Rapport SAE 3-03 : Exploration de données temporelles

Plan :

Partie 1: Choix et Présentation de la série

Partie 2: Méthode analyse saisonnier

1) Création de la table de données
2) Représentation graphique du jeu de données
3) Calcul des moyennes mobiles
4) Calcul des coefficients saisonniers
5) Calcul de la série corrigée Ycvs
6) Etude des résidus
7) Lissage Exponentiel de la série
8) Prévision sur l'année suivante.

Partie 3: Analyse des données grâce à l'analyse saisonnière

1) Analyse des données traitées et résultats déduits
2) Conclusion de la SAE

FINTZ Jules
GUESDON Theo

EMS
04/12/2022

Partie 1: Choix et Présentation de la série

Pour cette SAE, nous avons cherché nos données temporelles sur plusieurs plateformes tel que data.gouv ou l'INSEE. C'est sur le site de l'INSEE que nous avons trouvé notre jeu de données. Nous avons choisi la représentation de la quantité de touristes par année qui arrive en France métropolitaine de 20XX-2020.
exemple:

Institut national de la statistique et des études économiques
Insee
Mesurer pour comprendre

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

data.gouv.fr

Après avoir récupérer le jeu de données choisi :

Dans un premier temps nous avons commencé par utiliser un jeu de données tout autre pour ainsi tester nos fonctions et processus. c'est pour cela que nous avons une partie du programme appelé data test qui nous a servi de base de données test. (voir ci-dessous)

```
#total
annee<-c(2020,2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018)
mois<-c(1,12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1)
val_y<-c(13.2,18.8, 13.9, 15.5, 16.4, 27.1, 20.6, 19.5, 15.8, 18.2, 14.3, 14.1, 13.4, 18.4, 13.4, 15.6, 17.4, 27, 22, 16.4, 19.9, 17.3, 14.1, 13, 14.5)
#2018
mois_2018<-c(12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1)
val_2018<-c(18.4, 13.4, 15.6, 17.4, 27, 22, 16.4, 19.9, 17.3, 14.1, 13, 14.5)
#2019
mois_2019<-c(12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1)
val_2019<-c(18.8, 13.9, 15.5, 16.4, 27.1, 20.6, 19.5, 15.8, 18.2, 14.3, 14.1, 13.4)
#2020
mois_2020<-c(1)
val_2020<-c(13.2)
```

Nous avons procédé ainsi car les données que l'on voulait étudier étaient nombreuses et complexes, il était donc plus efficace de tester nos fonctions avec un jeu de données simple et court pour faciliter les tests.

Ensuite,nous avons dû procéder à un traitement de nos données afin de pouvoir les insérer dans le logiciel RStudio. Nous avons donc modifier les données initiales dans le logiciel excel. L'étape réalisée a été :

- Recodage du type de variable (Caractère -> entier) (voir ci-dessous)

2022-07	21,0 (P)
2022-06	17,2 (P)
2022-05	18,8 (P)
2022-04	16,4 (P)
2022-03	13,2 (P)
2022-02	14,3 (P)

- Vérification données correctes, pas de données parasites ou erronées.

Enfin nous allons utiliser notre vrai jeu de données nettoyé, et nous l'avons importé dans notre fichier Rstudio à l'aide de la fonction read_excel.

Partie 2: Méthode analyse saisonnier

Lors de cette SAE nous avons fait attention à respecter le procédé vue en cours pour l'analyse de donnée temporelle.
Nous avons donc procédé ainsi:

1) Création de la table de données

Nous avons trouvé notre jeu de données en cherchant des séries chronologiques sur l'INSEE. Ce dernier représente le nombre de personnes ayant passé au moins une nuit dans un hôtel en France entre Janvier 2011 et Décembre 2019. Il comporte deux variables :

- La date composée du mois et de l'année
- Le nombre de nuitées passées dans l'ensemble des hôtels métropolitains
- les données sont exprimé en milliers de personne (12—> 12 000)

Pour créer notre jeu de donnée optimal pour l'analyse nous avons procédé de cette manière:

Dans un premier temps sur excel, nous avons ajouté un titre et transformé le type de chaque variable.

mois	nuité
2019-12	4538
2019-11	3180

Dans un second temps, nous avons séparé les données récupérées en créant une table contenant les données de chaque année réparties en colonnes.

