Developing Data Science Skills Using Call of Duty® Data

Matt Slifko (mds6457@psu.edu)
Department of Statistics
The Pennsylvania State University

PennState
Eberly College
of Science

Goals and Background

Introduction

The GAISE report provides many useful recommendations for statistics and data science educators including that we should

- teach statistics as an investigative process of problem-solving and decision-making
- give students experience with multivariable thinking

For both new and veteran educators, there is a strong demand for interesting new data sets and examples that we can use to promote learning.

The GOAL of this work is to share 1) resources and 2) experiences for using data from the Call of Duty franchise for developing data science skills.

Resources

Our data focuses on a player's performance in an online multiplayer match. The gameplay dataset contains a variety of pre and post game information as shown in Figure 1.

You may obtain the following resources:

- Data dictionary
- Two gameplay datasets (one semi-clean and one messy)
- Maps dataset
- Weapons dataset
- Game modes dataset

by visiting:

https://github.com/matthewdslifko/CallOfDutyProject

	Map1	Map2	Choice	MapVote	Result	Eliminations	Deaths	Score	Damage	TotalXP	XPType	GameType
1	Moscow	Miami Strike	Miami Strike	5 to 0	100–97	22	17	4070	634	11002	10% Boost	HC – TDM
2	Moscow	WMD	Moscow	2 to 0	76–89	20	15	5305	560	9451	10% Boost	HC – TDM
3	*NA*	*NA*	Yamantau	*NA*	100–92	18	11	3335	483	12948	10% Boost	HC – TDM
4	Drive-In	Jungle	Drive-In	2 to 0	80–100	10	19	2170	280	11502	Double XP + 10%	HC – TDM
5	Collateral Strike	Hijacked	Collateral Strike	3 to 3	71–100	11	19	2195	308	11133	Double XP + 10%	HC – TDM

Figure 1: Preview of gameplay datasets. 12 of 25 variables shown.

Experiences

Where have I used this data?

- Small university-level courses, ranging from 7 to 40 students
- Introductory R/Data Science courses (No prerequisites)
- Introductory Statistical Learning (R and Intro Stat perquisites)

How have I used this data?

- Used in mini-projects and final projects to practice/extend data wrangling, data visualization, and modeling concepts covered in class
- Also valuable for demonstrations during class

Suggestions:

- Due to mature subject matter of the game, I would limit use to a college classroom
- Have students work in groups initially to help those who are unfamiliar with these types of games
- Provide ample background information so that lack of familiarity with game is not a problem
- Let student expertise drive discussion

Exploratory Data Analysis

Problem

Background information: “Damage” represents the amount of damage issued by the player on the opposing team’s players, weapons, vehicles .

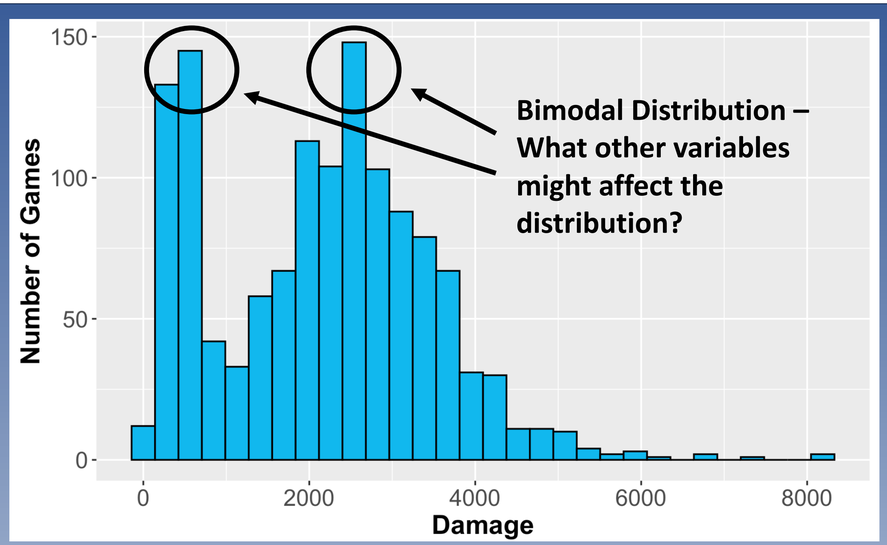
Task: Explore the distribution of Damage.

