AD 699 Assignment 2

Hiten Ladkani

2025-03-01

Simple Linear Regression

1. Bring player_dataset.csv into your R environment.
2. Use either the str() function or the glimpse() function from dplyr to learn more about this dataset.

a. After taking a look at the dataset description and seeing the results here, which variables in the dataset should be seen as numeric, and which should be seen as categorical?
b. Among the numeric variables, which ones are continuous, and which ones are discrete?

```
nhl_df <- read.csv("nhl_players.csv")
head(nhl_df)
```

```
##                name   Team_y Position_y HANDED GP  G  A  P  Sh Sh_perc
## 1 Zemgus Girgensons      BUF          C   Left 68 12  7 19  85    0.14
## 2        Zack Smith      CHI          C   Left 50  4  7 11  43    0.09
## 3      Zack Kassian      EDM          W  Right 59 15 19 34  99    0.15
## 4       Zach Parise      MIN          W   Left 69 25 21 46 155    0.16
## 5        Zach Hyman      TOR          W  Right 51 21 16 37 106    0.20
## 6     Zach Bogosian BUF, T.B          D  Right 27  1  6  7  29    0.03
##       SALARY PIM Giveaways Takeaways Hits Hits.Taken blocked_shots PlusMinus
## 1 $1,600,000  10        11        13  110         71            20        -1
## 2 $3,250,000  29        14        21  112         71            18         2
## 3 $2,000,000  69        45        26  157         54             8         0
## 4 $9,000,000   8        22        21   27         60            38       -11
## 5 $2,500,000  23        16        32   52        101            23        13
## 6 $6,000,000  22        11         4   30         21            27         0
```

```
str(nhl_df)
```

```
## 'data.frame':    568 obs. of  18 variables:
##  $ name         : chr  "Zemgus Girgensons" "Zack Smith" "Zack Kassian" "Zach Parise" ...
##  $ Team_y       : chr  "BUF" "CHI" "EDM" "MIN" ...
##  $ Position_y   : chr  "C" "C" "W" "W" ...
##  $ HANDED       : chr  "Left" "Left" "Right" "Left" ...
##  $ GP           : int  68 50 59 69 51 27 27 57 70 68 ...
##  $ G            : int  12 4 15 25 21 1 1 6 10 31 ...
##  $ A            : int  7 7 19 21 16 6 6 7 20 28 ...
##  $ P            : int  19 11 34 46 37 7 7 13 30 59 ...
##  $ Sh           : int  85 43 99 155 106 29 29 98 110 197 ...
##  $ Sh_perc      : num  0.14 0.09 0.15 0.16 0.2 0.03 0.03 0.06 0.09 0.16 ...
##  $ SALARY       : chr  "$1,600,000" "$3,250,000" "$2,000,000" "$9,000,000" ...
##  $ PIM          : int  10 29 69 8 23 22 22 28 49 12 ...
##  $ Giveaways    : int  11 14 45 22 16 11 11 21 14 41 ...
##  $ Takeaways    : int  13 21 26 21 32 4 4 20 48 42 ...
##  $ Hits         : int  110 112 157 27 52 30 30 136 78 9 ...
##  $ Hits.Taken   : int  71 71 54 60 101 21 21 99 114 66 ...
##  $ blocked_shots: int  20 18 8 38 23 27 27 30 20 14 ...
##  $ PlusMinus    : int  -1 2 0 -11 13 0 0 6 -5 -2 ...
```

Answer: a. Categorical variables - “name”, “Team_y”, “Position_y”, “HANDED” Numeric variables - “GP”, “G”, “A”, “Sh”, “Sh_perc”, “SALARY”, “PIM”, “Giveaways”, “Takeaways”, “Hits”, “Hits.Taken”, “blocked_shots” and “plusMinus”.

Answer: b. Continuous numeric variables - “sh_perc” Discrete numeric variables - “GP”, “G”, “A”, “P”, “Sh”, “SALARY”,“PIM”, “Giveaways”,“Takeaways”, “Hits”, “Hits.Taken”, “blocked_shots” and “plusMinus”.

3. Check for NAs. Are there any NA values in this dataset? How do you know this? Remove all rows that contain any NA values now.

```
sum(is.na(nhl_df))
```

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

Answer: There are no NA values in the dataset so we need not remove any rows.

4. Variable name clean-up. We’ve got some variable names that are a bit annoying here, thanks to some extra characters. Clean up any variable names that would be better without the extra characters.

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
nhl_df <- nhl_df %>%
          rename(Team = Team_y, Position = Position_y, Handed = HANDED,
                 Goals = G, Assists = A, Gamesplayed = GP, Shots = Sh,
                 Salary = SALARY, HitsTaken = Hits.Taken, blockedshots =
                 blocked_shots)
```

Cleaned up a few other variables to ensure consistency in the naming taxonomy, so it is easier to execute column wise functions

5. Check for duplicates. Are there any duplicate player names here? If so, remove one of the rows associated with this duplication. How many total rows of data remain now?

