

Actuariat de l'Assurance Non-Vie # 7

A. Charpentier (UQAM & Université de Rennes 1)



ENSAE ParisTech, Octobre / Décembre 2016.

<http://freakonometrics.hypotheses.org>

Modélisation des coûts individuels

Références: Frees (2010), chapitre 13, de Jong & Heller (2008), chapitre 8, et Denuit & Charpentier (2005), chapitre 11.

```
1 > sinistre=read.table("http://freakonometrics.free.fr/
    sinistreACT2040.txt",header=TRUE,sep=";")
2 > contrat=read.table("http://freakonometrics.free.fr/contractACT2040.
    txt",header=TRUE,sep=";")
3 > contrat=contrat[,1:10]
4 > names(contrat)[10]="region"
5 > sinistre_D0=sinistre[(sinistre$garantie=="2D0")&(sinistre$cout>0),]
6 > sinistre_RC=sinistre[(sinistre$garantie=="1RC")&(sinistre$cout>0),]
7 > base_D0=merge(sinistre_D0,contrat)
8 > dim(base_D0)
9 [1] 1735    13
10 > base_RC=merge(sinistre_RC,contrat)
11 > dim(base_RC)
12 [1] 1924    13
```

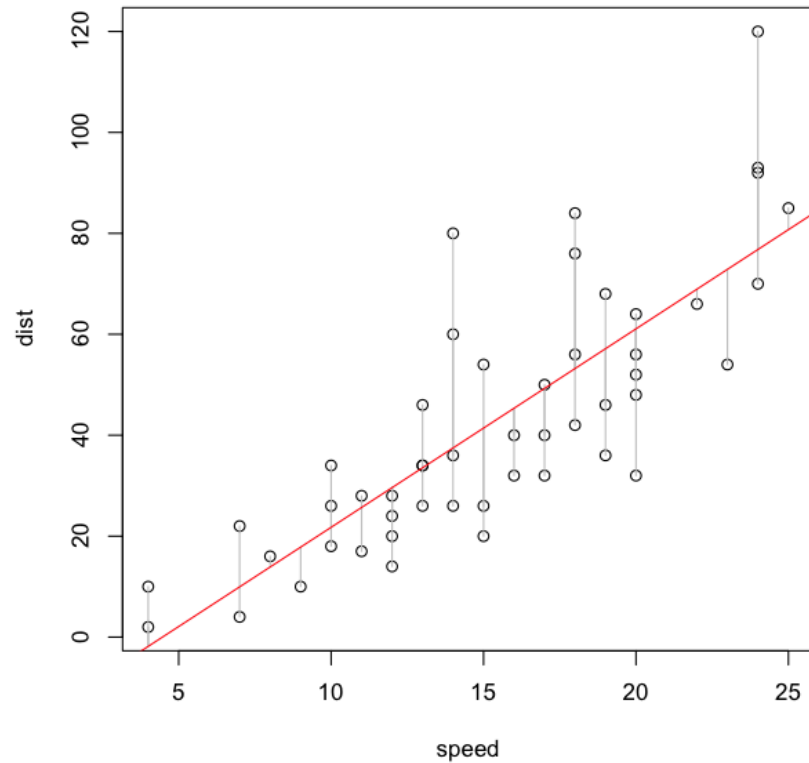
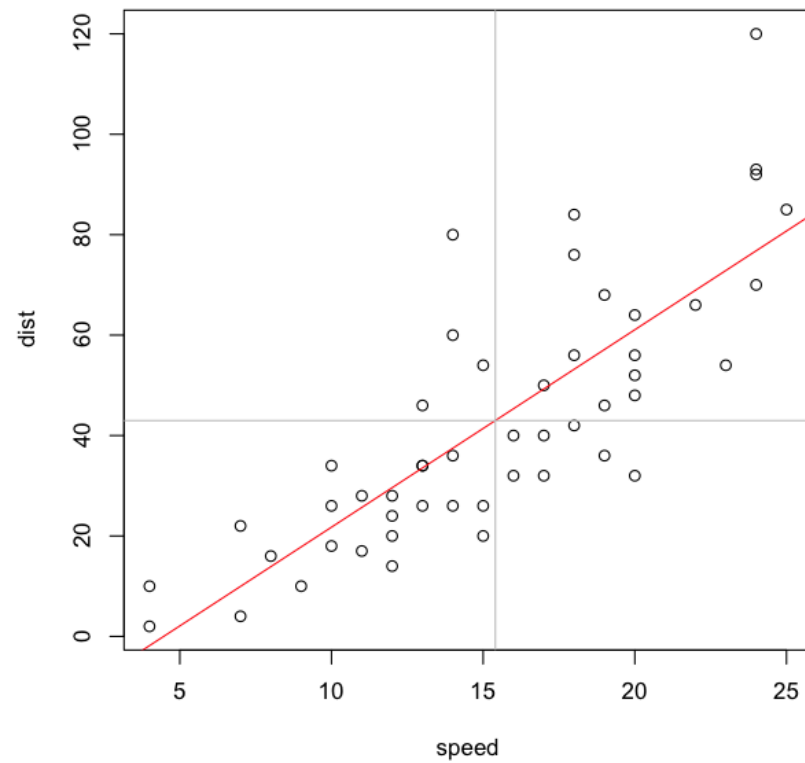
Modélisation des coûts individuels

Préambule: avec le **modèle linéaire**, nous avons $\sum_{i=1}^n Y_i = \sum_{i=1}^n \hat{Y}_i$

```
1 > reg=lm(dist~speed,data=cars)
2 > sum(cars$dist)
3 [1] 2149
4 > sum(predict(reg))
5 [1] 2149
```

C'est lié au fait que $\sum_{i=1}^n \hat{\varepsilon}_i = 0$, i.e.

“la droite de régression passe par le barycentre du nuage”.



Cette propriété était conservée avec la régression **log-Poisson**, nous avons que

$$\sum_{i=1}^n Y_i = \sum_{i=1}^n \hat{\mu}_i E_i, \text{ où } \hat{\mu}_i \cdot E_i \text{ est la prédiction faite avec l'exposition, au sens où}$$

```

1 > sum(freq$nombre_RC)
2 [1] 1924
3 > reg=glm(nombre_RC~1+offset(log(exposition)),data=freq,
4 + family=poisson(link="log"))
5 > sum(predict(reg,type="response"))
6 [1] 1924
7 > sum(predict(reg,newdata=data.frame(exposition=1),
8 + type="response")*freq$exposition)
9 [1] 1924

```

et ce, quel que soit le modèle utilisé !

```

1 > reg=glm(nombre_RC~offset(log(exposition))+ageconducteur+
2 + zone+carburant ,data=freq,family=poisson(link="log"))
3 > sum(predict(reg,type="response"))
4 [1] 1924

```

... mais c'est tout. En particulier, cette propriété n'est pas vérifiée si on change de fonction lien,

```
1 > reg=glm(nombre_RC~1+log(exposition),data=freq,
2 > sum(predict(reg,type="response"))
3 [1] 1977.704
```

ou de loi (e.g. binomiale négative),

```
1 > reg=glm.nb(nombre_RC~1+log(exposition),data=freq)
2 > sum(predict(reg,type="response"))
3 [1] 1925.053
```

Conclusion: de manière générale $\sum_{i=1}^n Y_i \neq \sum_{i=1}^n \hat{Y}_i$

La loi Gamma

La densité de Y est ici

$$f(y) = \frac{1}{y\Gamma(\phi^{-1})} \left(\frac{y}{\mu\phi}\right)^{\phi^{-1}} \exp\left(-\frac{y}{\mu\phi}\right), \quad \forall y \in \mathbb{R}_+$$

qui est dans la famille exponentielle, puisque

$$f(y) = \exp \left[\frac{y/\mu - (-\log \mu)}{-\phi} + \frac{1 - \phi}{\phi} \log y - \frac{\log \phi}{\phi} - \log \Gamma(\phi^{-1}) \right], \quad \forall y \in \mathbb{R}_+$$

On en déduit en particulier le **lien canonique**, $\theta = \mu^{-1}$ (fonction de lien inverse). De plus, $b(\theta) = -\log(\mu)$, de telle sorte que $b'(\theta) = \mu$ et $b''(\theta) = -\mu^2$. La **fonction variance** est alors ici $V(\mu) = \mu^2$.

Enfin, la déviance est ici

$$D = 2\phi[\log \mathcal{L}(y, y) - \log \mathcal{L}(\mu, y)] = 2\phi \sum_{i=1}^n \left(\frac{y_i - \mu_i}{\mu_i} - \log \left(\frac{y_i}{\mu_i} \right) \right).$$

La loi lognormale

La densité de Y est ici

$$f(y) = \frac{1}{y\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log y - \mu)^2}{2\sigma^2}\right), \quad \forall y \in \mathbb{R}_+$$

Si Y suit une loi lognormale de paramètres μ et σ^2 , alors $Y = \exp[Y^*]$ où $Y^* \sim \mathcal{N}(\mu, \sigma^2)$. De plus,

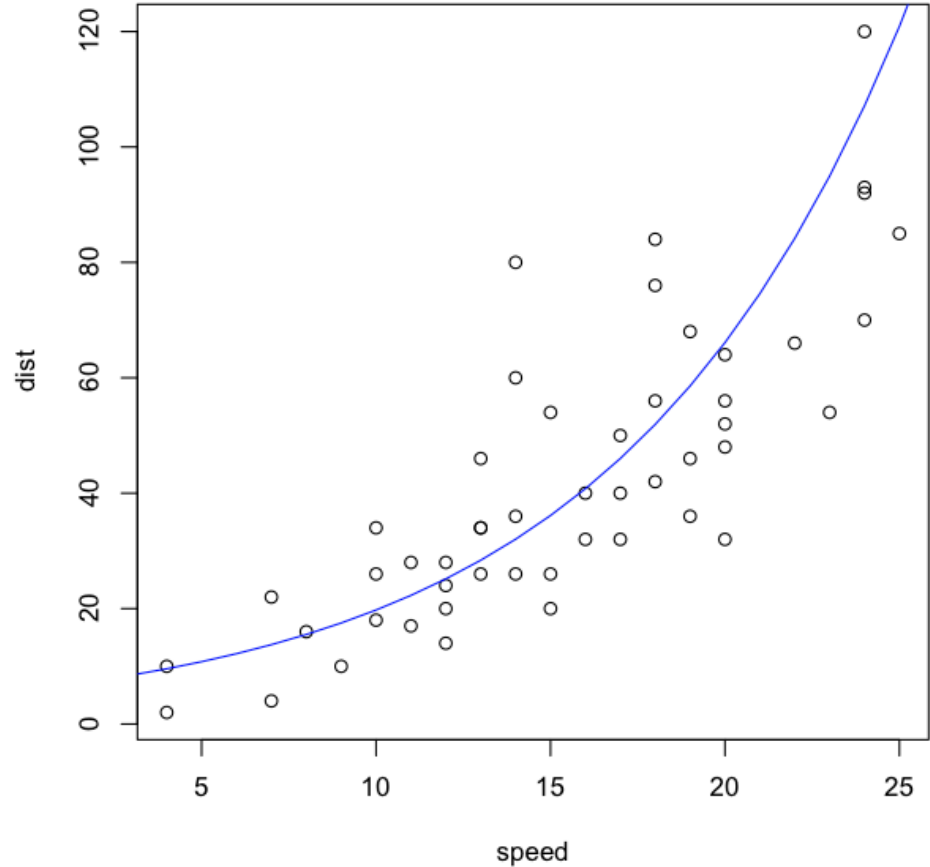
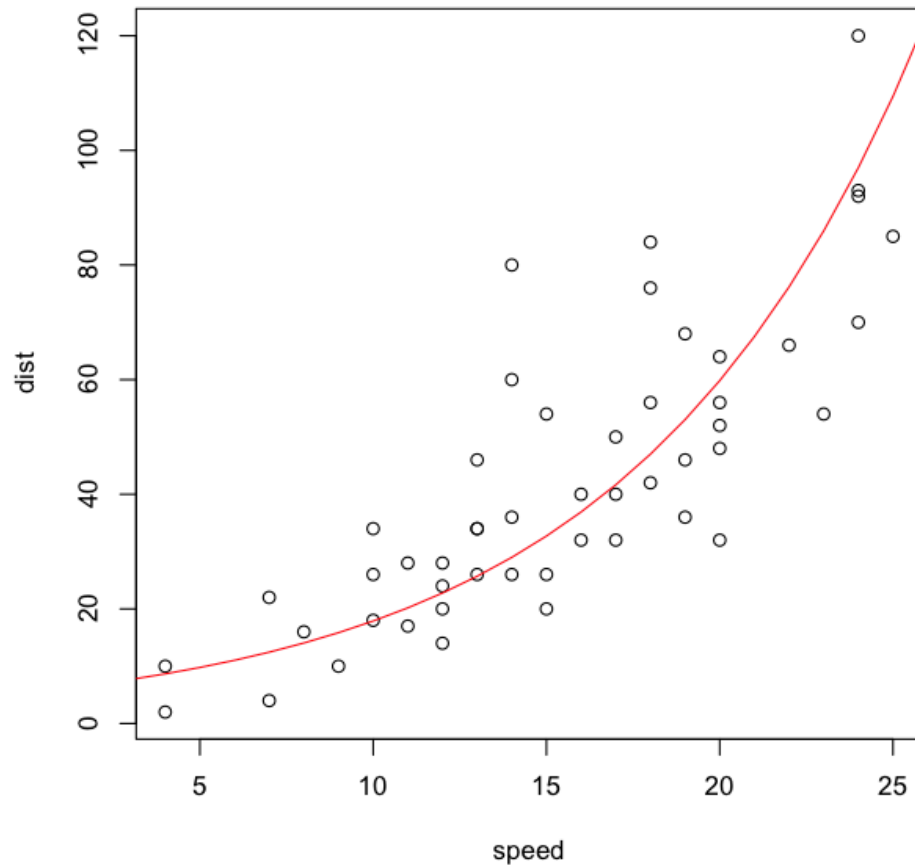
$$\mathbb{E}(Y) = \mathbb{E}(\exp[Y^*]) \neq \exp[\mathbb{E}(Y^*)] = \exp(\mu).$$

Rappelons que $\mathbb{E}(Y) = e^{\mu+\sigma^2/2}$, et $\text{Var}(Y) = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$.

```
1 > plot(cars)
2 > regln=lm(log(dist)~speed,data=cars)
3 > nouveau=data.frame(speed=1:30)
4 > preddist=exp(predict(regln,newdata=nouveau))
```



```
5 > lines(1:30, preddist, col="red")
6 > (s=summary(regln)$sigma)
7 [1] 0.4463305
8 > lines(1:30, preddist*exp(.5*s^2), col="blue")
```



