**BERGUIGA**
**Oussama**

# UNIVERSITE PARIS DAUPHINE – MASTER STATISTIQUE ET BIG DATA
# MODELE DE SURVIE
# CHALLENGE KAGGLE – PREVISION DU TEMPS DE TRAJET DES TAXIS DE PORTO


kaggle

Taxi
Service
Trip Time
Prediction Challenge @ ECML PKDD 2015

# Table des matières

# INTRODUCTION

Ce projet est issu du concours Kaggle 2015 « Taxi Trip Time prediction ». Il consiste en la prévision du temps de trajet des courses de Taxi de la ville de Porto en 2015. La problématique peut être résolue par la théorie « Modèle de Survie », et en particulier le modèle de Cox. La démarche consiste à trouver le meilleur modèle appris sur les données « Train », puis prédire le temps médian de la course pour la centaine d'observations des données « Test ».

# I) DONNEES

## 1) Présentation des différentes covariables

Le jeu de données est constitué de 200 000 lignes qui comportent les informations suivantes :

- CALL_TYPE : variable catégorielle, correspond au type de commande de la course (A=commande par le central, B=commande au taxi directement, C=autre)
- wday : correspondant au jour de la semaine de la course (du lundi à dimanche, de 0 à 6)
- hour : entier correspond à la plage horaire de la course (de 0 à 23)
- d_st : nombre réel, correspond à la distance du centre-ville de la prise en charge de la course
- heading : nombres réels correspond à l'angle de direction de la course
- xs,ys : nombres réels correspond aux coordonnées du point de prise en charge de la course
- xe,ye : nombres réels correspond au coordonnées du point d'arrêt de la course
- len : entier correspond aux nombres de balises GPS durant la course
- Missing_Data : variable catégorielle qui vaut « vrai » si des données GPS n'ont pu être captées, « faux » sinon

## 2) Valeurs Manquantes

Le jeu de données comporte une partie des lignes dont toutes les valeurs sont « -1 ». Nous interprétons ces lignes comme des données manquantes (qui représentent environ 2% des cellules du jeu de données, donc négligeable). Nous choisissons donc de supprimer ces lignes.

## 3) Temps de trajet

Il est écoulé 15 secondes entre 2 bornes GPS. Nous en déduisons donc le temps de trajet :

$$Temps\ de\ trajet\ =\ 15 * (len - 1)$$

## 4) Censure

Soit T la durée et C une variable aléatoire, on dit que C censure T à droite lorsque :

$$T_C = \min(T, C)\ et\ \delta = 1_{T<=C}$$

Dans ce jeu de données, la variable MISSING_Data vaut « VRAI » si le taxi n'a pu capter des données GPS (passage sous un tunnel) et donc le temps prédit est censuré. Dans notre jeu de données « Train », il n'y a qu'une donnée censurée de cette manière.

Cependant, il apparaît que des temps de trajet sont aberrants (par exemple des temps de trajet à 58 060 secondes) qui doivent probablement correspondre à des erreurs de mesure. Nous choisissons dans la suite :

