### WEEK 1

**VARIOUS DATA PRE PROCESSING TECHNIQUES**

### 1. Import Data and Read the data from Dataset.csv and perform the following task checking for missing values from the dataset.

### Explanation:

### Before we start preparing our data, first we need to download and load it in RStudio IDE.

### Handling the missing data

From the dataset, the Age and Salary column report missing data. Before implementing our machine learning models, this problem needs to be solved, otherwise it will cause a serious problem to our machine learning models. Therefore, it’s our responsibility to ensure this missing data is eliminated from our dataset using the most appropriate technique.

Here are two techniques we can use to handle missing data:

1. **Delete the observation reporting the missing data:**

This technique is suitable when dealing with big datasets and with very few missing values i.e. deleting one row from a dataset with thousands of observations can not affect the quality of the data. When the dataset reports many missing values, it can be very dangerous to use this technique. Deleting many rows from a dataset can lead to the loss of crucial information contained in the data.

To ensure this does not happen, we make use of an appropriate technique that has no harm to the quality of the data.

1. **Replace the missing data with the average of the feature in which the data is missing:**

This technique is the best way so far to deal with the missing values. Many statisticians make use of this technique over that of the first one.

**SOURSE CODE:**

### Dataset = read\_csv('data.csv')

### This above code imports our data stored in CSV format.

### view(Dataset)

### Upon executing we obtain our dataset as below.

### OUTPUT:

### 

### Handling the missing data:

### OUTPUT:

Dataset$Age = ifelse(is.na(Dataset$Age),

ave(Dataset$Age, FUN = function (x)mean(x, na.rm = TRUE)),

Dataset$Age)

### 

### The missing value that was in the Age column of our data set has successfully been replaced with the mean of the same column.

Dataset$Salary = ifelse(is.na(Dataset$Salary),

ave(Dataset$Salary, FUN = function (x)mean(x, na.rm = TRUE)),

Dataset$Salary)

### 

### The missing value that was in the Salary column was successfully replaced with the mean of the same column.

### Exercise:1.1

### Import Data and Read the data from Dataset.csv and perform the following task Encoding the categorical data.

### Explanation:

### Encoding categorical data

Encoding refers to transforming text data into numeric data. Encoding Categorical data simply means we are transforming data that fall into categories into numeric data.

In our dataset, the Country column is Categorical data with 3 levels i.e. France, Spain, and Germany. The purchased column is Categorical data as well with 2 categories, i.e. YES and NO.

The machine models we built on our dataset are based on mathematical equations and it’s only take numbers in those equations.

Keeping texts of a categorical variable in the equation can cause some troubles to the machine learning models and this why we encode those variables. To transform a categorical variable into numeric, we use the factor() function.

* Encoding refers to transforming text data into numeric data. Encoding Categorical data simply means we are transforming data that fall into categories into numeric data.
* In our dataset, the Country column is Categorical data with 3 levels i.e. France, Spain, and Germany. The purchased column is Categorical data as well with 2 categories, i.e. YES and NO.

Dataset$Country = factor(Dataset$Country,

levels = c('France','Spain','Germany'),

labels = c(1.0, 2.0 , 3.0 ))

### OUTPUT:

### 

Our country names were successfully replaced with numbers.

We do the same for the purchased column.

Dataset$Purchased = factor (Dataset$Purchased,

levels = c('No', 'Yes'),

labels = c(0, 1))

Dataset$Purchased[is.na(Dataset$Purchased)] <- 0

as.factor(Dataset$Purchased)

### Using the view() function we obtain.

### 

### .

### Exercise:1.2

### Import Data and Read the data from Dataset.csv and perform the following task splitting the dataset into the training and test set then do Feature scaling.

**Explanation:**

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

**Example:** If an algorithm is not using the feature scaling method then it can consider the value 3000 meters to be greater than 5 km but that’s actually not true and in this case, the algorithm will give wrong predictions. So, we use Feature Scaling to bring all values to the same magnitudes and thus, tackle this issue.

**Techniques to perform Feature Scaling**  
consider the two most important ones:

* **Min-Max Normalization:**This technique re-scales a feature or observation value with distribution value between 0 and 1.  
   
