THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

Episode 23

HPCC SYSTEMS®

Brought to you by the HPCC Systems Developer Community

April 25, 2019

Welcome!

- Please share: Let others know you are here with #HPCCTechTalks
- Ask questions! We will answer as many questions as we can following each speaker.
- Look for polls at the bottom of your screen. Exit full-screen mode or refresh your screen if you don't see them.
- We welcome your feedback - please rate us before you leave today and visit our blog for information after the event.
- Want to be one of our featured speakers? Let us know! techtalks@hpccsystems.com

Bob Foreman
Senior Software Engineer
LexisNexis® Risk Solutions
Robert.Foreman@lexisnexisrisk.com

HPCC SYSTEMS®

Community announcements

Platform updates:

- Latest release now available:
    - 7.2.4 Gold
- More information is available on the improvements and features included the 7.2.x series which include:
    - IDE Improvements
    - Java Embed
    - Alternative ways of embedding C and C++ code
    - Spark Improvements
    - Additions to ECL standard library
    - Thor improvements
    - New geospatial library from Uber.

Read the latest blogs on the community portal

- TextVectors – Machine Learning for Textual Data
- ECL Tips – The Seven Faces (Forms) of LOOP FUNCTION

Catch up on our 5 Questions with a Developer series

- Anupam Sengupta, GuardHat
- Jo Prichard, Data Scientist, LexisNexis Risk Solutions

Information on our annual Community Day event in the Fall coming soon!

- Day 1 includes a hands-on workshop and poster competition
- Day 2 includes both general and breakout sessions

Jessica Lorti
Director, Marketing
LexisNexis Risk Solutions
Jessica.Lorti@lexisnexisrisk.com

2019 HPCC Systems Community Day

Watch for Details Announced Soon!

HPCC SYSTEMS®

Today's speakers

Jeremy Meier
Undergraduate Student and Research Assistant
Clemson University
jjmeier@g.clemson.edu

Jeremy is a senior undergraduate student, majoring in Computer Science at Clemson University. He is originally from Greenville, South Carolina, and he is conducting research with Dr. Amy Apon's group with a focus on time series analysis. In the past, he has worked with HPCC Systems in the development of text analysis libraries. His other interests include bioengineering and animation.

David Noh
Undergraduate Student and Research Assistant
Clemson University
dnoh@g.clemson.edu

David is a senior undergraduate student, majoring in Computer Science at Clemson University. He is working on research with a focus on machine learning algorithms and time series analysis. His interests include machine learning algorithms and high performance computing.

CLEMSON UNIVERSITY

HPCC SYSTEMS®

Today’s speakers

Roger Dev
Senior Architect
LexisNexis® Risk Solutions
roger.dev@lexisnexisrisk.com

Roger is a Senior Architect working on the Machine Learning Team. Roger has been involved in the implementation and utilization of machine learning and AI techniques for many years, and he has over 20 patents in diverse areas of software technology. Roger has also served as a mentor to a number of HPCC Systems interns and is a strong supporter in our academic community.

Allan Wrobel
Consulting Software Engineer
LexisNexis® Risk Solutions
allan.wrobel@lexisnexis.com

Allan has spent his career working in the technology industry for over 40 years and has been working with databases since the mid-eighties.

Allan has worked with LexisNexis Risk Solutions since 2011 and the inception of LexisNexis Risk Solutions in the UK. Initially working with Data Operations, Allan is now serves as an ECL developer on both Thor and ROXIE. Allan is a passionate ambassador for the HPCC Systems community and has contributed several video tutorials on YouTube for users.

HPCC SYSTEMS®

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

An Investigation into Time Series Analysis

Jeremy Meier
Undergraduate Student
and Research Assistant

David Noh
Undergraduate Student
and Research Assistant

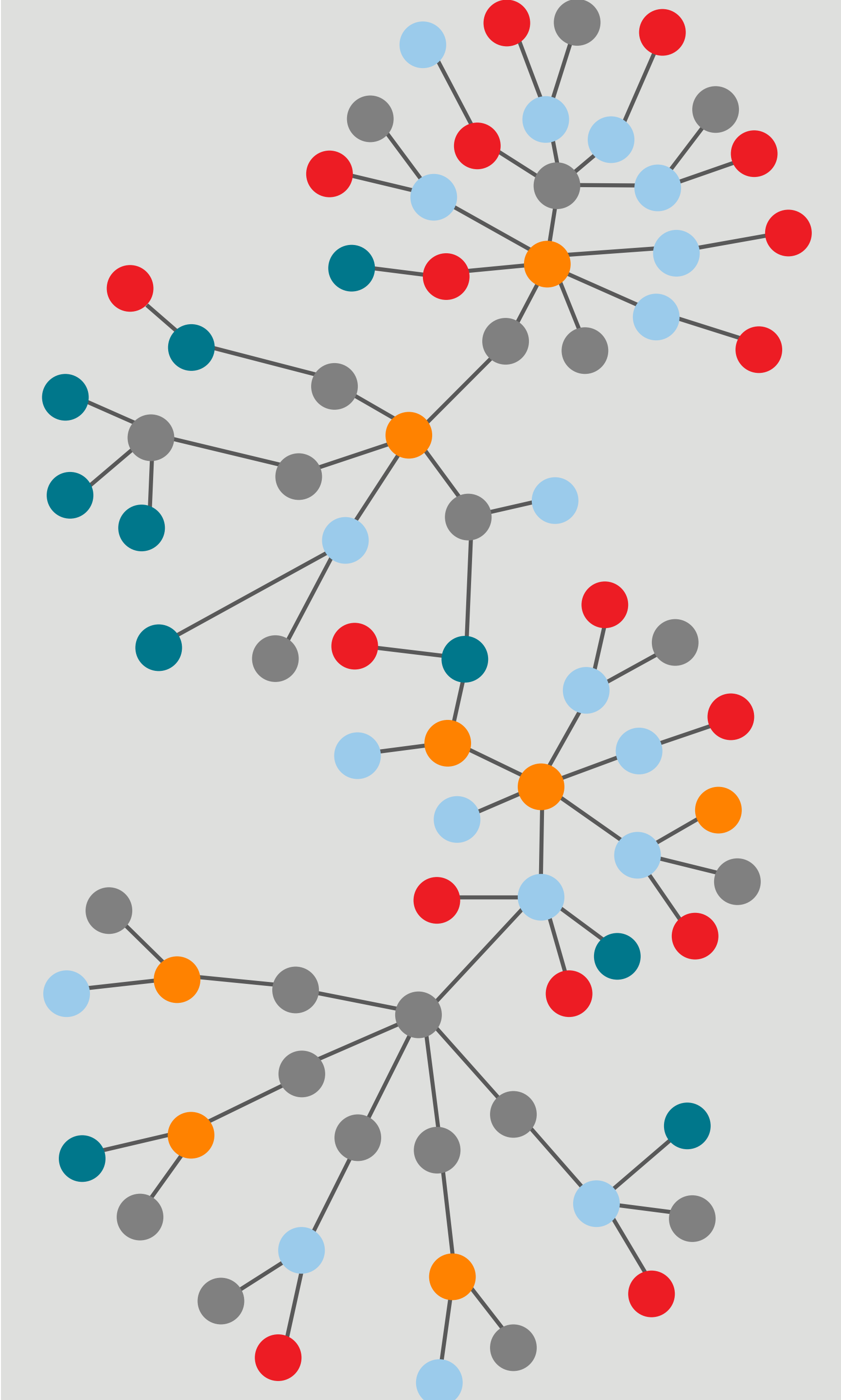
HPCC SYSTEMS®

CLEMSON UNIVERSITY

Quick poll:

How large are the time series data sets that you deal with?

See poll on bottom of presentation screen



HPCC SYSTEMS®

What is a Time Series?

- A series of data points that are measured at a regular or semi regular interval

Time Series Example

	A	B
1	date	value
2	9/6/2017	531974.19
3	9/7/2017	484704.26
4	9/8/2017	693635.27
5	9/9/2017	420176.65
6	9/10/2017	257548.74
7	9/11/2017	212416.06
8	9/12/2017	410240.57
9	9/13/2017	559267.26
10	9/14/2017	556496.67
11	9/15/2017	813277.37
12	9/16/2017	600138.13
13	9/17/2017	371246.62
14	9/18/2017	319319.61
15	9/19/2017	561685.94
16	9/20/2017	650536.61
17	9/21/2017	599229.88

