

Ordinal Logistic Regression in Predicting Basketball Player Salary







Skyler Chauff, John Beggs

Who is the highest paid NBA player?

FW

STEPHEN CURRY

THE NBA FINALS STATS










| 2015 | 2016 | 2017 | 2018 | 2019 | 2022 |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  |  |  |  |  |  |
| 26.0 PPG 5.2 RPG 6.3 APG 1.8 SPG 44.6 FG% 39.8 3P% 92.5 FT% | 22.6 PPG 4.9 RPG 3.7 APG 0.9 SPG 41.5 FG% 40.4 3P% 95.8 FT% | 26.8 PPG 8.0 RPG 9.4 APG 2.2 SPG 42.8 FG% 38.2 3P% 90.0 FT% | 27.5 PPG 6.0 RPG 6.8 APG 1.5 SPG 38.3 FG% 38.8 3P% 100.0 FT% | 30.5 PPG 5.2 RPG 6.0 APG 1.5 SPG 40.8 FG% 33.8 3P% 94.8 FT% | 31.2 PPG 6.0 RPG 5.0 APG 2.0 SPG 48.2 FG% 43.7 3P% 85.7 FT% |



So How Can We Predict, Historically, If Someone is Worth the Salary?

ORDINAL LOGISTIC REGRESSION



| | | | | |
|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|  \$74,742 |  \$92,857 |  \$122,741 |  \$150,000 |  \$230,620 |
|  \$250,000 |  \$268,032 |  \$330,000 |  \$333,333 |  \$555,217 |
|  \$576,230 |  \$702,311 |  \$999,200 |  \$1,017,781 |  \$1,017,781 |
|  \$1,017,781 |  \$1,017,781 |  \$1,017,781 |  \$1,017,781 |  \$1,017,781 |

| | | | | |
|--------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|  \$45.7 M STEPH CURRY |  \$44.3 M JOHN WALL |  \$44.2 M RUSSELL WESTBROOK |  \$44.2 M CHRIS PAUL |  \$43.8 M JAMES HARDEN |
|  \$43.7 M DAMIAN LILLARD |  \$41.2 M LEBRON JAMES |  \$40.9 M KEVIN DURANT |  \$39.3 M PAUL GEORGE |  \$39.3 M GIANNIS ANTETOKOUNMPO |

Purpose, Background, and Motivation



Ordinal Logistic Regression Can Predict NBA Player Worth by Assigning them a Salary Bin!



Learning Goals

1. Understand how Ordinal Logistic Regression will be applied to an NBA dataset and predicting NBA salary.
2. Understand the Theory and Application of Ordinal Logistic Regression (OLR).
3. Prepare data for OLR.
4. Construct an OLR model in R using relevant variables from an NBA data set.
5. Interpret OLR model coefficients and their implications for predicting NBA salary.
6. Apply OLR to predict salary and evaluate your model's performance.

Data

- Contains information on various measurables in professional basketball from the 2021-2022 NBA regular season
- 812 player-team stints for the season.

Predicting NBA Salaries with Machine Learning

Building a machine learning model with Python to predict NBA salaries and analyze the most impactful variables



Gabriel Pastorello · [Follow](#)

Published in Towards Data Science · 9 min read · Aug 24, 2023

Okay, but what variables might we want to look at?

Variables Selected

Explanatory Variables
(Everything else)

Player: The name of the basketball player.

Team: The team the player is currently playing for.

Games Played: The number of games the player has participated in.

Games Started: The number of games the player has started. **Center Position:** A binary variable indicating whether the player primarily plays the center position (1 if yes, 0 if no).

Power Forward Position: A binary variable indicating whether the player primarily plays the power forward position (1 if yes, 0 if no).

Point Guard Position: A binary variable indicating whether the player primarily plays the point guard position (1 if yes, 0 if no).

Small Forward Position: A binary variable indicating whether the player primarily plays the small forward position (1 if yes, 0 if no).

Shooting Guard Position: A binary variable indicating whether the player primarily plays the shooting guard position (1 if yes, 0 if no).

Field Goals Made per Game: The average number of field goals made by the player per game (non free throw)

3-Point Field Goals Made per Game: The average number of three-point field goals made by the player per game.

Steals per Game: The average number of steals made by the player per game.

Blocks per Game: The average number of shots blocked by the player per game.

Points per Game: The average number of points scored by the player per game.

Total Points: The total number of points scored by the player.

