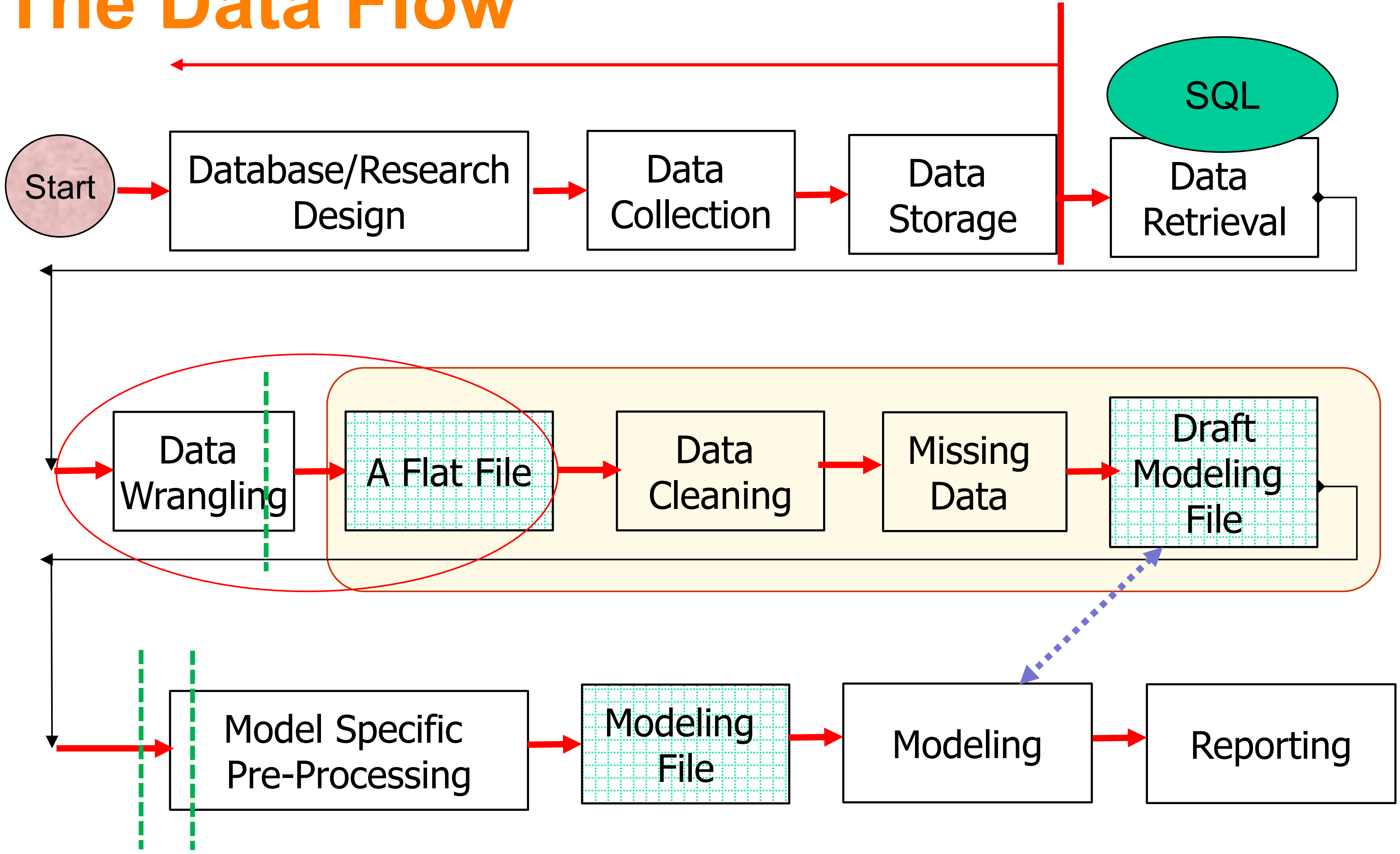
Data Wrangling and Cleaning Steps: 1-5, 6-7, 8-11

HASLAM
COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Session Objectives

- To highlight typical data wrangling problems
- To review cleaning/inspection techniques **1-5, 6-7,8-11**
- To understand the challenges in each step
- To highlight a few R functions to assist in these steps

