Remise à niveau SAS

J. Collet

28 octobre 2021

Table des matières

Chapitre 1

Régression linéaire

Le source SAS correspondant à ce chapitre est `lineairesig.sas`.

1.1 Fonctionnalités élémentaires

1.1.1 Bibliothèques

A l'aide de la commande `libname`, définir une bibliothèque de tables pointant vers un répertoire nommé `'tables'`. A l'aide de l'interface, lister les tables de cette bibliothèque et en visualiser une.

1.1.2 Importation

Le fichier `uscrime.html` contient pour chaque État des USA, le taux de criminalité `R`, et d'autres caractéristiques de l'État (lire le fichier dans un navigateur). Il faut dans un premier lire ces données avec `SAS`, en sautant les 37 premières lignes, et en utilisant la tabulation comme séparateur (`dlm='09'x`).
Faire de même à l'aide l'interface graphique d'import des données

1.1.3 Calcul des corrélations

A l'aide de la procédure `corr`, calculer les corrélations entre le taux de criminalité et les autres variables numériques de la table. Choisir une variable pour une régression linéaire simple.

1.1.4 Régression linéaire simple

Tracer le nuage de points où les coordonnées sont le prix et la variable explicative choisie. Construire le modèle de régression linéaire. Tracer le graphique avec les observations sous forme de points, et les prévisions sous forme de lignes.

1.2 Différences entre significativité des corrélations 2 à 2 et apport d'information dans une régression

1.2.1 Données réelles

On a calculé les corrélations entre `R` et les autres variables : quelles sont les variables significativement corrélées avec `R` ?
Construisez ensuite un modèle de régression, en sélectionnant les variables par la méthode `stepwise`. Quelles sont les variables sélectionnées ? On constate en particulier que les variables `Age, U2, X` sont sélectionnées, alors qu'elles n'étaient pas significativement corrélées avec `R`. Á l'opposé, on

constate que `N` n'apparaît pas dans la régression, alors que sa corrélation avec `Y` est statistiquement significative.

1.2.2 Simulation

On veut construire un modèle de régression où `Y` est la variable expliquée, `X1` et `X2` les variables explicatives. On veut que les coefficients de `X1` et `X2` dans la régression complète soient significatifs, mais que la corrélation entre `Y` et `X1` soit non-significative..
Pour cela, on peut choisir `X1` et `X2` indépendants. Alors, si le coefficient de `X2` est nettement plus grand que celui de `X1`, l'effet de `X1` sur `Y` est indétectable sans information sur `X2`.
Pour générer les données, on utilise les éléments suivants :

- Il est possible, dans une étape `data`, d'utiliser des boucles : `do observation=1 to 100;XXX; end;`.
- Dans une étape `data`, l'instruction `output;` fait que `SAS` ajoute une ligne à la table de sortie, contenant toues les variables qu'il manipule, avec leurs valeurs courantes.
- Il y a des fonctions de tirages de nombres au hasard !

1.3 Différences entre sens des corrélations 2 à 2 et effet dans une régression

Maintenant, on veut construire un modèle de régression, avec les mêmes variables ; on veut que les coefficients de `X1` et `X2` dans la régression complète soient positifs, mais que la corrélation entre `Y` et `X1` soit significativement négative.
Pour cela, il faut que la corrélation entre `X1` et `X2` soit très proche de -1 : alors, si le coefficient de `X2` est nettement plus grand que celui de `X1`, l'effet de `X1` sur `Y`, directement et « à travers » `X2` est au total négatif.

Chapitre 2

Statistiques élémentaires

Le source SAS correspondant à ce chapitre est `td-stat.sas`.

2.1 Statistiques descriptives, tests d'adéquation à une loi

Associer à `ltables` la bibliothèque stockée dans le répertoire `'tables'`.
A l'aide de la procédure `univariate`, décrire la distribution de la variable `load` de la table `ltables.loadcal` et tester sa normalité. Tester différentes hypothèses de distribution sur les variables de la table et tracer les `QQPlot`.
A l'aide de la procédure `boxplot`, réaliser un ensemble de boîtes à moustaches relatives à la variable `load` pour chaque valeur de la variable `daytype`. On triera préalablement les données par `daytype`.
Utiliser la procédure `freq` pour calculer le tableau de fréquence des occurrences des modalités de la variable `offset`.

2.2 Corrélation

Utiliser la procédure `corr` pour déterminer les corrélations entre toutes les variables de la table `loadcal`. Tester la nullité du coefficient de corrélation dans chaque cas.

