Project1-452

Thu Tran

2023-04-05

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(rpart)
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(vip)
```

```
##
## Attaching package: 'vip'
##
## The following object is masked from 'package:utils':
##
##     vi
```

```
library(AppliedPredictiveModeling)
data("abalone")
```

Part 1: Exploratory Data Analysis

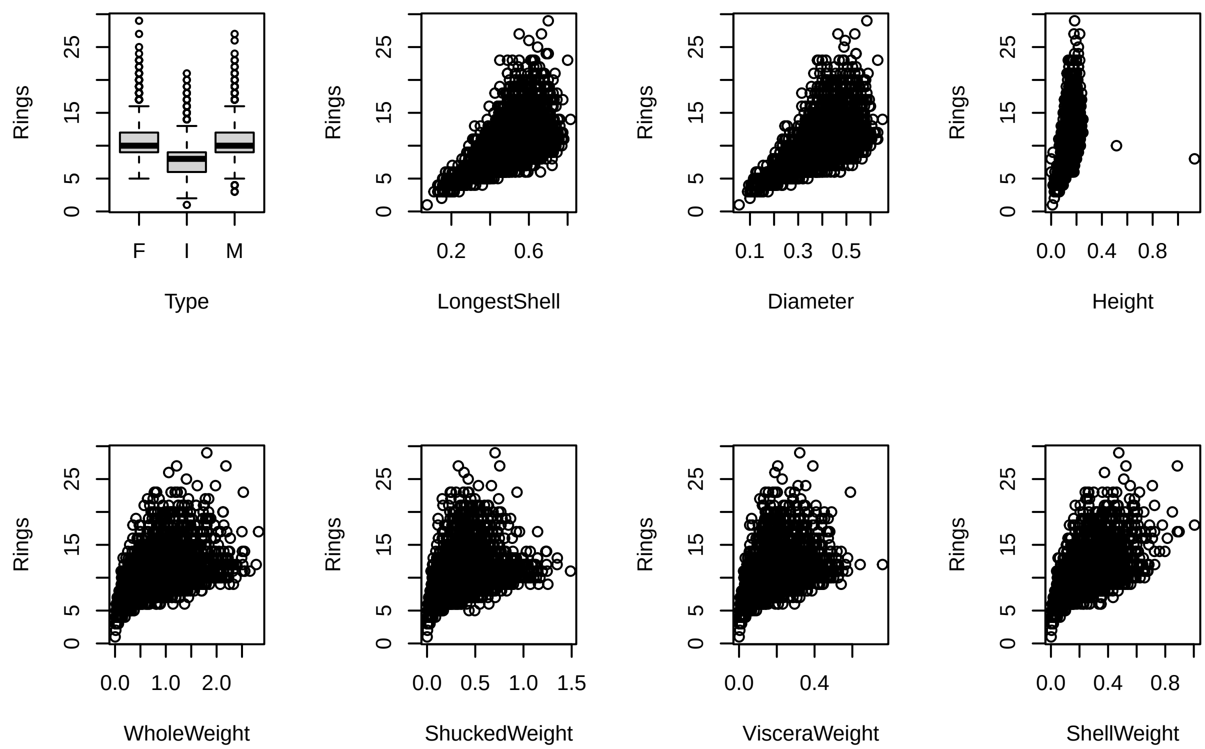
```
glimpse (abalone)
```

