TP3

Andrieu Carla et Zaari Abdelouahab

18/10/2021

IV. Real estate data

```
# TODO : Adjust the path of the housedata file
data <- read.table("C://Users//Lenovo//Desktop//housedata.csv",header=TRUE,
                   sep=',')
#removing id and date and zipcode columns
data$id <- NULL
data$date <- NULL
data$zipcode <- NULL
```

```
medianHousePrice=median(data$price);
data$medHousePriceBin=as.numeric(data$price>medianHousePrice);
```

Pourcentage des données manquantes : 0%

```
sum(is.na(data)) / (nrow(data) *ncol(data))
```

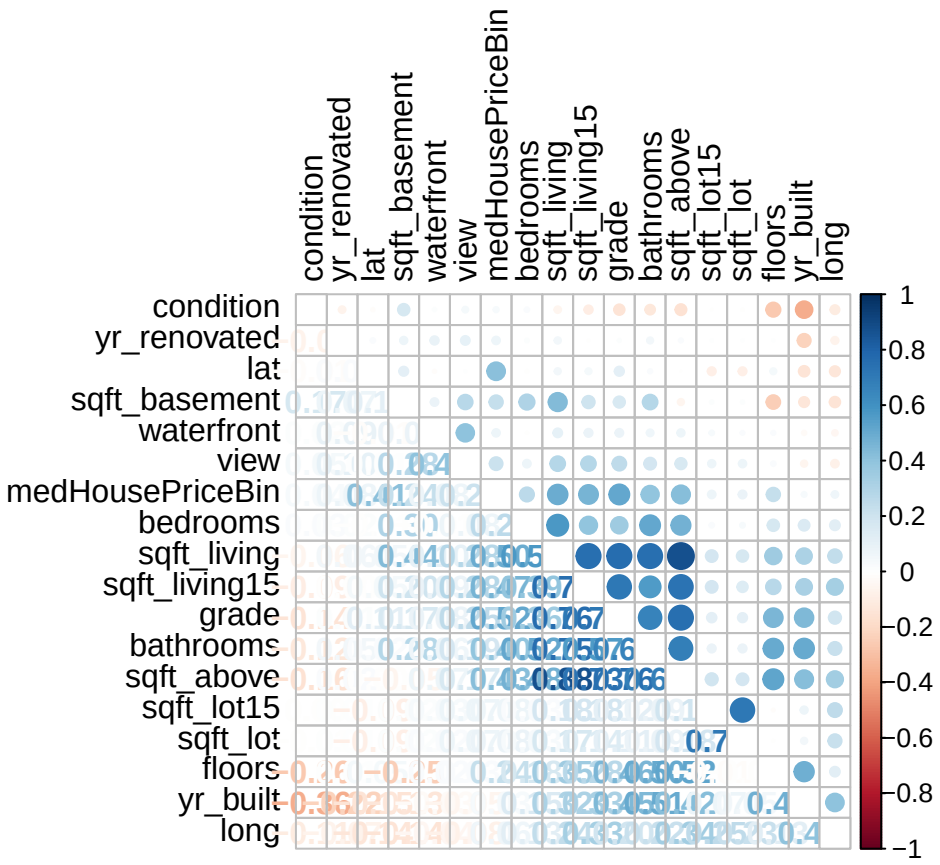
```
## [1] 0
```

On veut prédire la variable medHousePriceBin donc il n'est pas nécessaire de garder la variable price dans notre modèle

```
data$price <- NULL
```

