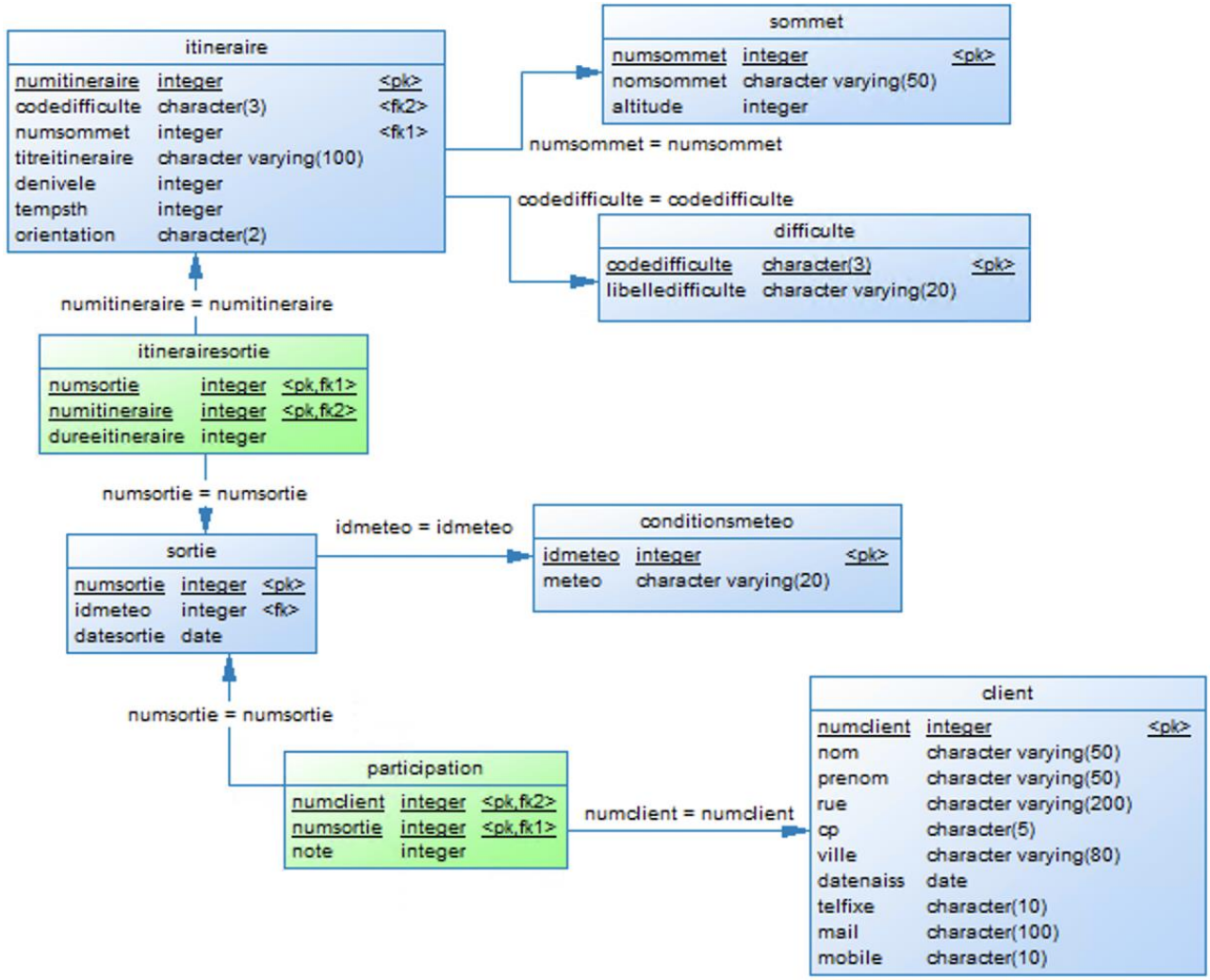
R2.06 Bases de données
TP5 – INSERTION DE DONNEES (PARTIE 2)

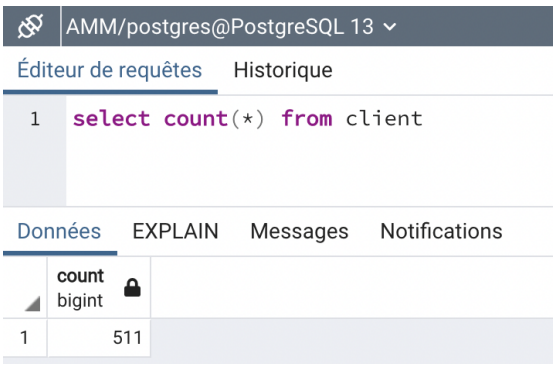
Ce TP fait suite au TP4. Il faut donc avoir terminé le TP4.

Rappel du modèle de données :



Vous devez avoir :

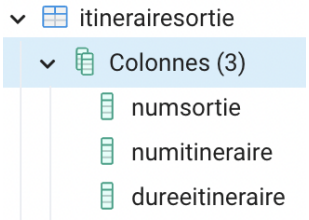
- 511 lignes dans la table `CLIENT` :



- 176 lignes dans la table `ITINERAIRE`
- 170 lignes dans la table `SORTIE`

Q1 : Génération des données dans la table `ITINERAIRESORTIE`.
Nous pouvons utiliser Excel ou réaliser cette insertion de données en utilisant SQL.
Nous allons le faire en SQL.

Rappel de la table `ITINERAIRESORTIE` :



La durée de l'itinéraire est la durée réelle : soit la durée totale de la sortie si la sortie n'est composée que d'un seul itinéraire (boucle), soit la durée de la montée et celle de la descente si la sortie est constituée de 2 itinéraires.

Etapes :

- Etape 1 : réaliser une requête affichant le n° de sortie, la date de sortie, l'idmeteo et générant un numéro aléatoire d'itinéraire compris entre 100 et 180 (les numéros d'itinéraire générés en TP4 vont de 100 à 199), **uniquement pour les sorties de 2023** (celles générées en TP4).
  Code du numéro d'itinéraire généré : `floor((random() * (180-100+1)) + 100)::int` en suivant l'exemple disponible ici : https://www.geeksforgeeks.org/postgresql-random-number-generation/
  Résultat :

	numsortie integer	datesortie date	idmeteo integer	numitineraire integer
1	200	2023-01-01	2	126
2	201	2023-01-02	1	130
3	202	2023-01-03	3	153
4	203	2023-01-04	1	130
5	204	2023-01-05	2	156
6	205	2023-01-06	3	122
7	206	2023-01-07	2	168
8	207	2023-01-08	3	151
9	208	2023-01-09	2	154
10	209	2023-01-10	3	116
11	210	2023-01-11	2	173
12	211	2023-01-12	2	147
13	212	2023-01-13	1	154
14	213	2023-01-14	3	108
15	214	2023-01-15	3	143
16	215	2023-01-16	2	147
17	216	2023-01-17	3	102
18	217	2023-01-18	1	170
19	218	2023-01-19	2	146
20	219	2023-01-20	1	161
Total de lignes: 59 sur 59		Requête terminée 00:00:00.112		

(59 lignes)

- Etape 2 : créer une table nommée `Q1a` à partir de la requête précédente en utilisant l'ordre SQL : `CREATE TABLE… AS SELECT…` (Cf. CM S1 ou https://www.w3schools.com/sql/sql_create_table.asp). Cette table contiendra 59 lignes, correspondant aux 59 sorties réalisées en 2023. Chaque sortie ne concernera cependant qu'un seul itinéraire.

