Modèle de comptage sur les étoiles de l'Euromillion

Axel-Cleris Gailloty

Hamet Niang

2021-01-17

Sommaire

1 Introduction

Nous travaillons sur les étoiles du loto dans le cadre d'un modèle de comptage. Le but est de créer une situation où nous pouvons compter la combinaison des valeurs que peuvent prendre les deux étoiles. C'est en fait à nous de créer la variable endogène à partir des deux étoiles.

1.1 Description de l'Euromillion

L'EuroMillions est une loterie organisée à travers neuf pays européens. Les tirages ont lieu les mardis et vendredis soir avec un gain minimum de 17 millions d'euros, qui peut aller jusqu'à la somme impressionante de 210 millions d'euros [1].

Le jeu consiste à choisir 5 numéros entre 1 et 50 ainsi que 2 étoiles numérotées de 1 à 12, avec deux tirages par semaine.

1.2 Description des étoiles

Les étoiles sont une variable aléatoire discrète contenue entre 1 à 12. Voici ce que nous aimerions modéliser dans le cadre de ce projet : **Combien de fois la somme des étoiles est supérieure à la valeur de chaque boule tirée individuellement?**

La base de données contient 53 colonnes et 175 observations. Nous affichons certaines colonnes de la base de données notamment les colonnes qu contiennent les 5 boules et celles qui contiennent les 2 étoiles.

[1] Définition officielle d'Euromillion https://www.euro-millions.com/fr/

B1	B2	B3	B4	B5	E1	E2
9	19	26	31	6	12	11
16	10	46	39	6	8	11
40	17	24	19	18	8	4
5	13	7	19	31	9	2
46	24	42	15	3	12	9
9	6	1	47	34	12	7
17	43	26	4	30	11	6
47	43	19	23	12	2	6
13	23	26	47	32	6	10
9	26	16	2	36	6	7

Il convient maintenant de créer notre variable endogène. Nous faisons cela en créant une fonction qui prend en compte la règle que nous avons fixée plus haut et génère l'endogène.

```
DB$sommeEtoile <- DB$E1 + DB$E2
Boules <- strsplit(DB$boules_gagnantes_en_ordre_croissant, "-")

countMoreThan <- function(x, arr) {
  count <- 0
  for (nb in arr[2:6]) {
    if (x > as.integer(nb)) {
      count <- count + 1
    }
  }
  return(count)
}

nombreTirage <- nrow(DB)
Y <- vector(mode = "numeric", length = nombreTirage)
for (i in seq_len(nombreTirage)) {
  Y[i] <- countMoreThan(DB$sommeEtoile[i], Boules[[i]])
}
```

2 Modèle de comptage

Dans un modèle de comptage la variable endogène y prend un petit nombre de valeur positives. y est en réalité le résultat d'un comptage, c'est le nombre d'occurrence d'un évènement. Les évènements peuvent être :

- Le nombre de téléviseurs dans une maison
- Le nombre d'enfants d'un couple
- Le nombre d'élèves par classe
- Le nombre de voyages en avion sur une année pour une personne
- Le nombre de buts marqués dans un championnat
- Le nombre de médailles par nation
- Le nombre de voitures d'un ménage

Dans notre cas l'évènement que nous étudions est le nombre de fois que la somme des étoiles soit supérieure à une boule.

2.1 La régression de Poisson

La régression de Poisson est un modèle de prédiction qui s'applique lorsque la variable cible y est une variable de comptage (nombre d'apparition d'un évènement durant un laps de temps).

$$P(Y = y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

Avec :

- Y une variable aléatoire de Poisson.
- $y = 1,2,3,4,\ldots$, un entier naturel.

- λ le nombre d'occurence de l'évènement dans un intervalle, nombre réel strictement positif.

Dans notre étude cela implique que nous allons devoir estimer le modèle suivant (appelé modèle de régression de Poisson qui utilise la fonction génératrice des moments de la loi de poisson) de la forme :

$$E[Y] = \lambda = exp(cst + \beta X)$$

On estime cst et β par la méthode du maximum de vraisemblance, on en déduit $\hat{\lambda}$:

$$\widehat{E[Y]} = \hat{\lambda} = exp(\widehat{cst} + \hat{\beta} X)$$

On en déduit aussi la probabilité du nombre d'occurrences :

$$P(\widehat{Y = y}) = \frac{\hat{\lambda}^y e^{\widehat{-\lambda}}}{y!}$$

3 Application

Nous appliquons maintenant cette procédure que nous venons de décrire à notre travail. Nous voulons modéliser le nombre de fois que la somme des étoiles est supérieure à chacune des valeurs des boules B1 à B5.

y qui représente le nombre d'occurences est compris entre 0 et 5. Lorsque le nombre est égal à 0 alors toutes les boules tirées sont supérieure à 24. Lorsque $y = 1$ alors une boule tirée est inférieure à 24 et ainsi de suite jusqu'à ce que $y = 5$ qui signifie que toutes les boules tirées sont inférieures à 24.

