

Rapport de Projet de Fin de Module

Méthodes de prévision



Réalisé Par :

-  M.ABDELLI Mohamed Saad
-  M.BEL FATMI ALAE
-  ELKHAZNAJI Ahmed
-  LAAMRANI Doae
-  ELKASSAB Oumaima

Demandé par :

-  M. EL MEROUANI Mohamed

Objectif :

L'objectif de ce projet est de faire nos premiers pas sur le logiciel SAS « *Statistical Analysis System* », se familiariser avec ce dernier, bien comprendre ses généralités et vers la fin essayer de choisir une série temporelle et appliquer les exemples du cours sur cette série, dans notre nous avons utilisé le modèle ARIMA.

Table de matière :

Introduction.....	5
SAS « Statistical Analysis System »	6
1. Description.....	6
2. Un programme SAS.....	6
3. Les instructions de bas SAS.....	7
Installation du SAS University Edition	7
Les manipulations sur SAS.....	10
Réalisation.....	16
1. La procédure ARIMA.....	16
2. Les trois étapes de modélisation ARIMA.....	16
3. Description de procédure.....	17
4. La stationnarité.....	18
5. Réalisation.....	19
Conclusion.....	26

Table des figures

Figure 1 : Interface d'accueil du SAS.....	6
Figure 2 : Démarrage de la machine virtuelle SAS.....	9
Figure 3 : Ouverture du SAS depuis le navigateur google chrome.....	10
Figure 4 : Importation des données.....	12
Figure 5 : Impression des données.....	13
Figure6 : PROC MEANS.....	14
Figure 7 : PROC PLOT.....	15
Figure 8 : Les variables différentielles.....	20
Figure 9 : la syntaxe du code ARIMA (cas normal et le cas de paramètre de bruit).....	21
Figure 10 : les résultats de la première procédure ARIMA.....	21
Figure 11 : les résultats de la deuxième procédure ARIMA.....	22
Figure 12 : AR(1).....	23
Figure 13 : MA(1).....	23
Figure 14 : Le test de régression (modèle1).....	24
Figure 15 : Le test de régression (modèle2).....	25

I. Introduction :

« Je ne crois aux statistiques que lorsque je les ai moi-même falsifiées ».

Il suffit d'allumer son ordinateur ou d'écouter les informations à la radio pour constater que les statistiques sont partout. Ceci révèle que le monde moderne est presque entièrement tourné vers le quantitatif et le mesurable. D'où l'intérêt de la statistique, discipline relativement récente, mais qui correspond parfaitement à cette orientation du monde moderne.

Le but de la statistique est de dégager les significations de données, numériques ou non, obtenues au cours de l'étude d'un phénomène.

Le Sas est un logiciel complet qui couvre une large gamme des méthodes d'analyse en Statistique, de conception américaine, il est développé par la société SAS Institute Inc.

Le système SAS se présente sous la forme d'un ensemble de modules logiciels adaptés pour la gestion et l'analyse statistique de gros volumes de données (tableaux de plusieurs giga-octets) et la création de rapports de synthèse.

Dans ce rapport nous allons vous présenter dans un premier lieu comment installer le SAS Studio, nos premières essais de manipulation sur le SAS, définir le modèle ARIMA et finalement appliquer ce dernier sur notre série temporelle et essayer de commenter les résultats.

II. SAS « *Statistical Analysis System* » :

1. Description :

L'objectif principal de ce fascicule est l'initiation à l'utilisation du logiciel SAS (Statistical Analysis System). Le système SAS offre un large éventail de traitements séquentiel de données permettant d'assurer pratiquement toutes ou presque toutes les fonctions de gestion et d'analyse des données à l'aide des nombreuses procédures statistiques qu'il propose. La documentation importante de ce logiciel traduit sa richesse mais aussi sa complexité qui peut toutefois s'avérer un inconvénient pour certains utilisateurs. Comme il est visiblement impossible de couvrir SAS en entier, ce guide se veut donc qu'un résumé des aspects de base les plus intéressants pour être en mesure de comprendre comment élaborer un programme SAS et d'être suffisamment à l'aise avec certains mots-clés, certaines commandes nécessaires car beaucoup de programmes SAS se ressemblent et ne diffèrent que par les données utilisées et par les traitements effectués. Il existe donc des éléments communs à l'ensemble des programmes. Ce sont ces éléments qui seront d'abord présentés.

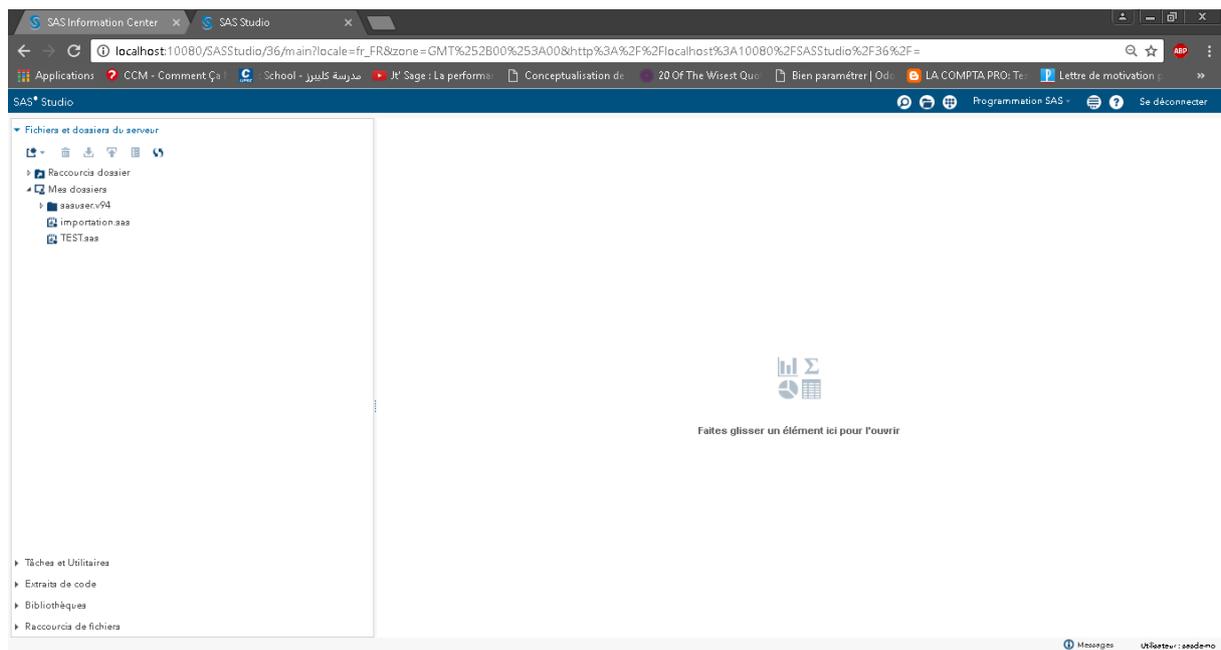


Figure 1 : Interface d'accueil du SAS.

2. Programme SAS :

Un programme SAS se décompose en deux étapes (types) d'instructions :

- ✓ Un (ou des) bloc(s) **DATA** pour les données : c'est un regroupement d'instructions SAS permettant de créer une, voire plusieurs tables SAS à partir de données se trouvant dans

des fichiers ASCII ou des tables SAS créées auparavant. C'est à l'intérieur d'un bloc DATA que les transformations à apporter aux données sont effectuées.