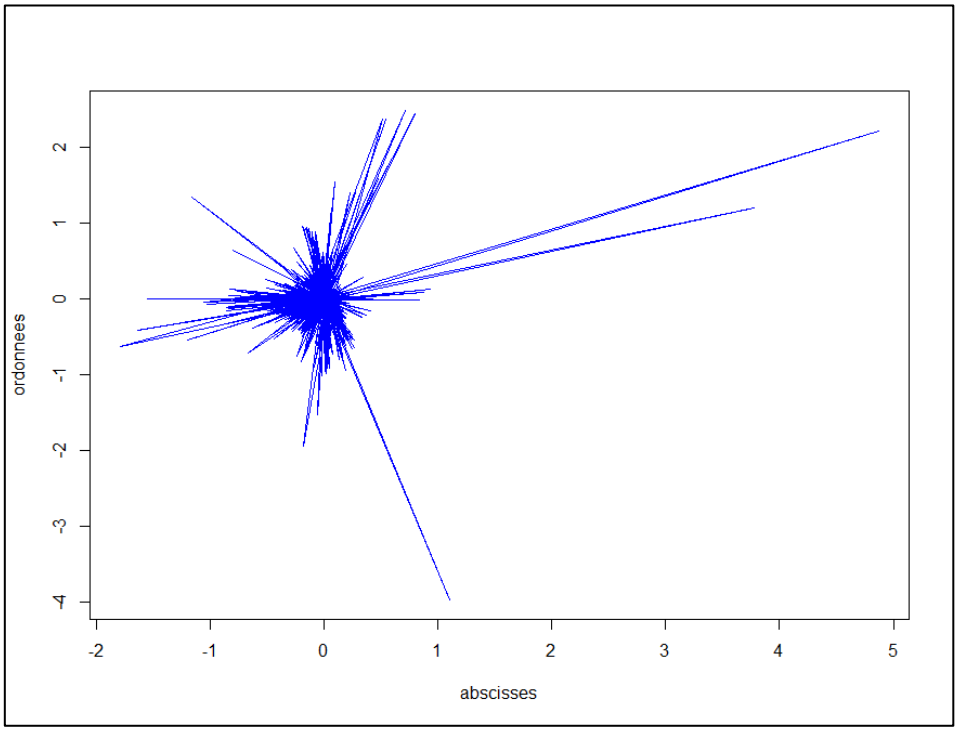
- De déterminer un temps critique Tc égal au quantile 90% des temps de trajet (ici 1545 secondes, soit environ 25 min).
- Puis nous forçons la censure (delta=0) pour tous les temps strictement supérieurs à Tc, et enfin nous forçons le temps de trajet = Tc pour tous les temps strictement supérieurs à Tc.

## 5) Exploration des données « Train »

Suivant le code R et insertion des graphiques

- Trajectoires des différentes observations :

