



# data.table an introduction and best practices

Bill Gold

September 17<sup>th</sup> 2018



# alternate title



**RESTRICTED**

This presentation is rated R, for excessive data.table processing speed and is intended for mature audiences

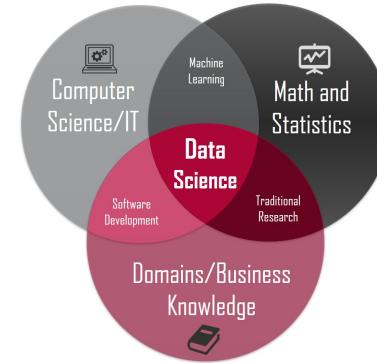
# data.table Where are We Going?

- I. why data.table?
- II. DT [ **i**, **j**, **by** ] [ **c** ] - syntax
- III. data exploration
- IV. something unexpected
- V. cool next steps imho

# about bill



Management Consulting  
Implementation business permanent project simply  
improving skills organization development performing  
industry proprietary objective frameworks efficient  
helping help strategy company hiring  
consultation technology practices consultants  
effective practice



Ataeva  
HAYSTREAM

+\$500MM ROI

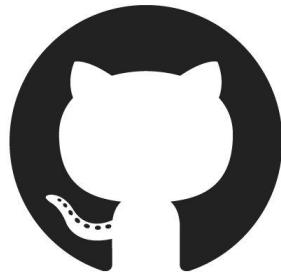
Hundreds of Models

+10 Platforms



COLUMBIA UNIVERSITY  
IN THE CITY OF NEW YORK





[linkedin.com/in/\*\*billcgold\*\*](https://linkedin.com/in/billcgold)

[github.com/\*\*billcgold\*\*](https://github.com/billcgold)

**billcgold@gmail.com**

Tonights Presentation <http://bit.ly/2NkuRJi>

## Is hiring

- 2 data scientists / consultants
- 3-5 years experience with R (or Python)
- Fintech experience a +

# About `data.table`

- v1.0 released by CRAN in 2006
- by Matt Dowle et all
- high-performance version of `data.frame`
- 4th largest Stack Overflow tag for an R package

# data.table Where are We Going?

- I. why data.table?
- II. DT [ i, j, by ] [ c ] - syntax
- III. data exploration
- IV. something unexpected
- V. cool next steps imho

