If You Take The Road Less Travelled by, Does it Make a Difference?

An Analysis on Developmental Paths to National Hockey League

Minyue Fan, Steve Kim, Kexiong Shen, Linda Yang

April 2021

Abstract

After being drafted, a player can train in different leagues before they they can play for the famous National Hockey League (NHL). Such training through different leagues is called a developmental path. However, the transition does not happen as often in some leagues as others. We address the question of whether different developmental paths for hockey players affect their success. For this specific project, we evaluate their success by whether the player transitions into the NHL and by the number of games they play in the NHL. We examine data from season 2000 to 2020 for players from various leagues. From exploratory data analysis, it appears that more players from the NCAA succeed than those from other leagues. However, the NCAA is competitive to get into in the first place, and the player is naturally better than those from other leagues. Therefore, we perform a causal inference analysis using propensity score weighting and Bayesian Additive Regression Tree to isolate the treatment effect by the league, and evaluate whether taking different developmental path affects their success. We found that taking the NCAA path in draft year one does not significantly impact a prospect's future success in the NHL.

1 Introduction

Hockey players play in various developmental leagues as part of their professional development before entering the National Hockey League (NHL). There are multiple traditional development paths, determined by the leagues participated in, that hockey prospects can take to get to the NHL. Some of the notable ones include going from US High School Leagues such as the United States Hockey League (USHL) to College, and then to the NHL, and for International players, going from the Kontinental Hockey League (KHL) to the NHL. Unlike some other major professional sports in North America such as football and basketball, where most prospects enter the professional leagues right after they are drafted, most hockey prospects do not immediately go to the NHL when they become eligible for the NHL draft, even if they are drafted. For example, only

0.9% of players drafted in the 2020 NHL draft entered the NHL that season [1]. Instead, they stay in or move to different developmental leagues before playing professionally for an NHL team.

Which developmental league to play in is an important career decision to make for hockey prospects because it could potentially impact their chances at getting into and performing well in the NHL. Therefore, the effect of playing in different developmental leagues on prospects future career in the NHL is crucial information when making this decision. General managers and scouts of professional hockey teams in the NHL also find this information valuable because when they are making draft decisions, they usually know which developmental leagues the prospects are going to play in next season and knowing this information could impact the teams' decisions. Our study investigates the effect of development paths in projecting a prospect's chances at having success at the NHL level. Our research question is:

- How do players' development paths impact their performance and success in the NHL?

One obstacle when attempting to examine this effect is that development path a player takes is correlated with their quality level. For example, one common observation is that American players who take the NCAA path have higher success rates than other paths, but the NCAA player pool is already better than most of the other developmental leagues in terms of quality [2]. Our study, then, intends to establish a causal relationship between taking any particular development path and players future success in the NHL.

The client for this project is Sam Ventura, who is the Director of Hockey Operations and Hockey Research at Pittsburgh Penguins. His team wants to utilize the findings of this project to learn more about the development of prospective hockey players and to potentially guide drafting decisions in the NHL draft.

2 Data

Variable	Definition
Player	Player name
Position	Player's position, which includes forwards and defensemen.
DateofBirth	In MM/DD/YYYY format.
Height	player's height in both inches and meters.
Weight	players' weight in both pounds and kilograms.
Nation	Nationality of player. Dual citizen-ships are included
Shoots	Handedness of the player. Either L (left) or R (right).

Table 1: Variables associated with players' biographical information

The data for this project is web scraped from Elite Prospects's website [3]. We maintain two datasets: one is players' biographical information, the other is their performance in every season from 2001-2020.

For the biography datasets, 15,786 players were included. Table 1 lists the 7 variables that were kept.

The performance dataset has 266,326 entries with information on 15,220 players. Table 2 lists the 19 variables that were kept.

Variable	Definition
Player	Player's Name
Season	The season in which the player played. Example: 2001-02.
Team	Player's team during the season
League	Player's league during the season
Games	Number of games played during the season
Goals	Number of goals during the season
Assists	Number of times player enables the goal to a scoring teammate. There is a maximum of two assists per goal
TotalPoints	Total points achieved during the season
PenaltyMin	Total minutes the player was on penalty
PlusMinus	Evaluation on performance relative to other teammates

Table 2: Variables associated with players' performance each season

To qualitatively understand players' transition in a 5-year time frame after they are eligible for being drafted at age 18, we plot the following two alluvial plots, conditioning on players in USHL in their first draft year (DY0). Each line represents a player and his transition between leagues from year t to year t+1, while the color of that line is determined by his new league. In this plot, we use green for NHL, orange for AHL, blue for NCAA and pink for USHL. On the y-axis, we keep track of the number of players in each year, and on x-axis, we condition on draft years. Each box represents a league, while box size indicates how many players are in that league in a certain year.

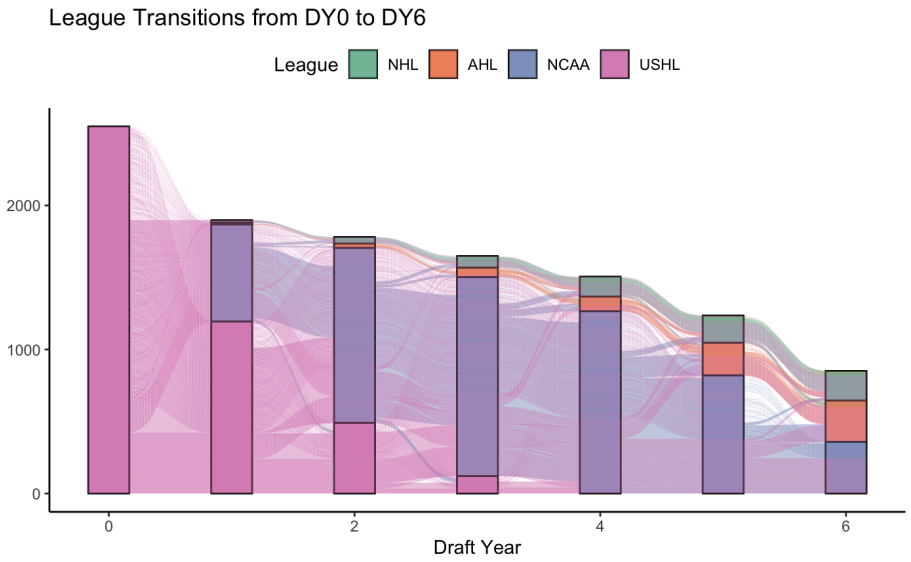


Figure 1: Players' Transition into NHL in 5 years (4 Leagues of Interest)

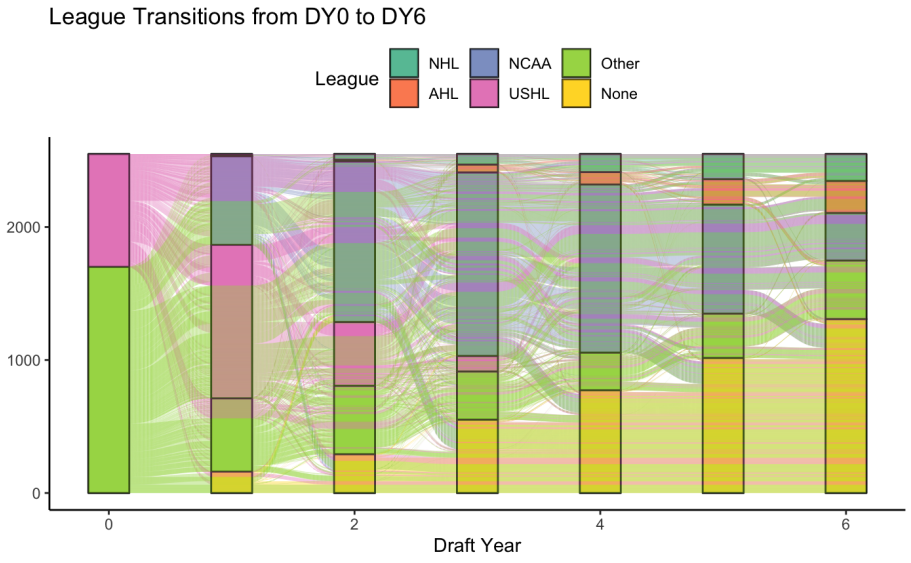


Figure 2: Players' Transitions into NHL in 5 years (A Full Picture)

3 Methods

3.1 Markov Chain

3.1.1 Simple Markov Chain Model

Our baseline model is a simple Markov chain model in which each state is uniquely determined by a league. The following variables are used for our analysis:

Variable	Definition
Occ(s)	the number of occurrences of state as observed
Occ(s, t)	the number of occurrences of state s being immediately followed by state t
transition probability function	Occ(s, s_0)/Occ(s). Bounded between (0, 1]

This model gives us a basic understanding of different probabilities of transitioning into NHL from NCAA and USHL, regardless of other variables.

3.1.2 Markov Chain Model with Logistic Regression

We add one more variable, players' age information, to each state because usually younger players are favored and therefore are more likely to play for the NHL. Therefore, including player's age will allow us to more accurately calculate the transition probability between states. Theoretically, adding more biographical and performance information such as height and goals will make the prediction more accurate because intuitively, taller players with higher goals will be favored. However, these variables are continuous, and will result in infinitely many states, which not only requires more computing power to calculate but also more data to be split on. Instead of grouping continuous variables and dividing them into categorical ones, we fit a logistic regression in each state to model the transition probability. This combination of Markov chain model and logistic regression not only lets us perform a more accurate analysis on probability conditioning on players' age compared to a generic logistic regression on all data, but also allows for more precise projection from different values in the continuous variable.

3.2 Propensity Scores

We are interested in the causal effects of developmental path on a player's success once in the NHL. We looked at players that were in the USHL, a popular developmental league, at the age of 18 and assessed whether they go into the NCAA, another common developmental league, the following year or stay in the USHL. Propensity score weighting was used to assess the causal impact of developmental league at age 19 on a player's success in the NHL.

Since the data that we have on players and their career paths are observational, propensity score weighting was used to remove any selection bias in the treatment assignments among the players. The treatment effect was being in the NCAA at age 19 while the control effect was being in the USHL at age 19. We used player success metrics at age 18 such as assists, goals, draft value, plus minus, and penalty minutes and other information such as height and weight to predict the treatment effect at age 19. This allowed us to remove any selection effect of the treatment and control assignments among the players. The probability of being in the NCAA at age 19 served as our propensity scores. Based on the propensity scores, players were weighted in the final model. We explored two methods:

Method 1

For treatment group

$$\omega = \frac{1}{1 - \hat{e}(x)}$$

For control group

$$\omega = \frac{1}{\hat{e}(x)}$$

Method 2

$$\omega = \hat{e}(x)(1 - \hat{e}(x))$$

$$\hat{e}(x) = P(T = NCAA | X)$$

To ensure that we are exploring the nuances between defense men and forward positioned players, we performed propensity score weighting separately for both position types.

Selection bias was confirmed to be removed after the weighting process by running a logistic regression model with the treatment effect as the response and player metrics (assists, goals, plus minus, weight, draft value, etc.) as predictors. The value and significance of the coefficient estimates were then assessed. If our estimates are close to zero at some statistically significant degree, then we can confirm that weighting was done correctly.

Once the selection effect was removed, we predicted the average number of NHL games played since being drafted into the NHL based on success metrics at age 18 and the treatment effect (USHL vs NCAA) on the weighted data set. We then assessed the significance of the coefficient estimate of the treatment variable to determine the causal effect of developmental path on success in the NHL.

3.3 Bayesian Additive Regression Trees

To quantitatively understand the causal effect between different developmental paths and number of games played in NHL per season, we used Bayesian additive regression trees (BART) [4]. Consistent with the initial data sets in the propensity score method, we looked at the players that became just eligible to

be drafted at age 18, or draft year 0. We also filter out those prospect players who play on the NHL in the following season because we would like to investigate the treatment effect of being in different leagues in that year.
As a non-parametric model, BART provides precise modeling of the response because it has more control for the confounding variables compared to parametric models. We use the average NHL games played in a season for each player's career span as our response variable, and calculate it as the total number of games played by a player divided by the number of years in NHL.
As performance data such as goals and points can be totally different for forward and defense players, we also separate players based on their positions. For players who switched positions in their careers, we take their major position and rule them as either defense or forward.
Using the bartMachine package [5] in RStudio, we fit a BART model, using the developmental league as the treatment, while treating all other variables as the confounding variables. We than extract the posterior distribution from the fitted model to make predictions on how many NHL games on average the prospect would play, given their performance statistics and biographical information, as well as the information on the developmental league they play in during that season.

As we recognize the importance of goals and assists to both forward and defense players, we also ask the question: if a player's performance improves by one increment, how does that affect his probability of getting into NHL? We therefore use BART to simulate projection with one more goals, and compared to the predicted the results with his original goals. We look at whether the difference between projected goals is larger than 0 to investigate if having one more increment in goal/assist/height/weight positively affects one's chance of getting into NHL.

4 Results

4.1 EDA Analysis

From plot 1, we see that more and more players transition to NCAA year after year, before they are too old to play for USHL in draft year 4. Because we plot NHL, AHL, NCAA and USHL in the order from the top to the bottom, we can visually see if a player goes "up" to a better league, or goes "down" to a less prestigious league. We see that more "upward" transitions to NHL happen from NCAA than USHL, which matches our prior knowledge.
It is also worth noting that the number of players decrease gradually. From plot 2, we see that some goes to other leagues, while more and more drop out of hockey.

4.2 Markov Chain

While conditioning only on leagues in our simple markov chain model, this baseline model presents itself as a simple counting problem. Out of the 15,786 players, we calculate a transition probability of 2% from NCAA to NHL while the number for USHL is only 0.7%. Note this model does not tell us about the treatment effect of going through different leagues, but is merely a retrospective analysis of the past transitioning events.

4.3 Propensity Scores

For forwards, we were successful in removing selection bias after propensity score weighting. In figure 3, we modeled the effects of predictors at draft year 0 on the choice of developmental path at draft year 1. The coefficient estimates are all insignificant so we have successfully removed selection effect for forward position players.

In figure 4, we have the model estimates from predicting the average number of games played in the NHL since being eligible based on predictors in draft year 0 (plus minus, goals, penalty minutes, assists, height, weight, and draft value) and developmental league at draft year 1 on the weighted data. We can see development league is insignificant while draft value is significant. This indicates that after the removal of selection effect for forwards, developmental league is not a significant predictor of success in the NHL; however, draft value is.

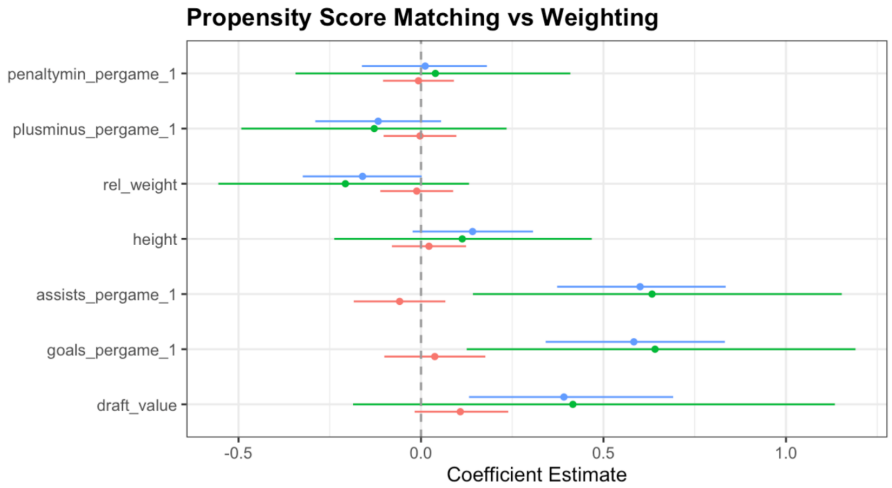


Figure 3: Predicting Treatment on Weighted Data for Forward Positioned Players

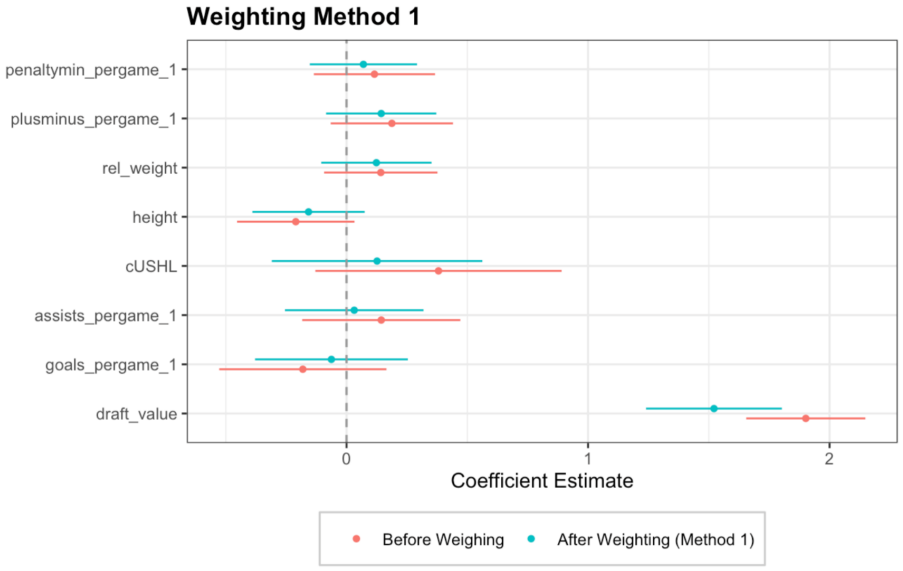


Figure 4: Predicting NHL Success on Weighted Data for Forward Positioned Players

When predicting treatment effect based on draft year 0 predictors for defense men on the weighted data, we observe a significant estimate for draft value; however, selection bias was removed overall based on the insignificant coefficient estimates among all other variables. These estimates can be seen in figure 5.

After weighting the data based on the propensity scores, we see that developmental league is not a significant predictor of success in the NHL in figure 6. Instead, we see that draft value and weight are better predictors of success in the NHL for defense men.

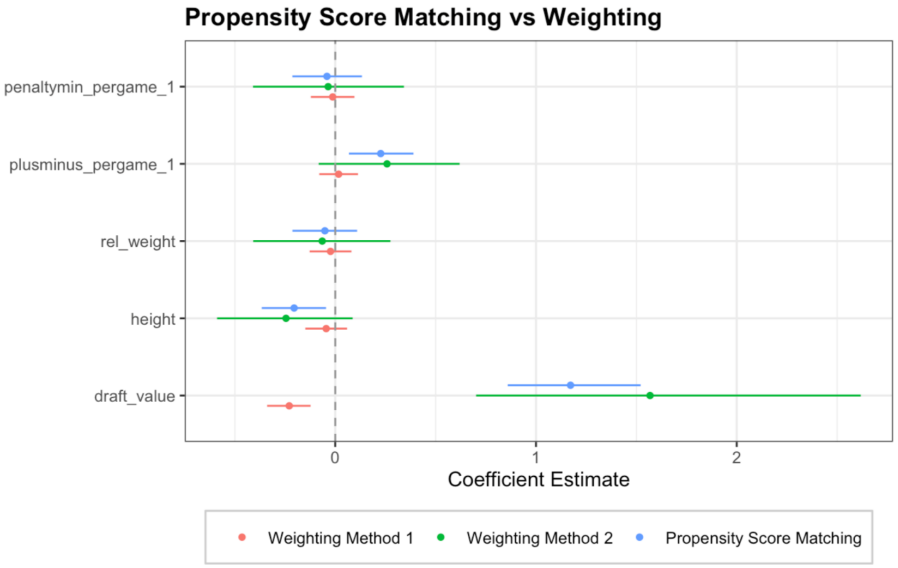


Figure 5: Predicting Treatment on Weighted Data for Defense Men

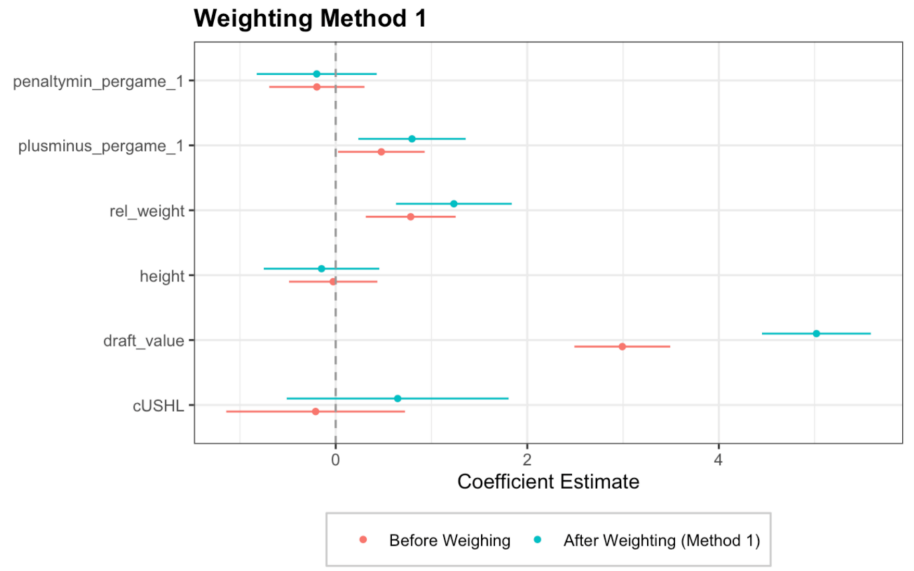


Figure 6: Predicting Treatment on Weighted Data for Defense Men

4.4 Bayesian Additive Regression Trees

We fitted two separate BART models for forwards and defensemen using the set of predictors discussed in the methods section. Table 3 includes the calculated conditional average treatment effects of going into the NCAA in draft year one from both the estimated BART models and the propensity score weighting method used in the previous section. The "observed average difference" column displays the difference in average number of games played per season in the NHL between the group of players who entered the NCAA in draft year one and the group of players who stayed in the USHL in draft year one. As we can see from the results displayed in table 3, the estimated conditional average treatment effects of going into the NCAA from both BART and propensity score weighting are much smaller than the observed average differences from the original data and are all insignificant. These results indicate that after considering the selection bias, going into the NCAA in draft year one does not have a significant impact on the expected number of games per season the prospect plays in the NHL.