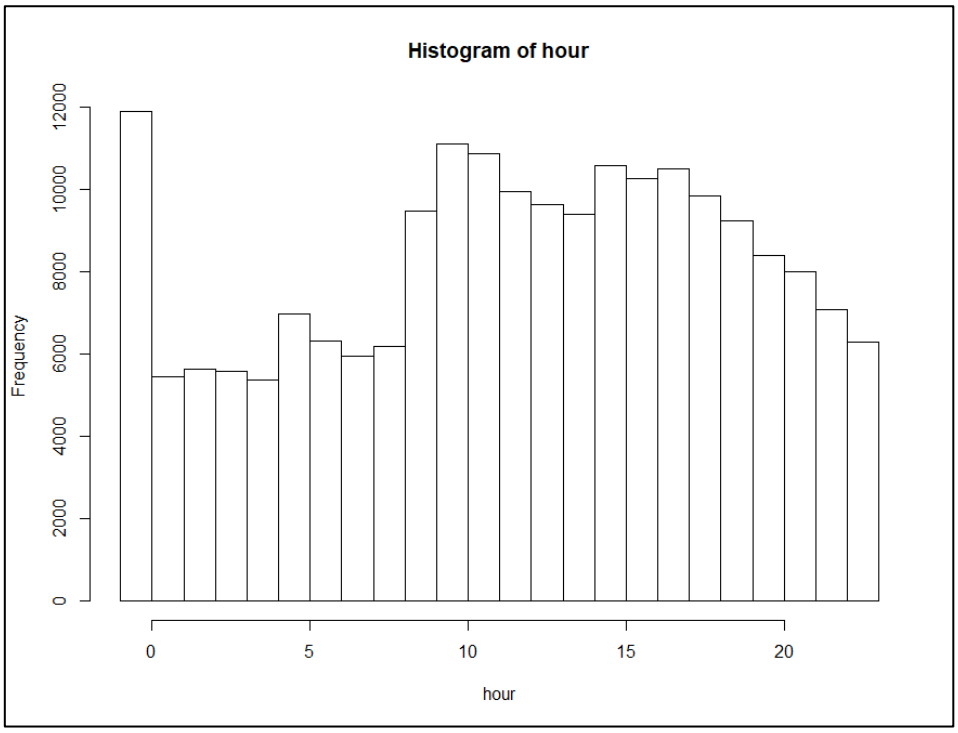
*Fig.1 : trajectoires des courses*




On remarque que les trajets sont très concentrés vers le centre de Porto.

- Influence de l'heure de la journée :

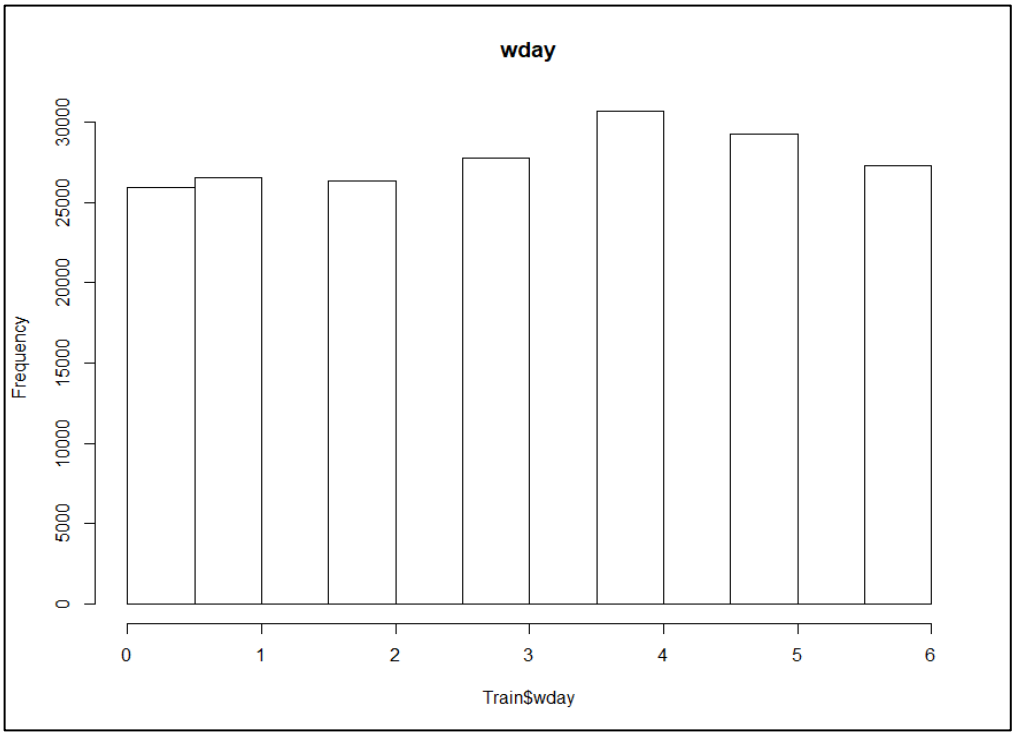
*Fig.2 : Histogramme de la répartition des courses par tranche horaire*




On voit sur ce graphique qu'il y a plus de courses le jour (entre 9h et 23h) que la nuit (entre 0h et 8h).