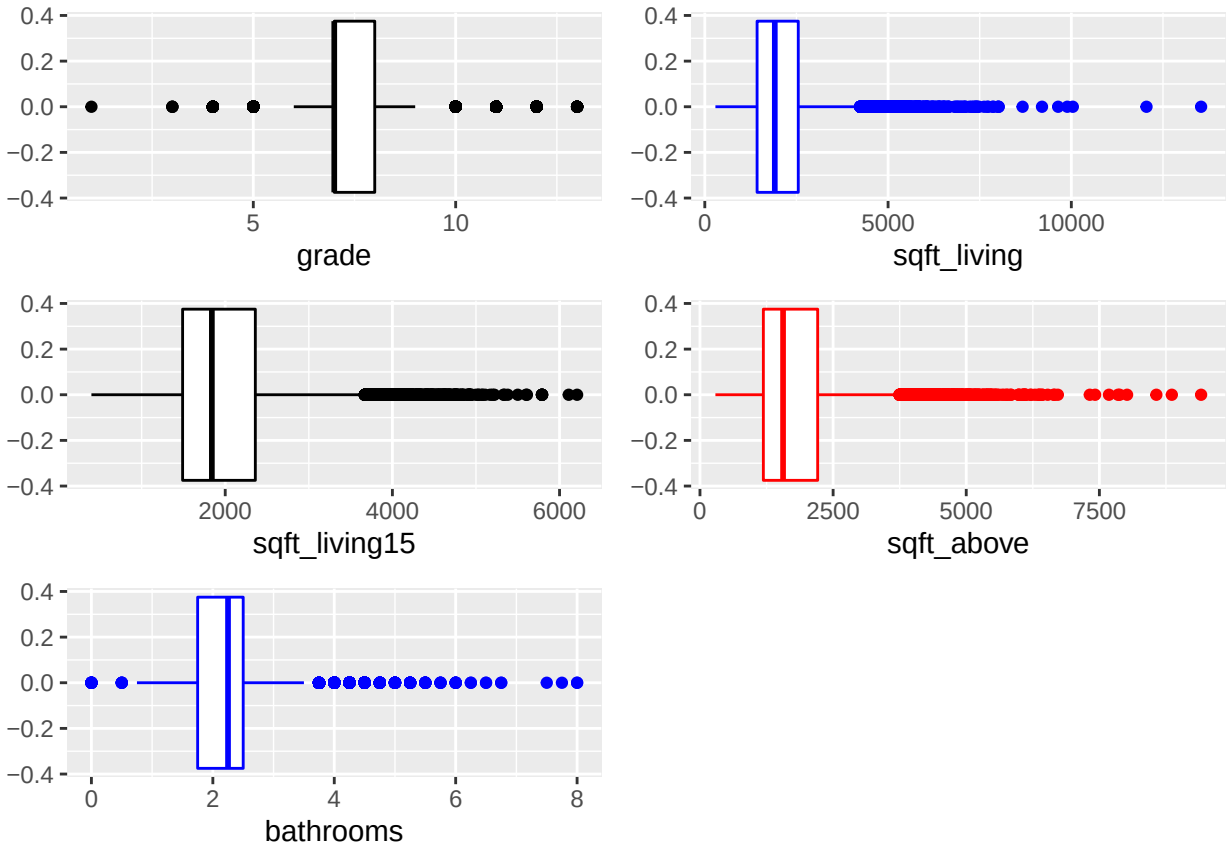
Visualisons la colinéarité entre la variable que l'on veut prédire avec les autres variables :

```
M <- cor(data)
corrplot.mixed(M, tl.col="black",order = 'AOE',tl.pos = "lt")
```



On trace des boxplots pour connaître la corrélation entre les variables :

```
p_1 <- ggplot(data , aes(x=grade))+
  geom_boxplot(col='black') + labs(x='grade')
p_2 <- ggplot(data , aes(x=sqft_living))+
  geom_boxplot(col='blue') + labs(x='sqft_living')
p_3 <- ggplot(data , aes(x=sqft_living15))+
  geom_boxplot(col='black') + labs(x='sqft_living15')
p_4 <- ggplot(data , aes(x= sqft_above))+
  geom_boxplot(col='red') + labs(x='sqft_above')
p_5 <- ggplot(data , aes(x=bathrooms))+
  geom_boxplot(col='blue') + labs(x='bathrooms')
grid.arrange(p_1,p_2,p_3,p_4,p_5, ncol=2, nrow = 3)
```