```
table(nhl_df$name)[table(nhl_df$name) > 1]
```

```
##
##     Andrej Sekera       Corey Perry Kevin Shattenkirk        Paul Byron
##                 2                 2                 2                 2
##      Troy Brouwer Valeri Nichushkin     Zach Bogosian
##                 2                 2                 2
```

```
nhl_df <- nhl_df %>%
          distinct(name, .keep_all = TRUE)
table(nhl_df$name)[table(nhl_df$name) > 1]
```

```
## named integer(0)
```

Answer: The "nhl_df" dataframe was reduced to 561 records from 568 records after removing duplicate records of few players.

6. Handling the "Salary" variable. Currently, the Salary variable contains $ symbols in each cell, as well as commas. Clean this variable up so that the commas and $ symbols are gone.

```
nhl_df$Salary <- as.numeric(gsub("[$,]", "", nhl_df$Salary))
str(nhl_df$Salary)
```

```
##  num [1:561] 1600000 3250000 2000000 9000000 2500000 6000000 1000000 6300000 9000000 5900000
...
```

Using a seed value based on either your street address or a lucky number of your choice, create a data partition. Assign approximately 60% of the records to your training set, and the other 40% to your validation set. (For instance, someone who lives at 201 Canal St would use 201 as a seed value, but a person living at 880 Washington Street would use 880. If your lucky number is 88, use it as your seed value – if your lucky number is 1234, use that instead).Keep in mind that a seed value has no relationship to the data itself – it's just an arbitrary number. You can use any method that results in 60% of rows going to training, and 40% to validation, with no overlapping rows, no rows thrown away, and random selection.

```
set.seed(317)
train_index <- sample(1:nrow(nhl_df), size = 0.60 * nrow(nhl_df))
#split data into 60% training and 40% validation set
train_data <- nhl_df[train_index, ]
validation_data <- nhl_df[-train_index, ]
```

a. Why is it important to partition the data before doing any sort of in-depth analysis of the variables? [note: this question is *not* asking why we do a data partition, but is instead asking why we do the partition prior to analyzing the full dataset]

Answer: If we perform predictive analysis before partitioning the data, we introduce an "optimism" bias. This is because when we choose the model that works best with the entire dataset, the bias can come from two sources: 1. the model itself is superior in predictive analysis. 2. There is a chance that the data perfectly matches with the chosen model better than other models. In the latter case, the problem of overfitting comes into picture and we would realise that the model has poor performance to new data (test data) on which it is tested.

8. Let's imagine that we wish to know more about the relationship between a player's total points and his salary. Using ggplot, create a scatterplot that depicts Salary on the y-axis and Points on the x-axis. Add a best-fit line to this scatterplot. Use only your training set data to build this plot. Be sure that your graph axes do not show numbers in scientific notation. What does this plot suggest about the relationship between these variables? Does this make intuitive sense to you? Why or why not?