	Observed Average Difference	CATE using BART	CATE from propensity score weighting
Forwards	13.36513	0.0705	2.0297
Defense	14.65782	0.0325	0.1871

Table 3: Estimated conditional average treatment effects of going into the NCAA in draft year one from BART and propensity score weighting.

From the plot of the variable importance (as measured by the inclusion proportion in predicting the average number of NHL games played per season of each variable), we see that the draft value has the highest inclusion proportion, followed by the height, relative weight, the total penalty minutes played and the plus-minus rating. Relative to the above variables, whether prospects played in USHL in the following season and their general position demonstrated less inclusion proportion.

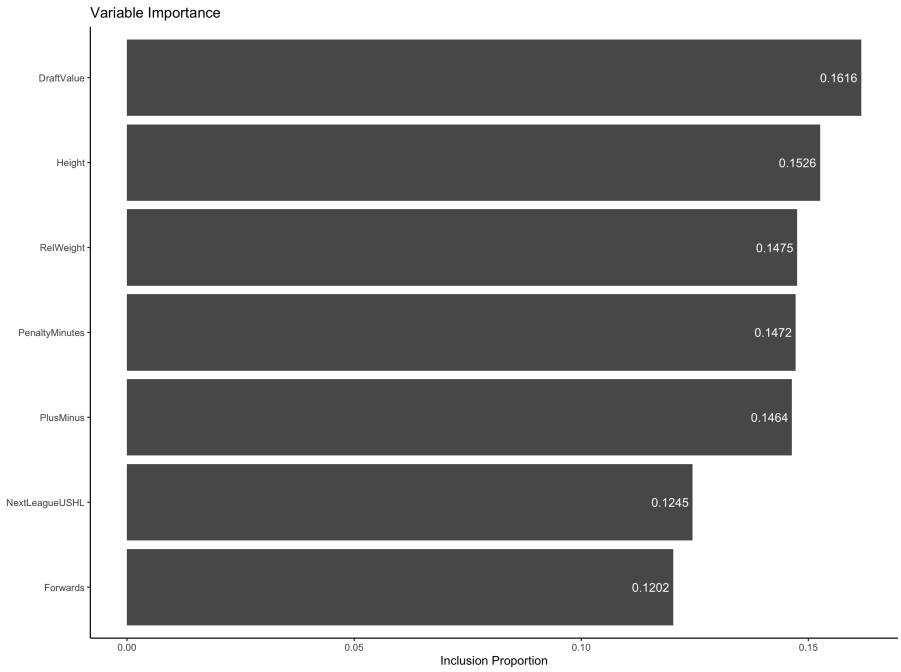


Figure 7: Variable Importance from the BART model

4.5 BART Inference

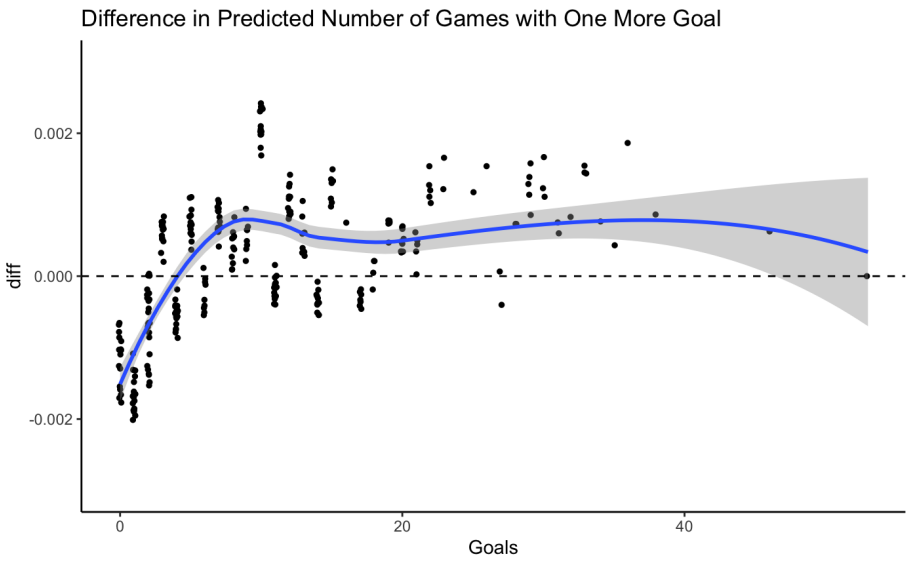


Figure 8: What If Forward Players Have One More Goal?

We see in figure 8 that for players with 5 more goals, their probability of transitioning into NHL increases. This is especially significant for players having goals between 6 and 11. Having one more goal increases their probability by at most 0.2%. For players with more than 11 goals, the increase in projection in less significant, but they are still larger than 0, indicating a positive increase in chance. Note that for players with less than 5 goals, the predicted difference is less than 0. It is against common sense, but usually these are the players that NHL will not be interested in. Therefore, we can ignore the abnormality for now.

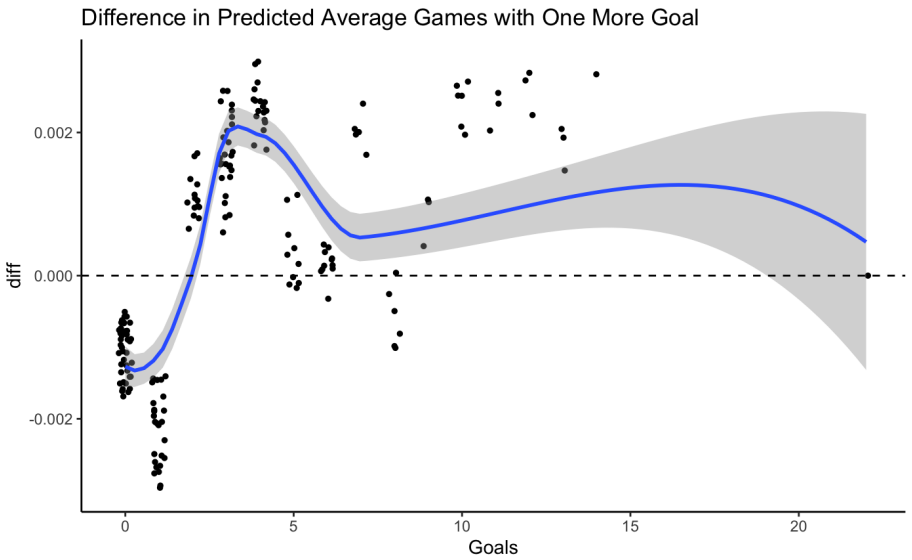


Figure 9: What If Defense Players Have One More Goal?

We repeat the same process for defense players. Note that goals have a different meaning for defense players: they prevent scoring from their opponent, and goals here represent how many goals they block. We therefore see the range of goals shrink from 0 to 51 in forward players to 0 to 23 for defense players. In figure 9, we see that in general, having one more goal contributes positively to the transition probability for players with more than 4 goals. We observe the same abnormality for players with less than 4 goals, but again, we ignore them for now.

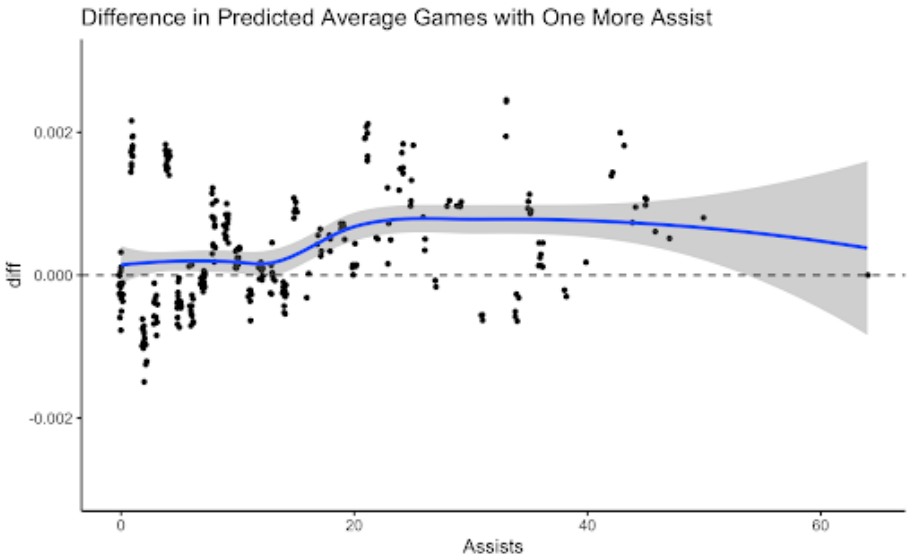


Figure 10: What If Forward Players Have One More Assist?

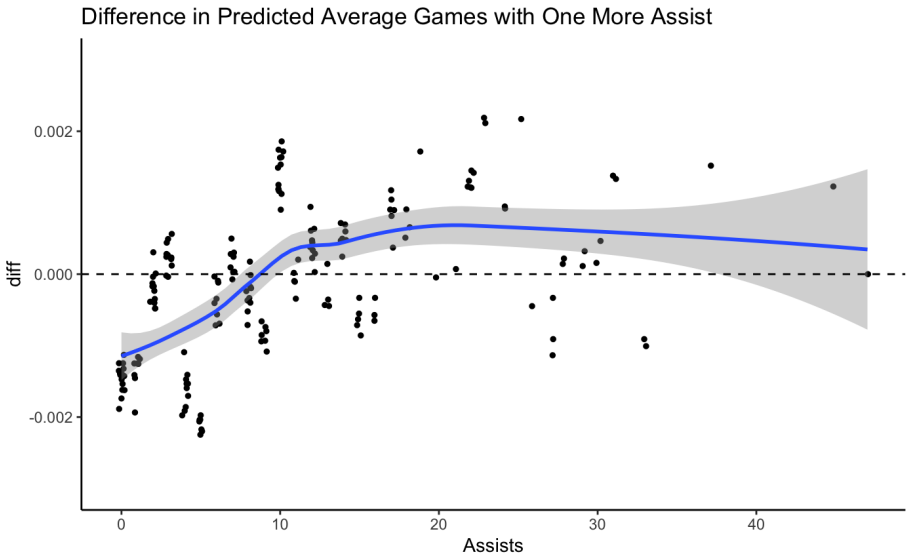


Figure 11: What Defense Players Have One More Assist?

We ask the same question for assists, and have an overall positive impact with having one more assist.

5 Discussion

Across both the propensity score methods and BART modeling, we arrive at effectively the same conclusion: for those prospect who are entering their draft year and had played in the USHL in the most recent season, whether or not they move on to stay in the USHL or move on to play in the NCAA did not seem to have much causal relationship with the average number of NHL games played per season. The observed differences in the conditional average treatment effect for both methods is reasonably small. Propensity score weighting allowed us to remove any confounding effects from our predictors by re-weighting players based on the probability of their treatment assignments. We observed that the developmental path is a non-significant predictor from this method across both forwards and defense men, while we notice that the developmental path has less variable importance compared to the player bio as well as the draft information - this suggests that, contrary to the common belief that those who play in the NCAA typically have better chances of playing in the NHL in their career. Such findings may even suggest the players to select their developmental league based on other factors, such as potential playing time being offered, rather than on the name value of the league early on in their developmental phase (very early in their draft years).

Interestingly, both the propensity score methods and BART model seem to suggest that the draft value is statistically significant in predicting for the average number of NHL games per season. However, keeping in mind that we did our analysis for this project on the prospects entering their draft year (players who are just drafted) and played in a developmental league in the United States (USHL), the fact that the draft value seems to have the highest significance may not be surprising - in fact, it may indicate good scouting being done (presumably based on the prospects' performance, prior to entering the draft) to pick the best available prospects earlier in the draft.

There are a few notes to mention on our analysis: first we would like to address the potential bias that may exist in our selected response variable. We chose the average number of NHL games per season throughout the individual player's career by taking the total number of NHL games divided by the number of seasons that the player has at least played in a single NHL game. However, we realized that this response variable may have some faulty behavior, as there may be players who only play a handful of games, either due to roster eligibility issues or the veteran status of the player - in the future, we may prefer selecting a typical age range which may be seen widely as the prime years of a hockey player. There may also be cases where a player only participated for only a few minutes, and yet would be given the same number of games as another player who played close to a full game; we may also prefer to use completely different hockey stat based on their actual performance. While simply taking the goals / assists per game may disregard the positional differences, we may devise a genuine metric that accounts for the defensive aspects of the game and weigh them equally as the scoring-related statistics.

Second, we should take into account that, for the purposes of this project, we

took a subset of all prospects from different backgrounds. To be specific, we only considered players who played in the USHL just before entering the draft and went on to play in either USHL or NCAA. We should be aware of the existence of other developmental leagues, such as the AHL, as well as overseas leagues, which may actually have more causal relationship, compared to the USHL or the NCAA. In the future, we would also like to address and implement other developmental leagues. Also, we only looked at players going from the initial draft year to the following year. However, the effect of developmental path, albeit be the same path, may be different depending on which stage of the developmental phase that the prospects may be in. For example, while either playing in the USHL or NCAA in their following season after entering the draft year may not seem to matter, it may hold true that after 5 to 6 years since the initial draft year that playing in the NCAA does in fact have causal relationship with reasonable success at the NHL.

References

[1] Wayne Jones. *What percentage of NHL draft picks make it to the NHL?* URL: https://hockeyanswered.com/what-percentage-of-nhl-draft-picks-make-it-to-the-nhl/. (accessed: 05.11.2021).

[2] Nate Ewell. *Study: NCAA Leads in NHL Draft Success*. URL: http://collegehockeyinc.com/articles/2014/06/study-ncaa-leads-nhl-draft-success.php. (accessed: 05.11.2021).

[3] *Elite Prospects*. URL: http://eliteprospects.com/.

[4] Jennifer L. Hill. "Bayesian Nonparametric Modeling for Causal Inference". In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 217–240. DOI: 10.1198/jcgs.2010.08162.

[5] *Bart Machine Package*. URL: https://cran.r-project.org/web/packages/bartMachine/bartMachine.pdf.

Propensity Score Weighting

Propensity Score Weighting

Data Cleaning

In the data cleaning process below, we are mapping draft position to a numerical value in order to use it as a predictor in order model. Additionally, we are converting player metrics such as plus minus, assists , goals,and penalty minus to a per game level.

```
draft_value = c(917, 871, 826, 783, 741, 702, 665, 629, 592, 565, 535, 507, 481, 456, 433, 413, 395, 379
           175, 176, 176, 177, 177, 177, 176, 176, 174, 173, 171, 169, 167, 165, 164, 162, 160, 158, 150
other = rep(50, 83)
length(c(draft_value, other))
```

```
## [1] 293
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
bio = read.csv("complete_bio_updated_withID.csv")
stat = read.csv("complete_stat_updated_withID.csv")
draft = read.csv("player_draft.csv")
draft_value = c(917, 871, 826, 783, 741, 702, 665, 629, 592, 565, 535, 507, 481, 456, 433, 413, 395, 379
           175, 176, 176, 177, 177, 177, 176, 176, 174, 173, 171, 169, 167, 165, 164, 162, 160, 158, 150
other = rep(50, 83)
draft_value = (c(draft_value, other))
Draftposition = 1:293
draft_values =as.data.frame(cbind(Draftposition, draft_value ))
nhl = stat %>% left_join(bio, by = 'ID')
nhl = nhl %>% left_join(draft, by = 'ID')
nhl = nhl %>% left_join(draft_values, by = 'Draftposition')
library(tidyr)
nhl[which(nhl$Player == 'Aaron Ave'),]
```

```
##  [1] Player.x       Season         Team           League         Games
##  [6] Goals          Assists        TotalPoints    PenaltyMinutes PlusMinus
## [11] ID             Player.y       Position       DateofBirth    Height
## [16] Weight         Nation         Shoots         Draftposition  draft_value
## <0 rows> (or 0-length row.names)
```

```
nhl= nhl %>% separate(DateofBirth, into = c('month' , 'birthdate', 'BirthYear'), sep = " ")
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 254 rows [92868,
## 92869, 92870, 92871, 92872, 96874, 96875, 96876, 96877, 96878, 100110, 100111,
## 100112, 100113, 102733, 102734, 102735, 102736, 102933, 102934, ...].
```

```
nhl =  nhl  %>% separate(Season, into = c('Season', 'endseason'), sep = '-')
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 77478 rows [6, 7,
## 9, 15, 19, 21, 26, 30, 35, 37, 38, 40, 41, 42, 43, 45, 46, 48, 49, 50, ...].
```

```
nhl$BirthYear = as.numeric(nhl$BirthYear)
nhl$Season = as.numeric(nhl$Season)
nhl = nhl[complete.cases(nhl$Season), ]

nhl$age_actual =  nhl$Season - nhl$BirthYear
nhl$Goals = as.numeric(nhl$Goals)
```

```
## Warning: NAs introduced by coercion
```

```
nhl$PlusMinus = as.numeric(nhl$PlusMinus)
```

```
## Warning: NAs introduced by coercion
```

```
nhl$PenaltyMin = as.numeric(nhl$PenaltyMinutes)
```

```
## Warning: NAs introduced by coercion
```

```
nhl$GamesPlayed = as.numeric(nhl$Games)
```

```
## Warning: NAs introduced by coercion
```

```
nhl$Assists = as.numeric(nhl$Assists)
```

```
## Warning: NAs introduced by coercion
```

```
nhl$TotalPoints = as.numeric(nhl$TotalPoints)
```

```
## Warning: NAs introduced by coercion
```

```
library(tidyr)
updated_nhl = nhl
updated_nhl$goals_pergame = updated_nhl$Goals/updated_nhl$GamesPlayed
updated_nhl$plusminus_pergame = updated_nhl$PlusMinus/updated_nhl$GamesPlayed
updated_nhl$PenaltyMin_pergame = updated_nhl$PenaltyMin/updated_nhl$GamesPlayed
updated_nhl$Assists_pergame = updated_nhl$Assists/updated_nhl$GamesPlayed
updated_nhl$TotalPoints_pergame = updated_nhl$TotalPoints/updated_nhl$GamesPlayed
updated_nhl[which(updated_nhl$Player == 'Aaron Ave'),]
```

```
##  [1] Player.x            Season              endseason
##  [4] Team                League              Games
##  [7] Goals               Assists             TotalPoints
## [10] PenaltyMinutes      PlusMinus           ID
## [13] Player.y            Position            month
## [16] birthdate           BirthYear           Height
## [19] Weight              Nation              Shoots
## [22] Draftposition       draft_value         age_actual
## [25] PenaltyMin          GamesPlayed         goals_pergame
## [28] plusminus_pergame   PenaltyMin_pergame  Assists_pergame
## [31] TotalPoints_pergame
```

```
## <0 rows> (or 0-length row.names)
```