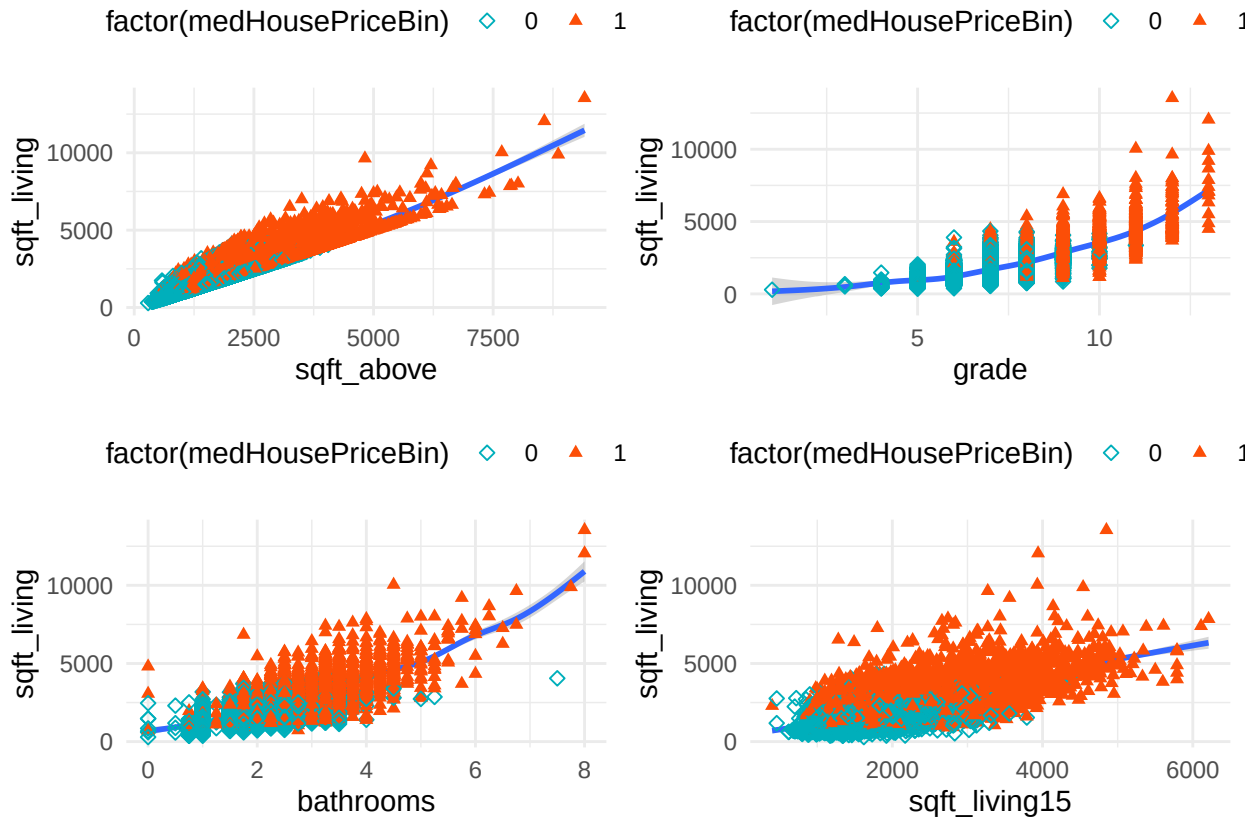


Ensuite, on veut visualiser la colinéarité de ces variables et la distribution avec les autres variables du modèle

```
p1 <- ggplot(data,aes(x= sqft_above, y=sqft_living)) +
  geom_smooth() + geom_point(aes(shape =factor(medHousePriceBin),color=factor(medHousePriceBin)))+
  scale_shape_manual(values = c(5,17)) +
  scale_color_manual(values = c("#00AFBB", "#FC4E07"))+
  theme_minimal() +
  theme(legend.position = "top")

p2 <- ggplot(data,aes(x= grade, y=sqft_living)) +
  geom_smooth() + geom_point(aes(shape =factor(medHousePriceBin),color=factor(medHousePriceBin)))+
  scale_shape_manual(values = c(5,17)) +
  scale_color_manual(values = c("#00AFBB", "#FC4E07"))+
  theme_minimal() +
  theme(legend.position = "top")

p3 <- ggplot(data,aes(x= bathrooms, y=sqft_living)) +
  geom_smooth() + geom_point(aes(shape =factor(medHousePriceBin),color=factor(medHousePriceBin)))+
  scale_shape_manual(values = c(5,17)) +
  scale_color_manual(values = c("#00AFBB", "#FC4E07"))+
  theme_minimal() +
  theme(legend.position = "top")

p4 <- ggplot(data,aes(x= sqft_living15, y=sqft_living)) +
  geom_smooth() + geom_point(aes(shape =factor(medHousePriceBin),color=factor(medHousePriceBin)))+
  scale_shape_manual(values = c(5,17)) +
  scale_color_manual(values = c("#00AFBB", "#FC4E07"))+
```

```
  theme_minimal() +
  theme(legend.position = "top")

grid.arrange(p1, p2, p3, p4, ncol=2, nrow = 2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



On peut clairement observer d'après les 4 graphiques montrant la distribution de la variable sqft_living que les autres variables sont corrélées à celle-ci. On constate que la valeur des variable grade, sqft_above, bathrooms, sqft_living15 augmente avec la surface de la maison ce qui engendre une augmentation du prix (voir les observations en orange). Ceci est logique car l'augemantation de la surface entraine l'augmentation des autres caractéristiques de la maison qui conduit par la suite à un prix élevé de la maison.

D'après les observations des graphes, on peut clairement constater que la distribution des observations entre sqft_above et sqft_living entraine une grande colinéarité qui peuvent aussi affecter les résultats du modèle. Nous choisissons alors de supprimer la variable sqft_above de notre base de données :

