**1-Create Database & Import Data**

--First we need to create a database 'DataScience'

Create Database DataScience

--After we create the database we need to access it

Use DataScience

--The let's create a schema ('Statistics' & 'Wrangle') to organize the tables

Create Schema [Statistics]

Create Schema Wrangle

----

--Let's import manually the data

--Explore the columns of the table by running the below or simply click Alt+F1 by highlighting the table

SELECT Column\_Name

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = N'train'

**2-Explore the Data.sql**

--Exploring the table 'train' by running select

Select \* From train

--Exploring the top 10 rows from the table 'train'

Select top 10 \* From train

**3-Check Aggregate functions**

--Let's explore some aggregate functions

--Check Average of age

Select Avg(age) as avg\_age

From train

--Check minimum and maximum of age

Select

min(age) as min\_age

,max(age) as max\_age

From train

--Check total rows of the table

Select

count(\*) as Nb\_Count

From train

--Check distinct value of 'Sex' column

Select

Distinct Sex

From train

--Check distinct count of column

Select

COUNT (DISTINCT Sex) as Count\_Distinct

From train

**4-Check Data Distribution Part1**

--https://www.mssqltips.com/sqlservertutorial/9128/sql-server-statistical-window-functions-percentiledisc-and-percentilecont/

--Get Statistics for Pclass column (STDEVP,min,max,AVG)

Select Distinct

'Pclass' as ColumnName

,STDEVP(Pclass) std

,min(Pclass) as min

,max(Pclass) as max

,Avg(Pclass) as mean

,AVG(CAST(Pclass AS DECIMAL(12,2))) as mean

From train

--Get Percentile for Age column

Select distinct

'Age' as ColumnName

,PERCENTILE\_CONT(0.25) WITHIN GROUP (ORDER BY age) OVER (PARTITION BY 1) as '25%'

,PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY age) OVER (PARTITION BY 1) as '50%'

,PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY age) OVER (PARTITION BY 1) as '75%'

From train

where age is not null

group by age

**4-Check Data Distribuation\_Part2\_Describe\_Nmeric\_Data**

--Drop table if exists '[Statistics].Data\_Aggregate' & '[Statistics].Data\_Percentile'

drop table if exists [Statistics].Data\_Aggregate

drop table if exists [Statistics].Data\_Percentile

--Create table [Statistics].Data\_Aggregate

CREATE TABLE [Statistics].Data\_Aggregate(

[ColumnName] [varchar](50) NOT NULL,

[Count] int,

[mean] [decimal](38, 6) NULL,

[std] [float] NULL,

[min] [int] NULL,

[max] [int] NULL

) ON [PRIMARY]

GO

CREATE TABLE [Statistics].Data\_Percentile(

[ColumnName] [varchar](50) NOT NULL,

[25%] decimal(18,10) NULL,

[50%] decimal(18,10) NULL,

[75%] decimal(18,10) NULL)

--------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------

Declare @Stat\_Aggregate nvarchar(max) ='

insert into [Statistics].[Data\_Aggregate] (ColumnName,Count,std,min,max,mean)

Select Distinct

''col\_name'' as ColumnName

,count(\*) Count

,STDEVP(col\_name) std

,min(col\_name) as min

,max(col\_name) as max

,AVG(CAST(col\_name AS DECIMAL(12,2))) as mean

From train'

Declare @Stat\_PERCENTILE nvarchar(max) ='

insert into [Statistics].[Data\_Percentile]

Select distinct

''col\_name'' as ColumnName

,PERCENTILE\_CONT(0.25) WITHIN GROUP (ORDER BY col\_name) OVER (PARTITION BY 1) as ''[25%]''

,PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY col\_name) OVER (PARTITION BY 1) as ''[50%]''

,PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY col\_name) OVER (PARTITION BY 1) as ''[75%]''

From train

where col\_name is not null

group by col\_name'

Declare @Query\_1 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','PassengerId')

Declare @Query\_2 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','Survived')

Declare @Query\_3 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','Pclass')

Declare @Query\_4 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','Age')

Declare @Query\_5 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','SibSp')

