Amateur Hockey League Rankings

Mike Martz, Colorado State University Reyn Oyadomori, Colorado State University

Abstract

The Nashville Predators National Hockey League (NHL) franchise is looking for more efficient ways to identify amateur talent that will succeed at the NHL level. This statistical analysis examines logistic regression models, zero-inflated negative binomial models, and a deep learning model and how they can be used to analyze the NHL Entry Draft data to predict the success of a drafted amateur player at the NHL level. Since most drafted players never play a single game in the NHL, the zero-inflated negative binomial models are able to incorporate over-dispersion and handle the excess zeros in the NHL Entry Draft data. The deep learning model exhibited promising results in identifying different tiers of success at the NHL level based on a player's amateur stats.

Introduction

Background

The National Hockey League (NHL) Entry Draft is the annual meeting where NHL teams select the rights to amateur hockey players. Over the last 10 years, players have been drafted out of many different amateur leagues with varying levels of competition. An amateur hockey player's skill level is difficult to project, as evidenced by a large proportion of drafted players never playing a game in the NHL. The Nashville Predators are looking to identify more effective ways to evaluate amateur players that would potentially be selected in the NHL Entry Draft and improve their draft results by selecting more amateur players that succeed at the NHL level.

Objectives

Based on the last 10 amateur drafts, what statistical analyses can be used to answer the following questions?

1. Which amateur leagues appear to produce the most NHL players at the highest conversion rates?
2. Do certain amateur leagues produce better NHL players for certain positions?
3. What weight should be put on each player's stats based on their league?
4. Based on a player's stats and what amateur league the player is drafted from, is there a way to predict whether a drafted player will be successful at the NHL level?

Dalton Linkus, Research & Development Data Engineer with the Nashville Predators gathered a dataset on the 2,125 players drafted in the NHL Entry Draft from 2010-2019. The information includes each individual player's amateur league statistics, as well as their statistics in the ECHL (formerly the East Coast Hockey League), the American Hockey League (AHL), and the NHL (Linkus 2020).

Summary Statistics and Summary Figures

From 2010-2019, 2,125 amatuer players were drafted from 52 amateur leagues around the world. Since there are some leagues with minimal data/players drafted, we focused on amateur leagues with at least 8 players drafted. This is because highly touted prospects look to play in more popular amateur leagues to increase player exposure to scouts. With the 8 drafted player criteria, our analysis focused on 24 amateur leagues and 2,043 total drafted players.

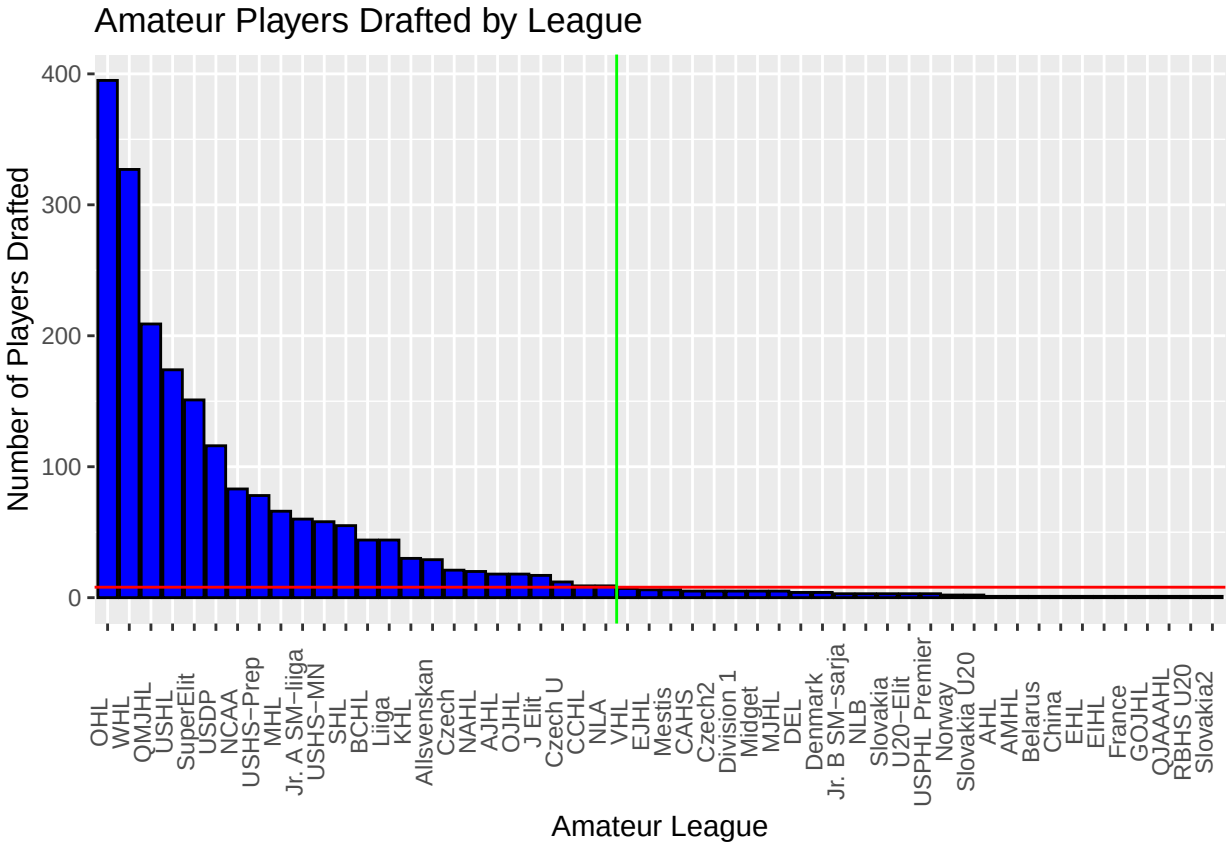


Figure 1 shows the number of players drafted out of each of the 52 amateur leagues from 2010-2019.

We examined the 2,043 players in the 24 leagues with more than 8 drafted players, the breakdown of the number of players drafted, and the success rate of players who played at least 1 game in the NHL.

League	Drafted Players	Success Pct	League	Drafted Players	Success Pct
AJHL	18	22%	NCAA	83	51%
Allsvenskan	29	41%	NLA	9	67%
BCHL	44	18%	OHL	395	45%
CCHL	9	33%	OJHL	18	33%
Czech	21	38%	QMJHL	209	35%
Czech U	12	17%	SHL	55	62%
J Elit	17	6%	SuperElit	151	28%
Jr. A SM-liiga	60	23%	USDP	116	50%
KHL	30	43%	USHL	174	29%
Liiga	44	59%	USHS-MN	58	22%
MHL	66	20%	USHS-Prep	78	17%
NAHL	20	10%	WHL	327	39%

Table 1 shows the number of players drafted out of each of the 24 amateur leagues between 2010 and 2019 with at least 8 players drafted and their NHL successful conversion rate.

Methods/Analysis

To determine whether there are amateur leagues that appear to produce more successful NHL players, we investigated a logistic regression model using the amateur league dataset by league. This would help to determine if there are any leagues that influence the probability of a drafted player playing in the NHL. Second, we used the same logistic regression model ideas and determined which player variables were most influential toward the probability a drafted player played at least 1 game in the NHL. The results from the logistic models helped us focus on specific variables when we leveraged the next set of models.

Between 2010 and 2019, over 63% of the players drafted from the 24 amateur leagues we analyzed did not play a single game in the NHL (missing number of NHL games played was treated as zero games played). Figure 2 below, outlined that the dataset has an excessive amount of failures (drafted players that played zero NHL games).

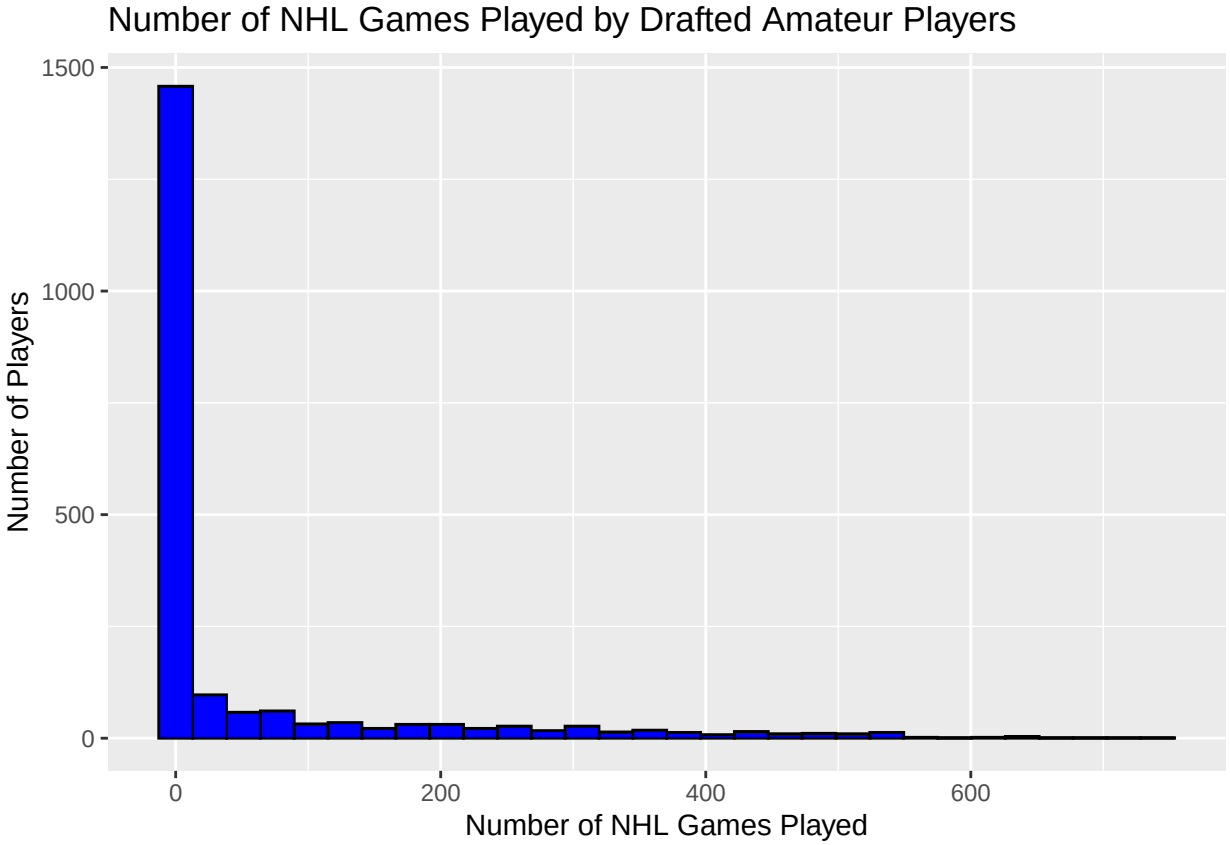


Figure 2 shows the distribution of the number of NHL games played by drafted amateur players in the 24 leagues that drafted more than 8 players between 2010 and 2019.

In order to deal with data that contains excessive failures more effectifvely, we modeled the number of games played at the NHL with a zero-inflated negative binomial (ZINB) regression model (Zeileis et al 2008). This model helped us identify which variables are significant in determining the success and the number of games played at the NHL level.

