lars v.s. Random forest

- Does random forest generalize better across divisions in the Hitters data than linear models (via lars) for the process of salary selection?
- Why might it be reasonable to think this given the way trees work?
- Use the plot and correlation of the cross predictions for lars vs salary and random forest vs salary to answer this question. (so 32 plots and correlations (including the self-prediction)

LARS

Predicting AW:

```{r}
# creating variables of the Hitters data base from ISLR
HittersAE <- Hitters%>%subset(League == "A")%>%subset(Division == "E")
HittersAW <- Hitters%>%subset(League == "A")%>%subset(Division == "W")
HittersNE <- Hitters%>%subset(League == "N")%>%subset(Division == "E")
HittersNW <- Hitters%>%subset(League == "N")%>%subset(Division == "W")
```

```{r}
# cleaning the data
HittersAEclean<-HittersAE%>% subset(!is.na(Salary))
HittersAWclean<-HittersAW%>% subset(!is.na(Salary))
HittersNEclean<-HittersNE%>% subset(!is.na(Salary))
HittersNWclean<-HittersNW%>% subset(!is.na(Salary))
```

```{r}
#Lars
HittersAEclean.lars<-lars(as.matrix(HittersAEclean[,-c(20,19,14,15)]),HittersAEclean$Salary)
I_AE<-HittersAEclean.lars$Cp==min(HittersAEclean.lars$Cp)
betalarsAE<-HittersAEclean.lars$beta[I_AE,]
betalarsAE
```

```
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years     CAtBat 
-3.3530002 10.0477043 -5.1870815  1.9252299  0.0000000  6.1607285  0.0000000 -0.3771068 
     CHits     CHmRun      CRuns       CRBI     CWalks    PutOuts    Assists     Errors 
 0.0000000  0.8418551  2.9480375  1.0145354 -1.1760056  0.3408382  0.1259118  0.0000000 
```

```{r}
HittersAWclean.lars<-lars(as.matrix(HittersAWclean[,-c(20,19,14,15)]),HittersAWclean$Salary)
I_AW<-HittersAWclean.lars$Cp==min(HittersAWclean.lars$Cp)
betalarsAW<-HittersAWclean.lars$beta[I_AW,]
betalarsAW
```

```
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years     CAtBat 
0.00000000 2.30243447 0.00000000 0.00000000 0.00000000 1.03768371 0.00000000 0.00000000 
     CHits     CHmRun      CRuns       CRBI     CWalks    PutOuts    Assists     Errors 
0.33434397 0.00000000 0.00000000 0.00000000 0.00000000 0.08998272 0.00000000 0.00000000 
```

```{r}
HittersNEclean.lars<-lars(as.matrix(HittersNEclean[,-c(20,19,14,15)]),HittersNEclean$Salary)
I_NE<-HittersNEclean.lars$Cp==min(HittersNEclean.lars$Cp)
betalarsNE<-HittersNEclean.lars$beta[I_NE,]
betalarsNE
```

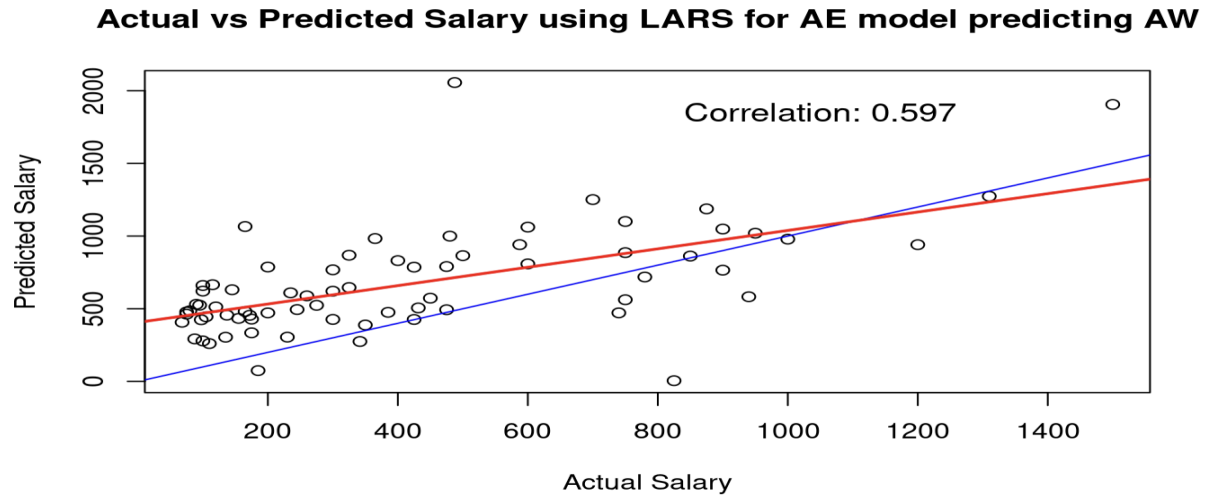
```
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years     CAtBat 
 -0.2481822  0.0000000  0.0000000  0.0000000  0.0000000  2.5323541  0.0000000  0.0000000 
     CHits     CHmRun      CRuns       CRBI     CWalks    PutOuts    Assists     Errors 
  0.0000000  0.0000000  0.0000000  0.8941327  0.0000000  0.4312271  0.7716719 -8.4960887 
```

```{r}
HittersNWclean.lars<-lars(as.matrix(HittersNWclean[,-c(20,19,14,15)]),HittersNWclean$Salary)
I_NW<-HittersNWclean.lars$Cp==min(HittersNWclean.lars$Cp)
betalarsNW<-HittersNWclean.lars$beta[I_NW,]
betalarsNW
```

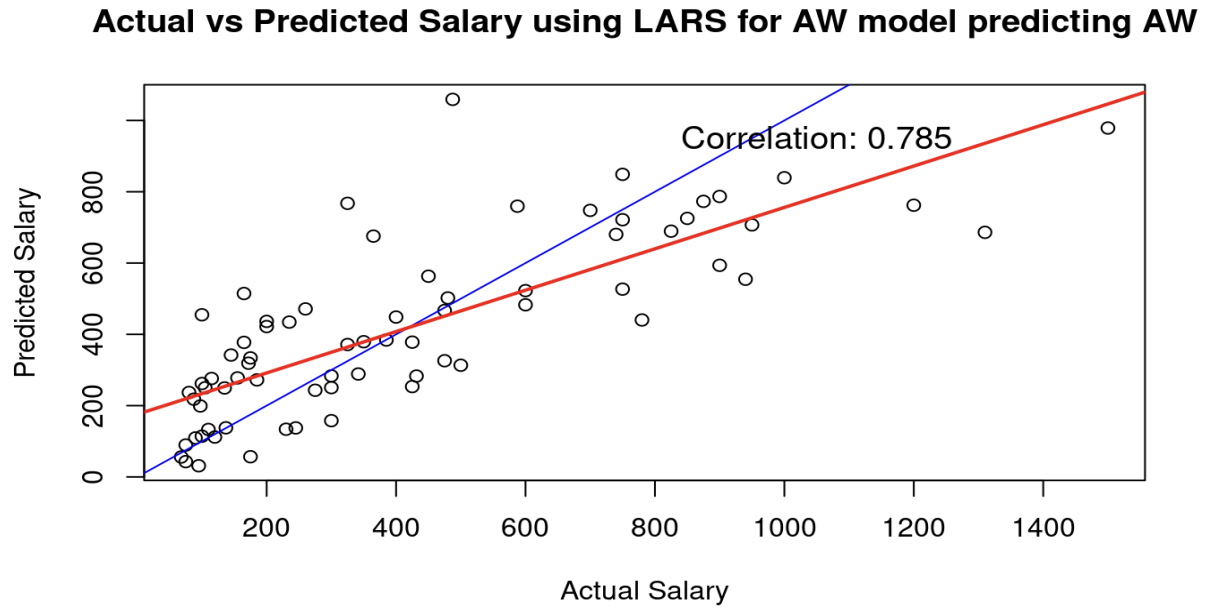
```
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years 
 1.3713374  1.9961871 32.2358918 -9.7338778 -11.9613345  7.3627544  7.7832856 
    CAtBat      CHits     CHmRun      CRuns       CRBI     CWalks    PutOuts 
 0.0000000  0.4414019  0.4438961  0.0000000  0.0000000 -0.9914020  0.0000000 
   Assists     Errors 
-0.7387407  3.8378122 
```

```r
predict_and_plot <-
  function(train_data, model_betas, test_data, train_division, test_division) {
    # Predict salaries for the test data
    pred <- as.matrix(test_data[,-c(20,19,14,15)]) %*% model_betas
    # Adjust the predicted values using the mean adjustment
    pred <- pred + (mean(train_data$Salary) - mean(pred))
    # Plot actual vs predicted salaries with Actual Salary on the x-axis
    # abline (0,1) adds a reference line with an intercept of 0 and a slope of 1 to the plot.
    # It represents a perfect prediction where predicted salary = actual salary.
    # lm_fit variable will add a linear regression line to the data
    plot(test_data$Salary, pred,
         main = paste("Actual vs Predicted Salary using LARS for",
                      train_division,"model predicting", test_division),
         xlab = "Actual Salary", ylab = "Predicted Salary")
    abline(0, 1, col = "blue")
    lm_fit <- lm(pred ~ test_data$Salary)
    abline(lm_fit, col = "red", lwd = 2)
 # Calculate the correlation
    correlation <- cor(pred, test_data$Salary)

    # Print the correlation on the plot
    text(x = max(test_data$Salary) * 0.7, y = max(pred) * 0.9,
         labels = paste("Correlation:", round(correlation, 3)),
         col = "black", cex = 1.2)

    return(correlation)
  }
#Red = Linear Regression Line
#Blue = Perfect prediction Line
```

