

T. D. n° 3

Régression linéaire simple

Résumé

Ce T.D. reprend rapidement des éléments du cours n°5 et n°6 et propose une mise en pratique interactive de la régression linéaire simple.

1 Quel film recommanderiez-vous à un esiearque ?

La démarche basée sur les dénombrements est extrêmement séduisante par sa simplicité. Un simple comptage permet de produire les probabilités conditionnelles et d'en déduire les règles d'associations. Toutefois, elle n'est pas viable en situation réelle. Pas assez précise et statique, nous préférons la technique de la régression qui est généralement utilisée pour construire des modèles prédictifs. La régression est un ensemble de méthodes statistiques très utilisées pour analyser la relation d'une variable par rapport à une ou plusieurs autres variables. En apprentissage statistique, nous distinguons deux types de problème :

- a) les problèmes de régression,
- b) les problèmes de classification.

Ainsi, nous considérons que les problèmes de prédiction d'une variable quantitative sont des problèmes de régression tandis que les problèmes de prédiction d'une variable qualitative sont des problèmes de classification. Certaines méthodes, comme la régression logistique, sont à la fois des méthodes de régression au sens où il s'agit de prédire la probabilité d'appartenir à chacune des classes et des méthodes de classification. Les applications sont nombreuses, certains touchent directement à la vie quotidienne :

- déterminer la viabilité d'un client sollicitant un crédit à partir de ses caractéristiques (exemple : âge, type d'emploi, niveau de revenu, autres crédits en cours, etc.).
- quantifier le risque de survenue d'un sinistre pour une personne sollicitant un contrat d'assurance.
- discerner les facteurs de risque de survenue d'une maladie cardio-vasculaire chez des patients (exemple : âge, sexe, tabac, alcool, regarder les matchs de l'équipe de France de football, etc.).
- pour une enseigne de grande distribution, cibler les clients qui peuvent être intéressés par tel ou tel type de produit.

Vous souhaitez prédire les notes de films d'un utilisateur quelconque, à partir de la base de données d'une plateforme publique (type *Allociné*), produisant une note moyenne, pour chaque film, à partir de toutes les notes données par l'ensemble des visiteurs de la plateforme publique.

Le package `ggplot2` est le package qui va nous servir à faire les graphiques pour ce T.D. Les graphiques issus de ce package ont un meilleur rendu : meilleure gestion de l'espace, des couleurs, légende insérée automatiquement. De plus, il est possible d'ajouter plus d'informations sur le graphique.

Commencez par charger votre jeu de données `MyData` : les données pour cet exercice, qui proviennent du site IMDb : <http://www.imdb.com/>

sont les notes pour 422 films enregistrés. Le site IMDb conserve les notes moyennes de films, construites par tous les utilisateurs dans un tableau qui contient des informations sur le titre du film, son réalisateur, sa durée, son année de sortie, le genre, la note moyenne (celle du site IMDb), et quelques autres variables moins intéressantes. Le fichier `Mydata.csv` contient ces informations ainsi que la note donnée par l'utilisateur que nous voulons étudier. Nous souhaitons en effet, prédire les notes que donnera cet utilisateur à des films, en se basant sur les notes que les autres utilisateurs ont déjà donné. La variable `IMDb` correspond à la note moyenne du film. La variable `mine` correspond à la note du film de notre utilisateur.

```
d <- read.csv("C:/Users/claey/Documents/cour/My TD/TD 3/MyData.csv",
+sep = ",")
```

La variable Y que vous voulez prédire est la note qui va être donnée par notre utilisateur (`mine`) pour chaque film. Le site IMDb vous permet de noter des films en utilisant le système d'étoiles, de une à dix étoiles. Les demi-étoiles et autres fractions ne sont pas autorisées. La note n'est donc évidemment pas une variable continue mais nous la traiterons comme telle pour ce T.D. Les figures ci-dessous illustrent la densité des notes de notre utilisateur (Figure 1) et de la densité des notes moyennes des utilisateurs d'IMDb (Figure 2) pour les 422 films enregistrés.

```
#Figure 1
>plot(density(d$mine))
#Figure 2
>plot(density(d$imdb))
```