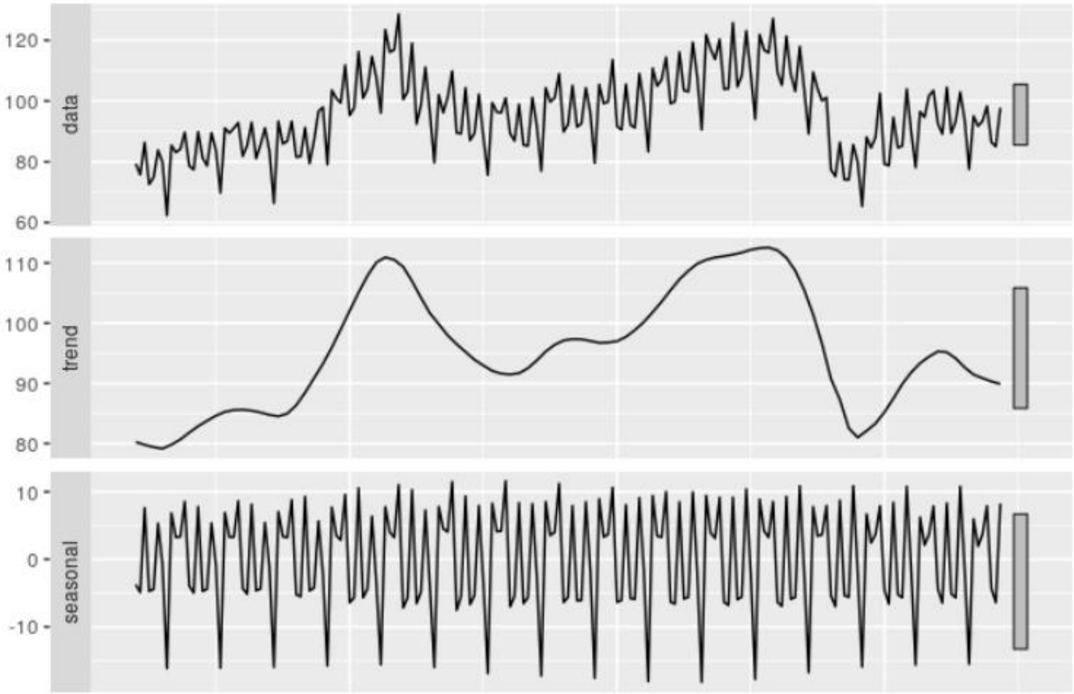
Regression Data Set

Loan ID	Date	Income per month	Loan type	Loan amount
ID207	15/07/18	25000	Car Loan	1000000
ID190	15/07/18	50000	Home Loan	2500000
ID007	22/07/18	70000	Personal Loan	1500000
ID433	29/07/18	45000	Education Loan	4500000
ID204	29/07/18	20000	Education Loan	5000000
ID611	08/08/18	80000	Business Loan	9000000
ID947	17/08/18	60000	Personal Loan	3700000
ID200	21/08/18	20000	Car Loan	500000
ID222	29/08/18	30000	Personal Loan	4300000

HPCC SYSTEMS®

What is a Time Series?

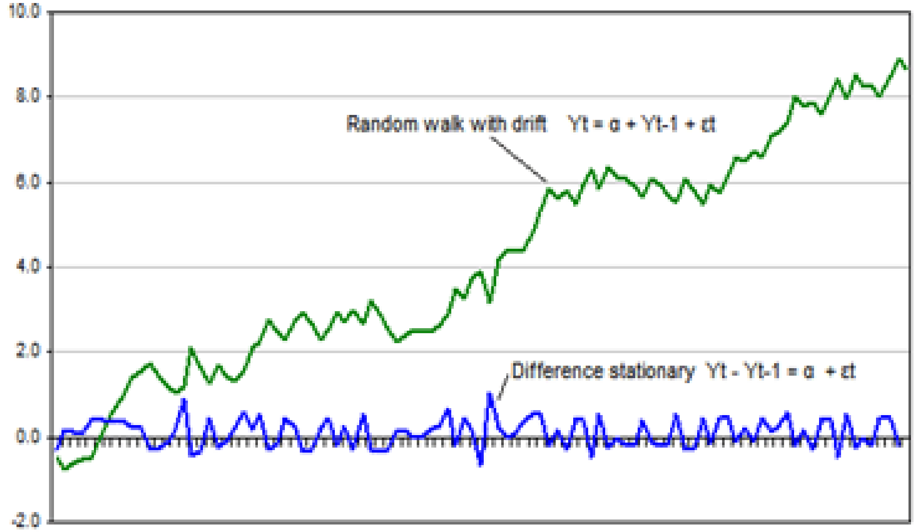
- Generally, time series have some sort of seasonality or trend.
  - Trend - A component of a time series that shows the overall movement in the series, ignoring the seasonality and any small random fluctuations
  - Seasonality - Presence of variations that occur at specific regular intervals



HPCC SYSTEMS®

Why is Stationarity important?

- Stationarity - A *stationary* time series is one whose statistical properties such as mean, variance, autocorrelation are all constant over time.
  - Thus, time series with trends, or with seasonality, are not stationary
- Most Statistical modeling methods assume or require the time series to be stationary to be effective
  - Easier to predict: one simply predicts that statistical properties will be the same in the future just as they have been in the past



HPCC SYSTEMS®

How do I know if my series is stationary?

- First, plot the time series and evaluate the variability of the time series
- Review the summary statistics for your data for seasons or random partitions and check for obvious or significant differences.
  - Split your time series into two (or more) partitions and compare the mean and variance of each group
- You can use statistical tests to check if the expectations of stationarity are met or have been violated
  - Augmented Dickey-Fuller test

How do I make my time series stationary?

- Making your data set stationary can usually be accomplished through the use of mathematical transformations
  - Differencing
    - X1, X2, X3,...........Xn
    - Difference of degree 1: (X2 - X1, X3 - X2, X4 - X3,.......Xn - X(n-1)
  - Transformation
    - Taking the log, square-root, etc.
- As you might expect, the series can be "untransformed" by reversing the mathematical transformation

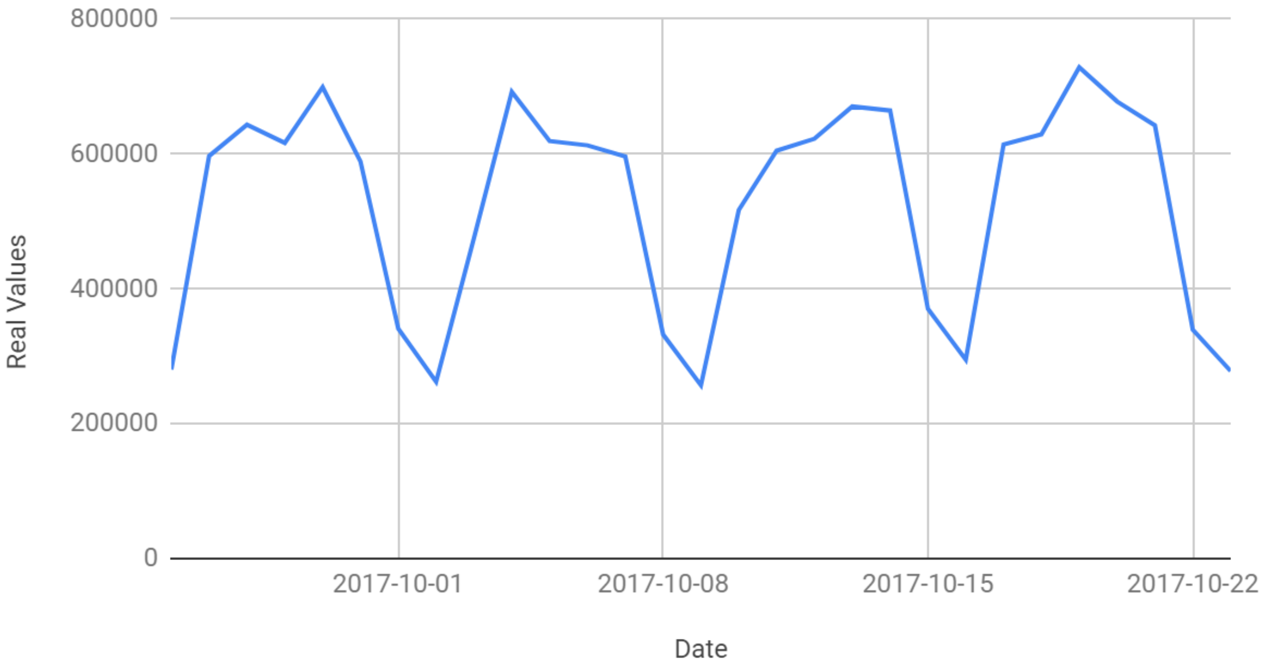
What is time series forecasting?

- Involves taking models fit on historical data and using them to predict future observations
- Components, such as trend and seasonality, may also be the most effective way to make predictions about future values, but not always
- The future is completely unavailable and must only be estimated from what has already happened
- Performance is determined by how well a model forecasts the future

The Data Set

- Stored Value Cards
- Around 16,000 total observations
- 115 accounts
- Opening balance values
  - Ranging from 0 – 10,000,000
  - Represent the balance in the account