After producing Figure 2, the distribution shows a bimodal pattern. **Invite discussion as to which other variables might affect the damage dealt.**

Additional background information: Some values in the “GameType” variable include the “HC” designation. Unlike Core games, in HardCore games, players begin with less health and health does not regenerate. Thus, there is less opportunity to do damage.

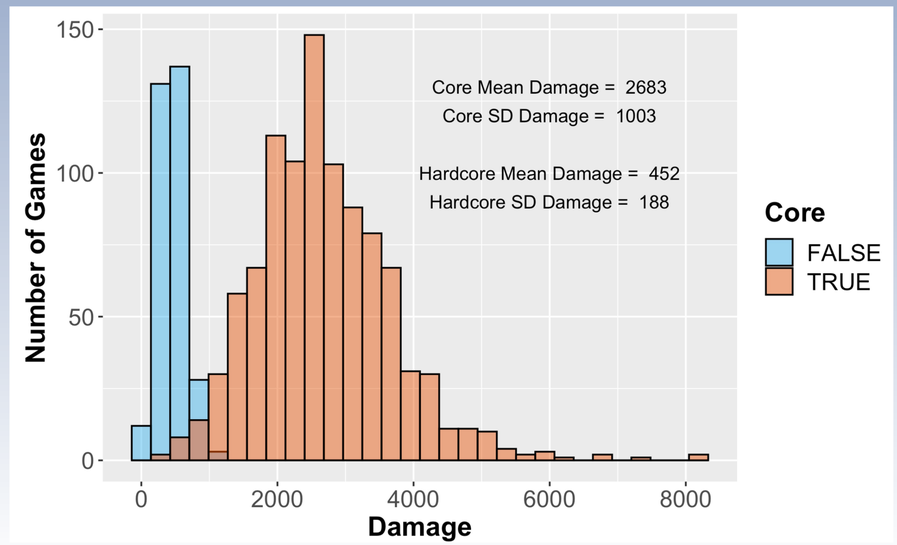
Updated Task: Explore the distribution of Damage using this information.

Figure 3 shows the updated plot. Summary statistics are added to support the visualization.



(Above) Figure 2: Distribution of damage variable.

(Below) Figure 3: Distribution of damage variable incorporating Core



Discussion

This example provides an opportunity to discuss:

- Data visualization
- Process of exploration
- Complementary roles of visualization and summary statistics
- Annotating plot with text
- String processing and new variable creation (to create Core variable)
- Multivariable thinking

Processing Character Data

Problem

Background information: Each online match consists of 2 competing teams that earn points for completing tasks. The “Result” variable shows the team scores at the end of the match. The player’s team score is listed first and the team with the higher score is considered the winner. Examples:

- Row 1: “100-97” means that the player’s team won by a score of 100 to 97
- Row 2: “76-89” means that the player’s team lost by a score of 89 to 76
- Row 36: “200-200” means that the teams played to a draw (i.e., tied 200 to 200)

Research Question: Is the player’s team more likely to win or lose? To answer the question, support your conclusion visually and numerically.

Table shows Result and 3 newly created variables. **There appears to be a logical error. Just because code ran without error does not make it flawless**. Invite discussion as to the cause of the mistake.

Figure 4 and Table 2 show the distribution of wins, losses, and draws based on the corrected results and lead us to a conclusion.

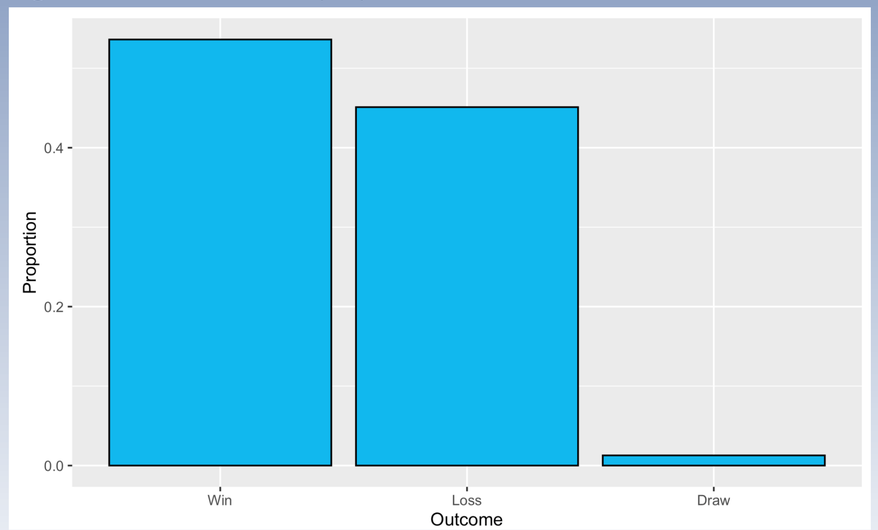
Table 1: Splitting ‘Result’ into new variables has not worked correctly.