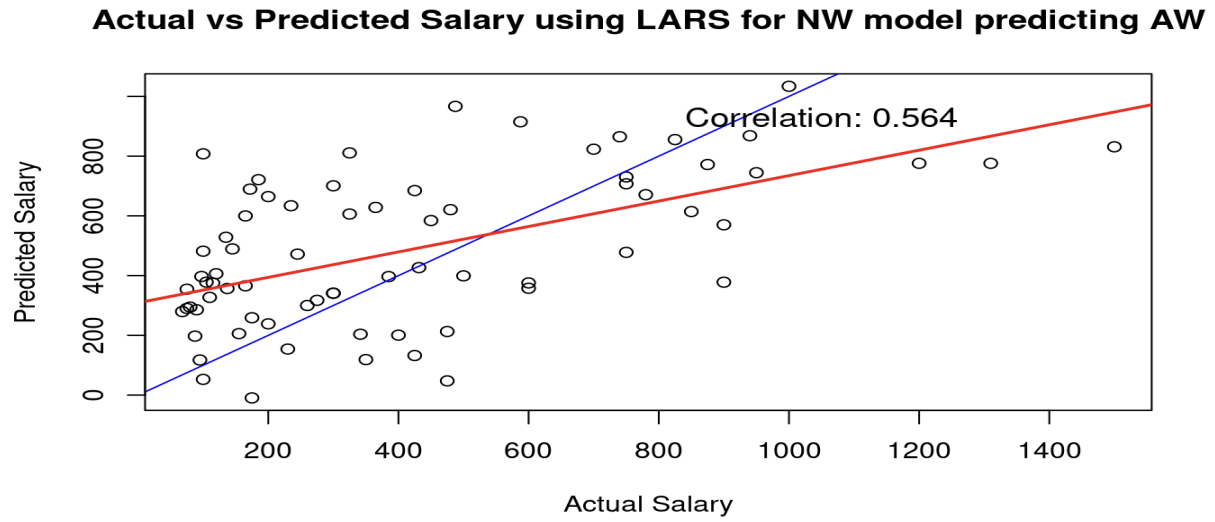
```r
# Predict for AW using AE
correlation_AW_AE <-
  predict_and_plot(HittersAEclean, betalarsAE, HittersAWclean, "AE", "AW")
```



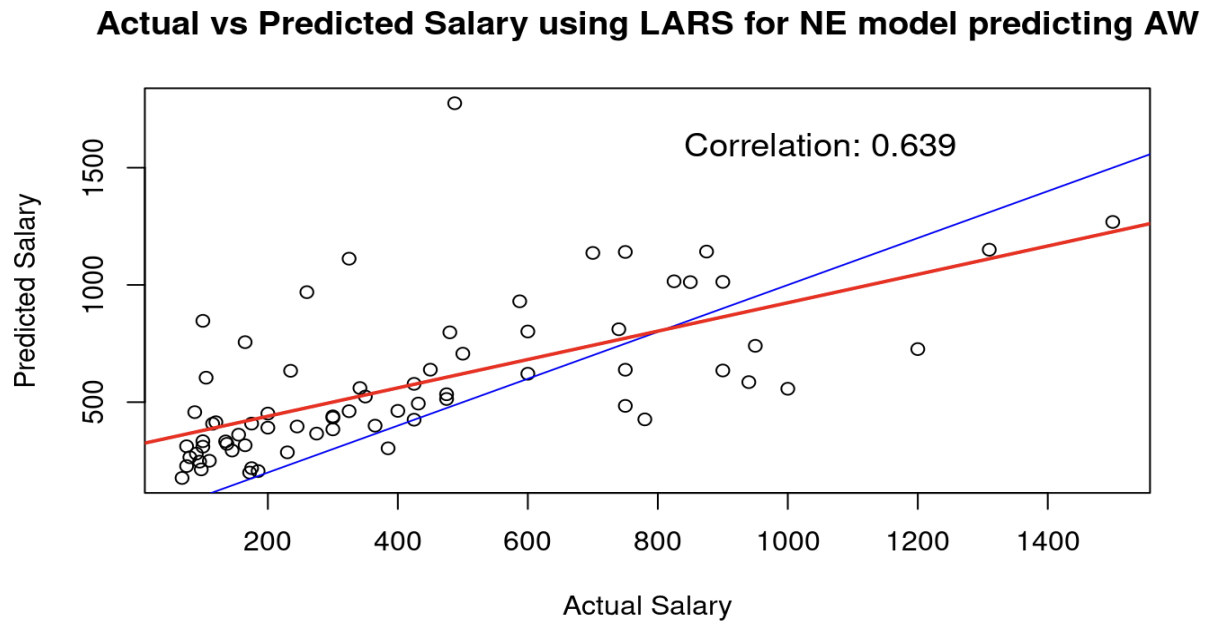

```r
#Predict for AW using AW model
correlation_AW_AW <-
  predict_and_plot(HittersAWclean, betalarsAW, HittersAWclean, "AW", "AW")
```



```r
# Predict for AW using NW
correlation_AW_NW <-
  predict_and_plot(HittersNWclean, betalarsNW, HittersAWclean, "NW", "AW")
```

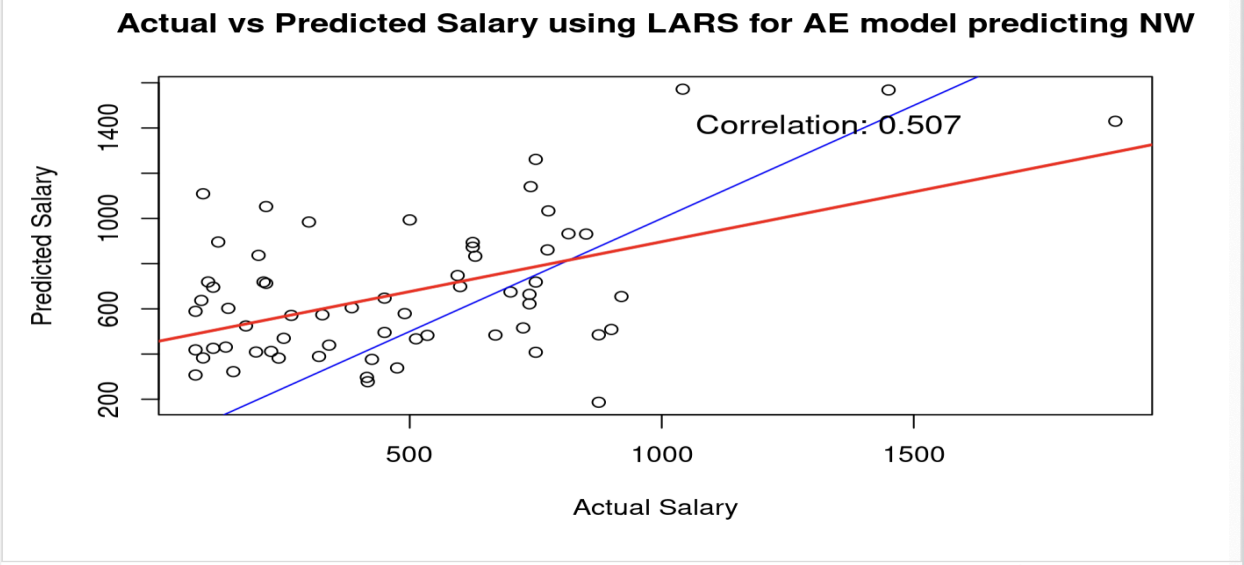



```r
# Predict for AW using NE
correlation_AW_NE <-
  predict_and_plot(HittersNEclean, betalarsNE, HittersAWclean, "NE", "AW")
```

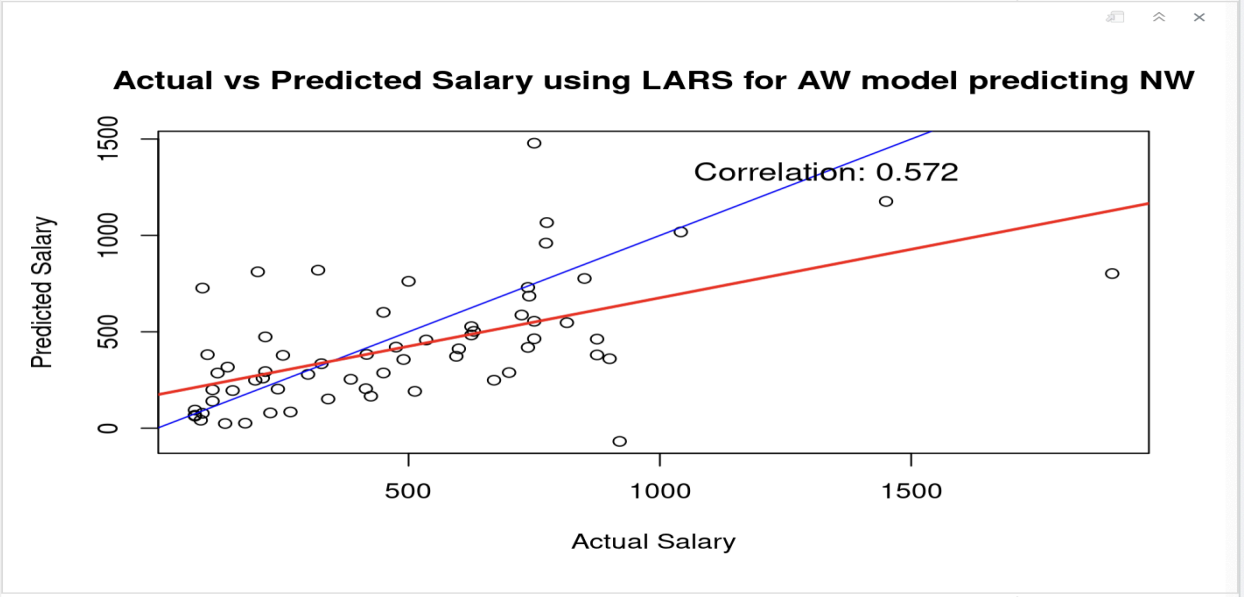


Predicting NW:

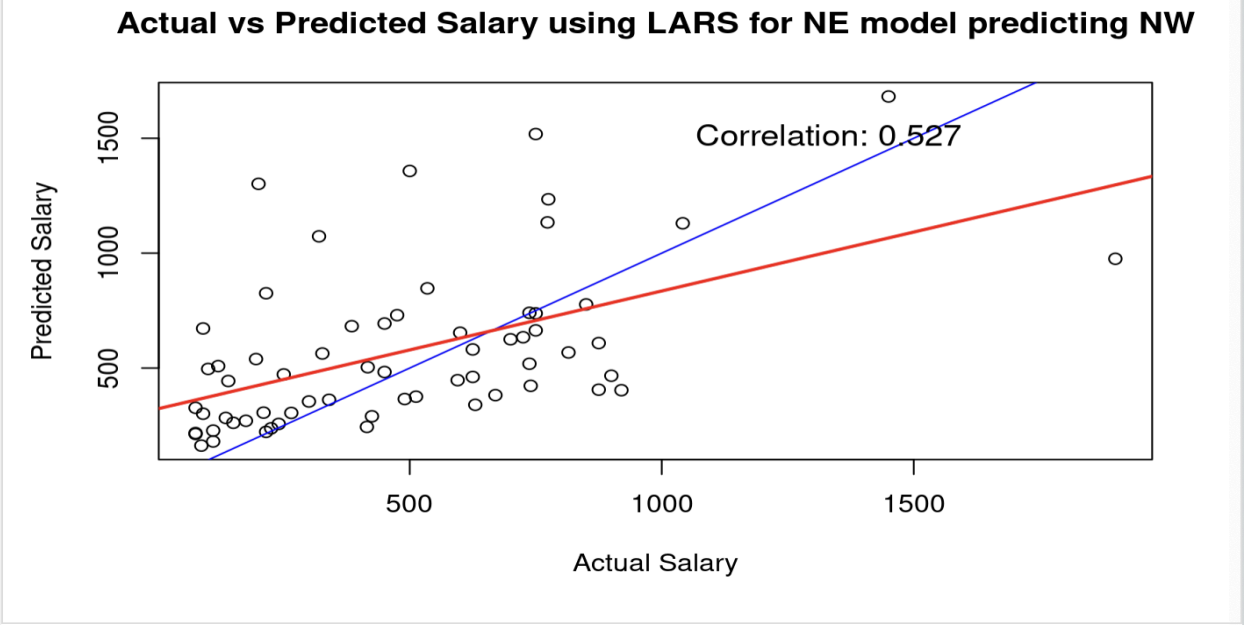
```r
# Red = Linear Regression Line
# Blue = Perfect Representation Line
# Predict for NW using AE model
correlation_NW_AE <-
  predict_and_plot(HittersAEclean, betalarsAE, HittersNWclean, "AE", "NW")
```



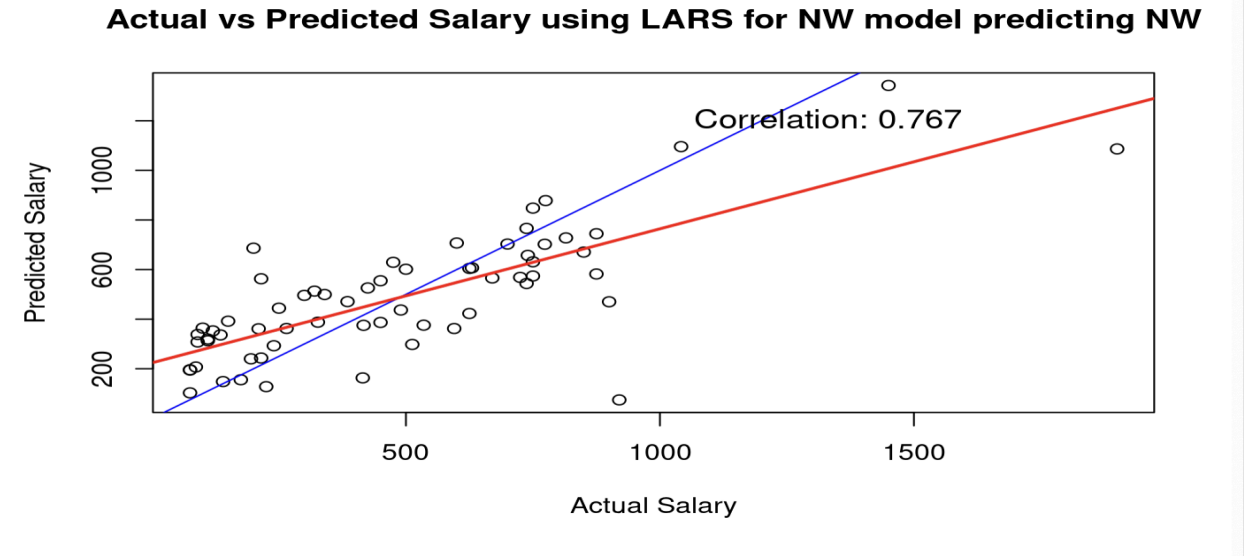

```r
# Predict for NW using AW
correlation_NW_AW <-
  predict_and_plot(HittersAWclean, betalarsAW, HittersNWclean, "AW", "NW")
```



```r
# Predict for NW using NE
correlation_NW_NE <-
  predict_and_plot(HittersNEclean, betalarsNE, HittersNWclean, "NE", "NW")
```

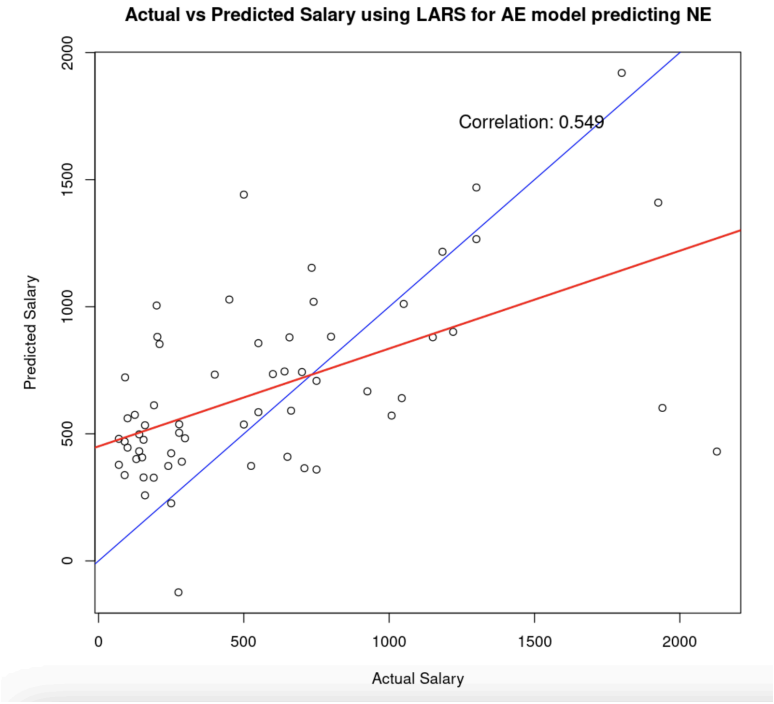


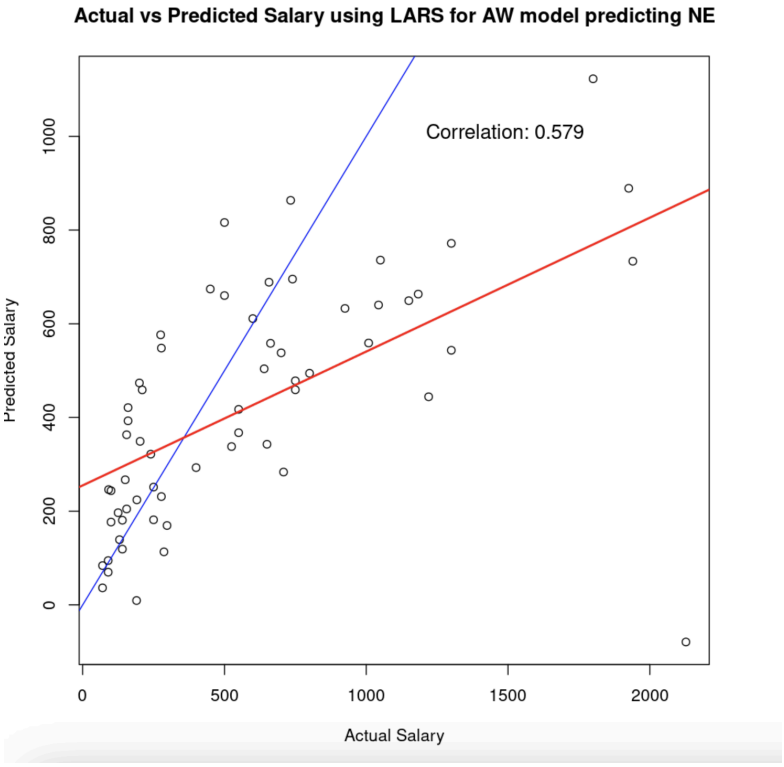

```r
# Predict for NW using NW
correlation_NW_NW <-
  predict_and_plot(HittersNWclean, betalarsNW, HittersNWclean, "NW", "NW")
```