```
library(ggplot2)
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.4.2
```

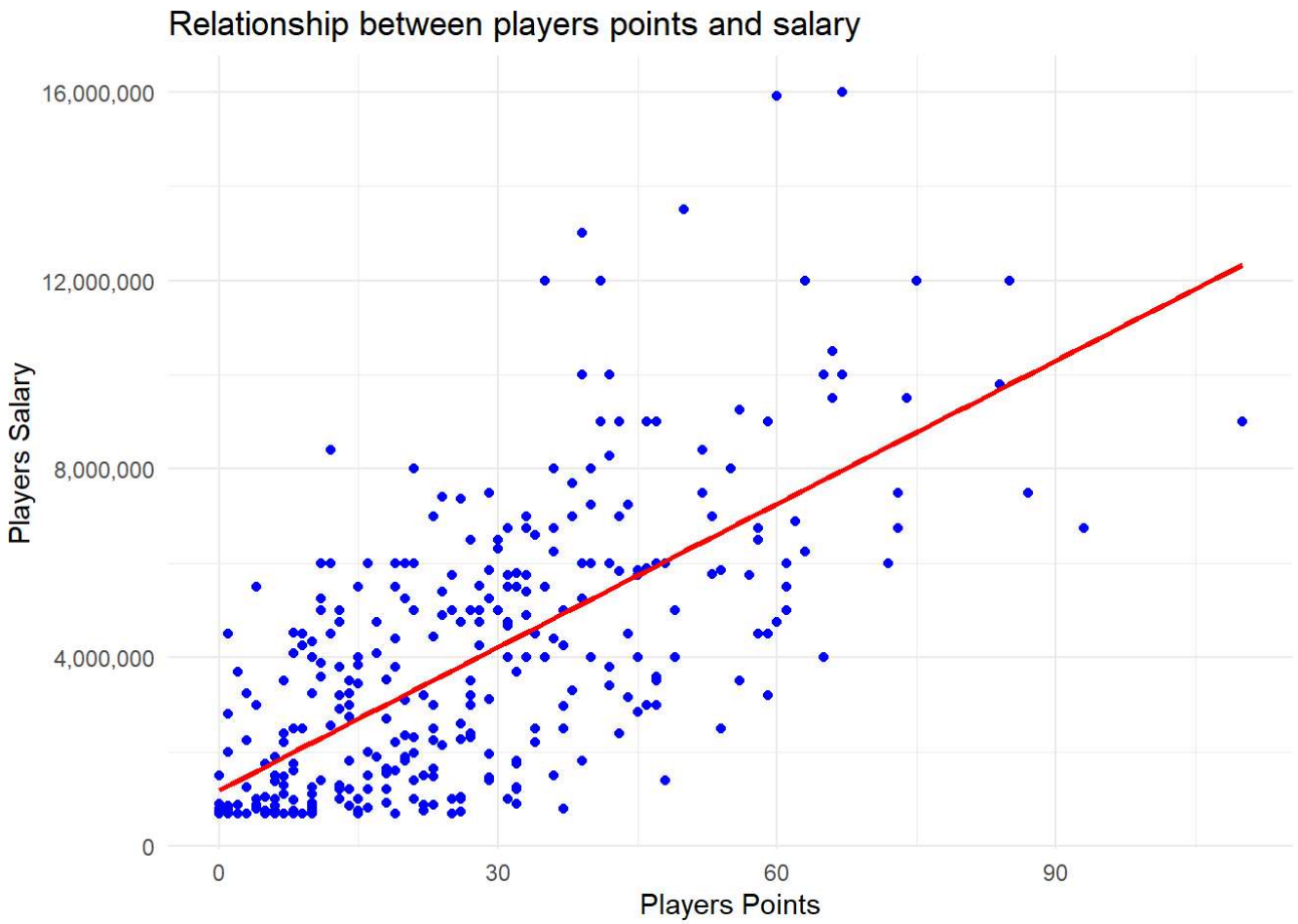
```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(train_data, aes(x = P, y = Salary))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", color = "red", se = FALSE)+
  xlab("Players Points")+
  ylab("Players Salary")+
  ggtitle("Relationship between players points and salary")+
  scale_x_continuous(labels = comma)+
  scale_y_continuous(labels = comma)+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_minimal();
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Answer: The relationship is positive linear one between points and salaries of players. It aligns with the sports reality as well, the more the points scored by a player, the more valuable they are and hence they receive a better salary than other players. Another expected pattern is that as we go higher up in points, the number of players decrease which you can see in most sports and hence they command higher salaries. There are few outliers too which we can assume might be legends or consistent performers season over season who have extremely high salaries.

9. Now, again using training set data only, find the correlation between Salary and Points. Then, use cor.test() to see whether this correlation is significant. What is this correlation? Is it a strong one? Is the correlation significant?

```
significance <- cor.test(train_data$Salary, train_data$P)
significance
```

```
##
##  Pearson's product-moment correlation
##
## data:  train_data$Salary and train_data$P
## t = 17.272, df = 334, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6258708 0.7395156
## sample estimates:
##       cor
## 0.6868692
```

Answer: The correlation between players salary and points is almost 0.69 which is strong positive relationship because the r value is closer to 1. The correlation is also statistically significant from the results as it rejects the null hypothesis with a strong t-statistic and a p-value of well below 5% suggesting its strong significance.

10. Using your training set, create a simple linear regression model, with Salary as your outcome variable and Points as your input variable. Use the summary() function to display the results of your model.

```
linear_model <- lm(train_data$Salary ~ train_data$P)
summary(linear_model)
```

```
##
## Call:
## lm(formula = train_data$Salary ~ train_data$P)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4647164 -1397551  -485708  1298288  8637472
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1185708     191827   6.181 1.85e-09 ***
## train_data$P   101280       5864  17.272  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2156000 on 334 degrees of freedom
## Multiple R-squared:  0.4718, Adjusted R-squared:  0.4702
## F-statistic: 298.3 on 1 and 334 DF,  p-value: < 2.2e-16
```

11. What are the minimum (most negative) and maximum (most positive) residual values in this model?

a. Find the observation whose rating generated the highest residual value in your model. What was that player's actual salary? What did the model predict that it would be? How is the residual calculated from the two numbers that you just found?

```
residuals <- residuals(linear_model)
max_residual_index <- which.max(abs(residuals))
predicted_salary <- predict(linear_model)[max_residual_index]
actual_salary <- train_data$Salary[max_residual_index]
#Manually calculating residual value from predicted & actual
analytical_residual = actual_salary - predicted_salary
analytical_residual
```

```
##     105
## 8637472
```

Answer: The minimum residual value is $4,647,164 and maximum value is $8,637,472. (a) - The 105th observation generated the highest residual value and the actual salary of that player was $15,900,000 and the model predicted the player's salary to be $7,262,528. The residual was calculated by subtracting predicted salary from the actual salary in the data.

b. Find the observation whose rating generated the lowest (i.e. most negative) residual value. What was that player's actual salary? What did the model predict that it would be? How is the residual calculated from the two numbers that you just found?

```
min_residual_index <- which.min((residuals))
predicted_salary_low <- predict(linear_model)[min_residual_index]
actual_salary_low <- train_data$Salary[min_residual_index]
#Manually calculating residual value from predicted & actual
min_residual <- actual_salary_low - predicted_salary_low
min_residual
```

```
##       72
## -4647164
```

Answer: The 72nd observation generated the minimum residual and the actual salary for the player was 1,400,000 and the predicted salary was 6,047,164. The residual was manually calculated subtracting these values.

c.It looks like there are some cases where this model is quite a bit "off the mark." Write a few sentences with your thoughts about why Points might be an okay starting point, but may not perfectly predict Salary (and do not just be general and say that 'other things might matter' – be more specific than that). [Note:this answer does NOT require any specific knowledge of hockey, beyond knowing that points are good, but that other things matter, too. As always, you may use any source].