fast

concise

integrates well with R

fast

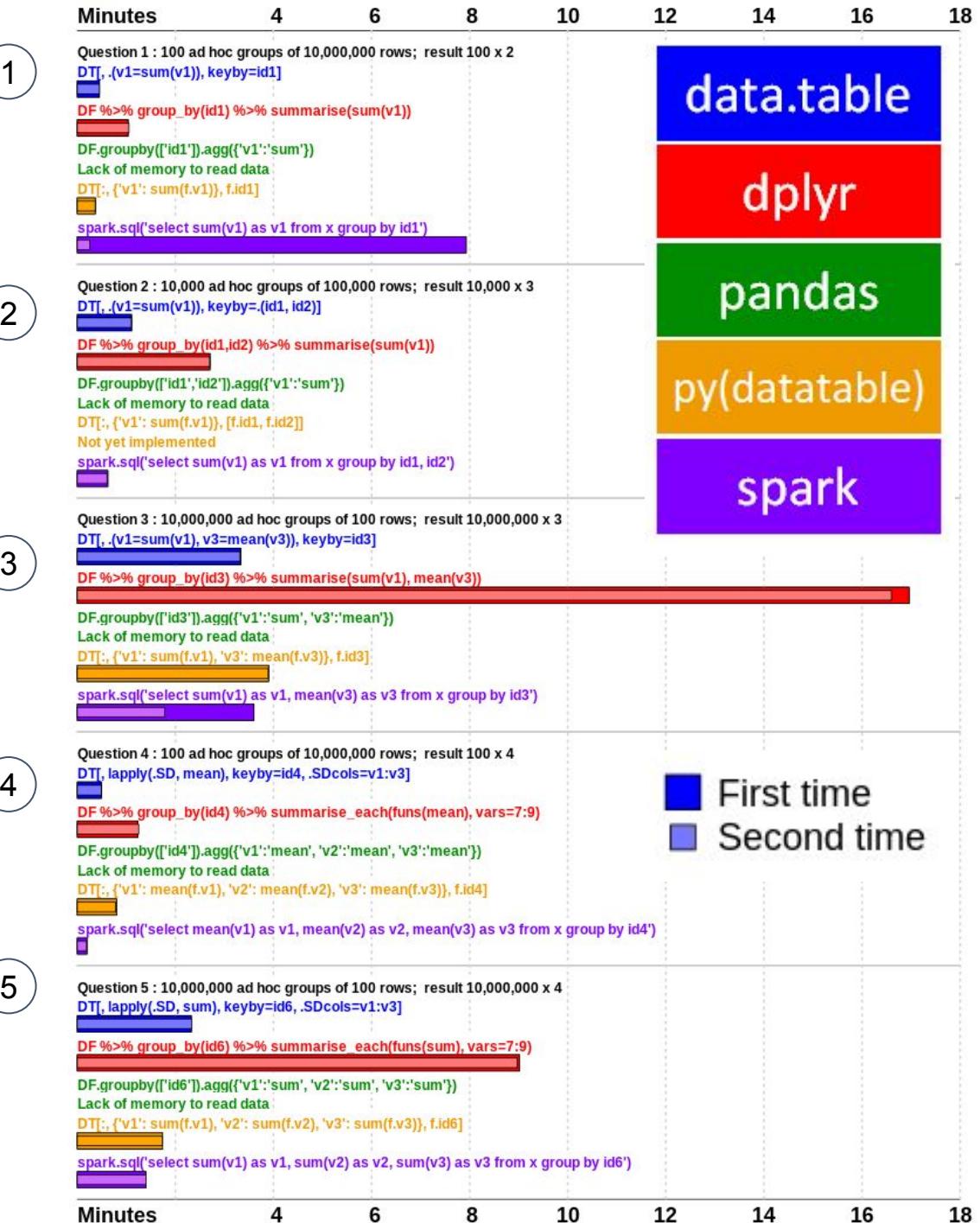
fast is relative



# data.table is objectively fast

1B rows, 9 columns, 50GB

fast



- data.table 1.11.5 - 2018-09-10 - Total: \$0.19 for 23 minutes
- dplyr 0.7.99.9000 - 2018-08-27 - Total: \$0.77 for 92 minutes
- pandas 0.23.4 - 2018-08-04 - Total: \$NA for NA minutes
- (py)datatable 0.6.0 - 2018-09-08 - Total: \$NA for NA minutes
- spark 2.3.1 - 2018-06-08 - Total: \$0.18 for 22 minutes

# why is this package different?

in memory

by reference

parallel processing (at times)

algorithms leveraging R's efficient internal structures (global character cache)

radix sorting from Terdiman and Herf, contributing back to base R

setkey contiguous, then use efficient contiguous methods

concise

MY PAPER IS ENTITLED,  
"TYRANNOSAURUS REX:  
FEARSOME PREDATOR OR  
LOATHSOME SCAVENGER?"



AHEM..."I SAY TYRANNOSAURS  
WERE PREDATORS, BECAUSE  
IT WOULD BE SO BOGUS IF  
THEY JUST ATE THINGS  
THAT WERE ALREADY DEAD.  
THE END."



© 1993 Watterson Distributed by Universal Uclick



I'M A CONCISE  
WRITER, OK???



aggregate & filter mtcars ...

	cyl	mpg-mean
1:	6	20.56667
2:	4	28.07500
3:	8	15.40000

sql

**SELECT**  
**FROM**  
**WHERE**  
**GROUP BY**

cyl, mean (mpg)  
mtcars  
am = 1  
cyl

65 characters  
27 repetitive



# data.frame

```
aggregate (  
  mtcars$mpg [mtcars$am==1]  
, by = list (cyl=mtcars$cyl[ mtcars$am==1 ]  
, FUN = mean )
```

function-centric

89 characters  
41 repetitive

data.table

mtcars [am == 1, mean (mpg), cyl ]

data-centric

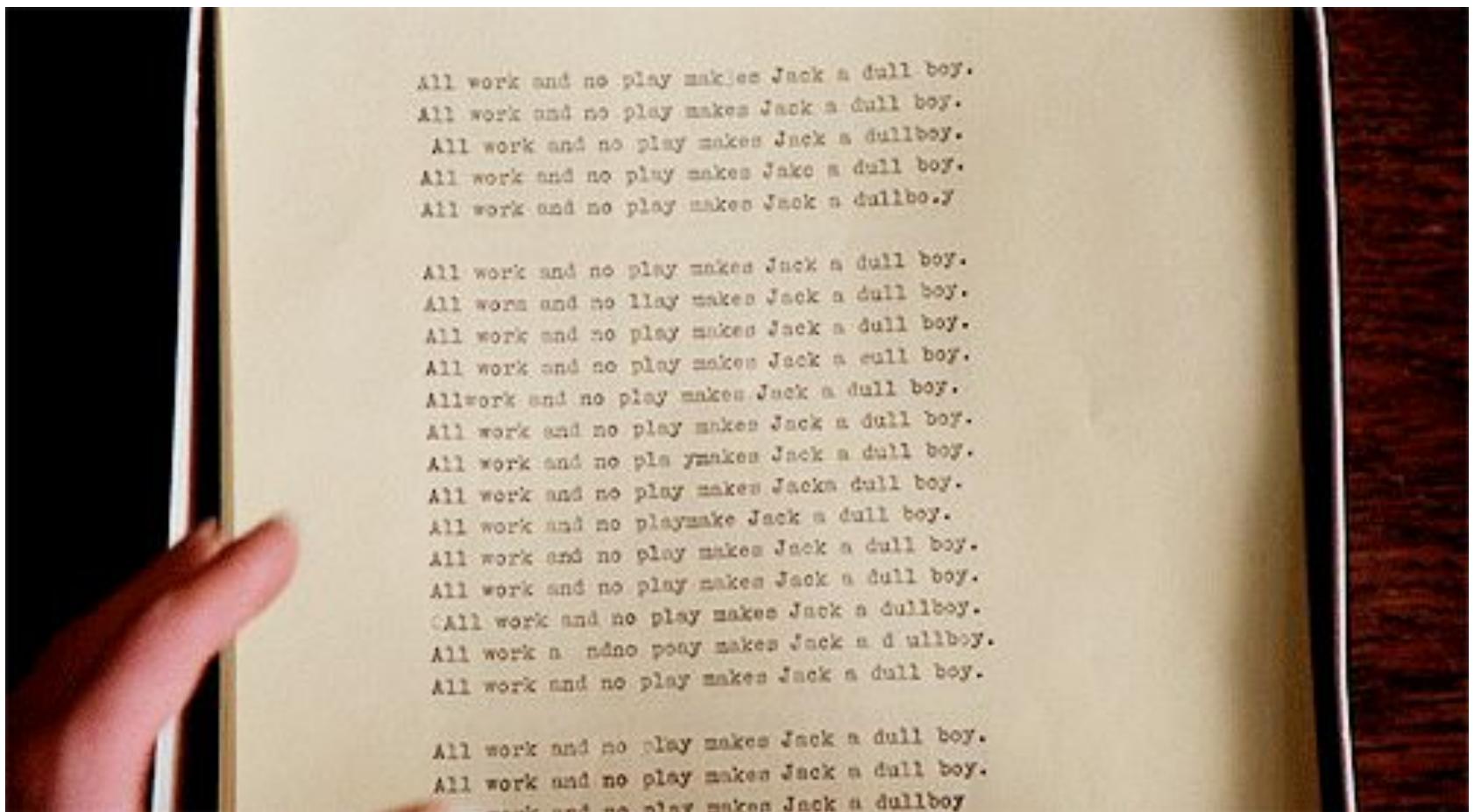
27 characters  
4 repetitive

6 vs 34

**data.table vs SQL**

**count of repetitive characters**

# concise



“any R function from any R package can be used in queries”