```
data$sqft_above <- NULL
```

On va utiliser maintenant la méthode de cross validation. Pour cela nous divisons les données de notre base de données en deux ensembles :

- un ensemble d'apprentissage
- un ensemble de test

```r
set.seed(2)
data_Split <- sort(sample(nrow(data),nrow(data)*.70))
train <- data[data_Split,]
test <- data[-data_Split,]
```

```r
#Création du modèle
model_1 <- glm(medHousePriceBin~.,data=train,family="binomial")
summary(model_1)
```

```
##
## Call:
## glm(formula = medHousePriceBin ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6706  -0.4804  -0.0444   0.4744   4.0515
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.391e+02  2.913e+01 -11.640  < 2e-16 ***
## bedrooms      -2.204e-01  3.695e-02  -5.965 2.45e-09 ***
## bathrooms      5.431e-01  6.149e-02   8.833  < 2e-16 ***
## sqft_living    9.067e-04  7.766e-05  11.675  < 2e-16 ***
## sqft_lot       7.076e-06  1.400e-06   5.053 4.35e-07 ***
## floors         6.154e-01  6.317e-02   9.741  < 2e-16 ***
## waterfront     2.329e+00  5.779e-01   4.030 5.59e-05 ***
## view           4.533e-01  5.032e-02   9.009  < 2e-16 ***
## condition      3.064e-01  4.261e-02   7.190 6.48e-13 ***
## grade          1.283e+00  4.658e-02  27.545  < 2e-16 ***
## sqft_basement  2.901e-04  8.592e-05   3.377 0.000734 ***
## yr_built      -3.378e-02  1.398e-03 -24.162  < 2e-16 ***
## yr_renovated   2.527e-05  6.989e-05   0.362 0.717688
## lat            1.026e+01  2.336e-01  43.941  < 2e-16 ***
## long           8.070e-01  2.145e-01   3.763 0.000168 ***
## sqft_living15  1.044e-03  7.276e-05  14.346  < 2e-16 ***
## sqft_lot15    -9.479e-07  1.858e-06  -0.510 0.609980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20972  on 15128  degrees of freedom
## Residual deviance: 10305  on 15112  degrees of freedom
## AIC: 10339
##
## Number of Fisher Scoring iterations: 6
```

On a pour ce premier modèle AIC: 10339 et on constate que les variables yr_renovated et sqft_lot15 ne sont pas significatives d'après les tests statistiques.

On va tester un deuxième modèle en utilisant la méthode de selection de variables backward, forward, stepwise pour avoir un modèle fiable pour réaliser les prédictions de notre variable medHousePriceBin