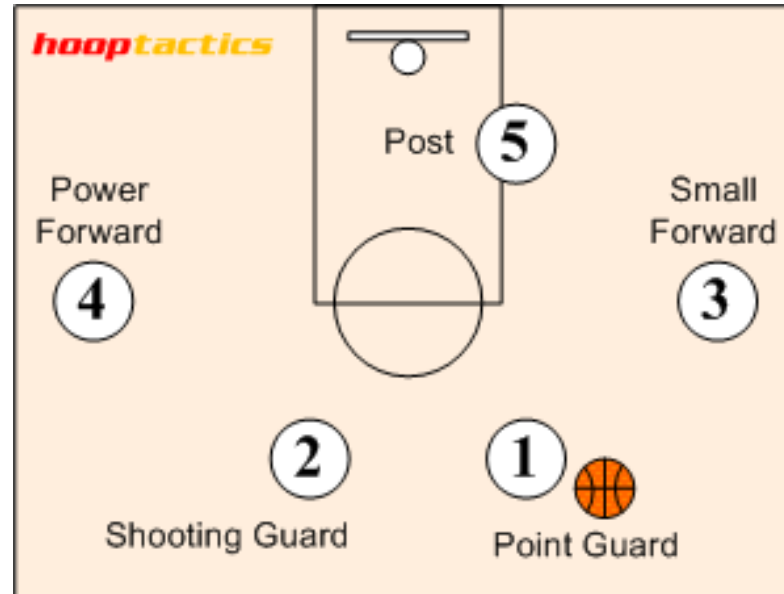
Salary: The salary of the player in dollars.

Age: The age of the player in years.

RESPONSE VARIABLE
(Binned)!