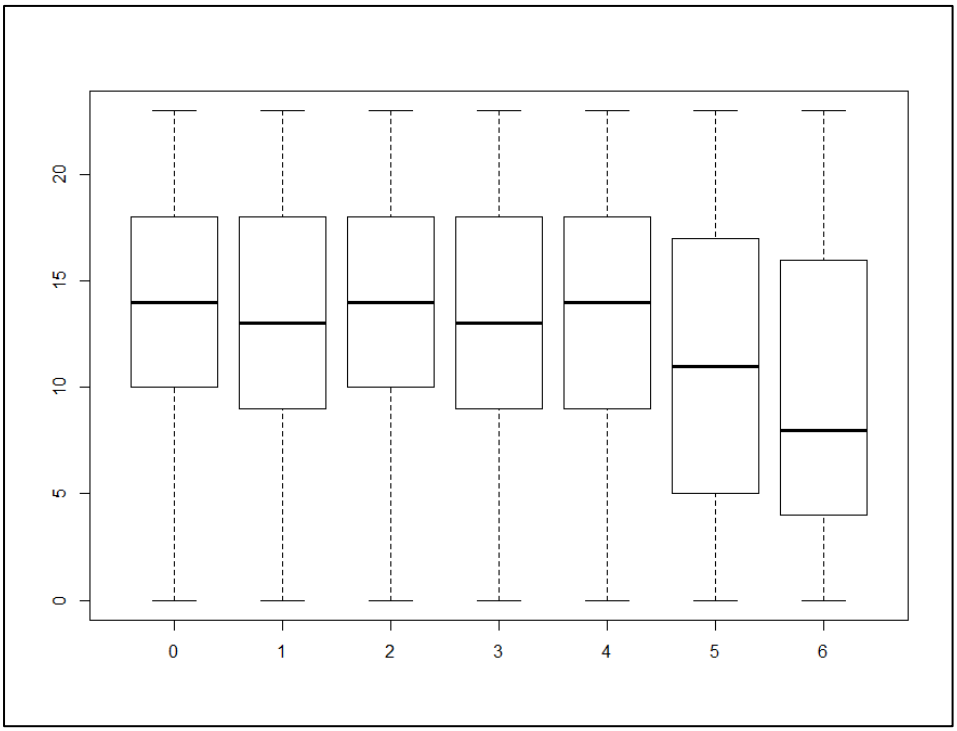
- Influence du jour de la semaine :

*Fig.3 : Histogramme de la répartition des courses par jour de la semaine*




On observe un léger pic d'affluence le vendredi et le samedi.

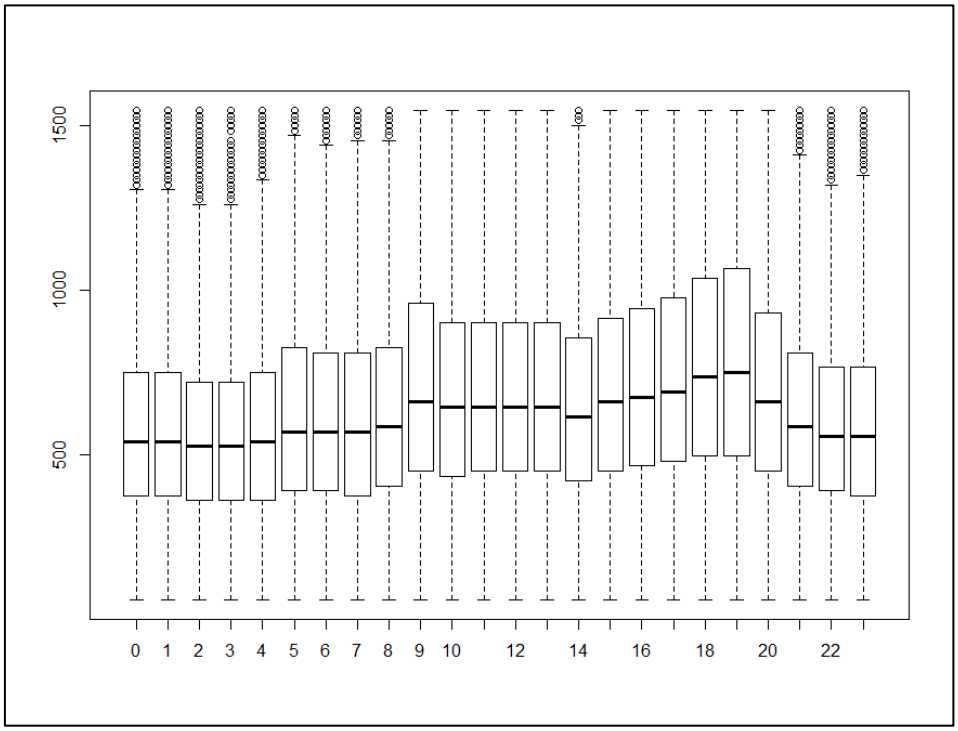
*Fig.4 : Répartition des courses par heure et par jour de la semaine*



