

useR! 2006

Data Profiling with R



*Discovering Data Quality Issues
as Early as Possible*



June 2006

Loyalty Matrix, Inc.
580 Market Street
Suite 600
San Francisco, CA 94104
(415) 296-1141
<http://www.loyaltymatrix.com>

Agenda

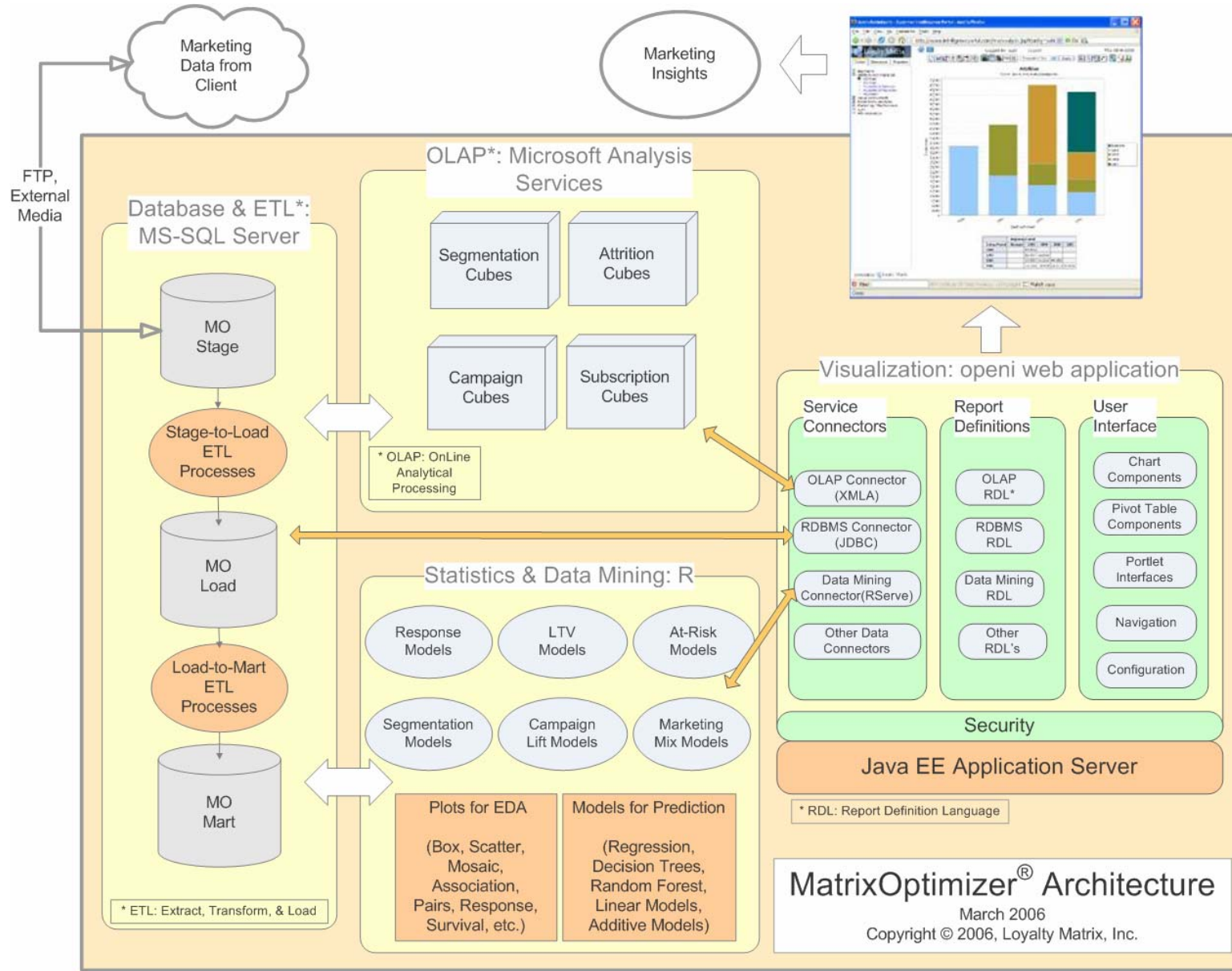
- Background & Problem Statement
- High Level Design
- R & SQL Integration Examples
- Grid Graphics for Summary Panels
- Some Real World Data Profiles
- Demo Profile Run (After Break)