The Data Flow

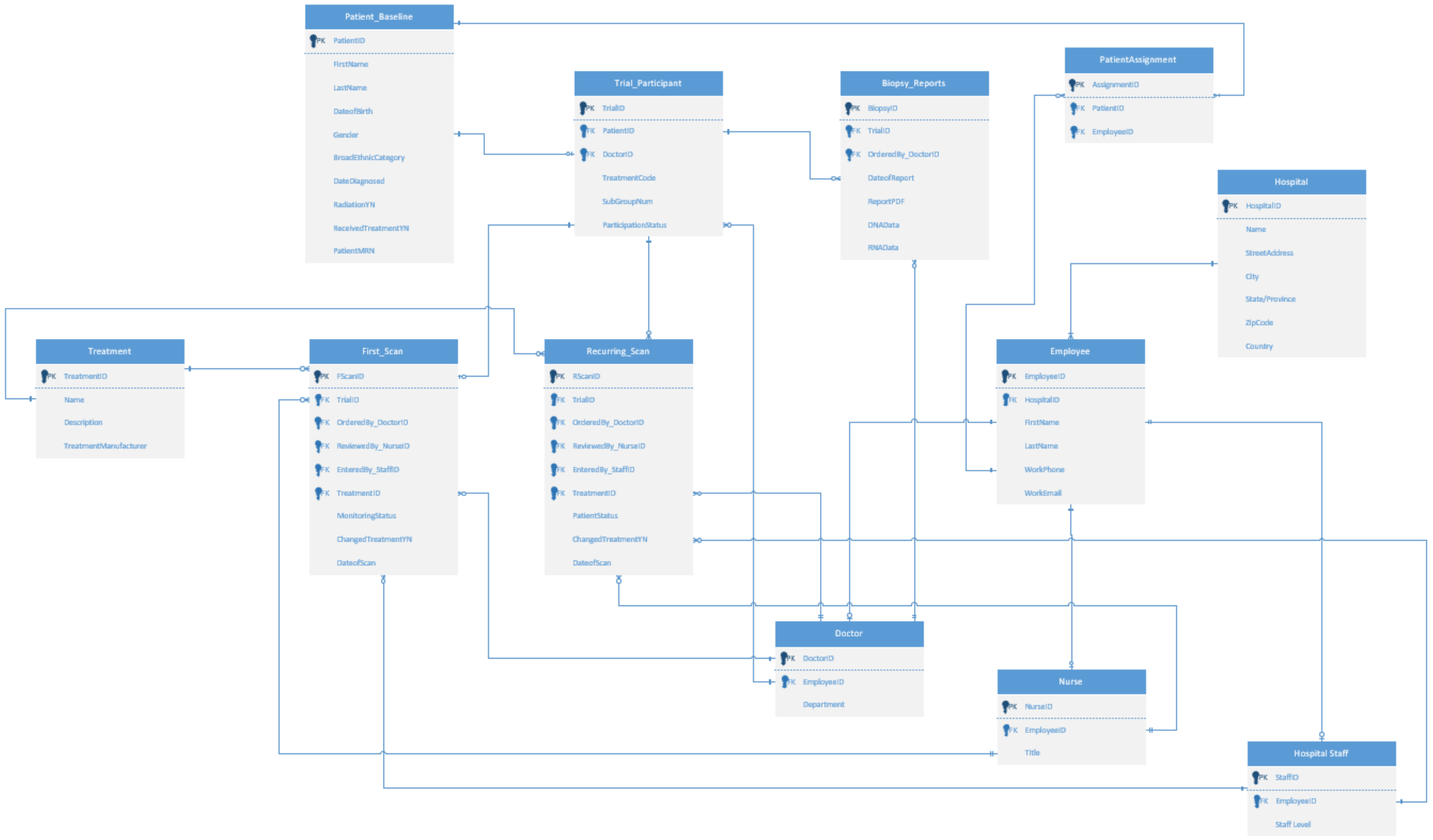


What is Data Wrangling?

- Hell if I know … but,
- For this class it will be defined as;
  - Taking “unTidy” data and making it Tidy (mainly rows and columns in this class)
  - Combining relational files into one flat file

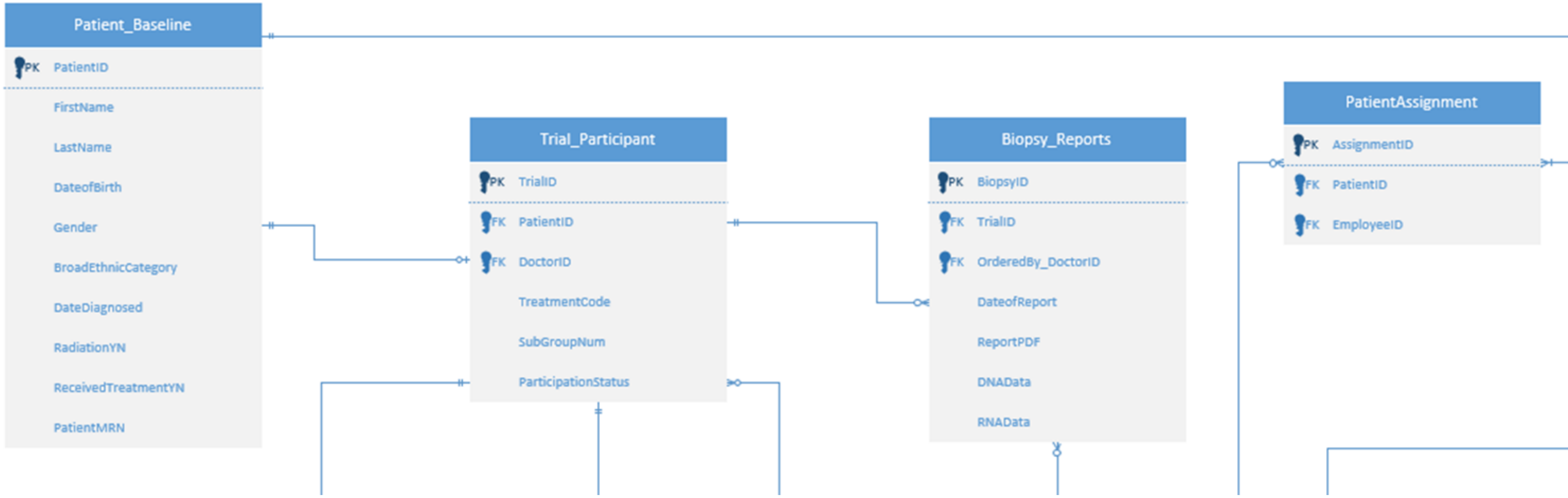
For key R functions to assist in Wrangling, see Tidyverse and dplyr slides and videos.

