Introduction

The world of Pokémon emerged in 1995 with the games Pokémon Red and Pokémon Green. For western audiences, the latter would become known as Pokémon Blue. These two games introduced a new flavor of turn-based gaming that would become known as Pokémon battling. Numerous other games have copied the Pokémon battling format, though none have been able to equal its widespread appeal and dedicated player base. By adding new items, Pokémon types, and of course new Pokémon into the game, the world of Pokémon continues to evolve and grow. The most recent iterations of Pokémon games, Pokémon Omega Ruby and Alpha Sapophire, have continued the tradition of adding layers to an already complex system of battling.

Since 2011 the program Pokémon Showdown has offered a stripped version of Pokémon games, based solely on the Pokémon battling system. With well-over 12,000 daily registered users and counting, this program has become the go-to program to test and practice battling strategies in the ultimate pursuit of becoming the very best that no one ever was.

A prominent line of inquiry embedded in the practice of game theory involves the process of decision making. This process has been explored rigorously in the past, especially so in the newly emerging field behavioral economics and behavioral game theory. I expand upon the recent studies in these fields by analyzing specific battling strategies and gauge their effectiveness in general play. Effectiveness is gauged using the experimental probability of winning given specific strategy sets. This will require categorizing different moves into sets indicative of specific behavior. To begin with I analyze the effectiveness of purely aggressive strategies, noted by the use of purely damaging moves. These aggressive strategy sets are compared to mixed defense-offense and purely defensive strategy sets. However my ultimate pursuit is to analyze and highlight the relevant conditions on which to base such hypotheses. By incorporating tenants of behavioral economics and game theory together, I hope to bridge two fields that up until recently have refrained from interdisciplinary inquiry.

Methodology

The data used is a compilation of battle logs from the Pokémon Showdown servers. Each battle log is stored as a separate .json file. The data spans across the year 2015, composed of four different months of data. The four months are March, June, September, and December. There are no lapses in data, i.e. each day of each month has numerous battle logs to account for. No dramatic overhaul was done to the Pokémon battling format at this time, though some minor adjustments were made. Furthermore only ranked games are included in the dataset, indicating that each player stands to gain or lose from the battle.

Pokémon Battling Basics

The Pokémon battle starts with Pokémon being sent out. For the purposes of the data used, one Pokémon is sent out for each opponent, totalling two Pokémon being out at any given time. Following this, each Pokémon has 4 moves to choose from, along with the option to switch to a different Pokémon (when applicable). After both players make a decision, the moves are weighted for priority and speed to determine the order of play. If both players decide not to switch one Pokémon will attack the other, after which the next Pokémon will do the same if it has not fainted. After each move has been executed the turn ends and the process is repeated. When one of the Pokémon faints, the player whose Pokémon fainted will be prompted to select another Pokémon from the bench. The first player to lose all of their Pokémon loses the battle.

Battle Formats

The data used for this study include two different Pokémon battling formats. The two formats are known as Over Used and Random Battles, abbreviated as OU and Randbats respectively. Both formats have teams of six Pokémon and only allow one Pokémon to be out at any given time. While both battle formats are subsets of what are known as single battles, each has their own unique spin on the Pokémon battling format.

Random Battles are the most frequently played format. Neither player gets to decide on their initial Pokémon nor do they have any input on the composition of the team. The format uses an algorithm to determine team compositions. However it is important to note that there are restrictions to the Randbats format that center around team composition and move composition for specific Pokémon.

The Over Used battle format includes team composition. By including team composition, players are able to decide what Pokémon to include on their team, the moves of each Pokémon, and other factors such as held items and abilities. Further restrictions to the OU format include banning specific Pokémon. The restricted Pokémon are included in the "Uber" tier along with the Pokémon Mega-Rayquaza. Additionally certain "hidden" abilities are locked for Pokémon.

