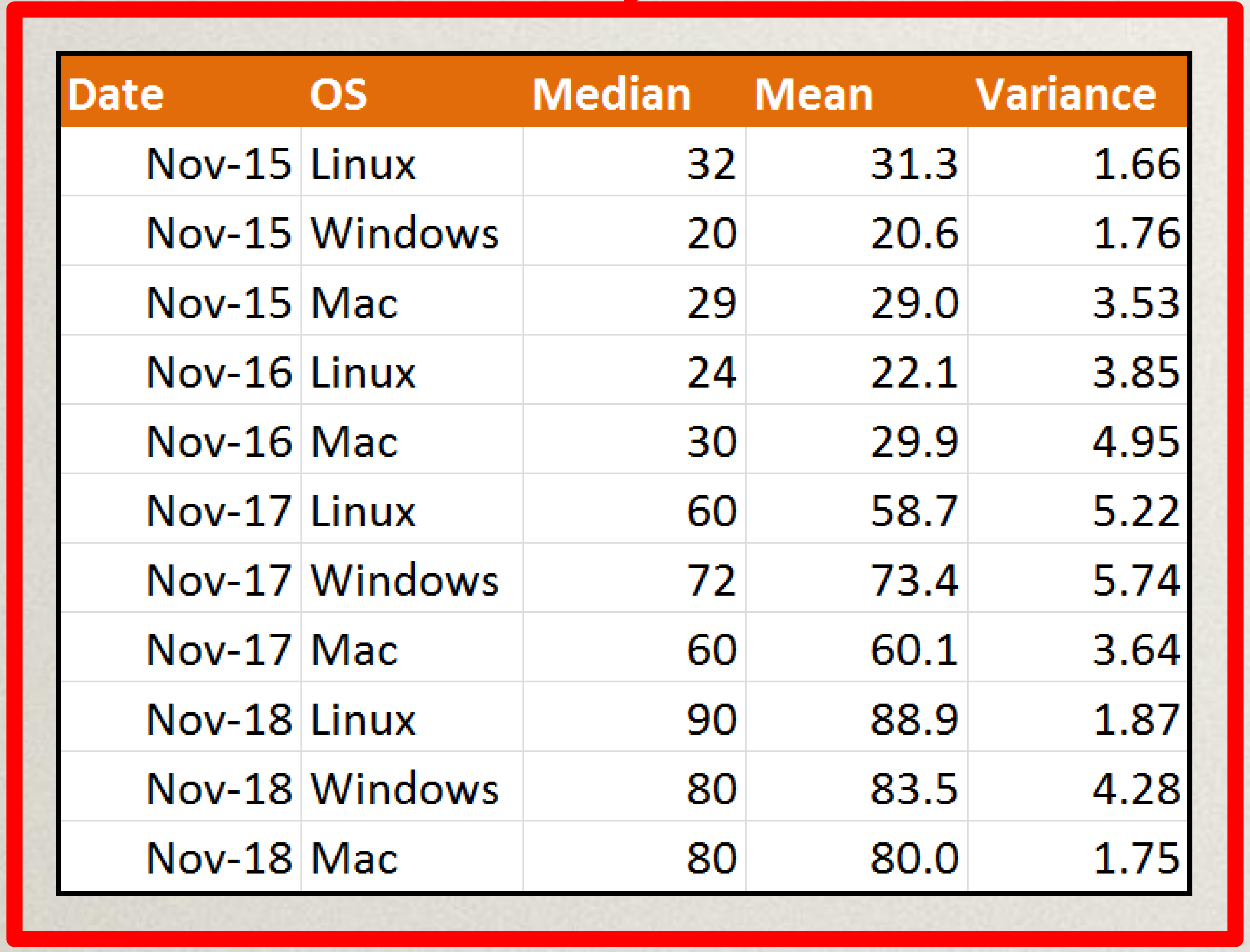
Advantages of Data Warehousing

Kyle Lahnakoski, November 2014

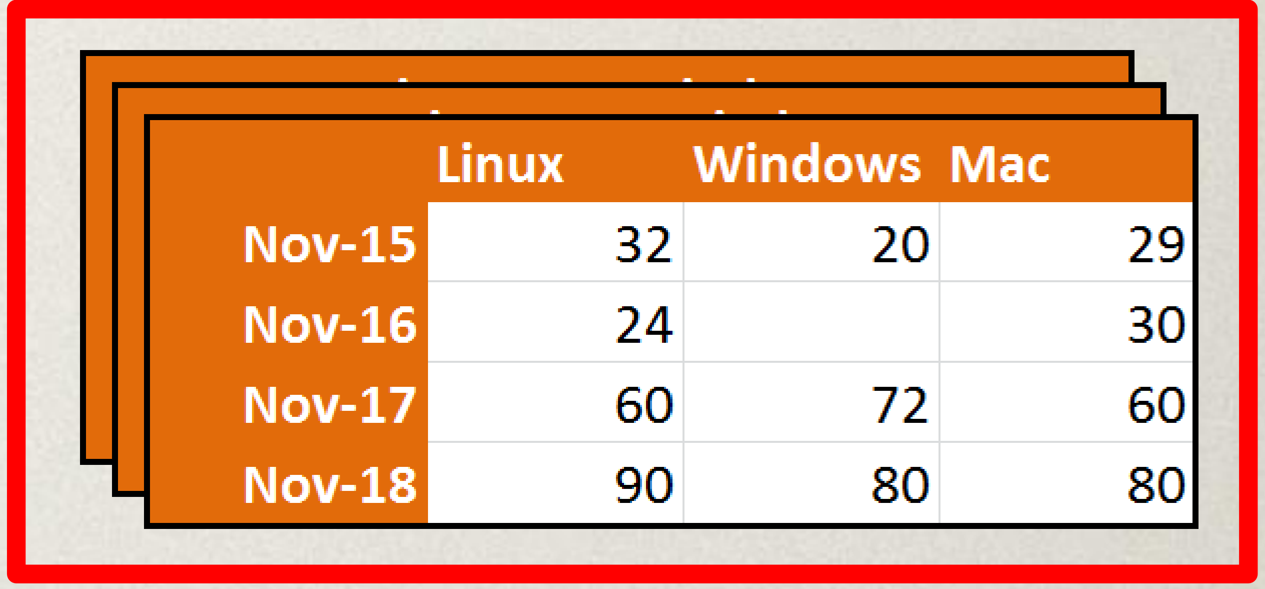
https://github.com/klahnakoski/Qb/blob/master/docs/Data%20Warehouse.pptx

