Arnav Dashaputra - Hitters data in ISLR

Step 1: Look at all 4 divisions. For each division choose the minimal Cp lars model.

Minimal Cp refers to the smallest value of the Cp statistic, which is a measure of model quality that balances the trade-off between the model's complexity and its fit to the data. Lower Cp would then mean that a model is well-suited to the data, without the model being overly complex. In order to find the minimal Cp for each division, I created an R function that fits the LARS model and extracts the Cp models. The following R function fits the LARS model to the data parameter (the division), and then finds the lowest Cp value and returns it.

```r
find_minimal_cp_model <- function(data) {

  # fitting the LARS model
  division_lars <- lars(x = as.matrix(data[, !names(data) %in% c("Salary", "League", "Division", "NewLeague", "FullDivision")]),
                        y = data$Salary, type = "lasso", max.steps = 100)

  # extract Cp values and locate indices of minimal Cps
  min_cp_index <- which.min(division_lars$Cp)

  return(list(index = min_cp_index, Cp_value = division_lars$Cp[min_cp_index]))
}
```

The LARS model is used to predict players' salaries based on their performance metrics. To select these metrics, I used most of the numerical data that attributes to their skill and performance. The columns of data seen in the 'c' parameter of the matrix fitting refers to data that was non-predictive and/or non-numeric, and were omitted from the calculation. After the model was fitted, the Cp statistic was calculated for each step in the model's path. The output of this function for each of the divisions is as follows:

```
$AE
[1] "AE Division: Minimal Cp is at model index 19 with Cp value of 11.9077675415999"

$AW
[1] "AW Division: Minimal Cp is at model index 5 with Cp value of 1.98473765552932"

$NE
[1] "NE Division: Minimal Cp is at model index 9 with Cp value of 3.53281289503602"

$NW
[1] "NW Division: Minimal Cp is at model index 17 with Cp value of 13.8888585555234"
```

Step 2: Check with scatterplots the predictions using the minimal CP lars model from the chosen division against actual salaries for each division (4 scatterplots, construct 4 predictions based on the 1 model)

