

NHL Goalie and Skater Heat Maps

Sasank Vishnubhatla

January 10, 2019

Last Update: 2019-01-30 19:17:03

GitHub Repository: [sv4u/goalie-and-skater-heat-maps](https://github.com/sv4u/goalie-and-skater-heat-maps)

Introduction

In the NHL, players have their *sweet-spots*. For skaters, it can be the top of a certain circle, or right in-front of the net. For goalies, it could be where their vision is best and where they have the best angle to cut down a shot. All players have these sweet-spots, but it is difficult to analytically say where they are. By using shot location data, we can determine these locations and create models show where goalies and skaters need improvement and where they succeed.

Before we jump in, let's clean up our R environment and also load in some libraries we will be using.

```
rm(list = ls())  
  
library(purrr)  
library(ggplot2)
```

Data Formatting

To start, we need to read in our data. Our data is formatted nicely in CSV format. We have data from the, 2015-2016 season, 2016-2017 season, 2017-2018 season, and 2018-2019 season (up to 1/7/19). This data was downloaded from MoneyPuck. Let's first start by loading in all three seasons of data:

```
data.2015 = read.csv("data/2015.csv")  
data.2016 = read.csv("data/2016.csv")  
data.2017 = read.csv("data/2017.csv")  
data.2018 = read.csv("data/2018.csv")
```

Note: this will take a *relatively* long time to compute as the datasets are large. Each dataset contains all shot data (**including** playoffs).

We'll only look at regular season data. The playoffs in the NHL are a beast of their own.

```
get.regular.season = function(data) {  
  subset(data, isPlayoffGame == 0)  
}  
  
season.2015 = get.regular.season(data.2015)  
season.2016 = get.regular.season(data.2016)  
season.2017 = get.regular.season(data.2017)  
season.2018 = get.regular.season(data.2018)
```

Now that we have our data, we can remove extraneous columns. Here is a table of what columns we are keeping, and what we are renaming them to:

| Old Column | New Column |
|-------------------|-------------|
| xCordAdjusted | x |
| yCordAdjusted | y |
| goal | goal |
| shotAngleAdjusted | angle |
| goalieNameForShot | goalie_name |
| shooterName | skater_name |
| game_id | game |

Now, here is the R code to do this subsetting of the original dataset.

```
get.helpful.data = function(data) {
  data.frame(x = data$xCordAdjusted,
    y = data$yCordAdjusted,
    goal = data$goal,
    angle = data$shotAngleAdjusted,
    goalie_name = data$goalieNameForShot,
    skater_name = data$shooterName,
    game = data$game_id)
}

analysis.2015 = get.helpful.data(season.2015)
analysis.2016 = get.helpful.data(season.2016)
analysis.2017 = get.helpful.data(season.2017)
analysis.2018 = get.helpful.data(season.2018)
```

Now, we have all the data we need.

Function Definitions

Generic

From our data, we can calculate some very important statistics like the following:

- Goal Percent: goals per total shots
- Save Percent: saves (total shots - goals) per total shots
- Shots per Goal: total shots per goal

Additionally, we can break up our data by game. There are some generic functions we can write to help for both goalies and skaters. Let's write them now!

```
get.goal.percent = function(data) {
  shots = length(data$goal)
  temp = subset(data, goal == 1)
  goals = length(temp$goal)
  goals / shots
}

get.save.percent = function(data) {
  shots = length(data$goal)
  temp = subset(data, goal == 1)
  goals = length(temp$goal)
  (shots - goals) / shots
}
```

```

}

get.shots.per.goal = function(data) {
  shots = length(data$goal)
  temp = subset(data, goal == 1)
  goals = length(temp$goal)
  shots / goals
}

```

Note: when using `get.shots.per.goal`, if there were no goals scored, R will handle the division by zero by returning infinity. This will be problematic when graphing data. I am still working on a good solution to this problem. Earlier, I used 200 as a substitute value. However, 200 still skews graphs, which is unideal.

```

get.games = function(data) {
  unique(data$game)
}

get.single.game = function(data, game_id) {
  subset(data, game == game_id)
}

get.all.games = function(data) {
  games = get.games(data)
  Map(function(x) get.single.game(data, x), games)
}

```

Now, we can create our game by game statistic functions:

```

get.game.goal.percent = function(data) {
  gameframe = get.all.games(data)
  games.gp = map(gameframe, function(x) get.goal.percent(x))
  unlist(games.gp, use.names = FALSE)
}

get.game.save.percent = function(data) {
  gameframe = get.all.games(data)
  games.sp = map(gameframe, function(x) get.save.percent(x))
  unlist(games.sp, use.names = FALSE)
}

get.game.shots.per.goal = function(data) {
  gameframe = get.all.games(data)
  games.spg = map(gameframe, function(x) get.shots.per.goal(x))
  unlist(games.spg, use.names = FALSE)
}

```

Also, we'll need a function to get match-ups between a specific goalie and skater. Let's write that here, instead of in our goalies *and* our skaters sections.

```

get.matchup.data = function(data, goalie, skater) {
  subset(data, goalie_name == goalie & skater_name == skater)
}

```

We've now written our generic data handling functions.

Goalies

Let's first start with a function to get data for a specific goalie.

```
get.goalie.data = function(data, name) {  
  subset(data, goalie_name == name)  
}
```

Skaters

Let's first start with a function to get data for a specific skater.

```
get.skater.data = function(data, name) {  
  subset(data, skater_name == name)  
}
```

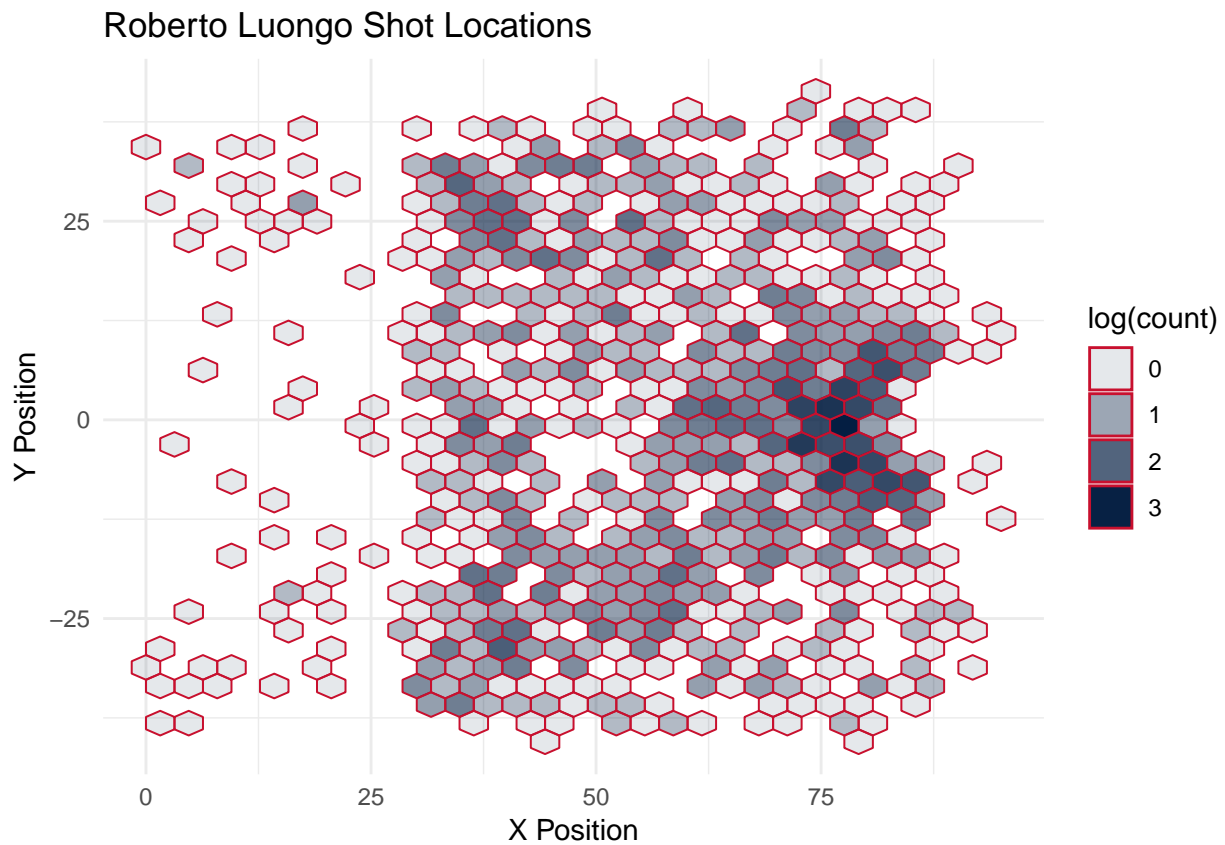
Graphing

Given specific data, we should be able to graph the location of shots. Let's write a function that uses ggplot to do so.

```
graph.shot.locations = function(data, primary, secondary, name) {  
  plot = ggplot(data) +  
    geom_hex(aes(x = x, y = y, alpha = log(..count..)), fill = primary, color = secondary) +  
    labs(title = paste(name, "Shot Locations", sep = " "), x = "X Position", y = "Y Position") +  
    theme_minimal()  
  plot  
}
```

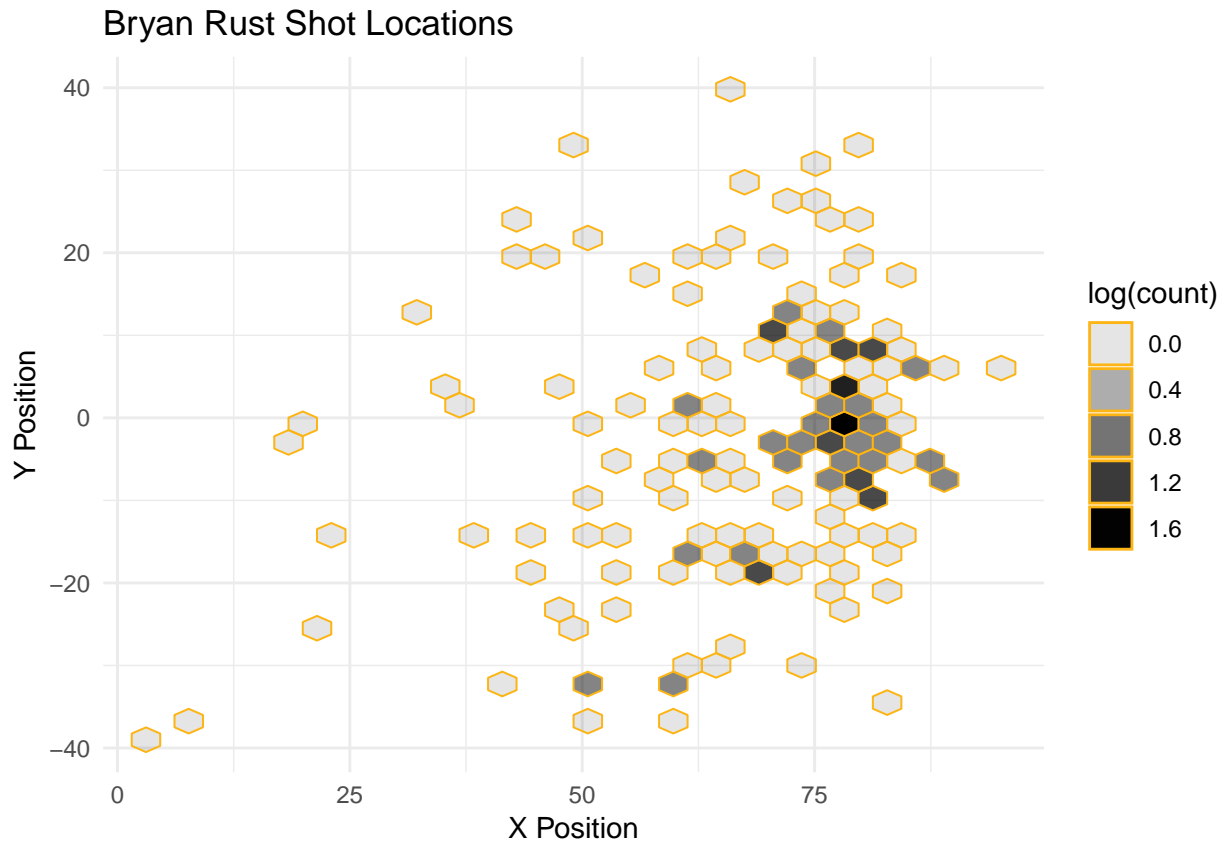
To see a test of what this does, let's quickly make a graph of Roberto Luongo's shots against him.

```
luongo = get.goalie.data(analysis.2017, "Roberto Luongo")  
plot = graph.shot.locations(luongo, "#041E42", "#C8102E", "Roberto Luongo")  
  
plot
```



Now, let's see how it looks for a skater. Let's look at Bryan Rust.

```
rust = get.skater.data(analysis.2017, "Bryan Rust")
plot = graph.shot.locations(rust, "#000000", "#FCB514", "Bryan Rust")
plot
```

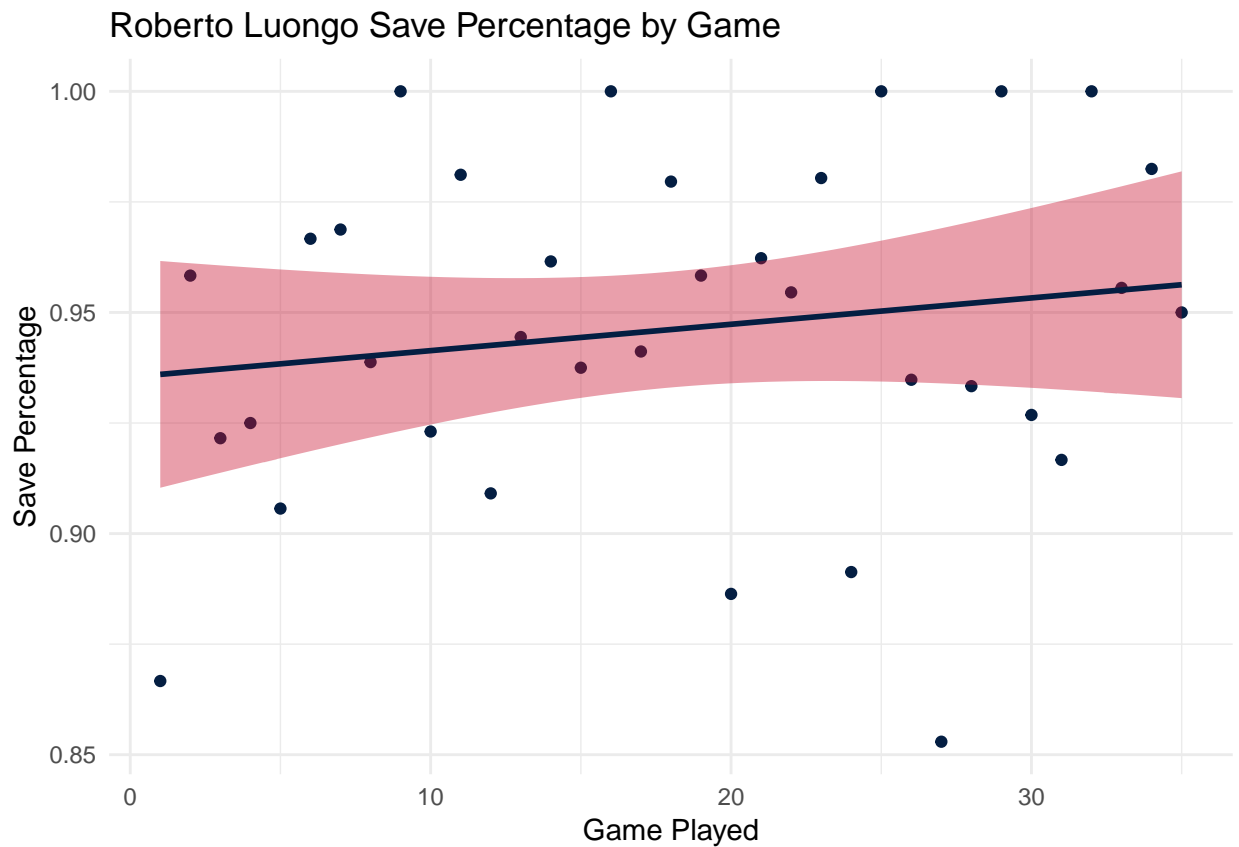


Furthermore, we should be able to graph trends in certain statistics.

```
graph.trend = function(trend, type, primary, secondary, name) {
  frame = data.frame(x = c(1:length(trend)), y = trend)
  disp = paste(name, type, "by Game", sep = " ")
  plot = ggplot(frame) +
    geom_point(aes(x = x, y = y), color = primary) +
    geom_smooth(aes(x = x, y = y), method = "lm", color = primary, fill = secondary) +
    labs(title = disp, x = "Game Played", y = type) +
    theme_minimal()
  plot
}
```

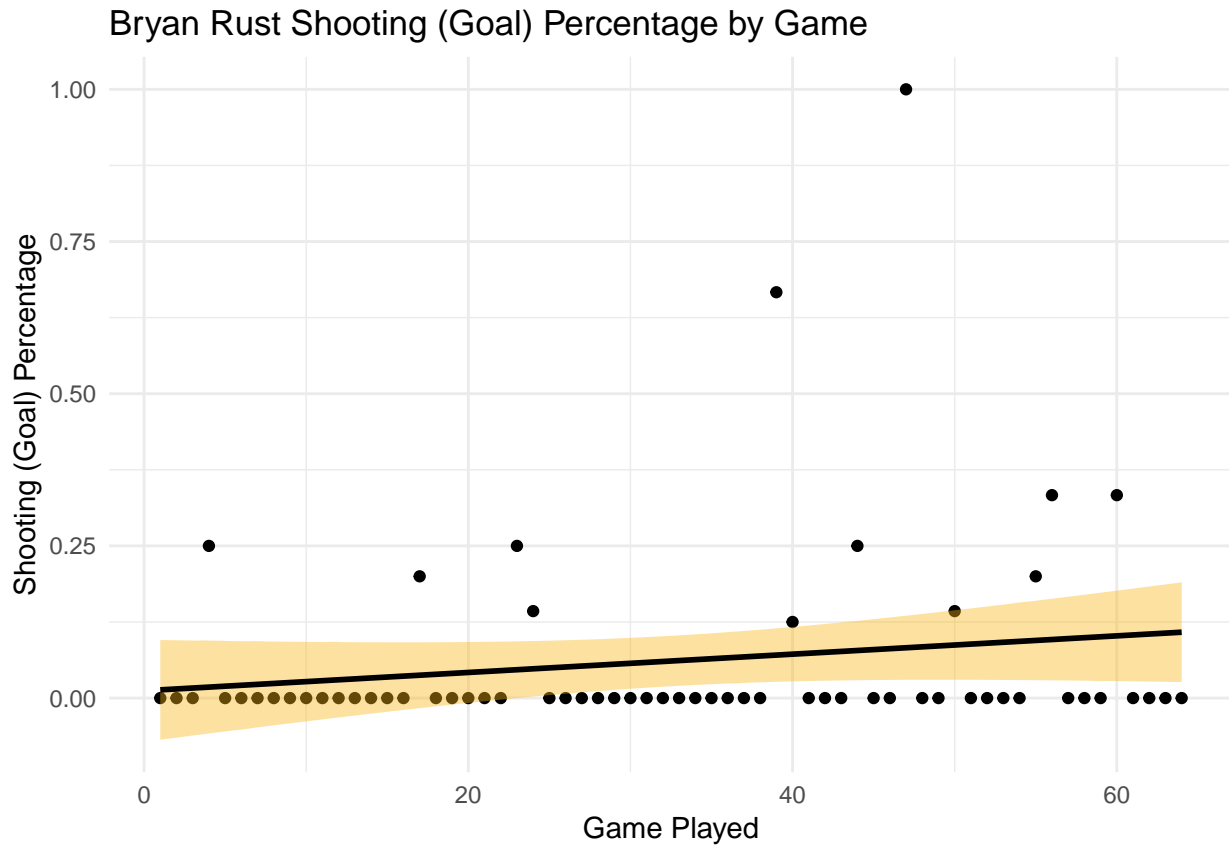
So, let's go back to Luongo and look at his save percentage per game.

```
luongo.game.sp = get.game.save.percent(luongo)
plot = graph.trend(luongo.game.sp, "Save Percentage", "#041E42", "#C8102E", "Roberto Luongo")
plot
```



We can also see the shooter's perspective. Let's look at Rust's shooting percentage (goal percent).

```
rust.game.gp = get.game.goal.percent(rust)
plot = graph.trend(rust.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB514", "Bryan Rust")
plot
```




```
murray.2015.spg = get.shots.per.goal(murray.2015)
murray.2016.spg = get.shots.per.goal(murray.2016)
murray.2017.spg = get.shots.per.goal(murray.2017)
murray.2018.spg = get.shots.per.goal(murray.2018)
```

Now, let's put it in a table:

| Season | Save Percent | Goal Percent | Shots Per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.9460043 | 0.0539957 | 18.52 |
| 2016-2017 | 0.9429599 | 0.0570401 | 17.5315315 |
| 2017-2018 | 0.9301837 | 0.0698163 | 14.3233083 |
| 2018-2019 | 0.9374185 | 0.0625815 | 15.9791667 |

We can look at his game by game data also.

```
murray.2015.games.sp = get.game.save.percent(murray.2015)
murray.2015.games.gp = get.game.goal.percent(murray.2015)
murray.2015.games.spg = get.game.shots.per.goal(murray.2015)

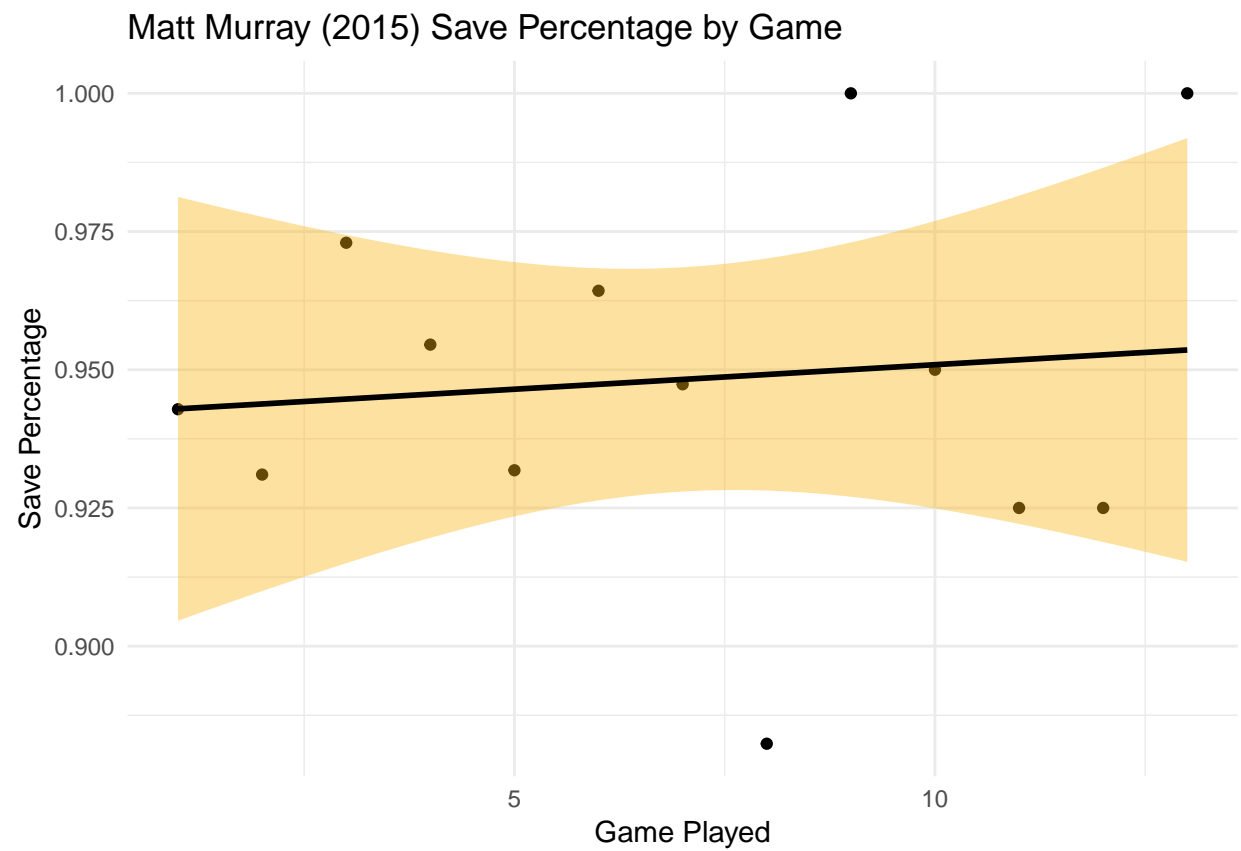
murray.2016.games.sp = get.game.save.percent(murray.2016)
murray.2016.games.gp = get.game.goal.percent(murray.2016)
murray.2016.games.spg = get.game.shots.per.goal(murray.2016)

murray.2017.games.sp = get.game.save.percent(murray.2017)
murray.2017.games.gp = get.game.goal.percent(murray.2017)
murray.2017.games.spg = get.game.shots.per.goal(murray.2017)

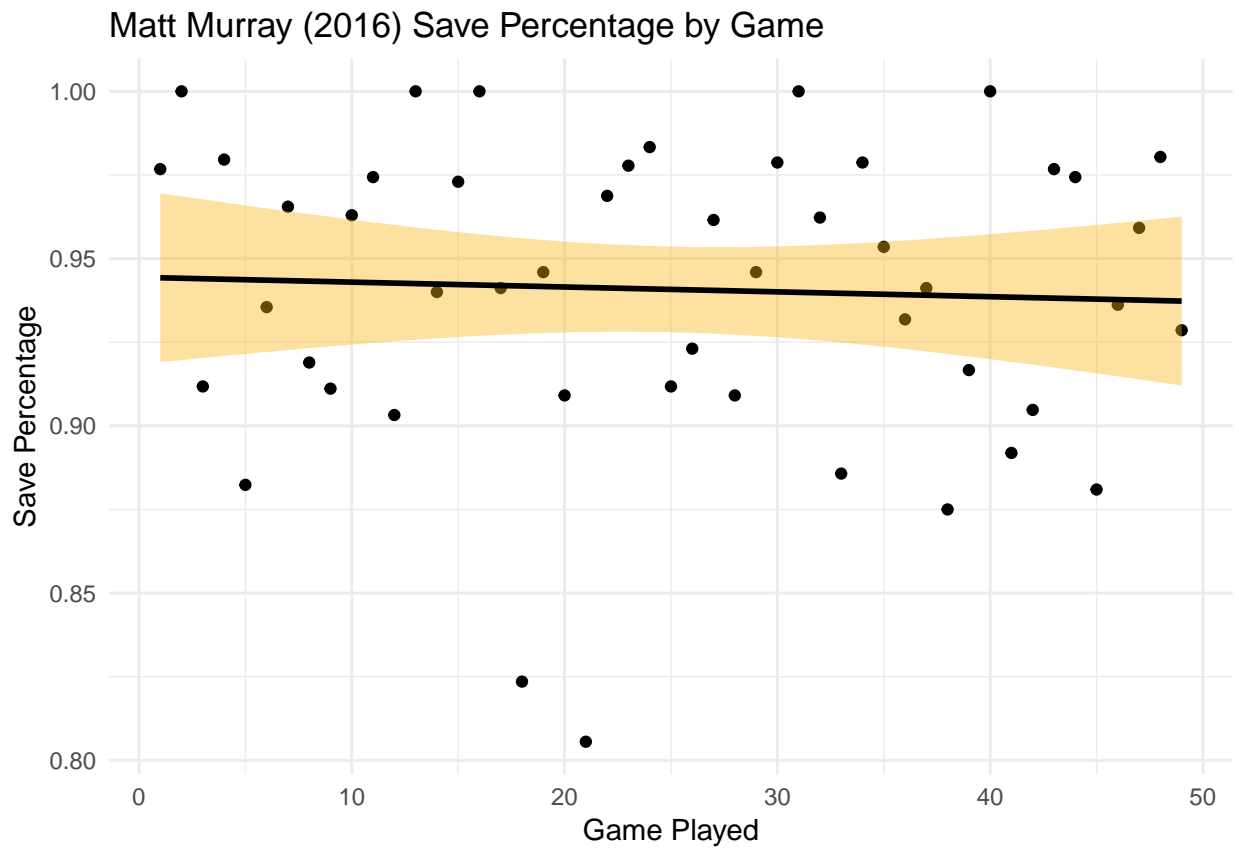
murray.2018.games.sp = get.game.save.percent(murray.2018)
murray.2018.games.gp = get.game.goal.percent(murray.2018)
murray.2018.games.spg = get.game.shots.per.goal(murray.2018)
```

Let's look at his save percentage graphs:

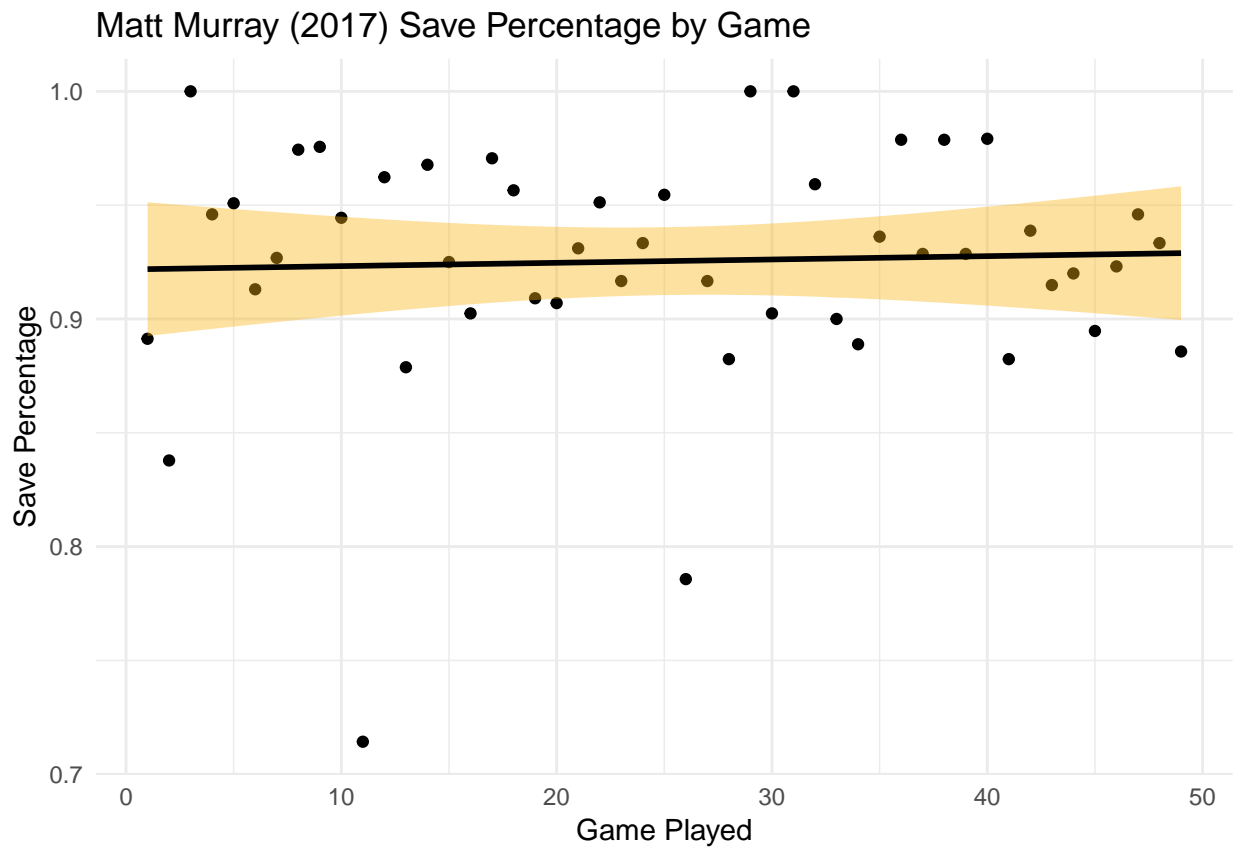
```
murray.2015.sp.plot = graph.trend(murray.2015.games.sp, "Save Percentage", "#000000", "#FCB514", "Matt L"
murray.2015.sp.plot
```



```
murray.2016.sp.plot = graph.trend(murray.2016.games.sp, "Save Percentage", "#000000", "#FCB514", "Matt Murray")
murray.2016.sp.plot
```