- ✓ Un (ou des) bloc(s) PROC pour les procédures : permet l'analyse des données à l'aide d'une procédure spécifique. Les différentes procédures d'un programme SAS seront détaillées par la suite.

Ces deux parties sont généralement indépendantes, il est donc inutile de ré-exécuter l'instruction **data**, dont les informations des données sont déjà stockées dans une table SAS (permanente ou temporaire), lors d'un enchaînement ou de la mise au point de procédures.

3. Instructions de base d'un programme SAS :

Il est important de se familiariser rapidement avec certains **mots-clés** et certaines **commandes**, nécessaires à l'élaboration d'un programme SAS, que l'on retrouve dans de nombreux programmes qui ne diffèrent que par les données utilisées et/ou par les traitements effectués. Il existe donc des instructions communes à l'ensemble des programmes SAS.

SAS étant principalement un logiciel de programmation. L'étape DATA et les procédures sont les deux familles principales d'instructions d'un programme SAS.

- ✓ L'étape **DATA** regroupe un ensemble d'instructions de programmation commençant par une ligne **DATA** et se terminant par l'instruction **RUN** ; qui clôt cette étape. Entre ces deux mot-clés, on utilisera des instructions pour la lecture d'une ou de plusieurs tables, d'un fichier externe, le calcul de nouvelles variables ou la modification de variables existantes.
- ✓ Les procédures sont des programmes déjà écrits de SAS, à la syntaxe beaucoup plus rigoureuse que celle de l'étape DATA, où il faut surtout renseigner un certain nombre de paramètres avec/sans d'options, qui accomplissent chacune une tâche spécialisée. Chaque procédure commence par une instruction **PROC** suivie de son nom, et s'achève sur une instruction **RUN** ; qui demande l'exécution de cette procédure.

III. Installation de SAS Studio University Edition:

1. Installation Oracle VirtualBox :

- Installez la dernière version d'Oracle VirtualBox à l'aide du lien fourni par votre administrateur de site ou consultez (<https://www.virtualbox.org/wiki/Downloads>).

- Voir la page de téléchargement de SAS University Edition (au [http://www.sas.com/en_us/software/university-Edition / download.html](http://www.sas.com/en_us/software/university-Edition/download.html)) Pour obtenir le vApp SAS University Edition.
- Lors du téléchargement de l'Université SAS Edition vApp, votre navigateur peut vous demander d'enregistrer ou d'exécuter le fichier.
- Cliquez sur **Enregistrer** pour enregistrer ce fichier dans votre répertoire de téléchargements.

2. Ajoutez le vApp SAS University Edition à VirtualBox:

- Lancez VirtualBox, puis **sélectionnez Fichier> Importer l'appliance**.
- Dans le répertoire Téléchargements, sélectionnez le fichier pour le vApp SAS University Edition (un fichier OVA) et Puis cliquez sur **Ouvrir**.
- Cliquez sur **Suivant** et puis cliquez sur **Importer**.

3. Créez un dossier pour vos données et vos résultats:

- Sur votre ordinateur local (dans un emplacement dont vous vous souviendrez), créez un dossier appelé SAS University Edition et un sous-dossier appelé myfolders .
- Vous allez enregistrer tous vos SAS University Edition à cet emplacement.
- Dans VirtualBox, sélectionnez le vApp SAS University Edition, puis sélectionnez Machine> Paramètres.
- Dans le volet de navigation de la boîte de dialogue Paramètres, sélectionnez Dossiers partagés,
- Puis cliquez sur OK.
- Dans la boîte de dialogue Ajouter un partage, sélectionnez Autre comme chemin du dossier.
- Dans la fenêtre Parcourir le dossier, ouvrez le dossier SAS University Edition et Myfolders sous dossier.
- Cliquez sur **OK** (ou choisissez, selon votre système d'exploitation).
- Dans la boîte de dialogue Ajouter un partage, vérifiez que **Lecture seule** n'est pas sélectionnée, puis sélectionnez l'option **Montage automatique Et faire des options permanentes** (si disponibles). Cliquez sur **OK**.
- Cliquez sur **OK** pour fermer la boîte de dialogue Paramètres.

4. Démarrez le vApp SAS University Edition:

- Dans VirtualBox, sélectionnez l'option SAS Enterprise Edition vApp, puis sélectionnez Machine> Démarrer. Il pourrait prendre quelques Minutes pour que la machine virtuelle démarre.
- Remarque: Lorsque la machine virtuelle est en cours d'exécution, l'écran avec le logo SAS est remplacé par une console noire (Appelé fenêtre de bienvenue). Vous pouvez réduire cette fenêtre, mais ne fermez pas le Jusqu'à ce que vous soyez prêt à terminer votre session SAS.

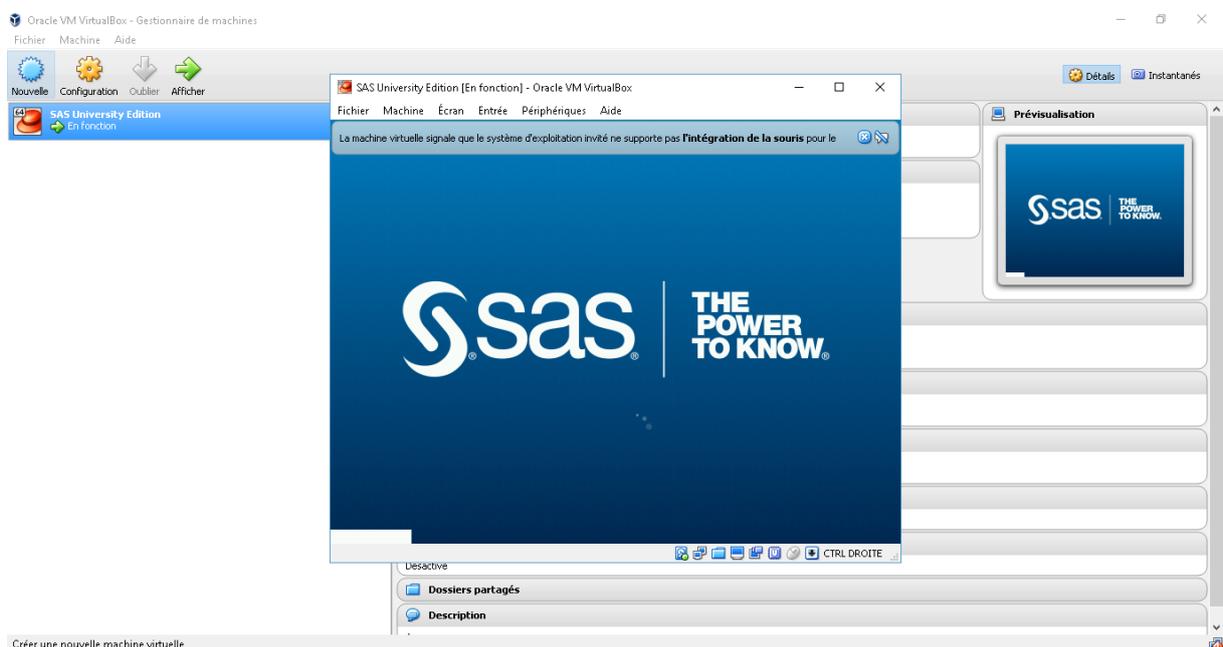


Figure 2 : Démarrage de la machine virtuelle SAS .

