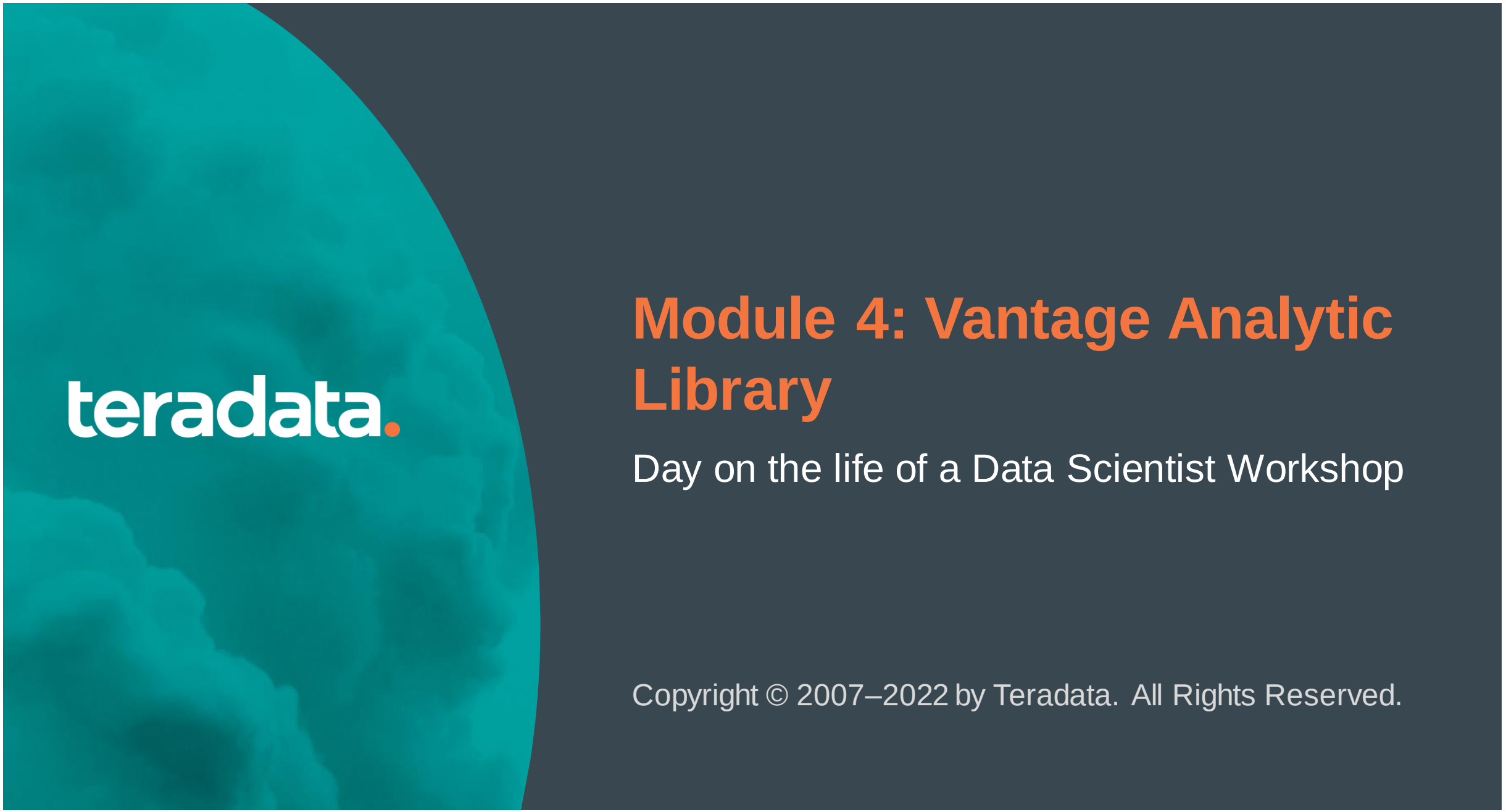


Module 4: Vantage Analytic Library

Day on the life of a Data Scientist Workshop

Objectives

After completing this module, you will be able to:

- Know more about Vantage Analytic Library functions, capabilities and use
- Understand how VAL works in Vantage

Topics

- **Introduction**
- User Experience
- VAL Functions

Analytics Library

What is Analytics Library?

A new set of advanced analytic functions:

- Runs on Vantage Advanced SQL Engine for all Vantage releases (Vantage 1.x and above)
- Deployable to all Vantage deployment options (on-premises and cloud)
- Executed as an external stored procedure
- Based on proven and matured analytics previously available as Teradata Warehouse Miner
- Easy and lightweight installation

Types of Function Included

- **Descriptive Statistics** – Provide univariate statistics, data distributions and data quality analysis on the entire data set
- **Data Transformations** – Transform and engineer data (e.g. value, format, structure) into a format suitable for modeling
- **Hypothesis Tests** – Perform statistical tests on the entire data set based upon an assumption or hypothesis
- **Advanced Analytics** – Set of commonly used multivariate statistical and machine learning algorithms such as linear & logistic regression, decision trees, K-means and association rules

Benefits: Analytics Library

- **Lightweight:** Functions run inside the Advanced SQL Engine as Java Stored Procedures. No additional infrastructure is required.
- **Proven and mature:** Based on functions previously offered by Teradata Warehouse Miner for 20+ years.
- **Use All of Your Data:** No need to move data out of Vantage.
- **Performant:** Functions run significantly faster than analytics running outside Vantage:
  - **Linearly scalable:** Inherits Vantage's MPP architecture and its linear scalability.
  - **Concurrency:** Multiple function calls can run without significantly impacting overall system performance.
  - **Workload Management:** Workloads associated to the functions can be easily managed using Vantage workload management feature.

