1) What is the distribution of the total number of air-travelers per year

2) What is the total air distance covered by each user per year

3) Which user has travelled the largest distance till date

4) What is the most preferred destination for all users.

5)Which route is generating the most revenue per year

6) What is the total amount spent by every user on air-travel per year

7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most

every year.

Task 1: What is the distribution of the total number of air-travelers per year

```
val JoinUserHoliday = spark.sql("select h.*, u.UserID, u.UserName from Userdetails_cls
u join holiday_cls h on u.UserID = h.UserID")
    //1
val groupyearwise = JoinUserHoliday.groupBy("Year").count.alias("").show()
```

```
+----+-----+
|Year|count|
+----+-----+
|1990|    8|
|1994|    1|
|1991|    9|
|1992|    7|
|1993|    7|
+----+-----+
```

Task 2: What is the total air distance covered by each user per year

```
val groupyearwisesum = JoinUserHoliday.groupBy("Year", "UserName").sum("Dist").show()
```

```
+----+--------+---------+
|Year|UserName|sum(Dist)|
+----+--------+---------+
|1991|    mark|      200|
|1990|  andrew|      200|
|1991|  andrew|      200|
|1991|    luke|      200|
|1993|    mark|      600|
|1991|   peter|      400|
|1993|    luke|      200|
|1991|  thomas|      200|
|1993|    john|      200|
|1991|    john|      400|
|1990|   annie|      200|
|1994|    mark|      200|
|1990|    mark|      200|
|1990|    lisa|      400|
|1992|  thomas|      400|
|1990|   james|      600|
|1993|   peter|      200|
|1991|    lisa|      200|
|1992|    luke|      200|
|1992|   annie|      200|
+----+--------+---------+
only showing top 20 rows
```

Task 3: Which user has travelled the largest distance till date

```
val mostPreferredDest = JoinUserHoliday.groupBy("UserName").sum("Dist")
val xg = mostPreferredDest.orderBy(desc("sum(Dist)")).first()
println("first "+xg)
```

```
first [mark,1600]
```

Task 4 : What is the most preferred destination for all users.

```
val x =  JoinUserHoliday.toDF().groupBy("TCountry").count()
val xdf = x.orderBy(desc("count")).first()
println("first "+xdf)
```

```
first [IND,9]
```

Task 5: Which route is generating the most revenue per year

```
val distinctroute = spark.sql("select h.FCountry,h.TCountry, h.FCountry + h.TCountry
as route,h.Year, h.Dist,t.ModeTravel, t.Amount from Transport_cls t join holiday_cls h
on t.ModeTravel = h.ModeTravel")
val Groupbydistinctroute =
distinctroute.groupBy("FCountry","TCountry","Year").sum("Amount").orderBy(desc("sum(Am
ount)")).toDF().show()
```

```
+--------+--------+----+-----------+
|FCountry|TCountry|Year|sum(Amount)|
```

```
+--------+--------+----+-----------+
|     IND|     RUS|1991|        340|
|     CHN|     IND|1993|        340|
|     IND|     AUS|1991|        340|
|     AUS|     CHN|1993|        340|
|     RUS|     IND|1992|        340|
|     CHN|     RUS|1992|        340|
|     CHN|     IND|1990|        340|
|     CHN|     AUS|1990|        170|
|     AUS|     IND|1992|        170|
|     PAK|     AUS|1993|        170|
|     RUS|     CHN|1993|        170|
|     CHN|     PAK|1991|        170|
|     IND|     PAK|1991|        170|
|     PAK|     IND|1993|        170|
|     RUS|     CHN|1992|        170|
|     CHN|     PAK|1990|        170|
|     IND|     CHN|1992|        170|
|     CHN|     RUS|1990|        170|
|     RUS|     IND|1990|        170|
|     AUS|     CHN|1990|        170|
+--------+--------+----+-----------+
only showing top 20 rows
```

Task 6: What is the total amount spent by every user on air-travel per year

```
val task6 = spark.sql("select h.UserID, h.Year, sum(t.Amount) from Transport_cls t
join holiday_cls h on t.ModeTravel = h.ModeTravel group by h.UserID, h.Year")
    task6.show(numRows = 32)
```

UserID	Year	sum(Amount)
3	1991	170
6	1993	170
3	1992	170
7	1990	510
10	1993	170
6	1991	340
2	1991	340
4	1991	170
5	1991	170
5	1994	170
8	1991	170
1	1990	170
5	1992	340
4	1990	340
3	1993	170
10	1990	170
2	1993	170
1	1993	510
9	1991	170
9	1992	340
8	1990	170
10	1992	170
8	1992	170