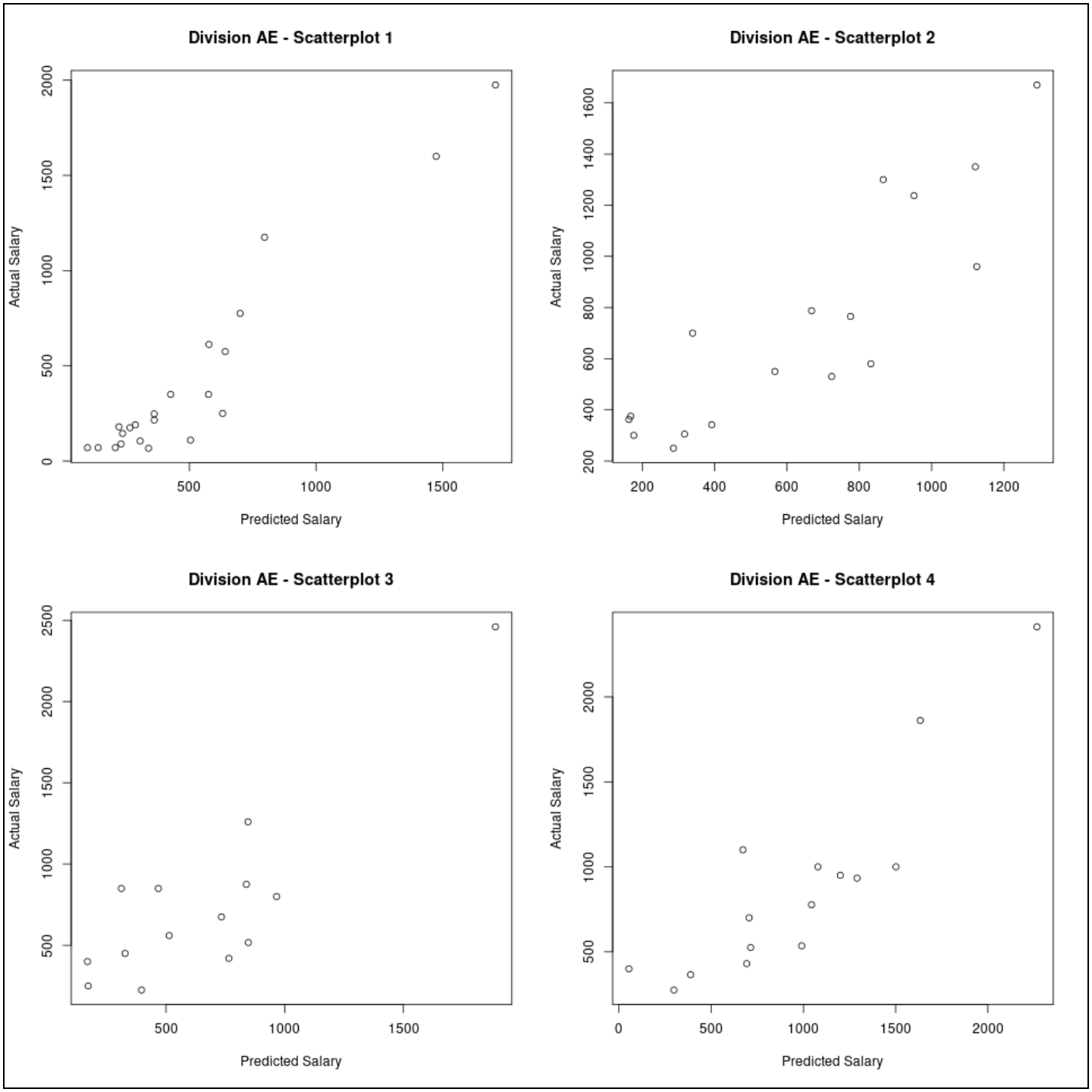
To clarify, for this step, I will be creating 4 scatterplots for each of the 4 divisions. Each scatterplot will be comparing the player salaries with the predicted salaries created by the optimal LARS model (the one with the minimal Cp value). This will illustrate how well the chosen model performs in different segments or conditions within each division.