Loyalty Matrix Background

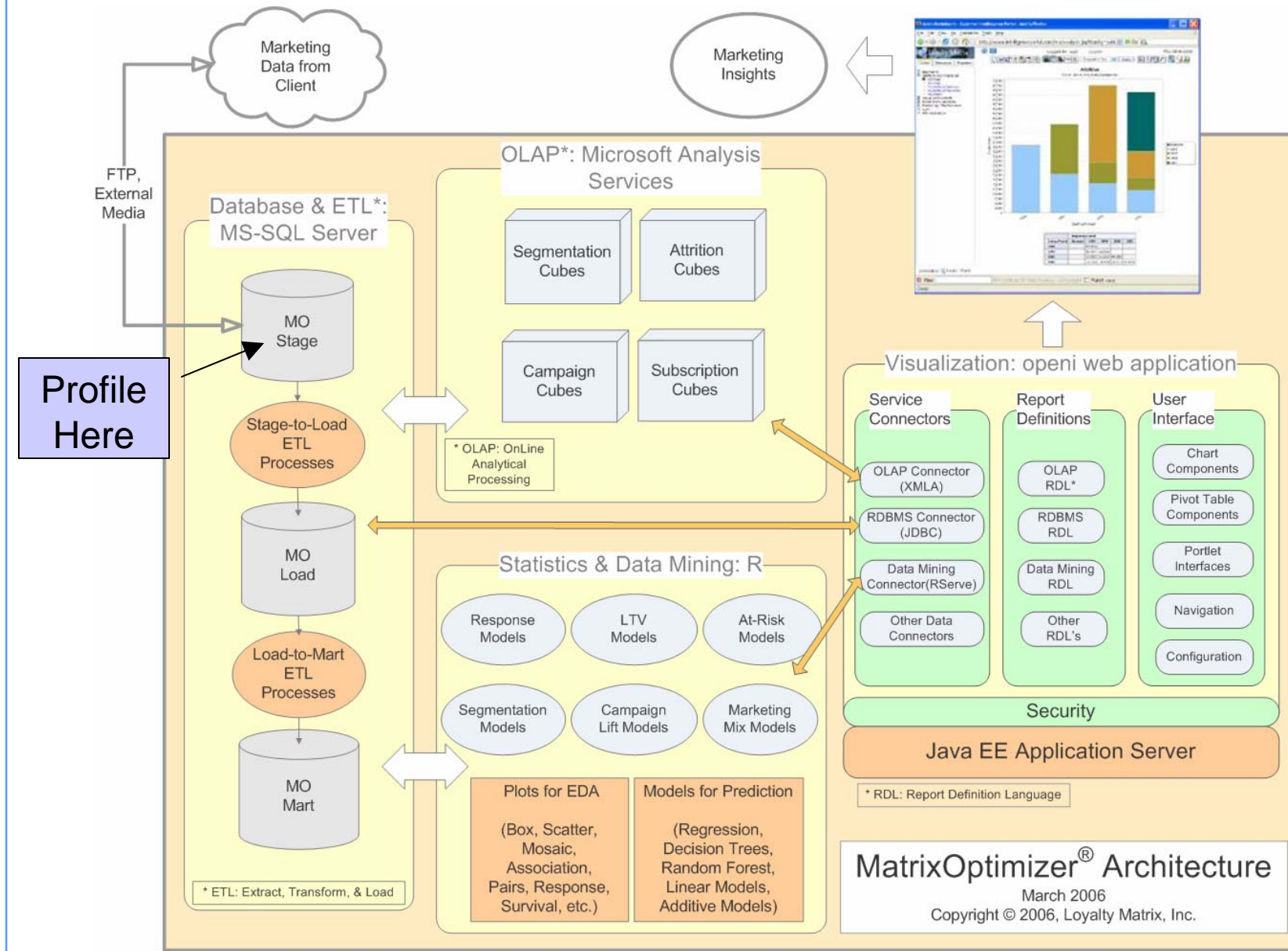
- 15-person San Francisco firm with an offshore team in Nepal
- Provide customer data analytics to optimize direct marketing resources
- OnDemand platform MatrixOptimizer® (version 3.1)
- Over 20 engagements with Fortune 500 clients
- Deliver actionable marketing actions based on real customer behavior



