

Outil d'Analyse de Risque

Stress Tests (scénarios de marché)

Projet VBA — Excel

Auteurs

RUIZ Julien
FERREIRA DA SILVA Marwan
BEURTHE Dorian



Cours : CY Tech — VBA
Enseignant : VERIN Renaud
Date : 12 octobre 2025

Ce rapport présente la conception et la mise en œuvre d'un outil d'analyse de risque sous Excel/VBA, reproduisant à échelle pédagogique des mécanismes de salles de marché : stress tests.

Table des matières

Introduction	1
1 Structure générale du projet	1
Feuilles principales	1
2 Interfaces utilisateurs (Forms)	2
2.1 frmAddPortfolio	2
2.2 frmRemovePortfolio	2
2.3 frmAddScenario	2
2.4 frmRemoveScenario	2
2.5 frmRunScenario	2
3 Modules VBA	2
3.1 mod_Main	3
3.2 mod_IO	3
3.3 mod_Analytics	3
3.4 mod_ScenariosRunner	3
3.5 mod_Visualization	3
3.6 mod_ExportPDF	3
4 Calculs financiers intégrés	3
5 Difficultés rencontrées	4
6 Résultats obtenus et bilan	4
A Exemple de structure de scénario	4
B Format attendu de Portfolio	5
C Captures d'écran	5
D Extraits de code VBA	17
D.1 Module principal (mod_Main)	17
D.2 Module d'analyse (mod_Analytics)	17
D.3 Module d'entrée/sortie (mod_IO)	17
D.4 Module d'exécution des scénarios (mod_ScenariosRunner)	17
D.5 Module de visualisation (mod_Visualization)	17
D.6 Module d'exportation PDF (mod_ExportPDF)	17
D.7 Formulaires utilisateurs (UserForms)	17

Introduction

Dès le départ, nous ne voulions pas réaliser un projet classique comme on peut en trouver facilement sur Internet. Nous étions très contents que le sujet soit libre car nous avons pu travailler sur un outil qui nous intéresse vraiment tout en renforçant nos connaissances en VBA. Notre objectif était de créer quelque chose de plus original, concret et utile dans les salles de marché. Notre but était également de créer un projet VBA concret que l'on puisse mettre sur notre CV et en parler en entretien.

Nous avons donc choisi de développer un outil d'analyse de risque inspiré du monde de la finance, un domaine où la rigueur et la précision sont essentielles. Ce projet s'inscrit dans une logique appliquée, puisqu'il reproduit des mécanismes réellement utilisés par les traders et les analystes financiers au quotidien, tels que les stress tests. L'idée est de concevoir un programme à la fois pédagogique et réaliste, capable de transformer Excel en un véritable outil d'aide à la décision pour la gestion de portefeuille.

Ce projet a consisté à concevoir sous Excel/VBA un outil complet d'analyse de risque permettant de simuler des scénarios de marché et d'évaluer l'exposition d'un portefeuille d'actifs. L'objectif principal était de reproduire, à une échelle pédagogique, le fonctionnement d'un modèle interne de gestion de risque utilisé dans les institutions financières. L'application devait offrir une interface intuitive pour l'utilisateur, tout en permettant la création et l'exécution de scénarios de *stress tests*.

1. Structure générale du projet

Le classeur Excel s'articule autour de plusieurs feuilles de calcul et modules VBA, chacun jouant un rôle bien défini dans le processus global.

Feuilles principales

- **Portfolio** : base du portefeuille d'actifs de l'utilisateur (nom des titres, type, exposition en euros, coefficient bêta). Point d'entrée pour les analyses.
- **Market_Data** et **Market_Returns** : données de marché de référence (historiques de prix, taux, bêtas, rendements) utilisées pour la simulation.
- **Scenarios** : enregistre les scénarios de stress créés par l'utilisateur (nom, type d'actif ciblé, pourcentage de choc, date de création).
- **VaR_ES_Output** : résultats des stress tests (pertes/gains simulés en valeur et en pourcentage) selon le scénario choisi.

- **DEV_LOG** : feuille annexe pour le suivi technique.

2. Interfaces utilisateurs (Forms)

Un ensemble de formulaires interactifs permet d'éviter la manipulation directe des feuilles Excel.

2.1 frmAddPortfolio

Ajoute un nouvel actif au portefeuille. L'utilisateur choisit un ticker depuis la base de données ; le programme récupère automatiquement le type d'actif et le bêta correspondant dans **Market_Data**. Des contrôles de validité empêchent l'ajout de doublons ou de valeurs incohérentes.

2.2 frmRemovePortfolio

Permet de supprimer un actif existant ou de réduire partiellement son exposition. Assure une gestion dynamique du portefeuille sans perturber les autres macros.

2.3 frmAddScenario

Crée des scénarios de stress personnalisés. Chaque scénario peut comporter plusieurs chocs, par exemple « Equity -12 % », « Rates +3 % », « Oil -45 % ». Une règle de validation empêche la saisie de chocs inférieurs à -100 %. Le scénario est enregistré dans **Scenarios** avec horodatage.

2.4 frmRemoveScenario

Supprime un scénario complet et toutes les lignes associées, afin de garder la base de données propre.

2.5 frmRunScenario

Sélectionne un scénario et exécute la simulation. Les impacts sont calculés pour chaque actif et affichés dans **VaR_ES_Output**, avec un code couleur : gains en vert, pertes en rouge. Les formulaires sont stylés dans un thème sombre inspiré de Bloomberg Terminal (fond anthracite, texte clair, boutons orange vif).

3. Modules VBA

Le code est organisé en modules indépendants pour garantir clarté et modularité.

3.1 mod_Main

Coeur du programme. Contient les principales procédures, dont RunStressTest, qui applique les chocs de scénario à chaque actif du portefeuille. Cette fonction invoque d'autres modules pour le tri des rendements, les calculs statistiques et la mise en forme des résultats.

3.2 mod_IO

Gère les entrées/sorties de données : extraction de ticker à partir du nom complet (p. ex. « MICROSOFT CORPORATION (XNAS :MSFT) »), lecture dans les feuilles et conversions numériques sécurisées.

3.3 mod_Analytics

Fonctions statistiques de base : calculs de rendements journaliers et utilitaires de tri. La VaR et l'Expected Shortfall (ES) ne sont pas implémentées dans cette version. Inclut des algorithmes de tri optimisés (QuickSort, BubbleSort) pour classer les rendements.

3.4 mod_ScenariosRunner

Relie frmRunScenario au moteur de calcul, charge la liste des scénarios et exécute celui choisi.

3.5 mod_Visualization

Présente les résultats : formats monétaires et pourcentages, ajustement de colonnes, coloration (rouge pour pertes, vert pour gains). Améliore la lisibilité du tableau des résultats.

3.6 mod_ExportPDF

Exporte en un clic les feuilles Scenarios, Portfolio, VaR_ES_Output et Market_Data vers un PDF unique (paysage, marges fines, centrage automatique), puis ouvre le document.

4. Calculs financiers intégrés

Le *stress test* applique des chocs définis par l'utilisateur aux différentes classes d'actifs. L'impact sur la valeur du portefeuille est calculé pour chaque actif, permettant d'estimer les pertes potentielles dans un scénario extrême (crash, remontée des taux, chute du pétrole, etc.).

Définition

Stress test. Procédure de simulation consistant à appliquer des *scénarios extrêmes* (historiques ou hypothétiques) sur des facteurs de marché (actions, taux, changes, matières premières, volatilité, corrélations) afin d'évaluer la sensibilité d'un portefeuille. L'objectif est d'identifier les vulnérabilités et d'anticiper les pertes potentielles en cas d'événements rares mais plausibles. Les sorties typiques sont la variation de valeur du portefeuille (P&L), les contributions par actif et les éventuels dépassements de limites de risque.

5. Difficultés rencontrées

- **Compatibilité Office 2019** : certaines fonctions de récupération de données en ligne non supportées (erreurs #NAME? dans `Market_Returns`). Tests sur Office 365 confirmant une incompatibilité logicielle et non un défaut de code.
- **Correspondance colonnes/scénarios** : évolution du format d'origine ⇒ erreurs de lecture ; ajustement du code pour lire les bonnes colonnes (A : nom, B : type d'actif, C : choc, D : date).
- **Validations renforcées** : blocage des chocs < -100 % et ignorance des cellules vides lors des calculs.
- **Organisation modulaire** : l'ajout progressif de fonctionnalités a nécessité une structuration stricte (calcul, affichage, données, interface, export) afin d'éviter les interférences entre procédures.

6. Résultats obtenus et bilan

L'outil final est complet, intuitif et professionnel : création/gestion de portefeuille, ajout/suppression d'actifs, scénarios de stress personnalisés, calcul des impacts en un clic. Les résultats sont affichés clairement avec code couleur. La focalisation sur les stress tests offre une lecture directe des impacts extrêmes sans modéliser VaR/ES. L'export PDF génère instantanément un rapport consolidé.

Le projet combine programmation VBA avancée, concepts de finance de marché et ergonomie d'interface. Il illustre comment Excel peut devenir un véritable outil d'analyse de risque et de simulation, proche des environnements professionnels.

A. Exemple de structure de scénario

Nom	Type d'actif	Choc (%)	Date
Crash actions	Equity	-12	2025-03-01
Hausse taux	Rates	+3	2025-03-01
Chute pétrole	Commodities	-45	2025-03-01

Table 1 – Exemple de lignes dans la feuille Scenarios.

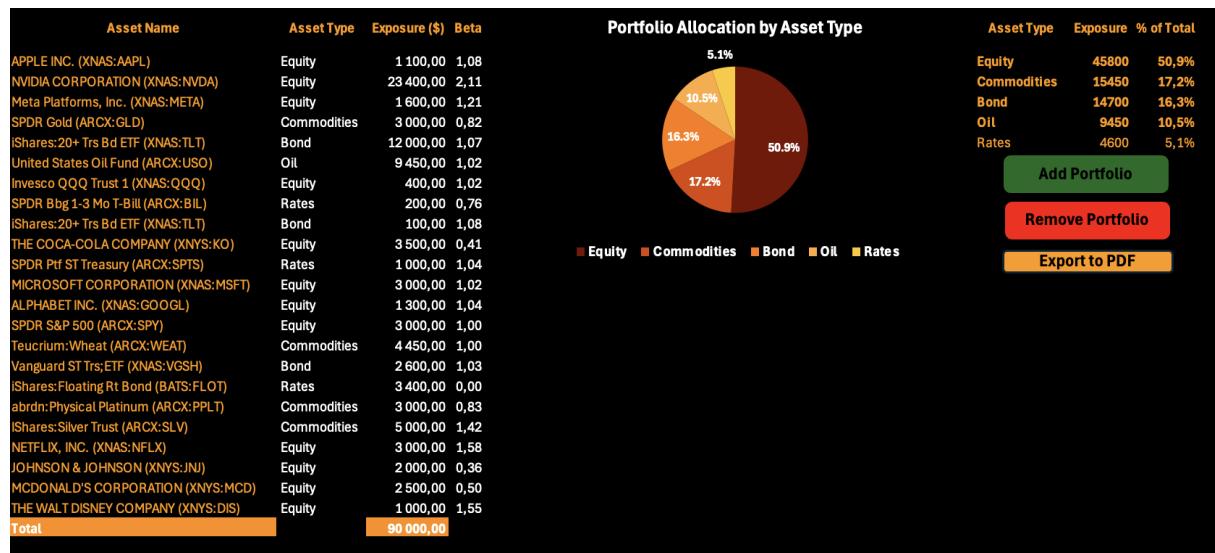
B. Format attendu de Portfolio

Ticker	Type	Exposition ()	Bêta
XNAS :MSFT	Equity	50 000	1,12
XPAR :BNP	Equity	30 000	1,05
CL1	Commodities	20 000	0,85

Table 2 – Champs minimaux pour la feuille Portfolio.

C. Captures d'écran

Cette section présente les principales interfaces et portions de code de notre projet VBA. Les images illustrent la structure du programme, les formulaires interactifs et le rendu des calculs dans Excel.