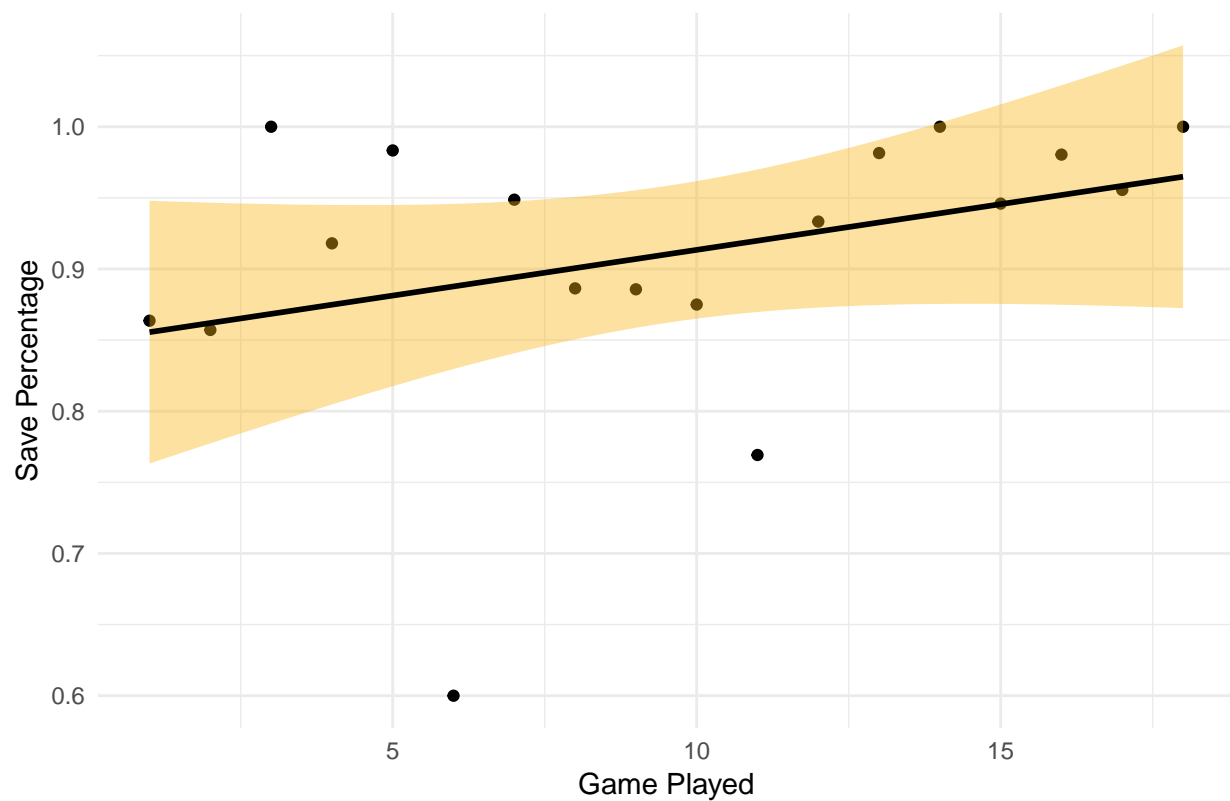


```
murray.2017.sp.plot = graph.trend(murray.2017.games.sp, "Save Percentage", "#000000", "#FCB514", "Matt Murray")
murray.2017.sp.plot
```



```
murray.2018.sp.plot = graph.trend(murray.2018.games.sp, "Save Percentage", "#000000", "#FCB514", "Matt Murray")
murray.2018.sp.plot
```

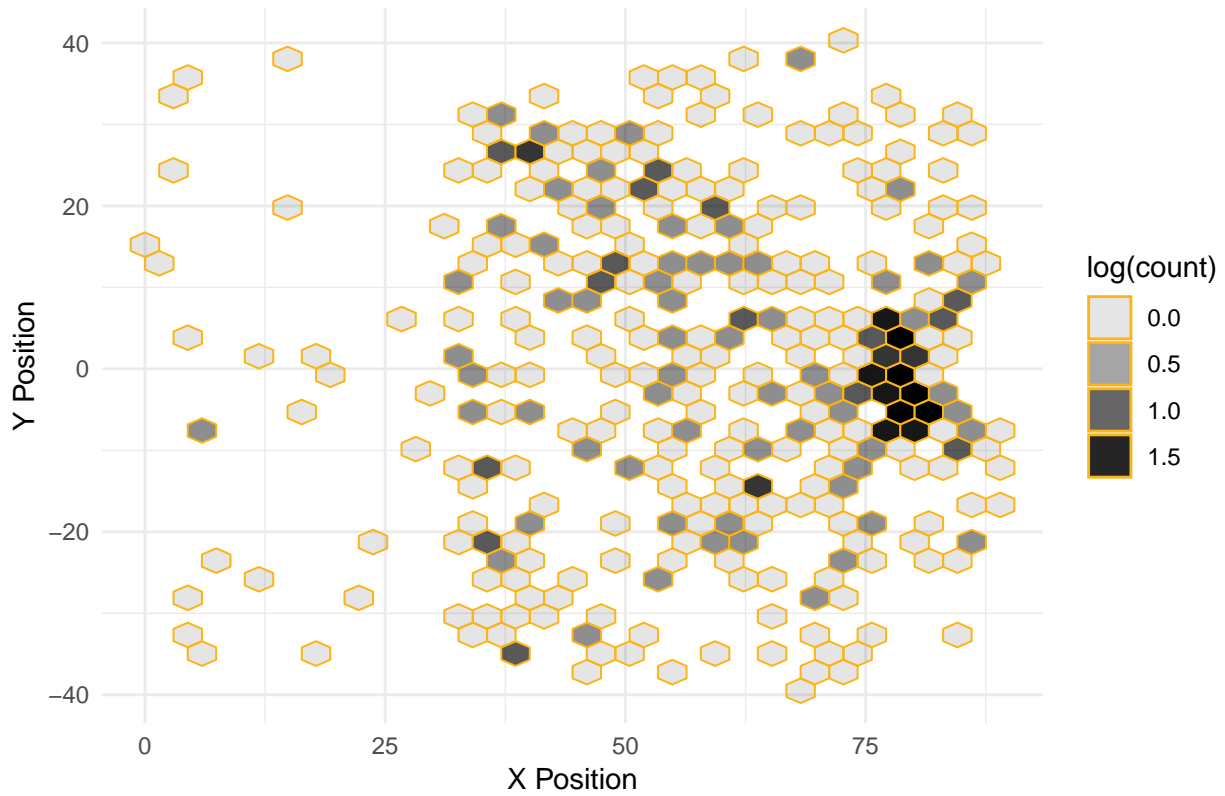
Matt Murray (2018) Save Percentage by Game



And now, let's take a look at his shot location data.

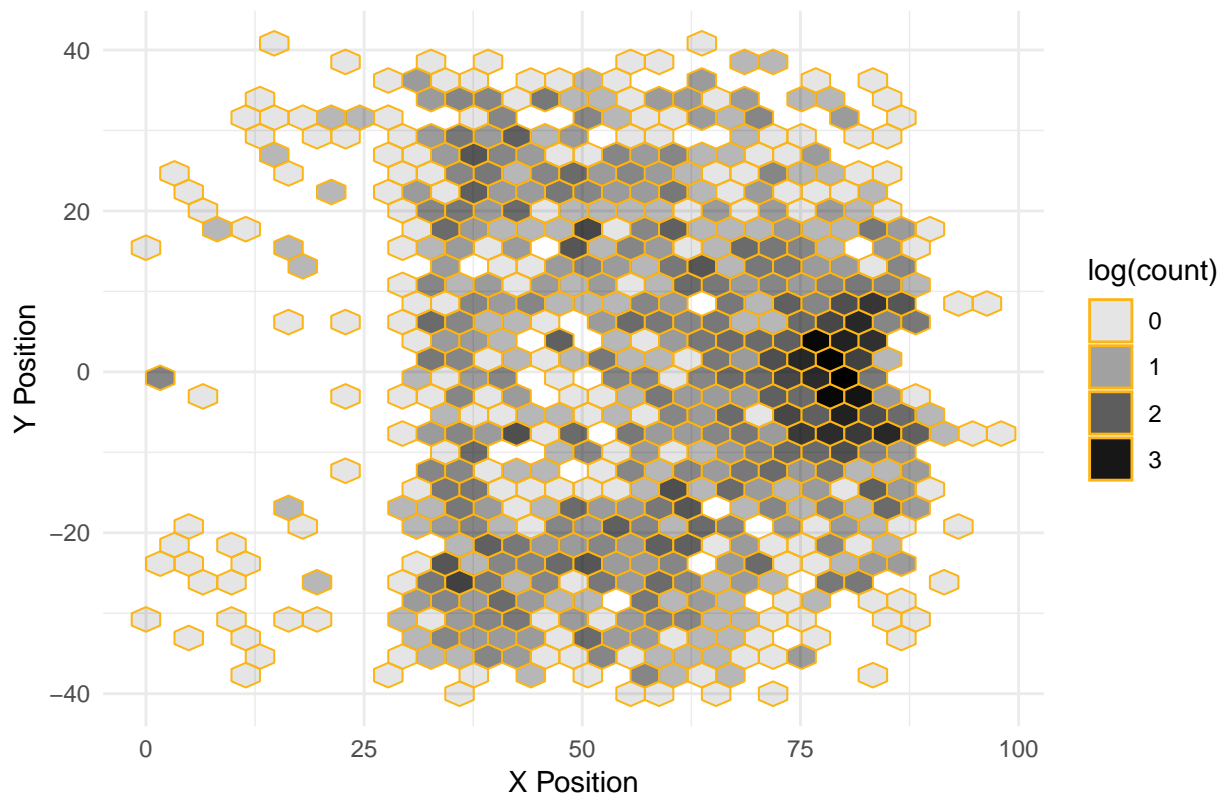
```
murray.2015.locations.plot = graph.shot.locations(murray.2015, "#000000", "#FCB514", "Matt Murray (2015)", "#000000", "#FCB514", "#000000", "#FCB514")
murray.2015.locations.plot
```

Matt Murray (2015) Shot Locations



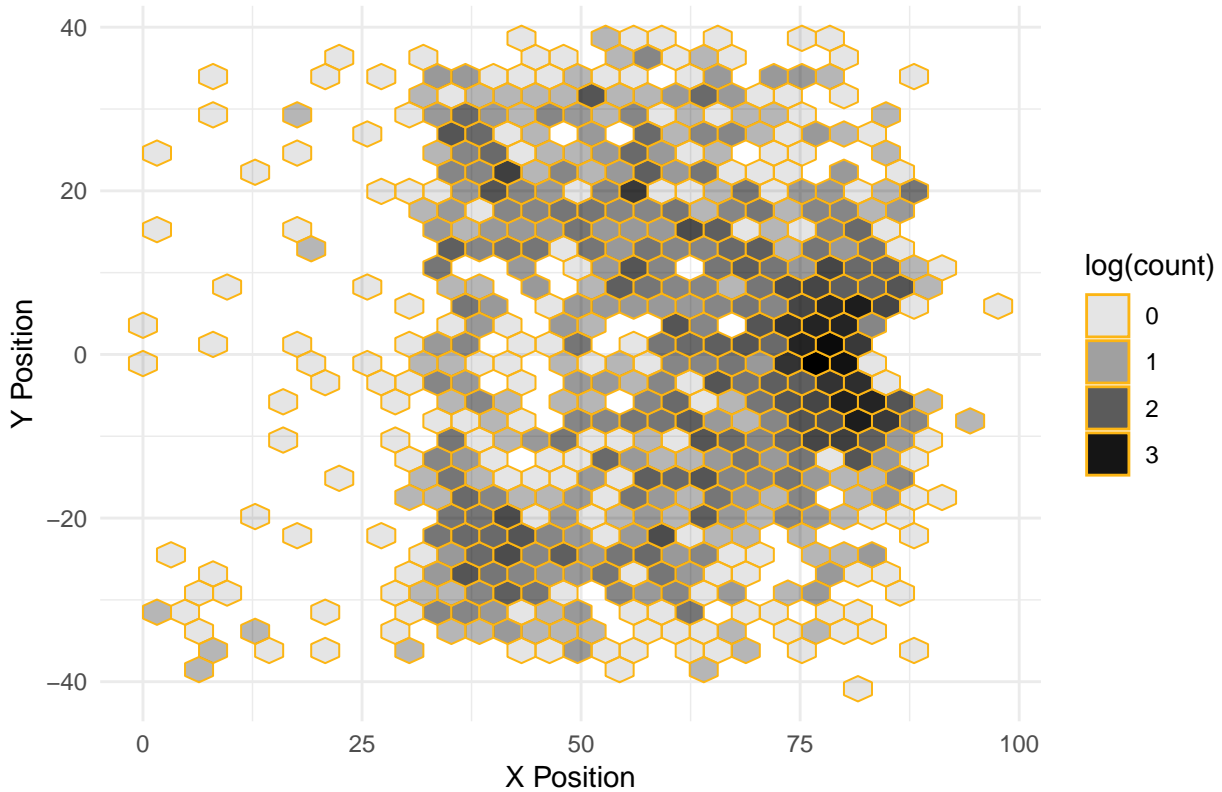
```
murray.2016.locations.plot = graph.shot.locations(murray.2016, "#000000", "#FCB514", "Matt Murray (2016)")
murray.2016.locations.plot
```

Matt Murray (2016) Shot Locations



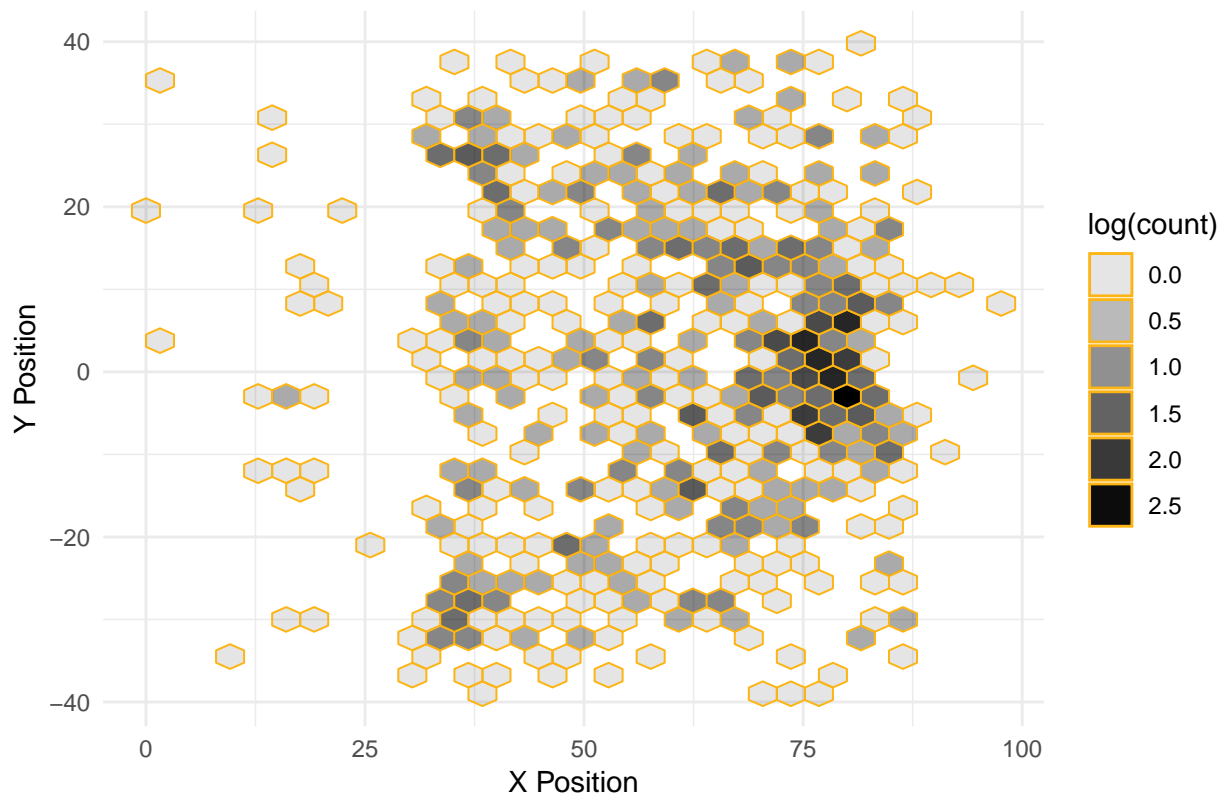
```
murray.2017.locations.plot = graph.shot.locations(murray.2017, "#000000", "#FCB514", "Matt Murray (2017)", "log(count)")
murray.2017.locations.plot
```

Matt Murray (2017) Shot Locations



```
murray.2018.locations.plot = graph.shot.locations(murray.2018, "#000000", "#FCB514", "Matt Murray (2018)")
murray.2018.locations.plot
```


Matt Murray (2018) Shot Locations



From the darkness of each hex bin, we can use that as a percent chance that Murray saved the shot. Though the darkness is not exactly the save percentage of the shots, it's the log chance of Murray saving the shot. This log chance is a good likelihood function of Murray saving the shot. We can see that Murray's weakness is very close to him. Those shots are most likely to be tipped or deflected shots, not actual shots. We can see his weakness is the top of the circles, based on the X and Y coordinates.

From the graph, we can see that Murray has struggled with shots near the circles. Disregarding tips, it seems like that Murray is positionally sound with goals against near him. However, compared to the right wing, Murray seems weak against shots coming from the left wing side. If teams were to exploit this, it would be beneficial to shoot towards Murray's glove side from the left wing circle.

Casey DeSmith

On January 11th, Casey DeSmith inked a 3 year extension with the Penguins with an average annual value of \$1.25 million. His rise to earn this paycheck started with exemplary play during the 2017-2018 season and that has carried over into this season. Let's see how he got here!

```
desmith.2017 = get.goalie.data(analysis.2017, "Casey DeSmith")
desmith.2018 = get.goalie.data(analysis.2018, "Casey DeSmith")
```

Let's start by calculating some of his stats for each season and then tabularizing.

```
desmith.2017.sp = get.save.percent(desmith.2017)
desmith.2018.sp = get.save.percent(desmith.2018)

desmith.2017.gp = get.goal.percent(desmith.2017)
desmith.2018.gp = get.goal.percent(desmith.2018)
```

```
desmith.2017.spg = get.shots.per.goal(desmith.2017)
desmith.2018.spg = get.shots.per.goal(desmith.2018)
```

Now, let's put it in a table:

| Season | Save Percent | Goal Percent | Shots Per Goal |
|-----------|--------------|--------------|----------------|
| 2017-2018 | 0.9427403 | 0.0572597 | 17.4642857 |
| 2018-2019 | 0.9422886 | 0.0577114 | 17.3275862 |

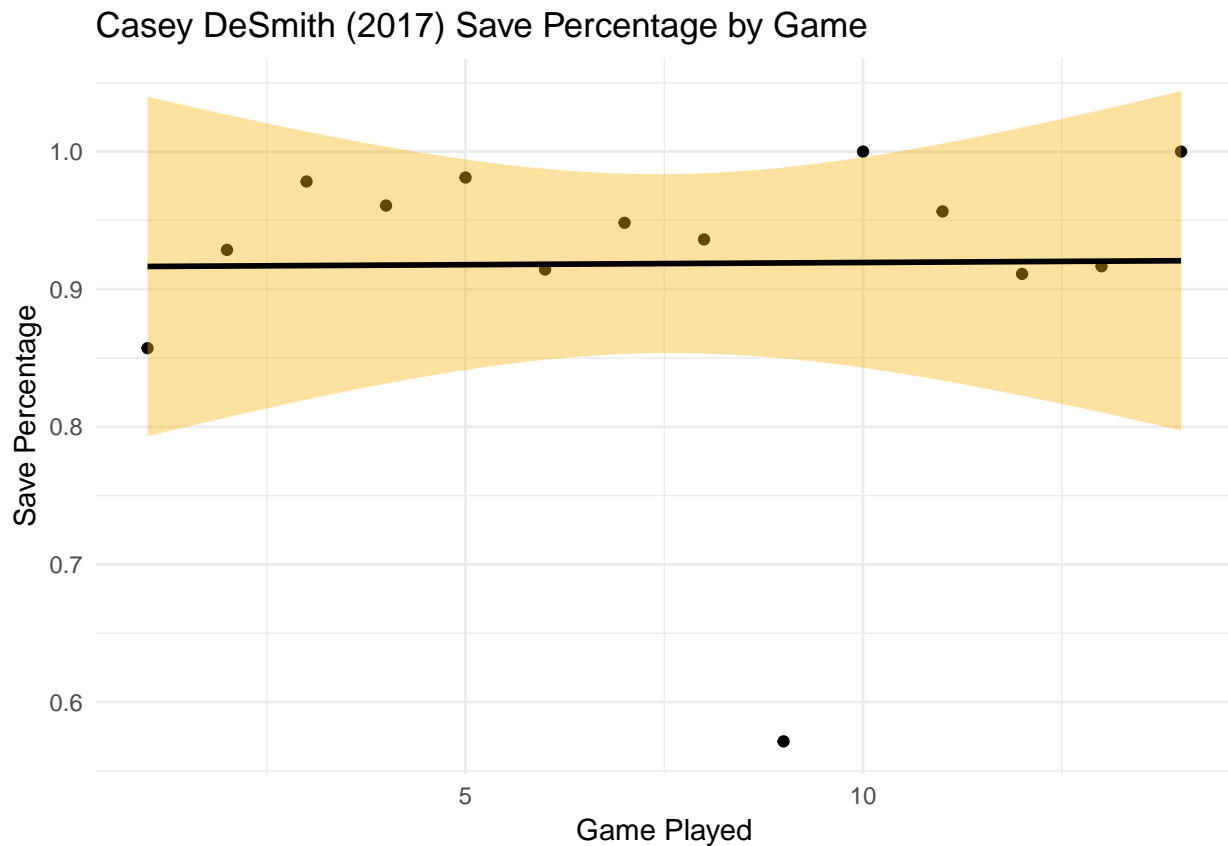
We can look at his game by game data also.

```
desmith.2017.games.sp = get.game.save.percent(desmith.2017)
desmith.2017.games.gp = get.game.goal.percent(desmith.2017)
desmith.2017.games.spg = get.game.shots.per.goal(desmith.2017)

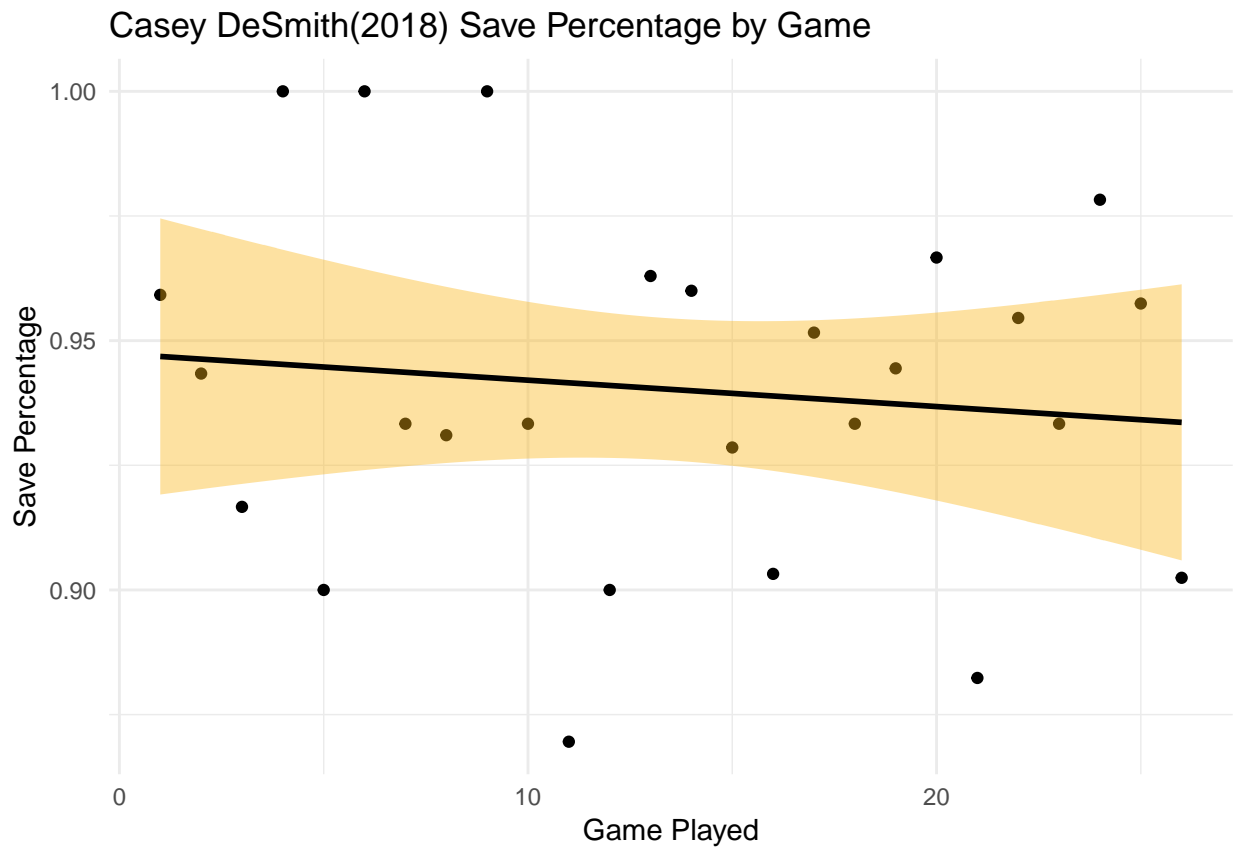
desmith.2018.games.sp = get.game.save.percent(desmith.2018)
desmith.2018.games.gp = get.game.goal.percent(desmith.2018)
desmith.2018.games.spg = get.game.shots.per.goal(desmith.2018)
```

Let's look at his save percentage graphs:

```
desmith.2017.sp.plot = graph.trend(desmith.2017.games.sp, "Save Percentage", "#000000", "#FCB514", "Casey DeSmith (2017) Save Percentage by Game")
desmith.2017.sp.plot
```



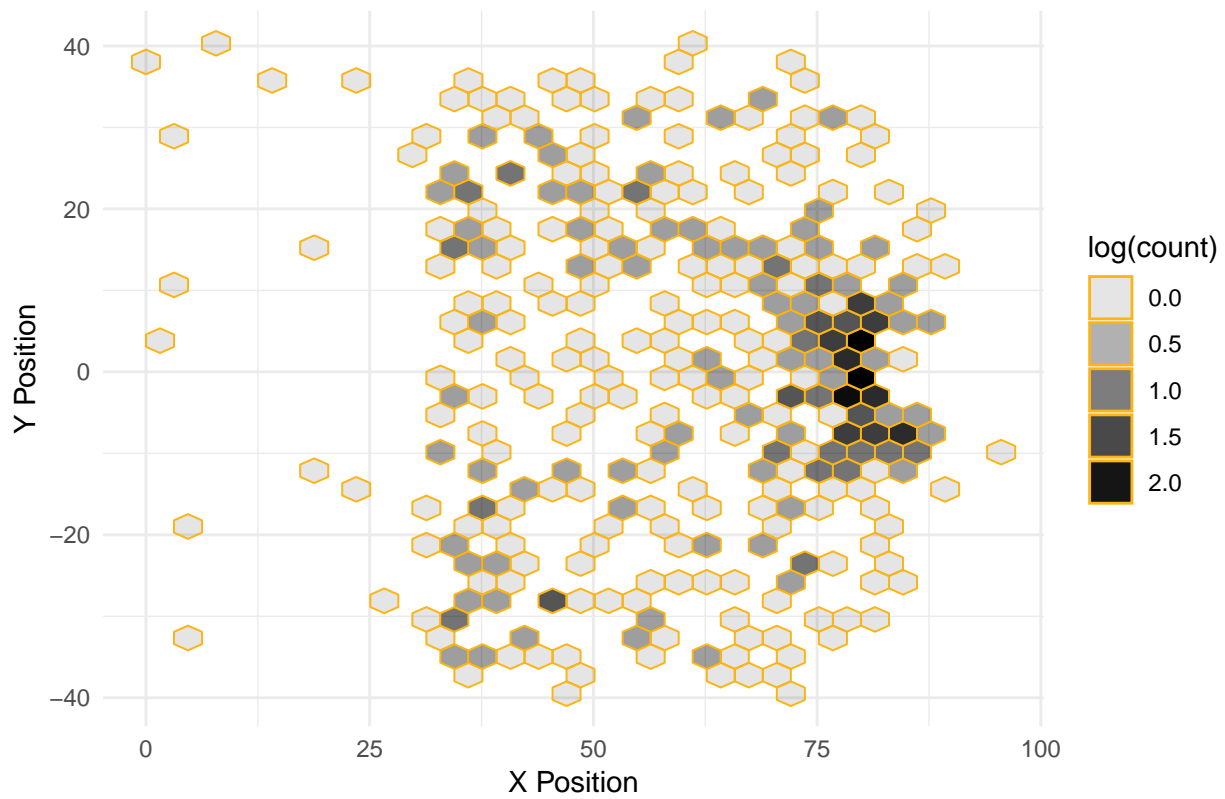
```
desmith.2018.sp.plot = graph.trend(desmith.2018.games.sp, "Save Percentage", "#000000", "#FCB514", "Casey DeSmith (2018) Save Percentage by Game")
desmith.2018.sp.plot
```



And now, let's take a look at his shot location data.