MatrixOptimizer®: Architecture Overview



MatrixOptimizer®: Profile Point

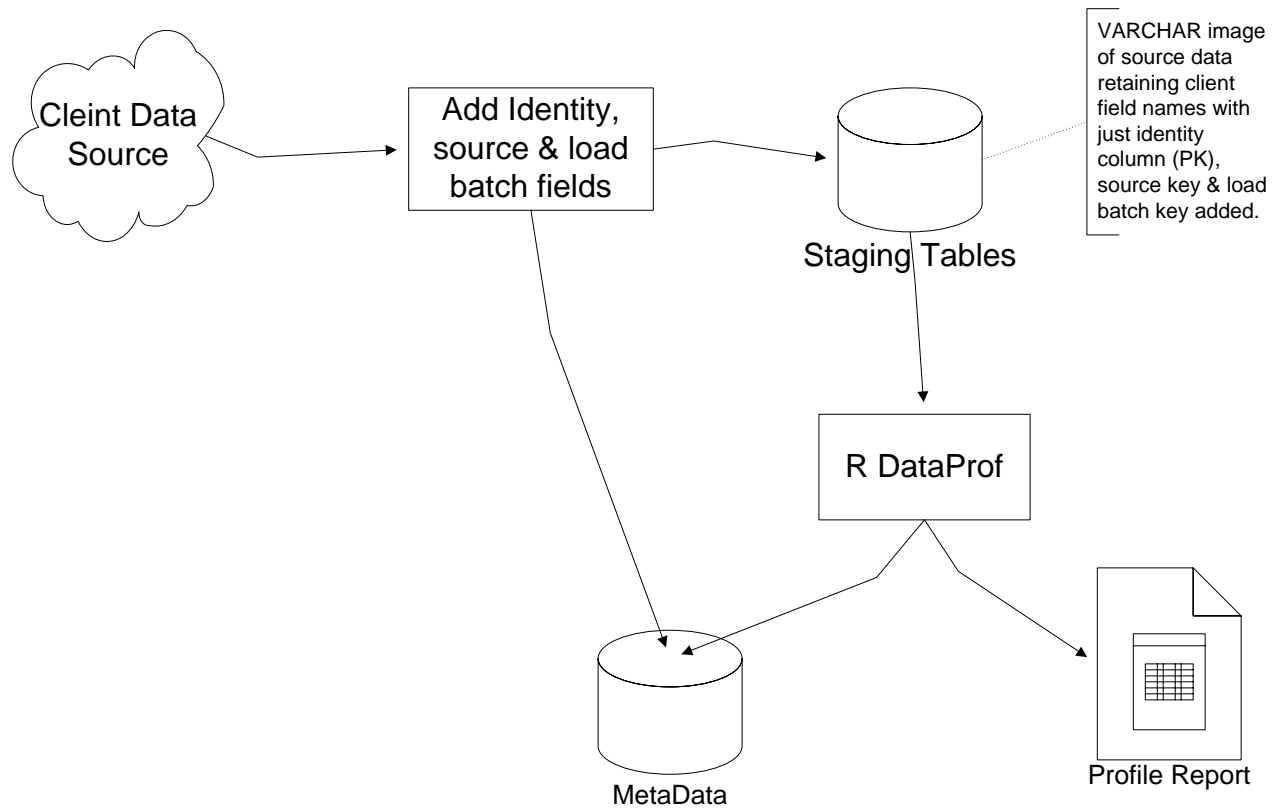


Ideal Data Profiler Requirements

- Require minimum input from analyst to run
- Intuitive output for DB pros – share results with client DBA.
- Column profiling (each column treated independently)
 - Simple statistics & plots
 - Patterns, exceptions & common domain detection
- Dependency profiling for intra-table dependencies
- Redundancy profiling for between table keys, overlaps
- Easy to use reports for analysts & clients
- Save findings in accessible data structure for subsequent use
- Low Cost or “Free”

- See: *Data Quality – The Accuracy Dimension* by Jack E. Olson

Profiling Data Flow



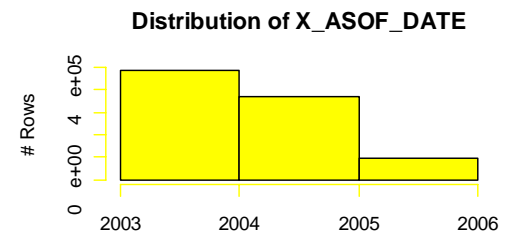
Profiler Output Panel

Header: Database details for field

Summary Counts & %'s
Empty, Numeric & Date only for
character strings

AMA_Stage . RECEIVABLE_TXN . X_ASOF_DATE							19	varchar(8000)
Rows	Nulls	Distinct	Empty	Numeric	Date			
# 3,861,249	2,908,244	28,564	0	0	953,005			
% 100.00	75.32	0.74	0.00	0.00	100.00			
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.			
2003-01-14	2003-08-12	2003-12-29	2004-02-12	2004-09-08	2005-10-17			

Head: NA|NA|NA|NA|NA|NA



Summary Stats if
numeric or date

Appropriate
plot type

Footer: Notes about field

Saved as .wmf in Plots sub-folder.

High Level Design

- User picks database to profile
- User selects specific tables or <all>
- For all selected tables:
 - For all columns in table:
 - Get basic stats
 - Select most appropriate plot type
 - Get data for plot
 - Write panel text & plot to .wmf

R & SQL Integration (1)

- Let SQL do the heavy lifting & minimize data sent to R
- RODBC also reads tables & columns:

```
odbcTables(cODBC)
lTables <- odbcFetchRows(cODBC)  ## fetch the list of tables
lTableNames <- lTables$data[[3]][lTables$data[[4]] == "TABLE" &
                                     lTables$data[[3]] != "dtproperties"]
```

```
odbcColumns(cODBC, ithTable)
lColumns <- odbcFetchRows(cODBC)
Columns <- data.frame(Name = lColumns$data[[4]],
                      Type = lColumns$data[[6]],
                      Width = lColumns$data[[8]])
```

- Get Number of Nulls

```
NumNull <- as.integer(sqlQuery(cODBC,
                               paste("SELECT COUNT(*) FROM ", ithTable,
                                     " WHERE [", ColName, "] IS NULL",
                                     sep = ")))
```

R & SQL Integration (2)

- Get Number of Distincts

```
NumDistinct <- as.integer(sqlQuery(cODBC,  
                                paste("SELECT COUNT(DISTINCT[",  
                                      ColName, "]) FROM ", ithTable,  
                                      sep = ")))
```

- Get Number of Empties

```
NumEmpty <- as.integer(sqlQuery(cODBC,  
                               paste("SELECT COUNT(*) FROM ", ithTable,  
                                     " WHERE LEN(LTRIM(RTRIM([",  
                                     ColName, "])))) = 0", sep = ")))
```

- Get Number of Numerics

```
NumNumeric <- as.integer(sqlQuery(cODBC,  
                                 paste("SELECT SUM(ISNUMERIC([", ColName,  
                                       "])) FROM ", ithTable, sep = ")))
```

- Get Number of Dates

```
NumDate <- as.integer(sqlQuery(cODBC,  
                              paste("SELECT SUM(ISDATE([", ColName,  
                                      "])) FROM ", ithTable, sep = ")))
```

R & SQL Integration (3)