There are two parts to a ZINB regression model. The first part will represent the probability the drafted hockey player will not play at least one game in the NHL. The second part will represent the total number games played at the NHL level modeled by a generalized linear negative binomial regression model. We then identified the most relevant variables that have the most influence on the number of NHL games played.

We fit a deep learning dense neural network to model various levels of 'NHL success' for drafted players. A categorical variable was created to represent varying degrees of success at the NHL level. A drafted player

could fall into one of five categories, 0-25 NHL games played, 26-50 NHL games played, 51-75 NHL games played, 76-100 NHL games played, and more than 100 NHL games played.

The draft dataset provided was separated into a training dataset, a validation dataset, and a testing dataset. The dense neural network conssited of four sequential dense layers, the first layer had 256 nodes or units, the second and third layers had 64 units, and the final layer had five units, the same number of categories the model classified data into. The first four layers, was followed with dropout, a method used to decrease the effect of model overfitting (Chollet and Allaire 2018). The model was compiled and run for 50 consectutive iterations and the best model measured by validation accuracy was saved and used to predict categories of 'unseen' test data.

Results and Conclusions/Discussion

We ran a logistical model on the 2,043 players in the 24 leagues with more than 8 drafted players. There are a total of 7 leagues with significant results, meaning there is evidence to suggest the 7 leagues have influence on the probability of an amateur player playing at least one game in the NHL (Appendix D1).

League	Drafted Players	Success Pct	League	Drafted Players	Success Pct
AJHL	*18*	*22%*	*NCAA*	*83*	*51%*
Allsvenskan	29	41%	*NLA*	*9*	*67%*
BCHL	44	18%	*OHL*	*395*	*45%*
CCHL	9	33%	OJHL	18	33%
Czech	21	38%	QMJHL	209	35%
Czech U	12	17%	*SHL*	*55*	*62%*
J Elit	17	6%	SuperElit	151	28%
Jr. A SM-liiga	60	23%	*USDP*	*116*	*50%*
KHL	30	43%	USHL	174	29%
Liiga	*44*	*59%*	USHS-MN	58	22%
MHL	66	20%	USHS-Prep	78	17%
NAHL	20	10%	WHL	327	39%

Table 2 shows the 7 leagues in *blue italics* out of the 24 amateur leagues analyzed that reported as significant influencers in a drafted player's chances of making it to the NHL.

The remaining 17 leagues do not have evidence to suggest that their amatuer league influences the probability a drafted player plays in the NHL. This implies these 17 amateur leagues are independent of a drafted player's chances of making it to the NHL.

We then looked at the draft data by position. We experimented with many logistical and ZINB models to determine the variables that influenced a drafted player's chances to play in the NHL and the number of games the player will play. The significant variables from the logisitcal models helped to narrow down which variables to focus on when picking which ZINB model was most effective. We used rootograms to verify the ZINB model's fit and effectiveness (Kleiber and Zeileis 2016). The most significant variables to influence a drafted player's NHL success are broken down by position in Table 3 below.

Position	Influence Draft Failure	Relationship	Influence NHL Games Played	Relationship
Defensemen	Amateur Assists	Inverse	Amateur PPG	Direct
Forward	Amateur Seasons Amateur Assists	Direct Inverse	Amateur Games Played Amateur Goals	Inverse Direct
Goalies	Amateur Seasons	Not Significant	Amateur Seasons	Inverse

Table 3 outlines the variables and their relationship to NHL success that influence each part of the ZINB model by each position.

Amateur assists are highly influential in whether the drafted player succeeds/fails to play in an NHL game (Appendix D3, D5). The more assists a drafted player has in their amateur league, the less likely that player

is to fail to play in an NHL game. This could imply that a key element that influences a drafted player's success is their ability to collaborate and work with the rest of their team.

Points per game is the most influential on how many NHL games a drafted defensemen would potentially play (Appendix D3). A defensemen's primary duty is to limit the number of goals an opponent scores, but the quality of the goalie and overall team strategy make evaluation of their counting and rate statistics potentially inconsistent. For example, a poor quality goalie or a prolific offense will negatively impact a defensemen's counting and rate statistics. A poor quality goalie will cause an uptick in the number of goals the opponent scores and a prolific offense lessens the need to be stingy on defense.

Amateur goals is a straightfoward indicator for how many NHL games a forward will play (Appendix D5). A forward's primary duty is to score goals, so being able to identify a prolific scorer and their scoring efficiency, as seen in the negative relationship with Amateur Games Played, are qualities that help predict a forward's success.

Because goalie is a specialized position, there were only 150 goalies in our analysis. The lack of sample size made it difficult to identify any variable that would help to predict a drafted goalie's NHL success (Appendix D6, D7).

The deep learning model created to predict drafted player's NHL success appears to stagnate and reach optimum modeling relatively rapidly (by approximate iteration five). This indicates the model performance levels off after 5-10 iterations. The model predicted NHL success category on unseen test data with about 74% accuracy as outlined in the bottom graph of Figure 3.

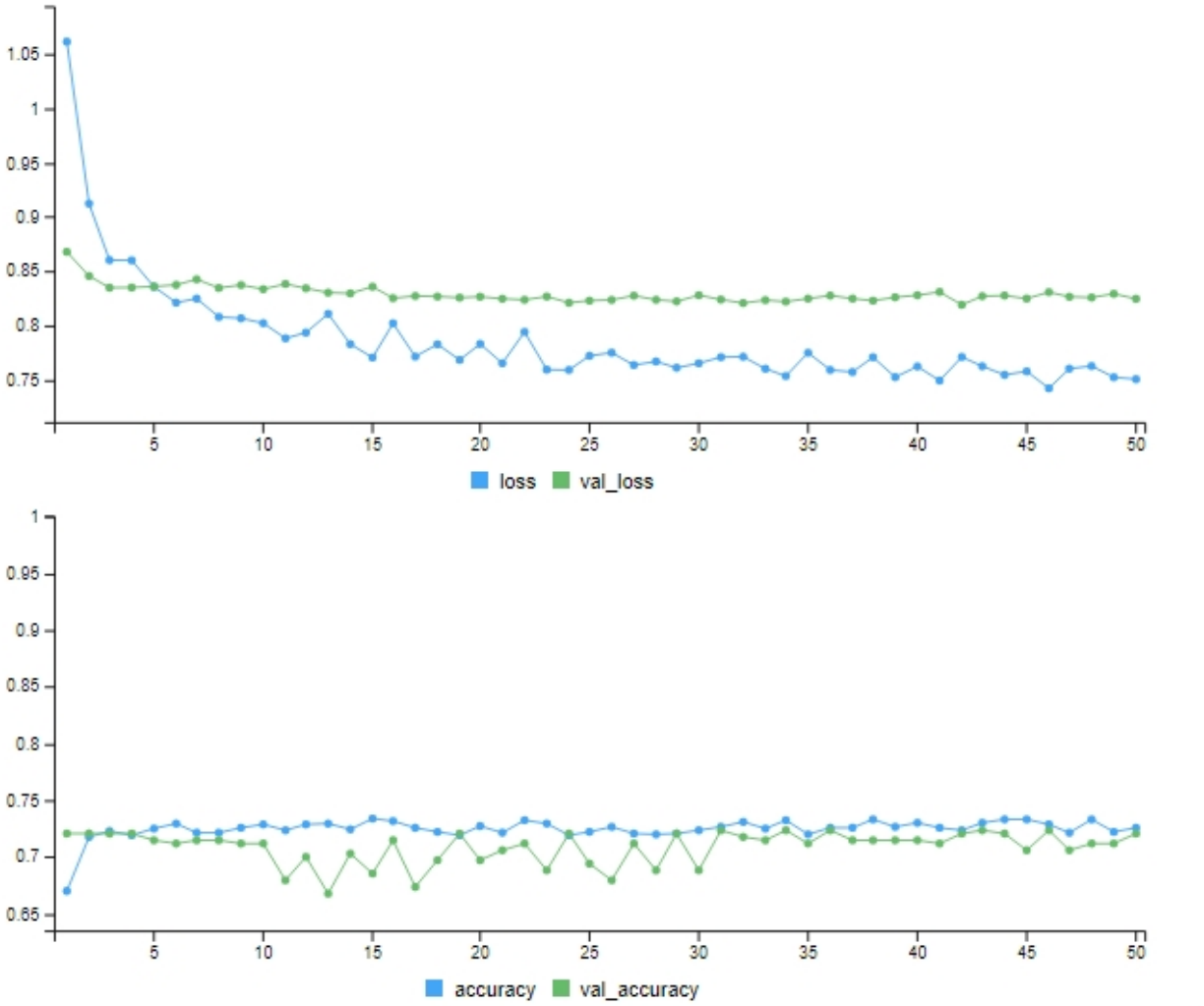


Figure 3 shows the results of the deep learning model by iteration as measured by a loss function score for training and validation sets (top graph) and training and validation accuracy (bottom graph).

In order to improve the logistic, ZINB, and deep learning models, we recommend gathering more amateur player data. Instead of the focus being only the amateur players who were drafted, include the data from all amateur players in the dataset. This is especially true for goalies, since their smaller sample size makes it more difficult to identify variables that influence NHL success.

Authors' Statements

Both authors contributed to the statistical analysis and writing for this report. Mike did the programming of the models and deep learning algorithm, while Reyn completed the summary statistics plots. Reyn developed the initial draft of the report, while Mike provided significant feedback/edits for the final product.

References

Abbas Moghimbeigi, Mohammed Reza Eshraghian, Kazem Mohammad, Brian Mcardle (2008) Multilevel zero-inflated negative binomial regression modeling for over-dispersed count data with extra

zeros, Journal of Applied Statistics, 35:10, 1193-1202, DOI: 10.1080/02664760802273203

Chollet F, Allaire JJ. 2018. Deep Learning with R. Shelter Island (NY): Manning Publications Co.

Chollet, F and others. 2015. Keras. (https://keras.io).

Christian Kleiber & Achim Zeileis (2016) Visualizing Count Data Regressions Using Rootograms, The American Statistician, 70:3, 296-303, DOI: 10.1080/00031305.2016.1173590

Linkus, Dalton. 2020. Amateur Hockey League Rankings Project (and associated data). Presented to STAA 556: Statistical Consulting, Colorado State University, May 21, 2020.

Fang, Rui. 2008. Zero-Inflated Negative Binomial Regression Model for Over-Dispersed Count Data with Excess Zeros and Repeated Measures, An Application to Human Microbiota Sequence Data. University of Colorado. April 19, 2013.

Zero Inflated Negative Binomial Regression. c2020 University of California. Accessed: June 4, 2020. https://stats.idre.ucla.edu/r/dae/zinb/.

Zeileis A, Kleiber C, Jackman S. 2008. Regression Models for Count Data in R. Journal of Statistical Software. 27(8):1-25.(https://www.jstatsoft.org/article/view/v027i08).