Nomenclature

Table / Relation Cube / Pivot Table



Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

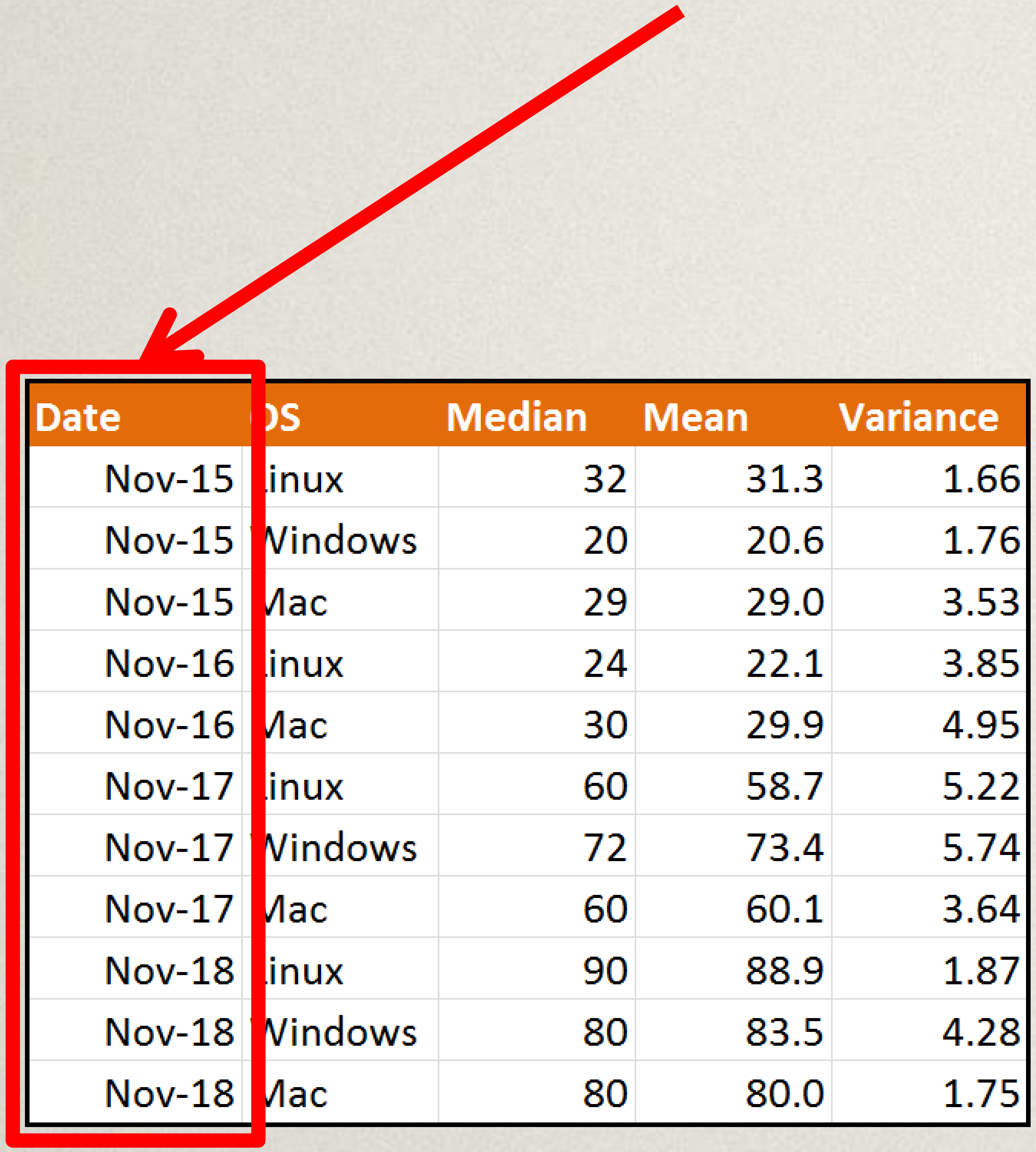


	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

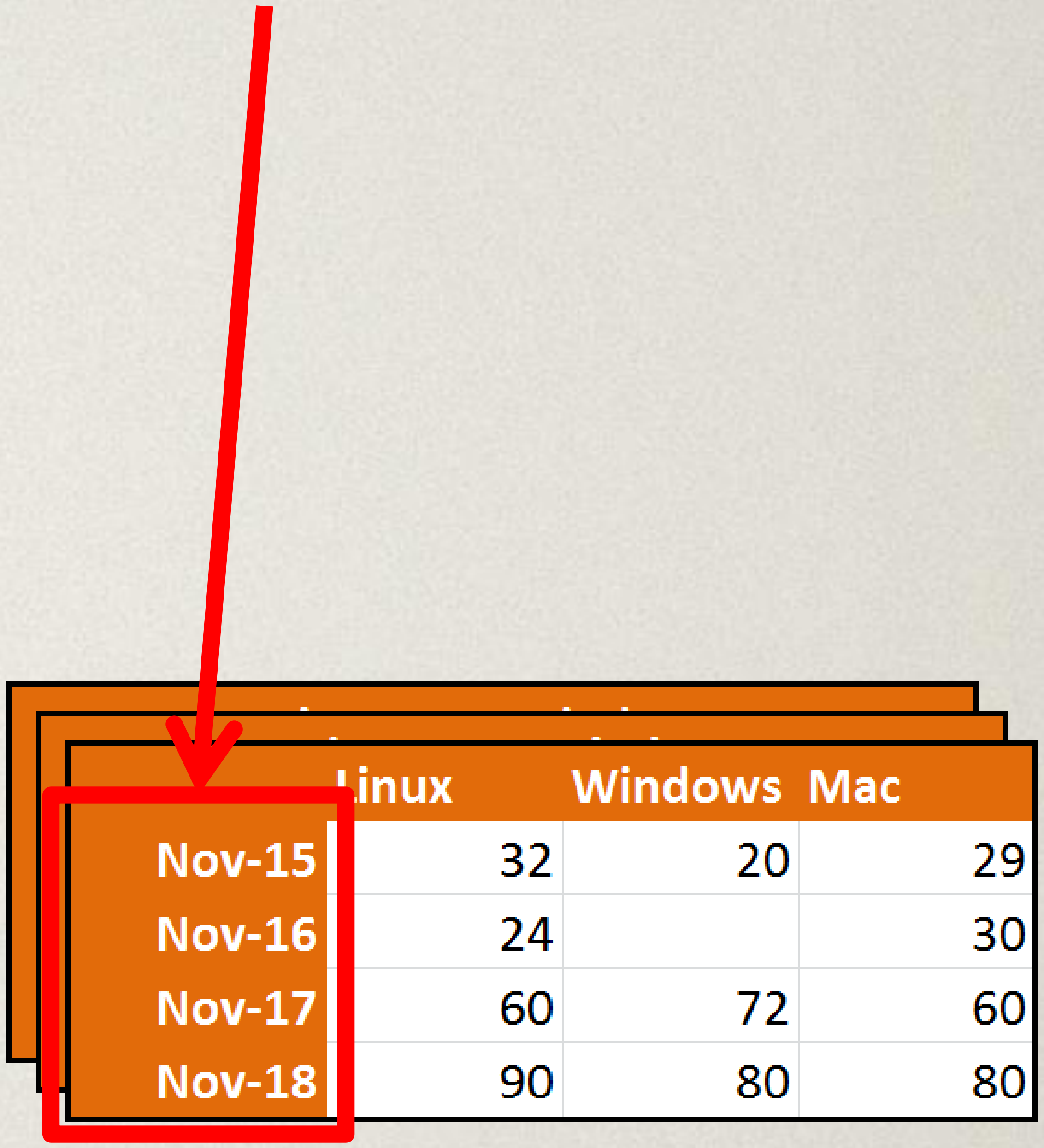
Nomenclature

Column / Attribute

Dimension

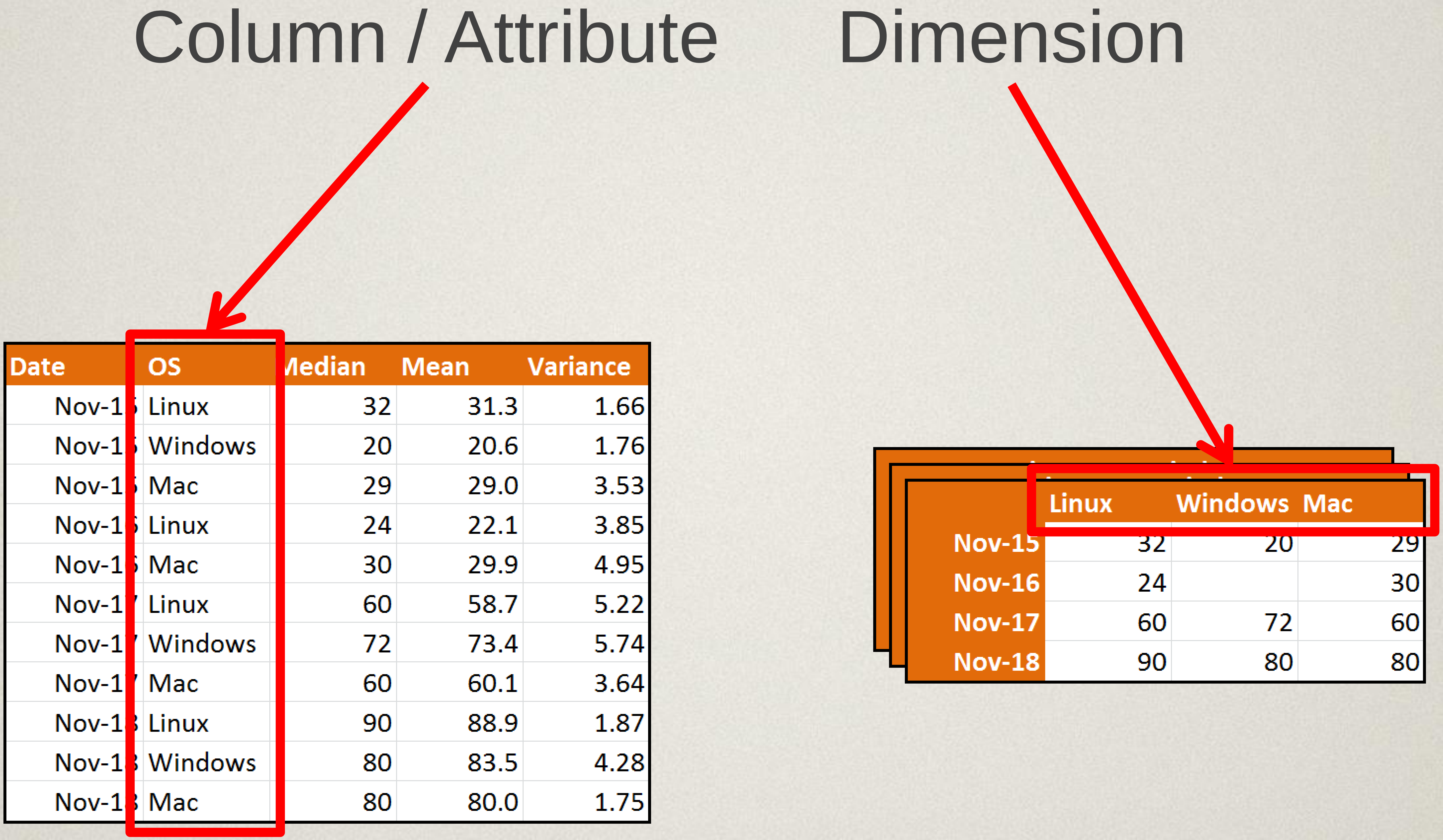


Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75