Nous affichons une distribution d'occurence de ces évènements.

```
table(Y)
```

```
Y
 0  1  2  3  4
49 65 42 18  1
```

Selon ce tableau, dans 49 cas sur 175 toutes les boules tirées sont supérieures à la somme des étoiles, c'est-à-dire supérieures à 24. Dans 65 cas une boule tirée est inférieure à 24, dans 42 cas deux boules tirées sont inférieures à 24, dans 18, trois boules tirées sont inférieures à 24 et dans un seul cas seulement 4 boules tirées sont inférieures à 24. Aucun cas ne mentionne le fait que toutes les 5 boules tirées sont inférieures à 24. C'est sur ce comptage que nous allons faire notre modélisation.

Mais avant de passer à la modélisation, nous estimons qu'il est important de choisir des variables exogènes au modèle, de présenter et de faire des statistiques descriptives sur ces variables exogènes en lien avec la distribution de la variable endogène.

Comme nous l'avons décrit au début de cette étude, le jeu de donnée contient 54 colonnes. On pourrait inclure toutes ces variables dans le modèle, nous avons assez de dégré de liberté de le faire puisque le nombre d'observation est de 175 soit plus de trois fois le nombre

de variables. Toutefois nous cherchons un modèle simple à interpréter. Aussi inclure beaucoup trop de variables dans un modèles risque d'introduire des biais dans l'analyse. Parmi ces biais nous pouvons compter :

- La multicolinéarité : c'est lorsque plusieurs variables prédisent la même chose. c'est lorsque ces variables sont très corrélés voire parfaitement corrélées. Un exemple c'est le jour du tirage. Le tirage de l'Euromillion se fait deux fois par semaine : le mardi et le vendredi. Si nous incluons dans le modèle le jour de tirage alors vendredi est un prédicteur parfait de mardi.
- Le biais de surapprentissage (overfitting) : c'est lorsque le modèle apprend trop les caractéristiques particulières du présent échantillon si bien qu'il est difficile de généraliser le résultat à d'autres échantillons.

3.1 Les hypothèses de recherche

Il convient de présenter les hypothèses que nous aimerions tester ici.

- Le nombre de gagnants au range 12 millions d'euros n'a pas d'effets sur la somme des étoiles.
- La somme de toutes les boules a un effet négatif sur le nombre d'occurence.
- Le nombre de boules jouées n'a pas d'effet sur le nombre d'occurences
- Il y a plus de grilles jouées les mardis que les vendredis
- Lorsque la première étoile est supérieure à la deuxième alors toute chose égale par ailleurs le nombre d'occurences est supérieure à 0.

3.1.1 Choix des variables

Pour ces raisons voici la liste des variables que nous choisissons :

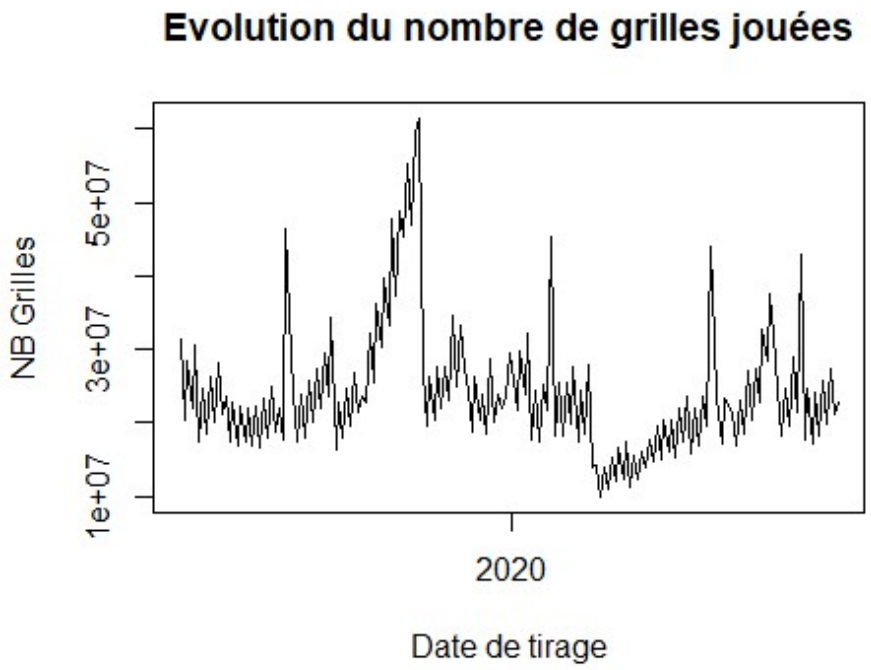
- Somme des boules B1 à B5
- Nombre de gagnants
- Nombre de boules jouées