Appendix/Supplemental Information

Appendix A - Data Dictionary

- **DraftID:** Unique identification for each draft pick. First 4 digits are the year of the draft, last three digits indicate the players overall selection number (e.g. First pick of the 2015 draft DraftID = 2015001)
- **Year:** Year player was drafted
- **Round:** Round of draft a player was selected (each draft has seven rounds)
- **Overall:** Players overall selection number in a given draft
- **Team:** NHL team player was drafted from
- **Player:** Player drafted full name
- **PlayerID:** Players unique identification number
- **Pos:** Position group for each player
  - D: Defenseman
  - F: Forward
  - G: Goalie
- **League:** Amateur league a player was drafted out of
- **(League).First.Season:** First season played in a specific league
- **(League).First.Season.10GP:** First season in which a player played in 10 or more games in a specific league
- **(League).Seasons:** Number of seasons played in a specific league
- **(League).GP:** Number of games played in specific league
- **(League).G:** Number of goals scored in a specific league
- **(League).A:** Number of assists accumulated in a specific league
- **(League).Pt:** Number of points accumulated in a specific league
  - Points are the sum of goals and assists
- **(League).PPG:** Points per game in a specific league
  - PPG = Pt/GP
- **(League).PIM:** Penalty minutes a player commits in a specific league
- **(League).+/-:** Plus/Minus accumulated in a specific league
  - A player receives a plus (1) when they are on the ice for a goal scored by their team
  - A player receives a minus (-1) when theyare on the ice for a goal scored against their team
- **(League).GAA:** Goals against average in a specific league (Only applies to goalies)
  - GAA = (# of goals allowed) * 60 / (Minutes Played)
  - GAA is a per 60 stat (60 minutes in regulation game)
- **(League).SV%:** Save percentage in a specific league (Only apples to goalies)
  - SV% = (Total Saves) / (Total Shots)

Appendix B - Figures and Tables

Figure 1 shows the number of players drafted out of each of the 52 amateur leagues from 2010-2019.

```
# Figure 1
# Amateur Players Drafted by League (all)
league_count <- draft_data %>%
  count(League) %>%
  arrange(n)
ggplot(league_count, aes(x = reorder(League, -n), y = n)) +
  geom_col(color = "black", fill = "blue") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0)) +
  geom_hline(yintercept = 8, color = "red") +
  geom_vline(xintercept = 24.5, color = "green") +
  ggtitle("Amateur Players Drafted by League")+
  xlab("Amateur League") +
  ylab("Number of Players Drafted")
```

Table 1 shows the number of players drafted out of each of the 24 amateur leagues between 2010 and 2019 with at least 8 players drafted and their NHL successful conversion rate.

```
# Table 1
# Table with League, number of players drafted, and conversion percentage
draft_tab_dat_join <- left_join(draft_tab_dat, league_count, by = "League")
draft_tab_dat_clean <- draft_tab_dat_join[draft_tab_dat_join$n >= 8,]
draft_tab_dat_edit <- draft_tab_dat_clean
draft_tab_dat_edit$NHL_gpgt0 <- as.factor(draft_tab_dat_edit$NHL_gpgt0)
draft_tab_dat_edit$Pct <- label_percent()(draft_tab_dat_edit$count / draft_tab_dat_edit$n)
draft_tab_dat_edit[seq(2,48,2), c(1,4,5)]
```

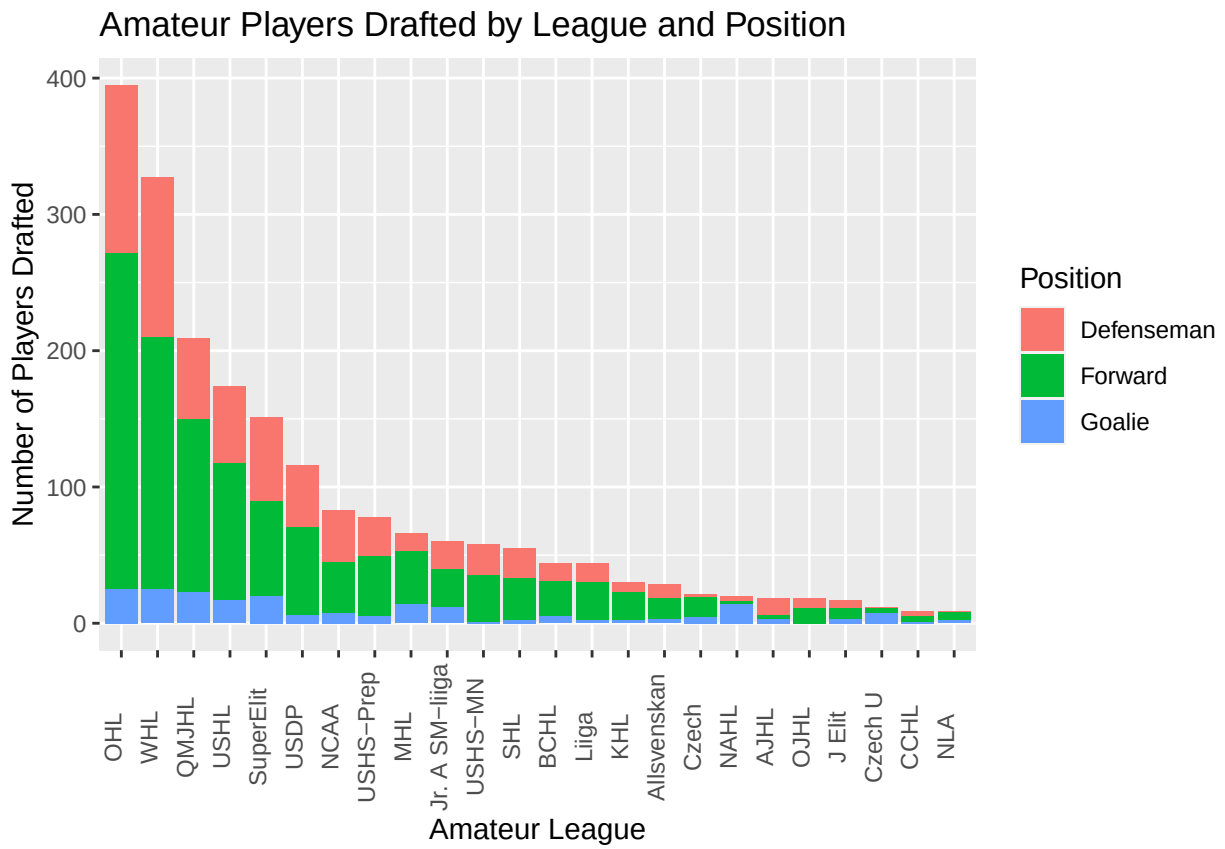
```
# Figure 2
# Distribution of the number of games played in the NHL
games_played <- left_join(nhl_gp_0, league_count, by = "League")
games_played_filter <- games_played[games_played$n >= 8,]
ggplot(games_played_filter, aes(x = NHL.GP)) + geom_histogram()
```

Table 2 shows the 7 leagues in *blue italics* out of the 24 amateur leagues analyzed that reported as significant influencers in a drafted player's chances of making it to the NHL.

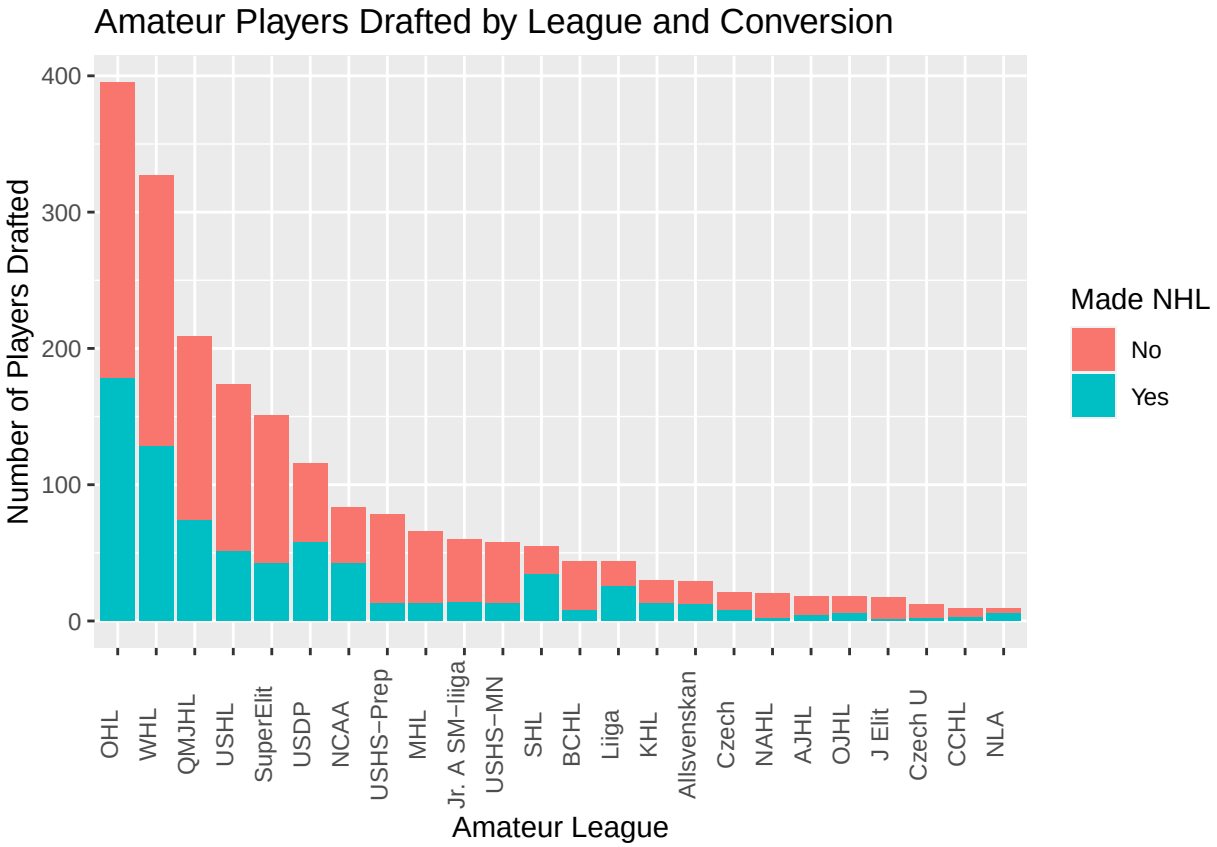
```
# Table 2
mod <- glm(NHL_gpgt0 ~ League, family = binomial(link = "logit"),
           data = player_gt8)
log.league <- summary(mod)
sig.coef <- c(1, 10, 13, 14, 15, 18, 20)

# significant coefficients (log odds)
log.league$coefficients[sig.coef]

# odds
odds <- exp(log.league$coefficients[sig.coef] + log.league$coefficients[1])
odds[1] <- exp(log.league$coefficients[1])
odds / (1 + odds)
```