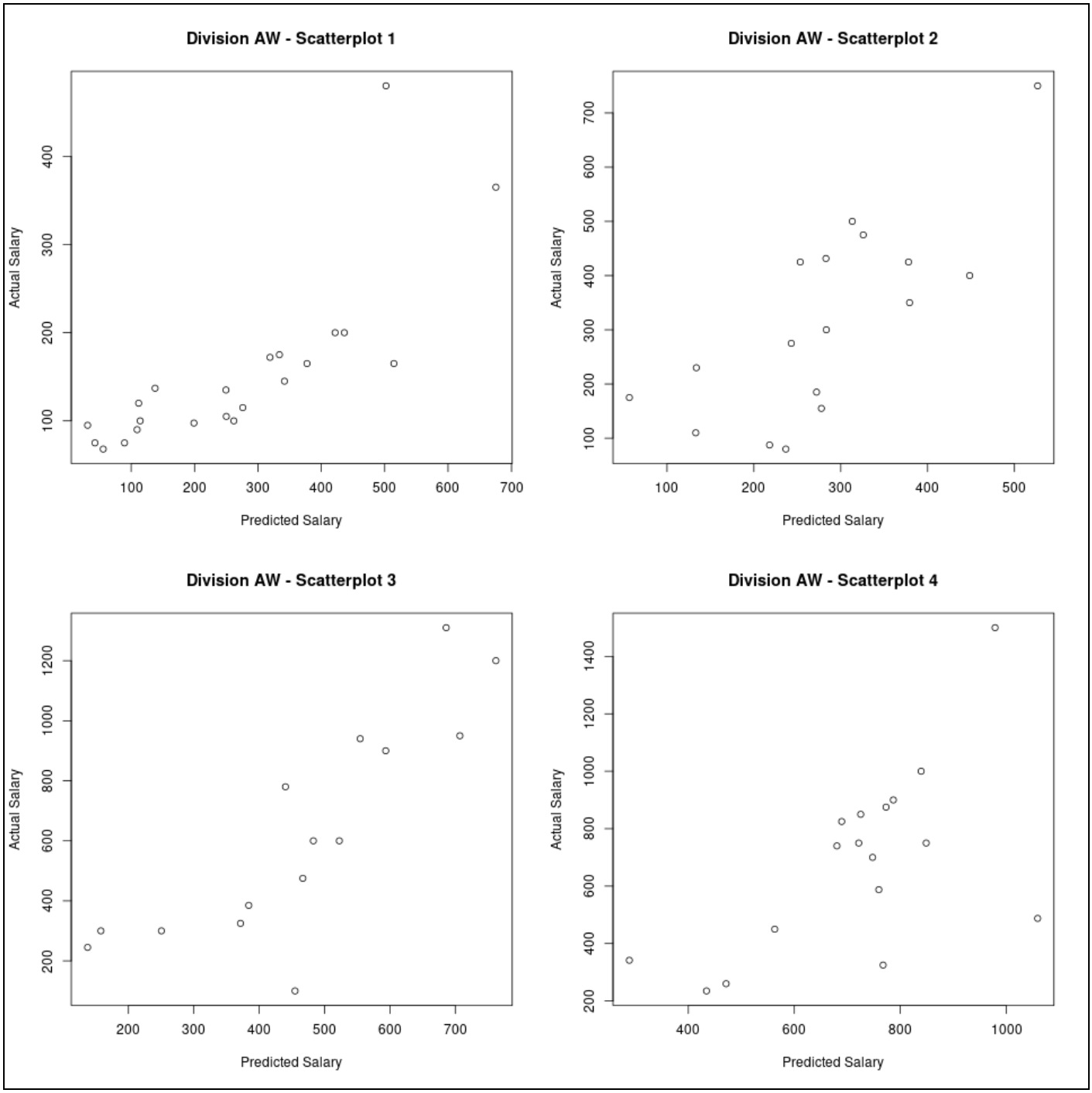
```
# Fit LARS model
division_lars <- lars(x = as.matrix(data[, !names(data) %in% c("Salary", "League", "Division", "NewLeague", "FullDivision")]),
                      y = data$Salary, type = "lasso")

# Predict salaries using the optimal LARS model
optimal_predictions <- predict(division_lars,
                               newx = as.matrix(data[, !names(data) %in% c("Salary", "League", "Division", "NewLeague", "FullDivision")]),
                               s = optimal_model_index)$fit
```