Variable Examples

Blocks



Steals

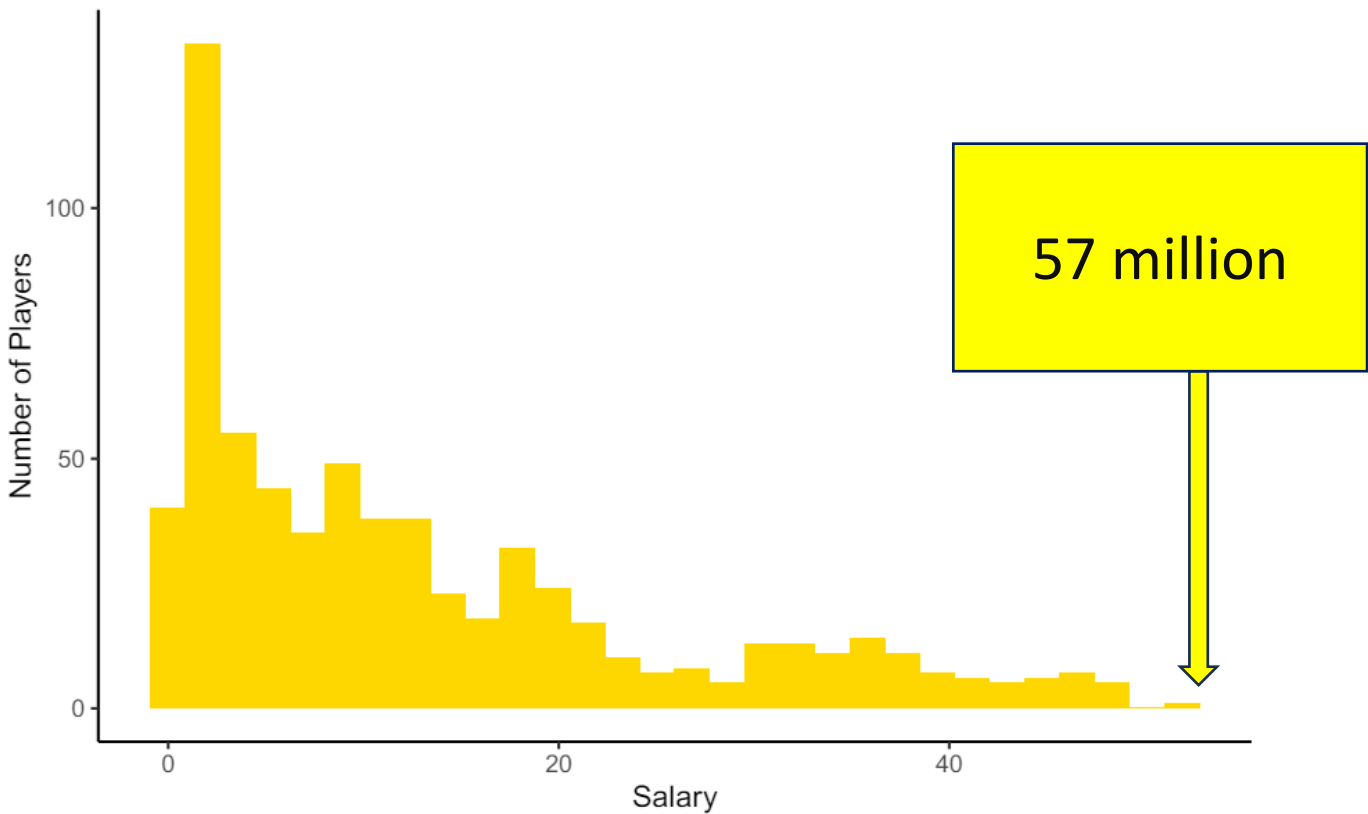


Binning NBA Salaries

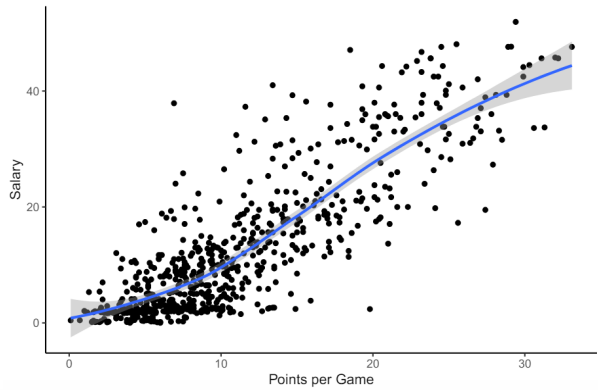
Salary Binning

Low Salary: < 2.53 million
Middle Salary: 2.54 - 9.74 million
High salary: > 9.75 million

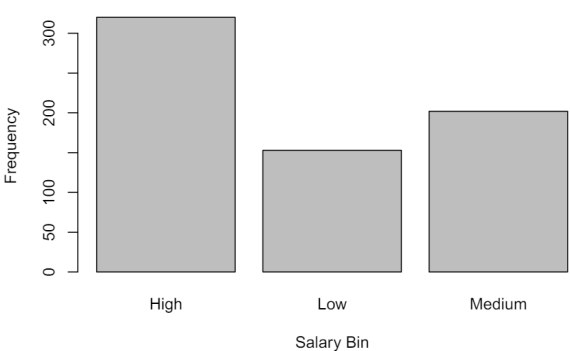
Distribution of NBA Salaries



Scatterplot of Points per Game vs. Salary



Counts of Salary Bins



Methods and Instructional Content

The equation for Ordinal Logistic Regression is defined as:

$$\text{logit}(P(Y \leq j)) = \alpha_j + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

Let's break down the model above in its individual components.

- Y : The dependent variable, representing the ordinal response variable. This is specifically the salary bin of NBA players.
- $P(Y \leq j)$: The probability of the dependent variable Y being less than or equal to category j .
- $\text{logit}(P(Y \leq j))$: The log odds of the dependent variable Y being less than or equal to category j .
- α_j : The intercept parameter associated with category j .
- $\beta_1, \beta_2, \dots, \beta_p$: The coefficients associated with the independent variables x_1, x_2, \dots, x_p respectively.
- x_1, x_2, \dots, x_p : The independent variables, representing the characteristics or attributes that influence the ordinal response variable Y . This includes player stats we are looking to use in order to “predict” the salary bin.

The odds of being less than or equal a particular category is defined as:

$$\frac{P(Y \leq j)}{P(Y > j)}$$

Assumptions

- Proportional odds assumption
 - The relationship between each pair of outcome categories is consistent across all levels of the independent variables
- Independent observations
 - Not necessarily independent **as teams are resourced differently**
- Linearity of Logit
 - Relationship between independent variables and log odds of outcome categories is linear



Exercises

Exercise 1: Binning NBA Salaries

At this point, we need to bin the NBA salaries. We will bin into three groups of low, medium, and high. This will then enable us to conduct the ordinal logistic regression in which we will predict the bin of players based on their performance in the game (low, medium or high).

In this, we recommend binning them based on percentiles. Because we want them in thirds, set the first bin at the 33rd percentile and the second and the 67th percentile. What are some other ways you might bin a numerical variable like salary?