Analytics Library Functions

Descriptive Statistics

- Values
- Frequency
- Histogram
- Statistics
- Data Explorer
- Overlap
- Adaptive Histogram
- Text Field Analysis

Data Transformations

- Bin Code
- Formula Derivation
- Design Code
- Null Value Replacement
- Recode
- Rescale
- Sigmoid
- Z-Score

Hypothesis Tests

PARAMETRIC TESTS

- Two Sample T-Test for Equal Means
- F-Test (N-Way)
- F-Test/ANOVA (2-Way)

BINOMIAL TESTS

- Binomial / Z Test
- Binomial Sign Test

TESTS BASED ON CONTINGENCY

- Chi Square Test
- Median Test

KOLMOGOROV / SMIRNOFF TESTS

- Kolmogorov / Smirnoff Test (One Sample)
- Lilliefors Test
- Shapiro-Wilk Test
- D'Agostino and Pearson Test
- Smirnov Test

RANK TESTS

- Mann-Whitney / Kruskal-Wallis Test
- Wilcoxon Signed Ranks Test
- Frieman Test with Kendall's Coefficient of Concordance & Spearman's Rho

Advanced Analytics

MATRIX BUILDING

- (E)SSCP
- Correlation
- Covariance

LINEAR REGRESSION

- Micro-modeling by segment
- Scoring
- Stepwise variants

LOGISTIC REGRESSION

- Micro-modeling by segment
- Scoring
- Stepwise variants

FACTOR ANALYSIS (PCA)

- Micro-reduction by segment
- Factor Scoring

FAST K-MEANS CLUSTERING

- K-Means Scoring

GAIN RATIO DECISION TREE

- Scoring

ASSOCIATION RULES / SEQUENCE ANALYSIS

- Association Rules / Sequence Analysis

Topics

- Introduction
- **User Experience**
- VAL Functions

User Experience (1 of 2)

- **User experience using SQL**: Users can call Analytics Library functions via any SQL-based tools (e.g., Editor or Teradata Studio) with simple syntax
  - See “Analytics Library User Guide” at Teradata Documentation for specific syntax
- **Workload Management**: Can manage impact of Analytics Library workload to other workloads via Teradata Active System Management (TASM)
  - Limit the number of concurrent active calls to Analytics Library functions via TASM system throttle on td_analyze() function

```
call twm.td_analyze('Kmeans’,'database=[$database];
                    tablename=telco_customers;
                    columns=age,support_calls,voicemail_
                    calls;
                        …
                        kvalue=3; iterations=10;
                    threshold=0.1; operatordatabase=VAL;
                    ');
```

User Experience (2 of 2)

- **User experience using Python**
  - Teradata package for Python (teradataml) provides Python wrapper for Analytics Library functions*
  - Supports all Analytics Library functions currently available
  - Functions can be executed using ‘valib’ object in teradataml

```
>>> # Use the generated matrix in building logistic regression model.
... log_reg_obj = valib.LogReg(data=df,
...                            columns=["age", "years_with_bank", "income"],
...                            response_column="nbr_children",
...                            response_value=1,
...                             matrix_data=mat_obj.result)
>>>
```

Topics

- Introduction
- User Experience
- **VAL Functions**

DATA EXPLORATION: Univariate Statistics / Distribution Functions

What Is It:

- **Data exploration** functions are used at the beginning, middle and end of any analytic modeling exercise.
- They address the need to understand the statistical properties of both the raw data and transformed data / features generated from the raw data as the data scientist iterates through the analytic process.

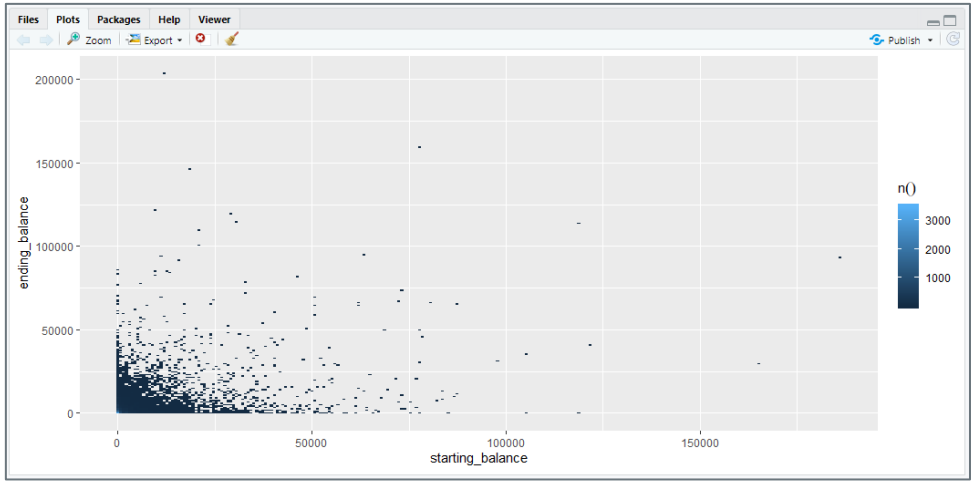
Sample Use Cases:

- They are used in any requiring analytic modeling.
- They can also be used more generically for data profiling or data quality assessment.