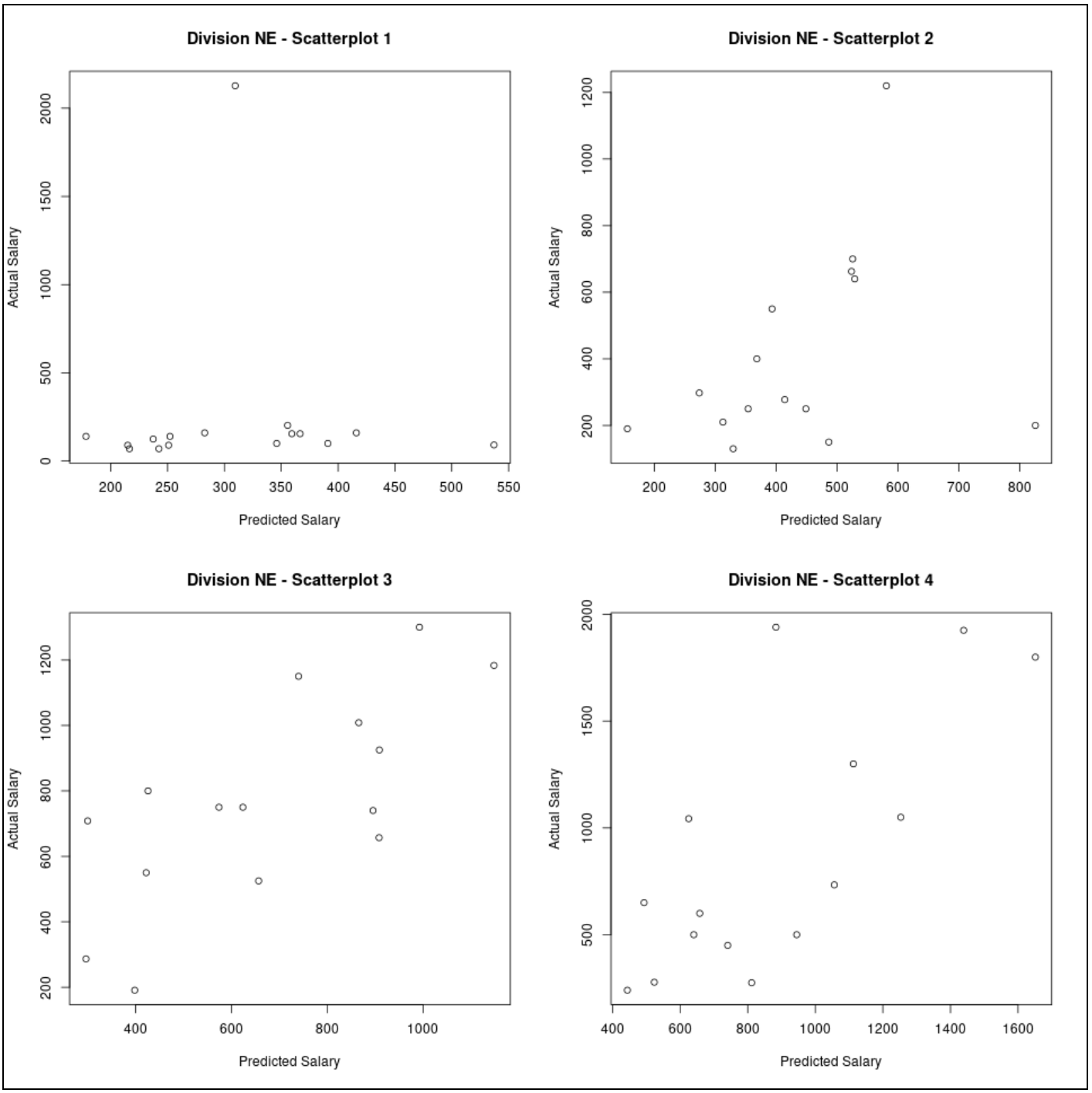
Subsets were determined by quartiles of time. The performance metrics used for the calculation (and used to determine the minimal Cp value) remained the same.