*Matt Dowle*

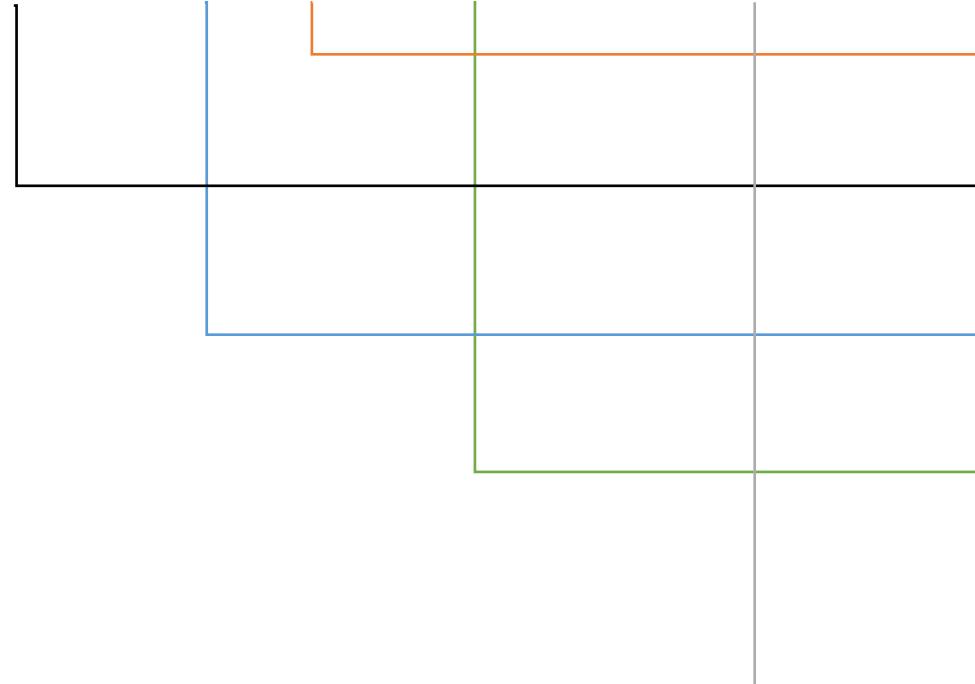


# data.table Where are We Going?

- I. why data.table?
- II. **DT [ i, j, by ] [ c ] - syntax**
- III. data exploration
- IV. something unexpected
- V. cool next steps imho

# data.table and SQL SELECT

**DT [ i, j, by ] [ c ]**



SELECT

select-list

FROM

table

WHERE

row filters

GROUP BY

aggregates

ORDER BY

HAVING

aggregate + filter

# setup example

```
library(data.table)
```

```
dt.mtcars <- data.table ( mtcars, keep.rownames=T )
```

# filter

```
dt.mtcars [ cyl == 8 ]
```

	rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	
1:	Hornet	Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
2:	Duster	360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
3:	Merc	450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
4:	Merc	450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
5:	Merc	450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
6:	Cadillac	Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
7:	Lincoln	Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
8:	Chrysler	Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
9:	Dodge	Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
10:	AMC	Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
11:	Camaro	Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
12:	Pontiac	Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
13:	Ford	Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
14:	Maserati	Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8

# filter multiple conditions

```
dt.mtcars [ cyl == 8 &  
           wt < 4 &  
           rn %like% 'Merc' ]
```

	rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1:	Merc 450SL	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
2:	Merc 450SLC	15.2	8	275.8	180	3.07	3.78	18.0	0	0	3	3

# filter row numbers

```
dt.mtcars [ 1:5 ]
```

		rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1:	Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4	4
2:	Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4	4
3:	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1	1
4:	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1	1
5:	Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2	2

# select-clause    vector output

```
dt.mtcars [ , rn ]
```

```
[1] "Mazda RX4"           "Mazda RX4 Wag"        "Datsun 710"          "Hornet 4 Drive"  
[5] "Hornet Sportabout"   "Valiant"            "Duster 360"         "Merc 240D"  
[9] "Merc 230"             "Merc 280"            "Merc 280C"          "Merc 450SE"  
[13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood" "Lincoln Continental"  
[17] "Chrysler Imperial"   "Fiat 128"            "Honda Civic"        "Toyota Corolla"  
[21] "Toyota Corona"       "Dodge Challenger"    "AMC Javelin"        "Camaro Z28"  
[25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"      "Lotus Europa"  
[29] "Ford Pantera L"      "Ferrari Dino"         "Maserati Bora"      "Volvo 142E"
```

# select-clause vector output

```
# same as previous, much faster  
dt.mtcars [["rn"]]
```

```
[1] "Mazda RX4"           "Mazda RX4 Wag"        "Datsun 710"      "Hornet 4 Drive"  
[5] "Hornet Sportabout"   "Valiant"            "Duster 360"     "Merc 240D"  
[9] "Merc 230"             "Merc 280"            "Merc 280C"       "Merc 450SE"  
[13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood" "Lincoln Continental"  
[17] "Chrysler Imperial"   "Fiat 128"            "Honda Civic"    "Toyota Corolla"  
[21] "Toyota Corona"       "Dodge Challenger"   "AMC Javelin"    "Camaro Z28"  
[25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"   "Lotus Europa"  
[29] "Ford Pantera L"      "Ferrari Dino"        "Maserati Bora"  "Volvo 142E"
```