**Figure 1** – Interface sombre inspirée du style Bloomberg.

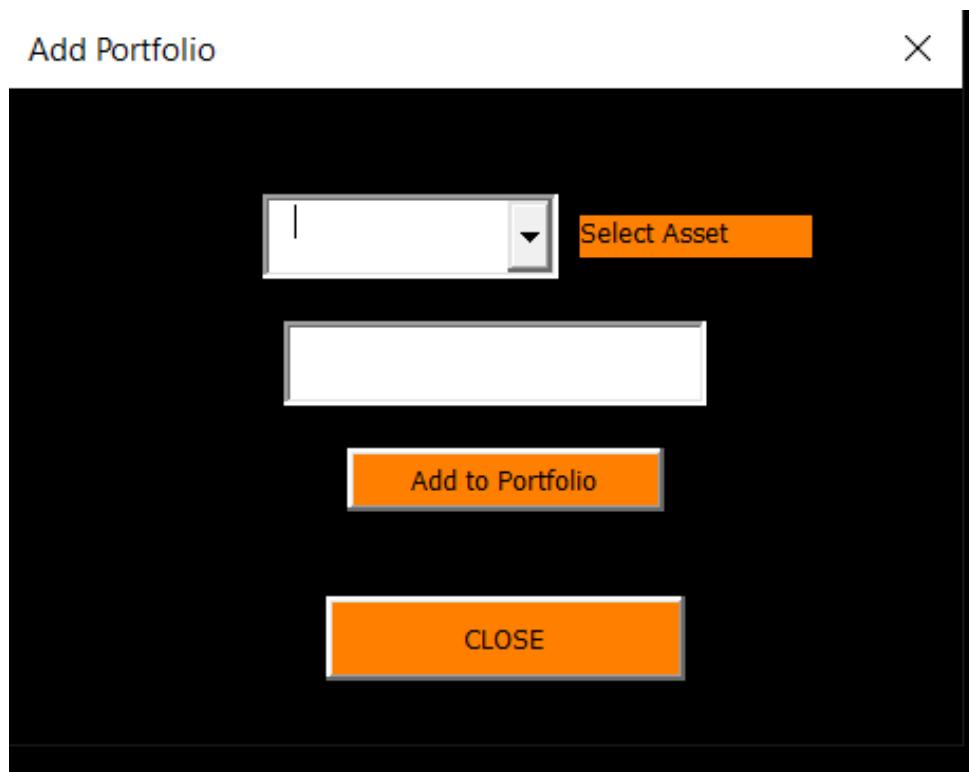


Figure 2 – Formulaire pour ajouter des actifs dans son portefeuille

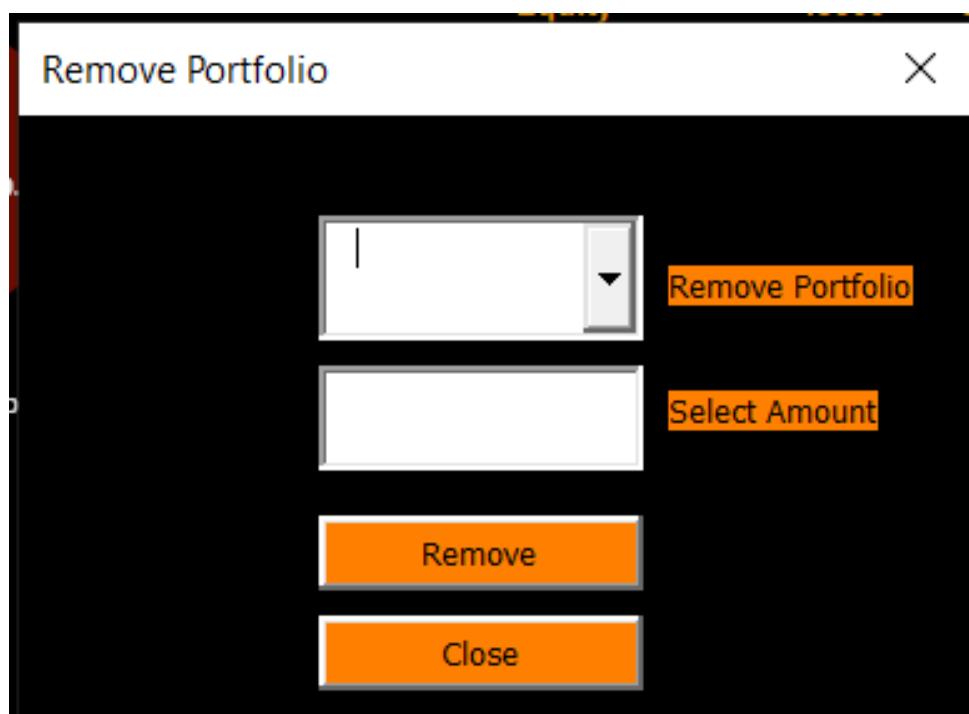


Figure 3 – Formulaire pour supprimer des actifs de son portefeuille

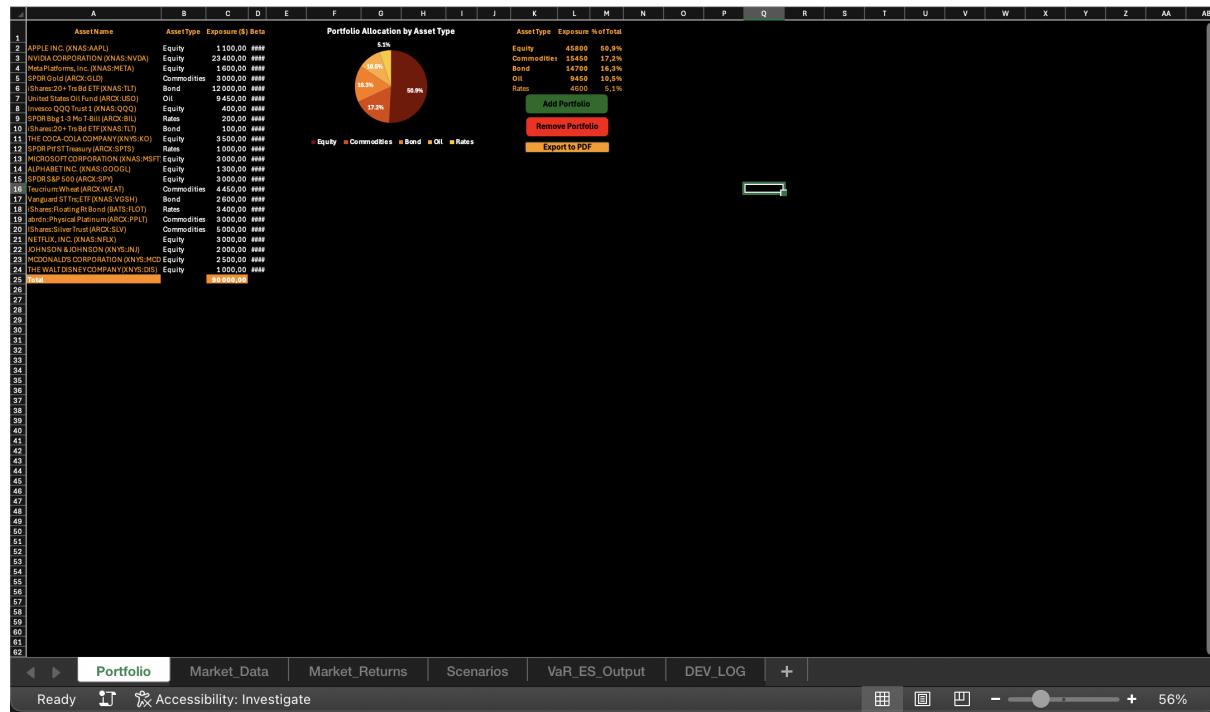


Figure 4 – image complète de la feuille "portfolio"

Ticker	Price	Beta	Market Cap	Asset Type	Returns	Previous close
AAPL	\$ 245,27	1,09	\$ 3 639 903 000 000	Equity	-3,45221%	\$ 254,04
NVDA	\$ 183,16	2,11	\$ 4 450 788 000 000	Equity	-4,88653%	\$ 192,57
MSFT	\$ 510,96	1,02	\$ 3 798 051 000 000	Equity	-2,18989%	\$ 522,40
META	\$ 705,30	1,22	\$ 1 771 813 000 000	Equity	-3,84589%	\$ 733,51
GOOGL	\$ 236,57	1,04	\$ 2 866 073 000 000	Equity	-2,05358%	\$ 241,53
SPY	\$ 653,02	1,00	\$ 672 726 673 351	Equity	-2,70278%	\$ 671,16
TLT	\$ 90,62	1,08	\$ 48 818 461 640	Bond	1,61471%	\$ 89,18
GLD	\$ 369,12	0,82	\$ 82 394 162 386,91	Commodities	1,00977%	\$ 365,43
USO	\$ 69,39	1,02	\$ 899 405 944,95	Oil	-4,30285%	\$ 72,51
QQQ	\$ 589,50	1,0232	\$ 385 755 187 983,10	Equity	-3,47143%	\$ 610,70
IEF	\$ 96,91	1,20	\$ 38 510 844 547,14	Bond	0,62299%	\$ 96,31
BPI	\$ 67,14	#FIELD!	#FIELD!	Bond	-0,63638%	\$ 67,57
WINT	\$ 102,88	#FIELD!	#FIELD!	Rates	-0,07808%	\$ 102,97
XLE	\$ 85,22	1,1548	\$ 26 926 057 790,77	Oil	-2,89426%	\$ 87,76
DBO	\$ 12,67	0,9418	\$ 219 370 903,76	Bond	-4,08781%	\$ 13,21
CNQ	\$ 4,90	3,5785	\$ 2 702 360,00	Oil	6,52174%	\$ 4,60
SLV	\$ 45,43	1,4173	\$ 23 276 965 580,08	Commodities	1,61038%	\$ 44,71
DBA	\$ 26,32	0,6887	\$ 791 801 255,06	Commodities	-0,64175%	\$ 26,49
WEAT	\$ 4,04	0,9987	\$ 117 346 706,78	Commodities	-1,22249%	\$ 4,09
GLDM	\$ 79,44	0,8592	\$ 20 854 265 334,53	Commodities	1,04299%	\$ 78,62
PPLT	\$ 145,79	0,8326	\$ 1 957 887 831,98	Commodities	-1,27311%	\$ 147,67
TIPS	\$ 26,99	0,9975	\$ 14 430 226 377,48	Rates	0,37189%	\$ 26,89
BATS	\$ 50,89	#FIELD!	\$ 8 918 495 068,16	Rates	-0,11776%	\$ 50,95
SPTS	\$ 29,32	1,0429	\$ 5 752 414 088,44	Rates	0,17082%	\$ 29,27
SPTL	\$ 27,28	1,0029	\$ 11 888 783 730,73	Rates	1,37495%	\$ 26,91
BIL	\$ 91,57	0,7564	\$ 42 140 690 455,39	Rates	0,04370%	\$ 91,53
BRK.B	\$ 489,13	0,762	\$ 1 055 469 000 000,00	Equity	-1,53199%	\$ 496,74
JPM	\$ 300,89	1,1496	\$ 827 373 400 000,00	Equity	-1,51867%	\$ 305,53
BAC	\$ 48,65	1,3493	\$ 360 348 000 000,00	Equity	-2,28962%	\$ 49,79
XOM	\$ 110,73	0,4732	\$ 472 069 300 000,00	Equity	-1,93074%	\$ 112,91
CVX	\$ 148,90	0,8155	\$ 304 856 900 000,00	Equity	-1,80691%	\$ 151,64
JNJ	\$ 190,72	0,3599	\$ 459 318 400 000,00	Equity	-0,18840%	\$ 191,08
KO	\$ 67,04	0,4142	\$ 288 517 800 000,00	Equity	1,00949%	\$ 66,37
MCD	\$ 297,01	0,5027	\$ 211 947 700 000,00	Equity	1,08914%	\$ 293,81