- Etape 3 : exécuter le code suivant permettant d'ajouter un second itinéraire emprunté sur quelques sorties :

```
INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
SELECT numsortie, datesortie, idmeteo, 190
FROM Q1a
WHERE numsortie=201;

INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
```

```
SELECT numsortie, datesortie, idmeteo, 191
FROM Q1a
WHERE numsortie=211;

INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
SELECT numsortie, datesortie, idmeteo, 192
FROM Q1a
WHERE numsortie=221;

INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
SELECT numsortie, datesortie, idmeteo, 193
FROM Q1a
WHERE numsortie=231;

INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
SELECT numsortie, datesortie, idmeteo, 194
FROM Q1a
WHERE numsortie=241;

INSERT INTO Q1a (numsortie, datesortie, idmeteo, numitineraire)
SELECT numsortie, datesortie, idmeteo, 195
FROM Q1a
WHERE numsortie=251;
```

Vérifier les données de la table `Q1a`.

Exécuter ensuite la requête suivante pour bien vérifier vos insertions :

Requête Historique

```
SELECT numsortie, COUNT(DISTINCT numitineraire) AS nbitineraires
FROM Q1a
GROUP BY numsortie;
```

Data Output Messages Notifications

	numsortie integer	nbitineraires bigint
1	200	1
2	201	2
3	202	1
4	203	1
5	204	1
6	205	1
7	206	1
8	207	1
9	208	1
10	209	1
11	210	1
12	211	2
13	212	1
14	213	1
15	214	1
16	215	1
17	216	1
18	217	1
19	218	1
20	219	1

Total de lignes: 59 sur 59 Requête terminée 00:00:00.155

(59 lignes)

Chaque sortie porte bien sur 1 seul itinéraire (une boucle) ou 2 itinéraires (1 pour l'aller et 1 pour le retour).
Ici toutes les données ne sont pas forcément cohérentes car l'itinéraire d'aller peut être lié à un sommet et celui de retour à un autre. Il serait donc possible d'améliorer cette génération de données.

- Etape 4 : il est maintenant nécessaire de générer le temps de parcours réel de chaque itinéraire. Nous allons considérer que la durée réelle de l'itinéraire est fonction des conditions météo.

Idmeteo (conditions météo)	*Durée itinéraire*
1 (= *Beau temps*)	Temps théorique de l'itinéraire * nombre aléatoire variant entre 80% et 90%
2 (= *Nuageux*)	Temps théorique de l'itinéraire * nombre aléatoire variant entre 90% et 110%
3 *(= Pluie/Neige)*	Temps théorique de l'itinéraire * nombre aléatoire variant entre 110% et 130%

Requête à compléter (vous aurez besoin des tables `Q1a` et `ITINERAIRE`) :

```
SELECT …,
      CASE q.idmeteo
            WHEN 1 THEN round((i.tempsth*floor((random() * (90-80+1)) +
80)/100)::numeric, -1)
            WHEN …
            WHEN …
      END AS dureeitineraire
FROM …
      JOIN … ON …
ORDER BY …
```

Résultat :

	numsortie integer	datesortie date	idmeteo integer	numitineraire integer	tempsth integer	dureeitineraire numeric
1	200	2023-01-01	2	126	350	350
2	201	2023-01-02	1	190	360	300
3	201	2023-01-02	1	130	460	410
4	202	2023-01-03	3	153	190	250
5	203	2023-01-04	1	130	460	370
6	204	2023-01-05	2	156	410	400
7	205	2023-01-06	3	122	360	410
8	206	2023-01-07	2	168	340	330
9	207	2023-01-08	3	151	350	410
10	208	2023-01-09	2	154	460	500
11	209	2023-01-10	3	116	420	460
12	210	2023-01-11	2	173	340	370
13	211	2023-01-12	2	191	190	190
14	211	2023-01-12	2	147	340	320
15	212	2023-01-13	1	154	460	410
16	213	2023-01-14	3	108	450	540
17	214	2023-01-15	3	143	170	200
18	215	2023-01-16	2	147	340	360
19	216	2023-01-17	3	102	490	600
20	217	2023-01-18	1	170	440	370

Total de lignes: 65 sur 65 | Requête terminée 00:00:00.112

(65 lignes)

- Etape 5 : créer la table `Q1b` à partir de la requête précédente. Cette table contiendra les données suivantes (le random s'est ré-exécuté !) :

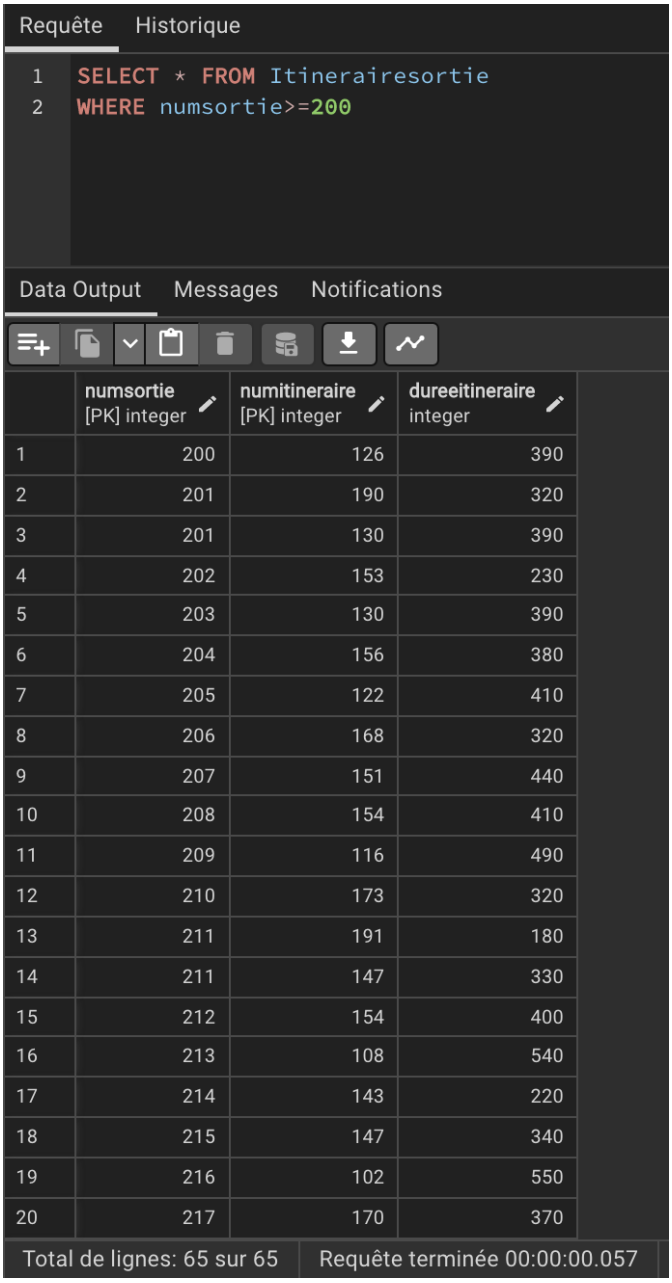
	numsortie integer	numitineraire integer	dureeitineraire numeric
1	200	126	390
2	201	190	320
3	201	130	390
4	202	153	230
5	203	130	390
6	204	156	380
7	205	122	410
8	206	168	320
9	207	151	440
10	208	154	410
11	209	116	490
12	210	173	320
13	211	191	180
14	211	147	330
15	212	154	400
16	213	108	540
17	214	143	220
18	215	147	340
19	216	102	550
20	217	170	370

Total de lignes: 65 sur 65 | Requête terminée 00:00:00.172

(65 lignes)

- Etape 6 : Vérifier les données de la table `Q1b`. Insérer les données de cette table dans la table `ITINERAIRESORTIE` en utilisant l'instruction :
  `INSERT INTO matable(champ1, champ2, ..)`
  `SELECT…`
  https://www.w3schools.com/sql/sql_insert_into_select.asp