A summary of the breakdown of the amateur players that were drafted out of the 24 leagues we are investigating.

```
# # Amateur Players Drafted by League and Position (filtered)
ggplot(draft_hist_dat_clean, aes(x = reorder(League,-n), y = count, fill = Pos)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0)) +
  ggtitle("Amateur Players Drafted by League and Position")+
  xlab("Amateur League") +
  ylab("Number of Players Drafted") +
  scale_fill_discrete(name = "Position", labels = c("Defenseman", "Forward", "Goalie"))
```




```
# Amateur Players Drafted by League and Conversion (filtered)
ggplot(draft_hist_dat_clean, aes(x = reorder(League,-n), y = count, fill = NHL_gpgt0)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0)) +
  ggtitle("Amateur Players Drafted by League and Conversion")+
  xlab("Amateur League") +
  ylab("Number of Players Drafted")+
  scale_fill_discrete(name = "Made NHL", labels = c("No", "Yes"))
```



Appendix C - Data Sets

C1 - Overall Data Sets

```
# All draft dataset with zero replacing NA, and adding flags for games played in NHL
nhl_gp_0 <- draft_data %>%
  mutate(
    NHL.Seasons = coalesce(NHL.Seasons, 0),
    NHL.GP = coalesce(NHL.GP, 0),
    NHL.G = coalesce(NHL.G, 0),
    NHL.A = coalesce(NHL.A, 0),
    NHL.Pt = coalesce(NHL.Pt, 0),
    NHL.PPG = coalesce(NHL.PPG, 0),
    `NHL.+/-` = coalesce(`NHL.+/-`, 0),
    NHL.GAA = coalesce(NHL.GAA, 0),
    `NHL.SV%` = coalesce(`NHL.SV%`, 0),
    NHL_gpgt0 = ifelse(NHL.GP > 0, 1, 0),
    NHL_gpgt10 = ifelse(NHL.GP > 10, 1, 0),
    NHL_gpgt25 = ifelse(NHL.GP > 25, 1, 0),
    NHL_gpgt50 = ifelse(NHL.GP > 50, 1, 0),
    NHL_gpgt100 = ifelse(NHL.GP > 100, 1, 0)) %>%
  select(DraftID, Year, Round, Overall,
         PlayerID, Pos, League,
         Amateur.Seasons:`Amateur.SV%`,
```

```
        NHL.Seasons:`NHL.SV%`,
        NHL_gpgt0:NHL_gpgt100)
```

```
# Data for position and NHL conversion plots and filtering for >= 8 players
draft_hist_dat <- nhl_gp_0 %>%
  group_by(League, Pos, NHL_gpgt0) %>%
  summarise(count = n())

draft_hist_dat_join <- left_join(draft_hist_dat, league_count, by = "League")
draft_hist_dat_clean <- draft_hist_dat_join[draft_hist_dat_join$n >= 8,]
draft_hist_dat_clean$NHL_gpgt0 <- as.factor(draft_hist_dat_clean$NHL_gpgt0)
```

```
#leagues greater than 8
player_gt8 <- nhl_gp_0 %>%
  group_by(League) %>%
  filter(n() >= 8) %>%
  select(Year, League,
        NHL.GP, NHL_gpgt0:NHL_gpgt100)
```

C2 - Position Data Sets

```
#make defensemen dataset
d_gt8 <- nhl_gp_0 %>%
  group_by(League, Pos) %>%
  filter(n() >= 8,
        Pos == "D") %>%
  select(Year, League,
        Amateur.Seasons:`Amateur.+/-`,
        NHL.GP, NHL_gpgt0:NHL_gpgt100)
```

```
#make forwards dataset
f_gt8 <- nhl_gp_0 %>%
  group_by(League, Pos) %>%
  filter(n() >= 8,
        Pos == "F") %>%
  select(Year, League,
        Amateur.Seasons:`Amateur.+/-`,
        NHL.GP, NHL_gpgt0:NHL_gpgt100)
```

```
#make goalie dataset
g_gt8 <- nhl_gp_0 %>%
  group_by(League, Pos) %>%
  filter(n() >= 8,
        Pos == "G") %>%
  mutate(
    Amateur_gps = Amateur.GP / Amateur.Seasons
  ) %>%
  select(Year, League, Amateur.Seasons,
        Amateur.GP, Amateur_gps, Amateur.GAA, `Amateur.SV%`,
        NHL.GP, NHL_gpgt0:NHL_gpgt100)
```

Appendix D - Models

D1 - Logistic Regression Model - Leagues

```
#logistic league effect > 8
mod <- glm(NHL_gpgt0 ~ League, family = binomial(link = "logit"),
           data = player_gt8)
summary(mod)
```

```
##
## Call:
## glm(formula = NHL_gpgt0 ~ League, family = binomial(link = "logit"),
##     data = player_gt8)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4823  -0.9967  -0.7290   1.2626   2.3804
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.25276    0.56695  -2.210   0.0271 *
## LeagueAllsvenskan     0.90446    0.68087   1.328   0.1841
## LeagueBCHL           -0.25131    0.68863  -0.365   0.7151
## LeagueCCHL            0.55962    0.90633   0.617   0.5369
## LeagueCzech           0.76726    0.72343   1.061   0.2889
## LeagueCzech U        -0.35667    0.95991  -0.372   0.7102
## LeagueJ Elit         -1.51983    1.17639  -1.292   0.1964
## LeagueJr. A SM-liiga  0.06318    0.64389   0.098   0.9218
## LeagueKHL             0.98450    0.67615   1.456   0.1454
## LeagueLiiga           1.62049    0.64455   2.514   0.0119 *
## LeagueMHL            -0.15258    0.64593  -0.236   0.8133
## LeagueNAHL           -0.94446    0.93647  -1.009   0.3132
## LeagueNCAA            1.27686    0.60797   2.100   0.0357 *
## LeagueNLA             1.94591    0.90633   2.147   0.0318 *
## LeagueOHL             1.05465    0.57589   1.831   0.0671 .
## LeagueOJHL            0.55962    0.75593   0.740   0.4591
## LeagueQMJHL           0.65155    0.58511   1.114   0.2655
## LeagueSHL             1.73460    0.63124   2.748   0.0060 **
## LeagueSuperElit       0.29908    0.59533   0.502   0.6154
## LeagueUSDP            1.25276    0.59658   2.100   0.0357 *
## LeagueUSHL            0.37240    0.59090   0.630   0.5285
## LeagueUSHS-MN         0.01105    0.64852   0.017   0.9864
## LeagueUSHS-Prep      -0.35667    0.64322  -0.555   0.5792
## LeagueWHL             0.81149    0.57816   1.404   0.1604
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2687.2  on 2042  degrees of freedom
## Residual deviance: 2559.8  on 2019  degrees of freedom
## AIC: 2607.8
##
```

```
## Number of Fisher Scoring iterations: 5
```

```
# if in league, your odds of playing in NHL will increase exp(coef) times over playing in one of the
```

D2 - Logistic Regression Model - Defensemen

```
# Defensemen Logistic Regression (filtered)
p_70 <- round(0.7 * nrow(d_gt8))
set.seed(13)
index <- sample(nrow(d_gt8), p_70)
x_train <- d_gt8[index, ]
x_test <- d_gt8[-index, ]

mod <- glm(NHL_gpgt0 ~ Year + Amateur.A,
           family = binomial(link = "logit"),
           data = d_gt8)
summary(mod)
```

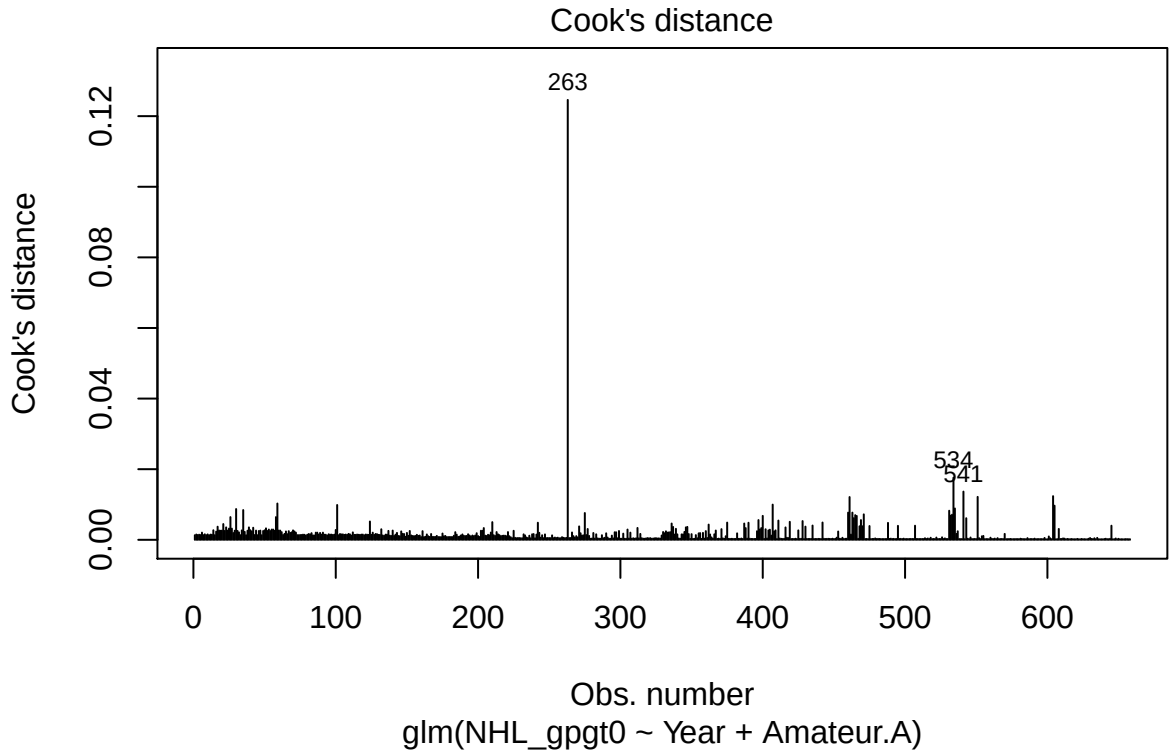
```
##
## Call:
## glm(formula = NHL_gpgt0 ~ Year + Amateur.A, family = binomial(link = "logit"),
##     data = d_gt8)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9021  -0.8888  -0.6078   1.1723   2.1282
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) 517.721897  66.172142   7.824 5.12e-15 ***
## Year         -0.257563   0.032869  -7.836 4.65e-15 ***
## Amateur.A     0.011851   0.003936   3.011  0.00261 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 837.19  on 657  degrees of freedom
## Residual deviance: 764.06  on 655  degrees of freedom
## AIC: 770.06
##
## Number of Fisher Scoring iterations: 3
```

```
anova(mod, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: NHL_gpgt0
```

```
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        657     837.19
## Year       1   63.873       656     773.32 1.327e-15 ***
## Amateur.A  1    9.257       655     764.06  0.002346 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Influential Points
plot(mod, which = 4, id.n = 3)
```




```
#extract points
model.data <- augment(mod) %>%
  mutate(index = 1:n())
model.data %>%
  top_n(3, .cooksd)
```

```
## # A tibble: 3 x 11
##   NHL_gpgt0  Year Amateur.A .fitted .se.fit .resid   .hat .sigma .cooksd
##       <dbl> <dbl>     <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>   <dbl>
## 1         0  2013       201   1.63    0.684  -1.90 0.0641   1.08  0.125
## 2         1  2018       110  -0.736   0.332   1.50 0.0241   1.08  0.0176
```

```
## 3         1  2018        94  -0.926   0.278   1.59 0.0157   1.08  0.0137
## # ... with 2 more variables: .std.resid <dbl>, index <int>
```

```
#the standardized residuals (.std.resid) are not > 3
ggplot(model.data, aes(index, .std.resid)) +
  geom_point(aes(color = NHL_gpgt0), alpha = .5) +
  theme_bw() +
  xlab("Index") + ylab("Standard Residuals") +
  ggtitle("Standardized Residuals - Defensemen")
```


```
#check for multicollinearity using the car package
car::vif(mod)
```

```
##      Year Amateur.A
##  1.033675  1.033675
```

```
# Diagnostics on classfication success
# Check confusion matrix from caret
glm_probs <- glm(NHL_gpgt0 ~ Year + Amateur.A,
                 family = binomial(link = "logit"),
                 data = x_train) %>%
  predict(newdata = x_test, type = "response")
glm_pred <- ifelse(glm_probs > 0.5, 1, 0)
g_pred <- ifelse(glm_pred == 1, "Yes", "No")
x_preds <- tibble(x_test, g_pred) %>%
```

```
  mutate(x_truth = ifelse(x_test$NHL_gpgt0 == 1, "Yes", "No"))

#caret confusionMatrix
confusionMatrix(as.factor(x_preds$g_pred),
                as.factor(x_preds$x_truth),
                positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  93  46
##        Yes 33  25
##
##                Accuracy : 0.599
##                  95% CI : (0.5269, 0.668)
##     No Information Rate : 0.6396
##     P-Value [Acc > NIR] : 0.8958
##
##                   Kappa : 0.094
##
##  Mcnemar's Test P-Value : 0.1770
##
##             Sensitivity : 0.3521
##             Specificity : 0.7381
##          Pos Pred Value : 0.4310
##          Neg Pred Value : 0.6691
##              Prevalence : 0.3604
##          Detection Rate : 0.1269
##    Detection Prevalence : 0.2944
##       Balanced Accuracy : 0.5451
##
##        'Positive' Class : Yes
##
```

