Projet final à rendre - Logiciel statistique R

Thibault LAURENT

A rendre avant le 30 Décembre 2018

Contents

Ce document a été généré directement depuis **RStudio** en utilisant l'outil Markdown. La version .pdf se trouve ici.

Résumé

L'objectif de ce projet est que vous appliquiez les outils que vous avez étudiés dans le cours d'initiation aux logiciels statistiques, dans le cas d'une étude de cas réelle. Le devoir est à faire seul et à rendre au format .html ou .pdf. Les codes que vous utiliserez pour répondre aux questions seront à intégrer dans le corps de votre rapport. C'est pourquoi l'utilisation de **R** Markdown est à privilégier, mais vous pouvez également utiliser d'autres éditeurs de textes...

En ce qui concerne l'organisation du travail à rendre, vous vous inspirerez de la façon dont est organisé le sujet du projet (vous n'êtes pas obliger de recopier l'énoncé).

Le barème est indiqué en face de chaque question. Il tient compte du code proposé et dans le cas où nous vous demandons de faire des commentaires sur les résultats obtenus, nous tiendrons compte des commentaires que vous aurez fait dans la notation.

IMPORTANT : il est demandé de commenter vos lignes de code. Vous pourriez être pénalisés si vous ne commentez pas vos codes.

1 Présentation du problème

Les services d'urgences de Toulouse disposent d'un outil informatique qui leur permet de récupérer le nombre de personnes qui sont entrées dans un service d'urgence pendant une journée.

On dispose d'une série temporelle journalière, c'est-à-dire un jeu de données où les individus observés sont les jours, allant du 1er janvier 2003 au 26 octobre 2009.

La variable étudiée est le "nombre de personnes entrées dans le service d'urgence".

La problématique du service d'urgence est de connaître au mieux cette variable "nombre de personnes entrées dans le service d'urgence" pour anticiper d'éventuels problèmes de suractivité.

Dans un premier temps, vous allez procéder aux étapes d'importation et de mise en forme des données, puis de création de nouvelles variables.

Ensuite, vous étudierez si la variable d'intérêt se comporte différemment en fonction du jour de la semaine, des mois, des années, etc.

Enfin, à partir d'observations météorologiques journalières observées dans la région de Toulouse, vous vérifierez s'il existe un lien entre le nombre d'entrants et les précipitations ou les températures.

2 Préparation des données

2.1 Description des fichiers .csv

2.1.1 Le fichier urgence.csv

Le fichier urgence.csv contient 366 observations et 8 variables. La première colonne **MMJJ** correspond à l'identifiant de la base dans sa forme actuelle. Il s'agit d'une date de type (1er janvier, 2 janvier, etc.) codée de la façon suivante : les deux derniers chiffres correspondent au numéro du jour et les premiers chiffres correspondent au mois. Par exemple, 403 signifie le '3 avril' et 215 le '15 février'.

La colonne qui s'appelle **A** correspond à l'année en cours, la colonne **A__1** correspond à l'année 2008 et ainsi de suite jusqu'à la colonne **A__6** qui correspond à l'année 2003.

- **Remarque 1** : la dernière ligne du fichier correspond au 29 février qui n'existe que les années bissextiles (2008 et 2004 pendant la période observée). Dans le fichier, ce jour est renseigné tous les ans; il s'agit donc d'une erreur et il faudra penser à supprimer ces jours qui n'existent pas dans la table finale.
- **Remarque 2** : la colonne **A** a des valeurs manquantes de 301 à 365. En effet, comme l'étude s'était arrêtée le 27 octobre 2009, au-delà de cette date, il n'y a plus d'observations.

2.1.2 Le fichier meteo.csv

Le fichier meteo.csv contient 2504 observations et 5 variables et correspond à la période du 1er janvier 2003 au 8 novembre 2009. Les colonnes **AN**, **MOIS** et **JOUR** sont les identifiants de cette base. Elles correspondent respectivement à l'année, le mois et le jour. La colonne **RR** correspond aux précipitations observées et la colonne 'M correspond à la température moyenne observée ladite journée.