	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature



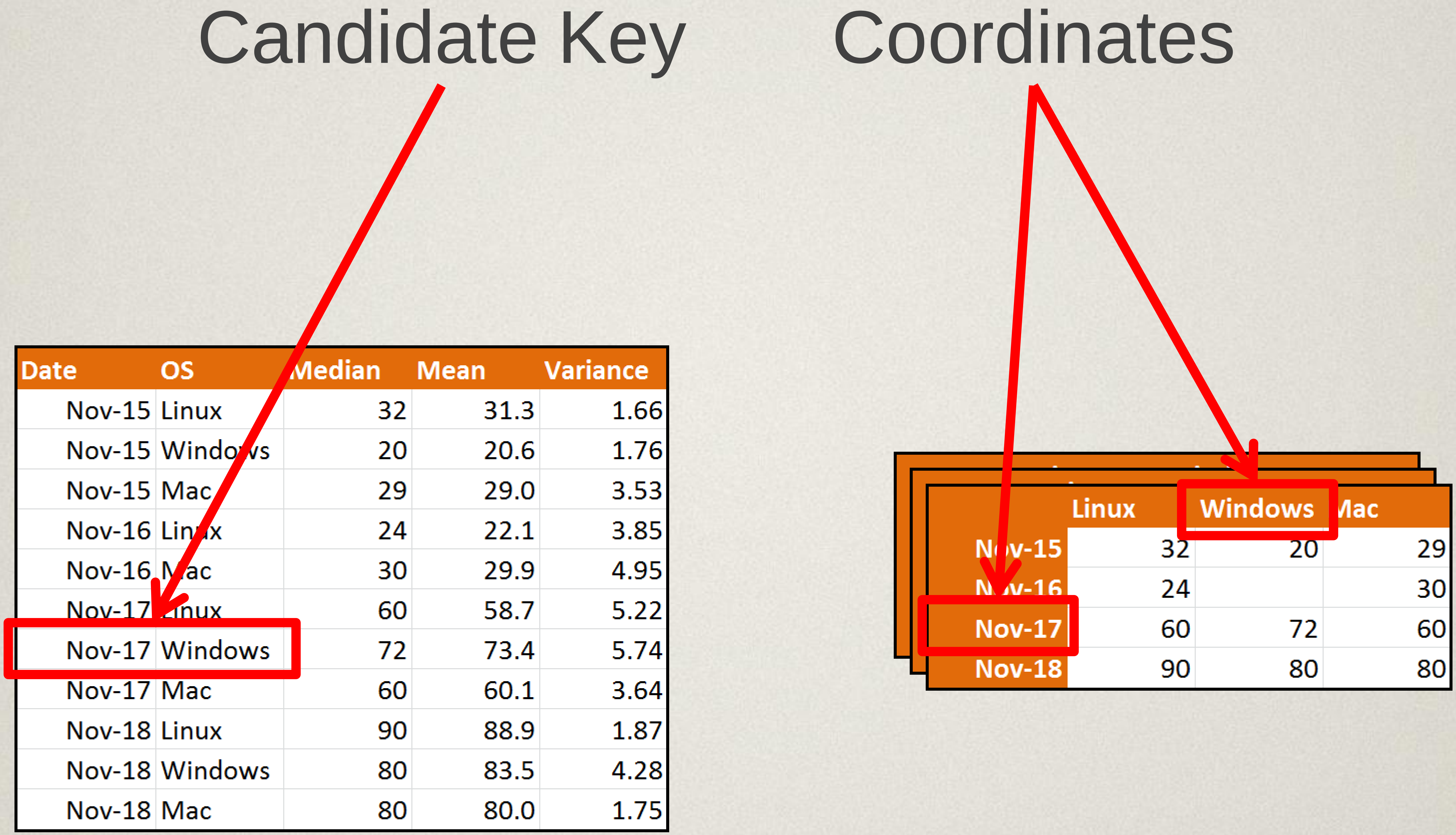
Column / Attribute

Dimension

Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature



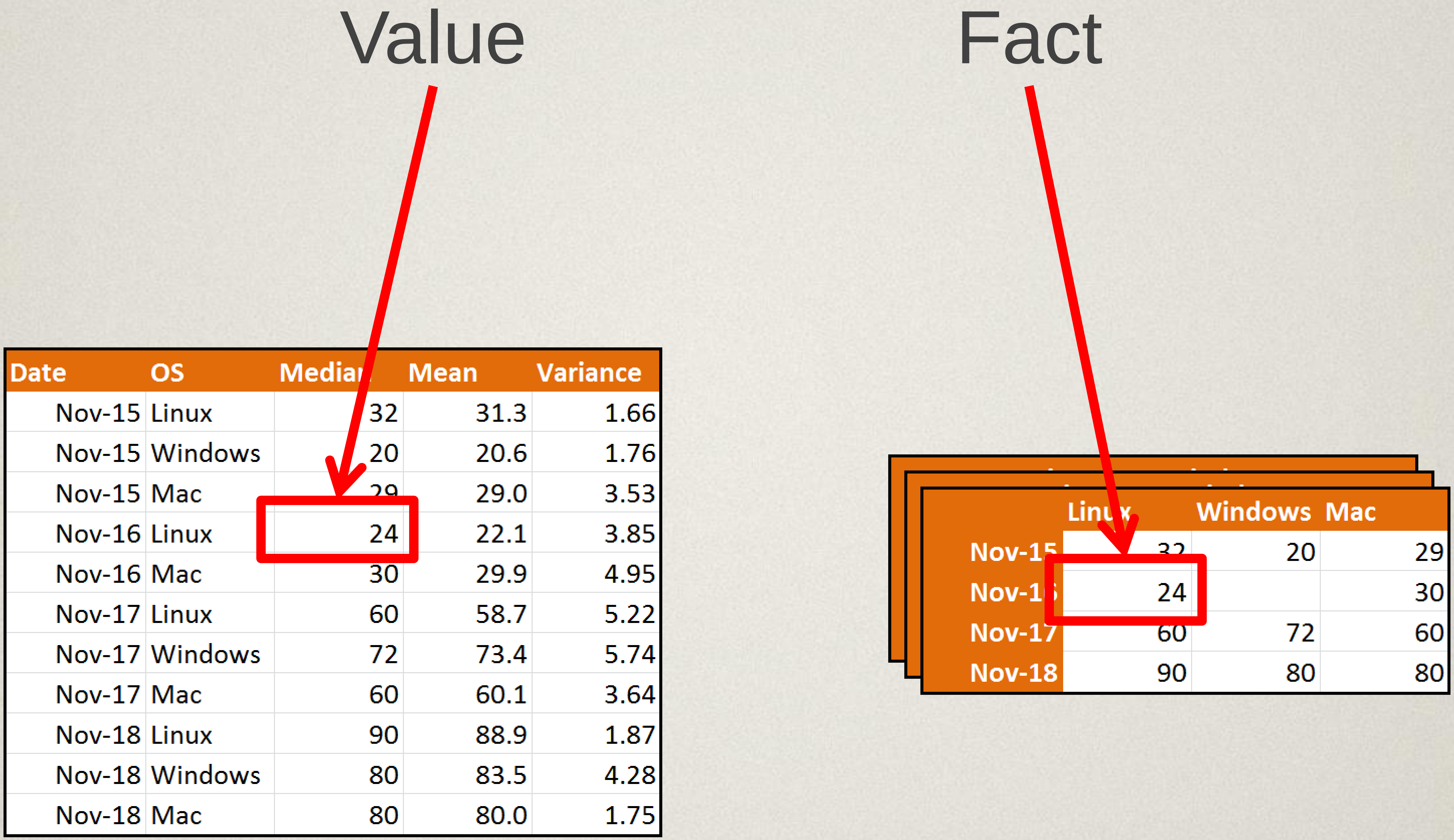
Candidate Key

Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

Coordinates