Next, we filtered the data so that we obtain each player's metrics at draft year 0 and their developmental league at draft year 1. We also calculated the average number of NHL games playerd for each player, which will serve as our response variable

```
players_ushl_18 = updated_nhl[which(updated_nhl$age_actual == 18 & updated_nhl$League == 'USHL'),]$ID
updated_nhl = updated_nhl[which(updated_nhl$ID %in% players_ushl_18), ]


updated_nhl$plusminus_pergame_draft_year = updated_nhl$plusminus_pergame
updated_nhl[which(updated_nhl$age_actual != 18),]$plusminus_pergame_draft_year = 0


updated_nhl$totalpoints_pergame_draft_year = updated_nhl$TotalPoints_pergame
updated_nhl[which(updated_nhl$age_actual != 18),]$totalpoints_pergame_draft_year= 0


updated_nhl$goals_pergame_draft_year = updated_nhl$goals_pergame
updated_nhl[which(updated_nhl$age_actual != 18),]$goals_pergame_draft_year = 0


updated_nhl$penalty_pergame_draft_year = updated_nhl$PenaltyMin_pergame
updated_nhl[which(updated_nhl$age_actual != 18),]$penalty_pergame_draft_year = 0


updated_nhl$assists_pergame_draft_year = updated_nhl$Assists_pergame
updated_nhl[which(updated_nhl$age_actual != 18),]$assists_pergame_draft_year = 0


updated_nhl$GamesPlayed_after_in_NHL = updated_nhl$GamesPlayed

updated_nhl[which(updated_nhl$League != 'NHL' ),]$GamesPlayed_after_in_NHL = 0
updated_nhl$league_19 = updated_nhl$League
updated_nhl[which(updated_nhl$age_actual != 19), ]$league_19 = ''
updated_nhl = updated_nhl %>% group_by(ID)  %>% mutate(total_games_in_nhl = sum(GamesPlayed_after_in_NHL
                                                       plusminus_pergame_1 = sum(plusminus_pergame_

updated_nhl[which(updated_nhl$Player =='A.J. Drobot'),]
```

```
## Warning: Unknown or uninitialised column: `Player`.
```

```
## # A tibble: 0 x 46
## # Groups:   ID [0]
## # ... with 46 variables: Player.x <chr>, Season <dbl>, endseason <chr>,
## #   Team <chr>, League <chr>, Games <chr>, Goals <dbl>, Assists <dbl>,
## #   TotalPoints <dbl>, PenaltyMinutes <chr>, PlusMinus <dbl>, ID <int>,
## #   Player.y <chr>, Position <chr>, month <chr>, birthdate <chr>,
## #   BirthYear <dbl>, Height <chr>, Weight <chr>, Nation <chr>, Shoots <chr>,
## #   Draftposition <dbl>, draft_value <dbl>, age_actual <dbl>, PenaltyMin <dbl>,
## #   GamesPlayed <dbl>, goals_pergame <dbl>, plusminus_pergame <dbl>,
## #   PenaltyMin_pergame <dbl>, Assists_pergame <dbl>, TotalPoints_pergame <dbl>,
## #   plusminus_pergame_draft_year <dbl>, totalpoints_pergame_draft_year <dbl>,
## #   goals_pergame_draft_year <dbl>, penalty_pergame_draft_year <dbl>,
## #   assists_pergame_draft_year <dbl>, GamesPlayed_after_in_NHL <dbl>,
## #   league_19 <chr>, total_games_in_nhl <dbl>, goals_pergame_1 <dbl>,
```

```
## #   plusminus_pergame_1 <dbl>, totalpoints_pergame_1 <dbl>,
## #   penaltymin_pergame_1 <dbl>, assists_pergame_1 <dbl>, c <chr>,
## #   retirement <dbl>
```

```
updated_nhl$position_new = 'backward'
updated_nhl[which(updated_nhl$Position %in% c('C', 'F', 'LW', 'RW')), ]$position_new = 'forward'
updated_nhl$more_than_10_games = 0
updated_nhl[which(updated_nhl$total_games_in_nhl > 10), ]$more_than_10_games = 1

updated_nhl = updated_nhl[which(updated_nhl$age_actual == 18), ]
updated_nhl = updated_nhl[which(updated_nhl$c %in% c('USHL', 'NCAA')), ]
updated_nhl$temp = 1
updated_nhl = updated_nhl %>% group_by(ID) %>% mutate(temp_2 = cumsum(temp))
updated_nhl = updated_nhl[which(updated_nhl$temp_2 == 1), ]
```

We then created height and weight information for each player. In our analysis, we saw that height and weight were very correlated with other. To account for this, we created a relative weight variable that removes any correlation it has with weight

```
updated_nhl = updated_nhl %>% separate(Height, c("blah","blah1", "height","blah3"), sep = " ")
```

```
## Warning: Expected 4 pieces. Missing pieces filled with `NA` in 1 rows [1690].
```

```
updated_nhl$height = as.numeric(updated_nhl$height)
```

```
## Warning: NAs introduced by coercion
```

```
updated_nhl = updated_nhl %>% separate(Weight, c("Weight","blah1h", "blahh2", "blahh3", "blahh4"), sep =
```

```
## Warning: Expected 5 pieces. Missing pieces filled with `NA` in 1 rows [1690].
```

```
updated_nhl$Weight = as.numeric(updated_nhl$Weight)
```

```
## Warning: NAs introduced by coercion
```

```
weight_height = lm(Weight ~ height, data = updated_nhl)
updated_nhl$exp_weight = predict(weight_height, newdata = updated_nhl)
updated_nhl$rel_weight = updated_nhl$Weight - updated_nhl$exp_weight



updated_nhl$games_per_year_in_nhl = updated_nhl$total_games_in_nhl/ (updated_nhl$retirement - updated_nh
updated_nhl[, c('Player.x', 'BirthYear', 'retirement', 'total_games_in_nhl', 'games_per_year_in_nhl')]
```

```
## # A tibble: 1,953 x 5
##    Player.x       BirthYear retirement total_games_in_nhl games_per_year_in_nhl
##    <chr>              <dbl>      <dbl>              <dbl>                <dbl>
##  1 Erik Cole           1978       2014                836                 46.4
##  2 Gary Suter          1964       2001               1129                 59.4
##  3 Mark Eaton          1977       2012                650                 38.2
##  4 Mike Peluso         1974       2003                  0                  0
##  5 Tyler Arnason       1979       2011                467                 33.4
##  6 Brian Swanson       1976       2011                 70                  4.12
##  7 Chris Ferraro       1973       2008                 74                  4.35
##  8 Peter Ferraro       1973       2008                 63                  3.71
##  9 Matt Henderson      1974       2003                  0                  0
## 10 Todd Rohloff        1974       2005                 40                  3.08
## # ... with 1,943 more rows
```

```
#updated_nhl = updated_nhl[which(updated_nhl$Season <= 2015), ]
```

We then filter out the data for forward positioned players and backward positioned players.

```
updated_nhl$y = 0
updated_nhl[which(updated_nhl$c == 'NCAA'),]$y = 1
updated_nhl_forward = updated_nhl[which(updated_nhl$position_new =='forward'),  ]
updated_nhl_backward = updated_nhl[which(updated_nhl$position_new =='backward'),  ]
```

For players that didn't make it to the draft or did not get a draft position, their draft value was set to 27

```
updated_nhl_forward[which(is.na(updated_nhl_forward$Draftposition)), ]$draft_value = 27
updated_nhl_forward$draft = 1
updated_nhl_forward[which(is.na(updated_nhl_forward$Draftposition)), ]$draft = 0
updated_nhl_forward[complete.cases(updated_nhl_forward$penaltymin_pergame_1, updated_nhl_forward$assists
```

```
## # A tibble: 854 x 62
## # Groups:   ID [854]
##    Player.x  Season endseason Team        League Games Goals Assists TotalPoints
##    <chr>      <dbl> <chr>     <chr>       <chr>  <chr> <dbl>   <dbl>       <dbl>
##  1 "Ryan Po~   2002 03        "Lincoln S~ USHL   54       35      43          78
##  2 "Paul St~   2003 04        "River Cit~ USHL   56       30      47          77
##  3 "Max Pac~   2006 07        "Sioux Cit~ USHL   60       21      42          63
##  4 "Andreas~   2005 06        "Sioux Fal~ USHL   58       29      30          59
##  5 "Chris P~   2002 03        "Lincoln S~ USHL   59       13      22          35
##  6 "Nate Ra~   2002 03        "River Cit~ USHL   48        5      17          22
##  7 "Corey E~   2003 04        "St. Louis~ USHL   57       12      17          29
##  8 "John Mc~   2004 05        "Des Moine~ USHL   60        8      10          18
##  9 "Ben Hol~   2005 06        "Sioux Fal~ USHL   56       10       9          19
## 10 "Travis ~   2002 03        "Chicago S~ USHL   60       21      26          47
## # ... with 844 more rows, and 53 more variables: PenaltyMinutes <chr>,
## #   PlusMinus <dbl>, ID <int>, Player.y <chr>, Position <chr>, month <chr>,
## #   birthdate <chr>, BirthYear <dbl>, blah <chr>, blah1 <chr>, height <dbl>,
## #   blah3 <chr>, Weight <dbl>, blah1h <chr>, blahh2 <chr>, blahh3 <chr>,
## #   blahh4 <chr>, Nation <chr>, Shoots <chr>, Draftposition <dbl>,
## #   draft_value <dbl>, age_actual <dbl>, PenaltyMin <dbl>, GamesPlayed <dbl>,
## #   goals_pergame <dbl>, plusminus_pergame <dbl>, PenaltyMin_pergame <dbl>,
## #   Assists_pergame <dbl>, TotalPoints_pergame <dbl>,
## #   plusminus_pergame_draft_year <dbl>, totalpoints_pergame_draft_year <dbl>,
## #   goals_pergame_draft_year <dbl>, penalty_pergame_draft_year <dbl>,
## #   assists_pergame_draft_year <dbl>, GamesPlayed_after_in_NHL <dbl>,
## #   league_19 <chr>, total_games_in_nhl <dbl>, goals_pergame_1 <dbl>,
## #   plusminus_pergame_1 <dbl>, totalpoints_pergame_1 <dbl>,
## #   penaltymin_pergame_1 <dbl>, assists_pergame_1 <dbl>, c <chr>,
## #   retirement <dbl>, position_new <chr>, more_than_10_games <dbl>, temp <dbl>,
## #   temp_2 <dbl>, exp_weight <dbl>, rel_weight <dbl>,
## #   games_per_year_in_nhl <dbl>, y <dbl>, draft <dbl>
```

Forward Position Players: Propensity Score Weighting Process

We predicted the treatment of each player (NCAA vs USHL at draft year 1) based on predictors in draft year 0 using a logistic model. The outcome of this model (P(T=NCAA)) became our propensity score. Players were then weighted based on this propensity score

```
library(Matching)
```

```
## Loading required package: MASS
```

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

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

```
## ##
## ##  Matching (Version 4.9-7, Build Date: 2020-02-05)
## ##  See http://sekhon.berkeley.edu/matching for additional documentation.
## ##  Please cite software as:
## ##   Jasjeet S. Sekhon. 2011. ``Multivariate and Propensity Score Matching
## ##   Software with Automated Balance Optimization: The Matching package for R.''
## ##   Journal of Statistical Software, 42(7): 1-52.
## ##
```

```
library(arm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: lme4
```

```
##
## arm (Version 1.11-2, built: 2020-7-27)
```

```
## Working directory is /Users/lindayang
```

```
library(dplyr)

updated_nhl_forward[c('penaltymin_pergame_1', 'plusminus_pergame_1',  'rel_weight', 'height', 'draft_val
mod1 = glm(y ~ penaltymin_pergame_1 + plusminus_pergame_1  +  rel_weight + height + draft_value , data =
summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + draft_value, family = "binomial", data = updated_nhl_forward)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1429  -1.0427  -0.8548   1.2431   1.7617
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.15333    0.07336  -2.090  0.03660 *
## penaltymin_pergame_1  0.07667    0.07762   0.988  0.32326
## plusminus_pergame_1   0.21131    0.07444   2.839  0.00453 **
## rel_weight           -0.21125    0.07400  -2.855  0.00431 **
```

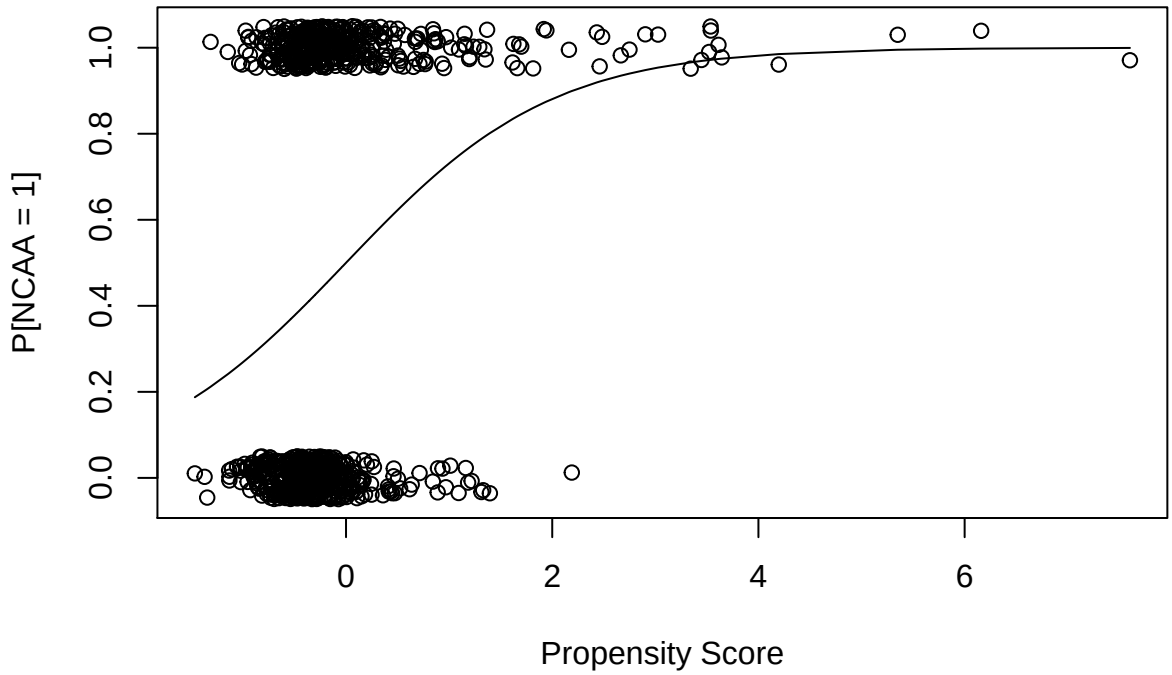
```
## height               -0.08480    0.07380  -1.149  0.25053
## draft_value           0.66709    0.13005   5.130  2.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1177.1  on 853  degrees of freedom
## Residual deviance: 1108.8  on 848  degrees of freedom
##   (173 observations deleted due to missingness)
## AIC: 1120.8
##
## Number of Fisher Scoring iterations: 5
```

```
complete_Cases =updated_nhl_forward[complete.cases(updated_nhl_forward[, c('y', 'penaltymin_pergame_1',
complete_Cases
```

```
## # A tibble: 854 x 62
## # Groups:   ID [854]
##    Player.x  Season endseason Team        League Games Goals Assists TotalPoints
##    <chr>      <dbl> <chr>     <chr>       <chr>  <chr> <dbl>   <dbl>       <dbl>
##  1 "Ryan Po~   2002 03        "Lincoln S~ USHL   54       35      43          78
##  2 "Paul St~   2003 04        "River Cit~ USHL   56       30      47          77
##  3 "Max Pac~   2006 07        "Sioux Cit~ USHL   60       21      42          63
##  4 "Andreas~   2005 06        "Sioux Fal~ USHL   58       29      30          59
##  5 "Chris P~   2002 03        "Lincoln S~ USHL   59       13      22          35
##  6 "Nate Ra~   2002 03        "River Cit~ USHL   48        5      17          22
##  7 "Corey E~   2003 04        "St. Louis~ USHL   57       12      17          29
##  8 "John Mc~   2004 05        "Des Moine~ USHL   60        8      10          18
##  9 "Ben Hol~   2005 06        "Sioux Fal~ USHL   56       10       9          19
## 10 "Travis ~   2002 03        "Chicago S~ USHL   60       21      26          47
## # ... with 844 more rows, and 53 more variables: PenaltyMinutes <chr>,
## #   PlusMinus <dbl>, ID <int>, Player.y <chr>, Position <chr>, month <chr>,
## #   birthdate <chr>, BirthYear <dbl>, blah <chr>, blah1 <chr>, height <dbl>,
## #   blah3 <chr>, Weight <dbl>, blah1h <chr>, blahh2 <chr>, blahh3 <chr>,
## #   blahh4 <chr>, Nation <chr>, Shoots <chr>, Draftposition <dbl>,
## #   draft_value <dbl>, age_actual <dbl>, PenaltyMin <dbl>, GamesPlayed <dbl>,
## #   goals_pergame <dbl>, plusminus_pergame <dbl>, PenaltyMin_pergame <dbl>,
## #   Assists_pergame <dbl>, TotalPoints_pergame <dbl>,
## #   plusminus_pergame_draft_year <dbl>, totalpoints_pergame_draft_year <dbl>,
## #   goals_pergame_draft_year <dbl>, penalty_pergame_draft_year <dbl>,
## #   assists_pergame_draft_year <dbl>, GamesPlayed_after_in_NHL <dbl>,
## #   league_19 <chr>, total_games_in_nhl <dbl>, goals_pergame_1 <dbl>,
## #   plusminus_pergame_1 <dbl>, totalpoints_pergame_1 <dbl>,
## #   penaltymin_pergame_1 <dbl>, assists_pergame_1 <dbl>, c <chr>,
## #   retirement <dbl>, position_new <chr>, more_than_10_games <dbl>, temp <dbl>,
## #   temp_2 <dbl>, exp_weight <dbl>, rel_weight <dbl>,
## #   games_per_year_in_nhl <dbl>, y <dbl>, draft <dbl>
```

```
p.scores = predict(mod1, type = 'link')
complete_Cases$psvalues = predict(mod1, type = 'response')
complete_Cases$weight.ATE = ifelse(complete_Cases$y == 1, 1/complete_Cases$psvalues, 1/(1- complete_Cas
complete_Cases$weight.ATE2 = complete_Cases$psvalues*(1 - complete_Cases$psvalues)
plot(p.scores, jitter(updated_nhl_forward[complete.cases(updated_nhl_forward[, c('y', 'penaltymin_pergam
o.scores = sort(p.scores)
```

```
lines(o.scores, exp(o.scores)/(1 + exp(o.scores)))
```




```
matches = matching(z = complete_Cases$y, score = o.scores)
matched = complete_Cases[matches$match.ind, ]
```