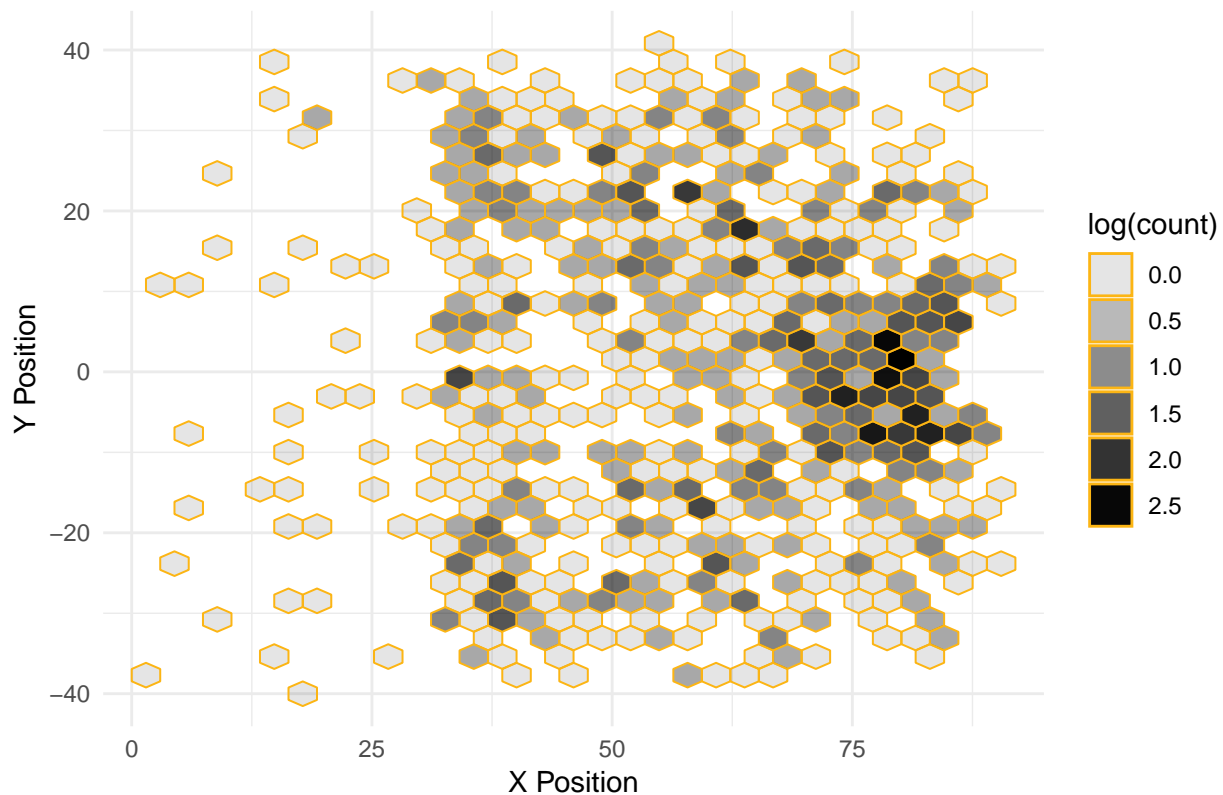
```
desmith.2017.locations.plot = graph.shot.locations(desmith.2017, "#000000", "#FCB514", "Casey DeSmith (")
desmith.2017.locations.plot
```

Casey DeSmith (2017) Shot Locations



```
desmith.2018.locations.plot = graph.shot.locations(desmith.2018, "#000000", "#FCB514", "Casey DeSmith (")
desmith.2018.locations.plot
```

Casey DeSmith (2018) Shot Locations



As we can see, DeSmith has played pretty phenomenally. He's earned his new extension and his play has been pretty consistent. He is the perfect complement for Matt Murray.

Marc-Andre Fleury

Before he was picked by the Vegas Golden Knights in the most recent NHL Expansion Draft, Marc-Andre Fleury was the franchise goaltender for the Pittsburgh Penguins. Drafted first overall in the 2003 NHL Entry Draft, Marc-Andre "Flower" Fleury made his NHL debut in October of 2003. Since then he's been the main net-protector for the Penguins (until 2017 that is). His athleticism plus his exceptionally quick reactions makes Flower a premier NHL goaltender. With 3 Stanley Cup victories (2009, 2016, and 2017), Fleury has cemented his place as a top-10 all-time goaltender. With 429 regular season wins, Fleury sits at number 9 for career regular season wins. Only two active goaltenders, Roberto Luongo and Henrik Lundqvist, sit higher above him.

Fleury has been playing phenomenally the past few seasons. What seems like a second-prime for the Canadian born goaltender, Fleury showed shades of his 2008-2009 season, when he led the Penguins to their third Stanley Cup.

```
fleury.2015 = get.goalie.data(analysis.2015, "Marc-Andre Fleury")
fleury.2016 = get.goalie.data(analysis.2016, "Marc-Andre Fleury")
fleury.2017 = get.goalie.data(analysis.2017, "Marc-Andre Fleury")
fleury.2018 = get.goalie.data(analysis.2018, "Marc-Andre Fleury")
```

Let's start by calculating some of his stats for each season and then tabularizing.

```
fleury.2015.sp = get.save.percent(fleury.2015)
fleury.2016.sp = get.save.percent(fleury.2016)
fleury.2017.sp = get.save.percent(fleury.2017)
```

```
fleury.2018.sp = get.save.percent(fleury.2018)

fleury.2015.gp = get.goal.percent(fleury.2015)
fleury.2016.gp = get.goal.percent(fleury.2016)
fleury.2017.gp = get.goal.percent(fleury.2017)
fleury.2018.gp = get.goal.percent(fleury.2018)

fleury.2015.spg = get.shots.per.goal(fleury.2015)
fleury.2016.spg = get.shots.per.goal(fleury.2016)
fleury.2017.spg = get.shots.per.goal(fleury.2017)
fleury.2018.spg = get.shots.per.goal(fleury.2018)
```

Now, let's put it in a table:

| Season | Save Percent | Goal Percent | Shots Per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.9427197 | 0.0572803 | 17.4580153 |
| 2016-2017 | 0.9314803 | 0.0685197 | 14.5943396 |
| 2017-2018 | 0.9475891 | 0.0524109 | 19.08 |
| 2018-2019 | 0.9365621 | 0.0634379 | 15.7634409 |

We can look at his game by game data also.

```
fleury.2015.games.sp = get.game.save.percent(fleury.2015)
fleury.2015.games.gp = get.game.goal.percent(fleury.2015)
fleury.2015.games.spg = get.game.shots.per.goal(fleury.2015)

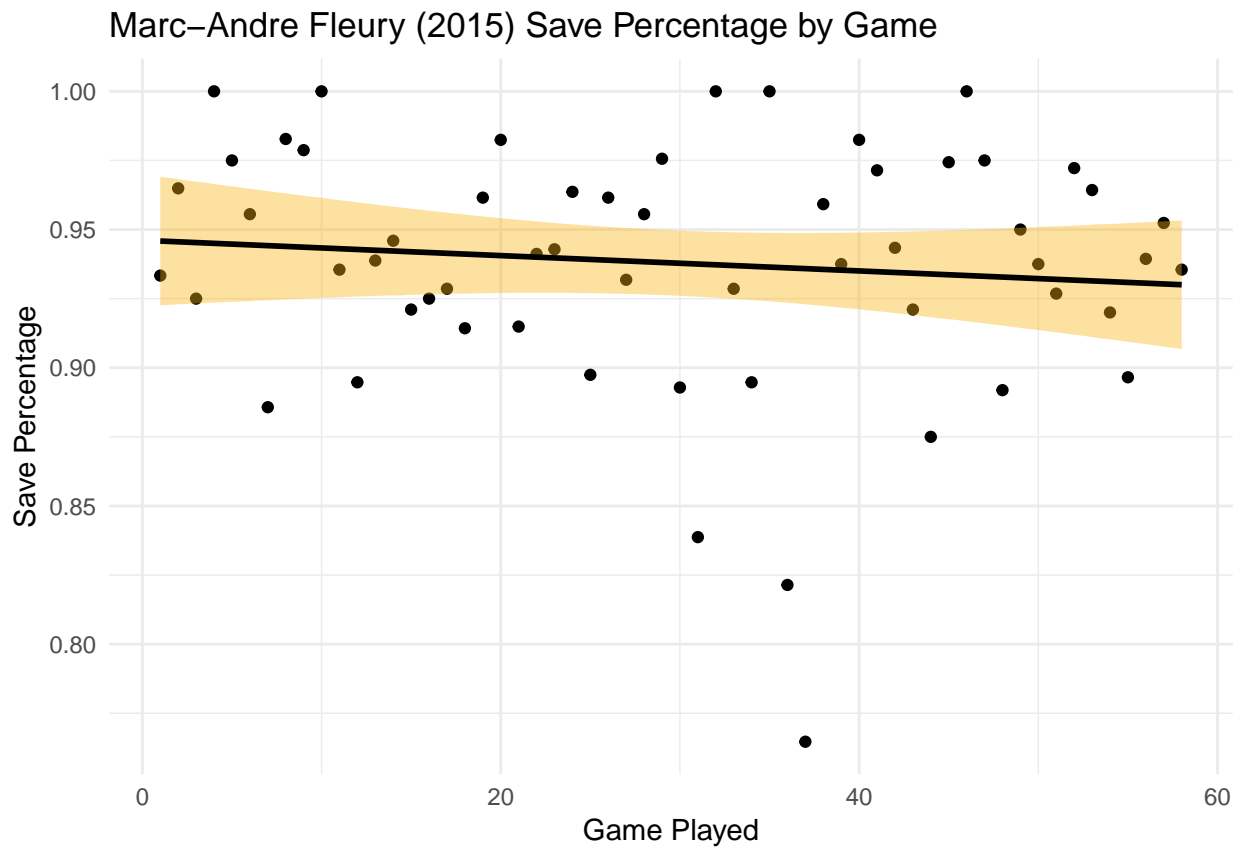
fleury.2016.games.sp = get.game.save.percent(fleury.2016)
fleury.2016.games.gp = get.game.goal.percent(fleury.2016)
fleury.2016.games.spg = get.game.shots.per.goal(fleury.2016)

fleury.2017.games.sp = get.game.save.percent(fleury.2017)
fleury.2017.games.gp = get.game.goal.percent(fleury.2017)
fleury.2017.games.spg = get.game.shots.per.goal(fleury.2017)

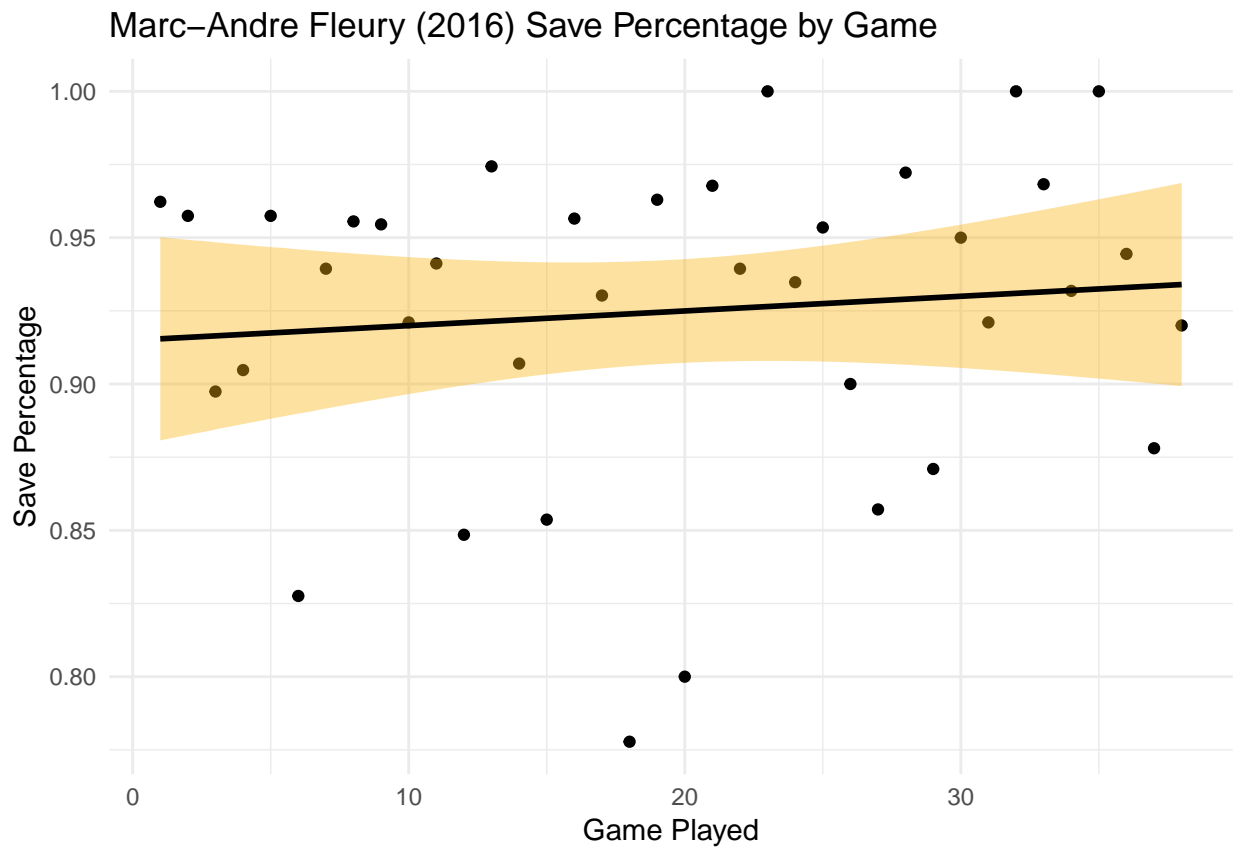
fleury.2018.games.sp = get.game.save.percent(fleury.2018)
fleury.2018.games.gp = get.game.goal.percent(fleury.2018)
fleury.2018.games.spg = get.game.shots.per.goal(fleury.2018)
```

Let's look at his save percentage graphs:

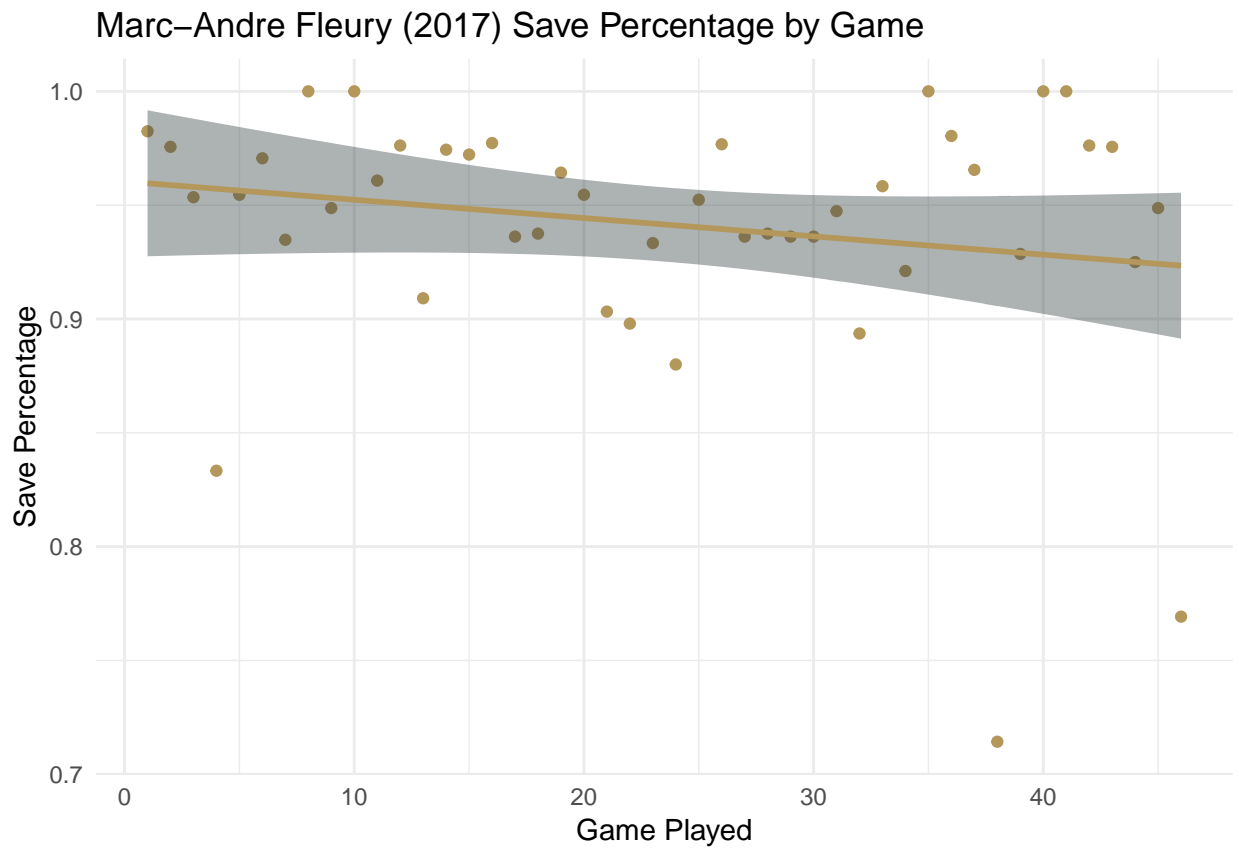
```
fleury.2015.sp.plot = graph.trend(fleury.2015.games.sp, "Save Percentage", "#000000", "#FCB514", "Marc-André Fleury")
fleury.2015.sp.plot
```



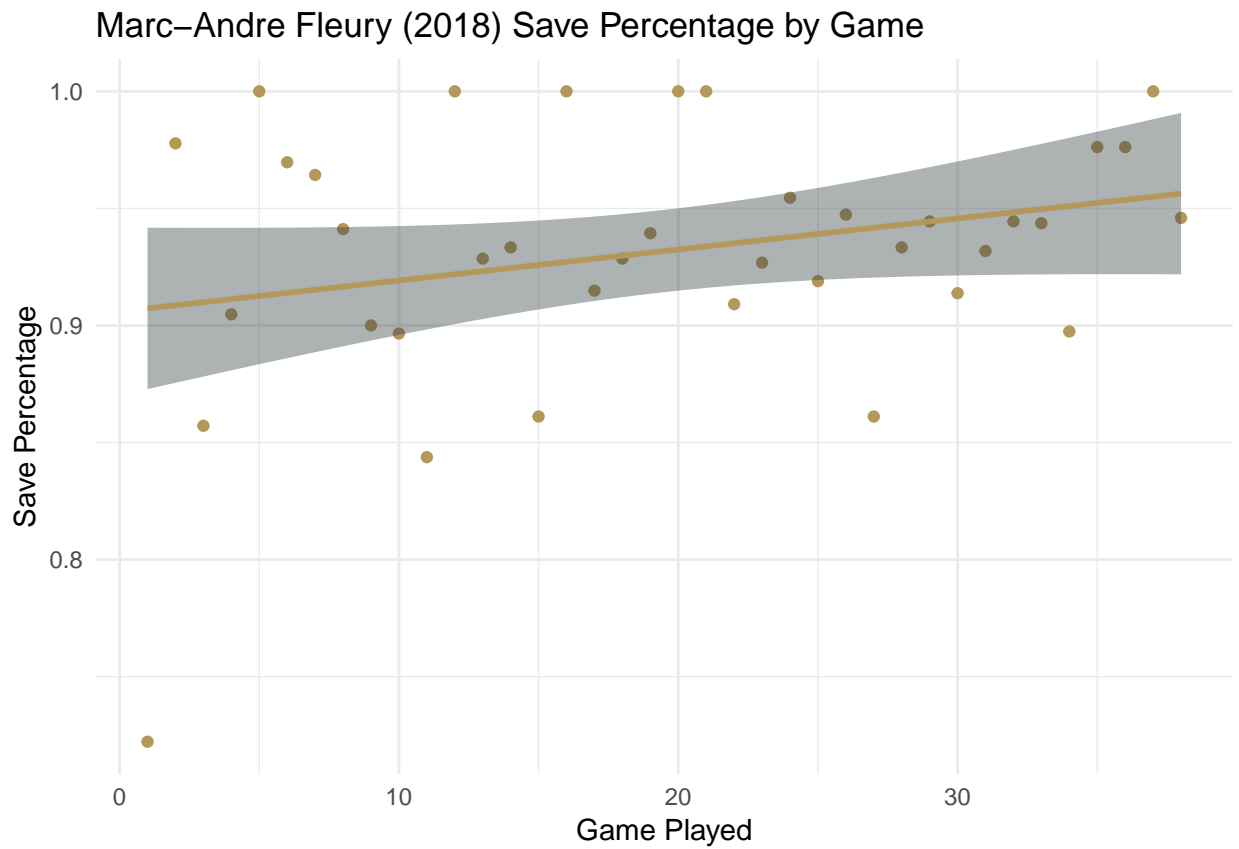
```
fleury.2016.sp.plot = graph.trend(fleury.2016.games.sp, "Save Percentage", "#000000", "#FCB514", "Marc-Andre Fleury")
fleury.2016.sp.plot
```



```
fleury.2017.sp.plot = graph.trend(fleury.2017.games.sp, "Save Percentage", "#B4975A", "#333F42", "Marc-Andre Fleury")
fleury.2017.sp.plot
```

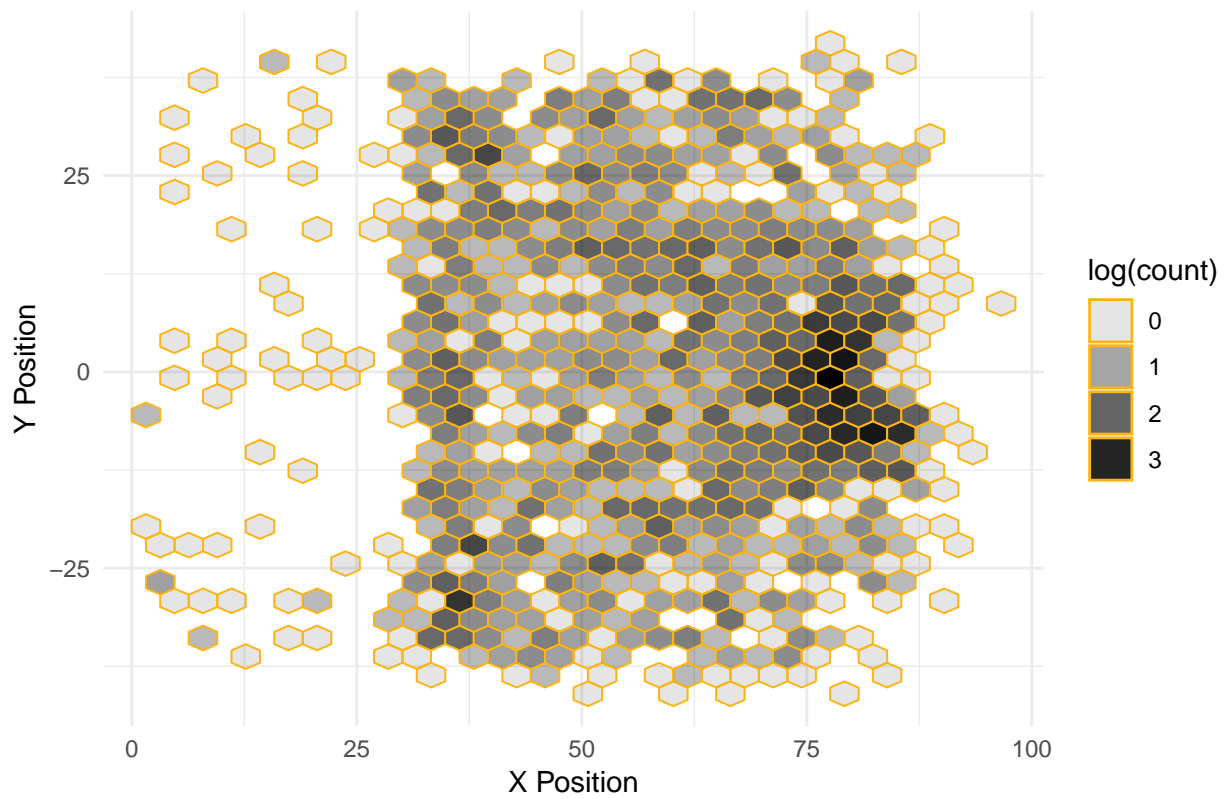
```
fleury.2018.sp.plot = graph.trend(fleury.2018.games.sp, "Save Percentage", "#B4975A", "#333F42", "Marc-Andre Fleury", "2017")
fleury.2018.sp.plot
```



And now, let's take a look at his shot location data.

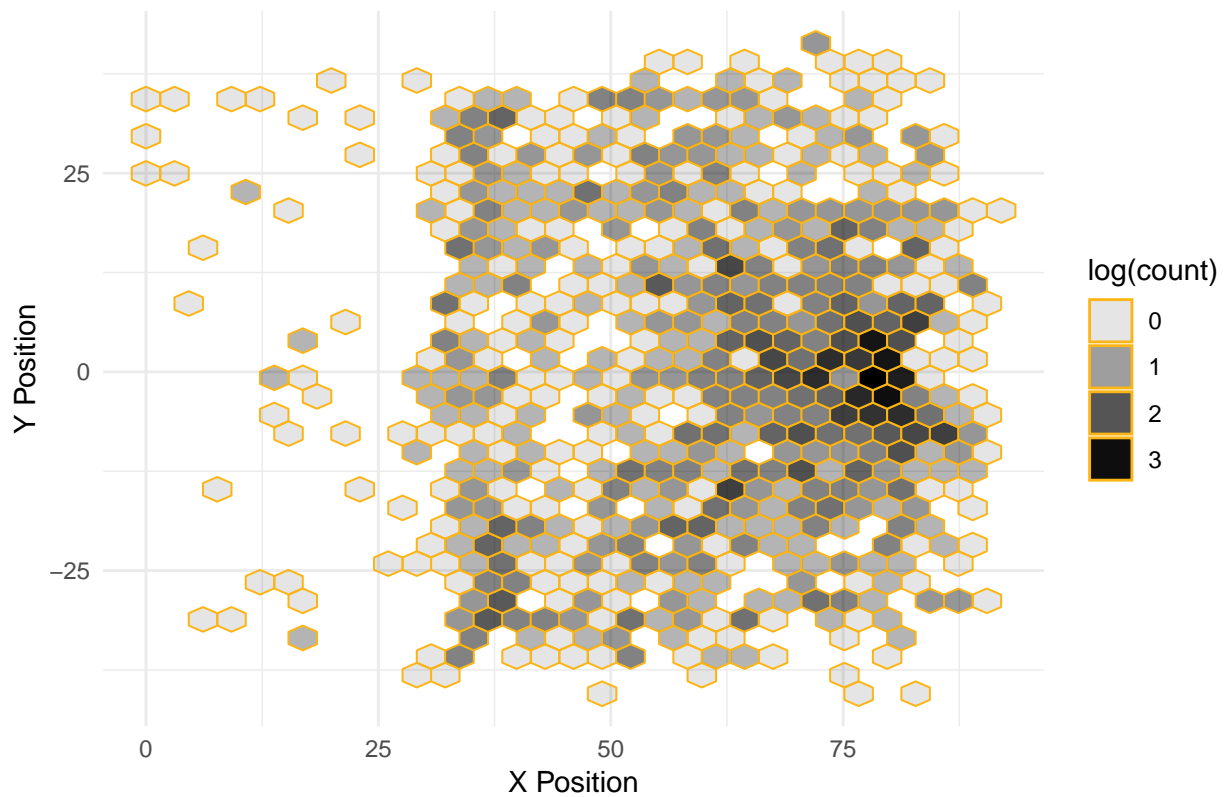
```
fleury.2015.locations.plot = graph.shot.locations(fleury.2015, "#000000", "#FCB514", "Marc-Andre Fleury")
fleury.2015.locations.plot
```

Marc-Andre Fleury (2015) Shot Locations



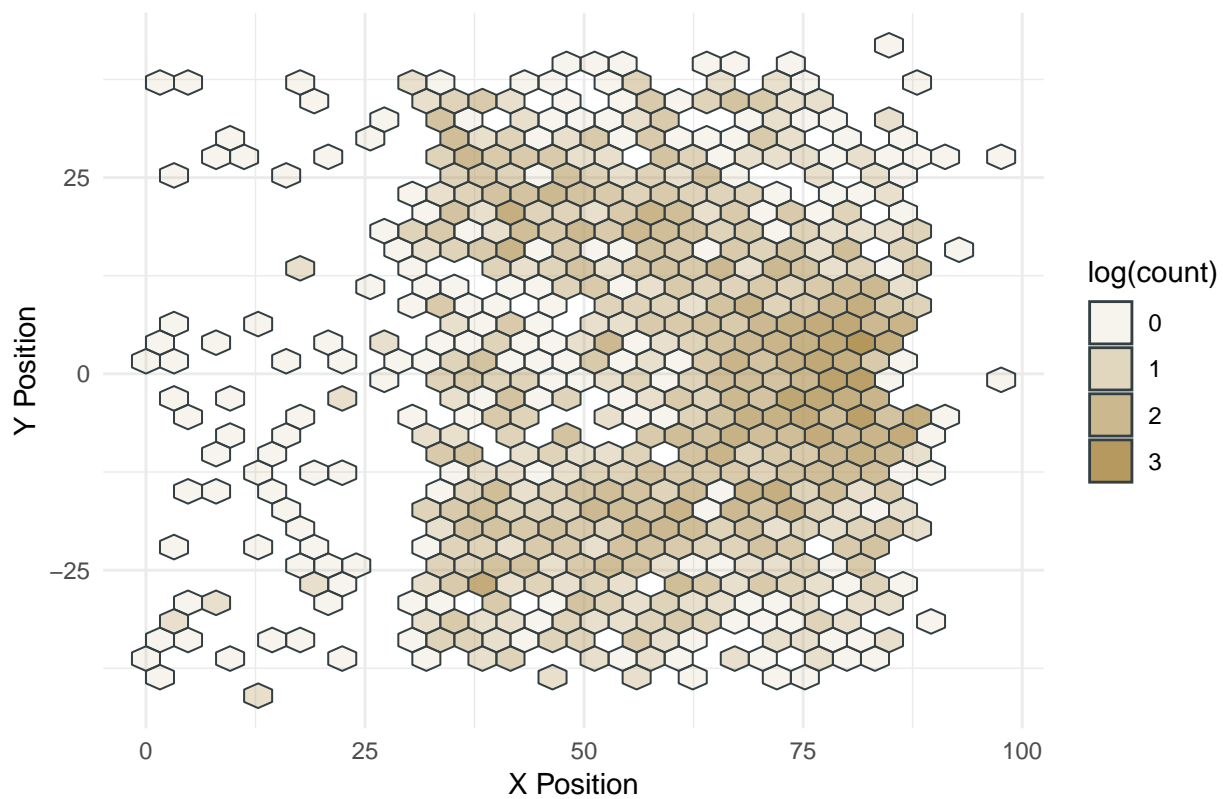
```
fleury.2016.locations.plot = graph.shot.locations(fleury.2016, "#000000", "#FCB514", "Marc-Andre Fleury")
fleury.2016.locations.plot
```

Marc-Andre Fleury (2016) Shot Locations



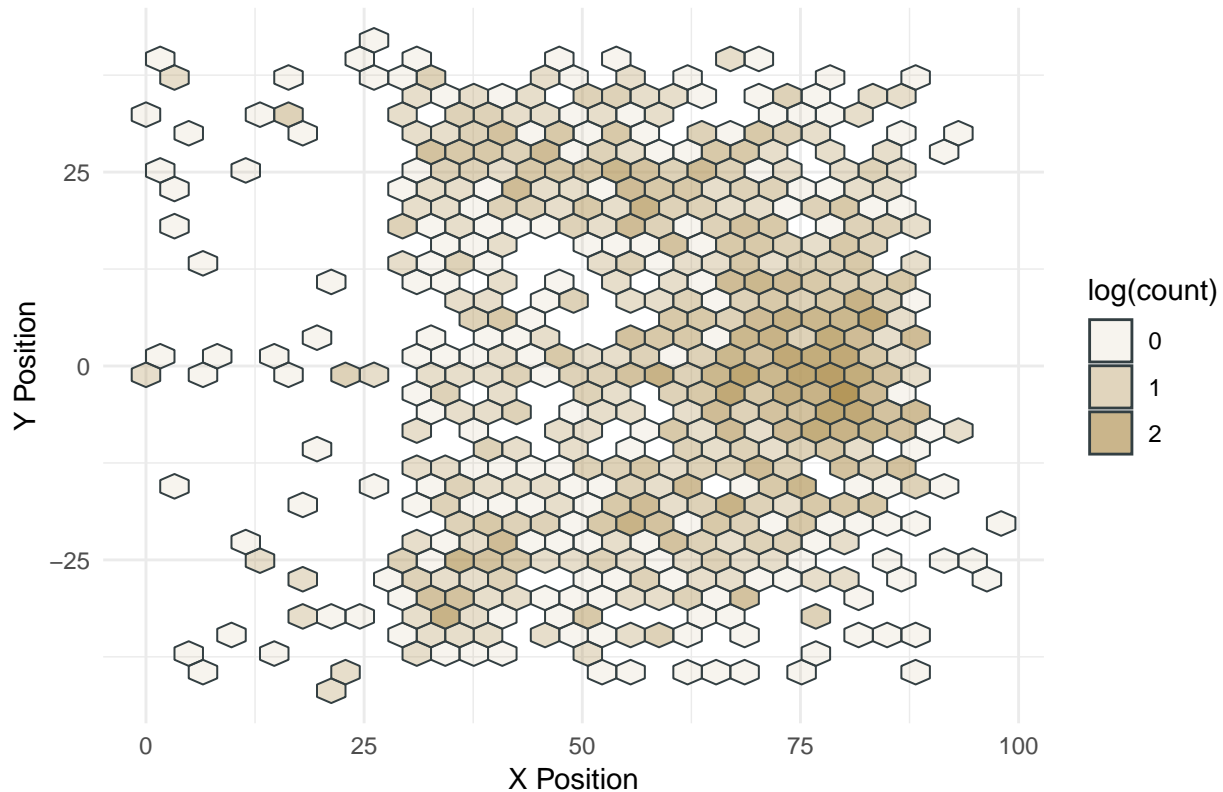
```
fleury.2017.locations.plot = graph.shot.locations(fleury.2017, "#B4975A", "#333F42", "Marc-Andre Fleury")
fleury.2017.locations.plot
```

Marc-Andre Fleury (2017) Shot Locations



```
fleury.2018.locations.plot = graph.shot.locations(fleury.2018, "#B4975A", "#333F42", "Marc-Andre Fleury")
fleury.2018.locations.plot
```

Marc-Andre Fleury (2018) Shot Locations



Fleury has shown exceptional consistency. For any young goaltender breaking into the league, he should be a top-role model.