FIGURE 1 – Fonction de distribution des notes de notre utilisateur

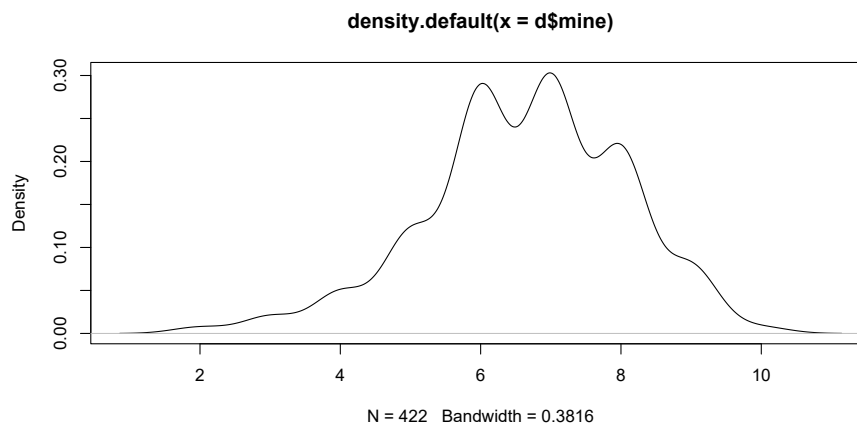
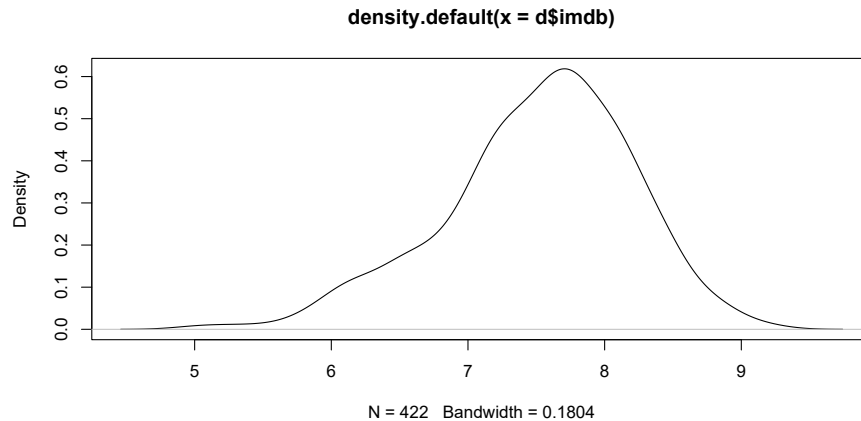


FIGURE 2 – Fonction de distribution des notes des utilisateurs d’IMDb



À vous !

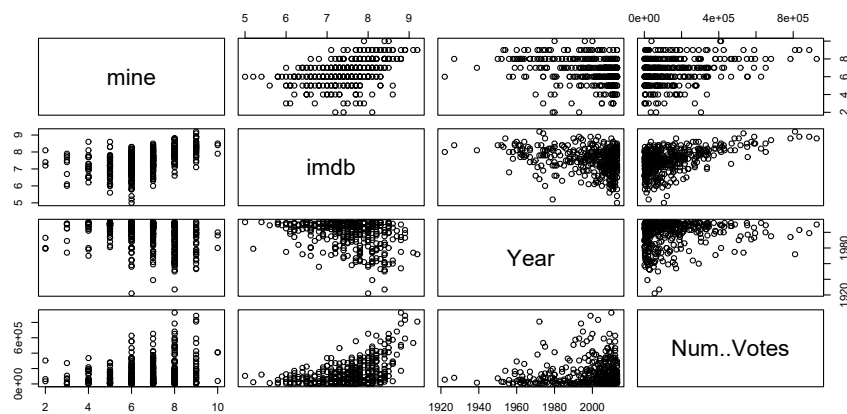
- Où se situe la moyenne sur chacune des deux courbes ?
- Estimez visuellement l'écart-type.
- Donnez la variance des notes de notre utilisateur et de celles de IMDb.
- Que pouvez-vous dire sur les notes de notre utilisateur (Figure 1) ?
- Que pouvez-vous dire sur les notes des utilisateurs d’IMDb (Figure 2) ?
- Essayez de justifier la différence entre ces deux courbes.