Les courses du week-end ayant lieu plus fréquemment de nuit, les courses ont lieu plus tôt en moyenne le week-end qu'en semaine.

- Répartition des durées de trajet par jour :

*Fig.5 : Répartition des temps de trajet par jour de la semaine*




Les courses de nuit durent sensiblement moins longtemps que les courses de jour, cela s'explique par une plus faible circulation sur la route.

# II) METHODES

## 1) Modèle de Cox

Afin de modéliser le temps de trajet des Taxis, nous utilisons un modèle de Cox. Rappelons brièvement en quoi consiste ce modèle de Cox :

$$\lambda(X,t) = \lambda_o(t) * e^{X*\beta}$$

$$\Lambda(X,t) = \Lambda_o(t) * e^{X*\beta}$$

$$\Lambda = \int \lambda dt$$

Où **λ** et **Λ** et sont respectivement les fonctions intensité et risque cumulé, X l'observation et β les coefficients linéaires.

De même nous avons la relation entre la fonction de survie et la fonction intensité cumulée :

$$\bar{F}(X,t) = e^{-\Lambda_o(t)*e^{X*\beta}}$$

## 2) Transformation des données

Deux possibilités pour exploiter les coordonnées :

- Nous créons les covariables (xe-xs) et (ye-ys) qui correspondent aux coordonnées du vecteur « trajectoire » (cf trajectoires dessinées dans l'exploration des données)
- Nous pouvons aussi utiliser la norme euclidienne de ce vecteur, donnée par

$$\sqrt{(xe-xs)^2+(ye-ys)^2}$$

La seconde option peut sembler moins intéressante car elle ne comporte pas l'angle de destination de la course, cependant cette information est déjà donnée par la covariable « heading »

## 3) Premières modélisations sur Train_Train

Dans un premier temps, nous essayons 2 modèles sur un premier échantillon aléatoire (90%) de Train :

```
modele1=coxph(Surv(time,censor)~CALL_TYPE+wday+hour+heading+I(xe-xs)+I(ye-ys)+d_st,data=Train_Train)
```

```
modele2=coxph(Surv(time,censor)~CALL_TYPE+wday+hour+heading+I(sqrt((ye-ys)^2+(xe-xs)^2))+d_st,data=Train_Train)
```

Puis nous appliquons une fonction « step » à ces 2 modèles, visant à sélectionner le sous-modèle qui minimise le critère AIC. Les modèles résultants sont soit identiques, soit avec la variable « heading » en moins. Dans la suite, nous allons donc comparer les 4 modèles suivants :

```
modele3=coxph(Surv(time, censor)~CALL_TYPE+wday+hour+I((xe-xs))+I((ye-ys))+d_st,data=Train_Train)
```

```
modele4=coxph(Surv(time, censor)~CALL_TYPE+wday+hour+I(sqrt((ye-ys)^2+(xe-xs)^2))+d_st,data=Train_Train)
```

Les modèles 1 et 3 d'une part, 2 et 4 d'autre étant emboités (seule la variable « heading » diffère), on a les indicateurs statistiques suivants :