D3 - Zero-Inflated Negative Binomial Model - Defensemen

```
# Zero inflated negative binomial - defensemen
library(countreg)
dg6 <- zeroinfl(NHL.GP ~
                  Amateur.PPG
                | Amateur.A,
                data = d_gt8, dist = "negbin",
                EM = TRUE)
summary(dg6)
```

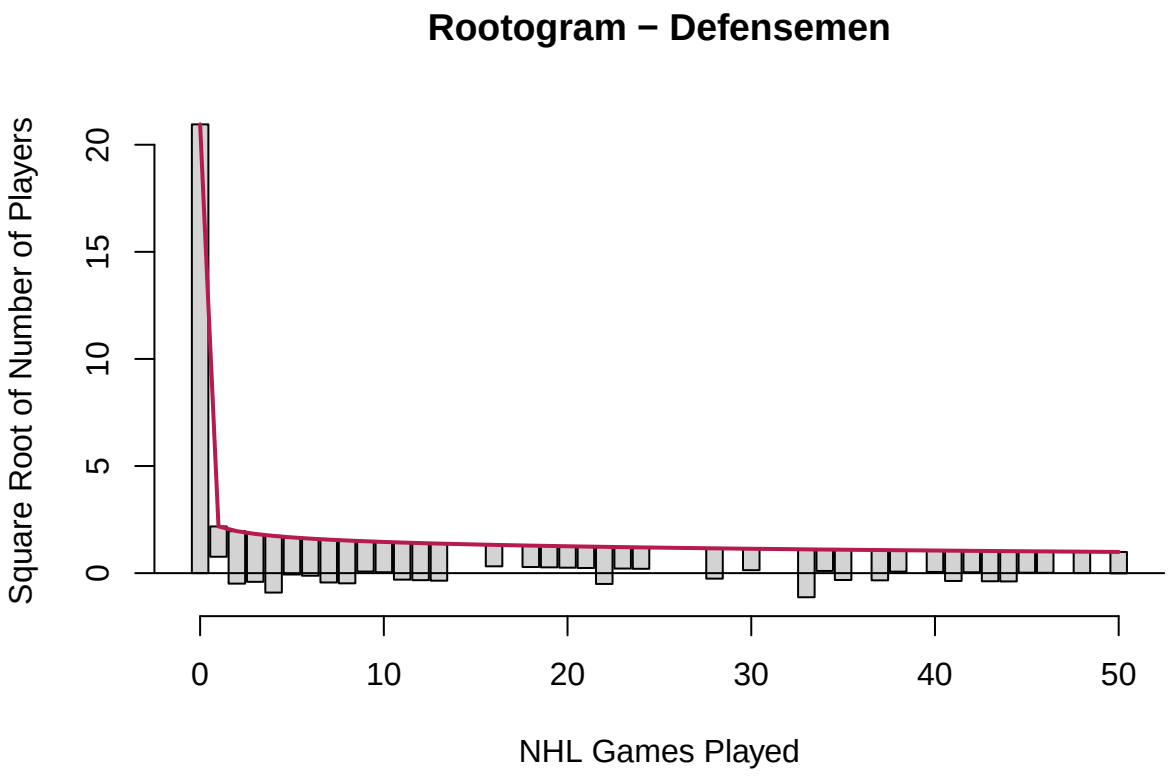
```
##
## Call:
## zeroinfl(formula = NHL.GP ~ Amateur.PPG | Amateur.A, data = d_gt8, dist = "negbin",
##     EM = TRUE)
##
```

```
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.5751 -0.3905 -0.3727 -0.2198  5.4580
##
## Count model coefficients (negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.6818     0.1928  24.278  < 2e-16 ***
## Amateur.PPG   0.6086     0.3751   1.622    0.105
## Log(theta)   -0.4695     0.1098  -4.276  1.9e-05 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.860986   0.141251   6.095 1.09e-09 ***
## Amateur.A   -0.007308   0.003715  -1.967   0.0492 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.6253
## Number of iterations in BFGS optimization: 1
## Log-likelihood: -1717 on 5 Df
```

```
AIC(dg6)
```

```
## [1] 3444.232
```

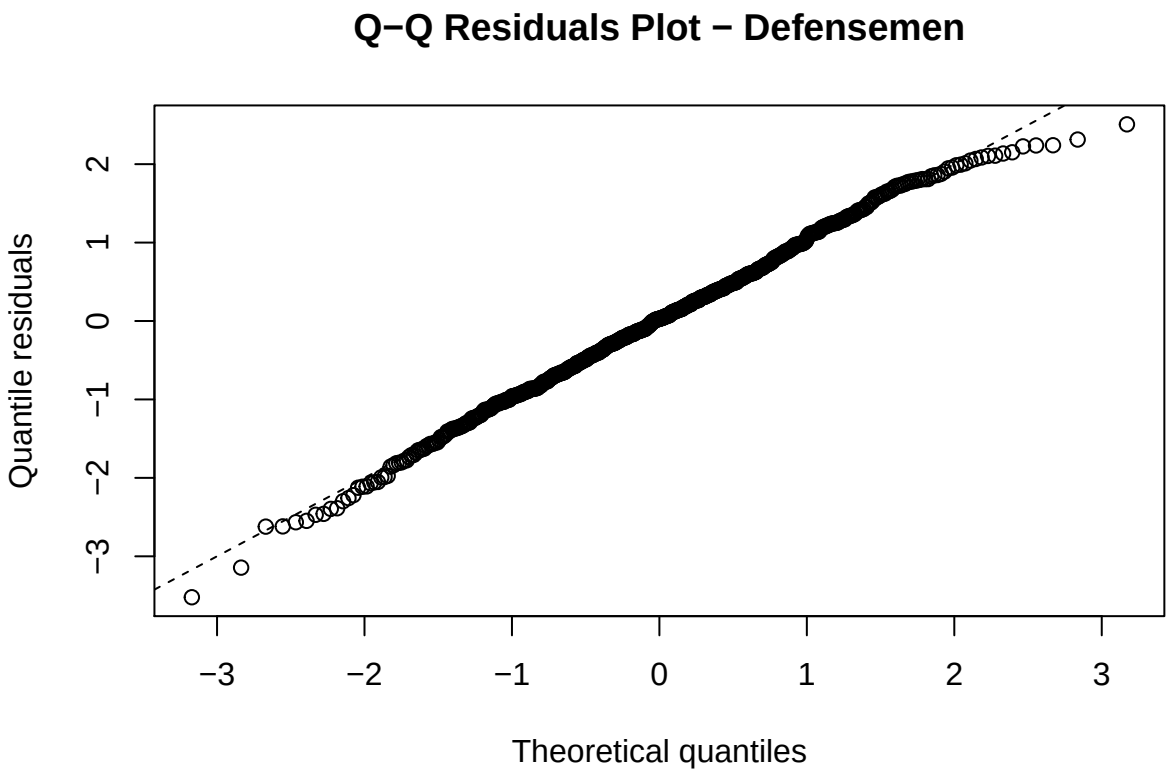
```
rootogram(dg6, xlab = "NHL Games Played",
          ylab = "Square Root of Number of Players",
          main = "Rootogram - Defensemen")
```




```
# Residuals
sqrt(mean(dg6$residuals^2))
```

```
## [1] 113.0641
```

```
qqrplot(dg6, main = "Q-Q Residuals Plot - Defensemen")
```




```
#Check to see if it is different from the intercept (null) model
dg0 <- update(dg6, . ~ 1)
waldtest(dg0, dg6)
```

```
## Wald test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.PPG | Amateur.A
##   Res.Df Df  Chisq Pr(>Chisq)
## 1    655
## 2    653  2 6.6851    0.03535 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(dg0, dg6)
```

```
## Likelihood ratio test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.PPG | Amateur.A
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   3 -1720.5
## 2   5 -1717.1  2 6.7655    0.03395 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

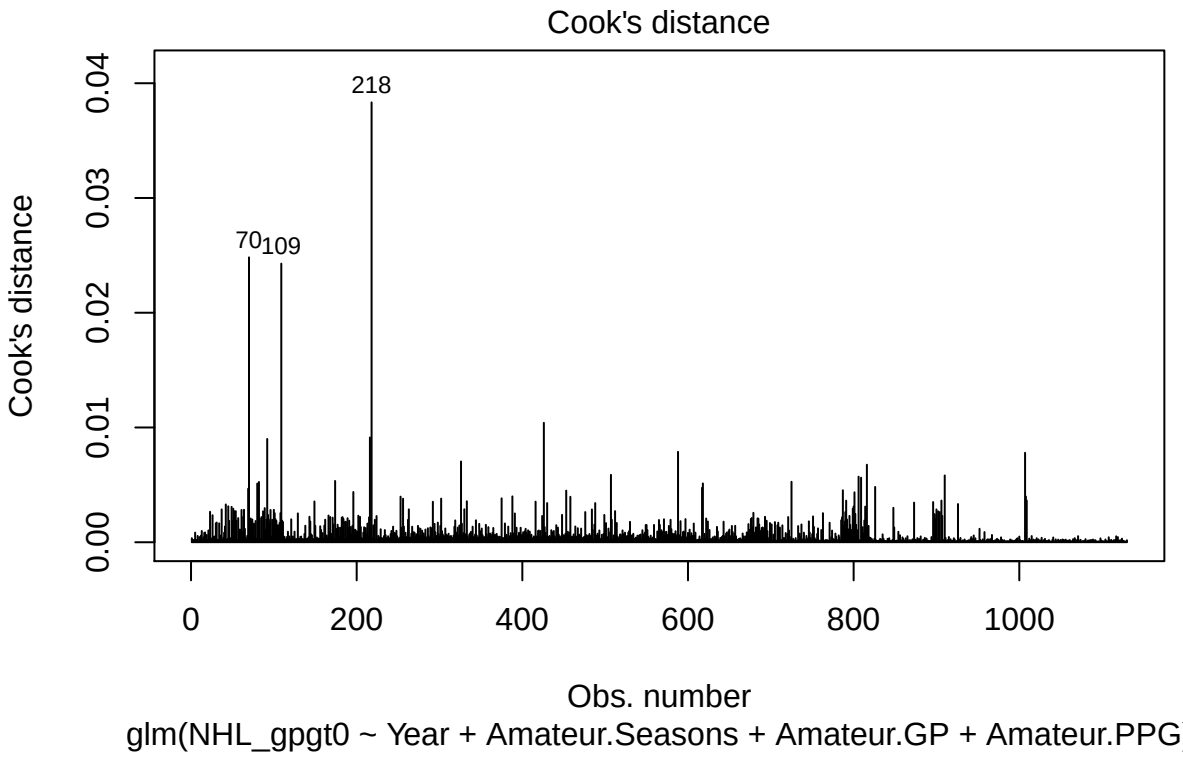
```
# both tests are significant
```

D4 - Logistic Regression Model - Forwards