1.1 Corrélation linéaire

La fonction `pairs()` produit une matrice de nuages de points et attend en entrée un objet de type matrice ou de type `dataframe`.

En probabilités et en statistique, étudier la corrélation linéaire entre deux ou plusieurs variables aléatoires ou variables statistiques (de type quantitatif), c’est étudier l’intensité de la liaison linéaire qui peut exister entre ces variables. Nous pouvons par exemple avoir une relation affine entre une variable quantitative X et une variable quantitative Y (de type $Y = a + b * X$), qui est une régression linéaire simple entre ces deux variables. La mesure de la corrélation linéaire entre ces deux variables se fait alors par le calcul du coefficient de corrélation linéaire, appelé aussi coefficient de bravais-Pearson, noté généralement $r(X, Y)$.

```
> pairs(d[,c(9,10,12,14)])
```

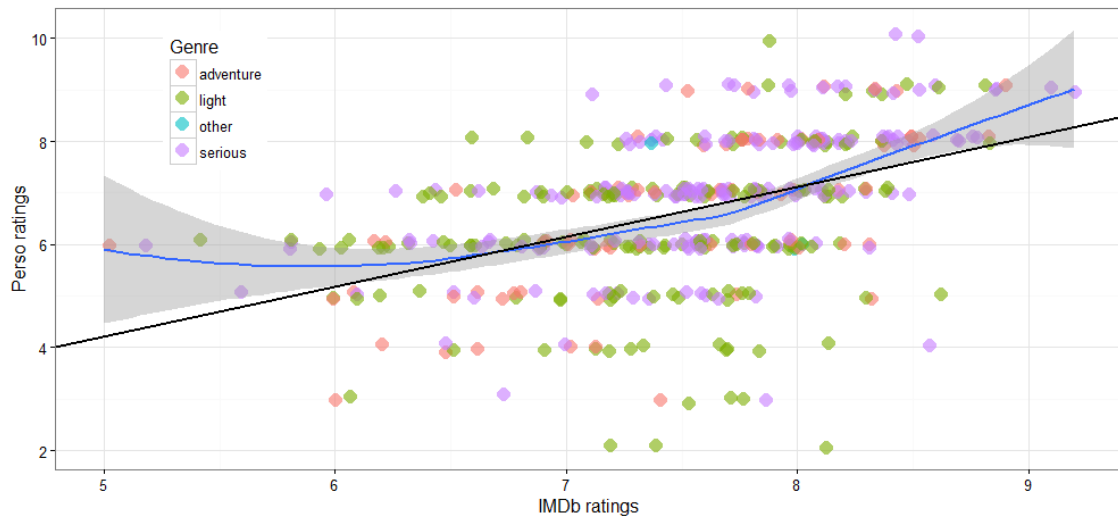
FIGURE 3 – Sortie graphique de la fonction `pairs()`

Nous décidons d’afficher la relation entre les notes des utilisateurs d’IMDb et celles de notre utilisateur.