	Result <chr>	PlayerScore <chr>	OpponentScore <chr>	PlayerResult <chr>
1	100-97	100	97	Loss
2	76-89	76	89	Loss
3	100-92	100	92	Loss
4	80-100	80	100	Win
5	71-100	71	100	Win
6	85-100	85	100	Win

6 rows

Some results are incorrect. What is happening?

Figure 4: Distribution of player’s results.



Discussion

This example provides an opportunity to discuss:

- String processing
- New variable creation
- If/else with > 2 conditions
- Importance of manually checking results and considering data types
- Data visualization
- Restructuring data for visualization
- Reordering factors for visualization

Bonus discussion: Repeating the activity on other gameplay dataset presents challenges when some Result values do not have a hyphen.

Bonus discussion: Do you think the difference in proportion of wins and losses is anything more than random noise?

Table 2: Proportions of wins, losses, and draws.

Outcome <fctr>	Proportion <dbl>	N <int>
Win	0.53617021	126
Loss	0.45106383	106
Draw	0.01276596	3

3 rows

Combining Data from Multiple Tables

Problem

Background information: The variable “Score” represents points earned by the player for actions in a match. For instance, capturing an enemy location earns 200 score points and eliminating an opponent earns 50.

Research Question: For complete matches, does the player’s score depend on the weapons class used?

Additional background: The “FullPartial” variable indicates whether the player participated in the complete match or only a partial match. (It is possible that a player is assigned to a game that is in progress and thus only plays a partial match.) For each match, the player selects a primary weapon to use, as indicated by the ”PrimaryWeaon” variable. As there are many weapons available, it might make more sense to examine the weapon class (e.g., shotgun, sniper rifle, melee, etc.) The weapon classes may be found in the CODWeapons dataset, as shown in Figure 5. Perform an appropriate join to bring the “Class” variable into the gameplay dataset and create an appropriate visualization.

Figure 6 shows side-by-side boxplots of score based on weapon class. Why is the boxplot for LMG a single line? Do you think the score depends on the weapon class?

Figure 5: Preview of Weapons dataset.

	Weapon	Class
1	Type 63	Tactical Rifle
2	M16	Tactical Rifle
3	AUG	Tactical Rifle
4	DMR 14	Tactical Rifle
5	CARV.2	Tactical Rifle
6	Nail Gun	Special Weapon
7	M79	Special Weapon
8	R1 Shadowhunter	Special Weapon
9	Ballistic Knife	Special Weapon
10	Ray Gun	Special Weapon
11	Pelington 703	Sniper Rifle

Figure 6: Score by weapon class. Diamonds depict the mean.

Discussion

This example provides an opportunity to discuss:

- Joining datasets (with no common variable names)
- Filtering (to remove partial matches)
- Data visualization
- Adding additional “layers” to visualization
- Categories with small *n* (for LMG, *n = 1*)
- Connection between mean and median in skewed distributions

Bonus discussion: **If you repeat this activity on the messier gameplay dataset, there are some additional challenges**. Some of the values for PrimaryWeapon are not found in the Weapons dataset. This is often the result of typos. Discussion points:

- How will this be handled?
- How do different types of joins handle the situations when there is no match?

Linear Regression, Indicators, and Interaction 1

Problem

Background information: Score, along with other factors not necessarily included in the dataset, is used to determine the eXperience Points (XP) earned.

Research Question: What is the relationship between the player's score the XP earned?

Figure 7 suggests that a linear relationship is plausible ($R^2 \approx 0.33$), but there might be some outliers with values of XP above 30000.

Multivariable thinking: What other variables might also influence XP?

Additional background: Players can earn and use tokens that grant double XP for short periods of time (e.g., 30 minutes). The "XPType" variable provides information for when the player used a token.