- Look for reasonable plot type & pull plot data:

```
## Come up with a reasonable plot type based on data types, coverage, etc.
PlotType <- ""

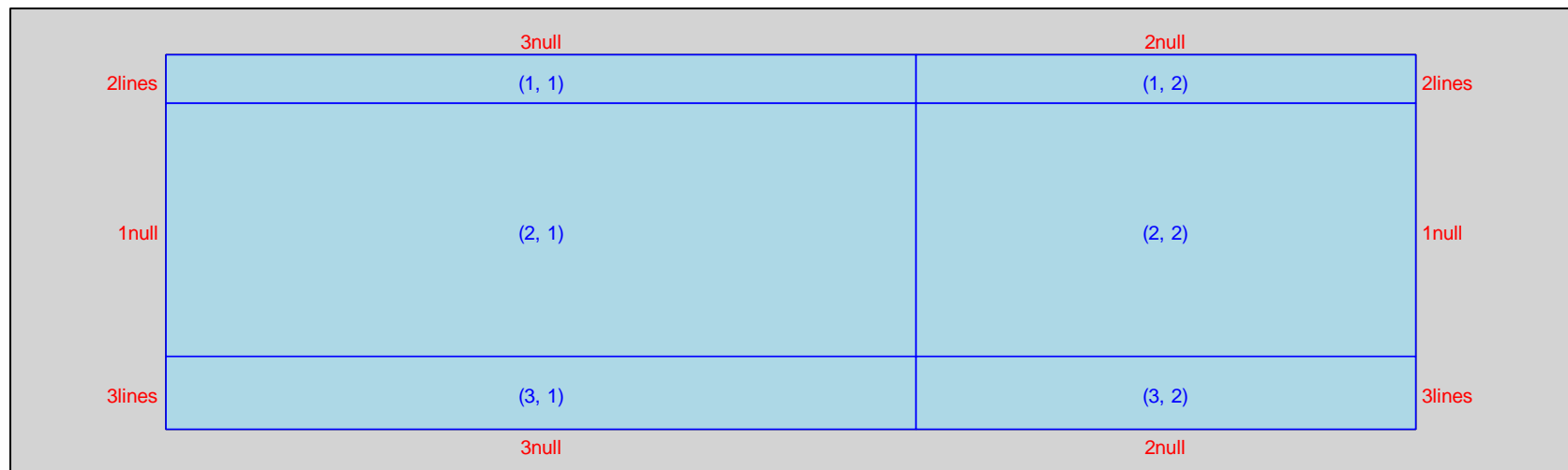
## when NumDistinct a small number of categories
if (PlotType == "" & NumDistinct <= 10) {
  PlotType <- "Category"
  PlotValues <- sqlQuery(cODBC, paste("SELECT [", ColName,
    "] ColValue, COUNT(*) NumRows FROM ", ithTable,
    " GROUP BY [", ColName, "] ORDER BY COUNT(*)",
    sep = ""))
}
```

- And so on for
 - Numbers or strings that mostly look like numbers
 - Dates or strings that mostly look like dates
 - Large number of categories
- Setting up for plot function
 - PlotType
 - PlotValues

Grid Graphics Tricks (1)

- Set up panel

```
windows(width = 10.5, height = 3, pointsize = 10)
TopLayout <- grid.layout(nrow = 3, ncol = 2,
                        widths = unit(c(3, 2), c("null", "null")),
                        heights = unit(c(2, 1, 3),
                                       c("lines", "null", "lines")))
#grid.show.layout(TopLayout)      ## <<<<<Debug only
```



Grid Graphics Tricks (2)

- Walk through the viewports starting with header:

```
pushViewport(vpTopLayout)
  grid.rect(gp = gpar(col = "blue", lwd = 3))

pushViewport(viewport(layout.pos.col = 1:2, layout.pos.row = 1))
  grid.rect(gp = gpar(col = "blue", lwd = 2))
  grid.text(paste(ColDesc$DB, ColDesc$Table,
                  ColDesc$Column, sep = " . "),
            x = unit(0.2, "char"), y = unit(0.6, "lines"),
            just = "left", gp = gpar(col="black", fontsize=18))
  grid.text(ColDesc$ColSeqNum,
            x = unit(0.8, "npc"), y = unit(0.6, "lines"),
            just = "right", gp = gpar(col="black", fontsize=18))
  grid.text(paste(ColDesc$ColType, "(", ColDesc$ColWidth,
                  ") ", sep = ""),
            x = unit(1, "npc"), y = unit(0.6, "lines"),
            just = "right", gp=gpar(col="black", fontsize=18))
popViewport()
```

Grid Graphics Tricks (3)