EXERCISE 1

```
# Your Code Here

perc_33 <- quantile(bball_clean$Salary, probs = 0.33)
perc_67 <- quantile(bball_clean$Salary, probs = 0.67)

bball_clean$sal_bin <- ifelse(bball_clean$Salary <= perc_33, "Low",
                             ifelse(bball_clean$Salary >= perc_67, "Medium", "High"))

kable(head(bball_clean[, 1:7]))
```

| Player | Age | Tm | G | GS | C | PF |
|--------------------------|-----|-----|----|----|---|----|
| Precious Achiuwa | 23 | TOR | 55 | 12 | 1 | 0 |
| Steven Adams | 29 | MEM | 42 | 42 | 1 | 0 |
| Bam Adebayo | 25 | MIA | 75 | 75 | 1 | 0 |
| Ochai Agbaji | 22 | UTA | 59 | 22 | 0 | 0 |
| Santi Aldama | 22 | MEM | 77 | 20 | 0 | 1 |
| Nickeil Alexander-Walker | 24 | MIN | 59 | 3 | 0 | 0 |

One way to bin them might be with simply making it so that they're all the same width or size. In other words, all of the levels have the same number of variables. Another solution may be to do it in a custom manner based on their position, meaning we bin for each position even if we have to group a few positions together.

We realize that we need to account for position. We need to do this because player position is a likely confounding variable that affects performance as players on the court have distinct responsibilities. Thus, we can also see how the position of the player influences salary bin.

In the chunk below, transform the Position column so that each position is now represented numerically (1 through 5, associated with the proper position: 1 is PG, 2 is SG, 3 is SF, 4 is PF, and 5 is C). Pay close attention to how the data is currently represented. A proper implementation will have a single variable called "Position".

Exercise 2: Filter for Players 25 and older

```
# Your Code Here

bball_clean$Position <- case_when(
  bball_clean$PG == 1 ~ 1,
  bball_clean$SG == 1 ~ 2,
  bball_clean$SF == 1 ~ 3,
  bball_clean$PF == 1 ~ 4,
  bball_clean$C == 1 ~ 5,
  TRUE ~ 0
)
```

Which position in basketball (PG, SG, SF, PF, or C) do you think gets paid the most? Why?

I would guess the PF gets paid the most because I would guess that they score the most points are the most efficient defensively. PFs have size, speed, and athleticism, all of which seem to me would result in higher pay.

We must now filter Basketball Clean dataframe so that it only includes players who are 25 and older. This is important because the rookie salary is calibrated differently and most rookies are under the age of 25.

Keep in mind that filtering for ages greater than 25 decreases the number of observations in the data set. Before you do this, you need to make sure you still have enough data to create a good model.

```
bball_clean <- bball_clean %>%
  filter(Age >= 25)
```

Let's check our work.

Using the data frame, create the visualizations as instructed.

First, create a barplot showing how many players are considered to be low, medium, or high.

Exercise 3: Visualize Salary Bins and Salary Ranges

Let's check our work.

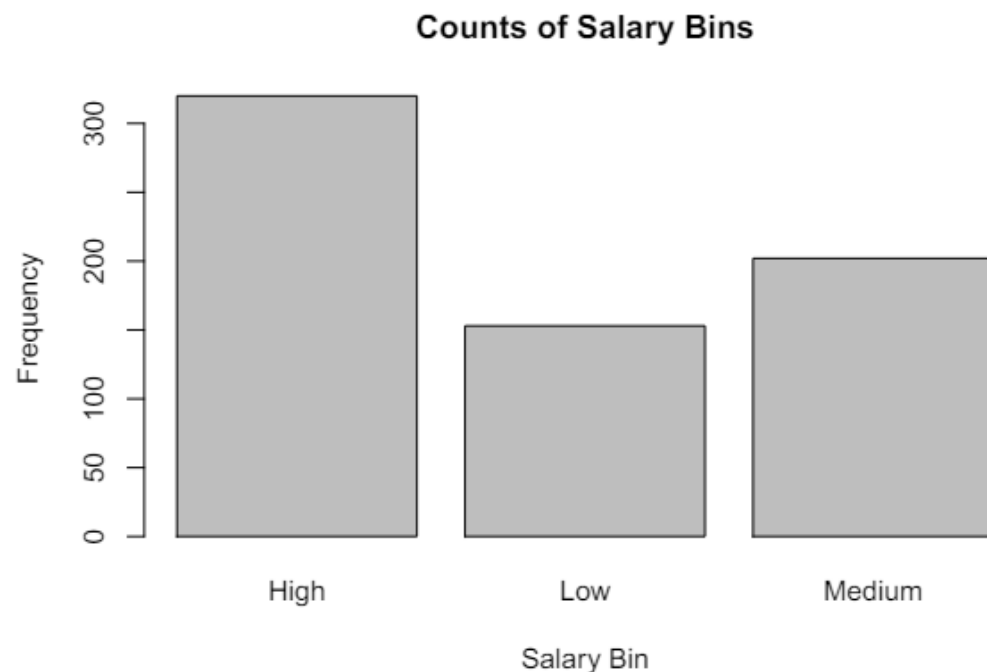
Using the data frame, create the visualizations as instructed.