Remarque là encore, $\sum_{i=1}^n Y_i \neq \sum_{i=1}^n \hat{Y}_i = \sum_{i=1}^n \exp \left(\hat{Y}_i^* + \frac{\sigma^2}{2} \right)$

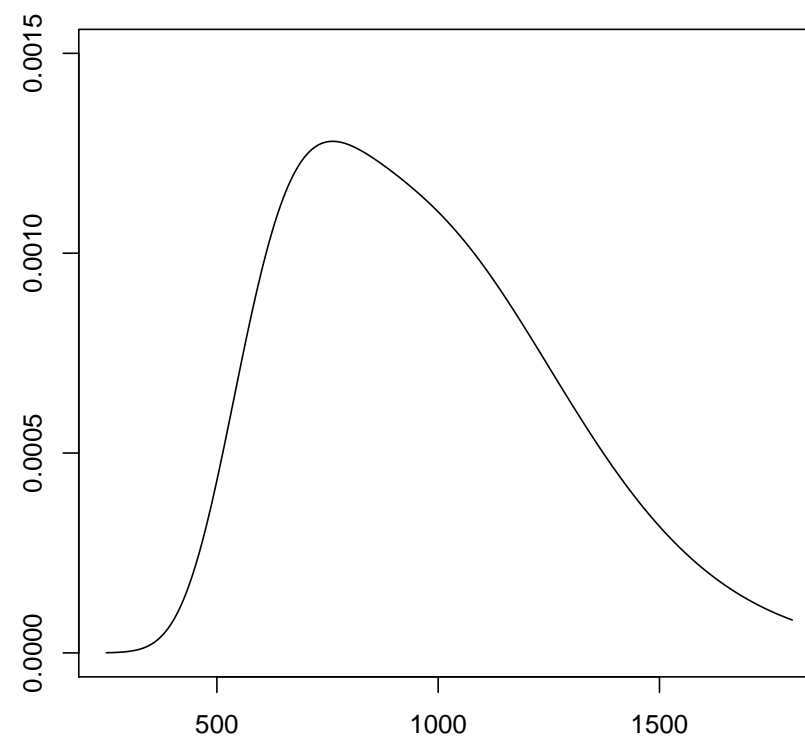
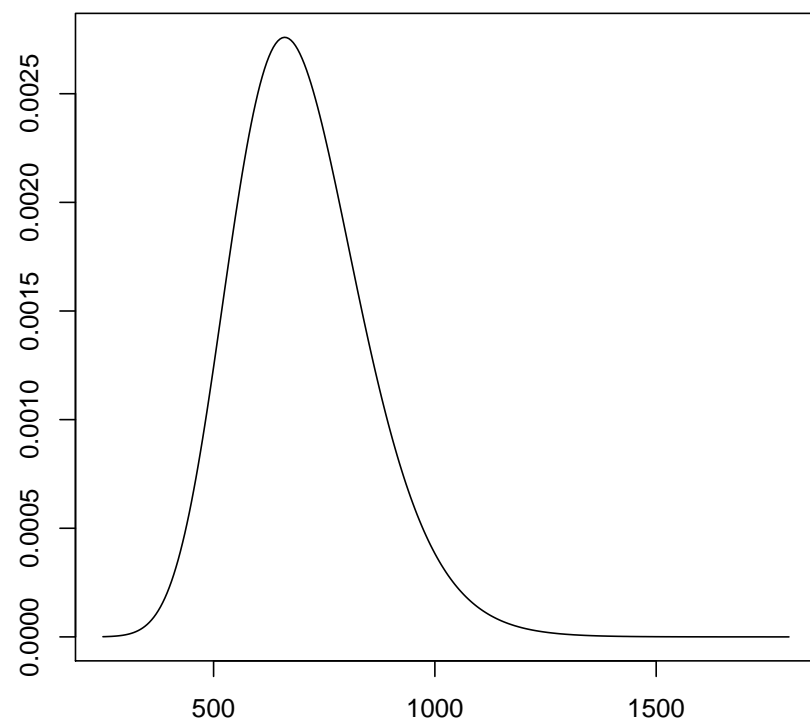
```
1 > sum(cars$dist)
2 [1] 2149
3 > sum(exp(predict(regln)))
4 [1] 2078.34
5 > sum(exp(predict(regln))*exp(.5*s^2))
6 [1] 2296.015
```

même si on ne régresse sur aucune variable explicative...

```
1 > regln=lm(log(dist)~1,data=cars)
2 > (s=summary(regln)$sigma)
3 [1] 0.7764719
4 > sum(exp(predict(regln))*exp(.5*s^2))
5 [1] 2320.144
```

(estimateur du maximum de vraisemblance \neq estimateur de la méthode des moments)

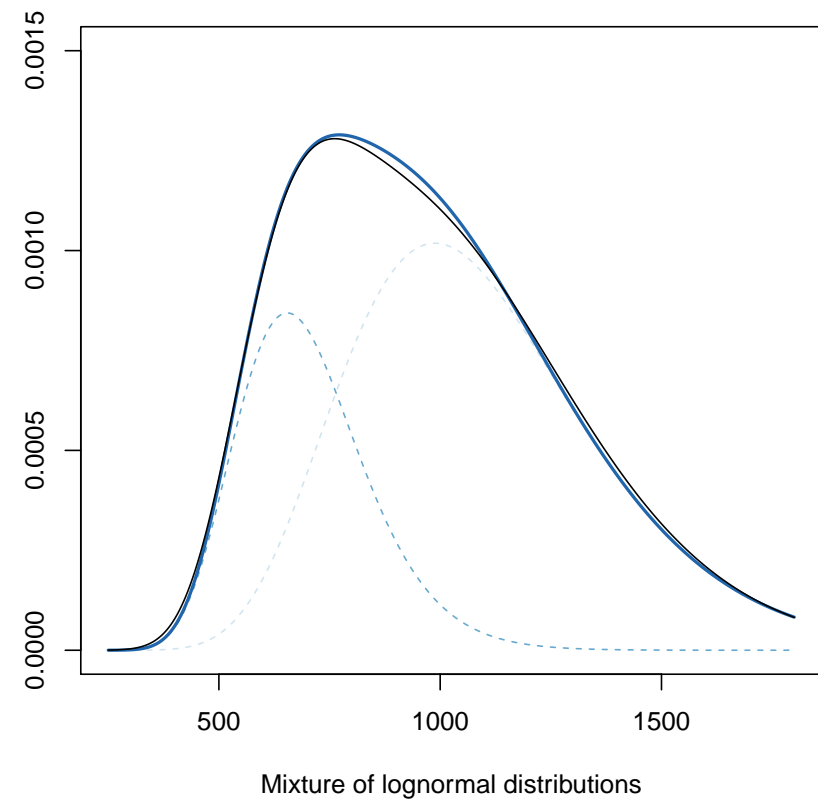
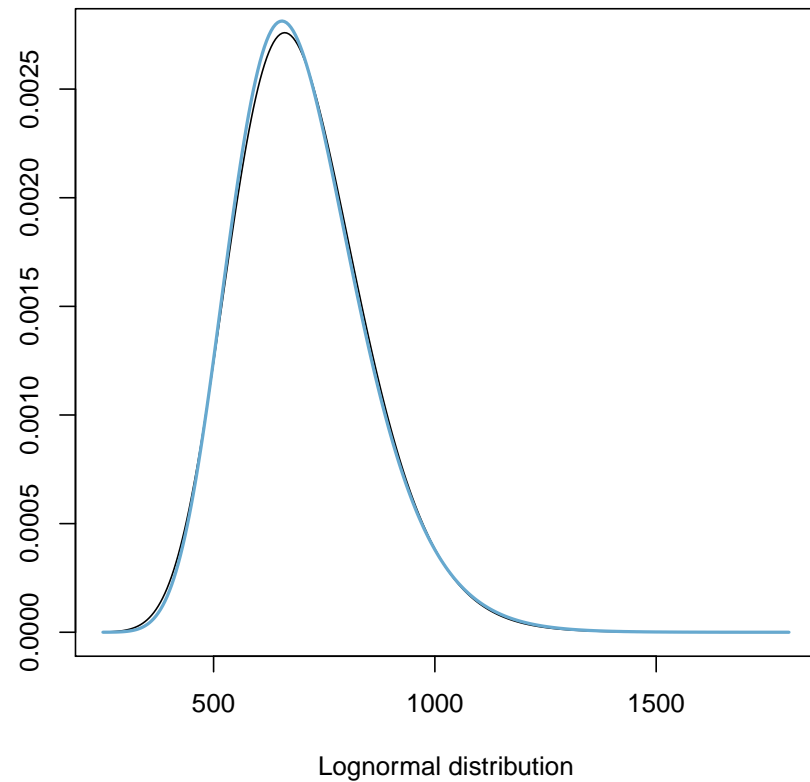
Loi Gamma ou loi lognormale ?



Mixture of two distributions

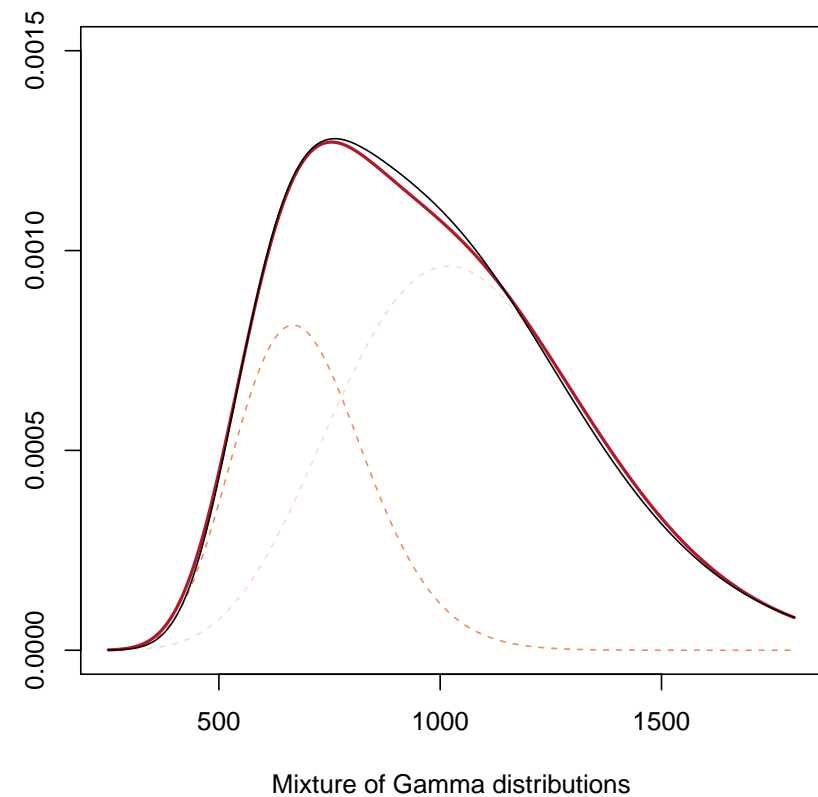
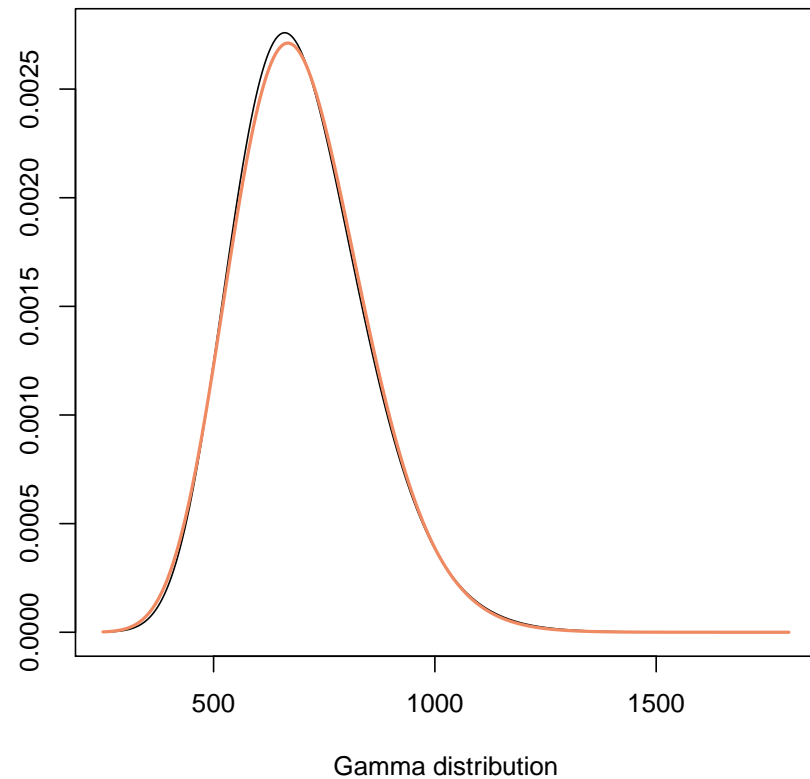
Loi Gamma ou loi lognormale ?

Loi Gamma ? Mélange de deux lois Gamma ?



Loi Gamma ou loi lognormale ?

Loi lognormale ? Mélange de deux lois lognormales ?



Autres lois possibles

Plusieurs autres lois sont possibles, au sein de la famille exponentielle, comme la loi inverse Gaussienne,

$$f(y) = \left[\frac{\lambda}{2\pi y^3} \right]^{1/2} \exp \left(\frac{-\lambda(y - \mu)^2}{2\mu^2 y} \right), \quad \forall y \in \mathbb{R}_+$$

de moyenne μ (qui est dans la famille exponentielle) ou la loi loi exponentielle

$$f(y) = \lambda \exp(-\lambda y), \quad \forall y \in \mathbb{R}_+$$

de moyenne λ^{-1} .

Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression **Gamma** (et un lien **log** i.e. $\mathbb{E}(Y|X) = \exp[X'\beta]$), on a

```
1 > regg=glm(cout~agevehicule+carburant+zone,data=base_RC,
2 +         family=Gamma(link="log"))
3 > summary(regg)
4 (Intercept)    7.72660    0.09300    83.079    < 2e-16 ***
5 agevehicule   -0.04674    0.00855    -5.466    5.27e-08 ***
6 carburantE     -0.14693    0.06329    -2.321    0.02038 *
7 zoneB         -0.14876    0.12690    -1.172    0.24124
8 zoneC         -0.04275    0.09924    -0.431    0.66668
9 zoneD         -0.11026    0.10416    -1.058    0.28998
10 zoneE         -0.12129    0.10478    -1.158    0.24719
11 zoneF         -0.47684    0.18142    -2.628    0.00865 **
12
13 (Dispersion parameter for Gamma family taken to be 1.686782)
```

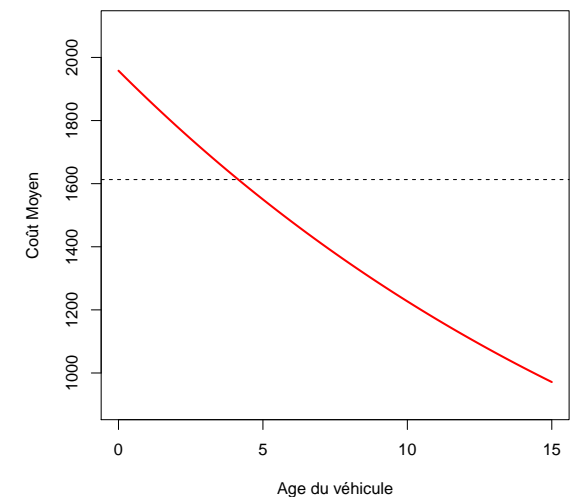
Les régressions Gamma

Régression Gamma, avec un lien logarithmique

```

1 > regg=glm(cout~agevehicule+carburant+zone,
  data=base_D0,family=Gamma(link="log"))
2 > nd=data.frame(agevehicule=seq(0,15,by=.25),
  carburant="E",zone="A")
3 > yp=predict(regg,newdata=nd,type="response")
4 > plot(seq(0,15,by=.25),yp,type="l",col="red",
  lwd=2)
5 > abline(h=mean(base_D0$cout),lty=2)

```



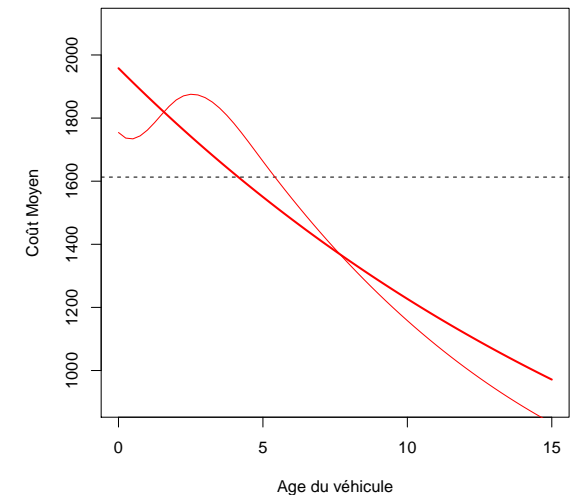
Les régressions Gamma

Régression Gamma, avec un lien logarithmique, avec du lissage (splines)

```

1 > library(splines)
2 > reggbs=glm(cout~bs(agevehicule)+carburant+
    zone,data=base_D0,family=Gamma(link="log"))
3 > yp=predict(regg,newdata=nd,type="response")
4 > plot(seq(0,15,by=.25),yp,type="l",col="red",
    lwd=2)

```



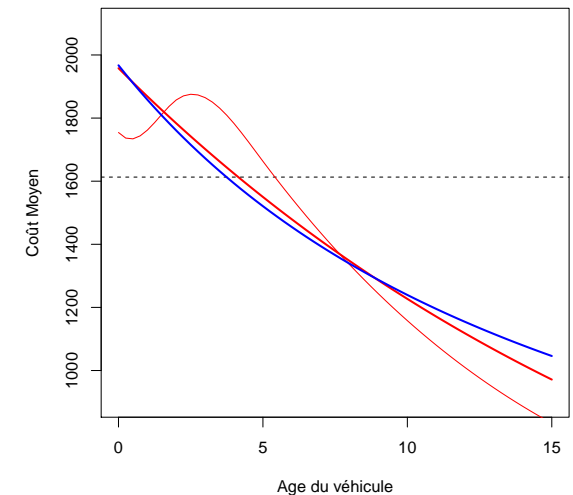
Les régressions Gamma

Régression Gamma, avec un lien inverse (lien canonique)

```

1 > regg=glm(cout~agevehicule+carburant+zone,
  data=base_D0,family=Gamma)
2 > nd=data.frame(agevehicule=seq(0,15,by=.25),
  carburant="E",zone="A")
3 > yp=predict(regg,newdata=nd,type="response")
4 > plot(seq(0,15,by=.25),yp,type="l",col="red",
  lwd=2)
5 > abline(h=mean(base_D0$cout),lty=2)

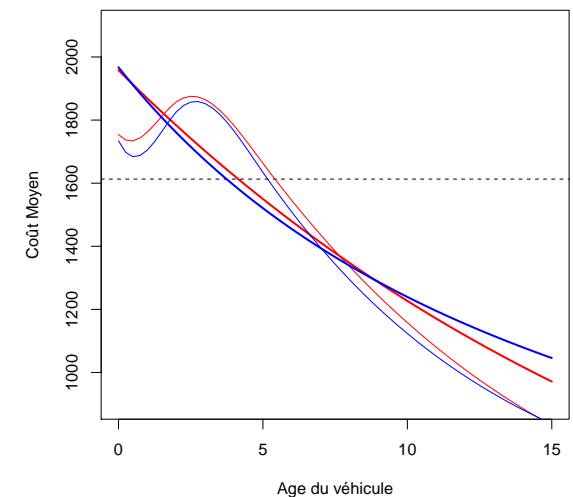
```



Les régressions Gamma

Régression Gamma, avec un lien inverse, avec du lissage (splines)

```
1 > library(splines)
2 > reggbs=glm(cout~bs(agevehicule)+carburant+
  zone ,data=base_D0,family=Gamma)
3 > yp=predict(regg,newdata=nd,type="response")
4 > plot(seq(0,15,by=.25),yp,type="l",col="red",
  lwd=2)
```



Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression **inverse-Gaussienne**, (et un lien **log** i.e. $\mathbb{E}(Y|X) = \exp[X'\beta]$),

```

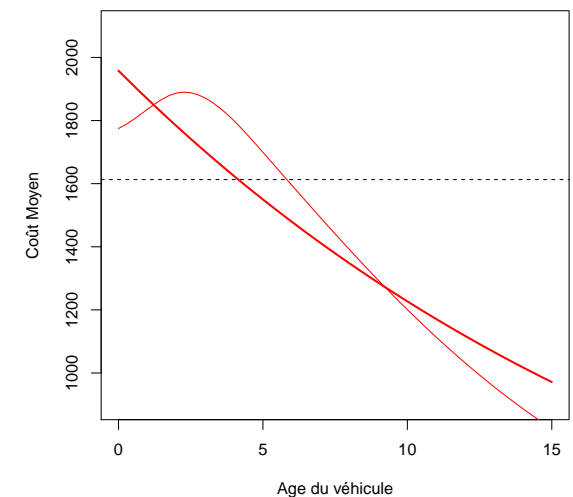
1 > regig=glm(cout~agevehicule+carburant+zone,data=base_D0,
2 +         family=inverse.gaussian(link="log"),start=coefficients(regg))
3 > summary(regig)
4 Coefficients:
5 (Intercept)    7.731661    0.093390    82.789    < 2e-16 ***
6 agevehicule   -0.046699    0.007016    -6.656  3.76e-11 ***
7 carburantE     -0.153028    0.061479    -2.489   0.01290 *
8 zoneB         -0.138902    0.123192    -1.128   0.25968
9 zoneC         -0.054040    0.098951    -0.546   0.58505
10 zoneD         -0.102103    0.102734    -0.994   0.32043
11 zoneE         -0.127266    0.103662    -1.228   0.21973
12 zoneF         -0.492622    0.155715    -3.164   0.00159 **
13
14 (Dispersion parameter for inverse.gaussian family taken to be
    0.001024064)

```

La régression Inverse Gaussienne

Régression Inverse Gaussienne, avec un lien logarithmique

```
1 > regig=glm(cout~agevehicule+carburant+zone,
  data=base_D0,family=inverse.gaussian(link=
    "log"),start=coefficients(regg))
2 > yp=predict(regig,newdata=nd,type="response")
3 > plot(seq(0,15,by=.25),yp,type="l",col="red",
  lwd=2)
4 > abline(h=mean(base_D0$cout),lty=2)
```



Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression log-normale i.e. $\mathbb{E}(\log Y|X) = X'\beta$, on a

```

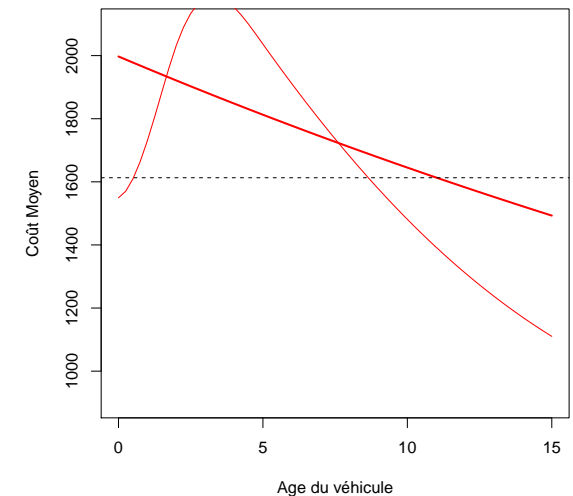
1 > regln=lm(log(cout)~agevehicule+carburant+zone,data=base_D0)
2 > summary(regln)
3 Coefficients:
4             Estimate Std. Error t value Pr(>|t|)
5 (Intercept)  6.776664   0.094371  71.809  <2e-16 ***
6 agevehicule -0.019397   0.008676  -2.236   0.0255 *
7 carburantE   -0.045508   0.064224  -0.709   0.4787
8 zoneB       -0.022196   0.128763  -0.172   0.8632
9 zoneC        0.056457   0.100695   0.561   0.5751
10 zoneD       -0.008894   0.105694  -0.084   0.9330
11 zoneE        0.017727   0.106321   0.167   0.8676
12 zoneF       -0.363002   0.184087  -1.972   0.0488 *
13
14 Residual standard error: 1.318 on 1727 degrees of freedom
15 > sigma=summary(regln)$sigma

```

La régression Log Normale

Régression Inverse Gaussienne, avec un lien logarithmique

```
1 > yp=exp(predict(regln,newdata=nd)+.5*sigma^2)
2 > plot(seq(0,15,by=.25),yp,type="l",col="red",
        lwd=2)
3 > abline(h=mean(base_D0$cout),lty=2)
```



Complément sur la régression Log Normale

Pour la régression **log-normale** on peut utiliser

```
1 > library(gamlss)
2 > regln=gamlss(cout~agevehicule+carburant+zone,data=base_D0,family=
        LOGNO(mu.link="identity"))
3 > summary(regln)
```

```

4  -----
5  Mu link function:  identity
6  Mu Coefficients:
7
8      Estimate Std. Error t value Pr(>|t|)
9  (Intercept)  6.776664   0.094153  71.975  <2e-16 ***
10 agevehicule -0.019397   0.008656  -2.241   0.0252 *
11 carburantE   -0.045508   0.064076  -0.710   0.4777
12 zoneB       -0.022196   0.128466  -0.173   0.8628
13 zoneC        0.056457   0.100463   0.562   0.5742
14 zoneD       -0.008894   0.105450  -0.084   0.9328
15 zoneE        0.017727   0.106076   0.167   0.8673
16 zoneF       -0.363002   0.183662  -1.976   0.0483 *
17 -----
18 Sigma link function:  log
19 Sigma Coefficients:
20
21      Estimate Std. Error t value Pr(>|t|)
22 (Intercept)  0.27370     0.01698  16.12  <2e-16 ***

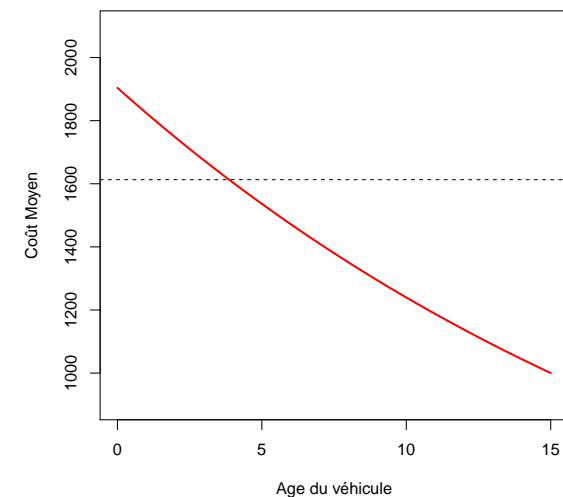
```


Régression Log Normale

```

1 > library(gamlss)
2 > yp=exp(predict(regln,what="mu",newdata=nd)
    +.5*(exp(regln$sigma.coefficients))^2)
3 > plot(seq(0,15,by=.25),yp,type="l",col="red",lwd=2)
4 > abline(h=mean(base_D0$cout),lty=2)

```

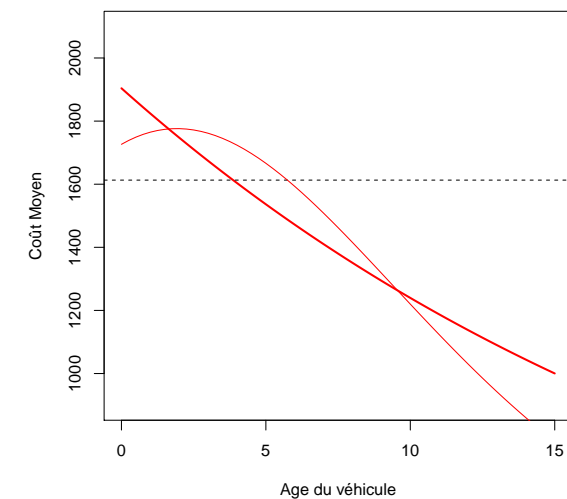


Au delà des lois de la famille exponentielle

Par exemple une loi de Weibull,

$$f(y_i|\mu_i, \sigma_i) = \frac{\sigma_i y_i^{\sigma_i-1}}{\mu_i^{\sigma_i}} \cdot \exp[-(y/\mu_i)^{\sigma_i}] \text{ avec } \begin{cases} \mu_i = \exp[\mathbf{x}_i^\top \boldsymbol{\alpha}] \\ \sigma_i = \exp[\mathbf{x}_i^\top \boldsymbol{\beta}] \end{cases}$$

```
1 > regweib=gamlss(cout~agevehicule+carburant+
  zone ,data=base_DO ,family=WEI(mu.link="
  log",sigma.link="log"))
2 > mu=exp(predict(regweib,what="mu",newdata=
  nd))
3 > sigma=exp(predict(regweib,what="sigma",
  newdata=nd))
4 > yp=mu*gamma((1/sigma)+1)
```



Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression **Gamma** (et un lien **log** i.e. $\mathbb{E}(Y|\mathbf{X}) = \exp[\mathbf{X}'\boldsymbol{\beta}]$), on a

```

1 > regg=glm(cout~agevehicule+carburant+zone,data=base_RC,
2 +         family=Gamma(link="log"))
3 > summary(regg)
4 Coefficients:
5             Estimate Std. Error t value Pr(>|t|)
6 (Intercept)   8.17615    0.22937  35.646  <2e-16 ***
7 agevehicule  -0.01715    0.01360  -1.261   0.2073
8 carburantE    -0.20756    0.14725  -1.410   0.1588
9 zoneB        -0.60169    0.30708  -1.959   0.0502 .
10 zoneC        -0.60072    0.24201  -2.482   0.0131 *
11 zoneD        -0.45611    0.24744  -1.843   0.0654 .
12 zoneE        -0.43725    0.24801  -1.763   0.0781 .
13 zoneF         0.24778    0.44852   0.552   0.5807
14
15 (Dispersion parameter for Gamma family taken to be 9.91334)

```

Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression **inverse-Gaussienne**, (et un lien **log** i.e. $\mathbb{E}(Y|X) = \exp[X'\beta]$),

```

1 > regig=glm(cout~agevehicule+carburant+zone,data=base_RC,
2 +         family=inverse.gaussian(link="log"),start=coefficients(regg))
3 > summary(regig)
4 Coefficients:
5             Estimate Std. Error t value Pr(>|t|)
6 (Intercept)  8.07065    0.23606  34.188  <2e-16 ***
7 agevehicule -0.01509    0.01118  -1.349   0.1774
8 carburantE   -0.18037    0.13065  -1.381   0.1676
9 zoneB       -0.50202    0.28836  -1.741   0.0819 .
10 zoneC       -0.50913    0.24098  -2.113   0.0348 *
11 zoneD       -0.38080    0.24806  -1.535   0.1249
12 zoneE       -0.36541    0.24975  -1.463   0.1436
13 zoneF        0.42854    0.56537   0.758   0.4486
14
15 (Dispersion parameter for inverse.gaussian family taken to be
    0.004331898)

```

Les régressions Gamma, lognormale et inverse Gaussienne

Pour la régression log-normale i.e. $\mathbb{E}(\log Y|X) = X'\beta$, on a

```
1 > regln=lm(log(cout)~agevehicule+carburant+zone,data=base_RC)
2 > summary(regln)
3 Coefficients:
4             Estimate Std. Error t value Pr(>|t|)
5 (Intercept)  6.876142   0.086483   79.508  <2e-16 ***
6 agevehicule -0.007032   0.005127   -1.372    0.170
7 carburantE   -0.042338   0.055520   -0.763    0.446
8 zoneB        0.080288   0.115784    0.693    0.488
9 zoneC        0.015060   0.091250    0.165    0.869
10 zoneD       0.099338   0.093295    1.065    0.287
11 zoneE       0.004305   0.093512    0.046    0.963
12 zoneF      -0.101866   0.169111   -0.602    0.547
```

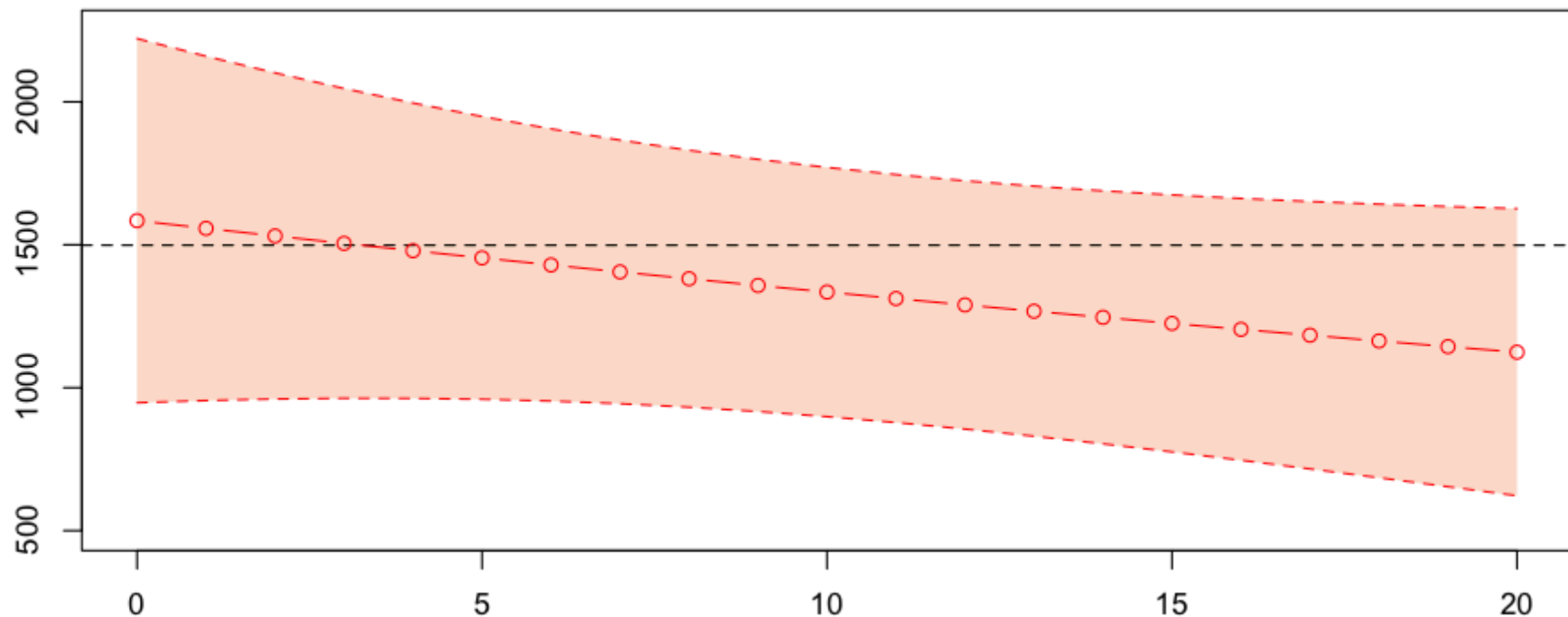
Les régressions Gamma, lognormale et inverse Gaussienne

On peut comparer les prédictions (éventuellement en fixant quelques covariables),

```
1 > nouveau=data.frame(agevehicule=0:20, carburant="E", zone="C")
2 > s=summary(regln)$sigma
3 > predln=predict(regln, se.fit=TRUE, newdata=nouveau)
4 > predg=predict(regg, se.fit=TRUE, type="response", newdata=nouveau)
5 > predig=predict(regig, se.fit=TRUE, type="response", newdata=nouveau)
```

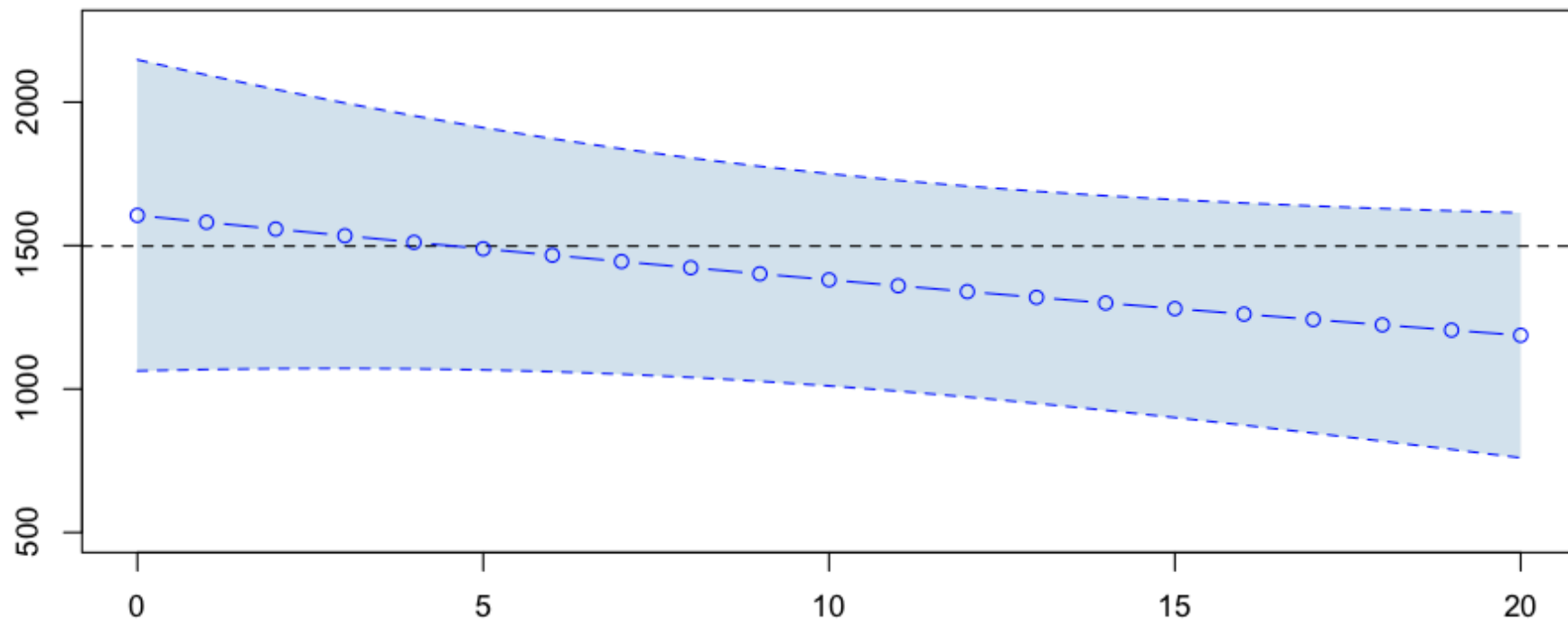
Pour le modèle **log-Gamma**, on a

```
1 > plot(0:20, predg$fit, type="b", col="red")  
2 > lines(0:20, predg$fit+2*predg$se.fit, lty=2, col="red")  
3 > lines(0:20, predg$fit-2*predg$se.fit, lty=2, col="red")
```



Pour le modèle log-inverse Gaussienne, on a

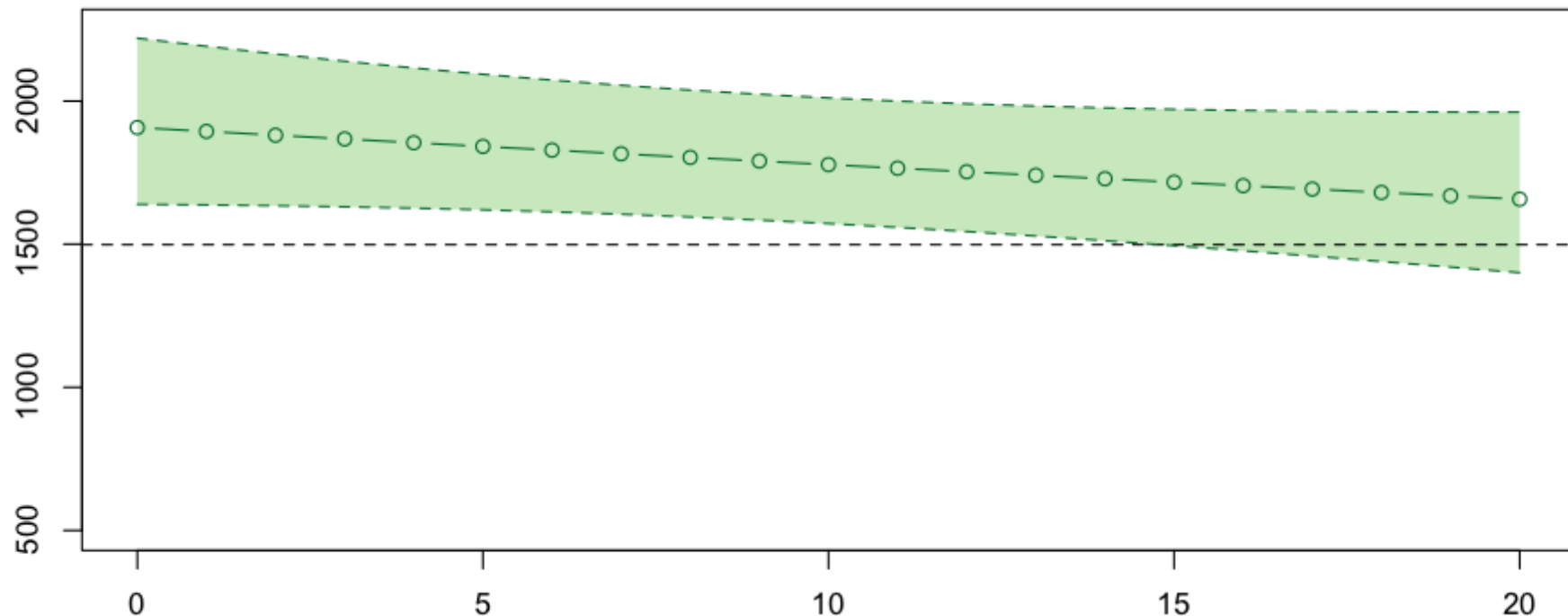
```
1 > plot(0:20, predig$fit, type="b", col="blue")  
2 > lines(0:20, predig$fit+2*predg$se.fit, lty=2, col="blue")  
3 > lines(0:20, predig$fit-2*predg$se.fit, lty=2, col="blue")
```



Pour le modèle `lognormal`, on a

```
1 > plot(0:20, exp(predln$fit+.5*s^2), type="b", col="green")
2 > lines(0:20, exp(predln$fit+.5*s^2+2*predln$se.fit), lty=2, col="green"
   )
3 > lines(0:20, exp(predln$fit+.5*s^2-2*predln$se.fit), lty=2, col="green"
   )
```

(les intervalles de confiance sur \hat{Y} n'ont pas trop de sens ici...)