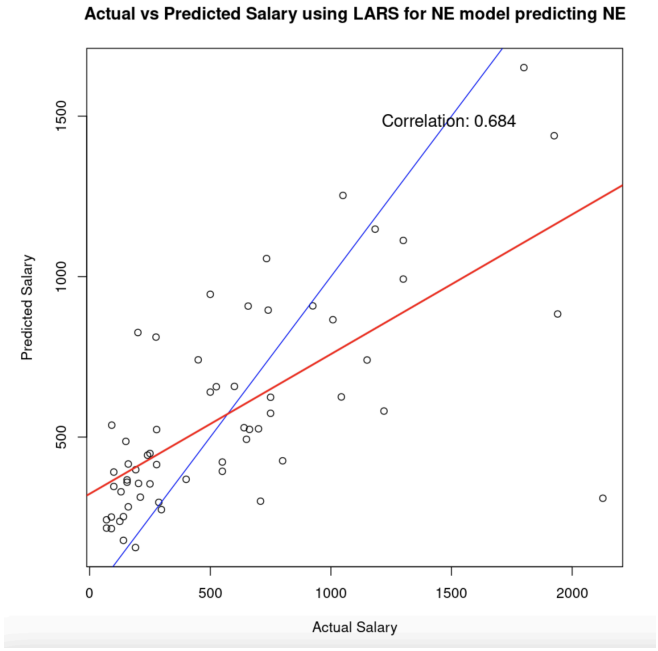


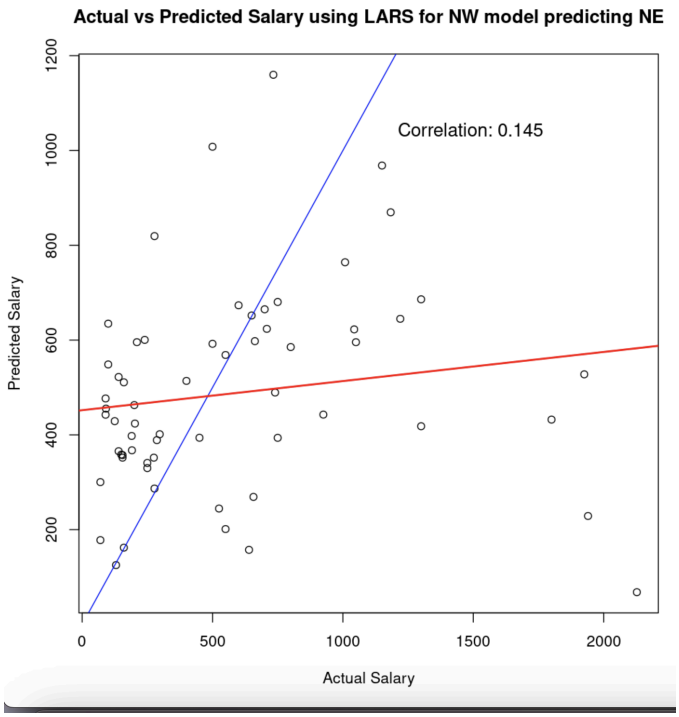
Predicting NE:

```
# Predict for NE using AE model
correlation_NE_AE <-
  predict_and_plot(HittersAEclean, betalarsAE, HittersNEclean, "AE", "NE")
# Predict for NE using AW
correlation_NE_AW <-
  predict_and_plot(HittersAWclean, betalarsAW, HittersNEclean, "AW", "NE")
# Predict for NE using NE
correlation_NE_NE <-
  predict_and_plot(HittersNEclean, betalarsNE, HittersNEclean, "NE", "NE")
# Predict for NE using NW
correlation_NE_NW <-
  predict_and_plot(HittersNWclean, betalarsNW, HittersNEclean, "NW", "NE")
```




Actual vs Predicted Salary using LARS for AW model predicting NE
Correlation: 0.579
Predicted Salary
Actual Salary
0
200
400
600
800
1000
0
500
1000
1500
2000


Actual vs Predicted Salary using LARS for NE model predicting NE
Correlation: 0.684
Predicted Salary
Actual Salary
500
1000
1500
0
500
1000
1500
2000

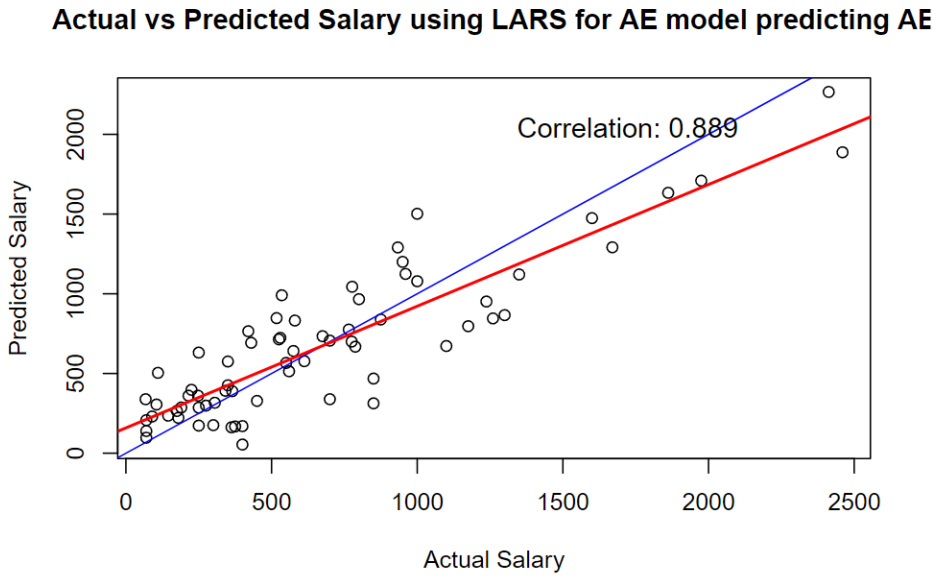


Predicting AE:

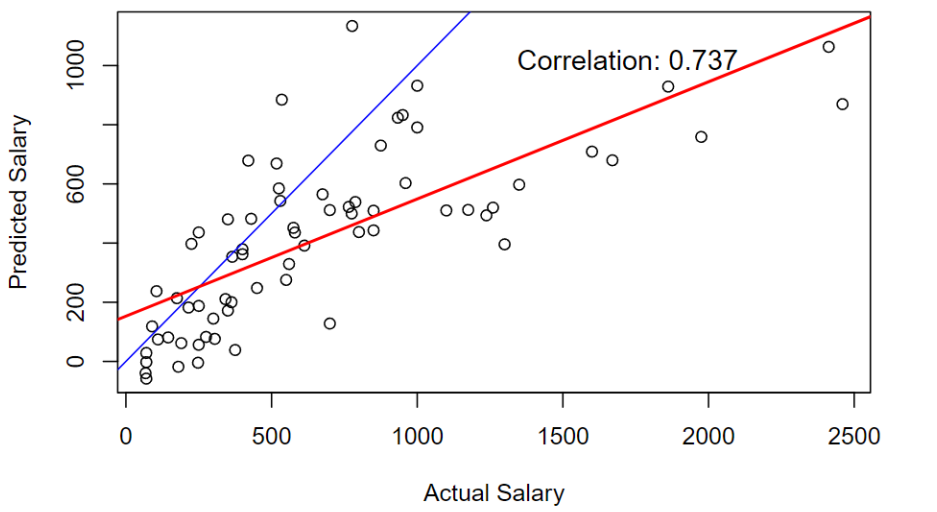
Red = Linear Regression Line

Blue = Perfect Representation Line

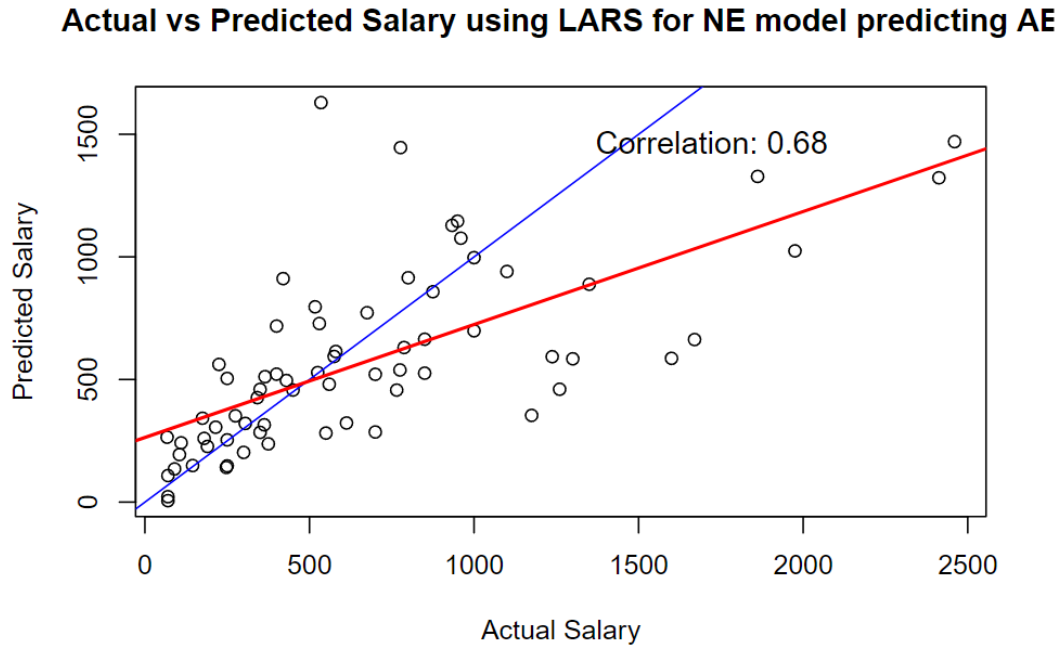
```
# Predict for AE using AE model
correlation_AE_AE <-
  predict_and_plot(HittersAEclean, betalarsAE, HittersAEclean, "AE", "AE")
```



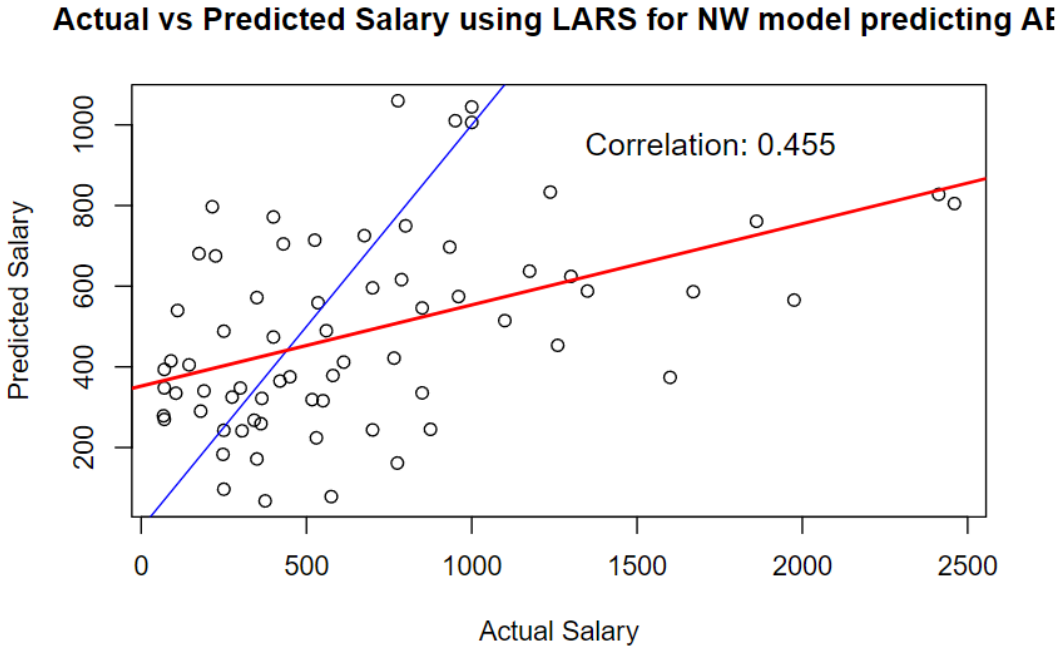

```
# Predict for AE using AW
correlation_AE_AW <-
  predict_and_plot(HittersAWclean, betalarsAW, HittersAEclean, "AW", "AE")
```



```r
# Predict for AE using NE
correlation_AE_NE <-
  predict_and_plot(HittersNEclean, betalarsNE, HittersAEclean, "NE", "AE")
```




```r
# Predict for AE using NW
correlation_AE_NW <-
  predict_and_plot(HittersNWclean, betalarsNW, HittersAEclean, "NW", "AE")
```