```
# Forwards Logistic Regression (filtered)
mod <- glm(NHL_gpgt0 ~ Year + Amateur.Seasons +
             Amateur.GP +
             Amateur.PPG, family = binomial(link = "logit"),
           data = f_gt8)
summary(mod)
```

```
##
## Call:
## glm(formula = NHL_gpgt0 ~ Year + Amateur.Seasons + Amateur.GP +
##     Amateur.PPG, family = binomial(link = "logit"), data = f_gt8)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1511  -0.9113  -0.5767   1.0519   2.1730
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     568.174852  49.376353  11.507  < 2e-16 ***
## Year             -0.282204   0.024512 -11.513  < 2e-16 ***
## Amateur.Seasons  -0.609540   0.107307  -5.680 1.34e-08 ***
## Amateur.GP        0.008279   0.002020   4.098 4.17e-05 ***
## Amateur.PPG       0.550057   0.155235   3.543 0.000395 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1526.5  on 1129  degrees of freedom
## Residual deviance: 1327.7  on 1125  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 1337.7
##
## Number of Fisher Scoring iterations: 3
```

```
#check influential points
plot(mod, which = 4, id.n = 3)
```




```
#extract points
model.data <- augment(mod) %>%
  mutate(index = 1:n())
model.data %>%
  top_n(3, .cooksd)
```

```
## # A tibble: 3 x 14
##   .rownames NHL_gpgt0  Year Amateur.Seasons Amateur.GP Amateur.PPG .fitted
##   <chr>         <dbl> <dbl>           <dbl>      <dbl>       <dbl>   <dbl>
## 1 70                0  2010               2         55        3.24   1.96
## 2 109               0  2010               4         50        3.56   0.879
## 3 218               0  2011               1         14        3.71   2.21
## # ... with 7 more variables: .se.fit <dbl>, .resid <dbl>, .hat <dbl>,
## #   .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>, index <int>
```

```
#the standardized residuals (.std.resid) are not > 3
ggplot(model.data, aes(index, .std.resid)) +
  geom_point(aes(color = NHL_gpgt0), alpha = .5) +
  theme_bw() +
  xlab("Index") + ylab("Standard Residuals") +
  ggtitle("Standardized Residuals - Forwards")
```


```
#check for multicollinearity using the car package
car::vif(mod)
```

```
##            Year Amateur.Seasons      Amateur.GP     Amateur.PPG
##        1.007022        1.970285        1.945564        1.044108
```

```
# Diagnostics on classfication success
# Check confusion matrix from caret
glm_probs <- glm(NHL_gpgt0 ~ Year + s_amateur_apg,
                 family = binomial(link = "logit"),
                 data = x_train) %>%
  predict(newdata = x_test, type = "response")
glm_pred <- ifelse(glm_probs > 0.5, 1, 0)
g_pred <- ifelse(glm_pred == 1, "Yes", "No")
x_preds <- tibble(x_test, g_pred) %>%
  mutate(x_truth = ifelse(x_test$NHL_gpgt0 == 1, "Yes", "No"))

#caret confusionMatrix
confusionMatrix(as.factor(x_preds$g_pred),
                as.factor(x_preds$x_truth),
                positive = "Yes")
```

D5 - Zero-Inflated Negative Binomial Model - Forwards

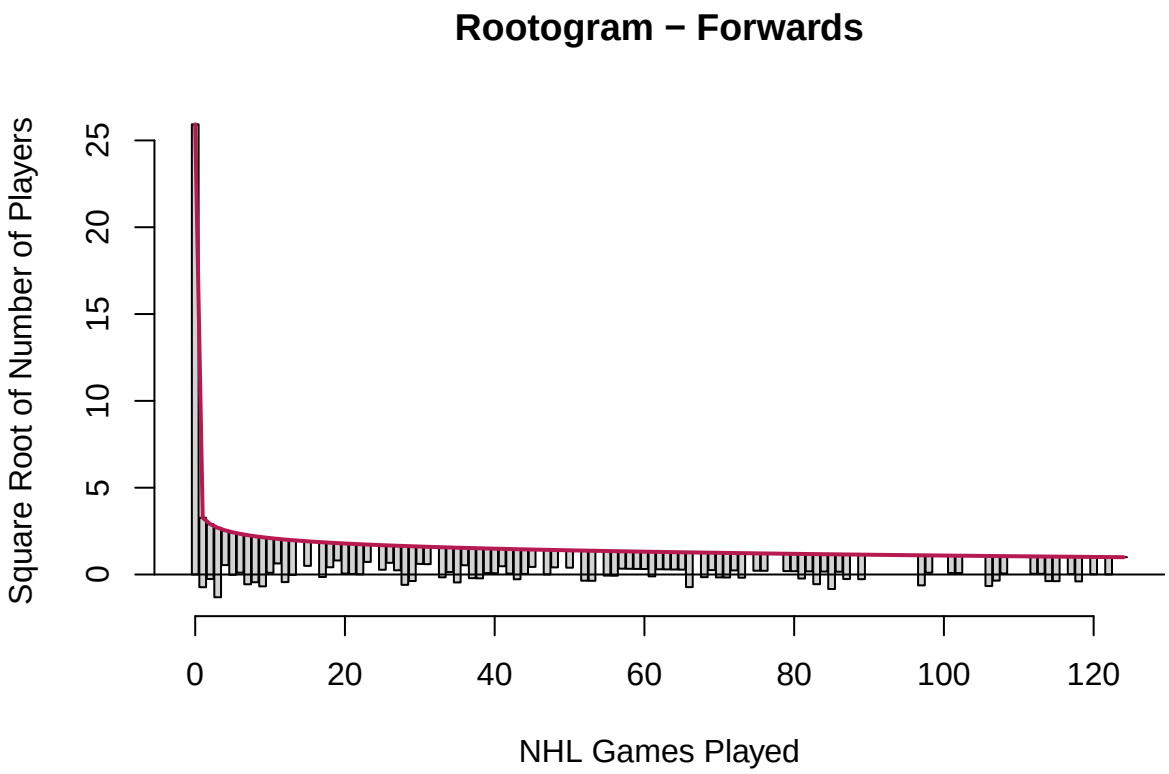
```
# Zero inflated negative binomial - defensemen
library(countreg)
fg4 <- zeroinfl(NHL.GP ~
                  Amateur.GP + Amateur.G
                | Amateur.Seasons + Amateur.A,
                data = f_gt8, dist = "negbin",
                EM = TRUE)
summary(fg4)
```

```
##
## Call:
## zeroinfl(formula = NHL.GP ~ Amateur.GP + Amateur.G | Amateur.Seasons +
##     Amateur.A, data = f_gt8, dist = "negbin", EM = TRUE)
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -0.63513 -0.43829 -0.36681 -0.08543  6.03139
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.211851   0.143350  36.357  < 2e-16 ***
## Amateur.GP  -0.007826   0.002121  -3.690 0.000225 ***
## Amateur.G    0.017304   0.004252   4.069 4.72e-05 ***
## Log(theta)  -0.534227   0.077164  -6.923 4.41e-12 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.185442   0.170194  -1.090    0.276
## Amateur.Seasons  0.587368   0.088326   6.650 2.93e-11 ***
## Amateur.A       -0.018467   0.002749  -6.716 1.86e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.5861
## Number of iterations in BFGS optimization: 1
## Log-likelihood: -3495 on 7 Df
```

```
AIC(fg4)
```

```
## [1] 7003.739
```

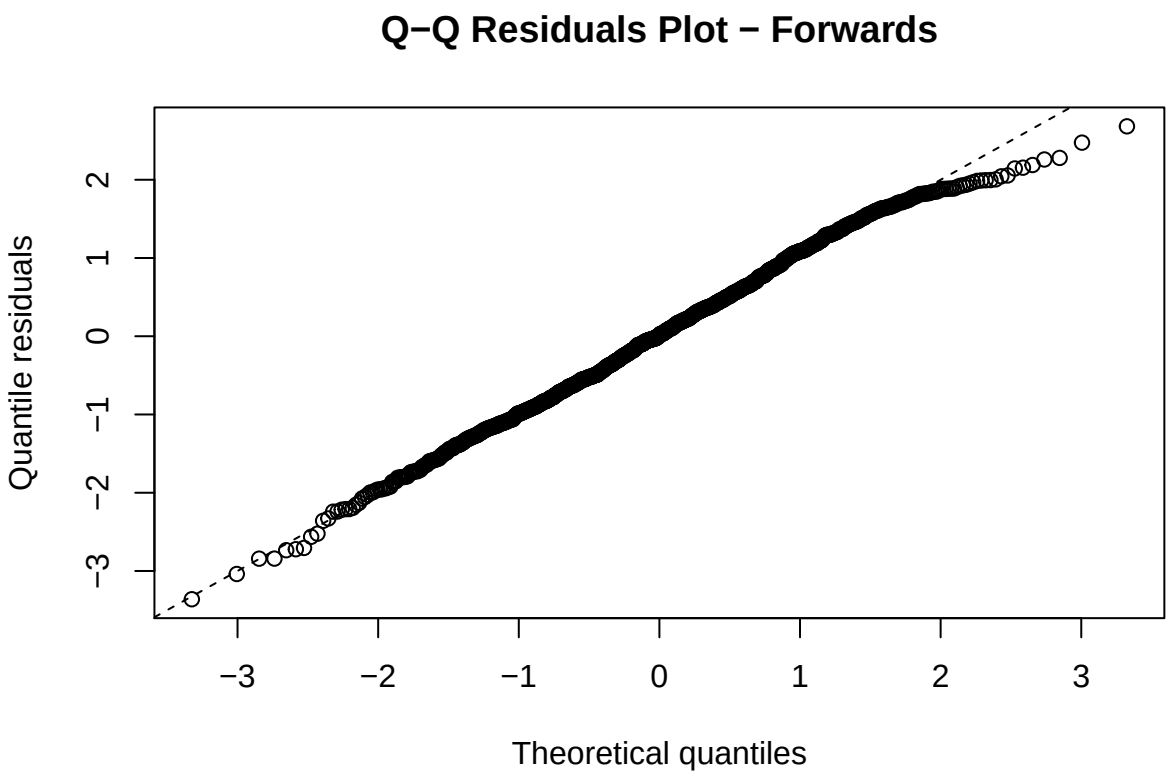
```
rootogram(fg4, xlab = "NHL Games Played",
          ylab = "Square Root of Number of Players",
          main = "Rootogram - Forwards")
```