Braden Holtby

After exercising his playoff demons during the 2018 Stanley Cup Playoffs, Braden Holtby has lived up to his expectations. His play has been phenomenal during both the regular season and playoffs. Let's see how he's been since 2015.

```
holtby.2015 = get.goalie.data(analysis.2015, "Braden Holtby")
holtby.2016 = get.goalie.data(analysis.2016, "Braden Holtby")
holtby.2017 = get.goalie.data(analysis.2017, "Braden Holtby")
holtby.2018 = get.goalie.data(analysis.2018, "Braden Holtby")
```

Let's start by calculating some of his stats for each season and then tabularizing.

```
holtby.2015.sp = get.save.percent(holtby.2015)
holtby.2016.sp = get.save.percent(holtby.2016)
holtby.2017.sp = get.save.percent(holtby.2017)
holtby.2018.sp = get.save.percent(holtby.2018)

holtby.2015.gp = get.goal.percent(holtby.2015)
holtby.2016.gp = get.goal.percent(holtby.2016)
holtby.2017.gp = get.goal.percent(holtby.2017)
holtby.2018.gp = get.goal.percent(holtby.2018)

holtby.2015.spg = get.shots.per.goal(holtby.2015)
```

```
holtby.2016.spg = get.shots.per.goal(holtby.2016)
holtby.2017.spg = get.shots.per.goal(holtby.2017)
holtby.2018.spg = get.shots.per.goal(holtby.2018)
```

Now, let's put it in a table:

| Season | Save Percent | Goal Percent | Shots Per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.9448141 | 0.0551859 | 18.1205674 |
| 2016-2017 | 0.9476936 | 0.0523064 | 19.1181102 |
| 2017-2018 | 0.9348104 | 0.0651896 | 15.3398693 |
| 2018-2019 | 0.9345048 | 0.0654952 | 15.2682927 |

We can look at his game by game data also.

```
holtby.2015.games.sp = get.game.save.percent(holtby.2015)
holtby.2015.games.gp = get.game.goal.percent(holtby.2015)
holtby.2015.games.spg = get.game.shots.per.goal(holtby.2015)

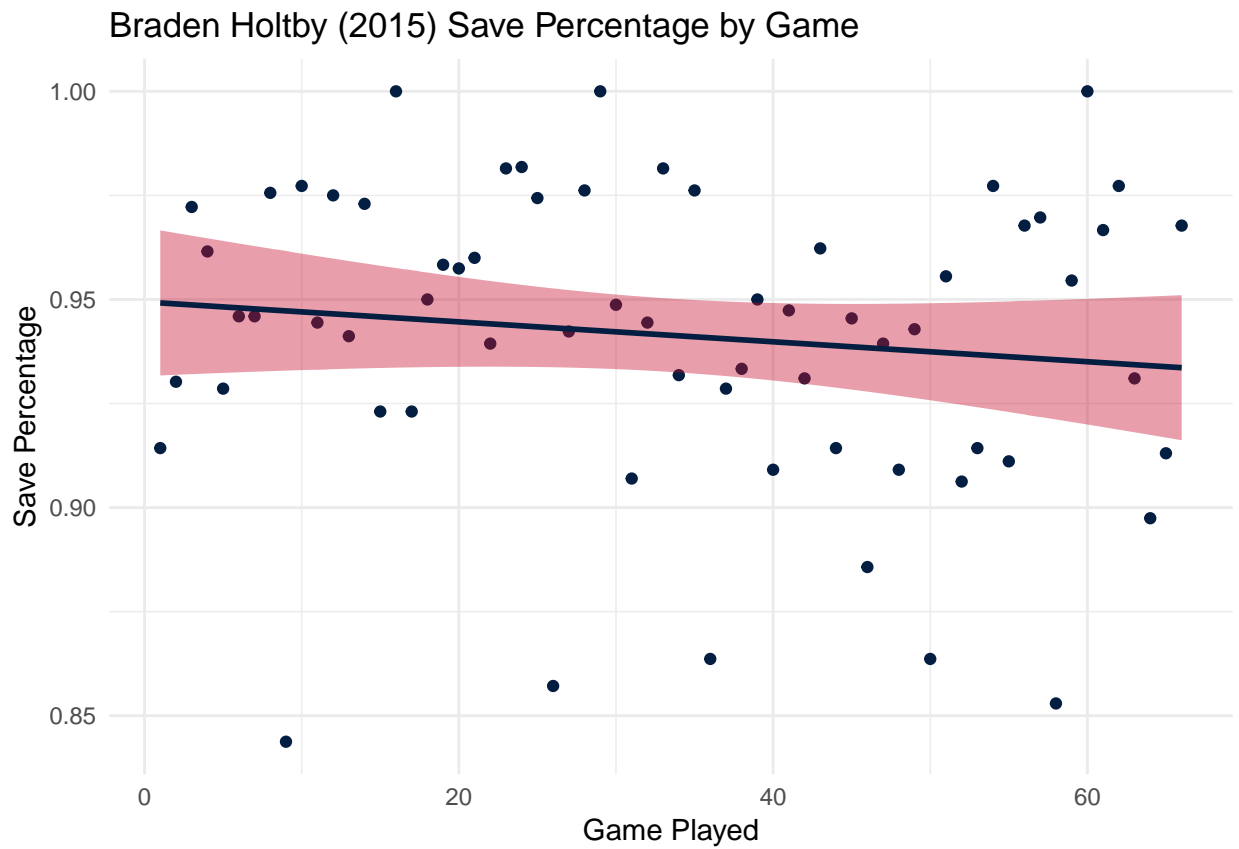
holtby.2016.games.sp = get.game.save.percent(holtby.2016)
holtby.2016.games.gp = get.game.goal.percent(holtby.2016)
holtby.2016.games.spg = get.game.shots.per.goal(holtby.2016)

holtby.2017.games.sp = get.game.save.percent(holtby.2017)
holtby.2017.games.gp = get.game.goal.percent(holtby.2017)
holtby.2017.games.spg = get.game.shots.per.goal(holtby.2017)

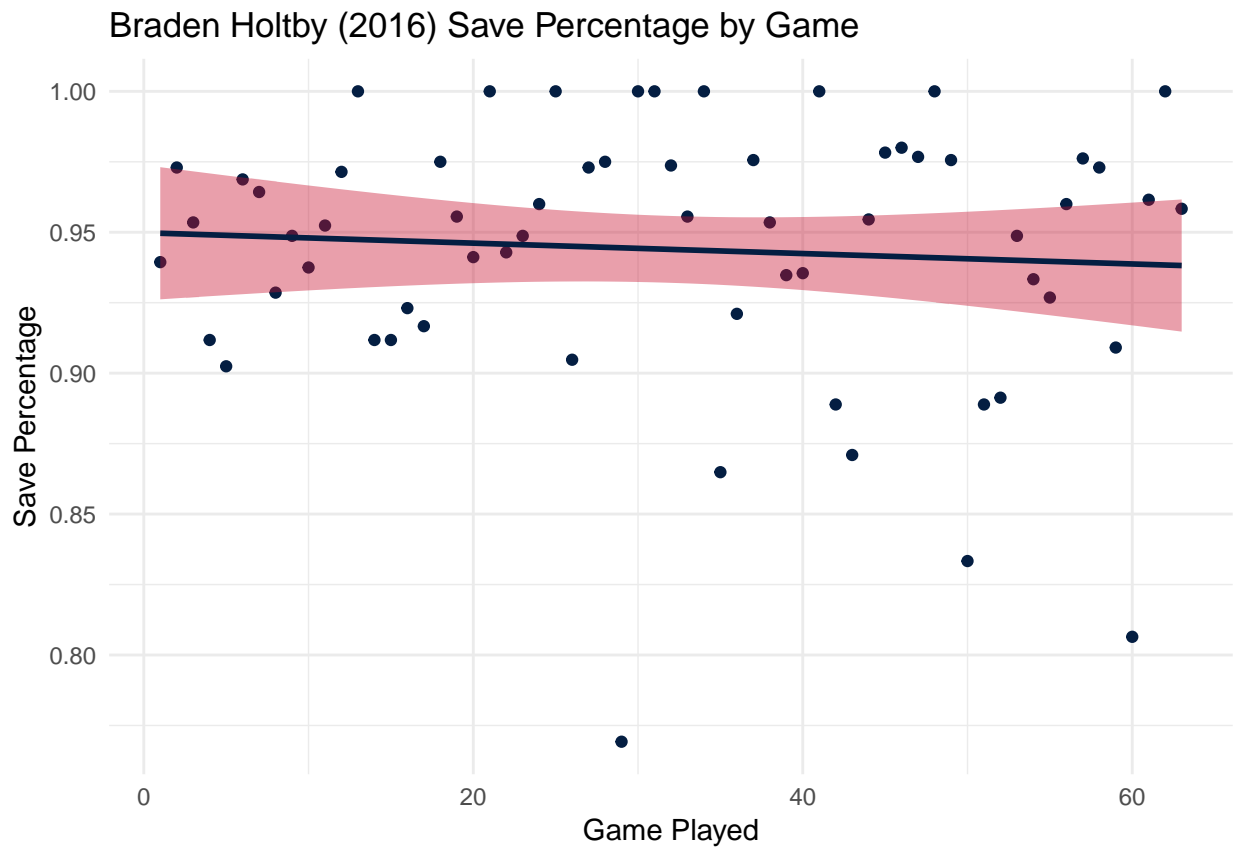
holtby.2018.games.sp = get.game.save.percent(holtby.2018)
holtby.2018.games.gp = get.game.goal.percent(holtby.2018)
holtby.2018.games.spg = get.game.shots.per.goal(holtby.2018)
```

Let's look at his save percentage graphs:

```
holtby.2015.sp.plot = graph.trend(holtby.2015.games.sp, "Save Percentage", "#041E42", "#C8102E", "Braden")
holtby.2015.sp.plot
```

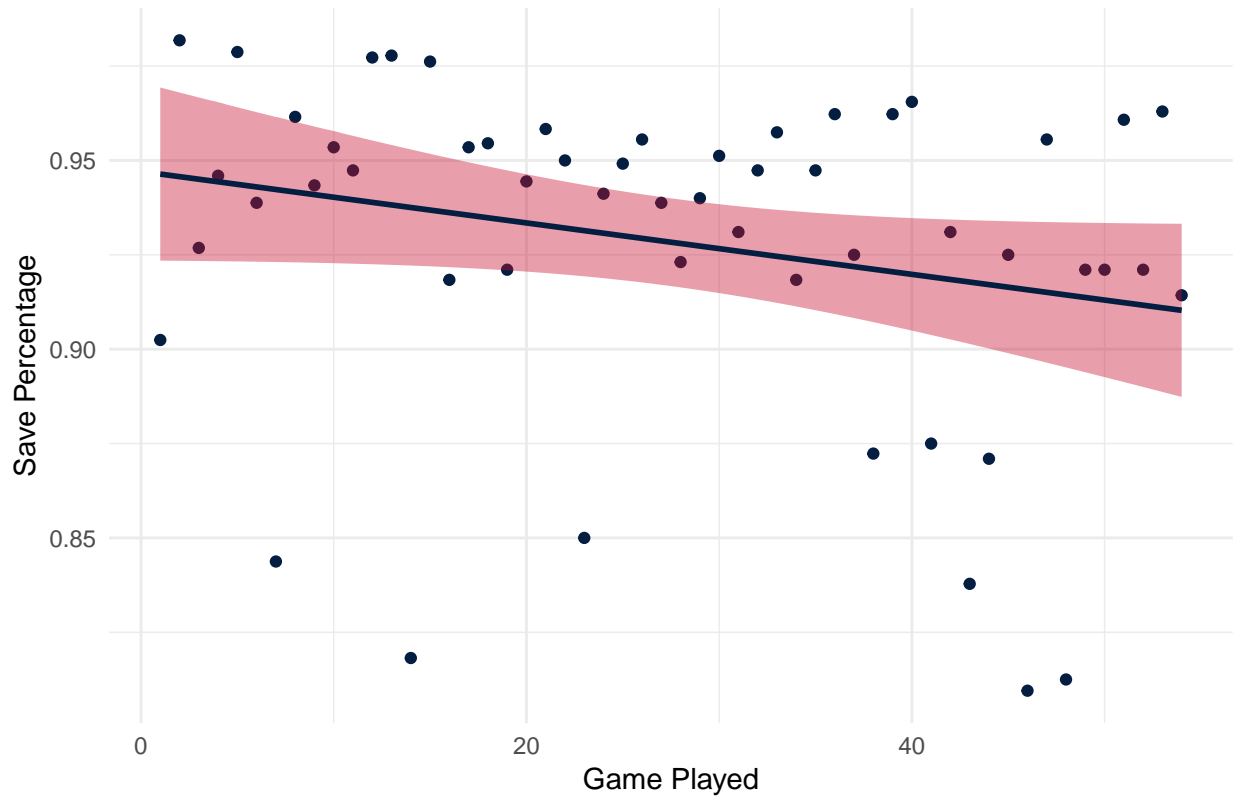


```
holtby.2016.sp.plot = graph.trend(holtby.2016.games.sp, "Save Percentage", "#041E42", "#C8102E", "Braden Holtby")
holtby.2016.sp.plot
```

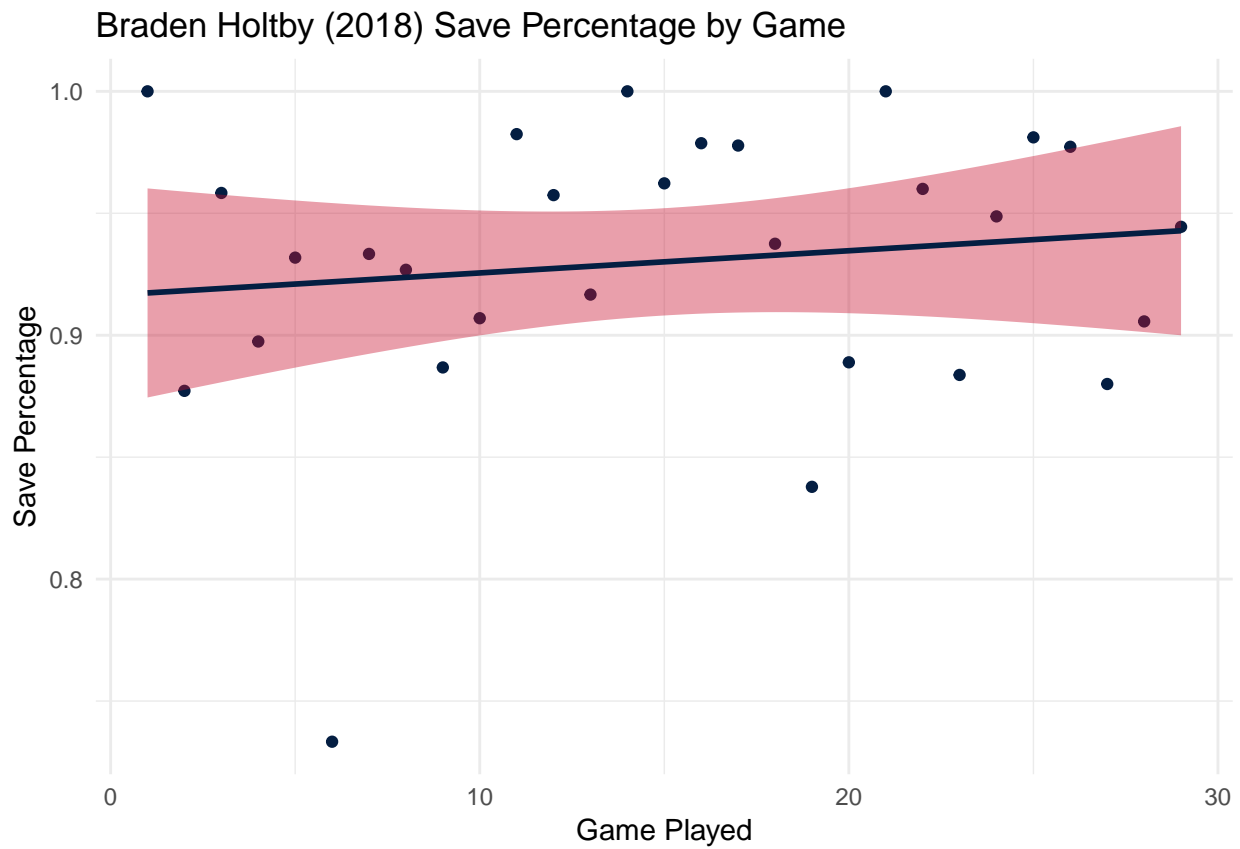



```
holtby.2017.sp.plot = graph.trend(holtby.2017.games.sp, "Save Percentage", "#041E42", "#C8102E", "Braden Holtby")
holtby.2017.sp.plot
```

Braden Holtby (2017) Save Percentage by Game



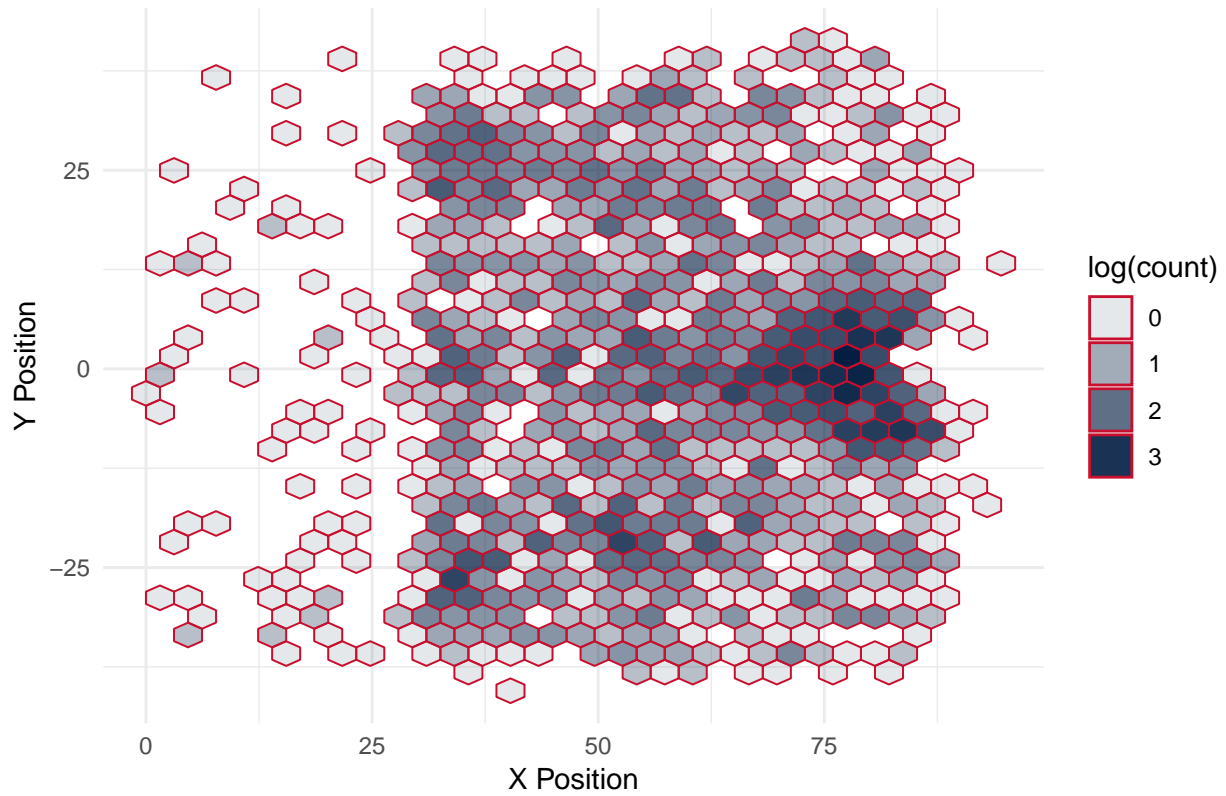
```
holtby.2018.sp.plot = graph.trend(holtby.2018.games.sp, "Save Percentage", "#041E42", "#C8102E", "Braden  
holtby.2018.sp.plot
```



And now, let's take a look at his shot location data.

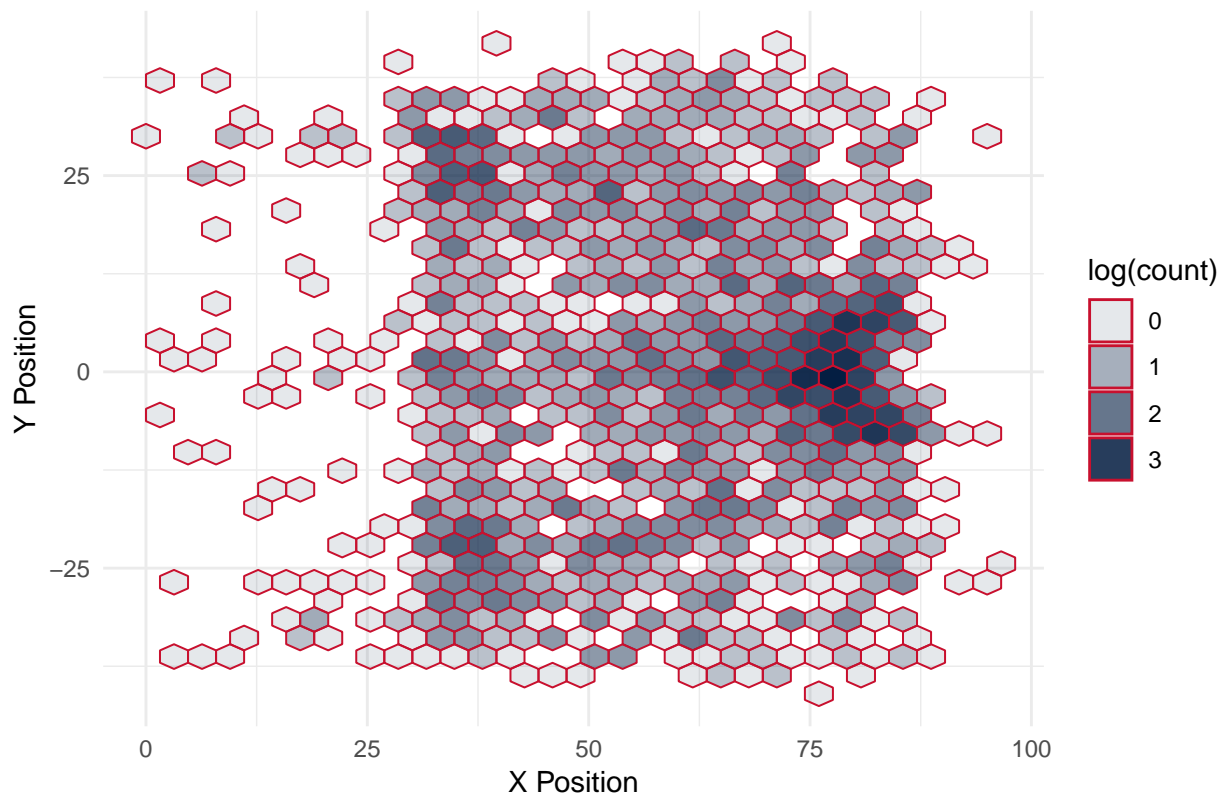
```
holtby.2015.locations.plot = graph.shot.locations(holtby.2015, "#041E42", "#C8102E", "Braden Holtby (2015)")
holtby.2015.locations.plot
```

Braden Holtby (2015) Shot Locations



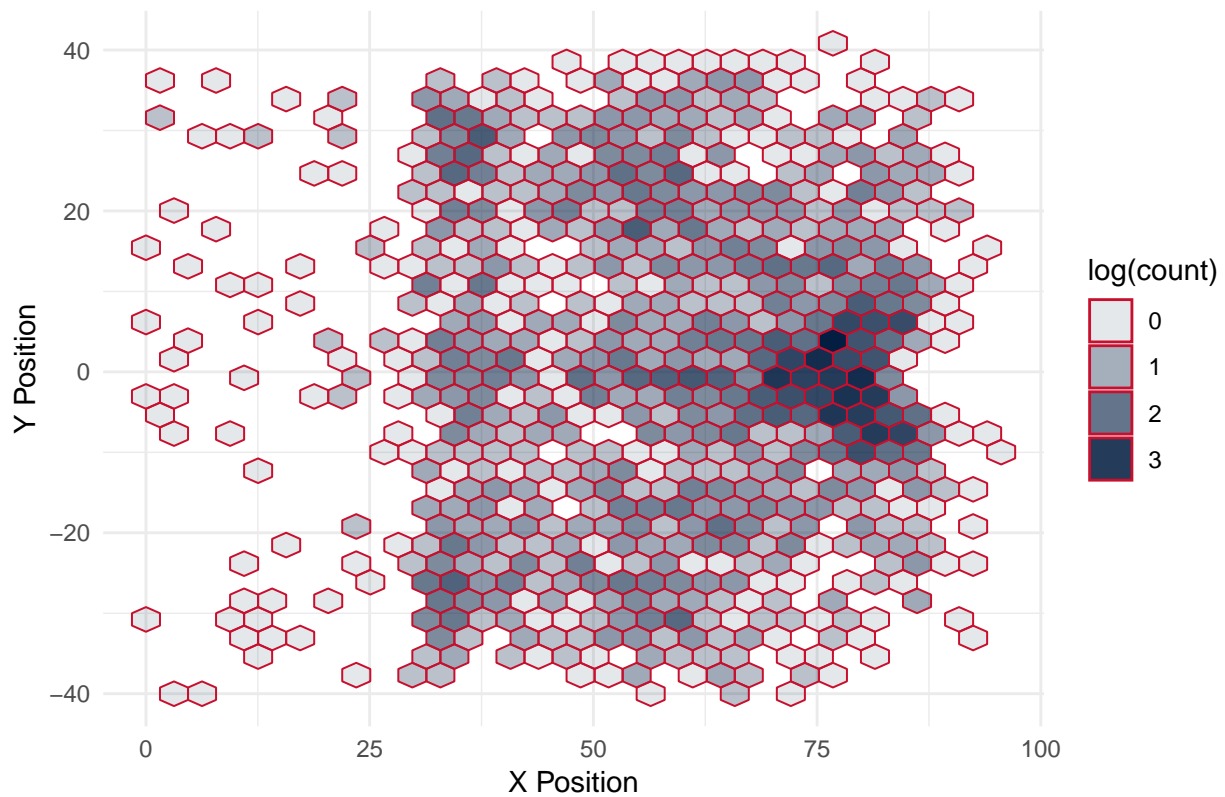
```
holtby.2016.locations.plot = graph.shot.locations(holtby.2016, "#041E42", "#C8102E", "Braden Holtby (2015) Shot Locations")
holtby.2016.locations.plot
```

Braden Holtby (2016) Shot Locations



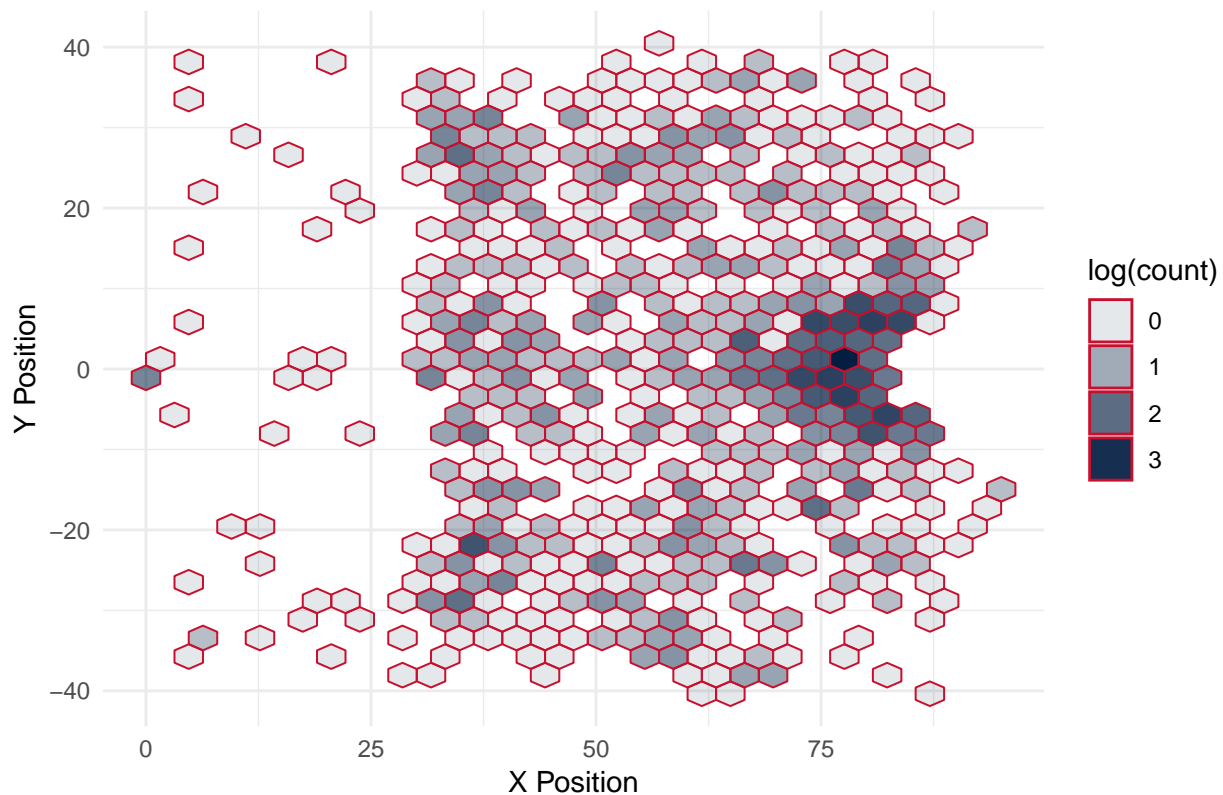
```
holtby.2017.locations.plot = graph.shot.locations(holtby.2017, "#041E42", "#C8102E", "Braden Holtby (2016)")
holtby.2017.locations.plot
```

Braden Holtby (2017) Shot Locations



```
holtby.2018.locations.plot = graph.shot.locations(holtby.2018, "#041E42", "#C8102E", "Braden Holtby (2017) Shot Locations")
holtby.2018.locations.plot
```

Braden Holtby (2018) Shot Locations



Holtby has had the trend of playing poorly near the end of the season. However, it seems like Holtby is currently going against his negative trend this season. His main weakness seems to be shots from the left circle. This is probably aimed towards his glove hand, which is significantly weaker than his positional play.