Random Forest

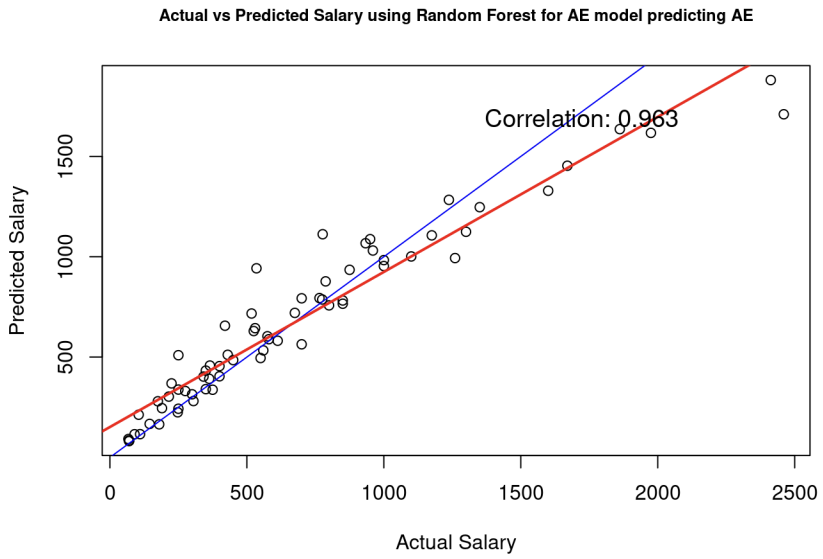
Predicting AE:

Predicting AE using other divisions and Random Forest

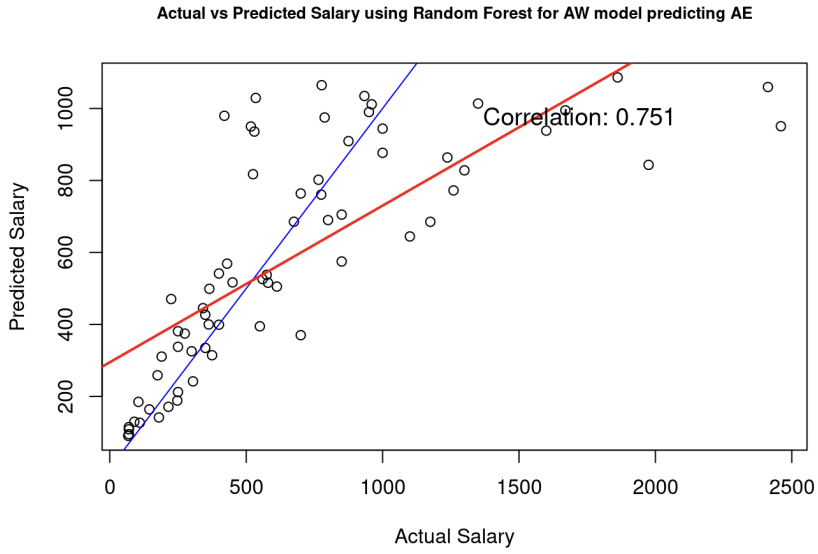
```
# function for predicting and plotting with Random Forest
predict_and_plot_rf <- function(train_data, rf_model, test_data, train_division, test_division) {
  pred <- predict(rf_model, test_data)

  plot(test_data$Salary, pred,
       main = paste("Actual vs Predicted Salary using Random Forest for",
                    train_division, "model predicting", test_division),
       xlab = "Actual Salary", ylab = "Predicted Salary",
       cex.main = 0.8)
  # reference line with an intercept of 0 and a slope of 1 to the plot that represents a perfect prediction where
predicted salary = actual salary
  abline(0, 1, col = "blue")

  # linear regression line
  lm_fit <- lm(pred ~ test_data$Salary)
  abline(lm_fit, col = "red", lwd = 2)

  # correlation
  correlation <- cor(pred, test_data$Salary)

  # print the correlation on the plot
  text(x = max(test_data$Salary) * 0.7, y = max(pred) * 0.9,
       labels = paste("Correlation:", round(correlation, 3)),
       col = "black", cex = 1.2)

  return(correlation)
}
```

```
# training the Random Forest models for each division
rfHittersAE <- randomForest(Salary ~ AtBat + Hits + HmRun + Runs + RBI + Walks + Years +
                              CAtBat + CHits + CHmRun + CRuns + CRBI + CWalks +
                              PutOuts + Assists + Errors,
                            data = HittersAEclean, ntree = 5000)

rfHittersAW <- randomForest(Salary ~ AtBat + Hits + HmRun + Runs + RBI + Walks + Years +
                              CAtBat + CHits + CHmRun + CRuns + CRBI + CWalks +
                              PutOuts + Assists + Errors,
                            data = HittersAWclean, ntree = 5000)

rfHittersNE <- randomForest(Salary ~ AtBat + Hits + HmRun + Runs + RBI + Walks + Years +
                              CAtBat + CHits + CHmRun + CRuns + CRBI + CWalks +
                              PutOuts + Assists + Errors,
                            data = HittersNEclean, ntree = 5000)

rfHittersNW <- randomForest(Salary ~ AtBat + Hits + HmRun + Runs + RBI + Walks + Years +
                              CAtBat + CHits + CHmRun + CRuns + CRBI + CWalks +
                              PutOuts + Assists + Errors,
                            data = HittersNWclean, ntree = 5000)
```

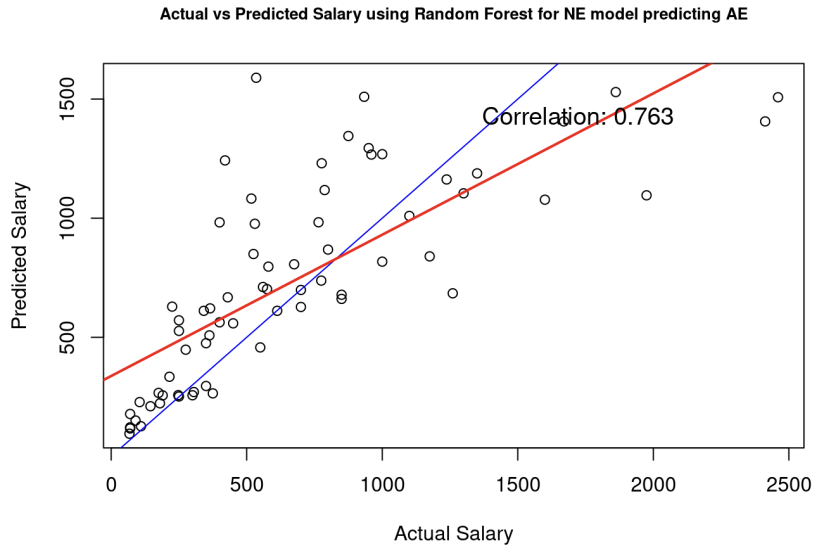
```
# Predict for AE using AE model
correlation_rf_AE_AE <- predict_and_plot_rf(HittersAEclean, rfHittersAE, HittersAEclean, "AE", "AE")
```



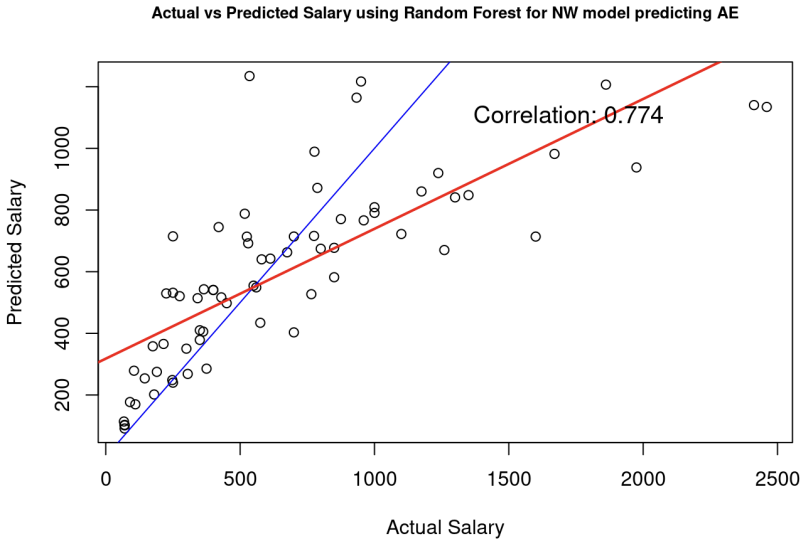
```
# Predict for AE using AW model
correlation_rf_AE_AW <- predict_and_plot_rf(HittersAWclean, rfHittersAW, HittersAEclean, "AW", "AE")
```




```
# Predict for AE using NE model
correlation_rf_AE_NE <- predict_and_plot_rf(HittersNEclean, rfHittersNE, HittersAEclean, "NE", "AE")
```