Figure 8 shows the residuals vs. predicted values for the model of XP using score. The points are coded by XPType. What do you notice?

Figure 9 updates Figure 7. How can we update the model based on this information? $R^2_{adj} \approx 0.66$

Figure 7: Experience points earned as a function of score

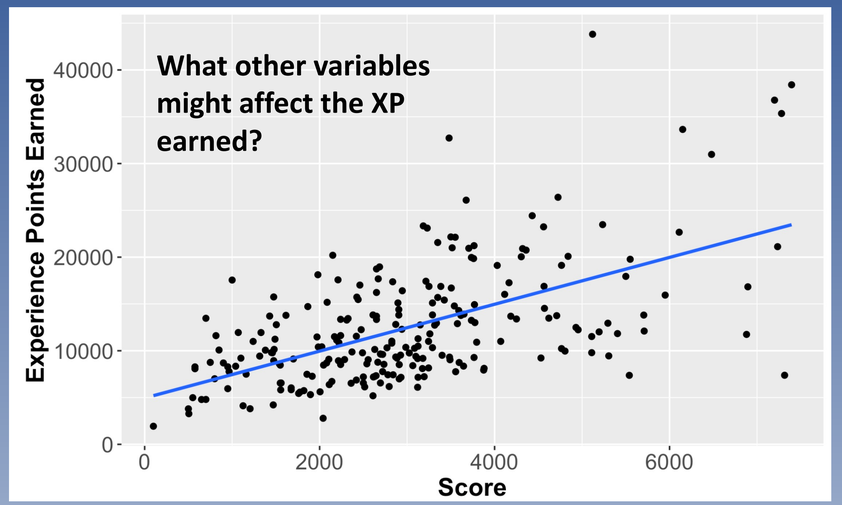


Figure 8: Residuals vs. predicted values coded by Type of XP

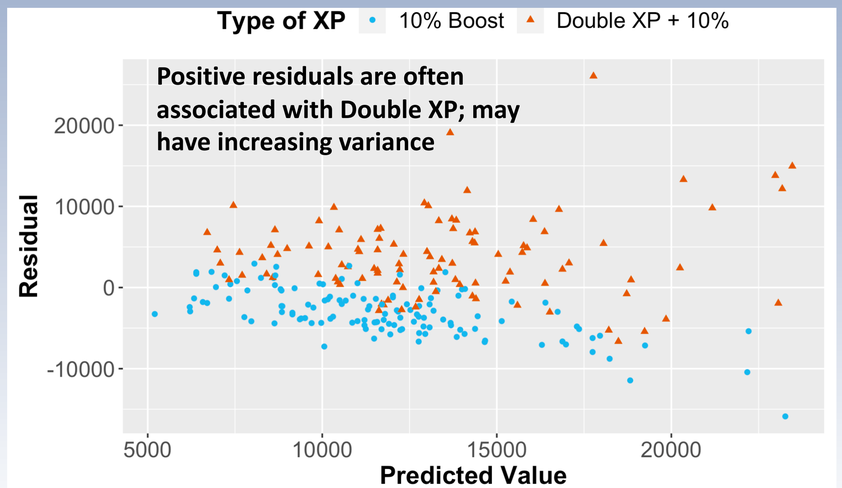


Figure 9: XP earned as function of score coded by type of XP

Discussion

This example provides an opportunity to discuss:

- Linear regression
- Residuals
- Checking regression assumptions
- Multivariable thinking and lurking variables
- Indicator variables
- Interaction terms
- Comparing models
- Statistical significance
- Investigation of outliers

Linear Regression, Indicators, and Interaction 2

Discussion

Even when we understand the methodology, there are frequently challenges that arise when implementing the methods.

Suppose we switch to the messy gameplay dataset. Several issues arise.

1. Applying the code that worked for the examples on the last page leads to an error, as shown in Figure 10. (This is due to a missing data problem.)

2. Creating the same visualization as before reveals that the type of XP has more than 2 levels as shown in Figure 11. The additional levels occur because of inconsistent naming and typographical errors.