Relational Database

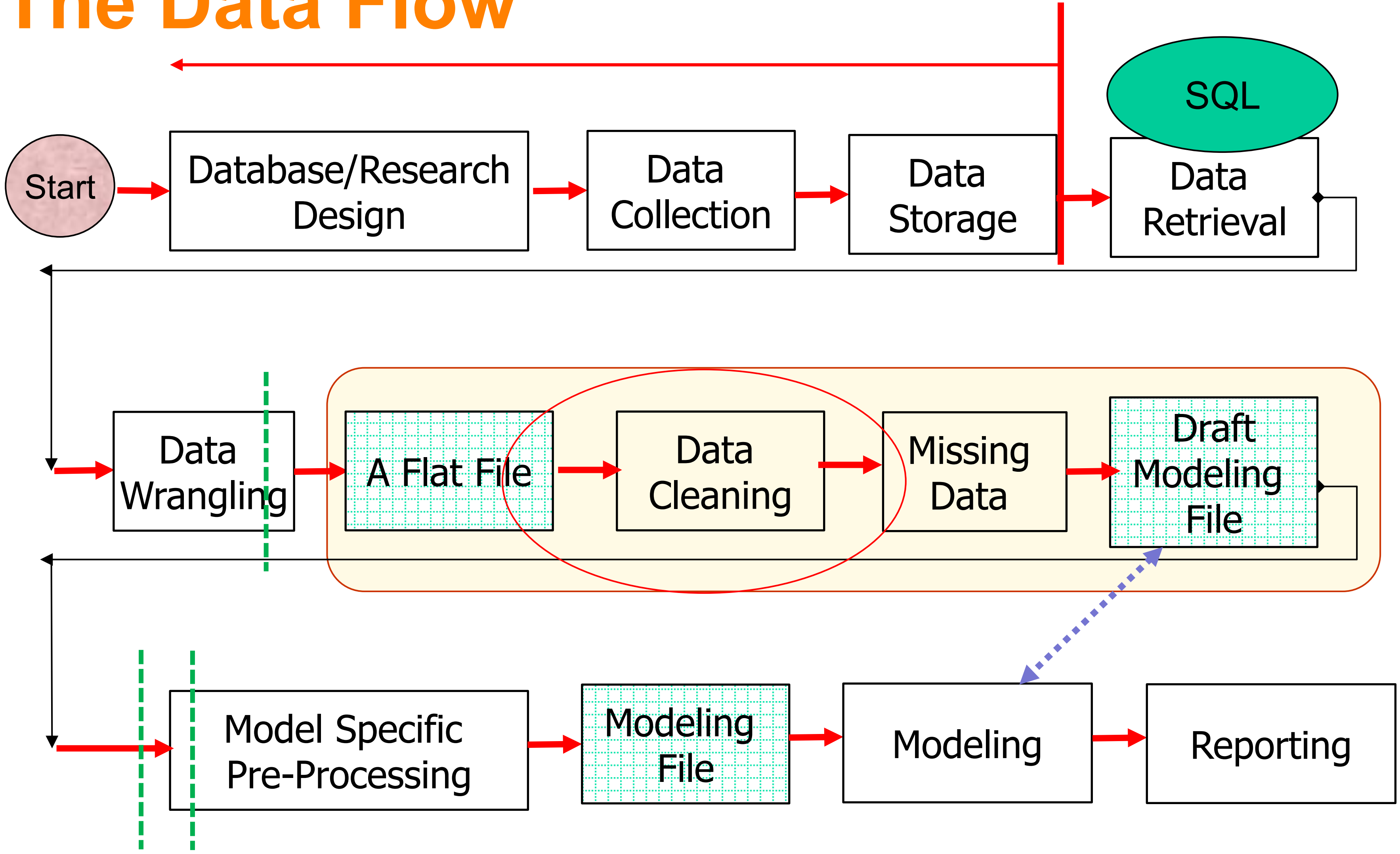


HASLAM
COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Notional Detail



The Data Flow



HASLAM COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Data Cleaning Steps

1. Open data in your software of choice
2. Review variables for common sense based on SME knowledge
3. Review how the software coded the variables (nominal, continuous)
4. Perform data integrity/validation checks (misspelled levels, bogus values, combine levels which represent the same thing, etc.)
5. Handle dates (extract relevant information, e.g. day of week, hour of day)
6. Handle categorical variables - keep as is, combine rare levels, combine similar levels
7. Handle zero-variance predictors, i.e., columns that contain the same number throughout
8. Handle near zero-variance predictors, i.e., columns that contain very little variety in values or who mostly contain a single value (very low information density)
9. Eliminate redundant columns and columns that are weighted sums of others.
10. Search for outliers and initial search for missing values
11. Sanity check using Decision Tree (1 to 2 splits)

HASLAM COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Step 1: Open data with your software(s) of choice

- We'll be mainly using R
- I like to use Excel and JMP for some activities
  - Easier (from my experience) to make initial visual exploration of the data
- Ultimately, we go to R (or other code) to create replicable work flow
  - Some univariate analyses can be done for understanding
    - Histograms
    - Bar Charts
    - Summary Statistics
    - Time Series
  - Some multivariate analysis
    - Correlation