- Première boule supérieure à la deuxième boule (variable binaire indicatrice)
- Le jour du tirage est mardi (variable binaire)
- Le mois de Septembre (variable binaire)
- Le jour du mois

3.2 Statistiques et visualisations

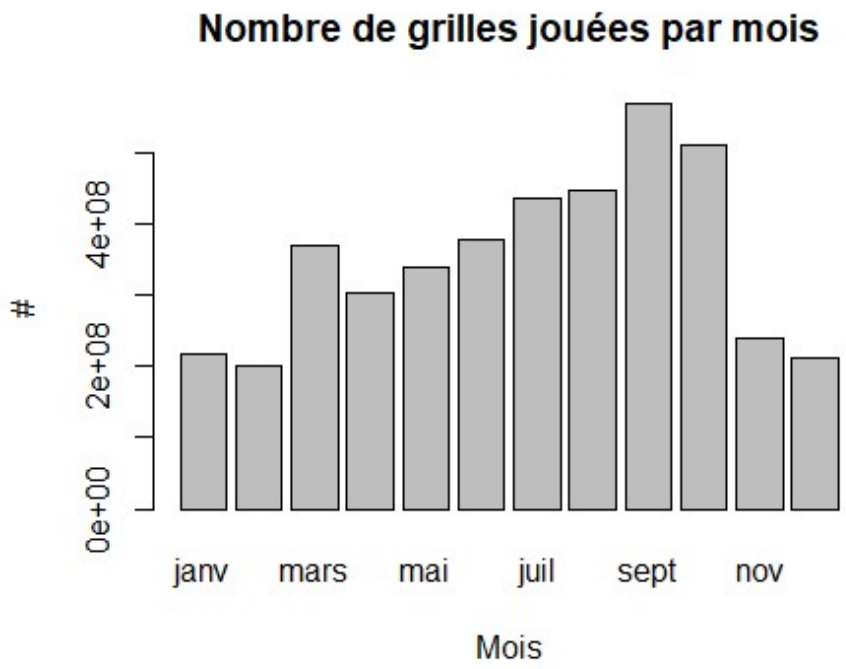
3.2.1 Evolution des grilles jouées

```
plot(x = DB$date_de_tirage, DB$nbGrillesJouees, main = "Evolution du nombre
de grilles jouées", type = "l", xlab = "Date de tirage", ylab = "NB Grilles")
```



Le nombre de grilles jouées dépend des périodes de l'années.

```
barplot(xtabs(nbGrillesJouees~Mois_fct, data = DB), main = "Nombre de grilles
jouées par mois", xlab = "Mois", ylab = "#")
```



Il y a beaucoup de grilles jouées en septembre et en octobre. Nous allons inclure dans le modèle La variable septembre.

```
t.test(nbGrillesJouees~Mardi, data = DB)


    Welch Two Sample t-test

data:  nbGrillesJouees by Mardi
t = 5.2249, df = 171.85, p-value = 4.999e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 3976947 8806108
sample estimates:
mean in group 0 mean in group 1
       27337378        20945850
```

Le test `t-test` de Welsh indique qu'il existe une différence significative entre les moyennes de grilles jouées mardi ou vendredi. Il y a plus de grilles qui sont jouées vendredi que mardi.

4 Estimation de régression de Poisson avec glm

Le langage de programmation statistique R vient avec des procédures statistiques et économétriques déjà implémentées. Il est donc facile d'estimer une grande gamme de méthodes statistiques et économétriques sans avoir à dépendre d'un package extérieur ou même sans avoir à les implémenter nous-même.