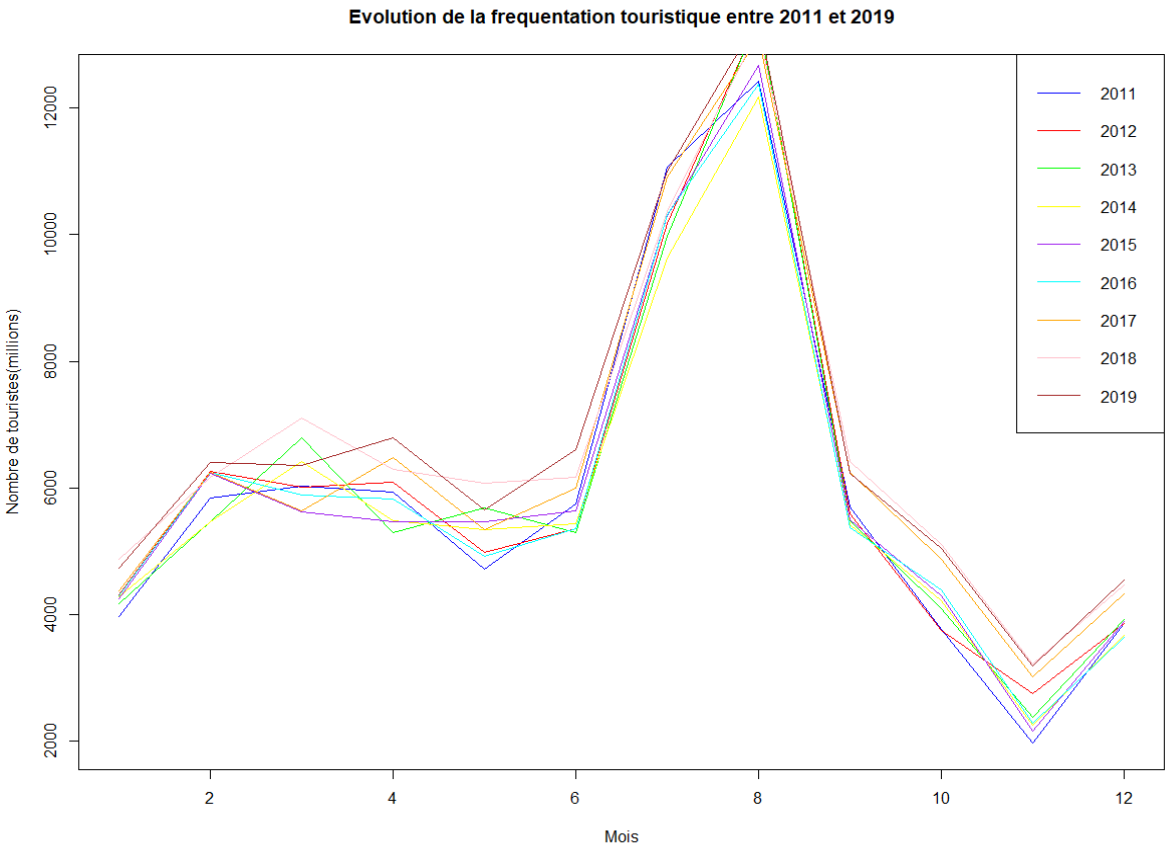
an_2011	an_2012	an_2013	an_2014	an_2015	an_2016	an_2017	an_2018	an_2019
3850	3848	3917	3657	3887	3626	4326	4458	4538
1965	2738	2365	2247	2144	2280	3014	3215	3180
3765	3737	4084	4212	4296	4383	4866	5102	5043
5693	5584	5481	5417	5491	5367	6237	6416	6228
12429	13410	13443	12185	12681	12388	13111	13241	13434
11069	10175	9969	9631	10308	10315	10919	10362	11010
5746	5349	5292	5425	5626	5357	5998	6168	6604
4707	4970	5682	5332	5467	4912	5335	6063	5651
5926	6088	5293	5479	5462	5823	6469	6297	6783
6029	6011	6783	6414	5619	5882	5627	7107	6359
5829	6259	5469	5467	6226	6248	6237	6166	6401
3958	4309	4166	4270	4238	4291	4369	4864	4728

Enfin, nous avons créé sur le logiciel R deux listes contenant respectivement les années et les mois de notre jeu de données :

```
annee<-c(2011,2012,2013,2014,2015,2016,2017,2018,2019)
mois<-c(12,11,10,9,8,7,6,5,4,3,2,1)
```

2) Représentation graphique du jeu de données

Une fois, les données réparties par année nous avons représenté graphiquement ces données à l'aide d'un graphique périodique afin de visualiser l'évolution de la fréquentation touristique entre 2011 et 2019.



