

# data.table: data.frame 2.0

## Examples

Mick Cooney

18 Feb 2015

# Eve Online



Massively Multiplayer Online game (MMO)

# Eve Online



Massively Multiplayer Online game (MMO)

- Internet Spaceships

# Eve Online



Massively Multiplayer Online game (MMO)

- Internet Spaceships
- Single-shard Universe

# Eve Online



Massively Multiplayer Online game (MMO)

- Internet Spaceships
- Single-shard Universe
- Well developed economy

# Eve Online



Massively Multiplayer Online game (MMO)

- Internet Spaceships
- Single-shard Universe
- Well developed economy
- Well-deserved reputation for being harsh and unforgiving

# Availability of Data

A game for data junkies...

# Availability of Data

A game for data junkies...

- CCP provides data API



# Availability of Data

A game for data junkies...

- CCP provides data API
- Provides access to market transactions, industry jobs, skill training, etc

# Availability of Data

A game for data junkies...

- CCP provides data API
- Provides access to market transactions, industry jobs, skill training, etc
- Third party data providers

# Availability of Data

A game for data junkies...

- CCP provides data API
- Provides access to market transactions, industry jobs, skill training, etc
- Third party data providers
- Real-time Datafeed – EMDR (scalable, high-availability)

# Examples of data.table Use

# Examples of data.table Use

- Calculate trading profitability

# Examples of data.table Use

- Calculate trading profitability
- Calculate item manufacturing cost

# Trading Profitability

```
linetype,characterID,characterName,transactTime,transactionID,transactionType,stationID,  
stationName,clientID,clientName,typeID,  
typeName,quantity,price,transactionFor
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 02:05:45,2594590547,buy,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,91727640,Lee Haxor,7447,  
425mm Prototype Gauss Gun,1,3901114.27,personal
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 02:19:33,2594596940,buy,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,1085859547,Roshan Sirdar,20353,  
1600mm Reinforced Steel Plates II,5,5000000.00,personal
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 02:19:45,2594597024,buy,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,1085859547,Roshan Sirdar,20353,  
1600mm Reinforced Steel Plates II,10,3000902.12,personal
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 02:43:14,2594607283,buy,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,572658539,Cornutus Metellus,20353,  
1600mm Reinforced Steel Plates II,5,5000000.00,personal
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 02:54:12,2594611985,sell,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,1779069929,Giurca II,3146,  
Heavy Neutron Blaster II,5,2167549.99,personal
```

```
TRADE,90192277,Gynax Gallenor,2012-09-01 04:00:42,2594642751,sell,60008494,  
Amarr VIII (Oris) - Emperor Family Academy,91882511,Kevin Reboolf,15729,  
Imperial Navy Energized Adaptive Nano Membrane,1,41677994.91,personal
```

```
...
```

---

```
1 get.trade.data <- function() {
2   trade.files <- dir('data/NOautoload', pattern = 'trades_', full.name = T);
3
4   read.trade.file <- function(file) {
5     file <- as.character(file);
6
7     return(read.csv(file,
8                     stringsAsFactors = F,
9                     colClasses = list(transactionID = 'character',
10                                      transactTime = 'POSIXct')));
11   }
12
13   alltrade.dt <- data.table(file = trade.files)[, read.trade.file(file), by = file];
14
15   alltrade.dt <- within(alltrade.dt, {
16     linetype = NULL;
17     file      = NULL;
18   });
19
20   trade.dt <- alltrade.dt[!duplicated(transactionID)];
21
22   return(trade.dt);
23 }
```

---



```
> head(trade.dt)
```

	characterID	characterName	transactTime	transactionID	transactionType	stationID
1:	90192277	Gynax Gallenor	2011-02-15 21:17:00	1722052107	sell	60010909
2:	90192277	Gynax Gallenor	2011-02-15 21:17:00	1722052096	buy	60010213
3:	90192277	Gynax Gallenor	2011-02-15 19:44:00	1721945501	buy	60011740
4:	90192277	Gynax Gallenor	2011-02-15 19:04:00	1721901880	buy	60011743
5:	90192277	Gynax Gallenor	2011-02-15 19:04:00	1721901804	buy	60011743
6:	90192277	Gynax Gallenor	2011-02-15 18:37:00	1721872768	buy	60011740

	stationName	clientID	clientName	typeID
1:	Oursulaert VII - Moon 3 - Duvalle Laboratories Research Center	1352680154	KuRSed Pestis	5489
2:	Noghere VIII - Moon 18 - CreoDron Warehouse	90210980	DSquare1980	226
3:	Oursulaert III - Federation Navy Testing Facilities	90279444	Jarpin	223
4:	Oursulaert IV - Federation Navy Testing Facilities	90365144	Mark SchultzIII	226
5:	Oursulaert IV - Federation Navy Testing Facilities	90365144	Mark SchultzIII	223
6:	Oursulaert III - Federation Navy Testing Facilities	90413887	Jackson Moore	226

	typeName	quantity	price	transactionFor
1:	Local Hull Conversion Expanded Cargo I	2	910000.17	personal
2:	Lead Charge M	400	9.17	personal
3:	Iron Charge M	4046	3.51	personal
4:	Lead Charge M	100	17.29	personal
5:	Iron Charge M	100	3.51	personal
6:	Lead Charge M	1872	17.29	personal