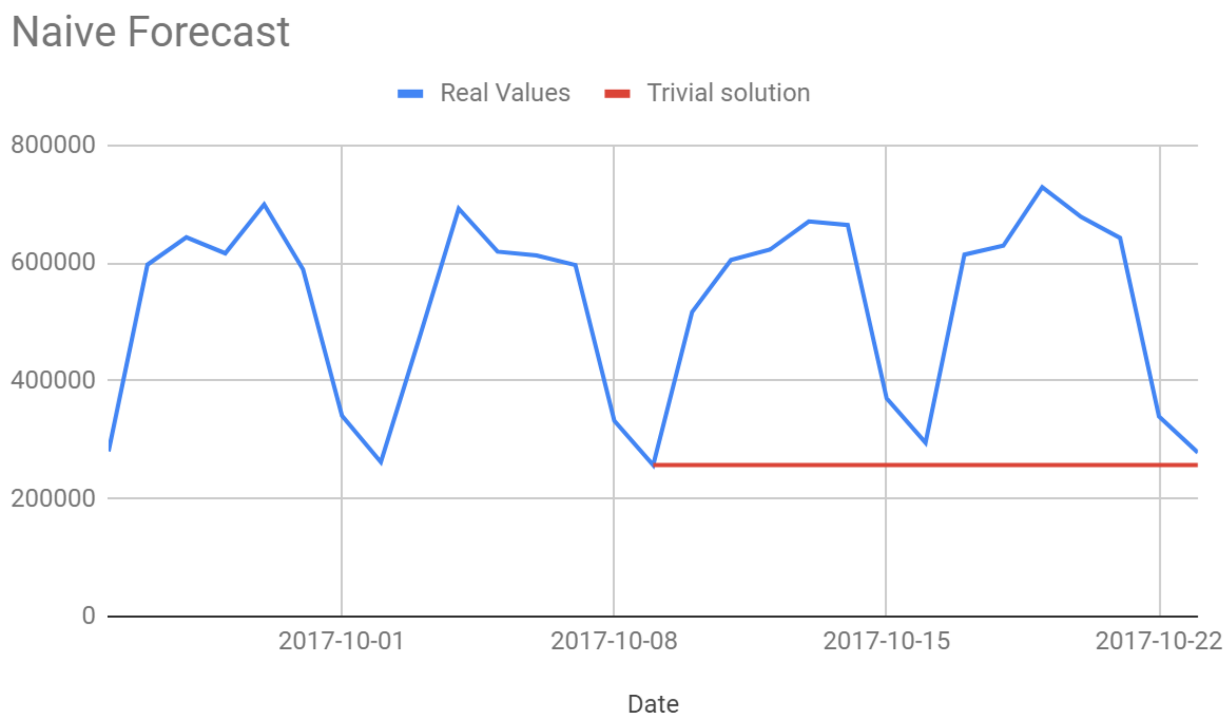
	A	B
1	date	value
2	9/6/2017	531974.19
3	9/7/2017	484704.26
4	9/8/2017	693635.27
5	9/9/2017	420176.65
6	9/10/2017	257548.74
7	9/11/2017	212416.06



HPCC SYSTEMS®

What is the Simple/Naive Method?

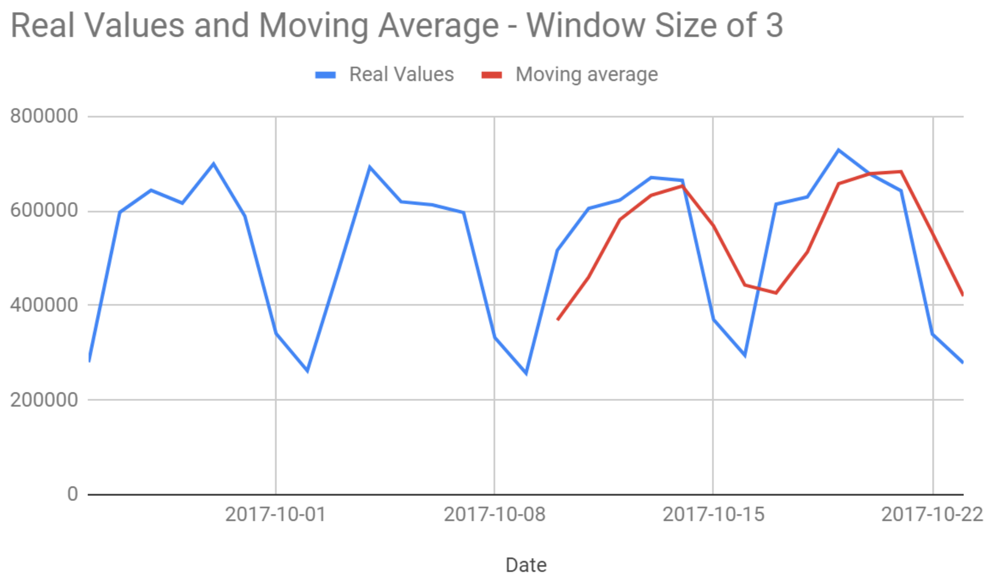
- In this forecasting technique, the value of the new data point is predicted to be equal to the previous data point. The result would be a flat line, since all new values take the previous values.

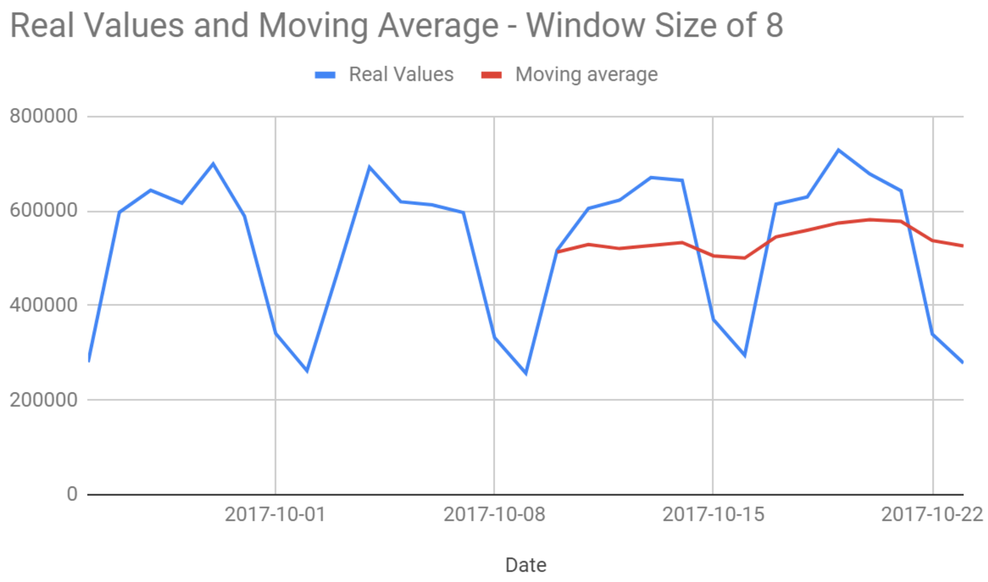


HPCC SYSTEMS®

What is Simple/Moving Averages?

- Simple Average
  - The next value is taken as the average of all the previous values.
- Moving Average
  - The next value is derived from the averages of successive segments.





HPCC SYSTEMS®

Quick poll:

How familiar are you with ARIMA prior to this talk?

See poll on bottom of presentation screen

HPCC SYSTEMS®

What is ARIMA?

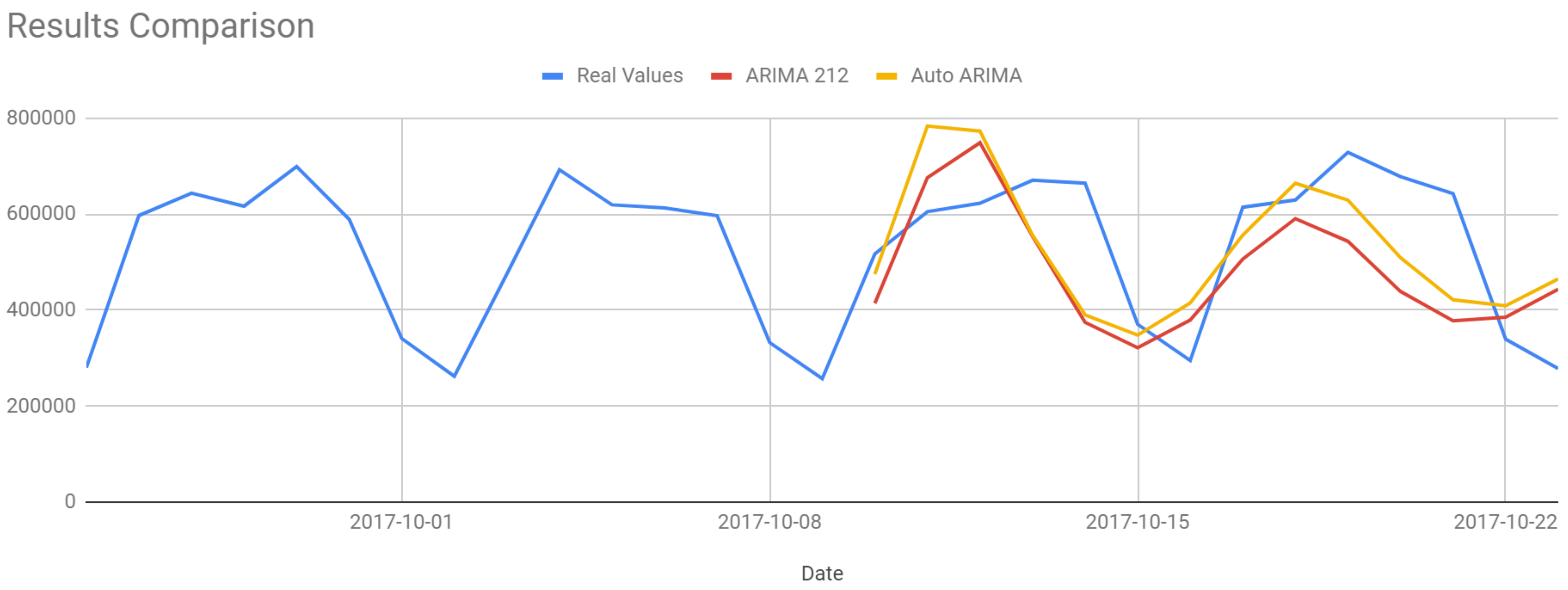
- **A**uto**R**egressive **I**ntegrated **M**oving **A**verage
  - a statistical analysis model that uses time series data to either better understand the data set or to predict future trends
- Autoregression
  - Model that shows a changing variable that regresses on its own lagged or prior values
- Integrated
  - Represents the differencing of raw observations to allow for the time series to become stationary
- Moving Average
  - Incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

What is Auto ARIMA?