```
## Rows: 4,177
## Columns: 9
## $ Type          <fct> M, M, F, M, I, I, F, F, M, F, F, M, M, F, F, M, I, F, M,~
## $ LongestShell  <dbl> 0.455, 0.350, 0.530, 0.440, 0.330, 0.425, 0.530, 0.545, ~
## $ Diameter      <dbl> 0.365, 0.265, 0.420, 0.365, 0.255, 0.300, 0.415, 0.425, ~
## $ Height        <dbl> 0.095, 0.090, 0.135, 0.125, 0.080, 0.095, 0.150, 0.125, ~
## $ WholeWeight   <dbl> 0.5140, 0.2255, 0.6770, 0.5160, 0.2050, 0.3515, 0.7775, ~
## $ ShuckedWeight <dbl> 0.2245, 0.0995, 0.2565, 0.2155, 0.0895, 0.1410, 0.2370, ~
## $ VisceraWeight <dbl> 0.1010, 0.0485, 0.1415, 0.1140, 0.0395, 0.0775, 0.1415, ~
## $ ShellWeight   <dbl> 0.150, 0.070, 0.210, 0.155, 0.055, 0.120, 0.330, 0.260, ~
## $ Rings         <int> 15, 7, 9, 10, 7, 8, 20, 16, 9, 19, 14, 10, 11, 10, 10, 1~
```

```
# Summary statistics for the variables
summary(abalone)
```

```
##   Type      LongestShell      Diameter          Height         WholeWeight
##   F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000   Min.   :0.0020
##   I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150   1st Qu.:0.4415
##   M:1528   Median :0.545   Median :0.4250   Median :0.1400   Median :0.7995
##            Mean   :0.524   Mean   :0.4079   Mean   :0.1395   Mean   :0.8287
##            3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650   3rd Qu.:1.1530
##            Max.   :0.815   Max.   :0.6500   Max.   :1.1300   Max.   :2.8255
##   ShuckedWeight    VisceraWeight     ShellWeight         Rings
##   Min.   :0.0010   Min.   :0.0005   Min.   :0.0015   Min.   : 1.000
##   1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300   1st Qu.: 8.000
##   Median :0.3360   Median :0.1710   Median :0.2340   Median : 9.000
##   Mean   :0.3594   Mean   :0.1806   Mean   :0.2388   Mean   : 9.934
##   3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290   3rd Qu.:11.000
##   Max.   :1.4880   Max.   :0.7600   Max.   :1.0050   Max.   :29.000
```

```
attach(abalone)
# Scatter plots
par(mfrow=c(2,4))
plot(Rings~Type)
plot(Rings~LongestShell)
plot(Rings~Diameter)
plot(Rings~Height)
plot(Rings~WholeWeight)
plot(Rings~ShuckedWeight)
plot(Rings~VisceraWeight)
plot(Rings~ShellWeight)
```



- It looks like Rings have a positive relationship with almost all predictors (LongestShell,Diameter,Height,WholeWeight,Shucke
except the Type predictors. It's not sure whether they have a linear association or not since there is a fanning pattern appears in the scatter plots between Rings and LongestShell,Diameter,WholeWeight, ShuckedWeight, VisceraWeight, ShellWeight

Part 2: Cross-Validation

a. Split to trainning and test set