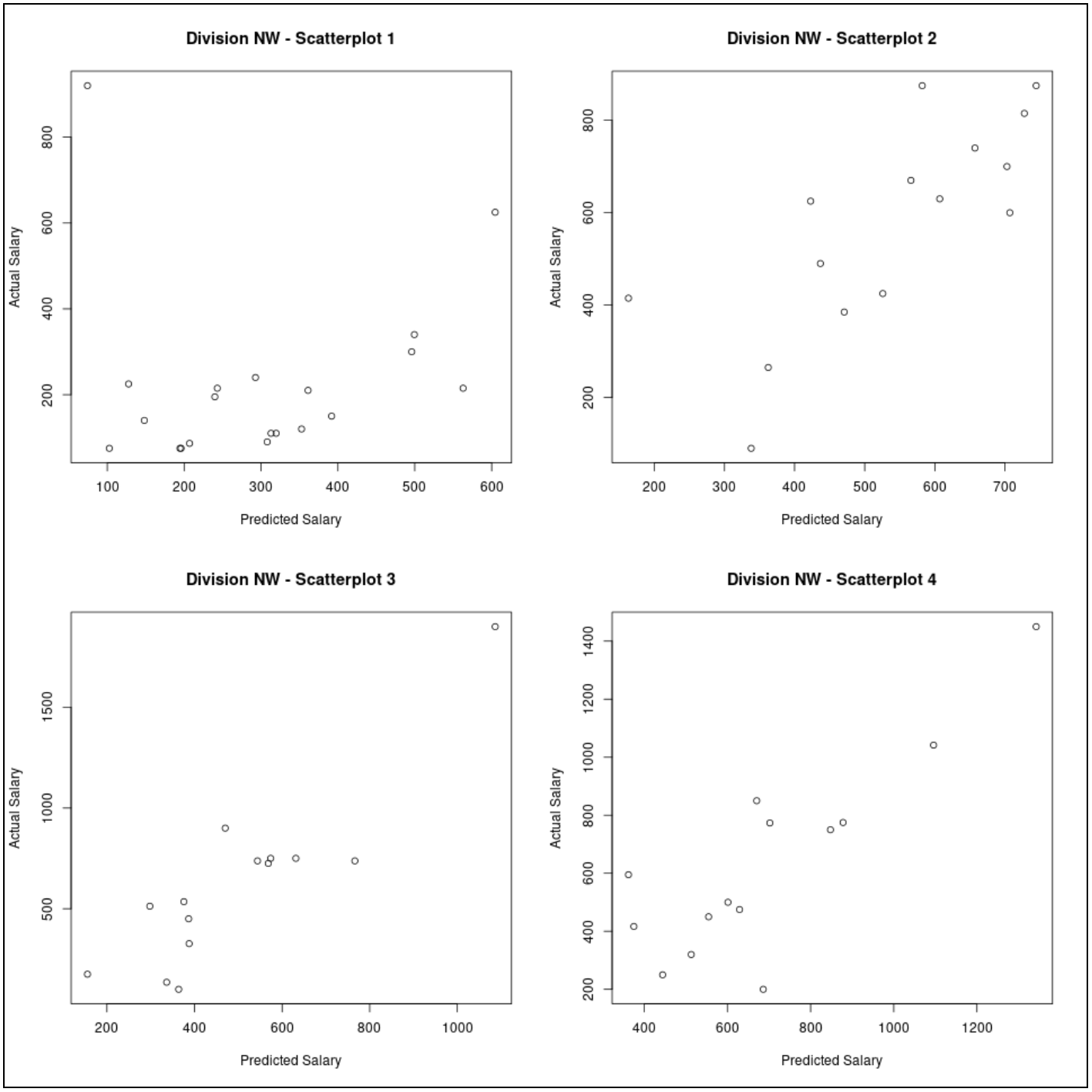
```
for (i in 1:4) {
  quartiles <- quantile(data$Years, probs = c(0.25, 0.5, 0.75))
  if (i == 1) {
    subset_indices <- which(data$Years <= quartiles[1])
  } else if (i == 2) {
    subset_indices <- which(data$Years > quartiles[1] & data$Years <= quartiles[2])
  } else if (i == 3) {
    subset_indices <- which(data$Years > quartiles[2] & data$Years <= quartiles[3])
  } else {
    subset_indices <- which(data$Years > quartiles[3])
  }
  subset_data <- data[subset_indices, ]
  subset_predictions <- optimal_predictions[subset_indices]
```

The plots were created with the predicted salary on the x-axis, and the actual salary on the y-axis. Additionally, the different minimal Cp values were used for each of the divisions. These 16 plots are shown below.


Division AE - Scatterplot 1
Actual Salary
Predicted Salary
0
500
1000
1500
2000
Division AE - Scatterplot 2
Actual Salary
Predicted Salary
200
400
600
800
1000
1200
1400
1600
Division AE - Scatterplot 3
Actual Salary
Predicted Salary
500
1000
1500
2000
2500
Division AE - Scatterplot 4
Actual Salary
Predicted Salary
0
500
1000
1500
2000


Division AW - Scatterplot 1
Actual Salary
Predicted Salary
100
200
300
400
500
600
700
Division AW - Scatterplot 2
Actual Salary
Predicted Salary
100
200
300
400
500
600
700
Division AW - Scatterplot 3
Actual Salary
Predicted Salary
200
400
600
800
1000
1200
Division AW - Scatterplot 4
Actual Salary
Predicted Salary
200
400
600
800
1000
1200
1400


Division NE - Scatterplot 1
Actual Salary
Predicted Salary
0
500
1000
1500
2000
200
250
300
350
400
450
500
550
Division NE - Scatterplot 2
Actual Salary
Predicted Salary
200
400
600
800
1000
1200
200
300
400
500
600
700
800
Division NE - Scatterplot 3
Actual Salary
Predicted Salary
200
400
600
800
1000
1200
400
600
800
1000
Division NE - Scatterplot 4
Actual Salary
Predicted Salary
500
1000
1500
2000
400
600
800
1000
1200
1400
1600



Step 3: Observe these plots to decide if it looks like there is a uniform process based on performance for choosing salary or not.