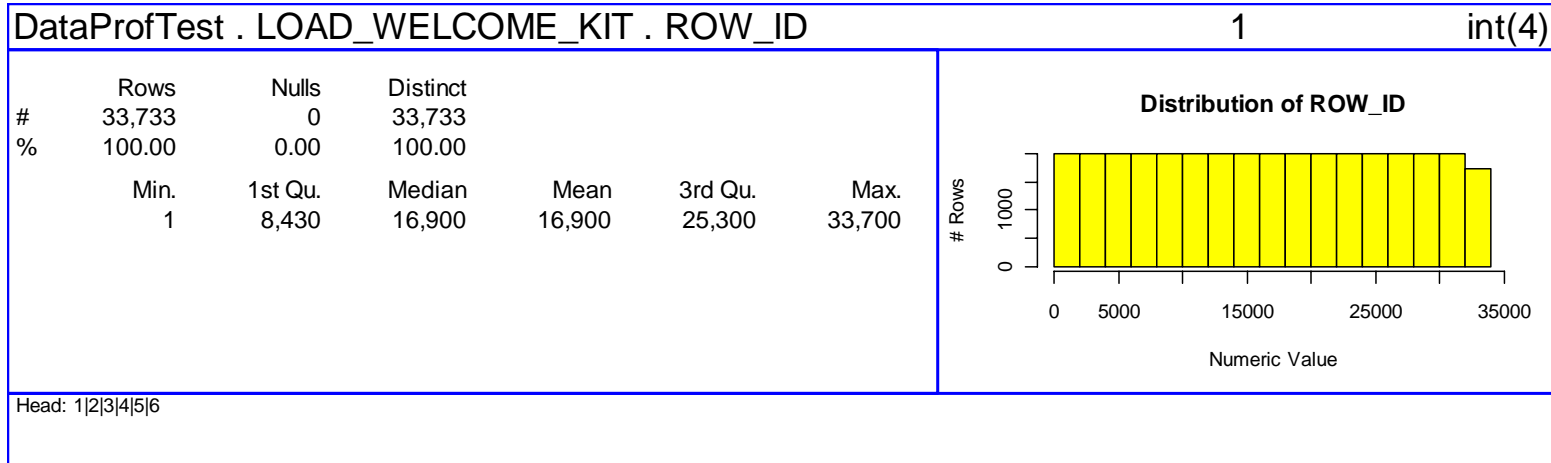
- Only tricky bit is allowing base graphics to do it's thing in plot area:

```
## Plot
pushViewport(viewport(layout.pos.col = 2, layout.pos.row = 2))
op <- par(no.readonly = TRUE) ## around all of plot options below
par(fig = gridFIG(), new = TRUE)
par(mfg = c(1, 1))
# a Category plot
if (ColDesc$ColPlot == "Category") {
  par(mar = c(4.5, 10, 1.8, 2) + 0.1)
  pPlot <- barplot(PlotValues$NumRows,
                  names.arg = as.character(PlotValues$ColValue),
                  horiz = TRUE, las = 1, col = "yellow",
                  main = paste("Categories in", ColDesc$Column),
                  ylab = NULL, xlab = "# Rows")
}

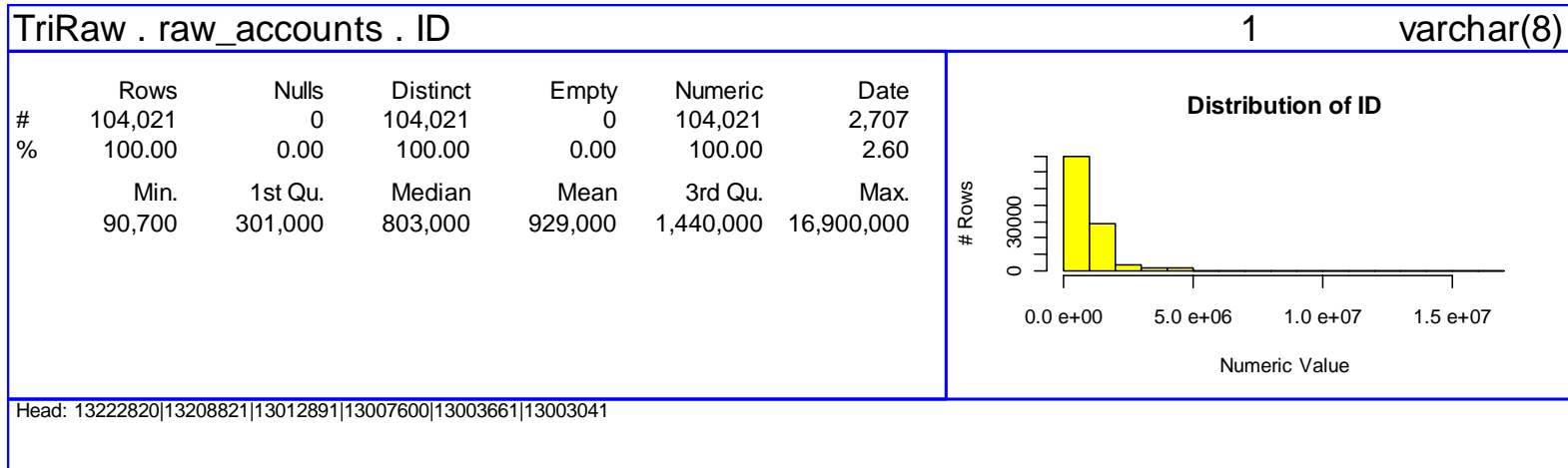
# etc for NumHist, CharHist, ...
```

Concluding Examples of Profiler Runs (1)