# select-clause data.table output

```
dt.mtcars [ 1:5 , list (rn) ]
```

```
          rn  
1:      Mazda RX4  
2:  Mazda RX4 Wag  
3:    Datsun 710  
4:  Hornet 4 Drive  
5: Hornet Sportabout
```

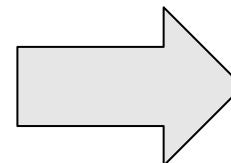
# select-clause data.table output

```
dt.mtcars [ 1:5 , list (rn, cyl, hp) ]
```

			rn	cyl	hp
1	Mazda	RX4		6	110
2	Mazda	RX4 Wag		6	110
3	Datsun	710		4	93
4	Hornet	4 Drive		6	110
5	Hornet	Sportabout		8	175



4 selects  
1 result



4 selects      1 result

```
dt.mtcars [ , list (rn, cyl, hp) ]
```

```
dt.mtcars [ , .(rn, cyl, hp) ]    # .() = list()
```

```
dt.mtcars [ , `c ('rn', 'cyl', 'hp') , with = F ]
```

```
dt.mtcars [ , c ( 1, 3, 5) ]
```

# select-clause data.table output

```
dt.mtcars [ 1:5 , .SD, .SDcols = rn:cyl ]
```

		rn	mpg	cyl
1:	Datsun	710	22.8	4
2:	Merc	240D	24.4	4
3:	Merc	230	22.8	4
4:	Fiat	128	32.4	4
5:	Honda	Civic	30.4	4

# variable column names

..

```
variable.col.name <- 'rn'  
dt.mtcars [ 1:5 , ..variable.col.name ]
```

```
      V1  
1: Mazda RX4  
2: Mazda RX4 Wag  
3: Datsun 710  
4: Hornet 4 Drive  
5: Hornet Sportabout
```

# group by

```
dt.mtcars [ , . (mean(mpg)) , by = cyl ]
```

```
cyl      v1
1:   6 19.74286
2:   4 26.66364
3:   8 15.10000
```

# group by (cont'd)

```
dt.mtcars [ , . ( mpg = mean(mpg) ) , by = cyl ]
```

```
cyl      mpg
1:   6 19.74286
2:   4 26.66364
3:   8 15.10000
```

# group by (cont'd)

```
dt.mtcars [ , lapply ( .SD, mean )  
            , .SDcols = mpg:carb  
            , by = cyl ]
```

	cyl	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1:	6	19.74286	6	183.3143	122.28571	3.585714	3.117143	17.97714	0.5714286	0.4285714	3.857143	3.428571
2:	4	26.66364	4	105.1364	82.63636	4.070909	2.285727	19.13727	0.9090909	0.7272727	4.090909	1.545455
3:	8	15.10000	8	353.1000	209.21429	3.229286	3.999214	16.77214	0.0000000	0.1428571	3.285714	3.500000

what's different?

# chaining



# having (chaining)

```
dt.mtcars [ , . (mpg = mean(mpg)) , by=cyl ] [ mpg > 16 ]
```

```
cyl      mpg
1:   6 19.74286
2:   4 26.66364
```

# order by

```
dt.mtcars [ , . (mpg = mean(mpg)) ,by=cyl] [order(-mpg)]
```

```
cyl      mpg  
1:   4 26.66364  
2:   6 19.74286  
3:   8 15.10000
```



what's our  
vector  
Victor?



# vectors and %in%

```
1:2  
1:6  
1:2 %in% 1:6  
1:6 %in% 1:2
```

```
> 1:2  
[1] 1 2  
> 1:6  
[1] 1 2 3 4 5 6  
> 1:2 %in% 1:6  
[1] TRUE TRUE  
> 1:6 %in% 1:2  
[1] TRUE TRUE FALSE FALSE FALSE FALSE
```

# vectors and %in% (cont'd)

```
dt.mtcars [ , cyl ]  
dt.mtcars [ , cyl ] %in% c(4,6)
```

```
> dt.mtcars [ , cyl ]  
[1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 4 8 6 8 4  
> dt.mtcars [ , cyl ] %in% c(4,6)  
[1] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
[16] FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE  
[31] FALSE TRUE
```

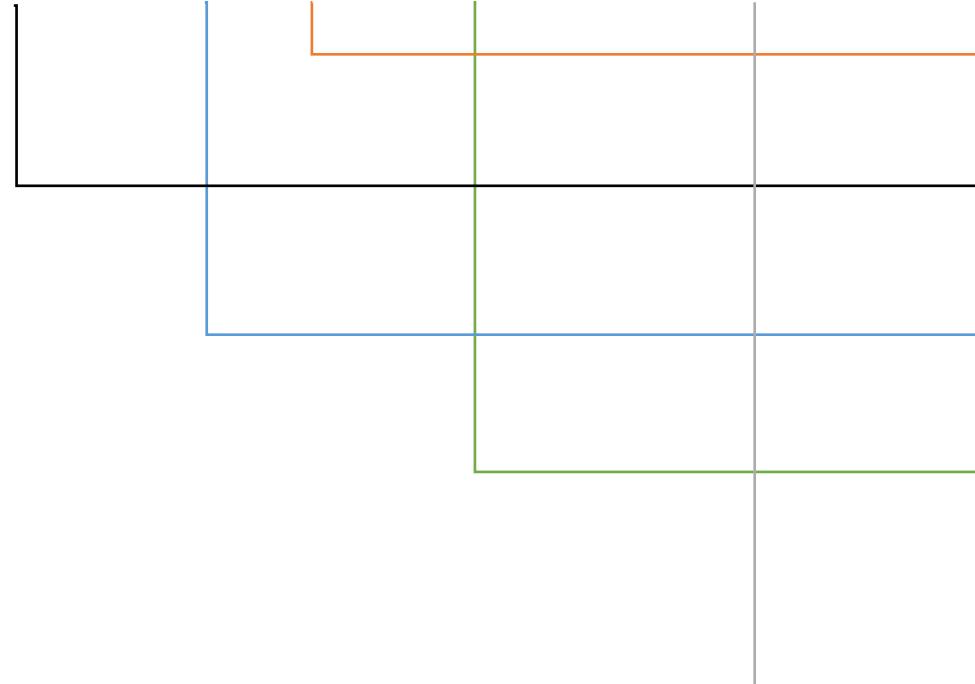
# filters vectors %in%

```
dt.mtcars [ cyl %in% c(4,6) ] [ 1:5 ] [ order (cyl) ]
```

