```
/* Accessing data */
%let path = /folders/myfolders;

options validvarname=any;
libname Sales xlsx "&path/Data_Business.xlsx";

*****************Data Quality**************;

title "Data quality assessment report";

%MACRO assess_data(file,exclude_freq,exclude_univ);
proc means data=&file N Nmiss;
var _numeric_;
run;

/*Categorical*/
proc freq data=&file(drop=&exclude_freq);
*format DO year4.; /*Identify invalid date*/
table _character_/nocum nopercent;
run;

/*Statistical data analysis*/
proc univariate data=&file(drop=&exclude_univ);
run;

%MEND;

%assess_data(SALES.CUSTOMERDEMOGRAPHIC, first_name Last_name, customer_id);
%assess_data(SALES.TRANSACTIONS, , transaction_id customer_id product_id product_first_sold_date);
%assess_data(SALES.CUSTOMERADDRESS, address, customer_id postcode);

/*Invalid date values*/

%MACRO date_check(file, date);

proc freq data=&file(keep=&date);
format &date year4.;
table &date/nocum nopercent;
run;

%MEND;

%date_check(SALES.CUSTOMERDEMOGRAPHIC, DOB);
%date_check(SALES.TRANSACTIONS, product_first_sold_date transaction_date);

/*Duplicate values*/

%Macro dup_check(Table, ID);

proc sql;
    select &ID, count(&ID) as Value_Count
    from &Table
    group by &ID
    having Value_Count > 1
    order by Value_Count desc;
quit;

%MEND;

%dup_check(Sales.CUSTOMERDEMOGRAPHIC, customer_id);
%dup_check(Sales.TRANSACTIONS, customer_id);
%dup_check(Sales.CUSTOMERADDRESS, customer_id);

******************Data Cleaning*******************;

/*Demographics Dataset*/
```

```
/*Converting DOB to Age*/
data data_demo;
    set sales.customerdemographic;
    Age = year(today())-year(DOB);
run;

/*Continuous variable*/
proc stdize data=data_demo out=data_demo method=median reponly;
var tenure age;
run;

/*Inconsistent values*/
data data_demo;
    set data_demo;
    if gender in ("F", "Femal", "Female") then gender="Female";
    else gender="Male";
run;

/*Transactions Dataset*/

data data_trans;
    set sales.transactions;
    attrib product_first_sold_date informat=ddmmyy10. format=ddmmyy10.;
    Product_active_years = year(today())-year(product_first_sold_date);
run;

/*Continuous variable*/
proc stdize data=data_trans out=data_trans method=median reponly;
var standard_cost Product_active_years;
run;

/*Address Dataset*/

data data_add;
    set sales.customeraddress(drop=);
    if state in ("NSW", "New South Wales") then state="New South Wales";
    else if state in ("VIC", "Victoria") then state="Victoria";
    else state="Queensland";
run;

/*To Check if datasets are modified correctly*/
%assess_data(data_demo, first_name Last_name, customer_id);
%assess_data(data_trans, , transaction_id customer_id product_id product_first_sold_date);
%assess_data(data_add, address, customer_id postcode);

**********Partional Clustering**********;

proc sql;
    create table cust_value as
        select customer_id,
               count(customer_id) as Total_purchases,
               sum(list_price) as Total_amount,
               avg(list_price) as Average_amount
        from
               Sales.TRANSACTIONS
        group by
               customer_id;
quit;

/*Customer Importance*/
data cust_value;
    set cust_value;

    if Total_amount>5000 then Importance="High Value";
    else if Total_amount>2000 then Importance="Medium Value";
    else Importance="Low Value";
run;
```

```
proc print data=cust_value(obs=50);
run;

/*Joining two tables*/

proc sql;
    create table sales_data as
        select d.customer_id,
               d.past_3_years_bike_related_purcha as past_purchases,
               c.Total_amount,
               c.Total_purchases
        from
            data_demo d
        inner join
            cust_value c
        on d.customer_id = c.customer_id;
quit;

proc sql;
    create table sales as
        select *
        from
            data_demo d
        inner join
            cust_value t on d.customer_id = t.customer_id
        inner join
            data_add a on t.customer_id = a.customer_id;
quit;

/*Standardardize the dataset*/

proc standard data=sales_data mean=0 std=1 out=data_scaled;
run;

proc print data=data_scaled;

/*K-Means Clustering using amount spent and number of purchases*/

ods graphics on;

proc cluster data=sales_data method=centroid ccc print=10 outtree=tree;
var past_purchases Total_amount;
run;

proc tree noprint ncl=4 out=out;
copy past_purchases Total_amount;
run;

proc candisc out = can noprint;
class cluster;
var past_purchases: Total_amount:;
run;

proc sgplot data = can;
title "Customer segmentation using cluster analysis";
scatter y = can2 x = can1 / group = cluster markerattrs=(symbol=circlefilled);
refline  0.0/ transparency=0.0 axis=y lineattrs=(color=black pattern=dash thickness=2);
refline -1.5/ transparency=0.0 axis=x lineattrs=(color=black pattern=dash thickness=2);
refline  2.5/ transparency=0.0 axis=x lineattrs=(color=black pattern=dash thickness=2);
refline  6.0/ transparency=0.0 axis=x lineattrs=(color=black pattern=dash thickness=2);
run;

/*K-means clustering using multiple features*/

data data_cluster;
    set sales(keep= past_3_years_bike_related_purcha tenure age total_purchases total_amount);

    idnum = _n_;
```

```
    rename past_3_years_bike_related_purcha=past_purchases ;

    if not cmiss(of _all_);

run;

ods graphics on;

proc surveyselect data=data_cluster out=traintest seed=1
    samprate=0.7 method=srs outall; /*70% training 30% testing*/
run;

data train;
    set traintest;
    if selected=1;
run;

data test;
    set traintest;
    if selected=0;
run;

/*Variables to be standardized*/

%let features = past_purchases tenure age total_purchases total_amount;

proc standard data=train out=clustvar mean=0 std=1;
var &features;
run;

/*K-means clustering*/

%macro kmeans(K);

proc fastclus data=clustvar out=outdata&K. outstat=cluststat&K.
              maxclusters=&k. maxiter=300;
              var &features;
run;

%mend;

%kmeans(1);
%kmeans(2);
%kmeans(3);
%kmeans(4);
%kmeans(5);
%kmeans(6);

/*Use Rsq to plot elbow curves*/

data clust1;
    set cluststat1;
    nclust=1;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;

data clust2;
    set cluststat2;
    nclust=2;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;
```

```
data clust3;
    set cluststat3;
    nclust=3;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;

data clust4;
    set cluststat4;
    nclust=4;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;

data clust5;
    set cluststat5;
    nclust=5;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;

data clust6;
    set cluststat6;
    nclust=6;

    if _type_='RSQ'; /*To extract rsq values*/

    keep nclust over_all; /*It contains rsq values*/
run;

/*All rsq values for all clusters*/
data clusrsquare;
    set clust1 clust2 clust3 clust4 clust5 clust6;
run;

/*Plot elbow curve: Examine results for 2, 3 and 5*/

proc sgplot data=clusrsquare;
           title "Elbow Curve";
           vline nclust/response=over_all lineattrs=(color=darkblue);
           yaxis grid;
run;

/*Got outdata&(K=3) using the macro*/

proc candisc data=outdata3 out=clustcan;
    class cluster;
    var &features;
run;
```