- A Surrogate Key

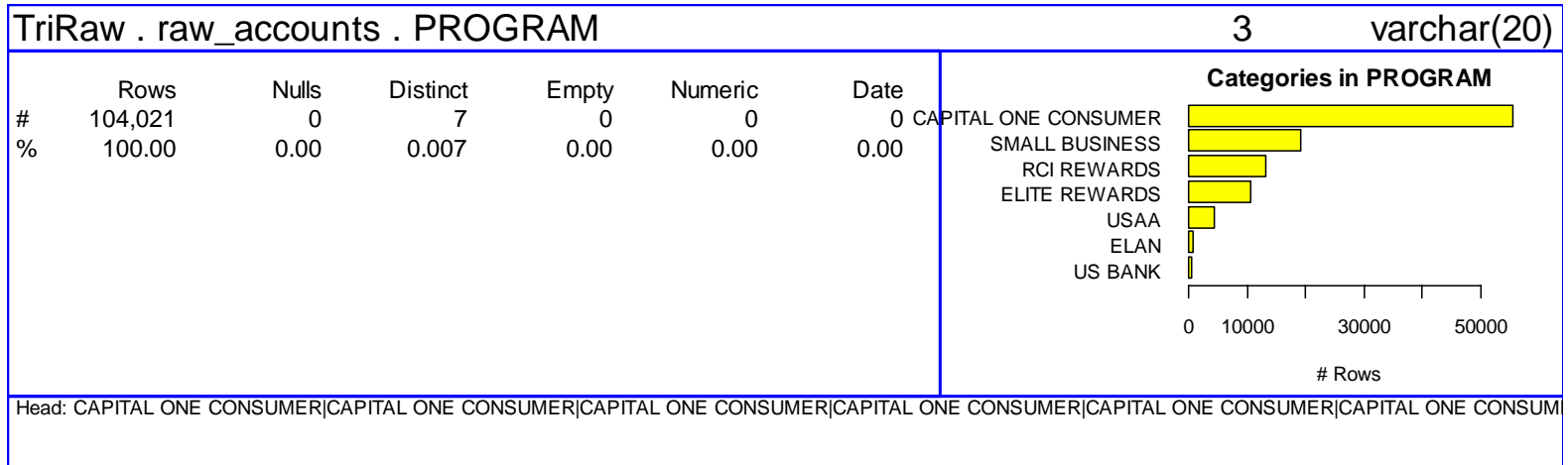


- Probable Business Key

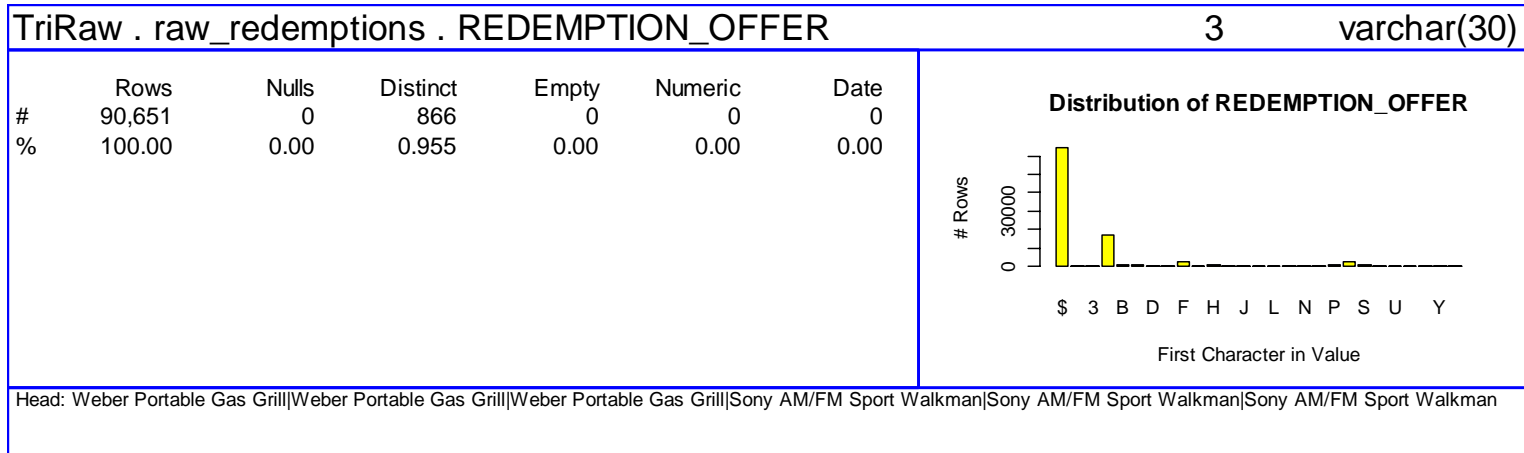


Concluding Examples of Profiler Runs (2)