Figure 5 – Les données du Marché

Date	APPLE INC. ↓ (XNAS:AAPL)	NVIDIA CORPORATIO N (XNAS:NVDA)	MICROSOFT CORPORATIO N (XNAS:MSFT)	Meta Platforms, Inc. (XNAS:METAC)	ALPHABET INC. (XNAS:GOOGL)	SPDR S&P 500 (ARCX:SPY)	iShares:20 + Trs Bd ETF (XNAS:TLT)					
Date	Close	Return	Close	Return	Close	Return	Close	Return	Close	Return	Close	Return
14/10/2024	\$ 231,30	\$ 138,07	\$ 419,14	\$ 590,42	\$ 164,96	\$ 584,32	\$ 93,77					
15/10/2024	\$ 233,85	0,011	\$ 131,80	-0,047	\$ 418,74	-0,001	\$ 586,27	-0,007	\$ 165,46	0,003	\$ 579,78	-0,0078
16/10/2024	\$ 231,78	-0,0089	\$ 135,72	0,0313	\$ 416,12	-0,0063	\$ 576,79	-0,0162	\$ 165,16	-0,002	\$ 582,30	0,0043
17/10/2024	\$ 232,15	0,0016	\$ 136,93	0,0089	\$ 416,72	0,00144	\$ 576,93	0,0002	\$ 162,93	-0,014	\$ 582,35	9E-05
18/10/2024	\$ 235,00	0,0013	\$ 138,00	0,0078	\$ 418,16	0,00346	\$ 576,47	-0,0008	\$ 163,42	0,003	\$ 584,59	0,0038
21/10/2024	\$ 236,48	0,0063	\$ 143,71	0,0414	\$ 418,78	0,00148	\$ 575,16	-0,0023	\$ 164,07	0,004	\$ 583,63	-0,0016
22/10/2024	\$ 235,86	-0,0026	\$ 143,59	-0E-04	\$ 427,51	0,02085	\$ 582,01	0,0119	\$ 165,14	0,0065	\$ 583,32	-0,0005
23/10/2024	\$ 230,76	-0,0216	\$ 139,56	-0,028	\$ 424,60	-0,0068	\$ 563,69	-0,0315	\$ 162,78	-0,014	\$ 577,99	-0,0091
24/10/2024	\$ 230,57	-0,0008	\$ 140,41	0,0061	\$ 424,73	0,00031	\$ 567,78	0,0073	\$ 162,72	-4E-04	\$ 579,24	0,0022
25/10/2024	\$ 231,41	0,0036	\$ 141,54	0,008	\$ 428,15	0,00805	\$ 573,25	0,0096	\$ 165,27	0,0157	\$ 579,04	-0,0003
28/10/2024	\$ 233,40	0,0086	\$ 140,52	-0,007	\$ 426,59	-0,0036	\$ 578,16	0,0086	\$ 166,72	0,0088	\$ 580,83	0,0031
29/10/2024	\$ 233,67	0,0012	\$ 141,25	0,0052	\$ 431,95	0,01256	\$ 593,28	0,0262	\$ 169,68	0,0178	\$ 581,77	0,0016
30/10/2024	\$ 230,10	-0,0153	\$ 139,34	-0,014	\$ 432,53	0,00134	\$ 591,80	-0,0025	\$ 174,46	0,0282	\$ 580,01	-0,0003
31/10/2024	\$ 225,91	-0,0182	\$ 132,78	-0,047	\$ 406,35	-0,0605	\$ 567,58	-0,0409	\$ 171,11	-0,019	\$ 568,64	-0,0196
01/11/2024	\$ 222,91	-0,0133	\$ 135,40	0,0199	\$ 410,37	0,00898	\$ 567,16	-0,0007	\$ 171,29	0,0011	\$ 571,04	0,0042
04/11/2024	\$ 222,01	-0,004	\$ 136,05	0,0048	\$ 408,46	-0,0047	\$ 560,68	-0,0114	\$ 169,24	-0,012	\$ 569,81	-0,0022
05/11/2024	\$ 223,45	0,0065	\$ 139,91	0,0284	\$ 411,46	0,00734	\$ 572,43	0,021	\$ 169,74	0,003	\$ 576,70	0,0121
06/11/2024	\$ 222,72	-0,0033	\$ 145,61	0,0407	\$ 420,18	0,02119	\$ 572,05	-0,0007	\$ 176,51	0,0399	\$ 591,04	0,0249
07/11/2024	\$ 227,48	0,0214	\$ 148,88	0,0225	\$ 425,43	0,01249	\$ 591,70	0,0344	\$ 180,75	0,024	\$ 595,61	0,0077
08/11/2024	\$ 226,96	-0,0023	\$ 147,63	-0,008	\$ 422,54	-0,0068	\$ 589,34	-0,004	\$ 178,35	-0,013	\$ 598,19	0,0043
11/11/2024	\$ 224,23	-0,012	\$ 145,26	-0,016	\$ 418,01	-0,0107	\$ 583,17	-0,0105	\$ 180,85	0,0112	\$ 598,76	0,001
12/11/2024	\$ 224,23	0	\$ 148,29	0,0209	\$ 423,03	0,01207	\$ 584,82	0,028	\$ 181,62	0,007	\$ 596,90	-0,0031
13/11/2024	\$ 225,12	0,004	\$ 146,27	-0,014	\$ 425,20	0,00513	\$ 580,00	-0,0082	\$ 178,88	-0,015	\$ 597,19	0,0005
14/11/2024	\$ 228,22	0,0138	\$ 146,76	0,0033	\$ 426,89	0,00397	\$ 577,16	-0,0049	\$ 175,58	-0,018	\$ 593,35	-0,0064
15/11/2024	\$ 225,00	-0,0141	\$ 141,98	-0,033	\$ 415,00	-0,0279	\$ 554,08	-0,04	\$ 172,49	-0,018	\$ 585,75	-0,0128
18/11/2024	\$ 228,02	0,0134	\$ 140,15	-0,013	\$ 415,76	0,00183	\$ 554,40	0,0006	\$ 175,30	0,0163	\$ 588,15	0,0041
19/11/2024	\$ 228,28	0,0011	\$ 147,01	0,0489	\$ 417,79	0,00488	\$ 561,09	0,0121	\$ 178,12	0,0161	\$ 590,30	0,0037
20/11/2024	\$ 229,00	0,0032	\$ 145,89	-0,008	\$ 415,49	-0,0055	\$ 565,52	0,0079	\$ 175,98	-0,012	\$ 590,50	0,0003
21/11/2024	\$ 228,52	-0,0021	\$ 146,67	0,0053	\$ 412,87	-0,0063	\$ 563,09	-0,0043	\$ 167,63	-0,047	\$ 593,67	0,0054
22/11/2024	\$ 229,87	0,0059	\$ 141,95	-0,032	\$ 417,00	-0,01	\$ 559,14	-0,007	\$ 184,76	-0,017	\$ 595,51	0,0031
25/11/2024	\$ 232,87	0,0181	\$ 136,02	-0,042	\$ 418,79	0,00429	\$ 565,11	0,0107	\$ 167,65	0,0175	\$ 597,53	0,0034
26/11/2024	\$ 235,06	0,0094	\$ 136,92	0,0066	\$ 427,99	0,02197	\$ 573,54	0,0149	\$ 169,12	0,0088	\$ 600,65	0,0052
27/11/2024	\$ 234,93	-0,0006	\$ 135,34	-0,012	\$ 422,99	-0,0117	\$ 569,20	-0,0076	\$ 169,23	0,0007	\$ 598,83	-0,003
29/11/2024	\$ 237,33	0,0102	\$ 138,25	0,0215	\$ 423,46	0,00111	\$ 574,32	0,009	\$ 168,95	-0,002	\$ 602,55	0,0062
02/12/2024	\$ 239,59	-0,0095	\$ 138,63	0,0027	\$ 430,98	0,01776	\$ 592,83	0,0322	\$ 171,49	0,015	\$ 603,63	0,0018
03/12/2024	\$ 242,65	0,0128	\$ 140,26	0,0118	\$ 431,20	0,00051	\$ 613,65	0,0351	\$ 171,34	-9E-04	\$ 603,91	0,0005
04/12/2024	\$ 243,01	0,0015	\$ 145,14	0,0348	\$ 437,42	0,01442	\$ 613,78	0,0002	\$ 174,37	0,0177	\$ 607,66	0,0062
05/12/2024	\$ 243,04	0,0001	\$ 145,06	-6E-04	\$ 442,62	0,01189	\$ 608,93	-0,0079	\$ 172,64	-0,01	\$ 606,66	-0,0016
06/12/2024	\$ 242,84	-0,0008	\$ 142,44	-0,018	\$ 443,57	0,00215	\$ 623,77	0,0244	\$ 174,71	0,012	\$ 607,81	0,0019
09/12/2024	\$ 246,75	0,0161	\$ 138,81	-0,025	\$ 446,02	0,00552	\$ 613,57	-0,0164	\$ 175,37	0,0038	\$ 604,68	-0,0051
10/12/2024	\$ 247,77	0,0041	\$ 135,07	-0,027	\$ 443,33	-0,006	\$ 619,32	0,0094	\$ 185,17	0,0559	\$ 602,80	-0,0031

Figure 6 – Market Returns

Scenario Name	Asset Class	Shock (%)	Time Added	Delete Scenario	Add Scenario	Run Stress Test
Global Equity Crash	Equity	-25	12/10/2025 21:30			
Global Equity Crash	Commodities	-10	12/10/2025 21:30			
Global Equity Crash	Bond	12	12/10/2025 21:30			
Global Equity Crash	Oil	-15	12/10/2025 21:30			
Global Equity Crash	Rates	20	12/10/2025 21:30			
Inflation Shock	Equity	-10	12/10/2025 21:32			
Inflation Shock	Commodities	12	12/10/2025 21:32			
Inflation Shock	Bond	-15	12/10/2025 21:32			
Inflation Shock	Oil	18	12/10/2025 21:32			
Inflation Shock	Rates	25	12/10/2025 21:32			
Deflation & recession	Equity	-20	12/10/2025 21:34			
Deflation & recession	Commodities	-8	12/10/2025 21:34			
Deflation & recession	Bond	18	12/10/2025 21:34			
Deflation & recession	Oil	-12	12/10/2025 21:34			
Deflation & recession	Rates	-25	12/10/2025 21:34			
Energy Crisis	Equity	-15	12/10/2025 21:35			
Energy Crisis	Commodities	10	12/10/2025 21:35			
Energy Crisis	Bond	-5	12/10/2025 21:35			
Energy Crisis	Oil	25	12/10/2025 21:35			
Energy Crisis	Rates	10	12/10/2025 21:35			

Figure 7 – Feuille Scenario

China Slowdown	Commodities	-14	12/10/2025 21:36
China Slowdown	Bond	6	12/10/2025 21:36
China Slowdown	Oil	-20	12/10/2025 21:36
China Slowdown	Rates	-8	12/10/2025 21:36
US Rate Spike	Equity	-18	12/10/2025 21:38
US Rate Spike	Commodities	-5	12/10/2025 21:38
US Rate Spike	Bond	-20	12/10/2025 21:38
US Rate Spike	Oil	-8	12/10/2025 21:38
US Rate Spike	Rates	35	12/10/2025 21:38
Geopolitical Conflict	Equity	-17	12/10/2025 21:39
Geopolitical Conflict	Commodities	15	12/10/2025 21:39
Geopolitical Conflict	Bond	-8	12/10/2025 21:39
Geopolitical Conflict	Oil	30	12/10/2025 21:39
Geopolitical Conflict	Rates	12	12/10/2025 21:39
Gold Rush	Equity	-10	12/10/2025 21:40
Gold Rush	Commodities	20	12/10/2025 21:40
Gold Rush	Bond	15	12/10/2025 21:40
Gold Rush	Oil	-10	12/10/2025 21:40
Gold Rush	Rates	-18	12/10/2025 21:40

Figure 8 – zoom de la feuille Scenario

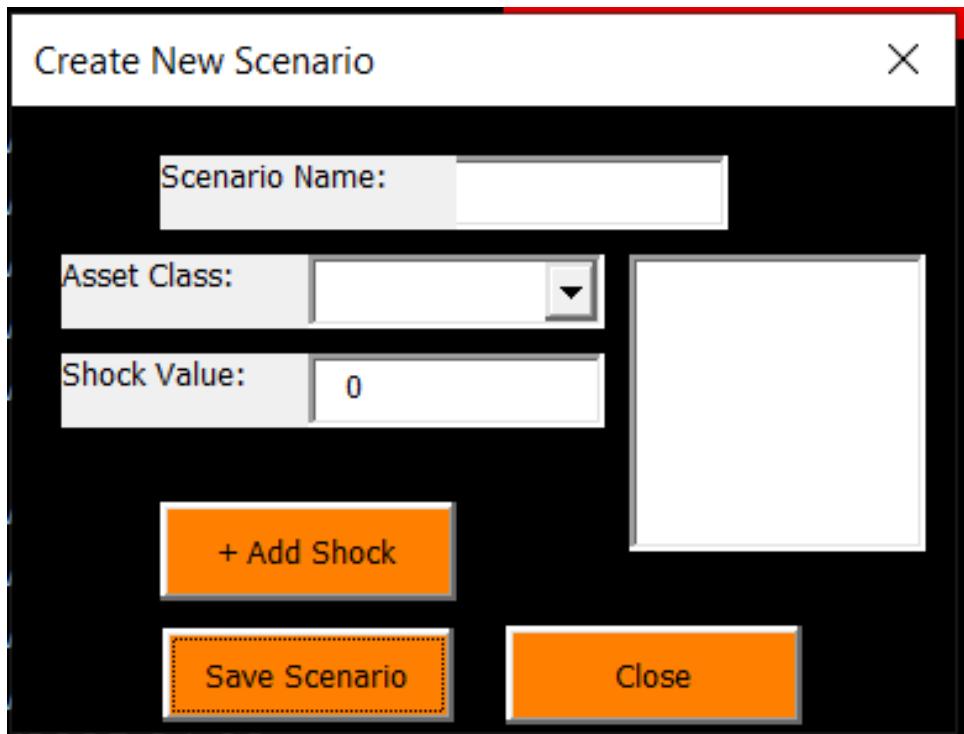


Figure 9 – Formulaire de création d'un nouveau Scenario

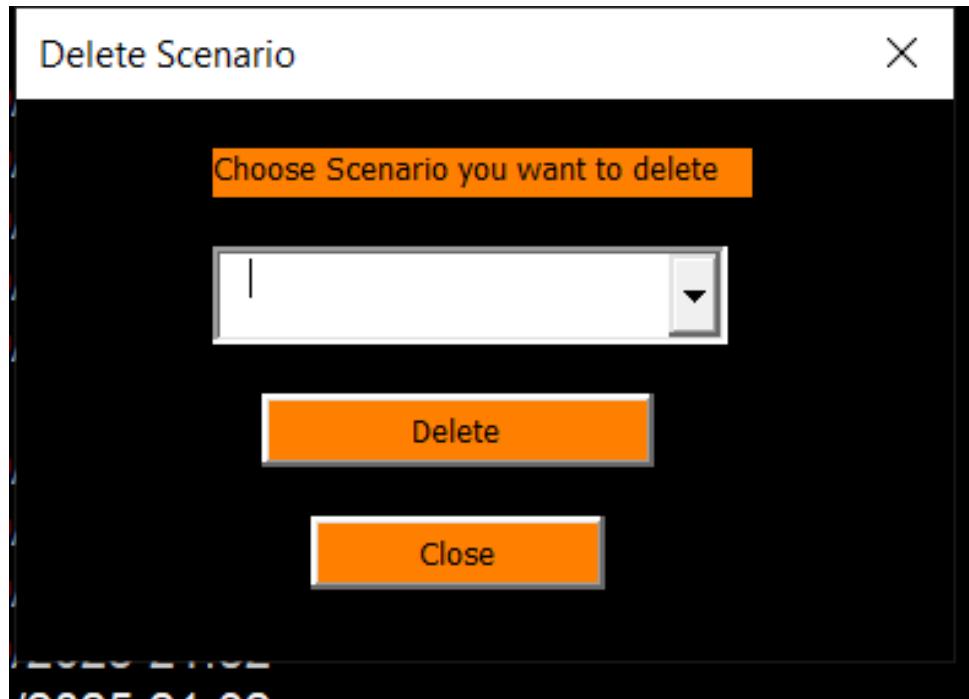


Figure 10 – Formulaire pour supprimer un Scénario

Asset	Asset Type	Shock	Applied PnL	Impact (\$)	PnL Impact (%)
Scenario: Global Equity Crash					
APPLE INC. (XNAS:AAPL)	Equity	-25,00%	-275,00		-25,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-25,00%	-5 850,00		-25,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-25,00%	-400,00		-25,00%
SPDR Gold (ARCX:GLD)	Commodities	-10,00%	-300,00		-10,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	12,00%	1 440,00		12,00%
United States Oil Fund (ARCX:USO)	Oil	-15,00%	-1 417,50		-15,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-25,00%	-100,00		-25,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	20,00%	40,00		20,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	12,00%	12,00		12,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-25,00%	-875,00		-25,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	20,00%	200,00		20,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-25,00%	-750,00		-25,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-25,00%	-325,00		-25,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-25,00%	-750,00		-25,00%
Teucrimum:Wheat (ARCX:WEAT)	Commodities	-10,00%	-445,00		-10,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	12,00%	312,00		12,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	20,00%	680,00		20,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-10,00%	-300,00		-10,00%
IShares:Silver Trust (ARCX:SLV)	Commodities	-10,00%	-500,00		-10,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-25,00%	-750,00		-25,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-25,00%	-500,00		-25,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-25,00%	-625,00		-25,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-25,00%	-250,00		-25,00%
Total - Global Equity Crash			-11 728,50		-13,03%

