Projet Actuariat - Tarification en assurance IARD avec les GLM et les GAM

Solène Corre, Florentin Dehooghe, François Delhaye

29 avril 2020

Table des matières

1 Présentation du projet

L'assurance est un contrat par lequel, moyennant le versement d'une prime dont le montant est fixé a priori (en début de période de couverture), l'assureur s'engage à indemniser l'assuré pendant toute la période de couverture (généralement un an). Cette prime doit refléter le risque associé au contrat.

Pour chaque police d'assurance, la prime est fonction de variables dites de tarification permettant de segmenter la population en fonction de son risque. Il est usuel d'utiliser une approche fréquence/sévérité ou une approche indemnitaire pour modéliser le coût annuel d'une police d'assurance. Sur les données utilisées dans ce projet, nous utiliserons cette dernière approche car on ne dispose pas des montants individuels de sinistre.

Le but de ce projet est de proposer un tarificateur en se basant deux méthodes : les modèles linéaires généralisés (GLM) et les modèles additifs généralisés (GAM). Ces derniers sont une extension des GLM (proposé par McCullagh et Nelder, 1989) en considérant une approche non-paramétrique pour le prédicteur.

Un second objectif sera, en plus de calculer une prime pure par police, de déterminer une commerciale intégrant une marge pour risque. Une approche par simulation sera réalisée pour juger de l'adéquation du chargement par rapport à la charge sinistre totale portefeuille.

2 Exploration des jeux de données freMPL1 et freMPL2

Un peu à la manière du machine learning, les données contenues dans freMPL2 serviront de données d'entraînement de notre modèle et les données de freMPL1 serviront pour tester notre modèle final.

2.1 Première visualisation des jeux de données

Les dimmensions du jeu de données **freMPL1** sont (30595, 22).

Ainsi, notre jeu contient 30595 données différentes, toutes définies par 22 caractéristiques différentes.

De même, les dimmensions du jeu de données **freMPL2** sont (48295, 22).

Ainsi, notre jeu contient 48295 données différentes, toutes définies par 22 caractéristiques différentes.

Les noms des caractéristiques des jeux de données sont les mêmes. es différentes caractéristiques sont :

- **Exposure** : il s'agit d'une donnée de type numérique qui correspond à la fréquence d'exposition aux risques d'un individu sur une année. Par exemple, si l'individu a été exposé 100 jours, le chiffre affiché est 0,27 (= 100/365,25).
- **LicAge** : c'est un nombre entier de mois correspondant à l'âge de la licence de la personne concernée.
- **RecordBeg** : cela correspond à la date de début d'exposition aux risques.
- **RecordEnd** : c'est la date de fin d'exposition au risque. Si elle n'est pas renseigner, c'est que la personne est toujours exposée.

— **VehAge** : Il correspond à l'âge du véhicule en année(s). Il est composé en 9 catégories distinctes : "0", "1", "2", "3", "4", "5", "6-7", "8-9" et "10+".
— **Gender** : c'est le sexe de l'individu.
— **MariStat** : il s'agit du statut marital de la personne. Elle est soit célibataire ("Alone") soit autre chose ("Other").
— **SocioCateg** : Cela correspond à la catégorie socioprofessionnelle de l'individu. Les valeurs, comprises entre "CSP1" et "CSP99", correspondent à la classification française (voir lien suivant : https://fr.wikipedia.org/wiki/Professions_et_catégories_socioprofessionnelles_en_France).
— **VehUsage** : Cela correspond à l'utilisation du véhicule par le propriétaire. Il est soit privée ("Private"), soit professionnel ("Professional"), ...
— **DrivAge** : C'est l'âge du conducteur (en années). Pour rappel, en France, la conduite est possible à partir de 18 ans.
— **HasKmLimit** : il s'agit d'une valeur numérique spécifiant si oui ("1") ou non ("0") l'assurance comporte une limite kilométique.
— **BonusMalus** : c'est un variable de type numérique, dont la valeur est comprise entre 50 et 350, précisant si la personne possède des bonus ou des malus. Si la valeur est inférieure à 100, l'individu a droit à des bonus. Sinon, la personne a des malus.
— **VehBody** : il s'agit du type de modèle concerné par l'assurance de l'individu.
— **VehPrice** : c'est un indicateur correspondant au prix du véhicule.
— **VehEngine** : cela correspond au type de moteur que possède le véhicule.
— **VehEnergy** : cela correspond au type d'énergie consommé par le véhicule que possède le véhicule
— **VehMaxSpeed** : c'est la vitesse maximum que peut atteindre le véhicule. Les différentes catégories sont:"1-130 km/h", "130-140 km/h", "140-150 km/h", "150-160 km/h", "160-170 km/h", "170-180 km/h", "180-190 km/h", "190-200 km/h", "200-220 km/h", "220+ km/h".
— **VehClass** : il s'agit de la classe du véhicule.
— **RiskVar** : Nombre compris entre 1 et 20 correspondant au risque inconnu probable.
— **ClaimAmount** : c'est le montant total de la garantie) laquelle peut prétendre l'assuré.
— **Garage** : il s'agit du type de garage auquel se rend l'assuré.
— **ClaimInd** : c'est un indicateur précisant si oui ou non l'assuré peut prétendre à une garantie.

Regardons maintenant les premiers éléments composant le jeu de données **freMPL1** :

	1	2	3
Exposure	0.583	0.200	0.083
LicAge	366	187	169
RecordBeg	2004-06-01	2004-10-19	2004-07-16
RecordEnd	NA	NA	2004-08-16
VehAge	2	0	1
Gender	Female	Male	Female
MariStat	Other	Alone	Other
SocioCateg	CSP1	CSP55	CSP1
VehUsage	Professional	Private+trip to office	Professional
DrivAge	55	34	33
HasKmLimit	0	0	0
BonusMalus	72	80	63
VehBody	sedan	microvan	other microvan
VehPrice	D	K	L
VehEngine	injection	direct injection overpowered	direct injection overpowered
VehEnergy	regular	diesel	diesel
VehMaxSpeed	160-170 km/h	170-180 km/h	170-180 km/h
VehClass	B	M1	M1
ClaimAmount	0	0	0
RiskVar	15	20	17
Garage	None	None	None

	1	2	3
ClaimInd	0	0	0

et aussi les premiers éléments composants **freMPL2** :

	1	2	3
Exposure	0.583	0.416	0.583
LicAge	579	361	366
RecordBeg	2004-06-01	2004-01-01	2004-06-01
RecordEnd	NA	2004-06-01	NA
VehAge	10+	1	2
Gender	Male	Female	Female
MariStat	Other	Other	Other
SocioCateg	CSP60	CSP1	CSP1
VehUsage	Private	Professional	Professional
DrivAge	83	55	55
HasKmLimit	0	0	0
BonusMalus	50	58	72
VehBody	sedan	sedan	sedan
VehPrice	N	D	D
VehEngine	injection	injection	injection
VehEnergy	regular	regular	regular
VehMaxSpeed	190-200 km/h	160-170 km/h	160-170 km/h
VehClass	H	B	B
RiskVar	14	15	15
ClaimAmount	0	0	0
Garage	None	None	None
ClaimInd	0	0	0

2.2 Nettoyage de données

Remarquons qu'il serait intéressant de faire un peu de nettoyage de données avant d'effectuer quelconques travaux sur celles-ci. Pour cela, nous allons créer une fonction qui servira à nettoyer les 2 data frames.

Cette fonction (appelée nettoyage_dataframe) prend l'un des deux data frames en paramètres et effectue les opérations suivantes :

- Suppression des données des individus assurés moins d'un jour (Exposure)
- Modification des données des individus ayant un ClaimAmount négatif
- Suppression de la colonne associée au sexe de la personne
- Réduction du nombre de catégories socioprofessionnels
- Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)

2.3 Statistiques descriptives

Regardons maintenant plus précisement les valeurs particulières de ces colonnes (valeurs minimum et maximum, moyenne, médiane,quantiles, ...). Pour cela, on exécute l'instruction **summary(freMPLx)** (et plus précisement **dfSummary(freMPLx)** du package summarytools pour l'affichage) ce qui donne les résultats suivants :

- Pour **freMPL1** :

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	Exposure [numeric]	Mean (sd) : 0.4 (0.3) min < med < max: 0 < 0.4 < 1 IQR (CV) : 0.5 (0.6)	753 distinct values	0 (0%)
2	LicAge [integer]	Mean (sd) : 301.3 (163) min < med < max: 0 < 283 < 940 IQR (CV) : 263 (0.5)	787 distinct values	0 (0%)
3	RecordBeg [Date]	min : 2004-01-01 med : 2004-03-01 max : 2004-12-30 range : 11m 29d	363 distinct values	0 (0%)
4	RecordEnd [Date]	min : 2004-01-03 med : 2004-07-01 max : 2004-12-31 range : 11m 28d	364 distinct values	13984 (46.55%)
5	VehAge [factor]	1. 0 2. 1 3. 10+ 4. 2 5. 3 6. 4 7. 5 8. 6-7 9. 8-9	4573 (15.2%) 4645 (15.5%) 1535 (5.1%) 4839 (16.1%) 3790 (12.6%) 3297 (11.0%) 2722 (9.1%) 2882 (9.6%) 1760 (5.9%)	0 (0%)
6	MariStat [factor]	1. célibataire 2. autre	7303 (24.3%) 22740 (75.7%)	0 (0%)
7	SocioCateg [factor]	1. CSP1 2. CSP2 3. CSP3 4. CSP5 5. CSP6 6. CSP7 7. CSP4 8. CSP9	1803 (6.0%) 830 (2.8%) 487 (1.6%) 19905 (66.3%) 4592 (15.3%) 59 (0.2%) 2361 (7.9%) 6 (0.0%)	0 (0%)
8	VehUsage [factor]	1. privée 2. privée et trajet vers bur 3. professionnel 4. trajet professionnel	9793 (32.6%) 13264 (44.1%) 6407 (21.3%) 579 (1.9%)	0 (0%)
9	DrivAge [integer]	Mean (sd) : 46.3 (14.9) min < med < max: 18 < 45 < 97 IQR (CV) : 23 (0.3)	80 distinct values	0 (0%)
10	HasKmLimit [integer]	Min : 0 Mean : 0.1 Max : 1	0 : 26756 (89.1%) 1 : 3287 (10.9%)	0 (0%)
11	BonusMalus [integer]	Mean (sd) : 64.2 (18.3) min < med < max: 50 < 54 < 272 IQR (CV) : 26 (0.3)	92 distinct values	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
12	VehBody	1. cabriolet	1315 (4.4%)	0
	[factor]	2. microvan	1347 (4.5%)	(0%)
		3. autobus	156 (0.5%)	
		4. coupé	1302 (4.3%)	
		5. autre microvan	1661 (5.5%)	
		6. berline	19764 (65.8%)	
		7. SUV	1823 (6.1%)	
		8. break	1605 (5.3%)	
		9. camionnette	1070 (3.6%)	
13	VehPrice	1. A ·	148 (0.5%)	0
	[factor]	2. B ·	102 (0.3%)	(0%)
		3. C ·	446 (1.5%)	
		4. D ·	1583 (5.3%)	
		5. E ·	2177 (7.2%)	
		6. F ·	2383 (7.9%)	
		7. G ·	2343 (7.8%)	
		8. H ·	2362 (7.9%)	
		9. I ·	2209 (7.4%)	
		10. J ·	2788 (9.3%)	
		[17 others]	13502 (44.9%)	
14	VehEngine	1. carburation	508 (1.7%)	0
	[factor]	2. GPL	2 (0.0%)	(0%)
		3. injection	20458 (68.1%)	
		4. injection directe surpuis	6895 (22.9%)	
		5. électrique	6 (0.0%)	
		6. injection surpuissante	2174 (7.2%)	
15	VehEnergy	1. diesel	9254 (30.8%)	0
	[factor]	2. GPL	2 (0.0%)	(0%)
		3. électrique	6 (0.0%)	
		4. essence	20781 (69.2%)	
16	VehMaxSpeed	1. 1-130 km/h	212 (0.7%)	0
	[factor]	2. 130-140 km/h	1066 (3.5%)	(0%)
		3. 140-150 km/h	1257 (4.2%)	
		4. 150-160 km/h	3801 (12.6%)	
		5. 160-170 km/h	5205 (17.3%)	
		6. 170-180 km/h	4749 (15.8%)	
		7. 180-190 km/h	4593 (15.3%)	
		8. 190-200 km/h	3613 (12.0%)	
		9. 200-220 km/h	3250 (10.8%)	
		10. 220+ km/h	2297 (7.6%)	
17	VehClass	1. 0	743 (2.5%)	0
	[factor]	2. A	2931 (9.8%)	(0%)
		3. B	9400 (31.3%)	
		4. H	4804 (16.0%)	
		5. M1	7622 (25.4%)	
		6. M2	4543 (15.1%)	
18	ClaimAmount	Mean (sd) : 259.6 (2337.2)	1799 distinct values	0
	[numeric]	min < med < max:		(0%)
		0 < 0 < 163427		
		IQR (CV) : 0 (9)		

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
19	RiskVar [integer]	Mean (sd) : 13.2 (4.7) min < med < max: 1 < 15 < 20 IQR (CV) : 7 (0.4)	20 distinct values	0 (0%)
20	Garage [factor]	1. aucun 2. garage indépendant 3. concessionnaire	19678 (65.5%) 3870 (12.9%) 6495 (21.6%)	0 (0%)
21	ClaimInd [integer]	Min : 0 Mean : 0.1 Max : 1	0 : 26778 (89.1%) 1 : 3265 (10.9%)	0 (0%)

On constate ainsi que, pour ce data frame, l'âge moyen du conducteur est de 46,3 ans avec pour écart-type 14,9 ans. Le plus jeune conducteur a 18 ans(âge minimum légale pour conduire en France) et le plus âgé a 97 ans. L'écart interquartile (IQR), c'est-à-dire la mesure de dispersion qui s'obtient en faisant la différence entre le premier (25% des valeurs du data frame sont inférieures à ce quartile) et le troisième quartile(75 %), est de 23. Autrement dit, 50% des âges des conducteurs est compris entre 35 et 58 ans. Le coefficient de variation (CV), le rapport entre l'écart-type et la moyenne, est égale à 3.

De même, en ce qui concerne l'usage du véhicule par son propriétaire, on remarquera que la plupart des personnes renseignées utilise leur véhicule pour les trajets privés et pour se rendre à leur bureau (44,1%).