---

```
1 show.last.trades <- function(typeID, count = 10, side = 'buy', tradedata = trade.dt) {
2   showID <- typeID;
3
4   show.dt <- trade.dt[typeID %in% showID][transactionType == side]
5               [, list(transactionID, transactTime, transactionType,
6                       typeID, typeName, quantity, price)];
7
8   print(tail(show.dt, n = count));
9
10  return();
11 }
```

---

```
> show.last.trades(7447)
```

	transactionID	transactTime	transactionType	typeID	typeName	quantity
1:	2585507342	2012-08-20 15:04:16	buy	7447 425mm Prototype	Gauss Gun	1
2:	2585507388	2012-08-20 15:04:20	buy	7447 425mm Prototype	Gauss Gun	1
3:	2585859680	2012-08-20 22:49:44	buy	7447 425mm Prototype	Gauss Gun	3
4:	2589493779	2012-08-25 13:58:36	buy	7447 425mm Prototype	Gauss Gun	1
5:	2590013633	2012-08-26 02:06:24	buy	7447 425mm Prototype	Gauss Gun	5
6:	2591525948	2012-08-27 20:33:01	buy	7447 425mm Prototype	Gauss Gun	4
7:	2591934755	2012-08-28 11:27:04	buy	7447 425mm Prototype	Gauss Gun	4
8:	2592540494	2012-08-29 05:28:40	buy	7447 425mm Prototype	Gauss Gun	1
9:	2592662562	2012-08-29 10:51:30	buy	7447 425mm Prototype	Gauss Gun	5
10:	2594590547	2012-09-01 02:05:45	buy	7447 425mm Prototype	Gauss Gun	1

price

1:	3262343.41
2:	3262343.41
3:	3262343.45
4:	3262381.35
5:	3350000.02
6:	3800020.01
7:	3800056.04
8:	3800101.21
9:	3800107.37
10:	3901114.27

---

```
1 calculate.avg.price <- function(trade.dt) {
2   avgprice.dt <- trade.dt[, list(volume = sum(quantity),
3                                   cash   = sum(quantity * price),
4                                   avgprice = (sum(quantity * price) / sum(quantity))),
5                                   by = list(typeID, typeName, transactionType)];
6
7   setkey(avgprice.dt, typeID, transactionType);
8
9   return(avgprice.dt);
10 }
```

---

```
> avgprice.dt <- calculate.avg.price(trade.dt[transactTime >= as.POSIXct('2012-09-01')])
> avgprice.dt[, price := format(avgprice, big.mark = ',', scientific = F)]
```

	typeID	typeName	transactionType	volume	cash
1:	34	Tritanium	buy	49991443	277885841.55
2:	35	Pyerite	buy	23908532	258529648.64
3:	36	Mexallon	buy	2500000	137060701.94
4:	37	Isogen	buy	1456943	121081494.79
5:	38	Nocxium	buy	200000	103935941.80
---					
135:	28578	Ice Harvester Upgrade II	buy	18	19690309.70
136:	28578	Ice Harvester Upgrade II	sell	18	23821548.08
137:	30836	Salvager II	sell	1	1399753.99
138:	31055	Medium Trimark Armor Pump I	sell	5	6494667.75
139:	31754	Medium Anti-Thermal Screen Reinforcer I	buy	5	850035.00

  

	avgprice	price
1:	5.55866814146573e+00	5.55866814146573
2:	1.08132799052656e+01	10.81327990526562
3:	5.48242807760000e+01	54.82428077600000
4:	8.31065421159235e+01	83.10654211592355
5:	5.19679709000000e+02	519.67970900000000
---		
135:	1.09390609444444e+06	1,093,906.09444444440305
136:	1.32341933777777e+06	1,323,419.337777777757496
137:	1.39975399000000e+06	1,399,753.98999999999069
138:	1.29893355000000e+06	1,298,933.55000000004657
139:	1.70007000000000e+05	170,007.00000000000000

# Cost of Manufacturing

# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system

# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system
- Nanotech fabbing



# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system
- Nanotech fabbing
- Blueprint + Materials + Manufacturing Slot  $\rightarrow$  Item

# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system
- Nanotech fabbing
- Blueprint + Materials + Manufacturing Slot → Item
- Better blueprint → Less waste

# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system
- Nanotech fabbing
- Blueprint + Materials + Manufacturing Slot → Item
- Better blueprint → Less waste
- Major update to system with Crius Expansion (June 2014)

