

```

/*start_station_frequence select the most popular start
station*/
proc sql;
Create Table orion.start_station as
    select start_station,
           count(start_station) as start_station_frequence
    from orion.IMPORT_BIKESHARING
group by start_station
order by calculated start_station_frequence desc;
quit;
/*Find the top 10 popular start station*/
proc sql inobs=10;
Create Table orion.start_station_1 as
    select *
from orion.start_station
order by start_station_frequence desc;
quit;

/*end_station_frequence select the most populart end station*/
proc sql;
Create Table orion.end_station as
    select end_station,
           count(end_station) as end_station_frequence
    from orion.IMPORT_BIKESHARING
group by end_station
order by calculated end_station_frequence desc;
quit;
/*Find the top 10 popular end station*/
proc sql inobs=10;
Create Table orion.end_station_1 as
    select *
from orion.end_station;
quit;

/*Select the numbers of trip id in each station*/
proc sql;

```

```

create table orion.popular_station_time as
select hour(start_time,1) as time,
       start_station,
       count(trip_id) as Usage
from Orion.IMPORT_BIKESHARING
GROUP BY time,start_station
order by time;
quit;
/*find the max usage in each station*/
proc sql;
create table orion.popular_time_start_station_1 as
Select  time,
       start_station,
       max(usage) as max_usage
from orion.popular_station_time
group by time
HAVING usage=max_Usage;
quit;

/*count the numbers of trips by route_type*/
proc sql;
Create Table orion.trip_type as
       select trip_route_category,
       count (trip_id) as numbers_trips
from Orion.IMPORT_BIKESHARING
group by trip_route_category;
quit;

/*calculate the average_distance*/
proc sql;
create table orion.average_distance as
select avg(distance)*111 as average_distance
       from
       (select trip_ID,
              sum(Longitude_x,Latitude_y)      as      distance
format=comma10.2
       from (select trip_ID,
              ABS(start_lon-end_lon) as Longitude_x,
              ABS(start_lat-end_lat) as Latitude_y

```

```

        from (select trip_ID,
                    start_lon,
                    start_lat,
                    end_lon,
                    end_lat

                from orion.IMPORT_BIKESHARING
                where end_lon is not missing
                and end_lat is not missing

                )))
        where distance ne 0
        ;

quit;
/*clean_lon_lat for virtual station*/
proc sql;
create table orion.lon_lat_clean as
select trip_id,start_lat,start_lon,end_lat,end_lon
from orion.IMPORT_BIKESHARING
    where start_lat is not missing
    and start_lon is not missing
    and end_lat is not missing
    and end_lon is not missing;
quit;

/*find_out the distance by each trip_id*/
proc sql;
create table orion.sum_distance as
select trip_id,
        ABS(ABS(start_lat)-ABS(end_lat)) as Longitude_x,
        ABS(ABS(start_lon)-ABS(end_lon)) as Latitude_y
FROM orion.lon_lat_clean;
QUIT;
proc sql;
create table orion.distance as
    select trip_id,
        SUM(Longitude_X,Latitude_y)*111 as distance
format=comma10.2
from orion.sum_distance
where calculated distance ne 0
;
quit;

/*find the trip within or cross_city*/

```

```

proc sql;
create table orion.withincity as
select 'Total number within city' as Sum,
      count(trip_id) as Total,
      count(trip_id)/488481          as          Percentage
format=percent10.0
from orion.distance
where distance<5;
quit;

```

```

proc sql;
create table orion.crosscity as
      select 'Total number cross city' as Sum,
            count(trip_id) as Total,
            count(trip_id)/488481          as          Percentage
format=percent10.0
from orion.distance
where distance>=5;
quit;

```

```

proc sql;
create table orion.city as
      select *
from orion.withincity
union
      select *
from orion.crosscity;
quit;

```

```

/*find out the distribution of the bike_type*/
proc sql;
Create Table orion.bike_type as
      select bike_type,
            count(trip_id) as count
from orion.IMPORT_BIKESHARING
group by bike_type;
quit;

```

```

/*Average_distance_bike_type*/
proc sql;
create table orion.average_distance_type_bike as
      select avg(distance) as average_distance format=comma10.2,

```

```

        b.bike_type
from orion.distance as a
    inner join
orion.IMPORT_BIKESHARING as b
    on a.trip_id=b.trip_id
group by bike_type;
quit;

```

```

/*Time analysis*/
/*Query to calculate Time of the day with most rides on starting
time*/

```

```

proc sql;
create table orion.Higher_Demand_StartTime
as select hour(start_time) as Time,
        count(trip_id) as Frequency
    from orion.import_bikesharing
    group by Time
    order by Frequency desc;
quit;

```

```

/*Query to calculate Days with highest ride demand*/

```

```

proc sql;
create table orion.Highest_Demand_Days
as select weekday as day,
        count(weekday) as frequency

    from orion.import_bikesharing
    group by day
    order by day;
quit;