2.3 Tests de comparaison d'échantillons

A partir de la table `loadcal`, utiliser la variable `daytype` pour ne sélectionner que les jours ouvrables (c'est à dire `daytype=1`), puis la variable `wday`, égale à `weekday(date)`, pour sélectionner deux jours de la semaine à comparer (par exemple : mardi et vendredi).
Ensuite, déterminer à l'aide de la procédure `univariate` si la variable `load` suit une loi normale, pour chacun des jours de la semaine sélectionnés.
Déterminer alors à l'aide de la procédure `ttest`, si les consommations électriques moyennes sont identiques entre deux jours de la semaine.
Refaites le même test, en sélectionnant deux heures de la journée : 2 heures et 17 heures.
Refaites encore une fois le test, mais sur la consommation totale de la journée. Cela suppose d'avoir calculé au préalable cette consommation, à l'aide de la procédure `summary`.

Chapitre 3

Étude de distances de freinage

Le source SAS correspondant à ce chapitre est `freinage.sas`.

3.1 Données

Vous disposez du fichier de données `freinage.csv`, qui contient les colonnes suivantes :

Driver identifiant du conducteur

brake_number numéro de l'essai

car voiture utilisée

road_state état de la route

speed vitesse

brake_length distance de freinage

difference_with_professional_driver différence relative de distance avec un pilote professionnel

3.2 Première modélisation, utilisant des connaissances physiques

On sait que :

— le conducteur d'une voiture met un certain temps pour réagir,

— la décélération est ensuite constante,

— la décélération dépend de l'état de la route (sèche ou mouillée).

À partir de ces informations, proposez un modèle linéaire pour estimer la distance d'arrêt, en justifiant cette forme. On pourra pour cela se renseigner sur Internet (indiquer les sources consultées).
On rappelle que dans les procédures `GLM` et `GLMSelect`, le signe * signifie qu'il y a interaction entre variables. Si l'une est continue et l'autre discrète, cela signifie qu'on estimera une valeur du coefficient de la variable continue par valeur de la variable discrète.
Il faut ensuite estimer le modèle construit.

3.3 Modélisations plus complexes

Comment utiliser l'information sur le véhicule ? Plus précisément, avec quels termes du modèle construit précédemment peut-on envisager une interaction ?
On dispose aussi d'un numéro de conducteur. Que se passe-t-il si on intègre ce numéro dans le modèle ?
Enfin, on dispose aussi d'un numéro d'essai. Que se passe-t-il si on intègre ce numéro dans le modèle ? Quel sens donneriez-vous à ce résultat ?

Chapitre 4

Significativité et bootstrap

Le source SAS correspondant à ce chapitre est `bootstrap.sas`.

4.1 Régression simple

A l'aide de la commande `libname`, définir une bibliothèque de tables pointant vers le répertoire `'tables'`. A l'aide de l'interface, lister les tables de cette bibliothèque et en visualiser une.
A l'aide l'aide d'une étape `data`, créer une table `voitures` à partir du fichier texte `'input/voitures.txt'`. Visualiser le résultat.
Faire de même à l'aide l'interface graphique d'import des données.
A l'aide de la procédure `corr`, calculer les corrélations entre le prix et les autres variables numériques de la table. Choisir une variable pour une régression linéaire simple.
Tracer le nuage de points où les coordonnées sont le prix et la variable explicative choisie. Construire le modèle de régression linéaire. Tracer le graphique avec les observations sous forme de points, et les prévisions sous forme de lignes.

4.2 Étude des effets de l'échantillon

On se pose la question : et si nous n'avions eu que la moitié des observations, quel aurait été le modèle ? Aurait-il été très différent ?
Pour répondre à cette question, on va tirer au hasard des échantillons 2 fois plus petits que celui que nous avons, et étudier les modèles de régression estimés sur ces demi-échantillons.
Pour cela, on utilisera les éléments suivants.

- La procédure `surveyselect` (faite pour tirer des échantillons de sondage), en choisissant le tirage `Simple Random Sampling`, une taille d'échantillon de 9, et 100 demi-échantillons.
- **Toute** procédure `SAS` admet un paragraphe `by`, qui est un mécanisme très efficace de boucle implicite. Sia table d'entrée de la procédure est triée selon la variable discrète `toto`, alors `SAS` lit les premières lignes, jusqu'au changement de valeur de `toto`, traite alors les observations correspondant à la première valeur de `toto`, vide sa mémoire,et passe à la valeur suivante.

