Econ 1042 PS2

Owen Asnis

2023-02-13

Data Wrangling

```r
players <- read_delim("nhlps2.csv", delim = "\t",
    escape_double = FALSE, trim_ws = TRUE)

standings <- read_csv("standings_2018_2019.csv",
                      show_col_types = FALSE) %>%
  select(-c(...11, ...12, ...13, ...14, ...15)) %>%
  mutate(PTS_PG = PTS / GP)
```

Question 1

In baseball, expected home runs is a better representation than actual home runs because the expected statistics account for other variables. For example, one batter could be making hard contact but only have a few home runs because they play in Pittsburgh (ranked by Bleacher Report as the worst ballpark for hitters), whereas another batter could be making softer contact but have more home runs because they play in Colorado (considered to be the best ballpark for hitters). Even though the player from Pittsburgh has less home runs, we would still consider them to be a better hitter, as they would have more home runs than the player from Colorado if they played in the same conditions. In hockey, the total shots by each team while a player is on the ice might be a better indicator of a player's value than goals and assists because it similarly accounts for other variables. For example, one player could be generating many shots for and allowing few shots against but have scored no goals because their team was playing the best goalie in the league, whereas another player could be generating few shots for and allowing many shots against and scored a goal because their team was playing the worst goalie in the league. Again, the player generating many shots for and few shots against would be considered the better player because they would be more productive than the player generating few shots for and many shots against if they were playing in the same conditions. In basketball, thinking about shots doesn't work in the same way, because after one team scores, the other team gets the ball, leading to very similar shot totals. In hockey, if one team is dominating the other, there will be a large difference in shot totals.

Question 2

Corsi is the net difference in shots taken by your team and the other team, as defined by the problem set. Therefore, this variable is trying to measure how good a player is at generating shots for their team and limiting shots for the team they're playing. Like was said in the response to Question 1, Corsi is useful compared to just goals scored at both the team and the player level, because it accounts for other variables, like talent of the goalie, and shows how good a player is at producing offense and providing defense beyond just simple goal statistics.

Question 3

We care about a player's relative Corsi versus others on your team because a single player in hockey can only impact the game so much. There are 12 forwards and 6 defenders on a hockey team, and usually 5 players on the ice at any given time. Therefore, a player could have a negative Corsi when they are playing well if their team is struggling and another player could have a positive Corsi when they are struggling if their team is playing well. Relative Corsi accounts for the skill of the team a player is on and is therefore similar to the idea of a team fixed effect. For example, if you put Connor McDavid (the best hockey player in the NHL) on the Columbus Blue Jackets (the worst team in the NHL), his Corsi might be negative because although he's a great player, his team stinks. However, he would likely have a positive relative Corsi, because he would have a higher Corsi rating than many, if not all, of the players on his team. Here, relative Corsi is a better measure of the player's value.

Question 4

```
pdo_model <- lm(pdo ~ lagged_pdo,
                data = players)
summary(pdo_model)
```

```
##
## Call:
## lm(formula = pdo ~ lagged_pdo, data = players)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1792 -1.3352  0.0567  1.3235  6.9871
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 85.15844    2.09607   40.63  < 2e-16 ***
## lagged_pdo   0.14821    0.02093    7.08 1.91e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.04 on 2270 degrees of freedom
##   (734 observations deleted due to missingness)
## Multiple R-squared:  0.02161,    Adjusted R-squared:  0.02117
## F-statistic: 50.13 on 1 and 2270 DF,  p-value: 1.913e-12
```

According to this model, hockey relative shooting percentage (PDO) isn't very serially correlated, because both the estimate for the lagged PDO and the adjusted R-squared are very small, indicating that lagged PDO and PDO aren't intricately related.

```
pdo_variable <- tibble(lagged_pdo = 103.38)

pdo_prediction <- round(predict(pdo_model, newdata = pdo_variable),
                        digits = 2)

sprintf("PDO forecast in year t + 1: %s",
        pdo_prediction)
```

```
## [1] "PDO forecast in year t + 1: 100.48"
```

```
total <- players %>%
  nrow()

under <- players %>%
  filter(pdo < pdo_prediction) %>%
  nrow()

percentile <- round(((under/total) * 100), digits = 0)

sprintf("Corresponding percentile: %sst", percentile)
```

```
## [1] "Corresponding percentile: 59st"
```

Question 5

```
goals_model <- lm(goals ~ lagged_goals,
                  data = players)
summary(goals_model)
```

```
##
## Call:
## lm(formula = goals ~ lagged_goals, data = players)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2681  -3.6147  -0.7269   2.9414  24.2731
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.61964    0.23519   23.89   <2e-16 ***
## lagged_goals  0.44389    0.01866   23.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.153 on 2270 degrees of freedom
##   (734 observations deleted due to missingness)
## Multiple R-squared:  0.1996, Adjusted R-squared:  0.1993 
## F-statistic: 566.2 on 1 and 2270 DF,  p-value: < 2.2e-16
```

```
cfrel_model <- lm(cfrel_percent ~ lagged_cfrel_percent,
                    data = players)
summary(cfrel_model)
```

```
## 
## Call:
## lm(formula = cfrel_percent ~ lagged_cfrel_percent, data = players)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -12.5060  -1.9068   0.0462   2.0030  11.6241 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -0.05382    0.06339  -0.849    0.396    
## lagged_cfrel_percent  0.64322    0.01664  38.664   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 2.999 on 2270 degrees of freedom
##   (734 observations deleted due to missingness)
## Multiple R-squared:  0.3971, Adjusted R-squared:  0.3968 
## F-statistic:  1495 on 1 and 2270 DF,  p-value: < 2.2e-16
```

According to these models, Relative Corsi is more replicable year to year than goals, because both the estimate for the lagged variable and the adjusted R-squared for Relative Corsi are significantly larger. This shows that lagged Relative Corsi and Relative Corsi are significantly more related than lagged goals and goals. This matches expectations: The goal statistic can be a bit random, as some players could score a ton of goals by playing against worse goalies and with better teammates and other players could score only a few goals by playing against better goalies and with worse teammates. Year to year, variables like opposing goalie skill, teammates and bounces could work together to benefit or harm a player’s goal total. On the other hand, Relative Corsi eliminates many of these variables and is a better indication of a player’s true value and therefore, it’s expected that this variable would be more replicable year to year.

Question 6

```
players_nona <- players %>%
  drop_na()

spec1 <- lm(goals ~ lagged_pdo + lagged_goals + lagged_cfrel_percent +
              lagged_shots + lagged_assists + lagged_toi,
     data = players_nona)
summary(spec1)
```

```
##
## Call:
## lm(formula = goals ~ lagged_pdo + lagged_goals + lagged_cfrel_percent +
##     lagged_shots + lagged_assists + lagged_toi, data = players_nona)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3126  -3.4756  -0.5589   2.7547  24.6889
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          11.5392717  6.3502195   1.817   0.0693 .
## lagged_pdo           -0.0391872  0.0633679  -0.618   0.5364
## lagged_goals          0.3255200  0.0295790  11.005  < 2e-16 ***
## lagged_cfrel_percent  0.1616779  0.0323943   4.991 6.48e-07 ***
## lagged_shots          0.0334316  0.0044153   7.572 5.37e-14 ***
## lagged_assists        0.1267181  0.0231186   5.481 4.70e-08 ***
## lagged_toi           -0.0066624  0.0008838  -7.539 6.88e-14 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.982 on 2216 degrees of freedom
## Multiple R-squared:  0.2539, Adjusted R-squared:  0.2519
## F-statistic: 125.7 on 6 and 2216 DF,  p-value: < 2.2e-16
```

```r
spec2 <- lm(goals ~ lagged_goals + lagged_cfrel_percent + lagged_assists,
     data = players_nona)
summary(spec2)
```

```
##
## Call:
## lm(formula = goals ~ lagged_goals + lagged_cfrel_percent + lagged_assists,
##     data = players_nona)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0334  -3.4812  -0.6464   2.9483  24.1434
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.82198    0.28416  20.489  < 2e-16 ***
## lagged_goals          0.34390    0.02294  14.992  < 2e-16 ***
## lagged_cfrel_percent  0.23808    0.03165   7.521 7.82e-14 ***
## lagged_assists        0.05181    0.01761   2.942   0.0033 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.066 on 2219 degrees of freedom
## Multiple R-squared:  0.2275, Adjusted R-squared:  0.2265
## F-statistic: 217.9 on 3 and 2219 DF,  p-value: < 2.2e-16
```