Nous observons que l'allure des courbes de la fréquentation touristique de 2011 à 2019 est très similaire puisque l'on observe que les courbes sont très rapprochées.

On peut voir que la fréquentation touristique mensuelle reste constante autour de 6000 millions entre Février et Juin. Ensuite, elle connaît un pic au mois d'Août atteignant 13 000 millions de touristes. Cependant, elle chute jusqu'à 2000 millions au mois de Novembre pour devenir la fréquentation la plus faible de l'année.

3) Calcul des moyennes mobiles

```
> moyenne_mobile
 [1] 5913.750 5945.875 5976.917 5971.208 6007.542 6011.167 5957.375 5951.792 5969.500 5975.500 5992.667 6025.208 6042.708
[14] 6030.042 6028.958 6039.125 6036.208 6029.000 6018.042 6045.333 6041.875 6040.917 6040.167 6001.292 5984.500 5968.750
[27] 5969.167 5971.833 5916.750 5850.250 5841.708 5832.667 5825.833 5818.208 5802.750 5807.000 5820.917 5826.208 5825.417
[40] 5832.000 5855.750 5904.625 5941.208 5955.208 5960.125 5926.292 5924.792 5955.083 5942.875 5937.667 5946.958 5945.417
[53] 5928.042 5916.125 5905.208 5870.875 5862.792 5888.792 5900.667 5903.792 5935.167 5994.917 6045.625 6102.000 6168.375
[66] 6223.667 6275.542 6319.875 6364.417 6380.708 6369.625 6372.417 6381.167 6395.042 6413.250 6430.542 6443.417 6425.625
[79] 6409.500 6446.917 6470.083 6524.583 6583.292 6600.958 6624.917 6626.792 6622.875 6612.583 6612.792 6647.833 6693.000
[92] 6694.000 6697.083 6686.167 6664.792 6668.917
>
```

Nous avons commencé par définir notre période d'une durée de 12 mois. Pour calculer nos moyennes mobiles, il nous a fallu choisir la méthode de calcul adapté. Nous étions dans une méthode additive avec une période d'ordre pair. Nous avons donc appliqué la méthode vue en cours.

4) Calcul des coefficients saisonniers

```
> ymed
 [1] -2134.62500 -3594.66146 -1798.02083  -402.33854  6739.89062  4217.46354  -510.07292  -831.08333  -294.33854
[10]    28.85417  -172.21875 -1858.70833
```

Nous avons commencé par soustraire les moyenne mobile aux données. Ensuite, nous avons calculé la moyenne des résultats trouvés chaque mois. Pour cela, nous avons sommé les moyennes de chaque mois à l'aide d'une boucle et nous avons divisé cette somme par le nombre de mois, afin d'obtenir la moyenne des coefficients saisonniers mensuels.

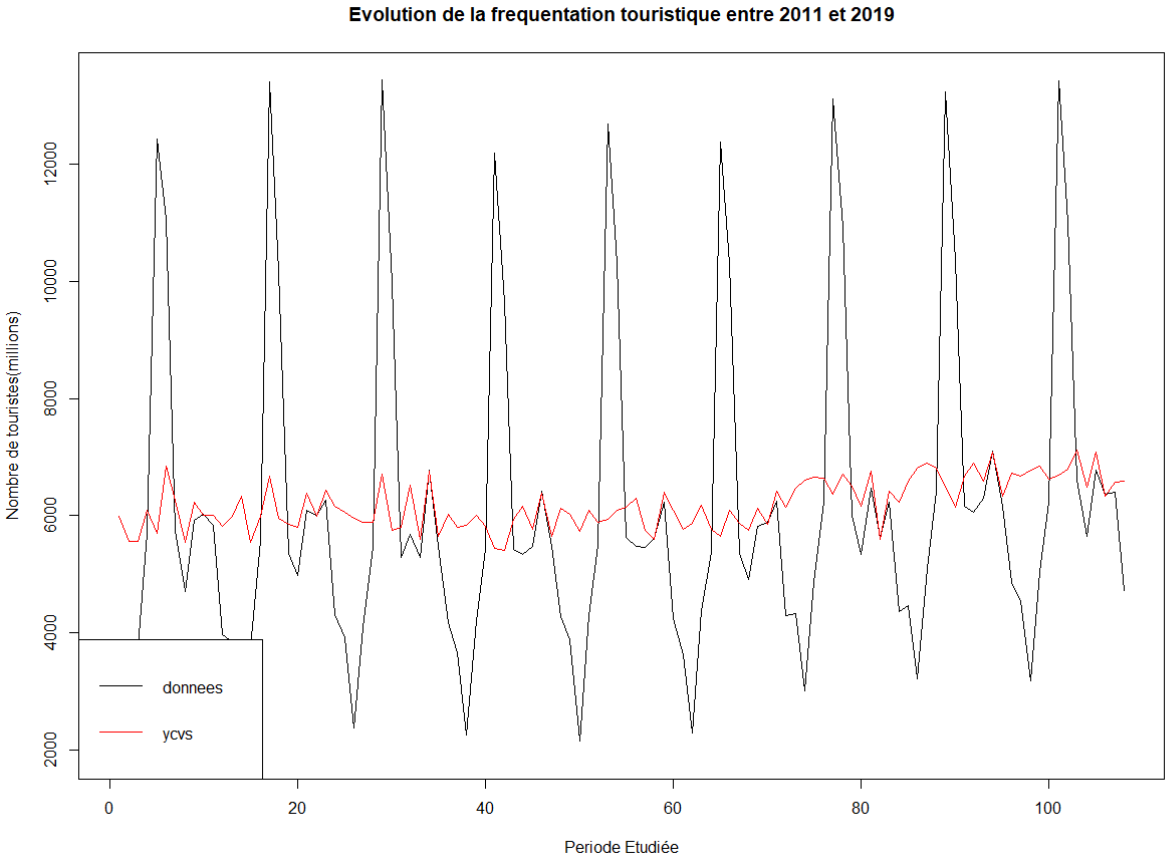
5) Calcul de la série corrigée Ycvs

Pour calculer Ycvs nous avons appliqué la méthode vue en cours, soit:
Nous avons soustrait les données par le coefficient saisonnier correspondant au mois.
Pour cela, nous avons fait une boucle "for" contenant un compteur s'incrémentant à chaque passage dans la boucle et se réinitialisant quand le compteur est égale à 12. Cela correspond à la fin de l'année.
Ce compteur nous permet alors de sélectionner seulement le mois désiré.

Exemple:
Si cpt = 2 alors on soustrait à la valeur le coefficient saisonnier de février.
on obtient alors la séries des valeurs corrigées suivante:

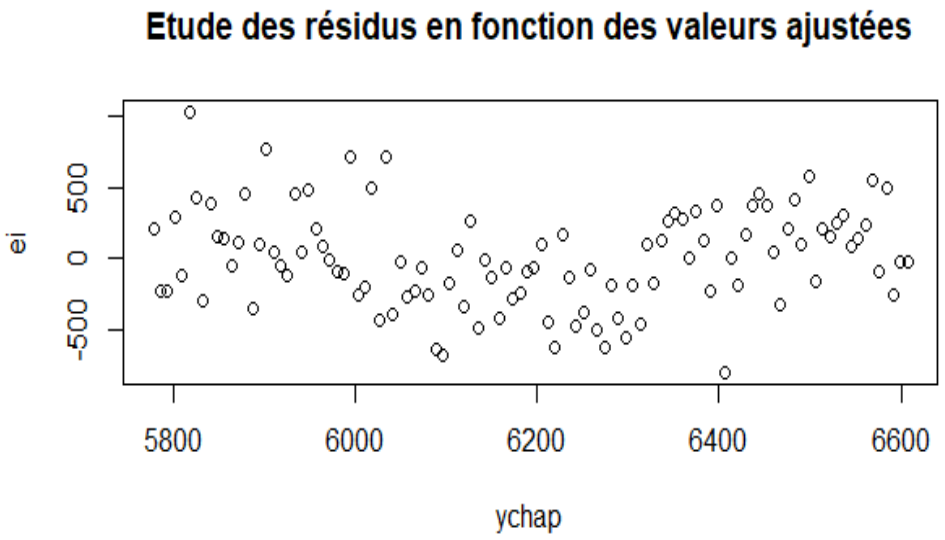
```
> ycvs
  [1] 5984.625 5559.661 5563.021 6095.339 5689.109 6851.536 6256.073 5538.083 6220.339 6000.146 6001.219 5816.708 5982.625
 [14] 6332.661 5535.021 5986.339 6670.109 5957.536 5859.073 5801.083 6382.339 5982.146 6431.219 6167.708 6051.625 5959.661
 [27] 5882.021 5883.339 6703.109 5751.536 5802.073 6513.083 5587.339 6754.146 5641.219 6024.708 5791.625 5841.661 6010.021
 [40] 5819.339 5445.109 5413.536 5935.073 6163.083 5773.339 6385.146 5639.219 6128.708 6021.625 5738.661 6094.021 5893.339
 [53] 5941.109 6090.536 6136.073 6298.083 5756.339 5590.146 6398.219 6096.708 5760.625 5874.661 6181.021 5769.339 5648.109
 [66] 6097.536 5867.073 5743.083 6117.339 5853.146 6420.219 6149.708 6460.625 6608.661 6664.021 6639.339 6371.109 6701.536
 [79] 6508.073 6166.083 6763.339 5598.146 6409.219 6227.708 6592.625 6809.661 6900.021 6818.339 6501.109 6144.536 6678.073
 [92] 6894.083 6591.339 7078.146 6338.219 6722.708 6672.625 6774.661 6841.021 6630.339 6694.109 6792.536 7114.073 6482.083
[105] 7077.339 6330.146 6573.219 6586.708
```

Ainsi on retrouve le graphique ci-dessous représentant les valeurs corrigées:



En observant le graphique ci-dessus, on constate que l'effet saisonnier a bien été supprimé sur la série ycvs puisque que les variations ne sont presque plus présentes. Il n'y a plus de périodicité, on ne visualise plus de variations périodiques entre les différentes années.

6) Etude des résidus



On observe que les valeurs des résidus sont assez éloignées de 0. Cependant, nous avons des données avec un ordre de grandeur élevé ce qui peut expliquer ces grands écarts.

De plus, lorsqu'on somme l'ensemble de ces derniers on obtient une valeur très proche de 0. Dès lors, l'ajustement semble correct.

```
> sum(ei)
[1] 3.158362e-12
```

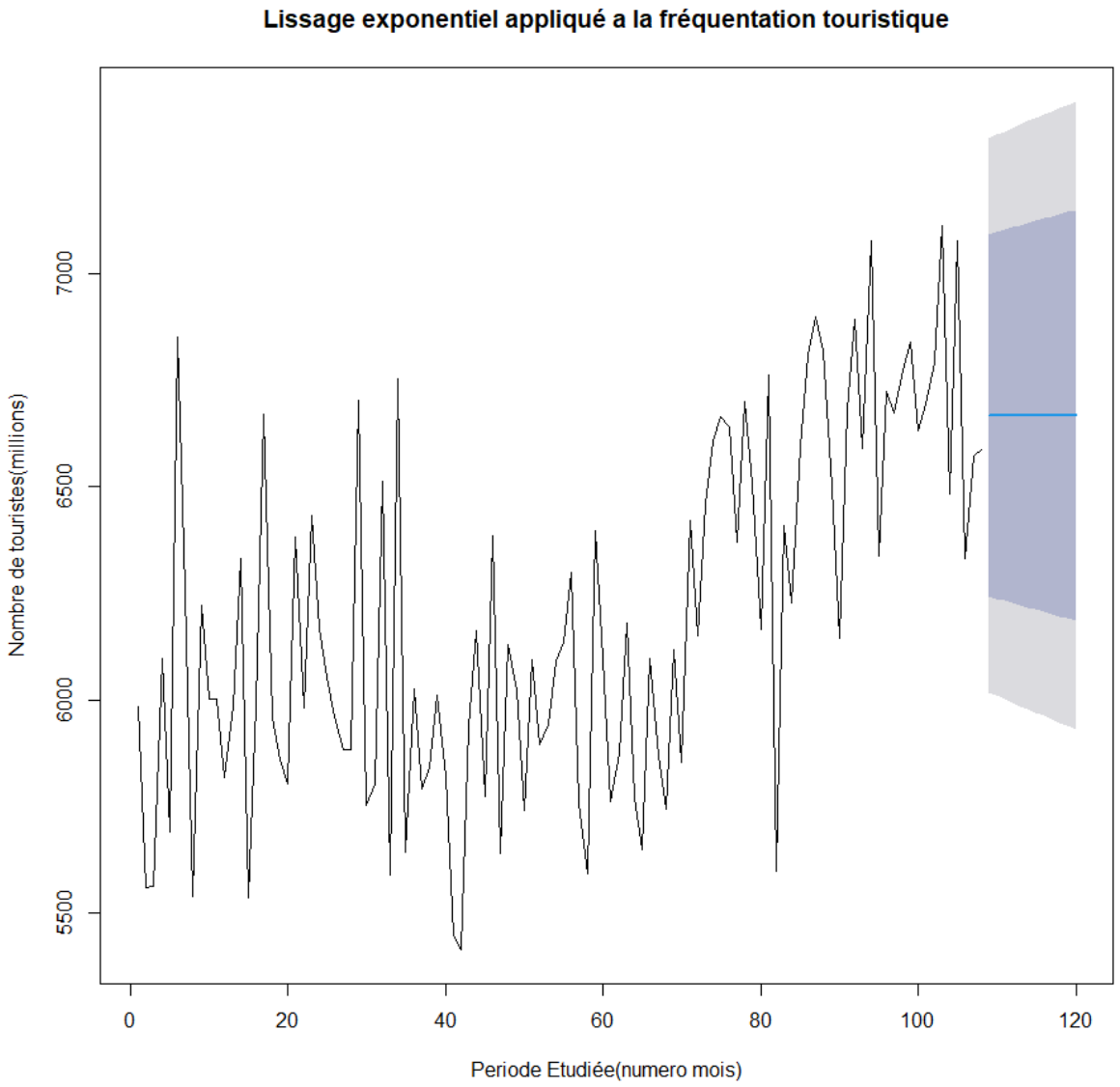
7) Lissage Exponentiel de la série

Pour analyser les données corrigées et effectuer les tests de prévision (voir point 8) nous avons décidé d'utiliser le lissage exponentiel. En effet ce lissage est précis bien que simple a implémenter.

Nous utilisons la méthode vue en cours et en tp pour effectuer un lissage exponentiel sur les données. Grâce à cela nous avons pu récupérer l'ensemble des valeurs. Ci dessous' une partie des données :

```
> m
  [1] 5984.625 5602.158 5566.935 6042.498 5724.448 6738.828 6304.348 5614.710 6159.776 6016.109
 [11] 6002.708 5835.308 5967.893 6296.185 5611.137 5948.818 6597.980 6021.581 5875.324 5808.507
 [21] 6324.955 6016.427 6389.740 6189.911 6065.454 5970.241 5890.843 5884.089 6621.207 5838.504
 [31] 5805.716 6442.347 5672.839 6646.015 5741.698 5996.407 5812.103 5838.706 5992.889 5836.694
 [41] 5484.268 5420.610 5883.627 6135.138 5809.518 6327.583 5708.055 6086.643 6028.127 5767.608
 [51] 6061.380 5910.143 5938.013 6075.284 6129.994 6281.274 5808.832 5612.014 6319.598 6118.997
 [61] 5796.462 5866.842 6149.603 5807.365 5664.035 6054.186 5885.784 5757.353 6081.340 5875.965
 [71] 6365.793 6171.317 6431.694 6590.965 6656.715 6641.076 6398.106 6671.193 6524.385 6201.913
 [81] 6707.196 5709.051 6339.202 6238.858 6557.248 6784.420 6888.461 6825.351 6533.534 6183.436
 [91] 6628.609 6867.536 6618.958 7032.227 6407.620 6691.199 6674.482 6764.644 6833.383 6650.643
[101] 6689.763 6782.259 7080.892 6541.964 7023.801 6399.511 6555.848 6583.622 5984.625 5559.661
[111] 5563.021 6095.339 5689.109 6851.536 6256.073 5538.083 6220.339 6000.146 6001.219 5816.708
[121] 5982.625 6332.661 5535.021 5986.339 6670.109 5957.536 5859.073 5801.083 6382.339 5982.146
[131] 6431.219 6167.708 6051.625 5959.661 5882.021 5883.339 6703.109 5751.536 5802.073 6513.083
[141] 5587.339 6754.146 5641.219 6024.708 5791.625 5841.661 6010.021 5819.339 5445.109 5413.536
[151] 5935.073 6163.083 5773.339 6385.146 5639.219 6128.708 6021.625 5738.661 6094.021 5893.339
[161] 5941.109 6090.536 6136.073 6298.083 5756.339 5590.146 6398.219 6096.708 5760.625 5874.661
[171] 6181.021 5769.339 5648.109 6097.536 5867.073 5743.083 6117.339 5853.146 6420.219 6149.708
[181] 6460.625 6608.661 6664.021 6639.339 6371.109 6701.536 6508.073 6166.083 6763.339 5598.146
[191] 6409.219 6227.708 6592.625 6809.661 6900.021 6818.339 6501.109 6144.536 6678.073 6894.083
```

Ensuite, nous pouvons ajouter ces données au tableau des Ycvs et des données, ainsi on se retrouve avec le graphique ci-dessous :



On observe en réalisant le graphique du lissage exponentiel appliquée à la fréquentation touristique que la périodicité de nos données à totalement disparu et l'on peut plus facilement comparer la fréquentation entre chaque mois.
De plus, on visualise plus facilement l'évolution de la fréquentation qui tend à rester constante comme on peut l'observer sur la fin de notre courbe. Les variations qui au 10e mois pouvaient atteindre 1500 millions ne représentent plus qu'environ 500 millions au 110e mois.

8) Prévision sur l'année suivante.

En suivant la méthode du lissage exponentiel, nous pouvons prédire les valeurs futures sur une courte période. Nous avons donc commencé par étirer le lissage exponentiel pour lancer une prédiction.

Maintenant, nous allons étudier grâce au package forecast, les valeurs possibles que pourrait prendre la courbe. En effet, bien que le lissage permette de prédire les valeurs sur l'année suivante, cela ne suffit pas. Il faut pour cela connaître un champ des possibles (montrer par un rectangle bleu) dans lesquelles nous aurions une grande chance de trouver les valeurs réelles.

Partie 3: Analyse des données grâce à l'analyse saisonnière

1) Analyse des données traitées et résultats déduits

Nous avons décidé d'observer les données sur le nombre de nuitées touristiques par mois en France métropolitaine. Après traitement effectué en partie 2, nous pouvons relever plusieurs informations:
-on relève une chute du nombre de nuitées durant le début d'années et fin d'année avec une chute maximale en novembre
-on relève une très forte augmentation du nombre de nuitées durant la période estivale avec un pic en août.
-Grâce à Ycvs nous observons que la fréquentation touristique augmente très légèrement sur ces 9 années..

Grâce au lissage exponentiel et à la prévision effectuée, nous pouvons affirmer que l'année 2020 devrait être légèrement en augmentation par rapport aux autres années. cela comparer avec l'année 2020 encouru nous permet de constater la perte touristique due au covid

2) Conclusion de la SAE

Ainsi, lors de cette SAE, nous avons mis en pratique les compétences acquises pendant les cours et TP. Nous avons fait face à plusieurs difficultés, sur l'implémentation sous R des méthodes mathématiques comme le lissage exponentiel ou le le calcul des valeurs corrigées saisonnières.

Nous avons donc beaucoup appris sur les méthodes de codage sous R et la gestion de données complexes et nombreuses.
En effet, nous ne pouvons pas vérifier directement sur les données si les valeurs étaient justes ou erronées, c'est pour cela que nous avons créé un jeu de données test pour vérifier nos fonctions et processus.

De plus, nous avons découvert la récupération de données tels que les données de l'insee et les méthodes de traitement qui pourront se révéler utiles au vue du nombre de données existantes sur le site de l'INSEE et data.gouv.
En conclusion, ce projet aura été un vrai défi très intéressant et utile.