```
set.seed(123)
n<-nrow(abalone)
train_index<-sample(1:n, round(n*0.7))
abalone_train<-abalone[train_index,]
abalone_test <-abalone[-train_index,]
```

b.Fit a multilinear regression model

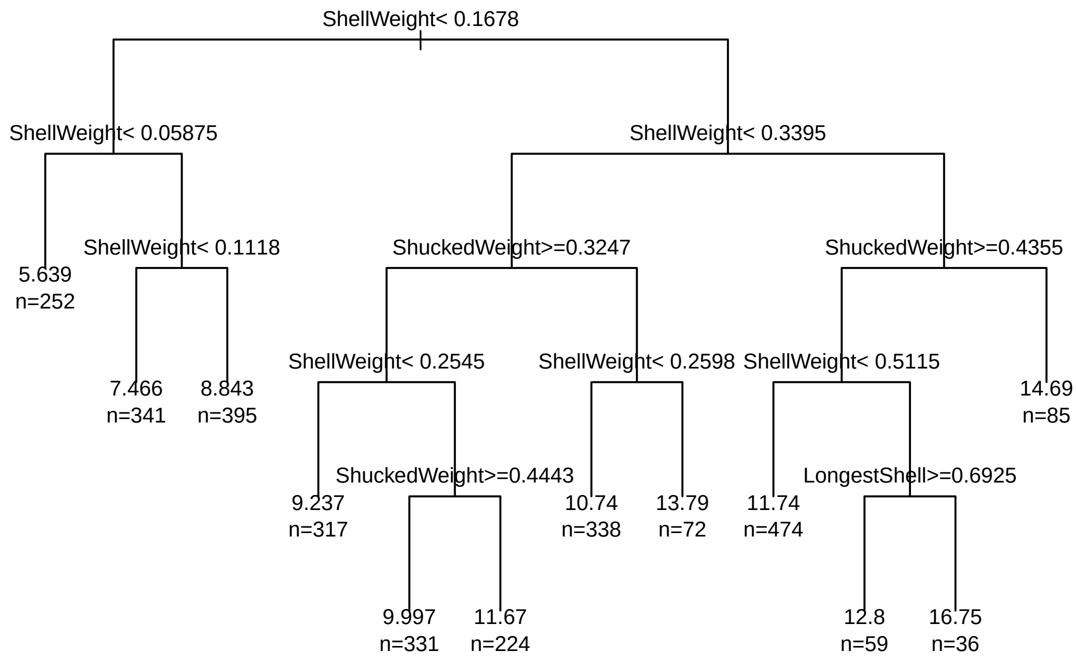
```
fit_ml<-lm(Rings~ .,data=abalone_train)
summary(fit_ml) # fix to just print the coefficient
```

```
##
```

```
## Call:
## lm(formula = Rings ~ ., data = abalone_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.444 -1.313 -0.336  0.880 14.136
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.0932     0.3492   8.858  < 2e-16 ***
## TypeI          -0.7255     0.1223  -5.931 3.37e-09 ***
## TypeM           0.1159     0.0993   1.167 0.243116
## LongestShell   -0.7745     2.1287  -0.364 0.716018
## Diameter        9.8634     2.6414   3.734 0.000192 ***
## Height         25.1272     2.7536   9.125  < 2e-16 ***
## WholeWeight     9.0528     0.8657  10.457  < 2e-16 ***
## ShuckedWeight -19.9145     0.9915 -20.086  < 2e-16 ***
## VisceraWeight -11.8409     1.5667  -7.558 5.46e-14 ***
## ShellWeight     7.1323     1.3232   5.390 7.60e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.186 on 2914 degrees of freedom
## Multiple R-squared:  0.5442, Adjusted R-squared:  0.5428
## F-statistic: 386.6 on 9 and 2914 DF,  p-value: < 2.2e-16
```

c. Fit a regresstion tree