John Gibson

Hailing from Pittsburgh Pennsylvania, John Gibson is the 25-year-old starter for the Anaheim Ducks. In his 6 years in the NHL, Gibson has played a total of 221 games, starting 213 games. In his career, Gibson boasts a 2.37 GAA and 92.2% save percentage. Let's start by looking at his growth per season.

```
gibson.2015 = get.goalie.data(analysis.2015, "John Gibson")
gibson.2016 = get.goalie.data(analysis.2016, "John Gibson")
gibson.2017 = get.goalie.data(analysis.2017, "John Gibson")
gibson.2018 = get.goalie.data(analysis.2018, "John Gibson")
```

Let's start by calculating some of his stats for each season and then tabularizing.

```
gibson.2015.sp = get.save.percent(gibson.2015)
gibson.2016.sp = get.save.percent(gibson.2016)
gibson.2017.sp = get.save.percent(gibson.2017)
gibson.2018.sp = get.save.percent(gibson.2018)

gibson.2015.gp = get.goal.percent(gibson.2015)
gibson.2016.gp = get.goal.percent(gibson.2016)
gibson.2017.gp = get.goal.percent(gibson.2017)
gibson.2018.gp = get.goal.percent(gibson.2018)
```

```
gibson.2015.spg = get.shots.per.goal(gibson.2015)
gibson.2016.spg = get.shots.per.goal(gibson.2016)
gibson.2017.spg = get.shots.per.goal(gibson.2017)
gibson.2018.spg = get.shots.per.goal(gibson.2018)
```

Now, let's put it in a table:

| Season | Save Percent | Goal Percent | Shots Per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.9439319 | 0.0560681 | 17.835443 |
| 2016-2017 | 0.9456901 | 0.0543099 | 18.412844 |
| 2017-2018 | 0.948018 | 0.051982 | 19.2374101 |
| 2018-2019 | 0.9442356 | 0.0557644 | 17.9325843 |

We can look at his game by game data also.

```
gibson.2015.games.sp = get.game.save.percent(gibson.2015)
gibson.2015.games.gp = get.game.goal.percent(gibson.2015)
gibson.2015.games.spg = get.game.shots.per.goal(gibson.2015)

gibson.2016.games.sp = get.game.save.percent(gibson.2016)
gibson.2016.games.gp = get.game.goal.percent(gibson.2016)
gibson.2016.games.spg = get.game.shots.per.goal(gibson.2016)

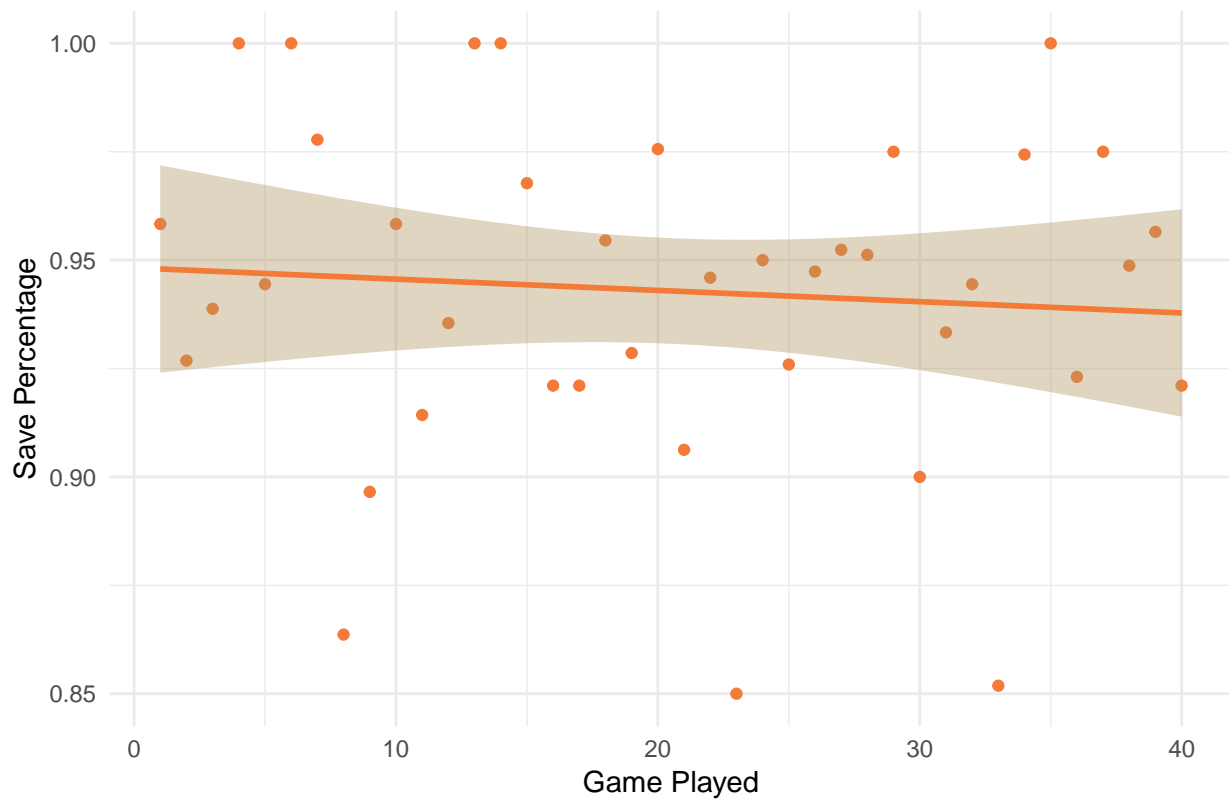
gibson.2017.games.sp = get.game.save.percent(gibson.2017)
gibson.2017.games.gp = get.game.goal.percent(gibson.2017)
gibson.2017.games.spg = get.game.shots.per.goal(gibson.2017)

gibson.2018.games.sp = get.game.save.percent(gibson.2018)
gibson.2018.games.gp = get.game.goal.percent(gibson.2018)
gibson.2018.games.spg = get.game.shots.per.goal(gibson.2018)
```

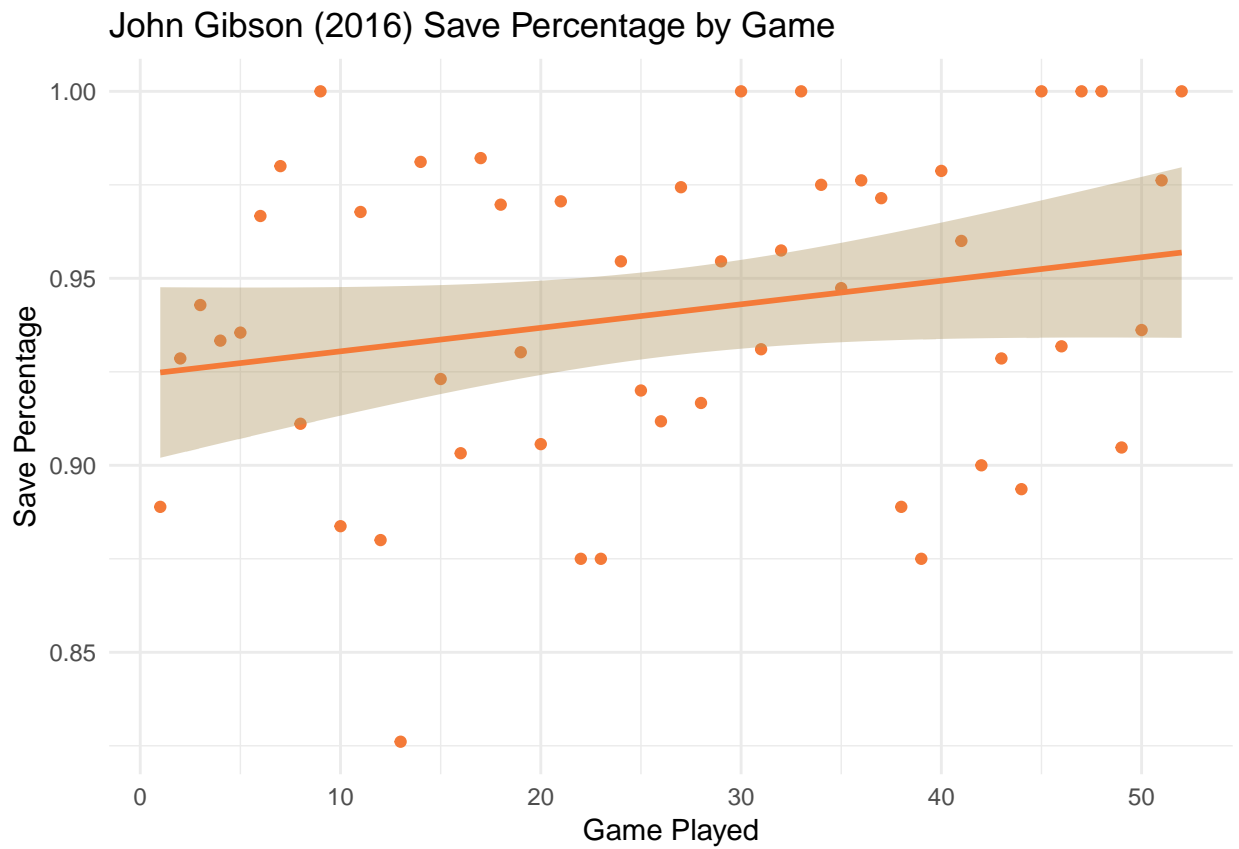
Let's look at his save percentage graphs:

```
gibson.2015.sp.plot = graph.trend(gibson.2015.games.sp, "Save Percentage", "#F47A38", "#B09862", "John C. Gibson")
gibson.2015.sp.plot
```


John Gibson (2015) Save Percentage by Game

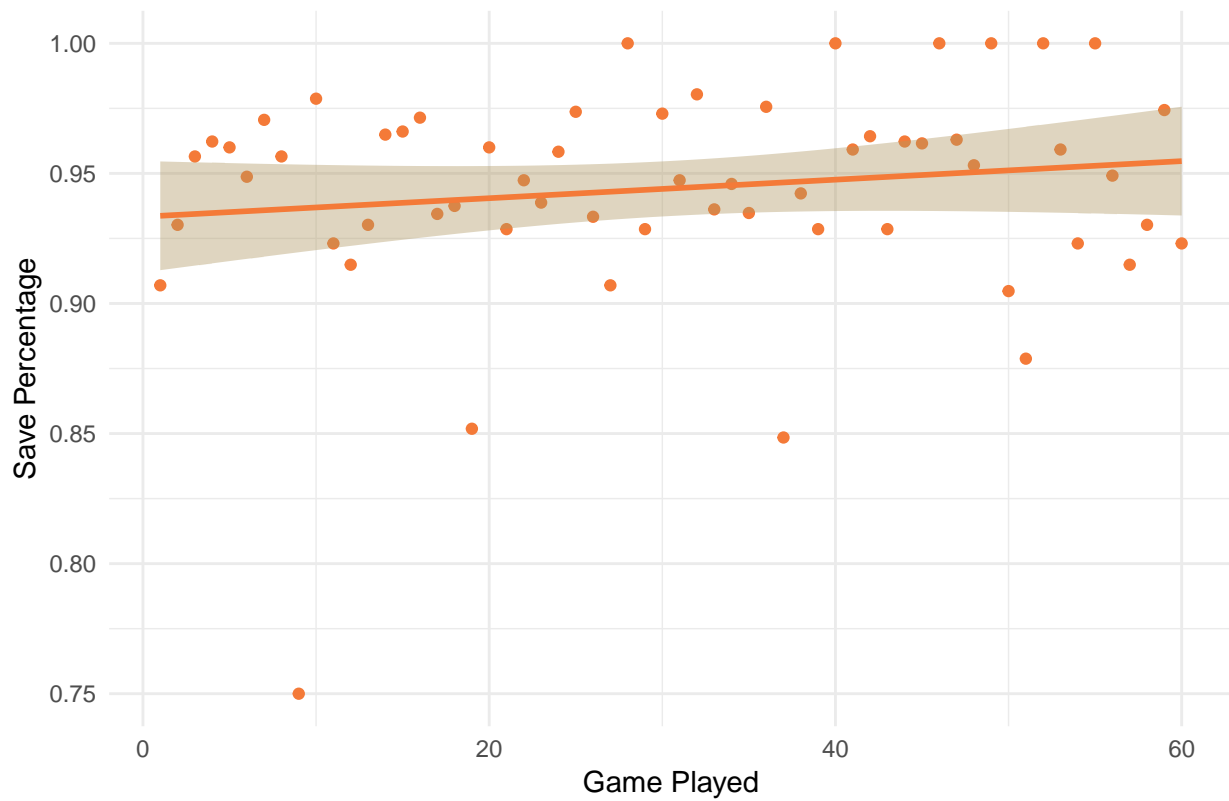


```
gibson.2016.sp.plot = graph.trend(gibson.2016.games.sp, "Save Percentage", "#F47A38", "#B09862", "John Gibson", "2015")
gibson.2016.sp.plot
```

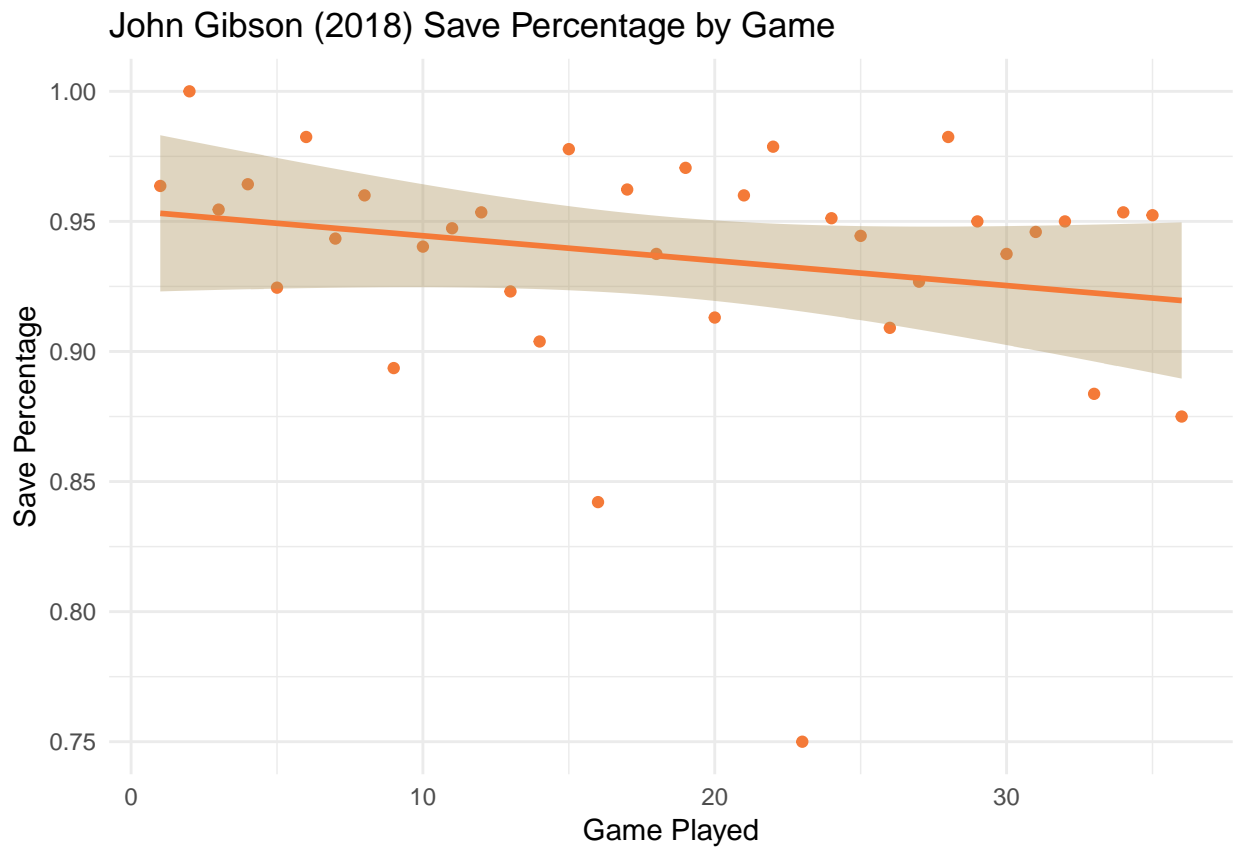


```
gibson.2017.sp.plot = graph.trend(gibson.2017.games.sp, "Save Percentage", "#F47A38", "#B09862", "John Gibson", "2016")
gibson.2017.sp.plot
```

John Gibson (2017) Save Percentage by Game



```
gibson.2018.sp.plot = graph.trend(gibson.2018.games.sp, "Save Percentage", "#F47A38", "#B09862", "John Gibson", "2017")
gibson.2018.sp.plot
```

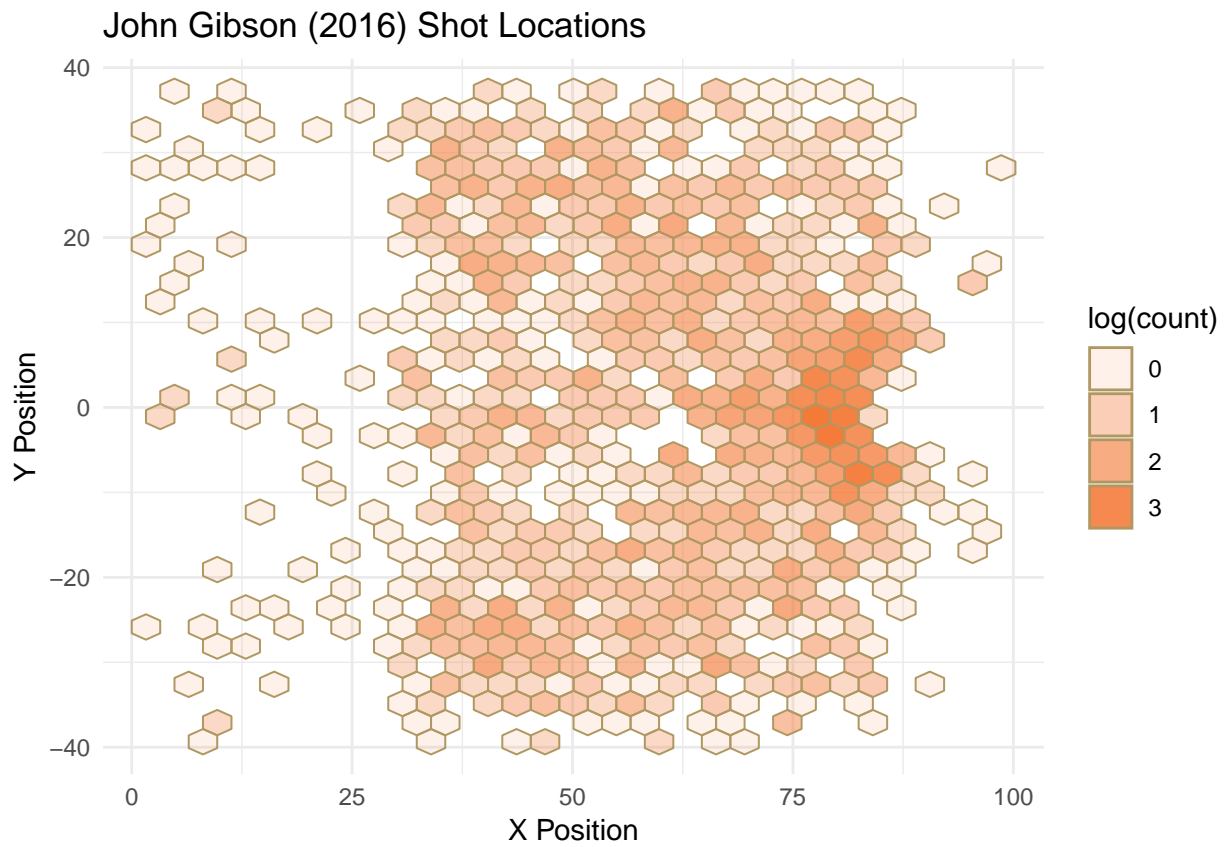


And now, let's take a look at his shot location data.

```
gibson.2015.locations.plot = graph.shot.locations(gibson.2015, "#F47A38", "#B09862", "John Gibson (2015)", "#B09862")
gibson.2015.locations.plot
```

A hexagonal bin plot showing the spatial distribution of log(count) values. The x-axis is 'X Position' (0 to 100) and the y-axis is 'Y Position' (-40 to 40). The plot shows a dense cluster of hexagons in the center, with colors ranging from light orange (log(count) = 0) to dark orange (log(count) = 3). A legend on the right indicates the color scale for log(count).

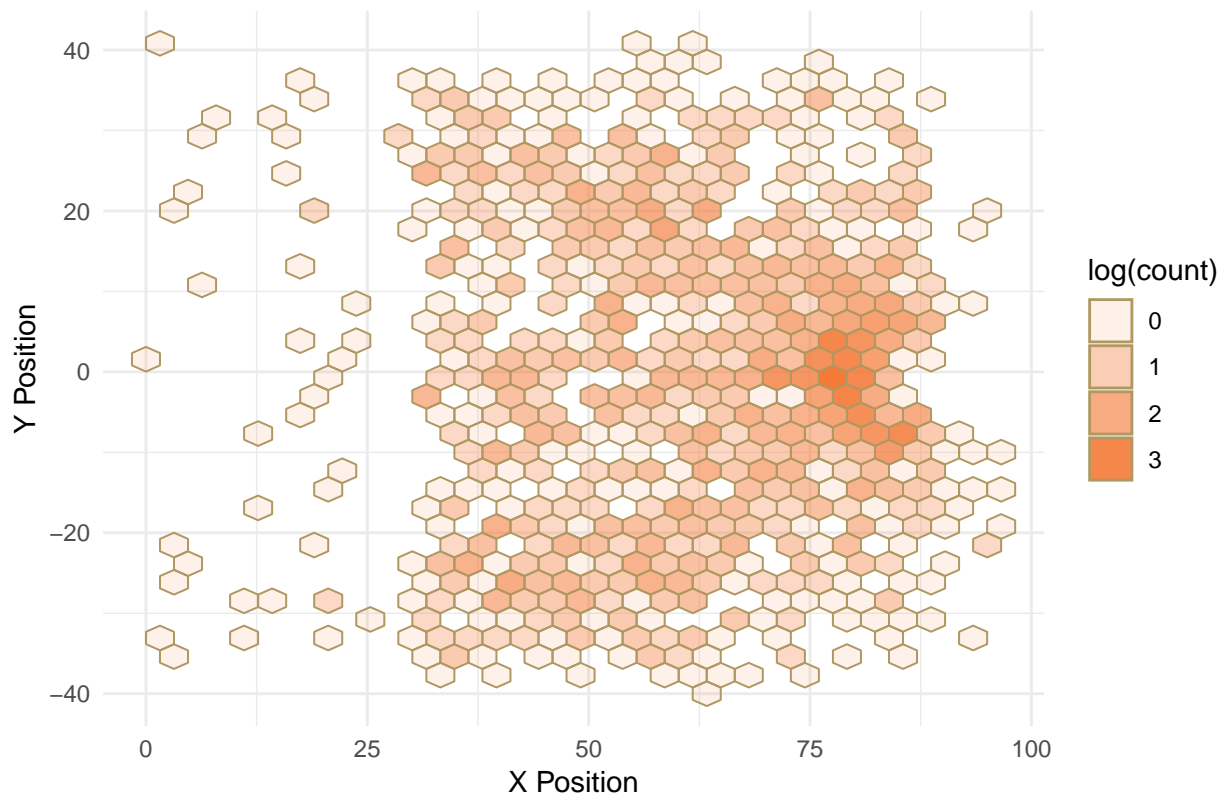
```
gibson.2016.locations.plot = graph.shot.locations(gibson.2016, "#F47A38", "#B09862", "John Gibson (2016)")
gibson.2016.locations.plot
```



```
gibson.2017.locations.plot = graph.shot.locations(gibson.2017, "#F47A38", "#B09862", "John Gibson (2017)", "#F47A38", "#B09862")
gibson.2017.locations.plot
```

```
gibson.2018.locations.plot = graph.shot.locations(gibson.2018, "#F47A38", "#B09862", "John")
gibson.2018.locations.plot
```

John Gibson (2018) Shot Locations



Skaters

Sidney Crosby

Already regarded as being in the top 5 all-time greatest NHL players, Sidney Crosby has been playing like an MVP for every single season he's been in the league. Let's look at his past few seasons in more detail.

```
crosby.2015 = get.skater.data(analysis.2015, "Sidney Crosby")
crosby.2016 = get.skater.data(analysis.2016, "Sidney Crosby")
crosby.2017 = get.skater.data(analysis.2017, "Sidney Crosby")
crosby.2018 = get.skater.data(analysis.2018, "Sidney Crosby")
```

Let's get some of his stats.

```
crosby.2015.sp = get.save.percent(crosby.2015)
crosby.2015.game.sp = get.game.save.percent(crosby.2015)
crosby.2016.sp = get.save.percent(crosby.2016)
crosby.2016.game.sp = get.game.save.percent(crosby.2016)
crosby.2017.sp = get.save.percent(crosby.2017)
crosby.2017.game.sp = get.game.save.percent(crosby.2017)
crosby.2018.sp = get.save.percent(crosby.2018)
crosby.2018.game.sp = get.game.save.percent(crosby.2018)

crosby.2015.gp = get.goal.percent(crosby.2015)
crosby.2015.game.gp = get.game.goal.percent(crosby.2015)
crosby.2016.gp = get.goal.percent(crosby.2016)
crosby.2016.game.gp = get.game.goal.percent(crosby.2016)
```



```

crosby.2017.gp = get.goal.percent(crosby.2017)
crosby.2017.game.gp = get.game.goal.percent(crosby.2017)
crosby.2018.gp = get.goal.percent(crosby.2018)
crosby.2018.game.gp = get.game.goal.percent(crosby.2018)

crosby.2015.spg = get.shots.per.goal(crosby.2015)
crosby.2015.game.spg = get.game.shots.per.goal(crosby.2015)
crosby.2016.spg = get.shots.per.goal(crosby.2016)
crosby.2016.game.spg = get.game.shots.per.goal(crosby.2016)
crosby.2017.spg = get.shots.per.goal(crosby.2017)
crosby.2017.game.spg = get.game.shots.per.goal(crosby.2017)
crosby.2018.spg = get.shots.per.goal(crosby.2018)
crosby.2018.game.spg = get.game.shots.per.goal(crosby.2018)

```

Here's a table of his data.