Figure 11 – Feuille "Value at Risk et Expected shortfall"

Scenario: Post-Crisis (2009)				
APPLE INC. (XNAS:AAPL)	Equity	35,00%	385,00	35,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	35,00%	8 190,00	35,00%
Meta Platforms, Inc. (XNAS:META)	Equity	35,00%	560,00	35,00%
SPDR Gold (ARCX:GLD)	Commodities	20,00%	600,00	20,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-10,00%	-1 200,00	-10,00%
United States Oil Fund (ARCX:USO)	Oil	25,00%	2 362,50	25,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	35,00%	140,00	35,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	5,00%	10,00	5,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-10,00%	-10,00	-10,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	35,00%	1 225,00	35,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	5,00%	50,00	5,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	35,00%	1 050,00	35,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	35,00%	455,00	35,00%
SPDR S&P 500 (ARCX:SPY)	Equity	35,00%	1 050,00	35,00%
Teucrimum:Wheat (ARCX:WEAT)	Commodities	20,00%	890,00	20,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	-10,00%	-260,00	-10,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	5,00%	170,00	5,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	20,00%	600,00	20,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	20,00%	1 000,00	20,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	35,00%	1 050,00	35,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	35,00%	700,00	35,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	35,00%	875,00	35,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	35,00%	350,00	35,00%
Total - Post-Crisis (2009)			20 242,50	22,49%

Figure 12 – Exemple 1 : Post crisis 2009

Scenario: COVID-19 Shock (2020)				
APPLE INC. (XNAS:AAPL)	Equity	-30,00%	-330,00	-30,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-30,00%	-7 020,00	-30,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-30,00%	-480,00	-30,00%
SPDR Gold (ARCX:GLD)	Commodities	-15,00%	-450,00	-15,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	10,00%	1 200,00	10,00%
United States Oil Fund (ARCX:USO)	Oil	-45,00%	-4 252,50	-45,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-30,00%	-120,00	-30,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	-60,00%	-120,00	-60,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	10,00%	10,00	10,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-30,00%	-1 050,00	-30,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	-60,00%	-600,00	-60,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-30,00%	-900,00	-30,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-30,00%	-390,00	-30,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-30,00%	-900,00	-30,00%
Teucrimum:Wheat (ARCX:WEAT)	Commodities	-15,00%	-667,50	-15,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	10,00%	260,00	10,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	-60,00%	-2 040,00	-60,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-15,00%	-450,00	-15,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	-15,00%	-750,00	-15,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-30,00%	-900,00	-30,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-30,00%	-600,00	-30,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-30,00%	-750,00	-30,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-30,00%	-300,00	-30,00%
Total - COVID-19 Shock (2020)			-21 600,00	-24,00%

Figure 13 – Exemple 2 : Crise Covid-19

Scenario: Deflation & recession				
APPLE INC. (XNAS:AAPL)	Equity	-20,00%	-220,00	-20,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-20,00%	-4 680,00	-20,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-20,00%	-320,00	-20,00%
SPDR Gold (ARCX:GLD)	Commodities	-8,00%	-240,00	-8,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	18,00%	2 160,00	18,00%
United States Oil Fund (ARCX:USO)	Oil	-12,00%	-1 134,00	-12,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-20,00%	-80,00	-20,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	-25,00%	-50,00	-25,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	18,00%	18,00	18,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-20,00%	-700,00	-20,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	-25,00%	-250,00	-25,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-20,00%	-600,00	-20,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-20,00%	-260,00	-20,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-20,00%	-600,00	-20,00%
Teucrium:Wheat (ARCX:WEAT)	Commodities	-8,00%	-356,00	-8,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	18,00%	468,00	18,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	-25,00%	-850,00	-25,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-8,00%	-240,00	-8,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	-8,00%	-400,00	-8,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-20,00%	-600,00	-20,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-20,00%	-400,00	-20,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-20,00%	-500,00	-20,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-20,00%	-200,00	-20,00%
Total - Deflation & recession			-10 034,00	-11,15%

Figure 14 – Exemple 3 : Deflation / Recession

Scenario: Energy Crisis				
APPLE INC. (XNAS:AAPL)	Equity	-15,00%	-165,00	-15,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-15,00%	-3 510,00	-15,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-15,00%	-240,00	-15,00%
SPDR Gold (ARCX:GLD)	Commodities	10,00%	300,00	10,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-5,00%	-600,00	-5,00%
United States Oil Fund (ARCX:USO)	Oil	25,00%	2 362,50	25,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-15,00%	-60,00	-15,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	10,00%	20,00	10,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-5,00%	-5,00	-5,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-15,00%	-525,00	-15,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	10,00%	100,00	10,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-15,00%	-450,00	-15,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-15,00%	-195,00	-15,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-15,00%	-450,00	-15,00%
Teucrium:Wheat (ARCX:WEAT)	Commodities	10,00%	445,00	10,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	-5,00%	-130,00	-5,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	10,00%	340,00	10,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	10,00%	300,00	10,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	10,00%	500,00	10,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-15,00%	-450,00	-15,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-15,00%	-300,00	-15,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-15,00%	-375,00	-15,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-15,00%	-150,00	-15,00%
Total - Energy Crisis			-3 237,50	-3,60%

Figure 15 – Exemple 4 : Crise énergétique

Scenario: Gold Rush					
APPLE INC. (XNAS:AAPL)	Equity	-10,00%	-110,00	-10,00%	
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-10,00%	-2 340,00	-10,00%	
Meta Platforms, Inc. (XNAS:META)	Equity	-10,00%	-160,00	-10,00%	
SPDR Gold (ARCX:GLD)	Commodities	20,00%	600,00	20,00%	
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	15,00%	1 800,00	15,00%	
United States Oil Fund (ARCX:USO)	Oil	-10,00%	-945,00	-10,00%	
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-10,00%	-40,00	-10,00%	
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	-18,00%	-36,00	-18,00%	
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	15,00%	15,00	15,00%	
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-10,00%	-350,00	-10,00%	
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	-18,00%	-180,00	-18,00%	
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-10,00%	-300,00	-10,00%	
ALPHABET INC. (XNAS:GOOGL)	Equity	-10,00%	-130,00	-10,00%	
SPDR S&P 500 (ARCX:SPY)	Equity	-10,00%	-300,00	-10,00%	
Teucruium:Wheat (ARCX:WEAT)	Commodities	20,00%	890,00	20,00%	
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	15,00%	390,00	15,00%	
iShares:Floating Rt Bond (BATS:FLOT)	Rates	-18,00%	-612,00	-18,00%	
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	20,00%	600,00	20,00%	
IShares:Silver Trust (ARCX:SLV)	Commodities	20,00%	1 000,00	20,00%	
NETFLIX, INC. (XNAS:NFLX)	Equity	-10,00%	-300,00	-10,00%	
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-10,00%	-200,00	-10,00%	
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-10,00%	-250,00	-10,00%	
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-10,00%	-100,00	-10,00%	
Total - Gold Rush			-1 058,00		-1,18%

Figure 16 – Exemple 5 : Gold Rush

Asset	Asset Type	Shock	Applied PnL	Impact (\$)	PnL Impact (%)
Scenario: Global Equity Crash					
APPLE INC. (XNAS:AAPL)	Equity	-25,00%	-275,00	-25,00%	
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-25,00%	-5 850,00	-25,00%	
Meta Platforms, Inc. (XNAS:META)	Equity	-25,00%	-400,00	-25,00%	
SPDR Gold (ARCX:GLD)	Commodities	-10,00%	-300,00	-10,00%	
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	12,00%	1 440,00	12,00%	
United States Oil Fund (ARCX:USO)	Oil	-15,00%	-1 417,50	-15,00%	
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-25,00%	-100,00	-25,00%	
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	20,00%	40,00	20,00%	
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	12,00%	12,00	12,00%	
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-25,00%	-875,00	-25,00%	
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	20,00%	200,00	20,00%	
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-25,00%	-750,00	-25,00%	
ALPHABET INC. (XNAS:GOOGL)	Equity	-25,00%	-325,00	-25,00%	
SPDR S&P 500 (ARCX:SPY)	Equity	-25,00%	-750,00	-25,00%	
Teucruium:Wheat (ARCX:WEAT)	Commodities	-10,00%	-445,00	-10,00%	
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	12,00%	312,00	12,00%	
iShares:Floating Rt Bond (BATS:FLOT)	Rates	20,00%	680,00	20,00%	
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-10,00%	-300,00	-10,00%	
IShares:Silver Trust (ARCX:SLV)	Commodities	-10,00%	-500,00	-10,00%	
NETFLIX, INC. (XNAS:NFLX)	Equity	-25,00%	-750,00	-25,00%	
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-25,00%	-500,00	-25,00%	
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-25,00%	-625,00	-25,00%	
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-25,00%	-250,00	-25,00%	
Total - Global Equity Crash			-11 728,50		-13,03%

Figure 17 – Exemple 6 : Global Equity Crash

Scenario: Inflation Shock				
APPLE INC. (XNAS:AAPL)	Equity	-10,00%	-110,00	-10,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-10,00%	-2 340,00	-10,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-10,00%	-160,00	-10,00%
SPDR Gold (ARCX:GLD)	Commodities	12,00%	360,00	12,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-15,00%	-1 800,00	-15,00%
United States Oil Fund (ARCX:USO)	Oil	18,00%	1 701,00	18,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-10,00%	-40,00	-10,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	25,00%	50,00	25,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-15,00%	-15,00	-15,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-10,00%	-350,00	-10,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	25,00%	250,00	25,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-10,00%	-300,00	-10,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-10,00%	-130,00	-10,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-10,00%	-300,00	-10,00%
Teucrimum:Wheat (ARCX:WEAT)	Commodities	12,00%	534,00	12,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	-15,00%	-390,00	-15,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	25,00%	850,00	25,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	12,00%	360,00	12,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	12,00%	600,00	12,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-10,00%	-300,00	-10,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-10,00%	-200,00	-10,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-10,00%	-250,00	-10,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-10,00%	-100,00	-10,00%
Total - Inflation Shock			-2 080,00	-2,31%

Figure 18 – Exemple 7 : Inflation Shock

Scenario: China Slowdown				
APPLE INC. (XNAS:AAPL)	Equity	-12,00%	-132,00	-12,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-12,00%	-2 808,00	-12,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-12,00%	-192,00	-12,00%
SPDR Gold (ARCX:GLD)	Commodities	-14,00%	-420,00	-14,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	6,00%	720,00	6,00%
United States Oil Fund (ARCX:USO)	Oil	-20,00%	-1 890,00	-20,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-12,00%	-48,00	-12,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	-8,00%	-16,00	-8,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	6,00%	6,00	6,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-12,00%	-420,00	-12,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	-8,00%	-80,00	-8,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-12,00%	-360,00	-12,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-12,00%	-156,00	-12,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-12,00%	-360,00	-12,00%
Teucrimum:Wheat (ARCX:WEAT)	Commodities	-14,00%	-623,00	-14,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	6,00%	156,00	6,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	-8,00%	-272,00	-8,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-14,00%	-420,00	-14,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	-14,00%	-700,00	-14,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-12,00%	-360,00	-12,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-12,00%	-240,00	-12,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-12,00%	-300,00	-12,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-12,00%	-120,00	-12,00%
Total - China Slowdown			-9 035,00	-10,04%

Figure 19 – Exemple 8 : China Slowdown

Scenario: Geopolitical Conflict				
APPLE INC. (XNAS:AAPL)	Equity	-17,00%	-187,00	-17,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-17,00%	-3 978,00	-17,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-17,00%	-272,00	-17,00%
SPDR Gold (ARCX:GLD)	Commodities	15,00%	450,00	15,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-8,00%	-960,00	-8,00%
United States Oil Fund (ARCX:USO)	Oil	30,00%	2 835,00	30,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-17,00%	-68,00	-17,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	12,00%	24,00	12,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-8,00%	-8,00	-8,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-17,00%	-595,00	-17,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	12,00%	120,00	12,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-17,00%	-510,00	-17,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-17,00%	-221,00	-17,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-17,00%	-510,00	-17,00%
Teucrium:Wheat (ARCX:WEAT)	Commodities	15,00%	667,50	15,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	-8,00%	-208,00	-8,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	12,00%	408,00	12,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	15,00%	450,00	15,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	15,00%	750,00	15,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-17,00%	-510,00	-17,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-17,00%	-340,00	-17,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-17,00%	-425,00	-17,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-17,00%	-170,00	-17,00%
Total - Geopolitical Conflict			-3 257,50	-3,62%