— Pour **freMPL2** :

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	Exposure [numeric]	Mean (sd) : 0.4 (0.3) min < med < max: 0 < 0.4 < 1 IQR (CV) : 0.5 (0.6)	755 distinct values	0 (0%)
2	LicAge [integer]	Mean (sd) : 274.2 (161.8) min < med < max: 0 < 246 < 940 IQR (CV) : 255 (0.6)	809 distinct values	0 (0%)
3	RecordBeg [Date]	min : 2004-01-01 med : 2004-03-11 max : 2004-12-30 range : 11m 29d	365 distinct values	0 (0%)
4	RecordEnd [Date]	min : 2004-01-03 med : 2004-07-01 max : 2004-12-31 range : 11m 28d	364 distinct values	22109 (46.55%)
5	VehAge [factor]	1. 0 2. 1 3. 10+ 4. 2 5. 3 6. 4 7. 5 8. 6-7 9. 8-9	4313 (9.1%) 3987 (8.4%) 14347 (30.2%) 4140 (8.7%) 3760 (7.9%) 3658 (7.7%) 3412 (7.2%) 4909 (10.3%) 4971 (10.5%)	0 (0%)
6	MariStat [factor]	1. célibataire 2. autre	13690 (28.8%) 33807 (71.2%)	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
7	SocioCateg	1. CSP1	2366 (5.0%)	0
	[factor]	2. CSP2	1721 (3.6%)	(0%)
		3. CSP3	918 (1.9%)	
		4. CSP5	32894 (69.2%)	
		5. CSP6	5731 (12.1%)	
		6. CSP7	80 (0.2%)	
		7. CSP9	9 (0.0%)	
		8. CSP4	3778 (8.0%)	
8	VehUsage	1. privée	16785 (35.3%)	0
	[factor]	2. privée et trajet vers bur	22051 (46.4%)	(0%)
		3. professionnel	7958 (16.8%)	
		4. trajet professionnel	703 (1.5%)	
9	DrivAge	Mean (sd) : 44.5 (14.7)	83 distinct values	0
	[integer]	min < med < max:		(0%)
		18 < 42 < 103		
		IQR (CV) : 23 (0.3)		
10	HasKmLimit	Min : 0	0 : 41029 (86.4%)	0
	[integer]	Mean : 0.1	1 : 6468 (13.6%)	(0%)
		Max : 1		
11	BonusMalus	Mean (sd) : 69 (20.4)	108 distinct values	0
	[integer]	min < med < max:		(0%)
		50 < 64 < 272		
		IQR (CV) : 35 (0.3)		
12	VehBody	1. cabriolet	1506 (3.2%)	0
	[factor]	2. microvan	1458 (3.1%)	(0%)
		3. autobus	220 (0.5%)	
		4. coupé	1761 (3.7%)	
		5. autre microvan	1837 (3.9%)	
		6. berline	34051 (71.7%)	
		7. SUV	1974 (4.2%)	
		8. break	2231 (4.7%)	
		9. camionnette	2459 (5.2%)	
13	VehPrice	1. A ·	765 (1.6%)	0
	[factor]	2. B ·	655 (1.4%)	(0%)
		3. C ·	1697 (3.6%)	
		4. D ·	3617 (7.6%)	
		5. E ·	3878 (8.2%)	
		6. F ·	4106 (8.6%)	
		7. G ·	4184 (8.8%)	
		8. H ·	3952 (8.3%)	
		9. I ·	3505 (7.4%)	
		10. J ·	3898 (8.2%)	
		[17 others]	17240 (36.3%)	
14	VehEngine	1. carburation	6513 (13.7%)	0
	[factor]	2. GPL	2 (0.0%)	(0%)
		3. injection	30663 (64.6%)	
		4. injection directe surpuis	6554 (13.8%)	
		5. électrique	6 (0.0%)	
		6. injection surpuissante	3759 (7.9%)	

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
15	VehEnergy	1. diesel	13521 (28.5%)	0
	[factor]	2. GPL	2 (0.0%)	(0%)
		3. électrique	6 (0.0%)	
		4. essence	33968 (71.5%)	
16	VehMaxSpeed	1. 1-130 km/h	1256 (2.6%)	0
	[factor]	2. 130-140 km/h	2286 (4.8%)	(0%)
		3. 140-150 km/h	4073 (8.6%)	
		4. 150-160 km/h	7075 (14.9%)	
		5. 160-170 km/h	7915 (16.7%)	
		6. 170-180 km/h	7933 (16.7%)	
		7. 180-190 km/h	5795 (12.2%)	
		8. 190-200 km/h	4567 (9.6%)	
		9. 200-220 km/h	3998 (8.4%)	
		10. 220+ km/h	2599 (5.5%)	
17	VehClass	1. 0	1901 (4.0%)	0
	[factor]	2. A	4140 (8.7%)	(0%)
		3. B	15229 (32.1%)	
		4. H	7034 (14.8%)	
		5. M1	11756 (24.8%)	
		6. M2	7437 (15.7%)	
18	RiskVar	Mean (sd) : 13.5 (4.7)	20 distinct values	0
	[integer]	min < med < max:		(0%)
		1 < 15 < 20		
		IQR (CV) : 6 (0.3)		
19	ClaimAmount	Mean (sd) : 86.8 (1232.5)	873 distinct values	0
	[numeric]	min < med < max:		(0%)
		0 < 0 < 120152.4		
		IQR (CV) : 0 (14.2)		
20	Garage	1. aucun	35092 (73.9%)	0
	[factor]	2. garage indépendant	4642 (9.8%)	(0%)
		3. concessionnaire	7763 (16.3%)	
21	ClaimInd	Min : 0	0 : 45363 (95.5%)	0
	[integer]	Mean : 0	1 : 2134 (4.5%)	(0%)
		Max : 1		

Pour ce data frame, l'âge moyen du conducteur est de 44,5 ans avec pour écart-type 14,7 ans. Le plus jeune conducteur a 18 ans(âge minimum légale pour conduire en France) et le plus âgé a 103 ans. L'écart interquartile (IQR) est de 23 ce qui veut dire que 50% des conducteurs ont un âge compris entre 33 et 56 ans. Le coefficient de variation (CV) est égale à 3.

De même, en ce qui concerne l'usage du véhicule par son propriétaire, on remarquera que la plupart des personnes renseignées utilise leur véhicule pour les trajets privés et pour se rendre à leur bureau (46,4%).

On remarquera également qu'il existe des données manquantes, pour les 2 tableaux de données, dans la colonne RecEnd, ce qui signifie que les individus concernés sont toujours assurés.

On peut aussi utiliser la fonction **describe()** du package Hmisc pour avoir un aperçu de la dispersion des données. En effet, cette fonction détermine le type de la variable (character, factor, numeric,...) et affiche un "résumé" concis en fonction de chacun. Vous trouvez un exemple d'exécution de la focntion describe en annexe.

2.4 Représentations graphiques des données

Dans cette partie, vous allez voir des représentations graphiques des colonnes les plus importantes de nos data frames. L'ensemble des graphiques est cependant disponible dans les annexes de ce rapport.

2.5 Méthodes des composantes principales

Nous allons maintenant rentrer dans des méthodes d'analyse descriptives plus complètes pour nous permettre d'établir nos modèles linéaires. Pour cela, nous allons appliquer les méthodes d'analyse en composantes principales (ACP) et d'analyse factorielle des correspondances (AFC). Le but de ces méthodes est de définir les informations les plus significatives de nos data frames et de découvrir si oui ou non il existe certaines similitudes entre nos différentes informations pour pouvoir obtenir un data frame optimisé sur lequel on appliquera nos 2 modèles linéaires (GLM, GAM).

2.5.1 Analyse en composantes principales (ACP)

L'ACP permet d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables quantitatives.

C'est une méthode statistique qui permet d'explorer des données dites multivariées (données avec plusieurs variables). Chaque variable pourrait être considérée comme une dimension différente.

L'analyse en composantes principales est utilisée pour extraire et de visualiser les informations importantes contenues dans une table de données multivariées. L'ACP synthétise cette information en seulement quelques nouvelles variables appelées composantes principales. Ces nouvelles variables correspondent à une combinaison linéaire des variables originels. Le nombre de composantes principales est inférieur ou égal au nombre de variables d'origine.

2.5.1.1 Calcul de l'ACP

Pour réaliser le calcul de l'ACP, plusieurs fonctions, de différents packages, sont disponibles dans le logiciel R :

- *prcomp()* et *princomp()* issus du package *stats*
- *PCA()* issu du package *FactoMineR*
- *dudi.pca()* issu du package *ade4*
- *epPCA()* issu du package *ExPosition*.

Parmi ces fonctions, nous avons decider d'utiliser la fonction **PCA()** du package **FactoMineR** car ce package nous permettra également de réaliser notre seconde analyse. Enfin, pour extraire et visualiser les résultats, nous allons utiliser les fonctions R fournies par le package **factoextra**.

Nous allons donc exécuter l'ACP sur notre tableau freMPL2 en prenant à ce que l'ensemble des valeurs que nous utilisons soit de type numérique (quitte à réaliser une conversion sur certaines de nos colonnes).

Une fois que nos données ont été converties, il faut veiller à la *standardisation des données*. Pour cela, on normalise nos variables afin que le résultat de l'ACP obtenue ne soient pas affecté (par exemple, par des différences d'unités).

Ainsi, l'objectif est de rendre les variables comparables en les normalisant généralement de manière à ce qu'elles aient un écart type égal à 1 et une moyenne nulle.

L'approche consiste à transformer les données en soustrayant à chaque valeur une valeur de référence (la moyenne de la variable) et en la divisant par l'écart type.

Pour normaliser les données, il est possible d'utiliser la fonction *scale()*. Cependant, par défaut, le fonction PCA() normalise automatiquement les données. Nous n'avons pas eu besoin de faire cette transformation.

Réalisons maintenant notre Analyse en Composantes Principales. Pour cela, il faut exécuter la commande suivante :

```
freMPL2.pca <- PCA(freMPL2.active, ncp = 5, graph = FALSE)
```

Notre fonction PCA() prend en compte un data frame freMPL2.active qui correspond aux colonnes du dataframe freMPL2 qui sont de type numérique et que l'on souhaite analyser, un paramètre ncp qui correspond au nombre de dimensions conservées dans les résultats finaux (par défaut, ce nombre est égal à 3) et un paramètre logique graph qui précise si oui (graph = TRUE) ou non (graph = FALSE) nous voulons qu'un graphique du résultat s'affiche.

La fonction PCA() crée un objet contenant de nombreuses informations comme les valeurs propres (la variance du facteur correspondant où un facteur est une combinaison linéaire des variables initiales), la moyenne et l'écart type des variables, le poids de ces variables, ...

2.5.1.2 Analyse des résultats

2.5.1.2.1 Valeurs propres

Regardons d'abord les **valeurs propres**. Elles mesurent la quantité de variance expliquée par chaque axe prinicipal.

Examinons donc ces valeurs propres (eigenvalue en anglais) afin de déterminer le nombre de composantes principales à prendre en considération. Les valeurs propres et la proportion de variances retenues par les composantes principales peuvent être extraites à l'aide de la fonction *get_eignevalue()* du package *factoextra*.

Voici le résultat que l'on obtient :

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	2.341	29.259	29.259
Dim.2	1.359	16.990	46.248
Dim.3	1.022	12.780	59.029
Dim.4	0.973	12.157	71.186
Dim.5	0.949	11.865	83.051
Dim.6	0.667	8.336	91.387
Dim.7	0.616	7.702	99.089
Dim.8	0.073	0.911	100.000

Dans ce tableau, nous avons les valeurs propres de chacune des 8 colonnes du dataframe freMPL2.active (Exposure, LicAge, DrivAge, HasKmLimit, BonusMalus, RiskVar, ClaimAmount, ClaimInd), la proportion de variance associée et la variance cumulée.

La somme de toutes les valeurs propres donne une variance total de 8 (le nombre de dimensions). Pour obtenir la proportion de variance de la deuxième colonne, il suffit de prendre la valeur propre associée, de diviser cette valeur par le nombre de dimensions et de le mettre en pourcentage. Par exemple, pour la dimension 1, 2,341 divisé par 8 donne 0,29259, ce qui donne 29,259% de la variance. Enfin, la dernière colonne correspond à la somme cumulée des variances. Par exemple, 59.029 correspond à la somme de 12.780 avec 16.990 et 29.259.

On notera ainsi qu'environ 46,25% de la variance totale est expliquée par nos 2 premières dimensions.

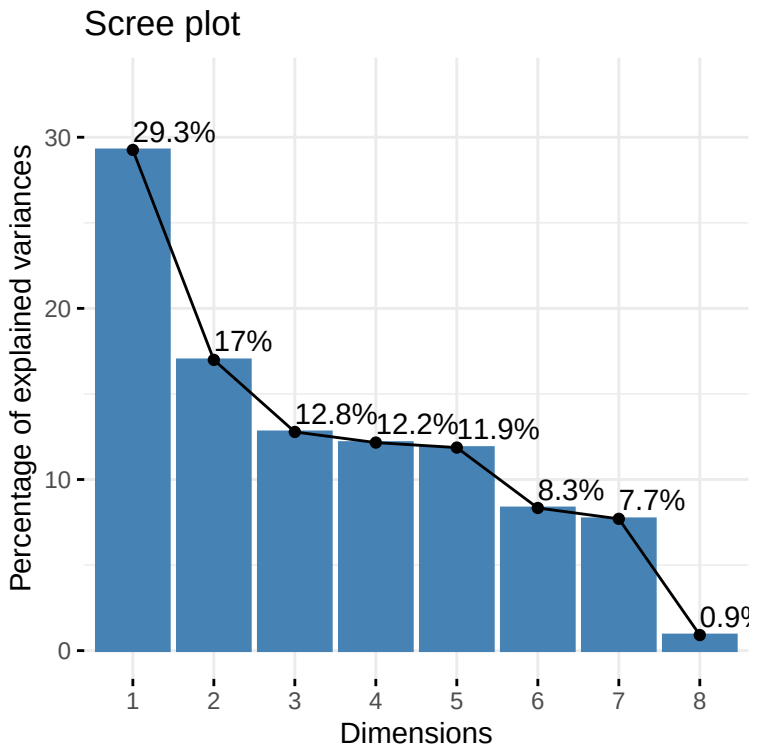
On peut utiliser ses valeurs propres pour déterminer le nombre d'axes principaux à conserver après l'ACP :

— Une valeur propre > 1 indique que la composante principale concernée représente plus de variance par rapport à une seule variable d'origine, lorsque les données sont standardisées. Ceci est généralement utilisé comme seuil à partir duquel les PC sont conservés (Dans ce cas, on aurait 3 composantes principales).

— On peut également limiter le nombre d'axes à un nombre qui représente une certaine fraction de la variance totale. Par exemple, si vous êtes satisfaits avec 70% de la variance totale expliquée, utilisez le nombre d'axes pour y parvenir (Dans ce cas, on aurait 4 dimensions).

Une autre méthode pour déterminer le nombre de composantes principales est de regarder le graphique des valeurs propres (appelé **scree plot**). Le nombre d'axes est déterminé par le point, au-delà duquel les valeurs propres restantes sont toutes relativement petites et de tailles comparables.

Le graphique des valeurs propres peut être généré à l'aide de la fonction *fviz_eig()* ou *fviz_screeplot()* du package *factoextra*.



Du graphique ci-dessus, nous pourrions vouloir nous arrêter à la cinquième composante principale puisque environ 83% des informations contenues dans les données sont conservées par les cinq premières composantes principales.

2.5.1.2.2 Les variables

Pour extraire les résultats pour les variables, à partir de l'ACP, il est possible d'utiliser la fonction *get_pca_var()*. Cette fonction retourne une liste d'éléments contenant tous les résultats pour les variables actives (coordonnées, corrélation entre les variables et les axes, cosinus-carré et contributions).

Les composants de get_pca_var() peuvent être utilisés dans le graphique des variables comme suit :

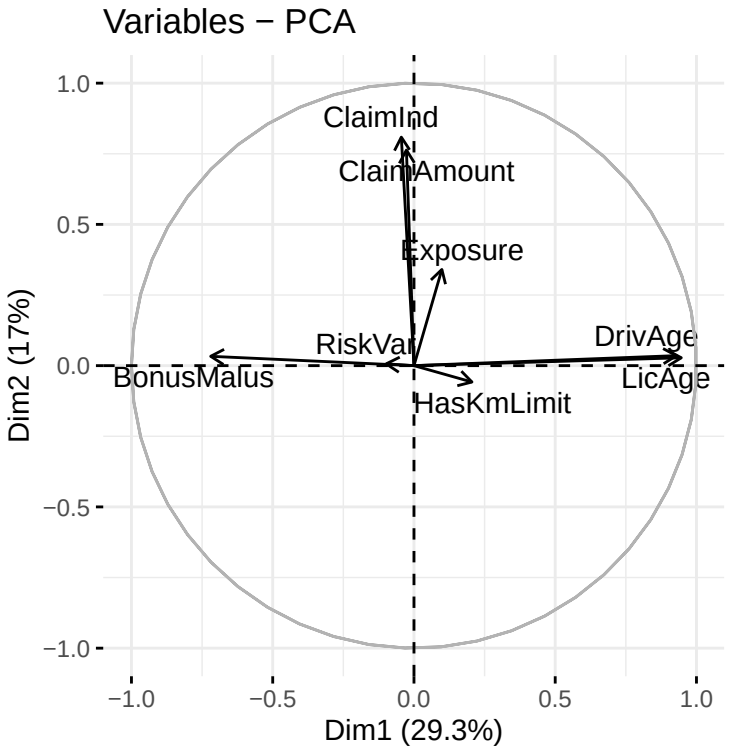
— *get_pca_var()$coord* : coordonnées des variables pour créer un nuage de points.
— *get_pca_var()$cos2* (cosinus carré des variables) : Représente la qualité de représentation des variables sur le graphique de l'ACP. Il est calculé comme étant les coordonnées au carré.
— *get_pca_var()$contrib* : contient les contributions des variables aux composantes principales.

Cercle de corrélation

Dans ce qui va suivre, nous allons visualiser les variables et tirer des conclusions à propos de leurs corrélations.

La corrélation entre une variable et une composante principale est utilisée comme coordonnées de la variable sur la composante principale. La représentation des variables diffère de celle des observations : les observations sont représentées par leurs projections, mais les variables sont représentées par leurs corrélations.