Example:

- Calculates basic data quality metrics: counts, uniques, NULLs, zeros, positive, negatives
- Calculates univariate statistics: min / max / mean / std, skewness, kurtosis, variance, modes, percentiles
- Performs distributions analysis on:
  - Continuous data (Histogram)
  - Discrete data (Frequency)

Visual: Output example - Raster Plot in R-Studio



DATA PREPARATION: Deriving / Aggregating / Encoding / Transforming

What Is It:

- **Data preparation** functions are used to transform raw data into a form for analytic modeling. This is typically a single row per entity being modeled, with missing values removed.
- For example, some algorithms require all numeric inputs, so you may have to perform one-hot encoding on desired categorical inputs. Similarly, some algorithms want normalized data so you may need to Z-Score all numeric inputs.

Sample Use Cases:

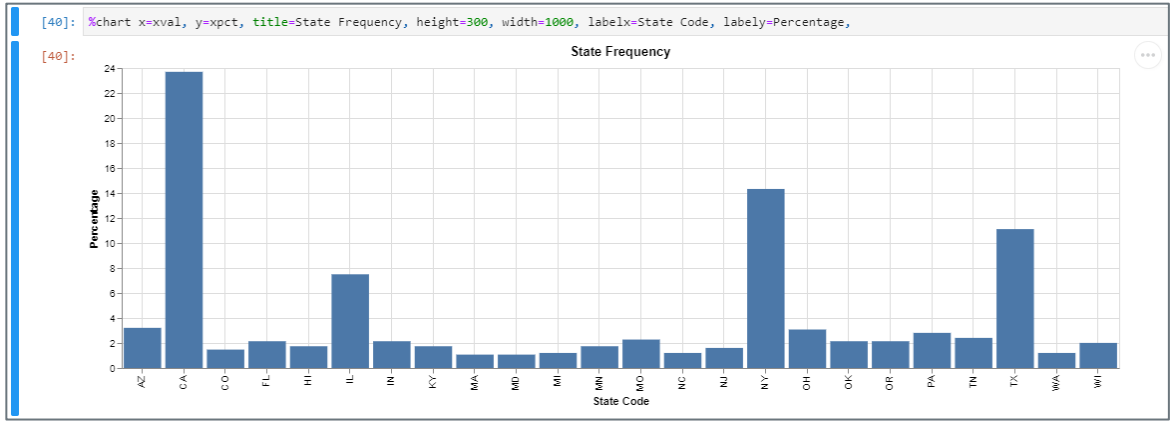
Any requiring analytic modeling – currently, problems falling into these categories:

- Classification / Regression – "Supervised Learning"
  - Decision Trees
  - Linear Regression
  - Logistic Regression
- Clustering / Segmentation – "Unsupervised Learning"
  - Association Rules
  - K-Means Clustering

Examples:

- **Bincode:** Represent a continuous data element as a set of discrete values, 1-n.
- **Design coding:** Encode categorical data elements as binary numeric equivalents (0/1).
- **Statistical transformations:** Transform numeric variables into normalized values through Z-Score or Sigmoid transformations.

Visual: Output example – One-hot encoding state code in Jupyer



VAL Variable Transformation


```
Scale housing data with Vantage Analytics Library function
variable transformation, 'vartran'

call TRNG_XSP.td_analyze('vartran','database=TRNG_TDU_TD01;
tablename=scale_housing;
keycolumns=id;
retain=columns(types,id);
rescale =
{rescalebounds (lowerbound/-1, upperbound/1),
columns (price,lotsize,bedrooms,bathrms,stories)}');
```

VAL Variable Transformation

types	id	price	lotsize	bedrooms	bathrms	stories
classic	1	-1	0.55431…	1	-1	-0.33333…
classic	2		-0.4763…	-1	-1	-1
classic	3	-0.67741…	-1	1	-1	-1
classic	4	-0.20430…	1	1	-1	-0.33333…
classic	5	-0.18279…	0.83844…	-1	-1	-1
bungalow	6	0.032258…	-0.3871…	1	-1	-1
bungalow	7	0.032258…	-0.5431…		1	-0.33333…
bungalow	8	0.161290…	-0.3871…	1	-1	0.33333…
bungalow	9	0.797849…	-0.0306…	1	-1	-1
bungalow	10	1	0.35933…	1	1	1

MODEL BUILDING AND SCORING: Logistic Regression

What Is It:

- **Logistic Regression** is one of the basic and popular algorithms to solve a classification problem. Its underlying technique is quite the same as Linear Regression; the term "Logistic" is taken from the Logit function that is used in this method of binary classification.
- Also includes GROUP BY and STEPWISE options.
- Scoring applies the model to predict binomial outcome values of data based on the input variable values.