- Auto ARIMA unlike the ARIMA model chooses the parameters and makes the data stationary.

	ARIMA	Auto ARIMA
Step 1	Load data	Load data
Step 2	Pre-process data	Pre-process data
Step 3	Make data stationary	Fit Auto ARIMA model
Step 4	Determine D value	Predict/Forecast values
Step 5	Determine P and Q values	Calculate error
Step 6	Fit ARIMA model	
Step 7	Predict/Forecast values	
Step 8	Calculate Error	

ARIMA vs Auto ARIMA

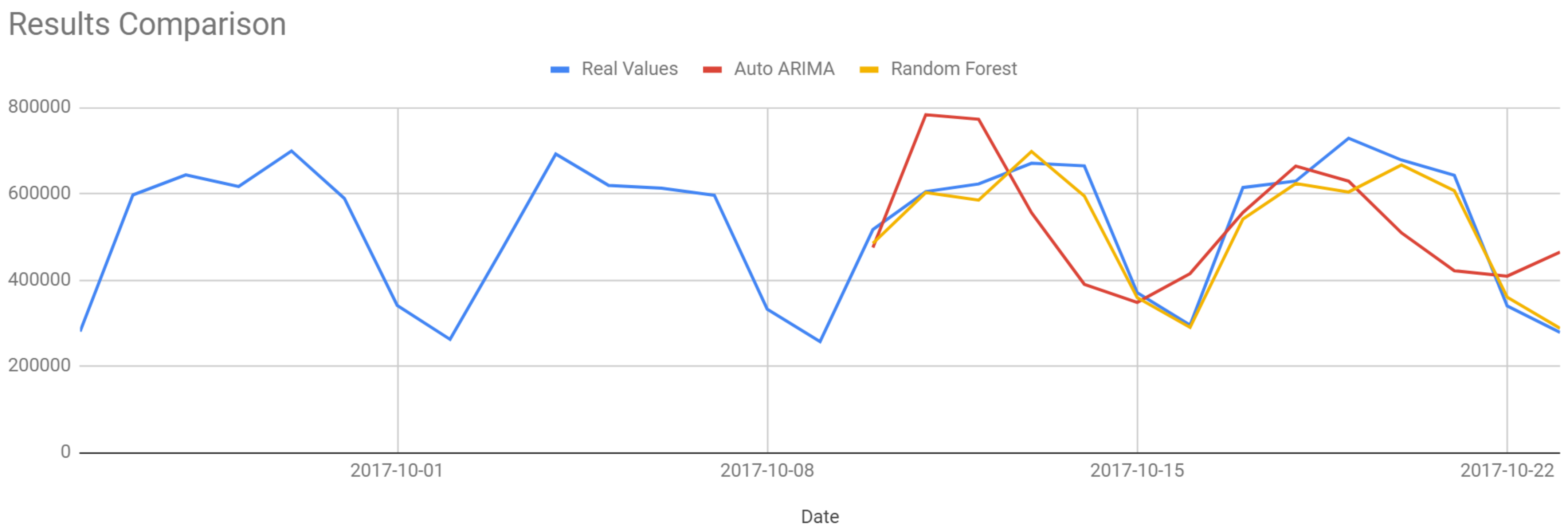


HPCC SYSTEMS®

What are some modern techniques for time series Analysis?

- Facebook Prophet
  - A procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
  - It works best with time series that have strong seasonal effects and several seasons of historical data.
  - Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.
- Random Forest
  - A supervised learning algorithm
  - Can be used for both classification and regression problems

Results

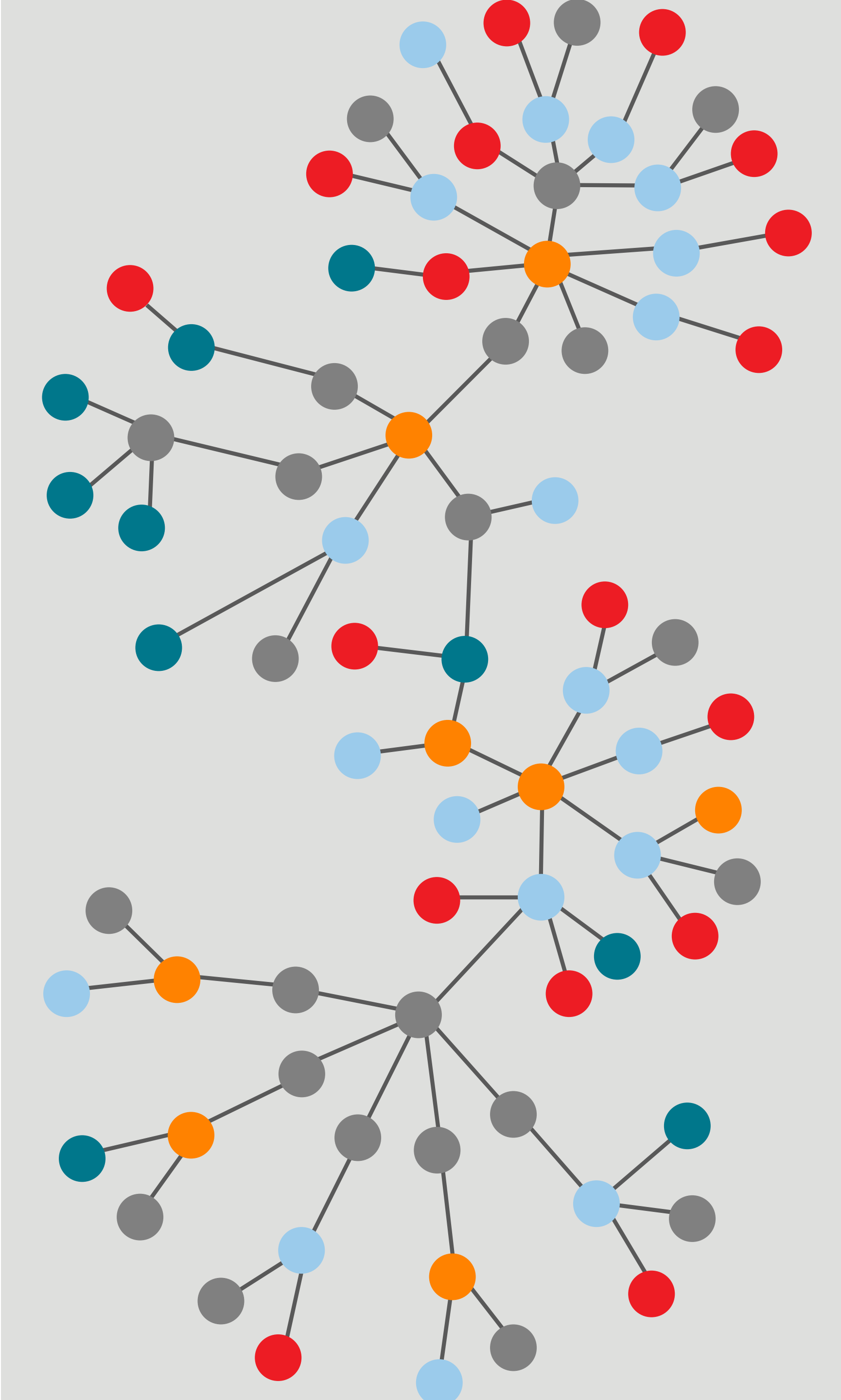


HPCC SYSTEMS®

Quick poll:

How likely is it that you will use time series analysis to solve your company's data problems?

See poll on bottom of presentation screen



HPCC SYSTEMS®

Questions?