Visualisons d'abord les variables :

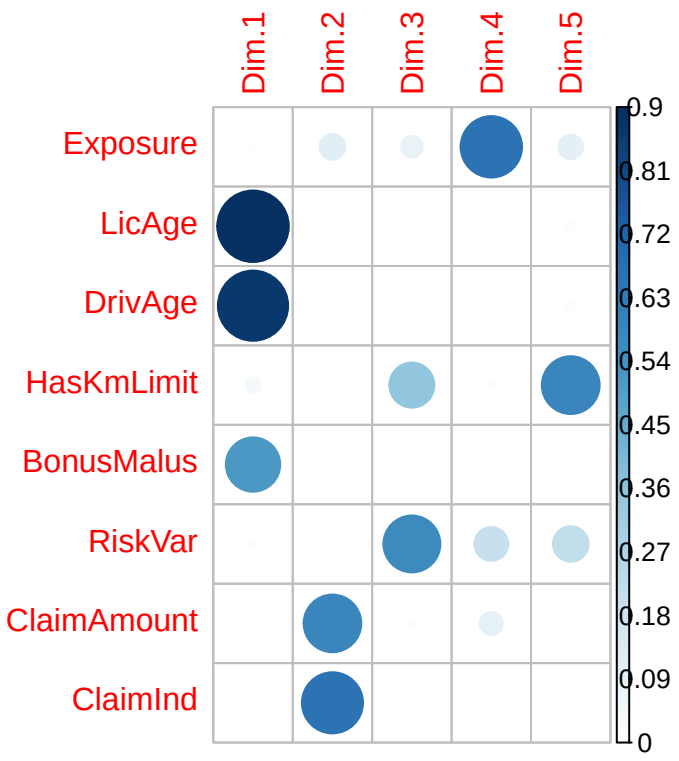


Le graphique ci-dessus est également connu sous le nom de **graphique de corrélation des variables**. Il montre les relations entre toutes les variables. Il peut être interprété comme suit:

— Les variables positivement corrélées sont regroupées.
— Les variables négativement corrélées sont positionnées sur les côtés opposés de l'origine du graphique (quadrants opposés).
— La distance entre les variables et l'origine mesure la qualité de représentation des variables. Les variables qui sont loin de l'origine sont bien représentées par l'ACP.

Qualité de la représentation

Pour visualiser la qualité de la représentation des variables sur la carte de l'ACP, nous allons utiliser le cosinus carré (*cos2*). Visualisons d'abord le cosinus carré des variables sur toutes les dimensions en utilisant le package *corrplot*. Voici le résultat :



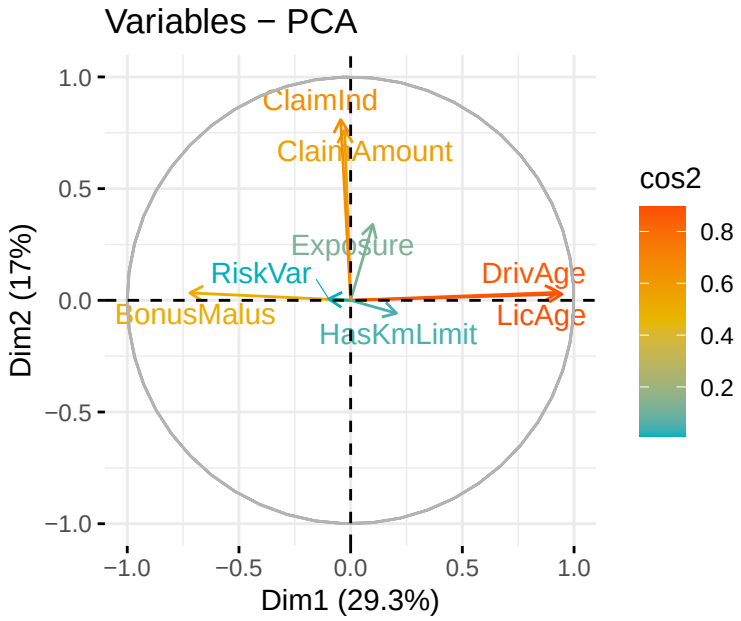
On remarquera qu'avec 5 axes principaux l'ensemble des 8 variables utilisées dans notre ACP sont plutôt bien représentées.

Pour visualiser le cosinus carré, nous aurions pu utiliser aussi la fonction *fviz_cos2()* du package *factoextra* pour créer un diagramme bâton du cosinus carré des variables.

Plus la valeur du cosinus carré est élévée, plus la réprésentation de la variable sur les axes prinicipaux pris en considération est bonne. Dans ce cas-là, la variable est positionnée à proximité de la circonférence du cercle de corrélation et le point associé dans le tableau de corrélation est gros et de couleur foncé.

Inversement, un faible cosinus carré indique que la variable n'est pas parfaitement représentée par les axes principaux. Dans ce cas, la variable est proche du centre du cercle et le point du tableau de corrélation est petit (voir inexistant).

Il est également possible de colorer les variables en fonction de la valeur de leurs cosinus carré.



On remarquera donc que les variables DrivAge et LicAge sont bien représentées par nos axes principaux tandis que la variable RiskVar n'est pas bien représenté par nos axes.

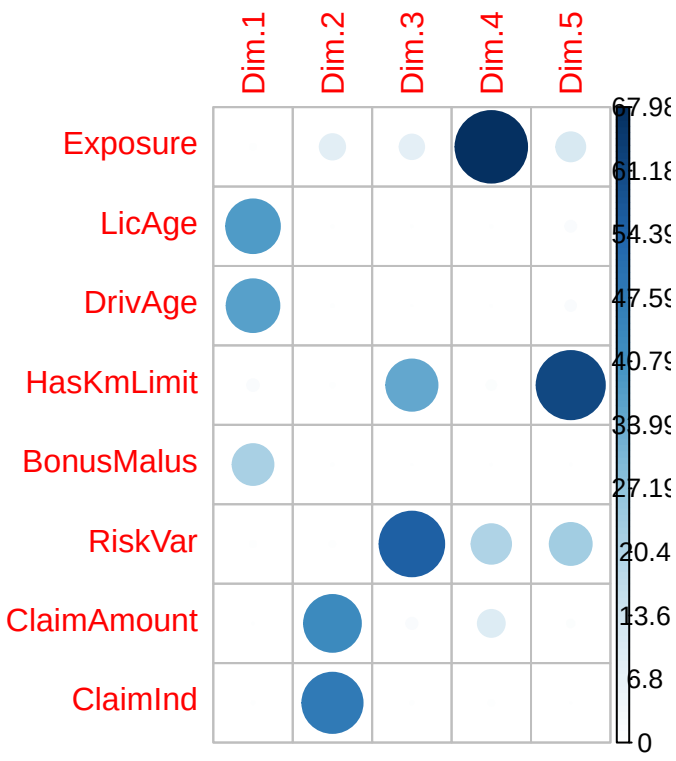
Contribution des variables aux axes principaux

Observons maintenant la contribution des variables aux axes principaux.

Les contributions des variables dans la définition d'un axe principal donné sont exprimées en pourcentage :

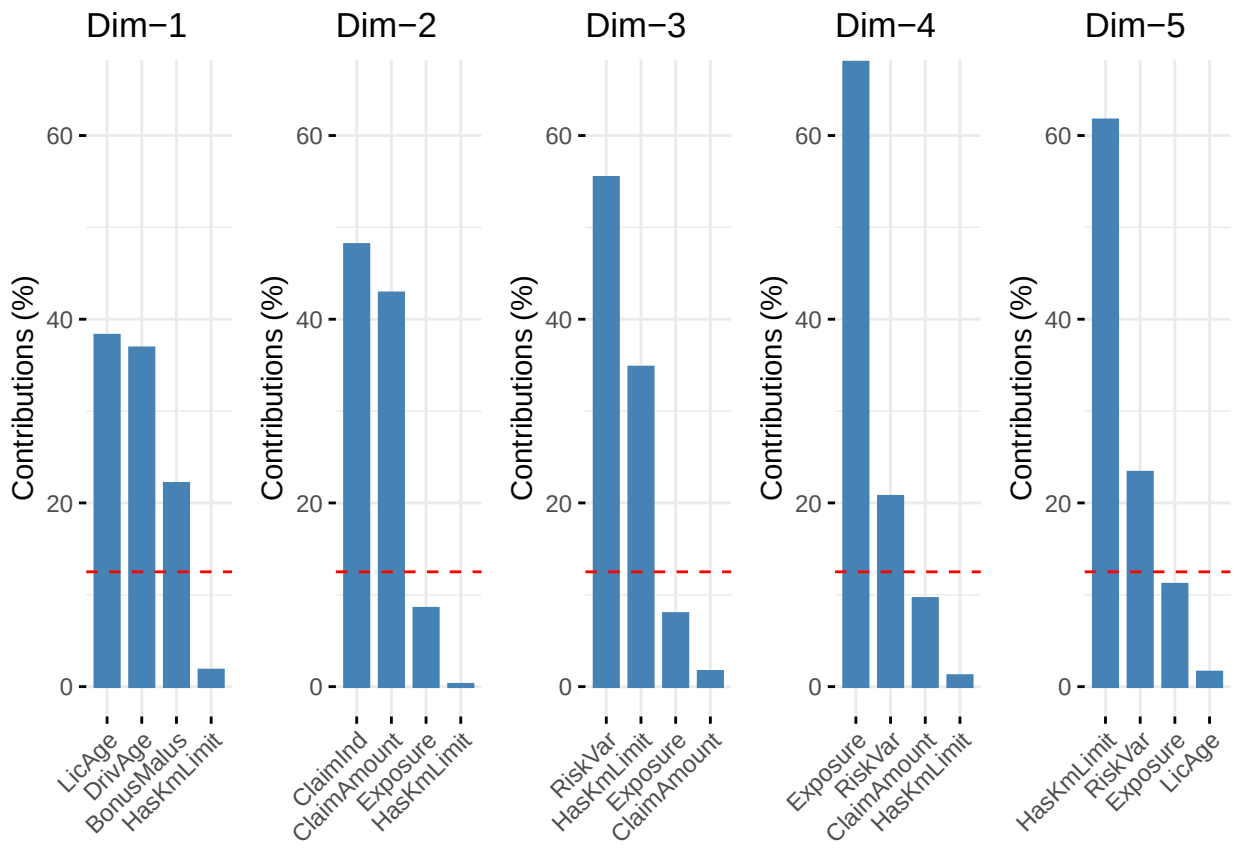
- Les variables corrélées par nos deux premiers axes sont les plus importantes pour expliquer la variabilité dans le jeu de données.
- Les variables qui ne sont pas en corrélation avec un axe ou qui sont corrélées avec les derniers axes sont des variables à faible apport et peuvent être supprimées pour simplifier l'analyse globale.

Comme pour la visualisation du cosinus carré, il est possible d'utiliser la fonction *corrplot()* pour mettre en évidence les variables les plus contributives pour chaque dimension:



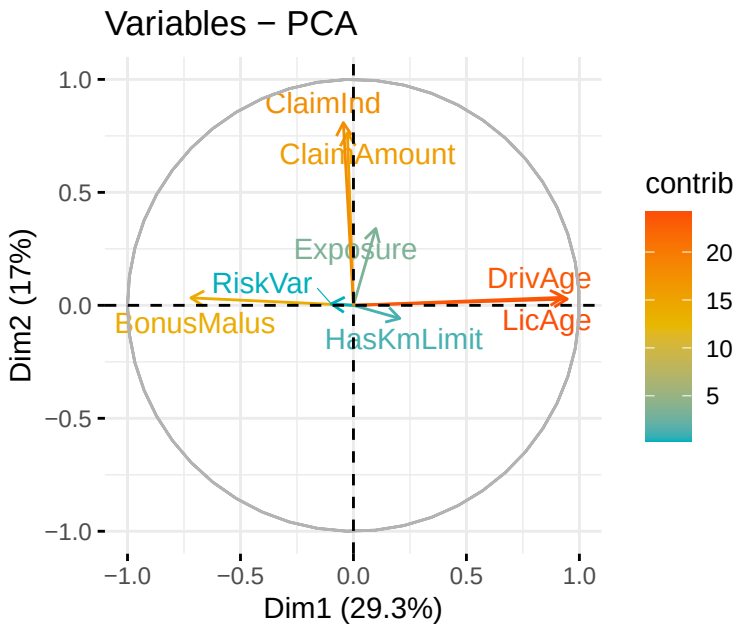
Grâce à ce graphique, on constate, par exemple, que les variables LicAge, DrivAge et BonusMalus représentent la première dimension (le premier axe principal).

La fonction *fviz_contrib()* peut être utilisée pour créer un diagramme bâton de la contribution des variables pour voir plus précisement la répartition des varaibles selon l'axe principal.



La ligne en pointillé rouge, sur les graphiques ci-dessus, indique la contribution moyenne attendue (dans notre cas, il est de 12,5%). Pour une composante donnée, une variable avec une contribution supérieure à ce seuil pourrait être considérée comme importante pour contribuer à la composante.

Enfin, on peut mettre en évidence les variables les plus importantes sur le graphe de corrélation.



Au final, on notera que 5 de nos variables ont plus d'importances que les autres : l'age du conducteur, l'âge de la licence de ce conducteur, son bonus ou son malus, s'il a eu un accident pendant qu'il était assuré et le montant auquel il peut prendre prétendre.

On a également vu que nos 8 variables peuvent être réduites en 5 nouvelles variables qui sont des combinaisons linéaires des anciennes variables, sans pour autant perdre d'informations ou très peu (17% de l'ensemble de nos données).

2.5.2 Analyse factorielle des correspondances (AFC)

L'analyse factorielle des correspondances est une extension de l'analyse en composantes principales pour analyser l'association entre deux variables qualitatives (ou catégorielles).

L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables catégorielles. Le tableau de contingence contient les fréquences formées par les deux variables. Il retourne les coordonnées des éléments des colonnes et des lignes du tableau de contingence. Ces coordonnées permettent de visualiser graphiquement l'association entre les éléments de lignes et de colonnes dans un graphique à deux dimensions.

Nous verrons donc comment calculer et interpréter l'AFC et nous tenterons de définir les éléments les plus importants expliquant les variations dans le jeu de données.

2.5.2.1 Calcul

Plusieurs fonctions de différents packages sont disponibles dans le logiciel R pour calculer l'AFC:

- *CA()* du package *FactoMineR*
- *ca()* du package *ca*
- *dudi.coa()* du package *ade4*
- *corresp()* du package *MASS*

— *epCA()* du package *ExPositio*

Cependant, nous allons utiliser la fonction *CA()* du package *FactoMineR* pour l'analyse et le package *factoextra* afin d'extraire et de visualiser les résultats de l'AFC.

Réalisons maintenant notre Analyse factorielle des correspondances. Pour cela, il faut exécuter la commande suivante :

```
freMPL2.ca <- CA (freMPL2.active, ncp=5, graph = FALSE)
```

Comme pour la fonction PCA() pour l'Analyse des Composantes Principales, notre fonction CA() prend en compte le data frame freMPL2.active que l'on souhaite analyser, le paramètre ncp qui correspond au nombre de dimensions conservées dans les résultats finaux et un paramètre logique graph qui précise si oui (graph = TRUE) ou non (graph = FALSE) nous voulons qu'un graphique du résultat s'affiche.

La fonction CA() crée un objet contenant de nombreuses informations sous forme de listes ou de matrices comme les valeurs propres (la variance du facteur correspondant où un facteur est une combinaison linéaire des variables initiales), le poids des lignes et des colonnes, le cosinus carré des lignes et des colonnes ...

2.5.2.2 Analyse des résultats

Pour analyser les résultats de notre AFC, nous pouvons utiliser les fonctions fournies par le package *factoextra* comme :

— *get_eigenvalue(freMPL2.ca)* pour obtenir les valeurs propres expliquées par chaque axe principal
— *fviz_eig(freMPL2.ca)* pour visualiser ces valeurs propres
— *get_ca_row(freMPL2.ca)* et *get_ca_col(freMPL2.ca)* pour avoir les résultats associés aux lignes ou aux colonnes.
— *fviz_ca_row(freMPL2.ca)* et *fviz_ca_col(freMPL2.ca)* pour visualiser ces résultats.

2.5.2.2.1 Conformité statistique : test de chi2

Pour interpréter l'AFC, la première étape consiste à évaluer s'il existe une dépendance significative entre les lignes et les colonnes.