	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature



Value

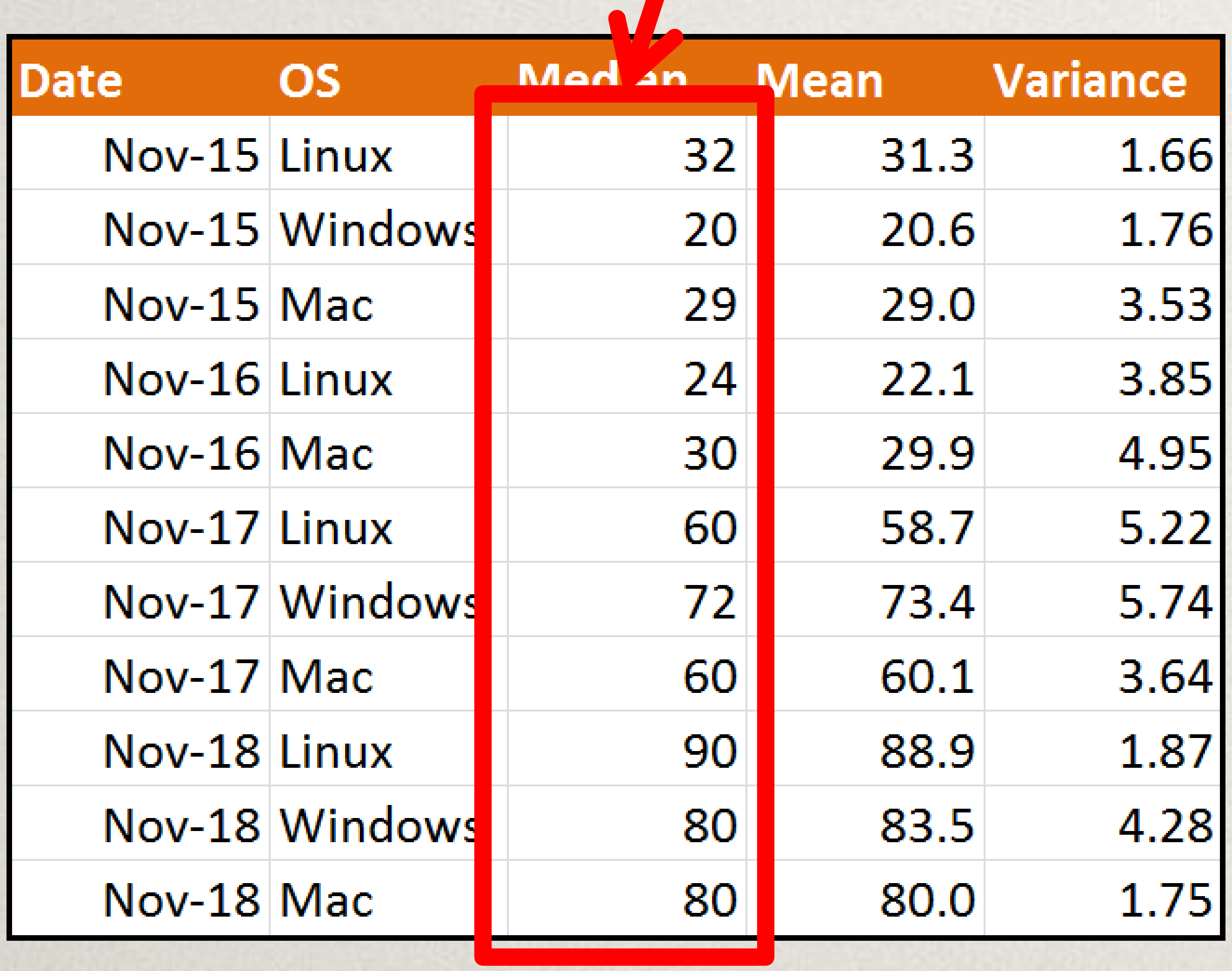
Fact

Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

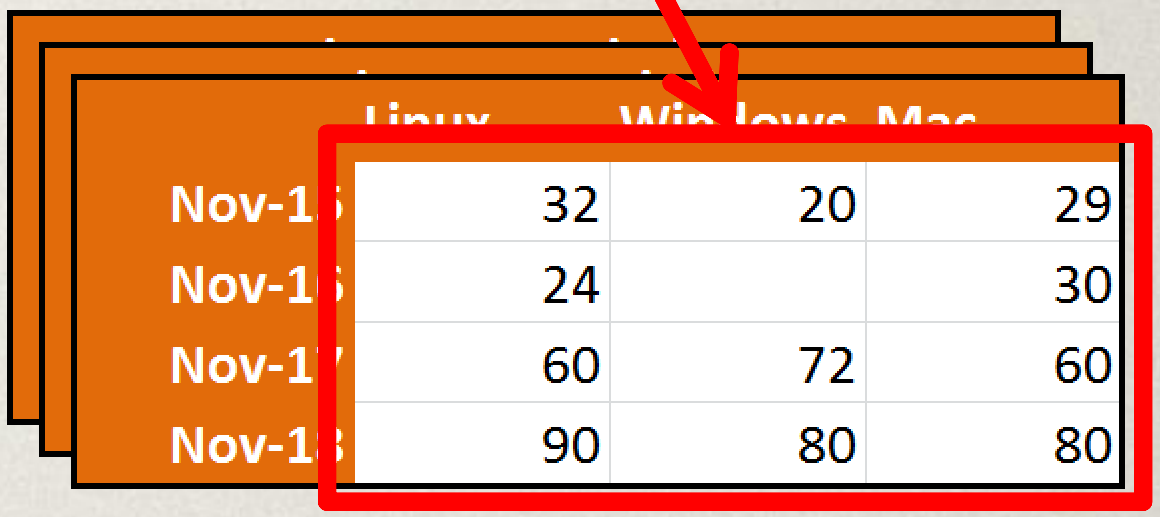
Nomenclature

Values



Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

Measure



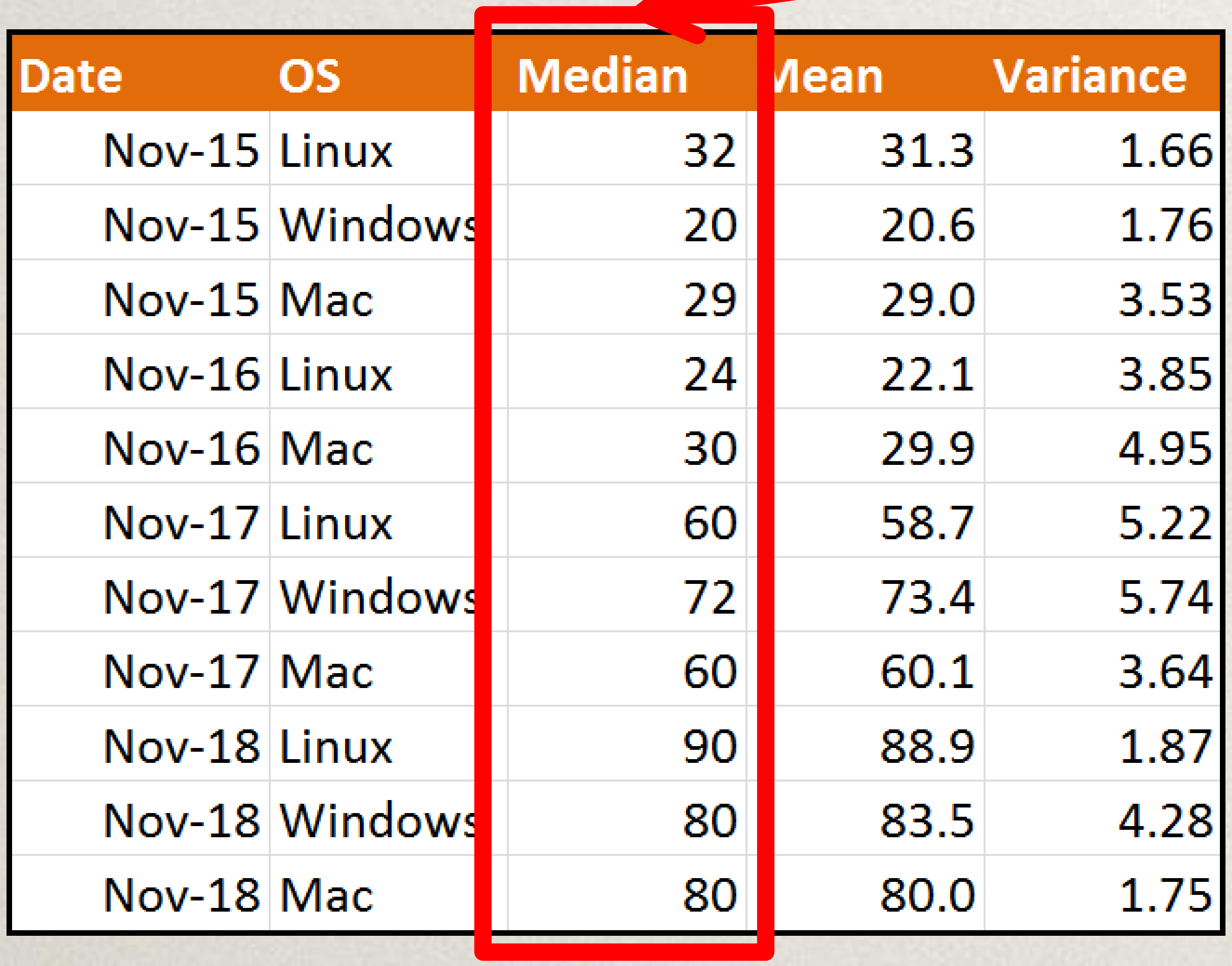
	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature

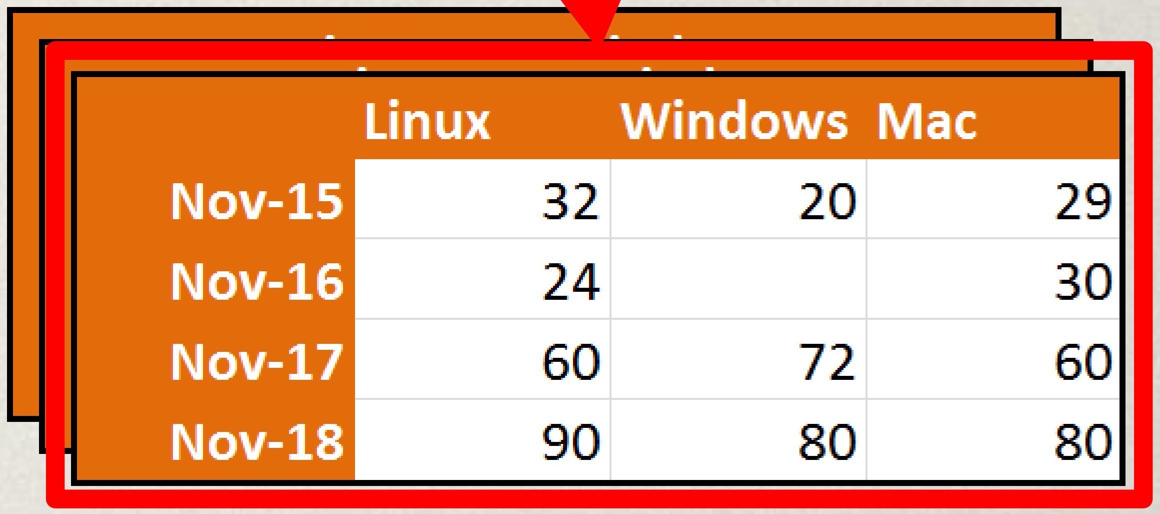
Values

Measure

Median



Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

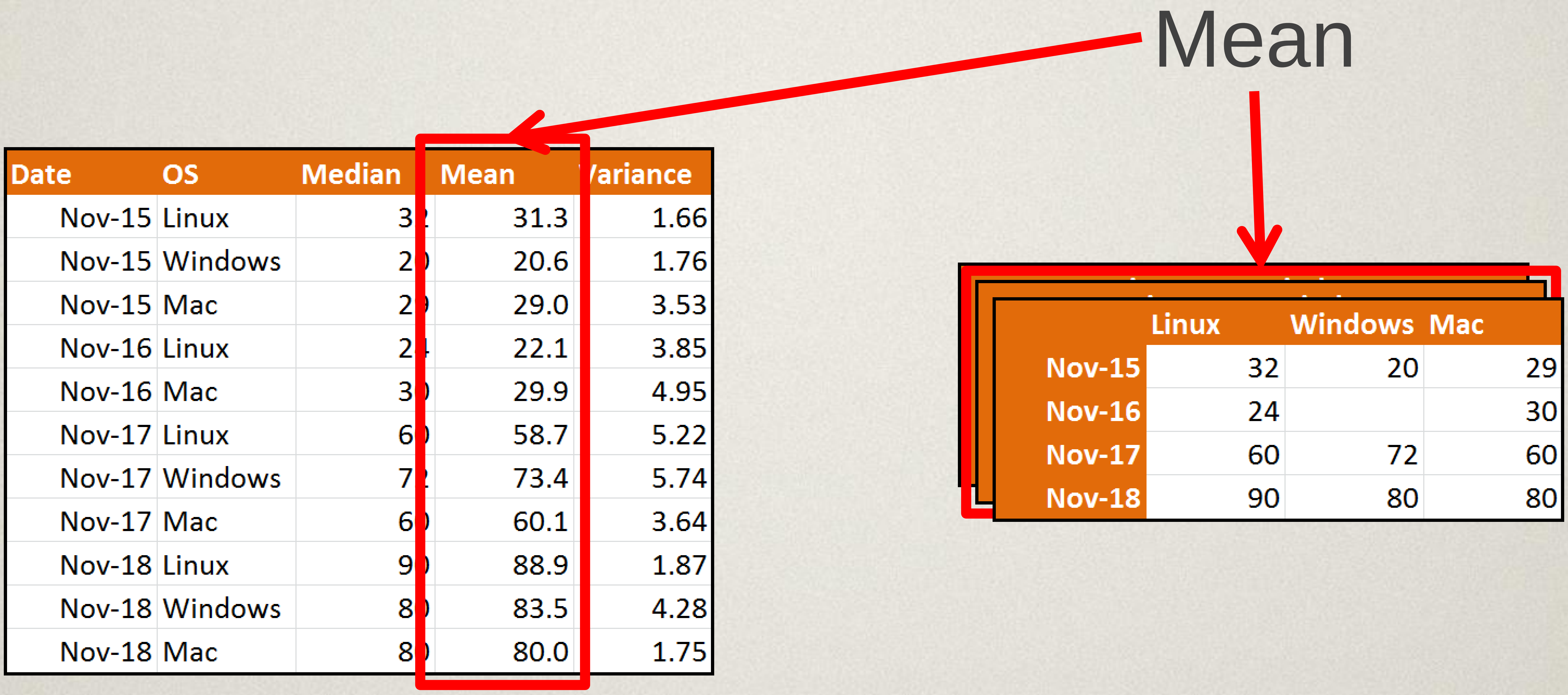


	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature

Values

Measure



Mean

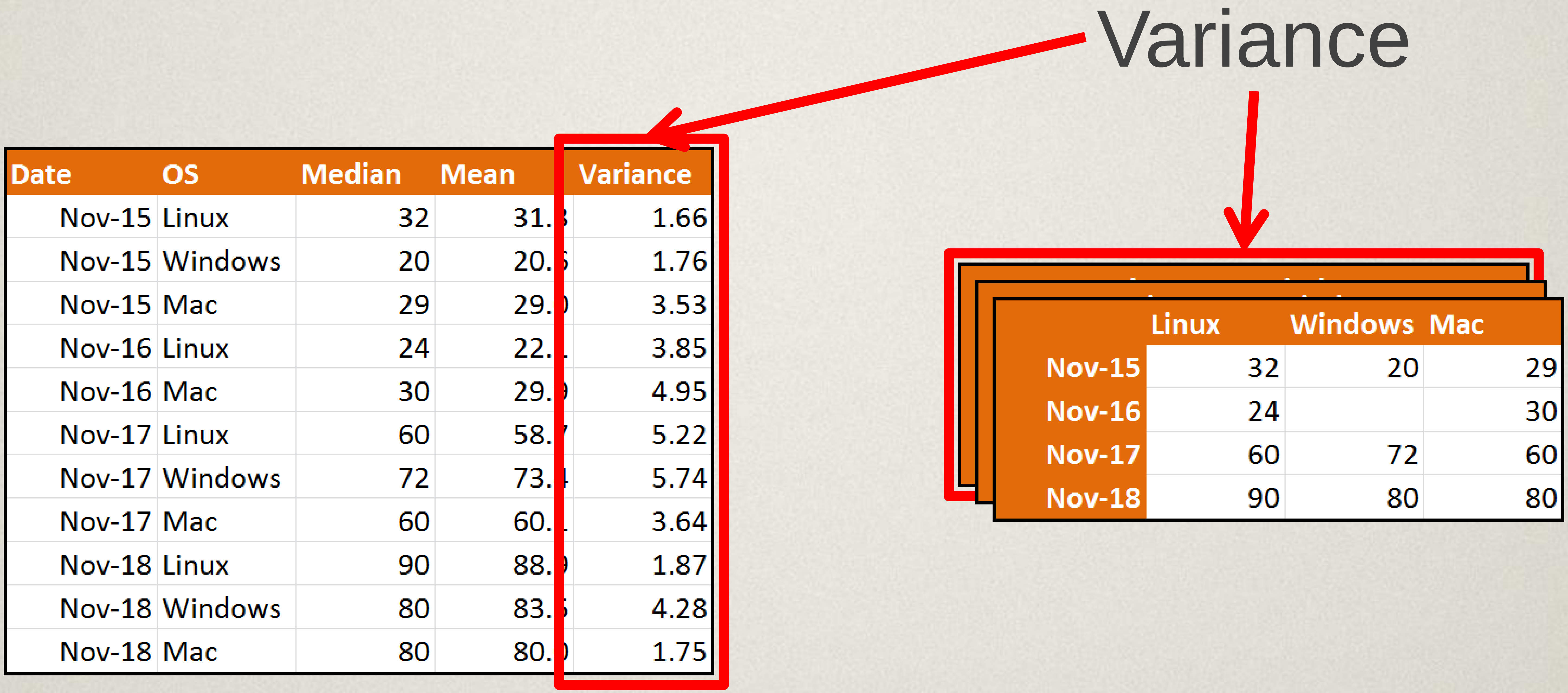
Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Nomenclature