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

# data.table and SQL SELECT

**DT [ i, j, by ] [ c ]**



SELECT

select-list

FROM

table

WHERE

row filters

GROUP BY

aggregates

ORDER BY

aggregate + filter

# join

```
dt.mtcars.cyl.aggr <- dt.mtcars [ , . (mpg.mean.cyl=mean(mpg)
                                         , mpg.sd.cyl=sd(mpg)
                                         , hp.mean.cyl=mean(hp)
                                         , hp.sd.cyl=sd(hp) )
                                         , by = cyl ]
```

```
> dt.mtcars.cyl.aggr
   cyl mpg.mean.cyl mpg.sd.cyl hp.mean.cyl hp.sd.cyl
1:  4    26.66364   4.509828    82.63636  20.93453
2:  6    19.74286   1.453567   122.28571  24.26049
3:  8    15.10000   2.560048   209.21429  50.97689
```

# join (cont'd)

```
setkeyv ( dt.mtcars, c('cyl') )  
setkeyv ( dt.mtcars.cyl.aggr, c('cyl') )
```

```
DT <- dt.mtcars [ dt.mtcars.cyl.aggr ]  
DT [ 1:5 ]
```

	rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	mpg.mean.cyl	mpg.sd.cyl	hp.mean.cyl	hp.sd.cyl
1:	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	26.66364	4.509828	82.63636	20.93453
2:	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	26.66364	4.509828	82.63636	20.93453
3:	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	26.66364	4.509828	82.63636	20.93453
4:	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	26.66364	4.509828	82.63636	20.93453
5:	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	26.66364	4.509828	82.63636	20.93453

# join (cont'd)

JOIN type	DT syntax	data.table::merge() syntax
INNER	X[Y, nomatch=0]	merge(X, Y, all=FALSE)
LEFT OUTER	Y[X]	merge(X, Y, all.x=TRUE)
RIGHT OUTER	X[Y]	merge(X, Y, all.y=TRUE)
FULL OUTER	-	merge(X, Y, all=TRUE)
FULL OUTER WHERE NULL (NOT INNER)	-	merge(X, Y, all=TRUE), subset NA

- list
- .. variable select-list
- X[Y] right outer join

:D big grin

:o) clown

:= update

# update

```
SELECT      cyl, mpg, am  
FROM        mtcars  
WHERE       am = 1
```

=

```
UPDATE      cyl = 1  
SET          mtcars  
FROM         WHERE  
           am = 1
```

:=

# update add a new column :=

```
dt.mtcars [ , N := 1 ]  
dt.mtcars [ 1:5 ]
```

			rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	N
1:	Mazda	RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4	4	1
2:	Mazda	RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4	4	1
3:	Datsun	710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1	1	1
4:	Hornet	4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1	1	1
5:	Hornet	Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2	2	1

# update :=

```
v.manufacturer <- gsub("[A-Za-z]+.*"  
, "\\\1"  
, dt.mtcars [ , rn ] )  
dt.mtcars [ , manufacturer := v.manufacturer ]  
dt.mtcars [ 1:5 ]
```

	rn	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	N	manufacturer
1:	Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4	1	Mazda
2:	Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4	1	Mazda
3:	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1	1	Datsun
4:	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1	1	Hornet
5:	Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2	1	Hornet

update :=

```
dt.mtcars [ manufacturer == 'Merc' , is.merc := 1 ]
```

```
dt.mtcars [ , .N, by = is.merc ]
```

```
> dt.mtcars [ , .N, by = is.merc ]  
  is.merc N  
1:      NA 25  
2:      1  7
```

# other functions

fread

fwrite

readRDS

saveRDS

setnames

setcolororder

dcast & melt

.N .SD .I .GRP .BY

format.q  
round

DT [ i, j, by ] [ c ]

DT [ where,  
select-clause,  
group by ] [ order | having ]

# data.table Where are We Going?

- I. why data.table?
- II. DT [ i, j, by ] [ c ] - syntax
- III. data exploration**
- IV. something unexpected
- V. cool next steps imho

# time series aggregate

```
1 #####  
2 ##### calculate percentile variables  
3 #####  
4 cutoffs <- c(0:10)/10  
5 by.vars <- c('FactorId.01', 'FactorId.02')  
6 setkeyv (time.series, by.vars)  
7 quantile.time.series <- time.series [ Period >= '2008-01-01'  
8 , . ( Variable.01      = quantile ( Variable.01, cutoffs, na.rm = T )  
9 ,      , Variable.02   = quantile ( Variable.02, cutoffs, na.rm = T )  
10 ,     , Variable.03  = quantile ( Variable.03, cutoffs, na.rm = T )  
11 ,    , Variable.04   = quantile ( Variable.04, cutoffs, na.rm = T )  
12 ,   , Variable.05   = quantile ( Variable.05, cutoffs, na.rm = T )  
13 ,  , Variable.06   = quantile ( Variable.06, cutoffs, na.rm = T )  
14 , by = by.vars ]
```