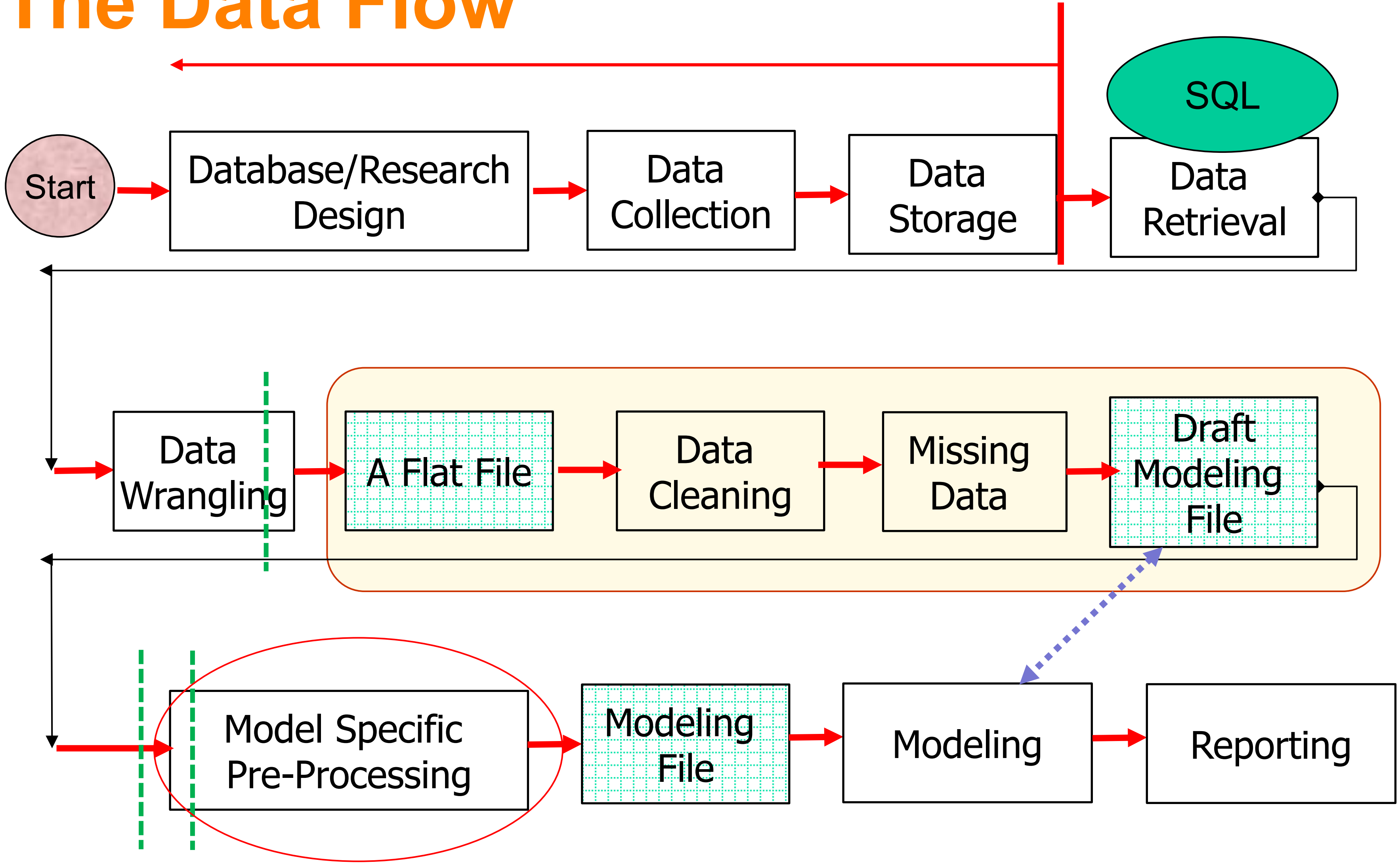
NOTE: some of this work belongs in Model Specific Processing

Step 1: Open data with your software(s) of choice (cont.)

- Some multivariate analysis
  - Correlation
  - Principal Components
  - Scatter plots
  - Cluster

NOTE: Most of this work belongs in Model Specific Processing

The Data Flow



HASLAM COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Step 2: Review with SMEs

- Do NOT be a lone wolf
- Review where/how you got the data
- Use/obtain Data Dictionary and discuss with SMEs for understand
- What do the SME's think about:
  - What a "good" model would accomplish
  - Amount/reason for missing data
  - Key variables

Step 3: Review Variable types

- Different software will make different assumptions
- You at least have to get Categorical and Quantitative correct
  - Ordinal can be important
- Remember Factors, Levels
  - Order of levels
- Somewhere in the process, metadata must be considered, I did not include in this class
  - Other than some variable naming conventions

Some R functions: *eyeball the Environment*

levels(), class(), str(), glimpse(), attributes(), fct_relevel()

Step 4: Data Integrity/Validation Checks

- Levels
  - How many? Too many?
  - Misspellings
  - Mean the same thing
  - Single/low frequency
- Outliers here? Or later?