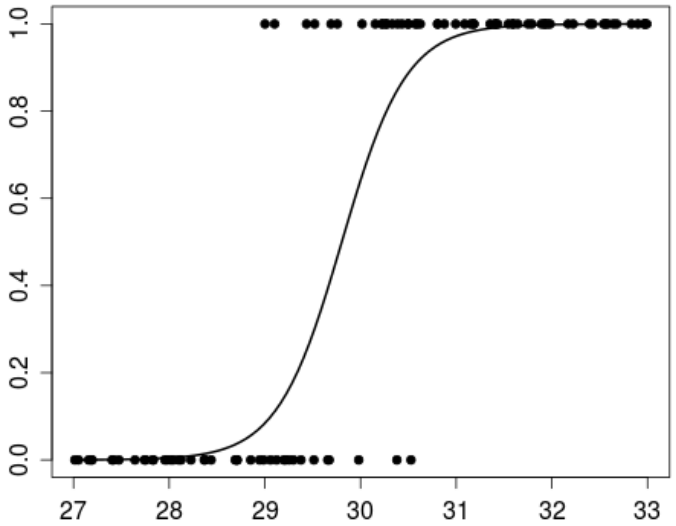
Sample Use Cases : Supervised Classification:

- Predicting a customer's propensity to purchase a particular product.
- Predicting which customers at a financial institution or telecommunications company will leave or churn.
- Predicting who will win the election in the US or whether a student will pass or fail an exam.

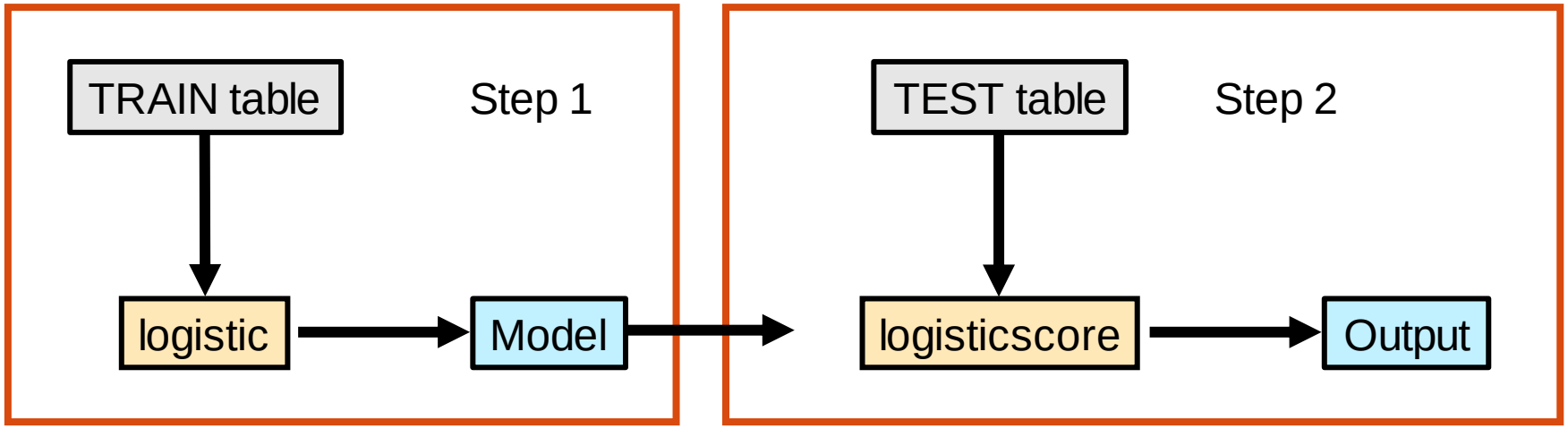
Example:

- Predicting the annual salary bands for new graduates based on a variety of factors such as education, geography, special skills, age, and other variables

Visual: $Y = 1 / (1 + e^{-x})$



Logistic Regression (VAL) – Workflow



Logistic Regression – Syntax

- Create a 'Classification' Model

```
call TRNG_XSP.td_analyze('logistic',
                         'database=TRNG_TDU_TD01;
                          tablename=housing_train_int_binary;
                          columns=price,lotsize,bedrooms,bathrms,
                          stories,garagepl,driveway,
                          recroom,fullbase,gashw,airco,prefarea;
                          dependent=homestyle;
                          statstable=true;
                          successtable=true;
                          thresholdtable=true;
                          lifttable=true;
                          outputdatabase=${QLID};
                          outputtablename=LogisticOut1;
                          overwrite=true;');
```