Figure 20 – Exemple 9 : Geopolitical Conflict

Scenario: US Rate Spike				
APPLE INC. (XNAS:AAPL)	Equity	-18,00%	-198,00	-18,00%
NVIDIA CORPORATION (XNAS:NVDA)	Equity	-18,00%	-4 212,00	-18,00%
Meta Platforms, Inc. (XNAS:META)	Equity	-18,00%	-288,00	-18,00%
SPDR Gold (ARCX:GLD)	Commodities	-5,00%	-150,00	-5,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-20,00%	-2 400,00	-20,00%
United States Oil Fund (ARCX:USO)	Oil	-8,00%	-756,00	-8,00%
Invesco QQQ Trust 1 (XNAS:QQQ)	Equity	-18,00%	-72,00	-18,00%
SPDR Bbg 1-3 Mo T-Bill (ARCX:BIL)	Rates	35,00%	70,00	35,00%
iShares:20+ Trs Bd ETF (XNAS:TLT)	Bond	-20,00%	-20,00	-20,00%
THE COCA-COLA COMPANY (XNYS:KO)	Equity	-18,00%	-630,00	-18,00%
SPDR Ptf ST Treasury (ARCX:SPTS)	Rates	35,00%	350,00	35,00%
MICROSOFT CORPORATION (XNAS:MSFT)	Equity	-18,00%	-540,00	-18,00%
ALPHABET INC. (XNAS:GOOGL)	Equity	-18,00%	-234,00	-18,00%
SPDR S&P 500 (ARCX:SPY)	Equity	-18,00%	-540,00	-18,00%
Teucrium:Wheat (ARCX:WEAT)	Commodities	-5,00%	-222,50	-5,00%
Vanguard ST Trs;ETF (XNAS:VGSH)	Bond	-20,00%	-520,00	-20,00%
iShares:Floating Rt Bond (BATS:FLOT)	Rates	35,00%	1 190,00	35,00%
abrdn:Physical Platinum (ARCX:PPLT)	Commodities	-5,00%	-150,00	-5,00%
iShares:Silver Trust (ARCX:SLV)	Commodities	-5,00%	-250,00	-5,00%
NETFLIX, INC. (XNAS:NFLX)	Equity	-18,00%	-540,00	-18,00%
JOHNSON & JOHNSON (XNYS:JNJ)	Equity	-18,00%	-360,00	-18,00%
MCDONALD'S CORPORATION (XNYS:MCD)	Equity	-18,00%	-450,00	-18,00%
THE WALT DISNEY COMPANY (XNYS:DIS)	Equity	-18,00%	-180,00	-18,00%
Total - US Rate Spike			-11 102,50	-12,34%

Figure 21 – Exemple 10 : US Rate Spike

Timestamp	Source	Details
08/10/2025 23:51	RunAllScenarios_ToOutput	START
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenarios found=5
08/10/2025 23:51	RunOneScenario_ToOutput	START LE SINGE
08/10/2025 23:51	RunOneScenario_ToOutput	OK LE SINGE assets_applied=6 total_pnl=-5910
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: LE SINGE rows_added=7
08/10/2025 23:51	RunOneScenario_ToOutput	START BLAbla
08/10/2025 23:51	RunOneScenario_ToOutput	OK BLAbla assets_applied=5 total_pnl=4650
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: BLAbla rows_added=6
08/10/2025 23:51	RunOneScenario_ToOutput	START Crash 2026
08/10/2025 23:51	RunOneScenario_ToOutput	OK Crash 2026 assets_applied=5 total_pnl=4485
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: Crash 2026 rows_added=6
08/10/2025 23:51	RunOneScenario_ToOutput	START test1
08/10/2025 23:51	RunOneScenario_ToOutput	OK test1 assets_applied=5 total_pnl=-8460
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: test1 rows_added=6
08/10/2025 23:51	RunOneScenario_ToOutput	START marwwan
08/10/2025 23:51	RunOneScenario_ToOutput	OK marwwan assets_applied=1 total_pnl=9000
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: marwwan rows_added=2
08/10/2025 23:51	RunAllScenarios_ToOutput	COMPLETED lastRow=43
08/10/2025 23:51	RunAllScenarios_ToOutput	START
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenarios found=5
08/10/2025 23:51	RunOneScenario_ToOutput	START LE SINGE
08/10/2025 23:51	RunOneScenario_ToOutput	OK LE SINGE assets_applied=6 total_pnl=-5910
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: LE SINGE rows_added=7
08/10/2025 23:51	RunOneScenario_ToOutput	START BLAbla
08/10/2025 23:51	RunOneScenario_ToOutput	OK BLAbla assets_applied=5 total_pnl=4650
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: BLAbla rows_added=6
08/10/2025 23:51	RunOneScenario_ToOutput	START Crash 2026
08/10/2025 23:51	RunOneScenario_ToOutput	OK Crash 2026 assets_applied=5 total_pnl=4485
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: Crash 2026 rows_added=6
08/10/2025 23:51	RunOneScenario_ToOutput	START test1
08/10/2025 23:51	RunOneScenario_ToOutput	OK test1 assets_applied=5 total_pnl=-8460
08/10/2025 23:51	RunAllScenarios_ToOutput	Scenario done: test1 rows_added=6

Figure 22 – Feuille Dev Log (afin de traquer)

D. Extraits de code VBA

Cette sous-section présente des captures d'écran du code et des modules clés du projet, illustrant la structure et la logique du programme. Les extraits ont été choisis pour mettre en évidence les parties essentielles du moteur de calcul, de la gestion des scénarios et des interfaces utilisateur.

D.1 Module principal (`mod_Main`)

D.2 Module d'analyse (`mod_Analytics`)

D.3 Module d'entrée/sortie (`mod_IO`)

D.4 Module d'exécution des scénarios (`mod_ScenariosRunner`)

D.5 Module de visualisation (`mod_Visualization`)

D.6 Module d'exportation PDF (`mod_ExportPDF`)

D.7 Formulaires utilisateurs (UserForms)

```

Option Explicit

Sub OpenScenarioForm()
    frmScenario.Show
End Sub

Sub ShowScenarioForm()
    frmScenario.Show
End Sub

Sub RunStressTest()
    Dim wsPortfolio As Worksheet, wsScenarios As Worksheet, wsOutput As Worksheet
    Dim lastRowPortfolio As Long, lastRowScenarios As Long, outputRow As Long
    Dim i As Long, j As Long
    Dim assetName As String, assetType As String
    Dim exposure As Double, beta As Double
    Dim shockType As String, shockValue As Double, shockUnit As String
    Dim pnlImpact As Double
    Dim totalPnL As Double, totalExposure As Double

    Set wsPortfolio = ThisWorkbook.Sheets("Portfolio")
    Set wsScenarios = ThisWorkbook.Sheets("Scenarios")
    Set wsOutput = ThisWorkbook.Sheets("VaR_ES_Output")

    wsOutput.Cells.ClearContents
    wsOutput.Range("A1:E1").Value = Array("Asset", "Asset Type", "Shock Applied", "PnL Impact ($)", "PnL Impact (%)")
    outputRow = 2

    lastRowPortfolio = wsPortfolio.Cells(wsPortfolio.Rows.Count, 1).End(xlUp).Row
    lastRowScenarios = wsScenarios.Cells(wsScenarios.Rows.Count, 1).End(xlUp).Row

    totalPnL = 0
    totalExposure = 0

    For i = 2 To lastRowPortfolio
        assetName = wsPortfolio.Cells(i, 1).Value
        assetType = Trim$(LCase$(wsPortfolio.Cells(i, 2).Value))
        exposure = wsPortfolio.Cells(i, 3).Value
        beta = wsPortfolio.Cells(i, 4).Value

        For j = 2 To lastRowScenarios
            shockType = Trim$(LCase$(wsScenarios.Cells(j, 2).Value))
            shockValue = wsScenarios.Cells(j, 3).Value

            If assetType = shockType Then
                pnlImpact = exposure * (shockValue / 100#)

                wsOutput.Cells(outputRow, 1).Value = assetName
                wsOutput.Cells(outputRow, 2).Value = wsPortfolio.Cells(i, 2).Value
                wsOutput.Cells(outputRow, 3).Value = CStr(shockValue)
                wsOutput.Cells(outputRow, 4).Value = Round(pnlImpact, 2)

                If exposure <> 0 Then
                    wsOutput.Cells(outputRow, 5).Value = Round(pnlImpact / exposure * 100#, 2)
                    wsOutput.Cells(outputRow, 5).NumberFormat = "0.00\%"
                End If
            End If
        Next j
    Next i
End Sub

```

Figure 23 – Début du module `mod_Main` — gestion du cœur du programme.

```

    If exposure <> 0 Then
        wsOutput.Cells(outputRow, 5).Value = Round(pnlImpact / exposure * 100#, 2)
        wsOutput.Cells(outputRow, 5).NumberFormat = "0.00\%"
    Else
        wsOutput.Cells(outputRow, 5).Value = "N/A"
    End If

    totalPnL = totalPnL + pnlImpact
    totalExposure = totalExposure + exposure
    outputRow = outputRow + 1
End If
Next j
Next i

wsOutput.Cells(outputRow, 1).Value = "Total Portfolio"
wsOutput.Cells(outputRow, 4).Value = Round(totalPnL, 2)
If totalExposure <> 0 Then
    wsOutput.Cells(outputRow, 5).Value = Round(totalPnL / totalExposure * 100#, 2)
    wsOutput.Cells(outputRow, 5).NumberFormat = "0.00\%"
Else
    wsOutput.Cells(outputRow, 5).Value = "N/A"
End If
wsOutput.Rows(outputRow).Font.Bold = True

With wsOutput
    .Range("A1:E1").Font.Bold = True
    .Range("A1:E1").HorizontalAlignment = xlCenter
    .Columns("A:E").AutoFit
    .Range("D2:D" & outputRow).NumberFormat = "#,##0.00 [$-407]"
    .Range("E2:E" & outputRow).NumberFormat = "0.00\%"
End With

MsgBox "Stress test complete! Check VaR_ES_Output sheet.", vbInformation
End Sub

Public Sub OpenAddPortfolioForm()
    Dim f As frmAddPortfolio
    Set f = New frmAddPortfolio
    f.Show
End Sub

Sub StyleMarketReturnsSheet_TextColor()
    Dim ws As Worksheet: Set ws = ThisWorkbook.Sheets("Market_Returns")
    Dim lastCol As Long: lastCol = ws.Cells(2, ws.Columns.Count).End(xlToLeft).Column
    Dim lastRow As Long: lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
    Dim j As Long
    Dim rng As Range

    Dim neutralBand As Double: neutralBand = 0.0005

    For j = 1 To lastCol
        If UCase$(Trim$(ws.Cells(2, j).Value)) = "RETURN" Then
            Set rng = ws.Range(ws.Cells(3, j), ws.Cells(lastRow, j))
            rng.FormatConditions.Delete

            Dim fcNeutral As FormatCondition
            Set fcNeutral = rng.FormatConditions.Add(Type:=xlCellValue, _
                Operator:=xlBetween, Formula1:=-neutralBand, Formula2:=neutralBand)
            With fcNeutral
                .Font.Color = RGB(130, 130, 130)
                .StopIfTrue = True
                .SetFirstPriority
            End With
        End If
    Next j
End Sub

```

Figure 24 – Suite du module principal — boucle de calcul et exécution des scénarios.

```

Dim fcNeg As FormatCondition
Set fcNeg = rng.FormatConditions.Add(Type:=xlCellValue, _
    Operator:=xlLess, Formula1:="=0")
fcNeg.Font.Color = RGB(220, 0, 0)

Dim fcPos As FormatCondition
Set fcPos = rng.FormatConditions.Add(Type:=xlCellValue, _
    Operator:=xlGreater, Formula1:="=0")
fcPos.Font.Color = RGB(0, 150, 0)
End If
Next j

MsgBox "Market_Returns styled Ñ text only: red < 0, green > 0, gray near 0.", vbInformation
End Sub

Sub ShowAddScenarioForm()
    frmAddScenario.Show
End Sub

Public Sub OpenRemovePortfolioForm()
    Dim f As frmRemovePortfolio
    Set f = New frmRemovePortfolio
    f.Show
End Sub

Public Sub OpenRemoveScenarioForm()
    Dim f As frmRemoveScenario
    Set f = New frmRemoveScenario
    f.Show
End Sub

Sub UpdateTotalRow()
    Dim ws As Worksheet
    Dim lastDataRow As Long
    Dim totalRow As Long
    Dim totalLabel As Range
    Dim totalValue As Range

    Set ws = ThisWorkbook.Sheets("Portfolio")

    lastDataRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    If LCase(Trim(ws.Cells(lastDataRow, "A").Value)) = "total" Then
        ws.Rows(lastDataRow).ClearContents
        ws.Rows(lastDataRow).Interior.ColorIndex = xlNone
        lastDataRow = lastDataRow - 1
    End If

    totalRow = lastDataRow + 1

    Set totalLabel = ws.Cells(totalRow, "A")
    totalLabel.Value = "Total"
    totalLabel.Font.Bold = True
    totalLabel.Interior.Color = RGB(255, 140, 0)
    totalLabel.Font.Color = RGB(255, 255, 255)

```

Figure 25 – Suite du module principal

```

Set totalValue = ws.Cells(totalRow, "C")
totalValue.Formula = "=SUM(C2:C" & lastDataRow & ")"
totalValue.Font.Bold = True
totalValue.NumberFormat = "#,##0.00"
totalValue.Interior.Color = RGB(255, 140, 0)
totalValue.Font.Color = RGB(255, 255, 255)

ws.Cells(totalRow, "D").ClearContents

With ws.Range("A" & totalRow & ":D" & totalRow).Borders
    .LineStyle = xlContinuous
    .Weight = xlThin
End With
End Sub

Sub UpdatePortfolioPieChart()
    Dim ws As Worksheet
    Dim chartObj As ChartObject
    Dim chartName As String
    Dim dict As Object
    Dim lastRow As Long, i As Long
    Dim assetType As String
    Dim exposure As Double
    Dim totalExposure As Double
    Dim chartLeft As Double, chartTop As Double
    Dim tmpRange As Range

    Set ws = ThisWorkbook.Sheets("Portfolio")
    Set dict = CreateObject("Scripting.Dictionary")
    chartName = "PortfolioPie"

    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row
    If LCase(Trim(ws.Cells(lastRow, "A").Value)) = "total" Then lastRow = lastRow - 1
    If lastRow < 2 Then Exit Sub

    For i = 2 To lastRow
        assetType = Trim(CStr(ws.Cells(i, "B").Value))
        exposure = val(ws.Cells(i, "C").Value)
        If Len(assetType) > 0 And exposure > 0 Then
            If Not dict.Exists(assetType) Then
                dict(assetType) = exposure
            Else
                dict(assetType) = dict(assetType) + exposure
            End If
        End If
    Next i

    For i = 0 To dict.Count - 1
        totalExposure = totalExposure + dict.Items()(i)
    Next i
    If totalExposure = 0 Then Exit Sub

    ws.Range("K:M").ClearContents
    ws.Range("K1").Value = "Asset Type"
    ws.Range("L1").Value = "Exposure"
    ws.Range("M1").Value = "% of Total"

    For i = 0 To dict.Count - 1
        ws.Cells(i + 2, "K").Value = dict.Keys()(i)
        ws.Cells(i + 2, "L").Value = dict.Items()(i)
        ws.Cells(i + 2, "M").Value = Round(dict.Items()(i) / totalExposure, 4)
    Next i