- A Few Categories

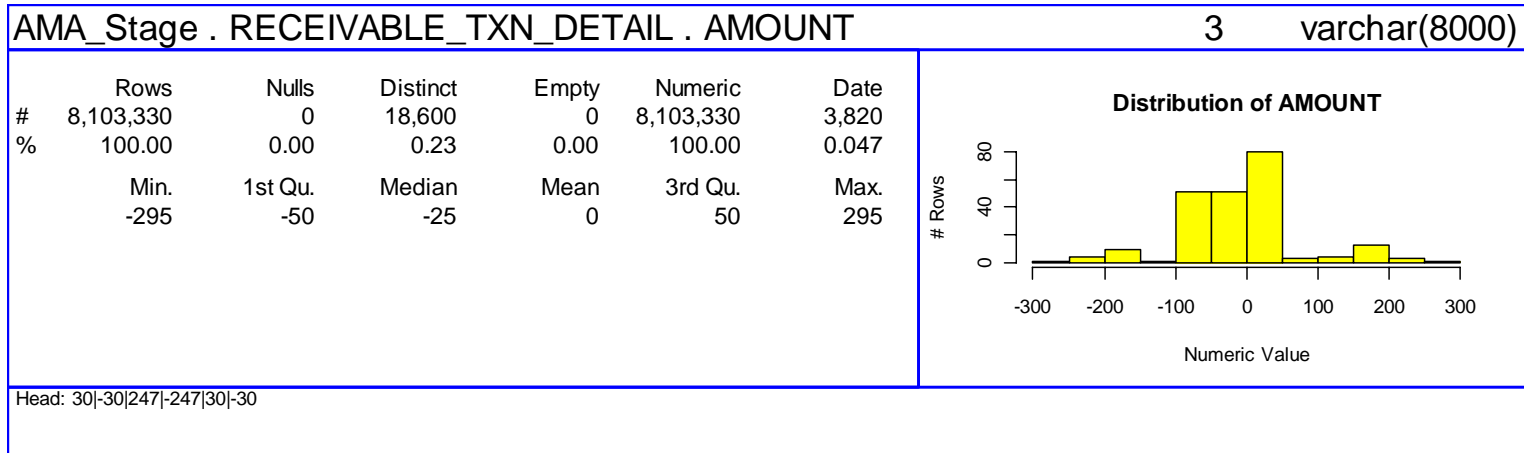


- Many Categories

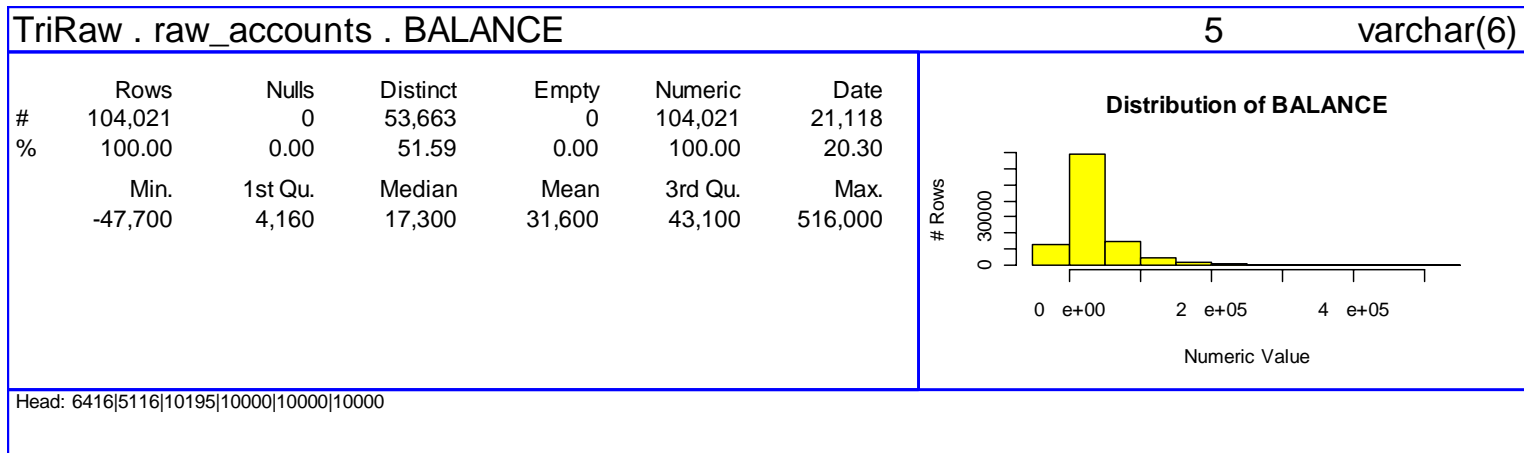


Concluding Examples of Profiler Runs (3)

- Numeric Value

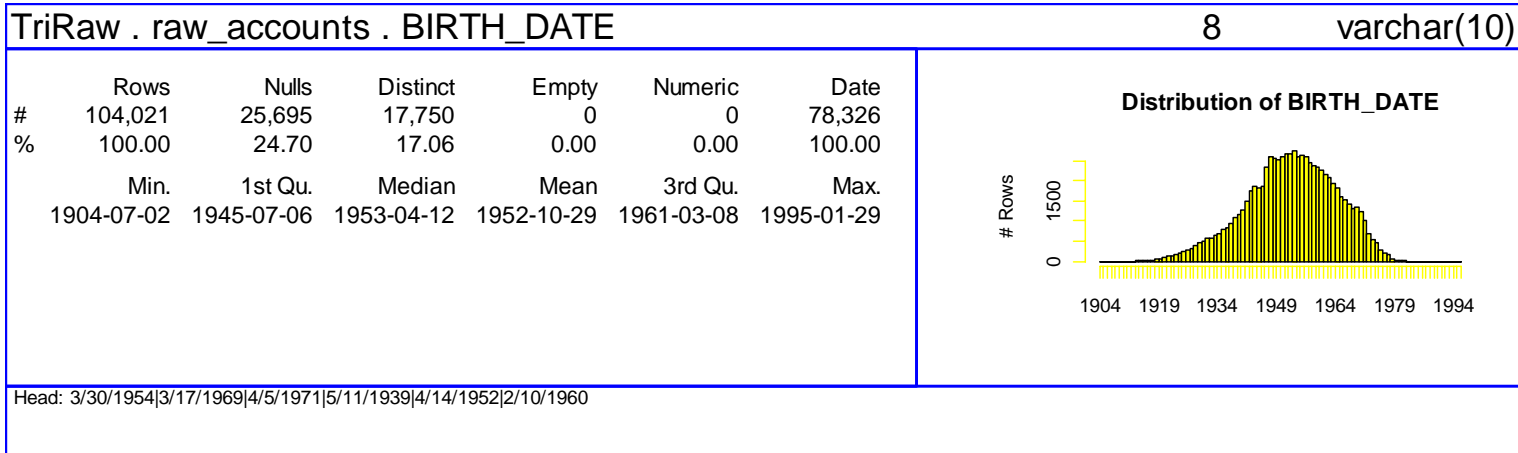


- Another Numeric Value

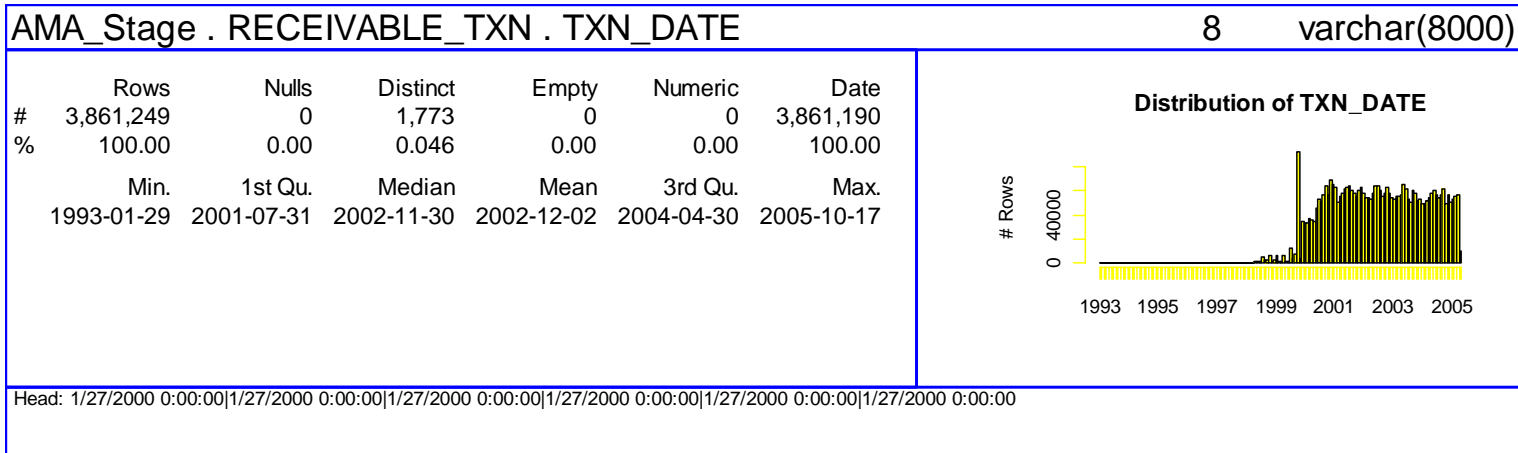


Concluding Examples of Profiler Runs (4)