Jeremy Meier
Undergraduate Student
and Research Assistant
jjmeier@g.clemson.edu

David Noh
Undergraduate Student
and Research Assistant
dnoh@g.clemson.edu

CLEMSON UNIVERSITY

HPCC SYSTEMS®

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

TextVectors - Machine Learning for Textual Data

HPCC SYSTEMS®

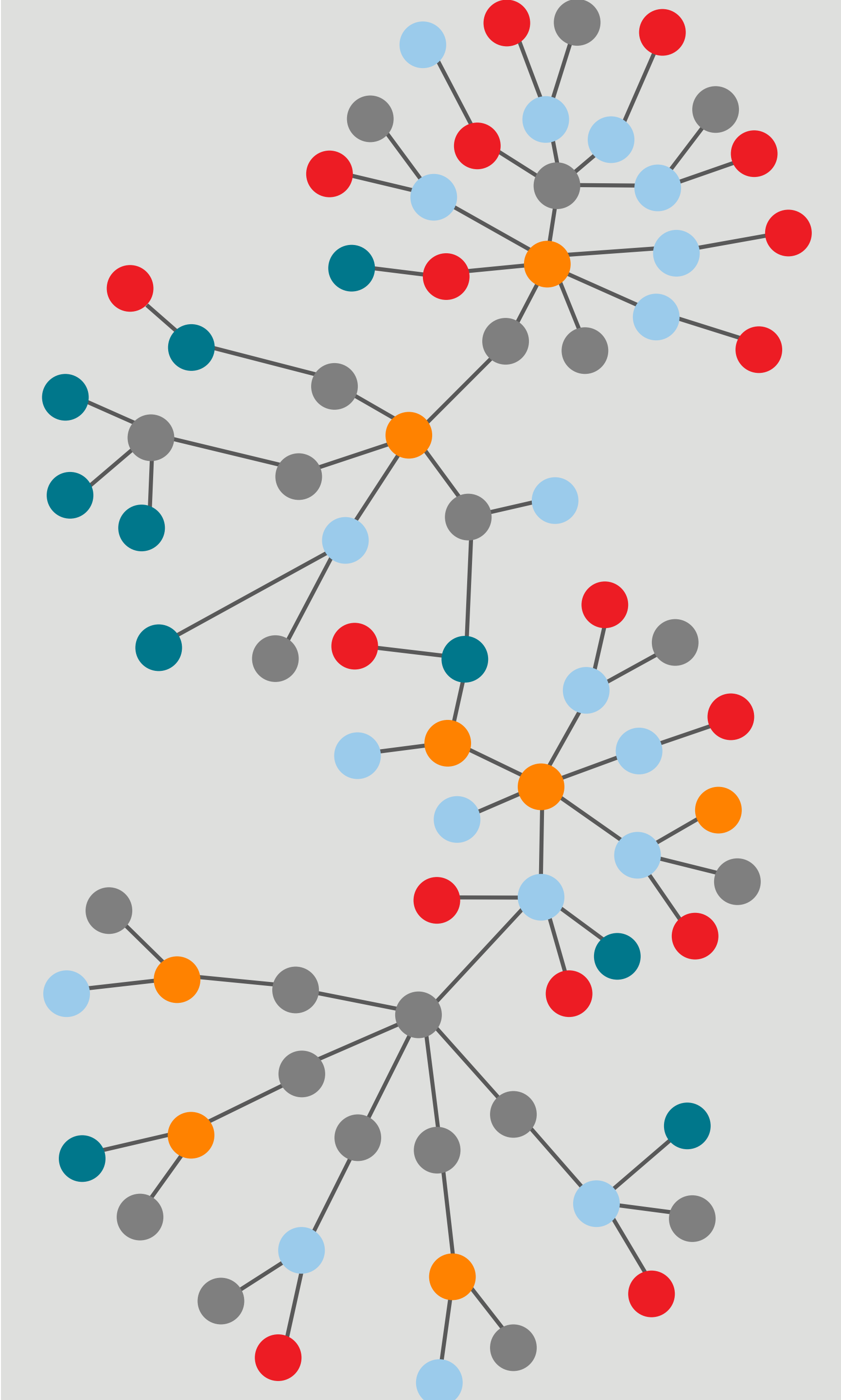
Roger Dev
Senior Architect
LexisNexis® Risk Solutions

LexisNexis® RISK SOLUTIONS

Quick poll:

Have you encountered situations in which important information was stored as free-form text?

See poll on bottom of presentation screen



TextVectors -- A new HPCC ML bundle

- Turns text into rich numerical data:
  - Words
  - Phrases
  - Sentences
- Completely automatic and unsupervised.
- Encodes the “meaning” of the text.
- Supports direct analysis of the vectors.
- Vectors can be used as features for any ML algorithm.
- Scalable, Parallelized, Enhanced version of the Sent2Vec algorithm.

Text Vectorization – The theory

"*You shall know a word by the company it keeps.*"

*- Linguist **John Rupert Firth**, 1957*

Or more rigorously:

"*The meaning of a word is closely associated with the distribution of the words that surround it in coherent text.*"

Understanding Vectorization – A Thought Experiment

- Let's pick a few words that are fairly closely related:
  - Cat
  - Dog
- Now let's pick another word that is fairly unrelated:
  - Piston

Thought Experiment -- continued

- There are many sentences in which you could just as likely find dog or cat:
  - A **dog / cat** is an animal.
  - **Dogs / cats** can make good pets.
  - I have a companion **dog / cat**.
  - My son was bitten by a **dog / cat**.
- Yet there are many sentences about dogs that would not likely be found about cats:
  - My **dog** weighs 120 pounds.
  - When I throw a ball, my **dog** brings it back to me.
  - My **dog** barks whenever the mailman comes.
  - Cocker Spaniels are a medium size **dog** breed.
- Note that **NONE** of those sentences are likely to be found about <u>Pistons</u>.

Thought Experiment -- continued

- Now imagine two words for that are interchangeable in any sentence where one is found…

Nice
Good

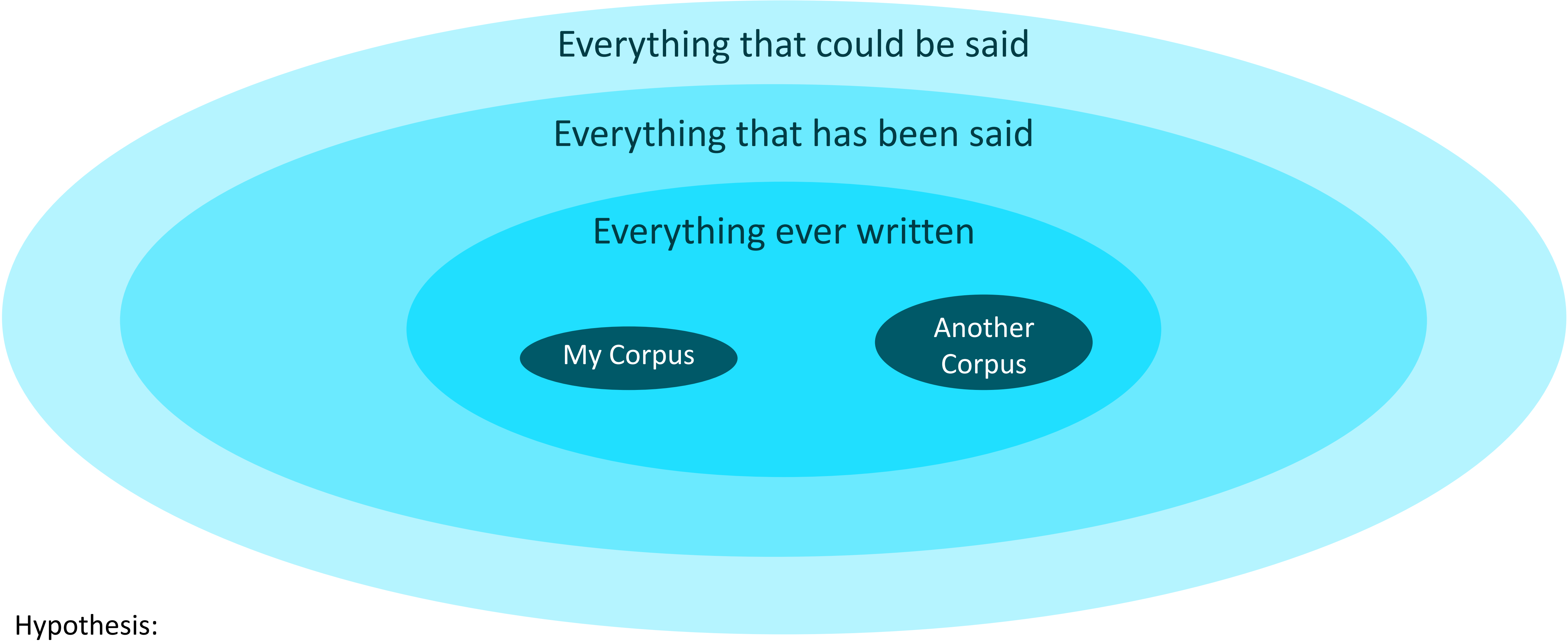
Little
Small

TV
Television

Car
Automobile

- If we think long enough on this, we have to concede that these two words must have essentially identical meaning – they are perfect synonyms.
- So John Rupert Firth was on to something: "*A word is known by the company it keeps*".
- In order to avoid philosophical argument, let's call this notion of meaning "Contextual Meaning".
- Contextual Meaning is not absolute. It is a function of the Corpus (i.e. the body of text) upon which it is based.