Logistic Regression – Results

- Create a 'Classification' Model

```
SELECT * FROM LogisticOut1;
```

Column Name	B Coefficient	Standard Error	Wald Statistic	T Statistic	P-Value	Odds Ratio	Lower	Upper	Partial R	Standardized Coefficient
(Constant)	63.7540201911286	100.509363437449	0.402348240427357	0.634309262448024	0.526389610583611					
recroom	1.64958783903626	1.86139767114417	0.785366839489263	0.886209252653832	0.376254561167507	5.2048341555675	0.135513426187966	199.90859466121	0	0.320562928701099
prefarea	0.460425262301585	1.72336231395329	0.071378119994872	0.267166839250069	0.789534610052472	1.58474777520185	0.0540793533091131	46.4396365217632	0	0.095895712561691
price	-2.15604179820962e-05	9.09818625959481e-06	5.61571042281475	-2.36974902105998	0.0184686077041444	0.999978439814442	0.9999606082405	0.999996271706361	-0.1005754983742	-1.39348571510017
lotsize	-0.00163365177819347	0.00051631014704928	10.0114655895826	-3.16409000971569	0.00172456542635575	0.998367681904516	0.997357895447108	0.999378490731822	-0.149710188965125	-2.05365814450112
stories	-0.559364143948065	1.09746448044192	0.259781548818994	-0.509687697339257	0.610666105364187	0.571572386087064	0.066513190345697	4.91173240735091	0	-0.277369652587048
fullbase	-0.818391360758338	1.41164330478996	0.336102797246397	-0.579743734115684	0.562547064766092	0.441140720311342	0.0277321913719301	7.01730103138493	0	-0.190870921336103
garagepl	-1.48097926189868	0.87712223456022	2.85087196853277	-1.68845253665384	0.092422061364361	0.227414880580275	0.0407578323981056	1.26889789928438	-0.0487896250836737	-0.644547137166363
bedrooms	-2.10762409868453	1.24835776005406	2.85041556459423	-1.68831737673763	0.092448021892078	0.121526358337006	0.0105212280024767	1.40370076260848	-0.0487765380595637	-0.939164712541167
airco	-10.5793705603842	3.81958524468553	7.6716228612407	-2.7697694599444	0.00597923427507987	2.54353514964594e-05	1.42626718058225e-08	0.0453601621460804	-0.12596475459124	-2.64994614233148
gashw	-11.6134143693516	4.47894382513032	6.72308516855151	-2.59289127588326	0.010010222400908	9.04395147180733e-06	1.39275372093508e-09	0.0587275819083729	-0.114949810911139	-1.09942139826144
bathrms	-13.2851162892105	4.72007479137303	7.92196574366256	-2.81459868252342	0.00522526753014535	1.69960227660763e-06	1.63159496952738e-10	0.0177044422948092	-0.12871475017909	-3.96344894398987
driveway	-19.5660272766856	99.0031694354894	0.039057740269768 9	-0.197630312122834	0.843475693072939	3.18113661891145e-09	1.70189089292972e-93	5.9461098418353e+75	0	-4.44280617655127

```
SELECT * FROM LogisticOut1_rpt;
```

rid	Total Observations	Total Iterations	Initial Log Likelihood	Final Log Likelihood	Likelihood Ratio Test G Stat	Chi-Square Degrees of Freedom
1	297	13	-178.72251158315	-11.6851324117444	334.074758342811	12

Chi-Square Value	Chi-Square Probability	McFaddens Pseudo R-Squa	Dependent Variable	Dependent Response Value	Total Distinct Values
21.0260698174829	0	0.934618575420434	homestyle	1	2

Logistic Regression – Required Arguments

- **Database:** Specify the database that has the input file
- **TableName:** Specify the table to build the model from
- **Dependent:** Specify the name of the column that contains the response variable (that is, what you want to predict). (**Y**-var)
- **Columns:** (**X**-var). Specify the names of columns that contain numeric predictor variables (which must be numeric values) and specify the names of the columns that contain the categorical predictor variables (which can be either numeric or VARCHAR values).

Logistic Regression – Optional Arguments

- **backward:** Whether to start with all independent variables in the model and do the following until no more independent variables can be removed from the model:
  - Take one backward step, removing the independent variable that worst explains the variance of the dependent variable
  - Take one forward step, adding the independent variable that best explains the variance of the dependent variable
- **backwardonly:** Like backward without the forward step
- **constant:** Whether the linear model includes a constant term. Default: True
- **convergence:** The convergence criterion. The algorithm stops iterating when the change in the log likelihood function falls below this value. Default: .001

Logistic Regression – Optional Arguments (cont.)

- **forward:** Whether to start with no independent variables in the model and do the following until no more independent variables can be added to the model:
  - Take one forward step, adding the independent variable that best explains the variance of the dependent variable
  - Take one backward step, removing the independent variable that wors explains the variance of the dependent variable
- **forwardonly:** Like forward without the backward step
- **groupby:** The input table columns for which to separately analyze each value or combination of values. Default behavior: Input is not grouped.
- **Overwrite:** When overwrite is set to true (default), the output tables are dropped before creating new ones.

Logistic Regression – Optional Arguments (cont.)

- **maxiterations:** The maximum number of attempts to converge on a solution. Default: 100
- **response:** The value assumed by the dependent column, to treat as the response value
- **stepwise:** Whether to perform the stepwise procedure (**forward**, **forwardonly**, **backward**, or **backwardonly**). Default: false
- **ColumnsToExclude:** If a column specifier such as all is used in the columns' parameter, the columnstoexclude parameter may be used to exclude specific columns from the analysis. For convenience, when the columnstoexclude parameter is used, dependent variable and group by columns, if any, are automatically excluded as input columns and do not need to be included as columnstoexclude.

