Experiment No.8

Title: Execution of ETL process and OLAP operations

Batch: A1 Roll No.: 16010422013 Experiment No.:8

Aim: Execution of ETL process and OLAP operations

Resources needed: Different RDBMS such as MySQL, Postgres and Excel, CSV, Rapidminer 5.3/ Latest vision

---

Theory

Data Warehouse:

An analytics-focused type of data management system called a data warehouse is intended to assist and allow business intelligence (BI) activities. Large amounts of historical data are frequently included in data warehouses, which are only designed to be used for queries and analysis. Application log files and transaction apps are only two examples of the many different sources from which the data in a data warehouse often comes.

Big data from various sources is centralised and combined in a data warehouse. Because of its analytical skills, businesses can get more out of their data and make better decisions. It gradually compiles a historical record that data scientists and business analysts can find quite useful. Because to these features, a data warehouse can be regarded as an organization's "single source of truth."

ETL:

Extract, Transform, Load (ETL) refers to a process in database usage and especially in data warehousing. Data extraction is where data is extracted from homogeneous or heterogeneous data sources; data transformation where the data is transformed for storing in the proper format or structure for the purposes of querying and analysis; data loading where the data is loaded into the final target database, more specifically, an operational data store, data mart, or data warehouse.

One may improve their chances of achieving better connection and scalability by employing a well-established ETL framework. A decent ETL tool must be able to interface with the several different relational databases and read the various file formats employed by a business. ETL solutions have started to move into Enterprise Application Integration, or even Enterprise Service Bus, systems that now encompass a lot more than simply the extraction, transformation, and loading of data. Converting CSV files into formats usable by relational databases is one frequent use case for ETL technologies. ETL solutions make it feasible for users to input csv-like data feeds/files and import it into a database with as little code as possible, facilitating a typical translation of millions of records. ESTL instruments

RapidMiner:

RapidMiner provides data mining and machine learning procedures including: data loading and transformation (Extract, transform, load (ETL)), data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. RapidMiner is written in the Java programming language. RapidMiner provides a GUI to design and execute analytical workflows. Those workflows are called “Processes” in RapidMiner and they consist of multiple “Operators”. Each operator performs a single task within the process, and the output of each operator forms the input of the next one.

Alternatively, the engine can be called from other programs or used as an API. Individual functions can be called from the command line. RapidMiner provides learning schemes, models and algorithms and can be extended using R and Python scripts.

OLAP:

In computing, online analytical processing, or OLAP is an approach to answering multi-dimensional analytical (MDA) queries. OLAP is part of the broader category of business intelligence, which also encompasses relational database report writing and data mining. Typical applications of OLAP include business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture. The term OLAP was created as a slight modification of the traditional database term OLTP (Online Transaction Processing).

OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives. OLAP consists of three basic analytical operations: consolidation (roll-up), drill-down, and slicing and dicing. Consolidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends. By contrast, the drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a region's sales. Slicing and dicing is a feature whereby users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

OLAP queries can be implemented by using analytical SQL functions
Oracle has extensions to ANSI SQL to allow to quickly computing aggregations and rollups.
These new statements include:

- rollup
- cube
- grouping

These simple SQL operators allow creating easy aggregations directly inside the SQL.

Creating tabular aggregates with ROLLUP:

ROLLUP enables an SQL statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total. ROLLUP is a simple extension to the GROUP BY clause, so its syntax is extremely easy to use. Create cross-tabular reports with CUBE:

In multidimensional jargon, a "cube" is a cross-tabulated summary of detail rows. CUBE enables a SELECT statement to calculate subtotals for all possible combinations of a group of dimensions. It also calculates a grand total.

This is the set of information typically needed for all cross-tabular reports, so CUBE can calculate a cross-tabular report with a single select statement

---

Activities:

For ETL:

1. Install https://rapidminer.software.informer.com/download/#downloading
2. Go through the tutorial provided by RapidMiner
3. Extract data from 2 to 3 heterogeneous sources such as excel, MYSQL, Postgres etc.
4. Download any data set from *https://www.kaggle.com/datasets* or similar website

5. Apply five different transformations and filters to the data with specific requirement
6. Prepare a report for the activities 2 and 4 (ETL part) with steps and visualisations applied.
7. Create and save clean dataset in csv file.
8. Import the csv file from step7 in PostgreSQL database.
9. Apply rollup and cube operations to the same

Questions:

1. **Elaborate on the operations applied and results generated to your dataset**
   OLAP stands for Online Analytical Processing Server. It is a software technology that allows users to analyze information from multiple database systems at the same time. It is based on multidimensional data model and allows the user to query on multi-dimensional data (eg. Delhi -&amp;gt; 2018 -&amp;gt; Sales data). OLAP databases are divided into one or more cubes and these cubes are known as Hyper-cubes.
   The ROLLUP is an extension of the GROUP BY clause. The ROLLUP option allows you to include extra rows that represent the subtotals, which are commonly referred to as super-aggregate rows, along with the grand total row. By using the ROLLUP option, you can use a single query to generate multiple grouping sets.
   Similar to the ROLLUP, CUBE is an extension of the GROUP BY clause. CUBE allows you to generate subtotals like the ROLLUP extension. In addition, the CUBE extension will generate subtotals for all combinations of grouping columns specified in the GROUP BY clause.
2. **Explain if Drill-down, Drill-across can be applied in relational database, Justify with a query implementation.**

The best way to think about how drill-down works, is to tell yourself that you're 'walking through the different levels of a hierarchy'. To drill-down, by definition, requires the use of hierarchical data where values are grouped into levels.
Here are some examples of levels:
If your data is date-based, your level groupings might be something like:
Year, Month and Day, OR, Year, Quarter, Day and Hour.
If your data is geographical, your levels might be something like:
Country, State/Province, Postal Code and City.

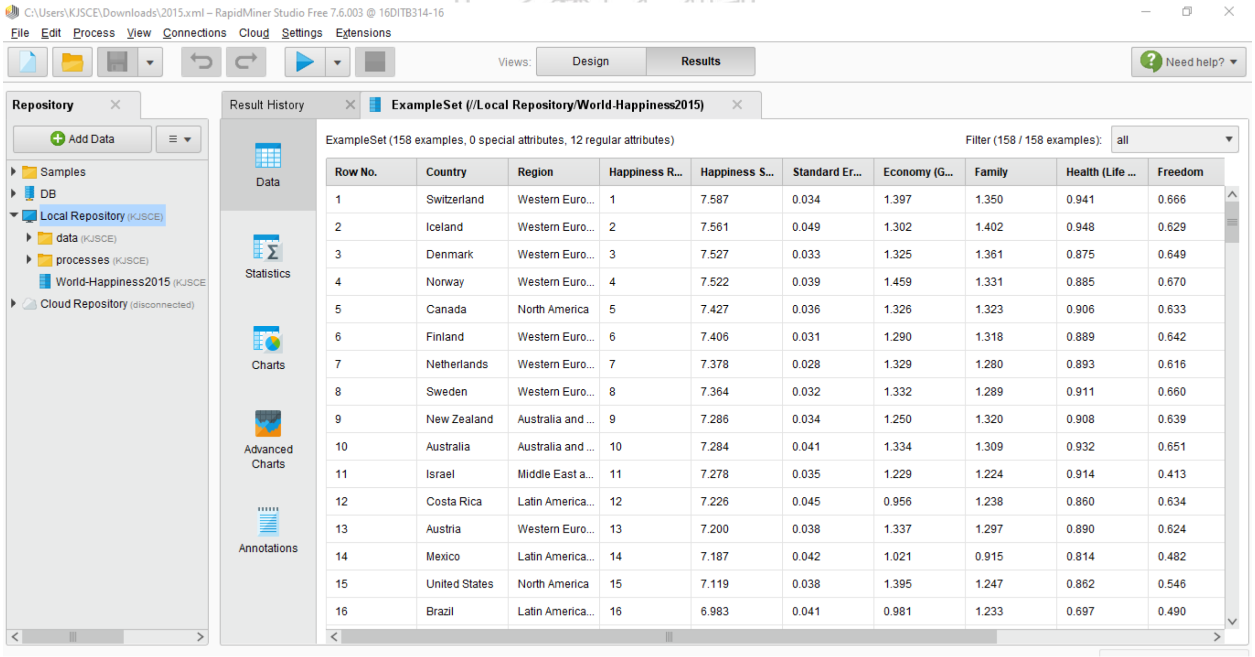
Drilling-across is the process by which you can drill from one application to another from your data visualizations. From the visualization below, a user can automatically drill across into a CRM system using the correct customer number.