Pour estimer une régression de Poisson nous pouvons utiliser la fonction `glm` (Generalized Linear Models). glm est utilisé pour ajuster des modèles linéaires généralisés, spécifiés en donnant une description symbolique du prédicteur linéaire et une description de la distribution d'erreur.

```
reg <- glm(Y
~sommeBoules+nbGagnants+nbGrillesJouees+Etoile1Sup+Mardi+Sept+Jour , data =
DB, family = "poisson")
summary(reg)


Call:
glm(formula = Y ~ sommeBoules + nbGagnants + nbGrillesJouees +
    Etoile1Sup + Mardi + Sept + Jour, family = "poisson", data = DB)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.72107  -0.81974  -0.00419   0.40494   1.66203

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      3.042e+00  4.715e-01   6.451 1.11e-10 ***
sommeBoules     -2.337e-02  2.934e-03  -7.965 1.65e-15 ***
nbGagnants      -1.188e-05  4.248e-06  -2.796  0.00518 **
nbGrillesJouees  5.522e-08  2.079e-08   2.655  0.00792 **
Etoile1Sup      -4.237e-02  1.417e-01  -0.299  0.76493
Mardi            7.579e-03  1.554e-01   0.049  0.96110
Sept             8.874e-02  2.709e-01   0.328  0.74322
Jour            -1.494e-02  8.166e-03  -1.829  0.06735 .
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 176.67  on 174  degrees of freedom
Residual deviance: 108.28  on 167  degrees of freedom
AIC: 421.18

Number of Fisher Scoring iterations: 5
```

4.1 Interprétation des coefficients

4.1.1 Significativité des coefficients estimés

La fonction `summary` de R opère divers calculs sur les coefficients estimés. Parmi ces calculs il y a le test de significativité des coefficients déduit à partir de la statistique `z`, nous avons donc `Pr(> |z|)` qui indique si le coefficient est significatif à quel seuil. Sur l'ensemble des variables que nous avons sélectionnées , la somme des boules, le nombre de gagnants et le nombre de grilles jouées sont significatives au seuil de 5%. Le jour n'est significative que si on étend le seuil de significativité à 10%.

Le fait que le tirage ait lieu un mardi, que la grille soit achetée en Septembre ou la première étoile soit supérieure à la seconde n'ont aucune significativité statistique.

4.2 Interprétation des coefficients

Pour une variable binaire X_j, le coefficient $\hat{a}_j$ représente l'écart des logarithmes des nombres espérés (λ) selon l'apparition ($X_j = 1$) ou pas ($X_j = 0$) de la caractéristique.

$$\widehat{a_j} = ln\hat{\lambda}_{(X_j=1)} - ln\hat{\lambda}_{(X_j=0)}$$

Puisque

$$\hat{a} = ln\widehat{\lambda_1} - ln\widehat{\lambda_0} = ln\frac{\lambda_1}{\lambda_0}$$

$$e^{\widehat{a_j}} = \frac{\lambda_1}{\lambda_0}$$

L'interprétation des coefficients estimés pour les prédicteurs de catégorie est faite par rapport au niveau de référence du prédicteur. Des coefficients positifs indiquent que l'événement a plus de chances d'avoir lieu à ce niveau du prédicteur qu'au niveau de référence du facteur. Des coefficients négatifs indiquent que l'événement a moins de chances d'avoir lieu à ce niveau du prédicteur qu'au niveau de référence.

Construction du modèle à estimer en prenant compte les coefficients coefficients estimés

$$E[Y] = \lambda$$

$= exp(2.67 - 0.023 *$ SommeBoules $- 1.293e - 05 *$ nbGagnants $+ 6.359e - 08$
$*$ nbGrillesJouees $- 4.237e - 02 *$ Etoile1Sup $+ 7.579e - 03 * Mardi + 8.874e - 02$
$* Sept - 1.494e - 02 * Jour)$

Selon le modèle que nous estimé nous voyons que la somme de sboules a un effet négatif sur le nombre d'occurence des boules inférieures à 24.

Nous résumons dans le tableau suivant les effets de chaque endogène sur le nombre d'occurences de boules tirées inférieures à 24.