5. Ouvrez SAS University Edition:

- Dans un navigateur Web sur votre ordinateur local, entrez <http://localhost:10080>.
- À partir de SAS University Edition: Centre d'information, cliquez sur Démarrer SAS Studio.

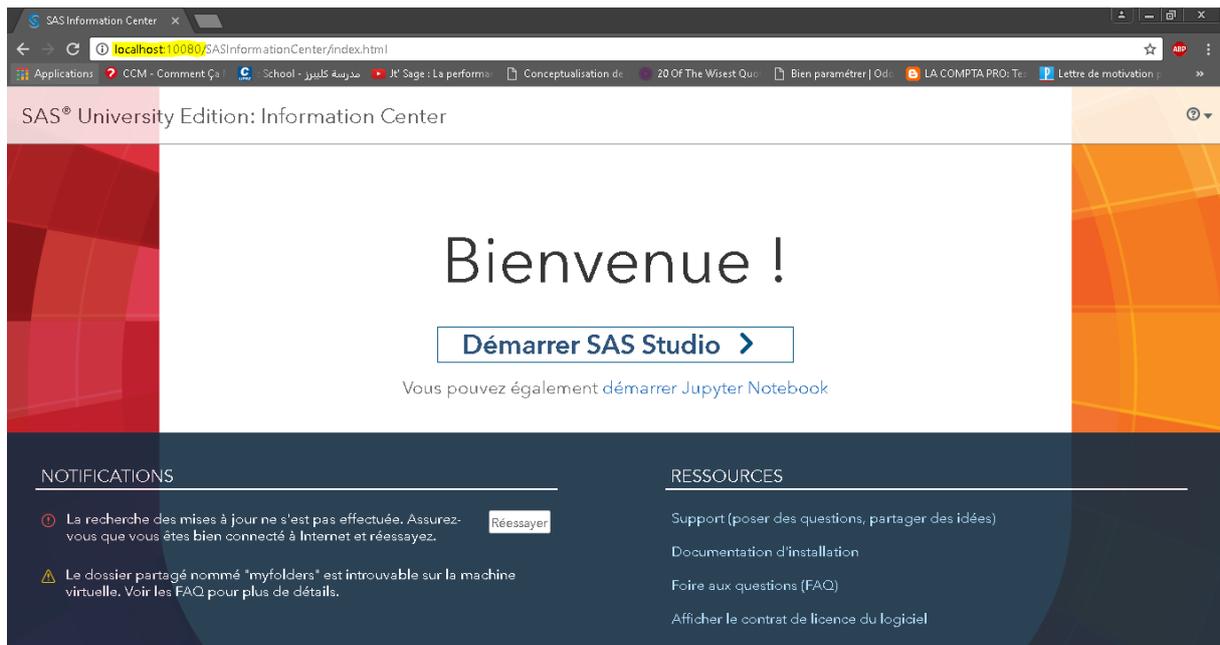


Figure 3 : Ouverture du SAS depuis le navigateur google chrome.

IV. Les manipulations sur SAS University Edition :

1. Etape DATA :

Nous appellerons TABLE SAS un tel fichier écrit dans ce format où les colonnes représentent les variables et les lignes les valeurs observées des variables. Les tables SAS peuvent être temporaires ou permanentes. Une séquence de lecture des données se présente de la façon suivante :

```
DATA < nom de la table SAS >;  
  INPUT < liste des variables >;  
  CARDS;  
  < les données sont entrées "à la main" >  
  ;  
RUN;
```

L'instruction CARDS – équivalente à DATALINES et à LINES – indique le début de la saisie des observations (une seule observation par ligne).

Les données sont alors séparées par des espaces, une valeur manquante étant représentée par un point. Notons que, dans la liste des variables, le séparateur est encore un blanc. Enfin, l'instruction RUN (facultative mais conseillée) termine l'étape DATA.

```

00001  /* exemple de programme; */           → ligne de commentaire
00002  DATA exemple;
00003  INPUT nom$ ddn DDMMYY10. CSP$ auto$ nbsin1 nbsin2;
00004  CARDS;
00005  Pierre 08/04/1944 lib C5 0 2
00006  Paul 12/03/1976 arti 307 0 0
00007  Jacques 09/10/1953 cadre C3 1 2
00008  Carole 11/12/1964 lib modus 1 0
00009  Caroline 21/08/1970 cadre polo 0 0
00010  Nathalie 14/06/1962 cadsup 607 3 4
00011  ;
00012  RUN;

```

2. Etape PROC :

Un programme SAS est composé d'une ou plusieurs étapes DATA (cf. séance n°1) suivies d'une ou plusieurs étapes PROC (ou procédures). Une étape PROC applique des traitements à des tables et génère des sorties, des graphiques ou encore de nouvelles tables.

□ Syntaxe:

Le début de l'étape est signalé par PROC suivi du nom de la procédure, puis du nom de la table à traiter. Il est important de spécifier la table sur laquelle on travaille avec l'instruction « DATA = < nom_tab > ». Par défaut, la procédure agit sur la dernière table créée. La fin de l'étape est reconnaissable à l'instruction RUN.

```

PROC < nom de la procédure > DATA = < nom de la Table SAS > < options >;
...
RUN;

```

Exemple : Importation de la série temporelle depuis le serveur (PROC IMPORT).

```

importation.sas x
CODE JOURNAL RESULTATS
Ligne
1 proc import datafile="/folders/myshortcuts/My_folders/timeseries_ppi.csv"
2 out=work.au_sales dbms=csv replace;
3 run;
4
5
/folders/myfolders/importation.sas UTF-8

```

importation.sas x

CODE JOURNAL RESULTATS **DONNÉES DE SORTIE**

Table: WORK.AU_SALES Afficher: Norme de colonnes Filtre: (néant)

Colonnes: Sélectionner tout

- yearqrt
- m3
- ppi
- cpi
- gdp
- m1nsa
- ddnaa
- t
- lnppi
- dppi

Propriété Valeur

Libellé

Nom

Longueur

Type

Format

Informat

Lignes totales : 169 Colonnes totales : 13 Lignes 1-100

	yearqrt	m3	ppi	cpi
1	01JAN60:00:00:00	692	25.4	19.280001
2	01APR60:00:00:00	701.59998	25.440001	19.4
3	01JUL60:00:00:00	709.5	25.370001	19.450001
4	01OCT60:00:00:00	722.09998	25.4	19.559999
5	01JAN61:00:00:00	726.20001	25.49	19.57
6	01APR61:00:00:00	735.29999	25.24	19.58
7	01JUL61:00:00:00	748.20001	25.24	19.68
8	01OCT61:00:00:00	764.70001	25.280001	19.700001
9	01JAN62:00:00:00	772.79999	25.42	19.74
10	01APR62:00:00:00	786.90002	25.280001	19.83
11	01JUL62:00:00:00	798.79999	25.41	19.91
12	01OCT62:00:00:00	817.29999	25.370001	19.959999
13	01JAN63:00:00:00	826.90002	25.280001	19.99
14	01APR63:00:00:00	841.29999	25.24	20.030001
15	01JUL63:00:00:00	852.5	25.34	20.18
16	01OCT63:00:00:00	872	25.360001	20.23
17	01JAN64:00:00:00	883.09998	25.389999	20.280001
18	01APR64:00:00:00	899.20001	25.27	20.32
19	01JUL64:00:00:00	913.40002	25.35	20.41

Messages: 1 Utilisateur: aedemo

Figure 4 : Importation des données.