For this task, I have to examine all 16 scatterplots, and evaluate whether the salary determination appears to be consistent across divisions based on player performance. The difference in the plots shows that there is a definite variation in how salary correlates with performance metrics. The degrees of scatter in each of the plots shows

that while there are some common factors influencing salary across all divisions, the strength as well as consistency of these factors vary between divisions. Some divisions exhibit a tighter clustering of points around a line of best fit, which would mean that the salary determination follows a more uniform process based on the performance metrics that were selected. Others show a more dispersed scatter, meaning less consistency and less uniformity. Overall, this means that while performance may be a common factor in determining a player's salary, other factors that are specific to each division likely play a role, leading to the conclusion that the process is not entirely uniform across divisions.

Step 4: After building the LARS model 4 times (so 16 plots), compare the lars and lm model for the same variables predicting the salary for 1 division. Compare the prediction and the coefficients.

To better complete this task, I once again created a function that fits this time the lm model to a specific division.

```
# Similar to the LARS model, extract the numerical data only
numeric_predictors <- data[, sapply(data, is.numeric)]
numeric_predictors <- numeric_predictors[, !names(numeric_predictors) %in% "Salary"]

# Fit LARS model
lars_fit <- lars(x = as.matrix(numeric_predictors), y = data$Salary, type = "lasso")

# Find smallest Cp value
cp_index <- which.min(lars_fit$Cp)

# Extract the coefficients
lars_coefs <- coef(lars_fit)[cp_index, ]
selected_vars <- names(lars_coefs[lars_coefs != 0])

# Fit the lm model
lm_formula <- as.formula(paste("Salary ~", paste(selected_vars, collapse = " + ")))
lm_fit <- lm(lm_formula, data = data)
```

For this task, this division was the AE division.

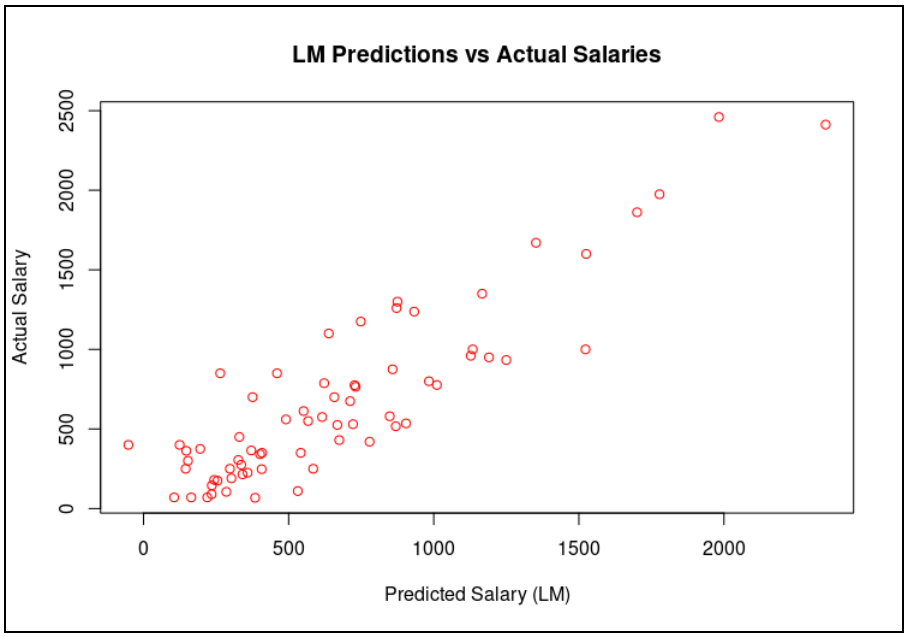
```
models_AE <- fit_and_compare_models(AE_data)
cp_values <- models_AE$lars_model$Cp
min_cp_index <- which.min(cp_values)
lars_coefs <- coef(models_AE$lars_model)[min_cp_index, ]
```