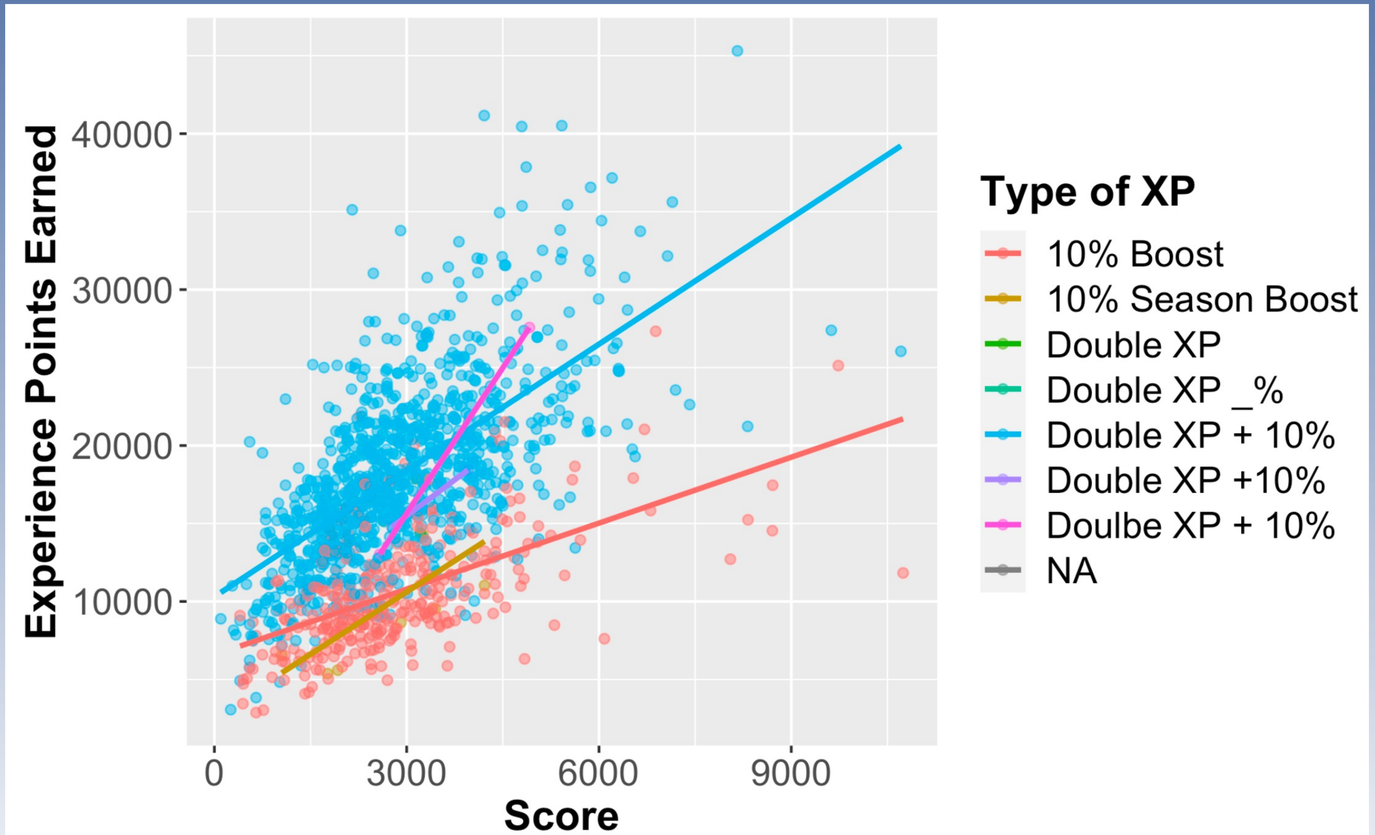
3. If we trained the model on the first (clean dataset from last slide) but tried to obtain predictions for new data (messy dataset), there would be new levels of “XPType” not seen in the train.

There are plenty to challenges to discuss!

Figure 10: Changing dataset leads to error message that previously worked.

```
Error in data.frame(Residuals = model3b$residuals, yhat = model3b$fitted.values,  :
  arguments imply differing number of rows: 1309, 1464
```

Figure 11: XP earned as function of score coded by type of XP



Thanks for your time!

Contact info:
Matt Slifko (mds6457@psu.edu)
Department of Statistics
The Pennsylvania State University

If you are interested in the resources, please visit:
https://github.com/matthewdslifko/CallOfDutyProject

PennState
Eberly College
of Science