#Figure 4

```
> p2 <- ggplot(d,aes(imdb, mine))+
  geom_point(position=position_jitter(width=0.1,height=.25),
    shape=16,
    size=4,alpha=0.6,
    aes(colour = new.genre, ))+
  stat_smooth(se = TRUE)+
  scale_x_continuous('IMDb ratings')+
  scale_y_continuous('Perso ratings')+
  theme_bw()+
  scale_colour_discrete(name="Genre")+
  scale_size_continuous(guide=FALSE)+
  theme(legend.position=c(0.15, 0.80))+
  geom_abline(size=1, aes(intercept=-0.6387, slope=0.9686))
> p2
```

FIGURE 4 – Représentation graphique et modélisation



La ligne noire est la droite de régression des moindres carrés, la ligne bleue représente un lissage de *loess* non-paramétrique qui suggère une certaine non-linéarité dans la relation que vous allez explorer plus tard.

À vous !

- Les variables `mine` et `imdb` sont-elles corrélées linéairement ?
- Existe-t-il d'autres corrélations linéaires entre les autres variables quantitatives ?
- Cherchez l'utilité de la fonction `aes()`.
- Sur quel genre de films pouvez-vous dire que l'utilisateur est difficile ?

1.2 Régression linéaire simple

Vous allez commencer avec un modèle de régression linéaire simple qui sera votre point de départ pour l'analyse statistique que vous allez mener. Ces résultats vous serviront de référence. Voici les estimations que le modèle `m1` prévoit sur les notes de notre utilisateur à partir des notes moyennes des utilisateurs d'IMDb :

```
> summary(m1<-lm(mine~imdb, data=d))
```

Call:

```
lm(formula = mine ~ imdb, data = d)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.2066	-0.7224	0.1808	0.7934	2.9871

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.6387	0.6669	-0.958	0.339
imdb	0.9686	0.0884	10.957	<2e-16 ***

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

Residual standard error: 1.254 on 420 degrees of freedom

Multiple R-squared: 0.2223, Adjusted R-squared: 0.2205

F-statistic: 120.1 on 1 and 420 DF, p-value: < 2.2e-16

À vous !

- Définissez les caractéristiques de la fonction `lm()`.
- Sur quelle base avez-vous entraîné votre modèle ?
- Quelle(s) variable(s) utilisez-vous pour construire votre modèle ?
- Quelle a été l'erreur maximale du modèle ?
- Commentez la qualité du modèle¹.

1.3 Erreur quadratique moyenne

L'erreur quadratique moyenne notée MSE (Mean Square Error) est très utile pour comparer plusieurs estimateurs, notamment lorsque l'un d'eux est biaisé. Si les deux estimateurs à comparer sont sans biais, l'estimateur le plus efficace est celui qui a la variance la plus petite. Nous pouvons effectivement exprimer l'erreur quadratique moyenne en fonction du biais de l'estimateur noté $\hat{\theta}$:

$$\text{MSE}(\hat{\theta}|\theta) = \text{Biais}^2(\hat{\theta}) + \text{Var}(\hat{\theta})$$