NOTE: some of this work belongs in Model Specific Processing

Some R functions: factor(), levels(), table(), mutate(), rename()

Step 5: Handle Dates

- Everybody hates dates, dates hate all software, especially Excel
- Get in Lubridate format
- Create sub variables
  - Month
  - Year
  - Season
  - Etc.

NOTE: some of this work belongs in Model Specific Processing

Some R functions: See Tidy/dplyr presentation

Step 6: Handle Categorical Variables

- Decide on rare levels
- Decide on too many levels
- Decide on order

NOTE: some of this work belongs in Model Specific Processing

Some R functions: table(), distinct(), Dr. Petrie’s rare_levels() function

Step 7: Eliminate Zero-Variance Predictors

- A variable must vary to be a variable!!
- Find via code

Some R functions: unique(), distinct(), table(), summary()

Step 8: Handle ***near*** zero-variance predictors

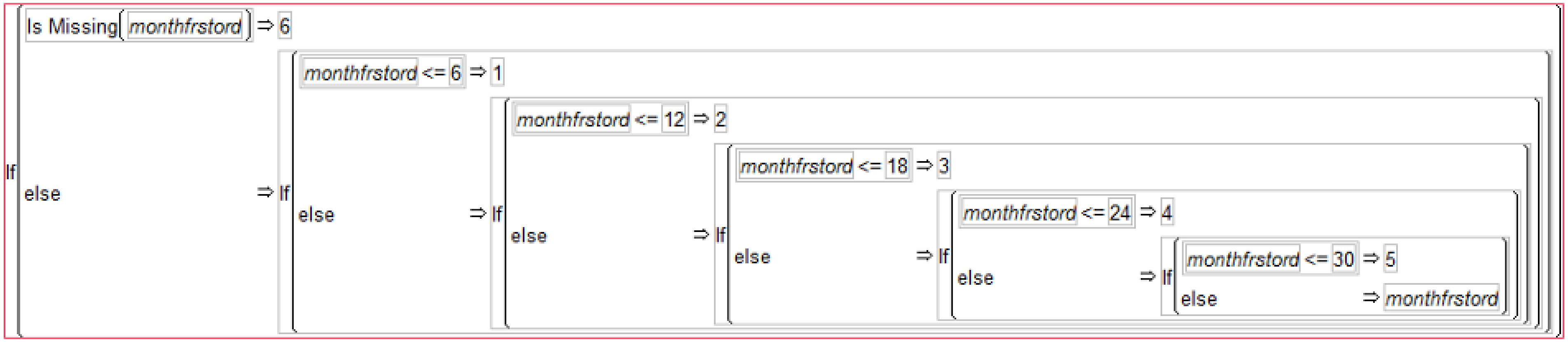
- Same as step 7, except not zero, just near zero
- Consider making quantitative variables categorical (bin)
- Can other values “appear”, impact on prediction
- Modeling implications if 1 value dominates

- NOTE: some of this work belongs in Model Specific Processing

Binning a Variable/Formula Editor

In the formula editor window, you can create a formula such as the one below. To bin a variable, use a series of If-Else statements to group the values of *monthfrstord* into 6 new groups

ifelse() and cut() in R.



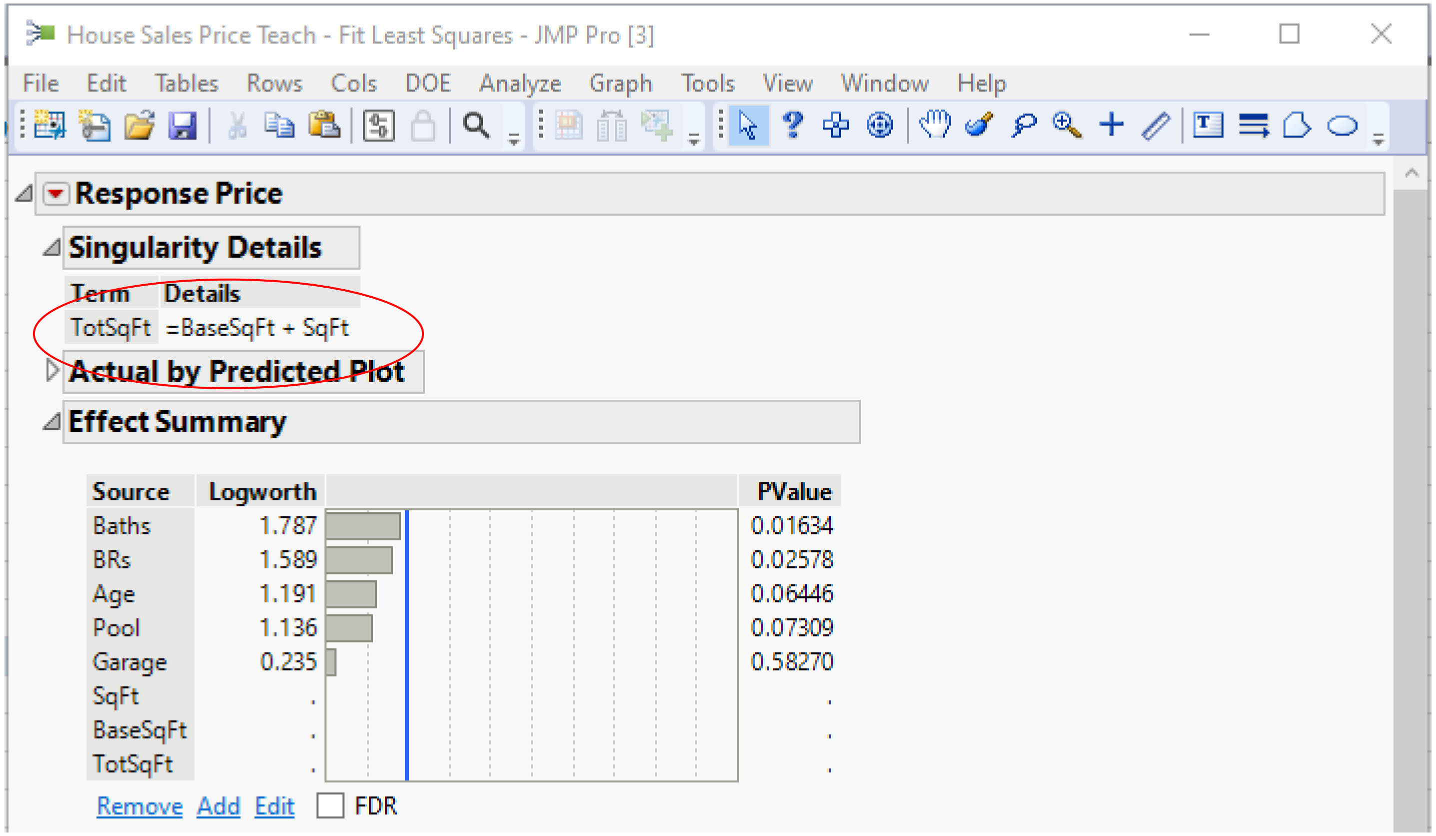
```
If  Is Missing(monthfrstord) ⇒ 6
    else ⇒ If  monthfrstord <= 6 ⇒ 1
               else ⇒ If  monthfrstord <= 12 ⇒ 2
                          else ⇒ If  monthfrstord <= 18 ⇒ 3
                                     else ⇒ If  monthfrstord <= 24 ⇒ 4
                                                else ⇒ If  monthfrstord <= 30 ⇒ 5
                                                           else ⇒ monthfrstord
```