```

Figure 26 – Suite du module principal

```

Set tmpRange = ws.Range("K1:L" & dict.Count + 1)

With ws.Range("K1:M" & dict.Count + 1)
    .Font.Color = RGB(255, 153, 0)
    .Columns.AutoFit
    .Interior.Color = RGB(0, 0, 0)

End With
ws.Range("M2:M" & dict.Count + 1).NumberFormat = "0.0%"

On Error Resume Next
ws.ChartObjects(chartName).Delete
On Error GoTo 0

chartLeft = ws.Columns("K").Left - 350
chartTop = ws.Rows(dict.Count + 4).Top - 146
Set chartObj = ws.ChartObjects.Add(Left:=chartLeft, Top:=chartTop, Width:=320, Height:=210)

chartObj.Name = chartName
chartObj.Chart.SetSourceData Source:=tmpRange
chartObj.Chart.ChartType = xlPie

With chartObj.Chart
    .HasTitle = True
    .ChartTitle.Text = "Portfolio Allocation by Asset Type"
    .ChartTitle.Font.Color = RGB(255, 255, 255)
    .ChartTitle.Font.Bold = True
    .ChartTitle.Font.Size = 15

    .ChartArea.Format.Fill.Visible = msoFalse
    .PlotArea.Format.Fill.Visible = msoFalse
    .ChartArea.Format.Line.Visible = msoFalse
    .PlotArea.Format.Line.Visible = msoFalse

    .HasLegend = True
    .Legend.Position = xlLegendPositionBottom
    .Legend.Font.Color = RGB(255, 255, 255)
    .Legend.Font.Size = 12
    .Legend.Font.Bold = True

Dim s As Series
Set s = .FullSeriesCollection(1)
s.ApplyDataLabels xlDataLabelsShowPercent

For i = 1 To s.Points.Count
    s.DataLabels(i).Text = Format(ws.Cells(i + 1, "M").Value, "0.0%")
Next i

With s.DataLabels
    .Font.Color = RGB(255, 255, 255)
    .Font.Size = 10
    .Font.Bold = True
End With

Dim colorMap As Object: Set colorMap = CreateObject("Scripting.Dictionary")
colorMap.Add "Equity", RGB(120, 15, 0)
colorMap.Add "Commodities", RGB(220, 70, 0)
colorMap.Add "Bond", RGB(255, 120, 0)
colorMap.Add "Oil", RGB(255, 170, 60)
colorMap.Add "Rates", RGB(255, 200, 40)
colorMap.Add "FX", RGB(255, 230, 80)

```

Figure 27 – Suite du module principal

```
Dim colorMap As Object: Set colorMap = CreateObject("Scripting.Dictionary")
colorMap.Add "Equity", RGB(120, 15, 0)
colorMap.Add "Commodities", RGB(220, 70, 0)
colorMap.Add "Bond", RGB(255, 120, 0)
colorMap.Add "Oil", RGB(255, 170, 60)
colorMap.Add "Rates", RGB(255, 200, 40)
colorMap.Add "FX", RGB(255, 230, 80)

Dim pt As Point
For i = 1 To s.Points.Count
    assetType = Trim(CStr(ws.Cells(i + 1, "K").Value))
    If colorMap.Exists(assetType) Then
        s.Points(i).Format.Fill.ForeColor.RGB = colorMap(assetType)
    Else
        s.Points(i).Format.Fill.ForeColor.RGB = RGB(130, 130, 130)
    End If
Next i
End With
End Sub
```

Figure 28 – Suite du module principal

```

mod_Analytics :

Option Explicit

Public Sub ComputeVaR95_FromMarketReturns()
    Const LABEL_ROW As Long = 2
    Const DATA_START As Long = 3

    Dim wsP As Worksheet, wsR As Worksheet, wsOut As Worksheet
    Dim lastRowP As Long, outRow As Long, i As Long
    Dim ticker As String, retCol As Long
    Dim lastRowR As Long, n As Long, r As Long
    Dim v As Variant, exposure As Double
    Dim arr() As Double, cnt As Long, idx As Long, var95 As Double
    Dim tickMap As Object

    On Error GoTo Fail

    Set wsP = ThisWorkbook.Sheets("Portfolio")
    Set wsR = ThisWorkbook.Sheets("Market_Returns")

    On Error Resume Next
    Set wsOut = ThisWorkbook.Sheets("Risk_Results")
    If wsOut Is Nothing Then
        Set wsOut = ThisWorkbook.Sheets.Add
        wsOut.Name = "Risk_Results"
    End If
    On Error GoTo 0

    wsOut.Cells.ClearContents
    wsOut.Range("A1:C1").Value = Array("Asset", "Asset Type", "VaR (95%)")
    outRow = 2

    Set tickMap = BuildReturnColumnMap(wsR)

    lastRowP = wsP.Cells(wsP.Rows.Count, 1).End(xlUp).Row
    Application.ScreenUpdating = False

    For i = 2 To lastRowP
        ticker = UCase$(ExtractTicker(CStr(wsP.Cells(i, 1).Value)))
        exposure = CDbl(wsP.Cells(i, 3).Value)

        wsOut.Cells(outRow, 1).Value = wsP.Cells(i, 1).Value
        wsOut.Cells(outRow, 2).Value = wsP.Cells(i, 2).Value

        If Not tickMap.Exists(ticker) Then
            wsOut.Cells(outRow, 3).Value = "No data"
            outRow = outRow + 1
            GoTo NextI
        End If

        retCol = CLng(tickMap(ticker))
        lastRowR = wsR.Cells(wsR.Rows.Count, retCol).End(xlUp).Row
        If lastRowR < DATA_START Then
            wsOut.Cells(outRow, 3).Value = "No data"
            outRow = outRow + 1
            GoTo NextI
        End If
    NextI:
    outRow = outRow + 1
End Sub

```

Figure 29 – Calculs statistiques et fonctions utilitaires du module mod_Analytics.

```

n = lastRowR - DATA_START + 1
ReDim arr(1 To n): cnt = 0
For r = DATA_START To lastRowR
    v = wsR.Cells(r, retCol).Value
    If IsNumeric(v) Then
        cnt = cnt + 1
        arr(cnt) = CDbl(v)
    End If
Next r

If cnt < 5 Then
    wsOut.Cells(outRow, 3).Value = "Insufficient data"
    outRow = outRow + 1
    GoTo NextI
End If

ReDim Preserve arr(1 To cnt)
QSortDbl arr, 1, cnt
idx = Application.WorksheetFunction.Max(1,
    Application.WorksheetFunction.RoundUp(0.05 * cnt, 0))
var95 = -arr(idx) * exposure

wsOut.Cells(outRow, 3).Value = Round(var95, 4)
wsOut.Cells(outRow, 3).NumberFormat = "#,##0.0000"
outRow = outRow + 1

NextI:
Next i

With wsOut
    .Range("A1:C1").Font.Bold = True
    .Columns("A:C").AutoFit
End With

Application.ScreenUpdating = True
MsgBox "VaR 95% calculée ? Risk_Results", vbInformation
Exit Sub

Fail:
Application.ScreenUpdating = True
MsgBox "Erreur ComputeVaR95_FromMarketReturns : " & Err.Description, vbExclamation
End Sub

Private Function BuildReturnColumnMap(ws As Worksheet) As Object
Dim d As Object: Set d = CreateObject("Scripting.Dictionary")
Dim lastCol As Long, c As Long
Dim head As String, lbl As String, tick As String

lastCol = ws.Cells(2, ws.Columns.Count).End(xlToLeft).Column

For c = 2 To lastCol
    lbl = ""
    If Not IsError(ws.Cells(2, c).Value) Then
        lbl = Trim$(UCase$(CStr(ws.Cells(2, c).Value)))
    End If

    If lbl = "RETURN" Or InStr(lbl, "RETURN") > 0 Then
        If Not IsError(ws.Cells(1, c - 1).Value) Then
            head = Trim$(CStr(ws.Cells(1, c - 1).Value))
            tick = ExtractTicker(head)
            If Len(tick) > 0 Then
                If Not d.Exists(tick) Then
                    d.Add tick, c
                    Debug.Print "Mapping ajouté : " & head & " -> colonne " & c
                End If
            End If
        End If
    End If
End Function

```

Figure 30 – Suite du module analytics

```

        End If
    End If
End If
Next c

Set BuildReturnColumnMap = d
End Function

Private Sub QSortDbl(arr() As Double, ByVal first As Long, ByVal last As Long)
    Dim i As Long, j As Long, p As Double, t As Double
    i = first: j = last
    p = arr((first + last) \ 2)
    Do While i <= j
        Do While arr(i) < p: i = i + 1: Loop
        Do While arr(j) > p: j = j - 1: Loop
        If i <= j Then
            t = arr(i): arr(i) = arr(j): arr(j) = t
            i = i + 1: j = j - 1
        End If
    Loop
    If first < j Then QSortDbl arr, first, j
    If i < last Then QSortDbl arr, i, last
End Sub

Private Function ExtractTicker(ByVal s As String) As String
    Dim pL As Long, pR As Long, core As String, pColon As Long
    s = Trim$(s)
    pL = InStrRev(s, "(")
    pR = InStrRev(s, ")")
    If pL > 0 And pR > pL Then
        core = Mid$(s, pL + 1, pR - pL - 1)
        pColon = InStrRev(core, ":")
        If pColon > 0 Then
            ExtractTicker = UCASE$(Mid$(core, pColon + 1))
        Else
            ExtractTicker = UCASE$(core)
        End If
    Else
        ExtractTicker = UCASE$(s)
    End If
End Function

Sub Debug_ShowReturnMapping()
    Dim wsR As Worksheet
    Dim map As Object
    Dim key As Variant
    Set wsR = ThisWorkbook.Sheets("Market_Returns")
    Set map = BuildReturnColumnMap(wsR)

    Debug.Print "==== Mapping détecté ==="
    For Each key In map.Keys
        Debug.Print key, map(key)
    Next key
    Debug.Print "==== Fin du mapping ==="
End Sub

```

Figure 31 – Suite du module analytics

```

mod_IO :

Option Explicit

Public Function SafeDouble(c As Range, ByRef outVal As Double) As Boolean
    On Error GoTo EH
    If c Is Nothing Then GoTo EH
    If IsError(c.Value) Then GoTo EH
    If Len(c.Value) = 0 Then GoTo EH
    If Not IsNumeric(c.Value) Then GoTo EH
    outVal = CDbl(c.Value)
    SafeDouble = True
    Exit Function
EH:
    SafeDouble = False
End Function

Option Explicit

Public Function GetAssetType(ticker As String) As String
    Dim ws As Worksheet, r As Range
    Set ws = ThisWorkbook.Sheets("Market_Data")

    Set r = ws.Columns(1).Find(What:=ticker, LookAt:=xlPart, MatchCase:=False)
    If Not r Is Nothing Then
        GetAssetType = CStr(ws.Cells(r.Row, "E").Value)
    Else
        GetAssetType = "Unknown"
    End If
End Function

Public Function GetBeta(ticker As String) As Double
    Dim ws As Worksheet, r As Range, v As Variant
    Set ws = ThisWorkbook.Sheets("Market_Data")

    Set r = ws.Columns(1).Find(What:=ticker, LookAt:=xlPart, MatchCase:=False)
    If Not r Is Nothing Then
        v = ws.Cells(r.Row, "C").Value
        If IsNumeric(v) Then
            GetBeta = CDbl(v)
        Else
            GetBeta = 0
        End If
    Else
        GetBeta = 0
    End If
End Function

```

Figure 32 – Fonctions de lecture et de conversion des données dans mod_IO.

```
Public Function ExtractTicker(ByVal s As String) As String
    Dim pL As Long, pR As Long, core As String, pColon As Long
    s = Trim$(s)
    pL = InStrRev(s, "(")
    pR = InStrRev(s, ")")
    If pL > 0 And pR > pL Then
        core = Mid$(s, pL + 1, pR - pL - 1)
        pColon = InStrRev(core, ":")
        If pColon > 0 Then
            ExtractTicker = UCASE$(Mid$(core, pColon + 1))
        Else
            ExtractTicker = UCASE$(core)
        End If
    Else
        ExtractTicker = UCASE$(s)
    End If
End Function
```

Figure 33 – Suite du module IO

```

mod_ScenarioRunner :

Option Explicit

Public Sub RunAllScenarios_ToOutput()
    On Error Resume Next

    Dim wsOut As Worksheet, wsP As Worksheet, wsS As Worksheet
    Dim scenNames As Collection, nm As Variant
    Dim lastRowP As Long, lastRowS As Long, lastRowOut As Long
    Dim outRow As Long, i As Long, r As Long

    Dim assetName As String, assetType As String
    Dim exposure As Double, duration As Double, shockVal As Double
    Dim pnl As Double, scenTotal As Double, scenExpo As Double

    Set wsOut = ThisWorkbook.Sheets("VaR_ES_Output")
    Set wsP = ThisWorkbook.Sheets("Portfolio")
    Set wsS = ThisWorkbook.Sheets("Scenarios")

    With wsOut
        .Cells.Clear
        .Cells.Interior.Color = RGB(0, 0, 0)
        .Cells.Font.Name = "Consolas"
        .Cells.Font.Color = RGB(255, 153, 0)
    End With

    With wsOut.Range("A1:E1")
        .Value = Array("Asset", "Asset Type", "Shock Applied", "PnL Impact ($)", "PnL Impact (%)")
        .Font.Bold = True
        .Font.Color = RGB(255, 153, 0)
        .Interior.Color = RGB(0, 0, 0)
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
    End With
    wsOut.Rows(1).RowHeight = 22

    outRow = 3

    Set scenNames = GetUniqueScenarioNames_Sheet()
    If scenNames.Count = 0 Then
        MsgBox "No scenarios found in 'Scenarios' sheet.", vbExclamation
        Exit Sub
    End If

    lastRowP = wsP.Cells(wsP.Rows.Count, 1).End(xlUp).Row
    lastRowS = wsS.Cells(wsS.Rows.Count, 1).End(xlUp).Row

    For Each nm In scenNames
        scenTotal = 0
        scenExpo = 0

        wsOut.Cells(outRow, 1).Value = "Scenario: " & CStr(nm)
        With wsOut.Cells(outRow, 1)
            .Font.Bold = True
            .Font.Color = RGB(255, 255, 255)
            .Font.Size = 12
        End With
        outRow = outRow + 1
    Next nm
End Sub