# scoring code

```
1 #####  
2 #### score a probit model  
3 #####  
4  
5 time.series [ , infinity.war.exp := (- 8675309  
6                                       + captain.america * 0.1111  
7                               + wonder.woman * 0.2222  
8                               + iron.man * 0.3333  
9                               + spider.man * 0.4444  
10                               + the.hulk * 0.5555  
11                               + aqua.man * -0.6666 ) ]  
12  
13 time.series [ , infinity.war.score := exp (exp.equation) / ( 1 + exp(exp.equation) ) ]  
14
```

# enhanced data dictionary (edd)

```
display.data ( edd (  
  dt = data.table (iris)  
  , path_out = '/home/bill/'  
  , file_out = 'iris.edd'  
  , return_edd = T ) )
```

Enhanced Data Dictionary (edd) for iris dataset															first_row	
	col_name	col_type	num_unique	num_na	num_blank	fill_rate	edd_mean	edd_pctl_01	edd_pctl_05	edd_pctl_25	edd_pctl_75	edd_pctl_95	edd_pctl_99	edd_sum	frequency_10	first_row
1	Sepal.Length	numeric	35	0	1.00	5.8433333333333	4.4	4.6	5.1	6.4	7.255	7.7	876.5	5:10,5.1:9,6.3:9,5.7:8,6.7:8,5.5:7,5.8:7,6.4:7,4.9:6,5.4:6,	5.1	
2	Sepal.Width	numeric	23	0	1.00	3.0573333333333	2.2	2.345	2.8	3.3	3.8	4.151	458.6	3:26,2:8,14,3:2:13,3:4:12,3:1:11,2:9:10,2:7:9,2:5:8,3:3:6,3:5:6,	3.5	
3	Petal.Length	numeric	43	0	1.00	3.758	1.149	1.3	1.6	5.1	6.1	6.7	563.7	1:4:13,1:5:13,4:5:8,5:1:8,1:3:7,1:6:7,5:6:6:4:5:4:7:5:4:9:5,	1.4	
4	Petal.Width	numeric	22	0	1.00	1.1993333333333	0.1	0.2	0.3	1.8	2.3	2.5	179.9	0:2:29,1:3:13,1:5:12,1:8:12,1:4:8,2:3:8,0:3:7,0:4:7,1:7:2:6,	0.2	
5	Species	factor	3	0	0	1.00								setosa:50,versicolor:50,virginica:50,	1	

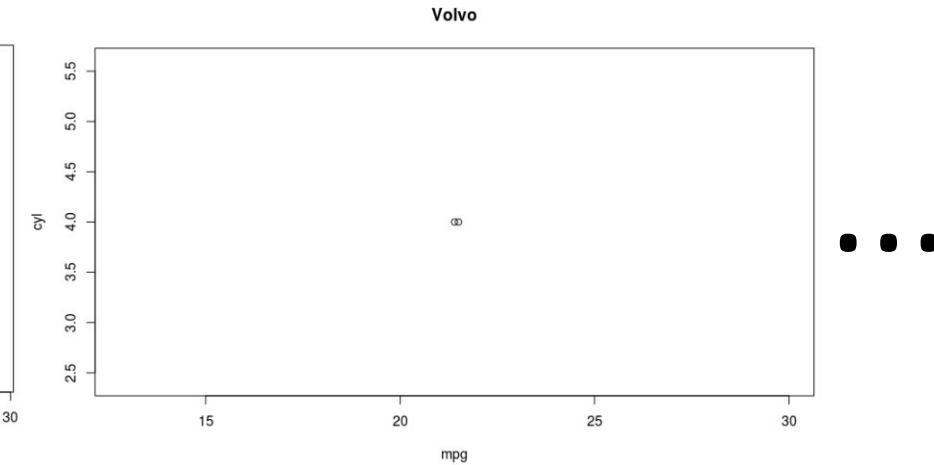
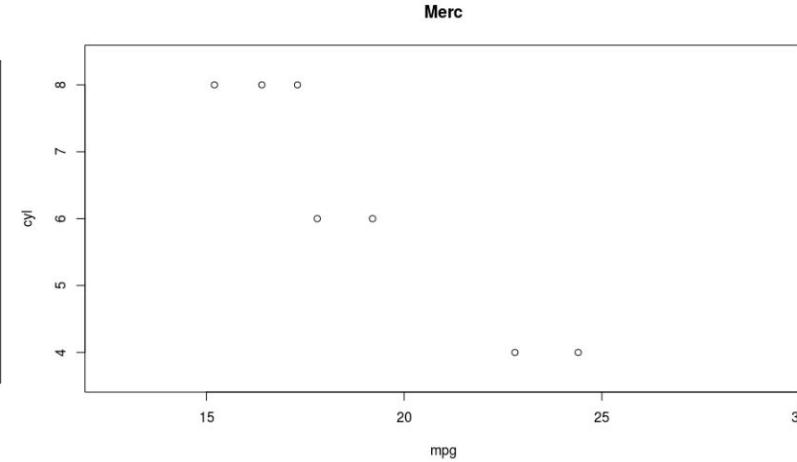
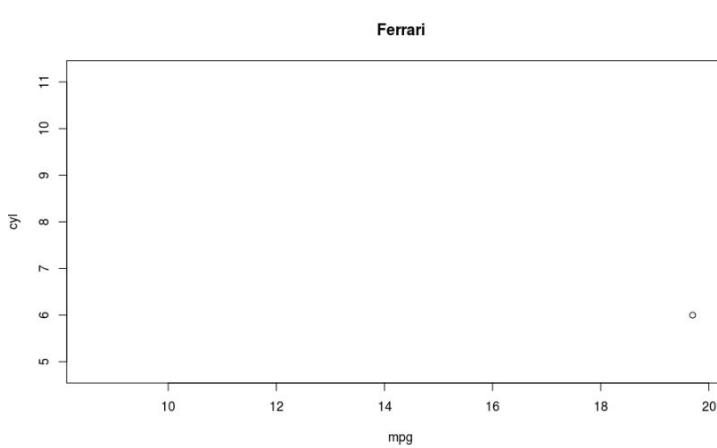
# intersection of multiple data.tables

```
venn.diagram ( dt.mtcars [ , carb ]  
, dt.mtcars [ , gear ]  
, 'carb'  
, 'gear' )
```