Values

Measure



Variance

Date	OS	Median	Mean	Variance
Nov-15	Linux	32	31.3	1.66
Nov-15	Windows	20	20.6	1.76
Nov-15	Mac	29	29.0	3.53
Nov-16	Linux	24	22.1	3.85
Nov-16	Mac	30	29.9	4.95
Nov-17	Linux	60	58.7	5.22
Nov-17	Windows	72	73.4	5.74
Nov-17	Mac	60	60.1	3.64
Nov-18	Linux	90	88.9	1.87
Nov-18	Windows	80	83.5	4.28
Nov-18	Mac	80	80.0	1.75

	Linux	Windows	Mac
Nov-15	32	20	29
Nov-16	24		30
Nov-17	60	72	60
Nov-18	90	80	80

Distinctive Features of DW

- Fast* filtering (fast “slicing”)
- Fast* aggregates
- API is a query language (SQL, MDX)
- A service, open to third party clients
- Uniform, Cartesian space of values
- Metadata on dimensions and measures
- Defines a standard for ETL
- Has a security model

* Virtually O(1)

Distinctive Features

Fast Slicing and Fast Aggregates

- Data is de-normalized to avoid expensive joins
- Creates and manages multiple indexes across many dimensions for fast slicing
- Manages materialized views (pre-aggregated data) for fast aggregates

Distinctive Features

API is a Query Language

MDX (via some wire protocol)

```
SELECT
  [Measures].[Performance].[Mean] ON COLUMNS
FROM
  [Talos]
WHERE
  [OS].[Windows]
```

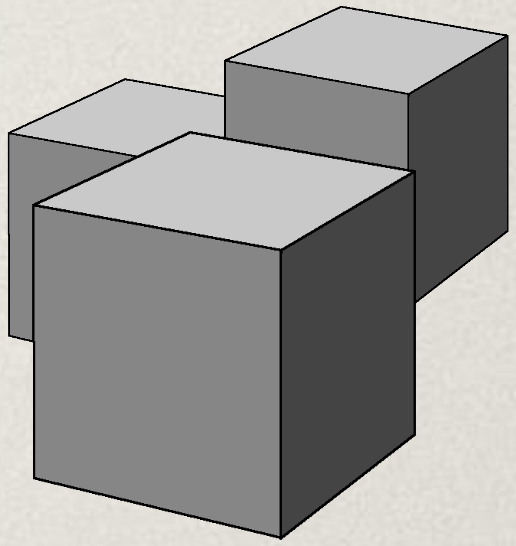
SQL (via ODBC?)

```
SELECT
  AVG(Mean) AS Mean
FROM
  Talos
GROUP BY
  OS
WHERE
  OS = “Windows”
```

Distinctive Features

API is a Query Language

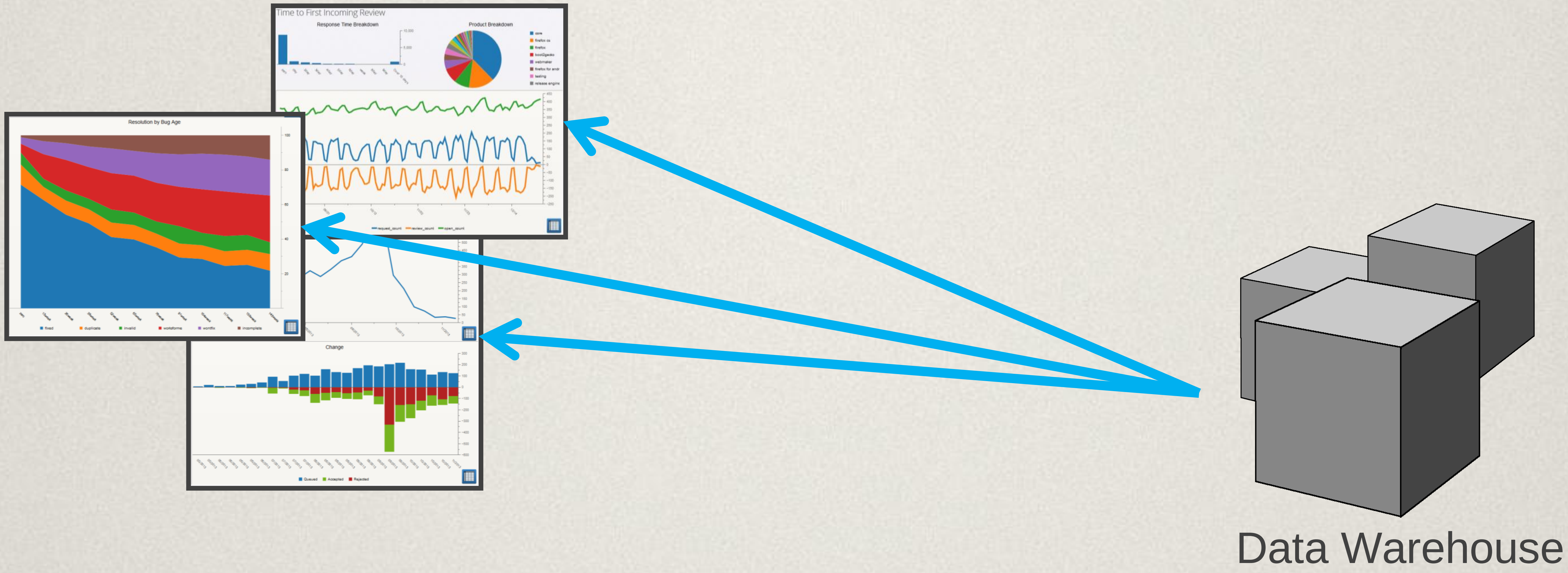
- Open to third party clients



Distinctive Features

API is a Query Language

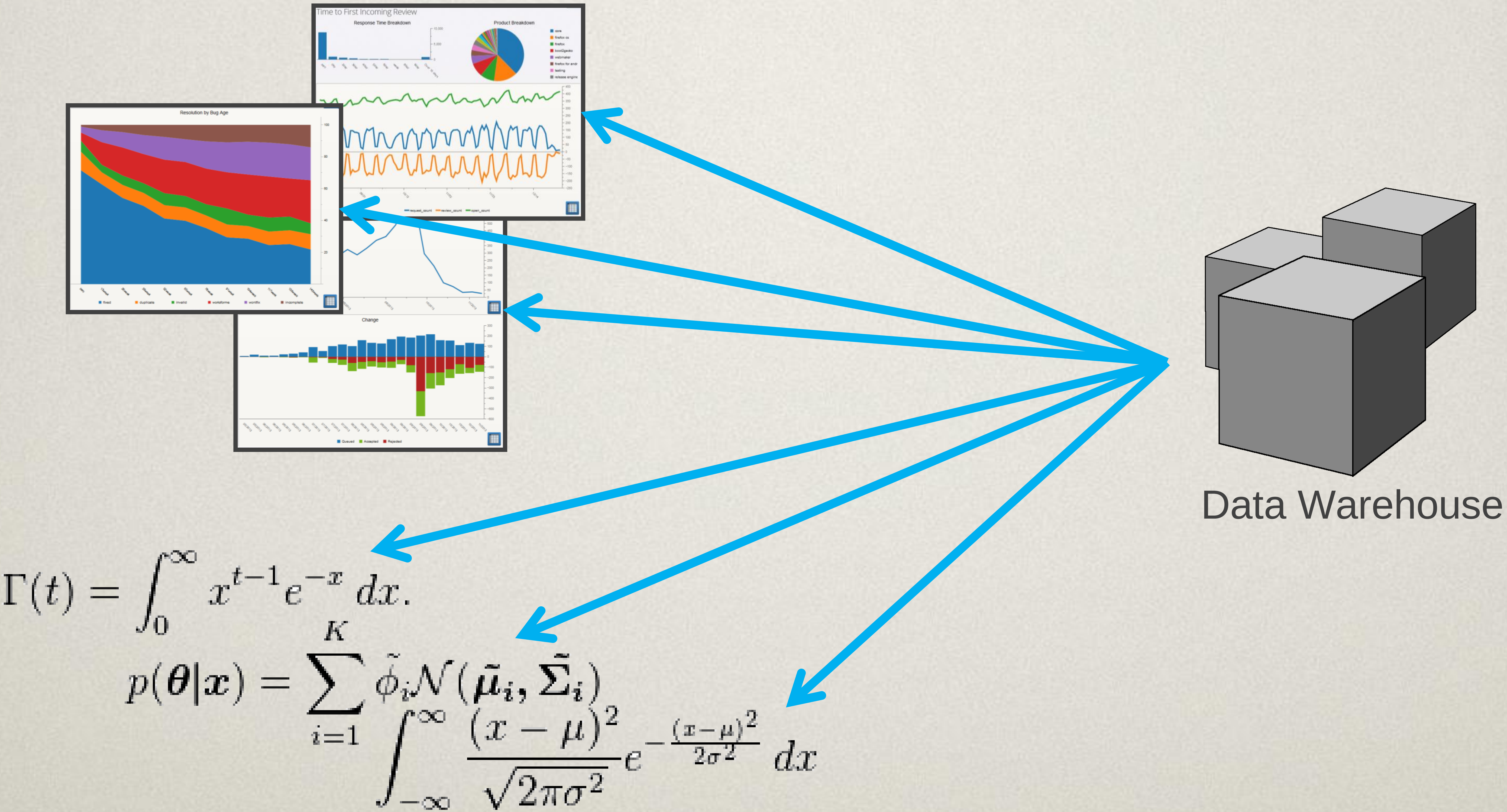
- Open to third party clients
  - Dashboards