Exemple : impression de la table exemple (PROC PRINT).

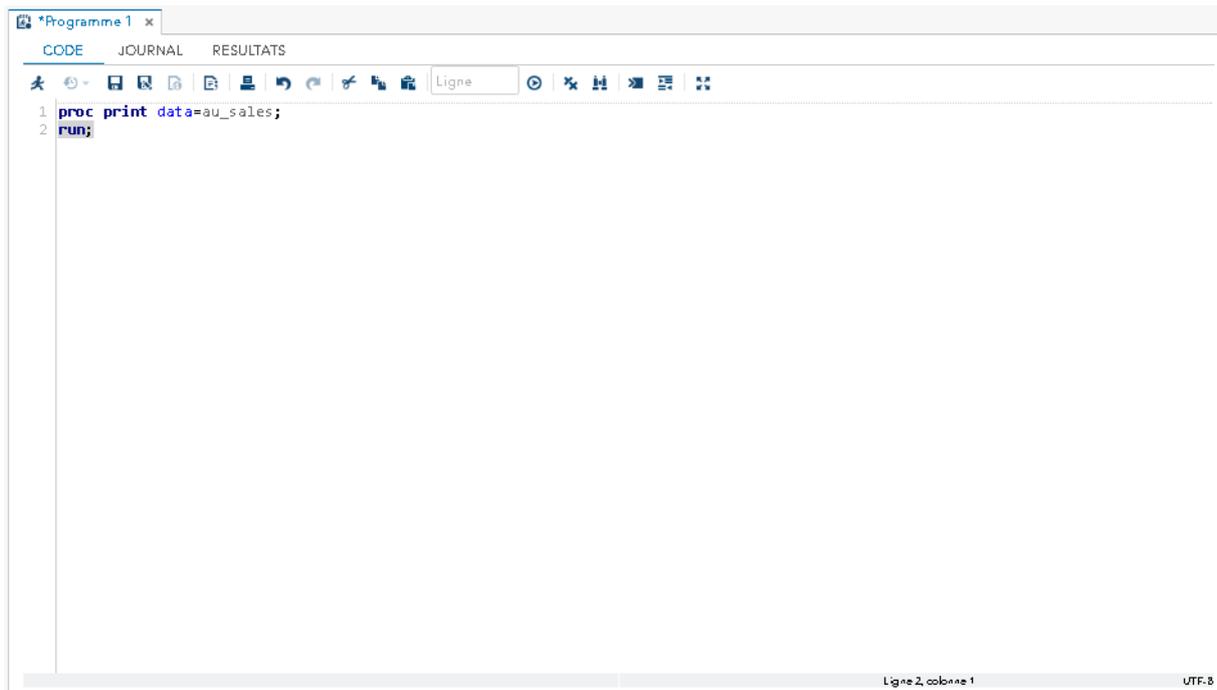


Table des matières

Obs.	yearqtr	m3	ppi	cpi	gdp	m1nsa	ddnsa	t	lnppi	dppi	dlnppi	lp
1	01JAN60:00:00:00	692	25.4	19.280001	2391	140.53	111.67	01APR60:00:00:00	3.2347491	.	.	.
2	01APR60:00:00:00	701.59998	25.440001	19.4	2379.2	138.42999	109.47	01JUL60:00:00:00	3.2363226	0.040000916	0.001573563	25.4400
3	01JUL60:00:00:00	709.5	25.370001	19.450001	2383.6001	139.60001	110.33	01OCT60:00:00:00	3.2335675	-0.069999695	-0.002755165	25.4400
4	01OCT60:00:00:00	722.09998	25.4	19.559999	2352.8999	142.67999	113.23	01JAN61:00:00:00	3.2347491	0.029998779	0.001181603	25.3700
5	01JAN61:00:00:00	726.20001	25.49	19.57	2366.5	142.22	113.4	01APR61:00:00:00	3.2382863	0.090000153	0.003537178	25.3700
6	01APR61:00:00:00	735.29999	25.24	19.58	2410.8	141.36	112.48	01JUL61:00:00:00	3.22843	-0.25	-0.009856224	25.3700
7	01JUL61:00:00:00	748.20001	25.24	19.68	2450.3999	142.00999	112.62	01OCT61:00:00:00	3.22843	0	0	25.3700
8	01OCT61:00:00:00	764.70001	25.280001	19.700001	2500.3999	146.62	116.71	01JAN62:00:00:00	3.2300136	0.040000916	0.001583576	25.3700
9	01JAN62:00:00:00	772.79999	25.42	19.74	2544	146.36	116.76	01APR62:00:00:00	3.2355363	0.13999939	0.005522728	25.2800
10	01APR62:00:00:00	786.90002	25.280001	19.83	2571.5	146.35001	115.38	01JUL62:00:00:00	3.2300136	-0.13999939	-0.005522728	25.2800
11	01JUL62:00:00:00	798.79999	25.41	19.91	2596.8	144.97	114.49	01OCT62:00:00:00	3.2351427	0.12999916	0.005129099	25.2800
12	01OCT62:00:00:00	817.29999	25.370001	19.959999	2603.3	149.23	118.29	01JAN63:00:00:00	3.2335675	-0.039999008	-0.001575232	25.2800
13	01JAN63:00:00:00	826.90002	25.280001	19.99	2634.1001	149.53	118.76	01APR63:00:00:00	3.2300136	-0.090000153	-0.003535867	25.3700
14	01APR63:00:00:00	841.29999	25.24	20.030001	2668.3999	148.85001	117.53	01JUL63:00:00:00	3.22843	-0.040000916	-0.001583576	25.2800
15	01JUL63:00:00:00	852.5	25.34	20.18	2719.6001	150.13	118.02	01OCT63:00:00:00	3.2323842	0.10000038	0.003954172	25.2800
16	01OCT63:00:00:00	872	25.360001	20.23	2739.3999	155.2	122.38	01JAN64:00:00:00	3.2331731	0.020000458	0.000788927	25.2800
17	01JAN64:00:00:00	883.09998	25.389999	20.280001	2800.5	154.95	122.28	01APR64:00:00:00	3.2343554	0.029998779	0.001182318	25.3800
18	01APR64:00:00:00	899.20001	25.27	20.32	2833.8	153.94	120.66	01JUL64:00:00:00	3.2296178	-0.11999893	-0.004737616	25.3899

Figure 5 : Impression des données.

Exemple : Vue de synthèse des variables en utilisant (PROC MEANS).

Cette procédure permet de donner tous les informations relatives aux variables spécifiques (la moyenne, l'écart-type, le minimum et le maximum) dans notre cas, nous intéressons sur les variables (ppi, dppi, et time).

```

12 * la premiere procedure MEANS donne une vision globale sur les variables de la table;
13 proc means data=au_sales;
14 var &ylist &dylist &time;
15 run;
16

```

La procédure MEANS

Variabale	N	Moyenne	Ec-type	Minimum	Maximum
ppi	169	64.6815385	30.2659545	25.2400000	110.4300000
dppi	168	0.4642857	0.9207450	-3.2100000	3.0800010
t	169	670564715	386030145	7862400.00	1333238400

Figure6 : PROC MEANS.