Answer: It is important to consider this from a hockey's business model context. If I own a team and want to buy players, I wouldn't just make decisions of purchase just based on the points they have made, reason being the team will have a mix of offensive and defensive players, players for specialized positions like Wing or Defenseman. Points is a good starting point, but it doesn't explain if the player has experience, which comes from games played which could be a significant factor in salary. Another important this to consider is points is only considering offensive players, how to assess salary for our

defense players, in that case blocked shots can be used. If we further explore, points is single dimension in many ways, I would value a player more who scores goals more consistently and is efficient and they would command more salary so "sh_per" is an imp variable. Lastly, dynamic performance level variables like takeaways and giveaways indicate current performance of player which might affect contract negotiations.

12. What is the regression equation generated by your model? Make up a hypothetical input value and explain what it would predict as an outcome. To show the predicted outcome value, you can either use a function in R, or just explain what the predicted outcome would be, based on the regression equation and some simple math.

```
# Let's say a player has scored 60 points then its salary will be
x <- 60
#Regression equation of the linear model created is
y <- 1185708 + 101280 * x
y
```

```
## [1] 7262508
```

13. Using the accuracy() function from the forecast package, assess the accuracy of your model against both the training set and the validation set. What is the purpose of making this comparison? Focus on RMSE and MAE here in particular

```
validation_data_model <- lm(validation_data$Salary ~ validation_data$P)
summary(validation_data_model)
```

```
##
## Call:
## lm(formula = validation_data$Salary ~ validation_data$P)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4473153 -1243859  -628284  1041804  9180741
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1475009     222927   6.617 2.69e-10 ***
## validation_data$P    96106       7311  13.146  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2227000 on 223 degrees of freedom
## Multiple R-squared:  0.4366, Adjusted R-squared:  0.4341
## F-statistic: 172.8 on 1 and 223 DF,  p-value: < 2.2e-16
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.4.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
accuracy(linear_model)
```

```
##                        ME    RMSE     MAE       MPE     MAPE      MASE
## Training set -7.609473e-11 2149201 1682231 -47.89095 74.20693 0.7051768
```

```
accuracy(validation_data_model)
```

```
##                       ME    RMSE     MAE       MPE     MAPE      MASE
## Training set -1.23186e-11 2217117 1671849 -50.62909 75.67967 0.7395876
```

```
mean(nhl_df$Salary)
```

```
## [1] 3746284
```

Answer: The purpose of comparing the performance of the model on training set v/s the validation set is to understand how accurate is the model predicting on unseen data i.e validation data. This helps understand if the model has captured sufficient variation in the dependent variable to explain its influence on outcome variable.This gives us direction for next steps which is based on this comparison, it might be further refining the model and testing again or proceed to cross-validation. In the comparison of performance, we know that the average salary is $3.75M and the model predicts within +- $2.15M error on training data while on validation data the RMSE slightly increases to +-2.21M which suggests possible overfitting in this case, but since the dataset is small we can say that the difference in RMSE is acceptable. On the other hand, MAE values are also close between validation & training data which suggests that the model makes reasonably accurate predictions.

14. How does your model's RMSE compare to the standard deviation of hockey player salaries in the training set? What can such a comparison teach us about the model?

```
sd_salaries <- sd(train_data$Salary)
sd_salaries
```

```
## [1] 2961560
```

Answer: The standard deviation of the Salary is $2.96M whereas the RMSE for the training set is $2.15M, a lower RMSE than SD of the outcome variable suggests that the model is doing well in explaining the variability/underlying pattern in the data. However, it is not significantly lower so there is room for improvement by adding more predictors.