Declare @Query\_6 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','Parch')

Declare @Query\_7 nvarchar(max)= replace(@Stat\_Aggregate,'col\_name','Fare')

exec (@Query\_1)

exec (@Query\_2)

exec (@Query\_3)

exec (@Query\_4)

exec (@Query\_5)

exec (@Query\_6)

exec (@Query\_7)

Declare @Query\_1\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','PassengerId')

Declare @Query\_2\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','Survived')

Declare @Query\_3\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','Pclass')

Declare @Query\_4\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','Age')

Declare @Query\_5\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','SibSp')

Declare @Query\_6\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','Parch')

Declare @Query\_7\_Perc nvarchar(max)= replace(@Stat\_PERCENTILE,'col\_name','Fare')

exec (@Query\_1\_Perc)

exec (@Query\_2\_Perc)

exec (@Query\_3\_Perc)

exec (@Query\_4\_Perc)

exec (@Query\_5\_Perc)

exec (@Query\_6\_Perc)

exec (@Query\_7\_Perc)

go

drop table if exists [Statistics].Data\_Describe\_Numerical

Select AG.ColumnName

,[Count]

,[mean]

,[std]

,[min]

,[25%]

,[50%]

,[75%]

,[max]

into [Statistics].Data\_Describe\_Numerical

From [Statistics].Data\_Aggregate AG

inner join [Statistics].Data\_Percentile Perc

on Perc.ColumnName = AG.ColumnName

go

Select

\*

From [Statistics].Data\_Describe\_Numerical

**5-Check Distinct value in table**

--Check Distinct value in table

Declare @SQL nvarchar(max)

set @SQL = ''

;with cols as (

select Table\_Schema, Table\_Name, Column\_Name, Row\_Number() over(partition by Table\_Schema, Table\_Name

order by ORDINAL\_POSITION) as RowNum

from INFORMATION\_SCHEMA.COLUMNS

)

select @SQL = @SQL + case when RowNum = 1 then '' else ' union all ' end

+ ' select ''' + Column\_Name + ''' as Column\_Name, count(distinct ' + quotename (Column\_Name) + ' ) As DistinctCountValue,

count( '+ quotename (Column\_Name) + ') as CountValue FROM ' + quotename (Table\_Schema) + '.' + quotename (Table\_Name)

from cols

where Table\_Name = 'train' --print @SQL

exec (@SQL)

**6-Check Data Distribuation\_Categorical**

--Drop table if exists '[Statistics].Data\_describe\_Categorical

drop table if exists [Statistics].Data\_describe\_Categorical

--Combine all queries in one table using ';with [Alias] as ('

;with Data

as

(

Select Distinct

'count' as ColumnName

,(Select cast(Count(Sex) as varchar(50)) from train ) as Sex

,(Select cast(Count(Ticket) as varchar(50)) from train ) as Ticket

,(Select cast(Count(Cabin) as varchar(50)) from train ) as Cabin

,(Select cast(Count(Embarked) as varchar(50)) from train ) as Embarked

From train

union

Select Distinct

'unique' as ColumnName

,(Select cast(Count(distinct Sex)as varchar(50)) from train ) as Sex

,(Select cast(Count(distinct Ticket)as varchar(50)) from train ) as Ticket

,(Select cast(Count(distinct Cabin)as varchar(50)) from train ) as Cabin

,(Select cast(Count(distinct Embarked)as varchar(50)) from train ) as Embarked

From train

union

Select Distinct

'top' as ColumnName

,(Select top 1 Sex from train where Sex is not null group by Sex order by count(\*) desc) as Sex

,(Select top 1 Ticket from train where Ticket is not null group by Ticket order by count(\*) desc) as Ticket

,(Select top 1 Cabin from train where Cabin is not null group by Cabin order by count(\*) desc) as Cabin

,(Select top 1 Embarked from train where Embarked is not null group by Embarked order by count(\*) desc) as Embarked

From train

union

Select

'freq' as ColumnName

,(Select distinct cast(MAX(count(Sex)) OVER () as varchar(50)) From train group by Sex) as 'Sex'