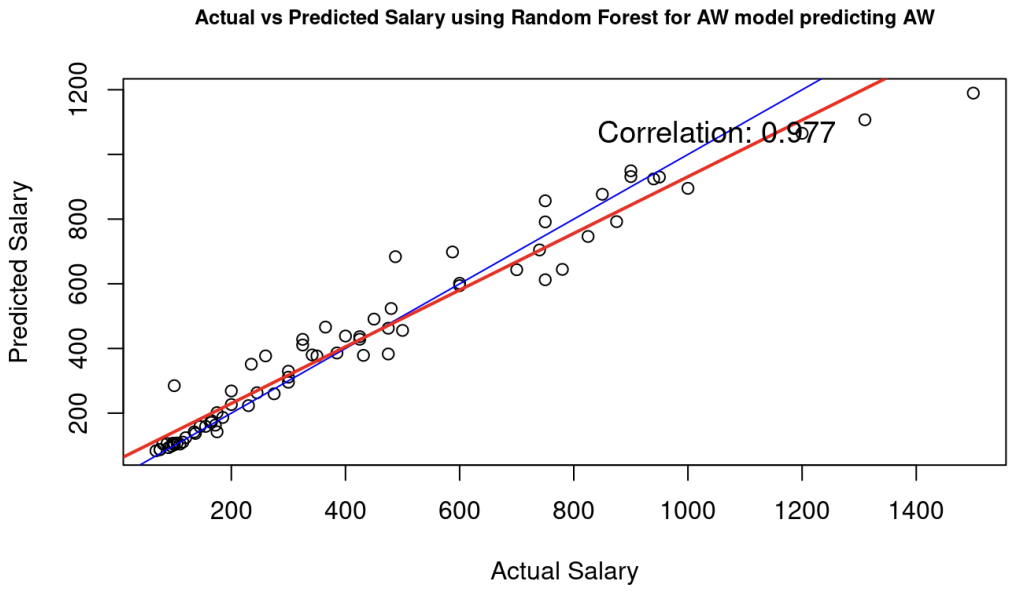


```r
# Predict for AE using NW model
correlation_rf_AE_NW <- predict_and_plot_rf(HittersNWclean, rfHittersNW, HittersAEclean, "NW", "AE")
```

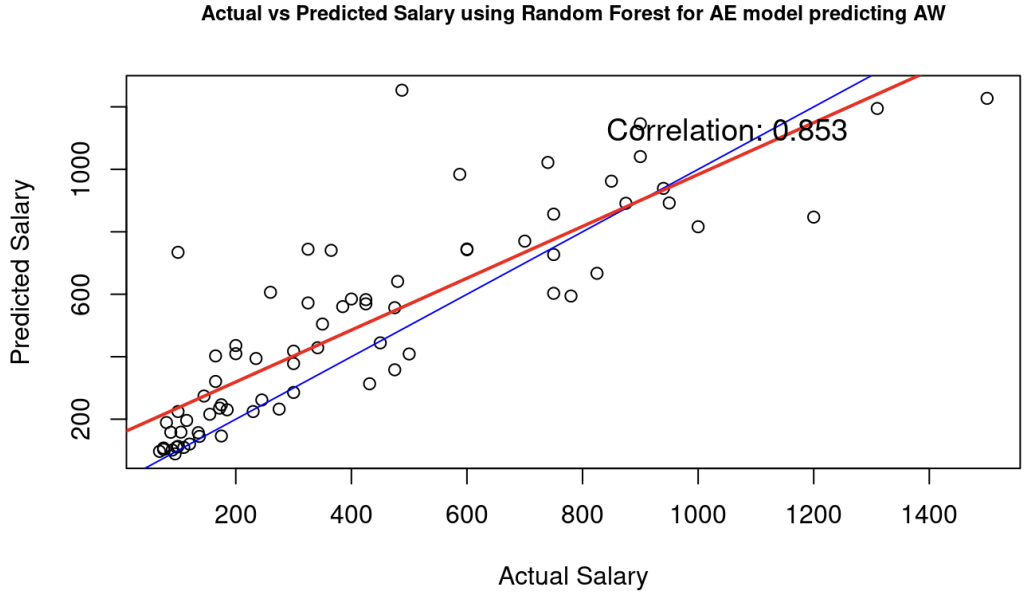


Predicting AW:

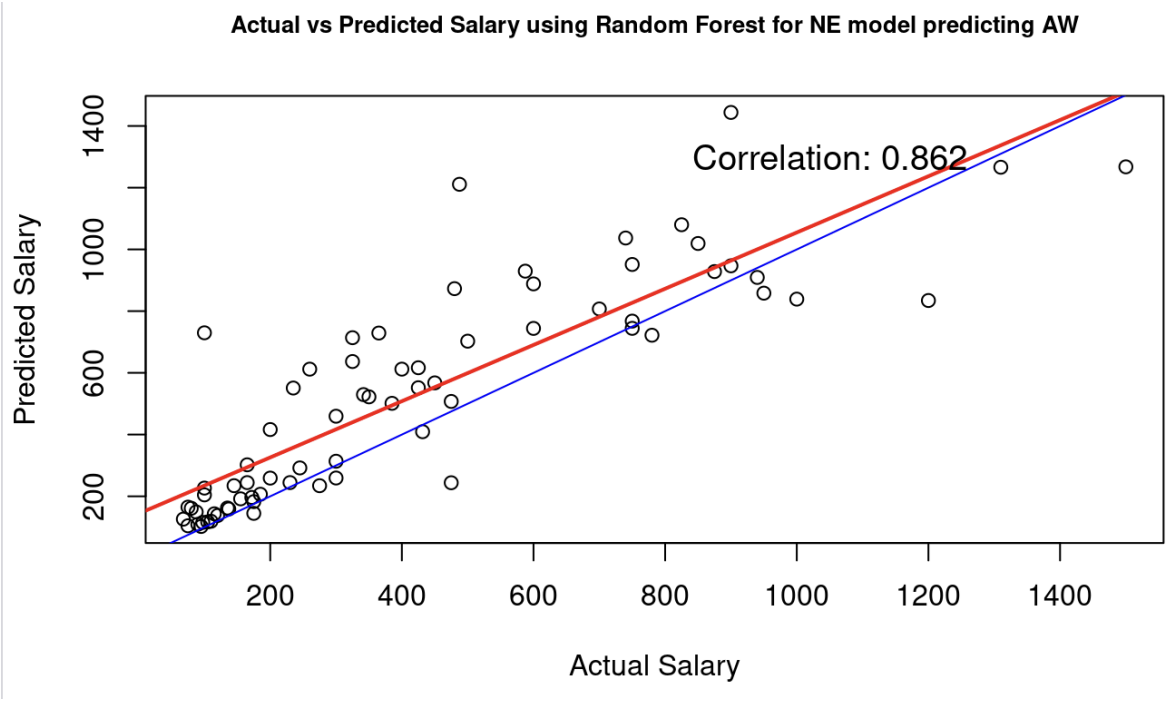
```
# Predict for AW using AW model
correlation_rfAW_AW <- predict_and_plot_rf(HittersAWclean, rfHittersAW, HittersAWclean, "AW", "AW")
```



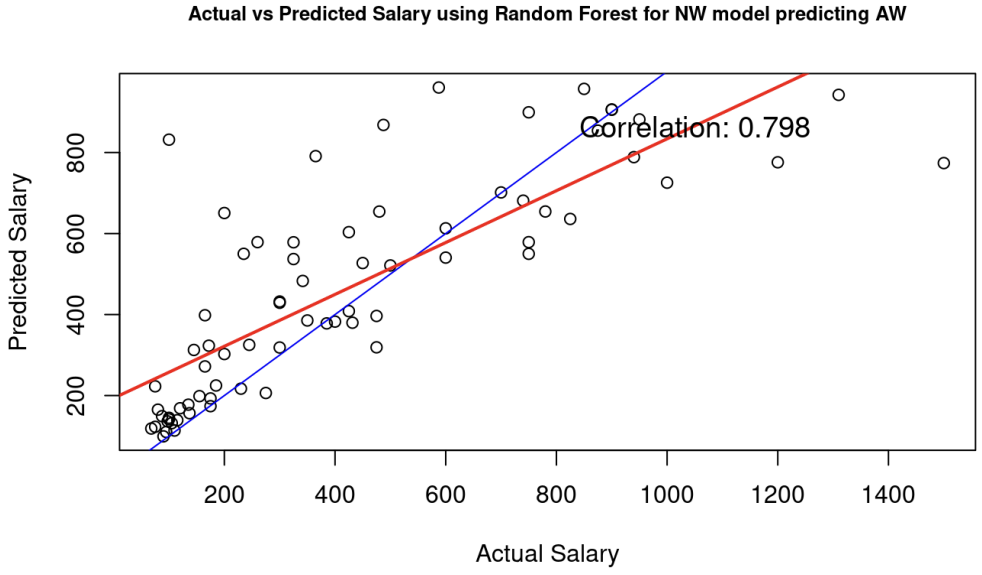

```
# Predict for AW using AE model
correlation_rf_AE_AW <- predict_and_plot_rf(HittersAEclean, rfHittersAE, HittersAWclean, "AE", "AW")
```



```r
# Predict for AW using NE model
correlation_rf_AE_NE <- predict_and_plot_rf(HittersNEclean, rfHittersNE, HittersAWclean, "NE", "AW")
```

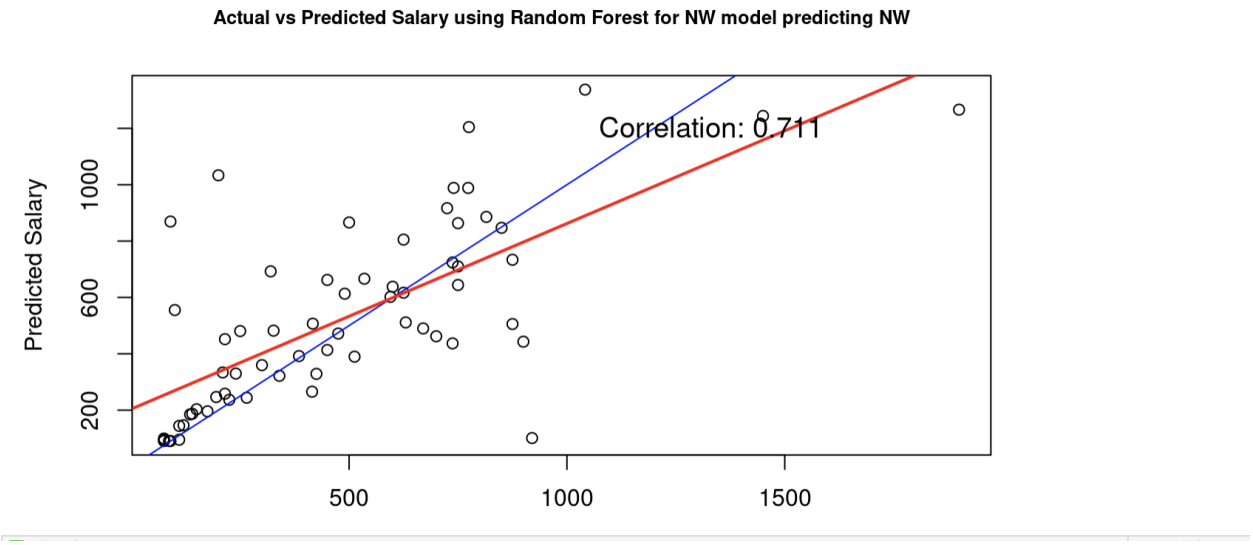



```r
# Predict for AW using NW model
correlation_rf_AE_NW <- predict_and_plot_rf(HittersNWclean, rfHittersNW, HittersAWclean, "NW", "AW")
```