Quelques instructions relatives aux procédures :

- L'instruction **PROC** (abréviation de procédure) est utilisée pour appeler une procédure SAS.

```
PROC < nom_proc > DATA = < nom_tab > < options >;
```

- L'instruction **VAR** (abréviation de variables) est suivie de la liste des variables à considérer.

```
VAR < liste des variables >;
```

- L'instruction **BY** permet le traitement par sous-groupes.

```
BY < sous-groupe de variables >;
```

Remarque La table SAS traitée par l'instruction BY doit avoir été triée suivant les variables mentionnées dans l'instruction.

- L'instruction **TITLE** définit le texte à imprimer en en-tête des pages de sortie. On peut ajouter jusqu'à 10 lignes de titres.

```
TITLEN 'titre';    où le n spécifie le nième titre.
```

- L'instruction **FOOTNOTE** définit le texte à imprimer en bas des pages de sortie. On peut ajouter jusqu'à 10 lignes de bas de pages.

```
FOOTNOTEN 'bas de page';    où le n spécifie le nième bas de page.
```

Exemple : Dessiner les graphes en utilisant (PROC PLOT).

Cette procédure sert à dessiner un graphe d'une variable en fonction d'autre variable, dans notre cas nous avons dessiné deux graphes, un pour la variable (ppi) en fonction du temps (time) et le deuxième c'est la différence (dppi) en fonction du temps (time).

```

17 *la procedure PLOT pour tracer une graphe de deux variables;
18 proc plot data=au_sales;
19 plot &ylist*&time;
20 plot &dylist*&time;
21 run;
22

```

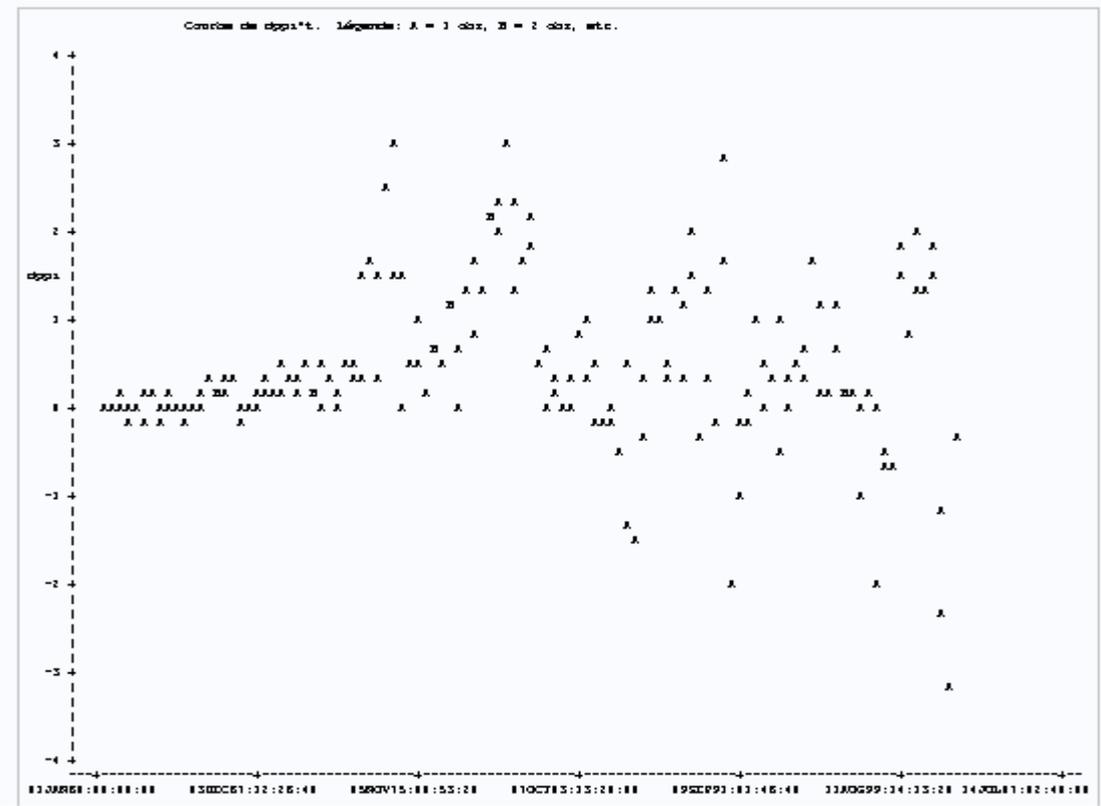
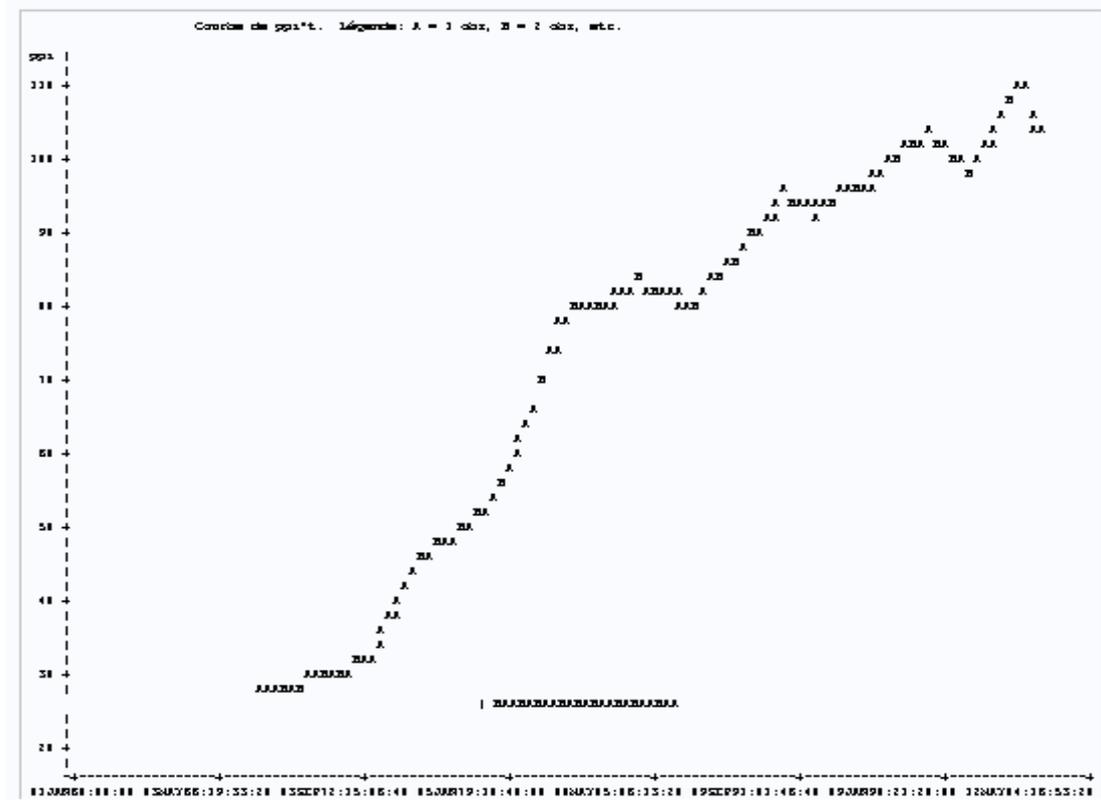


Figure 7 : PROC PLOT.