  Résultat :



	numsortie [PK] integer	numitineraire [PK] integer	dureeitineraire integer
1	200	126	390
2	201	190	320
3	201	130	390
4	202	153	230
5	203	130	390
6	204	156	380
7	205	122	410
8	206	168	320
9	207	151	440
10	208	154	410
11	209	116	490
12	210	173	320
13	211	191	180
14	211	147	330
15	212	154	400
16	213	108	540
17	214	143	220
18	215	147	340
19	216	102	550
20	217	170	370

Total de lignes: 65 sur 65 | Requête terminée 00:00:00.057

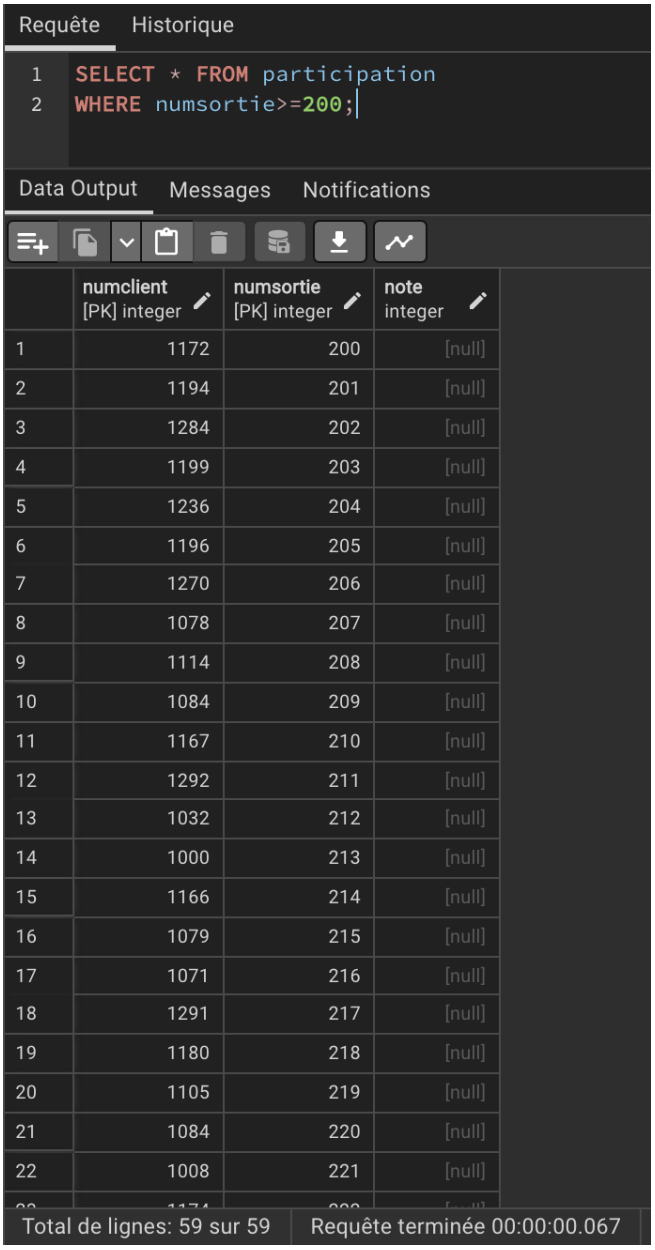
(65 lignes)

- Etape 7 : créer les instructions SQL supprimant les tables `Q1a` et `Q1b` qui ne sont plus utiles.

Remarque : plutôt que de créer des tables « normales », il est aussi possible de créer des tables temporaires qui ne sont manipulables que par des développeurs (https://docs.postgresql.fr/13/sql-createtableas.html). Nous y reviendrons dans le futur.

Nous n'avons pas mis à jour le script du TP4. Nous le ferons ultérieurement.

Q2 : En procédant de la même façon, ajouter des données dans la table `PARTICIPATION` : `numsortie` et `numclient`. Vous ajouterez des participations pour toutes les sorties de 2023 (59 sorties). Le numéro du client sera aléatoire et variera entre 1000 et 1300. Pour simplifier, il n'y aura qu'un seul client par sortie. Résultat : 59 lignes ajoutées.



Exécuter ensuite le script `generation_participations.sql` permettant d'ajouter des clients supplémentaires dans les sorties.
Ce script permet d'ajouter entre 1 et 10 clients (nombre aléatoire) supplémentaires par sortie pour les sorties de 2023 (n° de sorties variant de 200 à 258) et pour des n° de client choisis aléatoirement variant entre 1301 et 1499.
Si le code ne fonctionne pas, bien vérifier que vous avez des clients numérotés de 1301 à 1499 dans la table CLIENT.

Remarque : Vous ne serez capable d'écrire ce script (code procédural PL/pgSQL) que l'an prochain.

Résultat :

Requête Historique

```
1  SELECT * FROM participation
2  WHERE numsortie>=200
3  ORDER BY 1, 2;
```

Data Output Messages Notifications

	numclient [PK] integer	numsortie [PK] integer	note integer
1	1000	213	[null]
2	1008	221	[null]
3	1014	252	[null]
4	1024	246	[null]
5	1028	250	[null]
6	1032	212	[null]
7	1055	240	[null]
8	1057	236	[null]
9	1058	230	[null]
10	1063	258	[null]
11	1071	216	[null]
12	1071	245	[null]
13	1074	253	[null]
14	1078	207	[null]
15	1079	215	[null]
16	1084	209	[null]
17	1084	220	[null]
18	1088	256	[null]
19	1090	242	[null]
20	1097	223	[null]
21	1101	254	[null]
22	1105	219	[null]

Total de lignes: 383 sur 383 Requête terminée 00:00:00.054

383 lignes pour les sorties de 2023, mais ce nombre est aléatoire.

Q3 : Comme, nous pouvons le remarquer, aucune note n'a été renseignée. Les notes doivent être comprises entre 1 et 5.
Nous pouvons donc générer des nombres aléatoires compris entre 1 ou 5 ou, mieux, suivre une distribution des notes basée sur la loi normale, que vous verrez dans le module « R2.08 : Outils numériques pour les statistiques descriptives ».
Distribution / loi normale : https://www.irdp.ch/institut/normale-distribution-2136.html, https://fr.wikipedia.org/wiki/Loi_normale

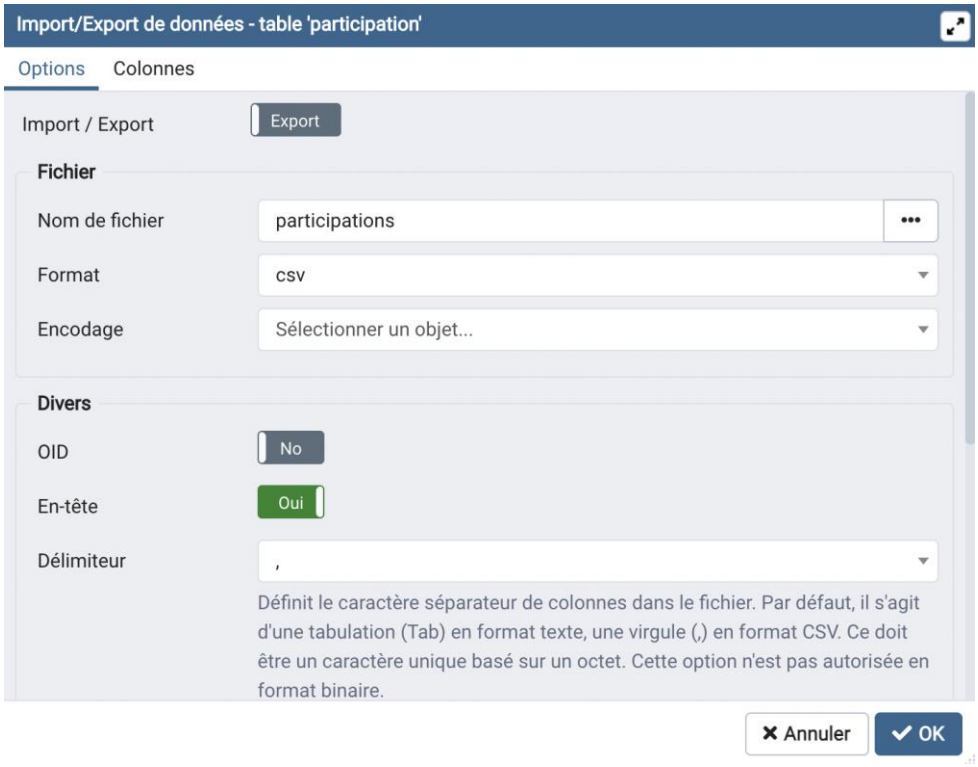
Il existe plusieurs façons de générer des nombres aléatoires suivant une distribution normale :