Prise en compte des gros sinistres

On a ici quelques *gros* sinistres. L'idée est de noter que

$$\mathbb{E}(Y) = \sum_i \mathbb{E}(Y|\Theta = \theta_i) \cdot \mathbb{P}(\Theta = \theta_i)$$

Supposons que Θ prenne deux valeurs, correspondant au cas $\{Y \leq s\}$ et $\{Y > s\}$.
Alors

$$\mathbb{E}(Y) = \mathbb{E}(Y|Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s)$$

ou, en calculant l'espérance sous $\mathbb{P}_{\mathbf{X}}$ et plus \mathbb{P} ,

$$\mathbb{E}(Y|\mathbf{X}) = \underbrace{\mathbb{E}(Y|\mathbf{X}, Y \leq s)}_A \cdot \underbrace{\mathbb{P}(Y \leq s|\mathbf{X})}_B + \underbrace{\mathbb{E}(Y|Y > s, \mathbf{X})}_C \cdot \underbrace{\mathbb{P}(Y > s|\mathbf{X})}_B$$

Prise en compte des gros sinistres

Trois termes apparaissent dans

$$\mathbb{E}(Y|\mathbf{X}) = \underbrace{\mathbb{E}(Y|\mathbf{X}, Y \leq s)}_A \cdot \underbrace{\mathbb{P}(Y \leq s|\mathbf{X})}_B + \underbrace{\mathbb{E}(Y|Y > s, \mathbf{X})}_C \cdot \underbrace{\mathbb{P}(Y > s|\mathbf{X})}_B$$

- le coût moyen des sinistres normaux, A
- la probabilité d'avoir un gros, ou un sinistre normal, si un sinistre survient, B
- le coût moyen des sinistres importants, C

Prise en compte des gros sinistres

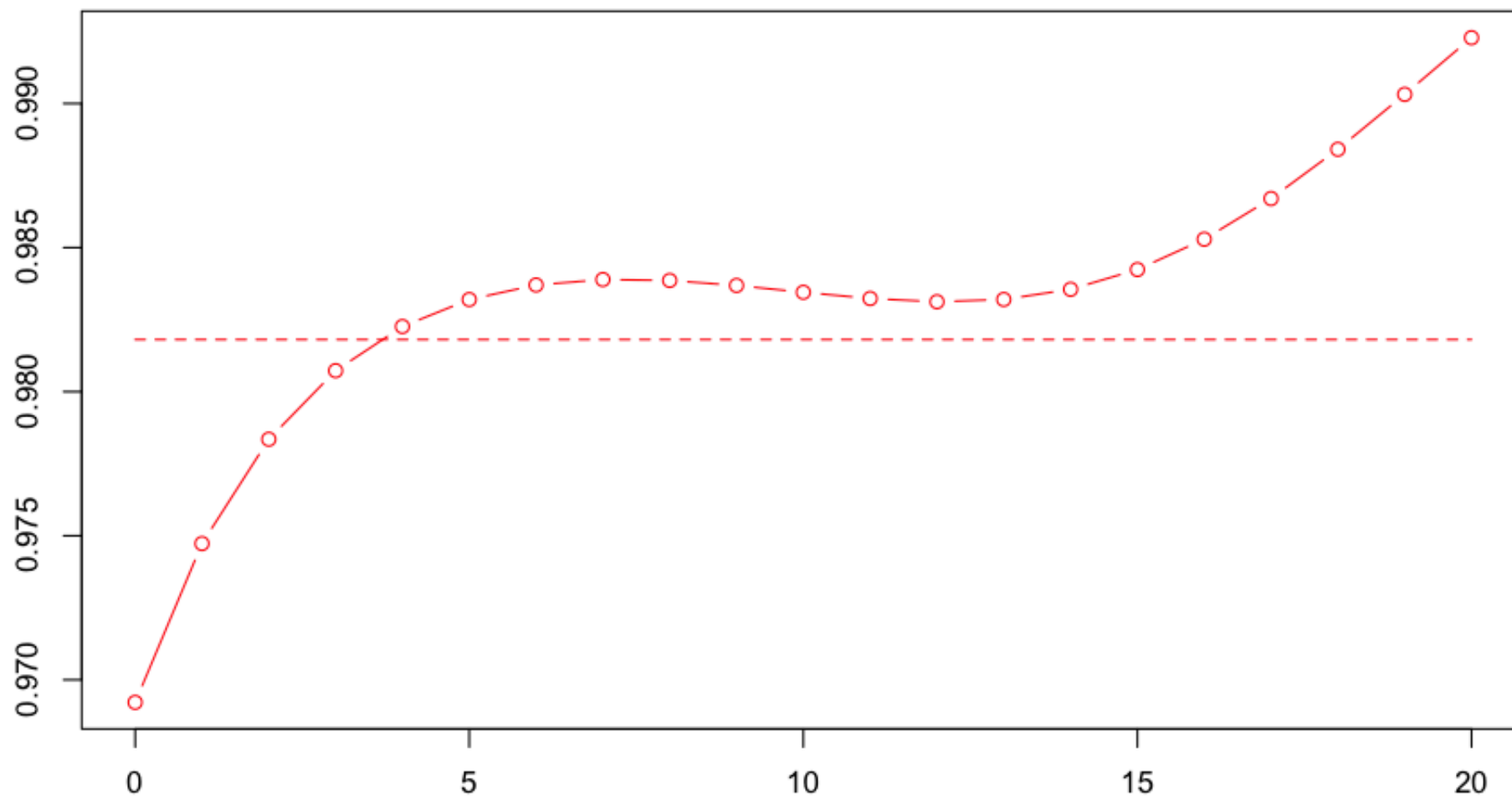
Pour le terme B , il s'agit d'une régression *standard* d'une variable de Bernoulli,

```

1 > s = 10000
2 > base_RC$normal=(base_RC$cout <=s)
3 > mean(base_RC$normal)
4 [1] 0.9818087
5 > library(splines)
6 > age=seq(0,20)
7 > regC=glm(normal~bs(agevehicule),data=base_RC,family=binomial)
8 > ypC=predict(regC,newdata=data.frame(agevehicule=age),type="response
  ")
9 > plot(age,ypC,type="b",col="red")
10 > regC2=glm(normal~1,data=base_RC,family=binomial)
11 > ypC2=predict(regC2,newdata=data.frame(agevehicule=age),type="
  response")
12 > lines(age,ypC2,type="l",col="red",lty=2)

```

Prise en compte des gros sinistres

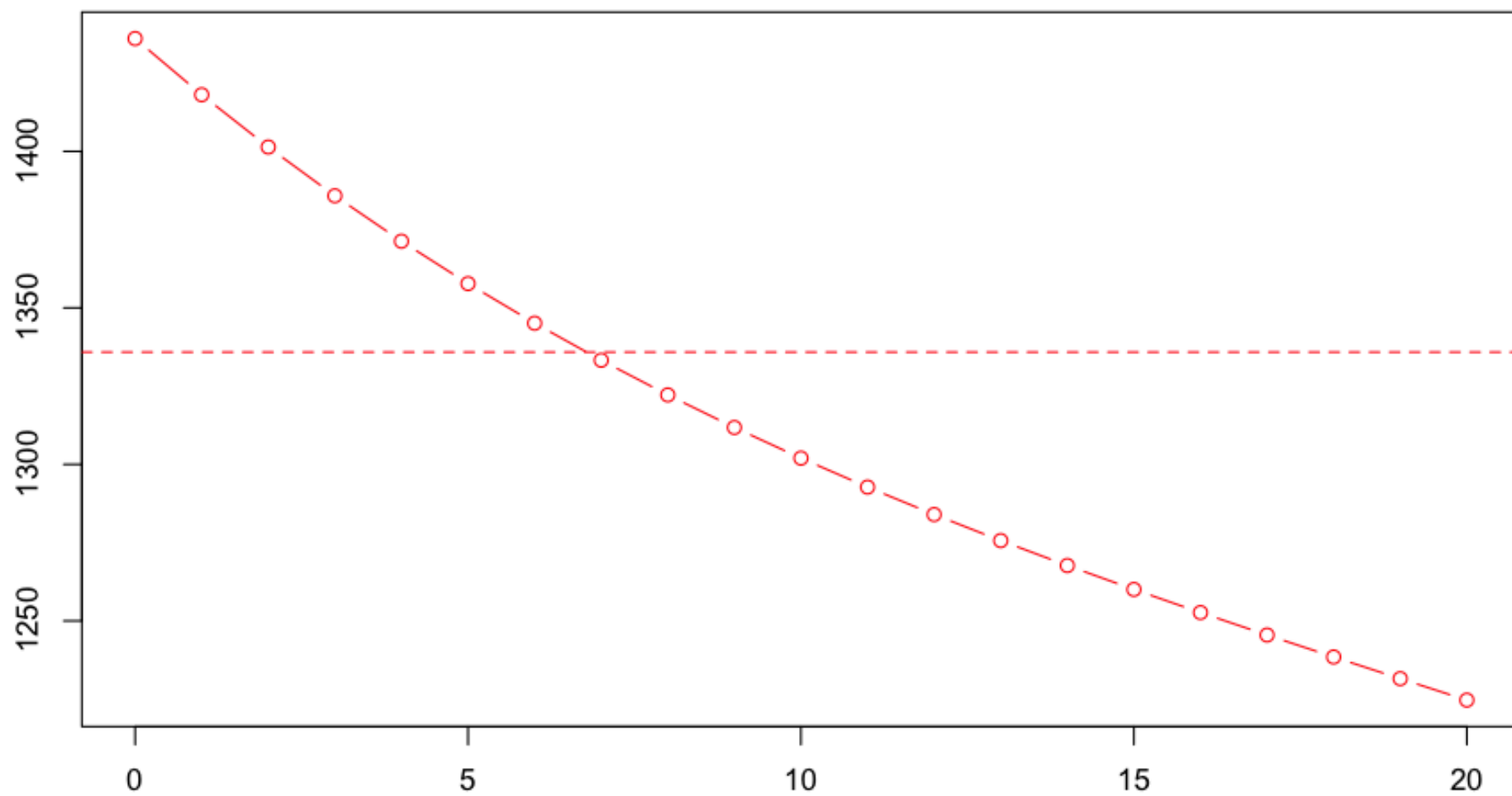


Prise en compte des gros sinistres

Pour le terme A , il s'agit d'une régression *standard* sur la base restreinte,

```
1 > indice = which(base_RC$cout <= s)
2 > mean(base_RC$cout[indice])
3 [1] 1335.878
4 > library(splines)
5 > regA = glm(cout ~ bs(agevehicule), data = base_RC,
6 + subset = indice, family = Gamma(link = "log"))
7 > ypA = predict(regA, newdata = data.frame(agevehicule = age), type = "response")
8 > plot(age, ypA, type = "b", col = "red")
9 > ypA2 = mean(base_RC$cout[indice])
10 > abline(h = ypA2, lty = 2, col = "red")
```

Prise en compte des gros sinistres

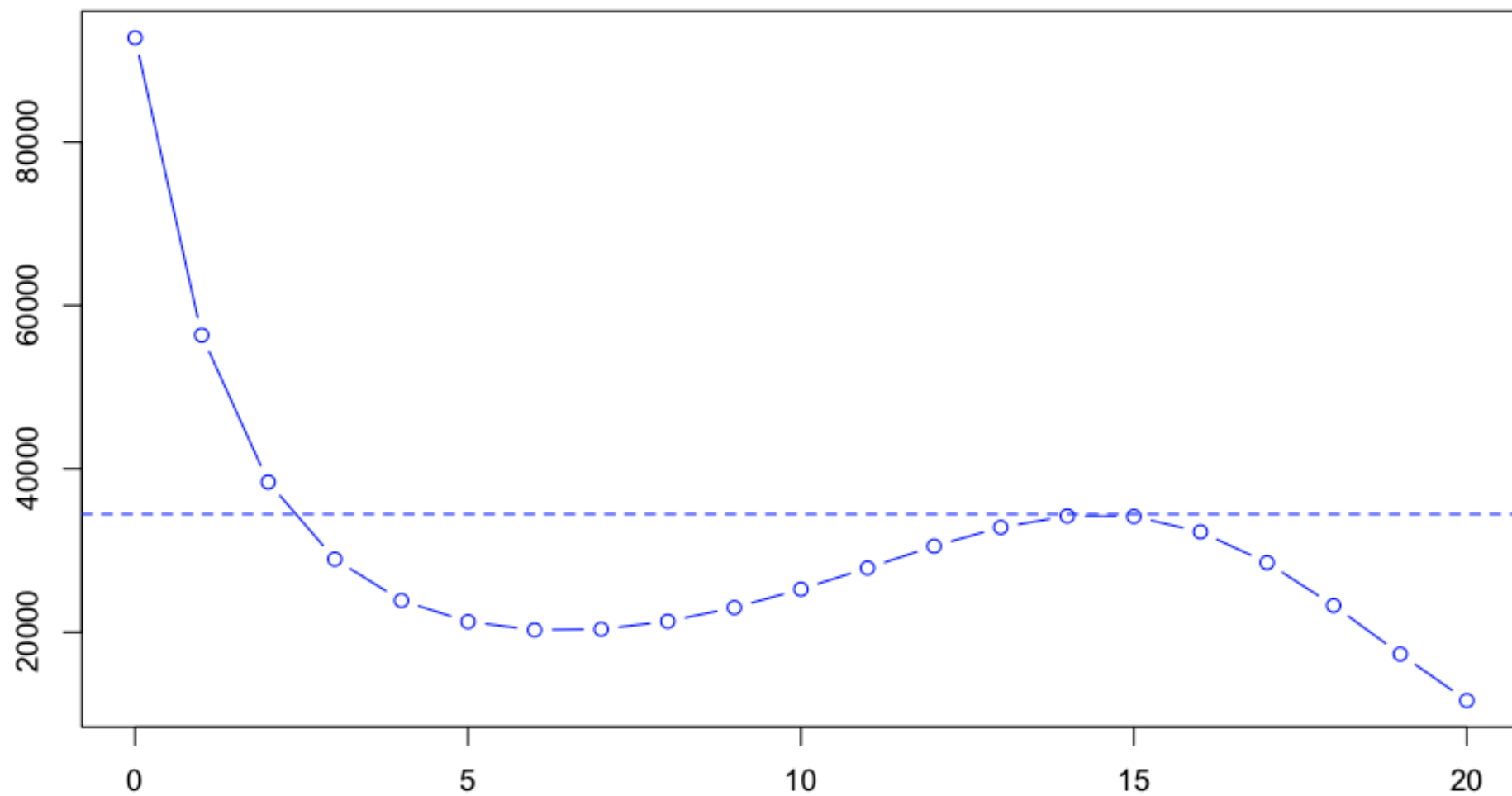


Prise en compte des gros sinistres

Pour le terme C , il s'agit d'une régression *standard* sur la base restreinte,

```
1 > indice = which(base_RC$cout>s)
2 > mean(base_RC$cout[indice])
3 [1] 34471.59
4 > regB=glm(cout~bs(agevehicule),data=base_RC,
5 + subset=indice,family=Gamma(link="log"))
6 > ypB=predict(regB,newdata=data.frame(agevehicule=age),type="
  response")
7 > plot(age,ypB,type="b",col="blue")
8 > ypB=predict(regB,newdata=data.frame(agevehicule=age),type="
  response")
9 > ypB2=mean(base_RC$cout[indice])
10 > plot(age,ypB,type="b",col="blue")
11 > abline(h=ypB2,lty=2,col="blue")
```