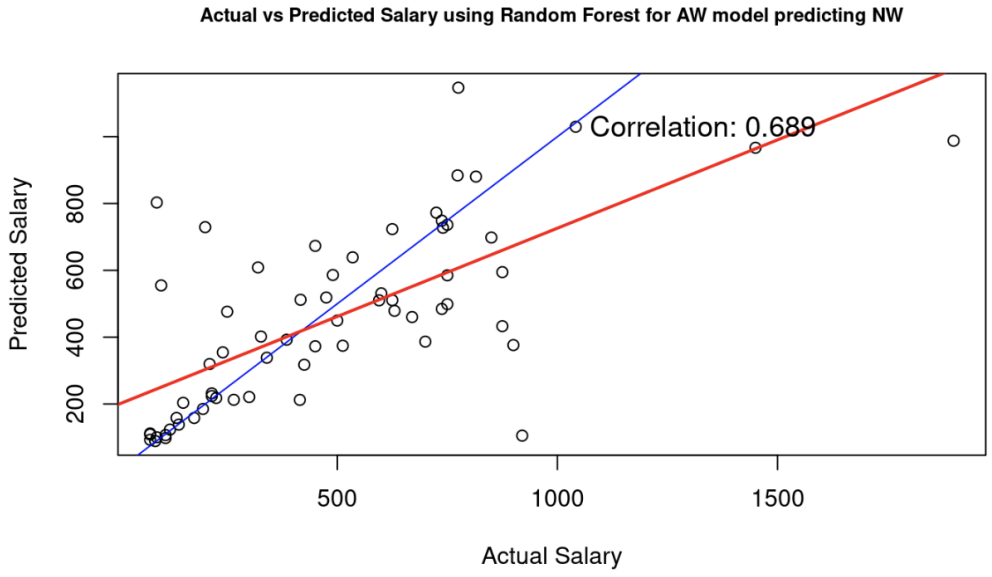


Predicting NW:

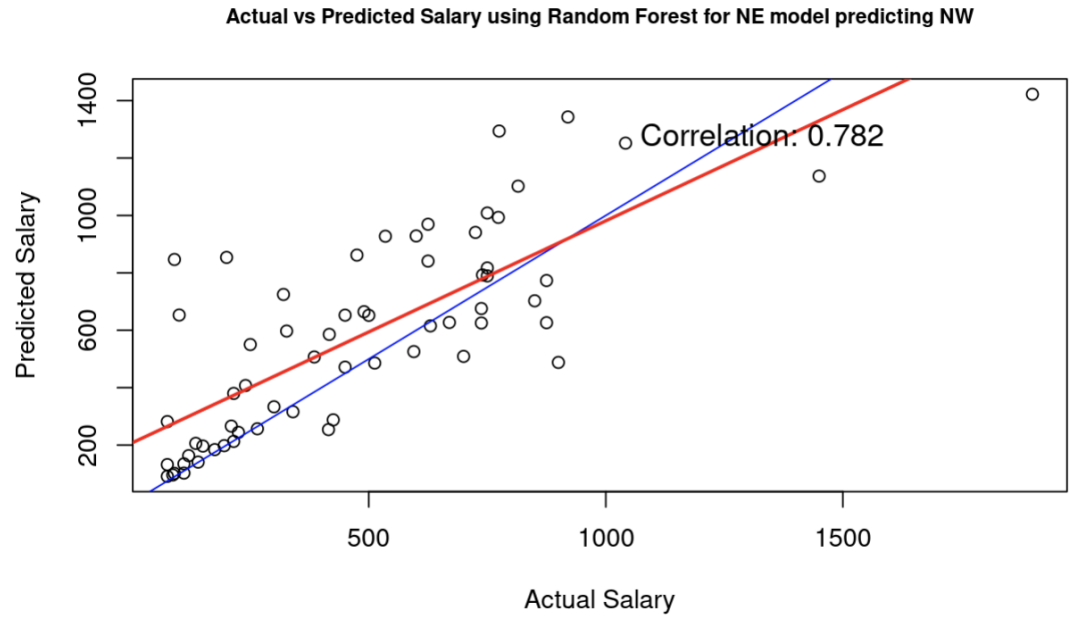
```{r}
# Predict for NW using AE model
correlation_rf_NW_AE <- predict_and_plot_rf(HittersAEclean, rfHittersAE, HittersNWclean, "NW", "NW")
```



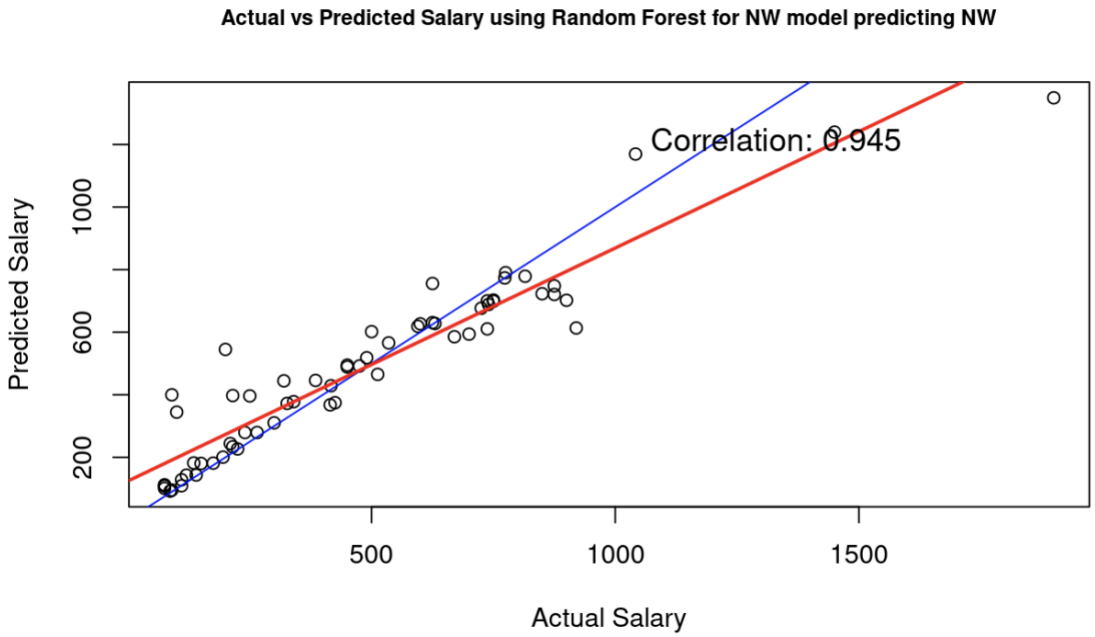
```
# Predict for NW using AW model
correlation_rf_AE_AW <- predict_and_plot_rf(HittersAWclean, rfHittersAW, HittersNWclean, "AW", "NW")

```




```
# Predict for NW using NE model
correlation_rf_NW_NE <- predict_and_plot_rf(HittersNEclean, rfHittersNE, HittersNWclean, "NE", "NW")
```

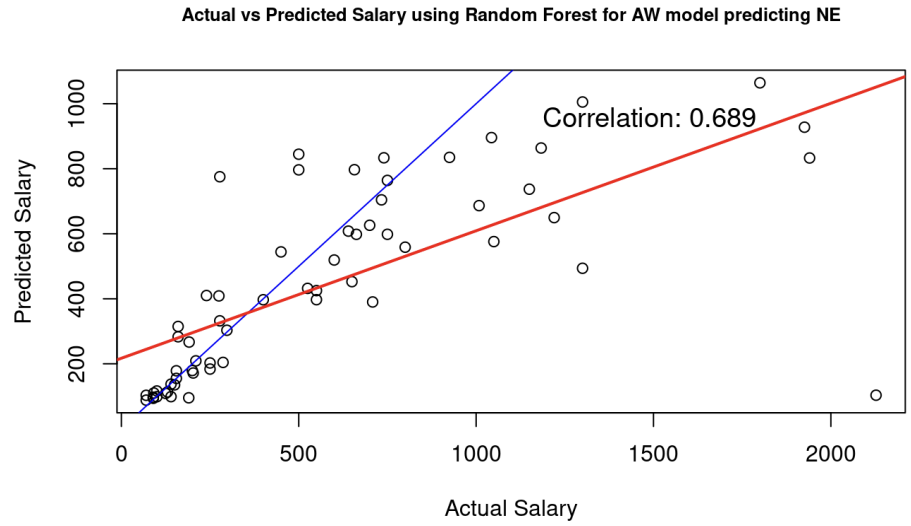


```
# Predict for NW using NW model
correlation_rf_NW_NW <- predict_and_plot_rf(HittersNWclean, rfHittersNW, HittersNWclean, "NW", "NW")
```

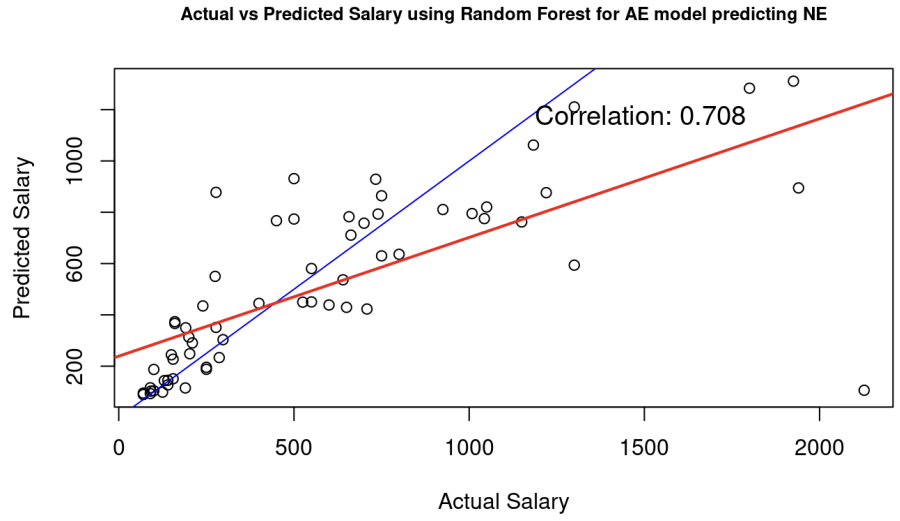


Predicting NE:

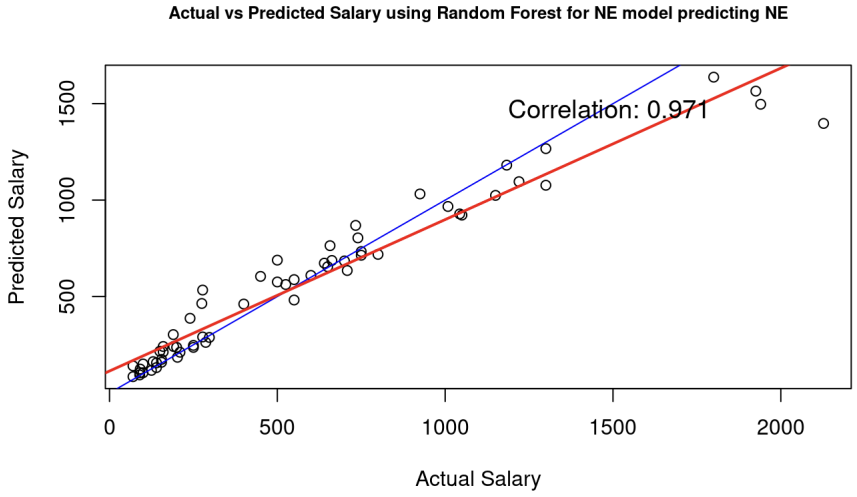
```{r}
# Predict for NE using AW model
correlation_rf_NE_AW <- predict_and_plot_rf(HittersAWclean, rfHittersAW, HittersNEclean, "AW",
"NE")
```



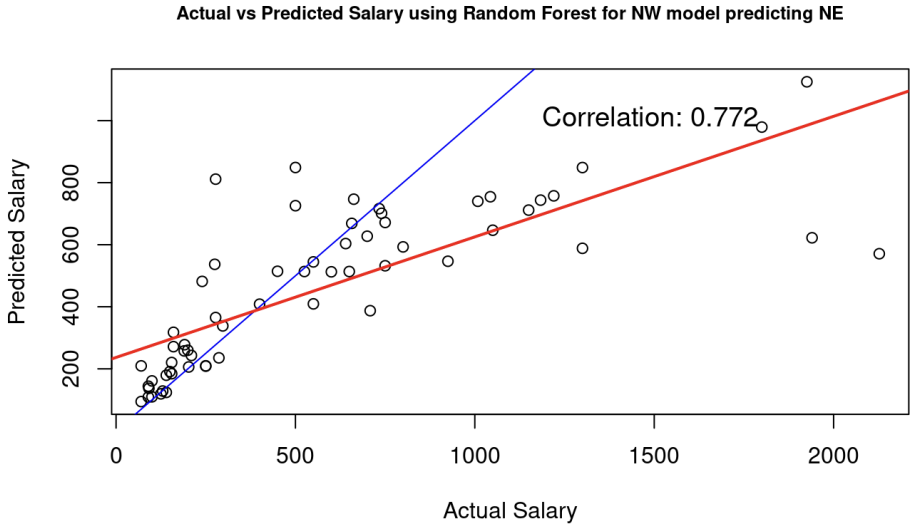

```{r}
# Predict for NE using AE model
correlation_rf_NE_AE <- predict_and_plot_rf(HittersAEclean, rfHittersAE, HittersNEclean, "AE",
"NE")
```



```r
# Predict for NE using NE model
correlation_rf_NE_NE <- predict_and_plot_rf(HittersNEclean, rfHittersNE, HittersNEclean, "NE",
"NE")
```




```r
# Predict for NE using NW model
correlation_rf_NE_NW <- predict_and_plot_rf(HittersNWclean, rfHittersNW, HittersNEclean, "NW",
"NE")
```



Analysis

Predicting AE Analysis:

- When predicting within the division (AE predicting AE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.889. The correlation for the random forest plot is 0.963.
- The random forest performs slightly better than the LARS when predicting across divisions (AW predicting AE). The correlation of the LARS model is 0.737. The correlation of the random forest model is 0.751.
- When predicting across divisions (NE predicting AE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.68. The correlation for the random forest plot is 0.763.
- When predicting across divisions (NW predicting AE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.455. The correlation for the random forest plot is 0.774.
- The red linear regression line in each random forest plot is significantly closer and similar to the blue perfect prediction line than in each corresponding LARS plot. This shows how random forest is superior in predicting salaries, as salaries have a lot of variability. LARS is a linear regression model that does not handle non-linearity like Random Forest. Also, Random Forest is non-parametric, meaning that it does not need a mean adjustment to correct for differences in scale or offset in the data. LARS is the far inferior model, as it would perform even worse without a mean adjustment, like we included. Random Forest combines multiple decision trees, reducing overfitting and outliers/noise. LARS is very

sensitive to outliers. This makes Random Forest better suited for salary prediction and other scenarios with intricate data patterns and variability.

Predicting AW Analysis:

- When predicting within the division (AW predicting AW), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.785. The correlation for the random forest plot is 0.977.
- The random forest performs slightly better than the LARS when predicting across divisions (AE predicting AW). The correlation of the LARS model is 0.597. The correlation of the random forest model is 0.858.
- When predicting across divisions (NE predicting AW), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.639. The correlation for the random forest plot is 0.862.
- When predicting across divisions (NW predicting AW), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.564. The correlation for the random forest plot is 0.798.

Overall:

The random forest model excels in making predictions both within individual divisions and across different divisions. Specifically, when predicting values within the AW division using AW data, the model performs better compared to cross-division predictions, achieving a notably high correlation of 0.977.

In the visual comparisons, the red linear regression line in each random forest plot aligns much more closely with the blue perfect prediction line than it does in the corresponding LARS plot.

This close alignment indicates that random forest provides superior predictive performance, especially for predicting salaries, which are known to exhibit substantial variability.

Predicting NE Analysis:

- When predicting within the division (AE predicting NE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.549. The correlation for the random forest plot is 0.708.
- The random forest performs slightly better than the LARS when predicting across divisions (AW predicting NE). The correlation of the LARS model is 0.549. The correlation of the random forest model is 0.689.
- When predicting across divisions (NE predicting NE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.684. The correlation for the random forest plot is 0.971.
- When predicting across divisions (NW predicting NE), the random forest model performs significantly better than LARS. The correlation of the LARS plot is 0.145. The correlation for the random forest plot is 0.772.

When predicting within the division (NE predicting NE), the model shows excellent performance with a correlation of 0.971. This is the strongest correlation among all models, which is expected since it uses data from the same division to predict itself.

- When using the AE (Atlantic East) model to predict NE salaries, the performance is moderate, with a correlation of 0.708. The scatter plot shows more dispersion around the prediction line, particularly at higher salary levels.

- Using the AW (Atlantic West) model to predict NE salaries yields slightly lower performance with a correlation of 0.689. The predictions show dispersion patterns similar to those of the AE model.
- The NW (Northwest) model predicting NE salaries shows relatively strong performance with a correlation of 0.772, which is better than both AE and AW models but not as good as the same-division prediction.

In all plots:

- The red line represents the linear regression fit of the predictions
- The blue line represents the ideal perfect prediction line (y=x)
- The closer these lines are to each other, the better the model's predictions

The analysis suggests that:

1. Same-division prediction (NE→NE) is significantly more accurate than cross-division predictions.
2. Among cross-division predictions, the NW model performs best for NE salary prediction.
3. AE and AW models show similar, moderate performance levels.
4. All models tend to have more prediction variance at higher salary levels, as shown by the increased scatter of points.
5. Random Forest is better.

Predicting NW Analysis:

When analyzing the predictions made within the division, specifically NW predicting NW, we can see that the Random Forest model does perform better than the LARS model. The LARS

model has a correlation value of 0.767 while the Random Forest model has a correlation of 0.945. This indicates that using Random Forest within the same divisions leads to a more accurate prediction, close to 1. The LARS model, while fairly accurate, doesn't perform as well compared to Random Forest

- Predicting across divisions
    - AE Predicting NW: When predicting across divisions in this scenario, Random Forest still performs better than LARS. The correlation value for the LARS model is 0.507, whereas the Random Forest model shows a correlation of 0.711. Random Power is more reliable compared to LARS in terms of predicting NW salaries using AE data
    - AW predicting NW: In this case, the Random Forest model also outperforms LARS with a correlation of 0.689 compared to LARS, which has a correlation of 0.572. Both models show they have a somewhat moderate predictive power, but Random Forest has a more reliable correlation compared to LARS.
    - NE predicting NW: In this case, the NE model shows a stronger performance when using Random Forest, with a correlation of 0.782. LARS, however, only achieves a correlation of 0.527, indicating that Random Forest is able to capture patterns in data across divisions. This allows it to consistently produce higher correlations than LARS.
- Further Analysis: When analyzing both the red and blue regression lines in each Random Forest plot, we can see that the red linear regression lines are closer to the blue perfect prediction lines in Random Forest compared to LARS plots. This indicates that Random Forest models perform better at predicting actual salary values compared to LARS. This

performance of the Random Forest models demonstrates Random Forest superiority in dealing with non-linearity and variability compared to LARS. Overall, Random Forest is non-parametric in nature which allows it to reduce overfitting and handle outliers more effectively, thus making it more suitable for salary prediction.

Conclusion Statement:

The data analysis makes it clear: Random Forest consistently outperforms LARS for salary predictions across all divisions. For instance, within the NE division, Random Forest achieved a 0.971 correlation compared to LARS's 0.684, and similarly, in NW, it scored 0.945 over LARS's 0.767. Even in cross-division cases—like NW predicting AE—Random Forest remained stronger, with a 0.774 correlation versus LARS's 0.455.

Random Forest's edge comes from handling non-linear data better, without needing mean adjustments, and its multiple decision trees make it more resilient to variability and outliers. Visual comparisons also support its reliability, with Random Forest's predictions closely aligning with the ideal trend line. In short, Random Forest proves itself as the more accurate, dependable choice for salary predictions.