The contextual hierarchy



Hypothesis:

“Contextual Meaning” approaches “Meaning” as Corpus Size approaches infinity

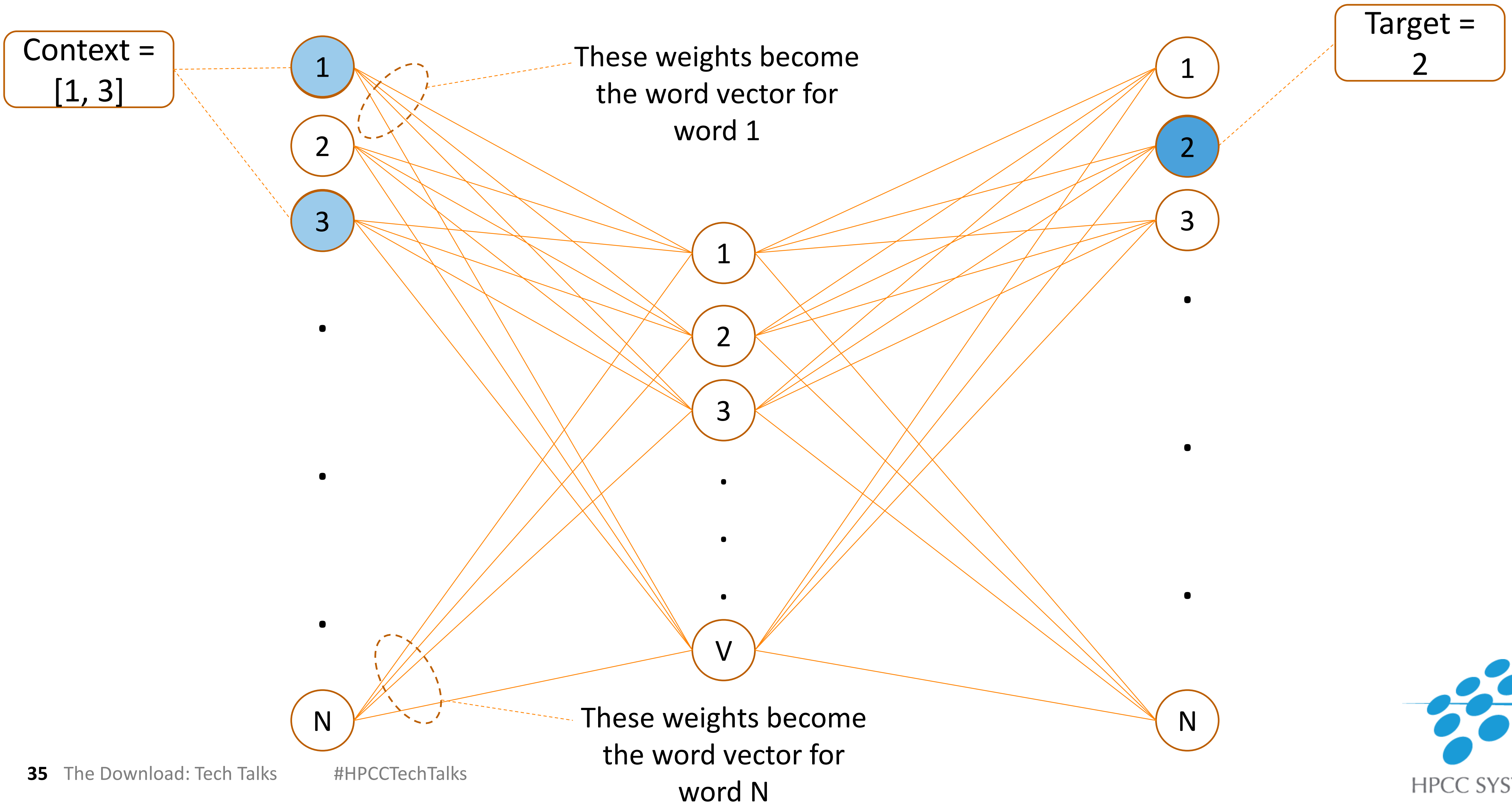
How do text vectors work?

- Vectors are best thought of as Coordinates in Space
  - A 2D vector [1.5, -3.2] can represent a coordinate in 2D space
  - A 3D vector [-.35, 1.2, 125.4] can represent a coordinate in 3D space
  - An N-Dimensional vector can represent a coordinate in ND space
- Text vectors are typically between 20 and 1000 dimensional.
- To create good text vectors, we only need to find the coordinates for each word so that it is close to all words with similar meaning and distant from all words with dissimilar meaning.
- This is an optimization problem.

Optimizing Text Vectors

HPCC SYSTEMS®

Continuous Bag of Words (CBOW) algorithm



HPCC SYSTEMS®

N-Grams

- N-Grams are combinations of words that imply different meaning than that of the individual words:
  - New York Times
  - High Performance Computing Cluster
  - Traffic Light
- Order of words in N-Grams is significant
- N-Grams may be sequences of any length
  - Unigram – A single word
  - Bigram – Two word sequence
  - Trigram – Three work sequence
- Note: Using e.g., Trigrams usually also includes Bigrams and Unigrams.

Case Study

- Anonymized public records – Violation Descriptions for every legal violation occurring within several US states.
- Violation Descriptions are entered by hand by clerks at 1000s of different courts.
  - Terse
  - Free-form
  - Many non-standardized abbreviations
  - Frequent typos and mis-spellings
- One million different Violation Descriptions.
- In a given year, approximately 300,000 new Violation Descriptions are seen.
- Vocabulary of over 16,000 Unigram words, 100,000 Trigram words.

HPCC SYSTEMS®

Case Study Results

- Training took ~40 minutes on a 20 node HPCC Cluster.
- We identified a set of interesting words in the corpus and asked TextVectors for the closest words in meaning:

text	closest
	Item
dog	dogs,cat,k9,animal,canine
boat	motorboat,mb,canoe,aircraft,vessel
speeding	speding,speed,spd,speedng,speedig
light	lgt,ligh,lght,lights,lamp
vehicle	veh,vehicl,vehic,vechicle,vechile
accident	accid,acc,scene,wl,crash
fish	fishing,trout,clams,creel,stocked

HPCC SYSTEMS®

Case Study Results -- continued

- We selected a small set of words and asked TextVectors to rate them by similarity to each word.

text	closest
	Item
dog	k9,animal,canine,fish,trout
boat	canoe,vessel,vehicle,crash,car
speeding	speed,sp,spedding,reckless,crash
light	brake,mirror,vessel,canine,vehicle
vehicle	car,canoe,vessel,boat,accident
accident	acc,crash,brake,vehicle,rd
fish	trout,bass,k9,dog,vessel

HPCC SYSTEMS®

Case Study Results -- continued

- We asked TextVectors to identify anomalous words within a set of words. We gave it the set:
  - dog, cat, canine, vehicle, terrier, animal, reckless
  - We asked for the two most anomalous words

id	text
4	vehicle
8	reckless

HPCC SYSTEMS®

Case Study Results -- continued

- We provided a set of sentences that were never seen in the training data and asked TextVectors to identify the closest sentences from the training data:

text	closest	similarity
	Item	Item
bicycles to ride to the right	fail to ride bike to the right,fail to ride to the right side	0.9739537835121155,0.9556015729904175
belt vio passenger	safety belt vio passenger,seat belt violation passenger	0.9506689310073853,0.920316755771637
burn trash	burn debris waste,burn rubbish waste	0.8604044318199158,0.8558459877967834
crash no proof of insurance	no proof of liability insurance,no proof of insurance scene	0.9724310040473938,0.9707409739494324
defect tail light pass side	defective pass side tail light,defective tail light pass side	0.987580418586731,0.987580418586731
driev w 2 earbuds	dr w 2 earphone,dr w 2 earphones	0.8863240480422974,0.8565295934677124
fail to yield fro stat emeg	fail to yield stat emer vhle,fail to yld stat emerg vhl	0.9484548568725586,0.9448829889297485
fictitious id to purchase alco	fake id to purchase alcohol,possess fict id to purch alco	0.9547269940376282,0.9437414407730103
firearm shoot in veh	discharge firearm in vehicle,disch firearm while in veh	0.9384040236473083,0.9299135208129883
going wrong way bicycle	ride bicycle wrong way one way,ride bicycle wrong way one way	0.9243994355201721,0.9243994355201721
floodway area allow encroachm	allow animal in roadway,rudee rocks unsafe area	0.7987234592437744,0.7928654551506042

Case Study Results -- continued

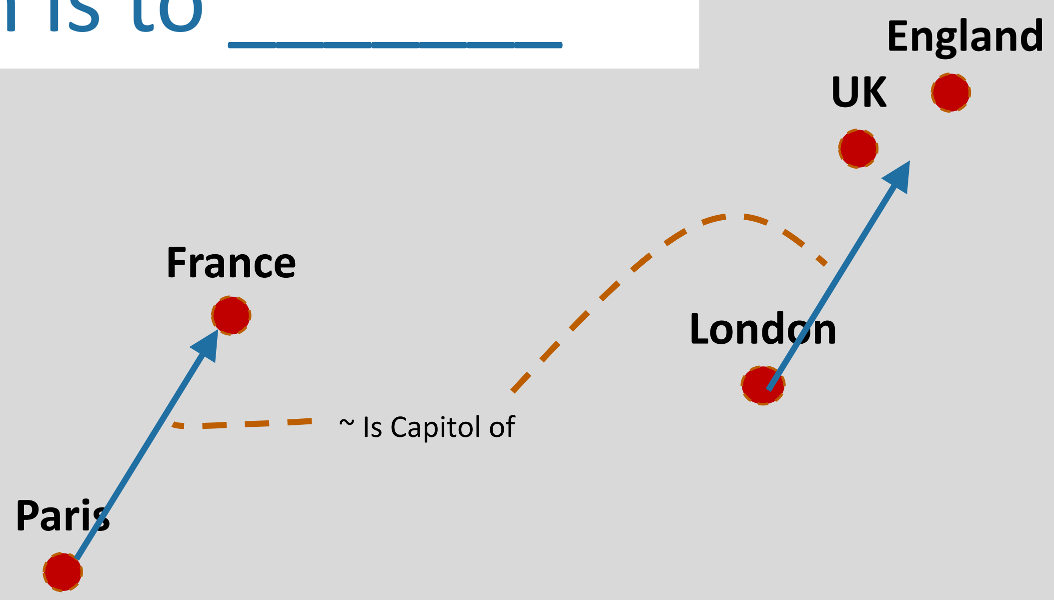
- We had a hard time finding a good analogy to solve, but this one seemed reasonable:

text	closest	similarity
	Item	Item
fishing is to trout as hunting is to:	hunt,waterfowl,bait,birds,spotlight	0.827839588165283,0.716