First, create a barplot showing how many players are considered to be low, medium, or high.

EXERCISE 3

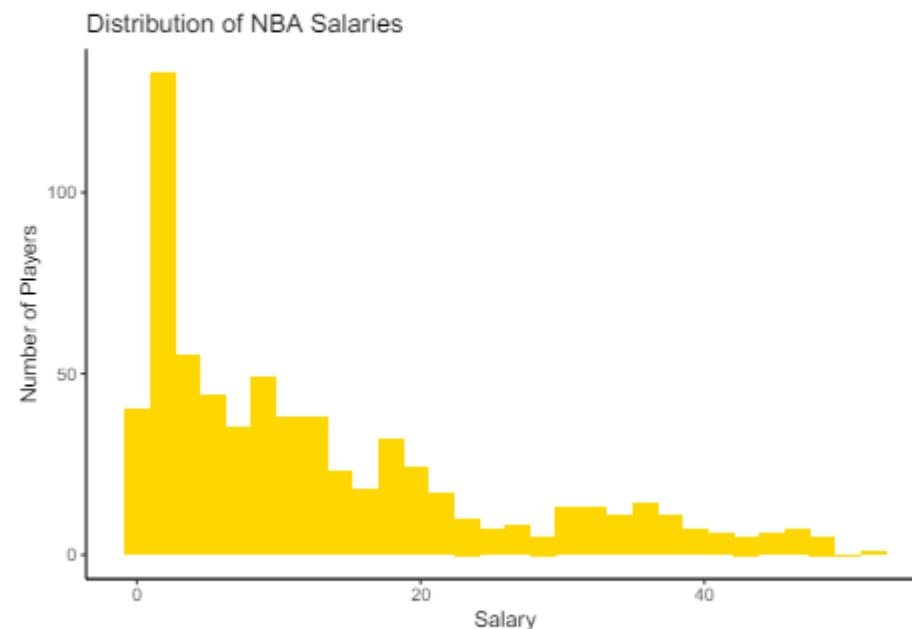
```
# Your Code Here
sal_bin_counts <- table(bball_clean$sal_bin)

barplot(sal_bin_counts,
        main = "Counts of Salary Bins",
        xlab = "Salary Bin",
        ylab = "Frequency")
```



Next, let's take a look at the distribution of NBA salaries to see if we have normality. Create a histogram representing the distribution of salaries in the NBA data.

```
# Your Code Here
ggplot(aes(x = `Salary $`, data = bball_clean)) +
  geom_histogram(fill = 'gold') +
  theme_classic() +
  labs(title = "Distribution of NBA Salaries",
       x = "Salary", y = "Number of Players")
```



Notice that there is a skew in the salaries, with a majority of players having a salary of approximately 2-6 million, with players with salaries upwards of 50 plus million. We need to keep this in account when we later on consider the validity conditions of our ordinal logistic regression model.

We are curious about looking further at the relationship between Field Goals made per game versus salary. This is something we will examine below.

Exercise 4: Visualize Points Per Game v. Salary

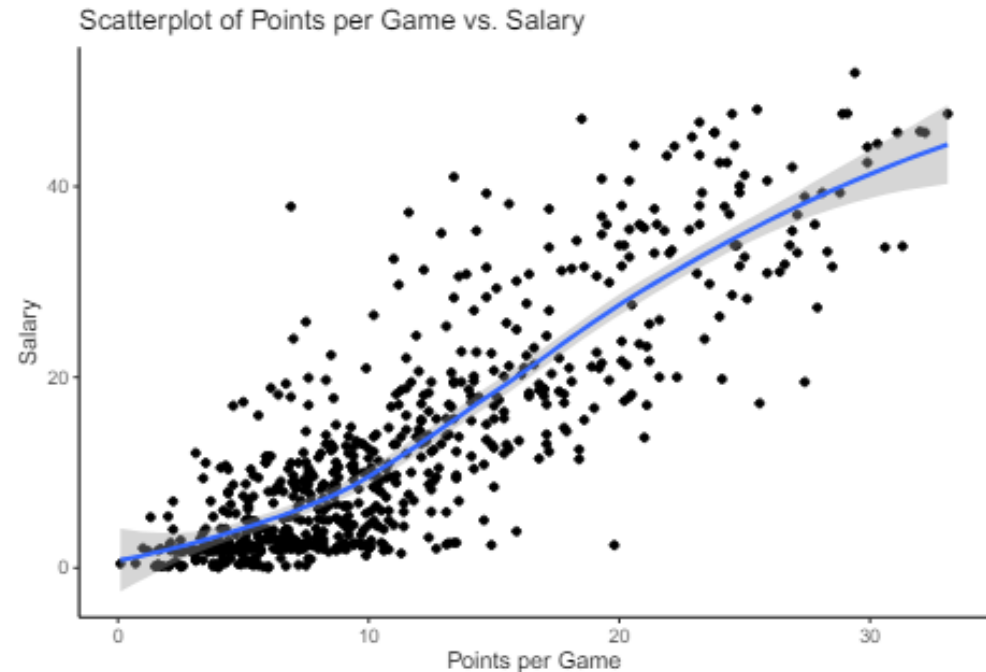
We are curious about looking further at the relationship between Field Goals made per game versus salary. This is something we will examine below.