- Avec Generatedata.com, par exemple :

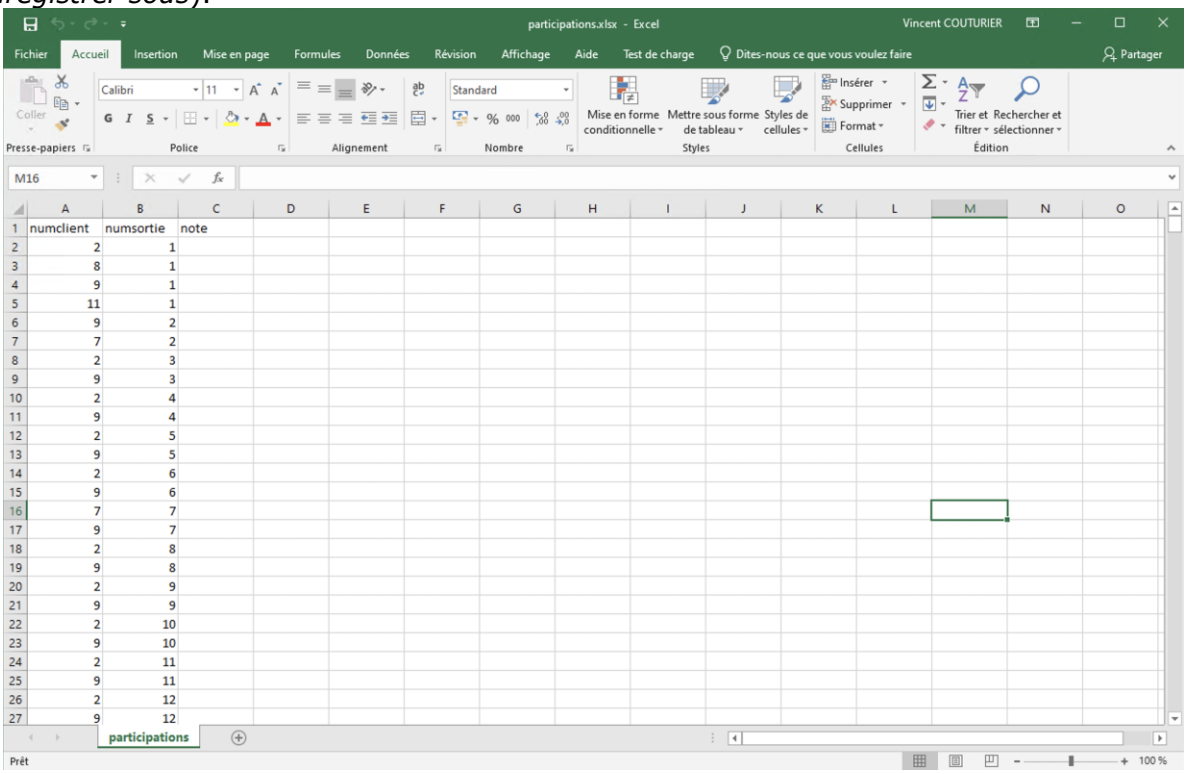
- Avec Microsoft Excel.

Nous allons utiliser Excel car comme des données sont déjà présentes dans la table `PARTICIPATION`, cela simplifiera notre tâche.

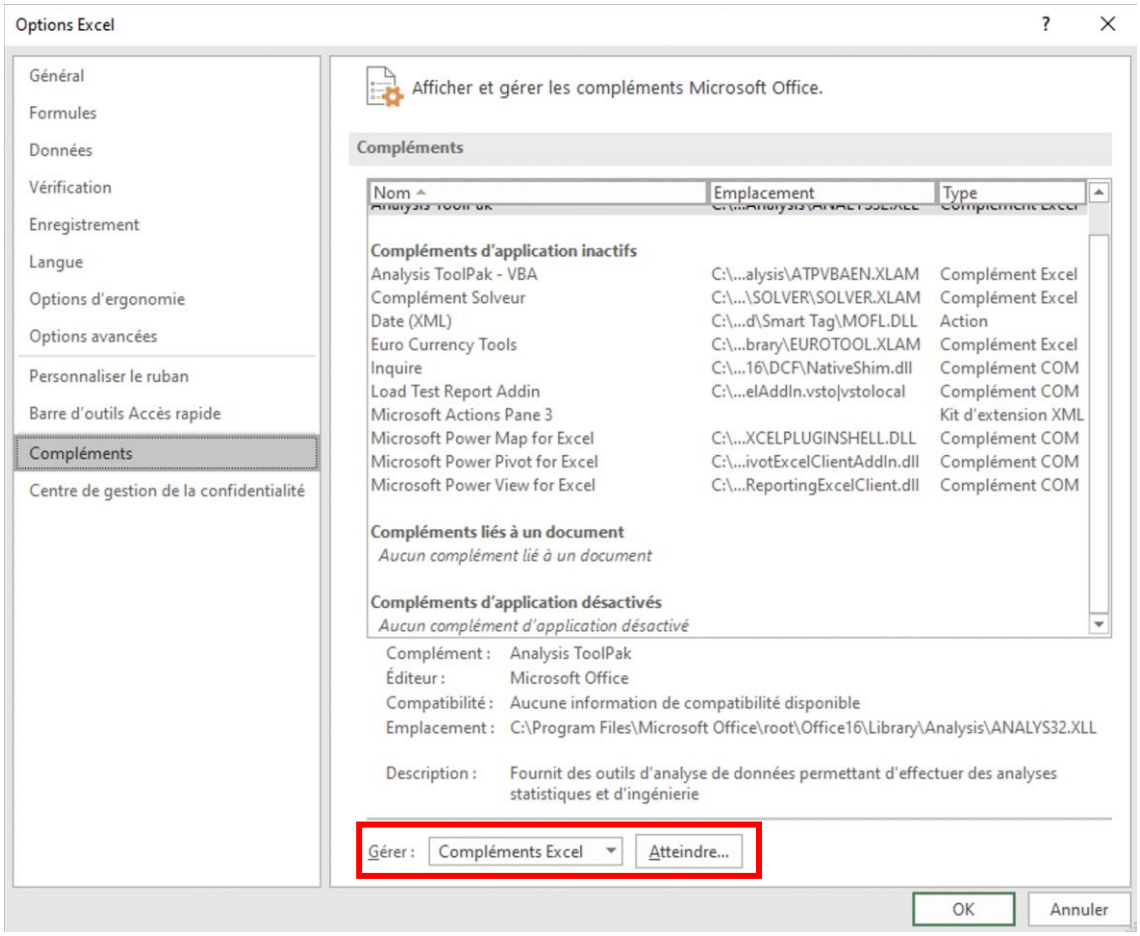
1. Dans PgAdmin4, exporter les données de la table `PARTICIPATION` au format CSV.