V. Réalisation :

1. La procédure ARIMA :

La procédure ARIMA analyse et prévoit des données de séries temporelles uni-variées à intervalle équitable, des données de fonctions de transfert et des données d'intervention en utilisant le modèle de la moyenne mobile autorégressive (ARIMA) ou de la moyenne mobile autorégressive (ARMA).

La procédure ARIMA fournit un ensemble complet d'outils pour l'identification uni-variée des modèles de séries chronologiques, l'estimation des paramètres et la prévision, et elle offre une grande souplesse dans les types de modèles ARIMA ou ARIMAX qui peuvent être analysés.

La procédure ARIMA supporte les modèles ARIMA saisonniers, sous-ensembles et factorisés; Modèles d'intervention ou de séries chronologiques interrompues; Analyse de régression multiple avec des erreurs ARMA; Et des modèles de fonction de transfert rationnels de toute complexité.

2. Les trois étapes de modélisation ARIMA :

L'analyse réalisée par PROC ARIMA est divisée en trois étapes, correspondant aux étapes décrites par Box et Jenkins (1976).

- **Identification:**

Dans l'étape d'identification, vous utilisez l'instruction IDENTIFY pour spécifier la série de réponses et identifier les modèles ARIMA candidats pour cela. L'instruction IDENTIFY lit les séries temporelles qui doivent être utilisées dans les instructions ultérieures, éventuellement les différenciant, et calcule les auto-corrélations, les auto-corrélations inverses, les auto-corrélations partielles et les corrélations croisées. Des tests de stationnarité peuvent être effectués pour déterminer si une différenciation est nécessaire. L'analyse de la sortie de l'instruction IDENTIFY suggère habituellement un ou plusieurs modèles ARIMA qui pourraient être adaptés. Les options vous permettent de tester la stationnarité et l'identification provisoire de l'ordre ARMA.

- **Estimation:**

Dans l'étape d'estimation et de vérification du diagnostic, vous utilisez l'instruction ESTIMATE pour spécifier le modèle ARIMA pour s'adapter à la variable spécifiée dans l'instruction IDENTIFY précédente et pour estimer les paramètres de ce modèle. La déclaration ESTIMATE produit également des statistiques de diagnostic pour vous aider à juger de la pertinence du modèle.

Les tests de signification pour les estimations de paramètres indiquent si certains termes du modèle pourraient être inutiles. Les statistiques de qualité d'ajustement aident à comparer ce modèle à d'autres. Les tests de résidus de bruit blanc indiquent si la série résiduelle contient des informations supplémentaires pouvant être utilisées par un modèle plus complexe. L'instruction OUTLIER fournit un autre outil utile pour vérifier si le modèle actuellement estimé tient compte de toute la variation de la série. Si les tests de diagnostic indiquent des problèmes avec le modèle, essayez un autre modèle, puis répétez l'étape d'estimation et de vérification du diagnostic.

- **Prévision:**

Dans la phase de prévision, vous utilisez l'instruction FORECAST pour prévoir les valeurs futures des séries temporelles et pour générer des intervalles de confiance pour ces prévisions à partir du modèle ARIMA produit par l'instruction ESTIMATE précédente.

3. Déclaration de procédure:

Les instructions IDENTIFY, ESTIMATE et FORECAST sont liées dans une hiérarchie. Une instruction IDENTIFY apporte une série temporelle à modéliser; Plusieurs énoncés ESTIMATE peuvent suivre pour estimer différents modèles ARIMA pour la série; Pour chaque modèle estimé, plusieurs instructions FORECAST peuvent être utilisées. Ainsi, une instruction FORECAST doit être précédée à un moment par une instruction ESTIMATE et une instruction ESTIMATE doit être précédée à un moment par une instruction IDENTIFY. Des instructions IDENTIFY supplémentaires peuvent être utilisées pour passer à la modélisation d'une série de réponses différentes ou pour modifier le degré de différenciation utilisé.

La procédure ARIMA peut être utilisée de manière interactive dans le sens où toutes les instructions de procédure ARIMA peuvent être exécutées n'importe quel nombre de fois sans réinvestir PROC ARIMA. Vous pouvez exécuter des instructions de procédure ARIMA individuellement ou en groupes en suivant la seule instruction ou groupe d'instructions avec

une instruction RUN. La sortie de chaque instruction ou groupe d'instructions est produite lorsque l'instruction RUN est entrée.

Une instruction RUN ne termine pas l'étape PROC ARIMA mais indique à la procédure d'exécuter les instructions fournies jusqu'à présent. Vous pouvez mettre fin à PROC ARIMA en envoyant une instruction QUIT, une étape DATA, une autre étape PROC ou une instruction ENDSAS.

La procédure ARIMA peut être utilisée de manière interactive dans le sens où toutes les instructions de procédure ARIMA peuvent être exécutées n'importe quel nombre de fois sans réinvestir PROC ARIMA. Vous pouvez exécuter des instructions de procédure ARIMA individuellement ou en groupes en suivant la seule instruction ou groupe d'instructions avec une instruction RUN. La sortie de chaque instruction ou groupe d'instructions est produite lorsque l'instruction RUN est entrée.

- Le programme complet PROC ARIMA pour cet exemple est le suivant:

❑ Déclaration de procédure:

```
proc arima data=test;
  identify var=sales nlag=24;
  run;
  identify var=sales(1);
  run;
  estimate p=1;
  run;
  estimate p=1 q=1;
  run;
  outlier;
  run;
  forecast lead=12 interval=month id=date out=results;
  run;
quit;
```

4. La stationnarité:

La série de bruit (ou résiduelle) pour un modèle ARMA doit être stationnaire, ce qui signifie que les valeurs attendues de la série et sa fonction d'auto-covariance sont indépendantes du temps.

La méthode standard pour vérifier la non-stationnarité consiste à tracer la série et sa fonction d'auto-corrélation. Vous pouvez examiner visuellement un graphique de la série au cours du temps pour voir si elle a une tendance visible ou si sa variabilité change sensiblement au fil du temps. Si la série n'est pas stationnaire, sa fonction d'auto-corrélation se décompose habituellement lentement.

Un autre moyen de vérifier la stationnarité consiste à utiliser les essais de stationnarité décrits dans la section Essais de stationnarité.

La plupart des séries temporelles sont non stationnaires et doivent être transformées en une série stationnaire avant que le processus de modélisation ARIMA puisse se poursuivre. Si la série a une variance non stationnaire, prendre le journal de la série peut aider. Vous pouvez calculer les valeurs du journal dans une étape DATA puis analyser les valeurs du journal avec PROC ARIMA.

Si la série a une tendance au fil du temps, une saisonnalité ou un autre motif non stationnaire, la solution habituelle consiste à prendre la différence de la série d'une période à l'autre, puis à analyser cette série différenciée. Parfois, une série peut-être besoin d'être différenciés plus d'une fois ou différenciés à des décalages de plus d'une période. (Si la tendance ou les effets saisonniers sont très réguliers, l'introduction de variables explicatives peut être une alternative appropriée à la différenciation.)