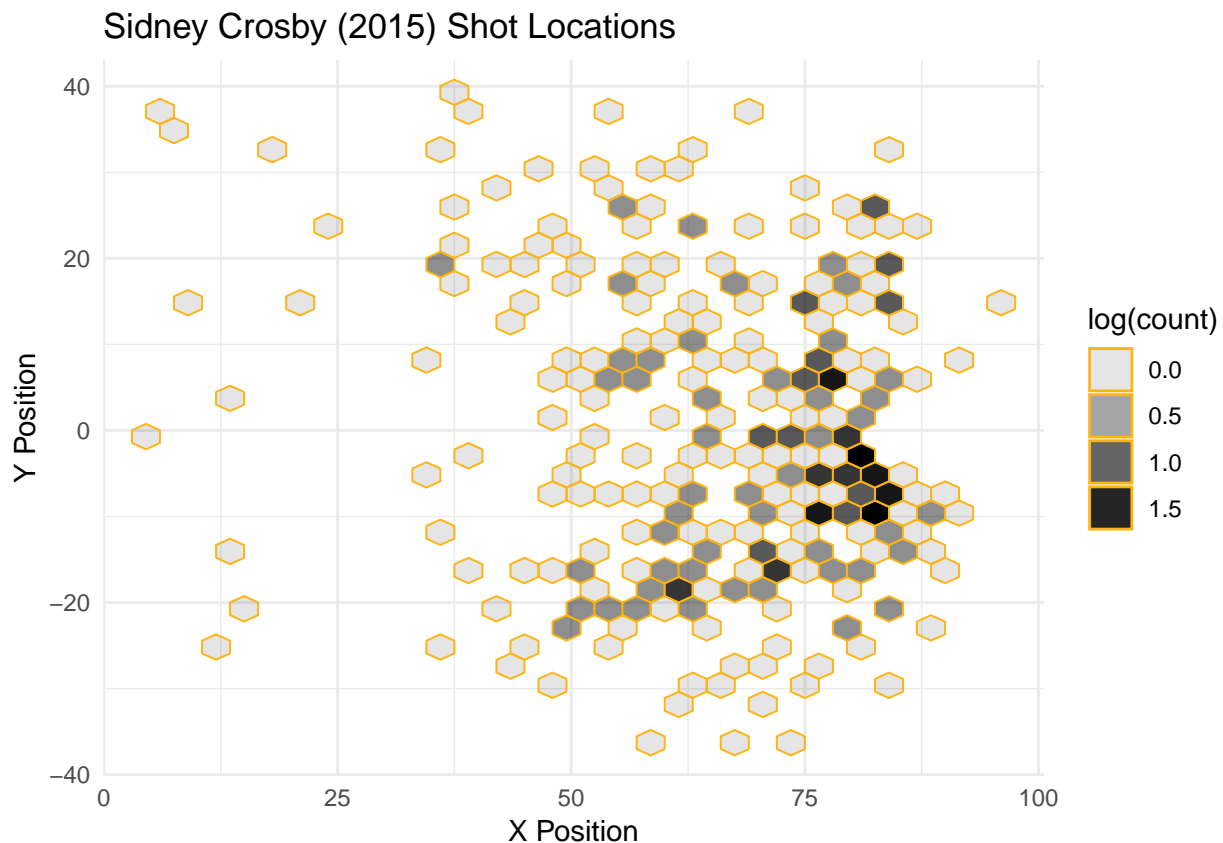
| Season | Save Percent | Goal Percent | Shots per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.8947368 | 0.1052632 | 9.5 |
| 2016-2017 | 0.8646154 | 0.1353846 | 7.3863636 |
| 2017-2018 | 0.914956 | 0.085044 | 11.7586207 |
| 2018-2019 | 0.8689655 | 0.1310345 | 7.6315789 |

Let's take a look at his heatmaps.

```

crosby.2015.locations.plot = graph.shot.locations(crosby.2015, "#000000", "#FCB514", "Sidney Crosby (2015)", "#000000")
crosby.2015.locations.plot

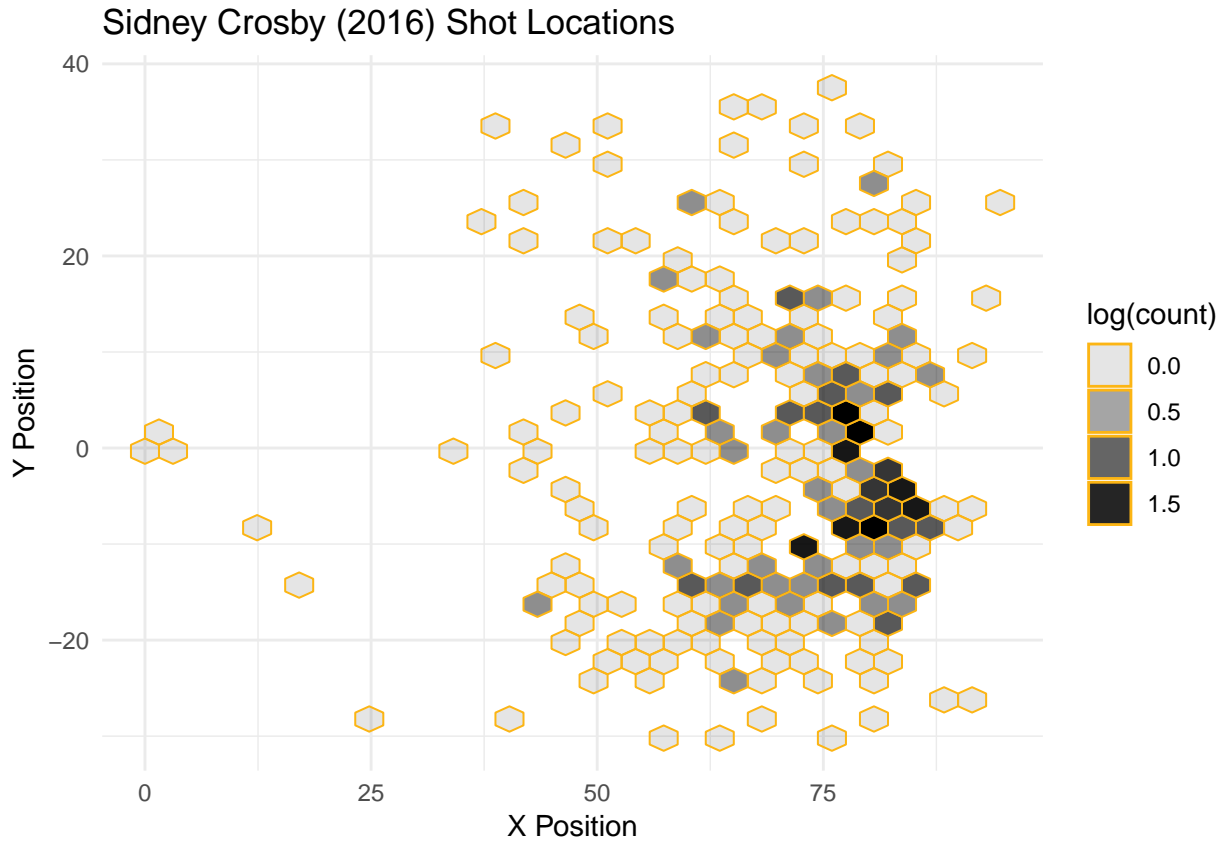
```



```

crosby.2016.locations.plot = graph.shot.locations(crosby.2016, "#000000", "#FCB514", "Sidney Crosby (2016)");
crosby.2016.locations.plot

```

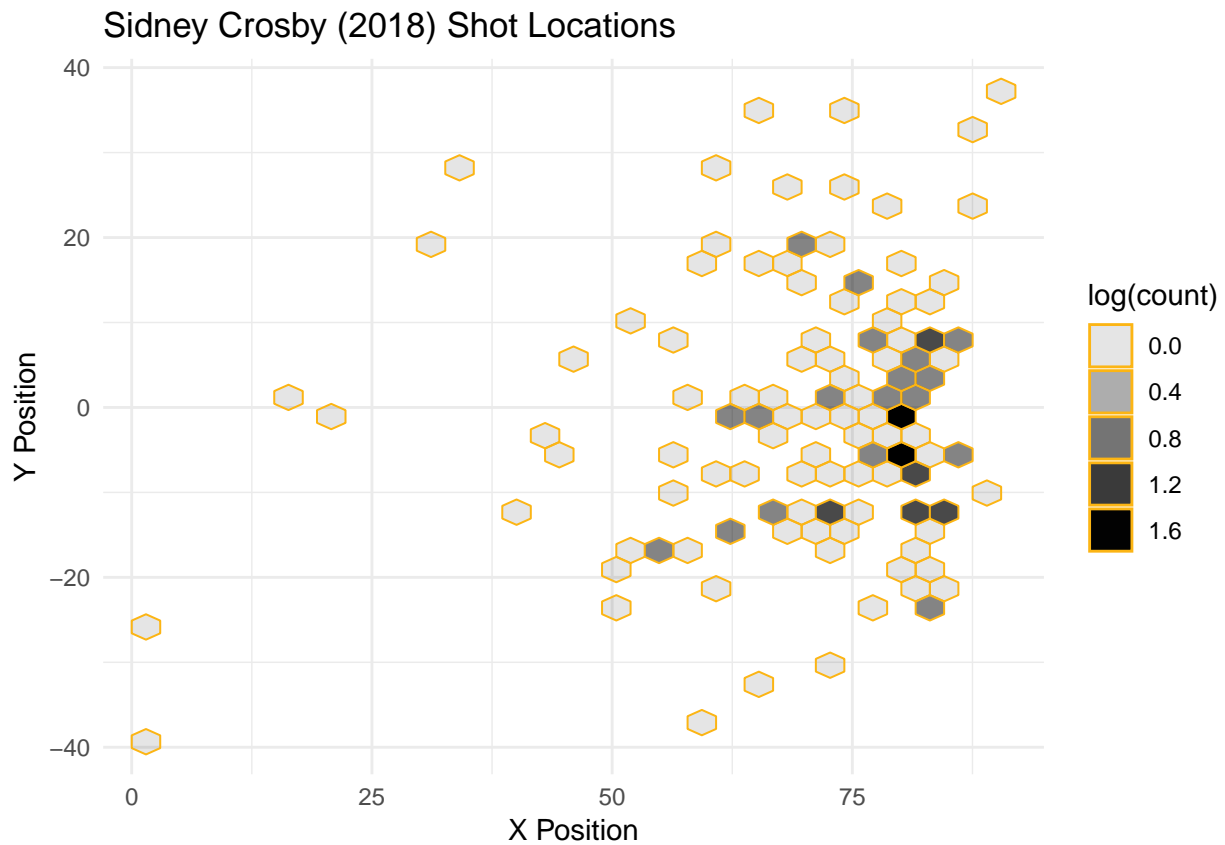


```
crosby.2017.locations.plot = graph.shot.locations(crosby.2017, "#000000", "#FCB514", "Sidney Crosby (2017)")
crosby.2017.locations.plot
```

Sidney Crosby (2017) Shot Locations



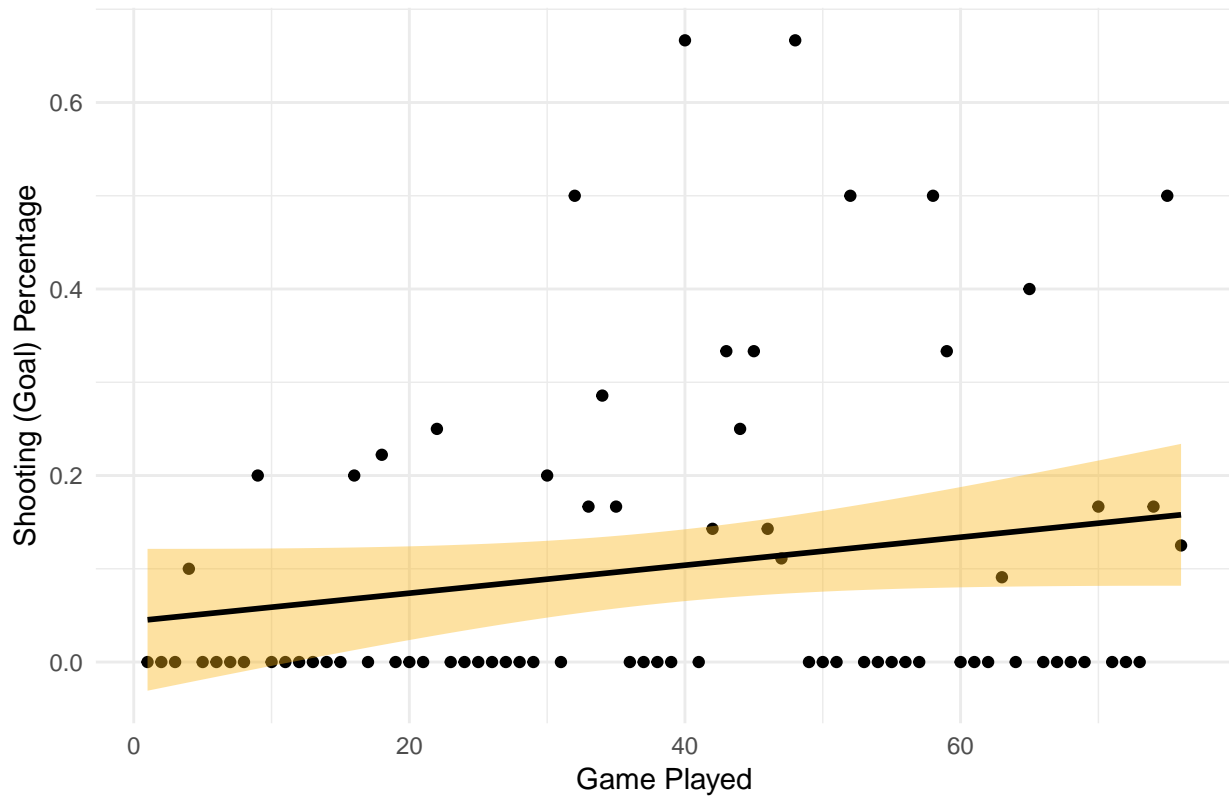
```
crosby.2018.locations.plot = graph.shot.locations(crosby.2018, "#000000", "#FCB514", "Sidney Crosby (2018)")
crosby.2018.locations.plot
```



From this, we can see that Sidney Crosby has a lot of success around the net. Let's look at this goal percentage per game now.

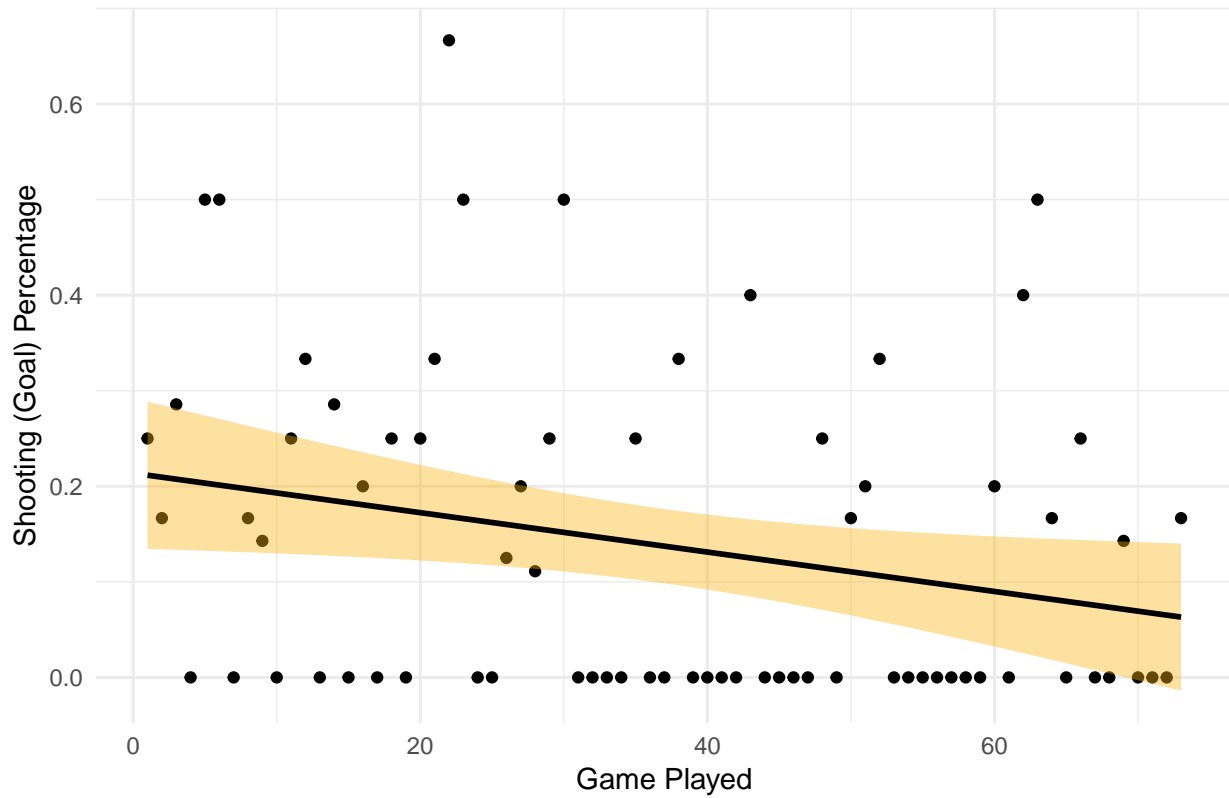
```
crosby.2015.gp.plot = graph.trend(crosby.2015.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB51D", "#000000")
crosby.2015.gp.plot
```

Sidney Crosby (2015) Shooting (Goal) Percentage by Game



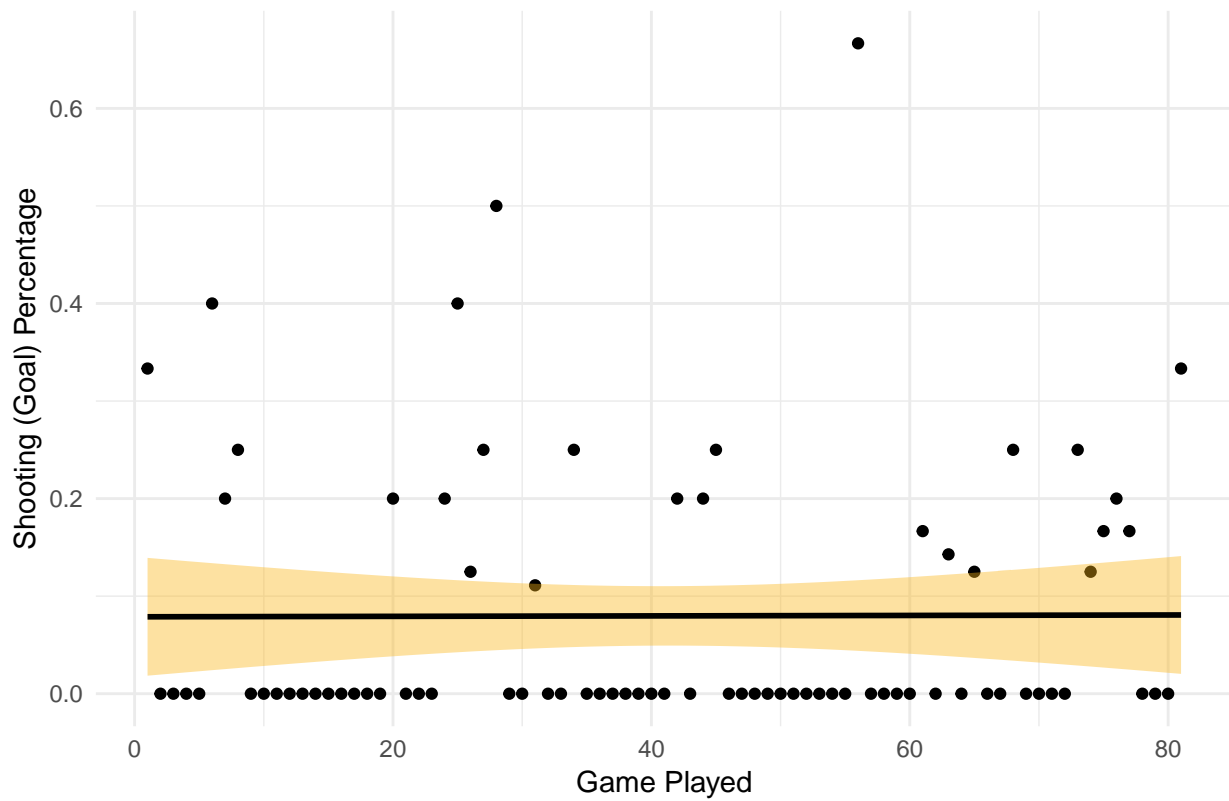
```
crosby.2016.gp.plot = graph.trend(crosby.2016.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB511", "#000000")
crosby.2016.gp.plot
```

Sidney Crosby (2016) Shooting (Goal) Percentage by Game

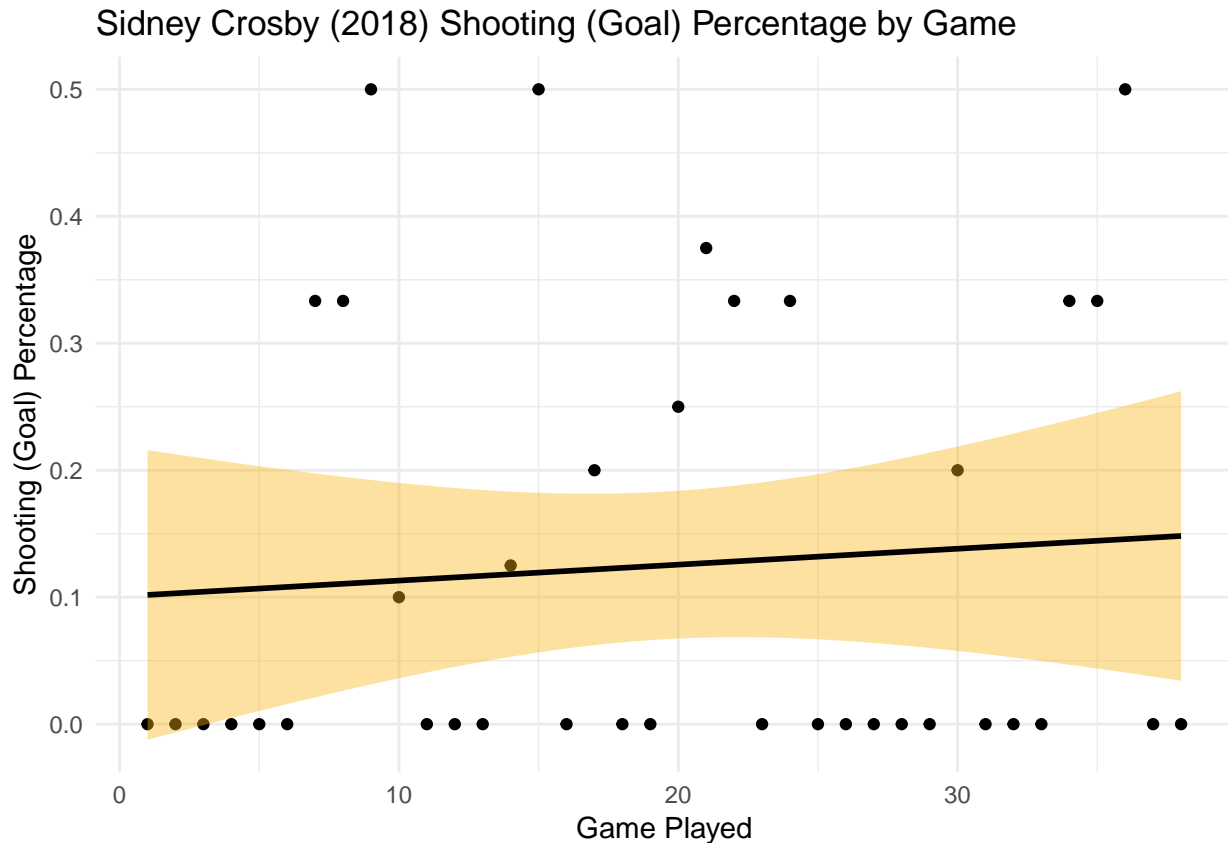


```
crosby.2017.gp.plot = graph.trend(crosby.2017.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB511", "#000000")
crosby.2017.gp.plot
```

Sidney Crosby (2017) Shooting (Goal) Percentage by Game



```
crosby.2018.gp.plot = graph.trend(crosby.2018.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB511", "#000000")
crosby.2018.gp.plot
```



Evgeni Malkin

A clear elite first line center, Evgeni Malkin has played behind Sidney Crosby his entire career. However, instead of fussing out this, he puts up first line center numbers every year. Let's see how he's done it his previous few years.

```
malkin.2015 = get.skater.data(analysis.2015, "Evgeni Malkin")
malkin.2016 = get.skater.data(analysis.2016, "Evgeni Malkin")
malkin.2017 = get.skater.data(analysis.2017, "Evgeni Malkin")
malkin.2018 = get.skater.data(analysis.2018, "Evgeni Malkin")
```

Let's get some of his stats.

```
malkin.2015.sp = get.save.percent(malkin.2015)
malkin.2015.game.sp = get.game.save.percent(malkin.2015)
malkin.2016.sp = get.save.percent(malkin.2016)
malkin.2016.game.sp = get.game.save.percent(malkin.2016)
malkin.2017.sp = get.save.percent(malkin.2017)
malkin.2017.game.sp = get.game.save.percent(malkin.2017)
malkin.2018.sp = get.save.percent(malkin.2018)
malkin.2018.game.sp = get.game.save.percent(malkin.2018)

malkin.2015.gp = get.goal.percent(malkin.2015)
malkin.2015.game.gp = get.game.goal.percent(malkin.2015)
malkin.2016.gp = get.goal.percent(malkin.2016)
malkin.2016.game.gp = get.game.goal.percent(malkin.2016)
malkin.2017.gp = get.goal.percent(malkin.2017)
```



```

malkin.2017.game.gp = get.game.goal.percent(malkin.2017)
malkin.2018.gp = get.goal.percent(malkin.2018)
malkin.2018.game.gp = get.game.goal.percent(malkin.2018)

malkin.2015.spg = get.shots.per.goal(malkin.2015)
malkin.2015.game.spg = get.game.shots.per.goal(malkin.2015)
malkin.2016.spg = get.shots.per.goal(malkin.2016)
malkin.2016.game.spg = get.game.shots.per.goal(malkin.2016)
malkin.2017.spg = get.shots.per.goal(malkin.2017)
malkin.2017.game.spg = get.game.shots.per.goal(malkin.2017)
malkin.2018.spg = get.shots.per.goal(malkin.2018)
malkin.2018.game.spg = get.game.shots.per.goal(malkin.2018)

```

Here's a table of his data.

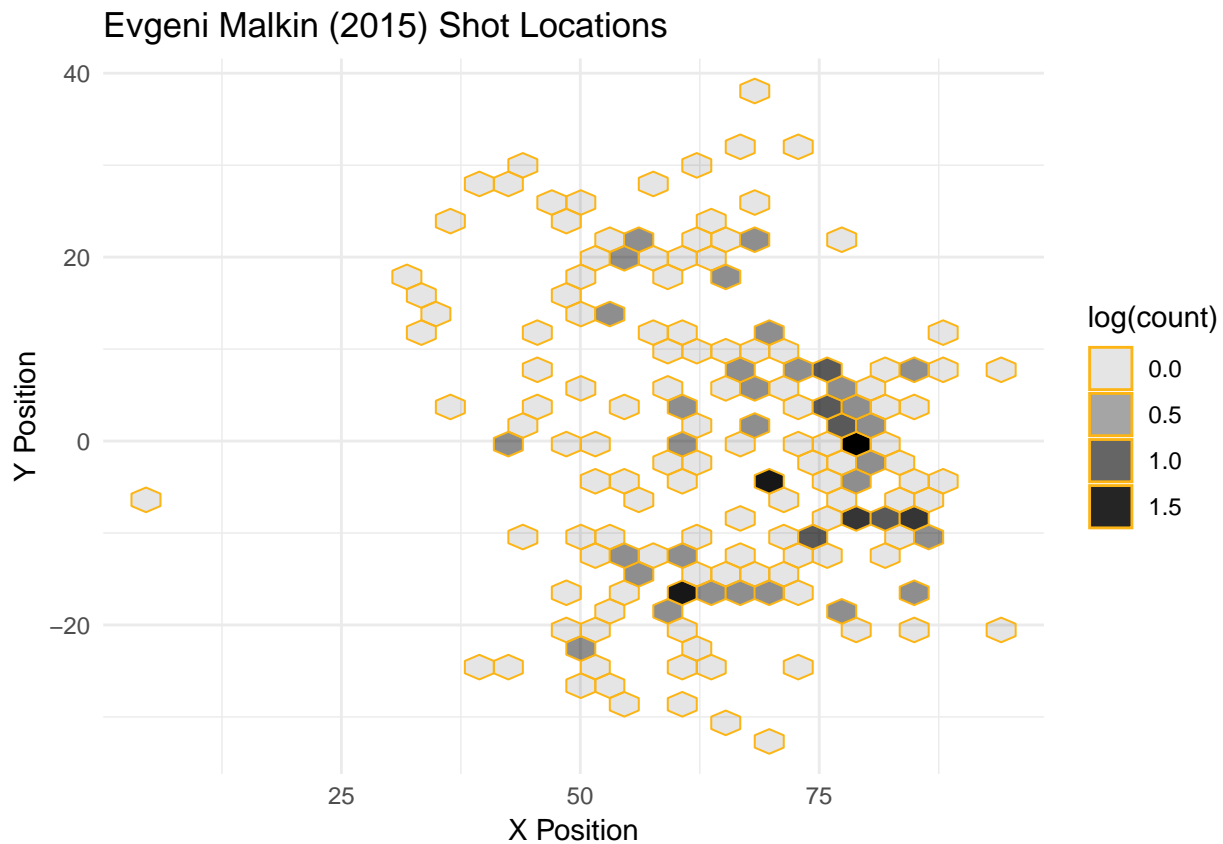
| Season | Save Percent | Goal Percent | Shots per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.8767123 | 0.1232877 | 8.111111 |
| 2016-2017 | 0.8674699 | 0.1325301 | 7.545454 |
| 2017-2018 | 0.8622951 | 0.1377049 | 7.261904 |
| 2018-2019 | 0.9064748 | 0.0935252 | 10.692307 |

Let's take a look at his heatmaps.

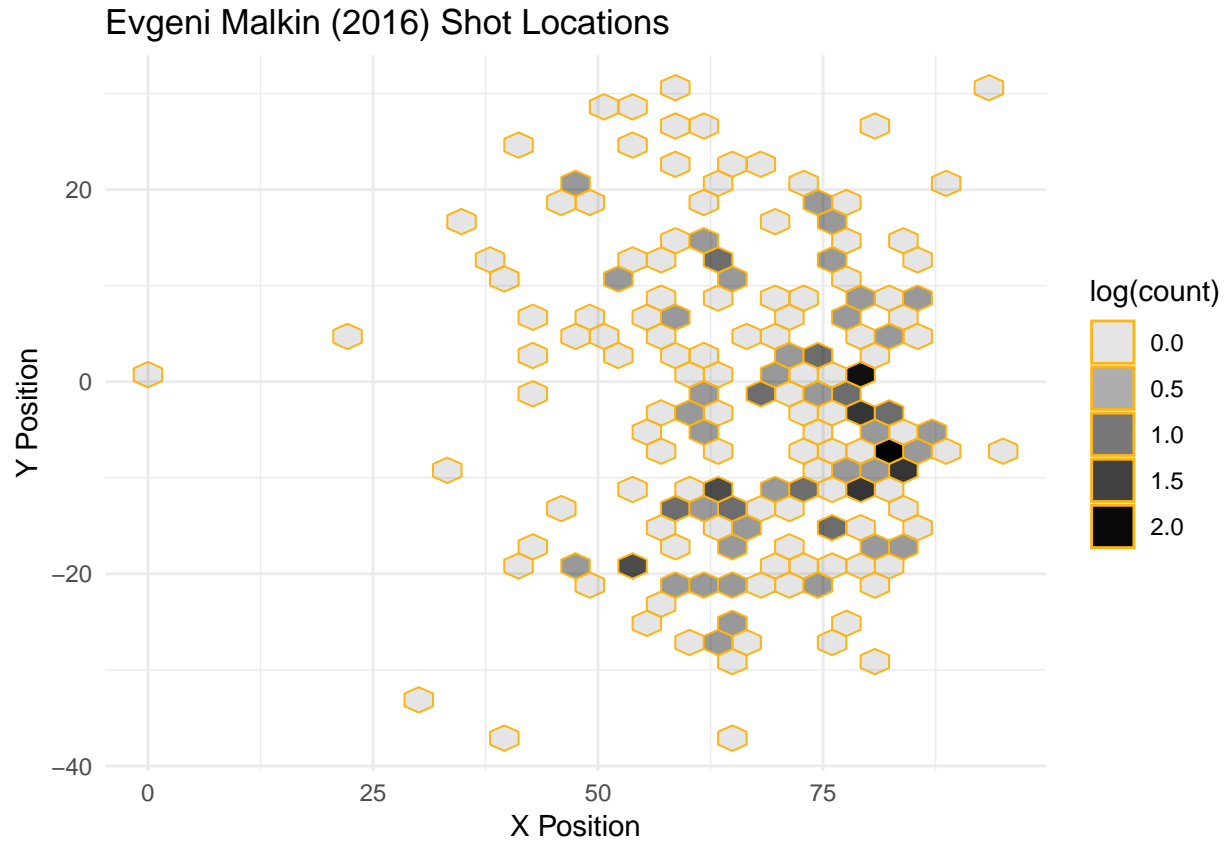
```

malkin.2015.locations.plot = graph.shot.locations(malkin.2015, "#000000", "#FCB514", "Evgeni Malkin (2015)", "#000000")
malkin.2015.locations.plot

```

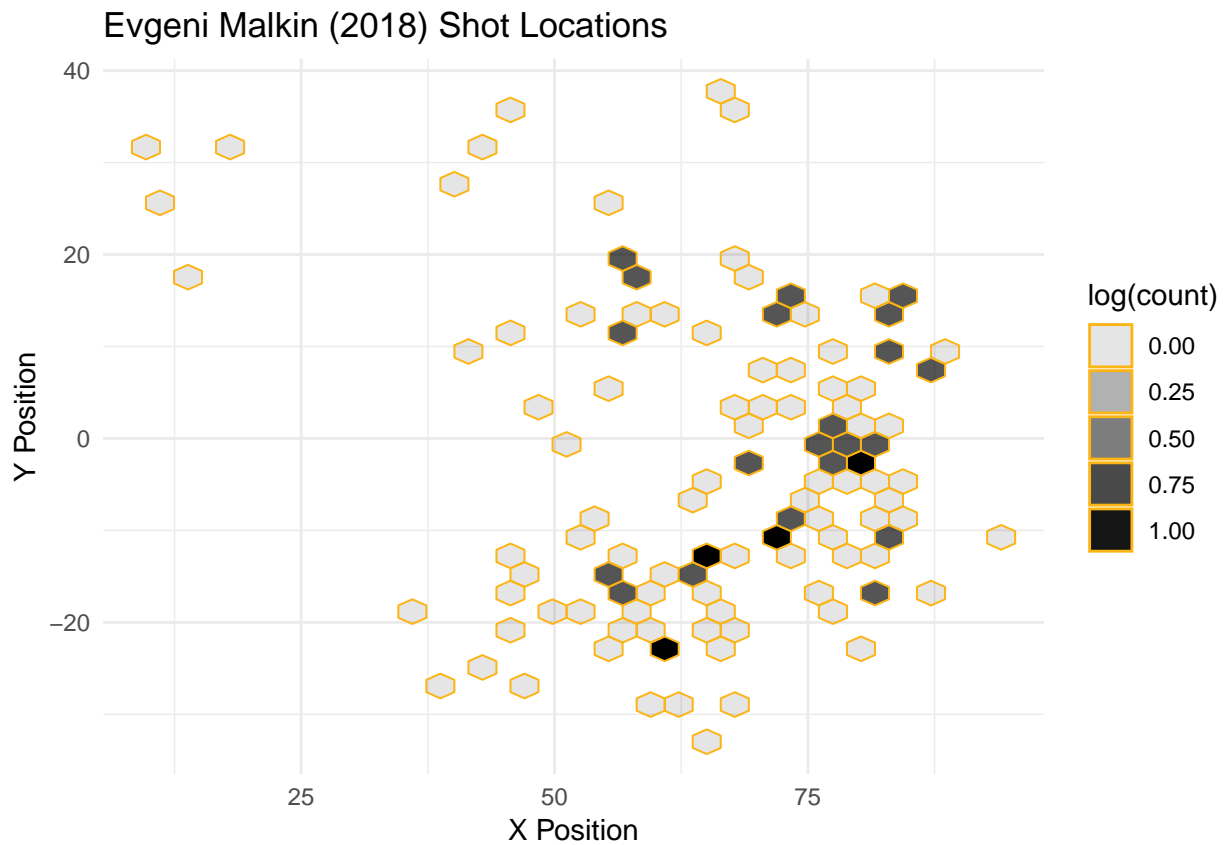


```
malkin.2016.locations.plot = graph_shot.locations(malkin.2016, "#000000", "#FCB514", "Evgeni Malkin (2016)")
malkin.2016.locations.plot
```