```
# Residuals
sqrt(mean(fg4$residuals^2))
```

```
## [1] 132.9388
```

```
qqrplot(fg4, main = "Q-Q Residuals Plot - Forwards")
```




```
#Check to see if it is different from the intercept (null) model
fg0 <- update(fg4, . ~ 1)
waldtest(fg0, fg4)
```

```
## Wald test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.GP + Amateur.G | Amateur.Seasons + Amateur.A
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   1128
## 2   1124  4 76.117  1.157e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(fg0, fg4)
```

```
## Likelihood ratio test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.GP + Amateur.G | Amateur.Seasons + Amateur.A
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   3 -3536.3
## 2   7 -3494.9  4 82.816  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# both tests are significant
```

D6 - Logistic Regression Model - Goalies

```
# Goalies Logistic Regression (filtered)
mod <- glm(NHL_gpgt0 ~ Year + Amateur.Seasons + Amateur.GAA,
           family = binomial(link = "logit"),
           data = g_gt8)
summary(mod)
```

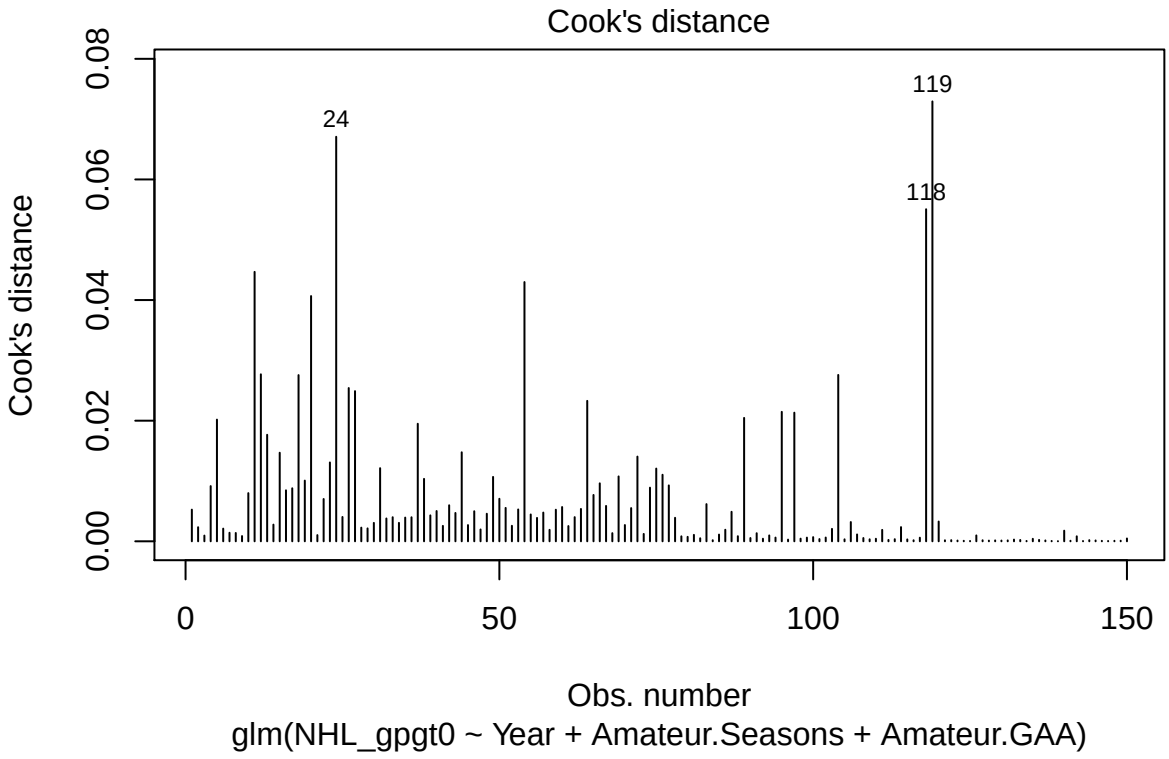
```
##
## Call:
## glm(formula = NHL_gpgt0 ~ Year + Amateur.Seasons + Amateur.GAA,
##     family = binomial(link = "logit"), data = g_gt8)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9830  -0.6955  -0.3432   0.8058   2.5255
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1006.83343  190.22111   5.293 1.20e-07 ***
## Year              -0.49971    0.09432  -5.298 1.17e-07 ***
## Amateur.Seasons    0.65196    0.26762   2.436   0.0148 *
## Amateur.GAA       -0.85800    0.35851  -2.393   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 189.54  on 149  degrees of freedom
## Residual deviance: 140.25  on 146  degrees of freedom
## AIC: 148.25
##
## Number of Fisher Scoring iterations: 5
```

```
anova(mod, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: NHL_gpgt0
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                              149     189.54
## Year             1   38.944       148     150.59 4.362e-10 ***
```

```
## Amateur.Seasons  1    4.138       147     146.46   0.04193 *
## Amateur.GAA      1    6.201       146     140.25   0.01277 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

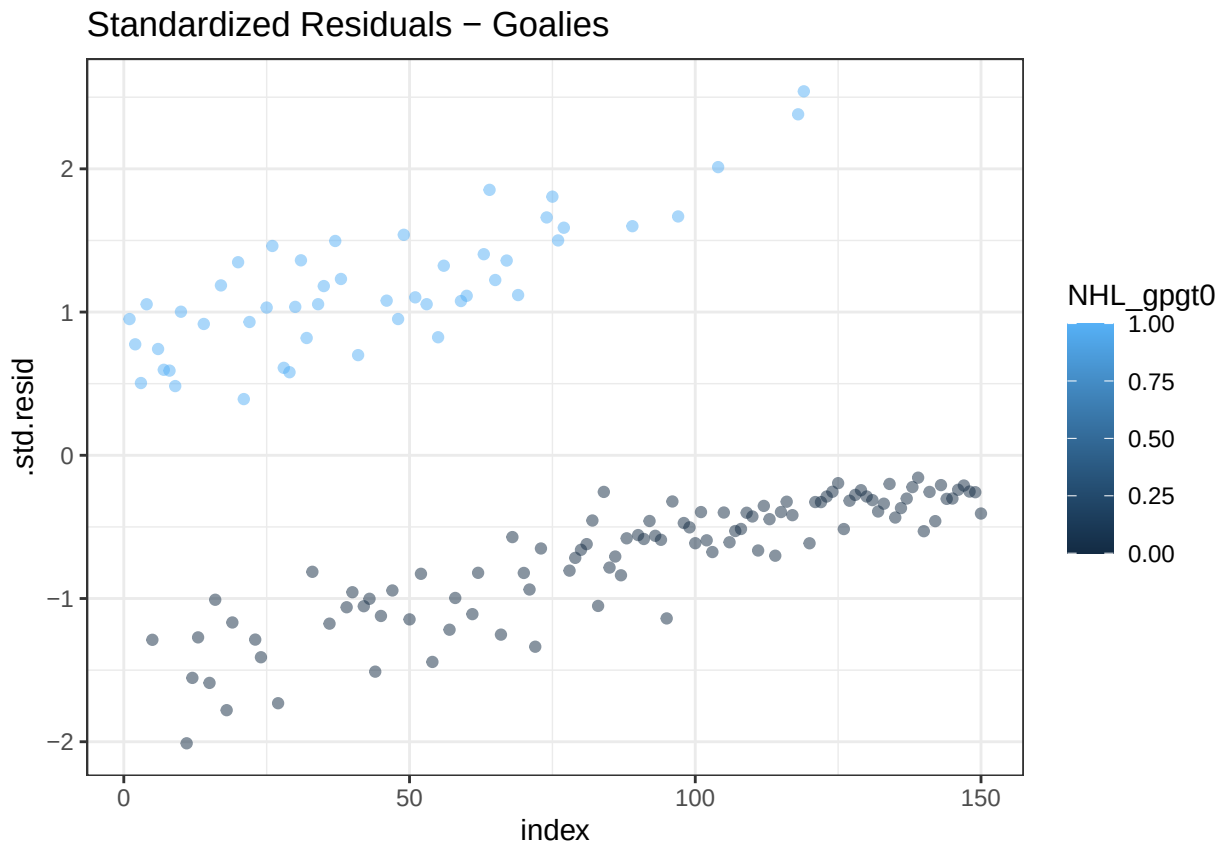
```
#check influential points
plot(mod, which = 4, id.n = 3)
```




```
#extract points
model.data <- augment(mod) %>%
  mutate(index = 1:n())
model.data %>%
  top_n(3, .cooksd)
```

```
## # A tibble: 3 x 12
##   NHL_gpgt0  Year Amateur.Seasons Amateur.GAA .fitted .se.fit .resid   .hat
##       <dbl> <dbl>           <dbl>       <dbl>   <dbl>   <dbl>  <dbl>  <dbl>
## 1         0  2011               4        4.93   0.290   0.773  -1.30 0.146
## 2         1  2017               2        3.44  -2.73    0.493   2.36 0.0140
## 3         1  2017               1        3.16  -3.15    0.556   2.53 0.0122
## # ... with 4 more variables: .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>,
## #   index <int>
```

```
#the standardized residuals (.std.resid) are not > 3
ggplot(model.data, aes(index, .std.resid)) +
  geom_point(aes(color = NHL_gpgt0), alpha = .5) +
  theme_bw() +
  ggtitle("Standardized Residuals - Goalies")
```




```
#check for multicollinearity using the car package
car::vif(mod)
```

```
##            Year Amateur.Seasons      Amateur.GAA
##        1.159013        1.088337        1.230648
```

D7 - Zero-Inflated Negative Binomial Model - Goalies

```
# Zero inflated negative binomial - goalie
library(countreg)
gg4 <- zeroinfl(NHL.GP ~ Amateur.Seasons
                | Amateur.Seasons,
                data = g_gt8, dist = "negbin",
                EM = TRUE)
summary(gg4)
```