Using the weighted data set, we then check to see if we removed selection if our coefficient estimates are zero when we predict treatment effect based on predictions in draft year 0 . Here we are comparing the two weighting methods with propensity score matching. We then use the weigthed data to predict the average number of games played in the NHL based on predictors in draft year 0 and league in draft year 1 (our treatment effect). This is where we can accurately assess the coefficient estimates and significance of the treatment

```
method_1 = glm(y ~ penaltymin_pergame_1 + plusminus_pergame_1 +rel_weight + height +   draft_value ,  fa
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
method_2 = glm(y ~ penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height +   draft_value ,  f
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
no_weights = glm(y ~ penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height +  draft_value,  f
```

```
#######

# now let's compare estimates of the effect of "watched" on
# post-test score, controlling for all the factors we have
# been interested in...

# the unmatched regression analysis
```

```r
mod1 = lm( games_per_year_in_nhl ~penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height +  c
summary(mod1)
```

```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + c + draft_value, data = complete_Cases)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.847  -0.252   0.038   0.329  44.449
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.2086     0.1763   1.183   0.2371
## penaltymin_pergame_1   0.1005     0.1270   0.791   0.4290
## plusminus_pergame_1    0.1729     0.1189   1.454   0.1463
## rel_weight             0.1301     0.1190   1.093   0.2746
## height                -0.2149     0.1197  -1.796   0.0729 .
## cUSHL                  0.3954     0.2415   1.637   0.1019
## draft_value            1.8993     0.1200  15.826   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.394 on 846 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.2409, Adjusted R-squared:  0.2355
## F-statistic: 44.75 on 6 and 846 DF,  p-value: < 2.2e-16
```

```r
coef(mod1)
```

```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##            0.2085813            0.1004745            0.1728982
##           rel_weight               height                cUSHL
##            0.1301017           -0.2149328            0.3954230
##          draft_value
##            1.8993104
```

```r
# the matched regression analysis

mod2 = lm(games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height +  c
summary(mod2)
```

```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + c + draft_value, data = complete_Cases,
##     weights = weight.ATE)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -18.522  -0.355  -0.048   0.326  91.801
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)           0.37806    0.16295   2.320   0.0206 *
## penaltymin_pergame_1  0.13376    0.12686   1.054   0.2920
## plusminus_pergame_1   0.08962    0.11534   0.777   0.4374
## rel_weight            0.21594    0.11949   1.807   0.0711 .
## height               -0.14074    0.11798  -1.193   0.2333
## cUSHL                 0.15482    0.23071   0.671   0.5024
## draft_value           1.69146    0.13853  12.210   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.747 on 846 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1576, Adjusted R-squared:  0.1516
## F-statistic: 26.37 on 6 and 846 DF,  p-value: < 2.2e-16
```

```
coef(mod2)
```

```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##           0.37806368           0.13376404           0.08961649
##           rel_weight               height                 cUSHL
##           0.21594101          -0.14073553           0.15481866
##          draft_value
##           1.69145724
```

```
mod3 = lm(games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height +  c
summary(mod3)
```

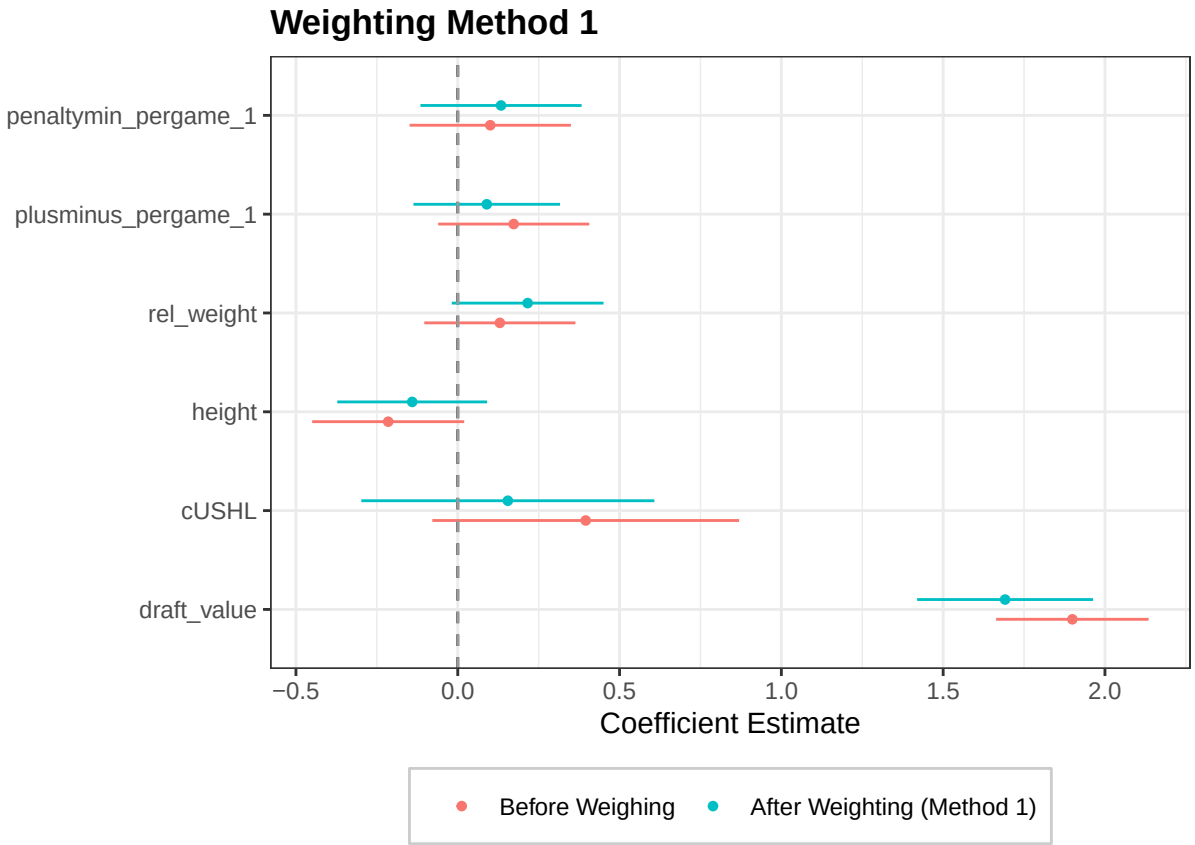
```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + c + draft_value, data = complete_Cases,
##     weights = weight.ATE2)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9981 -0.1096 -0.0484  0.0141 20.1563
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.211306   0.115872   1.824   0.0686 .
## penaltymin_pergame_1  0.042926   0.082973   0.517   0.6051
## plusminus_pergame_1   0.125853   0.080054   1.572   0.1163
## rel_weight            0.007882   0.079421   0.099   0.9210
## height               -0.073465   0.078929  -0.931   0.3522
## cUSHL                 0.180475   0.157212   1.148   0.2513
## draft_value           0.840815   0.141050   5.961 3.68e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.068 on 846 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.04555,    Adjusted R-squared:  0.03878
## F-statistic: 6.729 on 6 and 846 DF,  p-value: 5.625e-07
```

```
coef(mod3)
```

```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##          0.211305670          0.042925742          0.125853019
##           rel_weight               height                cUSHL
##          0.007881595         -0.073465475          0.180474813
##          draft_value
##          0.840815220
```

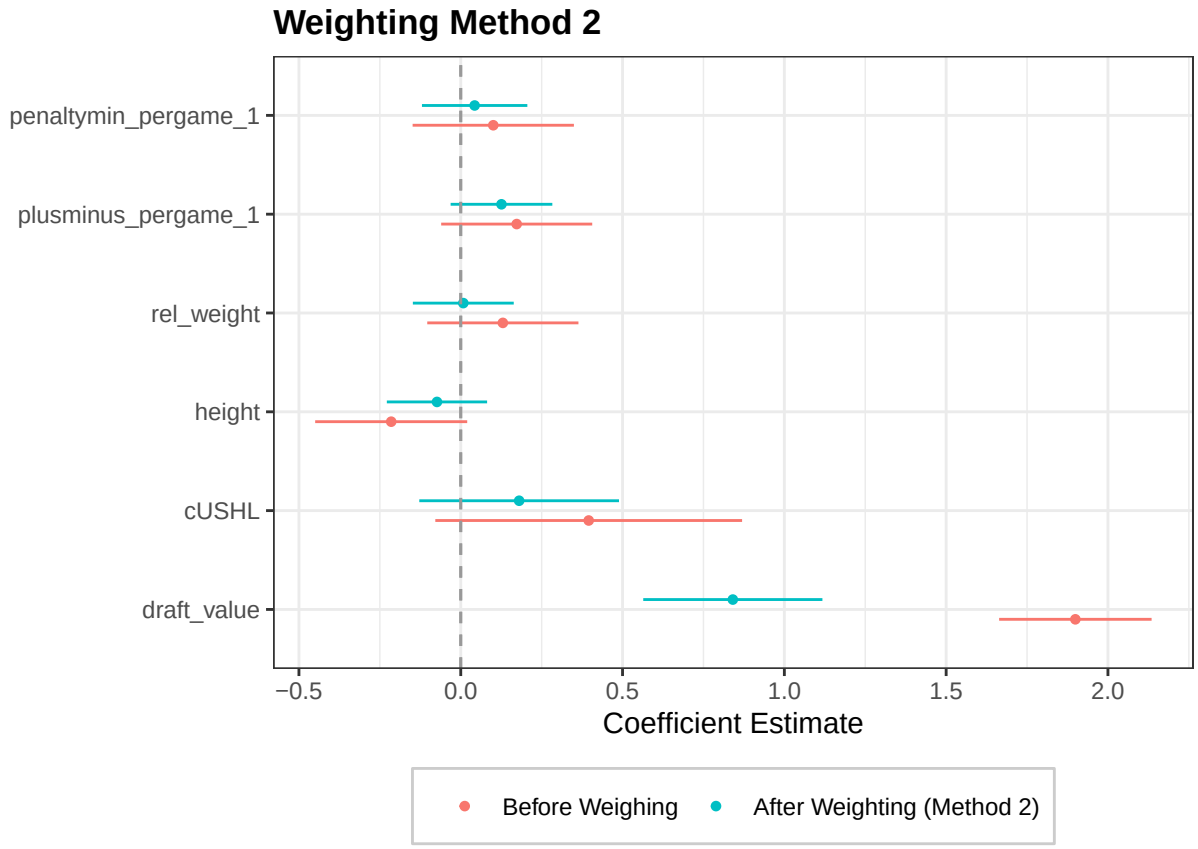
We then visualize the model coefficient estimates below

```
library(dotwhisker)
library(dplyr)
dwplot(list(mod1, mod2),
       vline = geom_vline(xintercept = 0,  linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Weighting Method 1") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Before
```

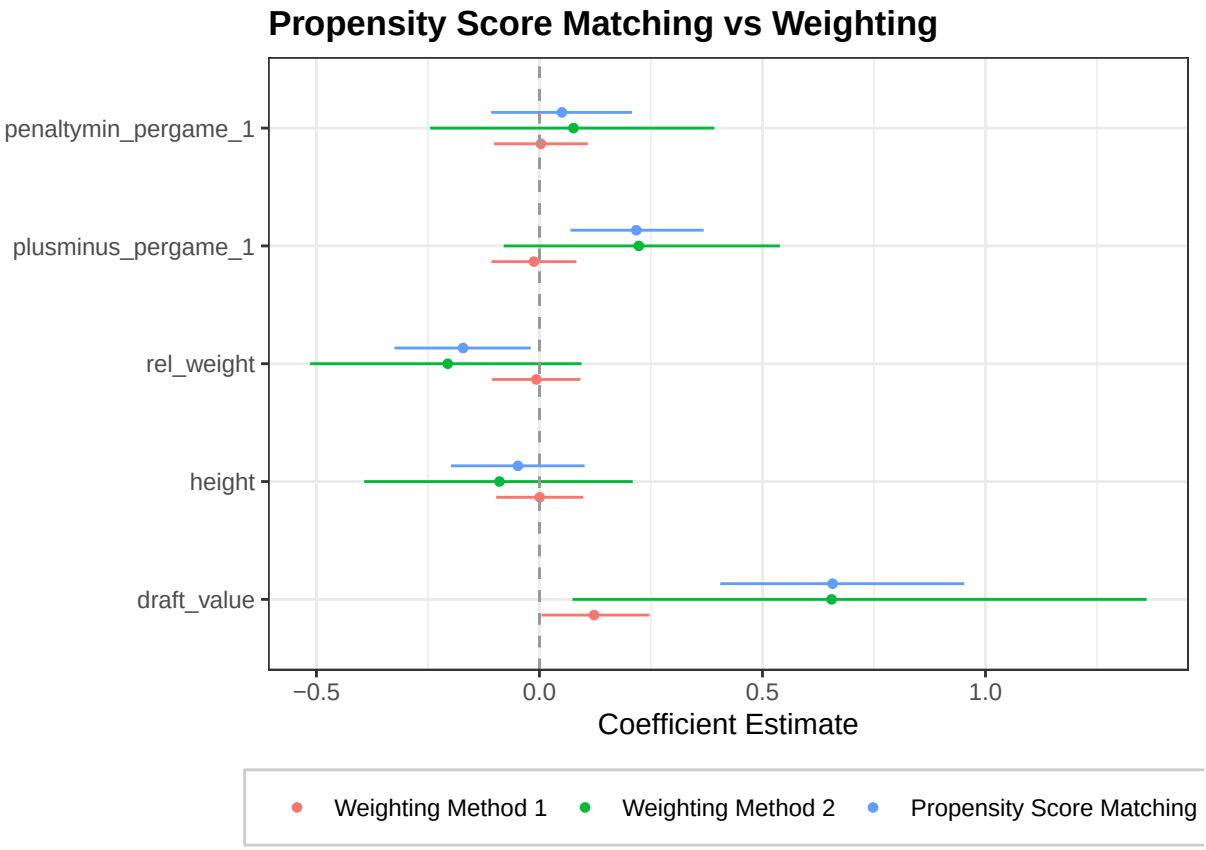



```
dwplot(list(mod1, mod3),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
```

```
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Weighting Method 2") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Before
```




```
library(dotwhisker)
library(dplyr)
dwplot(list(method_1, method_2, no_weights),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Propensity Score Matching vs Weighting") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Weight
```



Defense Men Propensity Score Weighting

We repeat the same process above for defense men. For players that didn't make it to the draft or did not get a draft position, their draft value was set to 27

```
updated_nhl_backward[which(is.na(updated_nhl_backward$Draftposition)), ]$draft_value = 27

updated_nhl_backward$draft = 1
updated_nhl_backward[which(is.na(updated_nhl_backward$Draftposition)), ]$draft = 0
```

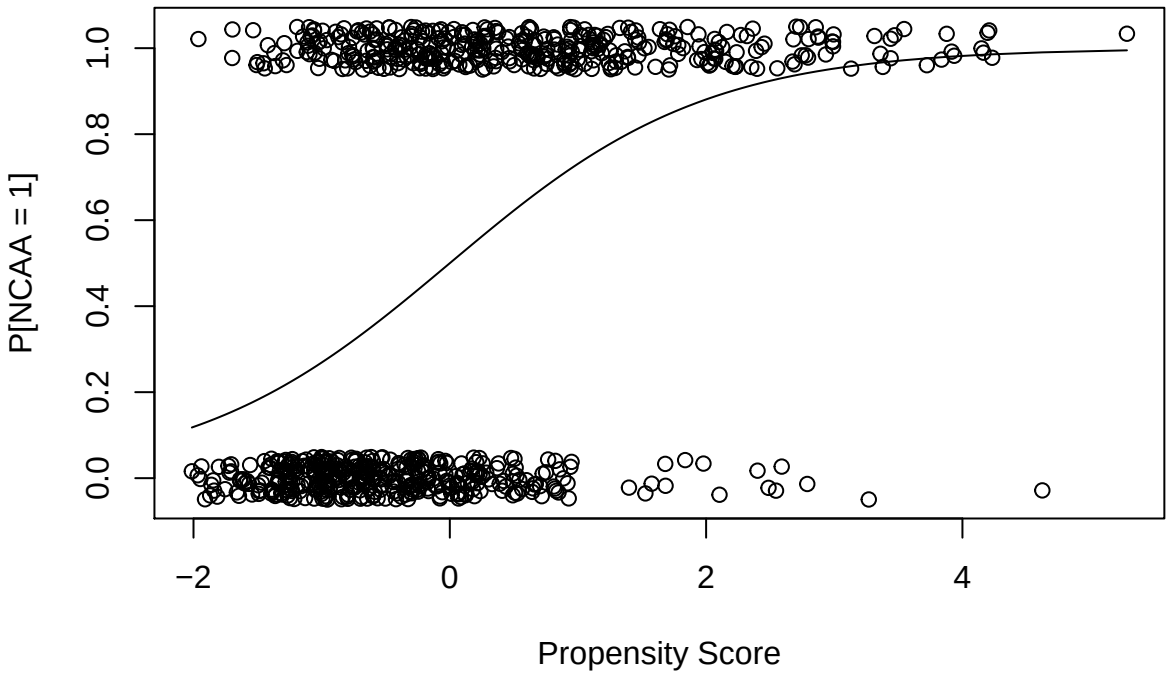
We predicted the treatment of each player (NCAA vs USHL at draft year 1) based on predictors in draft year 0 using a logistic model. The outcome of this model (P(T=NCAA)) became our propensity score. Players were then weighted based on this propensity score

```
library(Matching)
library(arm)


updated_nhl_backward[c('penaltymin_pergame_1', 'plusminus_pergame_1', 'rel_weight', 'height',   'assists
mod1 = glm(y ~  penaltymin_pergame_1 + plusminus_pergame_1  + rel_weight + height +goals_pergame_1 + as

summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + goals_pergame_1 + assists_pergame_1,
```

```
##     family = "binomial", data = updated_nhl_backward)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0441  -0.9142  -0.5304   0.9765   2.0462
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.02209    0.08357   0.264   0.7915
## penaltymin_pergame_1 -0.00441    0.09697  -0.045   0.9637
## plusminus_pergame_1   0.03731    0.09037   0.413   0.6797
## rel_weight           -0.07253    0.08556  -0.848   0.3966
## height                0.12428    0.08522   1.458   0.1447
## goals_pergame_1       0.27133    0.11327   2.396   0.0166 *
## assists_pergame_1     1.08137    0.12792   8.454   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1059.00  on 763  degrees of freedom
## Residual deviance:  874.09  on 757  degrees of freedom
##   (162 observations deleted due to missingness)
## AIC: 888.09
##
## Number of Fisher Scoring iterations: 4
```

```
complete_Cases =updated_nhl_backward[complete.cases(updated_nhl_backward[, c('y', 'penaltymin_pergame_1
p.scores = predict(mod1, type = 'link')
complete_Cases$psvalues = predict(mod1, type = 'response')
complete_Cases$weight.ATE = ifelse(complete_Cases$y == 1, 1/complete_Cases$psvalues, 1/(1- complete_Cas
# e*(1-e) fpr all
complete_Cases$weight.ATE2 = complete_Cases$psvalues*(1 - complete_Cases$psvalues)
plot(p.scores, jitter(updated_nhl_backward[complete.cases(updated_nhl_backward[, c('y', 'penaltymin_per
o.scores = sort(p.scores)
lines(o.scores, exp(o.scores)/(1 + exp(o.scores)))
```




```
matches = matching(z = complete_Cases$y, score = p.scores)
matched = complete_Cases[matches$match.ind, ]
```