HPCC SYSTEMS®

Word Analogies with TextVectors

Paris is to France as London is to ________



vec(Paris) – vec(France) + vec(London) ~= [UK, England]

HPCC SYSTEMS®

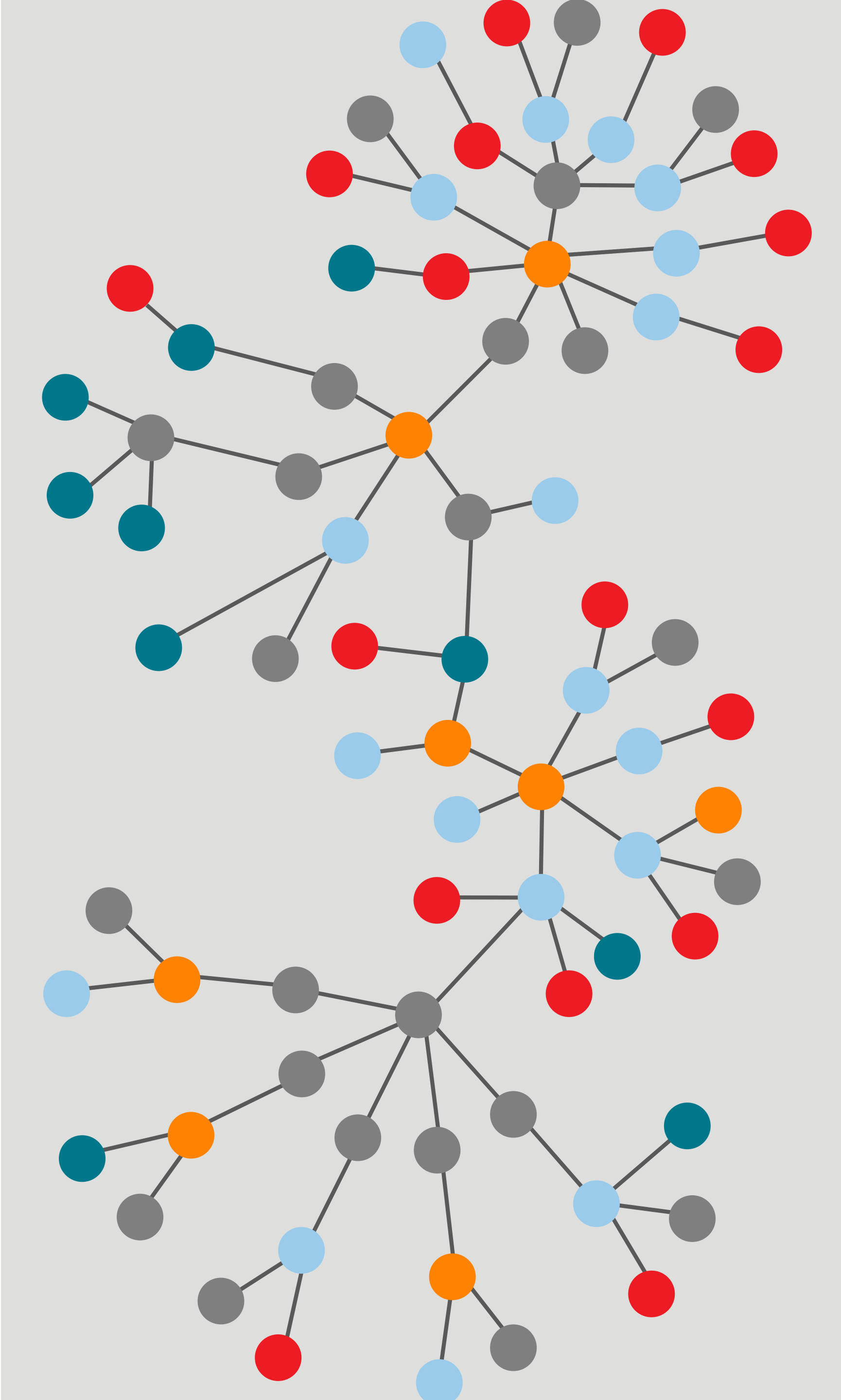
New areas for exploration

- I believe we are just scratching the surface with application of this type of technology.
  - Can semantic relationships be discovered as (word2 – word1)?
  - Can we uncover word hierarchies e.g., Animal -> Mammal -> Carnivore -> Canine?
  - Is there a way to standardize word vectors so that pre-computed vectors can be combined with contextual local meanings

Quick poll:

Do you think Text Vectors might be useful for your projects?

See poll on bottom of presentation screen



HPCC SYSTEMS®

Closing

- Thank you for attending.
- Feel free to contact me if you have projects where TextVectors could be helpful.
  - Roger.Dev@LexisNexisRisk.com
- For more information:
  - TextVectors blog article
    - https://hpccsystems.com/blog/TextVectors
  - All my blog articles:
    - https://hpccsystems.com/blogs-rogerdev

Questions?

Roger Dev
Sr Architect
LexisNexis® Risk Solutions
roger.dev@lexisnexisrisk.com

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

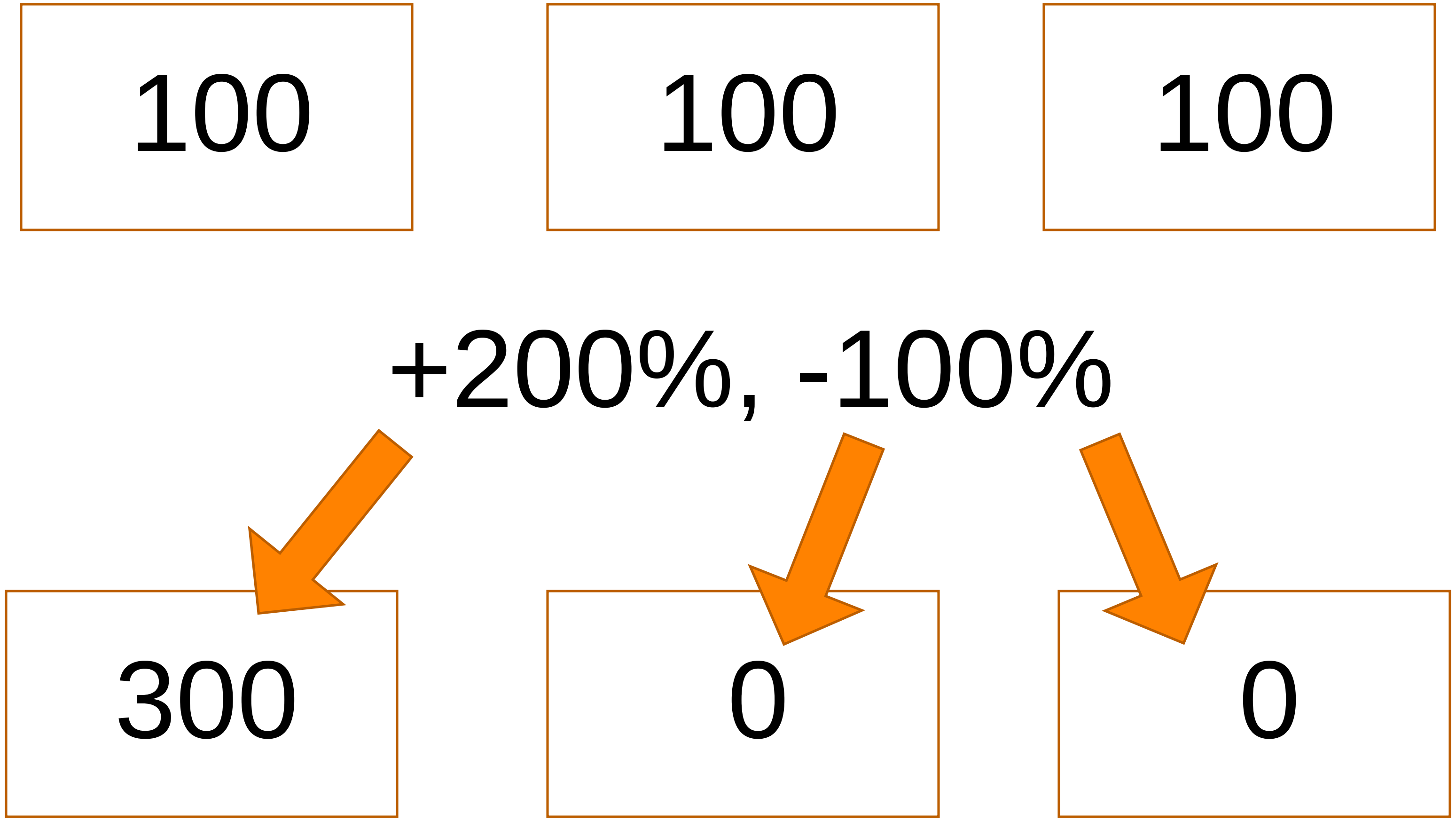
ECL Tip Part I: DISTRIBUTE

HPCC SYSTEMS®

Bob Foreman
Senior Software Engineer
LexisNexis Risk Solutions

LexisNexis® RISK SOLUTIONS

Cluster Skew



HPCC SYSTEMS®

DISTRIBUTE Function

The **DISTRIBUTE** function distributes records from the *recordset* across the nodes of the target cluster based on the specified *expression*. All records for which the *expression* evaluates the same end up on the same nodes.