Step 9: Redundant Columns and Linear Combination Columns

- Example: State names and State abbreviations
- How do I know?
  - Look
  - Logic
  - "Singularity Details"
  - Correlation matrix
  - Chi Square tables

Some R functions: associate(), cor_matrix(), lm()

Singularity Details


House Sales Price Teach - Fit Least Squares - JMP Pro [3]

File Edit Tables Rows Cols DOE Analyze Graph Tools View Window Help

Response Price

Singularity Details

Term	Details
TotSqFt	=BaseSqFt + SqFt

Actual by Predicted Plot

Effect Summary

Source	Logworth	PValue
Baths	1.787	0.01634
BRs	1.589	0.02578
Age	1.191	0.06446
Pool	1.136	0.07309
Garage	0.235	0.58270
SqFt	.	.
BaseSqFt	.	.
TotSqFt	.	.

Step 10: Outliers, Initial Search for Missing values

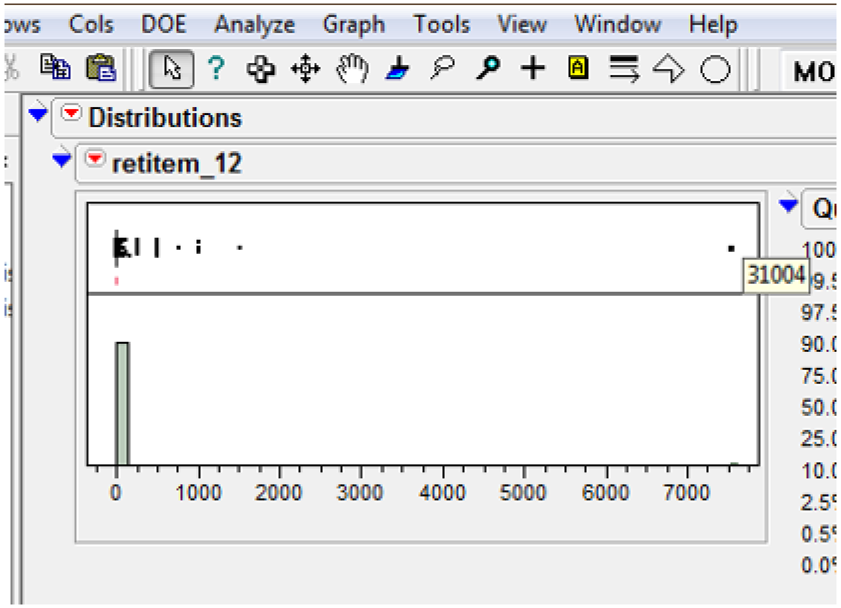
- Outliers ….. What is it good for? …
  - How to handle is more of a modeling and philosophy question, but
  - We can treat some outliers as missing data
  - What is an outlier??