Pokémon Attributes

Generally, there are a number of factors that are specific to each Pokémon. Some of these factors are considered static, meaning that they do not change over the course of the battle. These types of factors are noted as "Fixed" Attributes. However there are some factors that are generally regarded as Fixed Attributes but are affected by certain moves, at least when the Pokémon is sent out. These types of factors are considered "Mixed Attributes". Additionally there are attributes that are inherently influenced throughout the course of the battle. These are noted as "Variable Attributes". The terminology is largely taken from the Ho et al. paper for ease of translation. #### Pokémon Fixed Attributes Fixed attributes include the type(s) of the Pokémon, the four moves the Pokémon has learned, the one item the Pokémon holds, the Pokémon's one selected ability, the level of the Pokémon, and the Pokémon's baseline stats. The latter factor is divided into six categories. These categories include (baseline) Health, Attack, Special Attack, Defense, Special Defense, and Speed. There is further nuance with the inclusion of Pokémon natures and Individual Values, or IVs. These factors influence the base stats of each Pokémon. However due to the sheer number of trivial combinations of IV spreads and nature choices, these two factors will not be a pivotal aspect of framework used.

Pokémon Types

The type(s) of each Pokémon influence not only the potential weaknesses of each Pokémon, but also influence the amount of damage certain type-specific moves are able to do. Each Pokémon has at least one and at most two types. If a Pokémon uses a damaging move whose type corresponds to type of the Pokémon that used it, that Pokémon gets a same type attack bonus, abbreviated as a "stab" bonus. This causes the move to do 50% more damage, potentially 100% if the Pokémon also has the ability Adaptability.

Pokémon Mixed Attributes

Pokémon Variable Attributes

Variable Attributes include the current health of the Pokémon, the status of the Pokémon, the volatile status of the Pokémon, boost data of the Pokémon, and whether the Pokémon in question is currently active.

Literature Review

Scant rigorous or academic research has been conducted within the scope of Pokémon-related topics. The most frequent publications focusing on Pokémon have been strategies guides for each new Pokémon iteration. However these publications have focused more specifically on the games, not necessarily the mechanics of the battling in the games. The most recent publication of *The Official Hoenn Region Strategy Guide* focuses on both the game and the underlying system.

Typically academic papers have focused on the use of algorithms to simulate and play against human players in Pokémon Showdown. One paper gives a rudimentary background on Pokémon battling and focuses explicitly

on 1v1 battles (*Gildardo 2013*). Recent literature following that followed included more nuance. The most relevant publication focuses on the most current framework of Pokémon battling (*Ho et al. 2016*). This publication used the latest iteration of Pokémon Showdown, available at http://pokemonshowdown.com/. The site provides a hub for information on Pokémon battling basics and specific battle format descriptions. Replays and ladder ranking are available for public viewing, along with links to usage statistics and a damage calculator.

Specific usage statistics are found in a subsidiary website, found at http://sweepercalc.com/stats/. The usage statistics track the frequency of use for specific Pokémon, items, abilities, and a host of other relevant variables for Pokémon battling.

A number of links redirect users to the host site of this game: Smogon University. The host site can be found at http://www.smogon.com/. This website offers a wide variety of resources, similar to those found at the Pokémon Showdown website. Most importantly the Smogon forums are a prominent site for discussion of Pokémon battling strategies.

Unsurprisingly game theory vernacular have not entered into the discussion of Pokémon battling strategies. Applying such vernacular to the context of Pokémon battling offers a framework to discuss strategies. In this vein, exploring whether a player will switch Pokémon will necessarily invoke ideas found in behavioral economics. One specific concept engrained in behavioral economics that relates to Pokémon battling is the idea of "keeping doors open". In Chapter 6 of Ariel's foundational work, results indicates that players prefer to keep options available even if doing so incurs costs and/or reduces the payoff (*Predicably Irrational*). The idea of sunk costs in specific relation to Pokémon battling may also provide further insight into the decision making process involved in Pokémon battling. #Include Citation Here#