Using VAL Logistic Regression Score


```
call TRNG_XSP.td_analyze('logisticscore',
                         'database=${QLID};
                          tablename=housing_test_int_binary;
                          modeldatabase=${QLID};
                          modeltablename=LogisticOut1;
                          outputdatabase=${QLID};
                          outputtablename=LogisticScore1;
                          estimate=Estimate;
                          probability=Probability;
                          retain=homestyle;
                          samplescoresize=25;
                          lifttable=true;
                          successtable=true;
                          scoringmethod=scoreandevaluate');
```

View Prediction Results and Accuracy


```
SELECT * FROM LogisticScore1;
```

Actual | Predict

sn	homestyle	Probability	Estimate
13	1	0.9999999981385996	1
16	1	0.9999272619341869	1
25	1	0.9999999632752141	1
104	0	2.8505208410555573e-07	0
111	1	0.9999999999999988	1
132	1	0.9999999089883381	1
140	1	0.9999999251652865	1
142	1	0.9999998840737476	1
162	0	0.9930947017612649	1
195	1	0.9999999983487906	1

98% accurate

```
SELECT
(SELECT cast(count(*) as dec(4,2))
 FROM LogisticScore1
WHERE homestyle = Estimate)/
(SELECT cast(count(*) as dec(4,2))
 FROM LogisticScore1);
```

(Count(*)/Count(*))
.98

MODEL BUILDING AND SCORING: Linear Regression

What Is It:

- **Linear Regression** is a statistical technique used to determine values of a continuous outcome variable as a linear combination of multiple input variables.
- Implementation includes GROUP BY, in which multiple models can be built simultaneously based upon a discrete GROUP BY column as well as STEPWISE.
- Scoring applies the model to predict continuous outcome values of data based on the input variable values

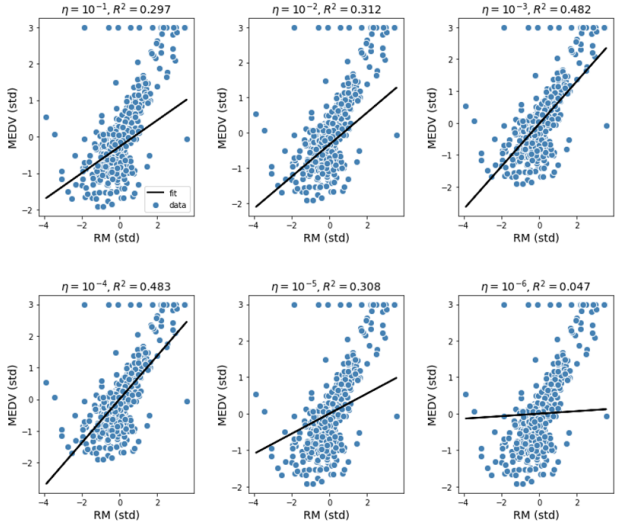
Sample Use Cases: Supervised Estimation

- Predicting incidence rate of disease in vulnerable urban areas.
- Predicting rates of manufacturing flaws over a time period.
- Estimating the lifetime value of a customer based upon their purchase history.

Example:

- Predicting the annual salary bands for new graduates based on a variety of factors such as education, geography, special skills, age, and other variables.

Visual: $Y = a + B(x_1) + B(x_2) + \ldots + B(x_n)$



MODEL BUILDING AND SCORING: Decision Tree

What Is It:

- **Decision Tree** is a modeling algorithm that predicts a binomial or multinomial dependent variable value based upon a set of independent variables.
- It uses Information Gain Ratio (a ratio of information gain to the intrinsic information) to reduce a bias towards multi-valued attributes.
- Decision Tree Scoring applies the Decision Tree model to predict values / categories on new data.

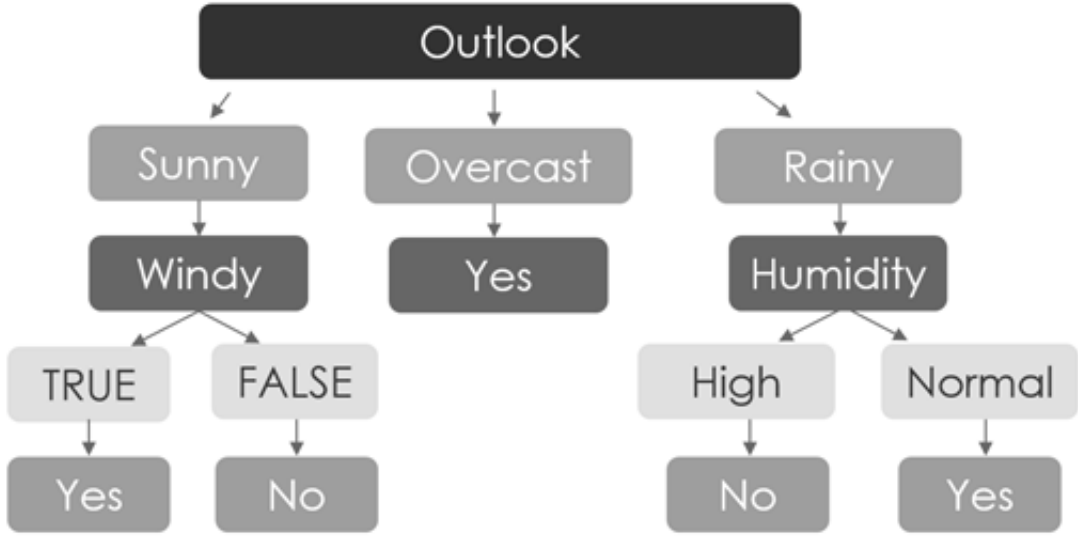
Sample Use Cases: Supervised Classification