- Reasonable Dates

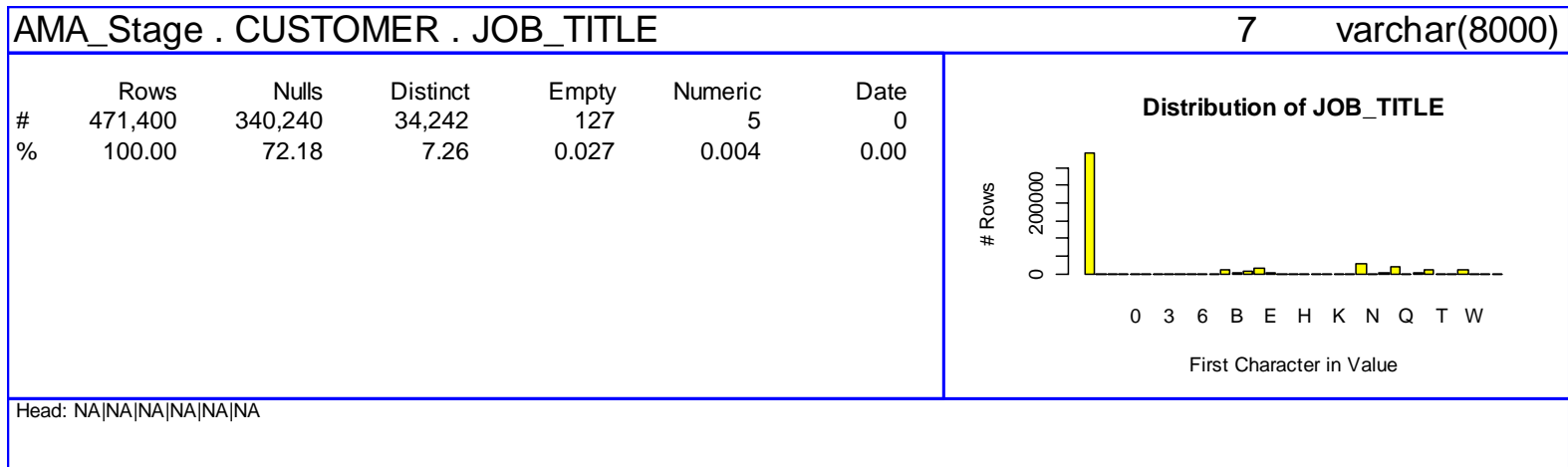


- Unusual Dates

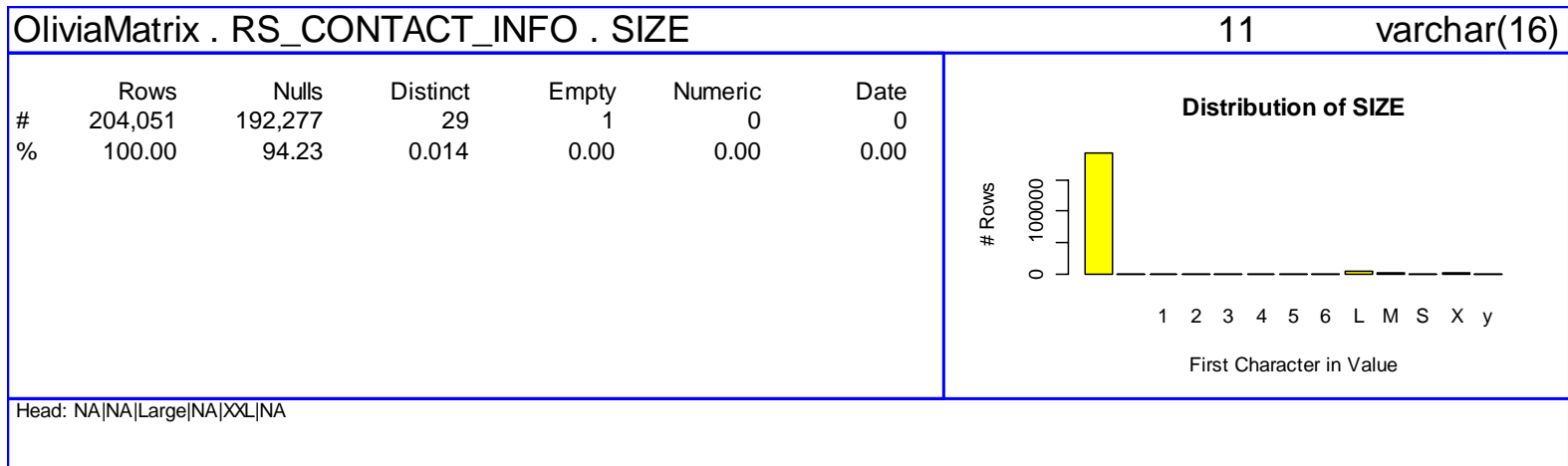


Concluding Examples of Profiler Runs (5)

- Customer Job Title not too useful



- T-Shirt Size also not reliable



Conclusion

- Even Current Version Useful in Production
 - Just column details picks up data quality problems
- Next To Do
 - Add DBMS interface layer & support MySQL, etc.
 - Enumerate patterns like 99999-9999, 9* A* A*, etc.
 - Add Hints for common domains
 - Metadata back to database
 - Dependency
 - Redundancy

Questions? Comments?



- Email JPorzak@loyaltymatrix.com
- Call 415-296-1141
- Visit <http://www.loyaltymatrix.com>
- Come by at:
580 Market Street, Suite 600
San Francisco, CA 94104