* **Standardization:**It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.  
   

Using our dataset, let’s split it into the training and test sets.

To begin with, we first load the required library.

### SOURCE CODE AND OUTPUT:

### 

### 

### From the results it clear that eight observations, 0.8 of our dataset observations, were split into the training set.

### 

### Feature scaling: The normalization technique is used when the data is normally distributed while standardization works with both normally distributed and the data that is not normally distributed.

The formula for these two techniques is shown below.

### 

### Let’s scale both the training set and test set of our dataset separately.

### 

### 

### Our training and test set were successfully scaled.

### If we fail to do so, R will show us an error.

### Such as:

### training\_set = scale(training\_set)# returns an error

### The reason is that our encoded columns are not treated as numeric entries.

### WEEK 2

**Data Warehouse Implementation**

**Extracting the data using sales data and Performing various OLAP operations like Slice and visualizing the multidimensional data for analysis.**

### Explaination:

### List of OLAP operations:

### Roll-up

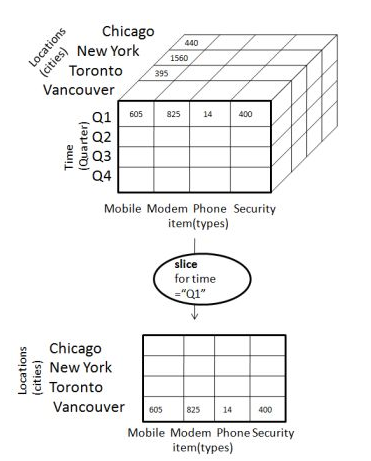
### Drill-down

### Slice and dice

### Pivot (rotate)

**Slice**

The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



* Here Slice is performed for the dimension "time" using the criterion time = "Q1".
* It will form a new sub-cube by selecting one or more dimensions.

**SOURCE CODE:**

### ## credits

### # https://dzone.com/articles/olap-operation-r

### # Setup the dimension tables

### state\_table <- data.frame(key=c("CA", "NY", "WA", "ON", "QU"),

### name=c("California", "new York", "Washington", "Ontario", "Quebec"),

### country=c("USA", "USA", "USA", "Canada", "Canada"))

### month\_table <- data.frame(key=1:12,

### desc=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),

### quarter=c("Q1","Q1","Q1","Q2","Q2","Q2","Q3","Q3","Q3","Q4","Q4","Q4"))

### prod\_table <- data.frame(key=c("Printer", "Tablet", "Laptop"), price=c(225, 570, 1120))

### # Function to generate the Sales table

### gen\_sales <- function(no\_of\_recs) {

### # Generate transaction data randomly

### loc <- sample(state\_table$key, no\_of\_recs, replace=T, prob=c(2,2,1,1,1))

### time\_month <- sample(month\_table$key, no\_of\_recs, replace=T)

### time\_year <- sample(c(2012, 2013), no\_of\_recs, replace=T)

### prod <- sample(prod\_table$key, no\_of\_recs, replace=T, prob=c(1, 3, 2))

### unit <- sample(c(1,2), no\_of\_recs, replace=T, prob=c(10, 3))

### amount <- unit\*prod\_table[prod,]$price

### sales <- data.frame(month=time\_month, year=time\_year, loc=loc, prod=prod, unit=unit, amount=amount)

### # Sort the records by time order

### sales <- sales[order(sales$year, sales$month),]

### row.names(sales) <- NULL

### return(sales)

### } # Now create the sales fact table

### sales\_fact <- gen\_sales(500) # Look at a few records

### head(sales\_fact)

### # Build up a cube

### revenue\_cube <- tapply(sales\_fact$amount, sales\_fact[,c("prod", "month", "year", "loc")], FUN=function(x){return(sum(x))})