*Fig.7 : Indicateurs statistiques des différents modèles*

| | **Modèle1** | **Modèle 3** | **Modèle 2** | **Modèle 4** |
|---|---|---|---|---|
| **P-valeurs de Wald (Ho :βj=0)** | Toutes significatives (<<5%) sauf pour « heading » | Toutes significatives (<<5%) | Toutes significatives (<<5%) sauf pour « heading » | Toutes significatives (<<5%) |
| **P-valeurs de Wald global (Ho :β=0)** | 0 donc très favorable au modèle | 0 donc très favorable au modèle | 0 donc très favorable au modèle | 0 donc très favorable au modèle |
| **P-valeur du test de Rapport de Log Vraisemblance (Ho :β=0)** | 0 donc très favorable au modèle | 0 donc très favorable au modèle | 0 donc très favorable au modèle | 0 donc très favorable au modèle |
| **Concordance** | 0,57 | 0,57 | 0,71 | 0,71 |
| **AIC** | 3 716 781 | 3 716 781 | 3 682 914 | 3 682 914 |
| **Anova (Test de Fisher)** | P-valeur=0.17 >5% en faveur de Modèle 3 | | P-valeur=0.15 >5% en faveur de Modèle 4 | |
| **P-valeur du test de Rapport de Log Vraisemblance (modèles emboîtés)** | P-valeur=0.16 >5% en faveur de Modèle 3 | | P-valeur=4 e-12 <5% en faveur de Modèle 2 | |

Ces indicateurs statistiques sont plutôt en faveur des 4 modèles, sans qu'un modèle ne se distingue particulièrement.

Remarque : on a également tenté de modéliser une transformation « pspline », cependant ce modèle comporte des polynômes de degré 14, par parcimonie, nous nous restreignons donc aux 4 premiers modèles.