Union:	7
carb Only:	4
Intersection:	2
gear Only:	1

# data.table & plot

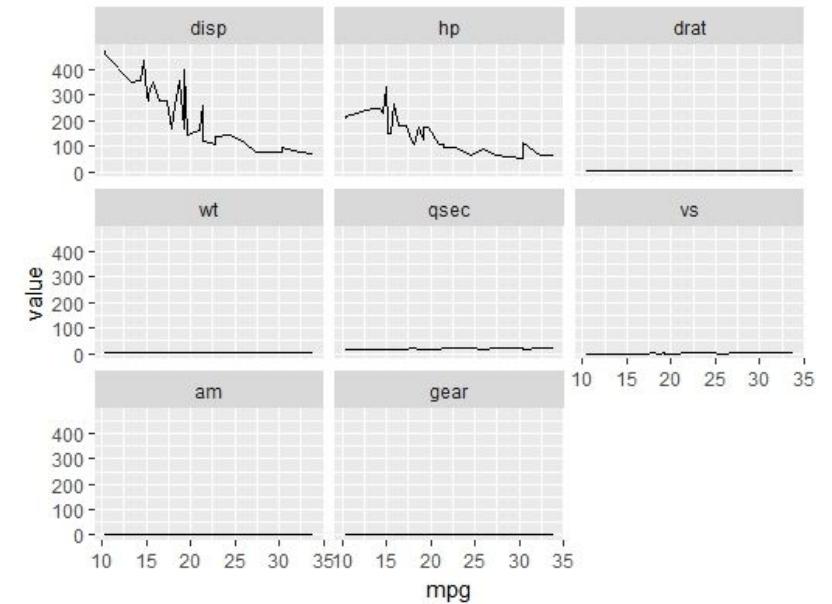
```
dt.mtcars [ , plot ( x = mpg  
                      , y = cyl  
                      , main = manufacturer )  
                     , keyby = manufacturer ]
```



# data.table & ggplot

```
plot.All.XY.by.Z <- function (dt, x, y, z) {
  # numerics only
  dt[, (y):= lapply( .SD, function(x) {as.numeric(as.character(x))}), .SDcols = y]
  dts <- melt(dt, id = c(x,z), measure = y)
  p <- ggplot(dts, aes_string(x = colnames(dt)[x], y = "value", colours = colnames(dt)[z])) +
    geom_line() +
    facet_wrap(~ variable)
  print (p)
}

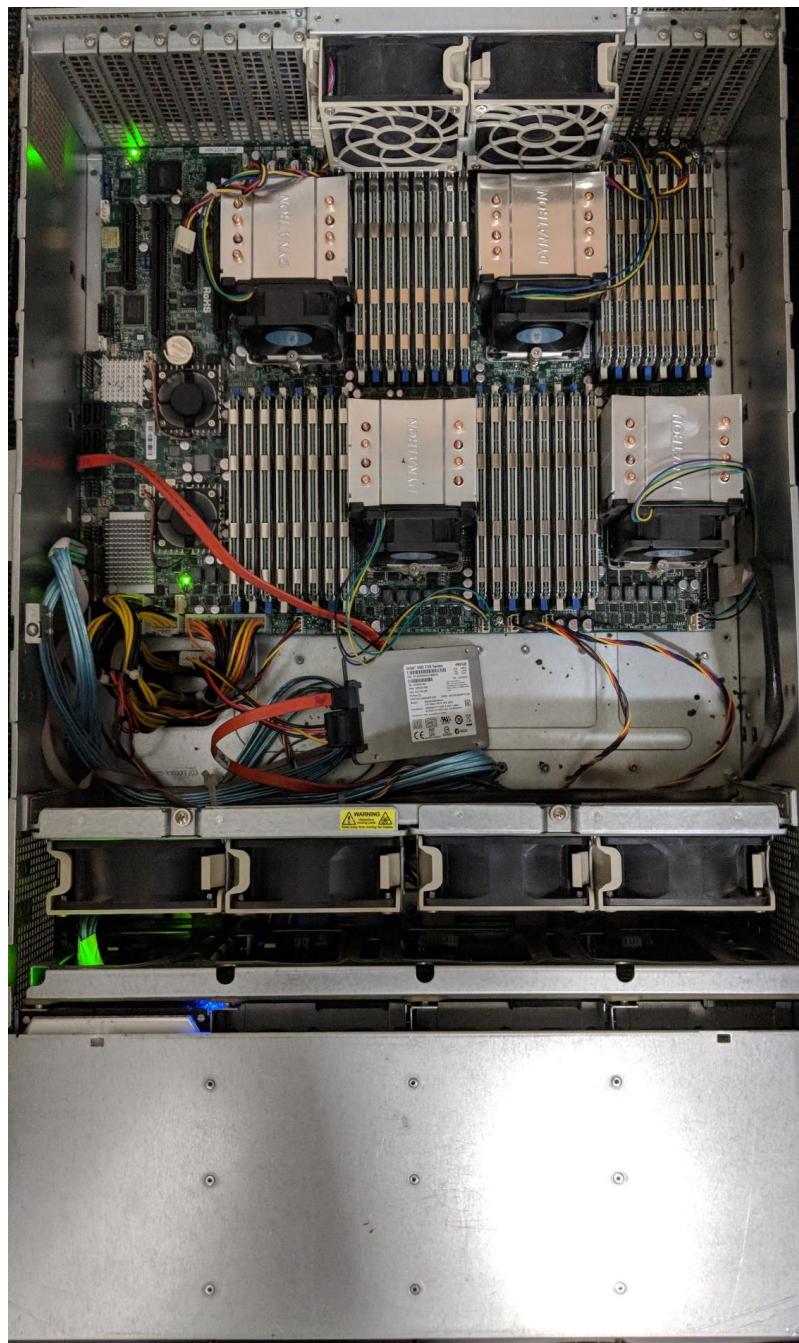
plot.All.XY.by.Z ( dt.mtcars, x=2, y=4:11, z=2)
```