```
> sqrt(mean(residuals(m1)^2))
[1] 1.251231
```

À vous !

- Comparez cette valeur avec celle que vous avez obtenue avec la sortie de la fonction `lm()`.
- Quelle est l'utilité d'observer l'erreur quadratique moyenne par rapport à la variance ?

1. Le test de Fisher n'est lisible que si les données suivent une loi normale

1.4 Limites de prédiction

La fonction suivante vous permet de calculer les limites d'un objet de prédiction `preds()` contenant des valeurs ajustées, l'erreur standard, et une estimation du niveau d'ensemble du bruit. Comme les données viennent de deux sources (indépendantes), cette fonction combine les écarts-types par « ajout en quadrature ». Plus simplement, en régression linéaire simple, il existe deux types d'intervalle de confiance que vous pouvez utiliser pour la prédiction :

- Construire un intervalle de confiance autour de la moyenne conditionnelle donnée pour une valeur de X .
- Construire un intervalle de confiance pour les valeurs réalisées Y pour une valeur de X donnée.

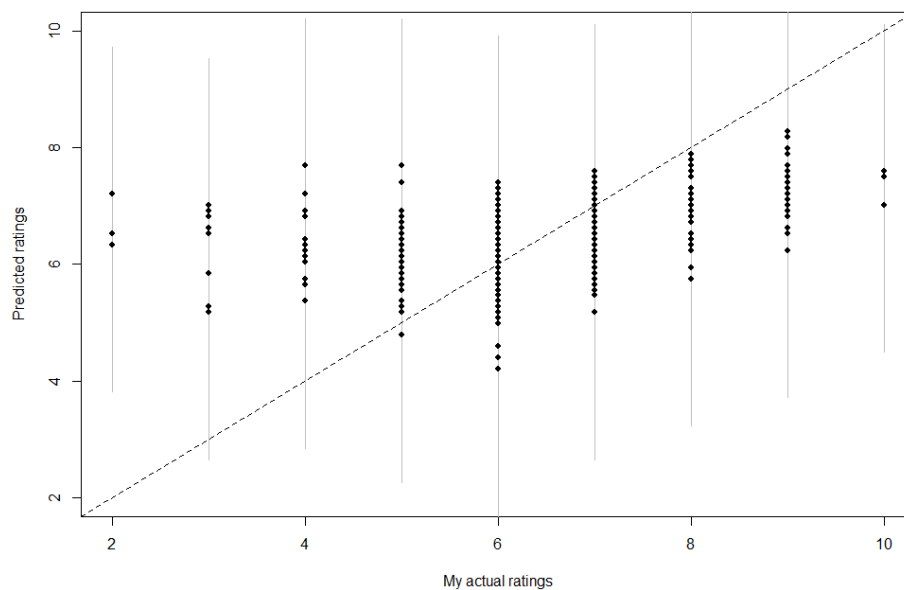
La fonction `predlims()` calcule un intervalle de prédiction approximatif en prenant l'écart-type estimé (à partir de l'erreur standard de la moyenne conditionnelle), et en le multipliant par 2 (valeur arrondie de 1,96).

```
predlims <- function(preds,sigma) {
  prediction.sd <- sqrt(preds$se.fit^2+sigma^2)
  upper <- preds$fit+2*prediction.sd
  lower <- preds$fit-2*prediction.sd
  lims <- cbind(lower=lower,upper=upper)
  return(lims)
}

> preds.lm <- predict(m1,se.fit=TRUE)
> predlims.lm <- predlims(preds.lm,sigma=summary(m1)$sigma)
> mean(d$mine <= predlims.lm[, "upper"]
      & d$mine >= predlims.lm[, "lower"])
[1] 0.957346

#Figure 5.
> plot(d$mine,preds.lm$fit,type="n", xlim=c(2,10), ylim=c(2,10),
      xlab="My actual ratings",ylab="Predicted ratings", main="")
> segments(d$mine,predlims.lm[, "lower"],
          d$mine,predlims.lm[, "upper"], col="grey")
> abline(a=0,b=1,lty="dashed")
> points(d$mine,preds.lm$fit,pch=16,cex=0.8)
```

FIGURE 5 – Intervalles de confiance des valeurs prédites



À vous !

- Qu'appelons-nous la probabilité de couverture? Quelle est sa valeur pour notre modèle?
- Rappelez la valeur du coefficient de détermination d'IMDb que vous aviez obtenu plus tôt.
- Que cela suppose-t-il?
- Que constatez-vous pour les bonnes notes? Pour les mauvaises notes?
- Concluez sur les cas où le modèle est « bon » et les cas où le modèle est « mauvais ».

1.5 Fonction de densité des notes

Il est possible d'observer la fonction de densité des notes distribuées par l'utilisateur. Nous observons les notes données par l'utilisateur à travers deux sous-ensembles de films : les films centrés dans l'écart-type des notes d'IMDb, et ceux dans les bornes externes.