,(Select distinct cast(MAX(count(Ticket)) OVER () as varchar(50)) From train group by Ticket) as 'Ticket'

,(Select distinct cast(MAX(count(Cabin)) OVER () as varchar(50)) From train group by Cabin) as 'Cabin'

,(Select distinct cast(MAX(count(Embarked)) OVER () as varchar(50)) From train group by Cabin) as 'Embarked'

)

--Insert the result into a new table [Statistics].Data\_Describe\_Categorical

Select

\*

into [Statistics].Data\_Describe\_Categorical

From Data

--Explore table [Statistics].Data\_Describe\_Categorical

Select \* From [Statistics].Data\_Describe\_Categorical

**7-Check Correlating**

--Find correlation between Age and survival rate

Select

Age,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Train

group by Age

order by Age asc

--Find correlation between Pclass and survival rate

Select

Pclass,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean,

(Select count(\*) from Train T where T.Pclass = A.Pclass) as Count

From Train A

group by Pclass

order by mean desc

--Find correlation between Sex and survival rate

Select

Sex,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean,

(Select count(\*) from Train T where T.Sex = A.Sex) as Count

From Train A

group by Sex

order by mean desc

--Find correlation between SibSp and survival rate

Select

SibSp,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Train

group by SibSp

order by mean desc

--Find correlation between Parch and survival rate

Select

Parch,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Train

group by Parch

order by mean desc

**8-Explore Data Statistics**

--Lets explore the table [Statistics].Data\_Describe\_Numerical

Select \* From [Statistics].Data\_Describe\_Numerical

**9-Wrangle data**

drop table if exists Wrangle.train

drop table if exists Wrangle.test

go

--Take a copy of the table 'train' and 'test' into another schema 'Wrangle'

Select \* into Wrangle.train From train

Select \* into Wrangle.test From test

--Check the result

Select top 2 \* From Wrangle.train

**10-Drop features Ticket and Cabin**

--Drop columns from 'Wrangle.train' and 'Wrangle.train'

--Drop features Ticket and Cabin if exists by altering the tables and droping the column

alter table Wrangle.train drop column if exists Ticket

alter table Wrangle.train drop column if exists Cabin

alter table Wrangle.test drop column if exists Ticket

alter table Wrangle.test drop column if exists Cabin

**11-Create Regular Expression Function**

**12-Feature extraction – Title**

--Add Title column

alter table Wrangle.train drop column if exists Title

alter table Wrangle.Test drop column if exists Title

alter table Wrangle.train add Title varchar(50) null

alter table Wrangle.test add Title varchar(50) null

--Extract Title from name using 'RegexMatch' function by passing the variables (Name and '([A-Za-z]+)\.')

Select

dbo.RegexMatch([Name], '([A-Za-z]+)\.') as Title

,Name

From Wrangle.train

----------------------------------------------------

--Update Title by running the regular explression

Declare @regex\_pattern varchar(max) = '([A-Za-z]+)\.'

Update Wrangle.Train Set Title = replace(dbo.RegexMatch([Name], @regex\_pattern),'.','')

Update Wrangle.Test Set Title = replace(dbo.RegexMatch([Name], @regex\_pattern),'.','')

----------------------------------------

--Explorer title count grouped

Select

Title

,Sex

,Count(\*) as TotalNb

From Wrangle.Train

Group by Title,Sex

order by Sex,TotalNb desc

go

----------------------------------------

--Replace common name with Rare

Select

Title

,Count(\*) as TotalNb

From Wrangle.Train

Group by Title

order by TotalNb desc

--# We can convert the categorical titles to ordinal.