- Determine the nature of the transaction (e.g., fraud versus legitimate)
- Determine optimal interest rate band of loan based on credit risk of user
- Determine next product offer based on rules generated by tree

Example:

- Airline bumps a passenger by using decision tree model to go through a set of rules to determine next action (e.g. refund, extra miles):
  - Is s/he part of loyalty program (yes/no)
  - Was ticket higher or lower than $X
  - Frequency of travel, etc.,

Visual: Single Decision Tree Flow



MODEL BUILDING AND SCORING: K-Means

What Is It:

- **K-Means** clustering is one of the simplest and popular unsupervised machine learning algorithms. The objective of K-means is simple: group similar data points together based upon their centroids – a location representing the center of the cluster. The K-Means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster.
- K-Means cluster scoring is effectively a final iteration of the algorithm that defines the final clusters.

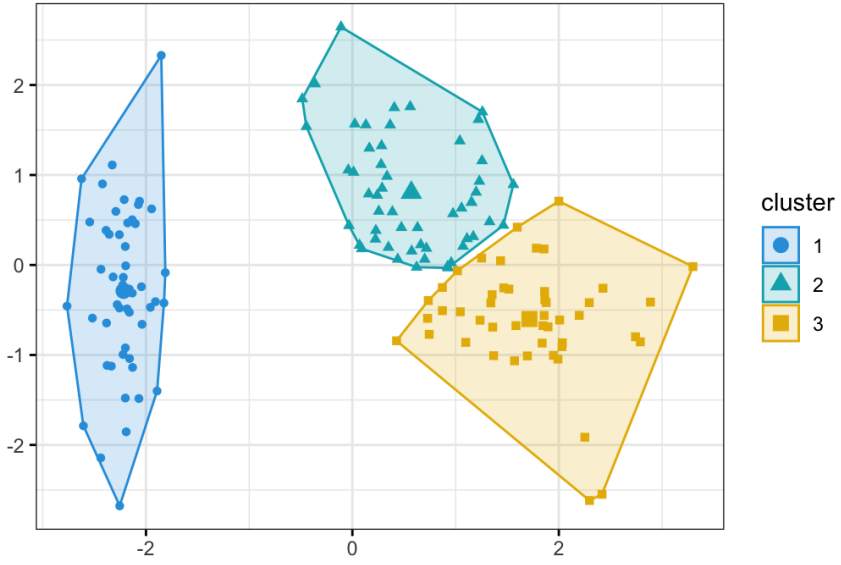
Sample Use Cases: Unsupervised Segmentation

- Document classification based on tags, topics, and the content.
- Customer segmentation based on purchase history, interests, or activity monitoring.
- Insurance fraud detection based on its proximity to clusters that indicate fraudulent patterns.

Example:

- Use clustering to get a meaningful understand of the structure of the data.
- Cluster-then-predict where different models will be built for different subgroups. An example of that is clustering patients into different subgroups and build a model for each subgroup to predict the probability of the risk of having heart attack.
- This assumes that we believe there is a wide variation in the behaviors of different subgroups.

Visual: Visualizing Clusters



HYPOTHESIS TESTING: Parametric / Binomial / Contingency / Non-Parametric / Rank Tests

What Is It:

Hypothesis or statistical tests are used to assess the plausibility of a hypothesis by using sample data.
Statistical tests may be:

- **Parametric**: Tests of normally distributed data.
- **Non-Parametric**: Tests of non-normal distributions
- **Binomial**: Tests the distribution of binary (i.e. indicator 0/1) variables.
- **Contingency**: Tests the difference in expected and observed frequencies.
- **Rank**: Tests on the relationships between a set of ranked variables).

Sample Use Cases:

Test the public opinion (e.g. presidential approval rating)

- Take poll results and determine NULL and alternative hypothesis:
  1) Null hypothesis: The proportions who would approve/disapprove are each 0.50.
  2) Alternative hypothesis: Fewer than 0.50, or 50%, of the population would answer disapprove.
- Collect and summarize data into a Z statistic (Sample proportion - random draw) / Standard deviation).
- Calculate the pValue - if less than 0.05, accept the alternative hypothesis.