Tracer le faisceau des 100 droites de prévisions. On utilisera la syntaxe suivante :

```
Plot ordonnee*abcisse=classif;
```

4.2.1 Le bootstrap

Des études théoriques montrent que, pour étudier les effets du hasard d'échantillonnage, il vaut mieux faire du bootstrap : dans un échantillon de taille n, tirer de nombreux échantillons de taille n, **avec remise**.
On refera donc le travail précédent en remplaçant le tirage de demis-échantillons par des tirages d'échantillons bootstrap. Cela implique de modifier les paramètres de la procédure `surveyselect` :

on remplace le `Simple Random Sampling` par le `Unrestricted Random Sampling`, et la donnée de la taille de sous-échantillon par la donnée du taux d'échantillonnage, égale à 1.

4.2.2 Utilisation de la procédure `SQL`

Tirer n valeurs dans un échantillon de taille n est très simple : il suffit de tirer n fois un numéro d'observation entre 1 et n, pour ensuite faire la jointure entre ces numéros tirés et les observations de départ.s Il faudra néanmoins avoir auparavant mis des numéros d'observation dans dans cette table, à l'aide de la variable automatique`_n_`.

Chapitre 5

Simulation et écriture de macros

Le source SAS correspondant à ce chapitre est `khi2simu.sas`.

5.1 Simulation, vérification de la fiabilité du test du χ^2

Simuler 1000 échantillons de 100 observations deux variables `X` et `Y`, de loi discrète uniforme entre 1 et une constante `nbmod`. Ces variables étant indépendantes, quelle devrait être le résultat d'un test du χ^2 ?
Pour effectuer le test sur chacun des 1000 échantillons, on utilisera la clause `BY`. On exportera les résultats obtenus dans une table, de façon à calculer la fréquence empirique de rejet de l'hypothèse d'indépendance.

5.2 Écriture d'une macro

Transformer votre programme précédent en une macro dépendant de trois paramètres `nbech`, `nbobs` et `nbmod` afin de pouvoir faire varier facilement le nombre et la taille des échantillons.

5.2.1 Éléments de syntaxe

Macro-variable Manuellement, on crée et affecte une macro-variable :

```
%let mac=2;
%let mac2=C:\Users\Jerome\Desktop\tds\TD2011\TP8
```

On peut utiliser une macro variable dans n'importe quel contexte, en appelant `&mac`. Par exemple :

```
libname lib "&mac2";

data lib.toto; infile "‘&mac2.\donnees.txt" firstobs=&mac;
  ...; run;
```

Macro-langage Dès lors que l'on a écrit un code correct, on peut le répéter automatiquement :

```
%macro titi;
  * du code correct;
%mend;
```

Alors, dans la suite du code, écrire `%titi;` reviendra à répéter tout le code correct. Il est aussi possible de paramétrer ce code, par des macro-variables.

5.3 Affectation d'une macro-variable à partir d'une table

On veut voir le résultat du calcul du χ^2 sur l'échantillon qui semble (par hasard) le plus éloigné de l'indépendance. Pour cela, il faut récupérer dans les résultats le numéro de l'échantillon pour lequel la p-valeur du test est la plus faible.
Or, on peut aussi créer et affecter une macro-variable, **avec le contenu d'une table** :

```
data toto; set toto;
  call symputx('mac3',nom_de_variable_de_la_table);
  ... ; run;
```

Chapitre 6

Analyse de données

Le source SAS correspondant à ce chapitre est `ACP.sas`.

Le jeu de données est disponible sous la forme d'un fichier texte. Il faut donc créer une table SAS à partir de ce fichier. Le fichier est `voitures.csv`.
Le jeu de données correspond à différentes caractéristiques pour 30 modèles de voitures :

1. nom du modèle (texte)
2. puissance
3. cylindrée
4. vitesse
5. longueur
6. largeur
7. poids
8. capacité du réservoir
9. consommation
10. prix
11. origine (1=France, 2=Europe sauf France, 3=Autres)
12. équipement (1=Base, 2=Bon, 3=Très bon, 4=Sport)
13. taille coffre

Remarques :

1. La première ligne du fichier contient le nom des variables et les données sont séparées par des points-virgules (il faut donc utiliser l'option `dlm=";"` dans l'instruction `infile` ainsi que l'option `firstobs=2`).
2. Nommez la variable poids de la voiture autrement que `poids` pour pouvoir utiliser les macros ultérieurement, par exemple `poidsv`.

