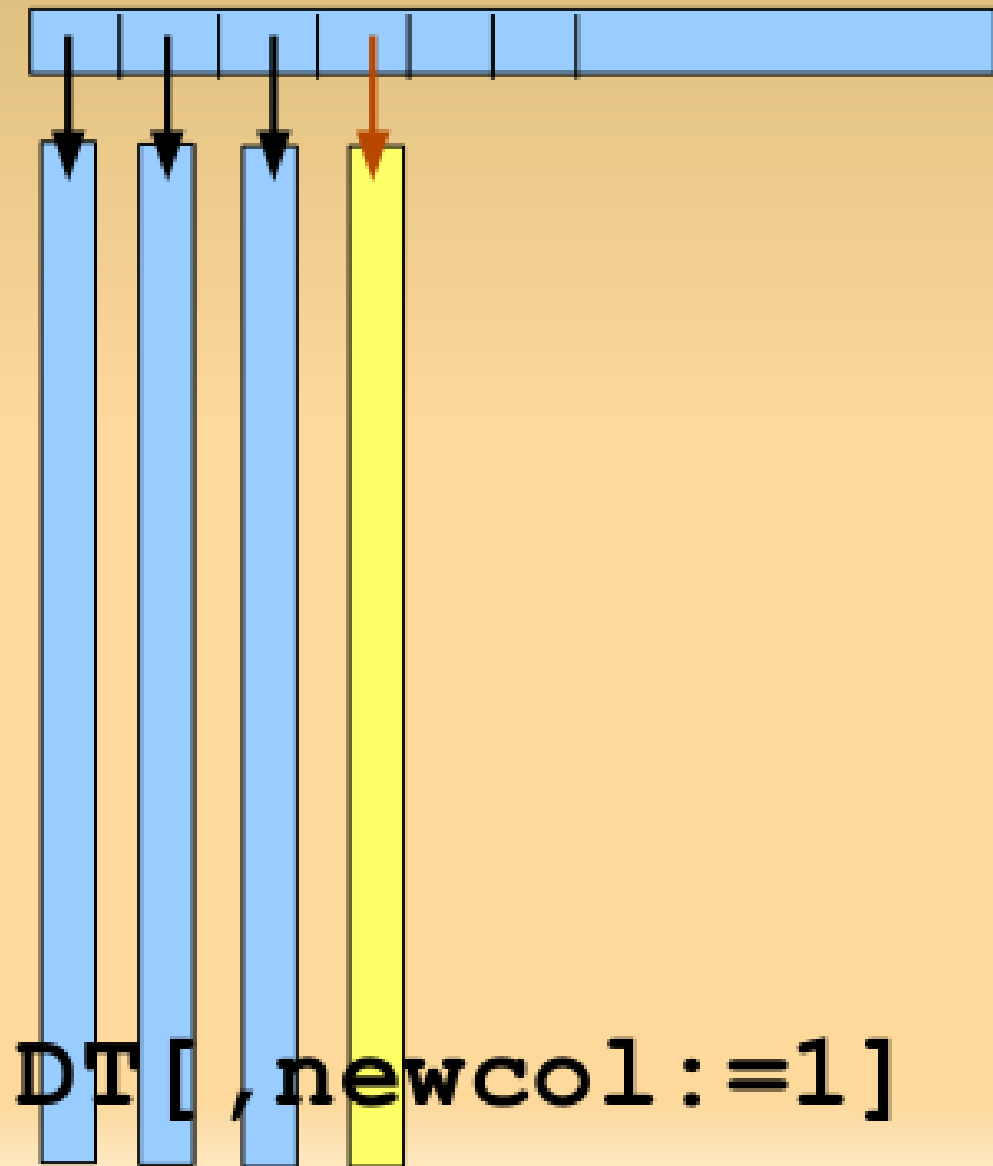
<code>datatable.verbose</code>	<code>FALSE</code>
<code>datatable.nomatch</code>	<code>NA_integer_</code>
<code>datatable.optimize</code>	<code>Inf</code>
<code>datatable.print.nrows</code>	<code>100L</code>
<code>datatable.print.topn</code>	<code>5L</code>
<code>datatable.allow.cartesian</code>	<code>FALSE</code>
<code>datatable.alloccol</code>	<code>quote(max(100L,ncol(DT)+64L))</code>
<code>datatable.integer64</code>	<code>"integer64"</code>

# Over allocation

data.frame



data.table





`:= and ` := ` ( )`

```
DT[col1==something, col2:=col3+1]
```

```
DT[, ` := ` (newCol1=mean(colA),  
           newCol2=sd(colA)),  
    by=sector]
```

# set\* functions

- `set()`
- `setattr()`
- `setnames()`
- `setcolorder()`
- `setkey()`
- `setkeyv()`

**53 examples in :**

**`example(data.table)`**

# Joins: X[Y]

- Vector search vs binary search
- One column == is ok, but not 2+ (revisit example in intro)
- J(), .(), list(), data.table()
- CJ()
- SJ()
- nomatch
- mult

# Rolling joins

```
roll = [-Inf,+Inf] | TRUE | FALSE
```

```
rollends = c(FALSE,TRUE)
```

By example on whiteboard

# by -vs- keyby

Order is always maintained :

- of groups (by order of first appearance)
- rows within groups.

keyby is a by as usual, followed by `setkeyv(DT, by)`

# Analogous to SQL

```
DT [ where,  
    select | update,  
    group by]  
[ having ]  
[ order by ]  
[ ]...[ ]
```

# Variable name repetition

```
DF[with(DF, order(-z, b)), ]
```

```
DT[order(-z, b)]
```

Stack Overflow :

How to sort a data.frame by columns in R



# Miscellaneous features

```
DT[, (myvar) := NULL]
```

Space and specials; e.g., `by="a, b, c"`

```
DT[4:7, newCol := 8] []
```

- extra `[]` to print at prompt
- auto fills rows 1:3 with NA

**Thank you!**

**<http://datatable.r-forge.r-project.org/>**