

```

/* Importation des tables */

/*Macro programme et macro variables*/

-----
%macro importation(table,chemin);

PROC IMPORT DATAFILE=&chemin
  DBMS=XLSX
  OUT=WORK.&table;
  GETNAMES=YES;
RUN;

%mend;

%importation(client,'/home/u59586376/Projet_SAS/client.xlsx');
%importation(commande,'/home/u59586376/Projet_SAS/commande.xlsx');
%importation(service,'/home/u59586376/Projet_SAS/service.xlsx');
%importation(point_vente,'/home/u59586376/Projet_SAS/point_vente.xlsx');
%importation(publicite,'/home/u59586376/Projet_SAS/publicite.xlsx');

/* PLAN DU PROGRAMME:
I. L'effet spatial
II. Moyens de communication et de publicité
III. Service offert aux clients
IV. La variété des produits proposés
V. Vue globale
*/

/* Pour chaque thème (variable) nous avons calculer des indicateurs pour évaluer l'effet de la variable sur le chiffre d'

/* I.L'EFFET SPACIAL */

-----
/* 1- Pour chaque point de vente, le nombre moyen de chaque type de produit */
Proc sql ;
  SELECT pv.nom, mean(co.pizza_qt) as moy_pizza, mean(co.sandwich_qt) as moy_sandwich, mean(co.escalope_qt) as moy_esca
  FROM client c
  JOIN point_vente pv
  ON pv.id = c.point_vente_id
  JOIN commande co
  ON c.id = co.client_id
  GROUP BY pv.nom;
Quit;

-----
/* 2- Pour chaque point de vente, le montant moyen dépensé sur chaque type de produit */
Proc sql ;
  SELECT pv.nom, mean(co.pizza_eu) as moy_pizza, mean(co.sandwich_eu) as moy_sandwich, mean(co.escalope_eu) as moy_esca
  FROM client c
  JOIN point_vente pv
  ON pv.id = c.point_vente_id
  JOIN commande co
  ON c.id = co.client_id
  GROUP BY pv.nom;
Quit;

-----
/* 3- Le nombre de fois qu'un canal particulier a été utilisé pour chaque point de vente */
Proc sql ;
  SELECT pv.nom, p.canal, COUNT(*) as nbr_visite
  FROM client c
  JOIN publicite p
  ON c.id = p.client_id
  JOIN point_vente pv
  ON pv.id = c.point_vente_id
  GROUP BY pv.nom, p.canal
  ORDER BY nbr_visite DESC;
Quit;

/* II.MOYENS DE COMMUNICATION ET DE PUBLICITE */

-----
/* 1- Le canal qui a été le plus fréquemment utilisé par la plupart des comptes */
Proc sql outobs=10;
  SELECT c.id, p.canal, COUNT(*) as utilisation_canal
  FROM client c
  JOIN publicite p
  ON c.id = p.client_id
  GROUP BY c.id, p.canal
  ORDER BY utilisation_canal DESC;
Quit;

-----
/* 2- Le nombre total de fois ou chaque type de canal de publicité a été utilisé */

```

```

Proc sql;
    SELECT p.canal, COUNT(*)
    FROM publicite p
    GROUP BY p.canal;
Quit;

/* 3- Toutes les informations concernant les personnes qui ont été contactées via le canal Facebook ou Twitter et qui ont
Proc sql;
    SELECT *
    FROM publicite
    WHERE canal IN (facebook, twitter) AND temps_visite BETWEEN '2016-01-01' AND '2017-01-01'
    ORDER BY temps_visite DESC;
Quit;

/* 4- Le nombre moyen d'événements par jour pour chaque canal */
Proc sql ;
    SELECT canal, mean(visite) as moy_visite
    FROM (SELECT DAY(temps_visite) as day,
                canal, COUNT(*) as visite
            FROM publicite
            GROUP BY 1,2) sub
    GROUP BY canal
    ORDER BY 2 DESC;
Quit;

/* 5- Pour le client qui a dépensé le plus (au total au cours de sa vie en tant que client) total_eu, combien de visites
/*Requêtes imbriquées (nested queries)
Proc sql;
    SELECT c.id, p.canal, COUNT(*)
    FROM client c
    JOIN publicite p
    ON c.id = p.client_id AND c.id = (SELECT id
                                    FROM (SELECT c.id, SUM(co.total_eu) as tot_dep
                                        FROM commande co
                                        JOIN client c
                                        ON c.id = co.client_id
                                        GROUP BY c.id
                                        ORDER BY 2 DESC) inner_table)

    GROUP BY 1, 2
    ORDER BY 3 DESC;
Quit;*/
/* OU BIEN LA METHODE */
proc sql outobs=1;
    create table tbl as
    SELECT c.id, SUM(co.total_eu) as tot_dep
    FROM commande co
    JOIN client c
    ON c.id = co.client_id
    GROUP BY 1
    ORDER BY 2 DESC;
quit;
proc sql;
    SELECT c.id, p.canal, COUNT(*)
    FROM client c
    JOIN publicite p
    ON c.id = p.client_id and c.id = 3411 /*tbl.id*/
    GROUP BY 1, 2
    ORDER BY 3 DESC;
Quit;

/* III.SERVICE OFFERT AUX CLIENTS */

/* 1- Le type de service dans chaque point de vente avec le plus grand montant de ventes total_eu */
/*Requêtes imbriquées (nested queries)
Proc sql ;
WITH t1 AS (
    SELECT s.type as s_type, pv.nom as pv_nom, SUM(co.total_eu) as total_eu
    FROM service s
    JOIN commande co
    ON s.commande_id = co.id
    JOIN client c
    ON co.client_id = c.id
    JOIN point_vente pv
    ON pv.id = c.point_vente_id
    GROUP BY 1,2
    ORDER BY 3 DESC),
t2 AS (
    SELECT pv_nom, MAX(total_eu) as total_eu
    FROM t1
    GROUP BY 1)
SELECT t1.s_type, t1.pv_nom, t1.total_eu