Create a scatter plot showing the relationship between points per game and salary. Add a line to it using `geom_smooth()`.

EXERCISE 4

```
# Your Code Here
ggplot(data = bball_clean, aes(x = PTS_per_game, y = `Salary $`)) +
  geom_point() +
```

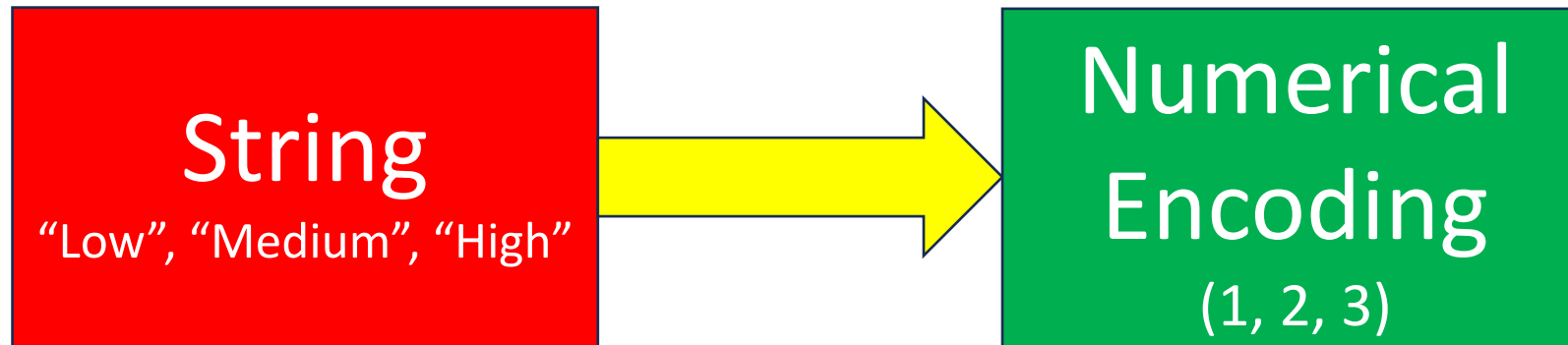
```
geom_smooth() +
labs(x = "Points per Game", y = "Salary") +
ggtitle("Scatterplot of Points per Game vs. Salary") +
theme_classic()
```



Based on the visualization above, what do you think `geom_smooth()` does?

Geom smooth adds a smooth mean line to a plot that is useful for visualizing the relationship between two variables.

Exercise 5: Turn Response into Factor



We now must turn the "low", "medium", and "high" values into factors (1,2,3) as this numerical encoding is needed for Ordinal Logistic Regression to work.

EXERCISE 5

```
# Turn the response (sal_bin) into a factor (1, 2, 3)
bball_clean$sal_bin <- factor(bball_clean$sal_bin,
                             levels = c("Low", "Medium", "High"), labels = c(1, 2, 3))
```