```
fit_tree<-rpart(Rings~ .,data= abalone_train, method= "anova")
# Plot the tree
par(cex=0.7,xpd=NA)
plot(fit_tree, uniform= TRUE)
text(fit_tree, use.n=TRUE)
```

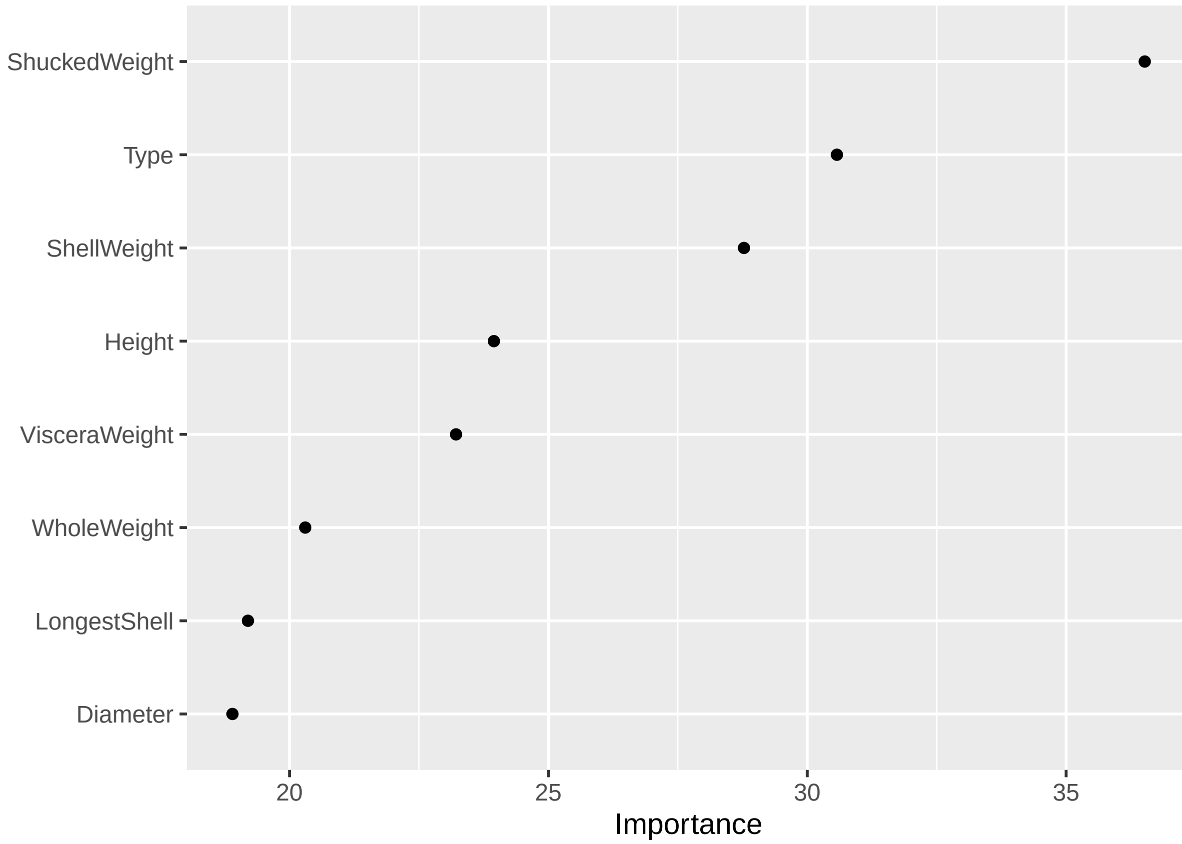


d. Fit model with randomForest

```
fit_rf<-randomForest(Rings~ ., data= abalone_train, importance = TRUE)
fit_rf
```

```
##
## Call:
##  randomForest(formula = Rings ~ ., data = abalone_train, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 4.679903
##                     % Var explained: 55.22
```

```
# Importance plot
vip(fit_rf,geom ="point")
```



e. Make prediction on the test set for multiple linear regression, regression tree, and random forests

```
# Make prediction
pred_ml<-predict(fit_ml, newdata = abalone_test)
pred_rf <- predict(fit_rf, newdata = abalone_test)
pred_tree <- predict(fit_tree, newdata = abalone_test)
```