```r
model_1_backward = step(model_1,direction='backward');
```

```
## Start:  AIC=10338.74
```

```
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + yr_renovated + lat + long + sqft_living15 + sqft_lot15
##
##                 Df Deviance   AIC
## - yr_renovated   1    10305 10337
## - sqft_lot15     1    10305 10337
## <none>                10305 10339
## - sqft_basement  1    10316 10348
## - long           1    10319 10351
## - waterfront     1    10323 10355
## - sqft_lot       1    10333 10365
## - bedrooms       1    10340 10372
## - condition      1    10357 10389
## - bathrooms      1    10384 10416
## - view           1    10392 10424
## - floors         1    10400 10432
## - sqft_living    1    10446 10478
## - sqft_living15  1    10515 10547
## - yr_built       1    10953 10985
## - grade          1    11192 11224
## - lat            1    13096 13128
##
## Step:  AIC=10336.87
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + lat + long + sqft_living15 + sqft_lot15
##
##                 Df Deviance   AIC
## - sqft_lot15     1    10305 10335
## <none>                10305 10337
## - sqft_basement  1    10316 10346
## - long           1    10319 10349
## - waterfront     1    10324 10354
## - sqft_lot       1    10333 10363
## - bedrooms       1    10341 10371
## - condition      1    10357 10387
## - bathrooms      1    10386 10416
## - view           1    10393 10423
## - floors         1    10400 10430
## - sqft_living    1    10447 10477
## - sqft_living15  1    10515 10545
## - yr_built       1    11016 11046
## - grade          1    11193 11223
## - lat            1    13099 13129
##
## Step:  AIC=10335.13
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + lat + long + sqft_living15
##
##                 Df Deviance   AIC
## <none>                10305 10335
## - sqft_basement  1    10316 10344
```

```
## - long           1    10319 10347
## - waterfront     1    10324 10352
## - bedrooms       1    10341 10369
## - condition      1    10358 10386
## - sqft_lot       1    10370 10398
## - bathrooms      1    10387 10415
## - view           1    10393 10421
## - floors         1    10401 10429
## - sqft_living    1    10447 10475
## - sqft_living15  1    10515 10543
## - yr_built       1    11016 11044
## - grade          1    11194 11222
## - lat            1    13101 13129
```