Une méthode consiste à utiliser le test statistique *chi2* pour examiner l'association entre les modalités des lignes et celles des colonnes. Dans notre exemple, l'association est très significative puisque nous avons un résultat pour chi-square égal à 22101115 pour une p-value nulle (Un score élevé signifie un lien fort entre les lignes et les colonnes).

2.5.2.2.2 Valeurs propres

L'observation des valeurs propres permet de déterminer le nombre d'axes principaux à considérer. Elles correspondent à la quantité d'informations retenue par chaque axe. Elles sont grandes pour le premier axe et petites pour l'axe suivant.

Les valeurs propres et la proportion de variances pour les différents axes peuvent être extraites à l'aide de la fonction *get_eigenvalue()*.

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	0.837	87.891	87.891
Dim.2	0.104	10.964	98.854
Dim.3	0.005	0.565	99.419
Dim.4	0.002	0.251	99.670
Dim.5	0.002	0.215	99.884
Dim.6	0.001	0.062	99.946
Dim.7	0.001	0.054	100.000

Les dimensions sont ordonnées de manière décroissante et listées en fonction de la quantité de variance expliquée. La dimension 1 explique la plus grande variance, suivie de la dimension 2 et ainsi de suite.

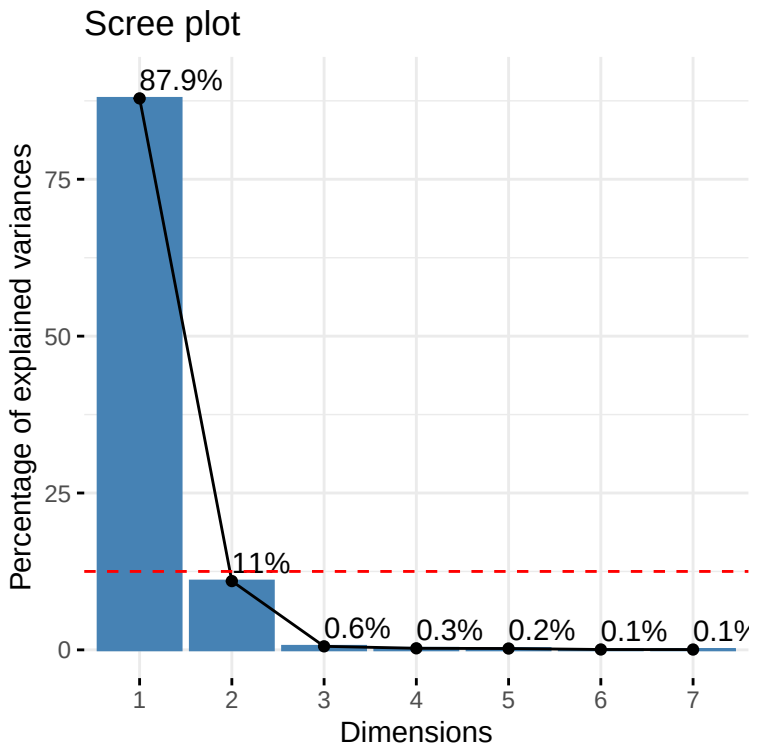
Le pourcentage cumulé expliqué est obtenu en ajoutant les proportions successives de variances expliquées pour obtenir le total courant. Par exemple, 87.89% plus 10.96% est égal à 98.85%. Par conséquent, environ 98.85% de la variance totale est expliquée par les deux premières dimensions.

Les valeurs propres peuvent être utilisées pour déterminer le nombre d'axes à retenir. Il n'y a pas de «règle générale» pour choisir le nombre de dimensions à conserver pour l'interprétation des données.

Dans notre analyse, les deux premiers axes expliquent 98.85% de la variance totale. C'est un pourcentage plus qu'acceptable.

Il est également possible de calculer une valeur propre moyenne au-dessus de laquelle l'axe doit être conservé dans le résultat. Dans notre cas, prenons 12,5% (1*100/8) comme valeur propre moyenne.Ainsi, tout axe avec une contribution supérieure devrait être considéré comme important et inclus dans la solution pour l'interprétation des données.

On peut voir cela sur le graphique des valeurs propres afin de déterminer le nombre de dimensions à l'aide de la fonction ou *fviz_screeplot()*.



Selon le graphique ci-dessus, seule la dimension 1 doit être considérées pour l'interprétation de la solution. La dimension 2 explique seulement 11% de l'inertie totale, ce qui est inférieur à la valeur moyenne des axes (12,5%) et trop petit pour être éventuellement conservé pour une analyse plus approfondie.

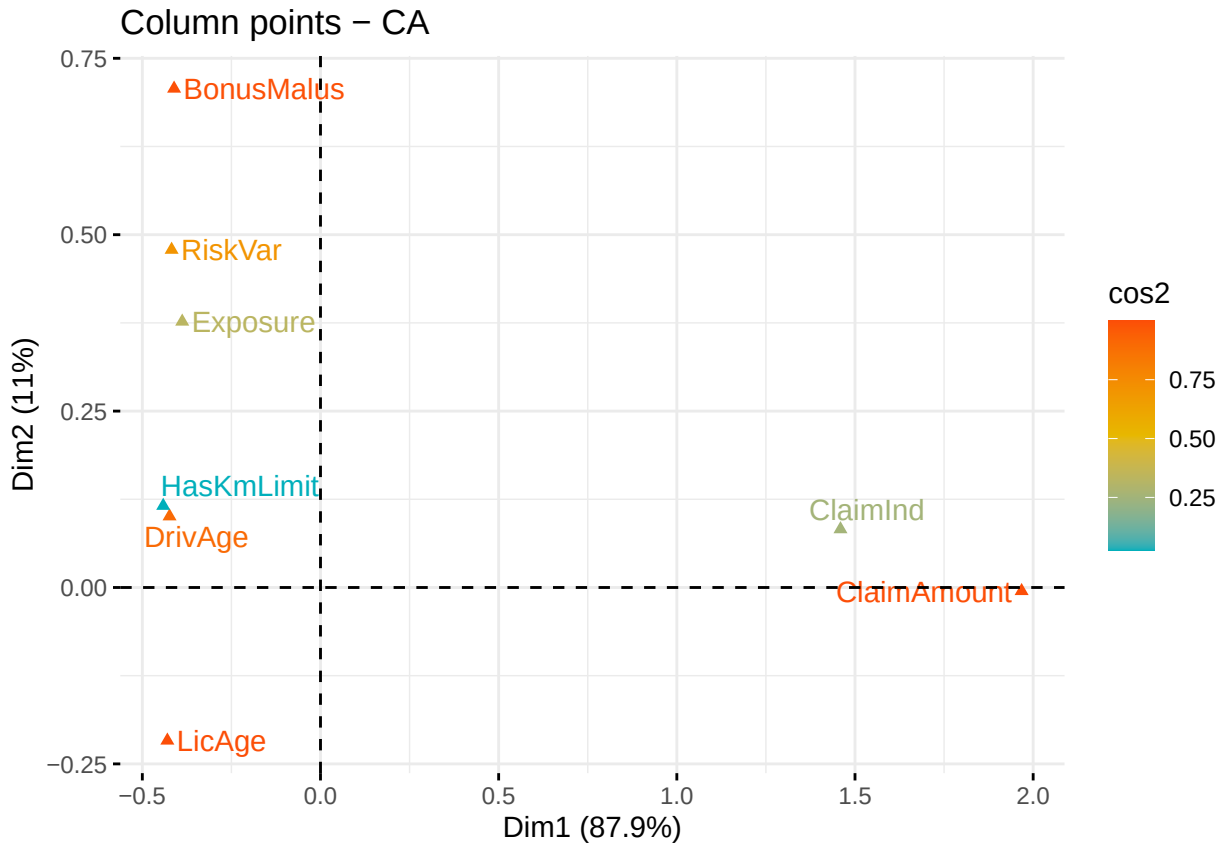
La dimension 1 explique environ 87,9% de l'inertie totale. Plus la rétention est élevée, plus la subtilité contenue dans les données d'origine est conservée dans la solution de l'AFC à faible dimension.

2.5.2.2.3 Les variables colonnes

La fonction *get_ca_col()* sert à extraire les résultats pour les colonnes. Cette fonction renvoie une liste contenant les coordonnées, le cos2, la contribution et l'inertie des colonnes.

Qualité et contribution pour les colonnes

Pour visualiser la qualité et la contribution des colonnes dans notre tableau de données, on peut utiliser la fonction *fviz_ca_col()*. Voici ce qu'elle affiche :



Comme pour l'ACP, on constate que 5 variables sont plutôt bien représentés. En effet, les colonnes LicAge, DrivAge, ClaimAmount, BonusMalus et RiskVar sont les variables les mieux représentées.

3 GLM

3.1 Qu'est-ce qu'un modèle linéaire généralisé (GLM) ?

Les modèles linéaires généralisés aussi appelés GLM sont une extension des modèles linéaires classiques.

La famille exponentielle est à la base des fonctions de distribution utilisées dans le modèle linéaire généralisé, qui comprend la plupart des modèles de régression en statistique et en économétrie.

La famille exponentielle regroupe les lois de probabilités dont la densité est donnée par

$$f(x;\theta) - exp(\sum_{j=1}^{k} \eta_j(\theta)T_j(x) - B(\theta))h(x), x \in \chi$$

où

$$\eta(.), T(.) : \mathbb{R}^k \to \mathbb{R}^k, h(.) : \mathbb{R}^d \to \mathbb{R}^k$$

la fonction de base

$$B : \mathbb{R} \to \mathbb{R} \& \theta \in \Longleftrightarrow \subset \mathbb{R}^k$$

est le vecteur de paramètres naturelles.

Voici un récapitulatif des lois usuelles utilisées en statistiques.

Loi	f.m.p./densité	θ	ϕ	$a(x)$	$b(x)$	Espérance	Var. fonction	Support
$\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ^2	x	$\frac{x^2}{2}$	$\mu=\theta$	1	$\mathbb{R}$
$\mathcal{G}_1(\alpha,\beta)$	$\frac{\beta^\alpha x^{\alpha-1}}{\Gamma(\alpha)}e^{-\beta x}$	$-\frac{\beta}{\alpha}=\frac{1}{\mu}$	$\frac{1}{\alpha}$	x	$-\ln(-x)$	$\mu=-\frac{1}{\theta}$	μ^2	$\mathbb{R}_+$
$\mathcal{G}_2(\nu,\mu)$	$\left(\frac{\nu x}{\mu}\right)^\nu \frac{e^{-\nu x/\mu}}{x\Gamma(\nu)}$	$\frac{-1}{\mu}$	$\frac{1}{\nu}$	x	$-\ln(-x)$	$\mu=-\frac{1}{\theta}$	μ^2	$\mathbb{R}_+$
$\mathcal{IG}_1(\mu,\lambda)$	$\sqrt{\frac{\lambda}{2\pi x^3}}e^{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}}$	$-\frac{1}{2\mu^2}$	$\frac{1}{\lambda}$	x	$-\sqrt{-2x}$	$\mu=(-2\theta)^{-\frac{1}{2}}$	μ^3	$\mathbb{R}_+$
$\mathcal{B}(\mu)$	$\mu^x(1-\mu)^{1-x}$	$\log(\frac{\mu}{1-\mu})$	1	x	$-\ln(1+e^x)$	$\mu=\frac{e^\theta}{1+e^\theta}$	$\mu(1-\mu)$	$\{0,1\}$
$\mathcal{P}(\mu)$	$\frac{\mu^x}{x!}e^{-\mu}$	$\log(\mu)$	1	1	e^x	$\mu=e^\theta$	μ	$\mathbb{N}$
$\mathcal{OP}(\phi,\mu)$	$\frac{\mu^{\frac{x}{\phi}}}{\frac{x}{\phi}!}e^{-\mu}$	$\log(\mu)$	ϕ			$\phi\mu$	$\mu(1+\phi\mu)$	$\mathbb{N}$

Cependant les modèles linéaires classiques sont utilisés uniquement lorsque la variable réponse est de type numérique continue. Or dans le cas que nous étudions, nous allons principalement utiliser des variables binaires avec lesquelles nous devrons utiliser la loi de Bernouilli.

De ce fait l'erreur qui résulte de notre modèle linéaire classique ne peut donc pas suivre une loi normale de moyenne nulle et de variance constante, nos résultats étant soit 0 ou 1.

Un GLM est composé de trois éléments : 1. Un prédicteur linéaire 2. Une fonction de lien 3. Une structure des erreurs

Les prédicteurs linéaires sont un ensemble de variables prédictives induisant une variable dépendante que l'on nomme réponse.

$$\eta \ = \sum_{j=1}^{p} \beta_j \ X_{ij}$$

La fonction de lien est une transformation par une fonction mathématique de la prédiction moyenne. Il s'agit donc d'une fonction qui transforme les valeurs du prédicteur linéaire. G étant ici notre fonction lien

$$g(\underset{\sim}{y}) = \sum_{j=1}^{p} \beta_j \ X_{ij}$$

Le but d'une fonction de lien est primordial dans notre exemple, celle-ci va contraindre les valeurs prédites dans l'échelle des valeurs observées. On comprendra alors que cette fonction lien nous est nécessaire pour pouvoir analyser nos variables binaires.

Enfin la structure des erreurs va donc devoir être adaptée par rapport à nos modèles linéaires classique afin qu'ils puissent correspondre à nos nouvelles données. Pour cela il existe plusieurs lois comme la loi de Poisson ou la loi Binomiale nous offrant une distribution des erreurs et des réponses qui seront différents.

La loi de Poisson est principalement utilisée lorsqu'il s'agit de problèmes de comptage (nombre de poissons dans une rivière, nombre de buts marqués dans une saison etc.)

Lorsque les données sont continues, nous pouvons donc utiliser une distribution Gaussienne, mais il existe également des distributions Binomiales négatives qui a pour but de modéliser des variables de comptage lorsque celles-ci sont sur-dispersées

Ainsi, dans notre cas les distribution de Poisson ou binomiale négative peuvent être utiliser pour représenter les fréquences des sinistres

Et Gamma et Inverse gauss pour représenter la sévérité des sinistres

3.2 Modélisation de la fréquence et de la sévérité des sinistres par les GLM

Dans cette partie, nous allons voir comment on modélise la fréquence et la sévérité des sévérités grâce à des GLM afin de pouvoir, par la suite, calculer la prime pure.

Les compagnies d'assurance prennent de nombreux critères en compte pour le calcul de votre prime d'assurance automobile. Parmi eux, on trouve :

- Les caractéristiques du véhicule
- L'usage du véhicule
- L'endroit de stationement du véhicule
- Le niveau de protection souhaité (assurance tous risques, assurance au tiers, etc.) ;
- Le kilométrage parcouru chaque année
- Le profil de conducteur : âge, date de permis, bonus-malus, antécédents d'assurance, etc.

Bien que nous ne possédons pas la totalité de ses informations (par exemple, l'endroit où le véhicule est stationné ou le nombre de kilomètres parcourus), nous avons une quantité suffisante d'informations pour pouvoir calculer notre propre prime d'assurance.

3.2.1 Fréquence des sinistres

Avant de calculer cette prime, il faut d'abord s'intéresser à la fréquence des sinistres pour une police (un individu) donné.

Pour celà, nous allons nous intéresser à la variable ClaimInd qui est un indice correspondant à l'événement "L'individu a eu un accident sur la période pour laquelle il était assuré". En effet, cette variable est à 1 si l'individu a subi un accident et à 0 sinon.

Ainsi, cette variable peut être associée à une expérience de Bernoulli, c'est-à-dire une expérience aléatoire comportant 2 issues, un succés ou un échec. Dans notre cas, le "succés" correspondrait au fait d'avoir eu au moins un accident et l'échec de ne pas avoir eu d'accident durant la durée d'exposition (période pendant lequel l'individu est assuré). La prédiction de notre modèle GLM correspondra donc à la probabilité qu'un individu avec ses caractéristiques associées d'avoir au moins un accident pendant qu'il est assuré.