Using the weighted data set, we then check to see if we removed selection if our coefficient estimates are zero when we predict treatment effect based on predictions in draft year 0 . Here we are comparing the two weighting methods with propensity score matching. We then use the weigthed data to predict the average number of games played in the NHL based on predictors in draft year 0 and league in draft year 1 (our treatment effect). This is where we can accurately assess the coefficient estimates and significance of the treatment

```
library(arm)
library(foreign)
#b.stats = balance(complete_Cases[, c('y', 'plusminus_pergame', 'goals_pergame', 'position_new', 'Penal
#plot(b.stats)

method_1 = glm(y ~  penaltymin_pergame_1 + plusminus_pergame_1  + rel_weight + height +goals_pergame_1 +

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
method_2 = glm(y ~   penaltymin_pergame_1 + plusminus_pergame_1  + rel_weight + height ++goals_pergame_

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
matching = glm(y ~  penaltymin_pergame_1 + plusminus_pergame_1  + rel_weight + height +goals_pergame_1 +



#######

# now let's compare estimates of the effect of "watched" on
# post-test score, controlling for all the factors we have
# been interested in...

# the unmatched regression analysis
```

```
mod1 = lm(games_per_year_in_nhl ~  penaltymin_pergame_1 + plusminus_pergame_1 + rel_weight + height + g
summary(mod1)
```

```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + goals_pergame_1 + assists_pergame_1 +
##     c, data = complete_Cases)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.184  -2.388  -0.963   0.315  48.537
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.72160    0.36492   4.718 2.84e-06 ***
## penaltymin_pergame_1 -0.10953    0.26663  -0.411 0.681339
## plusminus_pergame_1   0.15899    0.25138   0.632 0.527270
## rel_weight            0.81085    0.25131   3.226 0.001307 **
## height                0.90927    0.24755   3.673 0.000257 ***
## goals_pergame_1       0.87956    0.28497   3.086 0.002099 **
## assists_pergame_1     1.41508    0.31692   4.465 9.22e-06 ***
## cUSHL                -0.01532    0.53538  -0.029 0.977175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.58 on 755 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1041, Adjusted R-squared:  0.09579
## F-statistic: 12.53 on 7 and 755 DF,  p-value: 2.973e-15
```

```
coef(mod1)
```

```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##           1.72160087          -0.10953120           0.15899101
##           rel_weight               height      goals_pergame_1
##           0.81085324           0.90926511           0.87956144
##    assists_pergame_1                cUSHL
##           1.41507962          -0.01532257
```

```
# the matched regression analysis

mod2 = lm(games_per_year_in_nhl ~  penaltymin_pergame_1 + plusminus_pergame_1 +rel_weight + height + goa
summary(mod2)
```

```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + goals_pergame_1 + assists_pergame_1 +
##     c, data = complete_Cases, weights = weight.ATE)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -37.778  -3.534  -0.956   1.095 170.441
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.84199    0.39161   4.704 3.04e-06 ***
## penaltymin_pergame_1 -0.55547    0.25799  -2.153 0.031630 *
## plusminus_pergame_1   0.55766    0.24044   2.319 0.020643 *
## rel_weight             1.17827    0.27868   4.228 2.65e-05 ***
## height                 1.00476    0.28768   3.493 0.000506 ***
## goals_pergame_1        3.12510    0.31456   9.935  < 2e-16 ***
## assists_pergame_1    -0.02206    0.32294  -0.068 0.945556
## cUSHL                  0.18709    0.54597   0.343 0.731938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.72 on 755 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.174,  Adjusted R-squared:  0.1663
## F-statistic: 22.71 on 7 and 755 DF,  p-value: < 2.2e-16
```

```
coef(mod2)
```

```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##           1.84198714          -0.55547329           0.55765626
##           rel_weight               height      goals_pergame_1
##           1.17827441           1.00476364           3.12510296
##    assists_pergame_1                cUSHL
##          -0.02206045           0.18709125
```

```
mod3 = lm(games_per_year_in_nhl ~  penaltymin_pergame_1 + plusminus_pergame_1 +rel_weight + height +goal
summary(mod3)
```

```
##
## Call:
## lm(formula = games_per_year_in_nhl ~ penaltymin_pergame_1 + plusminus_pergame_1 +
##     rel_weight + height + goals_pergame_1 + assists_pergame_1 +
##     c, data = complete_Cases, weights = weight.ATE2)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3858 -0.9968 -0.4345  0.0889 21.5658
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.6839     0.3074   5.477 5.89e-08 ***
## penaltymin_pergame_1   0.0798     0.2436   0.328 0.743302
## plusminus_pergame_1    0.2892     0.2283   1.266 0.205813
## rel_weight             0.6219     0.2150   2.893 0.003928 **
## height                 0.7900     0.2143   3.686 0.000244 ***
## goals_pergame_1        0.5840     0.2850   2.049 0.040780 *
## assists_pergame_1      1.3555     0.3402   3.985 7.41e-05 ***
## cUSHL                 -0.1229     0.4513  -0.272 0.785545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

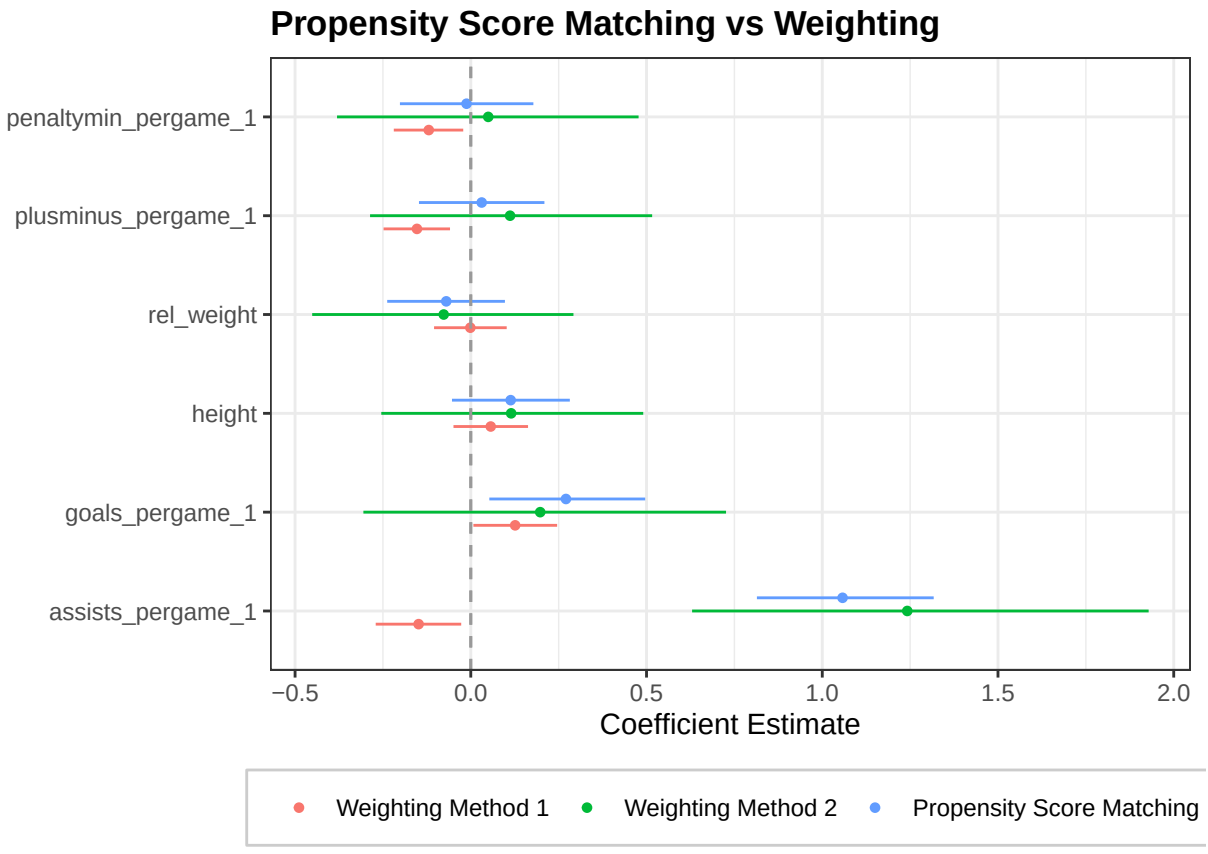
```
## Residual standard error: 2.51 on 755 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.07121,    Adjusted R-squared:  0.0626
## F-statistic:  8.27 on 7 and 755 DF,  p-value: 9.617e-10
```

```
coef(mod3)
```

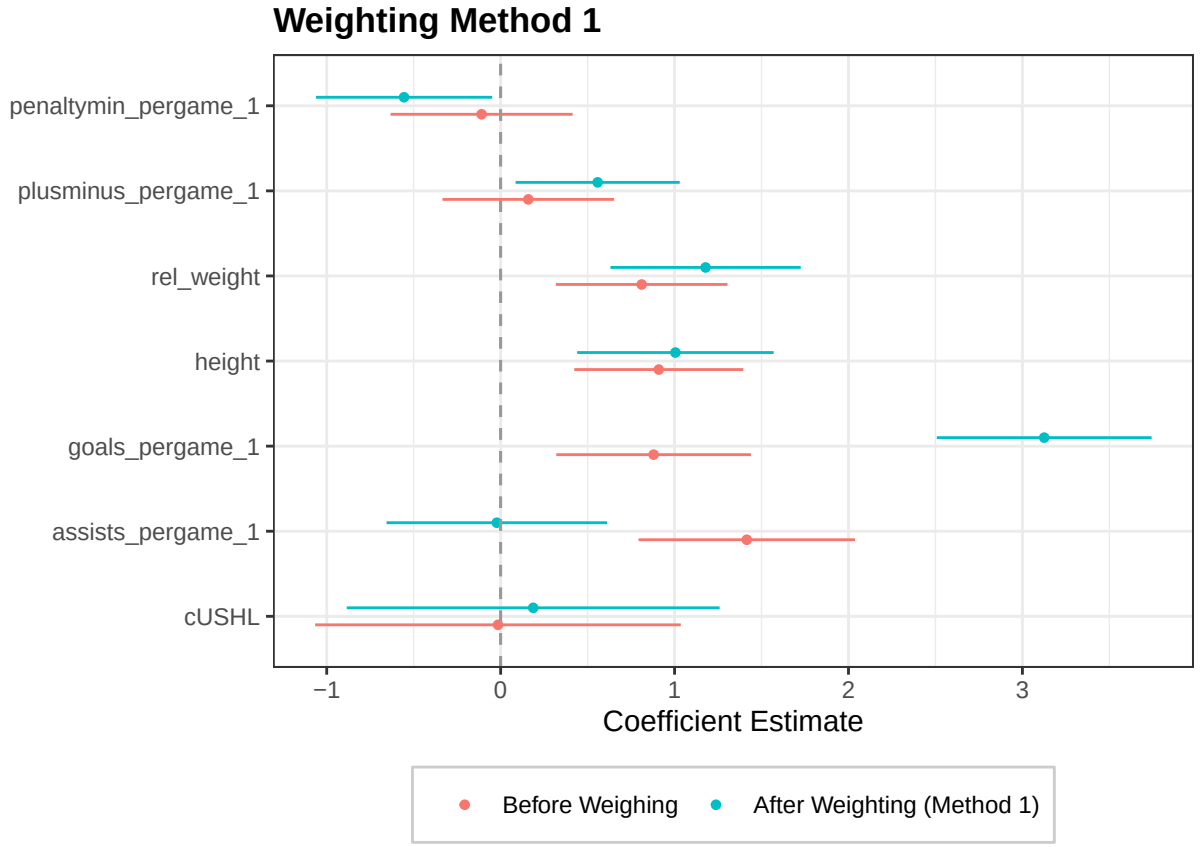
```
##          (Intercept) penaltymin_pergame_1  plusminus_pergame_1
##            1.6838546            0.0797980            0.2891549
##           rel_weight               height      goals_pergame_1
##            0.6218465            0.7899487            0.5839750
##    assists_pergame_1                 cUSHL
##            1.3554736           -0.1228524
```

We then visualize model coefficient estimates

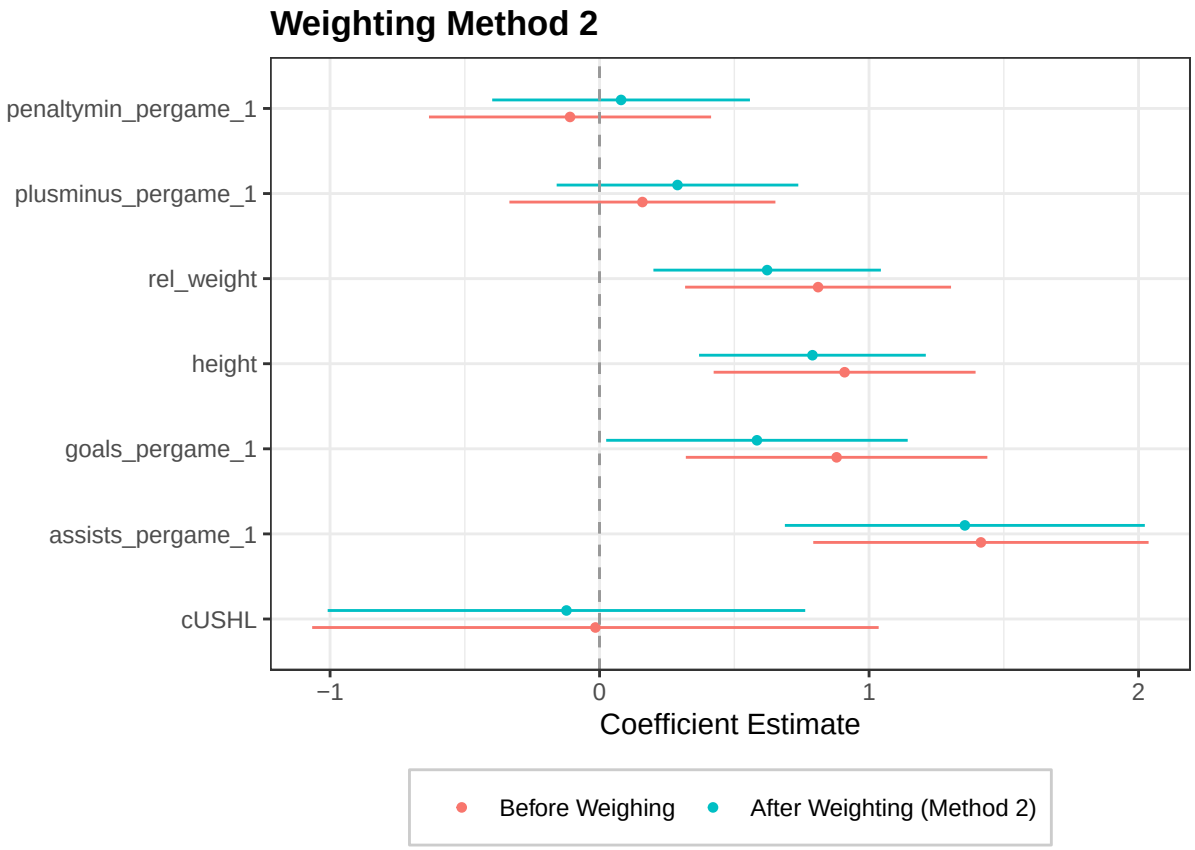
```
library(dotwhisker)
library(dplyr)
dwplot(list(method_1, method_2, matching),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Propensity Score Matching vs Weighting") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Weight
```



```
dwplot(list(mod1, mod2),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Weighting Method 1") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Before
```




```
dwplot(list(mod1, mod3),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) +
    theme_bw() + xlab("Coefficient Estimate") + ylab("") +
    geom_vline(xintercept = 0, colour = "grey60", linetype = 2) +
    ggtitle("Weighting Method 2") +
  theme(legend.position="bottom")+
    theme(plot.title = element_text(face="bold"),


          legend.background = element_rect(colour="grey80"),
          legend.title = element_blank()) +   scale_color_discrete(name = "Method", labels = c("Before
```


Weighting Method 2
penaltymin_pergame_1
plusminus_pergame_1
rel_weight
height
goals_pergame_1
assists_pergame_1
cUSHL
−1
0
1
2
Coefficient Estimate
Before Weighing
After Weighting (Method 2)

BART

```
library(dplyr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(alluvial)
library(ggalluvial)
library(tidyr)
library(tidyverse)
library(lubridate)

library(date)

leagues <- read.csv("withNextLeague_updated.csv") %>%
  select(Player, Season, NextLeague, ID)

# Calculate players' birth year
bio <- read_csv("complete_bio_updated_withID.csv")%>%
  mutate(BirthYear = as.integer(substring(DateofBirth,
                                          nchar(DateofBirth)-3,
                                          nchar(DateofBirth))))

# Transform height and weight into inches and pounds
bio$feetandinches <- str_split_fixed(bio$Height," / ",n=2)[,1] %>%
  str_remove("\"")
bio$feet <- str_split_fixed(bio$feetandinches,"'",n=2)[,1]
bio$inch <- str_split_fixed(bio$feetandinches,"'",n=2)[,2]
bio$Height_inch <- as.integer(bio$feet)*12+as.integer(bio$inch)
bio$Weight_lbs <- str_split_fixed(bio$Weight," / ",n=2)[,1] %>%
  str_remove(" lbs") %>%
  as.integer()

stat <- read_csv("complete_stat_updated_withID.csv") %>%
  left_join(leagues, by = c("ID", "Season")) %>%
  mutate(Season = as.integer(substr(Season, 0,4))) %>%
  fill(Season)

# Calculate draft year for each player each season
nhl_data <- bio %>% left_join(stat, by = "ID" )%>%
  mutate(Age = as.numeric((as.Date(paste(Season, "-09-15",
                                         sep = ""))-mdy(DateofBirth))/365.25),
         DraftYear = as.integer(Age - 18 + 1),
         Performance = as.integer(TotalPoints)/as.integer(Games)) %>%
  #select(Player, Age, League, Performance) %>%
  mutate(Performance = replace_na(Performance, 0)) %>%
  mutate(GeneralPosition =
                      case_when(Position %in% c("RW", "LW",
```

```
                                                  "C", "C/RW",
                                                  "C/LW", "F",
                                                  "LW/C", "LW/RW",
                                                  "RW/LW", "C/W",
                                                  "LW/D","RW/D","W/C", "RW/C", "C/D", "W", "F/D", "RF", "
```

```
write.csv(nhl_data, "merged_with_id.csv")
```

```
unique(nhl_data$GeneralPosition)
```

```
## [1] "Forward" "Defense" NA
```

```
nhl_data %>% filter(is.na(GeneralPosition))
```

```
## # A tibble: 1 x 30
##   Player Position DateofBirth Height Weight Nation Shoots     ID BirthYear
##   <chr>  <chr>    <chr>       <chr>  <chr>  <chr>  <chr>   <dbl>     <int>
## 1 Rob S~ -        -           - / -  - / -  -      -      380884        NA
## # ... with 21 more variables: feetandinches <chr>, feet <chr>, inch <chr>,
## #   Height_inch <dbl>, Weight_lbs <int>, Player.x <chr>, Season <int>,
## #   Team <chr>, League <chr>, Games <chr>, Goals <chr>, Assists <chr>,
## #   TotalPoints <chr>, PenaltyMinutes <chr>, PlusMinus <dbl>, Player.y <chr>,
## #   NextLeague <chr>, Age <dbl>, DraftYear <int>, Performance <dbl>,
## #   GeneralPosition <chr>
```

BART

```
library(readr)
library(dplyr)
options(java.parameters = "-Xmx5g")
library(bartMachine)
set_bart_machine_num_cores(4)
```

```
## bartMachine now using 4 cores.
```

```
nhl_data <- read_csv("merged_with_id.csv")
nhl_data <- nhl_data %>% select(-X1)
nhl_data <- nhl_data[!duplicated(nhl_data),]


# Calculate NHL games per season
average_nhl <- nhl_data %>%
  filter(League == "NHL") %>%
  group_by(ID) %>%
  summarize(gp = sum(as.integer(Games)),
            num.seasons = n(),
```

```
    nhl.games.per.season = gp/num.seasons, na.rm = TRUE) %>%
  select(ID, nhl.games.per.season)

final_nhl_data <- nhl_data %>%
  mutate(Goals = as.integer(Goals),
         Games = as.integer(Games),
         Assists = as.integer(Assists),
         TotalPoints = as.integer(TotalPoints)) %>%
  left_join(average_nhl)


final_nhl_data %>%
  mutate(PenaltyMinutes = as.numeric(gsub(PenaltyMinutes, "-", "0")))
```

```
## # A tibble: 256,899 x 31
##    Player Position DateofBirth Height Weight Nation Shoots    ID BirthYear
##    <chr>  <chr>    <chr>       <chr>  <chr>  <chr>  <chr>  <dbl>     <dbl>
##  1 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  2 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  3 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  4 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  5 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  6 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  7 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  8 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
##  9 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
## 10 Jarom~ RW       Jul 01, 19~ "6'1\~ 209 l~ Canada R       9036      1977
## # ... with 256,889 more rows, and 22 more variables: feetandinches <chr>,
## #   feet <dbl>, inch <dbl>, Height_inch <dbl>, Weight_lbs <dbl>,
## #   Player.x <chr>, Season <dbl>, Team <chr>, League <chr>, Games <int>,
## #   Goals <int>, Assists <int>, TotalPoints <int>, PenaltyMinutes <dbl>,
## #   PlusMinus <dbl>, Player.y <chr>, NextLeague <chr>, Age <dbl>,
## #   DraftYear <dbl>, Performance <dbl>, GeneralPosition <chr>,
## #   nhl.games.per.season <dbl>
```

```
# Select observations with players in USHL in their draft year 0
bart.dat <- final_nhl_data %>%
  filter(DraftYear == 0 & League == "USHL") %>%
  group_by(ID) %>%
  select(Player,  Games, League, Goals, Assists,
         PenaltyMinutes, PlusMinus,
         Position, Nation, Shoots, Performance,
         Height_inch, Weight_lbs, NextLeague,
         nhl.games.per.season, GeneralPosition,ID) %>%
  filter(nhl.games.per.season != Inf) %>%
  as.data.frame()
```

forward, defense

```
forward.dat <-  bart.dat %>% filter(GeneralPosition == "Forward")

X.forward <- forward.dat %>%
```

```
  select(Games, Goals, Assists, PenaltyMinutes,
                        Shoots, Performance, NextLeague,
                        Height_inch, Weight_lbs) %>%
  as.data.frame()