To perform a drill-down analysis, we might start with a query that aggregates the total
sales by region:
SELECT region, SUM(quantity) AS total_sales FROM Sales GROUP BY region;
This will give us a result set showing the total sales for each region. To drill down and
see the sales data for each individual store within a region, we can modify the query as
follows:
SELECT region, store, SUM(quantity) AS total_sales FROM Sales GROUP BY region, store;

This will give us a result set showing the total sales for each store within each region.
To perform a drill-across analysis, we might start with a query that aggregates the total
sales by product_name:
SELECT product_name, SUM(quantity) AS total_sales FROM Sales GROUP BY product_name;
This will give us a result set showing the total sales for each product name. To drill
across and compare the sales data by product name across different regions, we can
modify the query as follows:
SELECT product_name, region, SUM(quantity) AS total_sales FROM Sales GROUP BY product_name, region;
This will give us a result set showing the total sales for each product name within each region. Overall, drill-down and drill-across are useful techniques for analyzing data in relational databases, and SQL queries provide a straightforward way to implement them.

Results:

Basic Display of Data:



ExampleSet (158 examples, 0 special attributes, 12 regular attributes)

Row No.	Country	Region	Happiness R...	Happiness S...	Standard Er...	Economy (G...	Family	Health (Life ...	Freedom
1	Switzerland	Western Euro...	1	7.587	0.034	1.397	1.350	0.941	0.666
2	Iceland	Western Euro...	2	7.561	0.049	1.302	1.402	0.948	0.629
3	Denmark	Western Euro...	3	7.527	0.033	1.325	1.361	0.875	0.649
4	Norway	Western Euro...	4	7.522	0.039	1.459	1.331	0.885	0.670
5	Canada	North America	5	7.427	0.036	1.326	1.323	0.906	0.633
6	Finland	Western Euro...	6	7.406	0.031	1.290	1.318	0.889	0.642
7	Netherlands	Western Euro...	7	7.378	0.028	1.329	1.280	0.893	0.616
8	Sweden	Western Euro...	8	7.364	0.032	1.332	1.289	0.911	0.660
9	New Zealand	Australia and ...	9	7.286	0.034	1.250	1.320	0.908	0.639
10	Australia	Australia and ...	10	7.284	0.041	1.334	1.309	0.932	0.651
11	Israel	Middle East a...	11	7.278	0.035	1.229	1.224	0.914	0.413
12	Costa Rica	Latin America...	12	7.226	0.045	0.956	1.238	0.860	0.634
13	Austria	Western Euro...	13	7.200	0.038	1.337	1.297	0.890	0.624
14	Mexico	Latin America...	14	7.187	0.042	1.021	0.915	0.814	0.482
15	United States	North America	15	7.119	0.038	1.395	1.247	0.862	0.546
16	Brazil	Latin America...	16	6.983	0.041	0.981	1.233	0.697	0.490

Happiness score vs Happiness rank:

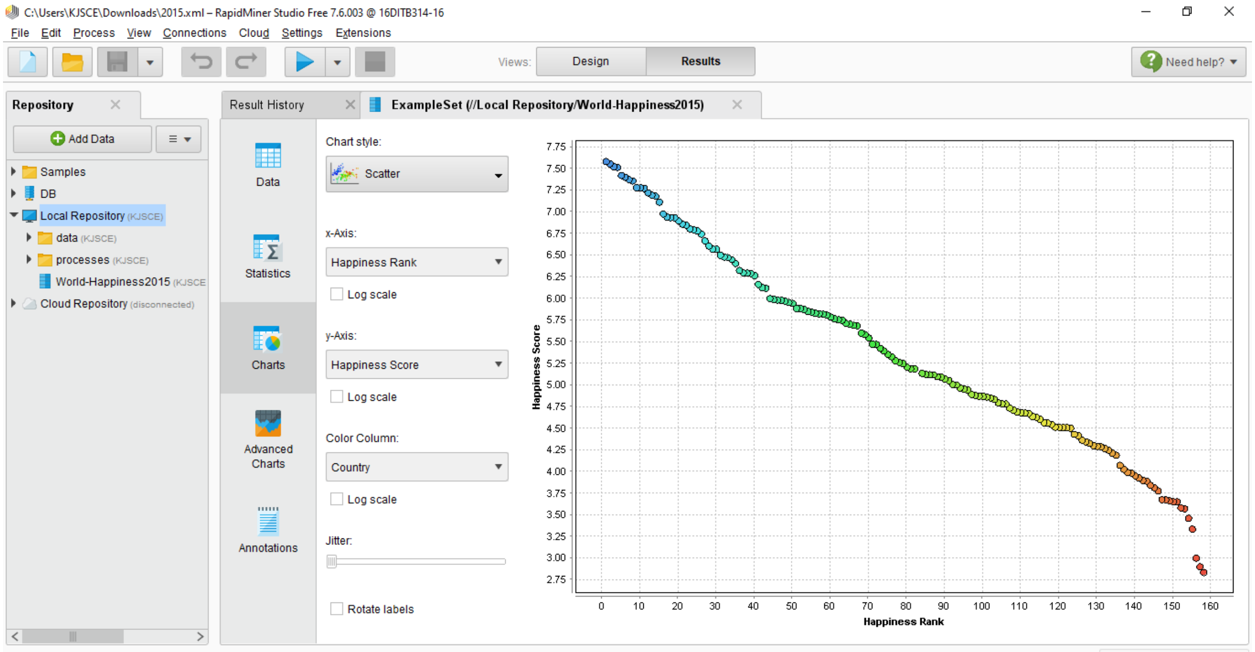


Chart style: Scatter

x-Axis: Happiness Rank

☐ Log scale

y-Axis: Happiness Score

☐ Log scale

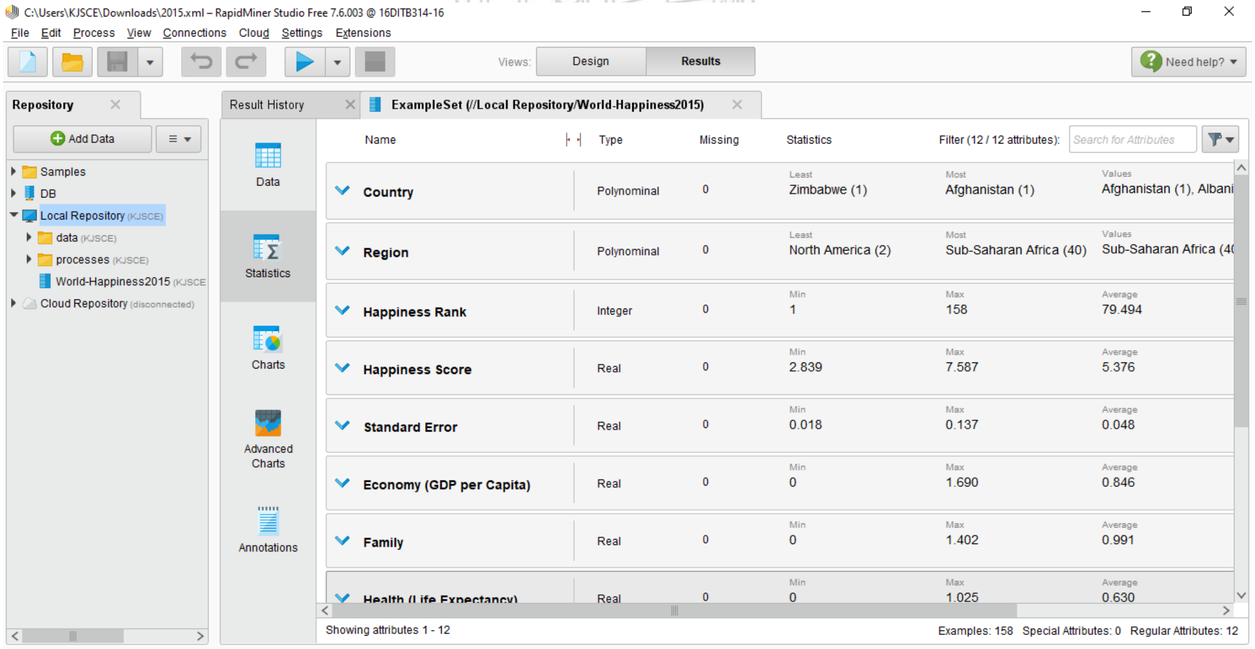
Color Column: Country

☐ Log scale

Jitter:

☐ Rotate labels

Statistics:


C:\Users\KJSCE\Downloads\2015.xml – RapidMiner Studio Free 7.6.003 @ 16DITB314-16
File Edit Process View Connections Cloud Settings Extensions
Views: Design Results
Need help?
Repository
Add Data
Samples
DB
Local Repository (KJSCE)
data (KJSCE)
processes (KJSCE)
World-Happiness2015 (KJSCE
Cloud Repository (disconnected)
Result History
ExampleSet (//Local Repository/World-Happiness2015)
Data
Statistics
Charts
Advanced Charts
Annotations
Search for Attributes

Filter (12 / 12 attributes):

Name	Type	Missing	Statistics		
Country	Polynominal	0	Least Zimbabwe (1)	Most Afghanistan (1)	Values Afghanistan (1), Albani
Region	Polynominal	0	Least North America (2)	Most Sub-Saharan Africa (40)	Values Sub-Saharan Africa (4(
Happiness Rank	Integer	0	Min 1	Max 158	Average 79.494
Happiness Score	Real	0	Min 2.839	Max 7.587	Average 5.376
Standard Error	Real	0	Min 0.018	Max 0.137	Average 0.048
Economy (GDP per Capita)	Real	0	Min 0	Max 1.690	Average 0.846
Family	Real	0	Min 0	Max 1.402	Average 0.991
Health (Life Expectancy)	Real	0	Min 0	Max 1.025	Average 0.630

Showing attributes 1 - 12

Examples: 158 Special Attributes: 0 Regular Attributes: 12

Happiness Score vs Country:

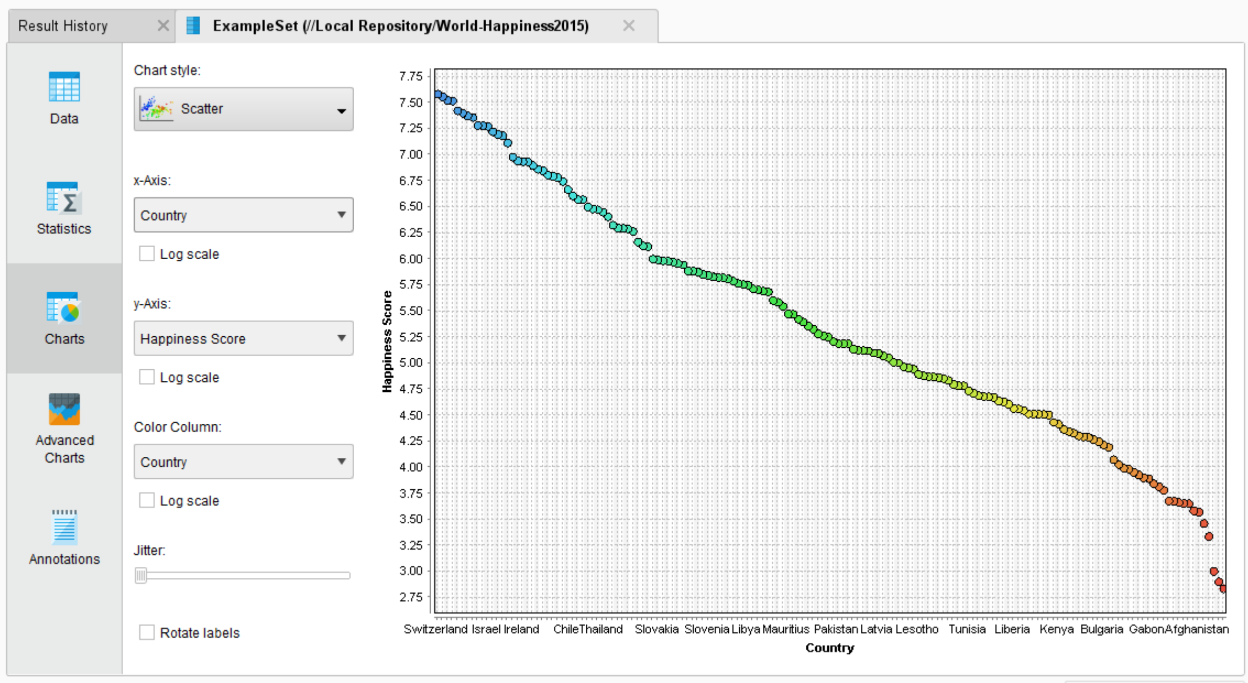

Result History
ExampleSet (//Local Repository/World-Happiness2015)
Data
Statistics
Charts
Advanced Charts
Annotations
7.75
7.50
7.25
7.00
6.75
6.50
6.25
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
Happiness Score
Switzerland Israel Ireland Chile Thailand Slovakia Slovenia Libya Mauritius Pakistan Latvia Lesotho Tunisia Liberia Kenya Bulgaria Gabon Afghanistan
Country

Chart style:
Scatter

x-Axis:
Country
□ Log scale

y-Axis:
Happiness Score
□ Log scale

Color Column:
Country
□ Log scale

Jitter:

□ Rotate labels

Happiness Rank vs Country:

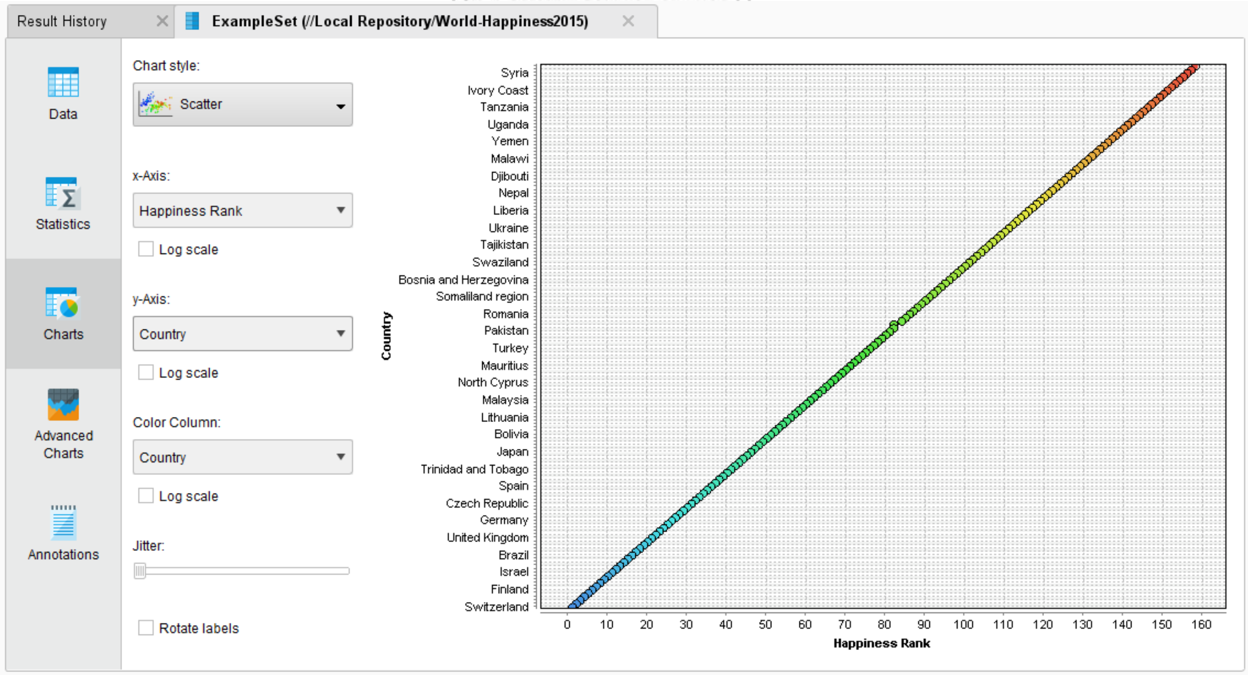


Chart style:
Scatter

x-Axis:
Happiness Rank
□ Log scale

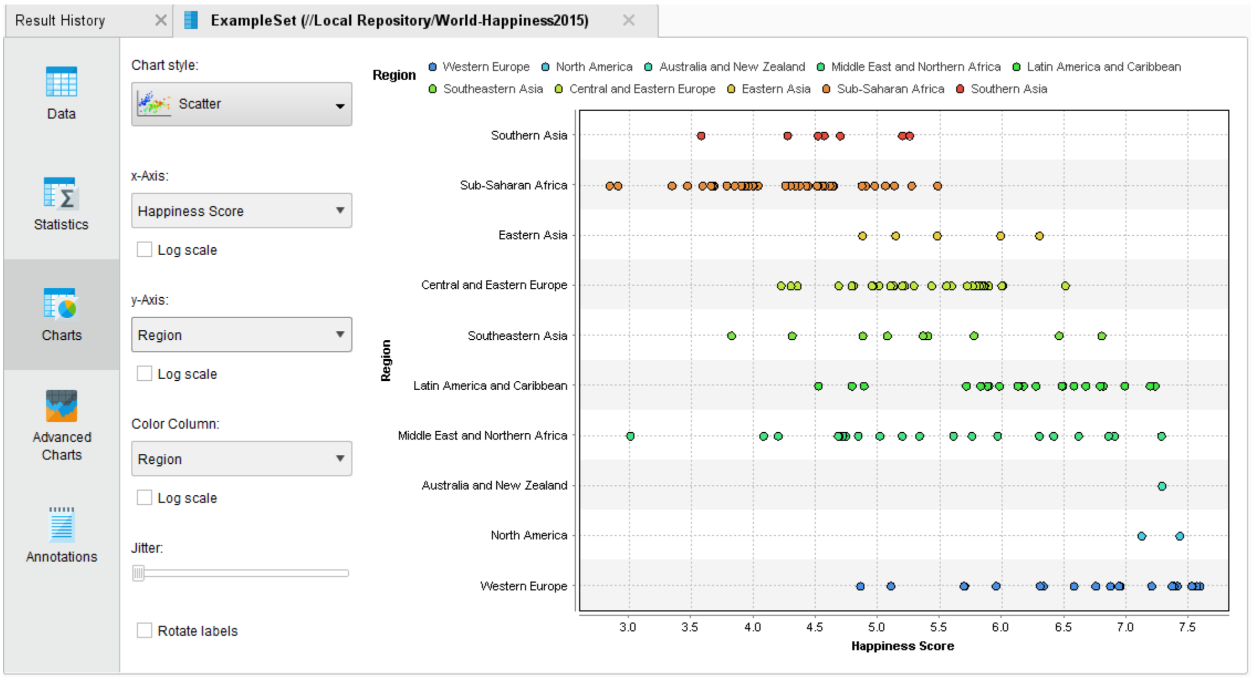
y-Axis:
Country
□ Log scale

Color Column:
Country
□ Log scale

Jitter:

□ Rotate labels

Region vs Happiness score:



Result History

ExampleSet (//Local Repository/World-Happiness2015)

Chart style: Scatter

x-Axis: Happiness Score

□ Log scale

y-Axis: Region

□ Log scale

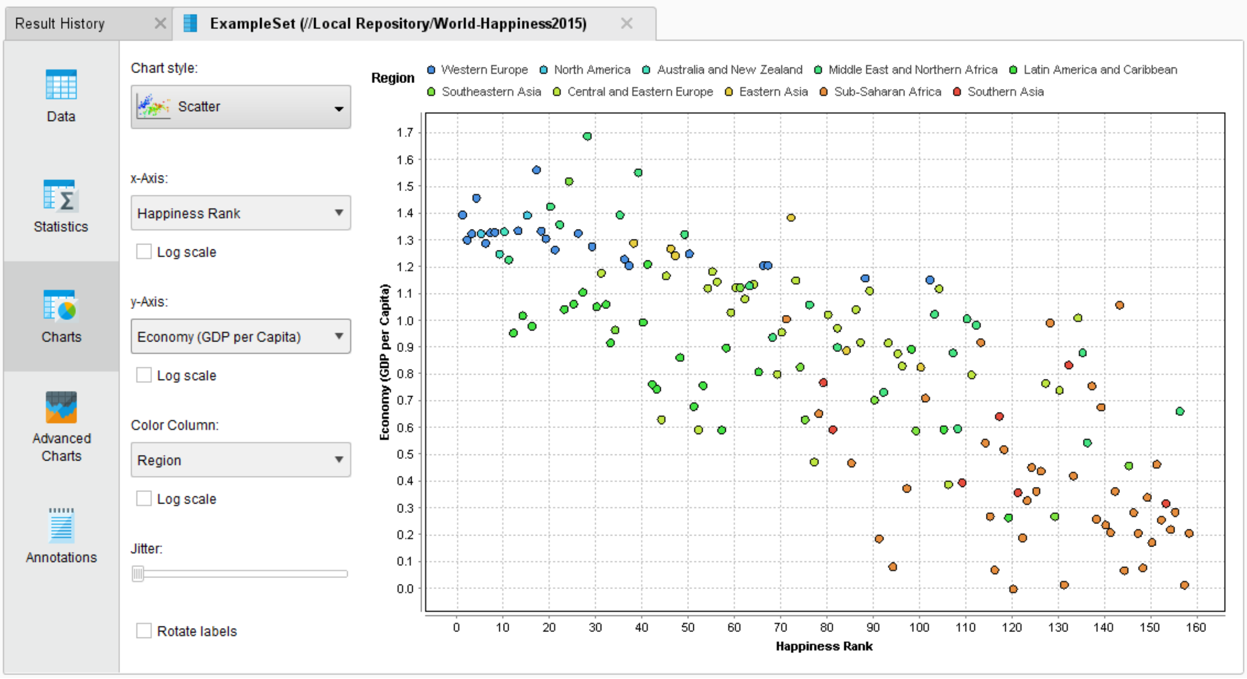
Color Column: Region

□ Log scale

Jitter:

□ Rotate labels

Happiness rank vs Economy:



Result History

ExampleSet (//Local Repository/World-Happiness2015)

Chart style: Scatter

x-Axis: Happiness Rank

□ Log scale

y-Axis: Economy (GDP per Capita)

□ Log scale

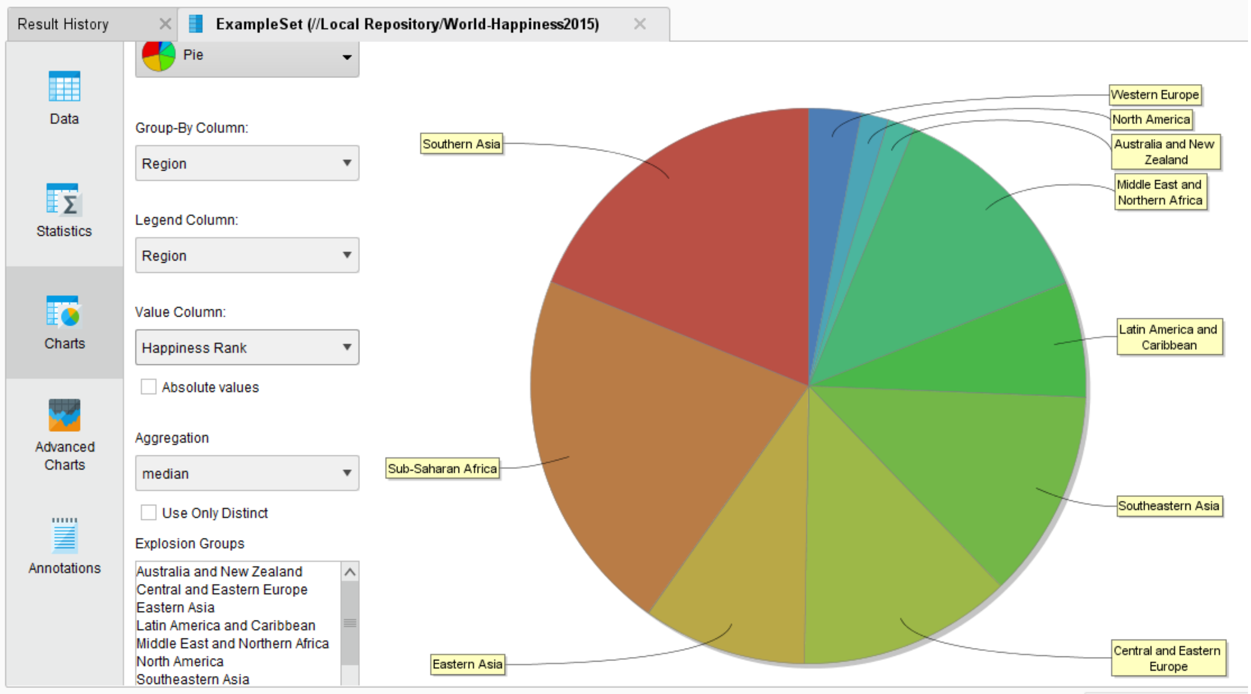
Color Column: Region

□ Log scale

Jitter:

□ Rotate labels

Data visualization using median aggregation - pie chart


Result History
ExampleSet (//Local Repository/World-Happiness2015)
Data
Statistics
Charts
Advanced Charts
Annotations
Western Europe
North America
Australia and New Zealand
Middle East and Northern Africa
Latin America and Caribbean
Southeastern Asia
Central and Eastern Europe
Eastern Asia
Sub-Saharan Africa
Southern Asia

Pie

Group-By Column: Region

Legend Column: Region

Value Column: Happiness Rank

□ Absolute values

Aggregation: median

□ Use Only Distinct

Explosion Groups

Australia and New Zealand
Central and Eastern Europe
Eastern Asia
Latin America and Caribbean
Middle East and Northern Africa
North America
Southeastern Asia