Avant de faire une calibration de notre modèle, il est important de segmenter nos données continues comme l'âge du conducteur. En effet, l'âge du conducteur, s'il n'est pas diviser en plusieurs catégories, n'aura aucune influence puisqu'un coefficient unique lui serait associés. Pour catégoriser nos données, il faut utiliser l'instruction **cut()** en R. Voici un exemple d'utilisation :

```
freMPL2$DrivAge <- cut(freMPL2$DrivAge, breaks = c(17,2:8*10,103))
levels(freMPL2$DrivAge)
```

```
## [1] "(17,20]"  "(20,30]"  "(30,40]"  "(40,50]"  "(50,60]"  "(60,70]"  "(70,80]"
## [8] "(80,103]"
```

Dans cet exemple, l'âge du conducteur est divisé de la façon suivante :

- groupe 1 : les conducteurs agés entre 18 et 20 ans
- groupe 2 : les conducteurs agés entre 21 et 30 ans
- groupe 3 : les conducteurs agés entre 31 et 40 ans
- groupe 4 : les conducteurs agés entre 41 et 50 ans
- groupe 5 : les conducteurs agés entre 51 et 60 ans

— groupe 6 : les conducteurs agés entre 61 et 70 ans
— groupe 7 : les conducteurs agés entre 71 et 80 ans
— groupe 8 : les conducteurs agés de 81 ans ou plus.

Pour utiliser un GLM avec R, il suffit d'employer la fonction **glm()** du package **stats**. Cette fonction s'écrit de la façon suivante :

glm(variable à expliquer ~ variable(s) explicative(s), type de loi(fonction de lien), . . .)

Dans le premier modèle que l'on va éxécuter, la variable à expliquer correspond au ClaimInd, les variables explicatives correspondent à l'ensemble des autres variables de notre tableau.

Pour le type de loi, nous utilisons l'argument family de la fonction glm() et, plus précisement, la famille binomiale. Nous avons utilisé cette famille-là car elle correspond aux données associées à un schéma de Bernoulli comme notre variable à expliquer. Les autres familles possibles sont : gaussian, Gamma, inverse.gaussian, poisson, quasi, quasibinomial et quasipoisson.

Enfin, nous allons utilisé la fonction de lien **logit** qui est associé aux lois binomiales.

Voici le premier modèle que nous avons utilisé :

```
glmfreqinit <- glm(ClaimInd~., offset = log(Exposure), family=binomial(link="logit"), data=freMPL2)
```

On remarquera la présence de 2 nouveaux arguments : offset et data. L'argument data sert à préciser le dataframe utilisé pour entraîner notre modèle et l'argument offset qui fixe le coefficient associé à la variable spécifié à 1.

Pour juger de la pretinence de notre modèle, nous pouvons regarder le résumé de notre modèle (disponible en Annexes : Modèle GLM pour fréquence n°= 1). On notera que les coefficients associés à l'intercept (une constante qui correspond à la valeur moyenne de ClaimInd lorsque toutes les variables sont nulles) et à l' "Exposure" sont trop importantes.

On peut également évalué la performance d'une régression logistique avec des métriques clés spécifiques comme :

— l' AIC (Critère d'information d'Akaike) qui mesure l'ajustement lorsqu'une pénalité est appliquée au nombre de paramètres.
— Null deviance : Il s'agit de la déviance du modèle nul, c'est-à-dire qu'il n'est caractérisé par aucun facteur.
— Residual deviance : Il s'agit de la déviance du modèle avec toutes les variables.
— Number of Fisher Scoring iterations : Il s'agit du nombre d'itérations avant la convergence.

Le critère le plus souvent utilisé pour améliorer son modèle est le critère AIC. En effet, plus les valeurs d'AIC sont petites, plus notre modèle est plus proche de la vérité.

Pour trouver le modèle qui se rapproche donc le plus de la vérité, nous allons tester différents modèles avec une méthode progressive. C'est-à-dire que nous partons d'un modèle globale (contenant toutes les variables que l'on peut conserver) et que l'on ajoute et on supprime itérativement des variables dans le modèle afin de trouver le sous-ensemble de variables donnant le modèle le plus performant, c'est-à-dire un modèle qui réduit l'erreur de prédiction.

Pour cela, nous allons utiliser la fonction **step()** qui réalise cette sélection pas à pas. Voici donc le modèle GLM qui se rapproche le plus de la vérité :

```
glmfreqfinal <- glm(ClaimInd ~ Exposure + LicAge + RecordBeg + VehUsage + DrivAge +
    HasKmLimit + BonusMalus + VehBody + VehEngine + VehClass +
    RiskVar, offset = log(Exposure), family=binomial(link="logit"), data=freMPL2)
```

Voici la comparaison des performances entre nos 2 modèles :

	glm initial	glm final	différence
AIC	8947.5307446	1.6510924×10^4	-7563.3937287
déviance nulle	9007.8183728	1.6694188×10^4	-7686.3692968
déviance	8749.5307446	1.6422924×10^4	7673.3937287
nombre d'itération	11	10	1

```
##    ClaimInd Predict_ClaimInd_final
## 1         0                    0.05
## 2         0                    0.04
## 3         0                    0.05
## 4         0                    0.03
## 5         0                    0.01
## 6         0                    0.05
## 7         1                    0.11
## 8         0                    0.10
## 9         0                    0.03
## 10        0                    0.05
```

term	estimate	p.value
(Intercept)	7.4008188	0.0313432
Exposure	-0.4107231	0.0000420
LicAge(36,120]	-0.5013963	0.0002741
LicAge(120,240]	-1.0135217	0.0000000
LicAge(240,360]	-1.0337715	0.0000000
LicAge(360,480]	-1.1264340	0.0000000
LicAge(480,600]	-1.1252448	0.0000000
LicAge(600,720]	-1.1622387	0.0000114
LicAge(720,960]	-0.3558457	0.3773554
RecordBeg	-0.0006807	0.0121894
VehUsageprivée et trajet vers bureau	0.1677576	0.0032098
VehUsageprofessionnel	0.2320542	0.0012295
VehUsagetrajet professionnel	0.2317048	0.2076661
DrivAge(20,30]	-0.3099223	0.2574375
DrivAge(30,40]	0.0277059	0.9210963
DrivAge(40,50]	-0.0184779	0.9487449
DrivAge(50,60]	0.0887675	0.7641605
DrivAge(60,70]	0.2481255	0.4248226
DrivAge(70,80]	-0.1036408	0.7599400
DrivAge(80,103]	0.1622203	0.6841873
HasKmLimit1	-0.1154083	0.1259918
BonusMalus(100,350]	0.4728058	0.0000016
VehBodymicrovan	-0.0420086	0.8051183
VehBodyautobus	-0.4930079	0.2032957
VehBodycoupé	-0.0706163	0.6663497
VehBodyautre microvan	-0.1302188	0.4219627
VehBodyberline	-0.1856677	0.1320949
VehBodySUV	-0.0100828	0.9488913
VehBodybreak	-0.6246240	0.0002937
VehBodycamionnette	-0.1553241	0.3493435
VehEngineGPL	-8.4763186	0.9514713
VehEngineinjection	0.3819029	0.0000010
VehEngineinjection directe surpuissante	0.4812376	0.0000005

term	estimate	p.value
VehEngineélectrique	2.0483854	0.0659657
VehEngineinjection surpuissante	0.4144728	0.0001723
VehClassA	-0.4775388	0.0017465
VehClassB	-0.3923098	0.0033430
VehClassH	-0.1697628	0.2299550
VehClassM1	-0.3914327	0.0028873
VehClassM2	-0.3106034	0.0267901
RiskVar(4,8]	0.0881130	0.4597007
RiskVar(8,12]	0.1464974	0.1947564
RiskVar(12,16]	-0.0041769	0.9692561
RiskVar(16,20]	0.2193579	0.0430136

```
##    DrivAge frequence_observee_somme frequence_predite_final_model_somme
## 1  (17,20]                       20                                19.99
## 2  (20,30]                      402                               401.79
## 3  (30,40]                      604                               604.00
## 4  (40,50]                      411                               411.19
## 5  (50,60]                      401                               400.35
## 6  (60,70]                      197                               197.03
## 7  (70,80]                       69                                68.71
## 8 (80,103]                       30                                30.03
```

```
##      LicAge frequence_observee_somme frequence_predite_final_model_somme
## 1   (-1,36]                       90                                89.99
## 2  (36,120]                      405                               404.87
## 3 (120,240]                      586                               585.84
## 4 (240,360]                      447                               447.19
## 5 (360,480]                      368                               367.49
## 6 (480,600]                      178                               178.03
## 7 (600,720]                       45                                44.69
## 8 (720,960]                       15                                14.99
```

```
##   VehAge frequence_observee_somme frequence_predite_final_model_somme
## 1      0                      216                               202.92
## 2      1                      190                               196.08
## 3    10+                      534                               572.61
## 4      2                      230                               208.55
## 5      3                      192                               175.02
## 6      4                      180                               173.83
## 7      5                      163                               155.07
## 8    6-7                      211                               222.96
## 9    8-9                      218                               226.05
```

3.2.2 Sévérité des sinistres

```
glmsevinit <- glm(ClaimAmount~., family=Gamma(link = "log"), data=freMPL2.posclaim)
#glmsevfinal<- step(glmsevinit)
glmsevfinal <- glm(ClaimAmount ~ Exposure + LicAge + RecordBeg + VehAge + MariStat +
    VehUsage + DrivAge + HasKmLimit + BonusMalus + VehBody +
```

```
    VehEngine + VehEnergy + VehMaxSpeed + VehClass + RiskVar +
    Garage, family=Gamma(link = "log"), data=freMPL2.posclaim)
```

term	estimate	p.value
(Intercept)	8.5023307	0.1780393
Exposure	-0.3504966	0.0495540
LicAge(36,120]	-0.1691749	0.4737136
LicAge(120,240]	-0.1720450	0.4897297
LicAge(240,360]	-0.0926311	0.7381955
LicAge(360,480]	-0.1014967	0.7428897
LicAge(480,600]	-0.1922878	0.5934141
LicAge(600,720]	-0.1173925	0.8038292
LicAge(720,960]	-1.1588824	0.1118739
RecordBeg	0.0000165	0.9734414
VehAge1	0.2362062	0.1911327
VehAge10+	-0.1368285	0.4474217
VehAge2	0.1767800	0.3091205
VehAge3	0.1732252	0.3486787
VehAge4	0.1206356	0.5229630
VehAge5	0.3504180	0.0757393
VehAge6-7	-0.0604393	0.7486359
VehAge8-9	-0.2324099	0.2225077
MariStatautre	0.0948650	0.3501179
VehUsageprivée et trajet vers bureau	0.0272030	0.7837892
VehUsageprofessionnel	-0.1695126	0.1854213
VehUsagetrajet professionnel	-0.3407605	0.2958110
DrivAge(20,30]	0.0233090	0.9595256
DrivAge(30,40]	0.0390760	0.9343148
DrivAge(40,50]	-0.2135075	0.6626117
DrivAge(50,60]	-0.0272074	0.9569650
DrivAge(60,70]	0.1688650	0.7529938
DrivAge(70,80]	0.0687956	0.9053671
DrivAge(80,103]	1.0855567	0.1295824
HasKmLimit1	-0.2036862	0.1302068
BonusMalus(100,350]	0.0316081	0.8519096
VehBodymicrovan	0.8002581	0.0114918
VehBodyautobus	-0.1680490	0.8099603
VehBodycoupé	-0.1350727	0.6460884
VehBodyautre microvan	0.2226803	0.4678328
VehBodyberline	-0.0684795	0.7635584
VehBodySUV	0.0230232	0.9367684
VehBodybreak	0.2305800	0.4689070
VehBodycamionnette	-0.0329336	0.9233211
VehEngineinjection	-0.2487577	0.1456990
VehEngineinjection directe surpuissante	-0.4016675	0.1442571
VehEngineélectrique	-1.1928535	0.5190968
VehEngineinjection surpuissante	-0.1685412	0.4931484
VehEnergyessence	-0.3005566	0.0524806
VehMaxSpeed130-140 km/h	-0.5939348	0.0604775
VehMaxSpeed140-150 km/h	-0.4797771	0.1229528
VehMaxSpeed150-160 km/h	-0.6594079	0.0304925
VehMaxSpeed160-170 km/h	-0.4963599	0.1138736
VehMaxSpeed170-180 km/h	-0.3473724	0.2833392

term	estimate	p.value
VehMaxSpeed180-190 km/h	-0.6365487	0.0609109
VehMaxSpeed190-200 km/h	-0.4755505	0.1731607
VehMaxSpeed200-220 km/h	-0.5710419	0.1132330
VehMaxSpeed220+ km/h	-0.4028479	0.2952948
VehClassA	-0.2160233	0.4812885
VehClassB	0.1725520	0.5266119
VehClassH	0.0135059	0.9629464
VehClassM1	-0.0811366	0.7605071
VehClassM2	-0.1170001	0.6827473
RiskVar(4,8]	-0.0840285	0.6878041
RiskVar(8,12]	-0.0156177	0.9374464
RiskVar(12,16]	0.0538807	0.7779097
RiskVar(16,20]	0.0687386	0.7210434
Garagegarage indépendant	-0.0603167	0.6824388
Garageconcessionnaire	-0.0599278	0.6050396

comparaison modèle initial et modèle final :

	glm initial	glm final	différence
AIC	3.6253113×10^4	3.6295268×10^4	-42.1550291
déviance nulle	3143.9483434	3143.9483434	0
déviance	2614.7274762	2726.6198777	111.8924014
nombre d'itération	25	25	0

```
##    ClaimAmount Predict_ClaimAmount
## 1            0              4150.29
## 2            0              2223.29
## 3            0              1980.88
## 4            0              5298.57
## 5            0              3538.33
## 6            0              3008.69
## 7         1204              1970.13
## 8            0              1976.75
## 9            0              2943.52
## 10           0              2243.04
```

3.2.3 Calcul de la prime pure établi par le GLM

Soit X le coût monétaire au risque

Selon le modèle général,

$$X = \sum_{i=1}^{N} B_k$$

où N correspond au nombre de sinistres
et Bk correspond au montant de sinistres

Autrement dit, N représente la fréquence (variable discrète) et Bk la sévérité (variable continue positive)

En admettant que la fréquence n'a pas d'influence sur la sévérité et que les montants des sinistres ont le même comportement aléatoire, on a : E(X) = E(N).E(B) (prime pure).

Comment calculer E(N)? Comment calculer E(B)?

3.2.3.1 Calcul de l'espérance du nombre de sinistres (E(N))

fonction de lien logit :

$$log_{it}(x) = ln(\frac{X}{1-x})$$

fonction réciproque :

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

```
##      1
## 207.51
```

Application du modèle sur notre 2nd tableau

```
##     1
## 99.04
```

4 Gestion de projet

Durant ce projet, nous avons dû nous adapter aux circonstances actuelles et ainsi appréhender la gestion de projet d'une manière totalement nouvelle pour nous. Nous avons réussi à nous organiser afin que le projet se déroule le mieux possible.

Pour cela, plusieurs alternatives ont été mises en place :

Nous faisions environ un point par semaine avec nos référents sur Teams afin de :

- Avoir un suivi de projet.
- Poser des questions si certaines parties du projet ne sont pas totalement maîtrisées.
- Procéder à des modifications si certaines tâches effectuées ne correspondent pas à leurs attentes.

Nous faisions également une réunion par semaine avec seulement les membres du groupe de projet afin de :

- Pouvoir entrer plus dans les détails.
- Faire une synthèse des tâches à effectuer et les répartir.
- Se fixer des objectifs hebdomadaires.

Pour le partage d'informations, nous utilisions GitHub. De plus, lorsque nous avions des questions à poser à nos référents, nous échangions par mail.

La communication entre nous s'est faite via Messenger ou Discord, cela nous a permis d'échanger de manière instantanée et également d'avoir un historique des différents éléments que nous avions partagés.