Multiple Linear Regression

1. Build a correlation table in R that depicts the correlations among all of the numerical variables that you might use as predictors (use your training set to build this). Are there any variable relationships that suggest that multicollinearity could be an issue here? If so, for any strongly correlated variable pair, remove a variable that should be taken out of the model. If you removed any, how did you decide which ones to remove? If not,why did you keep the ones that you have left?

```
correlation_matrix <- cor(train_data[, c("Gamesplayed","Goals","Assists","P","Shots",
                                  "Sh_perc","PIM","Giveaways","Takeaways",
                                  "Hits","HitsTaken","blockedshots","PlusMinus"
                                  )])
round(correlation_matrix, 2)
```

```
##              Gamesplayed Goals Assists    P Shots Sh_perc  PIM Giveaways
## Gamesplayed         1.00  0.56    0.64 0.66  0.75    0.10 0.49      0.66
## Goals               0.56  1.00    0.72 0.89  0.84    0.42 0.27      0.45
## Assists             0.64  0.72    1.00 0.95  0.82    0.18 0.29      0.73
## P                   0.66  0.89    0.95 1.00  0.89    0.30 0.30      0.67
## Shots               0.75  0.84    0.82 0.89  1.00    0.18 0.35      0.68
## Sh_perc             0.10  0.42    0.18 0.30  0.18    1.00 0.06      0.00
## PIM                 0.49  0.27    0.29 0.30  0.35    0.06 1.00      0.39
## Giveaways           0.66  0.45    0.73 0.67  0.68    0.00 0.39      1.00
## Takeaways           0.70  0.66    0.75 0.77  0.73    0.20 0.31      0.61
## Hits                0.46  0.10    0.04 0.07  0.22   -0.02 0.59      0.28
## HitsTaken           0.67  0.25    0.38 0.35  0.45   -0.04 0.40      0.54
## blockedshots        0.43 -0.10    0.24 0.11  0.23   -0.23 0.30      0.53
## PlusMinus           0.08  0.11    0.28 0.22  0.14    0.05 0.02      0.09
##              Takeaways  Hits HitsTaken blockedshots PlusMinus
## Gamesplayed       0.70  0.46      0.67         0.43      0.08
## Goals             0.66  0.10      0.25        -0.10      0.11
## Assists           0.75  0.04      0.38         0.24      0.28
## P                 0.77  0.07      0.35         0.11      0.22
## Shots             0.73  0.22      0.45         0.23      0.14
## Sh_perc           0.20 -0.02     -0.04        -0.23      0.05
## PIM               0.31  0.59      0.40         0.30      0.02
## Giveaways         0.61  0.28      0.54         0.53      0.09
## Takeaways         1.00  0.17      0.40         0.20      0.17
## Hits              0.17  1.00      0.49         0.38     -0.03
## HitsTaken         0.40  0.49      1.00         0.59      0.17
## blockedshots      0.20  0.38      0.59         1.00      0.19
## PlusMinus         0.17 -0.03      0.17         0.19      1.00
```

```
correlation_matrix
```

```
##              Gamesplayed       Goals   Assists          P     Shots
## Gamesplayed   1.00000000  0.56451513 0.6444206 0.65868127 0.7544046
## Goals         0.56451513  1.00000000 0.7184702 0.89271156 0.8430254
## Assists       0.64442061  0.71847022 1.0000000 0.95482483 0.8228781
## P             0.65868127  0.89271156 0.9548248 1.00000000 0.8932883
## Shots         0.75440458  0.84302544 0.8228781 0.89328831 1.0000000
## Sh_perc       0.09941094  0.42183535 0.1780453 0.29557400 0.1842455
## PIM           0.48985981  0.27090359 0.2906290 0.30402933 0.3524177
## Giveaways     0.66085553  0.45148313 0.7294006 0.66544533 0.6823268
## Takeaways     0.70090113  0.66156566 0.7476917 0.76705090 0.7341262
## Hits          0.46432618  0.09993169 0.0424536 0.07019896 0.2156997
## HitsTaken     0.67322567  0.24885747 0.3827279 0.35427814 0.4524551
## blockedshots  0.43252041 -0.10072830 0.2414972 0.11342292 0.2255127
## PlusMinus     0.08440308  0.10616380 0.2755828 0.22389816 0.1375232
##                   Sh_perc        PIM    Giveaways Takeaways        Hits
## Gamesplayed   0.099410944 0.48985981  0.660855529 0.7009011  0.46432618
## Goals         0.421835349 0.27090359  0.451483125 0.6615657  0.09993169
## Assists       0.178045298 0.29062903  0.729400647 0.7476917  0.04245360
## P             0.295574000 0.30402933  0.665445334 0.7670509  0.07019896
## Shots         0.184245547 0.35241772  0.682326799 0.7341262  0.21569972
## Sh_perc       1.000000000 0.05738867 -0.001109148 0.2045980 -0.01823202
## PIM           0.057388667 1.00000000  0.392470471 0.3085863  0.58869863
## Giveaways    -0.001109148 0.39247047  1.000000000 0.6090072  0.28391811
## Takeaways     0.204598027 0.30858634  0.609007162 1.0000000  0.17320211
## Hits         -0.018232023 0.58869863  0.283918111 0.1732021  1.00000000
## HitsTaken    -0.038468093 0.40308152  0.535941765 0.4013249  0.48605939
## blockedshots -0.234066113 0.29737069  0.532930477 0.1973755  0.37624720
## PlusMinus     0.046379327 0.02064210  0.094599126 0.1681962 -0.03046354
##                HitsTaken blockedshots   PlusMinus
## Gamesplayed   0.67322567    0.4325204  0.08440308
## Goals         0.24885747   -0.1007283  0.10616380
## Assists       0.38272790    0.2414972  0.27558281
## P             0.35427814    0.1134229  0.22389816
## Shots         0.45245508    0.2255127  0.13752323
## Sh_perc      -0.03846809   -0.2340661  0.04637933
## PIM           0.40308152    0.2973707  0.02064210
## Giveaways     0.53594177    0.5329305  0.09459913
## Takeaways     0.40132488    0.1973755  0.16819617
## Hits          0.48605939    0.3762472 -0.03046354
## HitsTaken     1.00000000    0.5877669  0.16560082
## blockedshots  0.58776689    1.0000000  0.18854340
## PlusMinus     0.16560082    0.1885434  1.00000000
```