*Fig.8 : Test de linéarité des coefficients sur les variables continues*

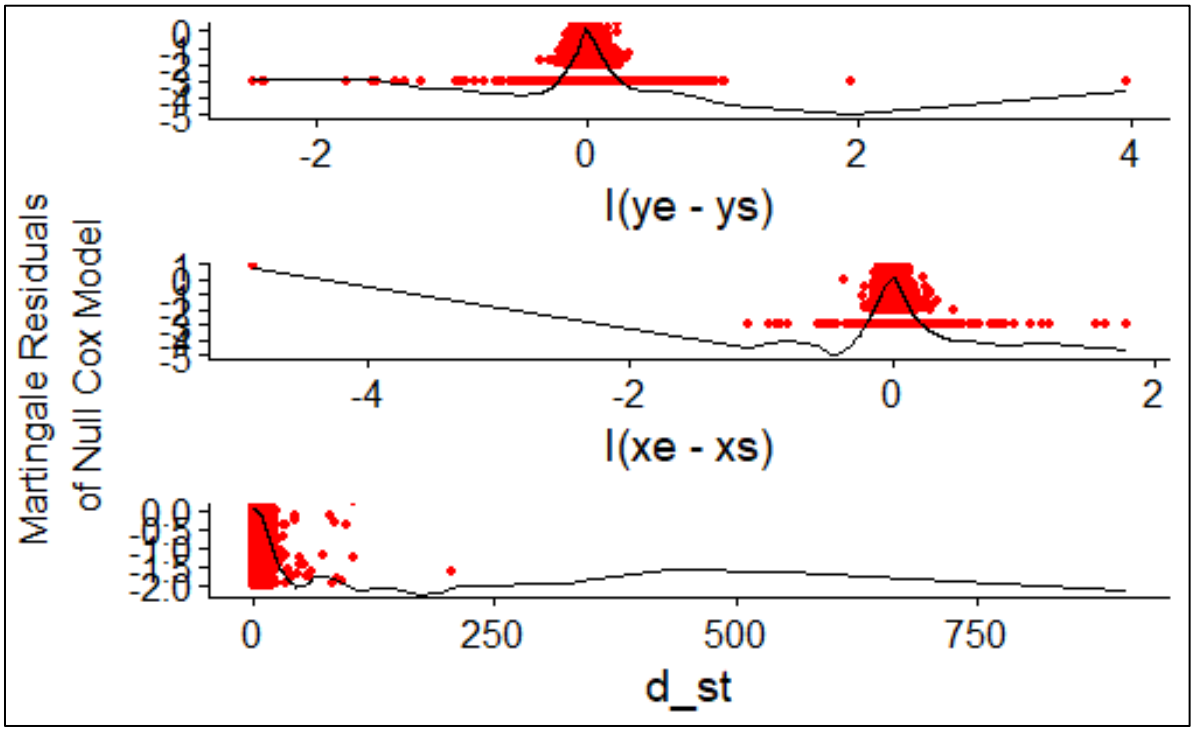
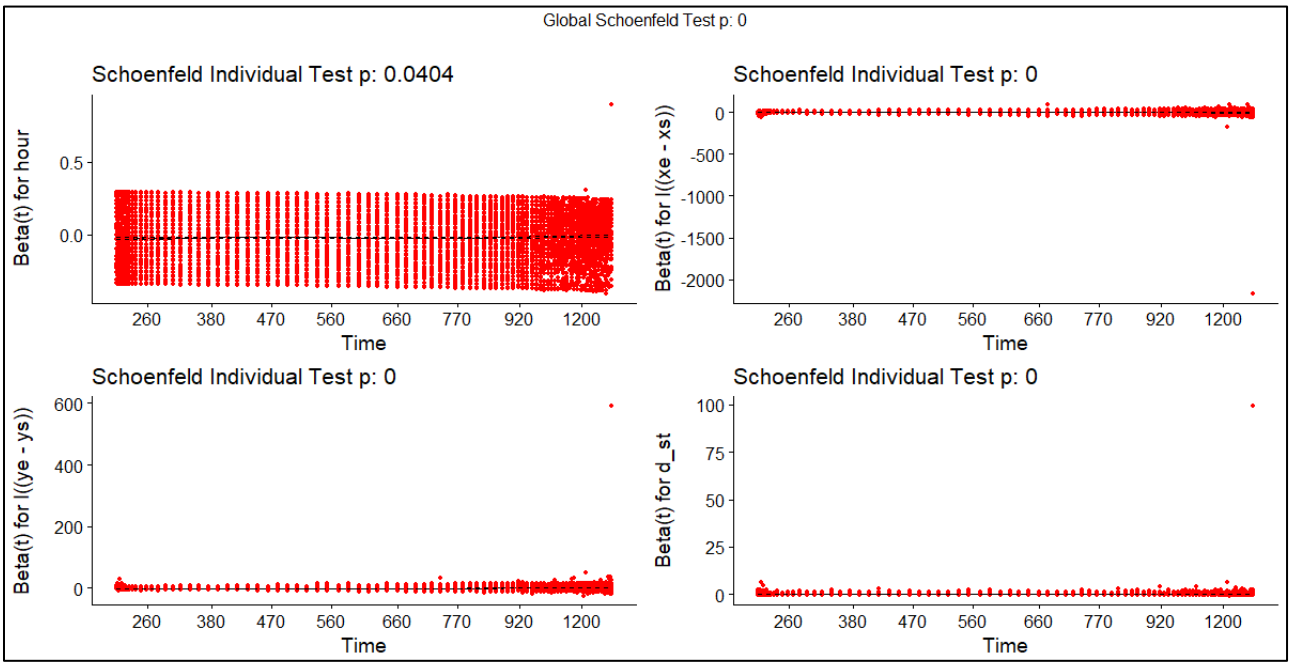



*Fig.9 : Test de la non dépendance des coefficients de β par rapport au temps*



- Le premier graphique montre le résultat du test la linéarité du modèle en les covariables, qui semble positif
- Le second graphique montre le résultat du test de non dépendance des coefficients linéaires βj par rapport au temps. Les p-valeurs ne semblent pas en faveur de l'hypothèse « Ho = βj indépendant du temps » …

### 4) Evaluation des modèles

Pour évaluer le modèle, le site « Kaggle » propose comme métrique de performance :

(1) $$\sqrt{\frac{1}{n} * \sum_{i=1}^{n} (\ln(p_i + 1) - \ln(a_i + 1))^2}$$

Avec $p_i$ la médiane prédite par le modèle pour l'observation i et $a_i$ le temps réel de la course de l'observation i, et n le nombre d'observations.