Data visualization using median aggregation - bar chart

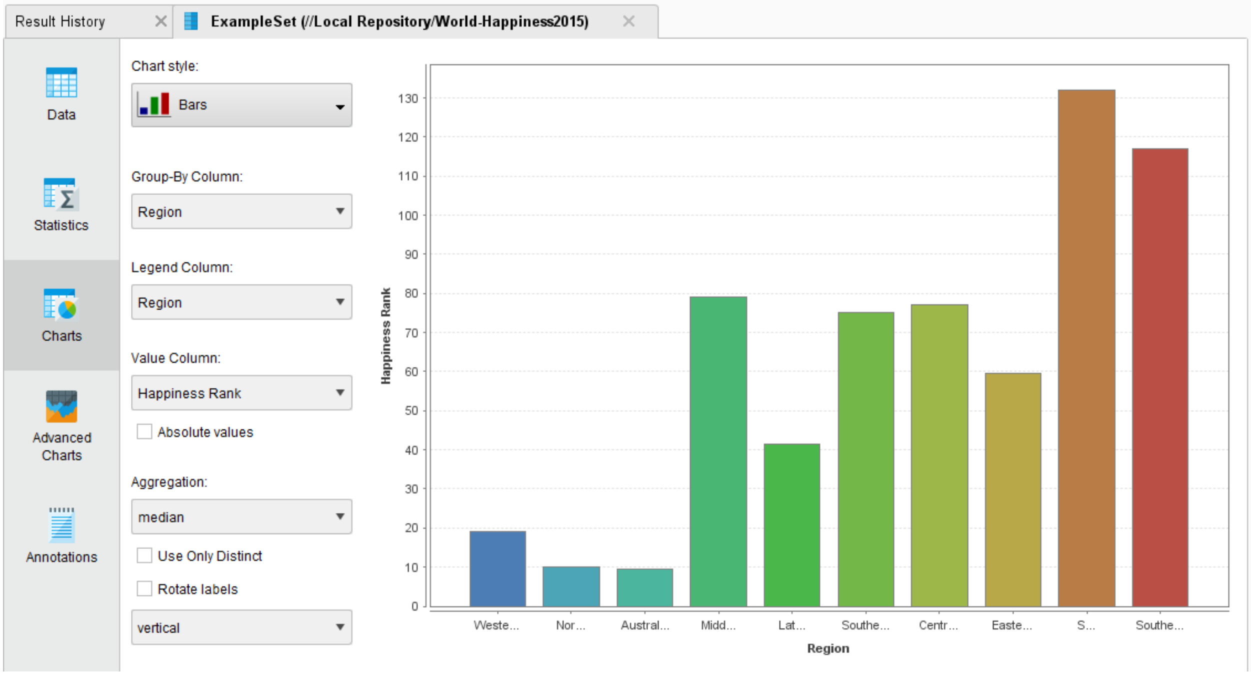


Chart style: Bars

Group-By Column: Region

Legend Column: Region

Value Column: Happiness Rank

□ Absolute values

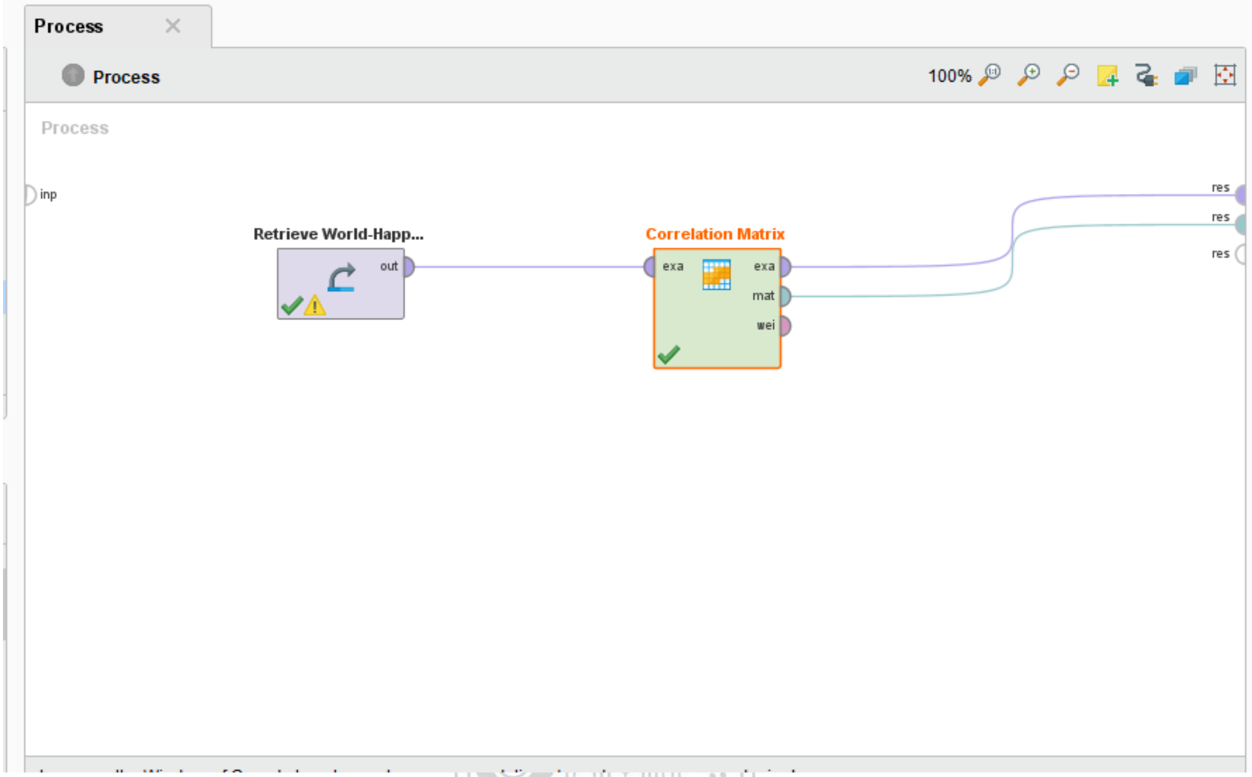
Aggregation: median

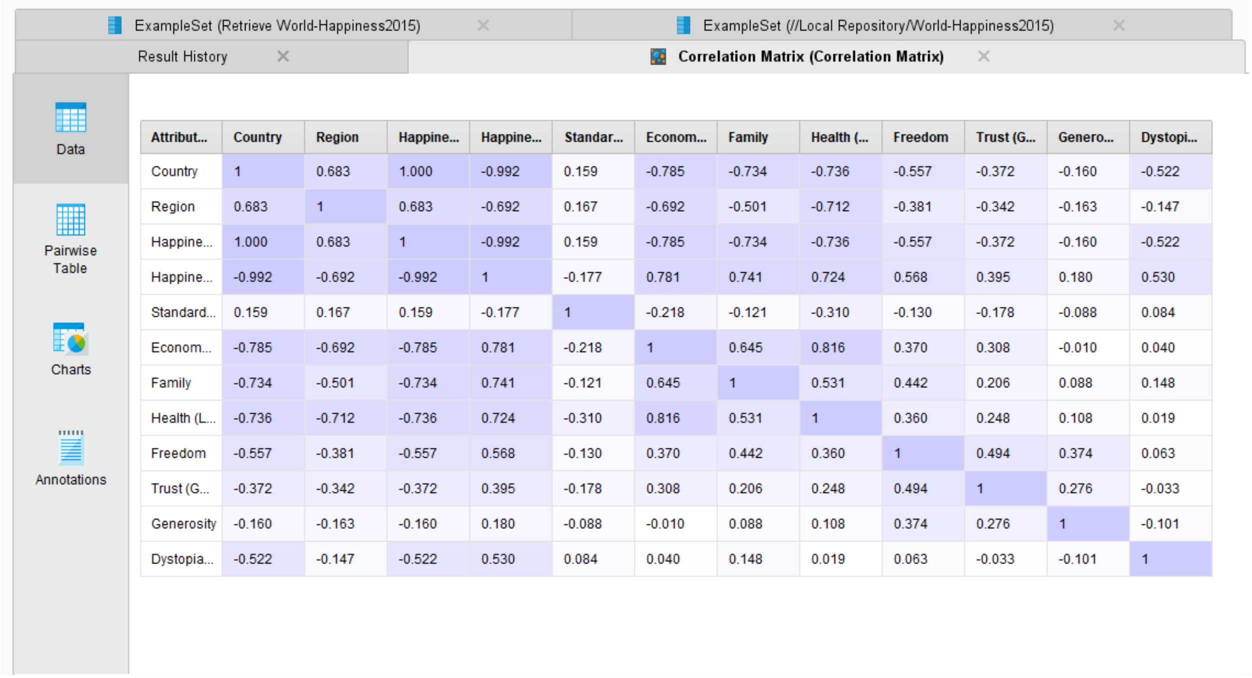
□ Use Only Distinct

□ Rotate labels

vertical

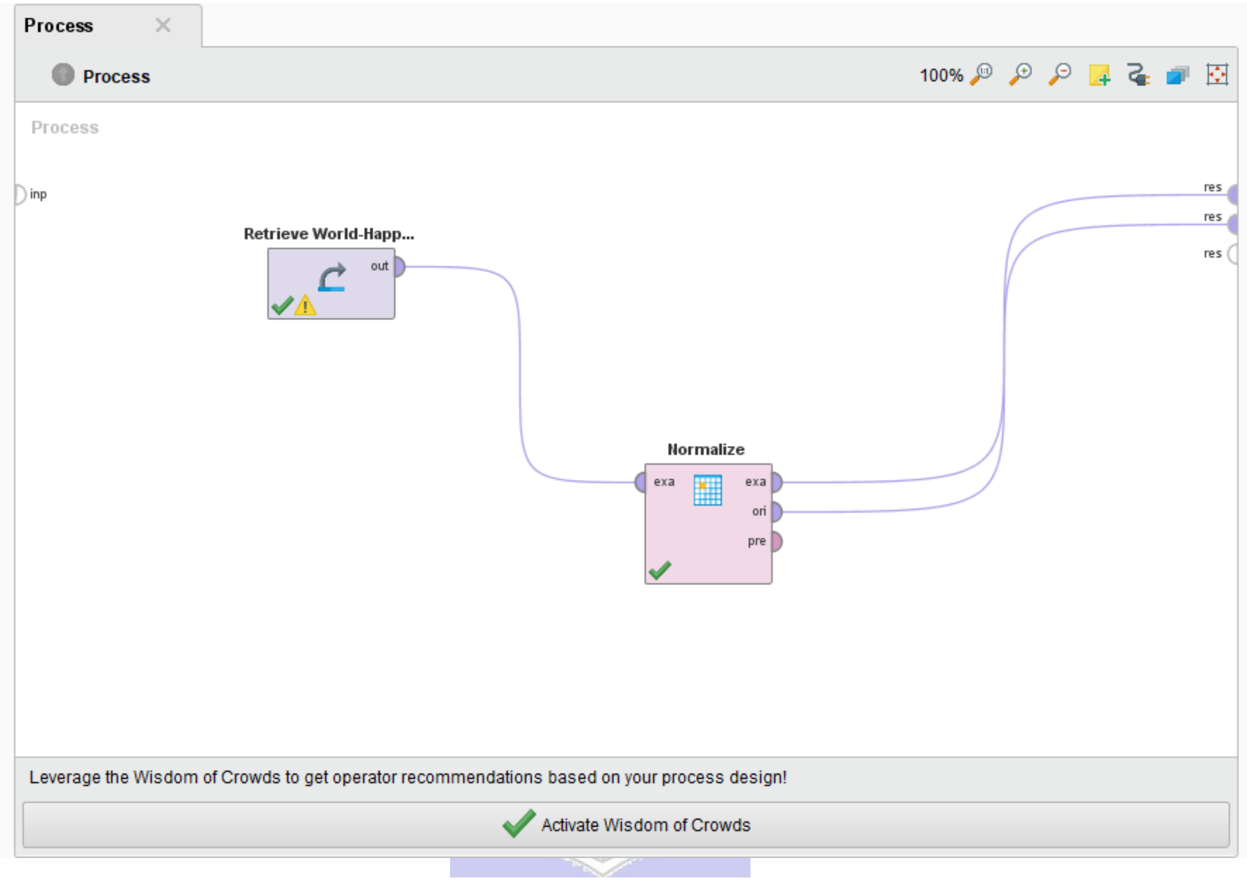
Correlation matrix:

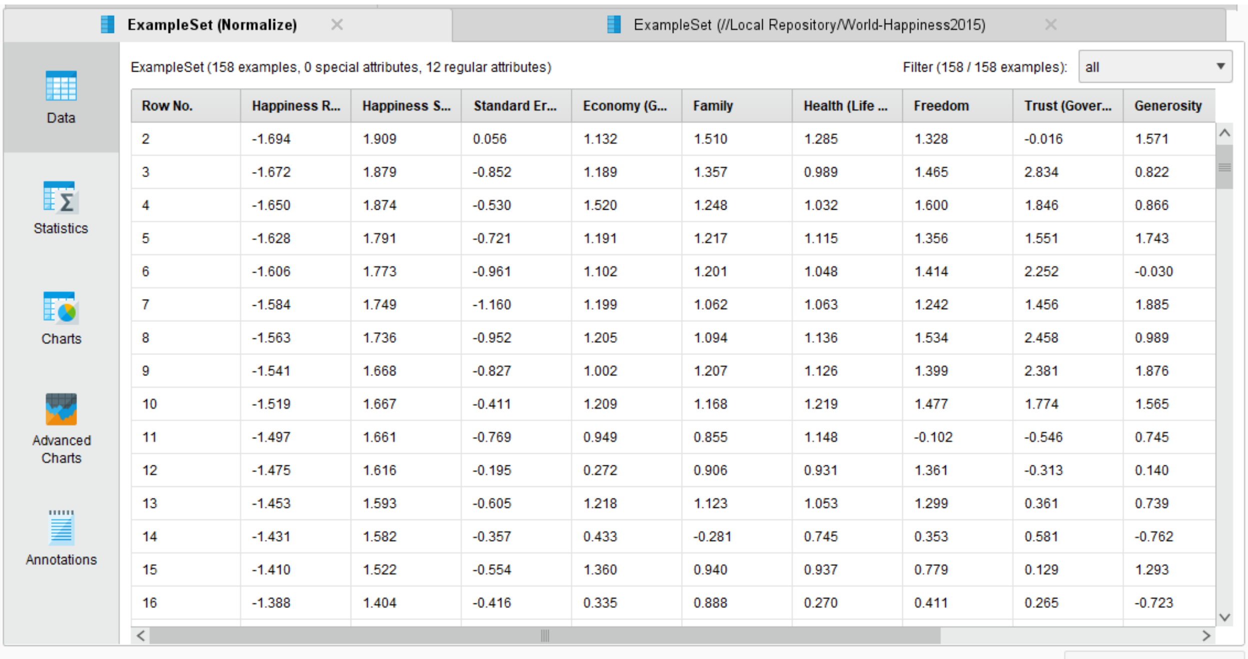



ExampleSet (Retrieve World-Happiness2015)
ExampleSet (//Local Repository/World-Happiness2015)
Result History
Correlation Matrix (Correlation Matrix)
Data
Pairwise Table
Charts
Annotations

Attribut...	Country	Region	Happine...	Happine...	Standar...	Econom...	Family	Health (...	Freedom	Trust (G...	Genero...	Dystopi...
Country	1	0.683	1.000	-0.992	0.159	-0.785	-0.734	-0.736	-0.557	-0.372	-0.160	-0.522
Region	0.683	1	0.683	-0.692	0.167	-0.692	-0.501	-0.712	-0.381	-0.342	-0.163	-0.147
Happine...	1.000	0.683	1	-0.992	0.159	-0.785	-0.734	-0.736	-0.557	-0.372	-0.160	-0.522
Happine...	-0.992	-0.692	-0.992	1	-0.177	0.781	0.741	0.724	0.568	0.395	0.180	0.530
Standard...	0.159	0.167	0.159	-0.177	1	-0.218	-0.121	-0.310	-0.130	-0.178	-0.088	0.084
Econom...	-0.785	-0.692	-0.785	0.781	-0.218	1	0.645	0.816	0.370	0.308	-0.010	0.040
Family	-0.734	-0.501	-0.734	0.741	-0.121	0.645	1	0.531	0.442	0.206	0.088	0.148
Health (L...	-0.736	-0.712	-0.736	0.724	-0.310	0.816	0.531	1	0.360	0.248	0.108	0.019
Freedom	-0.557	-0.381	-0.557	0.568	-0.130	0.370	0.442	0.360	1	0.494	0.374	0.063
Trust (G...	-0.372	-0.342	-0.372	0.395	-0.178	0.308	0.206	0.248	0.494	1	0.276	-0.033
Generosity	-0.160	-0.163	-0.160	0.180	-0.088	-0.010	0.088	0.108	0.374	0.276	1	-0.101
Dystopia...	-0.522	-0.147	-0.522	0.530	0.084	0.040	0.148	0.019	0.063	-0.033	-0.101	1

Data normalization:


Process
Process
100%
Process
inp
Retrieve World-Happ...
out
Normalize
exa
exa
ori
pre
res
res
res
Leverage the Wisdom of Crowds to get operator recommendations based on your process design!
Activate Wisdom of Crowds

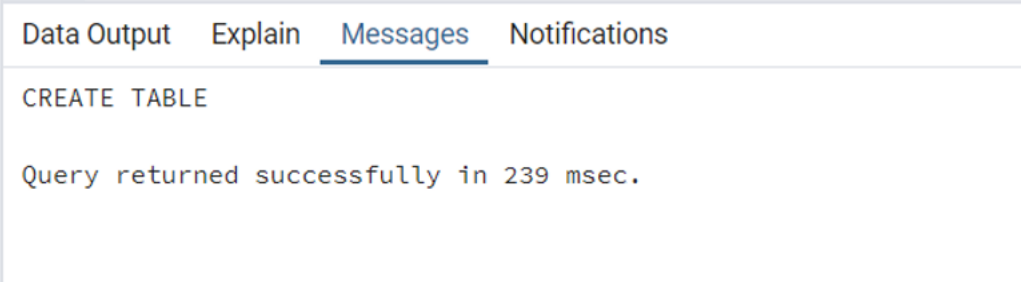


ExampleSet (158 examples, 0 special attributes, 12 regular attributes)