### # Showing the cells of the cube

### revenue\_cube

### dimnames(revenue\_cube)

### # Slice

### # cube data in Jan, 2012

### revenue\_cube[, "1", "2012",]

### # cube data in Jan, 2012

### revenue\_cube["Tablet", "1", "2012",]

### OUTPUT:

month year loc prod unit amount

1 1 2021 CA Camera 1 NA

2 1 2021 QU Camera 2 NA

3 1 2021 QU Camera 1 NA

4 1 2021 NY Blutooth 1 NA

5 1 2021 CA Mobile 1 NA

6 1 2021 NY Mobile 2 NA

, , year = 2021, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

$prod

[1] "Blutooth" "Camera" "Mobile"

$month

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

$year

[1] "2021" "2022"

$loc

[1] "CA" "NY" "ON" "QU" "WA"

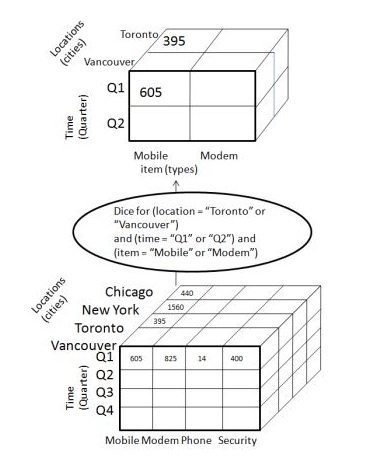
**Exercises:2.2**

**Extracting the data using sales data with following Product name Mobile, Camera and Bluetooth prices 300, 400 and 500 and Performing various OLAP operations like (Dice and Roll-up) and visualizing the Multidimensional data for analysis.**

**Explaination:**

### Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.



The dice operation on the cube based on the following selection criteria involves three dimensions.

* (location = "Toronto" or "Vancouver")
* (time = "Q1" or "Q2")
* (item =" Mobile" or "Modem")

### Roll-up

### Roll-up performs aggregation on a data cube in any of the following ways

### By climbing up a concept hierarchy for a dimension

### By dimension reduction

### The following diagram illustrates how roll-up works

### 

* Roll-up is performed by climbing up a concept hierarchy for the dimension location.
* Initially the concept hierarchy was "street < city < province < country".
* On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
* The data is grouped into cities rather than countries.

When roll-up is performed, one or more dimensions from the data cube are removed

**SOURSE CODE:**

### ## credits

### # https://dzone.com/articles/olap-operation-r

### # Setup the dimension tables

### state\_table <- data.frame(key=c("CA", "NY", "WA", "ON", "QU"),

### name=c("California", "new York", "Washington", "Ontario", "Quebec"),

### country=c("USA", "USA", "USA", "Canada", "Canada"))

### month\_table <- data.frame(key=1:12,

### desc=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),

### quarter=c("Q1","Q1","Q1","Q2","Q2","Q2","Q3","Q3","Q3","Q4","Q4","Q4"))

### prod\_table <- data.frame(key=c("Mobile", "Camera", "Blutooth"), price=c(225, 570, 1120))

### # Function to generate the Sales table

### gen\_sales <- function(no\_of\_recs) {

### # Generate transaction data randomly

### loc <- sample(state\_table$key, no\_of\_recs, replace=T, prob=c(2,2,1,1,1))

### time\_month <- sample(month\_table$key, no\_of\_recs, replace=T)

### time\_year <- sample(c(2022, 2021), no\_of\_recs, replace=T)

### prod <- sample(prod\_table$key, no\_of\_recs, replace=T, prob=c(1, 3, 2))

### unit <- sample(c(1,2), no\_of\_recs, replace=T, prob=c(10, 3))

### amount <- unit\*prod\_table[prod,]$price

### sales <- data.frame(month=time\_month, year=time\_year, loc=loc, prod=prod, unit=unit, amount=amount)

### # Sort the records by time order

### sales <- sales[order(sales$year, sales$month),]

### row.names(sales) <- NULL

### return(sales)

### } # Now create the sales fact table

### sales\_fact <- gen\_sales(500) # Look at a few records

### head(sales\_fact)

### # Build up a cube

### revenue\_cube <- tapply(sales\_fact$amount, sales\_fact[,c("prod", "month", "year", "loc")], FUN=function(x){return(sum(x))})