Distinctive Features

API is a Query Language

- Open to third party clients
  - Dashboards
  - Analysis Tools



Data Warehouse

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x}\, dx.$$

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{K} \tilde{\phi}_i \mathcal{N}(\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i)$$

$$\int_{-\infty}^{\infty} \frac{(x-\mu)^2}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx$$

Distinctive Features

API is a Query Language

- Open to third party clients
- More expressive than standard RESTful APIs

```
SELECT
  *
FROM
  bugs
WHERE
  whiteboard.contains(“[js:p1]”) AND
  component.beginsWith(“javascript”)
```

SQL

```
https://bugzilla.mozilla.org/buglist.cgi?
f1=status_whiteboard&
o1=substring&resolution=---&
o2=substring&query_format=advanced&
f2=component&v1=[js%3Ap1]&
v2=javascript
```

Bugzilla

Distinctive Features

API is a Query Language

- Open to third party clients
- More expressive than standard RESTful APIs
- Saves developer from implementing query features for third party apps.

Distinctive Features

API is a Query Language

- Open to third party clients
- More expressive than standard RESTful APIs
- Saves developer from implementing query features for third party apps.

- High demand on DW service
  - No joins – upper bound on cost of a request
  - Only filter and aggregates
- Security model is required

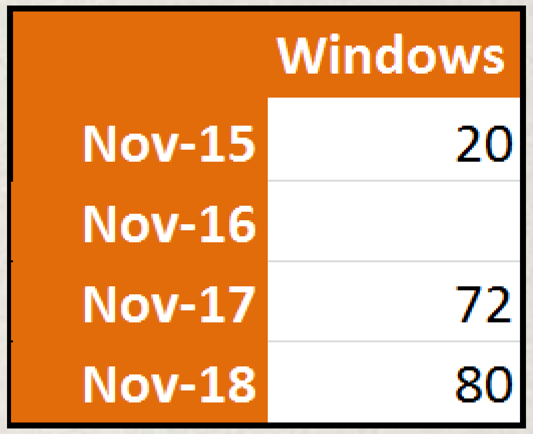
Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once

```
SELECT
   [Date] ON ROWS
   [Measures].[Median] ON COLUMNS
FROM
   [Talos]
WHERE
   [OS].[Windows]
```

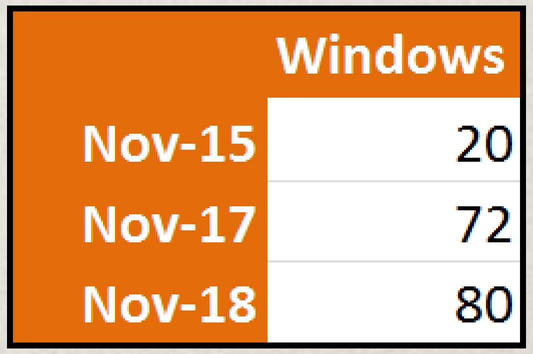
MDX



	Windows
Nov-15	20
Nov-16	
Nov-17	72
Nov-18	80

```
SELECT
  Date,
  AVG(Median) AS Median
FROM
  Talos
GROUP BY
  OS
WHERE
  OS = “Windows”
```

SQL



	Windows
Nov-15	20
Nov-17	72
Nov-18	80

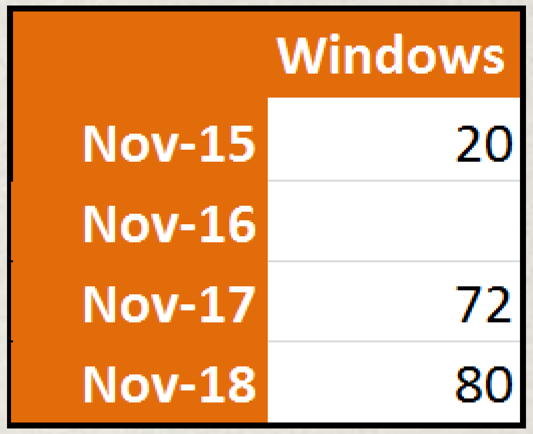
Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once

```
SELECT
  [Date] ON ROWS
  [Measures].[Median] ON COLUMNS
FROM
  [Talos]
WHERE
  [OS].[Windows]
```

MDX

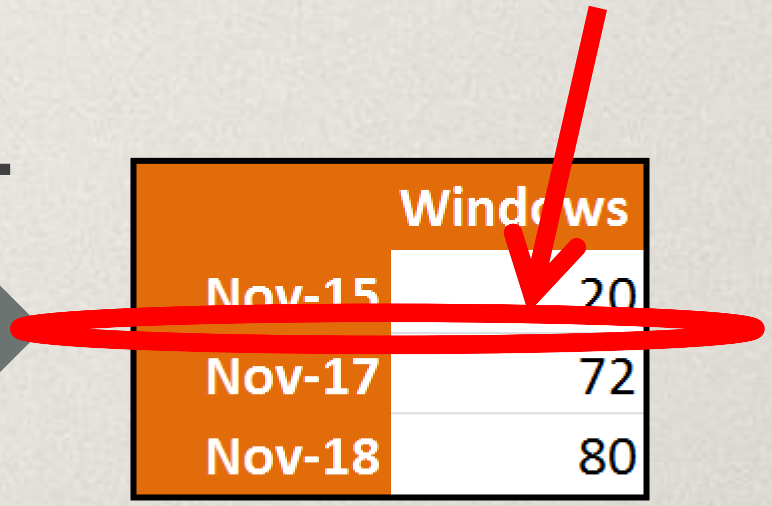


	Windows
Nov-15	20
Nov-16	
Nov-17	72
Nov-18	80

```
SELECT
  Date,
  AVG(Median) AS Median
FROM
  Talos
GROUP BY
  OS
WHERE
  OS = “Windows”
```

SQL

Where’s the 16th?



	Windows
Nov-15	20
Nov-17	72
Nov-18	80

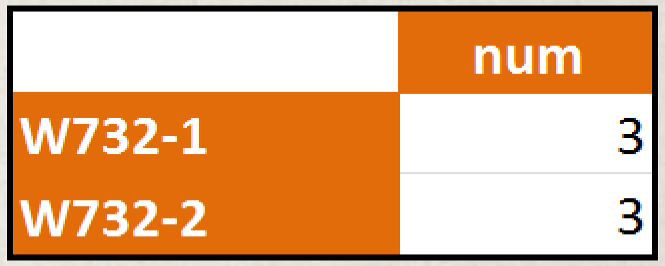
mozilla

Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once
- and only once

```
SELECT
   m.name,
   COUNT(t.Median) AS num
FROM
   Talos t
JOIN
   Machines m on m.os=t.os
GROUP BY
   m.Name
WHERE
   OS = “Windows”
```