6.1 ACP

6.1.1 Procédures de base

A l'aide de la procédure `Princomp`, réalisez une analyse en composantes principales sur les variables continues de jeu de données (sauf la variable `prix`). On utilisera la macro `%plotit` pour tracer le premier plan factoriel.

```
proc princomp data=donnees_d_entree out=projections N=5 outstat=axes;
var liste_des_variables;
run;
%plotit(data=projections,labelvar=nom, plotvars=Prin2 Prin1,
  color=black, colors=blue);
run;
```

6.1.2 Utilisation de macros

Les sorties SAS ne sont pas très complètes. Téléchargez sur la page de P.Besse (https://www.math.univ-toulouse.fr/ besse/pub/sas/) les macros suivantes :

1. acp.sas
2. gacpvx.sas
3. gacpixm.sas

Remarque : Ces sont des fichiers `.sas.txt` mais vous pouvez les sauver tels quels. Pour pouvoir les utiliser sous SAS, il faut utiliser la commande `%include "nom_macro.sas.txt";`
Exécutez ensuite ces trois macros et étudiez les listings et graphiques de sortie. Pour cela il faut ajouter une nouvelle variable dans le jeu de données appelée `weight` qui vaut 1 pour toutes les observations.

```
%acp(dataset=donnees_d_entree, ident=nom, listev=liste_des_variables, red=,q=3,
  poids=weight);
%gacpvx();/*cercle des correlations*/
%gacpixm(x=1,y=2,coeff=1);/*plans factoriels*/
```

Trouver comment représenter les variables illustratives suivantes :
- prix (quantitative)
- equipement et origine (qualitatives)

6.2 Classification ascendante hiérarchique

A l'aide de la procédure `cluster`, effectuez une classification ascendante hiérarchique par la méthode de Ward sur la table `voitures` en utilisant les mêmes variables continues que pour l'ACP.

```
proc cluster data=donnees_d_entree method=ward outtree=sortiecah standard;
  id nom;
var liste_des_variables;
run;
```

On pourra retraiter les sorties avec la procédure `tree`.

Chapitre 7

Réduction de dimension pour l'étude des naissances en France

Le source SAS correspondant à ce chapitre est `naissances.sas`.

Le but de ce TP est de pratiquer l'analyse des données avec `SAS`, alors que ce n'est vraiment pas l'outil idéal pour ce type d'analyses. Par conséquent, l'approche utilisée est un peu particulière, les composantes principales et les projections sur les axes principaux sont représentés de manière inhabituelle, mais assez utile dans de nombreux cas de séries temporelles.
Ce sujet est inspiré d'un TP sur les naissances en France, par R. Lobry. Ici, on propose en plus un exercice utilisant l'Analyse en Composantes Principales (ACP).
Une première question apparaît : faut-il, pour chaque mois, étudier le nombre moyen de naissances par jour ou le nombre total ? Et les années bissextiles ?
On peut ensuite dessiner les naissances par mois, pour plusieurs années à la fois, ou bien le nombre annuel de naissances.
On considère ensuite chaque année comme une observation, et les nombres de naissances mensuelles comme des variables, et on effectue une ACP.
L'ACP commence par calculer le point moyen, comment le représenter ?
Les composantes sont des jeux de coefficients, pour chaque mois, on peut donc les représenter graphiquement comme un profil. Que constate-t-on ?
Par ailleurs, les projections sur chaque composante dépendant de l'année, on peut donc utiliser cette dimension temporelle pour représenter ces projections. Que constate-t-on ?
On peut aussi comparer les résultats annuels issus de l'ACP avec ceux obtenus quand on calculait le nombre annuel de naissances.

Chapitre 8

Prévision de consommation d'électricité

Le source SAS correspondant à ce chapitre est `gesdon.sas`.

8.1 Manipulation de données, jointures

Nous disposons de plusieurs fichiers de données :

- des données de consommation, sur une ligne comprenant 48 valeurs par jour, fichier nommé `historique_consommation_puissance_2003.txt`,
- des données de température, sur une ligne comprenant 8 valeurs par jour, fichier nommé `TREA03L_V1_040419.txt`,
- des données sur les jours de fête et les jours EJP (une colonne pour chaque type de jour spécial), seuls les jours spéciaux sont mentionnés, fichier `speciauxR.txt`.

Il faut mettre les données en ligne sur une colonne unique, interpoler les températures, combler le trou dans la consommation dû au changement d'heure, fusionner les tables, les fusionner aussi avec la table des jours spéciaux.

8.2 Modélisation

On utilisera la procédure `GLM`, qui permet de construire des modèles linéaires, avec des interactions entre variables continues, entre variables discrètes, et aussi entre variables continues et discrètes.