- Getting a sense of the Missing Data challenge
  - May decide simple methods are enough
  - Prior visualization methods may have already done the trick
  - “Official” process and functions coming
    - Not all missing data is automatically imported as NA

Some R functions: summary(), hist(), table(), is.na(), anyNA(), complete.cases(), boxplot(), mvoutlier::aq.plot()

NOTE: some of this work belongs in Model Specific Processing

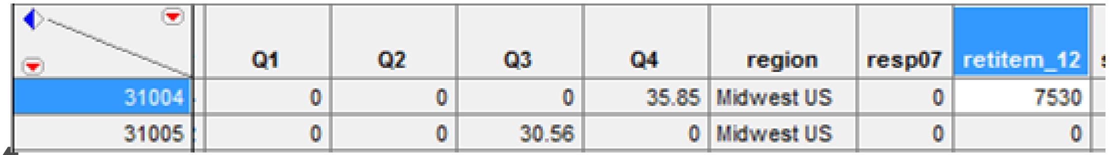
Checking for Outliers



Select the outlier by left-clicking once on the point in the histogram.

Once selected, the point will show up larger and a box will appear with the row number where that point is located in the data table.

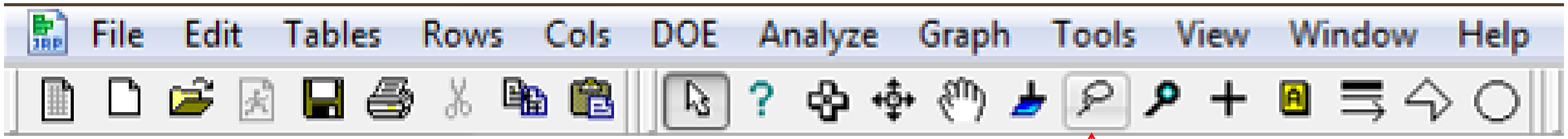
Go back to the data table and from the Rows menu, choose Next Selected. The table will automatically scroll to the selected row. Here you can see the outlier value.



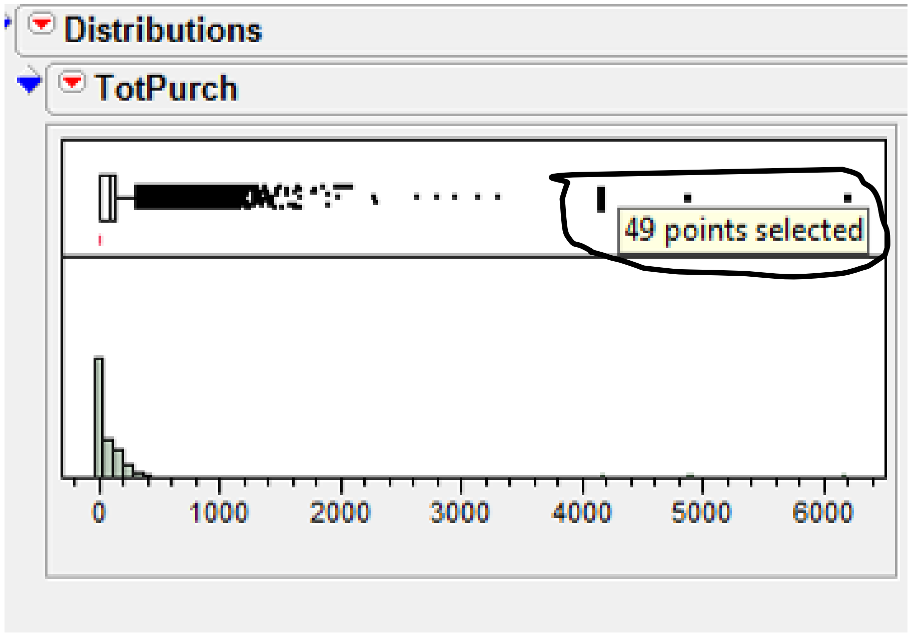
	Q1	Q2	Q3	Q4	region	resp07	retitem_12
31004	0	0	0	35.85	Midwest US	0	7530
31005	0	0	30.56	0	Midwest US	0	0

Checking for Outliers

Using the Lasso



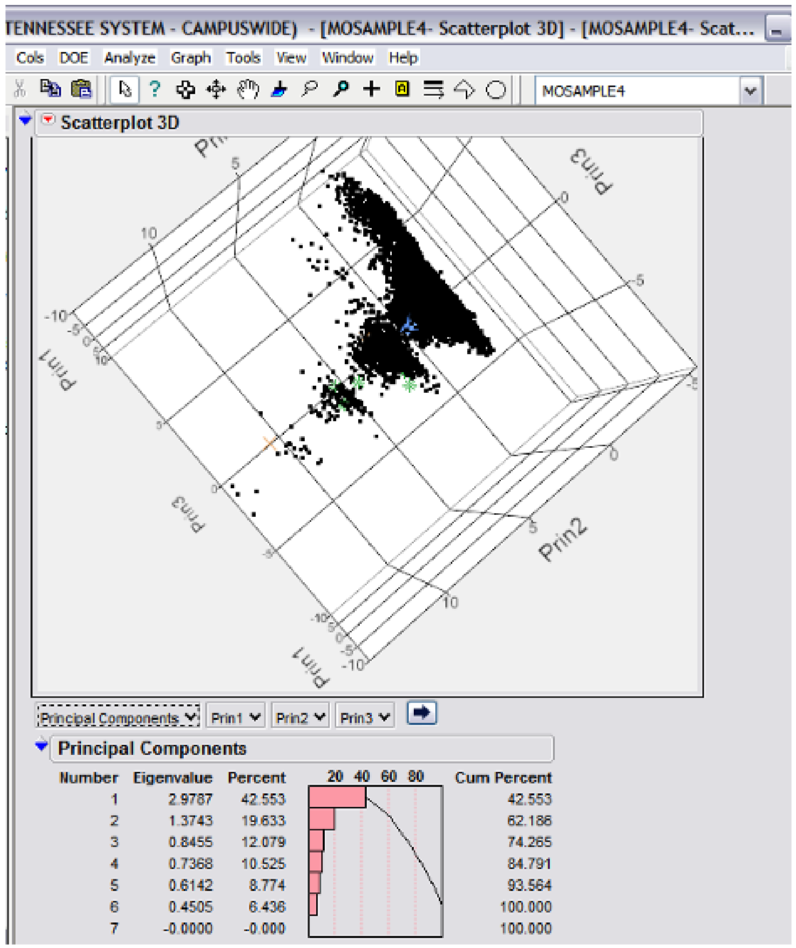
On the toolbar, click on the Lasso icon.