Answer: For identifying multicollinearity, we can keep a threshold of cor > 0.7, any variable pair with correlation above this will be identify as highly correlated. Following are the highly correlated variables: Goals & Points(0.89), Assists & Points (0.95), Goals & Assists (0.72), Shots & Goals, Shots & Assists are both above 0.8 and Takeaways & Assists are correlated (0.75). Since Points is product of goals and assists so we can just use Points, since we have already removed goals and assists we can keep Takeaways and Shots from these pairs.

2. What are dummy variables? In a couple of sentences, describe what they are and explain their purpose. (Question #2 is a general question, and is not directly related to this particular dataset or problem).

Answer: Dummy variables are binary (0/1) variables that are used to represent categorical data that have N categories in statistical models, because they cannot be directly used in mathematical equations in their original "char" form. There main purpose is to be used in regression models when they are supposed to be used as predictors, and creating them allows the model to use interpret and use them in explaining interrelationships with outcome variable.

3.Using backward elimination, build a multiple regression model with the remaining data in your training set, with the goal of predicting the variable Salary. Start with all of the potential predictors that you have left (if you eliminated any variables in a previous step in this section, don't bring them back – they're gone!) [Be sure to use a function for this process, rather than a manual approach] a. Show a summary of your resulting multiple linear regression model

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.4.2
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
MLR <- lm(train_data$Salary ~ Gamesplayed + P + Shots+ Sh_perc + PIM +
            Giveaways +Takeaways + Hits + HitsTaken + blockedshots + PlusMinus,
          data = train_data)
best_MLR <- stepAIC(MLR, direction = "backward")
```

```
## Start:  AIC=9770.87
## train_data$Salary ~ Gamesplayed + P + Shots + Sh_perc + PIM +
##     Giveaways + Takeaways + Hits + HitsTaken + blockedshots +
##     PlusMinus
##
##                Df  Sum of Sq        RSS    AIC
## - Sh_perc       1 2.2508e+11 1.3325e+15 9768.9
## - PIM           1 7.2040e+11 1.3330e+15 9769.1
## <none>                       1.3323e+15 9770.9
## - Giveaways     1 8.0789e+12 1.3403e+15 9770.9
## - Takeaways     1 9.9220e+12 1.3422e+15 9771.4
## - PlusMinus     1 1.2580e+13 1.3448e+15 9772.0
## - Shots         1 1.2769e+13 1.3450e+15 9772.1
## - Hits          1 1.3040e+13 1.3453e+15 9772.1
## - Gamesplayed   1 1.7960e+13 1.3502e+15 9773.4
## - HitsTaken     1 3.0658e+13 1.3629e+15 9776.5
## - blockedshots  1 8.3695e+13 1.4160e+15 9789.3
## - P             1 1.0600e+14 1.4383e+15 9794.6
##
## Step:  AIC=9768.93
## train_data$Salary ~ Gamesplayed + P + Shots + PIM + Giveaways +
##     Takeaways + Hits + HitsTaken + blockedshots + PlusMinus
##
##                Df  Sum of Sq        RSS    AIC
## - PIM           1 7.1133e+11 1.3332e+15 9767.1
## <none>                       1.3325e+15 9768.9
## - Giveaways     1 8.6294e+12 1.3411e+15 9769.1
## - Takeaways     1 9.8507e+12 1.3423e+15 9769.4
## - PlusMinus     1 1.2533e+13 1.3450e+15 9770.1
## - Hits          1 1.3418e+13 1.3459e+15 9770.3
## - Shots         1 1.3438e+13 1.3459e+15 9770.3
## - Gamesplayed   1 1.8020e+13 1.3505e+15 9771.4
## - HitsTaken     1 3.0643e+13 1.3631e+15 9774.6
## - blockedshots  1 8.6400e+13 1.4189e+15 9788.0
## - P             1 1.1118e+14 1.4437e+15 9793.9
##
## Step:  AIC=9767.11
## train_data$Salary ~ Gamesplayed + P + Shots + Giveaways + Takeaways +
##     Hits + HitsTaken + blockedshots + PlusMinus
##
##                Df  Sum of Sq        RSS    AIC
## <none>                       1.3332e+15 9767.1
## - Giveaways     1 8.9913e+12 1.3422e+15 9767.4
## - Takeaways     1 9.6855e+12 1.3429e+15 9767.5
## - PlusMinus     1 1.2682e+13 1.3459e+15 9768.3
## - Shots         1 1.3046e+13 1.3462e+15 9768.4
## - Hits          1 1.3774e+13 1.3470e+15 9768.6
## - Gamesplayed   1 1.7477e+13 1.3507e+15 9769.5
## - HitsTaken     1 3.0833e+13 1.3640e+15 9772.8
## - blockedshots  1 8.6620e+13 1.4198e+15 9786.3
## - P             1 1.1563e+14 1.4488e+15 9793.1
```