```

```

FROM t1
JOIN t2
ON t1.pv_nom = t2.pv_nom AND t1.total_eu = t2.total_eu;
Quit ;*/
/* OU BIEN LA METHODE */
Proc sql ;
    create table t1 as
        SELECT s.type as s_type, pv.nom as pv_nom, SUM(co.total_eu) as total_eu
        FROM service s
        JOIN commande co
        ON s.commande_id = co.id
        JOIN client c
        ON co.client_id = c.id
        JOIN point_vente pv
        ON pv.id = c.point_vente_id
        GROUP BY 1,2
        ORDER BY 3 DESC;

quit;
Proc sql ;
    create table t2 as
        SELECT pv_nom, MAX(total_eu) as total_eu
        FROM t1
        GROUP BY 1;

quit;
proc sql;
    SELECT t1.s_type, t1.pv_nom, t1.total_eu
    FROM t1
    JOIN t2
    ON t1.pv_nom = t2.pv_nom AND t1.total_eu = t2.total_eu;
Quit ;

/* IV.LA VARIETE DES PRODUITS PROPOSES */

/* 1- Le pourcentage des revenus provenant de chaque produit pour chaque commande */
Proc sql outobs=10;
    SELECT id, client_id,
        pizza_eu/(pizza_eu + sandwich_eu + escalope_eu + 0.01) as pizza_per, sandwich_eu /(pizza_eu + sandwich_eu + escalope_eu) as sandwich_per
    FROM commande;
Quit;

/* 2- La quantité totale de chaque produit commandées dans le tableau des commandes */
Proc sql;
    SELECT SUM(pizza_qt) as total_pizza_vente, SUM(escalope_qt) as total_escalope_vente, SUM(sandwich_qt) as total_sandwich_vente
    FROM commande;
Quit;

/* le nombre moyen par commande par chaque type de produit, ainsi que le montant moyen apporté par chaque type de produit */
Proc sql;
    SELECT mean(pizza_qt) as mean_pizza, mean(sandwich_qt) as mean_sandwich,
        mean(escalope_qt) as mean_escalope, mean(pizza_eu) as mean_pizza_eu,
        mean(sandwich_eu) as mean_sandwich_eu, mean(escalope_eu) as mean_escalope_eu
    FROM commande;
Quit;

/* V.ANALYSE GLOBALE DES TABLES*/

/* 1- Propriétés des variables*/
proc means data = client; run;
proc means data = commande; run;
proc means data = point_vente; run;
proc means data = publicite; run;
proc means data = service; run;

/* 2- Le chiffre d'affaires pour toutes les commandes de chaque année, classées du plus grand au moins*/
proc sql ;
    SELECT year(temps_com) as commande_year, SUM(total_eu) as total_vente
    FROM commande
    GROUP BY 1
    ORDER BY 2 DESC;
quit ;

/* 3- Les mois ou les restaurants ont réalisé les ventes les plus importantes. Tous les mois sont-ils uniformément représentés ? */
proc sql ;
    SELECT MONTH(temps_com) as commande_month, SUM(total_eu) as total_vente
    FROM commande
    /*WHERE temps_com >= 01/01/2014 AND temps_com <= 01/01/2017 puisque dans la requete precedente nous avons trouvé que
    GROUP BY 1
    ORDER BY 2 DESC;
quit ;
/* 4- L'année ou les restaurants ont réalisé les ventes les plus importantes en termes de nombre total de commandes ? To

```

```
Proc sql;
  SELECT YEAR(temps_com) as commande_year, COUNT(*) as total_vente
  FROM commande
  GROUP BY 1
  ORDER BY 2 DESC;
quit;
/* 5- Le mois ou les restaurants ont réalisé les ventes les plus importantes en termes de nombre total de commandes ? Tou
Proc sql;
  SELECT MONTH(temps_com) as commande_month, COUNT(*) as total_vente
  FROM commande
  GROUP BY 1
  ORDER BY 2 DESC;
quit;
/* 6- Le client qui a le plus de commandes*/
Proc sql outobs=1;
  SELECT c.id, COUNT(*) as nbr_commande
  FROM client as c
  JOIN commande as co
  ON c.id = co.client_id
  GROUP BY c.id
  ORDER BY nbr_commande DESC;
quit;

/* 7- client qui a dépensé le plus */
Proc sql outobs=1;
  SELECT c.id, SUM(co.total_eu) as total_vente
  FROM client c
  JOIN commande co
  ON c.id = co.client_id
  GROUP BY c.id
  ORDER BY total_vente DESC;
quit;
/* 8- Le nombre moyen, pour chaque client, de chaque type de produit qu'ils ont acheté pour leurs commandes*/
Proc sql ;
  SELECT c.id, mean(co.pizza_qt) as moy_pizza, mean(co.sandwich_qt) as moy_sandwich, mean(co.escalope_qt) as moy_escalope
  FROM client c
  JOIN commande co
  ON c.id = co.client_id
  GROUP BY c.id;
Quit;

/* 9- Le montant moyen, pour chaque client, dépensé sur chaque type de produit*/
Proc sql ;
  SELECT c.id, mean(co.pizza_eu) as moy_pizza, mean(co.sandwich_eu) as moy_sandwich, mean(co.escalope_eu) as moy_escalope
  FROM client c
```