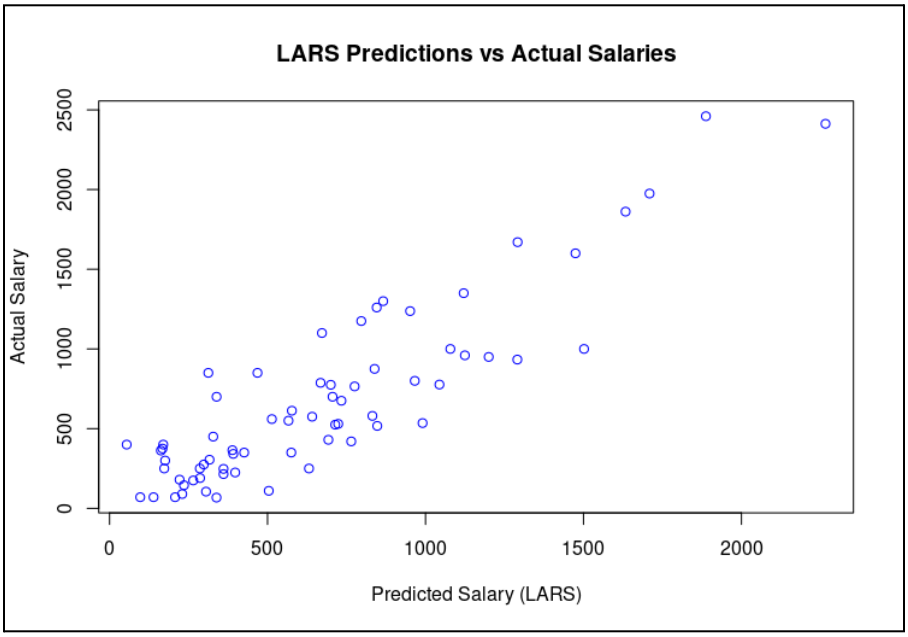
This was the result of running the function:

```
> print(lars_coefs)
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years     CAtBat      CHits     CHmRun      CRuns       CRBI     CWalks
-3.3530002 10.0477043 -5.1870815  1.9252299  0.0000000  6.1607285  0.0000000 -0.3771068  0.0000000  0.8418551  2.9480375  1.0145354 -1.1760056
   PutOuts    Assists     Errors
 0.3408382  0.1259118  0.0000000
> print(lm_coefs)
 (Intercept)        AtBat         Hits        HmRun         Runs        Walks       CAtBat       CHmRun        CRuns         CRBI       CWalks
333.91615317  -3.77014051  10.47129162  -4.36880711   1.89356499   6.28731152  -0.50437506  -0.08326206   3.66894235   1.52228576  -1.33810508
     PutOuts      Assists
  0.38200461   0.25849109
```

Lastly, for plotting:

```
lars_predictions_AE <- predict(models_AE$lars_model, newx = as.matrix(AE_data[, predictor_vars]), s = min_cp_index)$fit
lm_predictions_AE <- predict(models_AE$lm_model, newdata = AE_data)

plot(lars_predictions_AE, AE_data$Salary, main = "LARS Predictions vs Actual Salaries", xlab = "Predicted Salary (LARS)", ylab = "Actual Sal
plot(lm_predictions_AE, AE_data$Salary, main = "LM Predictions vs Actual Salaries", xlab = "Predicted Salary (LM)", ylab = "Actual Salary",
```