```

Figure 34 – Module Scenario : Exécution et transfert des scénarios vers le moteur de calcul.

```

For i = 2 To lastRowP
    assetName = wsP.Cells(i, 1).Value
    assetType = LCase$(Trim$(wsP.Cells(i, 2).Value))
    exposure = val(wsP.Cells(i, 3).Value)
    duration = IIf(IsNumeric(wsP.Cells(i, 6).Value), val(wsP.Cells(i, 6).Value), 0)

    shockVal = 0
    For r = 2 To lastRowS
        If StrComp(Trim$(wsS.Cells(r, 1).Value), CStr(nm), vbTextCompare) = 0 Then
            If LCase$(Trim$(wsS.Cells(r, 2).Value)) = assetType Then
                shockVal = val(wsS.Cells(r, 3).Value)
                Exit For
            End If
        End If
    Next r
    If shockVal = 0 Then GoTo NextAsset

    Select Case assetType
        Case "equity", "oil", "commodity", "commodities", "fx"
            pnl = exposure * (shockVal / 100#)
        Case "rates", "bond"
            If duration <> 0 Then
                pnl = -duration * exposure * (shockVal / 100#)
            Else
                pnl = exposure * (shockVal / 100#)
            End If
        Case Else
            pnl = exposure * (shockVal / 100#)
    End Select

    wsOut.Cells(outRow, 1).Value = assetName
    wsOut.Cells(outRow, 2).Value = wsP.Cells(i, 2).Value
    wsOut.Cells(outRow, 3).Value = shockVal / 100
    wsOut.Cells(outRow, 4).Value = Round(pnl, 2)
    wsOut.Cells(outRow, 5).Value = IIf(exposure <> 0, pnl / exposure, 0)

    scenTotal = scenTotal + pnl
    scenExpo = scenExpo + exposure
    outRow = outRow + 1

NextAsset:
Next i

wsOut.Cells(outRow, 1).Value = "Total - " & CStr(nm)
wsOut.Cells(outRow, 1).Font.Color = RGB(255, 255, 255)
wsOut.Cells(outRow, 4).Value = Round(scenTotal, 2)
wsOut.Cells(outRow, 5).Value = IIf(scenExpo <> 0, scenTotal / scenExpo, 0)
wsOut.Rows(outRow).Font.Bold = True
wsOut.Rows(outRow).Font.Color = RGB(255, 153, 0)
outRow = outRow + 2
Next nm

With wsOut
    lastRowOut = .Cells(.Rows.Count, "A").End(xlUp).Row
    .Columns("A:E").AutoFit

    .Range("C2:C" & lastRowOut).NumberFormat = "0.00%"
    .Range("D2:D" & lastRowOut).NumberFormat = "#,##0.00"
    .Range("E2:E" & lastRowOut).NumberFormat = "0.00%"

```

Figure 35 – Suite du module ScenarioRunner

```

.Range("A2:A" & lastRowOut).Font.Color = RGB(255, 153, 0)
.Range("B2:B" & lastRowOut).Font.Color = RGB(255, 255, 255)

.Range("C2:E" & lastRowOut).FormatConditions.Delete
With .Range("C2:E" & lastRowOut)
    With .FormatConditions.Add(xlCellValue, xlLess, "=0")
        .Font.Color = RGB(255, 80, 80)
    End With
    With .FormatConditions.Add(xlCellValue, xlGreater, "=0")
        .Font.Color = RGB(60, 255, 120)
    End With
End With

Dim cell As Range
For Each cell In .Range("A2:A" & lastRowOut)
    If InStr(1, cell.Value, "Scenario:", vbTextCompare) > 0 Or _
    InStr(1, cell.Value, "Total -", vbTextCompare) > 0 Then
        cell.Font.Color = RGB(255, 255, 255)
    Else
        cell.Font.Color = RGB(255, 153, 0)
    End If
Next cell

.Range("B2:B" & lastRowOut).Font.Color = RGB(255, 255, 255)

.Cells.Borders.Color = RGB(40, 40, 40)

Application.GoTo .Range("A1"), True
End With

MsgBox "? Stress test completed - results written to 'VaR_ES_Output'.", vbInformation
End Sub
Sub FixScenarioSpacing()
    Dim ws As Worksheet, lastRow As Long, i As Long
    Dim curName As String, prevName As String

    Set ws = ThisWorkbook.Sheets("Scenarios")
    Application.ScreenUpdating = False
    Application.EnableEvents = False

    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
    For i = 2 To lastRow
        ws.Cells(i, 1).Value = CleanText(ws.Cells(i, 1).Value)
    Next i

    For i = lastRow To 2 Step -1
        If WorksheetFunction.CountA(ws.Rows(i)) = 0 Then ws.Rows(i).Delete
    Next i

    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
    For i = lastRow To 3 Step -1
        curName = CleanText(ws.Cells(i, 1).Value)
        prevName = CleanText(ws.Cells(i - 1, 1).Value)
        If curName <> "" And prevName <> "" Then
            If StrComp(curName, prevName, vbTextCompare) <> 0 Then
                ws.Rows(i).Insert
            End If
        End If
    Next i

```

Figure 36 – Suite du module ScenarioRunner

```
Application.EnableEvents = True
Application.ScreenUpdating = True
MsgBox "Scenarios espacés correctement (1 ligne entre blocs).", vbInformation
End Sub

Private Function CleanText(ByVal s As String) As String
    Dim t As String
    t = Replace(CStr(s), Chr(160), " ")
    t = Application.WorksheetFunction.Clean(t)
    CleanText = Trim$(t)
End Function

Public Function GetUniqueScenarioNames_Sheet() As Collection
    Dim ws As Worksheet, lastRow As Long, c As New Collection, i As Long, nm As String
    Set ws = ThisWorkbook.Sheets("Scenarios")
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
    On Error Resume Next
    For i = 2 To lastRow
        nm = Trim$(CStr(ws.Cells(i, 1).Value))
        If Len(nm) > 0 Then c.Add nm, nm
    Next i
    On Error GoTo 0
    Set GetUniqueScenarioNames_Sheet = c
End Function
```

Figure 37 – Suite du module ScenarioRunner

```

mod_Visualization :

Sub FormatPnLColumn()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim i As Long

    Set ws = ThisWorkbook.Sheets("VaR_ES_Output")
    lastRow = ws.Cells(ws.Rows.Count, "D").End(xlUp).Row

    With ws
        For i = 2 To lastRow
            If IsNumeric(.Cells(i, 4).Value) Then
                .Cells(i, 4).NumberFormat = "$#,##0.00"
            End If
        Next i
    End With
End Sub

Sub FormatPnLPercent()
    Dim ws As Worksheet
    Dim lastRow As Long

    Set ws = ThisWorkbook.Sheets("VaR_ES_Output")
    lastRow = ws.Cells(ws.Rows.Count, "E").End(xlUp).Row

    With ws.Range("E2:E" & lastRow)
        .NumberFormat = "0.00%"
    End With
End Sub

Sub ApplyPnLColors()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim rng As Range

    Set ws = ThisWorkbook.Sheets("VaR_ES_Output")
    lastRow = ws.Cells(ws.Rows.Count, "D").End(xlUp).Row
    Set rng = ws.Range("D2:D" & lastRow)

    rng.FormatConditions.Delete

    With rng.FormatConditions.AddColorScale(ColorScaleType:=3)
        With .ColorScaleCriteria(1)
            .Type = xlConditionValueLowestValue
            .FormatColor.Color = RGB(255, 0, 0)
        End With
        With .ColorScaleCriteria(2)
            .Type = xlConditionValuePercentile
            .Value = 50
            .FormatColor.Color = RGB(255, 255, 255)
        End With
        With .ColorScaleCriteria(3)
            .Type = xlConditionValueHighestValue
            .FormatColor.Color = RGB(0, 176, 80)
        End With
    End With
End Sub

Sub StyleTotalRow()
    Dim ws As Worksheet
    Dim lastRow As Long

    Set ws = ThisWorkbook.Sheets("VaR_ES_Output")
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

```

Figure 38 – Module de visualisation : Mise en forme et stylisation des résultats du stress test.

```

Sub StyleTotalRow()
    Dim ws As Worksheet
    Dim lastRow As Long

    Set ws = ThisWorkbook.Sheets("VaR_ES_Output")
    lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

    With ws.Rows(lastRow)
        .Font.Bold = True
        .Interior.Color = RGB(230, 230, 230)
    End With
End Sub

Sub CleanUpVaRSheet()
    Call FormatPnLColumn
    Call FormatPnLPercent
    Call ApplyPnLColors
    Call StyleTotalRow
End Sub

Public Sub ColorReturnsColumn()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim cell As Range

    Set ws = ThisWorkbook.Sheets("Market_Data")

    lastRow = ws.Cells(ws.Rows.Count, "F").End(xlUp).Row

    For Each cell In ws.Range("F2:F" & lastRow)
        If IsNumeric(cell.Value) Then
            If cell.Value > 0 Then
                cell.Font.Color = RGB(0, 200, 0)
            ElseIf cell.Value < 0 Then
                cell.Font.Color = RGB(255, 50, 50)
            Else
                cell.Font.Color = RGB(255, 255, 255)
            End If
        Else
            cell.Font.Color = RGB(255, 255, 255)
        End If
    Next cell
End Sub

```

Figure 39 – Suite du module visualisation

```

mod_ExportPDF :

Option Explicit

Public Sub ExportKeySheetsToPDF()
    Dim wanted As Variant
    wanted = Array("Scenarios", "Portfolio", "VaR_ES_Output", "Market_Data")

    Dim present As Collection, nm As Variant, ws As Worksheet
    Set present = New Collection

    On Error Resume Next
    For Each nm In wanted
        Set ws = Nothing
        Set ws = ThisWorkbook.Worksheets(CStr(nm))
        If Not ws Is Nothing Then
            present.Add ws, ws.Name

            With ws.PageSetup
                .Orientation = xlLandscape
                .Zoom = False
                .FitToPagesWide = 1
                .FitToPagesTall = False
                .CenterHorizontally = True
                .CenterVertically = False
                .LeftMargin = Application.CentimetersToPoints(1.2)
                .RightMargin = Application.CentimetersToPoints(1.2)
                .TopMargin = Application.CentimetersToPoints(1.5)
                .BottomMargin = Application.CentimetersToPoints(1.5)
                .HeaderMargin = Application.CentimetersToPoints(0.8)
                .FooterMargin = Application.CentimetersToPoints(0.8)
            End With
        End If
    Next nm
    On Error GoTo 0

    If present.Count = 0 Then
        MsgBox "Aucune des feuilles attendues n'a été trouvée.", vbExclamation
        Exit Sub
    End If

    Dim defName As String, target As Variant
    defName = "RiskReport_" & Format(Now, "yyyymmdd_HHMM") & ".pdf"
    target = Application.GetSaveAsFilename(InitialFileName:=defName, _
        FileFilter:="PDF Files (*.pdf), *.pdf")
    If target = False Then Exit Sub

    Dim names() As String, i As Long
    ReDim names(1 To present.Count)
    For i = 1 To present.Count
        names(i) = present(i).Name
    Next i

    Dim prev As Worksheet
    Set prev = ActiveSheet

    Application.ScreenUpdating = False
    ThisWorkbook.Worksheets(names).Select
    ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, _
        FileName:=CStr(target), Quality:=xlQualityStandard, _
        IncludeDocProperties:=True, IgnorePrintAreas:=False, _
        OpenAfterPublish:=True
    prev.Select
    Application.ScreenUpdating = True

    If present.Count < UBound(wanted) + 1 Then
        MsgBox "Export PDF terminé. Certaines feuilles étaient manquantes et ont été ignorées.", vbInformation
    End If
End Sub

```

Figure 40 – Module d'exportation PDF : Procédure d'export automatique du rapport complet au format PDF.

```

frmAddPortfolio :

Option Explicit

Private Sub UserForm_Initialize()
    Dim ws As Worksheet, lastRow As Long, i As Long, v
    Set ws = ThisWorkbook.Sheets("Market_Data")

    cmbAsset.Clear
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    For i = 2 To lastRow
        On Error Resume Next
        v = ws.Cells(i, 1).Text
        On Error GoTo 0
        If Len(Trim$(v)) > 0 And v <> "Error 2015" Then
            cmbAsset.AddItem v
        End If
    Next i

    txtExposure.Value = ""
End Sub


Private Sub btnAdd_Click()
    Dim wsP As Worksheet
    Dim ticker As String, exposure As Double
    Dim assetType As String, beta As Double
    Dim lastRow As Long, i As Long
    Dim found As Boolean

    Set wsP = ThisWorkbook.Sheets("Portfolio")

    ticker = Trim$(cmbAsset.Value)
    If Len(ticker) = 0 Then
        MsgBox "Choisis un ticker.", vbExclamation
        Exit Sub
    End If

    If Not IsNumeric(txtExposure.Value) Then
        MsgBox "Exposition ($) invalide.", vbExclamation
        Exit Sub
    End If

    exposure = CDbl(txtExposure.Value)
    If exposure <= 0 Then
        MsgBox "L'exposition doit être supérieure à 0.", vbExclamation
        Exit Sub
    End If

    assetType = GetAssetType(ticker)
    beta = GetBeta(ticker)

    lastRow = wsP.Cells(wsP.Rows.Count, 1).End(xlUp).Row
    found = False

    If LCase(Trim(wsP.Cells(lastRow, 1).Value)) = "total" Then
        wsP.Rows(lastRow).Delete
        lastRow = lastRow - 1
    End If
End Sub