Exercise Interpretation

- **Coefficient Values:** the change in log-odds of being in a higher salary bin for a one-unit increase in the predictor, holding all other variables constant
 - i.e. “The more games played decreases the likelihood of being in a higher salary bin.”
- **Standard Error:** the level of uncertainty for the coefficient value
- **t-value:** ratio of the coefficient to its SE (larger means stronger evidence that the coefficient is 0)
 - i.e. “Although the ‘Games’ coefficient indicates ____, it’s not statistically significant.”
- **1|2:** the threshold between the first and second bins
 - Higher intercept values mean the log-odds of moving from a lower to a higher salary bin are higher
- **AIC:** evaluating how well a model fits the data
 - Lower is better

| | Value | Std. Error | t value | p value |
|--------------|------------|------------|-----------|-----------|
| 3P_per_game | 0.4171175 | 0.2263586 | 1.8427287 | 0.0653686 |
| STL_per_game | 0.8203322 | 0.3354755 | 2.4452822 | 0.0144739 |
| BLK_per_game | 0.6404837 | 0.3592316 | 1.7829269 | 0.0745982 |
| Age | 0.1601774 | 0.0296460 | 5.4030018 | 0.0000001 |
| PTS_per_game | 0.3767512 | 0.2037294 | 1.8492729 | 0.0644184 |
| PTS_totals | 0.0009817 | 0.0012976 | 0.7565778 | 0.4493029 |
| Position | 0.1478555 | 0.0885738 | 1.6692910 | 0.0950597 |
| 1 2 | 7.5413203 | 1.0822099 | 6.9684451 | 0.0000000 |
| 2 3 | 10.2369674 | 1.1335427 | 9.0309500 | 0.0000000 |

```
## Intercepts:
##      Value   Std. Error t value
## 1|2   7.5413    1.0822     6.9684
## 2|3  10.2370    1.1335     9.0309
##
## Residual Deviance: 831.2854
## AIC: 855.2854
```

Exercise Output Visualiztion

```
# Extract the coefficients table from the summary of model 'm'
ctable <- coef(summary(m))

# Calculate the p-values for the coefficients based on their t-values
# 'pnorm' is used for the normal distribution, 'lower.tail = FALSE' calculates the upper tail,
# and multiplying by 2 performs a two-tailed test
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2

# Add the p-values as a new column to the coefficients table
ctable <- cbind(ctable, "p value" = p)

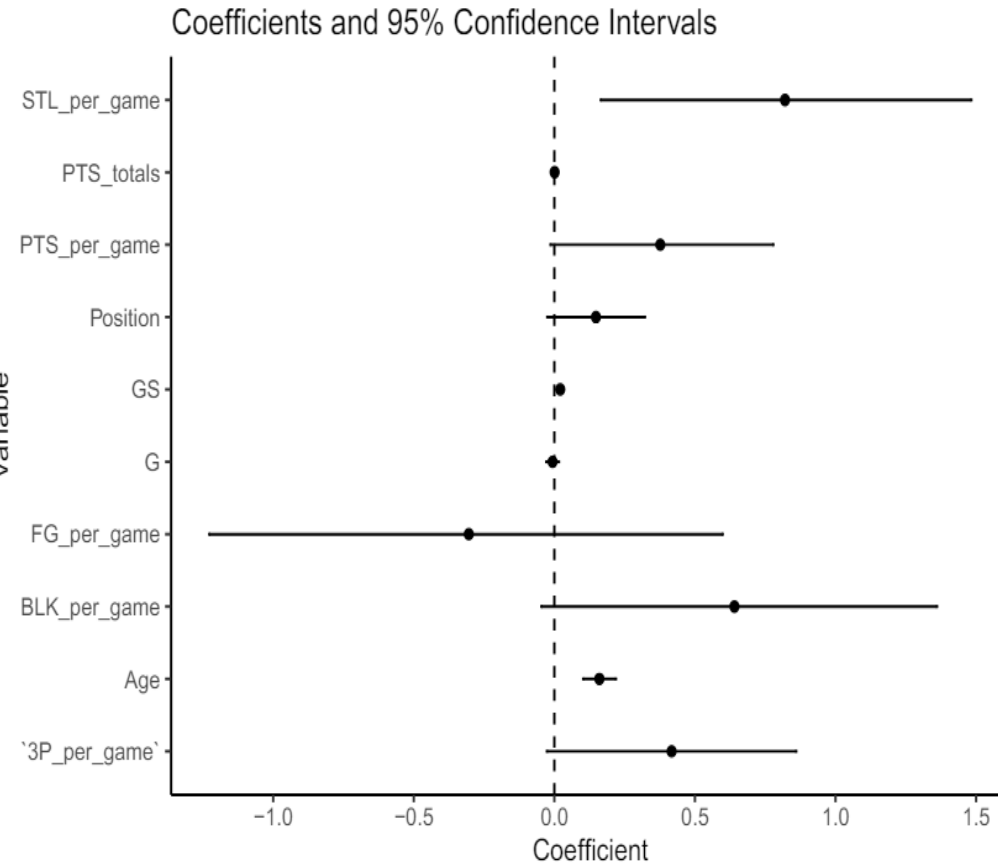
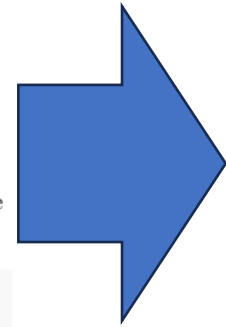
ci <- confint(m)

# Create a table from 'ctable' using 'kable()' for better formatting in markdown or HTML outputs
ctable %>% kable()
```