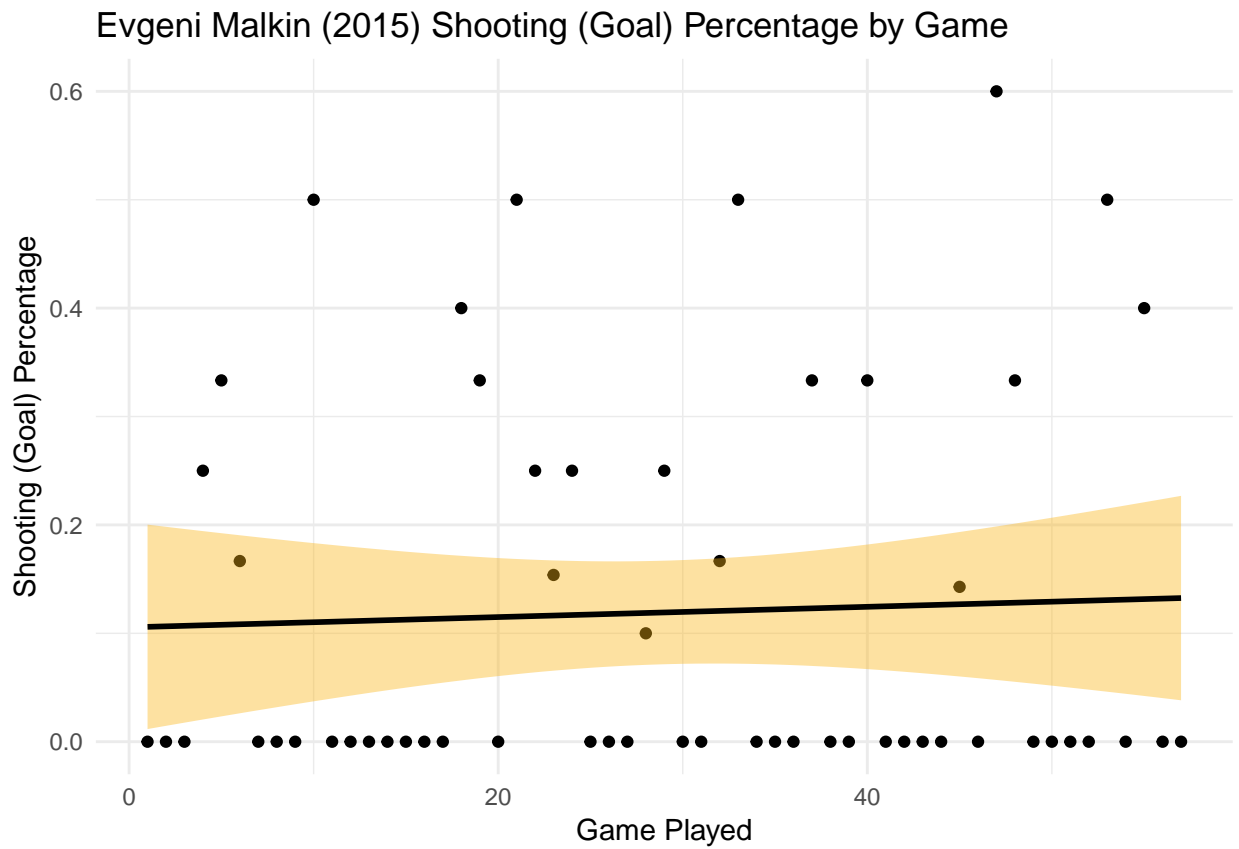
```
malkin.2017.locations.plot = graph_shot.locations(malkin.2017, "#000000", "#FCB514", "Evgeni Malkin (2017)")
malkin.2017.locations.plot
```

```
malkin.2018.locations.plot = graph.shot.locations(malkin.2018, "#000000", "#FCB514", "Evgeni Malkin (2018)")
malkin.2018.locations.plot
```



Let's look at this goal percentage per game now.

```
malkin.2015.gp.plot = graph.trend(malkin.2015.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB51D", "#000000")
malkin.2015.gp.plot
```

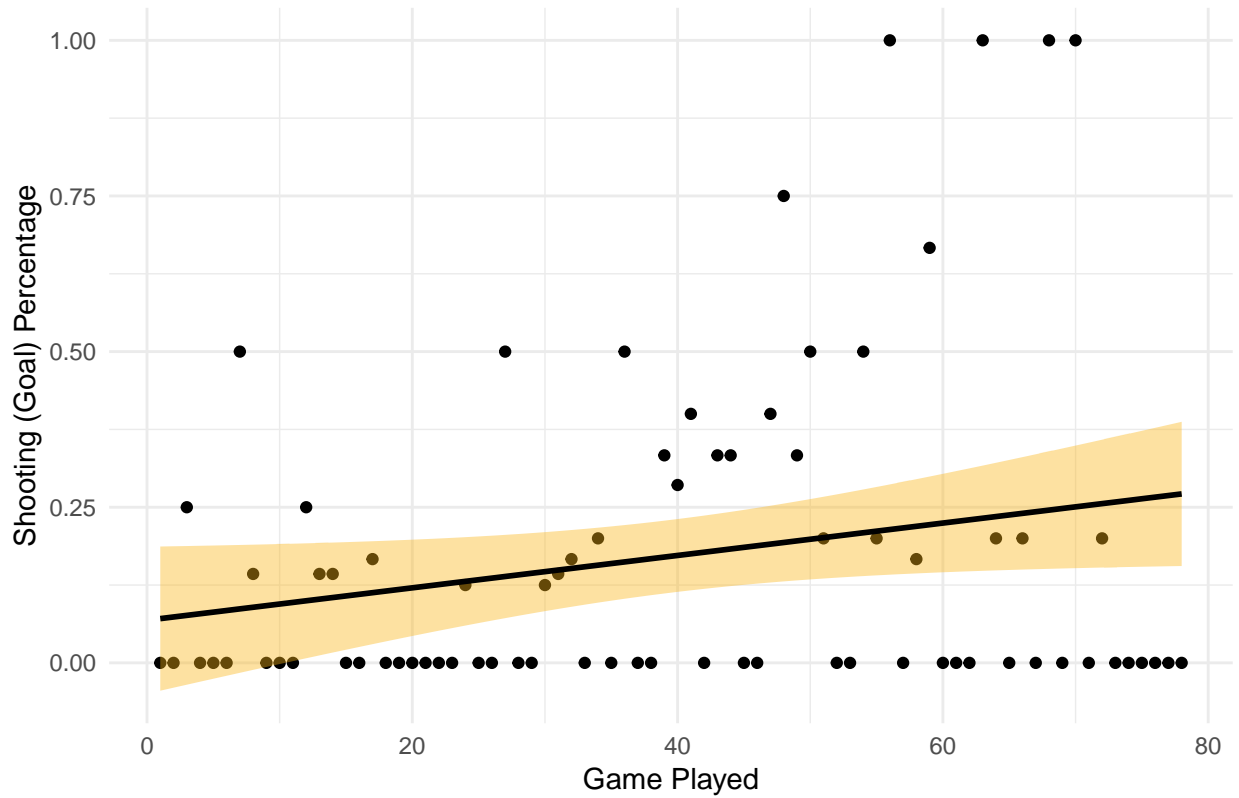


```
malkin.2016.gp.plot = graph.trend(malkin.2016.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB511", "#000000")
malkin.2016.gp.plot
```

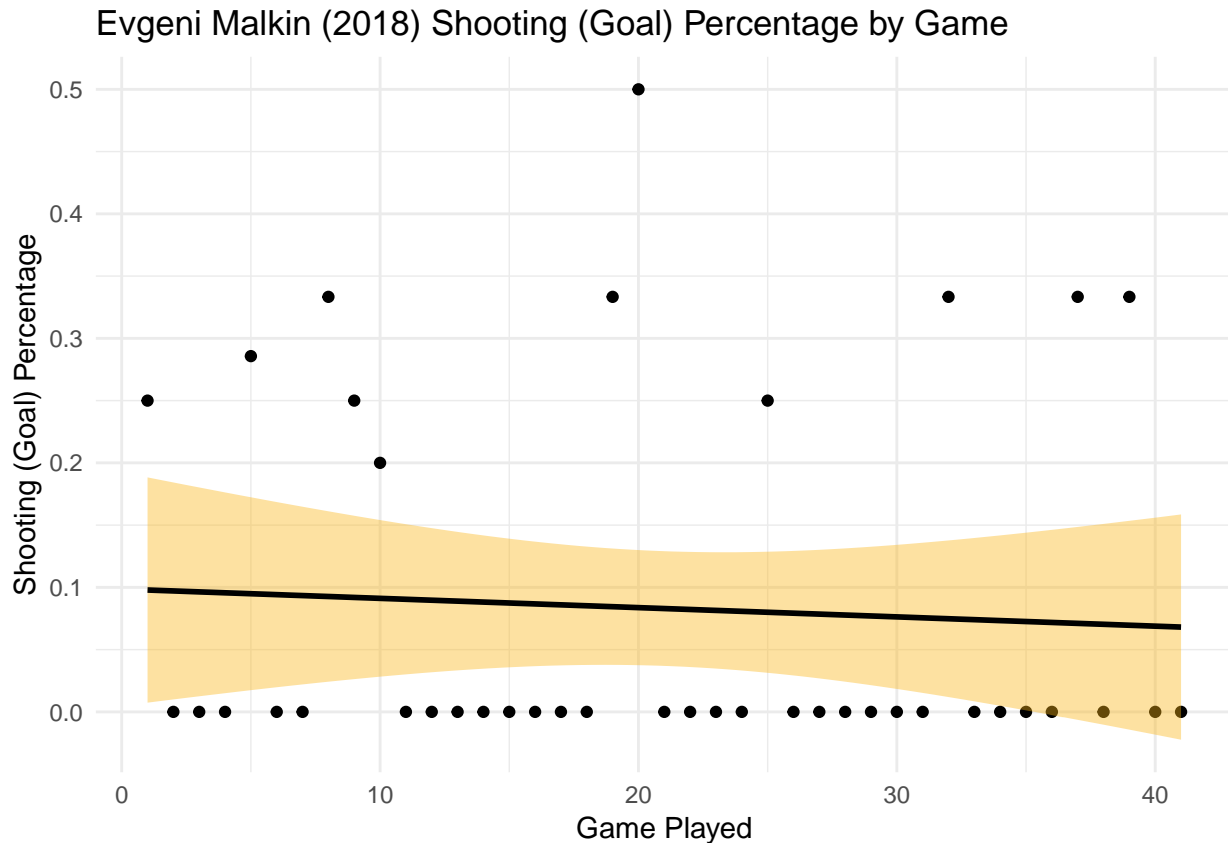
The plot displays the relationship between the number of games played and the shooting percentage. The y-axis, labeled 'Shooting (Goal) Percentage', ranges from 0.00 to 1.00. The x-axis, labeled 'Game Played', ranges from 0 to 60. Black dots represent individual game data points, showing a wide range of shooting percentages, including several 1.00 values. Brown dots represent the mean shooting percentage for each game, which are clustered around 0.15-0.20. A solid black line shows a slight upward trend, and a yellow shaded area represents the confidence interval.

```
malkin.2017.gp.plot = graph.trend(malkin.2017.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB51F")
malkin.2017.gp.plot
```

Evgeni Malkin (2017) Shooting (Goal) Percentage by Game



```
malkin.2018.gp.plot = graph.trend(malkin.2018.game.gp, "Shooting (Goal) Percentage", "#000000", "#FCB51D", "#000000")
malkin.2018.gp.plot
```



Alex Ovechkin

Alexander Ovechkin's style of play is simply electric. With an extremely hard and accurate slap shot mixed in with a deadly quick wrist shot, Ovechkin is a scoring machine. The Russian-born winger has cemented himself as one of the best Russians to ever play in the NHL.

```
ovechkin.2015 = get.skater.data(analysis.2015, "Alex Ovechkin")
ovechkin.2016 = get.skater.data(analysis.2016, "Alex Ovechkin")
ovechkin.2017 = get.skater.data(analysis.2017, "Alex Ovechkin")
ovechkin.2018 = get.skater.data(analysis.2018, "Alex Ovechkin")
```

Let's get some of his stats.

```
ovechkin.2015.sp = get.save.percent(ovechkin.2015)
ovechkin.2015.game.sp = get.game.save.percent(ovechkin.2015)
ovechkin.2016.sp = get.save.percent(ovechkin.2016)
ovechkin.2016.game.sp = get.game.save.percent(ovechkin.2016)
ovechkin.2017.sp = get.save.percent(ovechkin.2017)
ovechkin.2017.game.sp = get.game.save.percent(ovechkin.2017)
ovechkin.2018.sp = get.save.percent(ovechkin.2018)
ovechkin.2018.game.sp = get.game.save.percent(ovechkin.2018)

ovechkin.2015.gp = get.goal.percent(ovechkin.2015)
ovechkin.2015.game.gp = get.game.goal.percent(ovechkin.2015)
ovechkin.2016.gp = get.goal.percent(ovechkin.2016)
ovechkin.2016.game.gp = get.game.goal.percent(ovechkin.2016)
ovechkin.2017.gp = get.goal.percent(ovechkin.2017)
```



```

ovechkin.2017.game.gp = get.game.goal.percent(ovechkin.2017)
ovechkin.2018.gp = get.goal.percent(ovechkin.2018)
ovechkin.2018.game.gp = get.game.goal.percent(ovechkin.2018)

ovechkin.2015.spg = get.shots.per.goal(ovechkin.2015)
ovechkin.2015.game.spg = get.game.shots.per.goal(ovechkin.2015)
ovechkin.2016.spg = get.shots.per.goal(ovechkin.2016)
ovechkin.2016.game.spg = get.game.shots.per.goal(ovechkin.2016)
ovechkin.2017.spg = get.shots.per.goal(ovechkin.2017)
ovechkin.2017.game.spg = get.game.shots.per.goal(ovechkin.2017)
ovechkin.2018.spg = get.shots.per.goal(ovechkin.2018)
ovechkin.2018.game.spg = get.game.shots.per.goal(ovechkin.2018)

```

Here's a table of his data.

| Season | Save Percent | Goal Percent | Shots per Goal |
|-----------|--------------|--------------|----------------|
| 2015-2016 | 0.9111901 | 0.0888099 | 11.26 |
| 2016-2017 | 0.9274725 | 0.0725275 | 13.7878788 |
| 2017-2018 | 0.9068441 | 0.0931559 | 10.7346939 |
| 2018-2019 | 0.8823529 | 0.1176471 | 8.5 |

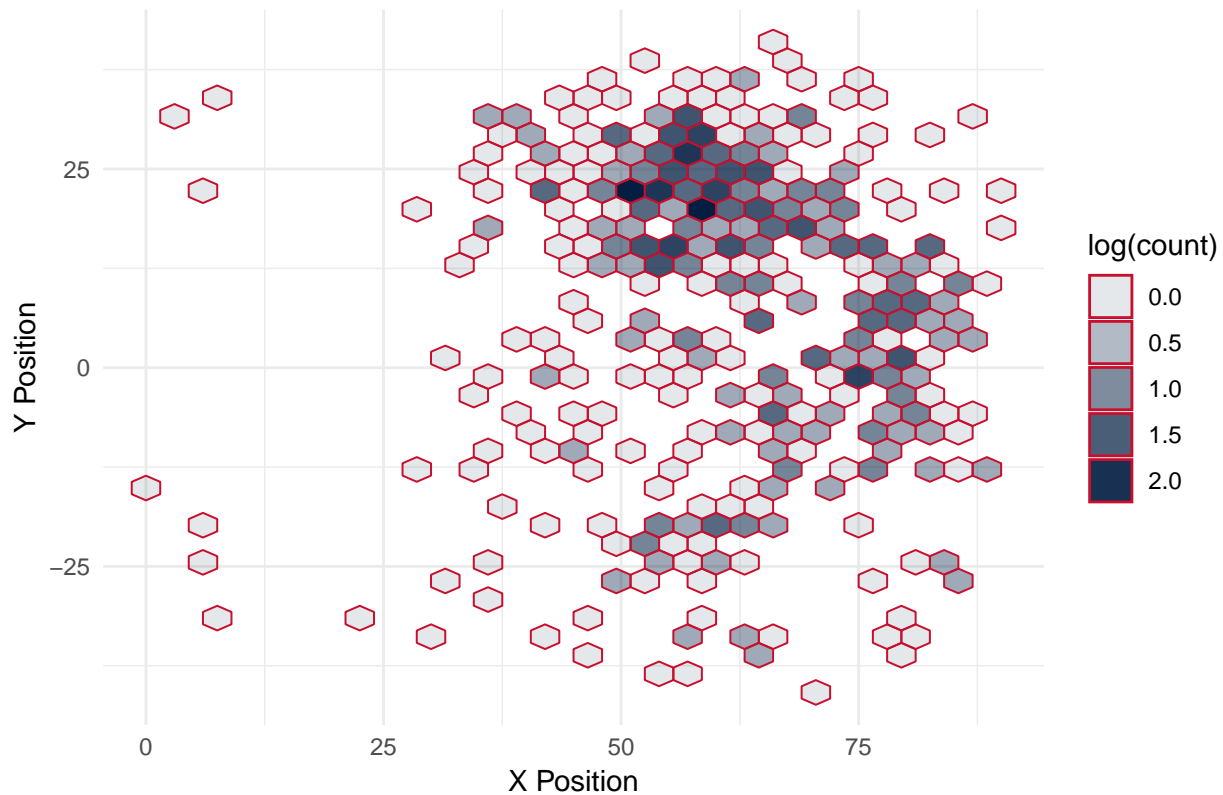
Let's take a look at his heatmaps.

```

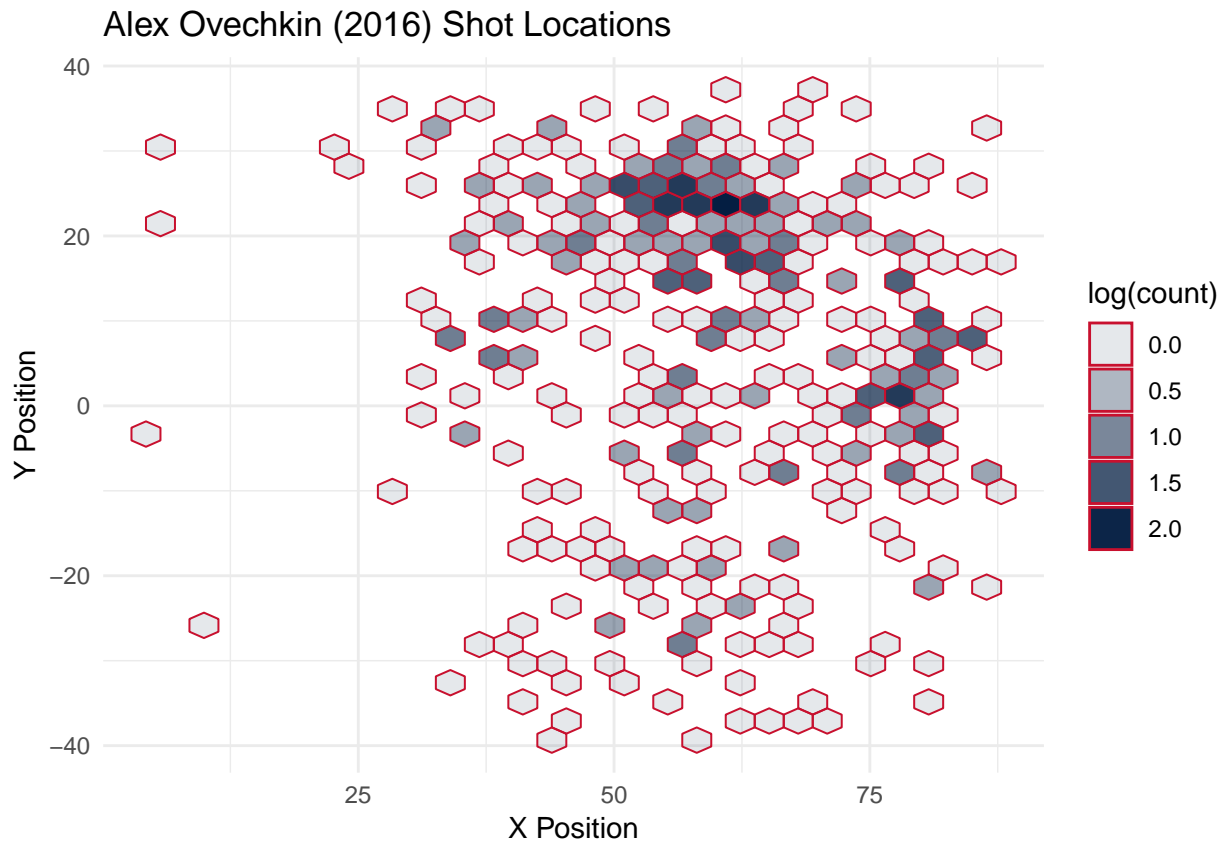
ovechkin.2015.locations.plot = graph.shot.locations(ovechkin.2015, "#041E42", "#C8102E", "Alex Ovechkin")
ovechkin.2015.locations.plot

```

Alex Ovechkin (2015) Shot Locations



```
ovechkin.2016.locations.plot = graph.shot.locations(ovechkin.2016, "#041E42", "#C8102E", "Alex Ovechkin")
ovechkin.2016.locations.plot
```

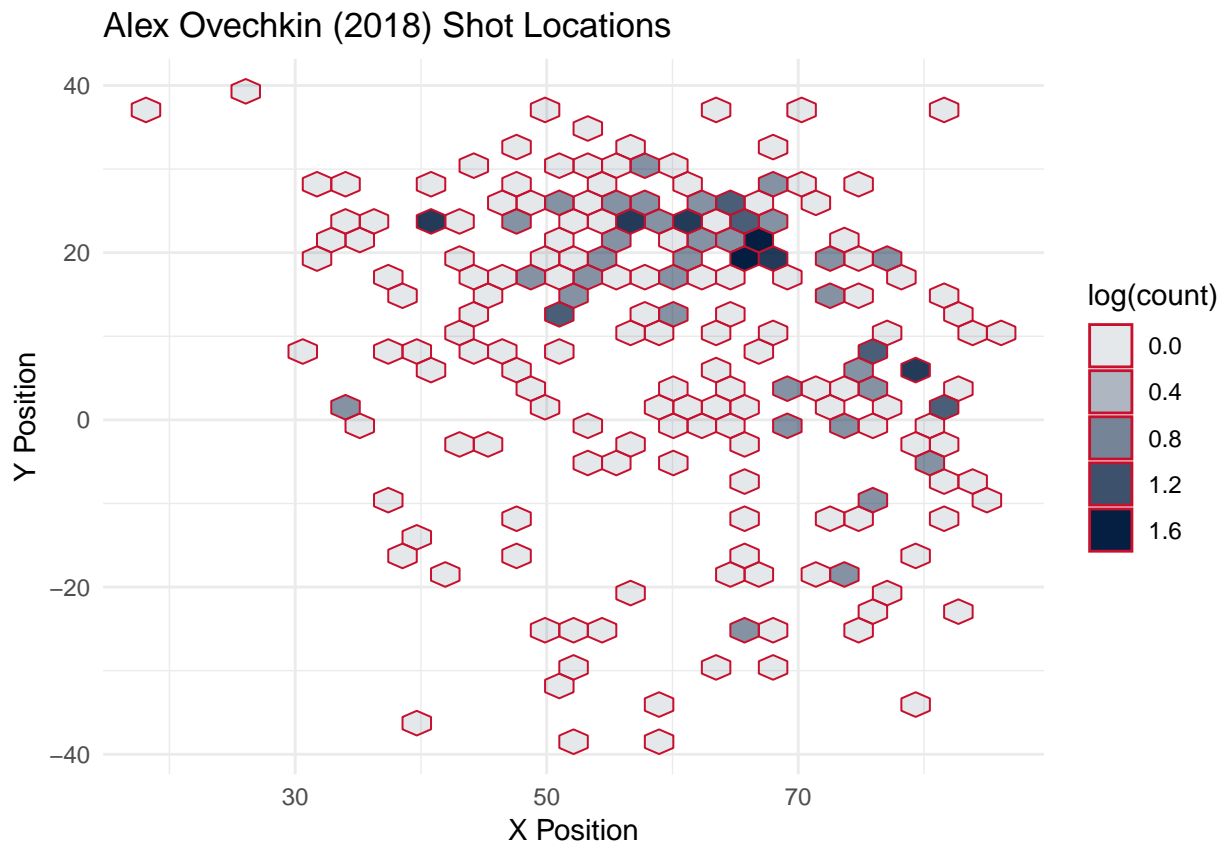


```
ovechkin.2017.locations.plot = graph.shot.locations(ovechkin.2017, "#041E42", "#C8102E", "Alex Ovechkin")
ovechkin.2017.locations.plot
```

Alex Ovechkin (2017) Shot Locations



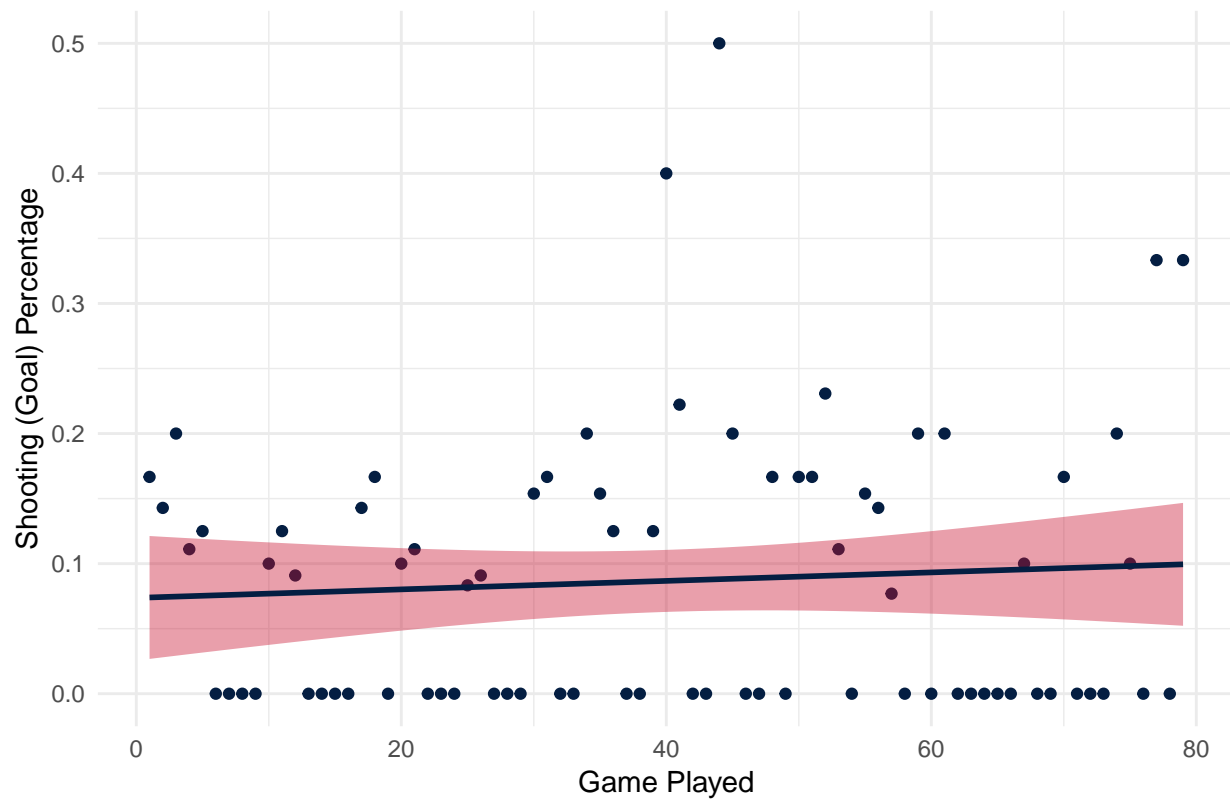
```
ovechkin.2018.locations.plot = graph.shot.locations(ovechkin.2018, "#041E42", "#C8102E", "Alex Ovechkin")
ovechkin.2018.locations.plot
```



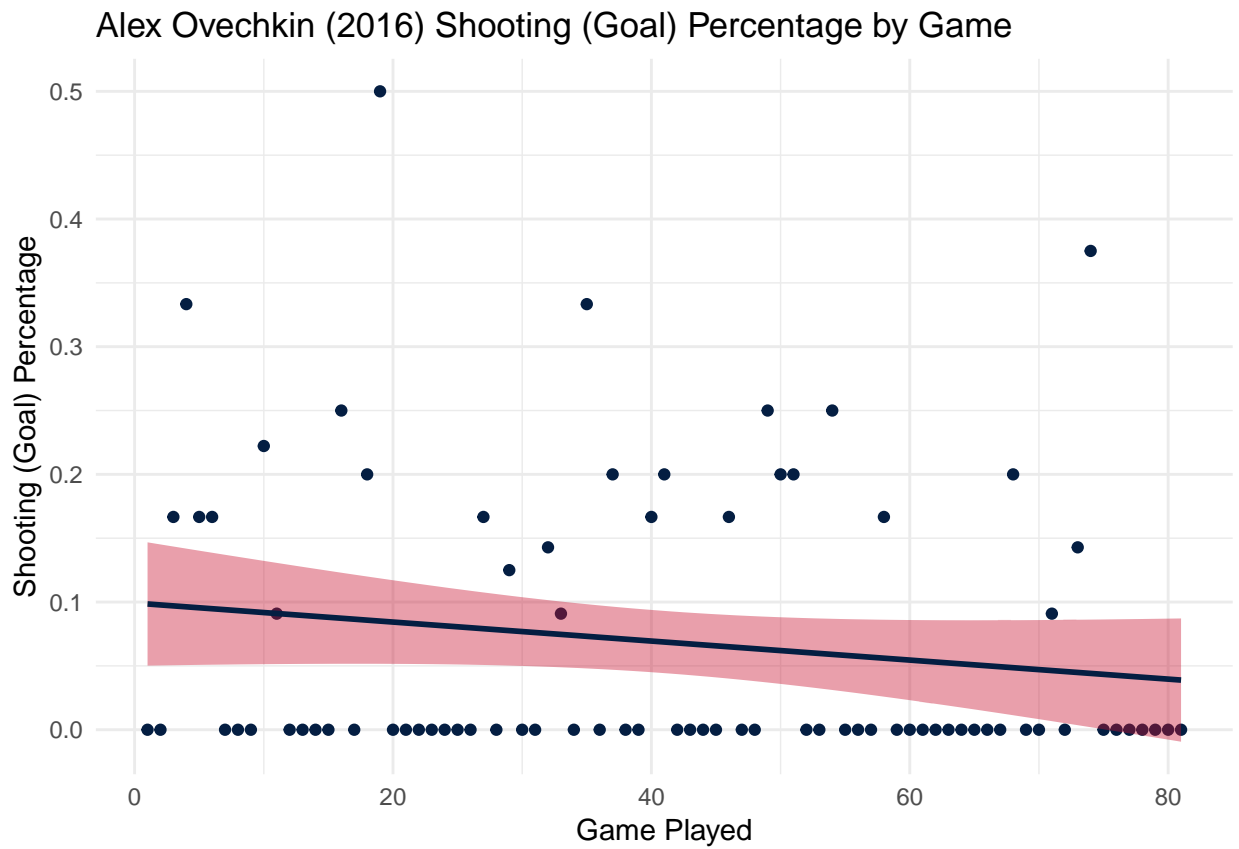
Let's look at this goal percentage per game now.

```
ovechkin.2015.gp.plot = graph.trend(ovechkin.2015.game.gp, "Shooting (Goal) Percentage", "#041E42", "#C44E52")
ovechkin.2015.gp.plot
```