Nous utilisons la médiane car nous n'avons pas accès à l'ensemble des valeurs de la fonction de Survie. La médiane de la fonction de survie est définie comme :

$$Médiane\ (X) = \inf \{\, t / \bar{F}(X, t) \leq 0.5 \,\}$$

Pour évaluer les modèles, nous procédons ainsi :

- Nous itérons une boucle de 10 découpages aléatoires de Train en Train_Train et Train_Test de taille respectivement 90% et 10% de Train initial (moins les valeurs manquantes), ce qui nous permet de faire 10 apprentissages distincts.
- Puis à l'intérieur de cette première boucle, nous itérons une seconde boucle sur les observations pour chacun des 4 modèles, où nous calculons la médiane du temps de trajet pour chacune des observations. Ceci nous permet de calculer la mesure de performance sur les 10 échantillons aléatoires d'apprentissage/test (cf équation (1))
- Enfin, nous calculons la moyenne de ces 10 mesures de performance pour chacun des 4 modèles
- Le modèle dont la mesure de performance est la plus faible est retenu

## III) RESULTATS

- Résultat des mesures de performance moyennes sur 10 itérations d'apprentissages distincts :

| Modèle 1 | Modèle 2 | Modèle 3 | Modèle 4 |
|---|---|---|---|
| 1.024845 | 1.217392 | 1.021556 | 1.215364 |

- Le modèle retenu est donc le modèle 3

- On applique ensuite ce modèle au fichier Test et on prédit les médianes des fonctions de survie pour les 100 observations dans le fichier .csv en pièce jointe.

## CONCLUSION

Le modèle avec la meilleure mesure de performance est :

modele retenu : coxph(Surv(time, censor)~CALL_TYPE+wday+hour+I((xe-xs))+I((ye-ys))+d_st,data=Train_Train)

Nous avons réussi à construire un modèle avec une erreur quadratique logarithmique proche de 1.02 (versus environ 0.53 pour les meilleurs scores du site « Kaggle »).

Les indicateurs statistiques du modèle retenu sont pertinents.

- Interprétation des coefficients :

Toutes choses égales par ailleurs, le risque d'arrêt du trajet (« hazard function ») est multiplié par :

- 1.19 pour les CALL_TYPE B par rapport au CALL_TYPE A
- 0.92 pour les CALL_TYPE C par rapport au CALL_TYPE A
- 0.98 par créneau horaire
- 1.03 par jour de la semaine (lundi=0, dimanche=6)
- 1.93 par déplacement dans le sens de la coordonnée « x »
- 0.30 par déplacement dans le sens de la coordonnée « y »
- 0.96 par unité de distance par rapport au centre-ville

Critiques du modèle :

- On aurait préféré connaître l'espérance du temps de trajet plutôt que la médiane de la fonction de Survie, cependant nous n'avons pas accès à l'intégralité de la fonction de Survie
- Le mois calendaire est une covariable, absente du jeu de données, qui aurait pu être une covariable pertinente, notamment par rapport à la saisonnalité de l'activité touristique de la ville de Porto. On peut également citer : l'expérience du chauffeur de taxi, la catégorie du véhicule, la météo etc.
- Sauf erreur de ma part, ou bug de l'algorithme, il semblerait que le test de non linéarité des coefficients par rapport au temps ne soit pas concluant