UPDATE Wrangle.Train SET Title =

CASE

WHEN Title = 'Mr' THEN '1'

WHEN Title = 'Miss' THEN '2'

WHEN Title = 'Mrs' THEN '3'

WHEN Title = 'Master' THEN '4'

WHEN Title = 'Rare' THEN '5'

END;

UPDATE Wrangle.Test SET Title =

CASE

WHEN Title = 'Mr' THEN '1'

WHEN Title = 'Miss' THEN '2'

WHEN Title = 'Mrs' THEN '3'

WHEN Title = 'Master' THEN '4'

WHEN Title = 'Rare' THEN '5'

END;

--Now we can drop the column name after extracting the title

alter table Wrangle.Train drop column if exists [Name]

---Update Title , replace it with Rare 5 when empty

Update Wrangle.train Set Title = '5' Where Title is null

Update Wrangle.train Set Title = '5' Where Title is null

Select top 5 \*

From Wrangle.Train

**13-Converting a categorical feature**

--Convert Female and Male to integer

--Explore Wrangle.train top 5 rows

Select

top 5 \*

From Wrangle.train

---Update Wrangle.train and Wrangle.test set Sex = 1 when female and 0 when male

Update Wrangle.[train] set Sex = '1' where Sex ='female'

Update Wrangle.[train] set Sex = '0' where Sex ='male'

Update Wrangle.test set Sex = '1' where Sex ='female'

Update Wrangle.test set Sex = '0' where Sex ='male'

Select

top 5 \*

From Wrangle.train

**14-Create Combined View**

--Drop view Wrangle.vwTrainTest is exists

drop view if exists Wrangle.vwTrainTest

Create view Wrangle.vwTrainTest

as

Select

PassengerId

,Survived

,Pclass

--,[Name]

,Sex

,Age

,SibSp

,Parch

,Fare

,Embarked

,Title

,'1' as isTrain

From Wrangle.[train]

union

Select

PassengerId

,null as Survived

,Pclass

--,[Name]

,Sex

,Age

,SibSp

,Parch

,Fare

,Embarked

,Title

,'0' as isTrain

From Wrangle.test

**15-Fill Empty Age by mean**

Select \* from Wrangle.vwTrainTest

where PassengerId = 6

Drop table if exists #Age\_Mean

--Create a new temp table as Age\_Mean

Select

Sex,

Pclass,

AVG(age) as Mean

into #Age\_Mean

From Wrangle.vwTrainTest

where age is not null

group by Sex,Pclass

order by sex

Update T

Set T.Age = cast(AM.Mean as int)

From Wrangle.train T

inner join #Age\_Mean AM

on AM.Sex= T.Sex and AM.PClass = T.Pclass

where T.Age is null

Update T

Set T.Age = cast(AM.Mean as int)

From [Wrangle].Test T

inner join #Age\_Mean AM

--Join on Sex and PClass

on AM.Sex= T.Sex and AM.PClass = T.Pclass

where T.Age is null

Select \* from Wrangle.vwTrainTest

where PassengerId = 6

**16-Create Age Band**

--Get Minimum age and Maximum

Select

min(age) as Age\_Minimum,

max(age) as Age\_Maximum

From Wrangle.vwTrainTest

--We will divide the age by 5 range

Select

\*,

Case

when Age > 0 and Age <= 16 then '0'

when Age > 16 and Age <= 32 then '1'

when Age > 0 and Age <= 48 then '2'

when Age > 0 and Age <= 64 then '3'

else '4' End AgeBand

From Wrangle.Train

--Update Age to replace age band

Update T

Set T.Age = Case

when Age > 0 and Age <= 16 then '0'

when Age > 16 and Age <= 32 then '1'

when Age > 0 and Age <= 48 then '2'

when Age > 0 and Age <= 64 then '3'

else '4' End

From Wrangle.Train T

Update T

Set T.Age = Case

when Age > 0 and Age <= 16 then '0'

when Age > 16 and Age <= 32 then '1'

when Age > 0 and Age <= 48 then '2'

when Age > 0 and Age <= 64 then '3'

else '4' End

From Wrangle.Test T

Select

top 5 \*

From Wrangle.Train

**17-Create Family Size**

--We will create new feature called FamilySize

alter table Wrangle.train drop column if exists FamilySize

alter table Wrangle.Test drop column if exists FamilySize

alter table Wrangle.train add FamilySize int

alter table Wrangle.test add FamilySize int

--Will update the value by adding SibSp + Parch + 1

Update Wrangle.train Set FamilySize = SibSp + Parch + 1

Update Wrangle.test Set FamilySize = SibSp + Parch + 1

Select top 5 \* From Wrangle.train

**18-Explore Family size survived**

Select

FamilySize,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Wrangle.train

group by FamilySize

order by mean desc

**19-Create feature isAlone**

--Create feature called IsAlone for passengers who don't have children/spouses/parents/siblings