```
> d1<-subset(d, d$imdb>6.49 & d$imdb<7.51)
> d2<-subset(d, d$imdb>7.51 & d$imdb<8.5)
```

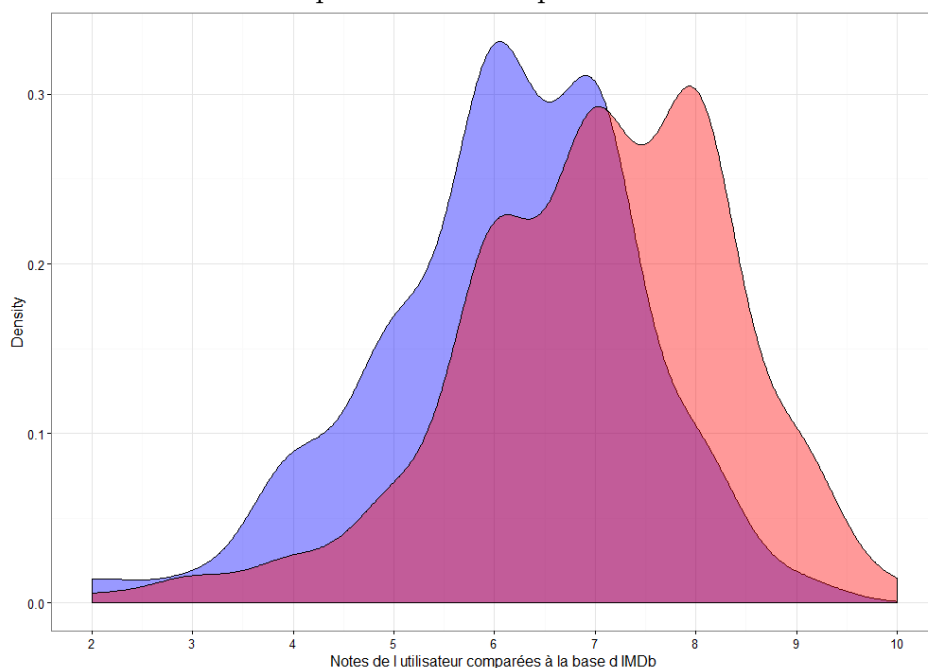
#Figure 6


```
> p3<-ggplot (NULL, aes(mine))+
  geom_density(data = d1, fill='blue',
  alpha=0.4,aes(x=mine, y = ..density..))+
  geom_density(data = d2, fill='red',
  alpha=0.4,aes(x=mine, y = ..density..))+
  scale_x_continuous
  ('Notes de l'utilisateur comparées à la base d'IMDb
  (blue: 6.5-7.5, red:7.5-8.5)', breaks=seq(2,10,1))
  + scale_y_continuous('Density')+theme_bw()+theme(legend.position="none")
> p3
```

À vous !

- Comparez les différents pics de fonction de densité pour les deux courbes.
- Concluez sur ce graphique. Notre utilisateur est-il un cinéphile ?

FIGURE 6 – Notes données par l'utilisateur pour deux sous-ensembles de films



1.6 Année de sortie

Nous sélectionnons les films à partir de l'année 1960. La dernière variable dans la régression linéaire ci-dessous est alors l'année de sortie du film.

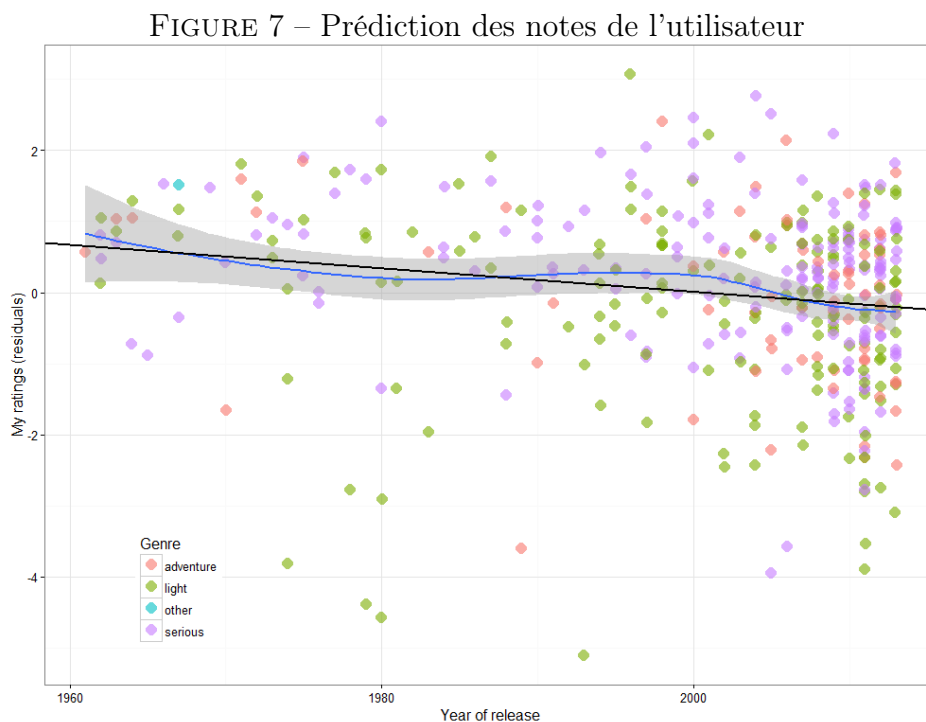
```
> d.60<-subset(d, Year>1960)
> d.60$r<-residuals(lm(d.60$mine~d.60$imdb))
```

#Figure 7.