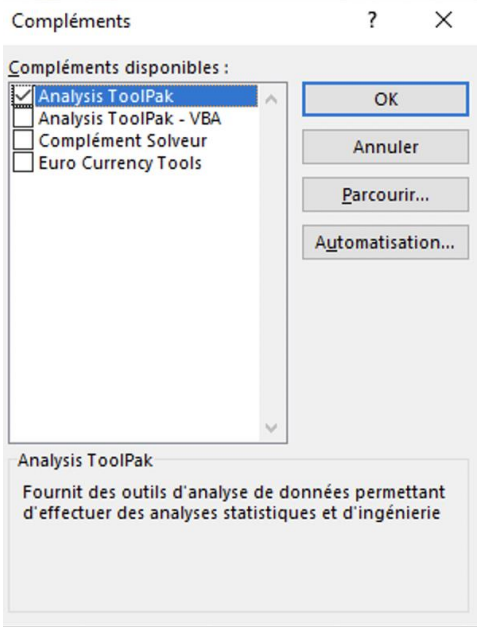
2. Ouvrir le fichier avec Excel puis l'enregistrer au format *Classeur Excel (*.xlsx)* (Menu *Fichier* > *Enregistrer sous*).

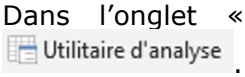


3. Génération de la distribution normale en utilisant le complément Excel « Analysis Toolpack » :
   a. Dans Menu *Fichier* > *Options*, onglet « Compléments », cliquer sur le bouton « Atteindre » de « Gérer : Compléments Excel ».



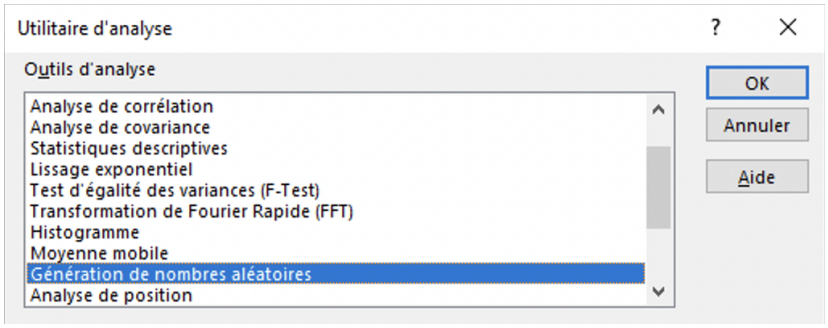
b. Activer le complément « Analysis Toolpak » s'il ne l'est pas déjà. Valider.



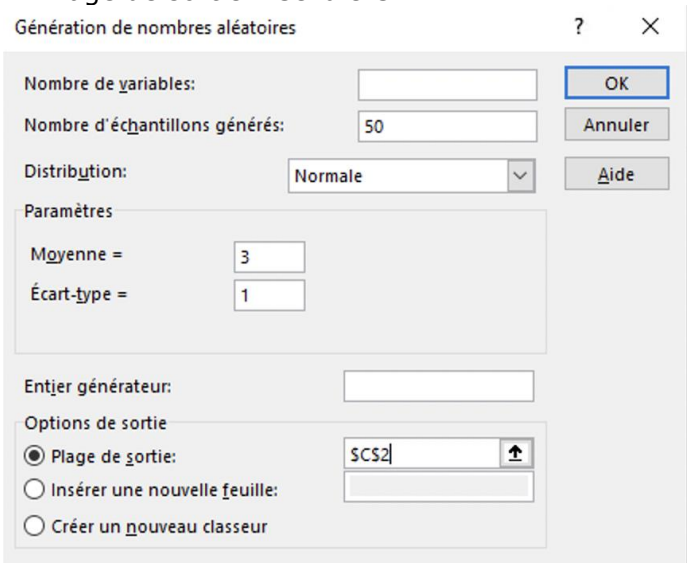
c. Dans l'onglet « Données » d'Excel, cliquer sur le bouton « Utilitaire d'analyse » 

.

d. Sélectionner « Génération de nombres aléatoires ».



e. Générer, par exemple, 50 notes, distribution normale, moyenne (= espérance) de 3 (sur 5), écart-type de 1. Plage de sortie : Cellule C2.



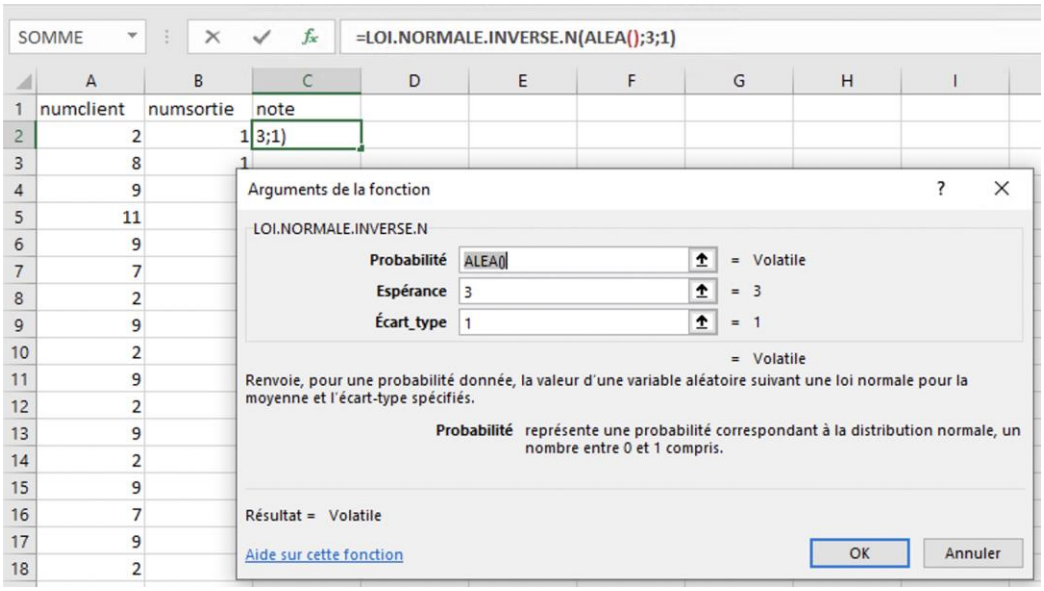
Résultat :

	A	B	C
1	numclient	numsortie	note
2	2	1	2,82076474
3	8	1	3,92237315
4	9	1	3,28569843
5	11	1	0,93196366
6	9	2	4,28831061
7	7	2	1,87954573
8	2	3	3,72240368
9	9	3	3,1001888
10	2	4	3,65320137
11	9	4	2,69624469
12	2	5	3,61762762
13	9	5	2,67478402
14	2	6	2,79020345
15	9	6	3,63465905
16	7	7	4,04155788
17	9	7	2,9491456

Le problème est que l'on ne peut pas faire varier les valeurs générées en fonction de paramètres. En effet, nous allons considérer que la note s'est améliorée à partir de 2023 et donc la sortie n°200.

Effacer les données générées.

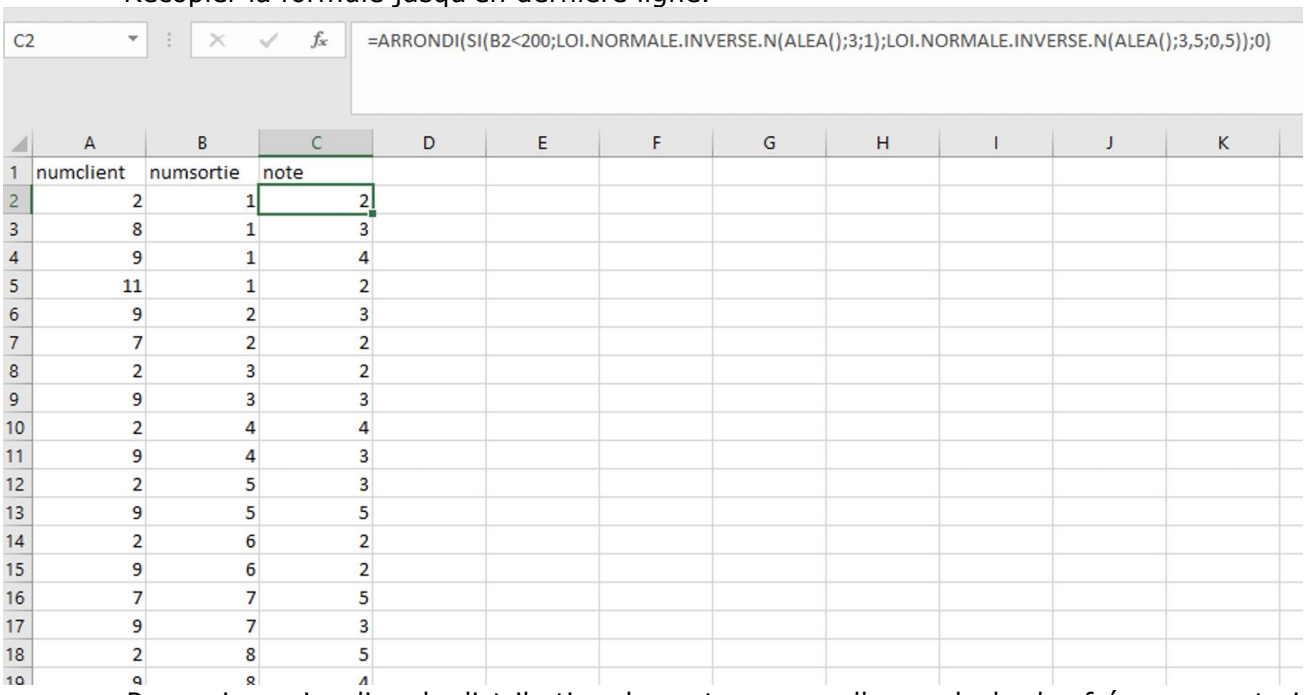
4. Génération de la distribution normale en utilisant les fonctions `LOI.NORMALE.INVERSE.N` et `ALEA` :
   a. Saisir la formule suivante en cellule C2 :



Fonctions Excel utilisées :
https://support.microsoft.com/fr-fr/office/loi-normale-inverse-n-loi-normale-inverse-n-fonction-54b30935-fee7-493c-bedb-2278a9db7e13
https://support.microsoft.com/fr-fr/office/alea-alea-fonction-4cbfa695-8869-4788-8d90-021ea9f5be73

b. Modifier la formule en utilisant la fonction `SI` :
   - Si *numsortie<200*, la distribution utilisera les paramètres précédents : Espérance : 3, Ecart-type : 1.
   - Sinon, la distribution aura pour paramètres : Espérance : 3,5, Ecart-type : 0,5.

   Arrondir la valeur à l'entier le plus proche car le type du champ note est `INTEGER`.
   Attention, il est possible de générer une note > 5.
   Recopier la formule jusqu'en dernière ligne.

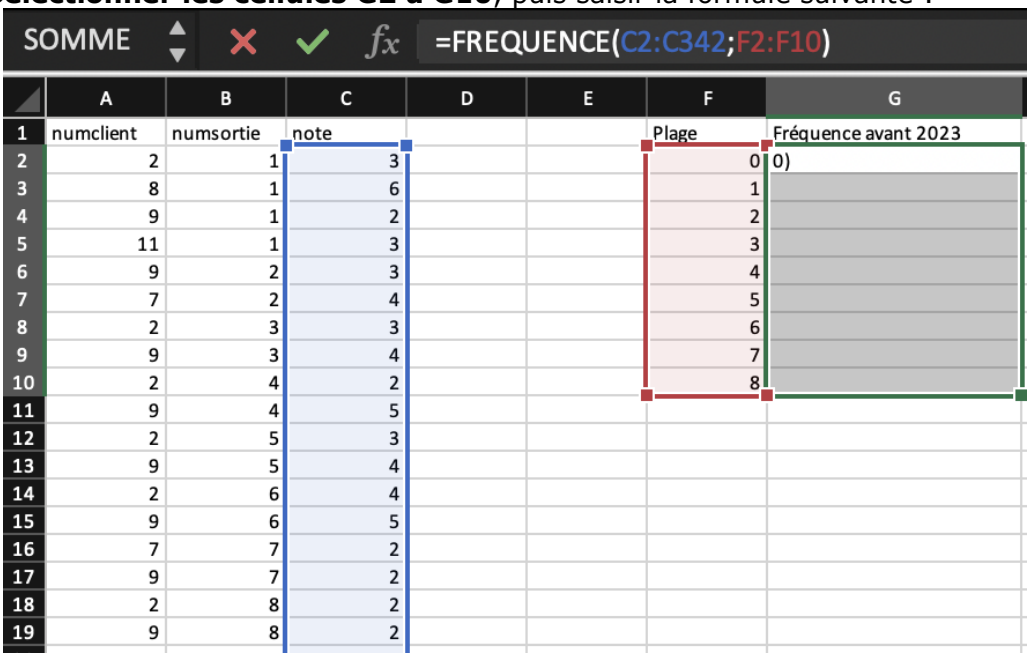


	numclient	numsortie	note
2	2	1	2
3	8	1	3
4	9	1	4
5	11	1	2
6	9	2	3
7	7	2	2
8	2	3	2
9	9	3	3
10	2	4	4
11	9	4	3
12	2	5	3
13	9	5	5
14	2	6	2
15	9	6	2
16	7	7	5
17	9	7	3
18	2	8	5

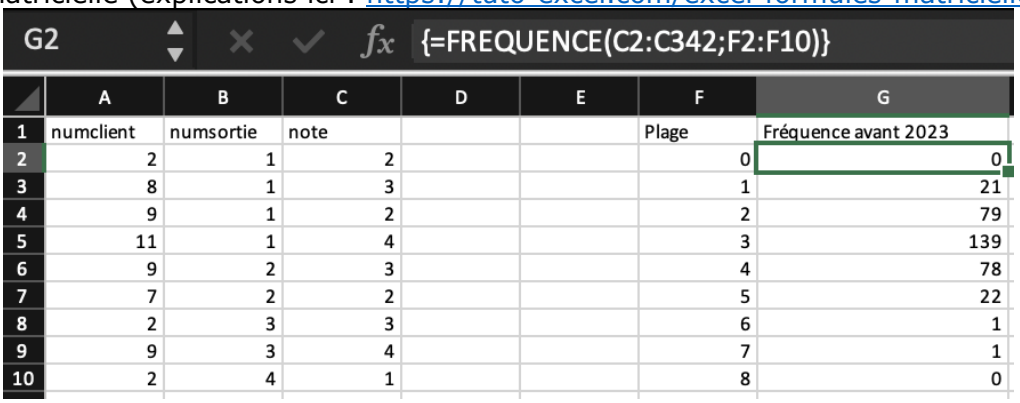
c. Pour mieux visualiser la distribution des notes, nous allons calculer les fréquences et ajouter un histogramme.
   Pour cela saisir d'abord les plages possibles en colonne F :

	A	B	C	D	E	F	G
1	numclient	numsortie	note			Plage	
2	2	1	3			0	
3	8	1	3			1	
4	9	1	5			2	
5	11	1	2			3	
6	9	2	3			4	
7	7	2	3			5	
8	2	3	2			6	
9	9	3	3			7	
10	2	4	3			8	
11	9	4	4				
12	2	5	3				
13	9	5	2				
14	2	6	2				
15	9	6	3				

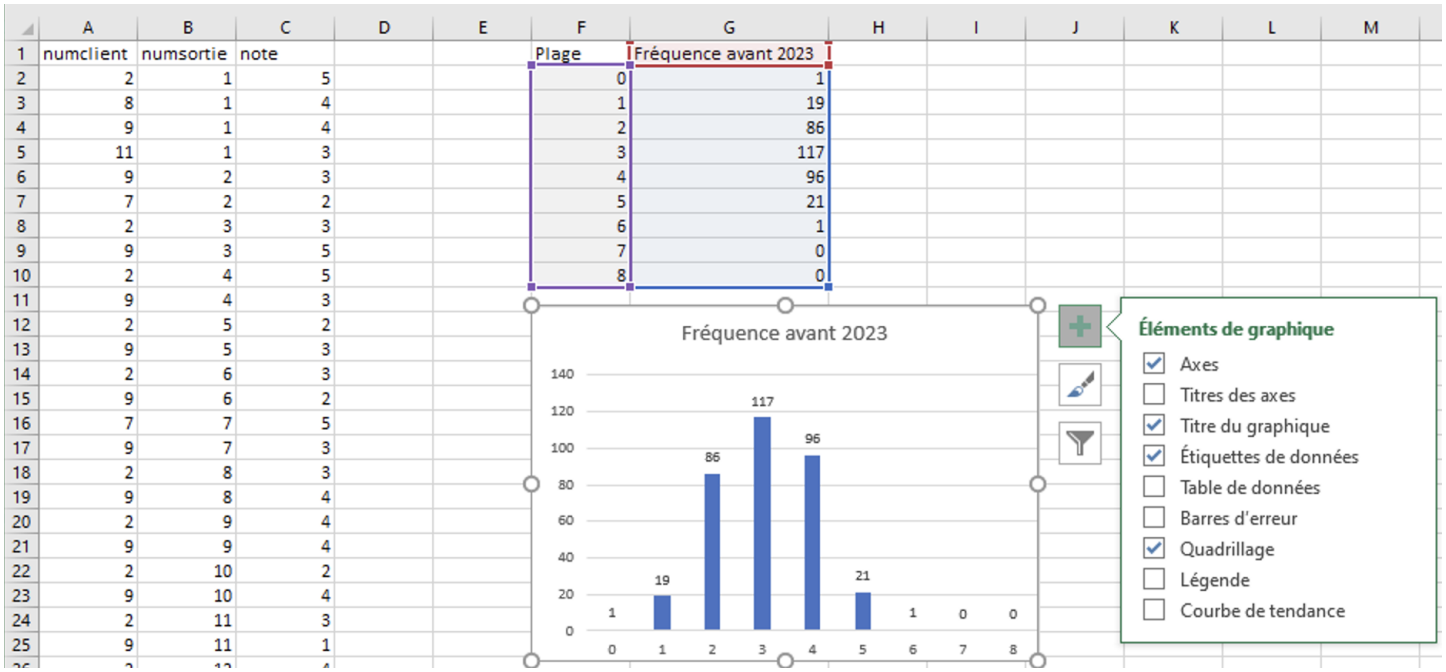
Saisir le texte « Fréquence avant 2023 » en cellule G1.
Sélectionner les cellules G2 à G10, puis saisir la formule suivante :



Valider la formule ensuite avec `CTRL + MAJ + ENTREE`, afin de l'appliquer sur l'ensemble des 7 cellules. Vous remarquerez les `{}` autour de la formule, qui indique qu'il s'agit d'une formule matricielle (explications ici : https://tuto-excel.com/excel-formules-matricielles.html).

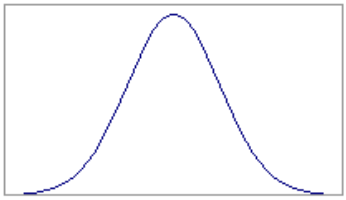


Sélectionner la plage F1 à G10, puis onglet *Insertion* > *Graphiques recommandés* et sélectionner l'histogramme. Nous pouvons visualiser la fréquence des notes avant 2023 (car nous n'avons appliqué la fréquence que sur la plage C2:C342 (et donc avant la sortie n°200).



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	numclient	numsortie	note			Plage	Fréquence avant 2023						
2	2	1	5			0	1						
3	8	1	4			1	19						
4	9	1	4			2	86						
5	11	1	3			3	117						
6	9	2	3			4	96						
7	7	2	2			5	21						
8	2	3	3			6	1						
9	9	3	5			7	0						
10	2	4	5			8	0						
11	9	4	3										
12	2	5	2										
13	9	5	3										
14	2	6	3										
15	9	6	2										
16	7	7	5										
17	9	7	3										
18	2	8	3										
19	9	8	4										
20	2	9	4										
21	9	9	4										
22	2	10	2										
23	9	10	4										
24	2	11	3										
25	9	11	1										

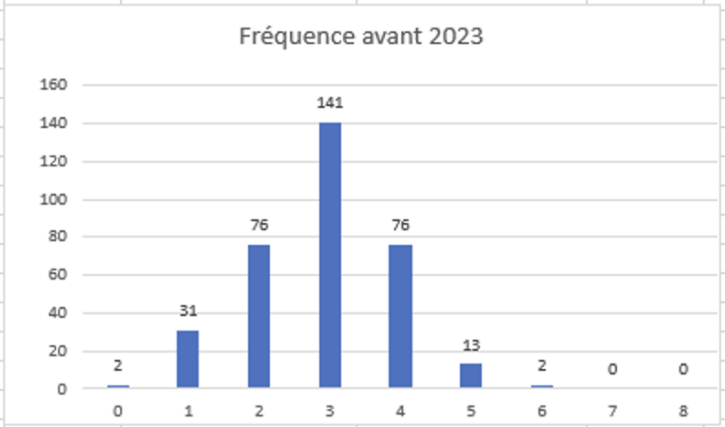
Nous avons bien obtenu une distribution normale :

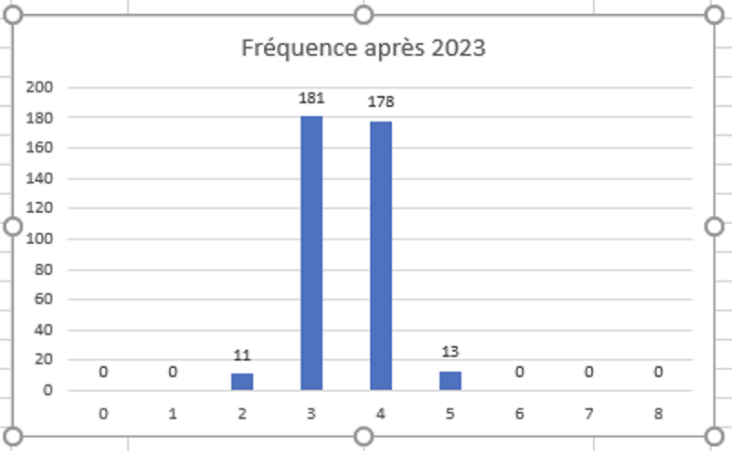


MAIS, nous avons généré des notes > 5 et/ou <1 (or une contrainte CHECK sur le champ `note` impose qu'elle soit comprise entre 1 et 5). **Nous verrons comment procéder ultérieurement.**

Faire de même pour les notes à partir de la sortie n°200 (la plage commencera en cellule C343 et ira jusqu'en dernière ligne de la colonne C). Résultat :

	A	B	C	D	E	F	G	H	I	J
1	numclient	numsortie	note			Plage	Fréquence avant 2023	Fréquence après 2023		
2	2	1	3			0	2	0		
3	8	1	3			1	31	0		
4	9	1	4			2	76	11		
5	11	1	2			3	141	181		
6	9	2	2			4	76	178		
7	7	2	3			5	13	13		
8	2	3	5			6	2	0		
9	9	3	3			7	0	0		
10	2	4	1			8	0	0		
11	9	4	4							
12	2	5	3							
13	9	5	4							
14	2	6	3							
15	9	6	2							
16	7	7	3							
17	9	7	4							
18	2	8	3							
19	9	8	4							
20	2	9	3							
21	9	9	3							
22	2	10	4							
23	9	10	3							
24	2	11	3							
25	9	11	3							
26	2	12	3							
27	9	12	3							
28	9	13	5							
29	9	14	3							
30	4	14	2							
31	3	14	3							
32	2	14	4							
33	6	14	-1							
34	8	15	4							
35	9	15	3							
36	2	16	1							
37	6	16	4							
38	9	17	2							
39	6	17	2							
40	2	18	3							
41	3	18	2							

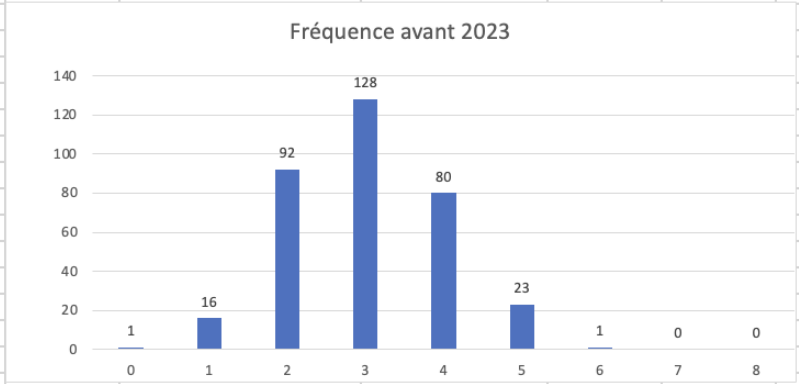


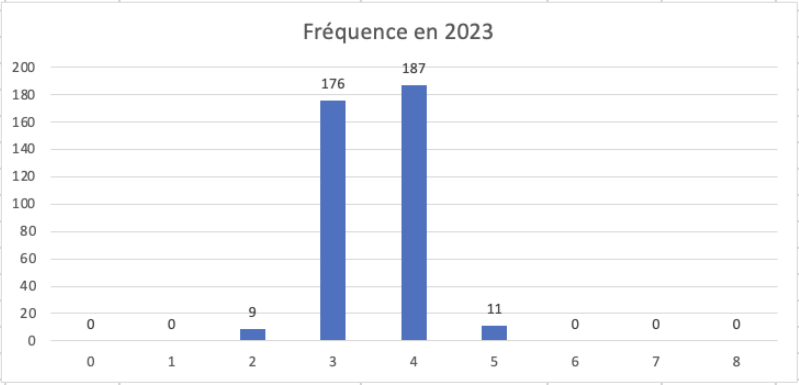


d. Gestion des notes > 5 ou <1 :
La formule suivante ne fonctionnera pas :
`=SI(ARRONDI(SI(B2<200;LOI.NORMALE.INVERSE.N(ALEA();3;1);LOI.NORMALE.INVERSE.N(ALEA();3,5;0,5));0)>5;5;ARRONDI(SI(B2<200;LOI.NORMALE.INVERSE.N(ALEA();3;1);LOI.NORMALE.INVERSE.N(ALEA();3,5;0,5));0))`
car 2 séries de nombres aléatoires sont générés dans le `SI`.

Le plus simple est d'ajouter une formule (en colonne D) utilisant 2 fonctions `SI` (ou une fonction `SI.CONDITIONS`) permettant de gérer ces cas. Pour les notes <1, les remplacer par la valeur 1 ; pour les notes >5, les remplacer par la valeur 5.

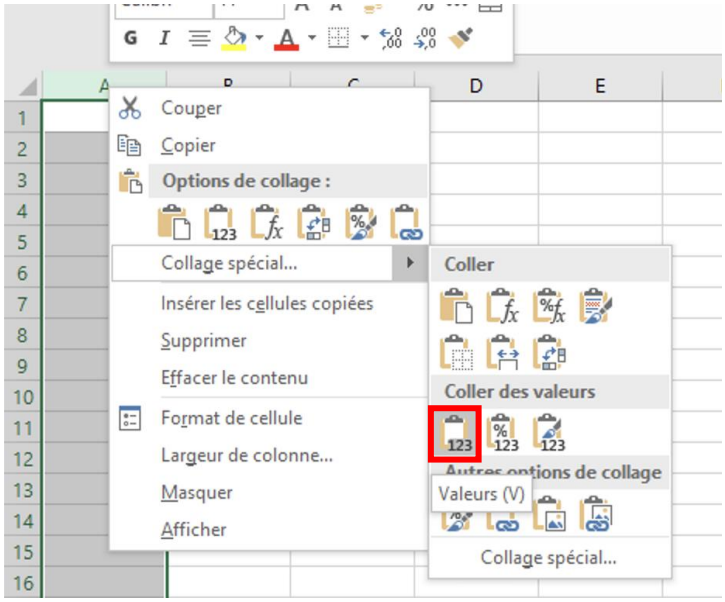
	A	B	C	D	E	F	G	H
1	numclient	numsortie	noteANC	note		Plage	Fréquence avant 2023	Fréquence en 2023
2	2	1	3	3		0	1	0
3	8	1	3	3		1	16	0
4	9	1	1	1		2	92	9
5	11	1	4	4		3	128	176
6	9	2	2	2		4	80	187
7	7	2	3	3		5	23	11
8	2	3	3	3		6	1	0
9	9	3	3	3		7	0	0
10	2	4	2	2		8	0	0
11	9	4	4	4				
12	2	5	4	4				
13	9	5	4	4				
14	2	6	4	4				
15	9	6	3	3				
16	7	7	5	5				
17	9	7	3	3				
18	2	8	4	4				
19	9	8	5	5				
20	2	9	1	1				
21	9	9	4	4				
22	2	10	3	3				
23	9	10	4	4				
24	2	11	2	2				
25	9	11	4	4				
26	2	12	3	3				
27	9	12	4	4				
28	9	13	4	4				
29	9	14	4	4				
30	4	14	2	2				
31	3	14	3	3				
32	2	14	4	4				
33	6	14	5	5				
34	8	15	3	3				
35	9	15	3	3				
36	2	16	2	2				
37	6	16	4	4				
38	9	17	2	2				
39	6	17	3	3				
40	2	18	5	5				
41	3	18	4	4				
42	2	19	2	2				





Enregistrer le classeur Excel.

Créer une nouvelle feuille de calcul.
Copier les colonnes A, B et D de la feuille 1.
Coller ces colonnes en feuille 2 en utilisant le *collage spécial* > *coller des valeurs* (bouton droit de la souris).



Le collage spécial des valeurs permet de ne conserver que les valeurs sans les formules, ce qui permettra qu'elles ne soient plus modifiées.

Enregistrer le classeur Excel.

Enregistrer ensuite la feuille active (feuille n°2) au format CSV (menu *Fichier* > *Enregistrer sous*).

5. Dans PgAdmin4, supprimer les données de la table `PARTICIPATION`. Importer les données du fichier CSV.
   Résultat :

Éditeur de requêtes | Historique

```
1  SELECT * FROM public.participation
2  ORDER BY numclient ASC, numsortie ASC
```

Données | EXPLAIN | Messages | Notifications

	numclient [PK] integer	numsortie [PK] integer	note integer
1	1	24	4
2	1	28	3
3	1	34	3
4	1	35	4
5	1	42	4
6	1	43	4
7	1	44	1
8	1	45	4
9	1	46	3
10	1	48	2
11	1	50	2
12	1	52	3
13	1	57	3
14	1	59	5
15	1	61	4
16	1	62	3
17	1	63	3
18	1	64	4

Q4 : Générer le script SQL de la base (comme en TP4) et compléter le script du fichier `Insertion BD AMM.sql`. Supprimer les lignes d'insert de `ITINERAIRESORTIE` et les remplacer par celles de la sauvegarde. Idem pour `PARTICIPATION`.

Q5 : Tester les différentes options de sauvegarde (notamment les formats *Personnalisé* et *Compressé TAR*) et de restauration de PostgreSQL. Documentation :
https://www.pgadmin.org/docs/pgadmin4/6.21/backup_dialog.html
Pour restaurer une sauvegarde de base de données, il est nécessaire de créer une nouvelle base (vide).