Y.forward <- forward.dat$nhl.games.per.season

# Run BART with regularization parameters k=25,50,100,150 for forwardds
forward_bart_150<- bartMachine(X.forward[,names(X.forward)!="Player"], Y.forward, verbose = FALSE,
                               k=150,

serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

```
forward_bart_100<- bartMachine(X.forward[,names(X.forward)!="Player"], Y.forward, verbose = FALSE,
                               k=100,

serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

```
forward_bart_50<- bartMachine(X.forward[,names(X.forward)!="Player"], Y.forward, verbose = FALSE,
                               k=50,

serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

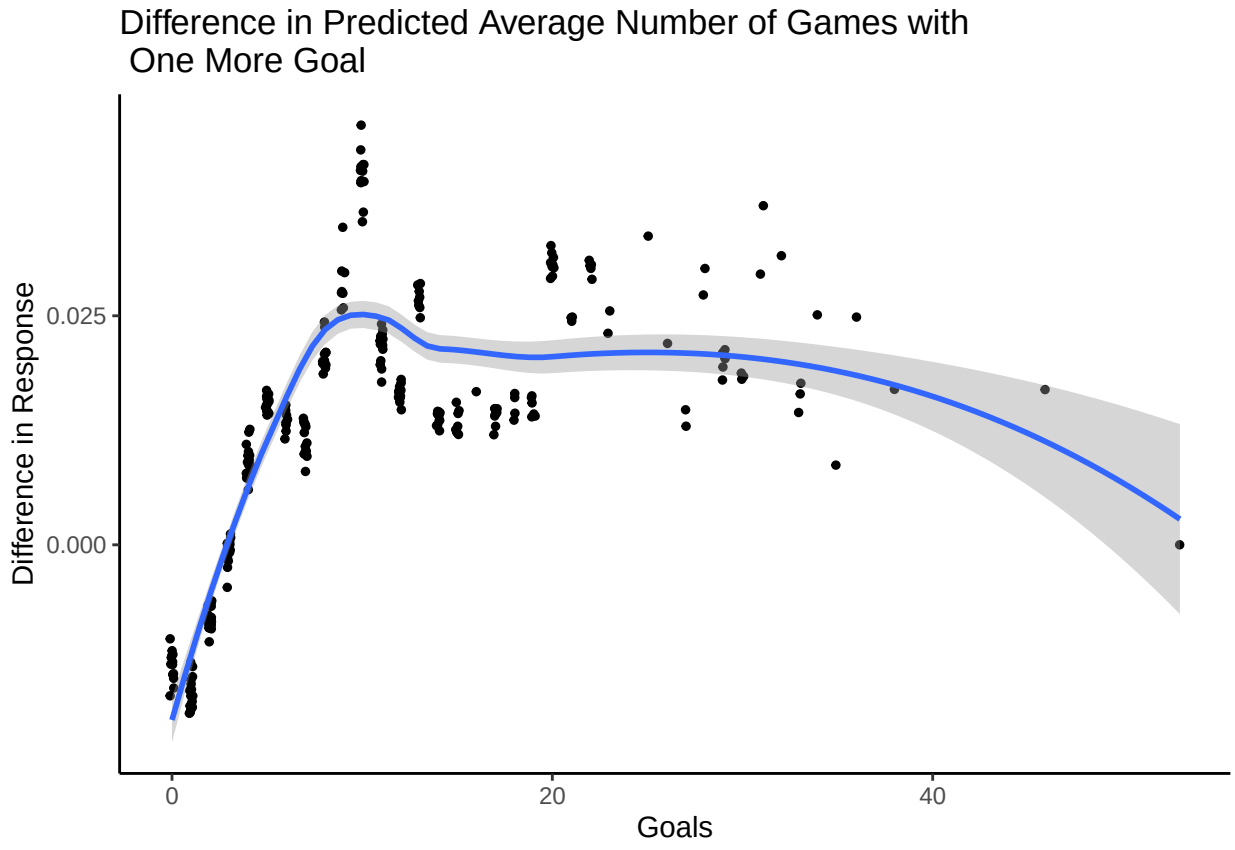
```
forward_bart_25<- bartMachine(X.forward[,names(X.forward)!="Player"], Y.forward, verbose = FALSE,
                               k=25,

serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

```
# Run BART for defensemen
defense.dat <-  bart.dat %>% filter(GeneralPosition == "Defense")

X.defense <- defense.dat %>%
  select(Games, Goals, Assists, PenaltyMinutes,
                        Shoots, Performance, NextLeague,
                        Height_inch, Weight_lbs) %>%
  as.data.frame()

Y.defense <- defense.dat$nhl.games.per.season

defense_bart_25<- bartMachine(X.defense[,names(X.defense)!="Player"], Y.defense, verbose = FALSE,
                               k=25,

serialize = TRUE, use_missing_data = TRUE)
```

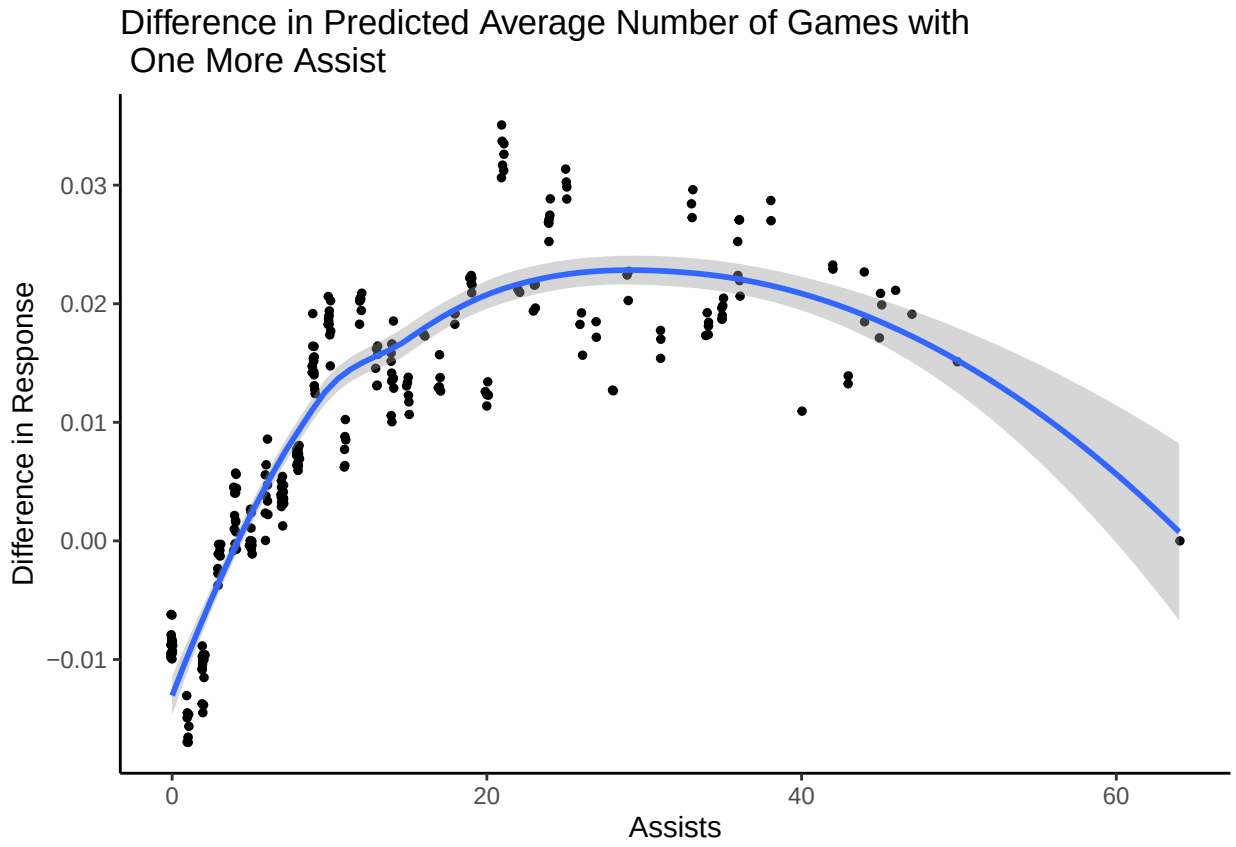
```
## serializing in order to be saved for future R sessions...done
```

```
# Calculate how one more goal affects players' predicted average games played per season
X.goals <- X.forward %>% mutate(Goals = Goals + 1)
y.new.goals <- predict(forward_bart_25, X.goals)

y.goals <-  predict(forward_bart_25, X.forward)
df <- cbind(X.forward$Goals, y.new.goals, y.goals, X.forward$NextLeague) %>% as.data.frame()
colnames(df) <- c("Goals","y.new","y.old","NextLeague")
df <- df %>% mutate(Goals = as.integer(Goals),
             y.new = as.double(y.new),
             y.old = as.double(y.old),
             diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Goals, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games with \n One More Goal"),
           y="Difference in Response")
```



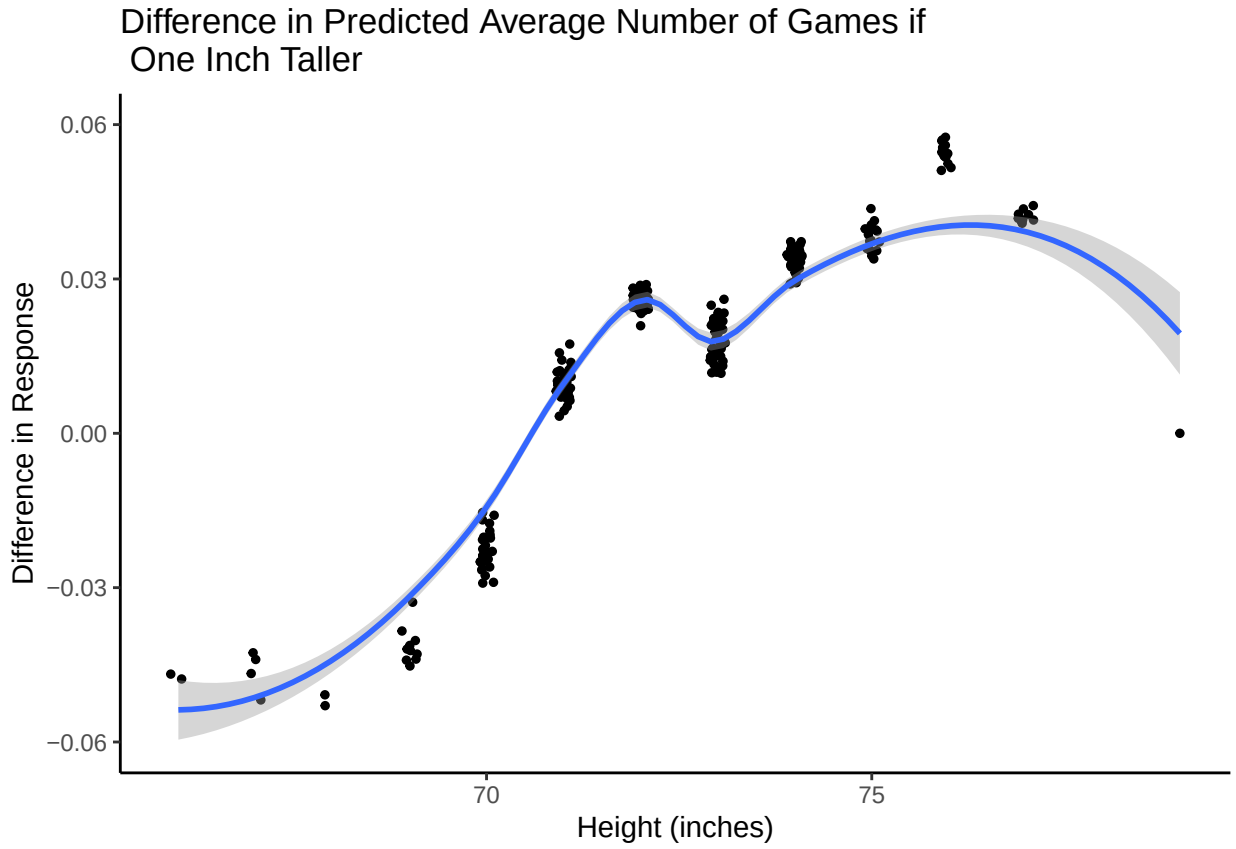

```
# Calculate how one more assist affects players' predicted average games played per season
X.assists <- X.forward %>% mutate(Assists = Assists + 1)
y.new.assists <- predict(forward_bart_25, X.assists)
```

```
y.assists <-  predict(forward_bart_25, X.forward)
df <- cbind(X.forward$Assists, y.new.assists, y.assists, X.forward$NextLeague) %>% as.data.frame()
colnames(df) <- c("Assists","y.new","y.old","NextLeague")
df <- df %>% mutate(Assists = as.integer(Assists),
             y.new = as.double(y.new),
             y.old = as.double(y.old),
             diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Assists, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games with \n One More Assist"),
           y="Difference in Response")
```



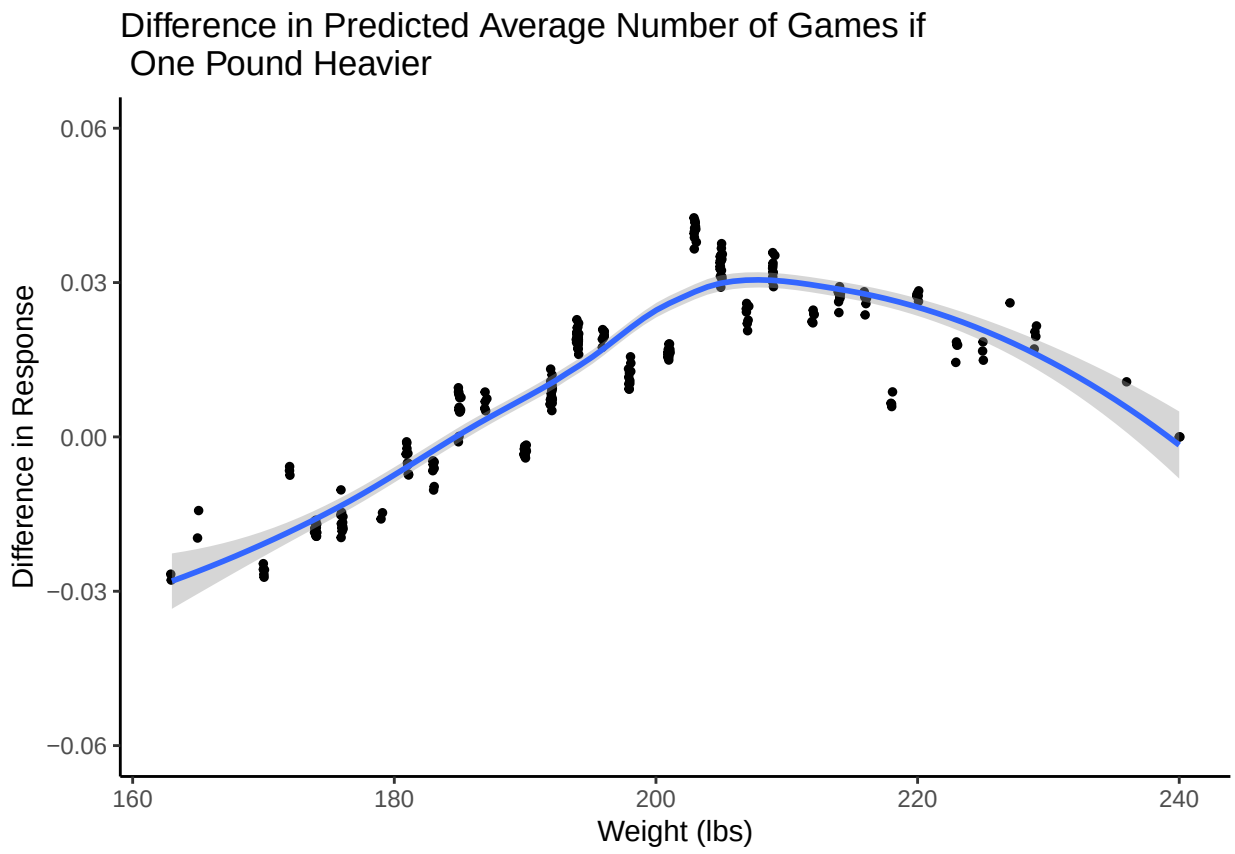

```
# Calculate how one more inch affects players' predicted average games played per season
X.hgt <- X.forward %>% mutate(Height_inch = Height_inch + 1)
y.new.hgt <- predict(forward_bart_25, X.hgt)

y.hgt <-  predict(forward_bart_25, X.forward)
df <- cbind(X.forward$Height_inch, y.new.hgt, y.hgt, X.forward$NextLeague) %>% as.data.frame()
colnames(df) <- c("Height_inch","y.new","y.old","NextLeague")
df <- df %>% mutate(Height_inch = as.integer(Height_inch),
             y.new = as.double(y.new),
```

```
            y.old = as.double(y.old),
            diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Height_inch, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games if \n One Inch Taller"),
          y="Difference in Response",x="Height (inches)")+ylim(-0.06,0.06)
```

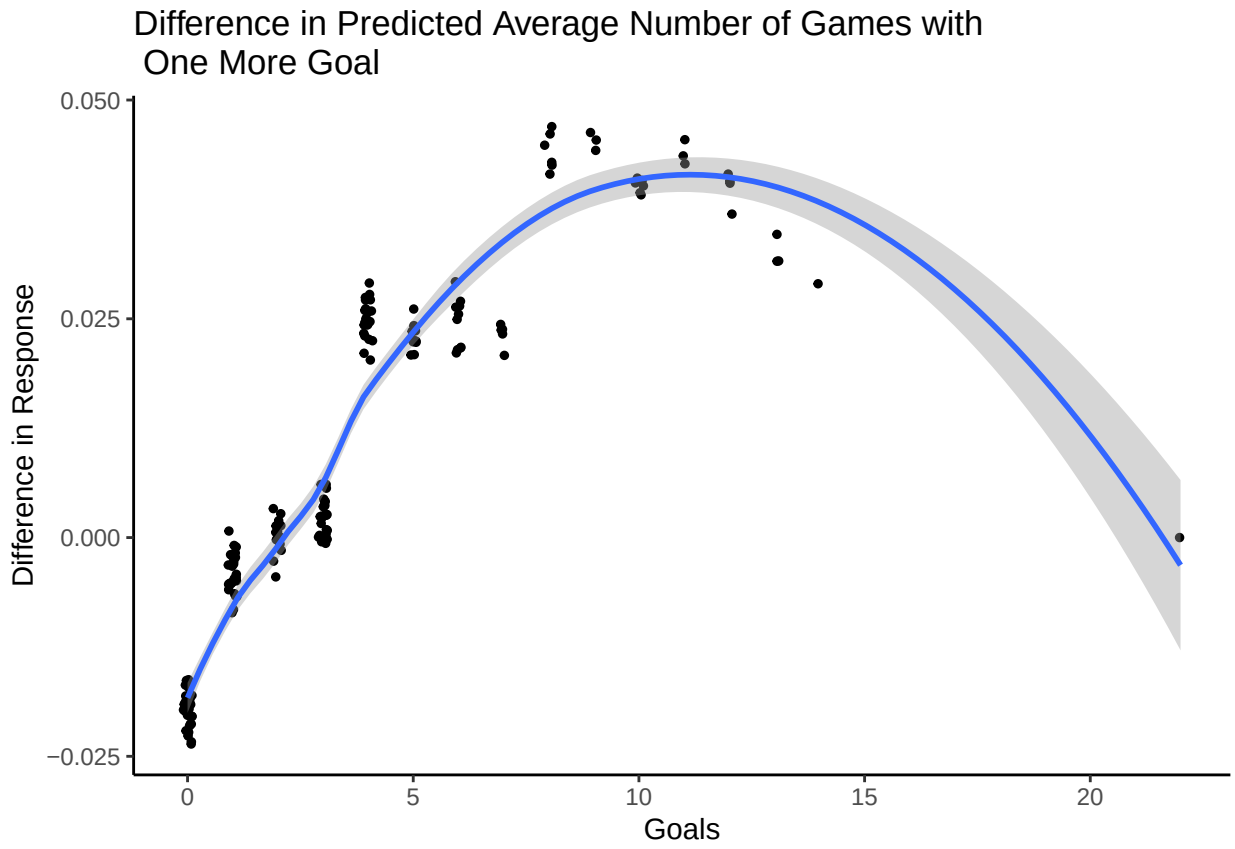