2.2 Importation et mise en forme (2 pts)

Importer les jeux de données "urgence.csv" et "meteo.csv", puis créer une base de données unique intitulée **projet**, qui contiendra en ligne chacun des jours observés entre le 1er janvier 2003 et le 31 octobre 2009 et qui aura en colonne les variables suivantes :

- **annee** : l'annee du jour d'entrée aux urgences
- **mois** : le mois du jour d'entrée aux urgences
- **jour** : le numéro du jour d'entrée aux urgences
- **precip** : les précipitations observées
- **temp** : les températures observées
- **date_j** : un **character** de la forme "dd/mm/aaaa" (par exemple "01/01/2003" pour le 1er janvier 2003).
- **entree** : le nombre de personnes entrantes

Remarque 1 : la principale difficulté va consister à transformer la table **urgence** dans le même format que celui demandé (par annee, mois et jour). Il y a plusieurs façons de procéder; penser à bien commenter ce que vous faites. Par ailleurs, vous tiendrez compte de ce qui a été dit ci-dessus concernant le 29 février.

Remarque 2 : entre le 27 et 31 octobre 2009, le nombre d'entrées aux urgences n'étant pas renseigné, on laissera la valeur **NA** à la variable **entree** durant cette période. Un des buts du devoir sera d'essayer de prédire les valeurs d'entrée pendant cette période.

Remarque 3 : penser à verifier que les variables **RR** et **TM** aient été importées en **numeric** et non pas en **factor**.

Remarque 4 : on pourra convertir la variable **date_j** en objet de classe **date** à l'aide la fonction *as.Date(, format = "%d/%m/%Y")*

Pour vérifier la validité de votre code, vous devrez faire afficher le code suivant :

```
str(projet)
```

```
## 'data.frame':    2496 obs. of  7 variables:
##  $ annee : int  2003 2003 2003 2003 2003 2003 2003 2003 2003 2003 ...
##  $ mois  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ jour  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ precip: num  3.4 0.8 0.2 2.4 0.6 1.2 1.2 3.4 0 0 ...
##  $ temp  : num  7.2 10.9 11 6.1 2 0.6 -0.9 4.3 5.4 -0.9 ...
##  $ date_j: Date, format: "2003-01-01" "2003-01-02" ...
##  $ entree: int  145 127 132 120 147 106 111 105 119 95 ...
```

2.3 Traitement des valeurs manquantes (2 pts)

- Identifier à quelles jours correspondent les valeurs manquantes dans les variables **temp** et **precip**
- Vous remplacerez ces valeurs manquantes par des valeurs de votre choix, en expliquant votre démarche.

Remarque : la notation tiendra fortement compte du raisonnement utilisé. Pour cela, vous pourrez par exemple vous appuyer sur les courbes de séries temporelles, des outils graphiques ou méthodes statistiques de votre choix ou bien utiliser des sites d'informations extérieures.

2.4 Création de variables (2 pts)

2.4.1 Variable jour_semaine

Créer dans la base **projet**, une variable appelée **jour_semaine** qui prend les valeurs :

- "lundi" si la journée était un lundi
- "mardi" si la journée était un mardi
- "mercredi" si la journée était un mercredi
- "jeudi" si la journée était un jeudi
- "vendredi" si la journée était un vendredi

- “samedi” si la journée était un samedi
- “dimanche” si la journée était un dimanche

Remarque : il y a plusieurs façons de procéder; on pourra par exemple s’appuyer sur le fait que le 1er janvier 2003 était un mercredi. Sinon, vous pourrez rechercher la fonction **R** qui renvoit le jour de la semaine lorsqu’on lui présente un objet de type **Date**

2.4.2 Variable saison

Créer dans la base **projet**, une variable **saison** qui vaut :