Example:

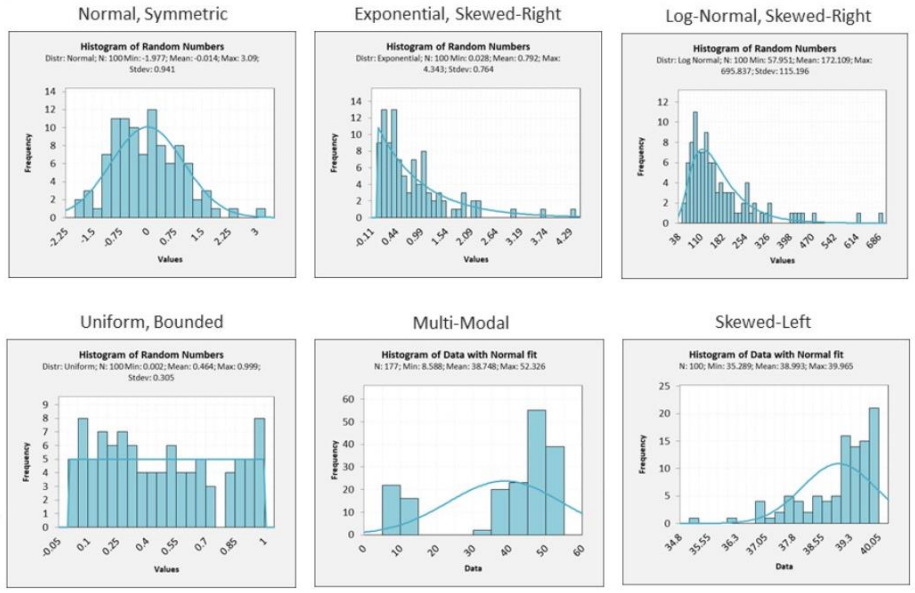
Choice of statistical test for independent observations*

		Outcome variable					
		Nominal	Categorical (>2 Categories)	Ordinal	Quantitative Discrete	Quantitative Non-Normal	Quantitative Normal
Input Variable	Nominal	X^2 or Fisher's	X^2	X^2-trend or Mann - Whitney	Mann-Whitney	Mann-Whitney or log-rank[a]	Student's *t* test
	Categorical (2>categories)	X^2	X^2	Kruskal-Wallis[b]	Kruskal-Wallis[b]	Kruskal-Wallis[b]	Analysis of variance[c]
	Ordinal (Ordered categories)	X^2-trend or Mann - Whitney	e	Spearman rank	Spearman rank	Spearman rank	Spearman rank or linear regression[d]
	Quantitative Discrete	Logistic regression	e	e	Spearman rank	Spearman rank	Spearman rank or linear regression[d]
	Quantitative non-Normal	Logistic regression	e	e	e	Plot data and Pearson or Spearman rank	Plot data and Pearson or Spearman rank and linear regression
	Quantitative Normal	Logistic regression	e	e	e	Linear regression[d]	Pearson and linear regression

*** Parametric and Non-parametric tests for comparing two or more groups**

Visual:

Different Distribution Shapes*

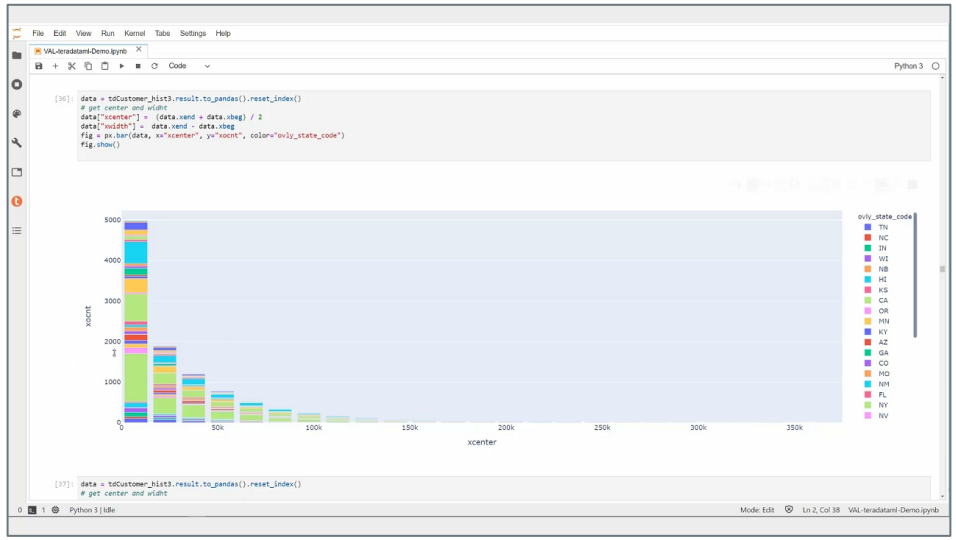


* Sigma Magic Analysis Software

For More Information – Overview and Demo Videos

TechBytes series: Teradata Vantage – Analytics Library

- Part 1. Overview (teradata.com / YouTube)
- Part 2. Demo - Data Exploration (teradata.com / YouTube)
- Part 3. Demo - Data Transformation and Matrix Building (teradata.com / YouTube)
- Part 4. Demo - Predictive Modeling, Model Evaluation, and Scoring (teradata.com / YouTube)



Thank you.

teradata.