```
summary(best_MLR)
```

```
##
## Call:
## lm(formula = train_data$Salary ~ Gamesplayed + P + Shots + Giveaways +
##     Takeaways + Hits + HitsTaken + blockedshots + PlusMinus,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5290394 -1237227  -299363  1241061  8465776
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1566569     320038   4.895 1.55e-06 ***
## Gamesplayed    -23879      11551  -2.067  0.03950 *
## P               80753      15187   5.317 1.96e-07 ***
## Shots            8267       4628   1.786  0.07502 .
## Giveaways       14903      10051   1.483  0.13910
## Takeaways       19598      12734   1.539  0.12479
## Hits            -5272       2873  -1.835  0.06739 .
## HitsTaken      -13145       4787  -2.746  0.00637 **
## blockedshots    22623       4916   4.602 5.99e-06 ***
## PlusMinus      -21324      12109  -1.761  0.07918 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2022000 on 326 degrees of freedom
## Multiple R-squared:  0.5463, Adjusted R-squared:  0.5337
## F-statistic: 43.61 on 9 and 326 DF,  p-value: < 2.2e-16
```

4. Model metrics

a. What is the total sum of squares for your model? (SST). This can be found by summing all of the squared differences from the mean for your outcome variable.
b. What is the total sum of squares due to regression for your model? (SSR). This can be found by summing all the squared differences between the fitted values and the mean for your outcome variable. Do not use any other SSR definition, besides the one listed here in the previous sentence.
c. What is your SSR / SST? Where can you also see this value in the summary of your regression model?

```
SST <- sum((train_data$Salary - mean(train_data$Salary))^2)
SST
```

```
## [1] 2.938231e+15
```

```
salary_pred_train <- predict(MLR, newdata = train_data)
SSR <- sum((salary_pred_train - mean(train_data$Salary))^2)
SSR
```

```
## [1] 1.605967e+15
```

```
R_squared <- SSR/SST
R_squared
```

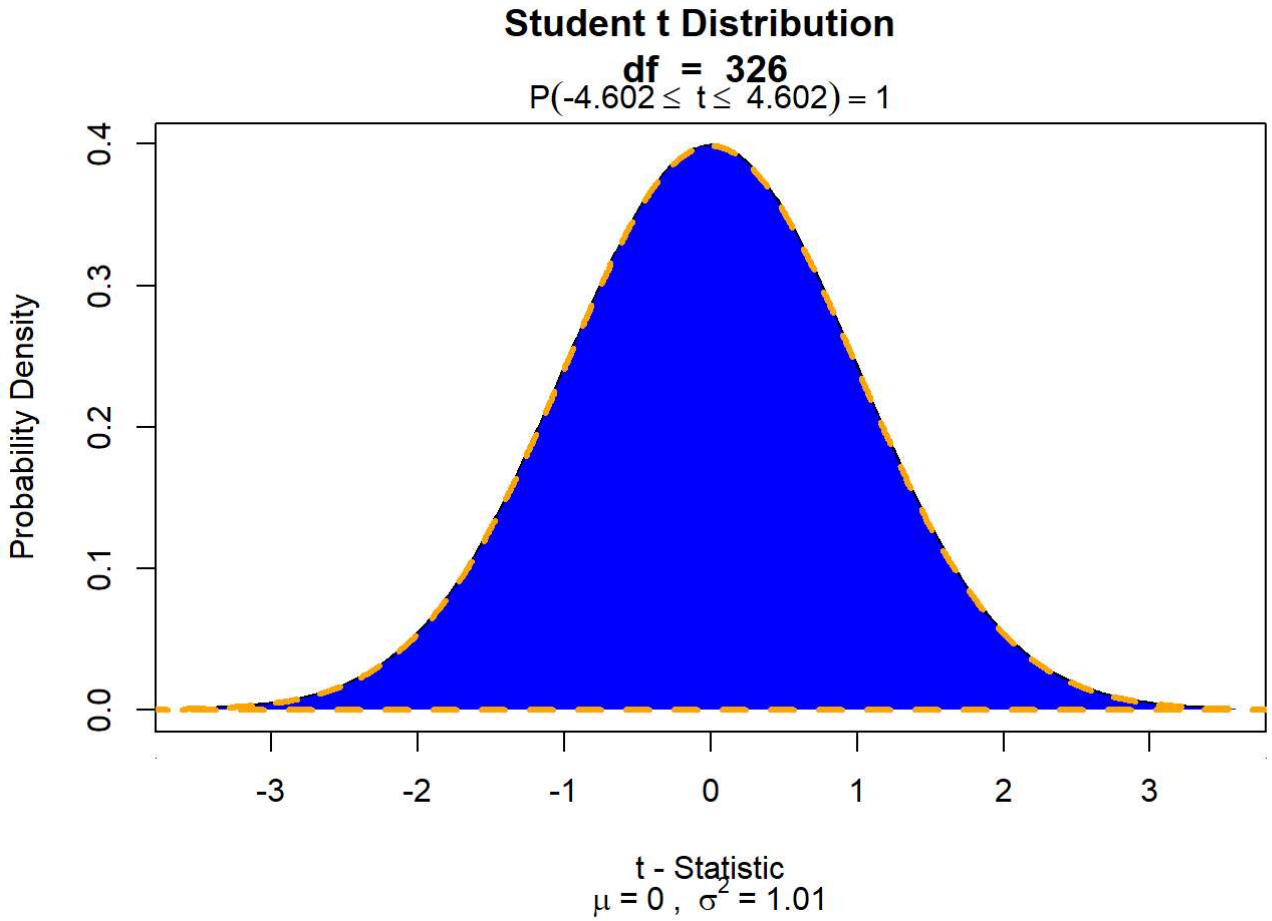
```
## [1] 0.546576
```

Answer: SSR/SST is R_square which explains the variability in the Salary of hockey players.

5. Getting from a t-value to a p-value. Choose one of the predictors from your model (it could be a numeric input variable or a single level from a categorical input). What is the t-value for that predictor? Using the visualize.t() function from the visualize package,create a plot of the t-distribution that shows the distribution for that t-value and the number of degrees of freedom in your model. What percent of the curve is shaded? How does this relate to the p-value for that predictor?