```
summary(model_1_backward)
```

```
##
## Call:
## glm(formula = medHousePriceBin ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + floors + waterfront + view + condition + grade +
##     sqft_basement + yr_built + lat + long + sqft_living15, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6519  -0.4808  -0.0444   0.4743   4.0396
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.394e+02  2.896e+01 -11.720  < 2e-16 ***
## bedrooms      -2.203e-01  3.689e-02  -5.972 2.35e-09 ***
## bathrooms      5.462e-01  6.103e-02   8.950  < 2e-16 ***
## sqft_living    9.064e-04  7.759e-05  11.682  < 2e-16 ***
## sqft_lot       6.515e-06  8.556e-07   7.614 2.66e-14 ***
## floors         6.164e-01  6.315e-02   9.761  < 2e-16 ***
## waterfront     2.325e+00  5.762e-01   4.034 5.48e-05 ***
## view           4.530e-01  5.031e-02   9.004  < 2e-16 ***
## condition      3.040e-01  4.219e-02   7.205 5.80e-13 ***
## grade          1.284e+00  4.657e-02  27.563  < 2e-16 ***
## sqft_basement  2.893e-04  8.590e-05   3.368 0.000756 ***
## yr_built      -3.391e-02  1.350e-03 -25.116  < 2e-16 ***
## lat            1.026e+01  2.335e-01  43.960  < 2e-16 ***
## long           8.026e-01  2.136e-01   3.758 0.000171 ***
## sqft_living15  1.041e-03  7.261e-05  14.339  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20972  on 15128  degrees of freedom
## Residual deviance: 10305  on 15114  degrees of freedom
## AIC: 10335
##
## Number of Fisher Scoring iterations: 6
```

Pour ce modèle de sélection backward on a trouvé que AIC=10335 c'est mieux que le modèle précédent. De plus, d'après les tests statistiqueles ce nouveau modèle les variables yr_renovated et sqft_lot15 sont considérées comme signifiatives.

Avec la sélection forward :

```
model_1_forward = step(model_1,direction='forward');
```

```
## Start:  AIC=10338.74
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + yr_renovated + lat + long + sqft_living15 + sqft_lot15
```

```
summary(model_1_forward)
```

```
##
## Call:
## glm(formula = medHousePriceBin ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + floors + waterfront + view + condition + grade +
##     sqft_basement + yr_built + yr_renovated + lat + long + sqft_living15 +
##     sqft_lot15, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6706  -0.4804  -0.0444   0.4744   4.0515
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.391e+02  2.913e+01 -11.640  < 2e-16 ***
## bedrooms      -2.204e-01  3.695e-02  -5.965 2.45e-09 ***
## bathrooms      5.431e-01  6.149e-02   8.833  < 2e-16 ***
## sqft_living    9.067e-04  7.766e-05  11.675  < 2e-16 ***
## sqft_lot       7.076e-06  1.400e-06   5.053 4.35e-07 ***
## floors         6.154e-01  6.317e-02   9.741  < 2e-16 ***
## waterfront     2.329e+00  5.779e-01   4.030 5.59e-05 ***
## view           4.533e-01  5.032e-02   9.009  < 2e-16 ***
## condition      3.064e-01  4.261e-02   7.190 6.48e-13 ***
## grade          1.283e+00  4.658e-02  27.545  < 2e-16 ***
## sqft_basement  2.901e-04  8.592e-05   3.377 0.000734 ***
## yr_built      -3.378e-02  1.398e-03 -24.162  < 2e-16 ***
## yr_renovated   2.527e-05  6.989e-05   0.362 0.717688
## lat            1.026e+01  2.336e-01  43.941  < 2e-16 ***
## long           8.070e-01  2.145e-01   3.763 0.000168 ***
## sqft_living15  1.044e-03  7.276e-05  14.346  < 2e-16 ***
## sqft_lot15    -9.479e-07  1.858e-06  -0.510 0.609980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20972  on 15128  degrees of freedom
## Residual deviance: 10305  on 15112  degrees of freedom
## AIC: 10339
##
## Number of Fisher Scoring iterations: 6
```

On a trouvé AIC: 10339, mais il y a encore dans le modèle des variables qui ne sont pas significatives.

Avec la selection stepwise :