5. Réalisation :

Dans cette partie nous allons attaquer la procédure ARIMA, voir les résultats et essayer de commenter les résultats obtenues.

Avant de commencer nous allons créer des variables différentielles (comme c'est indiqué dans la figure ci-dessous) pour les utiliser durant tous notre projet.

```

1 data au_sales; set au_sales; dppi=dif(ppi);
2 lppi=lag(ppi); ldpp=lag(dppi);
3 run;
4
5 %let ylist=ppi;
6 %let dylist=dppi;
7 %let time=t;
8 %let lylist=lppi;
9 %let trend= trend;
10 %let xlist=cpi gdp;
11

```

Figure 8 : Les variables différentielles.

- La procédure ARIMA :

Dans cette figure nous essayons de vérifier si la variable (ppi) est stationnaire dans un premier lieu et puis nous allons ajouter un paramètre de bruit blanc et comparer les résultats.

```

23 *La procédure ARIMA;
24 proc arima data=au_sales;
25 identify var=&ylist stationarity=(adf);
26 run;
27 *la procédure ARIMA en ajoutant un paramètre de bruit;
28 proc arima data=au_sales;
29 identify var=&ylist(1) stationarity=(adf);
30 run;
31

```

Figure 9 : la syntaxe du code ARIMA dans le cas normal et le cas ou en ajoutant le paramètre de bruit.

La procédure ARIMA

Nom de la variable = ppi	
Moyenne des séries de travail	64.68154
Ecart-type	30.17628
Nombre d'observations	169

Vérification de l'autocorrélation pour le bruit blanc									
Jusqu'au retard	Khi-2	DDL	Pr > khi-2	Autocorrélations					
6	960.86	6	<.0001	0.990	0.978	0.966	0.952	0.937	0.923
12	1789.38	12	<.0001	0.908	0.894	0.880	0.866	0.852	0.838
18	2489.96	18	<.0001	0.824	0.810	0.795	0.780	0.765	0.749
24	3048.92	24	<.0001	0.732	0.716	0.698	0.681	0.663	0.645

Tests de racine unitaire de Dickey-Fuller augmentés							
Type	Retards	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Moyenne zéro	0	0.9750	0.9071	5.66	0.9999		
	1	0.9132	0.8965	2.47	0.9969		
	2	0.8823	0.8908	2.10	0.9916		
Moyenne simple	0	-0.1024	0.9513	-0.26	0.9272	21.27	0.0010
	1	-0.3804	0.9346	-0.51	0.8853	4.88	0.0425
	2	-0.4918	0.9269	-0.61	0.8643	3.96	0.0911
Tendance	0	-1.4094	0.9819	-0.79	0.9635	0.32	0.9900
	1	-4.9336	0.8221	-1.46	0.8407	1.08	0.9570
	2	-5.4808	0.7807	-1.47	0.8348	1.13	0.9503

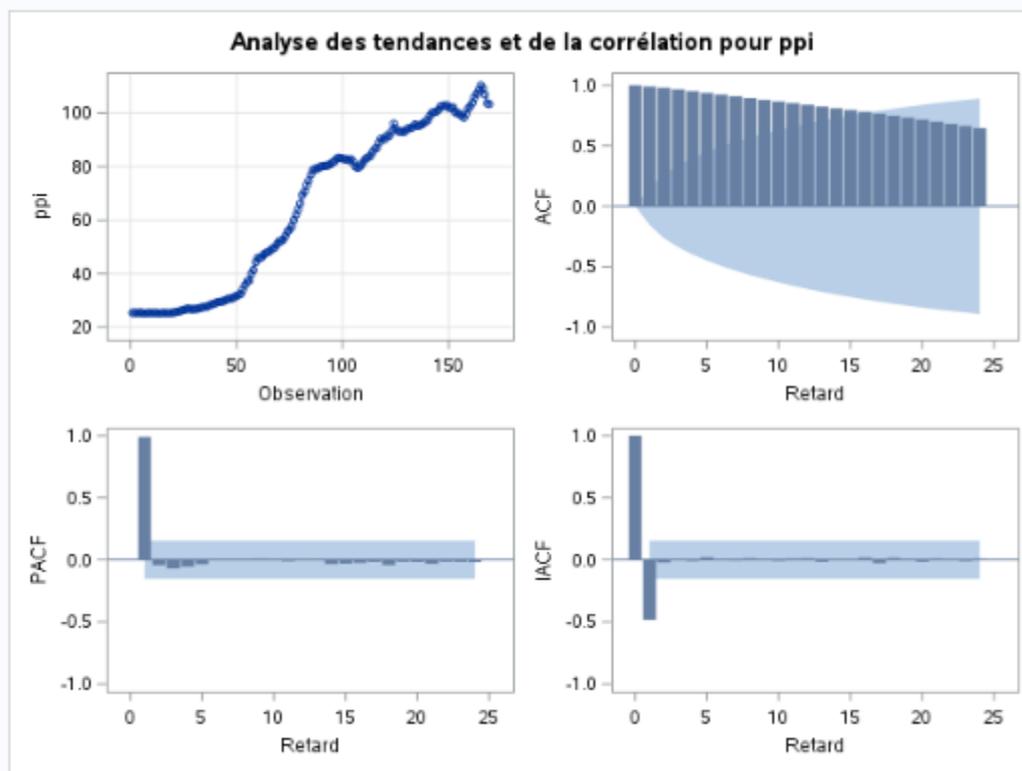


Figure 10 : les résultats de la première procédure ARIMA.

Nous concluons que la variable (ppi) n'est pas stationnaire si nous vérifions le graphe ACF les auto-corrélations diminuent lentement, et si on ajoutant le paramètre de bruit la variable n'est pas stationnaire aussi comme c'est indiqué dans la figure ci-dessous.

La procédure ARIMA

Nom de la variable = ppi	
Période(s) de différenciation	1
Moyenne des séries de travail	0.464286
Ecart-type	0.918001
Nombre d'observations	168
Observation(s) éliminée(s) par la différenciation	1

Vérification de l'autocorrélation pour le bruit blanc									
Jusqu'au retard	Khi-2	DDL	Pr > khi-2	Autocorrélations					
6	102.82	6	<.0001	0.553	0.335	0.319	0.216	0.086	0.153
12	106.35	12	<.0001	0.082	-0.078	-0.080	0.023	-0.008	-0.006
18	112.72	18	<.0001	0.112	0.069	0.048	0.039	0.084	0.077
24	117.91	24	<.0001	0.076	0.085	0.049	0.033	0.047	0.089

Tests de racine unitaire de Dickey-Fuller augmentés							
Type	Retards	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Moyenne zéro	0	-59.1439	<.0001	-5.97	<.0001		
	1	-45.1897	<.0001	-4.66	<.0001		
	2	-25.1469	0.0002	-3.27	0.0012		
Moyenne simple	0	-74.3553	0.0013	-6.86	<.0001	23.53	0.0010
	1	-64.7748	0.0013	-5.49	<.0001	15.08	0.0010
	2	-38.7515	0.0013	-3.85	0.0031	7.45	0.0010
Tendance	0	-74.3509	0.0005	-6.84	<.0001	23.41	0.0010
	1	-64.5966	0.0005	-5.47	<.0001	15.03	0.0010
	2	-38.2834	0.0008	-3.81	0.0184	7.51	0.0197