8.2.1 Construction des variables explicatives, linéarisation du modèle

Nous devons modéliser deux influences qui paraissent incompatibles, à première vue, avec un modèle linéaire : l'influence de la position dans l'année et celle de la température.
Dans le premier cas, nous utiliserons un résultat théorique : toute fonction périodique peut être approchée par une combinaison **linéaire** de fonctions trigonométriques (voir un livre sur l'analyse de Fourier pour un énoncé plus précis). Nous utiliserons donc ces fonctions trigonométriques comme variables explicatives.
Dans le second cas, nous utiliserons des connaissances physiques sur le phénomène à étudier : les chauffages électriques sont allumés quand la température extérieure est inférieure à 15 degrés. De plus, dans ce cas, les fuites thermiques étant proportionnelles à la différence entre l'intérieur et l'extérieur, la puissance consommée croît linéairement quand la température extérieure décroît.

8.2.2 Mise en œuvre

Il faudra créer les variables suivantes :

— des fonctions $\sin(2\pi\text{Jour})$ et $\cos(2\pi\text{Jour})$, pour modéliser l'influence de la position dans l'année sur la consommation,
— des fonctions $\max(0, 15 - t)$ et $\max(0, t - 18)$, pour modéliser l'impact du chauffage et de la climatisation.
— un type de jour, valant 0 pour les fêtes, et de 1 à 7 pour les autres jours.

Ensuite, il faudra réduire la taille de l'ensemble de données, en ne gardant que les demies-heures impaires.

On pourra alors construire un modèle, avec des interactions entre les fonctions trigonométriques, le type de jour et le chauffage d'une part, la demie-heure d'autre part. En revanche, la climatisation sera supposée indépendante des autres variables.

On constatera alors que l'erreur du modèle est assez importante : on éliminera successivement la période de Noël, la première quinzaine d'août, les jours EJP.

Chapitre 9

Vols à NYC

Le source SAS correspondant à ce chapitre est `NYC.sas`.

9.1 But de l'étude

On veut modéliser la durée d'un vol. Par exemple, on veut savoir si le temps de décollage dépend de la compagnie, ou uniquement des aéroports de départ et d'arrivée. On se demande aussi si la vitesse dépend du modèle, ou si la donnée du type de moteur suffit à prévoir la vitesse d'un vol.
Pour ces questions, on dispose de trois tables, qu'il va falloir fusionner. La première est nommée `flights` est une table de vols aux USA, comprenant :

year, month, day la date de départ du vol,

dep_time, sched_dep_time, arr_time, sched_arr_time les heures de départ et d'arrivée, prévues et réalisées, en minutes,

carrier, flight, tailnum les identifiants de la compagnie, du vol et de l'avion,

origin, dest les identifiants ds aéroports de départ et d'arrivée,

air_time le temps de vol en minutes.

La seconde est nommée `distances`, et donne, pour chaque couple d'origine et de destination, la distance en miles.
La troisième, nommée `planes`, comprend :

tailnum l'identifiant de l'avion,

yearplane type manufacturer model engines seats engine l'année de mise en service, le type d'ailes, le fabricant, le modèle, le nombre de moteurs, le nombre de sièges, le type de moteur.

Remarque Ce TP est largement inspiré de R for Data Science.

9.2 Modélisations

Pour cela, une première étape sera de faire le lien entre les vols, les caractéristiques des avions ayant fait ces vols et la longueur de ces vols, à l'aide d'étapes `MERGE` dans des étapes `DATA`.
On utilisera ensuite une proc `GLMSELECT` pour modéliser le temps de vol. Cette modélisation fait apparaître des coefficients liés à la distance parcourue, qui permettent donc de déduire des vitesses.
On tentera, dans une seconde étape, de déterminer les vitesses de vol moyennes pour chacun des types de moteurs utilisés. Il sera nécessaire de supprimer les observations où le moteur est `4 Cycle`, car les temps de vol ont manifestement été mal mesurées, et celles le moteur est `Turbo-shaft`, qui sont des vols d'hélicoptères. On retrouve alors des vitesses de vol de l'ordre de 800km/h.

Types de moteurs utilisés :

Reciprocating moteur à pistons, entraînant une hélice,

Turbo-fan turboréacteur double flux,

Turbo-jet turboréacteur

Turbo-prop turbopropulseur, turbine entraînant une hélice.

9.3 Compléments

Par ailleurs, on constate que la table des avions contient de nombreuses informations répétées : toute la description de l'avion est répétée pour les avions du même modèle.
On construira donc deux tables : une table de modèles, avec toutes les caractéristique techniques, et une table d'exemplaires, ne comprenannt que l'immatriculation et l'année de mie en service.
On montrera qu'un `MERGE` de ces deux tables permet de reconstituer la table `planes`.