```
model_1_stepwise = step(model_1,direction='both');
```

```
## Start:  AIC=10338.74
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + yr_renovated + lat + long + sqft_living15 + sqft_lot15
##
##                 Df Deviance   AIC
## - yr_renovated   1    10305 10337
## - sqft_lot15     1    10305 10337
## <none>                10305 10339
## - sqft_basement  1    10316 10348
## - long           1    10319 10351
## - waterfront     1    10323 10355
## - sqft_lot       1    10333 10365
## - bedrooms       1    10340 10372
## - condition      1    10357 10389
## - bathrooms      1    10384 10416
## - view           1    10392 10424
## - floors         1    10400 10432
## - sqft_living    1    10446 10478
## - sqft_living15  1    10515 10547
## - yr_built       1    10953 10985
## - grade          1    11192 11224
## - lat            1    13096 13128
##
## Step:  AIC=10336.87
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + lat + long + sqft_living15 + sqft_lot15
##
##                 Df Deviance   AIC
## - sqft_lot15     1    10305 10335
## <none>                10305 10337
## + yr_renovated   1    10305 10339
## - sqft_basement  1    10316 10346
## - long           1    10319 10349
## - waterfront     1    10324 10354
## - sqft_lot       1    10333 10363
## - bedrooms       1    10341 10371
## - condition      1    10357 10387
## - bathrooms      1    10386 10416
## - view           1    10393 10423
## - floors         1    10400 10430
## - sqft_living    1    10447 10477
## - sqft_living15  1    10515 10545
## - yr_built       1    11016 11046
## - grade          1    11193 11223
## - lat            1    13099 13129
##
## Step:  AIC=10335.13
```

```
## medHousePriceBin ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     yr_built + lat + long + sqft_living15
##
##                 Df Deviance   AIC
## <none>                10305 10335
## + sqft_lot15     1    10305 10337
## + yr_renovated   1    10305 10337
## - sqft_basement  1    10316 10344
## - long           1    10319 10347
## - waterfront     1    10324 10352
## - bedrooms       1    10341 10369
## - condition      1    10358 10386
## - sqft_lot       1    10370 10398
## - bathrooms      1    10387 10415
## - view           1    10393 10421
## - floors         1    10401 10429
## - sqft_living    1    10447 10475
## - sqft_living15  1    10515 10543
## - yr_built       1    11016 11044
## - grade          1    11194 11222
## - lat            1    13101 13129
```