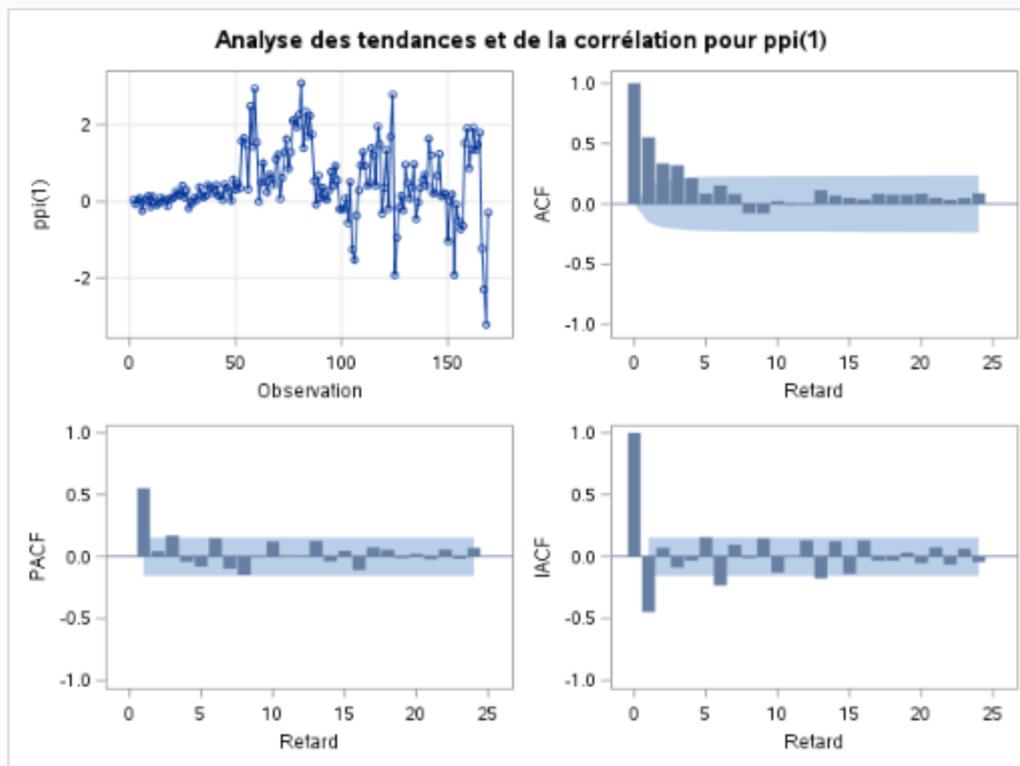


Figure 11 : les résultats de la deuxième procédure ARIMA.

- La procédure ARIMA (1,0, 0) équivalente à AR (1) :

Dans ce cas nous spécifions le paramètre de l'estimation $p=1$, voici les résultats obtenus.

```

38 *La procedure ARIMA en integrant la syntaxe d'estimation 1 dans ce cas la procedure ARIMA est
39 devenu AR(1);
40 proc arima data=au_sales;
41 identify var=&ylist;
42 estimate p=1;
43 run;
44

```

Estimation des moindres carrés conditionnels					
Paramètre	Estimation	Erreur type	Valeur du test t	Approx Pr > t	Retard
MU	26.75853	1.03715	25.80	<.0001	0
AR1,1	1.00000	0.0016585	602.95	<.0001	1

Figure 12 : AR(1).

- La procédure ARIMA (0,0, 1) équivalente à MA (1) :

Dans ce cas nous spécifions le paramètre de l'estimation $q=1$, voici les résultats obtenus.

```

45 *La procedure ARIMA en integrant la syntaxe d'estimation ESTIMATE q=1 dans ce
46 cas elle devient MA(1);
47 proc arima data=au_sales;
48 identify var=&ylist;
49 estimate q=1;
50 run;

```

Estimation des moindres carrés conditionnels					
Paramètre	Estimation	Erreur type	Valeur du test t	Approx Pr > t	Retard
MU	63.10136	2.39720	26.32	<.0001	0
MA1,1	-0.93793	0.02740	-34.23	<.0001	1

Figure 13 : MA(1).

- La procédure REG :

Dans cette procédure nous essayons de tester la régression du processus, dans ce cas du premier modèle la valeur estimée des paramètres est significativement différent de zéro, alors le processus n'est pas stationnaire.

```

32 *Teste de Regression;
33 proc reg data=au_sales;
34 model &dylist=&ylist;
35 model &dylist=&ylist &trend;
36 run;
--

```

La procédure REG
Modèle : MODEL1
Variable dépendante : dppi

Nb d'observations lues	169
Nb d'obs. utilisées	168
Nombre d'observations avec valeurs manquantes	1

Analyse de variance					
Source	DDL	Somme des carrés	Moyenne quadratique	Valeur F	Pr > F
Modèle	1	0.05661	0.05661	0.07	0.7970
Erreur	166	141.52119	0.85254		
Total sommes corrigées	167	141.57780			

Root MSE	0.92333	R carré	0.0004
Moyenne dépendante	0.46429	R car. ajust.	-0.0056
Coeff Var	198.87096		

Résultats estimés des paramètres					
Variable	DDL	Valeur estimée des paramètres	Erreur type	Valeur du test t	Pr > t
Intercept	1	0.50357	0.16827	2.99	0.0032
lppi	1	-0.00060951	0.00237	-0.26	0.7970

Figure 14 : Le test de régression (modèle1).

Le deuxième modèle nous ajoutons la variable de tendance et nous remarquons qu'elle significativement différente de zéro alors le processus n'est pas stationnaire.

La procédure REG
Modèle : MODEL2
Variable dépendante : dppi

Nb d'observations lues	169
Nb d'obs. utilisées	168
Nombre d'observations avec valeurs manquantes	1

Analyse de variance					
Source	DDL	Somme des carrés	Moyenne quadratique	Valeur F	Pr > F
Modèle	2	0.54332	0.27166	0.32	0.7282
Erreur	165	141.03448	0.85475		
Total sommes corrigées	167	141.57780			

Root MSE	0.92453	R carré	0.0038
Moyenne dépendante	0.46429	R car. ajust.	-0.0082
Coeff Var	199.12939		

Résultats estimés des paramètres					
Variable	DDL	Valeur estimée des paramètres	Erreur type	Valeur du test t	Pr > t
Intercept	1	0.58114	0.19737	2.94	0.0037
Ippi	1	-0.00839	0.01058	-0.79	0.4289
trend	1	0.00496	0.00657	0.75	0.4516

Figure 15 : Le test de régression (modèle2).

VI. Conclusion :

Ce projet nous a permis de manipuler le logiciel SAS University Edition et de bien maîtriser tous ses fonctionnalités, dans un premier pas nous avons établi nos premières manipulations en utilisant les procédures (IMPORT, PRINT, MEANS et PLOT), et vers la fin nous avons abordé la procédure ARIMA l'appliquer sur notre série temporelle et commenter les résultats afin de déterminer si le processus est stationnaire ou pas.