Alex Ovechkin (2015) Shooting (Goal) Percentage by Game

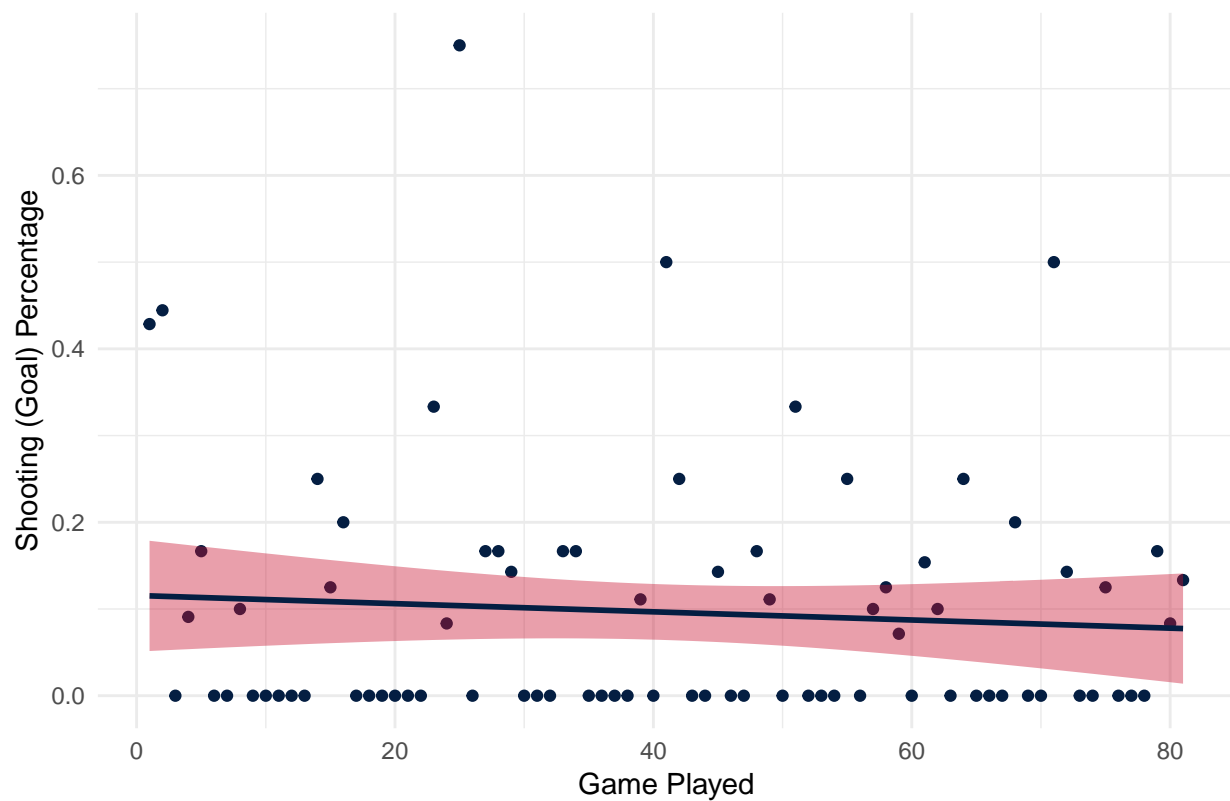


```
ovechkin.2016.gp.plot = graph.trend(ovechkin.2016.game.gp, "Shooting (Goal) Percentage", "#041E42", "#C43A3A", "#041E42", "#C43A3A")
ovechkin.2016.gp.plot
```

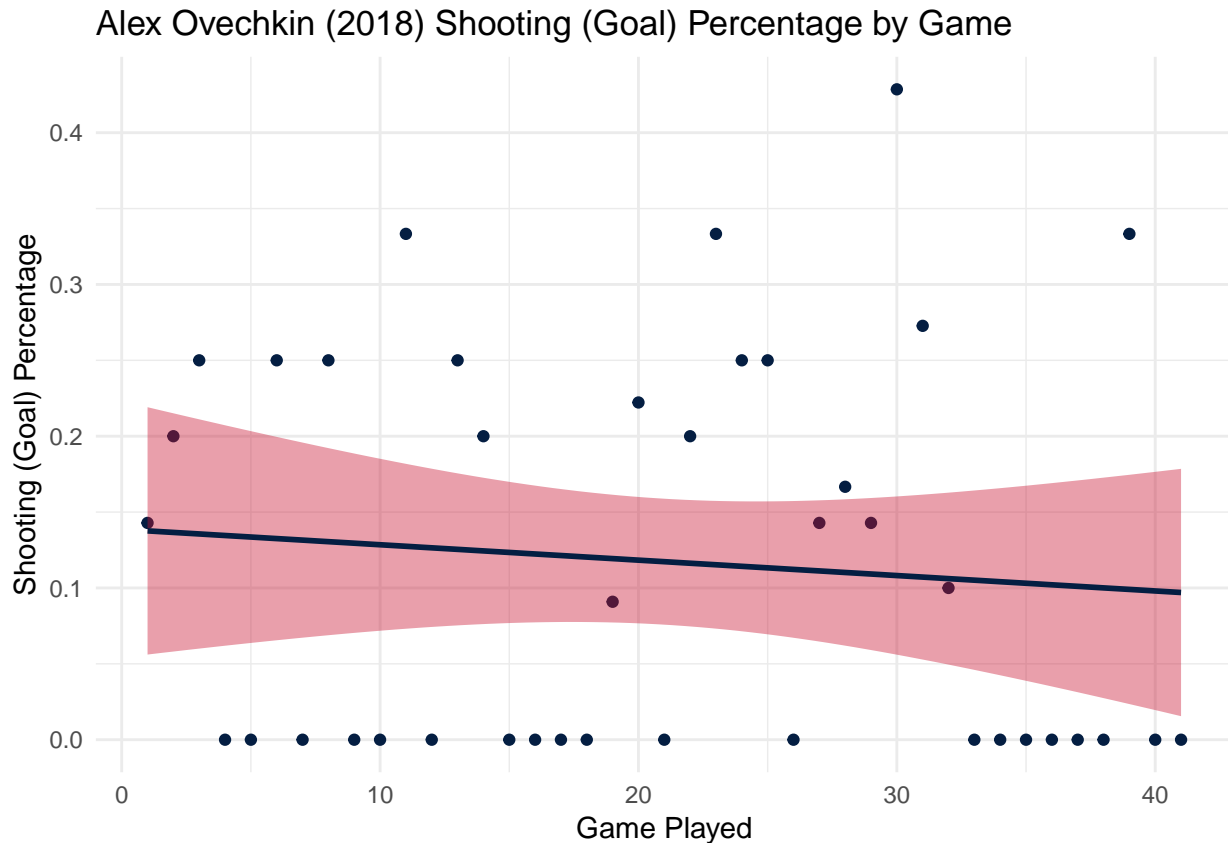


```
ovechkin.2017.gp.plot = graph.trend(ovechkin.2017.game.gp, "Shooting (Goal) Percentage", "#041E42", "#C85A3A", "#F08080")
ovechkin.2017.gp.plot
```

Alex Ovechkin (2017) Shooting (Goal) Percentage by Game



```
ovechkin.2018.gp.plot = graph.trend(ovechkin.2018.game.gp, "Shooting (Goal) Percentage", "#041E42", "#C8513D", "#041E42", "#C8513D")
ovechkin.2018.gp.plot
```



Jake Guentzel

Jake Guentzel has been an elite farm-system talent for the Penguins.

```
guentzel.2016 = get.skater.data(analysis.2016, "Jake Guentzel")
guentzel.2017 = get.skater.data(analysis.2017, "Jake Guentzel")
guentzel.2018 = get.skater.data(analysis.2018, "Jake Guentzel")
```

Let's get some of his stats.

```
guentzel.2016.sp = get.save.percent(guentzel.2016)
guentzel.2016.game.sp = get.game.save.percent(guentzel.2016)
guentzel.2017.sp = get.save.percent(guentzel.2017)
guentzel.2017.game.sp = get.game.save.percent(guentzel.2017)
guentzel.2018.sp = get.save.percent(guentzel.2018)
guentzel.2018.game.sp = get.game.save.percent(guentzel.2018)

guentzel.2016.gp = get.goal.percent(guentzel.2016)
guentzel.2016.game.gp = get.game.goal.percent(guentzel.2016)
guentzel.2017.gp = get.goal.percent(guentzel.2017)
guentzel.2017.game.gp = get.game.goal.percent(guentzel.2017)
guentzel.2018.gp = get.goal.percent(guentzel.2018)
guentzel.2018.game.gp = get.game.goal.percent(guentzel.2018)

guentzel.2016.spg = get.shots.per.goal(guentzel.2016)
guentzel.2016.game.spg = get.game.shots.per.goal(guentzel.2016)
guentzel.2017.spg = get.shots.per.goal(guentzel.2017)
```



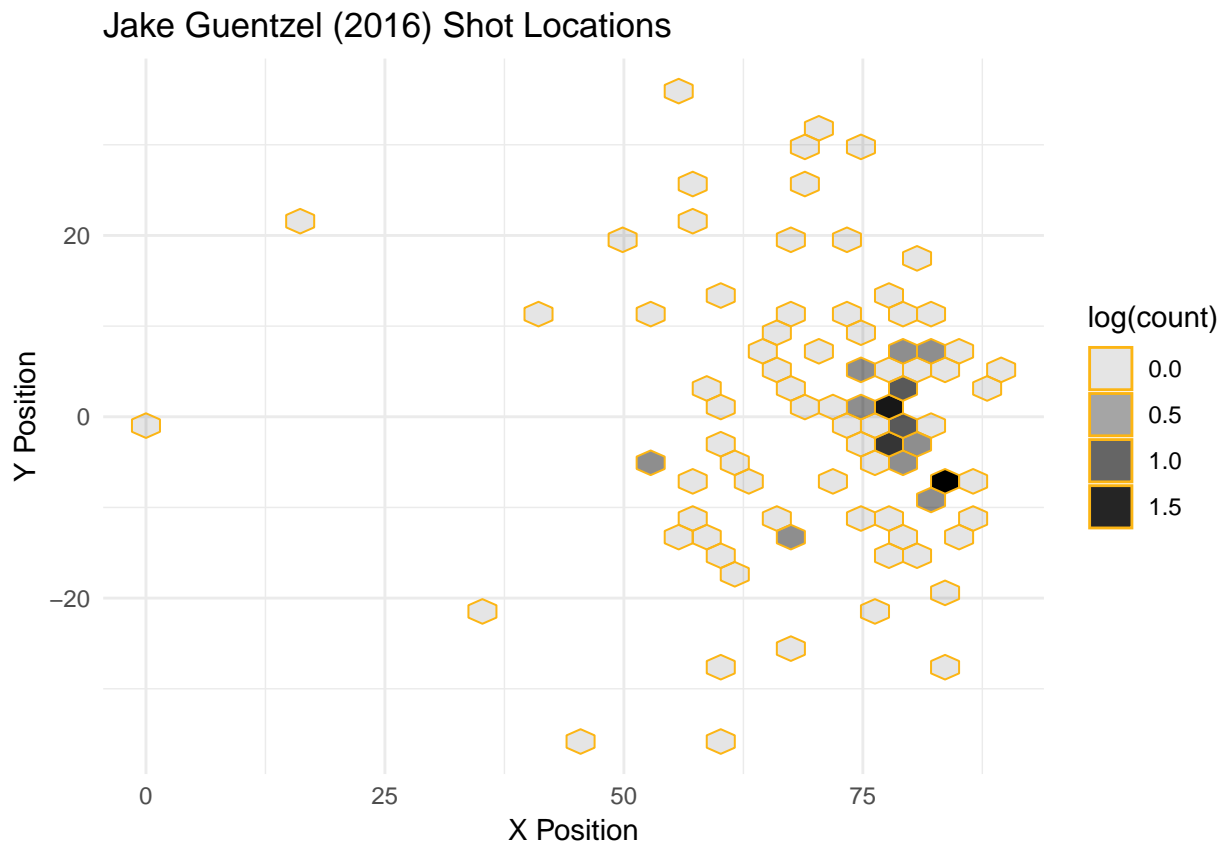
```
guentzel.2017.game.spg = get.game.shots.per.goal(guentzel.2017)
guentzel.2018.spg = get.shots.per.goal(guentzel.2018)
guentzel.2018.game.spg = get.game.shots.per.goal(guentzel.2018)
```

Here's a table of his data.

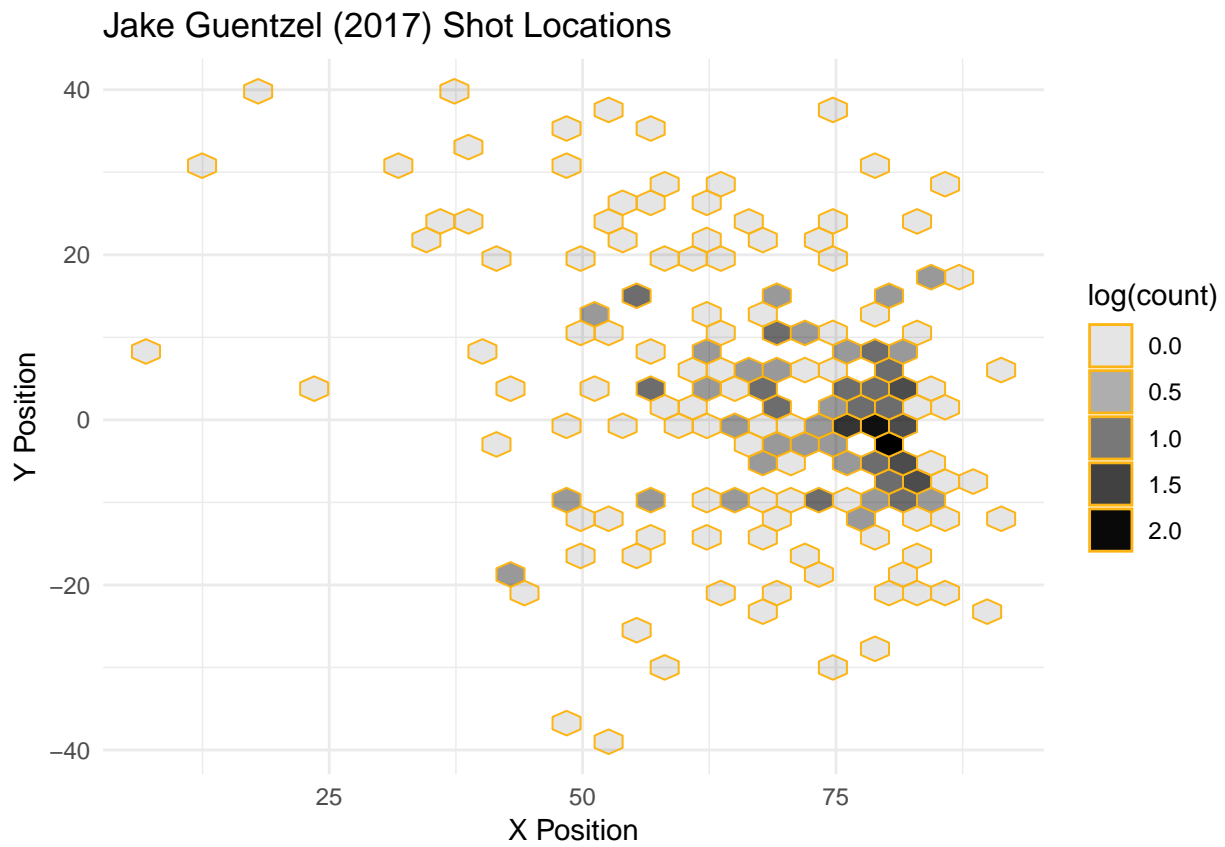
| Season | Save Percent | Goal Percent | Shots per Goal |
|-----------|--------------|--------------|----------------|
| 2016-2017 | 0.8518519 | 0.1481481 | 6.75 |
| 2017-2018 | 0.9079498 | 0.0920502 | 10.8636364 |
| 2018-2019 | 0.875 | 0.125 | 8 |

From this data, we can see that Guentzel isn't a sniper; instead he is a volume shooter. Let's take a look at his heatmaps.

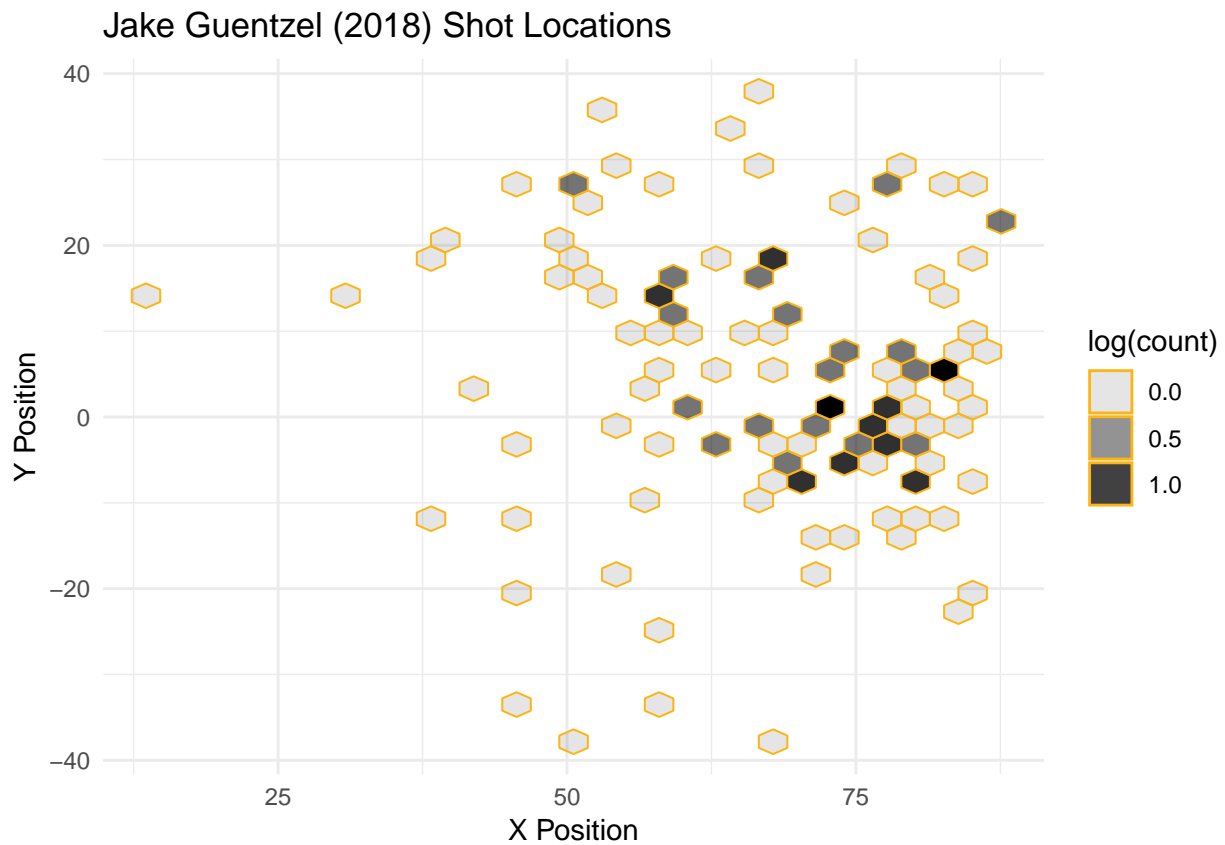
```
guentzel.2016.locations.plot = graph.shot.locations(guentzel.2016, "#000000", "#FCB514", "Jake Guentzel")
guentzel.2016.locations.plot
```



```
guentzel.2017.locations.plot = graph.shot.locations(guentzel.2017, "#000000", "#FCB514", "Jake Guentzel")
guentzel.2017.locations.plot
```



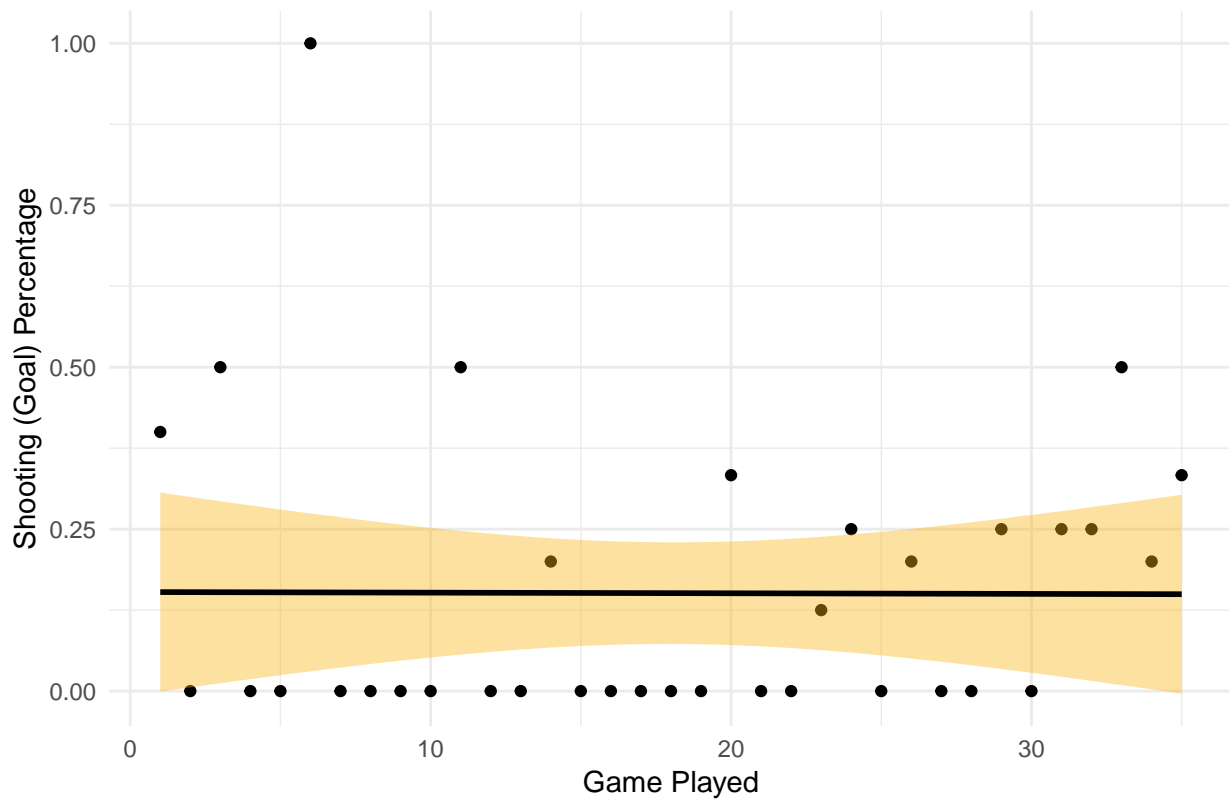
```
guentzel.2018.locations.plot = graph.shot.locations(guentzel.2018, "#000000", "#FCB514", "Jake Guentzel")
guentzel.2018.locations.plot
```



From this, we can see that Jake Guentzel has a lot of success around the net. Let's look at this goal percentage per game now.

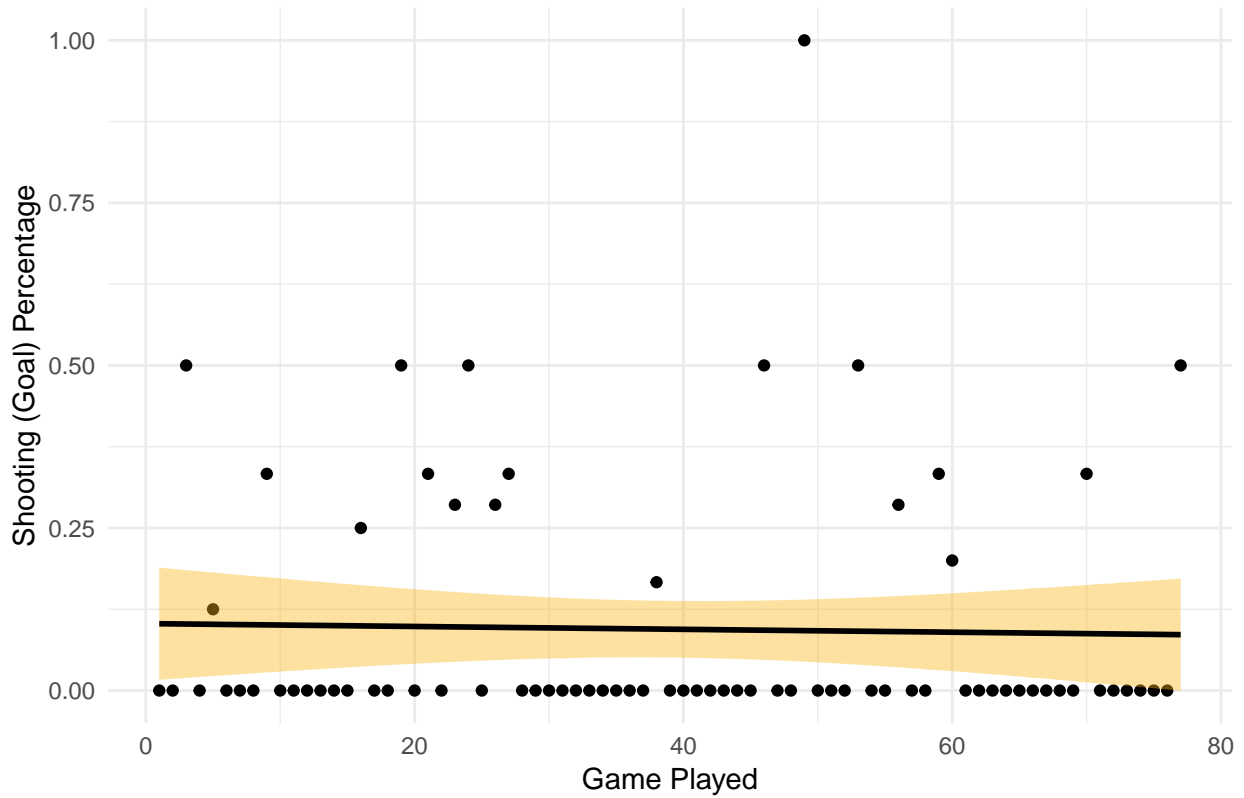
```
guentzel.2016.gp.plot = graph.trend(guentzel.2016.game.gp, "Shooting (Goal) Percentage", "#000000", "#F00000")
guentzel.2016.gp.plot
```

Jake Guentzel (2016) Shooting (Goal) Percentage by Game

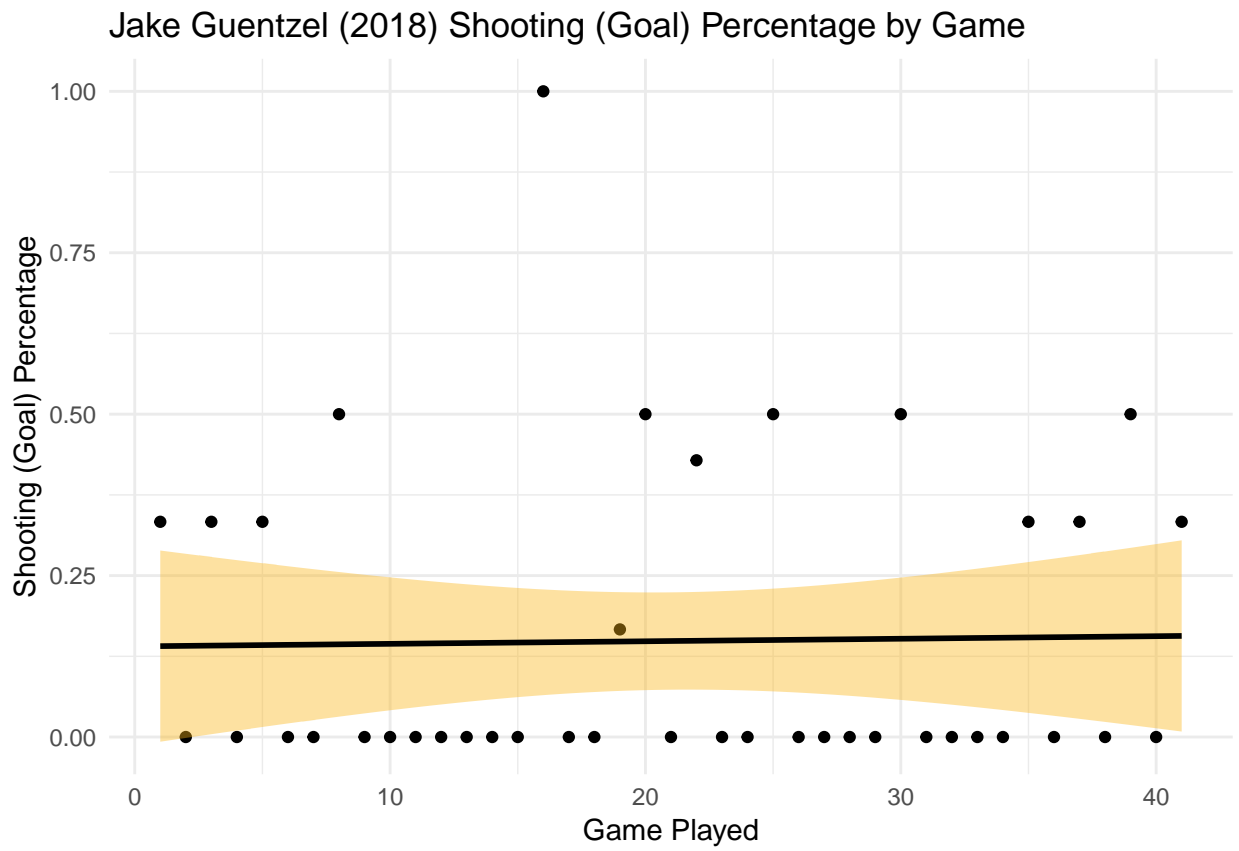


```
guentzel.2017.gp.plot = graph.trend(guentzel.2017.game.gp, "Shooting (Goal) Percentage", "#000000", "#F0E68C", "#F0E68C")
guentzel.2017.gp.plot
```

Jake Guentzel (2017) Shooting (Goal) Percentage by Game



```
guentzel.2018.gp.plot = graph.trend(guentzel.2018.game.gp, "Shooting (Goal) Percentage", "#000000", "#F08080")
guentzel.2018.gp.plot
```



Erik Karlsson

Kris Letang