Following the distribution process, all subsequent operations should be optimized by using LOCAL operation.

There are four types of DISTRIBUTE methods:

1. Random
2. Expression
3. Index
4. Skew

HPCC SYSTEMS®

DISTRIBUTE Methods

1. **"Random" DISTRIBUTE**

DISTRIBUTE(*recordset* **)**

This form redistributes the *recordset* "randomly" so there is no data skew across nodes, but without the disadvantages the RANDOM() function could introduce. This is functionally equivalent to **distributing by a hash of the entire record**.

2. **Expression DISTRIBUTE**

DISTRIBUTE(*recordset, expression* **)**

This form redistributes the *recordset* based on the specified *expression,* typically one of the HASH functions. Only the bottom 32-bits of the *expression* value are used, so either HASH or HASH32 are the optimal choices. Records for which the *expression* evaluates the same will end up on the same node. DISTRIBUTE implicitly performs a modulus operation if an *expression* value is not in the range of the number of nodes available. If the MERGE option is specified, the *recordset* must have been locally sorted by the *sorts* expressions. This avoids resorting.

HASH Functions

HASH(*expressionlist***)**
HASH32(*expressionlist***)**
HASH64(*expressionlist***)**
HASHCRC(*expressionlist***)**
HASHMD5(*expressionlist***)**

expressionlist – A comma-delimited list of values.

The **HASH** functions all return a hash value derived from all the values in the *expressionlist*.

```
Domains_Dist := DISTRIBUTE(Domains_Seq, HASH(zip, TRIM(prim_name), prim_range));
YP_Cont_Dist := DISTRIBUTE(YellowPages_Contacts,HASH32(TRIM(company_name),
                                                       TRIM(lname), zip));
```


HPCC SYSTEMS®

DISTRIBUTE Methods

Index-based DISTRIBUTE

DISTRIBUTE(*recordset, index* **[**, *joincondition* **])**

This form redistributes the *recordset* based on the existing distribution of the specified *index*, where the linkage between the two is determined by the *joincondition*. Records for which the *joincondition* is true will end up on the same node.

Skew-based DISTRIBUTE

DISTRIBUTE(*recordset,* **SKEW(** *maxskew* **[**, *skewlimit* **]))**

This form redistributes the *recordset,* but only if necessary. The purpose of this form is to replace the use of DISTRIBUTE(*recordset*,RANDOM()) to simply obtain a relatively even distribution of data across the nodes. This form will always try to minimize the amount of data redistributed between the nodes.

The skew of a dataset is calculated as:

```
MAX(ABS(AvgPartSize-PartSize[node])/AvgPartSize)
```

If the *recordset* is skewed less than *maxskew* then the DISTRIBUTE is a no-op. If *skewlimit* is specified and the skew on any node exceeds this, the job fails with an error message (specifying the first node number exceeding the limit), otherwise the data is redistributed to ensure that the data is distributed with less skew than *maxskew*.

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

ECL Tip Part II: Leveraging the Power of HPCC Systems? Use AGGREGATE.

Allan Wrobel
Consulting Software Engineer
LexisNexis Risk Solutions

HPCC SYSTEMS®

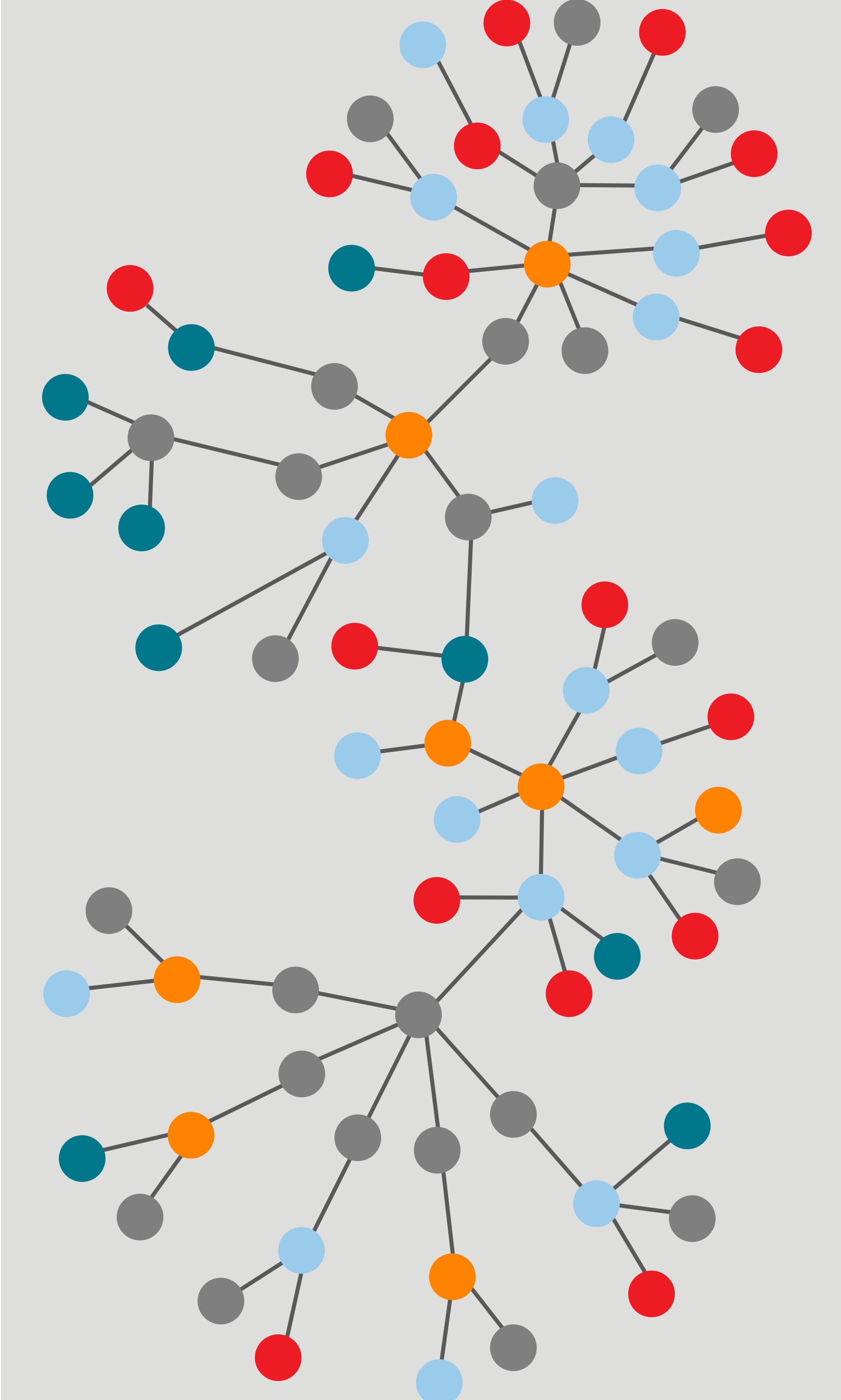
LexisNexis® RISK SOLUTIONS

Quick poll:

Do you already use AGGREGATE?

Do you see AGGREGATE as a ‘complex’ Built-in?

See poll on bottom of presentation screen



HPCC SYSTEMS®

AGGREGATE: The 2nd ‘Merge’ TRANSFORM

- 1st Iteration

LEFT
Gender: ‘F’
Calls: ‘’

RIGHT
Gender: ‘F’
Calls: ‘’

```
RTbl MergePhase(RTbl L,RTbl R) := TRANSFORM
   SELF.Calls   := L.Calls + L.Gender + CASE(LENGTH(L.Calls), 0 => '1',2 => '2',4 => '3','4');
   SELF         := L;
END;
```

- Result

SELF
Gender: ‘F’
Calls: ‘F1’

HPCC SYSTEMS®

AGGREGATE: The 2nd 'Merge' TRANSFORM

- 2nd Iteration

LEFT
Gender: 'F'
Calls: 'F1'

RIGHT
Gender: 'F'
Calls: ''

```
RTbl MergePhase(RTbl L,RTbl R) := TRANSFORM
   SELF.Calls  := L.Calls + L.Gender + CASE(LENGTH(L.Calls), 0 => '1',2 => '2',4 => '3','4');
   SELF        := L;
END;
```

- Result

SELF
Gender: 'F'
Calls: 'F1F2'

HPCC SYSTEMS®

Quick poll:

Has AGGREGATE been demystified for you?

See poll on bottom of presentation screen

HPCC SYSTEMS®

Questions?

Allan Wrobel
Consulting Software Engineer
LexisNexis® Risk Solutions
allan.wrobel@lexisnexis.com

Submit a talk for an upcoming episode!

- Have a new success story to share?
- Want to pitch a new use case?
- Have a new HPCC Systems application you want to demo?
- Want to share some helpful ECL tips and sample code?
- Have a new suggestion for the roadmap?
- Be a featured speaker for an upcoming episode! Email your idea to Techtalks@hpccsystems.com
- Visit The Download Tech Talks wiki for more information: https://wiki.hpccsystems.com/display/hpcc/HPCC+Systems+Tech+Talks

Watch for details to register for our next Tech Talk on May 24!

HPCC SYSTEMS®

Thank You!

Visit our Tech Talk wiki for more information and to browse past episodes:
https://hpccsystems.com/techtalks