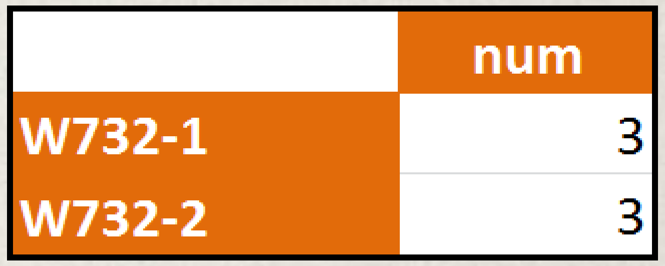
	num
W732-1	3
W732-2	3

Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once
- and only once

```
SELECT
   m.name,
   COUNT(t.Median) AS num
FROM
   Talos t
JOIN
   Machines m on m.os=t.os
GROUP BY
   m.Name
WHERE
   OS = “Windows”
```



	num
W732-1	3
W732-2	3

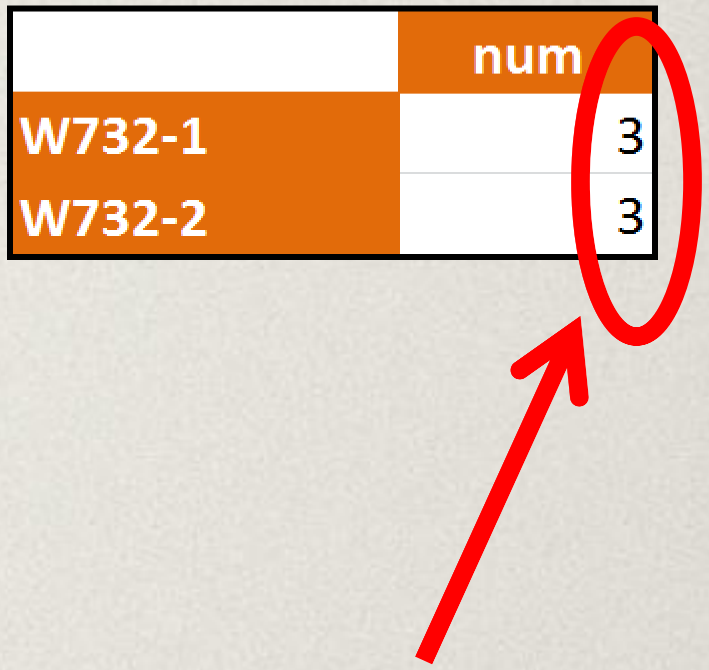
Bad logic, wrong assumption

Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once
- and only once

```
SELECT
   m.name,
   COUNT(t.Median) AS num
FROM
   Talos t
JOIN
   Machines m on m.os=t.os
GROUP BY
   m.Name
WHERE
   OS = “Windows”
```



	num
W732-1	3
W732-2	3

May conclude there are 6 tests

Distinctive Features

Clean, Cartesian spaces

- Dimension members are represented once
- and only once
- Dimensions are orthogonal (no functional dependencies)
- Important for SciPy, Pandas, R which operate on multidimensional arrays of data.

Distinctive Features

Metadata on Dimensions and Measures

- Dimensions can have sub-dimensions, type, name, natural ordering, formatting
- Measures have measurement units, default aggregation
- Extra context allows for exploration

Distinctive Features

Defines a standard for ETL

- Databases provide too much design choice:
  - You can choose to de-normalize for speed, or
  - Stay normalized for low selectivity relations.
  - Make a specific index, or
  - Write code to manage a fast aggregate.

Distinctive Features

Defines a standard for ETL

- Databases provide too much design choice:
  - You can choose to de-normalize for speed, or
  - Stay normalized for low selectivity relations.
  - Make a specific index, or
  - Write code to manage a fast aggregate.

Shape of the data in a database can be more complicated than the data demands. It can include side effects of implementation decisions.

Distinctive Features

Defines a standard for ETL

- Databases provide too much design choice
- DW takes away choices of data layout:
  - Always de-normalize
  - Redundant, even when extreme
  - Indexing decisions are made by the warehouse, not you.

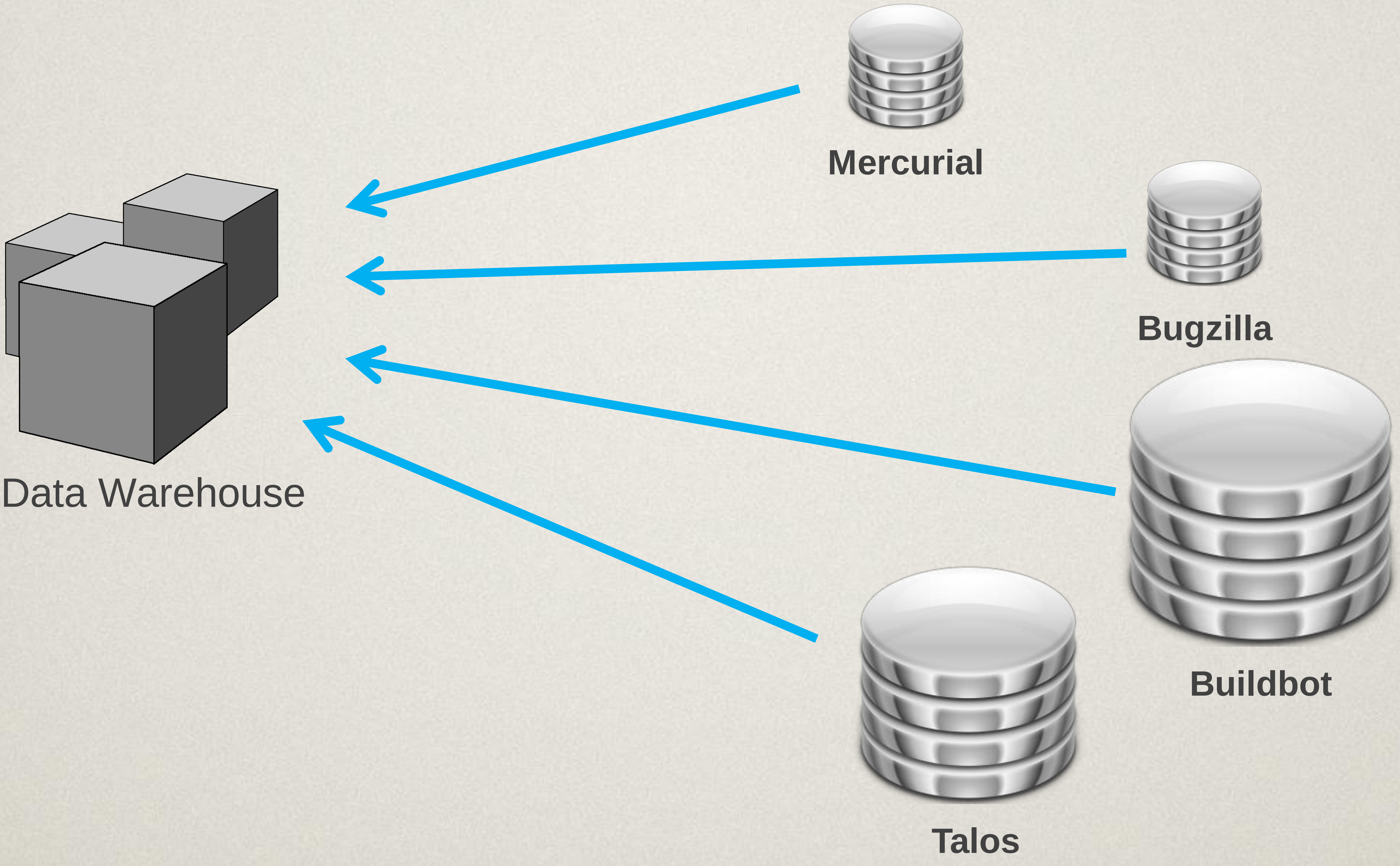
Distinctive Features

Defines a standard for ETL

- Databases provide too much design choice
- DW takes away choices of data layout
- DW demands all data is centralized
  - No deciding which system it best be in
  - Demands data that can be cross-referenced

Distinctive Features

Defines a standard for ETL



Data Warehouse?

Do you need a Data Warehouse?

- Are you caching?
  - cache filtered results
  - writing materialized views
  - *managing* materialized views (caching aggregates)
- Are you indexing? (for query optimization)
- Are you joining? (and delivering long result sets)
- Are you building a query interface?
- Can you accept “eventual consistency”?

Are you doing these all happening in a recognizable section of your code? - You need a data warehouse, or at least use it’s abstractions.

Data Warehouse?

Data warehouse is more than a NoSQL columnar data store:

- DW provides a query language with fast slices and aggregates
- DW includes extra metadata, how dimensions relate to each other and about the measures.

NoSQL data stores make an excellent base for data warehousing, but require additional work

Data Warehouse?

Existing Solutions?

- Open Source is about 20 years behind commercial software.
- Underlying Open Source technology is well developed, but the integration is non-existent.
- Good solutions are commercial solutions - Metrics uses Vertica and Tableau – both commercial products.
- Business Intelligence is very profitable: Open Source solutions disappear:
  - Pentaho – Went from subscription model to multi-license model (~July 2009?).
  - Mozilla was working with WebDetails, now bought by Pentaho.

Data Warehouse?

No Existing Solutions!?

- Mozilla may be unique:
  - BI is a means to an ends – We may be the first large and truly open company with BI needs.
  - We have a mandate to be open
  - We can define the integration standards
  - With standards, we can work with community and amplify each others skill

Data Warehouse?

Data Warehouse as Abstraction

- Dictate the two main DW standards:
  - Input - Multidimensional data cube
  - Output – Fast query service
- ETL designed to fill data cubes
- Clients leverage service to simplify internal design
- Tool discovery and software optimization for the data warehouse is independent of the peripheral software that uses it.

The End