```
summary(model_1_stepwise)
```

```
##
## Call:
## glm(formula = medHousePriceBin ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + floors + waterfront + view + condition + grade +
##     sqft_basement + yr_built + lat + long + sqft_living15, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6519  -0.4808  -0.0444   0.4743   4.0396
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.394e+02  2.896e+01 -11.720  < 2e-16 ***
## bedrooms      -2.203e-01  3.689e-02  -5.972 2.35e-09 ***
## bathrooms      5.462e-01  6.103e-02   8.950  < 2e-16 ***
## sqft_living    9.064e-04  7.759e-05  11.682  < 2e-16 ***
## sqft_lot       6.515e-06  8.556e-07   7.614 2.66e-14 ***
## floors         6.164e-01  6.315e-02   9.761  < 2e-16 ***
## waterfront     2.325e+00  5.762e-01   4.034 5.48e-05 ***
## view           4.530e-01  5.031e-02   9.004  < 2e-16 ***
## condition      3.040e-01  4.219e-02   7.205 5.80e-13 ***
## grade          1.284e+00  4.657e-02  27.563  < 2e-16 ***
## sqft_basement  2.893e-04  8.590e-05   3.368 0.000756 ***
## yr_built      -3.391e-02  1.350e-03 -25.116  < 2e-16 ***
## lat            1.026e+01  2.335e-01  43.960  < 2e-16 ***
## long           8.026e-01  2.136e-01   3.758 0.000171 ***
## sqft_living15  1.041e-03  7.261e-05  14.339  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20972  on 15128  degrees of freedom
## Residual deviance: 10305  on 15114  degrees of freedom
## AIC: 10335
##
## Number of Fisher Scoring iterations: 6
```

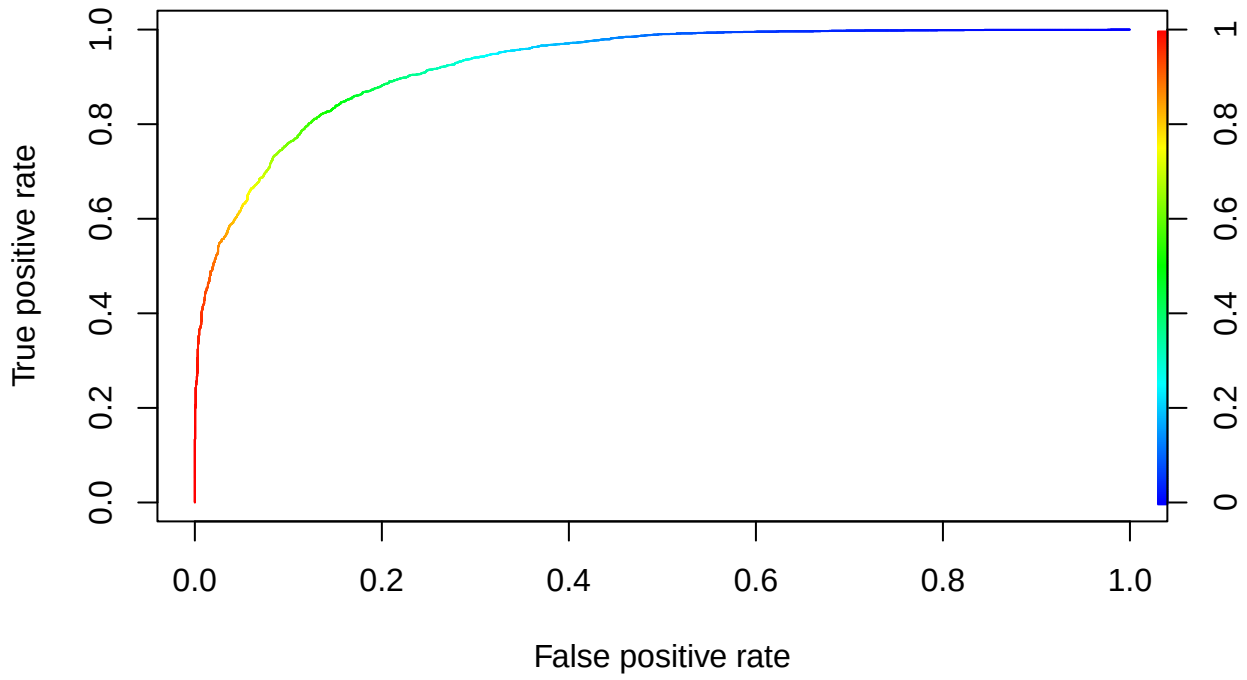
On a de meilleurs résultats avec la séléction forward et semblable au backward donc on peut garder ce modèle pour réaliser les prédictions.

On va choisir ce troisième modèle pour réaliser les prédictions.

```
#Evaluation du modèle
pred_test <- predict(model_1_stepwise,test,type="response")
```

On affiche le graphe ROC qui va nous aider à choisir threshold pour notre modèle

```
roc_pred <- prediction(pred_test,test$medHousePriceBin)
roc_test = performance(roc_pred, measure = "tpr", x.measure = "fpr")
plot(roc_test,colorize=TRUE)
```



D'après la ROC curve générée, on peut prendre un seuil de 0.5. En effet, c'est la meilleure valeur pour notre modèle afin d'avoir de bons résultats en terme de prédiction et donc moins d'erreur. Après avoir choisit ce seuil, on obtiens la matrice de confusion composée des faux et vrais positifs et négatifs.

```
pred_test <- ifelse(pred_test>0.5,1,0)
tab_prediction <- table(prediction=pred_test,actuelle=test$medHousePriceBin)
```

```r
tab_prediction
```

```
##           actuelle
## prediction    0    1
##          0 2738  538
##          1  482 2726
```

```r
model_accuracy <- (sum(diag(tab_prediction))/sum(tab_prediction))*100
paste("La précision de notre modèle est",model_accuracy,"%")
```

```
## [1] "La précision de notre modèle est 84.2689697717458 %"
```