```
# RMSE and R^2
RMSE <- function(y, y_hat) {
  sqrt(mean((y - y_hat)^2))
}
rmse<- c(RMSE(abalone_test$Rings,pred_ml),RMSE(abalone_test$Rings,pred_tree),
         RMSE(abalone_test$Rings,pred_rf))
r2<- c(cor(abalone_test$Rings, pred_ml)^2,cor(abalone_test$Rings, pred_tree)^2,
       cor(abalone_test$Rings, pred_rf)^2)
model<- c("Multiple Linear model","Regression Tree model","Random Forest model")
predict_tb<-data.frame(model,rmse,r2)
predict_tb
```

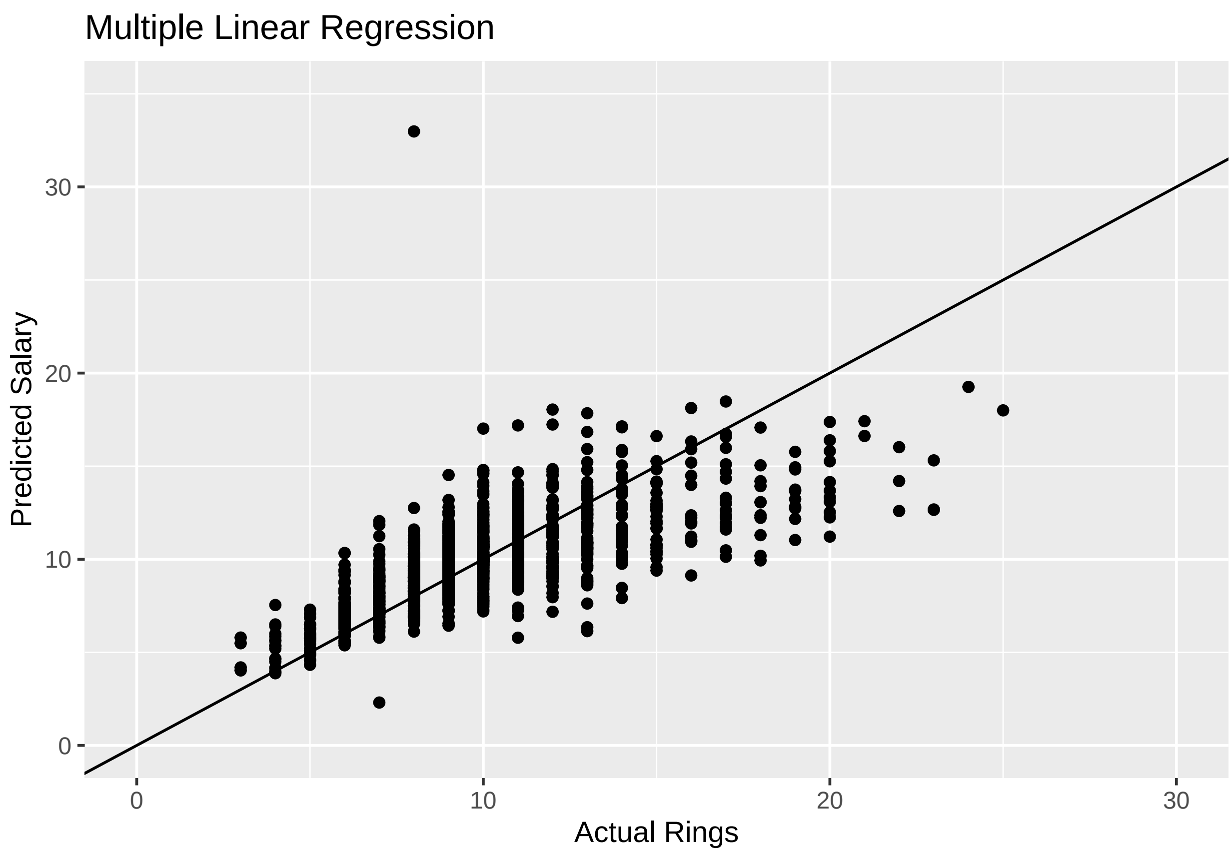
```
##                   model     rmse        r2
## 1 Multiple Linear model 2.288825 0.4955092
## 2 Regression Tree model 2.401850 0.4382897
## 3   Random Forest model 2.114117 0.5647526
```

f. Make plots of the predicted versus actual values