Variable	Signe
Somme des Boules	-
Nb gagnants	-
Nb grilles jouées	+
Etoile 1 > Etoile 2	-
Mardi	+
Septembre	-
Jour	-

Par exemple : La somme de toutes les boules est égale à 90, le nombre de gagnants est 69935, nombre de grilles jouées 23847582 et la première boule est supérieure à la deuxième.

```
l = exp(2.67 - 0.023 * 90 -1.293e-05 * 69935 + 6.359e-08 * 23847582 -6.842e-03 * 1)
```

4.2.1 Calcul du pseudo-R2

Le pseudo-R2 nous donne une idée de la qualité d'ajustement du modèle.

```
pseudoR2 <- function(y, y_hat) {
rd <- ifelse(y==0,0,y*log(y/y_hat))-(y-y_hat)
DS = 2 * sum(rd)
a0 <- log(mean(y))

rd0 <- ifelse(y==0,0,y *log(y)) - y * a0 - (y - exp(a0))
D0 <- 2 * sum(rd0)

R2 <- (D0-DS)/D0
return(R2)
}

pseudoR2(DB$y, reg$fitted.values)
```

```
[1] 0.3870637
```

Le pseudo-R2 est de 38%, ce qui indique que le modèle explique environ 38% de la variance du nombre de boules tirées qui sont inférieures à 24.

4.3 Estimation des probabilités d'occurence

Maintenant que nous avons estimé le modèle, nous pouvons estimer $\hat{\lambda}$ pour calculer les probabilités d'occurence du nombre de fois que les boules tirées sont inférieures à 24.

$$\widehat{P(Y = y)} = \frac{\hat{\lambda}^y e^{-\hat{\lambda}}}{y!}$$

Nous savons que $\hat{\lambda}$ est égale à $E[Y]$. A l'aide du modèle nous estimons $\hat{\lambda}$ comme suit :

$$\widehat{E[Y]} = \hat{\lambda} = exp(\widehat{cst} + \hat{\beta}X)$$

Pour estimer le λ il nous faut trouver les paramètres du tirage. Nous allons donc supposer que le tirage a eu lieu un mardi 15 Spetembre. La somme des boules tirées est égale à 90 et le nombre de gagnants est ce jour est de 120000, que le nombre de grilles tirées est de 25000000, que la première étoile est supérieure est à la seconde.

Nous utilisons ces paramètres dans les coefficients estimées

```
lambda = exp(3.041727 -2.337295e-02 * 90 -1.187523e-05 * 120000 + 5.521722e-
08 * 25000000 + 7.578674e-03*1 + 8.873867e-02*1 -1.493891e-02 * 15)
```

La valeur de $\hat{\lambda}$ est 2.151. Nous allons donc définir une fonction pour calculer les probabilités du nombre d'occurence.

```
Prob <- function(y, l) {
  (exp(-l) * l**y) / factorial(y)
}
```

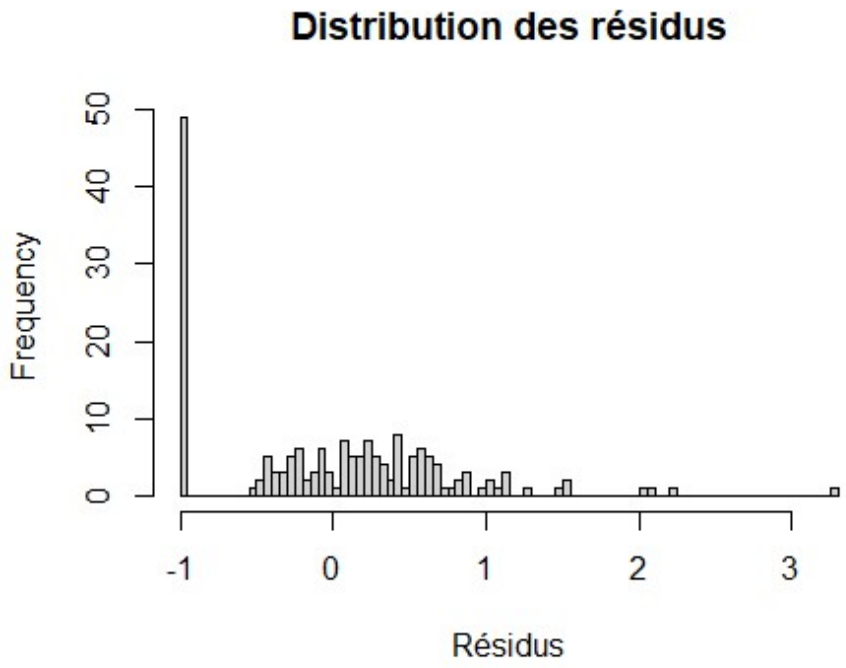
Dans le tableau suivant nous affichons les probabilités que le nombre d'occurence soit égal à 0, 1, 2, 3 et 4.