```
# Calculate how one more pound affects players' predicted average games played per season
X.wgt <- X.forward %>% mutate(Weight_lbs = Weight_lbs + 1)
y.new.wgt <- predict(forward_bart_25, X.wgt)

y.wgt <-  predict(forward_bart_25, X.forward)
df <- cbind(X.forward$Weight_lbs, y.new.wgt, y.wgt, X.forward$NextLeague) %>% as.data.frame()
colnames(df) <- c("Weight_lbs","y.new","y.old","NextLeague")
df <- df %>% mutate(Weight_lbs = as.integer(Weight_lbs),
            y.new = as.double(y.new),
            y.old = as.double(y.old),
            diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
```

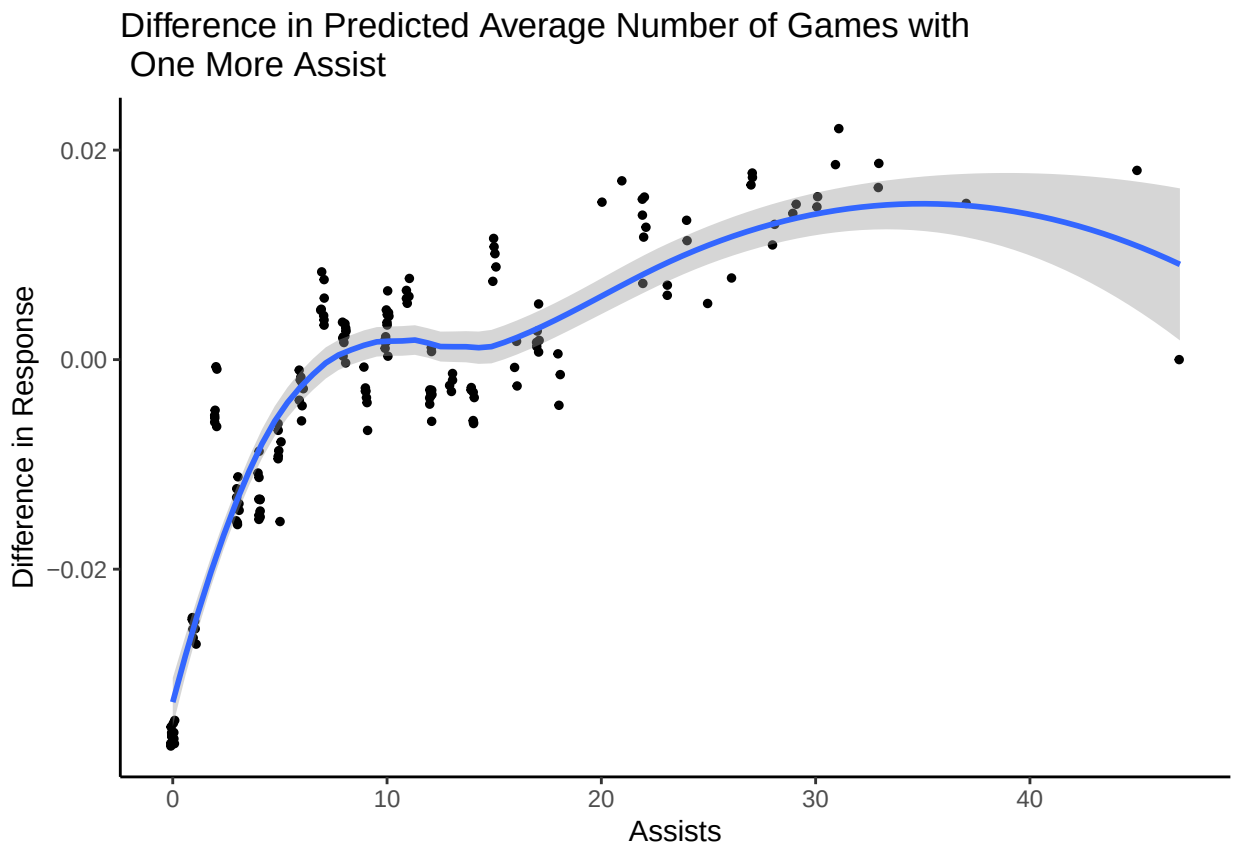
```r
ggplot(df, aes(x = Weight_lbs, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games if \n One Pound Heavier"),
           y="Difference in Response",x="Weight (lbs)")+ylim(-0.06,0.06)
```



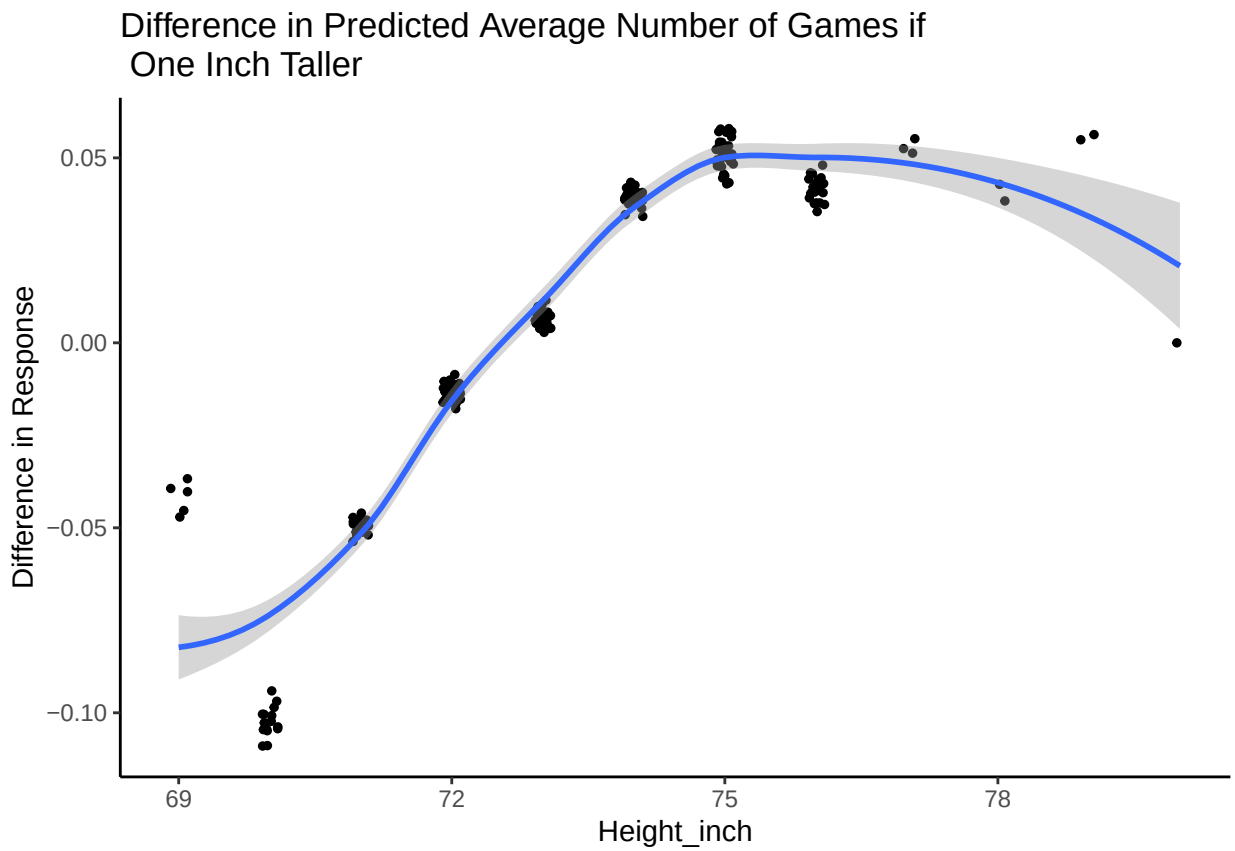

```r
# Calculate how one more goal affects players' predicted average games played per season for defensemen
X.goals <- X.defense %>% mutate(Goals = Goals + 1)
y.new.goals <- predict(defense_bart_25, X.goals)

y.goals <-  predict(defense_bart_25, X.defense)
df <- cbind(X.defense$Goals, y.new.goals, y.goals, X.defense$NextLeague) %>% as.data.frame()
colnames(df) <- c("Goals","y.new","y.old","NextLeague")
df <- df %>% mutate(Goals = as.integer(Goals),
              y.new = as.double(y.new),
              y.old = as.double(y.old),
              diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Goals, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games with \n One More Goal"),
           y="Difference in Response")
```



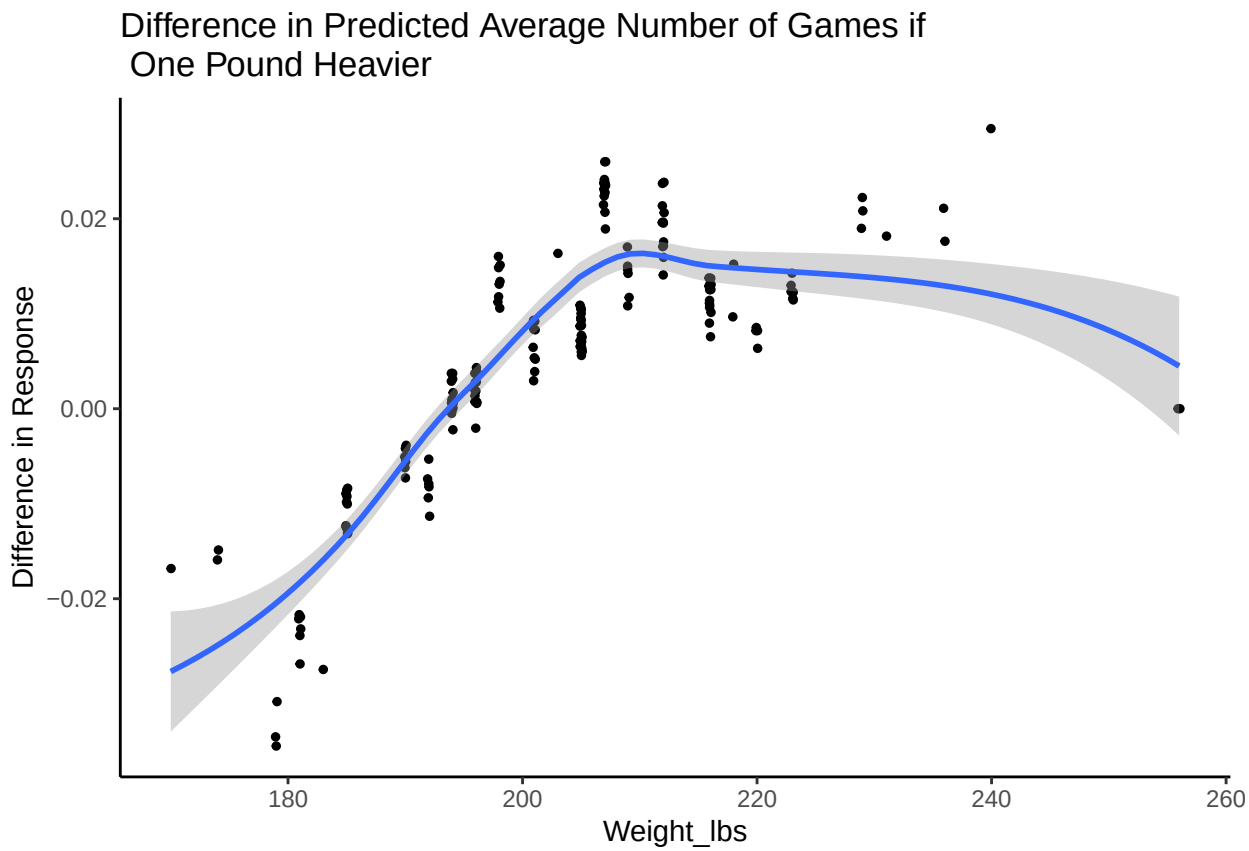

```
# Calculate how one more assist affects players' predicted average games played per season for defensem
X.assists <- X.defense %>% mutate(Assists = Assists + 1)
y.new.assists <- predict(defense_bart_25, X.assists)

y.assists <-  predict(defense_bart_25, X.defense)
df <- cbind(X.defense$Assists, y.new.assists, y.assists, X.defense$NextLeague) %>% as.data.frame()
colnames(df) <- c("Assists","y.new","y.old","NextLeague")
df <- df %>% mutate(Assists = as.integer(Assists),
              y.new = as.double(y.new),
              y.old = as.double(y.old),
              diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Assists, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games with \n One More Assist"),
           y="Difference in Response")
```




```r
# Calculate how one more inch affects players' predicted average games played per season for defensemen
X.hgt <- X.defense %>% mutate(Height_inch = Height_inch + 1)
y.new.hgt <- predict(defense_bart_25, X.hgt)

y.hgt <-  predict(defense_bart_25, X.defense)
df <- cbind(X.defense$Height_inch, y.new.hgt, y.hgt, X.defense$NextLeague) %>% as.data.frame()
colnames(df) <- c("Height_inch","y.new","y.old","NextLeague")
df <- df %>% mutate(Height_inch = as.integer(Height_inch),
              y.new = as.double(y.new),
              y.old = as.double(y.old),
              diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Height_inch, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games if \n One Inch Taller"),
           y="Difference in Response")
```




```
# Calculate how one more pound affects players' predicted average games played per season for defenseme
X.wgt <- X.defense %>% mutate(Weight_lbs = Weight_lbs + 1)
y.new.wgt <- predict(defense_bart_25, X.wgt)

y.wgt <-  predict(defense_bart_25, X.defense)
df <- cbind(X.defense$Weight_lbs, y.new.wgt, y.wgt, X.defense$NextLeague) %>% as.data.frame()
colnames(df) <- c("Weight_lbs","y.new","y.old","NextLeague")
df <- df %>% mutate(Weight_lbs = as.integer(Weight_lbs),
              y.new = as.double(y.new),
              y.old = as.double(y.old),
              diff = y.new - y.old)

#df <- df %>% mutate(Goals = as.integer(Goals))
#colnames(df) <- c("Goals", "y.new", "y", "NextLeague", "Diff")
ggplot(df, aes(x = Weight_lbs, y = diff)) +
  geom_jitter(size = 1, width = 0.1) +
    geom_smooth() + theme_classic() +
    labs(title = paste("Difference in Predicted Average Number of Games if \n One Pound Heavier"),
           y="Difference in Response")
```




```
# Calculate CATE for forwards
X.forward.NCAA <- X.forward %>% mutate(NextLeague="NCAA")
X.forward.USHL <- X.forward %>% mutate(NextLeague="USHL")
mean(predict(forward_bart_25,X.forward.NCAA)-predict(forward_bart_25,X.forward.USHL))
```

```
## [1] 0.07053334
```

```
# Calculate CATE for defensemen
X.defense.NCAA <- X.defense %>% mutate(NextLeague="NCAA")
X.defense.USHL <- X.defense %>% mutate(NextLeague="USHL")
mean(predict(defense_bart_25,X.defense.NCAA)-predict(defense_bart_25,X.defense.USHL))
```

```
## [1] 0.03253917
```

BART Variable Immportance

```
library(readr)
library(dplyr)
options(java.parameters = "-Xmx5g")
library(bartMachine)
set_bart_machine_num_cores(4)
```

```
## bartMachine now using 4 cores.
```

We look to investigate the variable importance in the BART model.
We first look to filter out those players who are entering their draft year and has played in the USHL.

```
nhl_data <- read_csv("final_nhl_data.csv")
nhl_data <- nhl_data %>% select(-X1)
nhl_data <- nhl_data[!duplicated(nhl_data),]
# remove characters
nhl_data$Games = as.numeric(nhl_data$Games)
nhl_data$Goals = as.numeric(nhl_data$Goals)
nhl_data$Assists = as.numeric(nhl_data$Assists)
nhl_data$TotalPoints = as.numeric(nhl_data$TotalPoints)
nhl_data$PenaltyMinutes = as.numeric(nhl_data$PenaltyMinutes)
### Players in their draft year, who played in the USHL
final_nhl_data_USHL <- nhl_data %>%
  filter(DraftYear  == 0 & League == "USHL")
# developmental leagues that we are concerned with right now
NextLeagueWanted <- c("NCAA", "USHL")
```

Then we manipulate the data to obtain the variables we want to investigate (these include the next league that the prospects have gone to play in, total penalty minutes played in the season, the plus minus rating, the position - either forwardsd or defensemen - height, weight and draft value, as well as the response variable, which is the number of nhl games played per season) and store in a new data frame.
Again, it is worthwhile to note that we took the relative weight, compared to the height of the player.

```
# filter for required data
USHL.bart_dat <- final_nhl_data_USHL %>%
  group_by(Player) %>%
  # only look for those who played in the next league that we are interested in
  filter(NextLeague %in% NextLeagueWanted) %>%
  select(Player, NextLeague, PenaltyMinutes, PlusMinus,
         GeneralPosition, Height_cm, RelWeight, DraftValue,
         nhl.games.per.season) %>%
  filter(nhl.games.per.season != Inf) %>%
  as.data.frame()
# binary variable for the next league, 1 if in the USHL, 0 if in the NCAA
USHL.bart_dat$NextLeagueUSHL <- ifelse(USHL.bart_dat$NextLeague == "USHL",
```

```
                                        1, 0)
# binary variable for the position. 1 if forwards, 0 if defensemen
USHL.bart_dat$Forwards <- ifelse(USHL.bart_dat$GeneralPosition == "Forward",
                                 1, 0)
```

We then organize the data to be used as inputs in bartmachine function.

```
# X and Y for bart
X <- USHL.bart_dat %>%
  select(NextLeagueUSHL, PenaltyMinutes, PlusMinus,
         Forwards, Height_cm, RelWeight, DraftValue) %>%
  as.data.frame()
Y <- USHL.bart_dat$nhl.games.per.season
# fit BART model
USHL.bart <- bartMachine(X[,names(X)!="Player"], Y, k = 25, seed = 726,
                         verbose = FALSE, serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

We can plot the convergence plots and the assumptions plots to investigate the model fitted.