# Where are We Going?

- I. why `data.table`? fast, concise, integrated
- II. syntax - getting comfortable - `DT [ i, j, by ] [ c ]`
- III. `data.table` and data exploration
- IV. something unexpected
- V. cool next steps

unexpected



unexpected



- 4 CPUs
- 40 Terabytes of disk - RAID6
- 1 Terabyte of RAM
- 6k cores

Out of Pocket Cost \$21k

# Where are We Going?

- I. why data.table? fast, concise, integrated
- II. syntax - getting comfortable - DT [ i, j, by ] [ c ]
- III. data.table and data exploration
- IV. something unexpected (economics)
- V. cool next steps

# 43x Performance Increase Based on GPU

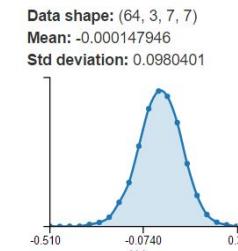
- Transition Migration Matrices
- Time to Execute
  - Pre-migration ~2.5 days
  - Post-migration < 1s

rows	Cores		
	32	6k	
R (sec)	Python (sec)	Python (sec)	
100k	0.7	1.6	0.06
500k	2.8	9.2	0.3
1 MM	6.4	17.5	0.7
2 MM	12.1	36.6	0.9
5 MM	26.3	84.3	2.1
10 MM	56.2	172.5	4.1

# computer vision & retail products

## Test Image

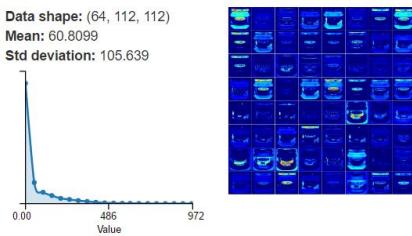
**conv1/7x7\_s2**  
Weights (Convolution layer)  
9,472 learned parameters



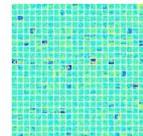
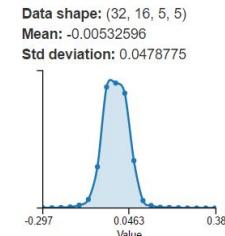
**Predictions**

Beech-Nut 0	100.0%
Little Duck 18	0.0%
Earth's Best 3	0.0%
Hot Kid 16	0.0%
Gerber 9	0.0%

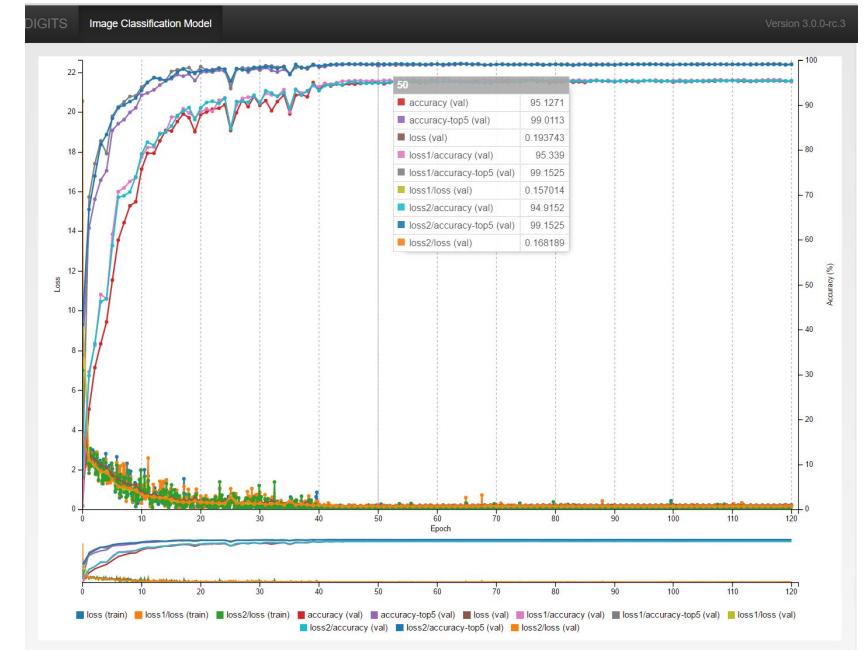
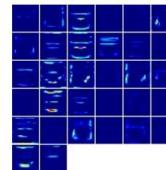
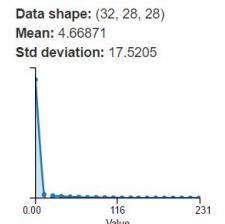
**conv1/7x7\_s2**  
Activation



**inception\_3a/5x5**  
Weights (Convolution layer)  
12,832 learned parameters



**inception\_3a/5x5**  
Activation



@ 50 Epochs  
95% Accuracy

A medium shot of a man with dark, wavy hair, wearing a dark suit jacket, a light-colored striped shirt, and a dark tie with a subtle pattern. He is looking slightly to his left with a neutral expression. The background is blurred, showing autumn foliage with orange and yellow leaves.

T. HANKS