Prise en compte des gros sinistres

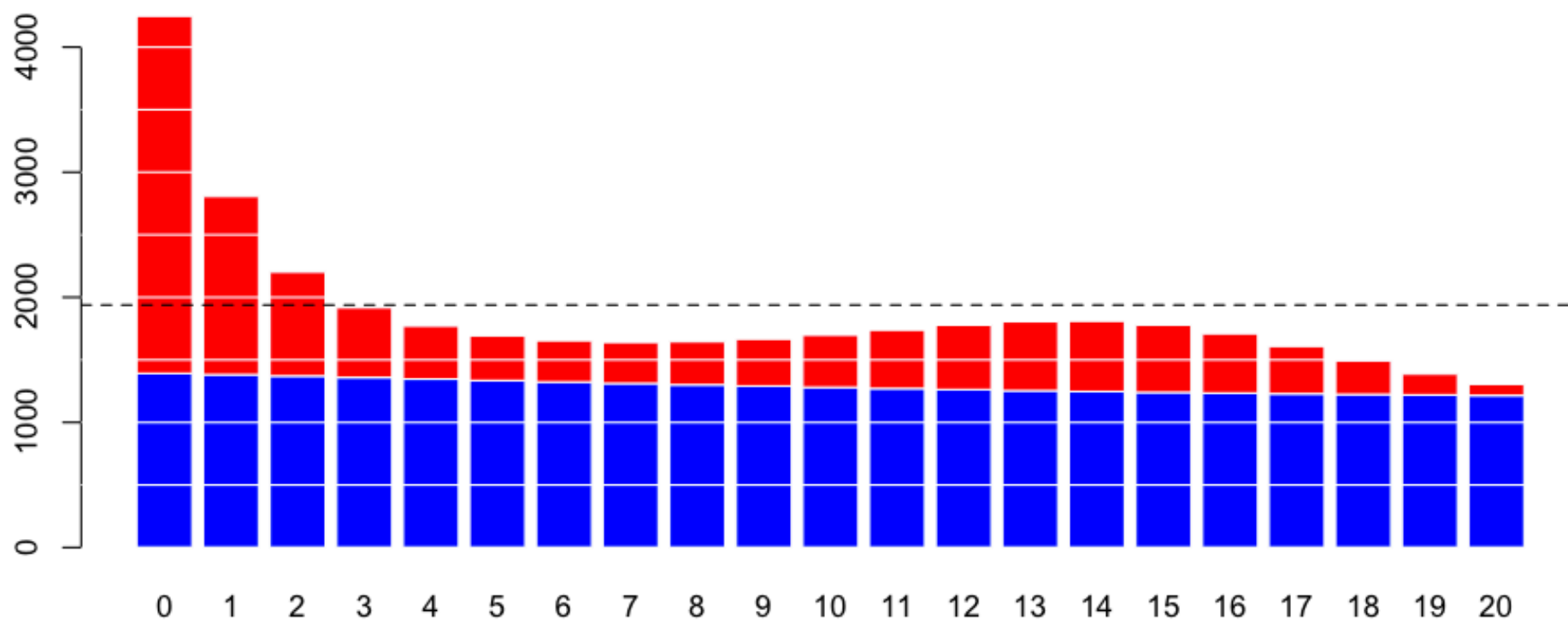


Prise en compte des gros sinistres

Reste à combiner les modèles, e.g.

$$\mathbb{E}(Y|\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s|\mathbf{X}) + \mathbb{E}(Y|Y > s, \mathbf{X}) \cdot \mathbb{P}(Y > s|\mathbf{X})$$

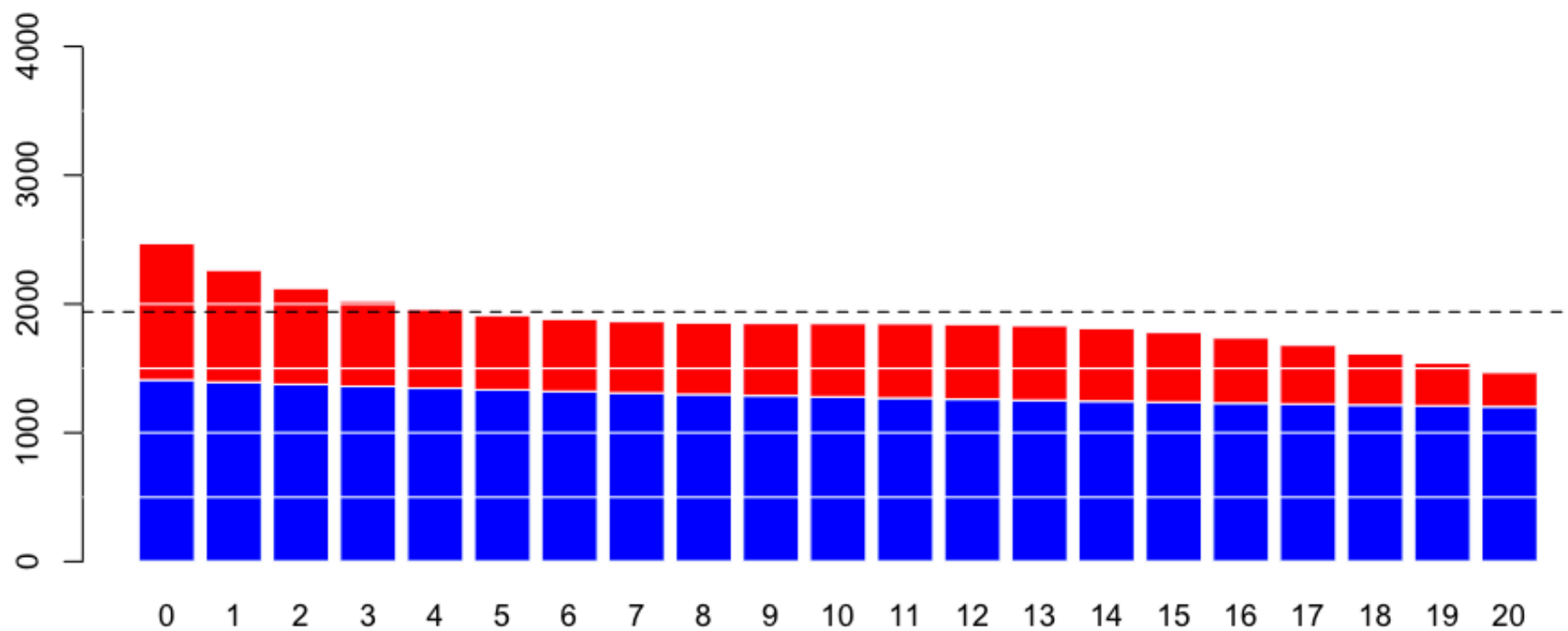
```
1 > indice = which(base_RC$cout>s)
2 > mean(base_RC$cout[indice])
3 [1] 34471.59
4 > prime = ypA*ypC + ypB*(1-ypC))
```



ou, e.g.

$$\mathbb{E}(Y|\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s|\mathbf{X}) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s|\mathbf{X})$$

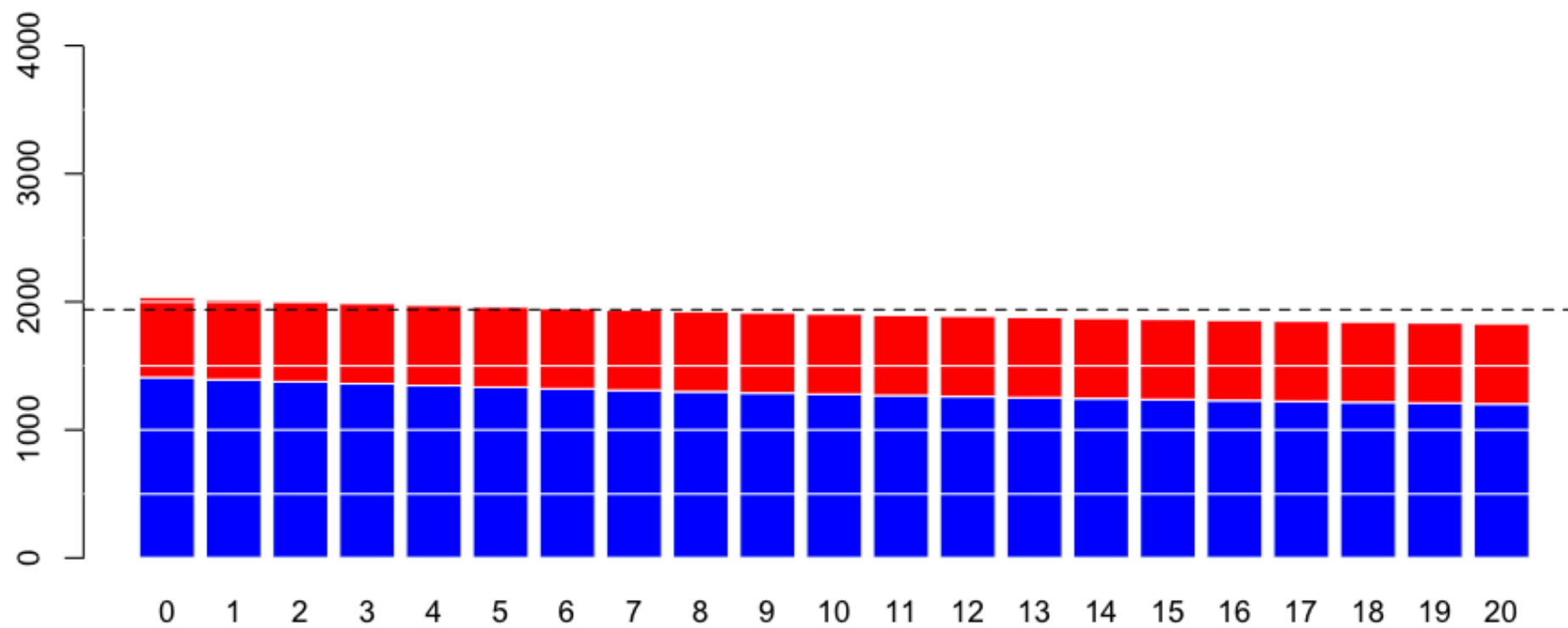
```
1 > indice = which(base_RC$cout>s)
2 > mean(base_RC$cout[indice])
3 [1] 34471.59
4 > prime = ypA*ypC + ypB2*(1-ypC))
```



voire e.g.

$$\mathbb{E}(Y|\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s)$$

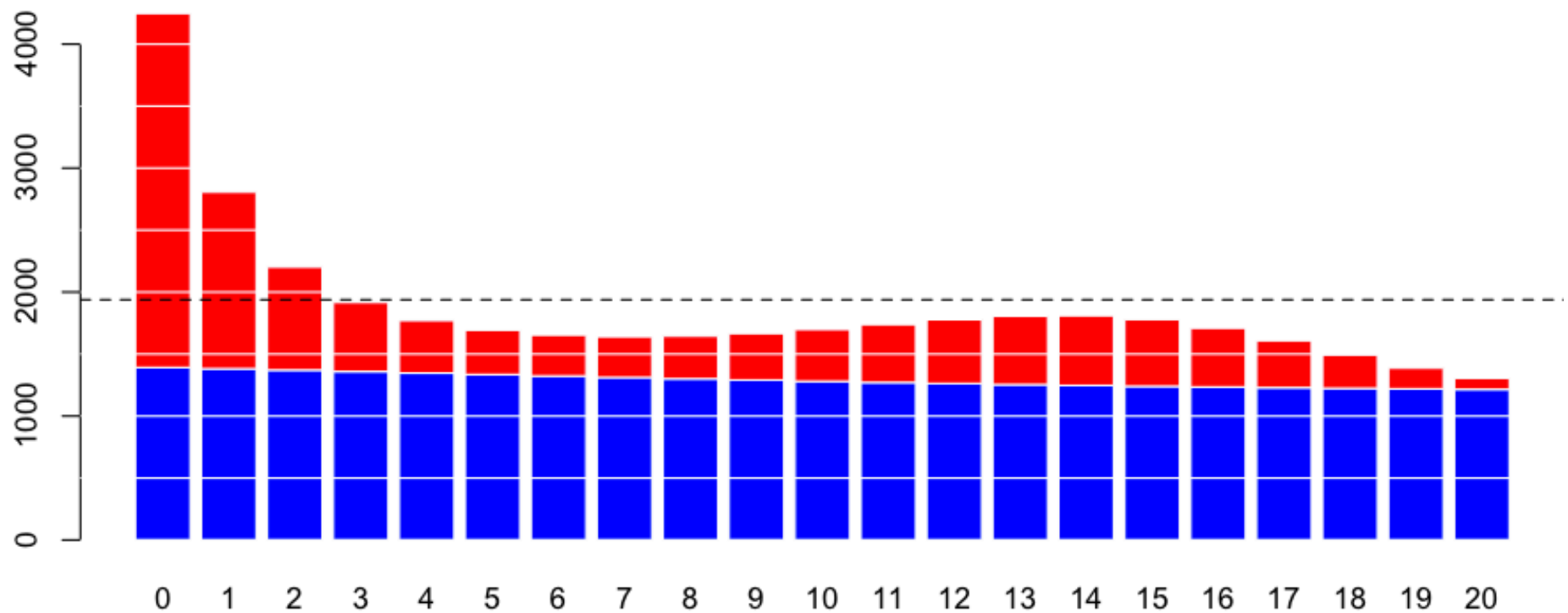
```
1 > indice = which(base_RC$cout>s)
2 > mean(base_RC$cout[indice])
3 [1] 34471.59
4 > prime = ypA*ypC + ypB2*(1-ypC))
```