### # Showing the cells of the cube

### revenue\_cube

### dimnames(revenue\_cube)

### # Slice

### # cube data in Jan, 2012

### revenue\_cube[, "1", "2012",]

### # cube data in Jan, 2012

### revenue\_cube["Tablet", "1", "2012",]

### # Dice

### revenue\_cube[c("Tablet","Laptop"), c("1","2","3"), , c("CA","NY")]

### # Rollup

### apply(revenue\_cube, c("year", "prod"), FUN=function(x) {return(sum(x, na.rm=TRUE))})

**OUTPUT:**

month year loc prod unit amount

1 1 2021 CA Camera 1 NA

2 1 2021 CA Camera 1 NA

3 1 2021 QU Blutooth 1 NA

4 1 2021 QU Blutooth 1 NA

5 1 2021 QU Camera 2 NA

6 1 2021 QU Camera 1 NA

, , year = 2021, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

$prod

[1] "Blutooth" "Camera" "Mobile"

$month

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

$year

[1] "2021" "2022"

$loc

[1] "CA" "NY" "ON" "QU" "WA"

**Exercise:2.1**

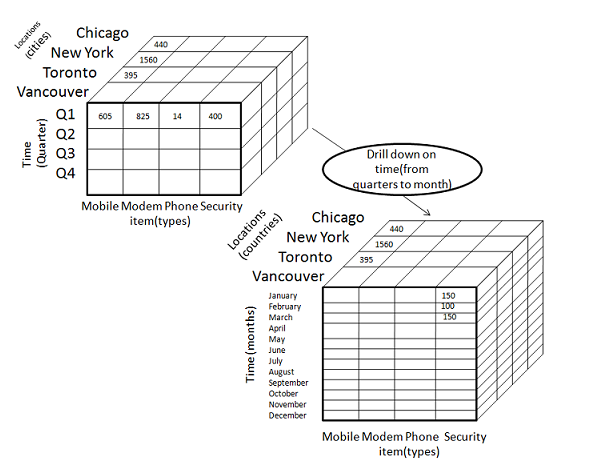
### Extracting the data using sales data with following Product name Mobile, Camera and Bluetooth prices 300, 400 and 500 performing various OLAP operations like (Drill-Down & Pivot) and visualizing the multidimensional data for analysis.

### Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways −

* By stepping down a concept hierarchy for a dimension
* By introducing a new dimension.

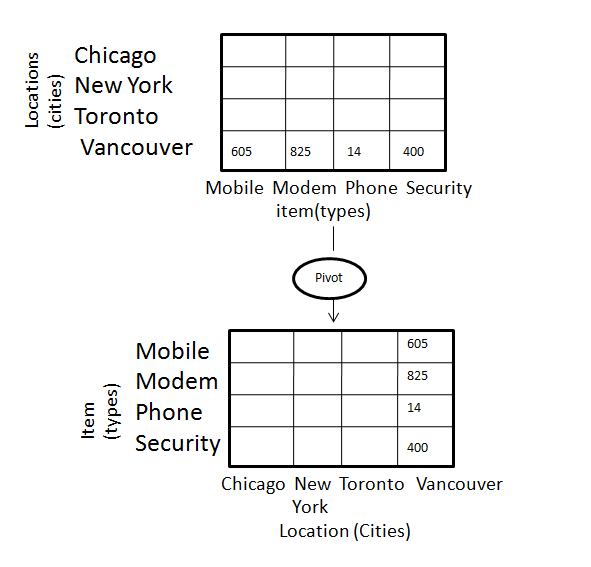
The following diagram illustrates how drill-down works



* Drill-down is performed by stepping down a concept hierarchy for the dimension time.
* Initially the concept hierarchy was "day < month < quarter < year."
* On drilling down, the time dimension is descended from the level of quarter to the level of month.
* When drill-down is performed, one or more dimensions from the data cube are added.
* It navigates the data from less detailed data to highly detailed data.

### Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



**COURSE CODE:**

### ## credits

### # https://dzone.com/articles/olap-operation-r

### # Setup the dimension tables

### state\_table <- data.frame(key=c("CA", "NY", "WA", "ON", "QU"),

### name=c("California", "new York", "Washington", "Ontario", "Quebec"),

### country=c("USA", "USA", "USA", "Canada", "Canada"))

### month\_table <- data.frame(key=1:12,

### desc=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),

### quarter=c("Q1","Q1","Q1","Q2","Q2","Q2","Q3","Q3","Q3","Q4","Q4","Q4"))

### prod\_table <- data.frame(key=c("Mobile", "Camera", "Blutooth"), price=c(300, 400, 500))

### # Function to generate the Sales table

### gen\_sales <- function(no\_of\_recs) {

### # Generate transaction data randomly

### loc <- sample(state\_table$key, no\_of\_recs, replace=T, prob=c(2,2,1,1,1))

### time\_month <- sample(month\_table$key, no\_of\_recs, replace=T)

### time\_year <- sample(c(2022, 2021), no\_of\_recs, replace=T)

### prod <- sample(prod\_table$key, no\_of\_recs, replace=T, prob=c(1, 3, 2))

### unit <- sample(c(1,2), no\_of\_recs, replace=T, prob=c(10, 3))

### amount <- unit\*prod\_table[prod,]$price

### sales <- data.frame(month=time\_month, year=time\_year, loc=loc, prod=prod, unit=unit, amount=amount)

### # Sort the records by time order

### sales <- sales[order(sales$year, sales$month),]

### row.names(sales) <- NULL

### return(sales)

### } # Now create the sales fact table

### sales\_fact <- gen\_sales(500) # Look at a few records

### head(sales\_fact)

### # Build up a cube

### revenue\_cube <- tapply(sales\_fact$amount, sales\_fact[,c("prod", "month", "year", "loc")], FUN=function(x){return(sum(x))})

### # Showing the cells of the cube

### revenue\_cube

### dimnames(revenue\_cube)

### # Drilldown

### apply(revenue\_cube, c("year", "month", "prod"), FUN=function(x) {return(sum(x, na.rm=TRUE))})

### # Pivot

### apply(revenue\_cube, c("year", "month"), FUN=function(x) {return(sum(x, na.rm=TRUE))})

### apply(revenue\_cube, c("prod", "loc"), FUN=function(x) {return(sum(x, na.rm=TRUE))})

**OUTPUT:**

month year loc prod unit amount

1 1 2021 CA Blutooth 2 NA

2 1 2021 WA Mobile 1 NA

3 1 2021 NY Camera 1 NA

4 1 2021 CA Camera 1 NA

5 1 2021 CA Blutooth 1 NA

6 1 2021 NY Camera 1 NA

, , year = 2021, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = CA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = NY

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = ON

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = QU

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2021, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

, , year = 2022, loc = WA

month

prod 1 2 3 4 5 6 7 8 9 10 11 12

Blutooth NA NA NA NA NA NA NA NA NA NA NA NA

Camera NA NA NA NA NA NA NA NA NA NA NA NA

Mobile NA NA NA NA NA NA NA NA NA NA NA NA

$prod

[1] "Blutooth" "Camera" "Mobile"

$month

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

$year

[1] "2021" "2022"

$loc

[1] "CA" "NY" "ON" "QU" "WA"

, , prod = Blutooth

month

year 1 2 3 4 5 6 7 8 9 10 11 12

2021 0 0 0 0 0 0 0 0 0 0 0 0

2022 0 0 0 0 0 0 0 0 0 0 0 0

, , prod = Camera

month

year 1 2 3 4 5 6 7 8 9 10 11 12

2021 0 0 0 0 0 0 0 0 0 0 0 0

2022 0 0 0 0 0 0 0 0 0 0 0 0

, , prod = Mobile

month

year 1 2 3 4 5 6 7 8 9 10 11 12

2021 0 0 0 0 0 0 0 0 0 0 0 0

2022 0 0 0 0 0 0 0 0 0 0 0 0

month

year 1 2 3 4 5 6 7 8 9 10 11 12

2021 0 0 0 0 0 0 0 0 0 0 0 0

2022 0 0 0 0 0 0 0 0 0 0 0 0

loc

prod CA NY ON QU WA

Blutooth 0 0 0 0 0

Camera 0 0 0 0 0

Mobile 0 0 0 0 0