```

```

/*Holiday effects*/

```

```

/*Query to Union for holidays and bike rentals table*/

```

```

proc sql;
create table orion.holiday_effects as
select trip_id,
        start_date,
        holiday

    from orion.import_bikesharing
    inner join

```

```

        orion.import_holiday
    on
        (start_date = holidays_date)

    order by holidays_date;

quit;

/*Query to calculate the trips on a particular holiday*/
proc sql;
create table orion.holiday_effects_2 as
select distinct holiday,
               count(trip_id) as num_trips

from orion.holiday_effects
    group by holiday;
quit;

/*Query to join holidays and bike rental data to count normal
days trips without holidays*/
proc sql;
create table orion.without_holiday as
select distinct start_time,
               count (trip_id) as usage

from orion.import_bikesharing
where start_date not in
    (
        select distinct start_date

        from orion.import_bikesharing
        inner join
            orion.import_holiday
        on
            start_date = holidays_date
    )
group by start_date;
quit;

/*Query to calculate average trips on normal days without
counting holidays*/
proc sql;
create table orion.without_holiday_1 as
select 'Total number trips non holiday' as Sum,

```

```

        avg(usage) as average_usage    format=comma10.2

from orion.without_holiday;
quit;

/*Query to join holidays and bike rental data to count only
holidays trips*/
proc sql;
create table orion.with_holiday as
select distinct start_date,
                count(trip_id) as usage

from orion.import_bikesharing
inner join
    orion.import_holiday
on
    start_date = holidays_date

group by  start_date;
quit;

/*Query to calculate average trips only for holidays*/
proc sql;
create table orion.with_holiday_1 as
select 'Total number trips on holiday' as Sum,
        avg(usage) as average_usage    format=comma10.2
from orion.with_holiday;
quit;

/*Query to compare average trips - Union to analyse normal days
and holidays together*/
proc sql;
create table orion.compare as
select *
from orion.without_holiday_1
union
select *
from orion.with_holiday_1;
quit;

/*Weather Analysis*/

```

```

/*Create a table to illustrate the average temperature, average
level of humid, and average wind speed per month*/
proc sql;
create table oriondb.avg_temp_per_month as
select
    Year,
    Month,
    sum(temp_avg)/count(temp_avg)    as    temp_avg    format    =
FORMAT5.2,
    sum(humid_avg)/count(humid_avg)    as    humid_avg    format    =
Format5.2,
    sum(wind_avg)/count(wind_avg)    as    wind_avg    format    =
Format5.2
from oriondb.import_weather
group by Year, month
order by Year, month;
run;
/*Create a table to find out the number of trip*/
proc sql;
create table oriondb.trips as
select distinct
    year,
    month,
    count(day(datepart(start_time)))    as    Num_trips    format    =
comma10.
from oriondb.import_bikesharing
group by year,month;
run;
/*Combine all needed data into a unify table for the analysis
*/
proc sql;
create table oriondb.temp_analysis as
select
    A.Year,
    A.Month,
    Num_trips,
    Temp_avg,
    Humid_avg,
    Wind_avg
from oriondb.avg_temp_per_month as A
inner join
    oriondb.trips as B
on A.Year = B.Year and A.Month = B.Month
order by A.Year, A.Month;

```



```

quit;
proc sql;
Title "Performance Review 2018-2019";
select year,
       round(sum(num_trips)/count(num_trips)) as average_num_trip
format = comma10.2,
       max(num_trips) as max_num_trip format = comma10.2,
       min(num_trips) as min_num_trip format = comma10.2,
       round(sum(temp_avg)/count(temp_avg)) as avg_temp format =
comma10.2,
       max(temp_avg) as max_temp format = comma10.2,
       min(temp_avg) as min_temp format = comma10.2,
       round(sum(humid_avg)/count(humid_avg)) as avg_humid format
= comma10.2,
       max(humid_avg) as max_humid format = comma10.2,
       min(humid_avg) as min_humid format = comma10.2,
       round(sum(wind_avg)/count(wind_avg)) as avg_wind format =
comma10.2,
       max(wind_avg) as max_wind format = comma10.2,
       min(wind_avg) as min_wind format = comma10.2
from oriondb.temp_analysis
group by year;
run;

```

```

/*Customer analysis*/
/*Create a table to find out the shifting trend of the type of
product by year*/
proc sql;
create table oriondb.passholder_type as
select
year, Passholder_Type,
count(Trip_ID) as frequency format = comma10.
from oriondb.import_bikesharing
group by Passholder_Type, year
having passholder_type ne 'Testing'
order by year, frequency desc;
run;
/*Create a table to indicate the average amount of time (minute)
that customer spent on each type of passholder*/
proc sql;
create table oriondb.avg_duration_passtype as
select
passholder_type,
avg(duration) as duration format = comma10.

```

```
from oriondb.import_bikesharing
group by passholder_type
having passholder_type ne 'Testing'
order by duration desc;
quit;
```