```

> p4<- ggplot(d.60, aes(Year, r))+
  geom_point(position=position_jitter(width=0.1,height=.25),shape=16,
    size=4,alpha=0.6,
    aes(colour = new.genre, ))+
  stat_smooth()+
  scale_x_continuous('Year of release')+
  scale_y_continuous('My ratings (residuals)')+
  theme_bw()+
  scale_colour_discrete(name="Genre")+
  scale_size_continuous(guide=FALSE)+
  theme(legend.position=c(0.15, 0.15))+
  geom_abline(size=1, aes(intercept=33.33, slope=-0.016659))
> p4

```



À vous !

- Affichez un `summary()` de la fonction `lm()` sur `d.60$r`, selon la variable `Year`.
- Un film ancien est-il généralement mieux noté ? Pourquoi selon vous ?
- Commentez les résidus de la régression linéaire simple pour le sous-ensemble de films après l'année 1960.
- Commentez la Figure 7.
- Quelle recommandation pourriez-vous donner pour améliorer la prédiction ?

- f) Utiliser une A.N.O.V.A pour tester la si modèle idéal peut être considéré, a posteriori, comme linéaire.

1.7 Ajout des prédicateurs

Nous allons ajouter d'autres variables pour voir comment améliorer la qualité du modèle. Nous voyons souvent, lorsqu'un film sort, que le fait d'ajouter un réalisateur connu donne un effet de levier (nous disons aussi qu'un réalisateur est "bankable"). De la même façon, certains types de films (comédie, action,...) sont plus facilement populaires que d'autres.

```
> #Linear model 2
> summary(m2<-lm(mine~imdb+d$comedy +d$romance+d$mystery
+d$Stanley.Kubrick..+d$Lars.Von.Trier..+d$Darren.Aronofsky..+year.c,
data=d))
```

Call:

```
lm(formula = mine ~ imdb + d$comedy + d$romance + d$mystery +
    d$Stanley.Kubrick.. + d$Lars.Von.Trier.. + d$Darren.Aronofsky.. +
    year.c, data = d)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.4265	-0.6212	0.1631	0.7760	2.5917