```
df_predict<-data.frame(
  Actual = abalone_test$Rings,
  Pred_ML=pred_ml,
  Pred_RF=pred_rf,
  Pred_TREE=pred_tree
)
```

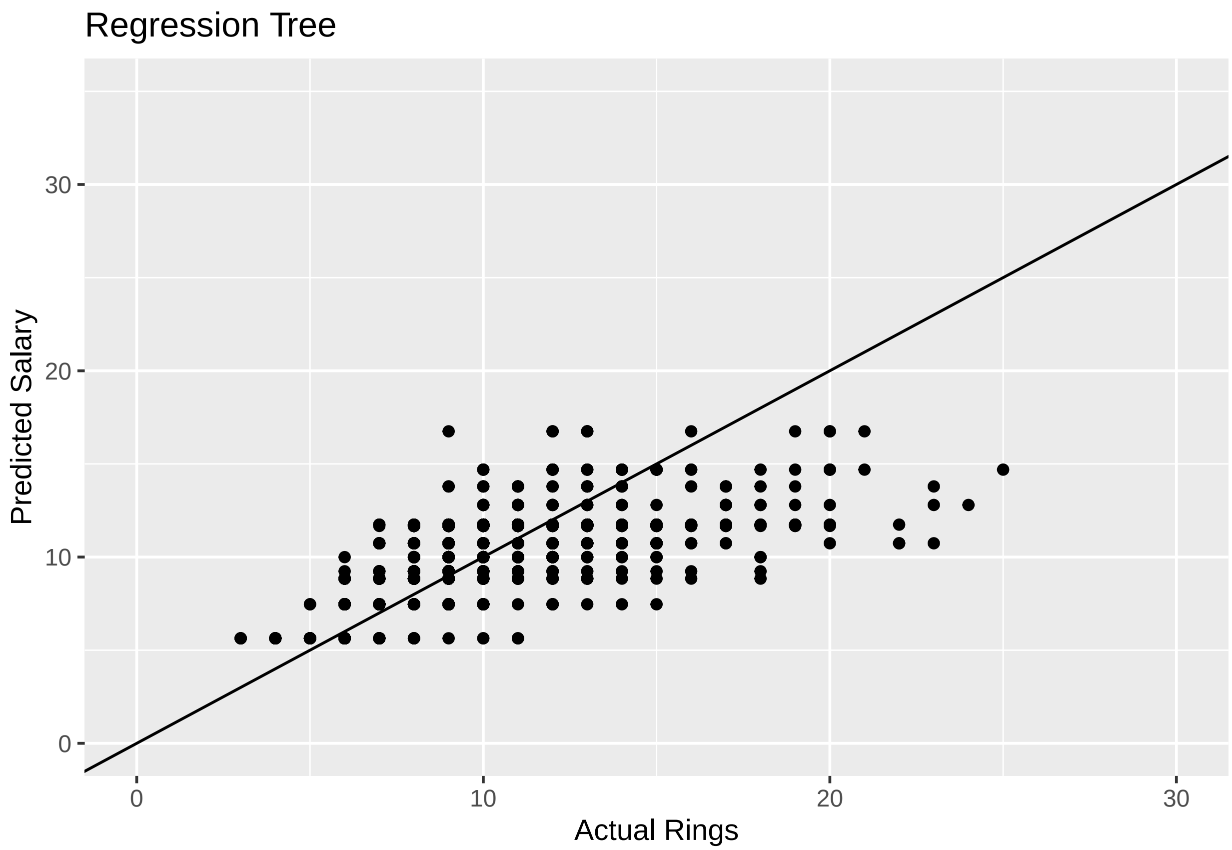
```
# Multiple linear
ggplot(df_predict,aes(x=Actual, y= Pred_ML))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  xlab("Actual Rings")+ ylab("Predicted Salary")+
  ggtitle("Multiple Linear Regression")+
  xlim(0,30)+ylim(0,35)
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