alter table Wrangle.train drop column if exists IsAlone

alter table Wrangle.Test drop column if exists IsAlone

alter table Wrangle.train add IsAlone int

alter table Wrangle.test add IsAlone int

--Reset the flag

Update Wrangle.train Set IsAlone = 0

Update Wrangle.train Set IsAlone = 0

Update Wrangle.train Set IsAlone = 1 where FamilySize = 1

Update Wrangle.test Set IsAlone = 1 where FamilySize = 1

--Find correlation between isAlone and survival rate

Select

IsAlone,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Wrangle.train

group by IsAlone

order by mean desc

Select top 5 \* From Wrangle.train

**20-Drop columns parch, Sibso and FamilySize**

--# remove Parch and SibSp and FamilySize

alter table Wrangle.train drop column Parch,SibSp,FamilySize

alter table Wrangle.test drop column Parch,SibSp,FamilySize

Select top 5 \* From Wrangle.train

**21-Completing a categorical feature**

Select

Embarked,

Count(\*) as Total

From Wrangle.[train]

group by Embarked

order by Total desc

--So we fill the empty with S

Update Wrangle.train set Embarked ='S' where Embarked is null

--Check correlation

Select

Embarked,

AVG(CAST(Survived AS DECIMAL(12,2))) as mean

From Wrangle.train

group by Embarked

order by mean desc

**22-Convert categorical feature to numeric**

--Now we need to convert Embarked to integer

Select

top 5 \*

From Wrangle.[train]

Update Wrangle.train set Embarked = '0' where Embarked ='S'

Update Wrangle.train set Embarked = '1' where Embarked ='C'

Update Wrangle.train set Embarked = '2' where Embarked ='Q'

Update Wrangle.test set Embarked = '0' where Embarked ='S'

Update Wrangle.test set Embarked = '1' where Embarked ='C'

Update Wrangle.test set Embarked = '2' where Embarked ='Q'

Select

top 5 \*

From Wrangle.[train]

**23-Quick completing and converting a numeric feature**

--Let explore the fare value , we have empty value in the test.

Select

\*

From Wrangle.test

where Fare is null

--https://sqlperformance.com/2012/08/t-sql-queries/median+

Declare @Median decimal(18,2)

SELECT @Median = PERCENTILE\_CONT(0.5)

WITHIN GROUP (ORDER BY fare) OVER ()

FROM Wrangle.Test

Select @Median

--Update Empty fare with the median

Update Wrangle.Test

Set Fare ='14.45'

where fare is null

**24-Create FareBrand feature**

;With Data

as

(

Select

Fare,

NTILE(4) OVER(ORDER BY Fare asc) AS Quartile

From [Wrangle].[train]

)

Select

Min(Fare) as Fare\_Min,

Max(Fare) as Fare\_max,

Quartile

From Data

Group by Quartile

go

--alter table Wrangle.train drop column FareBand int

--alter table Wrangle.test drop column FareBand int

--alter table Wrangle.train add FareBand int

--alter table Wrangle.test add FareBand int

--select \* into Wrangle.Train\_Test from Wrangle.Train

/\*

select \* into Wrangle.Train from Wrangle.Train\_Test

drop table Wrangle.Train

\*/

go

Update T

Set T.Fare = Case

when Fare >= 0 and Fare <= 7.8958 then '0'

when Fare > 7.8958 and Fare < 14.4542 then '1'

when Fare >= 14.4542 and Fare <= 31 then '2'

else '3' End

From Wrangle.Train T

go

Update T

Set T.Fare = Case

when Fare >= 0 and Fare <= 7.8958 then '0'

when Fare > 7.8958 and Fare < 14.4542 then '1'

when Fare >= 14.4542 and Fare <= 31 then '2'

else '3' End

From Wrangle.Test T

go

Select

top 5 \*

From Wrangle.Train