# Cost of Manufacturing

- Eve has an extensive manufacturing/crafting system
- Nanotech fabbing
- Blueprint + Materials + Manufacturing Slot → Item
- Better blueprint → Less waste
- Major update to system with Crius Expansion (June 2014)
- Pre-Crius:

$$\text{Waste} = \text{Base Waste (10\%)} \times \left( \frac{1}{1+\text{ME}} \right) \times \text{Material Amount}$$

```

1 calculate.construction.cost <- function(typeID, ME = 0, price.dt = pricedata.dt, verbose = F) {
2   material.dt <- within(get.blueprint.data(typeID), {
3     if(ME >= 0) {
4       waste = round((0.1/(1 + ME)) * quantity * wasteFactor, 0);
5     } else {
6       waste = round((0.1 * abs(ME)) * quantity * wasteFactor, 0);
7     }
8
9     required = quantity + waste;
10
11     waste = required - quantity;
12   });
13
14   setkey(material.dt, typeID);
15   setkey(price.dt, typeID);
16
17   material.dt <- merge(material.dt, price.dt[, list(typeID, price)], all.x = T, by = c('typeID'));
18
19   cost.dt <- within(material.dt, {
20     requiredCost = required * price;
21     wasteCost    = waste    * price;
22   });
23
24   if(!verbose) {
25     cost.dt <- cost.dt[, list(materialCost = sum(requiredCost),
26                               buildCost   = sum(requiredCost),
27                               buildWaste  = sum(wasteCost),
28                               maxWaste    = max(wasteCost),
29                               wasteRatio  = sum(wasteCost) / sum(requiredCost)),
30     by = list(constructTypeID, constructTypeName)];
31   }
32
33   return(cost.dt);
34 }

```

```
> calculate.construction.cost(627, ME = 30, price.dt = pricedata.dt, verbose = F)
constructTypeID constructTypeName materialCost buildCost buildWaste maxWaste wasteRatio
1:          627          Thorax  9896956.83 9896956.83  31554.05 10027.63 0.00318825781924725
```

```
> calculate.construction.cost(627, ME = 30, price.dt = pricedata.dt, verbose = T)
typeID constructTypeID constructTypeName typeName quantity wasteFactor required waste price
1:    34          627          Thorax Tritanium  524216             1  525907 1691   5.93
2:    35          627          Thorax Pyerite  131210             1  131633 423  11.99
3:    36          627          Thorax Mexallon  34124             1   34234 110  59.95
4:    37          627          Thorax Isogen   8270             1   8297 27 138.45
5:    38          627          Thorax Nocxium  2035             1   2042 7 698.00
6:    39          627          Thorax Zydrine   510             1    512 2 618.00
7:    40          627          Thorax Megacyte   130             1    130 0 1978.99
```

```
wasteCost requiredCost
1: 10027.63 3118628.51
2: 5071.77 1578279.67
3: 6594.50 2052328.30
4: 3738.15 1148719.65
5: 4886.00 1425316.00
6: 1236.00 316416.00
7:    0.00 257268.70
```

# Crius Changes

# Crius Changes

- Major Changes



# Crius Changes

- Major Changes
- Complete change in how Material Efficiency (ME) worked

# Crius Changes

- Major Changes
- Complete change in how Material Efficiency (ME) worked
- $\text{Required} = \text{Quantity} \times \text{Job Runs} \times (1 - \text{ME}\%)$

---

```
1 calculate.construction.cost <- function(typeID, ME = 0, runs = 1, price.dt = pricedata.dt,
2                                     verbose = FALSE, dbconnect = data.connection) {
3
4     stopifnot(ME >= 0 & ME <= 10);
5     stopifnot(runs > 1);
6
7     material.dt <- get.blueprint.data(typeID, dbconnect);
8
9     setkey(material.dt, typeID);
10    setkey(price.dt,    typeID);
11
12    material.dt <- merge(material.dt,
13                        price.dt[, list(matTypeID = typeID, price)],
14                        all.x = TRUE, by = c('matTypeID'));
15
16    cost.dt <- material.dt;
17    cost.dt[, required      := ceiling(quantity * runs * (1 - ME / 100))];
18    cost.dt[, requiredCost := required * price];
19
20
21    return(cost.dt);
22 }
```

---

# Cartesian Expansion

```
1 > set.seed(42); data.dt <- data.table(keyval = LETTERS[1:5], val = sample(1:10, 5))
2   keyval val
3 1:      A  10
4 2:      B   9
5 3:      C   3
6 4:      D   6
7 5:      E   4
8
9 > set.seed(42); data.dt[, .SD[rep(1, val)][, val := 1:val[1]], by = keyval]
10  keyval val
11 1:      A   1
12 2:      A   2
13 3:      A   3
14 4:      A   4
15 5:      A   5
16 6:      A   6
17 7:      A   7
18 8:      A   8
19 9:      A   9
20 10:     A  10
21 11:     B   1
22 ...
23 28:     D   6
24 29:     E   1
25 30:     E   2
26 31:     E   3
27 32:     E   4
28   keyval val
```

# Summary

`data.table` is the schnizzle