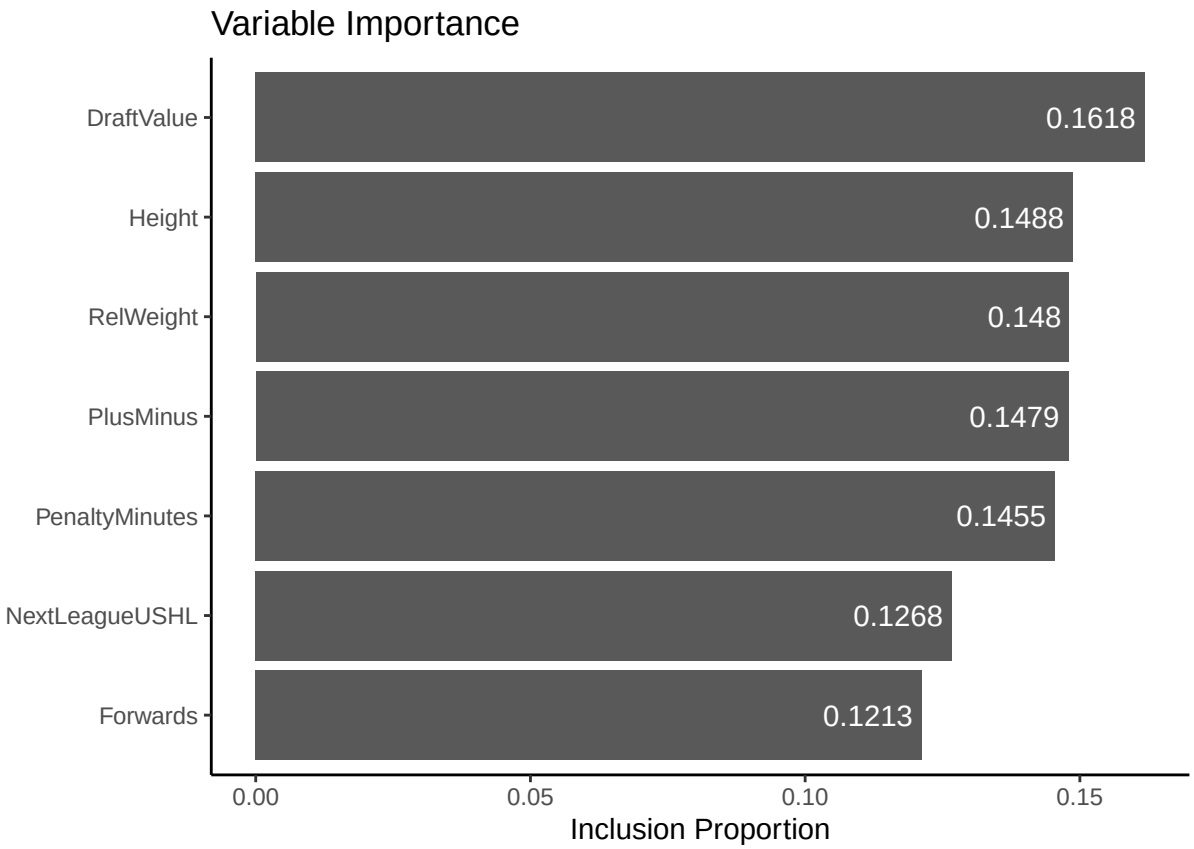
```
check_bart_error_assumptions(USHL.bart)
```

We notice that the residuals seem to demonstrate close to Normality assumptions, as the Q-Q plot (first plot) demonstrates close to a straight line. The plot of the residuals against the fitted values demonstrate that the residuals are centered around zero, with close to random pattern.
Next, we investigate the variable importance of the fitted model.
BartMachine package has the function `investigate_var_importance()`, which inherently measures the amount of inclusion proportion that each variable shows in predicting for the response varaible.

```
# variable importance
USHL.output <- investigate_var_importance(USHL.bart, num_replicates_for_avg = 20, plot = FALSE)
```

```
## .....................
```

```
USHL.avg_var_props <- USHL.output$avg_var_props
USHL.props_df <- data.frame(variables = names(USHL.avg_var_props),
                            props = USHL.avg_var_props)
USHL.props_df$variables[which(USHL.props_df$variables == "Height_cm")] = "Height"
library(ggplot2)
ggplot(data = USHL.props_df, aes(x = reorder(variables, props), y = props)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Variable Importance",
       x = "",
       y = "Inclusion Proportion") +
  theme_classic() +
  geom_text(aes(label=round(props, 4)), hjust = 1.1, col = "white")
```



We notice here that the draft value has the highest inclusion proportion, whereas the developmental path of USHL or NCAA has second-least proportion. General position seems to demonstrate the least amount of inclusion proportion in predicting for the number of NHL games played per season.
We also look to separate by players and investigate further.
For the forwards:

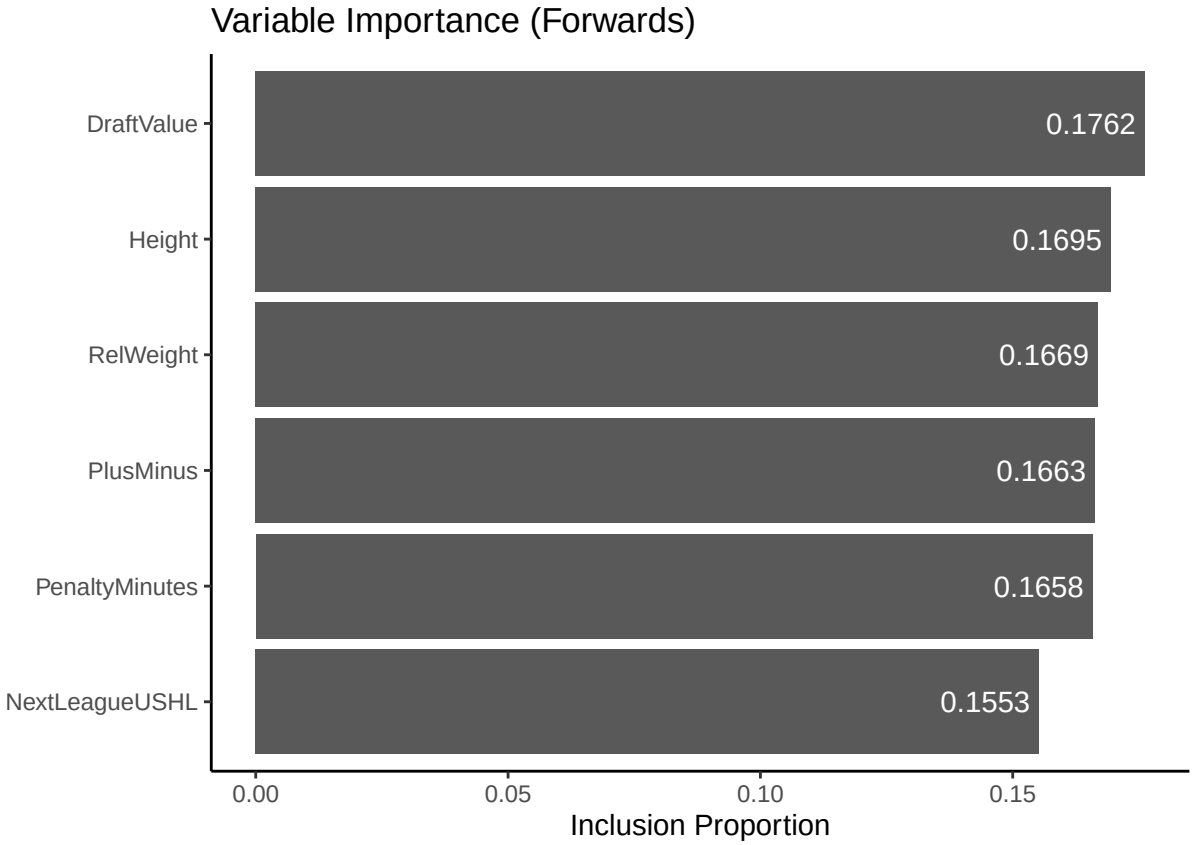
```
# forwards
USHL_bart.dat_forwards <- USHL.bart_dat %>%
  filter(GeneralPosition == "Forward")
X_forwards <- USHL_bart.dat_forwards %>%
  select(NextLeagueUSHL, PenaltyMinutes, PlusMinus,
         Height_cm, RelWeight, DraftValue) %>%
  as.data.frame()
Y_forwards <- USHL_bart.dat_forwards$nhl.games.per.season
USHL_forward.bart <- bartMachine(X_forwards[,names(X_forwards)!="Player"], Y_forwards, k = 25,  seed = 
                                 verbose = FALSE, serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

```
USHL_forwards.output <- investigate_var_importance(USHL_forward.bart, num_replicates_for_avg = 20, plot
```

```
## ..................
```

```
USHL_forwards.avg_var_props <- USHL_forwards.output$avg_var_props
USHL_forwards.props_df <- data.frame(variables = names(USHL_forwards.avg_var_props),
                                     props = USHL_forwards.avg_var_props)
USHL_forwards.props_df$variables[which(USHL_forwards.props_df$variables == "Height_cm")] = "Height"
ggplot(data = USHL_forwards.props_df, aes(x = reorder(variables, props), y = props)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Variable Importance (Forwards)",
       x = "",
       y = "Inclusion Proportion") +
  theme_classic() +
  geom_text(aes(label=round(props, 4)), hjust = 1.1, col = "white")
```



For the defensemen:

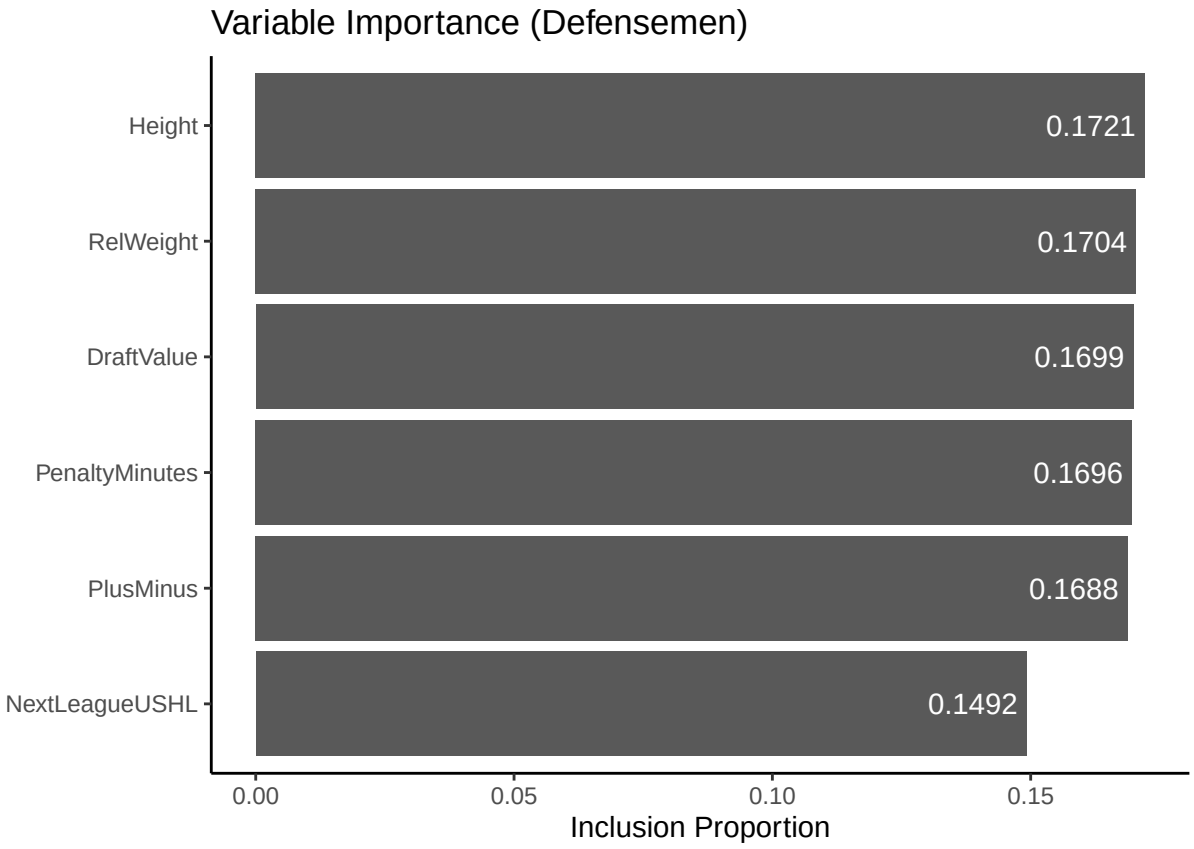
```
USHL_bart.dat_defensemen <- USHL.bart_dat %>%
  filter(GeneralPosition == "Defense")
X_defensemen <- USHL_bart.dat_defensemen %>%
  select(NextLeagueUSHL, PenaltyMinutes, PlusMinus,
         Height_cm, RelWeight, DraftValue) %>%
  as.data.frame()
Y_defensemen <- USHL_bart.dat_defensemen$nhl.games.per.season
USHL_defensemen.bart <- bartMachine(X_defensemen[,names(X_defensemen)!="Player"], Y_defensemen, k = 25,
                                    verbose = FALSE, serialize = TRUE, use_missing_data = TRUE)
```

```
## serializing in order to be saved for future R sessions...done
```

```
USHL_defensemen.output <- investigate_var_importance(USHL_defensemen.bart, num_replicates_for_avg = 20,
```

```
## ..................
```

```
USHL_defensemen.avg_var_props <- USHL_defensemen.output$avg_var_props
USHL_defensemen.props_df <- data.frame(variables = names(USHL_defensemen.avg_var_props),
                                       props = USHL_defensemen.avg_var_props)
USHL_defensemen.props_df$variables[which(USHL_defensemen.props_df$variables == "Height_cm")] = "Height"
ggplot(data = USHL_defensemen.props_df, aes(x = reorder(variables, props), y = props)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Variable Importance (Defensemen)",
       x = "",
       y = "Inclusion Proportion") +
  theme_classic() +
  geom_text(aes(label=round(props, 4)), hjust = 1.1, col = "white")
```



Across the two positions, we notice that for both, the next league had (considerably) lesser inclusion proportion compared to the other variables.
The main difference between the positions is that for the defensemen, the draft value did not seem to be as much included in predicting for the number of games played in the NHL per season - height and weight, as well as the penalty minutes, actually showed higher inclusion proportion.

alluvial

Minyue Fan

4/20/2021

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggplot2)
library(alluvial)
library(ggalluvial)
library(dplyr)
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(date)
```

Mutate Data into Desired Format

```r
bio <- read.csv("complete_bio.csv") %>%
  mutate(BirthYear = as.integer(substring(DateofBirth,
                                          nchar(DateofBirth)-3,
                                          nchar(DateofBirth))))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```r
stat <- read.csv("complete_stat.csv") %>%
  mutate(Season = as.integer(substr(Season, 0,4))) %>%
  fill(Season)




options(digits = 3)
combined.dat <- stat %>% left_join(bio, by = "Player") %>%
  mutate(Age = as.numeric((as.Date(paste(Season, "-09-15",
                                         sep = ""))-mdy(DateofBirth))/365.25),
         DraftYear = as.integer(Age - 18 + 1),
         Performance = as.integer(TotalPoints)/as.integer(Games)) %>%
  #select(Player, Age, League, Performance) %>%
  mutate(Performance = replace_na(Performance, 0))
```

```
## Warning: 206 failed to parse.

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

Prepare Data in Alluvial-Friendly Format

```r
dat.long <- combined.dat %>%
  filter(DraftYear  >=0  & DraftYear <= 6)  %>%
  select(Player, League, DraftYear) %>%
  mutate(League = ifelse(League %in% c("NHL",  "AHL", "NCAA", "USHL","Other"),
                         League,
                         "Other")) %>%
  as.data.frame()

dat.long$League <- factor(dat.long$League, levels = c("NHL",  "AHL", "NCAA", "USHL","Other"))

dat.wide <- dat.long %>%
  reshape(v.names = "League", timevar = "DraftYear", idvar = "Player", direction = "wide", sep = "") %>%
  as.data.frame()
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=0: first taken
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=1: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=2: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=3: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=4: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=5: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for DraftYear=6: first taken
```

```
dat <- dat.wide %>% group_by(League0, League1, League2, League3, League4, League5, League6) %>%
  summarise(freq = n()) %>%
  ungroup() %>%
  drop_na()
```

```
## `summarise()` has grouped output by 'League0', 'League1', 'League2', 'League3', 'League4', 'League5'
```

```
dat.long <- combined.dat %>%
  filter(DraftYear >=0  & DraftYear <= 6 & League %in% c("NHL",  "AHL", "NCAA", "USHL"))  %>%
  select(Player, League, DraftYear) %>%
  as.data.frame()

dat.long$League <- factor(dat.long$League, levels = c("NHL",  "AHL", "NCAA", "USHL"))
dat.long$DraftYear <- as.factor(dat.long$DraftYear)

dat.long <- dat.long %>% group_by(DraftYear) %>%
  summarise(freq = n()) %>%
  ungroup() %>%
  drop_na() %>%
  right_join(dat.long, by = "DraftYear") %>%
  mutate(LeagueGroup = League) %>%
  drop_na() %>%
  arrange(Player) %>%
  as.data.frame()
dat.long <- dat.long[!duplicated(dat.long[,c('DraftYear', 'Player')]),]

players.starts.with.ushl <- dat.long %>%
  filter(DraftYear == 0 & League == "USHL") %>%
  select(Player) %>%
  as.list()
```

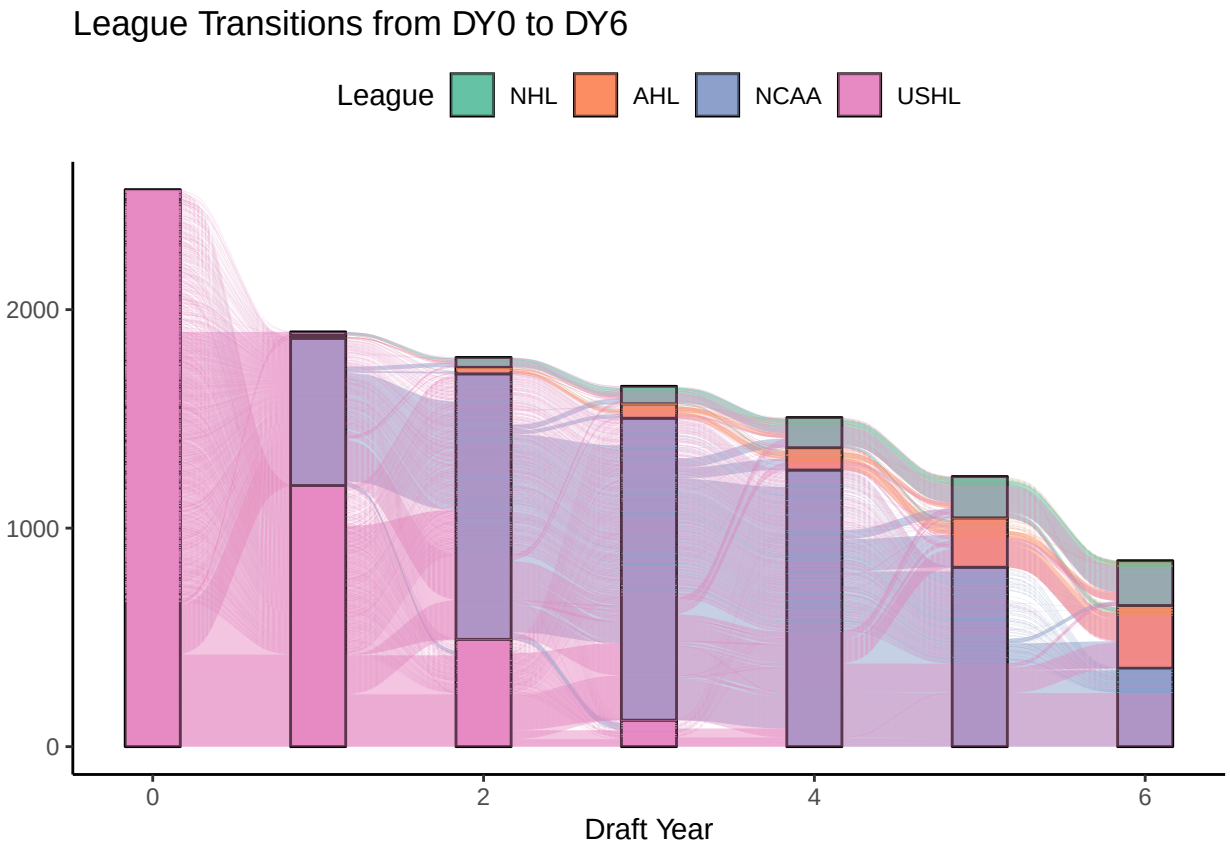
Transitions from DY0 to DY6

```
dat <- read.csv("alluvial.csv")
dat$League <- factor(dat$League, levels = c("NHL",  "AHL", "NCAA", "USHL"))
ggplot(dat,
       aes(x = DraftYear, stratum = League, alluvium = Player,
```

```
          fill = League, label = League)) +
  scale_fill_brewer(type = "qual", palette = "Set2") +
  geom_flow() +
  geom_stratum() +
  geom_alluvium() +
  #theme(legend.position = c(.2,.85), legend.direction = "horizontal") +
  ggtitle("League Transitions from DY0 to DY6") +
  xlab("Draft Year") +
  theme_classic() +
  theme(legend.position='top')
```

```
## Warning: The `.dots` argument of `group_by()` is deprecated as of dplyr 1.0.0.
```

```
## Warning in f(...): Some differentiation aesthetics vary within alluvia, and will be diffused by thei
## Consider using `geom_flow()` instead.
```



Transitions: A Full Picture

```
dat.long %>%
  mutate(DraftYear = as.integer(DraftYear)) %>%
  filter(0 <= DraftYear & DraftYear <= 6) %>%
  filter(Player %in% players.starts.with.ushl$Player) %>%
  mutate(League = factor(League,
                         levels=rev(c('None', 'USHL', 'NCAA', 'AHL', 'NHL')))) %>%
  complete(Player, DraftYear=0:6, fill=list(League='None')) %>%
```

```
ggplot(
       aes(x = DraftYear, stratum = League, alluvium = Player,
           fill = League, label = League)) +
  scale_fill_brewer(type = "qual", palette = "Set2") +
  geom_flow(stat = "alluvium", lode.guidance = "frontback") +
  geom_stratum() +
  geom_alluvium(aes(fill = League)) +
  ggtitle("League Transitions from DY0 to DY6") +
  xlab("Draft Year") +
  theme_classic() +
  theme(legend.position='top')
```

```
## Warning in f(...): Some differentiation aesthetics vary within alluvia, and will be diffused by thei
## Consider using `geom_flow()` instead.
```