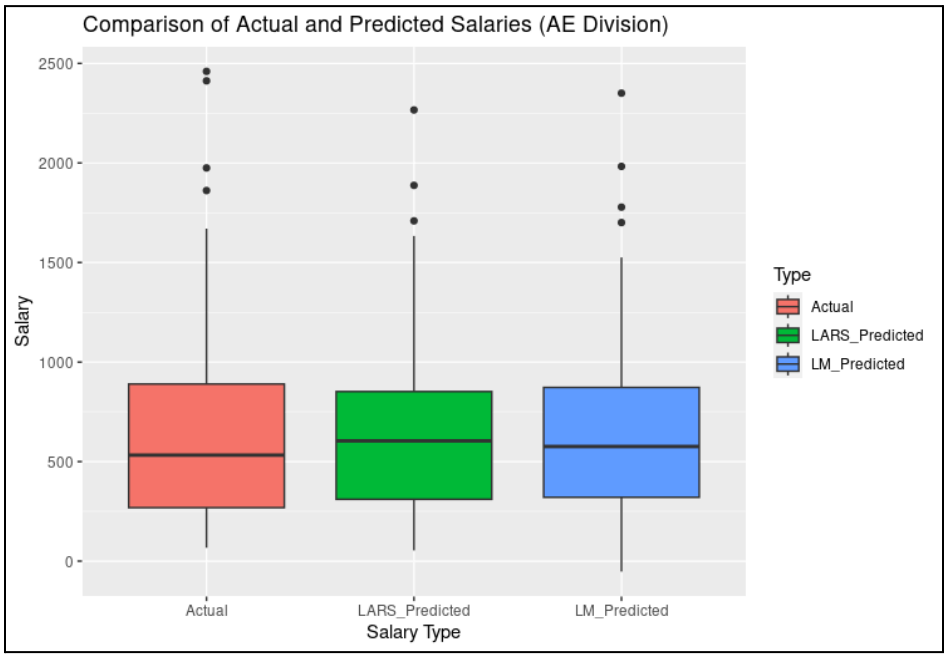
In the LARS model, a few variables have been shrunk towards zero, which shows the regularization effect that LARS uses to reduce overfitting and complexity. For instance, Years, CHits, and Errors were excluded (coefficients are zero), suggesting that they

may not be as significant when also considering other variables. Hits and Walks are positive predictors in both models, showing that there is a consistent positive relationship with a player's salary. The lm model, on the other hand, does not have the regularization that LARS does, and is forced to assign a non-zero coefficient to all variables. This means that there could be more complexity but there is also the risk of data overfitting. The differences in coefficients between the models, such as for the HmRun and CAtBat fields, shows how the variable selection used by LARS can create large differences when comparing against the collinearity (more than two associated predictor variables) in the lm model.

Bonus: Extra credit for boxplots with relevant discussion.

The boxplot that I decided to create highlights the distribution of the actual player salaries next to the player salaries predicted by the LARS and LM models for the AE division.

```
# Combine actual and predicted salaries
combined_data <- data.frame(
  Actual = AE_data$Salary,
  LARS_Predicted = lars_predictions_AE,
  LM_Predicted = lm_predictions_AE
)

reshape_data <- melt(combined_data, variable.name = "Type", value.name = "Salary") # reshape

# Boxplots
ggplot(reshape_data, aes(x = Type, y = Salary, fill = Type)) +
  geom_boxplot() +
  ggtitle("Comparison of Actual and Predicted Salaries (AE Division)") +
  xlab("Salary Type") +
  ylab("Salary")
```



As seen in the plot, the actual salaries have a wider range and greater variability, seen in the slightly longer box and box whiskers. This means that the actual salaries might be influenced by factors that are not fully captured by the two models.

The predicted salaries from both the LARS and LM models have smaller distributions, with the median of the LM player salaries being slightly higher than the LARS player salaries. This could mean there is a difference in how these models calculate the salary, which was discussed in the previous section when comparing scatter plots.

As for outliers, both models do have outliers, but the LARS model has fewer extreme values than the LM model. This could mean that both models capture the distribution of the players' salaries to some extent, but there are small differences in how the models handle extremities and spread. The LM model's predictions are closer to the upper range of salaries, which could be because there is less constraint on the coefficient estimates. The LARS model uses regularization to create more moderate predictions, as it decreases coefficients, providing a more balanced fit and prevents overfitting.

Some additional graphs that I came up with while working on the project: