Garrity\_Final\_Project

players <- read\_csv("player\_info.csv")

## Parsed with column specification:  
## cols(  
## player\_id = col\_double(),  
## firstName = col\_character(),  
## lastName = col\_character(),  
## nationality = col\_character(),  
## birthCity = col\_character(),  
## primaryPosition = col\_character(),  
## birthDate = col\_date(format = ""),  
## link = col\_character()  
## )

player\_game\_plays <- read\_csv("game\_plays\_players.csv")

## Parsed with column specification:  
## cols(  
## play\_id = col\_character(),  
## game\_id = col\_double(),  
## play\_num = col\_double(),  
## player\_id = col\_double(),  
## playerType = col\_character()  
## )

player\_game\_stats <- read\_csv("game\_skater\_stats.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double()  
## )

## See spec(...) for full column specifications.

plays <- read\_csv("game\_plays.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## play\_id = col\_character(),  
## event = col\_character(),  
## secondaryType = col\_character(),  
## periodType = col\_character(),  
## dateTime = col\_datetime(format = ""),  
## description = col\_character(),  
## rink\_side = col\_character()  
## )  
## See spec(...) for full column specifications.

games <- read\_csv("game.csv")

## Parsed with column specification:  
## cols(  
## game\_id = col\_double(),  
## season = col\_double(),  
## type = col\_character(),  
## date\_time = col\_date(format = ""),  
## date\_time\_GMT = col\_datetime(format = ""),  
## away\_team\_id = col\_double(),  
## home\_team\_id = col\_double(),  
## away\_goals = col\_double(),  
## home\_goals = col\_double(),  
## outcome = col\_character(),  
## home\_rink\_side\_start = col\_character(),  
## venue = col\_character(),  
## venue\_link = col\_character(),  
## venue\_time\_zone\_id = col\_character(),  
## venue\_time\_zone\_offset = col\_double(),  
## venue\_time\_zone\_tz = col\_character()  
## )

teams <- read\_csv("team\_info.csv")

## Parsed with column specification:  
## cols(  
## team\_id = col\_double(),  
## franchiseId = col\_double(),  
## shortName = col\_character(),  
## teamName = col\_character(),  
## abbreviation = col\_character(),  
## link = col\_character()  
## )

# Calculating Player Stats by Season

First combining the datasets for players and their stats by game, setting it into a new dataframe

player\_name\_stat <- player\_game\_stats %>%   
 left\_join(players) %>%   
 left\_join(games) %>%   
 left\_join(teams, by = c("team\_id" = "team\_id")) %>%   
 filter(type=="R") %>%   
 subset(select = -c(birthCity, link.x, link.y, date\_time\_GMT, home\_rink\_side\_start, venue, venue\_link, venue\_time\_zone\_id, venue\_time\_zone\_offset, venue\_time\_zone\_tz)) %>%   
 mutate(fullName = paste(firstName, lastName, sep = " "), timeOnIce = timeOnIce/60, season = str\_sub(season, -4))

## Joining, by = "player\_id"

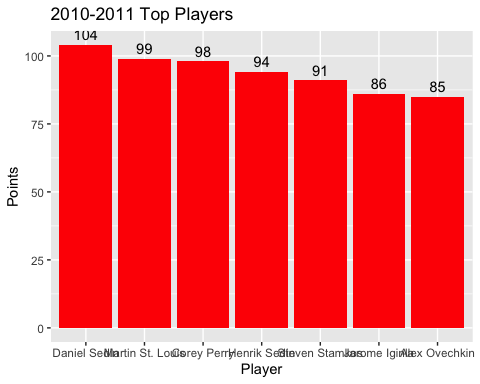
## Joining, by = "game\_id"

# Points by year

2010-2011

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 arrange(desc(tp)) %>%   
 filter(season==2011) %>%   
 slice(1:7) %>%   
 ggplot() +  
 geom\_col(aes(reorder(fullName,-tp), tp), fill = "red") +  
 geom\_text(aes(reorder(fullName,-tp),tp,label=tp),vjust=-.5) +  
 labs(x="Player", y="Points", title = "2010-2011 Top Players")

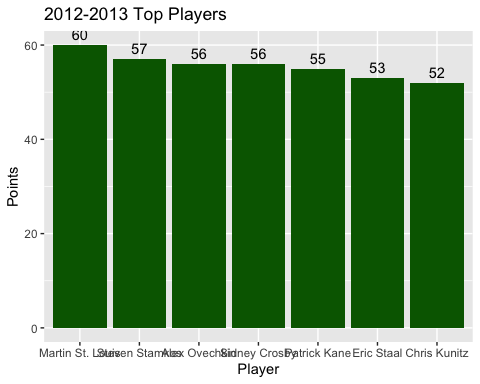
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



2012-2013 - Lockout Season

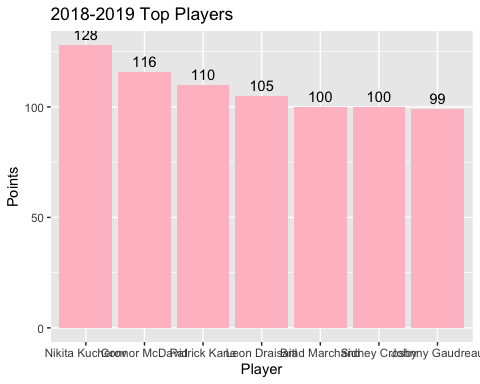
player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 arrange(desc(tp)) %>%   
 filter(season==2013) %>%   
 slice(1:7) %>%   
 ggplot() +  
 geom\_col(aes(reorder(fullName,-tp), tp), fill = "darkgreen") +  
 geom\_text(aes(reorder(fullName,-tp),tp,label=tp),vjust=-.5) +  
 labs(x="Player", y="Points", title = "2012-2013 Top Players")

## `summarise()` regrouping output by 'season' (override with `.groups` argument)

 2018-2019

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 arrange(desc(tp)) %>%   
 filter(season==2019) %>%   
 slice(1:7) %>%   
 ggplot() +  
 geom\_col(aes(reorder(fullName,-tp), tp), fill = "pink") +  
 geom\_text(aes(reorder(fullName,-tp),tp,label=tp),vjust=-.5) +  
 labs(x="Player", y="Points", title = "2018-2019 Top Players")

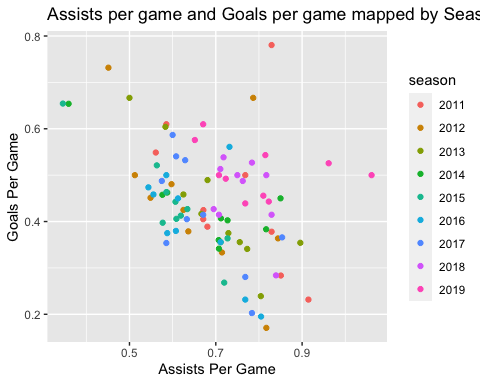
## `summarise()` regrouping output by 'season' (override with `.groups` argument)

 # Looking at Yearly Trends for Different Statistics

Creating Per Game statistics and mapping on scatter plot 10 best ppg by season

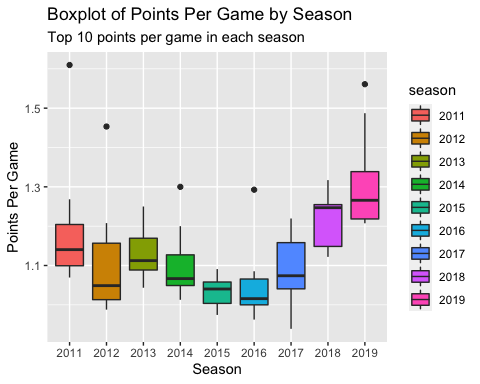
player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 mutate(gpg = tg/gp, apg = ta/gp, ppg = tp/gp) %>%   
 arrange(desc(ppg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(ppg)) %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_point(aes(x=apg, y=gpg, color = season)) +  
 labs(title = "Assists per game and Goals per game mapped by Season", x = "Assists Per Game", y = "Goals Per Game")

## `summarise()` regrouping output by 'season' (override with `.groups` argument)

 Looking at top Points Per Game by year

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 mutate(gpg = tg/gp, apg = ta/gp, ppg = tp/gp) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(ppg)) %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=ppg, group=season, fill = season)) +  
 labs(title = "Boxplot of Points Per Game by Season", subtitle = "Top 10 points per game in each season", x = "Season", y = "Points Per Game")

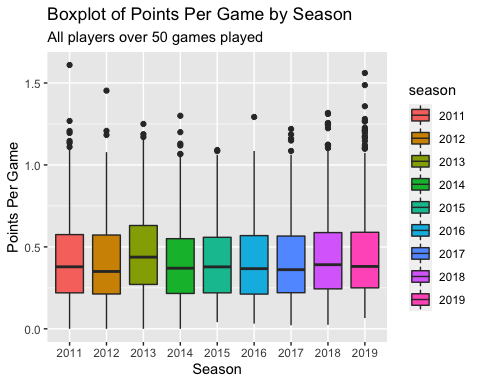
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Total Points Per Game by Season + 50 gp

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 mutate(gpg = tg/gp, apg = ta/gp, ppg = tp/gp) %>%   
 arrange(desc(ppg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(ppg)) %>%  
 ggplot() +  
 geom\_boxplot(aes(x=season, y=ppg, group=season, fill = season))+  
 labs(title = "Boxplot of Points Per Game by Season", subtitle = "All players over 50 games played", x = "Season", y = "Points Per Game")

## `summarise()` regrouping output by 'season' (override with `.groups` argument)

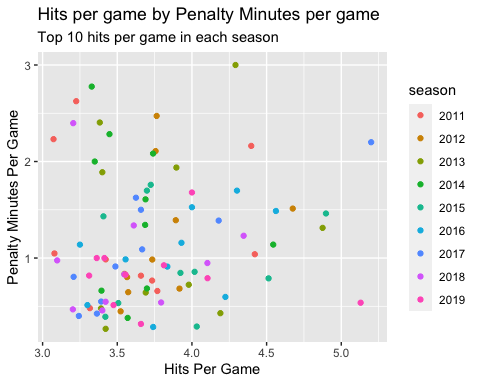


# Hits and Penalty Minutes by Year

Hit and Penalty Minutes on Scatterplot

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tpm = sum(penaltyMinutes)) %>%   
 mutate(hpg = th/gp, pmpg = tpm/gp) %>%   
 arrange(desc(hpg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(hpg)) %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_point(aes(x=hpg, y=pmpg, color = season))+  
 labs(title = "Hits per game by Penalty Minutes per game", subtitle = "Top 10 hits per game in each season", x = "Hits Per Game", y = "Penalty Minutes Per Game")

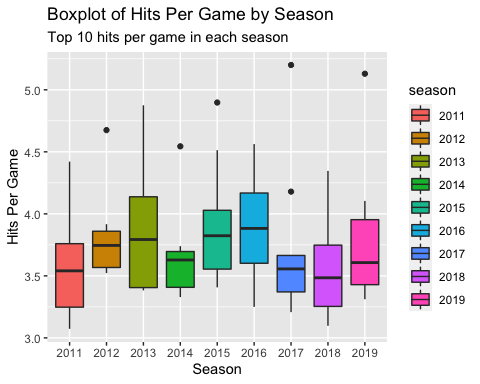
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Top 10 players hits per game boxplot

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tpm = sum(penaltyMinutes)) %>%   
 mutate(hpg = th/gp, pmpg = tpm/gp) %>%   
 arrange(desc(hpg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(hpg)) %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=hpg, group=season, fill = season)) +  
 labs(title = "Boxplot of Hits Per Game by Season", subtitle = "Top 10 hits per game in each season", x = "Season", y = "Hits Per Game")

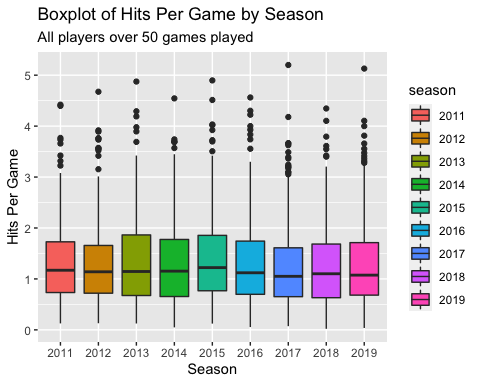
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Hits per game by season

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tpm = sum(penaltyMinutes)) %>%   
 mutate(hpg = th/gp, pmpg = tpm/gp) %>%   
 arrange(desc(hpg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(hpg)) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=hpg, group=season, fill = season)) +  
 labs(title = "Boxplot of Hits Per Game by Season", subtitle = "All players over 50 games played", x = "Season", y = "Hits Per Game")

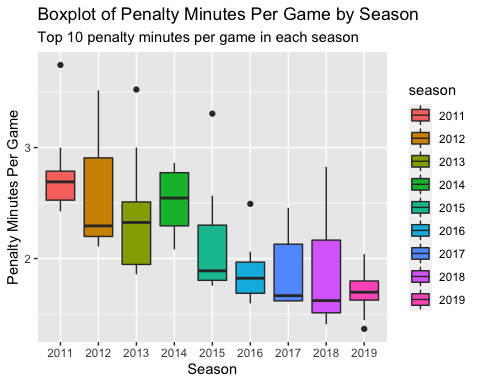
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Top 10 players penalty minutes per game boxplot

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tpm = sum(penaltyMinutes)) %>%   
 mutate(hpg = th/gp, pmpg = tpm/gp) %>%   
 arrange(desc(hpg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(pmpg)) %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=pmpg, group=season, fill = season))+  
 labs(title = "Boxplot of Penalty Minutes Per Game by Season", subtitle = "Top 10 penalty minutes per game in each season", x = "Season", y = "Penalty Minutes Per Game")

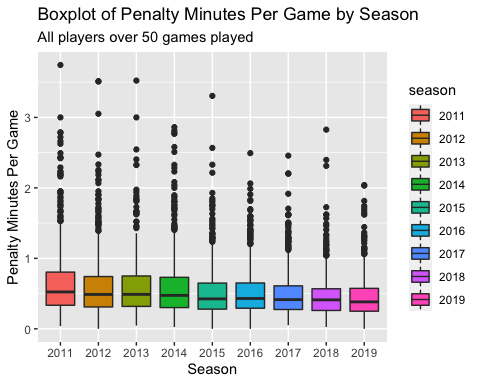
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Penalty minutes per game by sesason

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tpm = sum(penaltyMinutes)) %>%   
 mutate(hpg = th/gp, pmpg = tpm/gp) %>%   
 arrange(desc(hpg)) %>%   
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(pmpg)) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=pmpg, group=season, fill = season)) +  
 labs(title = "Boxplot of Penalty Minutes Per Game by Season", subtitle = "All players over 50 games played", x = "Season", y = "Penalty Minutes Per Game")

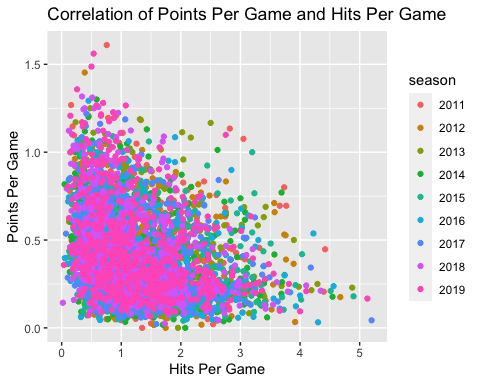
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Scatterplot of points per game adn hits per game

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(gp = n(), th = sum(hits), tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 mutate(hpg = th/gp, ppg = tp/gp) %>%  
 filter(gp>=40) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 ggplot()+  
 geom\_point(aes(hpg,ppg, color = season)) +  
 labs(title = "Correlation of Points Per Game and Hits Per Game", x="Hits Per Game", y = "Points Per Game")

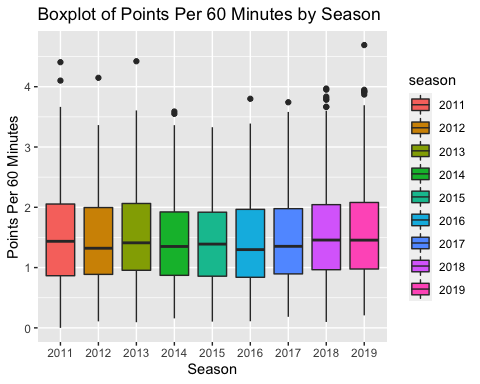
## `summarise()` regrouping output by 'season' (override with `.groups` argument)



Points per 60 minutes played

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(totalTime = sum(timeOnIce), tg = sum(goals), ta = sum(assists), tp = tg + ta) %>%   
 mutate(pp60 = (tp/totalTime)\*60) %>%   
 filter(totalTime >= 500) %>%   
 ungroup() %>%   
 group\_by(season) %>%   
 arrange(desc(pp60)) %>%   
 ggplot() +  
 geom\_boxplot(aes(x=season, y=pp60, group=season, fill = season)) +  
 labs(title = "Boxplot of Points Per 60 Minutes by Season", x="Season", y="Points Per 60 Minutes")

## `summarise()` regrouping output by 'season' (override with `.groups` argument)



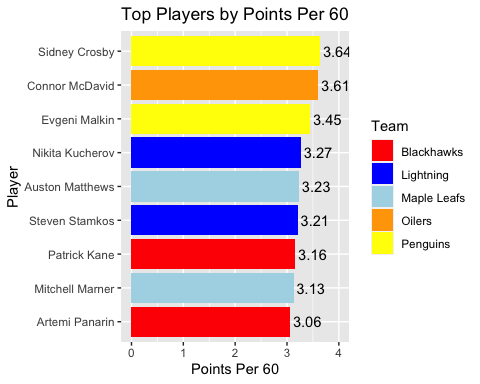
# Looking at top players by Statistics Per 60 Minutes Played

Best Player and Team - Evaluating Teammate Importance

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(totalTime = sum(timeOnIce), tg = sum(goals), ta = sum(assists), tp = tg + ta, teamName) %>%   
 unique() %>%   
 mutate(pp60 = (tp/totalTime)\*60) %>%   
 filter(totalTime >= 500) %>%   
 ungroup() %>%   
 group\_by(fullName) %>%   
 summarize(numSeason = n(), avgpp60 = mean(pp60), teamName) %>%   
 unique() %>%   
 arrange(desc(avgpp60)) %>%   
 ungroup() %>%   
 slice(1:9) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,avgpp60), y=avgpp60, fill = teamName)) +  
 ylim(0,4) +  
 geom\_text(aes(reorder(fullName,avgpp60),avgpp60,label=round(avgpp60,2)),hjust=-.1) +  
 scale\_fill\_manual(values=c("Blackhawks"="red", "Lightning"="blue", "Maple Leafs"="lightblue", "Oilers"="orange", "Penguins"="yellow")) +  
 labs(title = "Top Players by Points Per 60", x = "Player", y = "Points Per 60", fill = "Team") +  
 coord\_flip()

## `summarise()` regrouping output by 'season', 'fullName' (override with `.groups` argument)

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

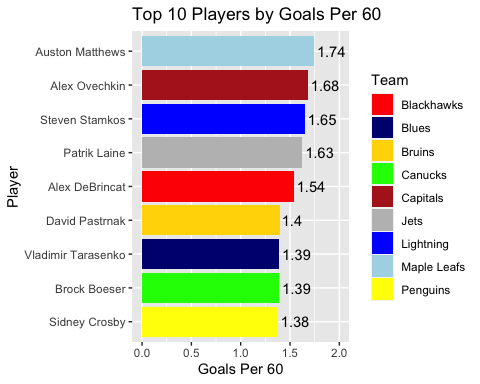


Goals per 60

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(totalTime = sum(timeOnIce), tg = sum(goals), teamName) %>%   
 unique() %>%   
 mutate(gp60 = (tg/totalTime)\*60) %>%   
 filter(totalTime >= 500) %>%   
 ungroup() %>%   
 group\_by(fullName) %>%   
 summarize(numSeason = n(), avggp60 = mean(gp60), teamName) %>%   
 unique() %>%   
 arrange(desc(avggp60)) %>%   
 ungroup() %>%   
 slice(1:9) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,avggp60), y=avggp60, fill = teamName)) +  
 ylim(0,2) +  
 geom\_text(aes(reorder(fullName,avggp60),avggp60,label=round(avggp60,2)),hjust=-.1) +  
 scale\_fill\_manual(values=c("Blackhawks"="red", "Blues"="navy", "Bruins"="gold", "Canucks"="green", "Capitals"="firebrick", "Jets"="grey","Lightning"="blue","Maple Leafs"="lightblue","Penguins"="yellow")) +  
 labs(title = "Top 10 Players by Goals Per 60", x = "Player", y = "Goals Per 60", fill = "Team") +  
 coord\_flip()

## `summarise()` regrouping output by 'season', 'fullName' (override with `.groups` argument)

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

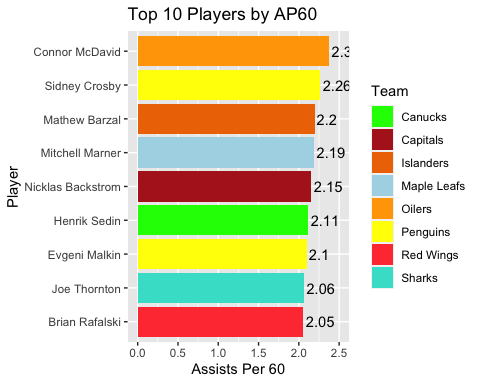


Assists per 60

player\_name\_stat %>%   
 group\_by(season, fullName) %>%   
 summarize(totalTime = sum(timeOnIce), ta = sum(assists), teamName) %>%   
 unique() %>%   
 mutate(ap60 = (ta/totalTime)\*60) %>%   
 filter(totalTime >= 500) %>%   
 ungroup() %>%   
 group\_by(fullName) %>%   
 summarize(numSeason = n(), avgap60 = mean(ap60), teamName) %>%   
 unique() %>%   
 arrange(desc(avgap60)) %>%   
 ungroup() %>%   
 slice(1:9) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,avgap60), y=avgap60, fill = teamName)) +  
 ylim(0,2.5) +  
 geom\_text(aes(reorder(fullName,avgap60),avgap60,label=round(avgap60,2)),hjust=-.1) +  
 scale\_fill\_manual(values=c("Canucks"="green", "Capitals"="firebrick", "Islanders"="darkorange2", "Maple Leafs"="lightblue", "Oilers"="orange", "Penguins"="yellow","Red Wings"="brown1","Sharks"="turquoise")) +  
 labs(title = "Top 10 Players by AP60", x = "Player", y = "Assists Per 60", fill = "Team") +  
 coord\_flip()

## `summarise()` regrouping output by 'season', 'fullName' (override with `.groups` argument)

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

 # Players Salaries and if they are deserved

NHL Salary data for 2018-2019 Year

url1 <- "https://www.capfriendly.com/browse/active/2019?contract=standard&hide=team,clauses,age,position,handed,expiry-status,caphit,skater-stats,goalie-stats&limits=gp-20-90"  
df1 <- url1 %>%  
 read\_html() %>%  
 html\_nodes("table") %>%  
 .[[1]] %>%  
 html\_table() %>%   
 as\_tibble()  
df1$PLAYER <- sub("\\d+.\\s", "", df1$PLAYER)  
  
for (i in 2:15) {  
 urltemp <- paste("https://www.capfriendly.com/browse/active/2019?contract=standard&hide=team,clauses,age,position,handed,expiry-status,caphit,skater-stats,goalie-stats&limits=gp-20-90&pg=", i, sep = "")  
 dftemp <-urltemp %>%  
 read\_html() %>%  
 html\_nodes("table") %>%  
 .[[1]] %>%  
 html\_table() %>%   
 as\_tibble()  
 dftemp$PLAYER <- sub("\\d+.\\s", "", dftemp$PLAYER)  
 dftemp <- dftemp %>%   
 mutate(SALARY = as.character(SALARY))  
 df1 <- full\_join(df1, dftemp)  
}

## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")  
## Joining, by = c("PLAYER", "SALARY")

df1$SALARY <- gsub('[$]','',df1$SALARY)  
df1$SALARY <- as.numeric(gsub(',','',df1$SALARY))  
df1 %>%   
 arrange(desc(SALARY))

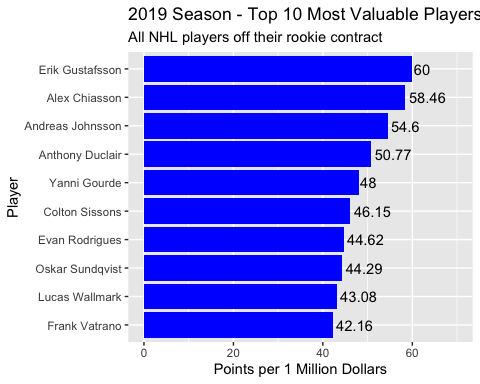
## # A tibble: 571 x 2  
## PLAYER SALARY  
## <chr> <dbl>  
## 1 John Tavares 15900000  
## 2 Connor McDavid 15000000  
## 3 Carey Price 15000000  
## 4 Jamie Benn 13000000  
## 5 Patrick Kane 12000000  
## 6 Jonathan Toews 12000000  
## 7 William Nylander 12000000  
## 8 Anze Kopitar 12000000  
## 9 John Carlson 12000000  
## 10 Jack Eichel 10000000  
## # … with 561 more rows

players\_salaries <- player\_name\_stat %>%   
 inner\_join(df1, by = c("fullName" = "PLAYER"))

Looking at best value by salary and points per game

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82) %>%   
 arrange(desc(pointsperdollar)) %>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 ylim(0,70) +  
 geom\_text(aes(reorder(fullName,pointsperdollar),pointsperdollar\*1000000,label=round(pointsperdollar\*1000000,2)),hjust=-.1) +  
 labs(title = "2019 Season - Top 10 Most Valuable Players by Total Points",subtitle = "All NHL players off their rookie contract", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

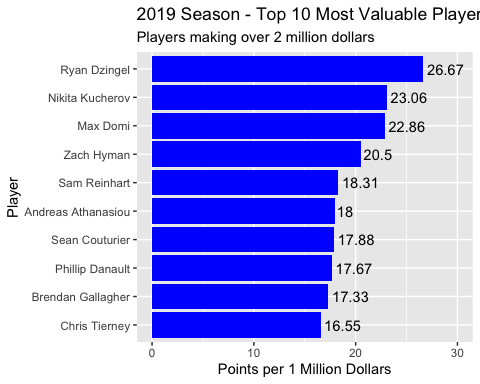
## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



Looking at Players making at least 2 Million

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82 & salary\_millions>=2) %>%   
 arrange(desc(pointsperdollar)) %>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 ylim(0,30) +  
 geom\_text(aes(reorder(fullName,pointsperdollar),pointsperdollar\*1000000,label=round(pointsperdollar\*1000000,2)),hjust=-.1) +  
 labs(title = "2019 Season - Top 10 Most Valuable Players by Total Points",subtitle = "Players making over 2 million dollars", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

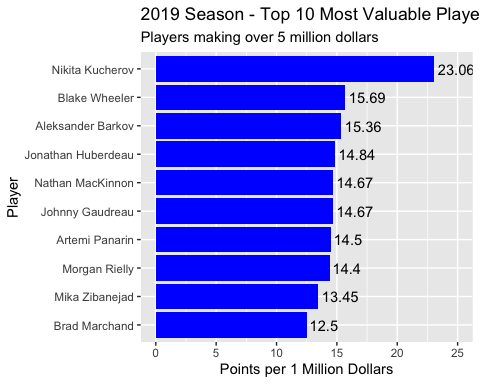
## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



At Least 5 Million

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82 & salary\_millions>=5) %>%   
 arrange(desc(pointsperdollar))%>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 ylim(0,25) +  
 geom\_text(aes(reorder(fullName,pointsperdollar),pointsperdollar\*1000000,label=round(pointsperdollar\*1000000,2)),hjust=-.1) +  
 labs(title = "2019 Season - Top 10 Most Valuable Players by Total Points",subtitle = "Players making over 5 million dollars", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

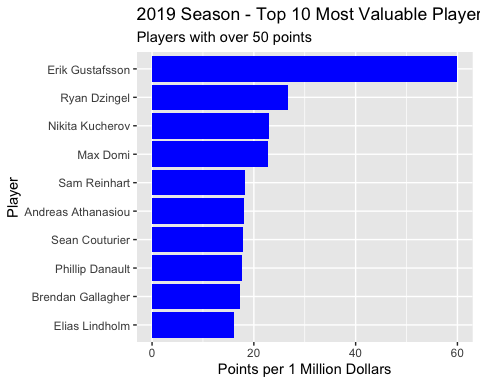
## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



Same as above but with players with over 50 points

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40, tp >= 50) %>%   
 arrange(desc(pointsperdollar))%>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 labs(title = "2019 Season - Top 10 Most Valuable Players by Total Points",subtitle = "Players with over 50 points", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

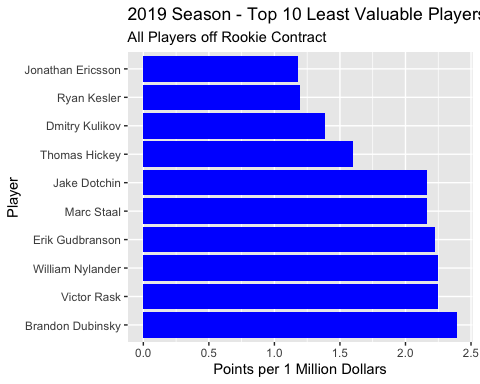
## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



Worst Player based on contract

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40) %>%   
 arrange(pointsperdollar) %>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,-pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 labs(title = "2019 Season - Top 10 Least Valuable Players by Total Points",subtitle = "All Players off Rookie Contract", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

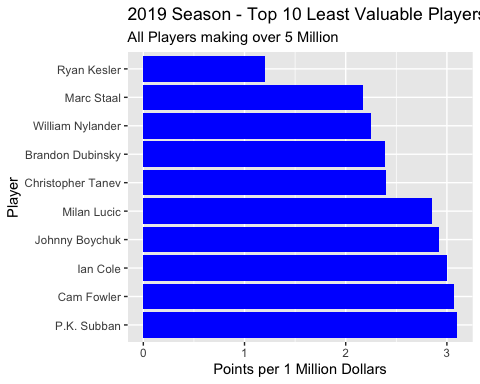
## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



Worst Making over 5 Million

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg,salary\_millions = SALARY/1000000, pointsperdollar = tp/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & salary\_millions >= 5) %>%   
 arrange(pointsperdollar) %>%   
 ungroup() %>%   
 slice(1:10) %>%   
 ggplot() +  
 geom\_col(aes(x=reorder(fullName,-pointsperdollar), y=pointsperdollar\*1000000), fill = "blue") +  
 labs(title = "2019 Season - Top 10 Least Valuable Players by Total Points",subtitle = "All Players making over 5 Million", x = "Player", y = "Points per 1 Million Dollars") +  
 coord\_flip()

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)



Best dollar value by points per game

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, ppg = tp/gp, salary\_millions = SALARY/1000000, pointspergameperdollar = ppg/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40) %>%   
 arrange(desc(pointspergameperdollar))

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 422 x 8  
## # Groups: fullName [418]  
## fullName gp ta tg tp ppg salary\_millions pointspergameperd…  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Alex Chiass… 73 16 22 38 0.521 0.65 0.000000801  
## 2 Erik Gustaf… 79 43 17 60 0.759 1 0.000000759  
## 3 Andreas Joh… 73 23 20 43 0.589 0.788 0.000000748  
## 4 Anthony Duc… 74 14 19 33 0.446 0.65 0.000000686  
## 5 Colton Siss… 75 15 15 30 0.4 0.65 0.000000615  
## 6 Evan Rodrig… 74 20 9 29 0.392 0.65 0.000000603  
## 7 Yanni Gourde 80 26 22 48 0.6 1 0.0000006   
## 8 Oskar Sundq… 74 17 14 31 0.419 0.7 0.000000598  
## 9 Devon Toews 48 13 5 18 0.375 0.65 0.000000577  
## 10 Brendan Lei… 62 16 7 23 0.371 0.65 0.000000571  
## # … with 412 more rows

Same as above but making over 2 Million

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, ppg = tp/gp, salary\_millions = SALARY/1000000, pointspergameperdollar = ppg/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & salary\_millions >= 2) %>%   
 arrange(desc(pointspergameperdollar))

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 290 x 8  
## # Groups: fullName [289]  
## fullName gp ta tg tp ppg salary\_millions pointspergameperd…  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Ryan Dzingel 78 30 26 56 0.718 2.1 0.000000342  
## 2 Zach Hyman 71 20 21 41 0.577 2 0.000000289  
## 3 Nikita Kuch… 82 87 41 128 1.56 5.55 0.000000281  
## 4 Max Domi 82 44 28 72 0.878 3.15 0.000000279  
## 5 Andreas Ath… 76 24 30 54 0.711 3 0.000000237  
## 6 Sean Coutur… 80 43 33 76 0.95 4.25 0.000000224  
## 7 Sam Reinhart 82 43 22 65 0.793 3.55 0.000000223  
## 8 Phillip Dan… 81 41 12 53 0.654 3 0.000000218  
## 9 Anthony Man… 67 23 25 48 0.716 3.3 0.000000217  
## 10 Nate Schmidt 61 21 9 30 0.492 2.3 0.000000214  
## # … with 280 more rows

Best Dollar value by points per 60

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, totalTime = sum(timeOnIce), pp60 = (tp/totalTime)\*60, pointsper60perdollar = pp60/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82) %>%   
 arrange(desc(pointsper60perdollar))

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 415 x 8  
## # Groups: fullName [414]  
## fullName gp ta tg tp totalTime pp60 pointsper60perdollar  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Andreas Johnsson 73 23 20 43 998. 2.59 0.00000328  
## 2 Anthony Duclair 74 14 19 33 940. 2.11 0.00000324  
## 3 Nic Dowd 64 14 8 22 659. 2.00 0.00000308  
## 4 Travis Boyd 53 15 5 20 521. 2.30 0.00000288  
## 5 Alex Chiasson 73 16 22 38 1239. 1.84 0.00000283  
## 6 Oskar Sundqvist 74 17 14 31 1023. 1.82 0.00000260  
## 7 Kenny Agostino 63 18 6 24 814. 1.77 0.00000253  
## 8 Brendan Leipsic 62 16 7 23 855. 1.61 0.00000248  
## 9 Daniel Sprong 63 9 14 23 753. 1.83 0.00000244  
## 10 Dominik Simon 71 20 8 28 946. 1.78 0.00000237  
## # … with 405 more rows

Same thing but total time over 1000 minutes

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, totalTime = sum(timeOnIce), pp60 = (tp/totalTime)\*60, pointsper60perdollar = pp60/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82 & totalTime > 1000) %>%   
 arrange(desc(pointsper60perdollar))

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 289 x 8  
## # Groups: fullName [289]  
## fullName gp ta tg tp totalTime pp60 pointsper60perdollar  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Alex Chiasson 73 16 22 38 1239. 1.84 0.00000283  
## 2 Oskar Sundqvist 74 17 14 31 1023. 1.82 0.00000260  
## 3 Colton Sissons 75 15 15 30 1203. 1.50 0.00000230  
## 4 Evan Rodrigues 74 20 9 29 1171. 1.49 0.00000229  
## 5 Yanni Gourde 80 26 22 48 1267. 2.27 0.00000227  
## 6 Lucas Wallmark 81 18 10 28 1198. 1.40 0.00000216  
## 7 Frank Vatrano 81 15 24 39 1188. 1.97 0.00000213  
## 8 Erik Gustafsson 79 43 17 60 1784. 2.02 0.00000202  
## 9 Brett Connolly 81 24 22 46 1080. 2.56 0.00000170  
## 10 Tyler Bertuzzi 73 26 21 47 1206. 2.34 0.00000167  
## # … with 279 more rows

Same thing but now total points over 50

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, totalTime = sum(timeOnIce), pp60 = (tp/totalTime)\*60, pointsper60perdollar = pp60/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82 & totalTime > 1000 & tp >= 50) %>%   
 arrange(desc(pointsper60perdollar))

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 88 x 8  
## # Groups: fullName [88]  
## fullName gp ta tg tp totalTime pp60 pointsper60perdoll…  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Erik Gustafsson 79 43 17 60 1784. 2.02 0.00000202   
## 2 Ryan Dzingel 78 30 26 56 1312. 2.56 0.00000122   
## 3 Max Domi 82 44 28 72 1425. 3.03 0.000000962  
## 4 Nikita Kucherov 82 87 41 128 1637. 4.69 0.000000845  
## 5 Andreas Athanasi… 76 24 30 54 1290. 2.51 0.000000837  
## 6 Brendan Gallagher 82 19 33 52 1345. 2.32 0.000000773  
## 7 Phillip Danault 81 41 12 53 1440. 2.21 0.000000736  
## 8 Sam Reinhart 82 43 22 65 1555. 2.51 0.000000707  
## 9 Eric Staal 81 30 22 52 1468. 2.13 0.000000607  
## 10 Sean Couturier 80 43 33 76 1770. 2.58 0.000000606  
## # … with 78 more rows

Worst of points per 60 per dollar

players\_salaries %>%   
 filter(season == 2019) %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), ta = sum(assists), tg = sum(goals), tp = ta + tg, totalTime = sum(timeOnIce), pp60 = (tp/totalTime)\*60, pointsper60perdollar = pp60/SALARY) %>%   
 unique() %>%   
 filter(gp >= 40 & gp <= 82 & totalTime > 1000) %>%   
 arrange(pointsper60perdollar)

## `summarise()` regrouping output by 'fullName' (override with `.groups` argument)

## # A tibble: 289 x 8  
## # Groups: fullName [289]  
## fullName gp ta tg tp totalTime pp60 pointsper60perdoll…  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Marc Staal 79 10 3 13 1534. 0.508 0.0000000847  
## 2 Niklas Hjalmarss… 82 10 0 10 1615. 0.371 0.0000000929  
## 3 Erik Gudbranson 76 8 2 10 1386. 0.433 0.0000000962  
## 4 Jonas Brodin 82 14 4 18 1685. 0.641 0.000000111   
## 5 Nikita Zaitsev 81 11 3 14 1658. 0.507 0.000000113   
## 6 Jay Bouwmeester 78 14 3 17 1617. 0.631 0.000000117   
## 7 Brent Seabrook 78 23 5 28 1490. 1.13 0.000000125   
## 8 Johnny Boychuk 74 16 3 19 1393. 0.818 0.000000126   
## 9 Ian Cole 71 13 2 15 1403. 0.642 0.000000128   
## 10 Christopher Tanev 55 10 2 12 1109. 0.649 0.000000130   
## # … with 279 more rows

# NHL Shooting Trends

This create an NHL rink visual to use with ggplot. I did not write this code. Credit to <https://github.com/mtthwastn> Link to code: <https://github.com/mtthwastn/statswithmatt/blob/master/hockey-with-r/gg-rink.R> Link to his article: <https://www.statswithmatt.com/post/hockey-rink-in-r/>

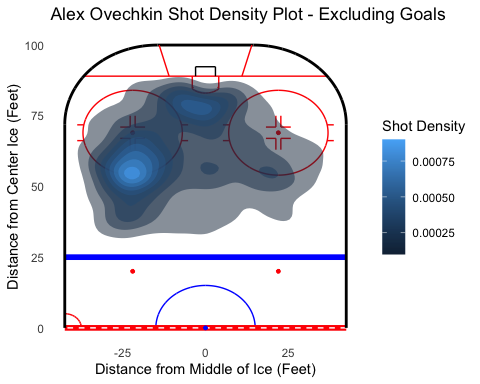
# Looking at the best goal scorer in the NHL's shot charts

Alexander Ovechkin - Best Modern NHL Goal Scorer - Maybe All Time Shoots that don't go in

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Blocked Shot","Missed Shot"), st\_x > 25) %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Excluding Goals", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

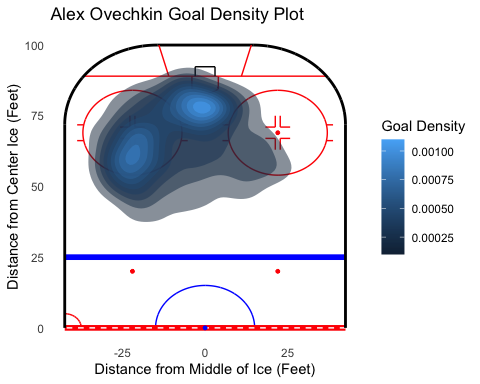


Alexander Ovechkin Goals

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event == "Goal", st\_x > 25) %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Goal Density Plot", fill = "Goal Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

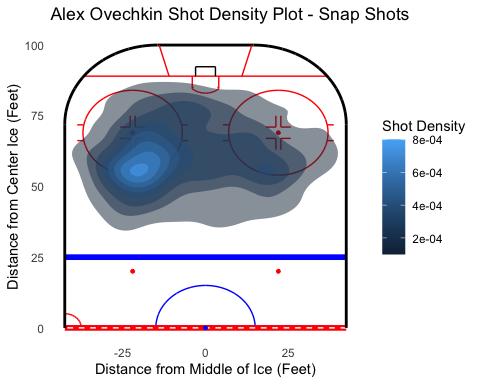


Looking at Shot Type Snap Shot

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter" | playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Goal"), st\_x > 25, secondaryType == "Snap Shot") %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Snap Shots", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

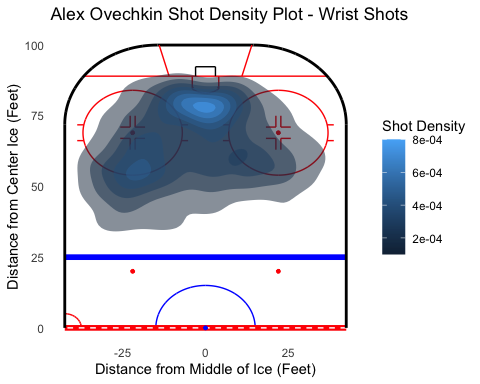


Wrist Shot

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter" | playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Goal"), st\_x > 25, secondaryType == "Wrist Shot") %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Wrist Shots", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

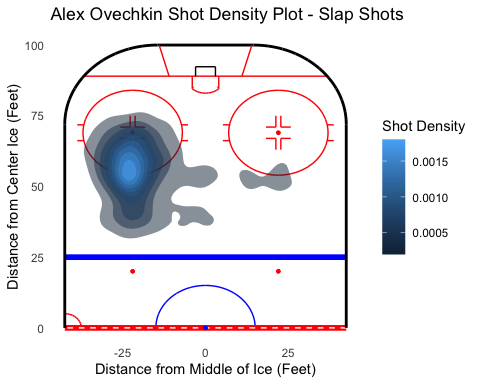


Slap Shot

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter" | playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Goal"), st\_x > 25, secondaryType == "Slap Shot") %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Slap Shots", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

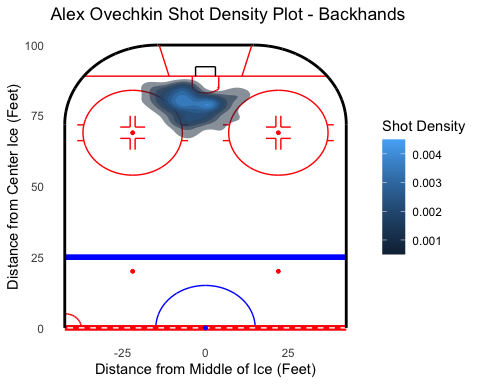


Backhand

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter" | playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Goal"), st\_x > 25, secondaryType == "Backhand") %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Backhands", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")

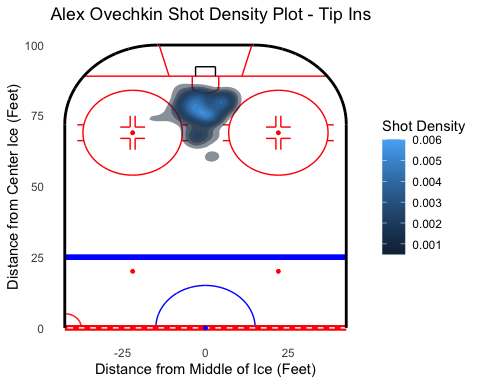


Tip-In

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 filter(playerType == "Shooter" | playerType == "Scorer", fullName == "Alex Ovechkin") %>%   
 inner\_join(plays) %>%   
 filter(event %in% c("Shot","Goal"), st\_x > 25, secondaryType == "Tip-In") %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Alex Ovechkin Shot Density Plot - Tip Ins", fill = "Shot Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip()

## Joining, by = "player\_id"

## Joining, by = c("play\_id", "game\_id", "play\_num")



Top 9 Goals scorers by Goals Per Game and their goal density map

Getting top 9 Players

top9gpg <- player\_name\_stat %>%   
 group\_by(fullName) %>%   
 summarize(gp = n(), tg = sum(goals)) %>%   
 mutate(gpg = tg/gp) %>%   
 filter(gp >= 200) %>%   
 arrange(desc(gpg)) %>%   
 select(fullName, gpg) %>%   
 slice(1:10)

## `summarise()` ungrouping output (override with `.groups` argument)

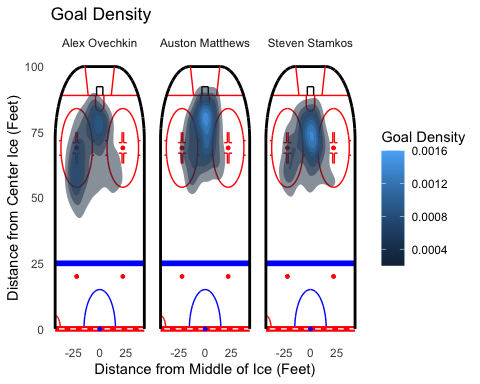
Combining Datasets 1-3

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 right\_join(top9gpg) %>%   
 filter(playerType == "Scorer", fullName %in% c("Alex Ovechkin", "Steven Stamkos", "Auston Matthews")) %>%   
 inner\_join(plays) %>%   
 filter(event == "Goal", st\_x > 25) %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Goal Density", fill = "Goal Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip() +  
 facet\_wrap(.~fullName)

## Joining, by = "player\_id"

## Joining, by = "fullName"

## Joining, by = c("play\_id", "game\_id", "play\_num")



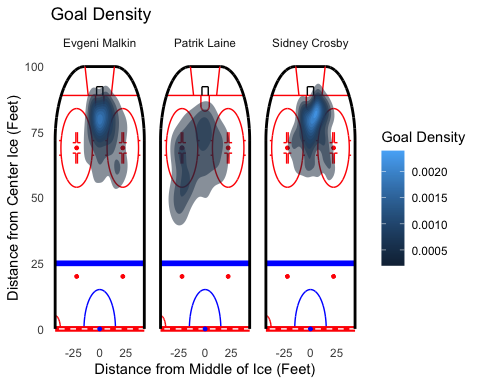
4-6

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 right\_join(top9gpg) %>%   
 filter(playerType == "Scorer", fullName %in% c("Patrik Laine", "Sidney Crosby", "Evgeni Malkin")) %>%   
 inner\_join(plays) %>%   
 filter(event == "Goal", st\_x > 25) %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Goal Density", fill = "Goal Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip() +  
 facet\_wrap(.~fullName)

## Joining, by = "player\_id"

## Joining, by = "fullName"

## Joining, by = c("play\_id", "game\_id", "play\_num")



7-9

players %>%   
 mutate(fullName = paste(firstName, lastName, sep = " ")) %>%   
 select(player\_id, fullName) %>%   
 left\_join(player\_game\_plays) %>%   
 right\_join(top9gpg) %>%   
 filter(playerType == "Scorer", fullName %in% c("Connor McDavid", "John Tavares", "Patrick Kane")) %>%   
 inner\_join(plays) %>%   
 filter(event == "Goal", st\_x > 25) %>%   
 ggplot(aes(st\_x,-st\_y)) +  
 gg\_rink(side="right") +  
 stat\_density2d(aes(fill = ..level..), alpha = .5, geom = "polygon") +  
 labs(title = "Goal Density", fill = "Goal Density", x = "Distance from Center Ice (Feet)", y = "Distance from Middle of Ice (Feet)") +  
 coord\_flip() +  
 facet\_wrap(.~fullName)

## Joining, by = "player\_id"

## Joining, by = "fullName"

## Joining, by = c("play\_id", "game\_id", "play\_num")