5 Bibliographie

5.1 Internet

- Pour la documentation R : https://www.rdocumentation.org/

— Pour connaître le fonctionnement de l'assurance automobile en France : https://fr.wikipedia.org/wiki/Assurance_automobile_en_France
— Pour l'analyse en composantes principales: http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/73-acp-analyse-en-composantes-principales-avec-r-l-essentiel/
— Pour l'analyse factorielle des correspondances : http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/74-afc-analyse-factorielle-des-correspondances-avec-r-l-essentiel/
— Pour les GLM : https://statistique-et-logiciel-r.com/introduction-aux-glm/
— Pour des compléments d'informations (lois de probabilités, ...): https://www.wikipedia.org/

5.2 Littérature

6 Annexes

6.1 Affichage de l'implementation de la fonction nettoyage_dataframe :

```
nettoyage_dataframe <- function(dt){

  # Suppression des données des individus assurés moins d'un jour (Exposure)
  dt <- subset(dt,dt$Exposure>1/365.25)

  # Modification des données des individus ayant un ClaimAmount négatif
  dt <- subset(dt,dt$ClaimAmount>=0)

  # Suppression de la colonne associée au sexe de la personne et de ClaimInd
  dt <- dt[,-6]
  dt <- dt[,-21]

  # Réduction du nombre de catégories socioprofessionnels
  levels(dt$SocioCateg) <- c(levels(dt$SocioCateg), "CSP4", "CSP6",
                                    "CSP9")
  for (i in 1:dim(dt)[1]){
    if (dt$SocioCateg[i]%in%c("CSP1","CSP16","CSP18","CSP19")){
      dt$SocioCateg[i]<-"CSP1"
      }
    if (dt$SocioCateg[i]%in%c("CSP2", "CSP20", "CSP21", "CSP22", "CSP23",
                                    "CSP25", "CSP26","CSP27", "CSP28")){
      dt$SocioCateg[i]<-"CSP2"
      }
    if (dt$SocioCateg[i]%in%c("CSP3", "CSP30", "CSP31", "CSP32", "CSP33",
                                    "CSP35", "CSP36","CSP37", "CSP38", "CSP39")){
      dt$SocioCateg[i]<-"CSP3"
      }
    if (dt$SocioCateg[i]%in%c("CSP40", "CSP41", "CSP42", "CSP43", "CSP46",
                                    "CSP47", "CSP48","CSP49")){
      dt$SocioCateg[i]<-"CSP4"
      }
    if (dt$SocioCateg[i]%in%c("CSP5", "CSP50", "CSP51", "CSP55", "CSP56",
                                    "CSP57", "CSP59")){
      dt$SocioCateg[i]<-"CSP5"
      }
```

```
  if (dt$SocioCateg[i]%in%c("CSP6", "CSP60", "CSP61", "CSP62", "CSP63",
                                "CSP65", "CSP66")){
    dt$SocioCateg[i]<-"CSP6"
    }
  if (dt$SocioCateg[i]%in%c("CSP7", "CSP70", "CSP73", "CSP74", "CSP77")){
    dt$SocioCateg[i]<-"CSP7"
    }
  if (dt$SocioCateg[i]%in%c("CSP9", "CSP91")){
    dt$SocioCateg[i]<-"CSP9"
    }
}
dt$SocioCateg <- droplevels(dt$SocioCateg)

# Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)
for (i in 1:dim(dt)[2]){
  # Type de véhicules
  if (colnames(dt)[i]=="VehBody"){
    levels(dt$VehBody) <- c(levels(dt$VehBody), "autobus", "coupé",
                                "autre microvan", "berline","SUV", "break",
                                "camionnette")
    dt$VehBody[dt$VehBody == "bus"]<-"autobus"
    dt$VehBody[dt$VehBody == "coupe"]<-"coupé"
    dt$VehBody[dt$VehBody == "other microvan"]<-"autre microvan"
    dt$VehBody[dt$VehBody == "sedan"]<-"berline"
    dt$VehBody[dt$VehBody == "sport utility vehicle"]<-"SUV"
    dt$VehBody[dt$VehBody == "station wagon"]<-"break"
    dt$VehBody[dt$VehBody == "van"]<-"camionnette"
    dt$VehBody <- droplevels(dt$VehBody)
    }
  # Statut marital
  if (colnames(dt)[i]=="MariStat"){
    levels(dt$MariStat) <- c(levels(dt$MariStat), "célibataire", "autre")
    dt$MariStat[dt$MariStat == "Alone"]<-"célibataire"
    dt$MariStat[dt$MariStat == "Other"]<-"autre"
    dt$MariStat <- droplevels(dt$MariStat)
    }
  # Utilisation du véhicule
  if (colnames(dt)[i]=="VehUsage"){
    levels(dt$VehUsage) <- c(levels(dt$VehUsage), "privée",
                                 "privée et trajet vers bureau", "professionnel",
                                 "trajet professionnel" )
    dt$VehUsage[dt$VehUsage == "Private"]<-"privée"
    dt$VehUsage[dt$VehUsage == "Private+trip to office"]<-
    "privée et trajet vers bureau"
    dt$VehUsage[dt$VehUsage == "Professional"]<-"professionnel"
    dt$VehUsage[dt$VehUsage == "Professional run"]<-
    "trajet professionnel"
    dt$VehUsage <- droplevels(dt$VehUsage)
    }
  # Moteur du véhicule
  if (colnames(dt)[i]=="VehEngine"){
    levels(dt$VehEngine) <- c(levels(dt$VehEngine),
                                  "injection directe surpuissante",
                                  "électrique", "injection surpuissante")
```

```
      dt$VehEngine[dt$VehEngine == "direct injection overpowered"]<-
      "injection directe surpuissante"
      dt$VehEngine[dt$VehEngine == "electric"]<-"électrique"
      dt$VehEngine[dt$VehEngine == "injection overpowered"]<-
      "injection surpuissante"
      dt$VehEngine <- droplevels(dt$VehEngine)
      }
    # Energie utilisée par le véhicule
    if (colnames(dt)[i]=="VehEnergy"){
      levels(dt$VehEnergy) <- c(levels(dt$VehEnergy), "électrique", "essence")
      dt$VehEnergy[dt$VehEnergy == "regular"]<-"essence"
      dt$VehEnergy[dt$VehEnergy == "eletric"]<-"électrique"
      dt$VehEnergy <- droplevels(dt$VehEnergy)
      }
    # Garage
    if (colnames(dt)[i]=="Garage"){
      levels(dt$Garage) <- c(levels(dt$Garage), "aucun", "garage indépendant",
                                    "concessionnaire")
      dt$Garage[dt$Garage == "None"]<-"aucun"
      dt$Garage[dt$Garage == "Private garage"]<-"garage indépendant"
      dt$Garage[dt$Garage == "Collective garage"]<-"concessionnaire"
      dt$Garage <- droplevels(dt$Garage)
    }
  }
  return (dt)
}
```

6.2 Affichage d'un exemple d'exécution de la fonction describe du package Hmisc

```
## freMPL2
##
##  21  Variables      47497  Observations
## ---------------------------------------------------------------------------
## Exposure
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0      755    0.999    0.437   0.3222    0.047    0.083
##      .25      .50      .75      .90      .95
##    0.187    0.416    0.666    0.833    0.916
##
## lowest : 0.003 0.005 0.006 0.008 0.009, highest: 0.994 0.996 0.997 0.998 1.000
## ---------------------------------------------------------------------------
## LicAge
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0      809        1    274.2    182.7       60       86
##      .25      .50      .75      .90      .95
##      141      246      396      500      566
##
## lowest :   0   1   2   3   4, highest: 887 912 914 930 940
## ---------------------------------------------------------------------------
## RecordBeg
##          n    missing   distinct       Info       Mean        Gmd        .05
##      47497          0        365      0.937 2004-04-19      128.7 2004-01-01
```

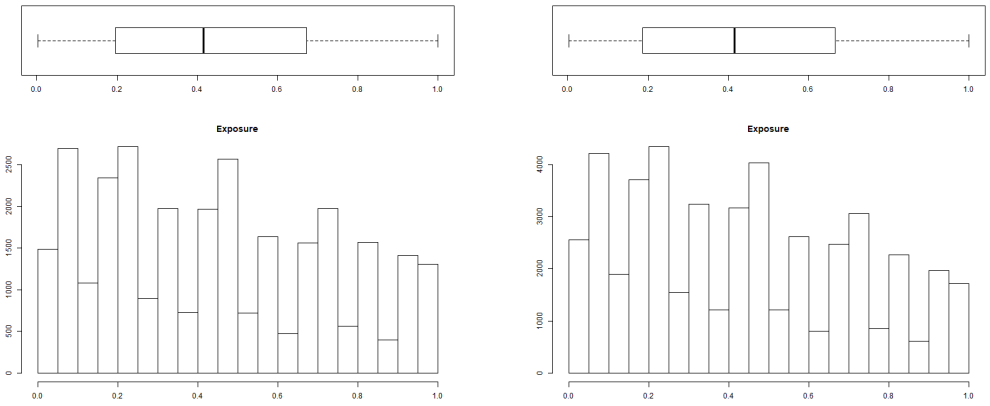
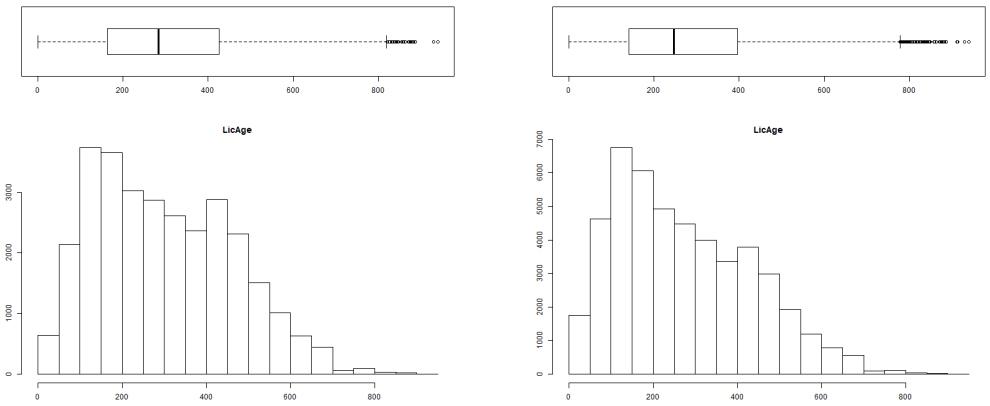
```
##        .10        .25        .50        .75        .90        .95
## 2004-01-01 2004-01-01 2004-03-11 2004-07-26 2004-10-29 2004-12-01
##
## lowest : 2004-01-01 2004-01-02 2004-01-03 2004-01-04 2004-01-05
## highest: 2004-12-26 2004-12-27 2004-12-28 2004-12-29 2004-12-30
## ---------------------------------------------------------------------------
## RecordEnd
##          n    missing   distinct       Info       Mean        Gmd        .05
##      25388      22109        364      0.999 2004-07-04      113.7 2004-02-01
##        .10        .25        .50        .75        .90        .95
## 2004-02-25 2004-04-07 2004-07-01 2004-10-01 2004-11-23 2004-12-01
##
## lowest : 2004-01-03 2004-01-04 2004-01-05 2004-01-06 2004-01-07
## highest: 2004-12-27 2004-12-28 2004-12-29 2004-12-30 2004-12-31
## ---------------------------------------------------------------------------
## VehAge
##        n  missing distinct
##    47497        0        9
##
## lowest : 0   1   10+ 2   3  , highest: 3   4   5   6-7 8-9
##
## Value          0     1   10+     2     3     4     5   6-7   8-9
## Frequency   4313  3987 14347  4140  3760  3658  3412  4909  4971
## Proportion 0.091 0.084 0.302 0.087 0.079 0.077 0.072 0.103 0.105
## ---------------------------------------------------------------------------
## MariStat
##        n  missing distinct
##    47497        0        2
##
## Value      célibataire       autre
## Frequency        13690       33807
## Proportion       0.288       0.712
## ---------------------------------------------------------------------------
## SocioCateg
##        n  missing distinct
##    47497        0        8
##
## lowest : CSP1 CSP2 CSP3 CSP5 CSP6, highest: CSP5 CSP6 CSP7 CSP9 CSP4
##
## Value       CSP1  CSP2  CSP3  CSP5  CSP6  CSP7  CSP9  CSP4
## Frequency   2366  1721   918 32894  5731    80     9  3778
## Proportion 0.050 0.036 0.019 0.693 0.121 0.002 0.000 0.080
## ---------------------------------------------------------------------------
## VehUsage
##        n  missing distinct
##    47497        0        4
##
## Value                            privée privée et trajet vers bureau
## Frequency                         16785                         22051
## Proportion                        0.353                         0.464
##
## Value                      professionnel         trajet professionnel
## Frequency                           7958                          703
## Proportion                         0.168                         0.015
```

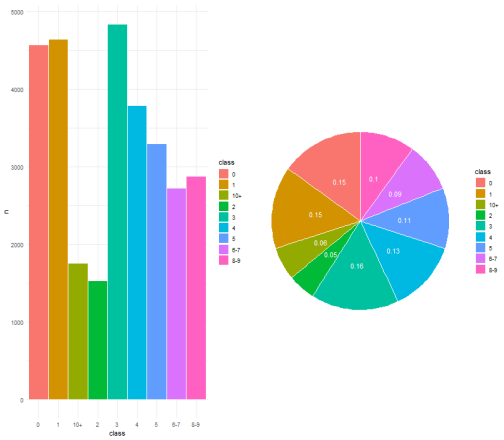
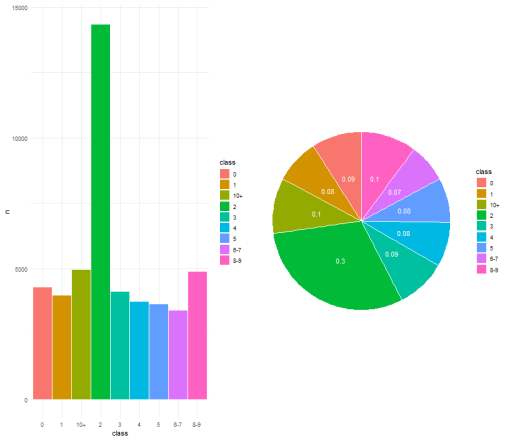
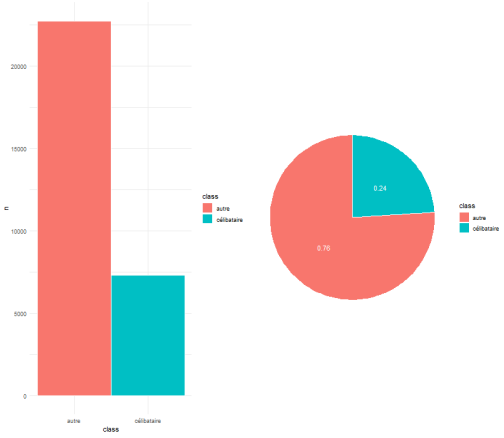
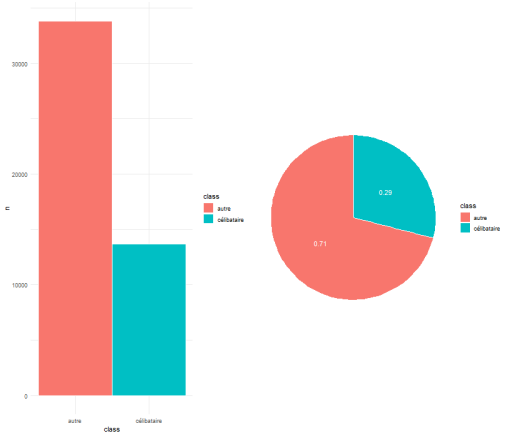
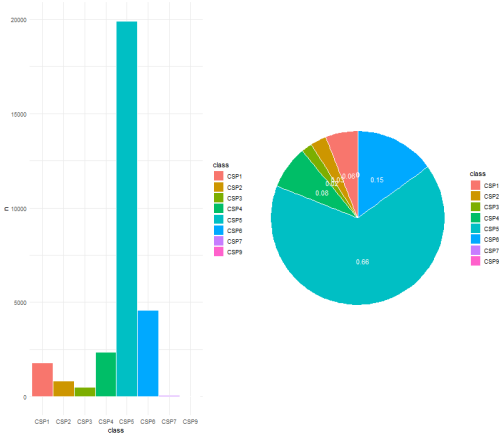
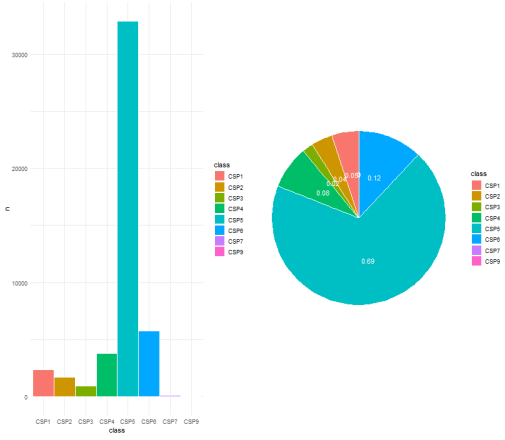
```
## ---------------------------------------------------------------------------
## DrivAge
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0       83        1    44.48    16.61       25       27
##      .25      .50      .75      .90      .95
##       32       42       55       65       72
##
## lowest :  18  19  20  21  22, highest:  96  97  98 102 103
## ---------------------------------------------------------------------------
## HasKmLimit
##        n  missing distinct     Info      Sum     Mean      Gmd
##    47497        0        2    0.353     6468   0.1362   0.2353
##
## ---------------------------------------------------------------------------
## BonusMalus
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0      108    0.954       69    21.99       50       50
##      .25      .50      .75      .90      .95
##       50       64       85      100      100
##
## lowest :  50  51  52  53  54, highest: 220 230 256 258 272
## ---------------------------------------------------------------------------
## VehBody
##        n  missing distinct
##    47497        0        9
##
## lowest : cabriolet      microvan       autobus        coupé          autre microvan
## highest: autre microvan berline        SUV            break          camionnette
##
## cabriolet (1506, 0.032), microvan (1458, 0.031), autobus (220, 0.005), coupé
## (1761, 0.037), autre microvan (1837, 0.039), berline (34051, 0.717), SUV (1974,
## 0.042), break (2231, 0.047), camionnette (2459, 0.052)
## ---------------------------------------------------------------------------
## VehPrice
##        n  missing distinct
##    47497        0       27
##
## lowest : A  B  C  D  E , highest: W  X  Y  Z  Z1
## ---------------------------------------------------------------------------
## VehEngine
##        n  missing distinct
##    47497        0        6
##
## lowest : carburation                   GPL                           injection
## highest: GPL                           injection                     injection directe surpuissante
##
## carburation (6513, 0.137), GPL (2, 0.000), injection (30663, 0.646), injection
## directe surpuissante (6554, 0.138), électrique (6, 0.000), injection
## surpuissante (3759, 0.079)
## ---------------------------------------------------------------------------
## VehEnergy
##        n  missing distinct
##    47497        0        4
##
```

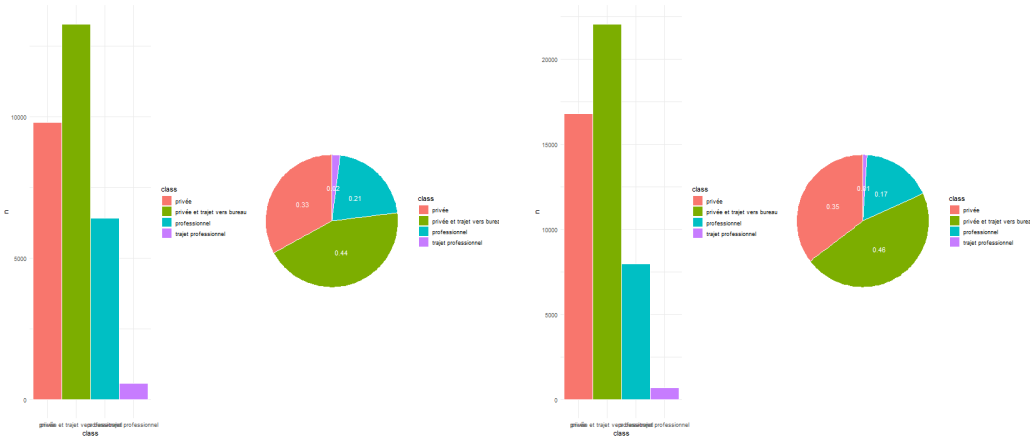
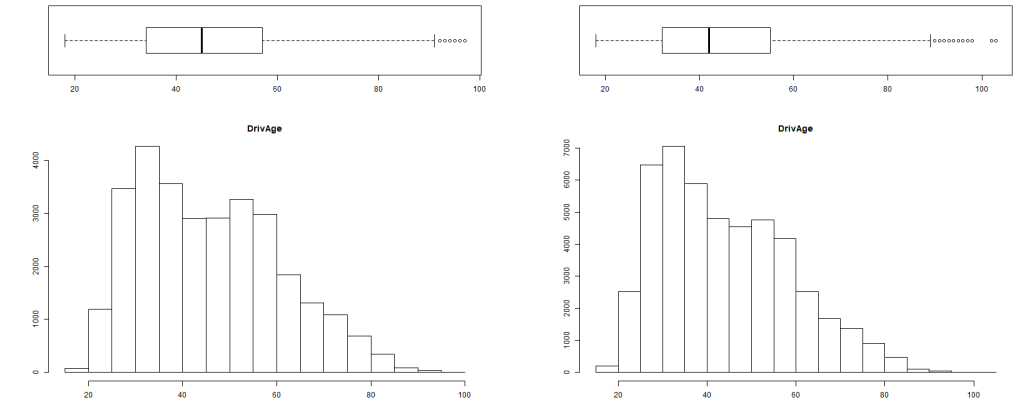
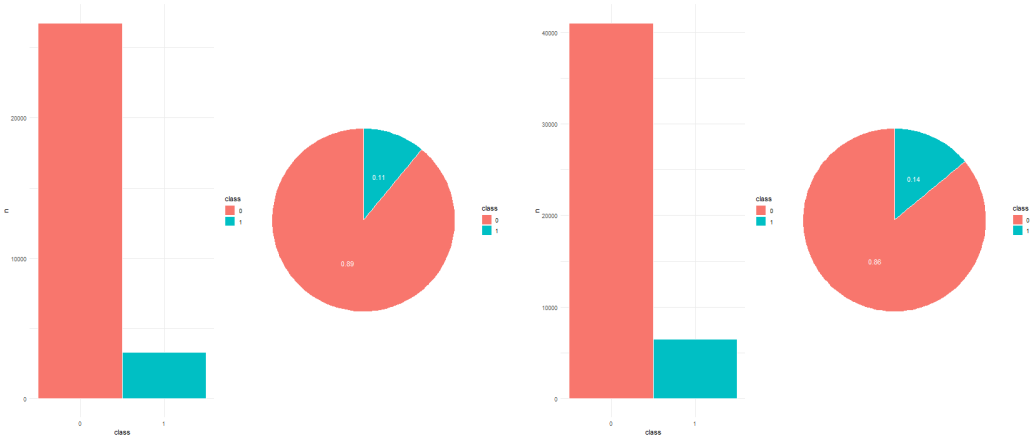
```
## Value          diesel        GPL électrique    essence
## Frequency       13521          2          6      33968
## Proportion      0.285      0.000      0.000      0.715
## ---------------------------------------------------------------------------
## VehMaxSpeed
##        n  missing distinct
##    47497        0       10
##
## lowest : 1-130 km/h   130-140 km/h 140-150 km/h 150-160 km/h 160-170 km/h
## highest: 170-180 km/h 180-190 km/h 190-200 km/h 200-220 km/h 220+ km/h
##
## Value        1-130 km/h 130-140 km/h 140-150 km/h 150-160 km/h 160-170 km/h
## Frequency          1256         2286         4073         7075         7915
## Proportion        0.026        0.048        0.086        0.149        0.167
##
## Value      170-180 km/h 180-190 km/h 190-200 km/h 200-220 km/h    220+ km/h
## Frequency          7933         5795         4567         3998         2599
## Proportion        0.167        0.122        0.096        0.084        0.055
## ---------------------------------------------------------------------------
## VehClass
##        n  missing distinct
##    47497        0        6
##
## lowest : 0  A  B  H  M1, highest: A  B  H  M1 M2
##
## Value          0     A     B     H    M1    M2
## Frequency   1901  4140 15229  7034 11756  7437
## Proportion 0.040 0.087 0.321 0.148 0.248 0.157
## ---------------------------------------------------------------------------
## RiskVar
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0       20    0.994    13.51    5.238        4        7
##      .25      .50      .75      .90      .95
##       11       15       17       19       20
##
## lowest :  1  2  3  4  5, highest: 16 17 18 19 20
##
## Value          1     2     3     4     5     6     7     8     9    10    11
## Frequency    590   501   754   700  1154  1041  1889  1630  1361  1513  2934
## Proportion 0.012 0.011 0.016 0.015 0.024 0.022 0.040 0.034 0.029 0.032 0.062
##
## Value         12    13    14    15    16    17    18    19    20
## Frequency   2896  3172  2496  5434  5632  4047  3078  3270  3405
## Proportion 0.061 0.067 0.053 0.114 0.119 0.085 0.065 0.069 0.072
## ---------------------------------------------------------------------------
## ClaimAmount
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    47497        0      873    0.129    86.83    170.3        0        0
##      .25      .50      .75      .90      .95
##        0        0        0        0        0
##
## lowest :      0.00      0.48      1.00      1.80      9.16
## highest:  57085.76  66892.58  80562.15  98152.44 120152.44
## ---------------------------------------------------------------------------
```

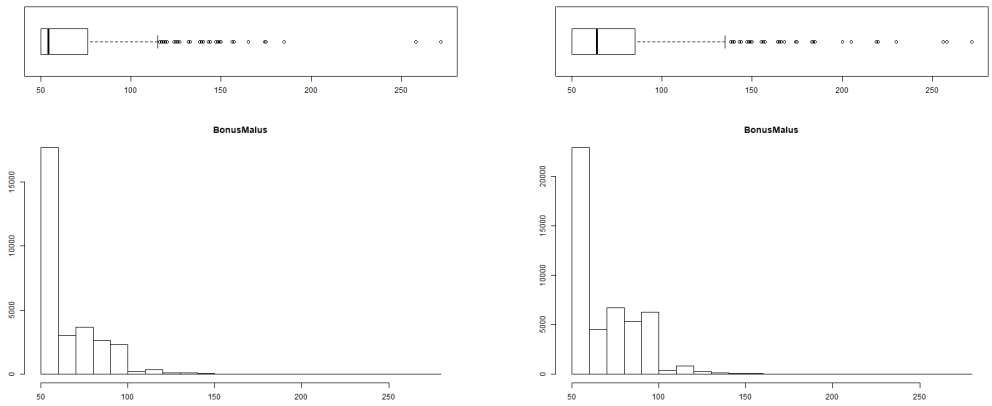
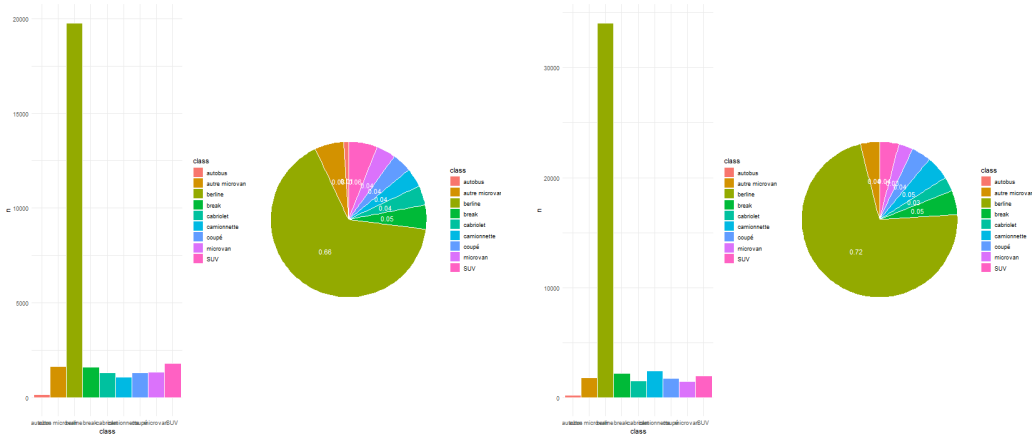
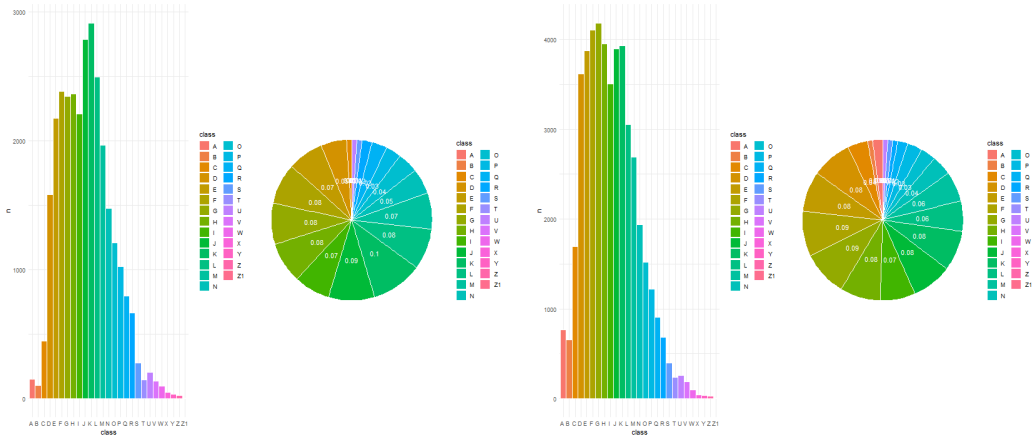
```
## Garage
##        n  missing distinct
##    47497        0        3
##
## Value                  aucun garage indépendant    concessionnaire
## Frequency                 35092                4642               7763
## Proportion                0.739               0.098              0.163
## ------------------------------------------------------------------------------
## ClaimInd
##        n  missing distinct     Info      Sum     Mean      Gmd
##    47497        0        2    0.129     2134  0.04493  0.08582
##
## ------------------------------------------------------------------------------
```

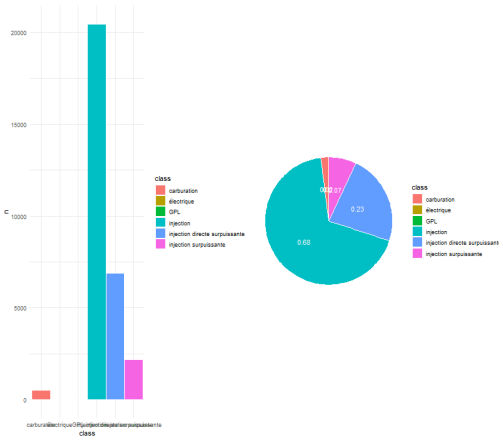
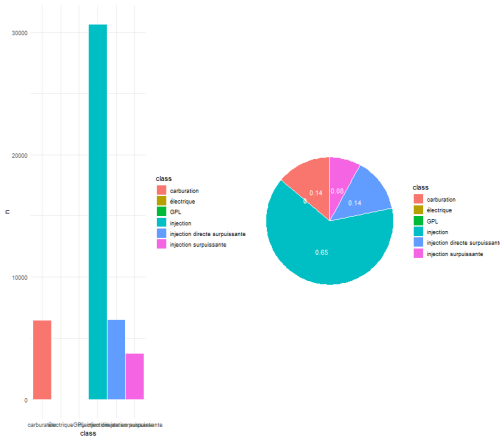
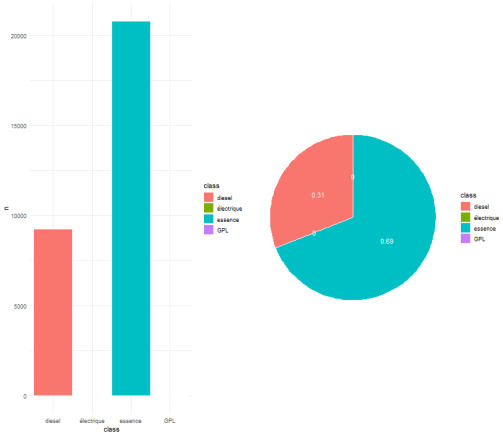
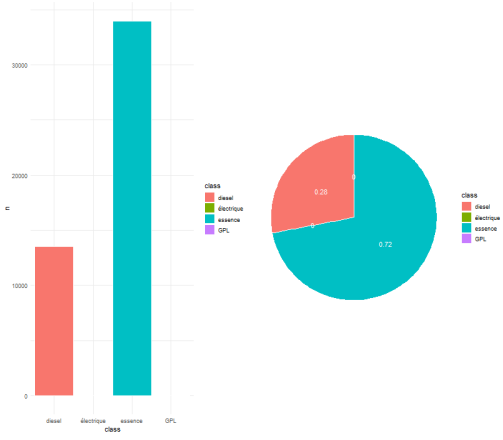
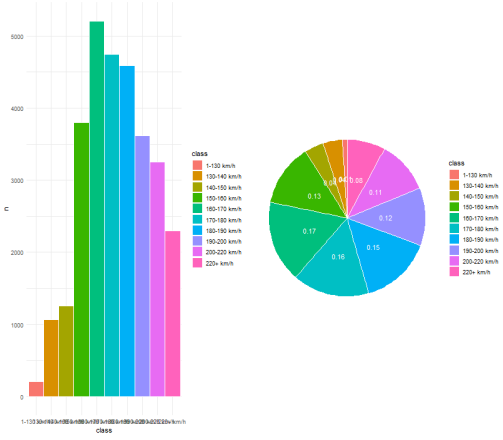
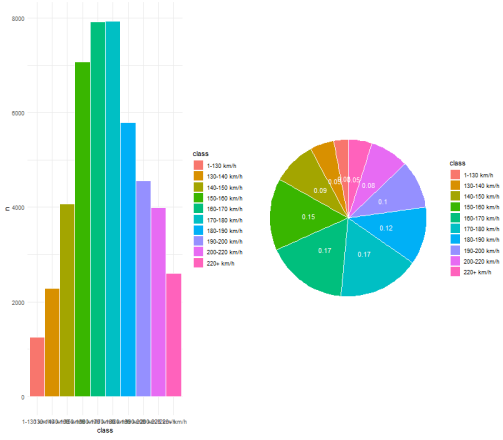
6.3 Affichage de l'ensemble des représentations graphiques

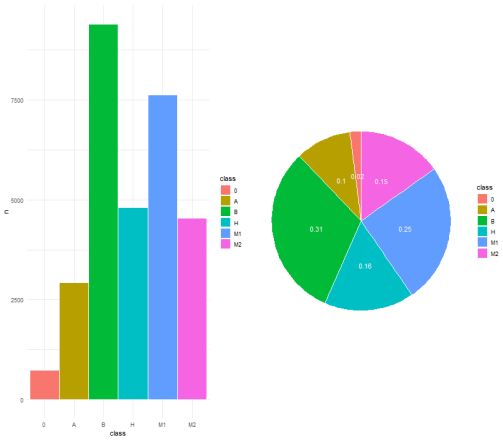
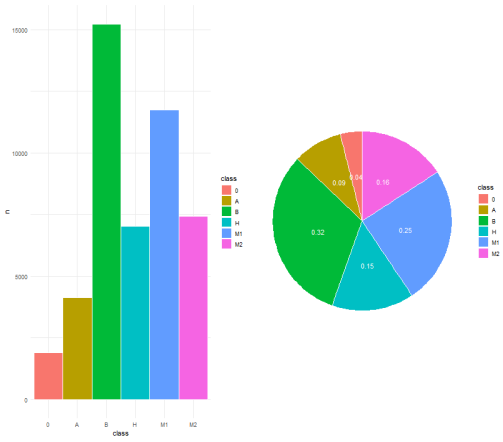
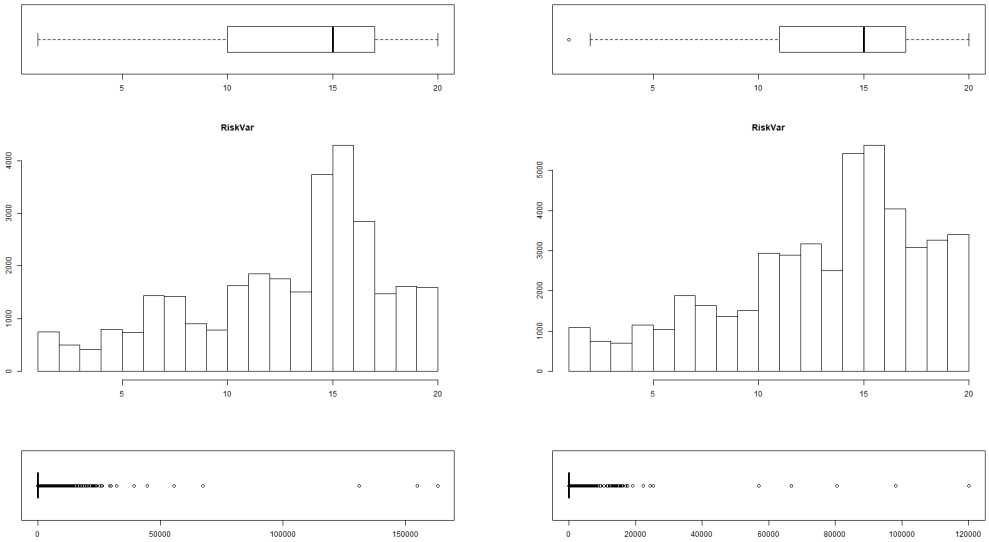
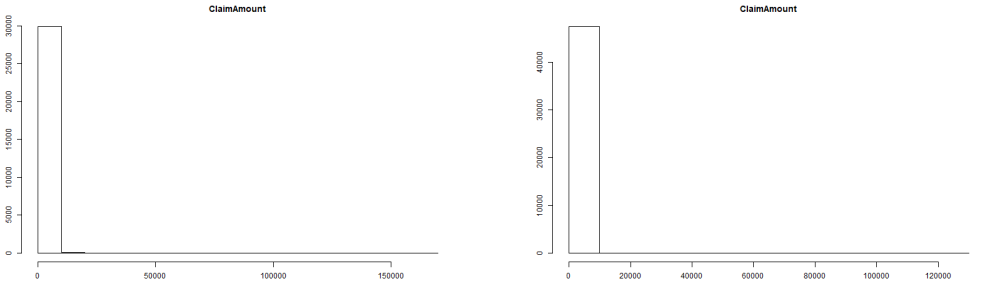
	freMPL1	freMPL2
Exposure		
LicAge		

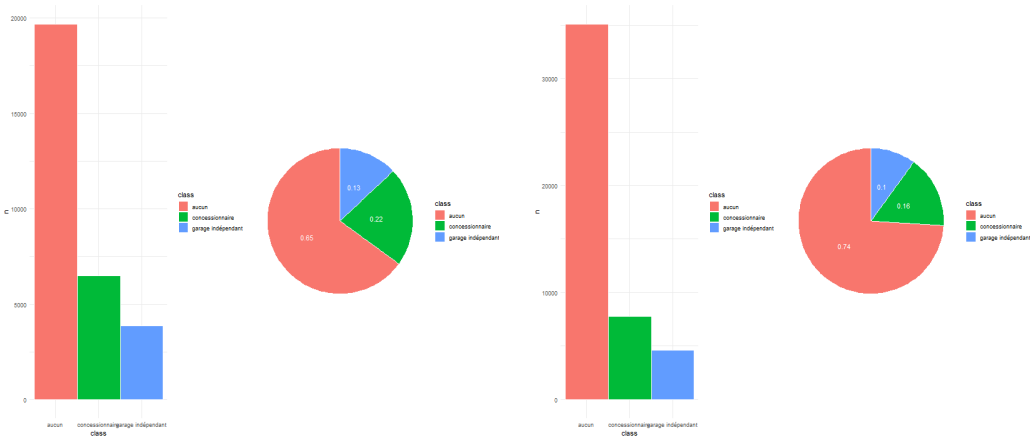
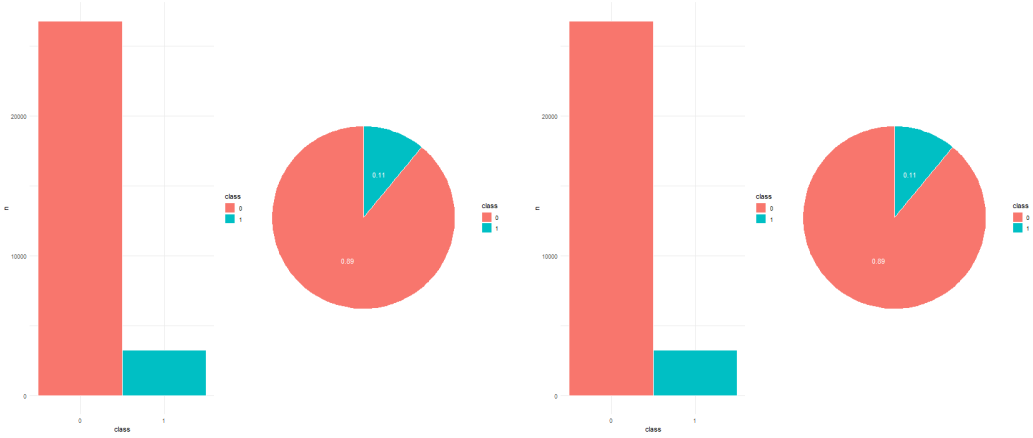
	freMPL1	freMPL2
VehAge		
MariStat		
SocioCateg		

	freMPL1	freMPL2
VehUsage		
DrivAge		
HasKmLimit		

	freMPL1	freMPL2
BonusMalus		
VehBody		
VehPrice		

	freMPL1	freMPL2
VehEngine		
VehEnergy		
VehMaxSpeed		

	freMPL1	freMPL2
VehClass		
RiskVar		
ClaimAmount		

	freMPL1	freMPL2
Garage		
ClaimInd		

6.4 Summary des modèles GLM testés

6.4.1 Modèle GLM pour fréquence n°= 1 :

term	estimate	p.value
(Intercept)	-14.5371138	0.0661785
Exposure	-97.4454095	0.0000502
LicAge(36,120]	-0.4274226	0.0270984
LicAge(120,240]	-0.8999444	0.0000107
LicAge(240,360]	-0.9975637	0.0000098
LicAge(360,480]	-1.2343094	0.0000009
LicAge(480,600]	-1.1079924	0.0000963
LicAge(600,720]	-1.3870787	0.0002390
LicAge(720,960]	-0.3357759	0.5262505
RecordBeg	-0.2643078	0.0000608
RecordEnd	0.2653423	0.0000521
VehAge1	0.0928512	0.5307565
VehAge10+	-0.0577577	0.6976274
VehAge2	0.2355487	0.0967943
VehAge3	0.0892504	0.5630879
VehAge4	0.1100626	0.4721830

term	estimate	p.value
VehAge5	0.0600270	0.7135042
VehAge6-7	-0.1177642	0.4495695
VehAge8-9	0.0223771	0.8851042
MariStatautre	0.1251064	0.1119550
SocioCategCSP2	0.2698013	0.2177705
SocioCategCSP3	0.4387593	0.0869583
SocioCategCSP5	0.1280679	0.3963413
SocioCategCSP6	0.0226095	0.9106107
SocioCategCSP7	0.1936441	0.7950467
SocioCategCSP9	2.4799265	0.0366922
SocioCategCSP4	0.0365919	0.8380416
VehUsageprivée et trajet vers bureau	0.1727977	0.0377543
VehUsageprofessionnel	0.2430745	0.0238778
VehUsagetrajet professionnel	0.3874621	0.1104137
DrivAge(20,30]	0.0312002	0.9410303
DrivAge(30,40]	0.3277161	0.4496951
DrivAge(40,50]	0.3830447	0.3887003
DrivAge(50,60]	0.5276206	0.2472137
DrivAge(60,70]	0.7187742	0.1319130
DrivAge(70,80]	0.5210198	0.3143430
DrivAge(80,103]	0.8935310	0.1232169
HasKmLimit1	-0.0429587	0.6745318
BonusMalus(100,350]	0.5267207	0.0000409
VehBodymicrovan	0.1342073	0.5789579
VehBodyautobus	-0.2173614	0.6639148
VehBodycoupé	-0.4384183	0.0604841
VehBodyautre microvan	-0.0432433	0.8483437
VehBodyberline	-0.3170252	0.0648152
VehBodySUV	-0.1889700	0.4122000
VehBodybreak	-0.6655514	0.0052004
VehBodycamionnette	-0.0445752	0.8590335
VehPriceB	-0.3289858	0.3962084
VehPriceC	-0.3698137	0.2381077
VehPriceD	-0.2593294	0.3629456
VehPriceE	-0.4096768	0.1594021
VehPriceF	-0.3248003	0.2701179
VehPriceG	-0.1203311	0.6902100
VehPriceH	-0.0947734	0.7615502
VehPriceI	-0.0727002	0.8212828
VehPriceJ	-0.0180442	0.9559398
VehPriceK	-0.0560452	0.8664322
VehPriceL	0.0486287	0.8878910
VehPriceM	0.0066992	0.9849796
VehPriceN	0.2239578	0.5436974
VehPriceO	0.4707025	0.2186845
VehPriceP	0.2099319	0.6083340
VehPriceQ	0.4091781	0.3488350
VehPriceR	-0.1634130	0.7391941
VehPriceS	0.1702197	0.7422748
VehPriceT	-0.8772786	0.2855695
VehPriceU	0.7740199	0.1280510
VehPriceV	-0.2188802	0.7637115

term	estimate	p.value
VehPriceW	-0.8576419	0.4387424
VehPriceX	0.7286590	0.4049130
VehPriceY	0.4331459	0.6284567
VehPriceZ	0.2873411	0.7996844
VehPriceZ1	-9.7875169	0.9579190
VehEngineGPL	-9.7277099	0.9662095
VehEngineinjection	0.2581627	0.0576310
VehEngineinjection directe surpuissante	0.1844513	0.4114301
VehEngineélectrique	2.3100438	0.0573746
VehEngineinjection surpuissante	0.2376131	0.2253493
VehEnergyessence	0.2644660	0.0347028
VehMaxSpeed130-140 km/h	0.3073270	0.2122095
VehMaxSpeed140-150 km/h	0.1864753	0.4749730
VehMaxSpeed150-160 km/h	0.3430838	0.1739194
VehMaxSpeed160-170 km/h	0.0528283	0.8415732
VehMaxSpeed170-180 km/h	0.1395786	0.6110917
VehMaxSpeed180-190 km/h	-0.2647504	0.3572604
VehMaxSpeed190-200 km/h	0.0187111	0.9494136
VehMaxSpeed200-220 km/h	-0.1510713	0.6249642
VehMaxSpeed220+ km/h	0.1121594	0.7475940
VehClassA	-0.3309431	0.1844906
VehClassB	-0.3176018	0.1435499
VehClassH	-0.2898008	0.2066987
VehClassM1	-0.3505731	0.0823459
VehClassM2	-0.2711091	0.2031215
RiskVar(4,8]	0.1192711	0.4779931
RiskVar(8,12]	0.1442821	0.3665566
RiskVar(12,16]	0.0662381	0.6646822
RiskVar(16,20]	0.2483412	0.1058307
Garagegarage indépendant	-0.1412016	0.2227603
Garageconcessionnaire	-0.1301959	0.1678623

6.4.2 Modèle GLM pour fréquence n°= 2 :

term	estimate	p.value
(Intercept)	7.4008188	0.0313432
Exposure	-0.4107231	0.0000420
LicAge(36,120]	-0.5013963	0.0002741
LicAge(120,240]	-1.0135217	0.0000000
LicAge(240,360]	-1.0337715	0.0000000
LicAge(360,480]	-1.1264340	0.0000000
LicAge(480,600]	-1.1252448	0.0000000
LicAge(600,720]	-1.1622387	0.0000114
LicAge(720,960]	-0.3558457	0.3773554
RecordBeg	-0.0006807	0.0121894
VehUsageprivée et trajet vers bureau	0.1677576	0.0032098
VehUsageprofessionnel	0.2320542	0.0012295
VehUsagetrajet professionnel	0.2317048	0.2076661
DrivAge(20,30]	-0.3099223	0.2574375
DrivAge(30,40]	0.0277059	0.9210963
DrivAge(40,50]	-0.0184779	0.9487449

term	estimate	p.value
DrivAge(50,60]	0.0887675	0.7641605
DrivAge(60,70]	0.2481255	0.4248226
DrivAge(70,80]	-0.1036408	0.7599400
DrivAge(80,103]	0.1622203	0.6841873
HasKmLimit1	-0.1154083	0.1259918
BonusMalus(100,350]	0.4728058	0.0000016
VehBodymicrovan	-0.0420086	0.8051183
VehBodyautobus	-0.4930079	0.2032957
VehBodycoupé	-0.0706163	0.6663497
VehBodyautre microvan	-0.1302188	0.4219627
VehBodyberline	-0.1856677	0.1320949
VehBodySUV	-0.0100828	0.9488913
VehBodybreak	-0.6246240	0.0002937
VehBodycamionnette	-0.1553241	0.3493435
VehEngineGPL	-8.4763186	0.9514713
VehEngineinjection	0.3819029	0.0000010
VehEngineinjection directe surpuissante	0.4812376	0.0000005
VehEngineélectrique	2.0483854	0.0659657
VehEngineinjection surpuissante	0.4144728	0.0001723
VehClassA	-0.4775388	0.0017465
VehClassB	-0.3923098	0.0033430
VehClassH	-0.1697628	0.2299550
VehClassM1	-0.3914327	0.0028873
VehClassM2	-0.3106034	0.0267901
RiskVar(4,8]	0.0881130	0.4597007
RiskVar(8,12]	0.1464974	0.1947564
RiskVar(12,16]	-0.0041769	0.9692561
RiskVar(16,20]	0.2193579	0.0430136