With the lasso, hold down the left mouse button and draw a loop around the data points you would like to select.

HASLAM COLLEGE OF BUSINESS
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Outlier Detection With 3D Graphs and Principal Components


TENNESSEE SYSTEM - CAMPUSWIDE) - [MOSAMPLE4- Scatterplot 3D] - [MOSAMPLE4- Scat...
Cols DOE Analyze Graph Tools View Window Help
MOSAMPLE4
Scatterplot 3D
Prin3
Prin1
Prin2
Principal Components
Prin1
Prin2
Prin3

Principal Components

Number	Eigenvalue	Percent	20 40 60 80	Cum Percent
1	2.9787	42.553		42.553
2	1.3743	19.633		62.186
3	0.8455	12.079		74.265
4	0.7368	10.525		84.791
5	0.6142	8.774		93.564
6	0.4505	6.436		100.000
7	-0.0000	-0.000		100.000

Step 11: Predictive Sanity Check (Decision Tree)

- Sometimes one variable is an exact or near exact match to our Y variable
  - One variable might be 0,1 the other Yes, No
  - Buy vs. Amount
  - Identifier Variable
- Easy to detect with a few splits of a Decision Tree

Some R functions: rpart(), rpart.plot()

Sanity check using Decision Tree (1 to 2 splits)

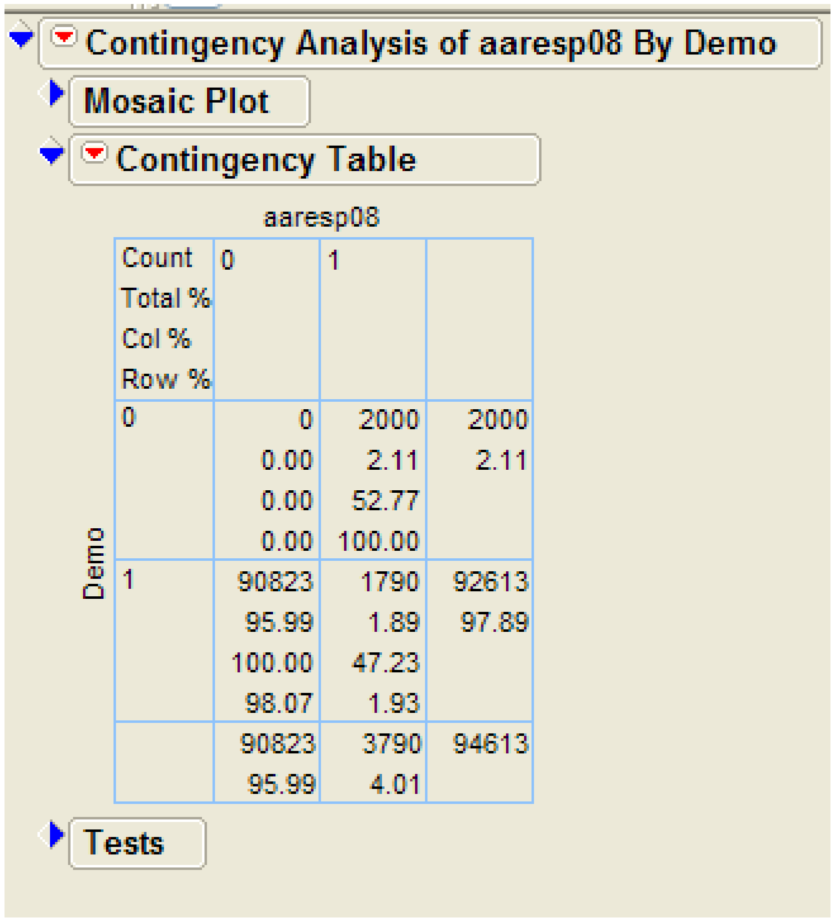
Contingency Table

The variable *Demo* is a near perfect predictor for the response variable.

A value of 0 for Demo will always result in a value of 1 for the response variable.

And 98% of the cases where Demo = 1, result in a value of 0 for the response variable.

Because of this relationship, we will exclude the variable *Demo*.


Contingency Analysis of aaresp08 By Demo

Mosaic Plot

Contingency Table

aaresp08

Demo	Count / Total % / Col % / Row %	0	1	
	0	0 0.00 0.00 0.00	2000 2.11 52.77 100.00	2000 2.11
	1	90823 95.99 100.00 98.07	1790 1.89 47.23 1.93	92613 97.89
		90823 95.99	3790 4.01	94613

Tests

Happy Hunting and Cleaning and Wrangling