Occurence	Probs
NB = 0	0.1164015
NB = 1	0.2503458
NB = 2	0.2692106
NB = 3	0.1929980
NB = 4	0.1037707

5 Evaluation de l'estimation

5.1 La distribution des résidus

```
hist(reg$residuals, main = "Distribution des résidus", xlab = "Résidus",
breaks = 100)
```



Nous n'avons pas une hypothèse particulière sur la distribution ds résidus pour l'évaluer. Mais à première vue cette distribution ne suit pas une loi normale.

5.1.1 Root Mean Square Percent Error (RMSPE)

Nous pouvons calculer un indicateur pour estimer la fiabilité de la prévision. Cet indicateur est le Root Mean Square Percent Error, c'est l'erreur de pourcentage quadratique moyen. Il permet de calculer le pourcentage moyen de déviation entre la valeur observée et la valeur prédite.

```
RMSPE <- function(y, y_hat) {
  N <- length(y)
  yi_y_hat <- ((y - y_hat)/y)^2

  y2 <- ifelse(yi_y_hat == Inf, 0, yi_y_hat)

  sum((1/N)*y2)
}

RMSPE(DB$y, reg$fitted.values)
```

```
[1] 0.114778
```

En moyenne, la valeur prédite dévie de la valeur observée de 11.5 %.

5.1.2 Matrice de confusion

Bien que le RMSPE permette d'estimer la déviation moyenne entre la valeur prédite et la valeur observée, nous remarquons que cet indicateur est utilisée pour des variables continues alors que manifestement une régression de Poisson concerne des valeurs discrètes. Nous pouvons donc calculer une matrice de confusion pour nous rendre compte de la performance du modèle.

En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée.

```
table(DB$y, round(reg$fitted.values))
```

```
    0  1  2  3  4  5
0 21 28  0  0  0  0
1  3 51 11  0  0  0
2  1 23 15  3  0  0
3  0  3  9  3  2  1
4  0  0  0  0  1  0
```

6 Conclusion

Ce travail réalisé dans le cadre du cours d'économétrie des variables qualitatives a pour intérêt de nous avoir ouvert des perspective sur l'estimation économétrique sur des variables discrètes. Les domaines d'application d'un modèle de comptage sont nombreux. Dans les évènements sportifs comme des matchs de Football plusieurs pronostiqueurs utilisent des modèles économétriques comme la régression de Poisson pour déviner l'issue d'un match.

Dans ce travail nous avons eu l'audace de faire une régression de Poisson sur des données de l'Euromillion. Or nous savons que l'Euromillion est un jeu de hazard, il est en cela impossible de prédire l'issue d'un tirage, mais la façon à laquelle nous avons formé notre variable endogène nous permet de remarquer quelques détails intéressants que nous avons découverts :

- La somme des boules a un effet négatif statistiquement significatif sur la probabilité que les boules tirées individuellement soient inférieures à 24.
- Encore plus surprenant nous trouvons que le nombre de gagnants et le nombre de grilles jouées sont toutes statistiquement significatives sur la probabilité que la somme des étoiles, c'est-à-dire 24, soit supérieure à chaque boule tirée individuellement; après des réflexions nous n'avons pas pu trouver une explication cohérente à ce phénomène.
- Le jour du mois où les tirages sont réalisés semblent avoir un effet significatif au seuil de 10% sur la probabilité que la somme des étoiles soit supérieure à chaque boule prise individuellement.

Au regard des résultats trouvés pouvons nous dire qu'un plus grand échantillon de données dans lequel se trouve des variables telles la météo, le climat et certaines variables économiques peuvent influencer sur les résultats du loto ? Peut-on dire qu'il est possible de modéliser l'aléatoire ?

7 Bibliographie

Hastie, T. J. and Pregibon, D. (1992) Generalized linear models. Chapter 6 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ricco Rakotomalala. (2010, 1 janvier). Régression de Poisson - Modèle de comptage. Laboratoire ERIC, Unité de Recherche Universitaire. http://eric.univ-lyon2.fr/~ricco/cours/slides/regression_poisson.pdf