Mais on peut aussi changer le seuil s dans

$$\mathbb{E}(Y|\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s)$$

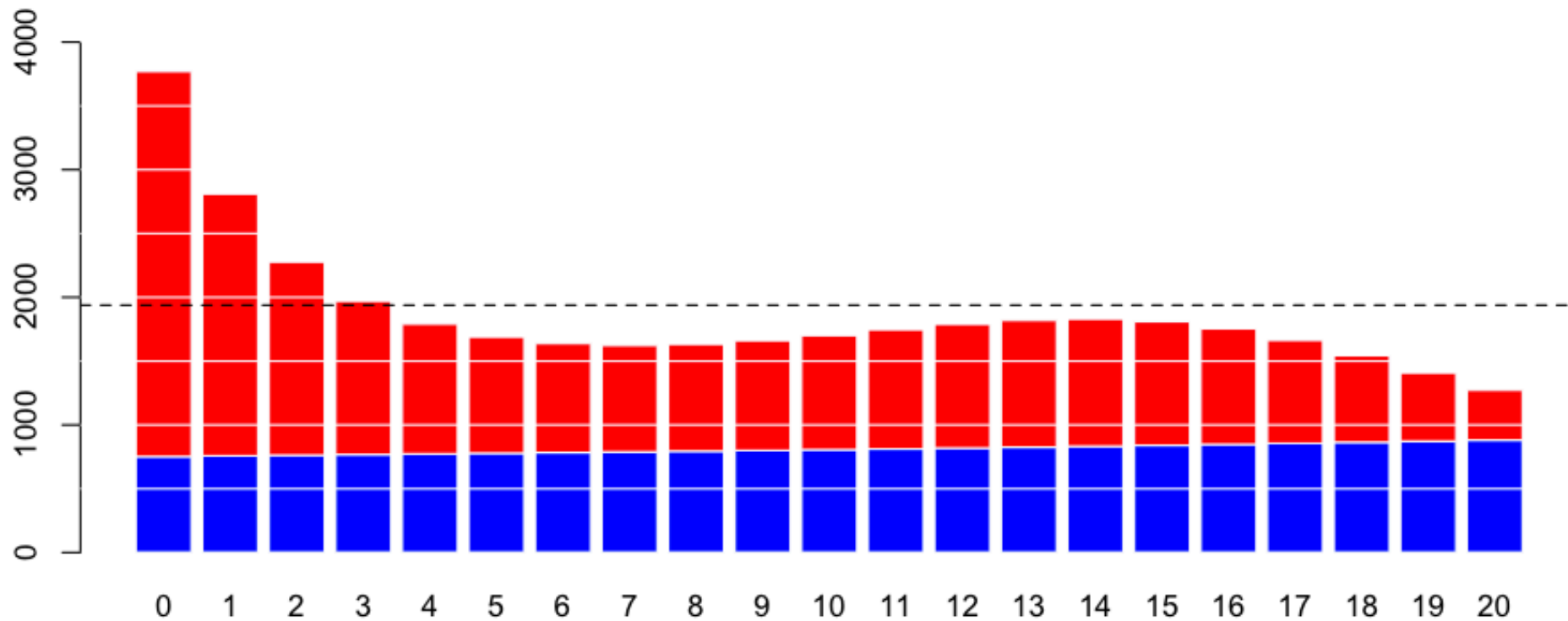
e.g. avec $s = 10,000\text{€}$,



Mais on peut aussi changer le seuil s dans

$$\mathbb{E}(Y|\mathbf{X}) = \mathbb{E}(Y|\mathbf{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s)$$

e.g. avec $s = 25,000\text{€}$,



Et s'il y avait plus que *deux* types de sinistres ?

Il est classique de supposer que la loi de Y (coût individuel de sinistres) est un mélange de plusieurs lois,

$$f(y) = \sum_{k=1}^K p_k f_k(y), \forall y \in \mathbb{R}_+$$

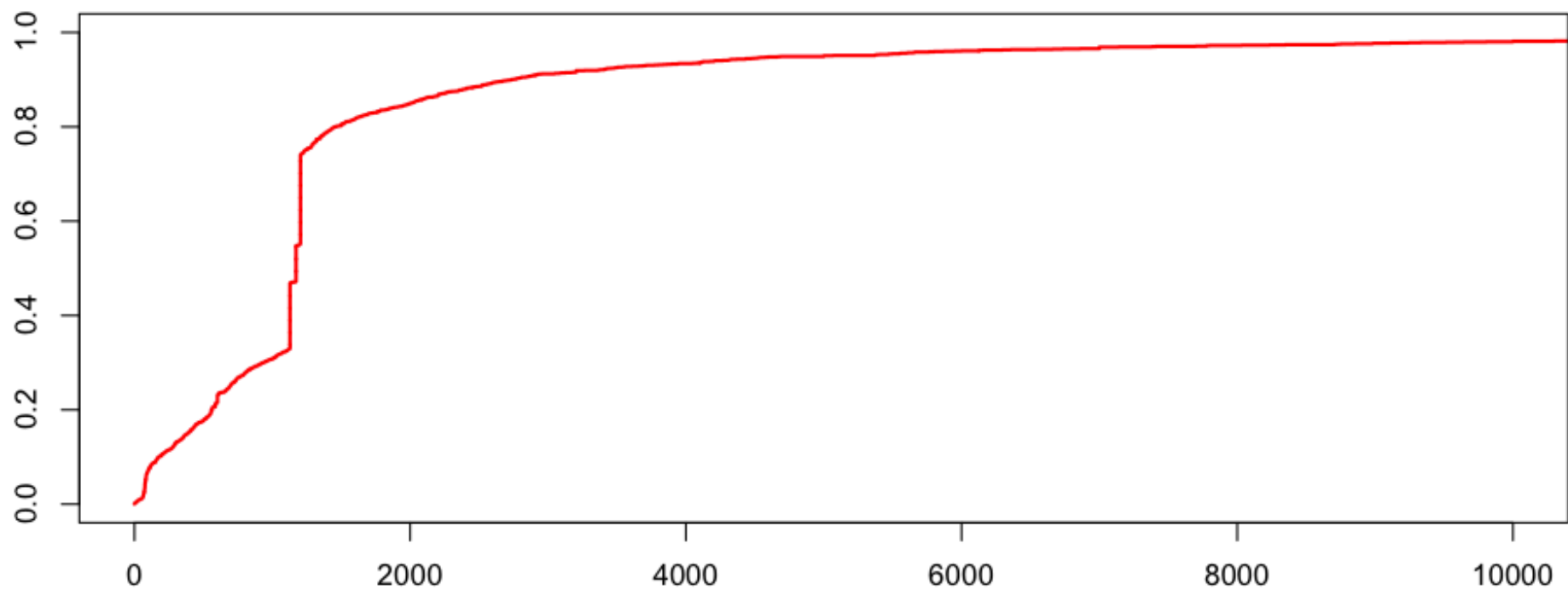
où f_k est une loi sur \mathbb{R}_+ et $\mathbf{p} = (p_k)$ un vecteur de probabilités. Ou, en terme de fonctions de répartition,

$$F(y) = \mathbb{P}(Y \leq y) = \sum_{k=1}^K p_k F_k(y), \forall y \in \mathbb{R}_+$$

où F_k est la fonction de répartition d'une variable à valeurs dans \mathbb{R}_+ .

Et s'il y avait plus que *deux* types de sinistres ?

```
1 > n=nrow(couts)
2 > plot(sort(base_RC$cout), (1:n)/(n+1), xlim=c(0,10000), type="s", lwd=2,
        col="red")
```



Et s'il y avait plus que *deux* types de sinistres ?

On peut considérer un *mélange* de trois lois,

$$f(y) = p_1 f_1(x) + p_2 \delta_\kappa(x) + p_3 f_3(x), \forall y \in \mathbb{R}_+$$

avec

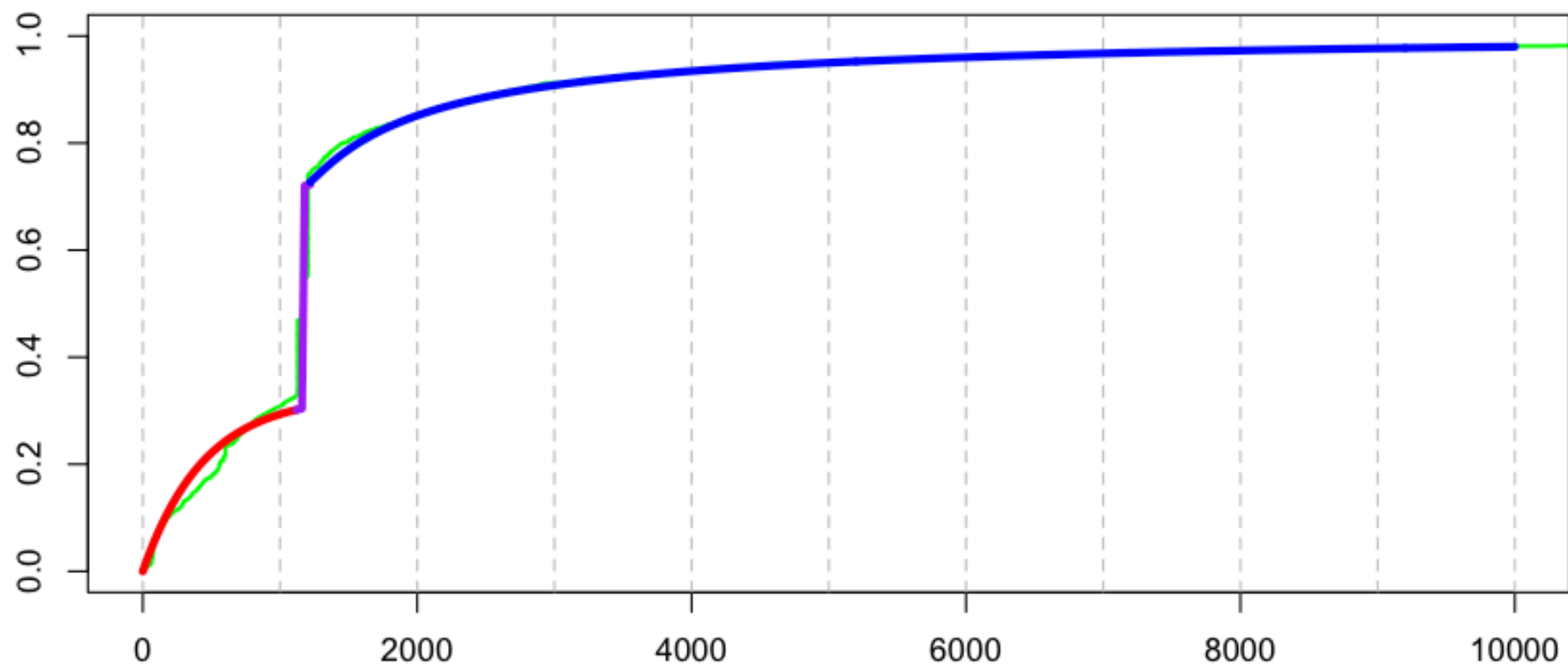
- une loi exponentielle pour f_1
- une masse de Dirac en κ (i.e. un coût fixe) pour f_2
- une loi lognormale (décallée) pour f_3

```

1 > I1=which(base_RC$cout < 1120)
2 > I2=which((base_RC$cout >= 1120) & (base_RC$cout < 1220))
3 > I3=which(base_RC$cout >= 1220)
4 > (p1=length(I1)/nrow(couts))
5 [1] 0.3284823
6 > (p2=length(I2)/nrow(couts))

```

```
7 [1] 0.4152807
8 > (p3=length(I3)/nrow(couts))
9 [1] 0.256237
10 > X=base_RC$cout
11 > (kappa=mean(X[I2]))
12 [1] 1171.998
13 > X0=X[I3]-kappa
14 > u=seq(0,10000,by=20)
15 > F1=pexp(u,1/mean(X[I1]))
16 > F2= (u>kappa)
17 > F3=plnorm(u-kappa,mean(log(X0)),sd(log(X0))) * (u>kappa)
18 > F=F1*p1+F2*p2+F3*p3
19 > lines(u,F,col="blue")
```



Prise en compte des coûts fixes en tarification

Comme pour les gros sinistres, on peut utiliser ce découpage pour calculer $\mathbb{E}(Y)$, ou $\mathbb{E}(Y|\mathbf{X})$. Ici,

$$\begin{aligned}\mathbb{E}(Y|\mathbf{X}) &= \underbrace{\mathbb{E}(Y|\mathbf{X}, Y \leq s_1)}_A \cdot \underbrace{\mathbb{P}(Y \leq s_1|\mathbf{X})}_{D, \pi_1(\mathbf{X})} \\ &\quad + \underbrace{\mathbb{E}(Y|Y \in (s_1, s_2], \mathbf{X})}_B \cdot \underbrace{\mathbb{P}(Y \in (s_1, s_2]|\mathbf{X})}_{D, \pi_2(\mathbf{X})} \\ &\quad + \underbrace{\mathbb{E}(Y|Y > s_2, \mathbf{X})}_C \cdot \underbrace{\mathbb{P}(Y > s_2|\mathbf{X})}_{D, \pi_3(\mathbf{X})}\end{aligned}$$

Les paramètres du mélange, $(\pi_1(\mathbf{X}), \pi_2(\mathbf{X}), \pi_3(\mathbf{X}))$ peuvent être associés à une loi multinomiale de dimension 3.

Loi multinomiale

Rappelons que pour la régression logistique, si $(\pi, 1 - \pi) = (\pi_1, \pi_2)$

$$\log \frac{\pi}{1 - \pi} = \log \frac{\pi_1}{\pi_2} = \mathbf{X}'\boldsymbol{\beta},$$

ou encore

$$\pi_1 = \frac{\exp(\mathbf{X}'\boldsymbol{\beta})}{1 + \exp(\mathbf{X}'\boldsymbol{\beta})} \text{ et } \pi_2 = \frac{1}{1 + \exp(\mathbf{X}'\boldsymbol{\beta})}$$

On peut définir une **régression logistique multinomiale**, de paramètre $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3)$ en posant

$$\log \frac{\pi_1}{\pi_3} = \mathbf{X}'\boldsymbol{\beta}_1 \text{ et } \log \frac{\pi_2}{\pi_3} = \mathbf{X}'\boldsymbol{\beta}_2$$

Loi multinomiale

ou encore

$$\pi_1 = \frac{\exp(\mathbf{X}'\beta_1)}{1 + \exp(\mathbf{X}'\beta_1) + \exp(\mathbf{X}'\beta_2)}, \pi_2 = \frac{\exp(\mathbf{X}'\beta_2)}{1 + \exp(\mathbf{X}'\beta_1) + \exp(\mathbf{X}'\beta_2)}$$

$$\text{et } \pi_3 = \frac{1}{1 + \exp(\mathbf{X}'\beta_1) + \exp(\mathbf{X}'\beta_2)}.$$

Remarque l'estimation se fait - là encore - en calculant numériquement le maximum de vraisemblance, en notant que

$$\mathcal{L}(\boldsymbol{\pi}, \mathbf{y}) \propto \prod_{i=1}^n \prod_{j=1}^3 \pi_{i,j}^{Y_{i,j}}$$

où Y_i est ici disjunctée en $(Y_{i,1}, Y_{i,2}, Y_{i,3})$ contenant les variables indicatrices de chacune des modalités. La log-vraisemblance est alors proportionnelle à

$$\log \mathcal{L}(\boldsymbol{\beta}, \mathbf{y}) \propto \sum_{i=1}^n \sum_{j=1}^2 (Y_{i,j} \mathbf{X}'_i \boldsymbol{\beta}_j) - n_i \log [1 + 1 + \exp(\mathbf{X}'\beta_1) + \exp(\mathbf{X}'\beta_2)]$$

Loi multinomiale (et GLM)

qui se résout avec un algorithme de type Newton-Raphson, en notant que

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\beta}, \mathbf{y})}{\partial \beta_{k,j}} = \sum_{i=1}^n Y_{i,j} X_{i,k} - n_i \pi_{i,j} X_{i,k}$$

i.e.