We notice that variables games started and age have statistically significant effects on on the ordinal response variable, as they have low p-values and their confidence intervals do not cross zero.

```
# Coefficients and confidence intervals
coef_df <- data.frame(
  Variable = rownames(ci),
  Coef = coef(m),
  CI_low = ci[,1],
  CI_high = ci[,2]
)

# Create forest plot
ggplot(coef_df, aes(x = Coef, y = Variable)) +
  geom_point() +
  geom_errorbarh(aes(xmin = CI_low, xmax = CI_high), height = 0) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "Coefficients and 95% Confidence Intervals",
       x = "Coefficient",
       y = "Variable") +
  theme_classic()
```





Exercise 6: Let's Apply Our Model To Individual Players!

We'll give you the stats, you guess the salary bin, then we check it with the model!

Let's See How You Do!



Lebron James

33-66 Binning

Low Salary: < 2.53 million

Middle Salary: 2.54 - 9.74 million

High salary: > 9.75 million

| Statistics | Lebron's Value |
|----------------------|----------------|
| Games Played | 55 |
| Games Started | 54 |
| Field Goals Per Game | 11.1 |
| Age | 38 |
| Points Per Game | 28.9 |



PLAYER 1: LeBron James

Your Code Here

First, make a data frame with only LeBron's information (within the clean data frame)

```
lebron_df <- bball_clean %>%  
  filter(Player == "LeBron James") %>%  
  dplyr::select(G, GS, FG_per_game, Age, PTS_per_game)
```

Output the table

```
lebron_df %>% kable()
```

| G | GS | FG_per_game | Age | PTS_per_game |
|----|----|-------------|-----|--------------|
| 55 | 54 | 11.1 | 38 | 28.9 |
| 56 | 56 | 11.4 | 37 | 30.3 |
| 45 | 45 | 9.4 | 36 | 25.0 |

Make your prediction below!

I predict LeBron will be in the high salary category.

Your Code Here

```
lebron_prediction <- predict(new_m, newdata = lebron_df, type = "class")  
lebron_prediction
```

```
## [1] 3 3 3  
## Levels: 1 2 3
```

Check Your Work

```
bball_clean %>% filter(Player == "LeBron James") %>%  
  dplyr::select(Player, sal_bin) %>% kable()
```

In the Code

Nicolas Batum

33-66 Binning

Low Salary: < 2.53 million

Middle Salary: 2.54 - 9.74 million

High salary: > 9.75 million

| Statistics | Lebron's Value |
|----------------------|----------------|
| Games Played | 78 |
| Games Started | 19 |
| Field Goals Per Game | 2.1 |
| Age | 34 |
| Points Per Game | 6.1 |



Justin Jackson

33-66 Binning

Low Salary: < 2.53 million

Middle Salary: 2.54 - 9.74 million

High salary: > 9.75 million

| Statistics | Lebron's Value |
|----------------------|----------------|
| Games Played | 7 |
| Games Started | 0 |
| Field Goals Per Game | 0.7 |
| Age | 26 |
| Points Per Game | 2.1 |



Seth Curry

33-66 Binning

Low Salary: < 2.53 million

Middle Salary: 2.54 - 9.74 million

High salary: > 9.75 million

| Statistics | Lebron's Value |
|----------------------|----------------|
| Games Played | 61 |
| Games Started | 7 |
| Field Goals Per Game | 3.4 |
| Age | 32 |
| Points Per Game | 9.2 |



Conclusion

The module introduced Ordinal Logistic Regression to predict NBA Salary Bin.

We saw that Ordinal Logistic Regression relied on NBA statistic data and assumptions of proportional odds, independent observations, and linearity of logit to help us accurately predict NBA salary bin (low, medium, or high).

Don't think this approach could only be used for predicting NBA salary bin. Other applications include

- “What fast food order at McDonalds is most likely to result in the customer ordering a “small,” “medium,” or “large” drink?
- What offensive line configurations are more likely to result in more running yards?
 - If you're starting on the 20 yd line, you Bin it at the 50 and the opposite 20 yd line.