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.074930	0.651223	1.651	0.099574 .
imdb	0.727829	0.087238	8.343	1.10e-15 ***
d\$comedy	-0.598040	0.133533	-4.479	9.74e-06 ***
d\$romance	-0.411929	0.141274	-2.916	0.003741 **
d\$mystery	0.315991	0.185906	1.700	0.089933 .
d\$Stanley.Kubrick..	1.066991	0.450826	2.367	0.018406 *
d\$Lars.Von.Trier..	2.117281	0.582790	3.633	0.000315 ***
d\$Darren.Aronofsky..	1.357664	0.584179	2.324	0.020607 *
year.c	0.016578	0.003693	4.488	9.32e-06 ***

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

Residual standard error: 1.156 on 413 degrees of freedom

Multiple R-squared: 0.3508, Adjusted R-squared: 0.3382

F-statistic: 27.89 on 8 and 413 DF, p-value: < 2.2e-16

```
> sqrt(mean(residuals(m2)^2)) #root mean squared error: 1.14
[1] 1.14322
```

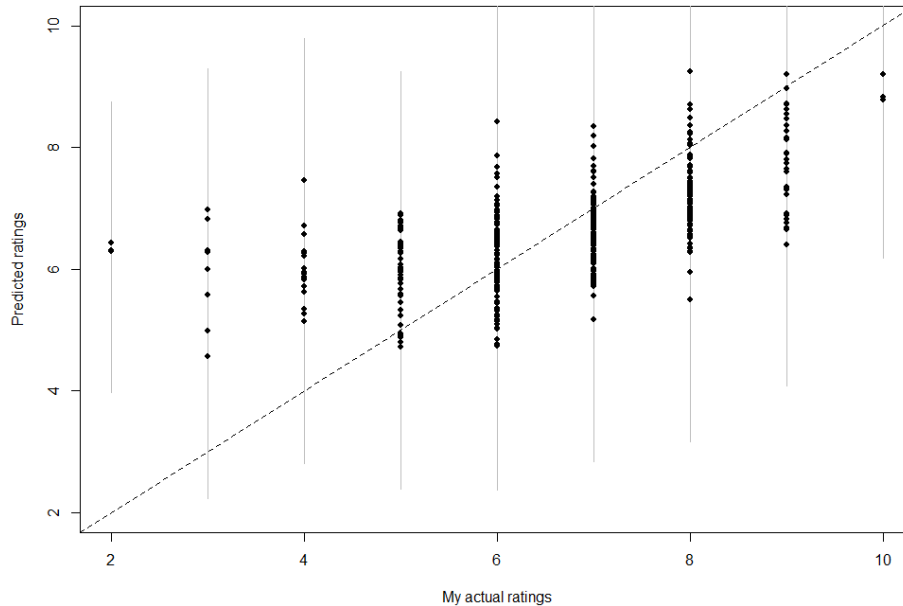
```

>
> preds.lm <- predict(m2,se.fit=TRUE)
> predlims.lm <- predlims(preds.lm,sigma=summary(m2)$sigma)
> mean(d$mine <= predlims.lm[, "upper"]
+      & d$mine >= predlims.lm[, "lower"]) #coverage of the prediction 96%
[1] 0.9597156

> #Figure 8.
> plot(d$mine,preds.lm$fit,type="n", xlim=c(2,10), ylim=c(2,10),
+      xlab="My actual ratings",ylab="Predicted ratings", main="")
> segments(d$mine,predlims.lm[, "lower"],
+          d$mine,predlims.lm[, "upper"], col="grey")
> abline(a=0,b=1,lty="dashed")
> points(d$mine,preds.lm$fit,pch=16,cex=0.8)

```

FIGURE 8 – Prédiction des notes de l'utilisateur



À vous !

- Comparez par rapport à l'erreur quadratique moyenne du premier modèle.
- Qu'avons-nous changé par rapport au premier modèle ?

- c) Pourquoi, selon vous, le modèle est-il potentiellement meilleur ? Regardez s'il est précisément meilleur sur un type de film particulier.
- d) Que pouvez-vous en conclure ?