```
library(visualize)
t_value <- 4.602
df <- 326

visualize.t(stat = c(-t_value, t_value), df = df, section = "bounded")
```



```
prob_shaded <- pt(t_value,df) - pt(-t_value,df)
percent_shaded <- prob_shaded * 100
percent_shaded
```

```
## [1] 99.9994
```

Answer: The probability of area shaded is 99.9% and this represents the CI for two-tailed hypothesis test where p-value measures how extreme the t-value is and this rejects null hypothesis and indicates that the blocked_shots predictor is significant.

6. What is your model's F-statistic? What does the F-Statistic measure? Using R, demonstrate where the F-Statistic comes from (you can use the formula/process shown in the class slides with the Sacramento example).

```
k <- length(coefficients(best_MLR)) - 1
n <- nrow(train_data)
SSE <- SST - SSR

F_statistic <- (SSR / k) / (SSE / (n - k - 1))
F_statistic
```

```
## [1] 43.66377
```

Answer: The F-statistic of the model from summary function is 43.61 and manually it comes close to that figure as seen above. The F-statistic collectively explains a significant amount of variance in the dependent variable (Salary).

7. Make up a fictional hockey player, and assign attributes to that player for each variable used in your model. What does your model predict that the player's salary will be? To answer this, you can use a function in R or just explain it using the equation and some simple math.

```
fictional_player <- data.frame(Gamesplayed = 70, P = 55,
  Shots = 180, Giveaways = 35, Takeaways = 50, Hits = 60, HitsTaken = 40,
  blockedshots = 20,PlusMinus = 5)
salary_prediction <- predict(best_MLR, newdata = fictional_player)
salary_prediction
```

```
##       1
## 6829635
```

8. Using the accuracy() function from the forecast package, assess the accuracy of your model against both the training set and the validation set. What do you notice about these results? Describe your findings in a couple of sentences. In this section, you should talk about the overfitting risk and also about the way your MLR model diered from your SLR model in terms of accuracy. Also, as you think about overall model accuracy, identify at least one important limitation of trying to predict income using these available variables.

```
validation_MLR_model <- predict(best_MLR, newdata = validation_data)
library(forecast)
validation_accuracy <- accuracy(validation_MLR_model, validation_data$Salary)
training_MLR_model <- predict(best_MLR, newdata = train_data)
training_accuracy <- accuracy(training_MLR_model, train_data$Salary)
print("Validation Set Accuracy:")
```

```
## [1] "Validation Set Accuracy:"
```

```
print(validation_accuracy)
```

```
##                ME    RMSE     MAE       MPE     MAPE
## Test set 36914.43 1992781 1504847 -37.65467 64.36569
```

```
print("Training Set Accuracy:")
```

```
## [1] "Training Set Accuracy:"
```

```
print(training_accuracy)
```

```
##                     ME    RMSE     MAE       MPE     MAPE
## Test set 5.390223e-09 1991949 1523040 -41.65882 66.80336
```

Answer: Both RMSE and MAE are almost similar for training and test data, which indicates that the MLR model is stable, this also indicates that the model is generalizing well and is not overfitting which in SLR seem to be the case by a small margin. MAE however is still high which states that model predicts salaries with a error of +- 1.5M, so maybe there is still room for improvement.If we compare the MLR model with SLR, we can see that RMSE has slightly improved but there is considerable improvement in MAE from 1.67M in SLR to 1.5M in MLR which suggests that the MLR model is better at predicting the sale prices than SLR. In terms of limitations of using these predictors, the most important I can think of is the absence of external variables in the dataset like economic factors, contract negotiations as well as there are no physical attributes of players that have been considered as they impact performance and would be one of the determinants in the salary of the player. By physical attributes I meant physical fitness quantified in some measure.