- **printemps** si le jour est compris entre le 21 mars et le 20 juin (inclus)
- **ete** si le jour est compris entre le 21 juin et le 20 septembre (inclus)
- **automne** si le jour est compris entre le 21 septembre et le 20 décembre (inclus)
- **hiver** si le jour est compris entre le 21 décembre et le 20 mars (inclus)

Vous afficherez le code suivant pour vérifier que la création de variables s’est correctement effectuée :

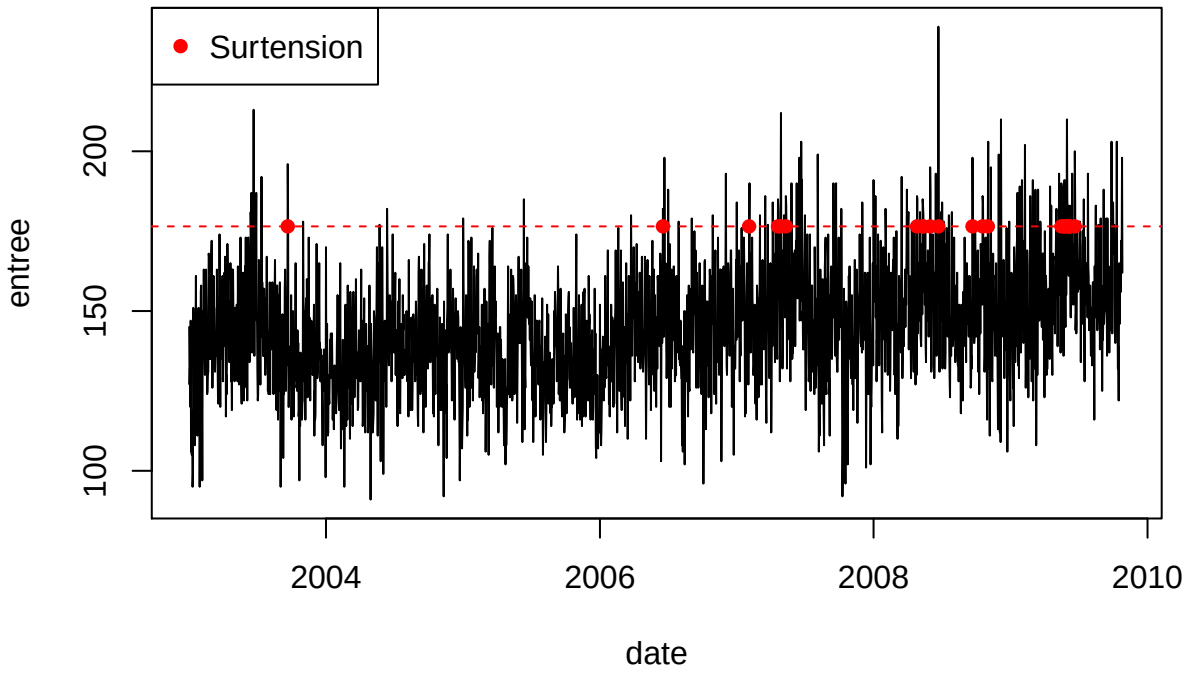
```
projet[1723:1726, ]
```

```
##      annee mois jour precip temp     date_j entree jour_semaine  saison
## 1723  2007    9   19    0.0 14.2 2007-09-19    155     mercredi     ete
## 1724  2007    9   20    0.0 15.0 2007-09-20    159        jeudi     ete
## 1725  2007    9   21    0.4 18.6 2007-09-21    136     vendredi automne
## 1726  2007    9   22    1.0 18.6 2007-09-22    167       samedi automne
```

3 Analyse descriptive

3.1 Etude de la variable entree (2 pt)

- A quel jour correspond la valeur maximale de la variable **entree** ?
- Représenter la boîte à moustache de la variable **entree**.
- Donner les dates des jours (i.e. “mm-jj”) et le jour de la semaine des observations qui sont situées au-dessus de la moustache supérieure (Q3 + 1.5 * (Q3 - Q1)). Y-a-t-il des dates ou jours qui apparaissent plus souvent ? Si oui, quelle est selon-vous la raison de ces pics ?
- En utilisant un test statistique vu en cours, pouvez-vous dire si la distribution de cette variable est gaussienne ?
- Quelle est la valeur du quantile d’ordre 0.95 ?
- Quelles sont les périodes pour lesquelles on observe plus de 2 jours consécutifs avec des valeurs supérieures aux quantiles d’ordre 0.95 ?
- Représenter la série temporelle du nombre d’entrée. Repésenter sur le même graphique par une droite horizontale la valeur du quantile d’ordre 0.95. Représenter également par des points rouges le début des périodes pour lesquelles on a observé pendant au moins 2 jours des valeurs supérieures au quantile d’ordre 0.95.



3.2 Nuage de points (1 pt)

Représenter dans une même fenêtre graphique (utiliser l'option **par(mfrow =)**) le nuage de points de la variable **entree** en fonction de la variable **precip**, puis le nuage de points de la variable **entree** en fonction de la variable **temp**. Vous ajouterez sur les figures, les droites de régression linéaires entre la variable y et x. En utilisant le test statistique approprié, pouvez-vous dire s'il existe un lien significatif de type linéaire entre ces variables ?

3.3 Boîte à moustache // (1 pt)

Représenter les boîtes à moustaches // de la variable **entree** en fonction de la variables **annee**. En utilisant le test statistique adéquat, pouvez-vous dire s'il y a des différences significatives de moyennes entre les groupes.

Répéter le même traitement en remplaçant la variable **annee** par les variable **mois**, puis **jour_semaine**, puis **saison**.

4 Représentation de la série agrégée par mois (3 pts)

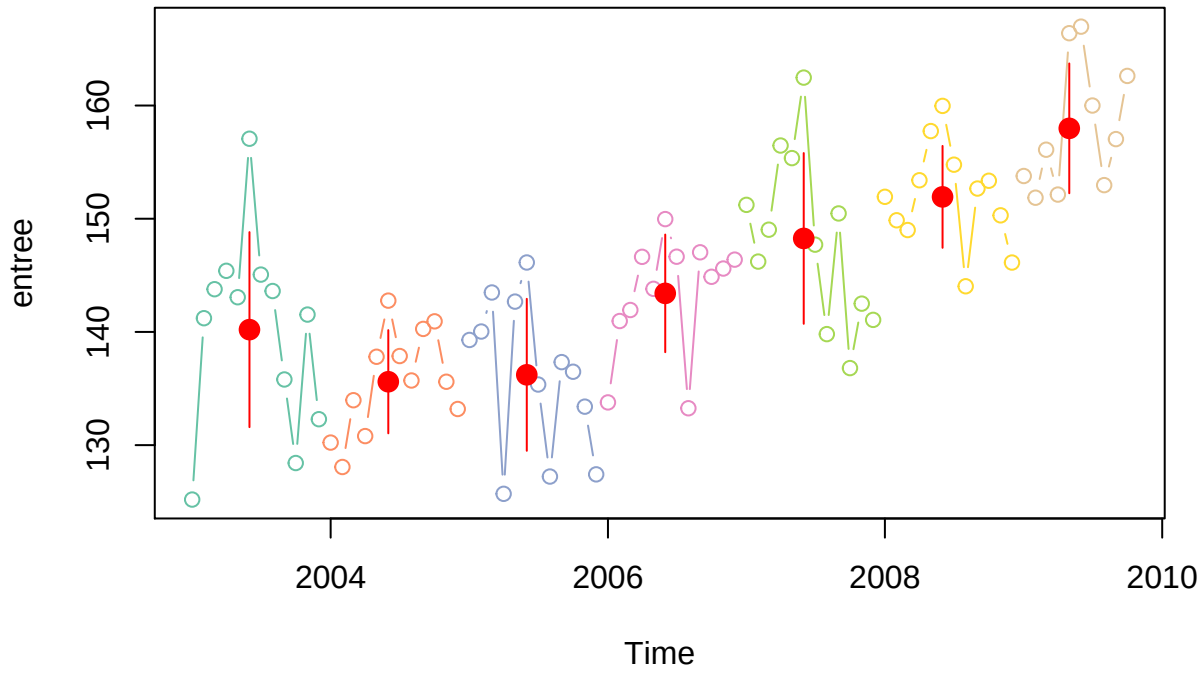
- Créer le **data.frame** nommé **projet_agg** qui contient les variables **entree**, **temp** et **precip** aggrégés (en prenant la moyenne) par mois/année. Ce **data.frame** aura donc 82 lignes et aura cette forme :

```
head(projet_agg, 3)
```

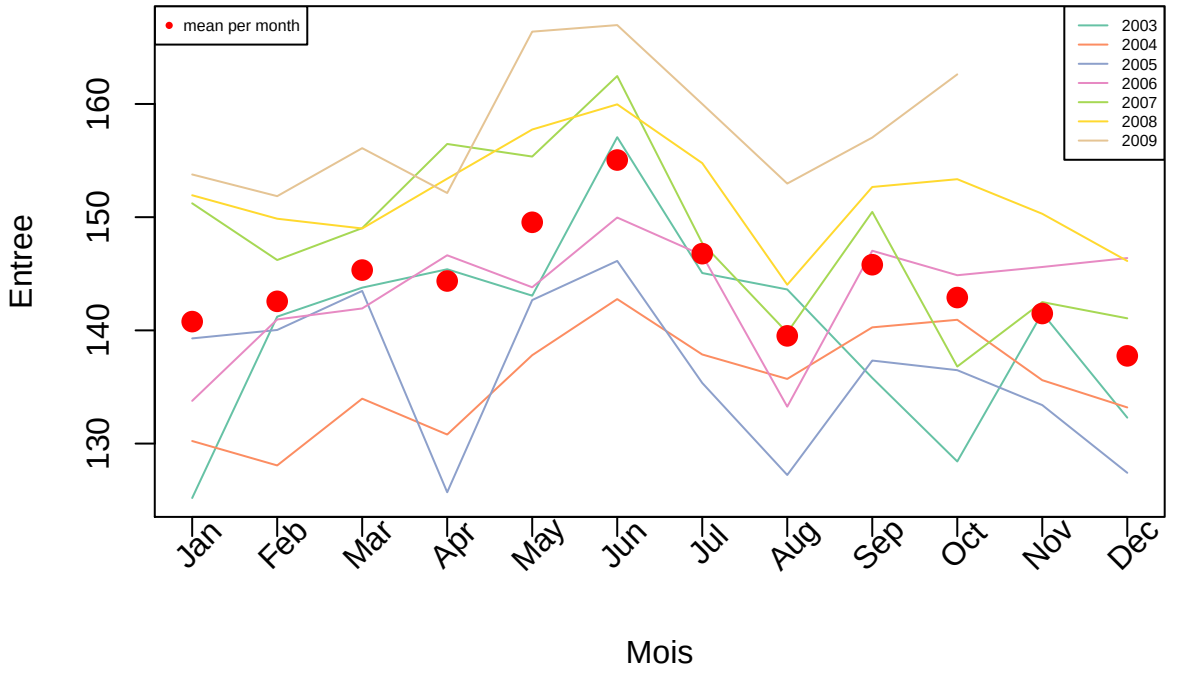
```
##   aaaa_mm   entree      temp   precip
## 1 2003-01 125.1935  4.477419 1.851613
## 2 2003-02 141.2143  5.642857 1.578571
## 3 2003-03 143.7742 11.422581 1.309677
```

- Vous représenterez sur un premier graphique cette série avec une couleur ou un symbole différent par année (vous choisirez les symboles ou couleurs que vous souhaitez) et en reliant uniquement les valeurs

d'une même année. Vous ajouterez également avec des points rouges, la moyenne observée pour chaque année. Vous ajouterez un trait vertical qui part du point représentant la moyenne par un segment de longueur égal à un écart-type calculé par année. A titre d'exemple, vous devrez obtenir un graphique le plus proche possible du graphique suivant.



- Sur un second graphique, vous représenterez sur les 12 mois de l'année, la série obtenue en 2003, celle obtenue en 2004, etc. en gardant les mêmes couleurs (ou symboles) que celles (ceux) utilisées dans le graphique précédent. Vous devrez également représenter par un point rouge la moyenne observée par mois. A titre d'exemple, vous devrez obtenir un graphique le plus proche possible du graphique suivant. N'oubliez pas d'indiquer les légendes.



- Est-ce qu'il vous semble qu'il y a une tendance au cours du temps ? Est-ce qu'il vous semble qu'il y a

une saisonalité mensuelle ?

Remarque : n'hésitez pas à faire preuve d'imagination en essayant d'améliorer l'esthétique de vos graphiques.

5 Lissage par moyenne mobile (3 pts)

5.1 Définition

Soit y_t une série, où $t = 1, ..., n$ représente le temps. Un lissage par moyenne mobile revient à associer à t la valeur lissée suivante :

$$\hat{y}_t = \frac{1}{2\alpha + 1} \sum_{i=t-\alpha}^{t+\alpha} y_i$$

où :

- α, le paramètre de lissage, est un entier compris entre 1 et $n/4$
- t est un entier compris entre $\alpha + 1$ et $n - \alpha$.

5.2 Fonction *lissage()*

Créer la fonction *lissage()* qui aura comme arguments d'entrée :

- **var_to_smooth**, un vecteur de type **numeric** qui correspond à la variable à lisser
- **alpha**, un scalaire de type **numeric** qui correspond au paramètre de lissage
- **var_time**, un vecteur de type **Date**, optionnel, qui correspond aux dates correspondant à la variable à lisser. Si cette variable n'est pas renseignée, on la remplacera par un vecteur allant de 1 à n où n est la longeur du vecteur **var_to_smooth**.

Cette fonction retournera un objet de type **list** comprenant les éléments suivants :

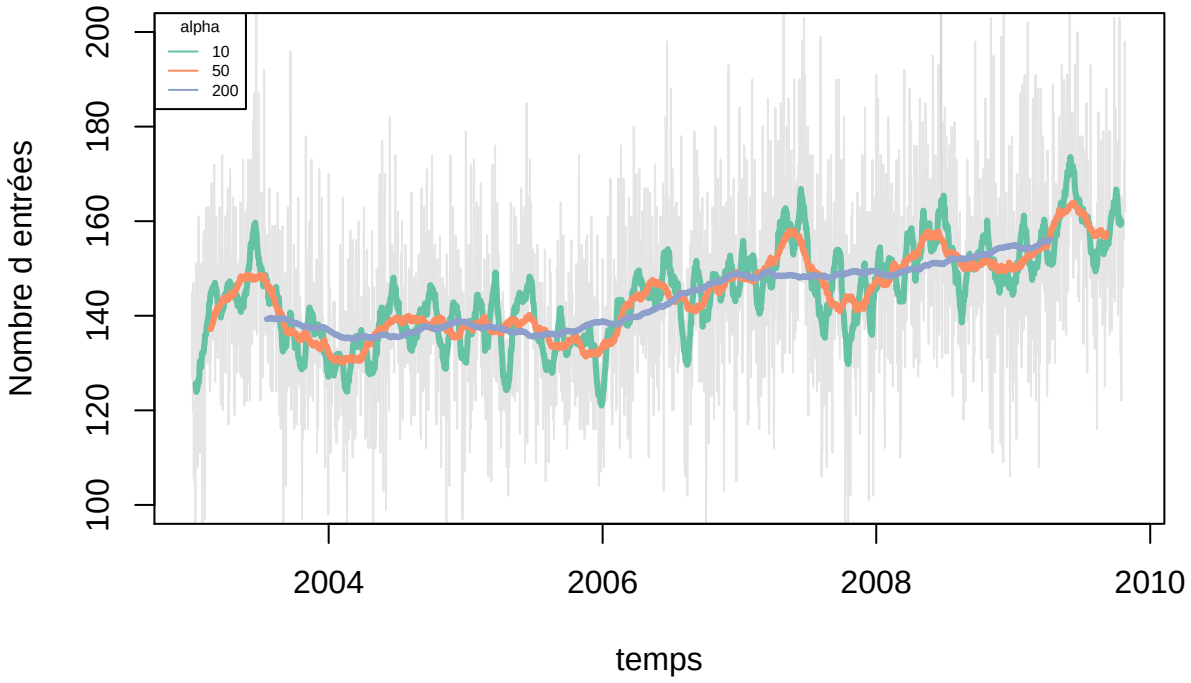
- le vecteur **valeurs_lissees** de taille $n - 2\alpha$ qui correspond au vecteur avec les valeurs lissées calculées d'après la formule ci-dessus,
- le vecteur **time_to_estimate** de taille $n - 2\alpha$ qui comprend les valeurs de **var_time** pour lesquelles on a fait le calcul des valeurs lissées.

Par ailleurs, la fonction procédéra aux vérifications suivantes :

- **var_to_smooth** et **alpha** doivent être de type **numeric**
- **alpha** doit être de longueur 1 et compris entre 1 et $n/4$
- si **var_time** est renseigné, il doit être de la même taille que **var_to_smooth** et de type **Date**

Application : vous représenterez sur un même graphique les valeurs lissées du nombre d'entrée pour différentes valeurs de α. Le résultat devra approcher la figure obtenue ci-après.

Question : Que se passe t-il quand α augmente ?



6 Régression linéaire (3 pts)

Vous allez créer une fonction *reg_lin()* qui prend comme arguments d'entrée :

- un vecteur **y** de taile n
- une matrice **x** de taille $n \times p$

Cette fonction devra retourner une liste contenant :

- l'objet **hat_beta**, un vecteur de taille p contenant le résultat du calcul $\hat{\beta} = (x'x)^{-1}(x'y)$
- l'objet **s2** contenant le résultat de $s^2 = \frac{1}{n-p}\sum_i \hat{\epsilon}_i^2$ où $\hat{\epsilon} = (y - \hat{y})$ et $\hat{y} = x\hat{\beta}$
- l'objet **t_value_beta** de taille p qui renvoit la statistique de test $\hat{\beta}_i / \hat{H}_{ii}$ où H_{ii} est le i-ème élément de la diagonale de $s^2(x'x)^{-1}$ pour i allant de 1 à p.
- l'objet **residuals** contenant le vecteur $\hat{\epsilon}$ de taille n

Elle fera les vérifications que **y** est un object de type **vector** et que **x** a le même nombre de lignes que la taille de **y**.

Application : vous appliquerez la fonction *reg_lin()* à la variable **entree** prise comme **y**. Pour la matrice **x**, on concaténera les variables **precip**, **temp**. On ajoutera à **x** une colonne contenant que des 1, une colonne contenant des 1 si **jour_semaine="lundi"**, une colonne contenant des 1 si **jour_semaine="mardi"**, une colonne contenant des 1 si **jour_semaine="mercredi"**, une colonne contenant des 1 si **jour_semaine="jeudi"**, une colonne contenant des 1 si **jour_semaine="vendredi"**, une colonne contenant des 1 si **jour_semaine="samedi"**. On enlèvera de **y** et **x** les observations pour lesquelles il y a des valeurs manquantes pour **entree**.

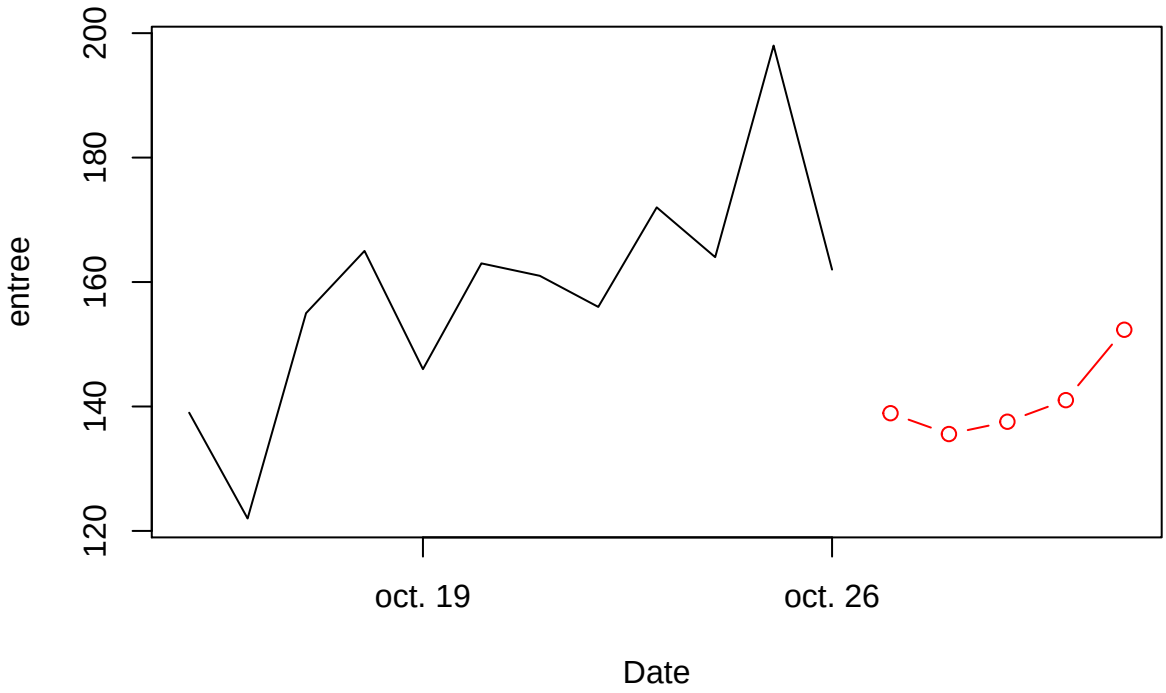
Vous comparerez vos résultats avec la commande :

```
res_lm <- lm(entree ~ precip + temp + jour_semaine,  data = projet)
summary(res_lm)
```

Prediction : vous allez prédire les 5 valeurs manquantes de **entree** en utilisant les résultats du modèle de régression linéaire précédent. Pour cela, vous ferez le calcul $X_{NA}\hat{\beta}$ où la matrice X_{NA} correpond à la matrice

x observées sur les 5 dernières observations.

Représentation: vous représenterez sur un nuage de points les observations de **entree** entre le 15 octobre 2009 et le 31 octobre 2008 en utilisant les valeurs que vous venez de calculer pour les 27, 28, 29, 30 et 31 octobre. Est-ce que les prédictions effectuées vous semblent "graphiquement" corrects ?



7 Conclusion

7.1 Conclusion (1 pts)

Faites le bilan du travail effectué dans ce projet, en gardant à l'esprit la problématique du service d'urgences (Cf. Introduction). Ensuite, donner d'autres idées d'exploration de ce jeu de données qui pourrait aider le service d'urgence à répondre à leur problématique.

Amélioration des prédictions : vous proposerez également des idées afin d'améliorer les prédictions que vous avez calculées.