```

Figure 41 – Code VBA du formulaire **frmAddPortfolio**.

```

lastRow = wsP.Cells(wsP.Rows.Count, 1).End(xlUp).Row
found = False

If LCase(Trim(wsP.Cells(lastRow, 1).Value)) = "total" Then
    wsP.Rows(lastRow).Delete
    lastRow = lastRow - 1
End If

For i = 2 To lastRow
    If StrComp(Trim$(wsP.Cells(i, 1).Value), ticker, vbTextCompare) = 0 Then

        wsP.Cells(i, 3).Value = wsP.Cells(i, 3).Value + exposure
        wsP.Cells(i, 2).Value = assetType
        wsP.Cells(i, 4).Value = beta
        found = True
        Exit For
    End If
Next i

If Not found Then
    lastRow = wsP.Cells(wsP.Rows.Count, 1).End(xlUp).Row + 1
    wsP.Cells(lastRow, 1).Value = ticker
    wsP.Cells(lastRow, 2).Value = assetType
    wsP.Cells(lastRow, 3).Value = exposure
    wsP.Cells(lastRow, 4).Value = beta
End If

wsP.Columns("C:D").NumberFormat = "#,##0.00"
wsP.Columns("A:D").AutoFit

If found Then
    MsgBox "Exposition mise à jour pour : " & ticker & vbCrLf & _
           "Nouvelle exposition totale : " & Format(wsP.Cells(i, 3).Value, "#,##0.00 $"), vbInformation
Else
    MsgBox "Ajouté : " & ticker & " (" & assetType & ")" & vbCrLf & _
           "Exposition : " & Format(exposure, "#,##0.00 $") & vbCrLf & _
           "Beta : " & Format(beta, "0.00"), vbInformation
End If

cmbAsset.ListIndex = -1
txtExposure.Value = ""

Call UpdateTotalRow

Call UpdatePortfolioPieChart
End Sub

Private Sub btnClose_Click()
    Unload Me
End Sub

```

Figure 42 – Suite du formulaire add to portfolio

frmRemovePortfolio

```

frmRemovePortfolio :

Option Explicit

Private Sub Label1_Click()
End Sub

Private Sub lblTitle_Click()
End Sub

Private Sub UserForm_Initialize()
    Dim ws As Worksheet, lastRow As Long, i As Long
    Dim dict As Object, v As Variant
    Set dict = CreateObject("Scripting.Dictionary")

    Set ws = ThisWorkbook.Sheets("Portfolio")
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    For i = 2 To lastRow
        v = ws.Cells(i, 1).Value
        If Not IsError(v) Then
            v = Trim$(CStr(v))
            If Len(v) > 0 And LCase(v) <> "total" Then dict(v) = 1
        End If
    Next i

    Me.cmbTicker.Clear
    If dict.Count > 0 Then Me.cmbTicker.List = dict.Keys

    Me.txtAmount.Value = ""
End Sub

Private Sub btnRemove_Click()
    Dim ws As Worksheet
    Dim lastRow As Long, i As Long
    Dim key As String
    Dim amtTxt As String
    Dim toRemove As Double
    Dim removedRows As Long
    Dim changedRows As Long

    key = Trim$(Me.cmbTicker.Value)
    If Len(key) = 0 Then
        MsgBox "Please select a ticker.", vbExclamation
        Exit Sub
    End If

    amtTxt = Trim$(Me.txtAmount.Value)

    If Len(amtTxt) > 0 Then
        If Not IsNumeric(amtTxt) Then
            MsgBox "Enter a valid amount to remove (numeric).", vbExclamation
            Exit Sub
        End If
        toRemove = CDbl(amtTxt)
        If toRemove <= 0 Then
            MsgBox "Amount must be > 0.", vbExclamation
            Exit Sub
        End If
    Else
        toRemove = 0
    End If

    Set ws = ThisWorkbook.Sheets("Portfolio")
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

```

Figure 43 – Formulaire frmRemovePortfolio — Partie 1 : initialisation et sélection de l'actif à modifier.

```

        ws.Rows(i).Delete
        removedRows = removedRows + 1
    End If
End If
Next i

Application.ScreenUpdating = True

If removedRows > 0 Then
    MsgBox removedRows & " row(s) removed for '" & key & "'.", vbInformation
Else
    MsgBox "No matching row found for: " & key, vbExclamation
End If

Else

Dim leftToRemove As Double: leftToRemove = toRemove
Dim expVal As Double, ok As Boolean

For i = lastRow To 2 Step -1
    If leftToRemove <= 0 Then Exit For

    If Not IsError(ws.Cells(i, 1).Value) Then
        If StrComp(Trim$(CStr(ws.Cells(i, 1).Value)), key, vbTextCompare) = 0 Then

            ok = False
            On Error Resume Next
            If IsNumeric(ws.Cells(i, 3).Value) Then
                expVal = CDbl(ws.Cells(i, 3).Value)
                ok = True
            End If
            On Error GoTo 0
            If Not ok Then GoTo NextI

            If expVal > leftToRemove Then

                ws.Cells(i, 3).Value = expVal - leftToRemove
                changedRows = changedRows + 1
                leftToRemove = 0
            Else

                leftToRemove = leftToRemove - expVal
                ws.Rows(i).Delete
                removedRows = removedRows + 1
            End If
        End If
    End If
NextI:
    Next i

Application.ScreenUpdating = True

If removedRows + changedRows = 0 Then
    MsgBox "No matching row found for: " & key, vbExclamation
Else
    Dim msg As String
    msg = "Ticker: " & key & vbCrLf & _
        "Rows changed: " & changedRows & vbCrLf & _
        "Rows removed: " & removedRows
    If leftToRemove > 0 Then
        msg = msg & vbCrLf & "Amount still not removed: " & Format(leftToRemove, "#,##0")
    End If
    MsgBox msg, vbInformation
End If

lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
If LCase(Trim$(ws.Cells(lastRow, 1).Value)) = "total" Then
    ws.Rows(lastRow).Delete
End If

Call UpdateTotalRow

Call UpdatePortfolioPieChart

Call UserForm_Initialize

Application.ScreenUpdating = True
End Sub

Private Sub btnClose_Click()
    Unload Me
End Sub

```

Figure 44— Formulaire frmRemovePortfolio — Partie 2 : suppression totale ou réduction partielle d'exposition.

frmAddScenario

```

frmAddScenario :
Option Explicit

Private Sub cmdAddShock_Click()
    Dim assetType As String
    Dim shockValue As String
    Dim displayText As String

    assetType = Trim(Me.cmbAsset.Value)
    shockValue = Trim(Me.txtShock.Value)

    If assetType = "" Then
        MsgBox "Please select an asset type.", vbExclamation
        Exit Sub
    End If

    If shockValue = "" Or Not IsNumeric(shockValue) Then
        MsgBox "Please enter a valid numeric shock value.", vbExclamation
        Exit Sub
    End If

    displayText = assetType & " | " & shockValue
    Me.lstShocks.AddItem displayText

    Me.cmbAsset.Value = ""
    Me.txtShock.Value = "0"
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdSaveScenario_Click()
    Dim scenarioName As String
    Dim i As Long
    Dim nextRow As Long
    Dim parts() As String
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim lastScenario As String

    scenarioName = Trim(Me.txtName.Value)

    If scenarioName = "" Then
        MsgBox "Please enter a scenario name.", vbExclamation
        Exit Sub
    End If

    If Me.lstShocks.ListCount = 0 Then
        MsgBox "Please add at least one shock.", vbExclamation
        Exit Sub
    End If

    Set ws = ThisWorkbook.Sheets("Scenarios")

    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    If lastRow >= 2 Then
        lastScenario = Trim(ws.Cells(lastRow, 1).Value)

        If lastScenario <> "" And StrComp(lastScenario, scenarioName, vbTextCompare) <> 0 Then
            ws.Rows(lastRow + 1).Insert
            lastRow = lastRow + 1
        End If
    End If

    nextRow = lastRow + 1

```

Figure 45 – Formulaire frmAddScenario — Partie 1 : structure du formulaire et ajout de chocs multiples.

```

Private Sub cmdSaveScenario_Click()
    Dim scenarioName As String
    Dim i As Long
    Dim nextRow As Long
    Dim parts() As String
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim lastScenario As String

    scenarioName = Trim(Me.txtName.Value)

    If scenarioName = "" Then
        MsgBox "Please enter a scenario name.", vbExclamation
        Exit Sub
    End If

    If Me.lstShocks.ListCount = 0 Then
        MsgBox "Please add at least one shock.", vbExclamation
        Exit Sub
    End If

    Set ws = ThisWorkbook.Sheets("Scenarios")

    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    If lastRow >= 2 Then
        lastScenario = Trim(ws.Cells(lastRow, 1).Value)

        If lastScenario <> "" And StrComp(lastScenario, scenarioName, vbTextCompare) <> 0 Then
            ws.Rows(lastRow + 1).Insert
            lastRow = lastRow + 1
        End If
    End If

    nextRow = lastRow + 1

    For i = 0 To Me.lstShocks.ListCount - 1
        parts = Split(Me.lstShocks.List(i), " | ")

        ws.Cells(nextRow, 1).Value = scenarioName
        ws.Cells(nextRow, 2).Value = parts(0)
        ws.Cells(nextRow, 3).Value = CDbl(parts(1))
        ws.Cells(nextRow, 4).Value = Now

        nextRow = nextRow + 1
    Next i

    MsgBox "Scenario '" & scenarioName & "' saved successfully!" & vbCrLf &
           "You can now run it manually using the 'Stress Test' button.", vbInformation

    Unload Me
End Sub

Private Sub UserForm_Initialize()
    Me.cmbAsset.Clear
    Me.cmbAsset.List = Array("Equity", "Rates", "Oil", "Commodities", "Bond")
    Me.txtShock.Value = "0"
End Sub

Sub ShowAddScenarioForm()
    frmAddScenario.Show
End Sub

```

Figure 46 – Formulaire frmAddScenario — Partie 2 : validation, enregistrement et rafraîchissement de la feuille Scenarios.

frmRemoveScenario

```

frmRemoveScenario :
Option Explicit

Private Sub UserForm_Initialize()
    Dim ws As Worksheet, lastRow As Long, r As Long
    Dim dict As Object, v As Variant

    Set dict = CreateObject("Scripting.Dictionary")
    Set ws = ThisWorkbook.Sheets("Scenarios")

    If ws Is Nothing Then
        MsgBox "Sheet 'Scenarios' not found.", vbExclamation
        Unload Me
        Exit Sub
    End If

    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
    For r = 2 To lastRow
        v = ws.Cells(r, 1).Value
        If Not IsError(v) Then
            v = Trim$(CStr(v))
            If Len(v) > 0 Then dict(v) = 1
        End If
    Next r

    cmbScenario.Clear
    If dict.Count > 0 Then cmbScenario.List = dict.Keys
End Sub

Private Sub btnDelete_Click()
    Dim ws As Worksheet
    Dim lastRow As Long, r As Long
    Dim nameToKill As String
    Dim deleted As Long
    Dim resp As VbMsgBoxResult

    nameToKill = Trim$(cmbScenario.Value)
    If Len(nameToKill) = 0 Then
        MsgBox "Please select a scenario name.", vbExclamation
        Exit Sub
    End If

    resp = MsgBox("Delete ALL rows for scenario:" & vbCrLf &
                 "<< " & nameToKill & " >> ?" & vbCrLf & vbCrLf & _
                 "This action cannot be undone.", _
                 vbQuestion + vbYesNo + vbDefaultButton2, "Confirm deletion")
    If resp <> vbYes Then Exit Sub

    Set ws = ThisWorkbook.Sheets("Scenarios")
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    Application.ScreenUpdating = False
    For r = lastRow To 2 Step -1
        If Not IsError(ws.Cells(r, 1).Value) Then
            If StrComp(Trim$(CStr(ws.Cells(r, 1).Value)), nameToKill, vbTextCompare) = 0 Then
                ws.Rows(r).Delete
                deleted = deleted + 1
            End If
        End If
    Next r
    Application.ScreenUpdating = True

    If deleted > 0 Then
        MsgBox deleted & " row(s) deleted for scenario << " & nameToKill & " >>.", vbInformation
    Else
        MsgBox "No rows found for scenario << " & nameToKill & " >>.", vbExclamation
    End If

    Call UserForm_Initialize
End Sub

Private Sub btnClose_Click()
    Unload Me
End Sub

```

Figure 47— Formulaire frmRemoveScenario — Suppression d'un scénario sélectionné et mise à jour automatique de la feuille Scenarios.

Conclusion

Nous avons choisi ce projet avant tout pour son lien direct avec la réalité des métiers de la finance de marché. L'objectif n'était pas de réaliser un projet académique classique, mais de concevoir un outil que nous pourrions réellement présenter en entretien, face à des traders ou à des portfolio managers, comme un exemple concret de notre compréhension des mécanismes de risque et des stress tests.

Au-delà de l'aspect technique, nous nous sommes véritablement amusés à développer cet outil. La combinaison entre programmation VBA, logique de marché et visualisation des impacts financiers a rendu le projet à la fois stimulant et formateur.

Ce travail nous a permis de manipuler des notions essentielles en finance, exposition, bêta, stress tests, scénarios macroéconomiques, tout en renforçant notre capacité à traduire des concepts théoriques en solutions opérationnelles. Il illustre notre volonté de relier nos connaissances quantitatives à des applications concrètes, comme le feraient des professionnels du trading ou de la gestion de portefeuille.