References

- Sanchez-Ante, Gildardo (2013, Dec) Sistemas Inteligentes: Reportes Finales Ago-Dic 2013. Retrieved from: https://www.researchgate.net/profile/Gildardo_Sanchez-Ante/publication/259343975_Sistemas_Inteligentes_Reportes_Finales_Ago-Dic_2013/links/0c96052b1d0b582e95000000.pdf#page=140
- Ho, Harrison & Ramesh, Varun (2016) Percymon: A Pokemon Showdown Artifical Intelligence. Retrieved from: http://robots.stanford.edu/cs221/2016/restricted/projects/vramesh2/final.pdf
- Usage Stats (2016). Retrieved from: http://sweepercalc.com/stats/
- Smogon University (2016). Retrieved from: http://www.smogon.com/
- Pokémon Showdown (2016). Retrieved from: http://pokemonshowdown.com/
- Ariely, Dan. Predictably Irrational: The Hidden Forces That Shape Our Decisions. New York, NY: Harper, 2009. Print. Chapter 8: Keeping Doors Open (pp. 139-154)
- Erickson, Paul. The World the Game Theorists Made. N.p.: n.p., n.d. Print. Chapter 6: Game Theory without Rationality (pp. 204-239)
- Pokémon Omega Ruby & Pokémon Alpha Sapphire: The Official Hoenn Region Strategy Guide. N.p.: n.p., n.d. Print.
- Camerer, Colin. Behavioral Game Theory: Experiments in Strategic Interaction. New York, NY: Russell Sage Foundation, 2003. Print.
- Pokémon Showdown Github Master Repository (2016). Retrieved from: https://github.com/Zarel/Pokemon-Showdown/tree/master/data in

End thesis Chapter 1

MARKDOWN TEMPLATE

R Markdown Basics

Here is a brief introduction into using *R Markdown*. *Markdown* is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. *R Markdown* provides the flexibility of *Markdown* with the implementation of **R** input and output. For more details on using *R Markdown* see http://rmarkdown.rstudio.com.

Be careful with your spacing in *Markdown* documents. While whitespace largely is ignored, it does at times give *Markdown* signals as to how to proceed. As a habit, try to keep everything left aligned whenever possible, especially as you type a new paragraph. In other words, there is no need to indent basic text in the Rmd document (in fact, it might cause your text to do funny things if you do).

Lists

It's easy to create a list. It can be unordered like

- Item 1
- Item 2

or it can be ordered like

1. Item 1
2. Item 2

Notice that I intentionally mislabeled Item 2 as number 4. *Markdown* automatically figures this out! You can put any numbers in the list and it will create the list. Check it out below.

To create a sublist, just indent the values a bit (at least four spaces or a tab). (Here's one case where indentation is key!)

1. Item 1
2. Item 2
3. Item 3
    - Item 3a
    - Item 3b

Line breaks

Make sure to add white space between lines if you'd like to start a new paragraph. Look at what happens below in the outputted document if you don't:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph. This should be a new paragraph.

Now for the correct way:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph.

This should be a new paragraph.

R chunks

When you click the **Knit** button above a document will be generated that includes both content as well as the output of any embedded **R** code chunks within the document. You can embed an **R** code chunk like this (`cars` is a built-in **R** dataset):

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Inline code

If you'd like to put the results of your analysis directly into your discussion, add inline code like this:

> The `cos` of 2π is 1.

Another example would be the direct calculation of the standard deviation:

> The standard deviation of `speed` in `cars` is 5.2876444.

One last neat feature is the use of the `ifelse` conditional statement which can be used to output text depending on the result of an **R** calculation:

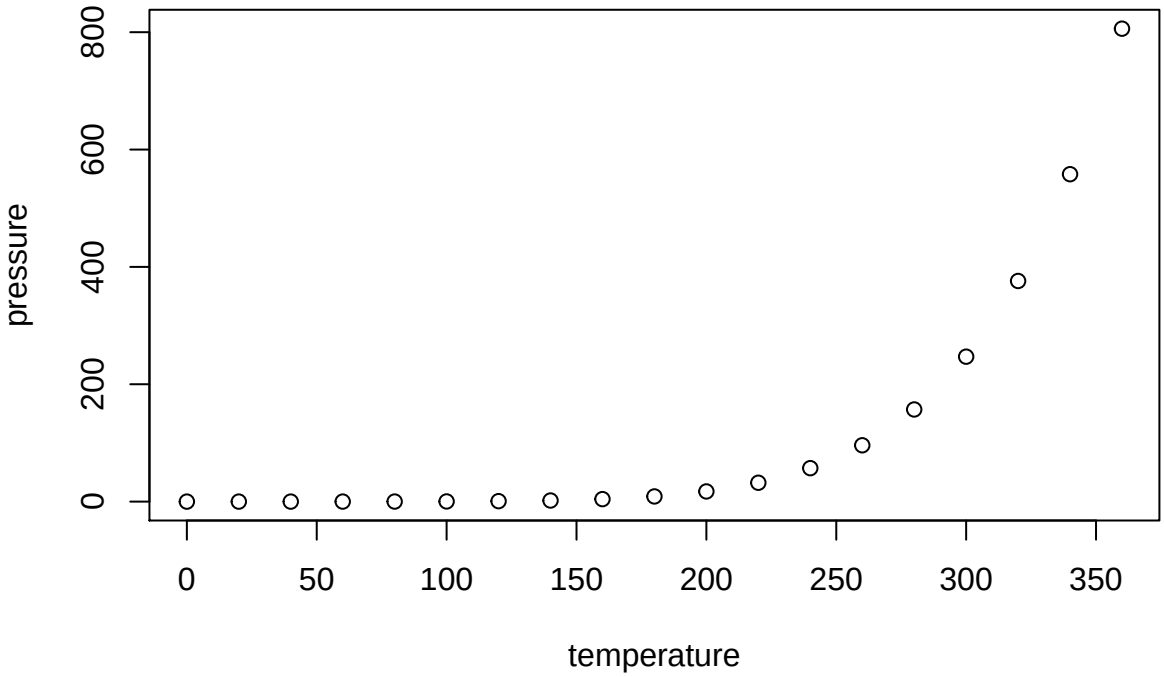
> The standard deviation is less than 6.

Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with `$2 \pi$` above, mathematics can be added by surrounding the mathematical text with dollar signs. More examples of this are in [Mathematics and Science] if you uncomment the code in [Math].

Including plots

You can also embed plots. For example, here is a way to use the base **R** graphics package to produce a plot using the built-in `pressure` dataset:



Note that the `echo=FALSE` parameter was added to the code chunk to prevent printing of the **R** code that generated the plot. There are plenty of other ways to add chunk options. More information is available at http://yihui.name/knitr/options/.

Another useful chunk option is the setting of `cache=TRUE` as you see here. If document rendering becomes time consuming due to long computations or plots that are expensive to generate you can use knitr caching to improve performance. Later in this file, you'll see a way to reference plots created in **R** or external figures.

Loading and exploring data

Included in this template is a file called `flights.csv`. This file includes a subset of the larger dataset of information about all flights that departed from Seattle and Portland in 2014. More information about this dataset and its **R** package is available at http://github.com/ismayc/pnwflights14. This subset includes only Portland flights and only rows that were complete with no missing values. Merges were also done with the `airports` and `airlines` data sets in the `pnwflights14` package to get more descriptive airport and airline names.

We can load in this data set using the following command:

```
flights <- read.csv("data/flights.csv")
```

The data is now stored in the data frame called `flights` in **R**. To get a better feel for the variables included in this dataset we can use a variety of functions. Here we can see the dimensions (rows by columns) and also the names of the columns.

```
dim(flights)
```

```
## [1] 52808    16
```

```
names(flights)
```

```
##  [1] "month"        "day"          "dep_time"     "dep_delay"
##  [5] "arr_time"     "arr_delay"    "carrier"      "tailnum"
##  [9] "flight"       "dest"         "air_time"     "distance"
## [13] "hour"         "minute"       "carrier_name" "dest_name"
```

Another good idea is to take a look at the dataset in table form. With this dataset having more than 50,000 rows, we won't explicitly show the results of the command here. I recommend you enter the command into the Console ***after*** you have run the **R** chunks above to load the data into **R**.

```
View(flights)
```

While not required, it is highly recommended you use the `dplyr` package to manipulate and summarize your data set as needed. It uses a syntax that is easy to understand using chaining operations. Below I've created a few examples of using `dplyr` to get information about the Portland flights in 2014. You will also see the use of the `ggplot2` package, which produces beautiful, high-quality academic visuals.

We begin by checking to ensure that needed packages are installed and then we load them into our current working environment:

```
# List of packages required for this analysis
pkg <- c("dplyr", "ggplot2", "knitr", "bookdown", "devtools")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg))
  install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages (thesisdown will load all of the packages as well)
library(thesisdown)
```

The example we show here does the following:

- Selects only the `carrier_name` and `arr_delay` from the `flights` dataset and then assigns this subset to a new variable called `flights2`.
- Using `flights2`, we determine the largest arrival delay for each of the carriers.

```
flights2 <- flights %>%
  select(carrier_name, arr_delay)
max_delays <- flights2 %>%
  group_by(carrier_name) %>%
  summarize(max_arr_delay = max(arr_delay, na.rm = TRUE))
```