The first specification included every lagged variable, whereas the second specification was simplified and only included 3 lagged variables. The preferred specification is the first one, which included every lagged variable, because it was a better fitted model according to adjusted R-squared.

```
players_nona$spec1_prediction <- predict(spec1)

players_nona %>%
  arrange(desc(spec1_prediction)) %>%
  select(name, season_start, spec1_prediction) %>%
  head(1)
```

```
## # A tibble: 1 x 3
##   name          season_start spec1_prediction
##   <chr>                <dbl>            <dbl>
## 1 Alex.Ovechkin         2009             24.7
```

```
players_nona$spec2_prediction <- predict(spec2)

players_nona %>%
  arrange(desc(spec2_prediction)) %>%
  select(name, season_start, spec2_prediction) %>%
  head(1)
```

```
## # A tibble: 1 x 3
##   name          season_start spec2_prediction
##   <chr>                <dbl>            <dbl>
## 1 Sidney.Crosby         2010             22.1
```

For the first specification, the most goals was forecasted for Alex Ovechkin in 2009, and for the second specification, the most goals was forecasted for Sidney Crosby in 2010.

Question 7

a)

```
PY_2019 <- standings %>%
  filter(Year == 2019.0) %>%
  mutate(PY_C2 = ((GF / GA)^2) / (((GF / GA)^2) + 1),
         PY_C5 = ((GF / GA)^5) / (((GF / GA)^5) + 1),
         W_PC = W / GP)

PY_2019 %>%
  select(PY_C2, PY_C5, W_PC)
```

```
## # A tibble: 31 x 3
##    PY_C2 PY_C5  W_PC
##    <dbl> <dbl> <dbl>
##  1 0.630 0.791  0.629
##  2 0.612 0.758  0.614
##  3 0.524 0.559  0.514
##  4 0.507 0.516  0.507
##  5 0.479 0.448  0.437
##  6 0.447 0.369  0.435
##  7 0.382 0.231  0.352
##  8 0.228 0.0451 0.239
##  9 0.555 0.634  0.594
## 10 0.584 0.699  0.594
## # ... with 21 more rows
```

2 and 5 were used for coefficients, and when the pythagorean win percentage is compared to the actual win percentage, it's clear that 2 fit the actual wins over the 2019 season best.

b)

```
PY_2018.5 <- standings %>%
  filter(Year == 2018.5) %>%
  mutate(PY_C2 = ((GF / GA)^2) / (((GF / GA)^2) + 1),
         PY_C5 = ((GF / GA)^5) / (((GF / GA)^5) + 1),
         W_PC = W / GP)

PY_C2_2018.5 <- lm(PTS_PG ~ PY_C2,
                   data = PY_2018.5)
RMSE_C2 <- round(sqrt(mean(PY_C2_2018.5$residuals^2)),
                 digits = 3)

sprintf("Root Mean Squared Error with a coefficient of 2: %s",
        RMSE_C2)
```

```
## [1] "Root Mean Squared Error with a coefficient of 2: 0.055"
```

```
PY_C5_2018.5 <- lm(PTS_PG ~ PY_C5,
                   data = PY_2018.5)
RMSE_C5 <- round(sqrt(mean(PY_C5_2018.5$residuals^2)),
                 digits = 3)

sprintf("Root Mean Squared Error with a coefficient of 5: %s",
        RMSE_C5)
```

```
## [1] "Root Mean Squared Error with a coefficient of 5: 0.059"
```

Using 2 as a coefficient rather than 5 minimizes the Root Mean Squared Error, which is in line with what was found in Question 7a. Pythagorean wins is an improvement over points per game in the first half, because pythagorean wins is better at determining how a team is truly playing. A team could have high points per game despite a negative goal differential and inversely, a team could have low points per game despite a positive goal differential. It was proven in baseball that run differential is a better predictor for future team performance than past winning percentage. Therefore, it would be expected that in hockey goal differential (which determines pythagorean wins) would be a better predictor for future team performance than past points percentage.

Question 8

I worked with Ty Thabit on this problem set.