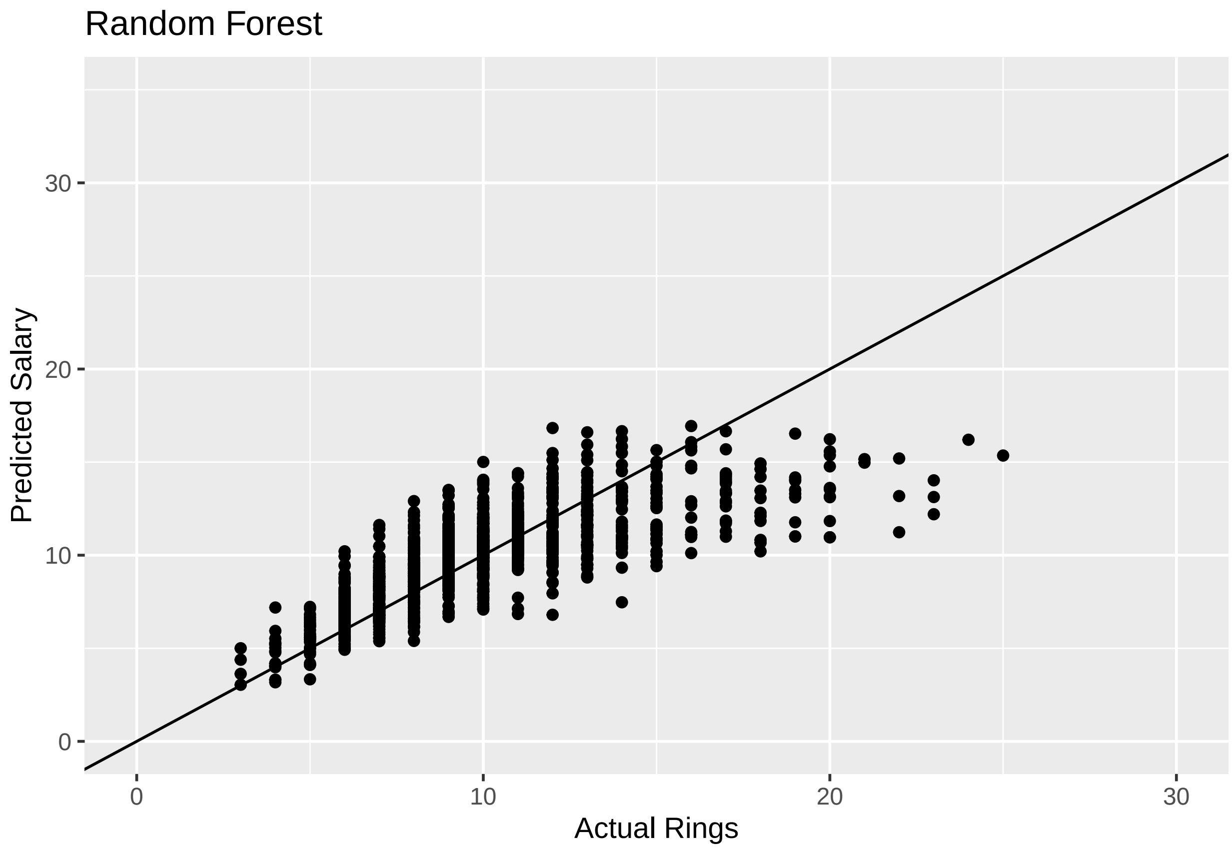



```
# Regression tree
ggplot(df_predict,aes(x=Actual, y= Pred_TREE))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1)+
```

```
  xlab("Actual Rings")+ ylab("Predicted Salary")+
  ggtitle("Regression Tree")+
  xlim(0,30)+ylim(0,35)
```




```
# Random Forest
ggplot(df_predict,aes(x=Actual, y= Pred_RF))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  xlab("Actual Rings")+ ylab("Predicted Salary")+
  ggtitle("Random Forest")+
  xlim(0,30)+ylim(0,35)
```



Interpret: - As visualizing the plots about the predicted versus actual values of different method, the random forest is the best fit version since the points are closed to the regression line. From the regression tree from c, there are 11 internal nodes which can be seen in predicted regression tree plot as 11 horizontal value of predicted salary. In the multiple linear regression, we can see an outlier that not fit in, so multiple linear regression maybe not a good model for prediction in this case.