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\beta}, \mathbf{y})}{\partial \beta_{k,j}} = \sum_{i=1}^n Y_{i,j} X_{i,k} - n_i \frac{\exp(\mathbf{X}'_i \boldsymbol{\beta}_j)}{1 + \exp(\mathbf{X}'_i \boldsymbol{\beta}_1) + \exp(\mathbf{X}'_i \boldsymbol{\beta}_2)} X_{i,k}$$

Loi multinomiale

Sous R, la fonction `multinom` de `library(nnet)` permet de faire cette estimation. On commence par définir les trois tranches de coûts,

```
1 > seuils=c(0,1120,1220,1e+12)
2 > base_RC$tranches=cut(base_RC$cout,breaks=seuils,
3 + labels=c("small","fixed","large"))
4 > head(couts,5)
5   nocontrat    no  garantie    cout  exposit  zone  puissance  agevehicule
6 1      1870 17219      1RC 1692.29    0.11    C           5           0
7 2      1963 16336      1RC  422.05    0.10    E           9           0
8 3      4263 17089      1RC  549.21    0.65    C          10           7
9 4      5181 17801      1RC  191.15    0.57    D           5           2
10 5      6375 17485      1RC 2031.77    0.47    B           7           4
11   ageconducteur  bonus  marque  carbur  densite  region  tranches
12 1           52    50    12      E      73     13    large
13 2           78    50    12      E      72     13    small
14 3           27    76    12      D      52     5     small
15 4           26   100    12      D      83     0     small
```

16	5	46	50	6	E	11	13	large
----	---	----	----	---	---	----	----	-------

Loi multinomiale

On peut ensuite faire une régression multinomiale afin d'expliquer π_i en fonction de covariables \mathbf{X}_i .

```
1 > reg=multinom(tranches~ageconducteur+agevehicule+zone+carburant ,data
    =base_RC)
2 # weights:  30 (18 variable)
3 initial    value 2113.730043
4 iter    10 value 2063.326526
5 iter    20 value 2059.206691
6 final     value 2059.134802
7 converged
```

```

1 > summary(reg)
2 Call:
3 multinom(formula = tranches ~ ageconducteur + agevehicule + zone +
4   carburant, data = couts)
5
6 Coefficients:
7   (Intercept) ageconducteur agevehicule      zoneB      zoneC
8 fixed  -0.2779176    0.012071029   0.01768260  0.05567183 -0.2126045
9 large  -0.7029836    0.008581459  -0.01426202  0.07608382  0.1007513
10      zoneD      zoneE      zoneF      carburantE
11 fixed  -0.1548064 -0.2000597 -0.8441011 -0.009224715
12 large   0.3434686  0.1803350 -0.1969320  0.039414682
13
14 Std. Errors:
15   (Intercept) ageconducteur agevehicule      zoneB      zoneC
16      zoneD
17 fixed   0.2371936    0.003738456    0.01013892  0.2259144  0.1776762
18      0.1838344

```

```

17 large    0.2753840    0.004203217    0.01189342    0.2746457    0.2122819
    0.2151504
18          zoneE      zoneF carburantE
19 fixed 0.1830139 0.3377169 0.1106009
20 large 0.2160268 0.3624900 0.1243560

```

Loi multinomiale

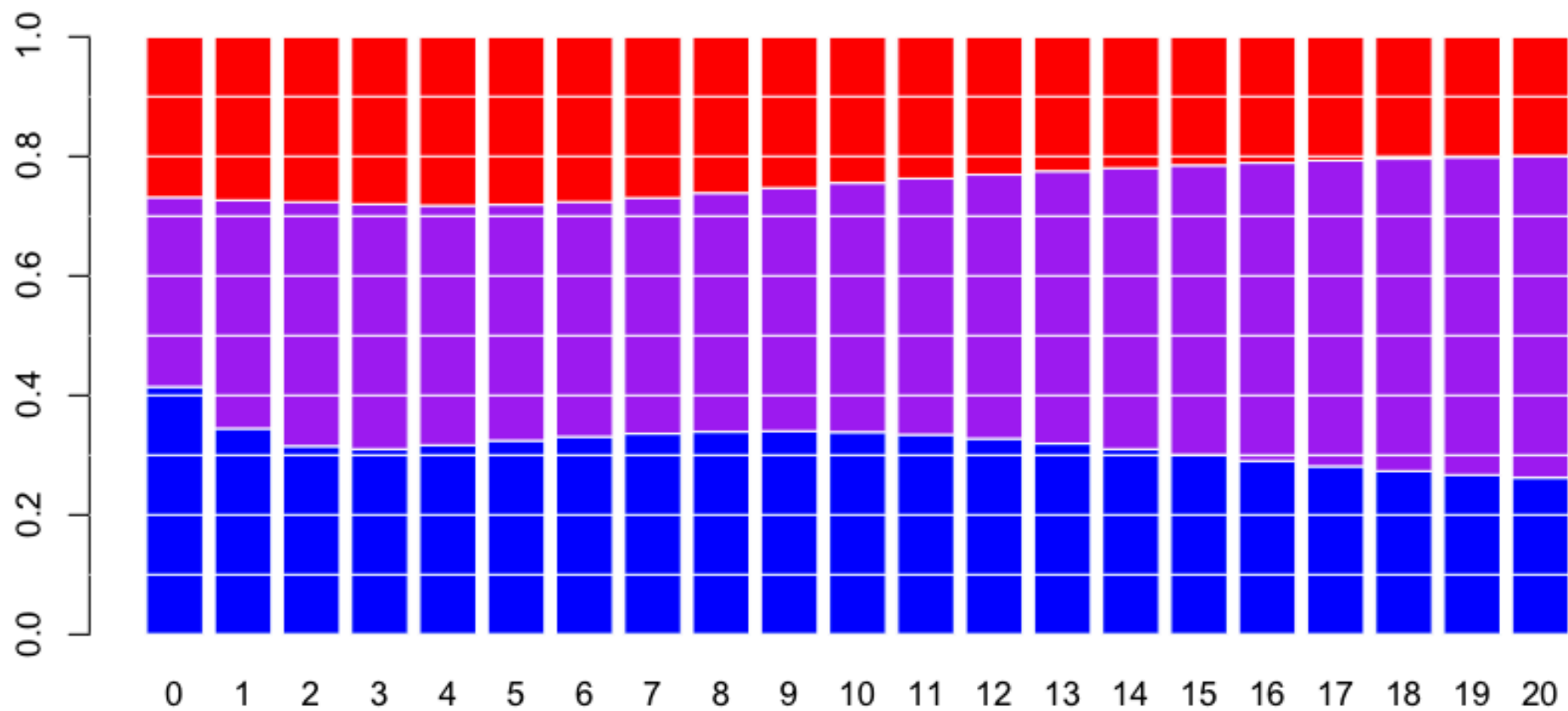
On peut régresser suivant l'ancienneté du véhicule, avec ou sans lissage,

```
1 > library(splines)
2 > reg=multinom(tranches~agevehicule ,data=base_RC)
3 # weights:  9 (4 variable)
4 initial  value 2113.730043
5 final    value 2072.462863
6 converged
7 > reg=multinom(tranches~bs(agevehicule),data=base_RC)
8 # weights:  15 (8 variable)
9 initial  value 2113.730043
10 iter   10 value 2070.496939
11 iter   20 value 2069.787720
12 iter   30 value 2069.659958
13 final   value 2069.479535
14 converged
```


Loi multinomiale

On peut alors prédire la probabilité, sachant qu'un accident survient, qu'il soit de type 1, 2 ou 3

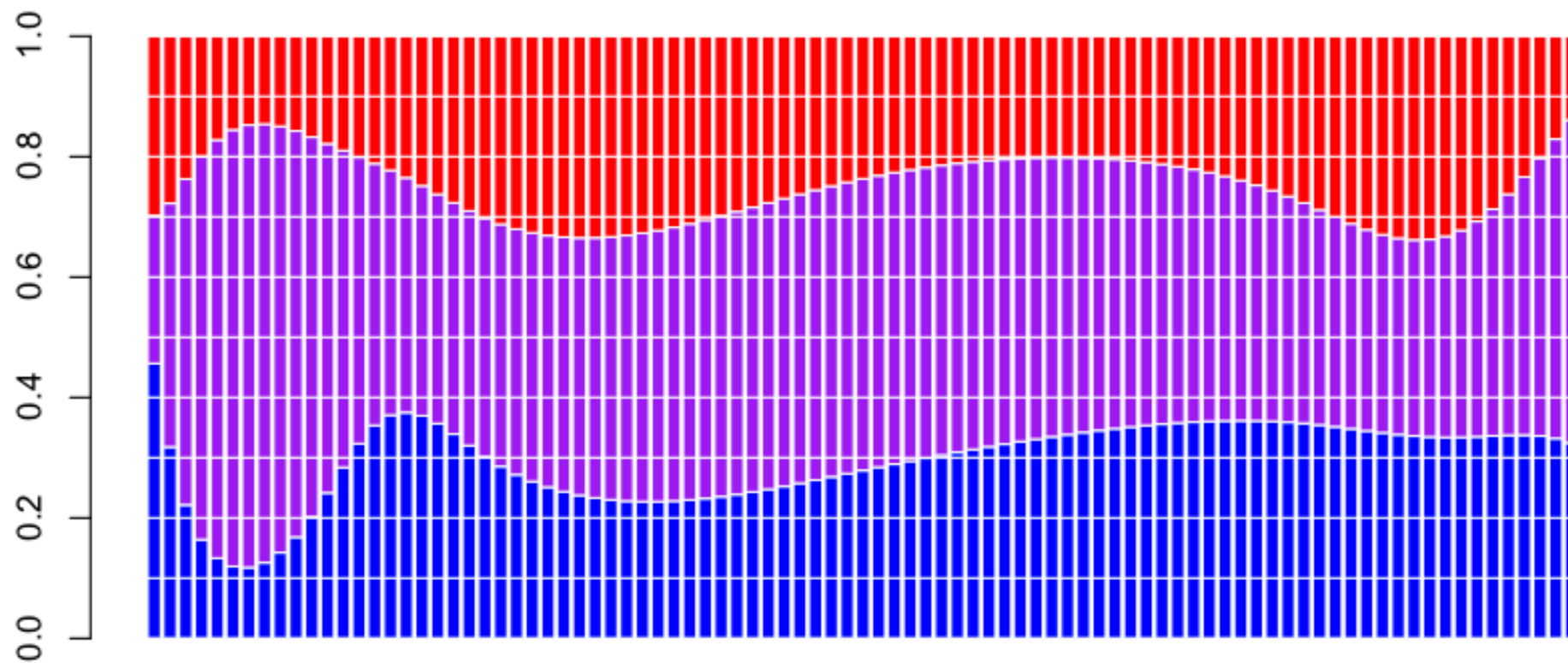
```
1 > predict(reg,newdata=data.frame(agevehicule=5),type="probs")
2      small      fixed      large
3 0.3388947 0.3869228 0.2741825
```



Loi multinomiale

ou en fonction de la densité de population

```
1 > reg=multinom(tranches~bs(densite),data=base_RC)
2 # weights:  15 (8 variable)
3 initial    value 2113.730043
4 iter    10 value 2068.469825
5 final    value 2068.466349
6 converged
7 > predict(reg,newdata=data.frame(densite=90),type="probs")
8      small      fixed      large
9 0.3484422 0.3473315 0.3042263
```



Loi multinomiale

Il faut ensuite ajuster des lois pour les trois régions A, B ou C

```

1 > reg=multinom(tranches~bs(densite),data=base_RC)
2 # weights:  15 (8 variable)
3 initial   value 2113.730043
4 iter    10 value 2068.469825
5 final    value 2068.466349
6 converged
7 > predict(reg,newdata=data.frame(densite=90),type="probs")
8      small      fixed      large
9 0.3484422 0.3473315 0.3042263

```

Pour A, on peut tenter une loi exponentielle (qui est une loi Gamma avec $\phi = 1$).

```

1 > regA=glm(cout~agevehicule+densite+carburant,data=sousbaseA,
2 + family=Gamma(link="log"))
3 > summary(regA, dispersion=1)
4
5 Coefficients:

```

```

6      Estimate Std. Error z value Pr(>|z|)
7 (Intercept)  6.0600491   0.1005279  60.282   <2e-16 ***
8 agevehicule  0.0003965   0.0070390   0.056   0.955
9 densite      0.0014085   0.0013541   1.040   0.298
10 carburantE   -0.0751446   0.0806202  -0.932   0.351
11 ---
12 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
13
14 (Dispersion parameter for Gamma family taken to be 1)

```

Pour **B**, on va garder l'idée d'une masse de Dirac en

```

1 > mean(sousbaseB$cout)
2 [1] 1171.998

```

(qui semble correspondre à un coût fixe.

Enfin, pour **C**, on peut tenter une loi Gamma ou lognormale décallée,

```

1 > k=mean(sousbaseB$cout)

```

```

2 > regC=glm((cout-k)~agevehicule+densite+carburant ,data=sousbaseC ,
3 + family=Gamma(link="log"))
4 > summary(regC)
5
6 Coefficients:
7             Estimate Std. Error t value Pr(>|t|)
8 (Intercept)  9.119879   0.378836  24.073   <2e-16 ***
9 agevehicule -0.013393   0.028620  -0.468   0.6400
10 densite     -0.010814   0.004831  -2.239   0.0256 *
11 carburantE   -0.530964   0.287450  -1.847   0.0653 .
12 ---
13 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
14
15 (Dispersion parameter for Gamma family taken to be 10.00845)

```