A useful function in the `knitr` package for making nice tables in *R Markdown* is called `kable`. It is much easier to use than manually entering values into a table by copying and pasting values into Excel or LaTeX. This again goes to show how nice reproducible documents can be! (Note the use of `results="asis"`, which will produce the table instead of the code to create the table.) The `caption.short` argument is used to include a shorter title to appear in the List of Tables.

```
kable(max_delays,
      col.names = c("Airline", "Max Arrival Delay"),
      caption = "Maximum Delays by Airline",
      caption.short = "Max Delays by Airline",
      longtable = TRUE,
      booktabs = TRUE)
```

Table 1: Maximum Delays by Airline

Airline	Max Arrival Delay
Alaska Airlines Inc.	338
American Airlines Inc.	1539
Delta Air Lines Inc.	651
Frontier Airlines Inc.	575
Hawaiian Airlines Inc.	407
JetBlue Airways	273
SkyWest Airlines Inc.	421
Southwest Airlines Co.	694
United Air Lines Inc.	472
US Airways Inc.	347
Virgin America	366

The last two options make the table a little easier-to-read.

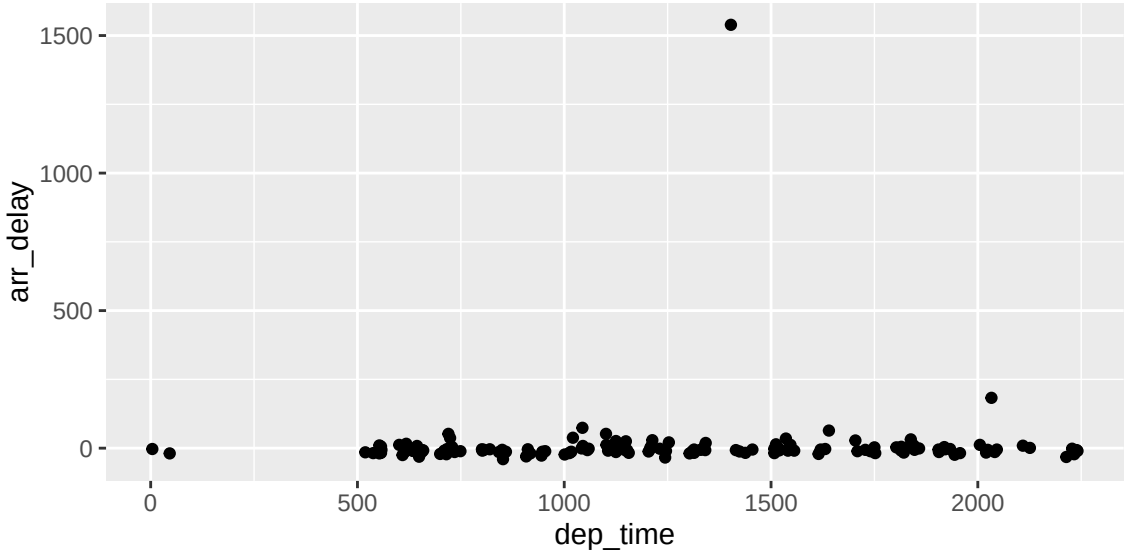
We can further look into the properties of the largest value here for American Airlines Inc. To do so, we can isolate the row corresponding to the arrival delay of 1539 minutes for American in our original `flights` dataset.

```
flights %>% filter(arr_delay == 1539,
                   carrier_name == "American Airlines Inc.") %>%
  select(-c(month, day, carrier, dest_name, hour,
            minute, carrier_name, arr_delay))
```

```
##   dep_time dep_delay arr_time tailnum flight dest air_time distance
## 1     1403      1553     1934  N595AA   1568  DFW      182     1616
```

We see that the flight occurred on March 3rd and departed a little after 2 PM on its way to Dallas/Fort Worth. Lastly, we show how we can visualize the arrival delay of all departing flights from Portland on March 3rd against time of departure.

```
flights %>% filter(month == 3, day == 3) %>%
  ggplot(aes(x = dep_time, y = arr_delay)) + geom_point()
```



Additional resources

- *Markdown* Cheatsheet - https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
- *R Markdown* Reference Guide - https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf
- Introduction to `dplyr` - https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html
- `ggplot2` Documentation - http://docs.ggplot2.org/current/