Filter (158 / 158 examples): all

Row No.	Happiness R...	Happiness S...	Standard Er...	Economy (G...	Family	Health (Life ...	Freedom	Trust (Gover...	Generosity
2	-1.694	1.909	0.056	1.132	1.510	1.285	1.328	-0.016	1.571
3	-1.672	1.879	-0.852	1.189	1.357	0.989	1.465	2.834	0.822
4	-1.650	1.874	-0.530	1.520	1.248	1.032	1.600	1.846	0.866
5	-1.628	1.791	-0.721	1.191	1.217	1.115	1.356	1.551	1.743
6	-1.606	1.773	-0.961	1.102	1.201	1.048	1.414	2.252	-0.030
7	-1.584	1.749	-1.160	1.199	1.062	1.063	1.242	1.456	1.885
8	-1.563	1.736	-0.952	1.205	1.094	1.136	1.534	2.458	0.989
9	-1.541	1.668	-0.827	1.002	1.207	1.126	1.399	2.381	1.876
10	-1.519	1.667	-0.411	1.209	1.168	1.219	1.477	1.774	1.565
11	-1.497	1.661	-0.769	0.949	0.855	1.148	-0.102	-0.546	0.745
12	-1.475	1.616	-0.195	0.272	0.906	0.931	1.361	-0.313	0.140
13	-1.453	1.593	-0.605	1.218	1.123	1.053	1.299	0.361	0.739
14	-1.431	1.582	-0.357	0.433	-0.281	0.745	0.353	0.581	-0.762
15	-1.410	1.522	-0.554	1.360	0.940	0.937	0.779	0.129	1.293
16	-1.388	1.404	-0.416	0.335	0.888	0.270	0.411	0.265	-0.723

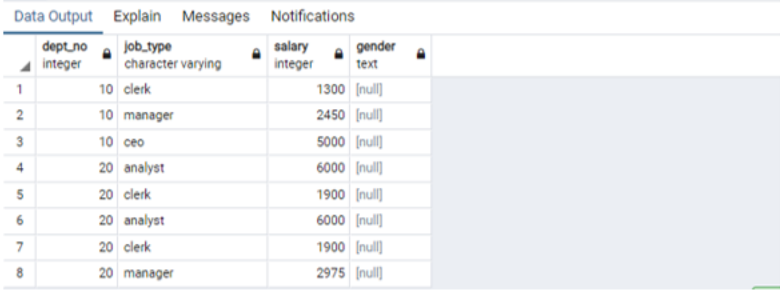
```
CREATE TABLE
employee (
dept_no int,
job_type
varchar,
salary
int,
gender
text
);
```



```
CREATE TABLE

Query returned successfully in 239 msec.
```

Inserting into table:

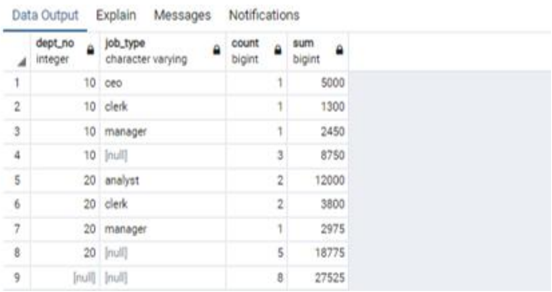
```
insert into
employee values(10,
'clerk', 1300),
(10, 'manager', 2450),
(10, 'ceo', 5000),
(20, 'analyst', 6000),
(20, 'clerk', 1900),

(20, 'analyst', 6000),
(20, 'clerk', 1900),
(20,'manager', 2975);
select * from employee;
```



	dept_no integer	job_type character varying	salary integer	gender text
1	10	clerk	1300	[null]
2	10	manager	2450	[null]
3	10	ceo	5000	[null]
4	20	analyst	6000	[null]
5	20	clerk	1900	[null]
6	20	analyst	6000	[null]
7	20	clerk	1900	[null]
8	20	manager	2975	[null]

Rollup:

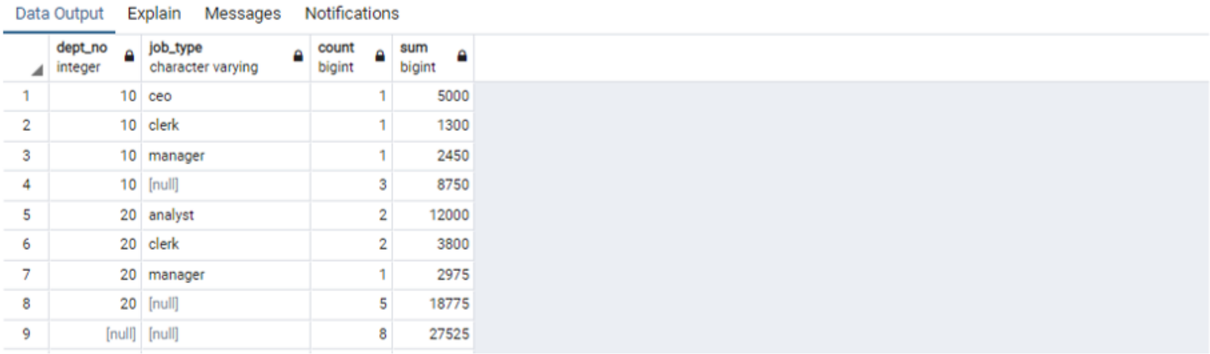
```
SELECT dept_no, job_type, count(*),
sum(salary)FROM employee
GROUP BY ROLLUP(dept_no, job_type);
```



	dept_no integer	job_type character varying	count bigint	sum bigint
1	10	ceo	1	5000
2	10	clerk	1	1300
3	10	manager	1	2450
4	10	[null]	3	8750
5	20	analyst	2	12000
6	20	clerk	2	3800
7	20	manager	1	2975
8	20	[null]	5	18775
9	[null]	[null]	8	27525

Cube:

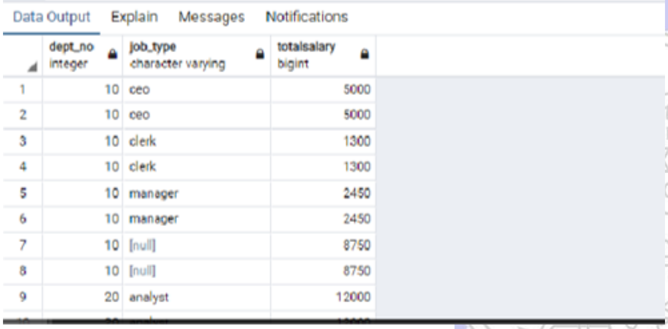
```
SELECT dept_no, job_type, count(*),
sum(salary)FROM employee
GROUP BY
CUBE(dept_no, job_type);
```



	dept_no integer	job_type character varying	count bigint	sum bigint
1	10	ceo	1	5000
2	10	clerk	1	1300
3	10	manager	1	2450
4	10	[null]	3	8750
5	20	analyst	2	12000
6	20	clerk	2	3800
7	20	manager	1	2975
8	20	[null]	5	18775
9	[null]	[null]	8	27525

Grouping sets:

```
SELECT dept_no, job_type, sum(salary) AS
TotalSalaryFROM employee
GROUP BY Grouping
Sets(RollUp(dept_no,job_type),
CUBE(dept_no,job_type));
```



	dept_no integer	job_type character varying	totalsalary bigint
1	10	ceo	5000
2	10	ceo	5000
3	10	clerk	1300
4	10	clerk	1300
5	10	manager	2450
6	10	manager	2450
7	10	[null]	8750
8	10	[null]	8750
9	20	analyst	12000

Outcomes:

CO4: Apply ETL Processing and Online Analytical Processing on the warehouse data

Conclusion: (Conclusion to be based on the outcomes achieved):

In this experiment we successfully understood OLAP operations and understood, grouping by, cube and roll up operations therefore yielding the desired result.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

- https://www.oracle.com/in/database/what-is-a-data-warehouse
- Paulraj Ponniah, “Data Warehousing: Fundamentals for IT Professionals”, Wiley India