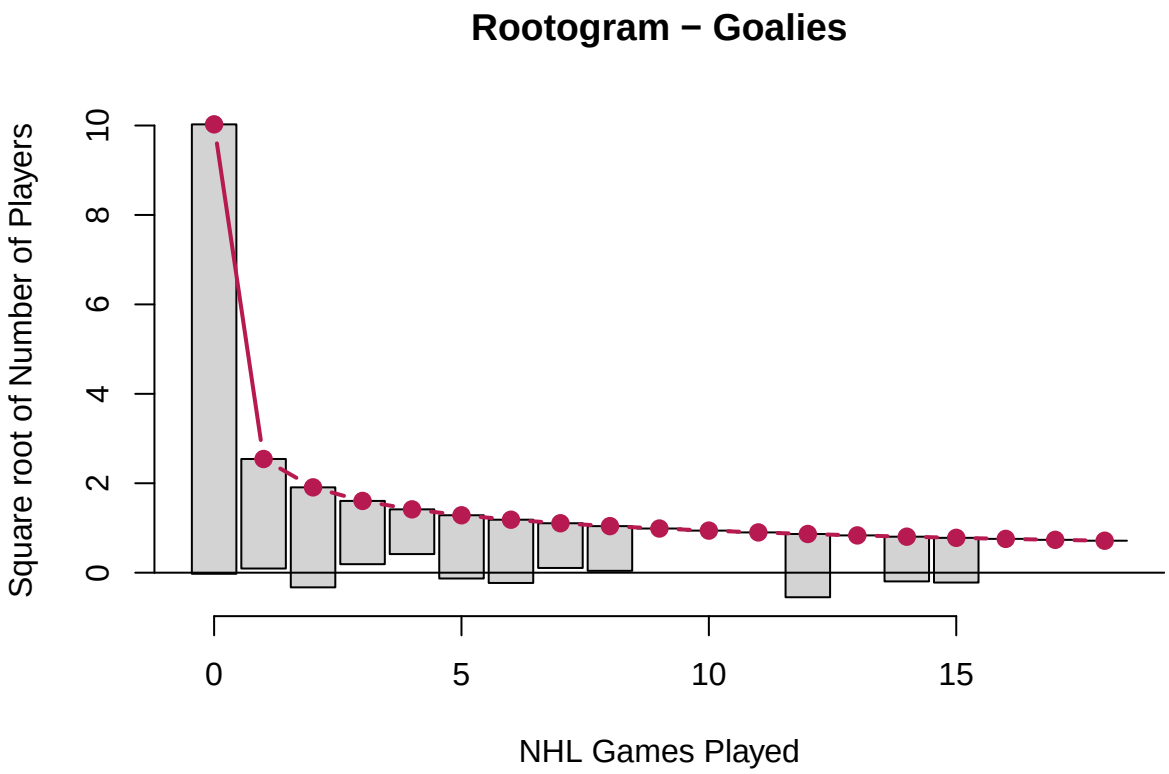
```
##
```

```
## Call:
## zeroinfl(formula = NHL.GP ~ Amateur.Seasons | Amateur.Seasons, data = g_gt8,
##     dist = "negbin", EM = TRUE)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.3612 -0.3060 -0.2390 -0.1888  4.7554
##
## Count model coefficients (negbin with log link):
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.8003     0.9976   3.810 0.000139 ***
## Amateur.Seasons  -0.2454     0.3851  -0.637 0.524029
## Log(theta)       -2.0193     0.3662  -5.514  3.5e-08 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.758      1.187   2.324   0.0201 *
## Amateur.Seasons   -1.885      1.011  -1.865   0.0622 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.1327
## Number of iterations in BFGS optimization: 3
## Log-likelihood: -316.9 on 5 Df
```

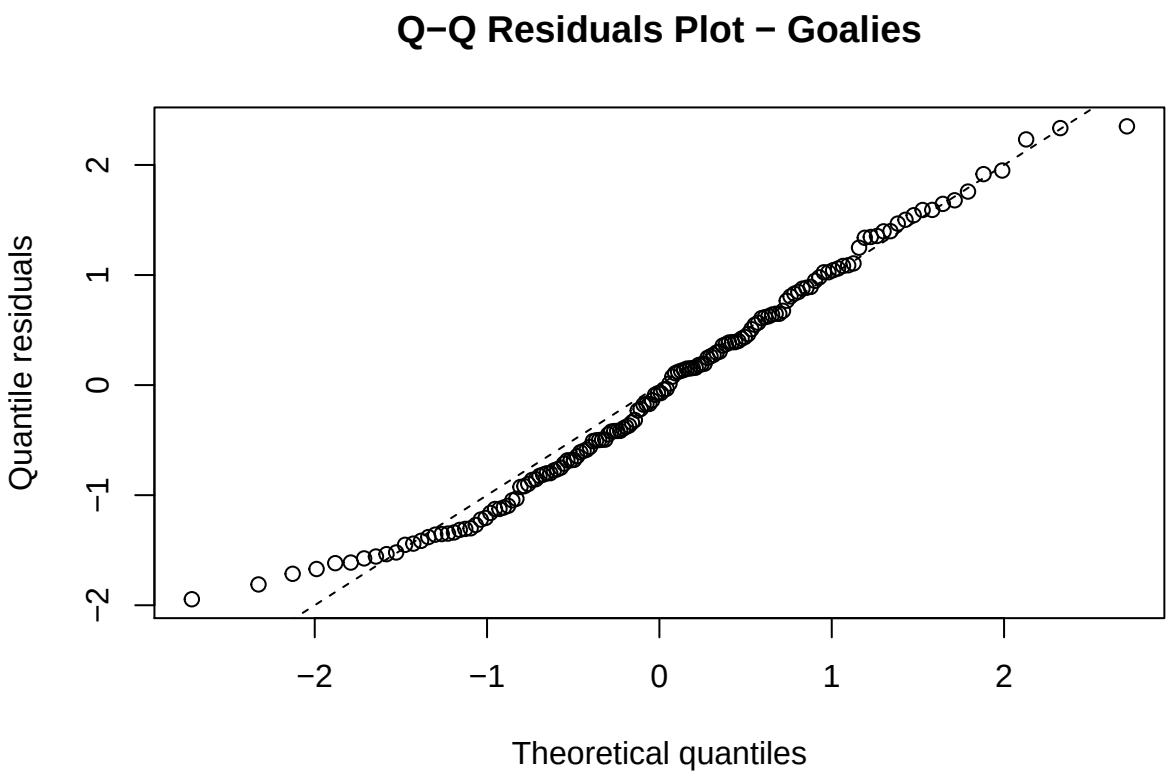
```
AIC(gg4)
```

```
## [1] 643.7751
```

```
rootogram(gg4, xlab = "NHL Games Played",
          ylab = "Square root of Number of Players",
          main = "Rootogram - Goalies")
```




```
qqrplot(gg4, main = "Q-Q Residuals Plot - Goalies")
```




```
gg0 <- update(gg4, .~1)
waldtest(gg0, gg4)
```

```
## Wald test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.Seasons | Amateur.Seasons
##   Res.Df Df  Chisq Pr(>Chisq)
## 1    147
## 2    145  2 4.1323     0.1267
```

```
lrtest(gg0, gg4)
```

```
## Likelihood ratio test
##
## Model 1: NHL.GP ~ 1
## Model 2: NHL.GP ~ Amateur.Seasons | Amateur.Seasons
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   3 -321.92
## 2   5 -316.89  2 10.062   0.006533 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix E - Deep Learning Algorithm

```
library(tidyverse)
library(magrittr)
library(keras)

# DATASET Deep2 from build_data.R is predictors
# response is y_tot from same script

# 1. Divide whole dataset into 80% training/validation (A)
#   and 20% testing data set (B).
# 2. Divide training/validation (A) into 80% training and
#   20% validation data

samp <- floor(0.8 * nrow(deep2))
set.seed(15)
index <- sample(seq_len(nrow(deep2)), size = samp)
x_train1 <- as.matrix(deep2[index, ])
x_test <- as.matrix(deep2[-index, ])

# now training/validation
samp1 <- floor(0.8 * nrow(x_train1))
set.seed(13)
index1 <- sample(seq_len(nrow(x_train1)), size = samp1)
x_train <- as.matrix(x_train1[index1, ])
x_val <- as.matrix(x_train1[-index1, ])

# now do response and change to categorical
n_class <- length(unique(y_tot))

# Now do response 80/20 test/train, & 80/20 train/val
y1 <- as.integer(y_tot[index])

#y3 = y_train, y4 = y_val
y3 <- as.integer(y1[index1])
y4 <- as.integer(y1[-index1])

#to_categorical requires integer vector from 0:nclasses
y_train <- to_categorical(y3, num_classes = 5)
y_test <- as.integer(y_tot[-index])
y_val <- to_categorical(y4, num_classes = 5)

#######################################
#Build neural net architecture - MLP
#######################################
model <- keras_model_sequential()

#model architecture
model %>%
  layer_dense(units = 256,
              input_shape = c(ncol(x_train))) %>%
  layer_activation("relu") %>%
  layer_dropout(0.5) %>%
```

```
  layer_dense(units = 64) %>%
  layer_activation("relu") %>%
  layer_dropout(0.5) %>%
  layer_dense(units = 64) %>%
  layer_activation("relu") %>%
  layer_dropout(0.5) %>%
  layer_dense(units = n_class) %>%
  layer_activation("softmax")

#compile network
model %>%
  compile(
    optimizer = "rmsprop",
    loss = "categorical_crossentropy",
    metrics = c("accuracy")
  )

#model summary
summary(model)

#make sure to use callbacks to save best model
# Run again and save best model
# We need filepaths
checkpoint_dir <- "checkpoints"
unlink(checkpoint_dir, recursive = TRUE)
dir.create(checkpoint_dir)
filepath <- file.path(checkpoint_dir, "weights.{epoch:02d}-{val_loss:.2f}.hdf5")

mc <- callback_model_checkpoint(
  filepath = filepath,
  monitor = 'val_acc',
  mode = 'max',
  save_best_only = TRUE
)

# fit the model
# use validation data to evaluate
history <- model %>%
  fit(
    x_train,
    y_train,
    epochs = 50,
    batch_size = 32,
    validation_data = list(x_val, y_val),
    callbacks = mc
  )

plot(history)

#evaluate the model
model %>% evaluate(x_test, y_test)

# We need a fresh model from a function
```

```
# see:  https://keras.rstudio.com/articles/tutorial_save_and_restore.html
# use the same architecture
create_model <- function() {
  model <- keras_model_sequential()

  #model architecture
  model %>%
    layer_dense(units = 256,
                input_shape = c(ncol(x_train))) %>%
    layer_activation("relu") %>%
    layer_dropout(0.5) %>%
    layer_dense(units = 64) %>%
    layer_activation("relu") %>%
    layer_dropout(0.5) %>%
    layer_dense(units = 64) %>%
    layer_activation("relu") %>%
    layer_dropout(0.5) %>%
    layer_dense(units = n_class) %>%
    layer_activation("softmax")

  #compile network
  model %>%
    compile(
      optimizer = "rmsprop",
      loss = "categorical_crossentropy",
      metrics = c("accuracy")
    )
  model
}

model <- create_model()

model %>% fit(
  x_train,
  y_train,
  epochs = 50,
  batch_size = 32,
  validation_data = list(x_val, y_val),
  callbacks = mc
)

p_class <- model %>%
  predict_classes(x_test)
p_class <- factor(p_class, levels=c(0:4))
t <- table(p_class, y_test, dnn = c('pred', 'resp'))
acc = sum(diag(t))/sum(t)
acc
```