Analyzing Direct Marketing Data with R

Liang Wei
Brendan Kitts
Lucid Commerce Inc.
Outline

• Who we (Lucid Commerce) are
  – Business problems
• Why R + SQL Server + SQL Server Reporting Service (SSRS)
  – Analyze data
  – Create and publish analytics product
• Case study:
  – Create a boxplot and publish the tool
  – Create a regression model using R and deploy it
• Summary
About Lucid Commerce

- **About Lucid Commerce**
  - Direct response television (DRTV)
    - “advertising that asks consumers to respond directly to the company --- usually either by calling an 800 number or by visiting a web site”
    - One example of DR Product: credit card application
  - Business problems
    - Lucid’s DRTV optimization
      - Customer targeting, station targeting, TV program targeting
    - Cross channel attribution
      - TV, radio and web.
      - How we can attribute web purchases to an ad on TV

- **Data**
  - Lucid provides data driven business decisions
  - About 3 million direct marketing purchasers as our panel
  - About 400 demographics variables appended to customers
  - Purchase behaviors
Why We Choose R

• Lucid uses Microsoft SQL Server for data management
  – One of the most popular database engines
  – SQL server lacks powerful data visualization
  – SQL Server database engine lacks interactive support for powerful exploration of data using advanced analytics

• R works very well with SQL Server
  – RODBC

• R has richer support for advanced analytics (that we need)
  – Clustering, classification, forecast

• R is powerful in generating graphics
  – Reporting & data visualization

• Quick to publish analytics results and implement R scripts to real-time products
R Analytics Workflow

SQL Server Report Service

Connect to SQL Server
Calls Stored Procedure

SQL Server Database Engine

Sends results to SSRS

SSRS Web Page

Rendering new page

User input

Send analytics results to database

R Analytics Engine

Calls R script
Pass data

Proceed
Case Study

• Media buyer’s questions
  – How is my campaign performing?
  – What is the best station to place an ad for my client?
  – What is the best time?
    • Weekday or Weekend
    • Morning or night
  – Local broadcast or national?
  – Visualize variations on different factors
    • boxplot

www.newamerica.net
# Case Study – Data Visualization

- **Example data**

<table>
<thead>
<tr>
<th>ProjectKey</th>
<th>SpotNaturalKey</th>
<th>RunDateTime</th>
<th>Daypart</th>
<th>DayOfWeek</th>
<th>Station</th>
<th>Channel</th>
<th>Revenue</th>
<th>MER</th>
<th>Orders</th>
<th>MC</th>
<th>SR</th>
<th>DR</th>
<th>WR</th>
</tr>
</thead>
<tbody>
<tr>
<td>10021</td>
<td>100210103</td>
<td>02/01/2010 09:30</td>
<td>16pm - 19pm</td>
<td>2</td>
<td>Weekday</td>
<td>WMBC</td>
<td>Broadcast</td>
<td>170.84</td>
<td>1.658</td>
<td>2</td>
<td>175,344,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210104</td>
<td>02/01/2010 07:00</td>
<td>2am - 5am</td>
<td>2</td>
<td>Weekday</td>
<td>DOC</td>
<td>Cable</td>
<td>119.94</td>
<td>0.62311,7647</td>
<td>1</td>
<td>165,693,870,813,153</td>
<td>287,589,412,728,238</td>
<td>252,545,611,532,515</td>
</tr>
<tr>
<td>10021</td>
<td>100210105</td>
<td>02/01/2010 08:00</td>
<td>4pm - 7pm</td>
<td>2</td>
<td>Weekday</td>
<td>WASA</td>
<td>Broadcast</td>
<td>170.84</td>
<td>1.3261</td>
<td>2</td>
<td>175,934,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210106</td>
<td>02/01/2010 22:00</td>
<td>7am - 10am</td>
<td>2</td>
<td>Weekday</td>
<td>WGCB</td>
<td>Broadcast</td>
<td>622.52</td>
<td>3.92396,022,81</td>
<td>4</td>
<td>165,229,170,523,937</td>
<td>336,322,923,191,611</td>
<td>252,545,611,532,515</td>
</tr>
<tr>
<td>10021</td>
<td>100210107</td>
<td>02/01/2010 10:00</td>
<td>9am - 12am</td>
<td>2</td>
<td>Weekday</td>
<td>DOC</td>
<td>Cable</td>
<td>230.78</td>
<td>2.20542,370,809</td>
<td>3</td>
<td>175,639,903,042,53</td>
<td>283,070,412,728,238</td>
<td>252,545,611,532,515</td>
</tr>
<tr>
<td>10021</td>
<td>100210108</td>
<td>02/01/2010 10:30</td>
<td>11am - 2pm</td>
<td>2</td>
<td>Weekday</td>
<td>WMBC</td>
<td>Broadcast</td>
<td>230.78</td>
<td>1.68741,7213</td>
<td>2</td>
<td>175,344,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210109</td>
<td>02/01/2010 11:30</td>
<td>13am - 4pm</td>
<td>2</td>
<td>Weekday</td>
<td>WMBC</td>
<td>Broadcast</td>
<td>240.88</td>
<td>1.20154,010,7</td>
<td>2</td>
<td>175,344,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210110</td>
<td>02/01/2010 12:30</td>
<td>15am - 6pm</td>
<td>2</td>
<td>Weekday</td>
<td>WMBC</td>
<td>Broadcast</td>
<td>240.88</td>
<td>1.20154,010,7</td>
<td>2</td>
<td>175,344,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210111</td>
<td>02/01/2010 13:30</td>
<td>2pm - 5pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210112</td>
<td>02/01/2010 14:30</td>
<td>3pm - 6pm</td>
<td>2</td>
<td>Weekday</td>
<td>WMBC</td>
<td>Broadcast</td>
<td>240.88</td>
<td>1.20154,010,7</td>
<td>2</td>
<td>175,344,431,193,762</td>
<td>208,763,851,394,956</td>
<td>561,956,870,760,672</td>
</tr>
<tr>
<td>10021</td>
<td>100210113</td>
<td>02/01/2010 15:30</td>
<td>4pm - 7pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210114</td>
<td>02/01/2010 16:30</td>
<td>5pm - 8pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210115</td>
<td>02/01/2010 17:30</td>
<td>6pm - 9pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210116</td>
<td>02/01/2010 18:30</td>
<td>7pm - 10pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210117</td>
<td>02/01/2010 19:30</td>
<td>8pm - 11pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
<tr>
<td>10021</td>
<td>100210118</td>
<td>02/01/2010 20:30</td>
<td>9pm - 12pm</td>
<td>2</td>
<td>Weekday</td>
<td>ANGL</td>
<td>Cable</td>
<td>320.72</td>
<td>2.66868,823</td>
<td>3</td>
<td>171,718,406,291,91</td>
<td>224,459,523,555,153</td>
<td>253,937,800,590,59</td>
</tr>
</tbody>
</table>

**Proceed**
Case Study – Data Visualization

• Building SSRS Report

![Image showing a boxplot for a project with user input and database connection points.]
• Stored procedure

  – “A stored procedure is a subroutine available to applications accessing a relational database system” [en.wikipedia.org/wiki/Stored_procedure](http://en.wikipedia.org/wiki/Stored_procedure)

```sql
USE [useR2010]
GO

CREATE PROCEDURE [dbo].[BoxplotProc]
    @ProjectKey INT,
    @StartDate DATETIME,
    @EndDate DATETIME,
    @VarName VARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @cmd varchar(200)
    DECLARE @RImageID BIGINT
    SELECT @RImageID = ABS( CHECKSUM( 1000000000*RAND()) )

    SET @cmd = 'R --no-save --args projectkey=10021 rimageid=584674235 startdate='2010-01-01' enddate='2010-07-11' varName='revenue' <c:\useR2010\boxplot.r

EXEC master..xp_cmdshell @cmd, NO_OUTPUT

SELECT RImage FROM useR2010.dbo.boxplotresult where RImageID = @RImageID
END
```

Compiles a command string

```
R --no-save --args projectkey=10021 rimageid=584674235 startdate='2010-01-01' enddate='2010-07-11' varName='revenue' <c:\useR2010\boxplot.r
```

Execute R script from SQL Server with `xp_cmdshell` operating-system command shell
Case study – Data Visualization

• R script snippet

```r
query <- sprintf("INSERT INTO user2010.dbo.boxplotresult (RImageID, CreatedDateTime, RImage) SELECT $d, GETDATE() as CreatedDateTime, * FROM OPENROWSET(BULK N'%s', SINGLE_BLOB) AS import", RImageID, file.name
)
sqlQuery(odbc.connect, query)
```

This R code inserts the image we created with R (file.name is the image file’s location on my hard disk) to the database. We use SQL Server’s bulk load function to load the image file to the database as VARBINARY type filed(which is defined in the table creation step). *

* The authors would like to thank Don Nebres for his advice on this image loading process.
• Evaluate the arguments sent from command shell
• Creates graph
• Save results to database
  – *sqlSave*: tables, numbers, texts
  – Bulk load function for images

Case study – Data Visualization

- SQL Server Database Engine
- Send data request
- Analyze data
- R analytics engine
- Send analytics results to database
- Evaluate arguments
- Pass data to R
- Calls R script
Case Study – Data Visualization

Boxplot for Project 851

Case Study – Forecasting Model

• Media buyer: how can we quantitatively predict our revenue?

• Forecast problem
  – given a specific airing (e.g. an ad for product A showing on ESPN at 9:00am PST)

• Multiple regression
  \[ y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i \]
Case Study – Forecasting Model

- A real time regression tool

Regression Formula

\[ \text{Revenue} \sim mc + sr \]

This model's RSquare is 0.839108
Case Study – Publishing Forecasting Results

• R script snippet
  – Saving forecasted results

```
sqlSave(odbc.connect, forecast.result, tablename = 'Model10Forecast',
        append=TRUE, rownames=FALSE)
```

The R code inserts the forecasted results data back to the database using `sqlSave` function from the RODBC library:

- `forecast.result` is the table that have our forecasted results,
- `tablename` is what my target table in the database,
- `append` means we are going to add those rows to the database instead of overwriting the table,
- `rownames` is the rownames of the result table, our target table doesn’t have this column so we get rid of it.

See `?sqlSave` for detailed explanation of this function.
Case study – Publishing Forecasting Results

- After a good model is carefully selected, we can deploy it into production which updates daily.
Summary

• Analytics problems solved
  – R provides powerful advanced analytics for analyzing DRTV data
  – Actionable plan for media buy

• Engineering problems solved
  – SQL Server, SSRS provides convenient mechanism in developing analytics products developed by R
  – Publish analytics results to web portal, expose reports to internal team or external clients
Thank You!

• The authors would like to thank Brian Burdick, Dyng Au and Amanda Powter for their great help, advice and discussions which make this work possible

• The authors can be contacted at:
  – lwei at lucidcommerce dot com

• R, SQL sample scripts and related DRTV marketing topics can be found here:
  – http://blog.lucidcommerce.com/