

Predicting Weekly Revenue Sales & Number of Weekly games for Publisher

By Group - 1

​

​**Team Members :**

**Sindhura Alla**

**Ameer Mittigar**

**Arshad Abdullah Mohammad**

**Manikantha Varaprasad Inakollu**

CONTEXT

[1. EXECUTIVE SUMMARY: 3](#_Toc134062050)

[2. PROBLEM DEFINITION AND SIGNIFICANCE: 3](#_Toc134062051)

[**3.** **METHODOLOGY USED:** 4](#_Toc134062052)

[**4.** **FEATURE ENGINEERING:** 4](#_Toc134062053)

[**5.** **PRIOR LITERATURE:** 4](#_Toc134062053)

[6. DATA SOURCE: 4](#_Toc134062054)

[7. VARIABLE CHOICE: 5](#_Toc134062055)

[8. DESCRIPTIVE ANALYSIS & DATA VISUALIZATIONS: 6](#_Toc134062056)

[a) CORRELATION POINT: 6](#_Toc134062057)

[b) BOX-AND-WHISKER PLOT: 6](#_Toc134062058)

[c) HISTOGRAM OF PROBABILITY RESIDUALS: 6](#_Toc134062059)

[9. MODELS: 8](#_Toc134062060)

[10. STARGAZER OUTPUT: 9](#_Toc134062061)

[11. INSIGHTS AND RECOMMENDATIONS: 10](#_Toc134062063)

[12. LIMITATIONS: 10](#_Toc134062064)

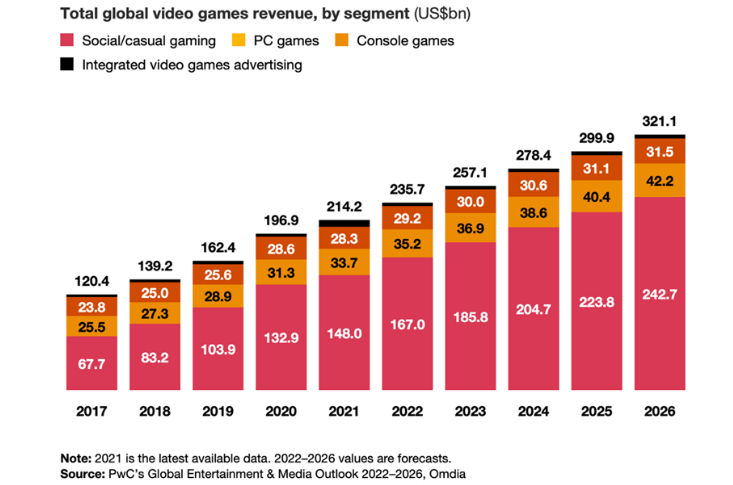
[13. APPENDIX: 11](#_Toc134062065)

[R CODE: 11](#_Toc134062066)

# 

# EXECUTIVE SUMMARY:

According to PwC’s Global Entertainment & Media Outlook, the global video game market was valued at approximately $162 billion in 2019, and revenues are expected to exceed $250 billion by 2023.



In this competitive market, publishers must continually innovate, develop new games, and adapt to changing market conditions to survive and be profitable. Several factors contribute to the success of video game publishers, including the quality of their games, marketing strategies, and the platforms they choose to release their games. This is where the world of Statistics enters and helps the players with the kind of insights that they need to make forecasts, plan resources, and optimize their marketing efforts. The critical findings generated with this study can significantly benefit both video game publishers and other relevant players in the industry in making informed decisions regarding resource allocation, marketing strategies, and game development. Accurately predicting game counts may help publishers better manage inventory, ensuring that popular games are always available and reducing the likelihood of stock shortages. This can ultimately lead to increased profitability and success in the highly competitive market.

# PROBLEM DEFINITION AND SIGNIFICANCE:

In an industry like Video gaming, publishers play a critical role in the success of video games because they provide the financial resources and expertise needed to bring fun to market. In addition, they are responsible for managing the development process, ensuring that games are completed on time and within budget, and coordinating with marketing and distribution partners to ensure the games reach the broadest possible audience. Therefore, Statistics and predictions are essential for publishers in the video gaming industry because they help them make informed business decisions.

By analyzing data on revenue, game counts, genre, Life to date sales, publishers can identify patterns and trends that inform their marketing, sales, and development strategies. Upon research, it has come to light that it has been a challenge “to predict revenue generated by the game accurately. Accurately predicting game count can help publishers better manage inventory, ensuring that popular games are always available and reducing the likelihood of stock shortages”*.* This is an exciting and essential problem because the video gaming industry is highly competitive and rapidly growing. Inaccurate predictions can lead to problems, including overproduction of specific games, which can lead to inventory stockpiles and lower profits, or underproduction of certain games, resulting in missed revenue opportunities and dissatisfied customers.

## **METHODOLOGY USED:**

Predictive models like the random effects model and linear mixed effects allow publishers to identify which factors are most influential in running their business and driving decisions, allowing them to focus on areas with tremendous growth potential. These predictions enable publishers to make more accurate revenue forecasts, plan budgets, and allocate resources more effectively.

Running with the idea and based on the prior research, with their historical data, predicting their revenue and game count helps publishers make improved business decisions, assess the overall performance, and better understand the impact of various factors on game counts, including past performance, genre trends, and pricing strategy.

## **FEATURE ENGINEERING:**

* The data has been scrapped from the Famitsu\_sales website.
* Read data from Excel into data frames.
* Excel data is already organized as time series (easy to reshape in scraping code using pandas python)
* The time column combines the year and week\_no columns into a single string with a dash separator. This creates a column that represents the period of each row in a standardized format.
* Two new columns, ranked\_tf and ranked, indicate whether a publisher reached the ranking each week.
* This is done by checking if the game\_count column is greater than 0. ranked\_tf is a character column that contains "True" or "False" values, while ranked is a numeric column that includes 1 for "True" and 0 for "False.”
* The count\_df data frame is created by selecting the publisher and time columns from the df data frame where the ranked queue is 1, indicating that the publisher made it to the ranking.
* The rows are then grouped by the publisher and counted to determine how many times each publisher made it to the order out of all weeks (405).

1. PRIOR LITERATURE:

Researchers Hsiao, Wu, and Chen proposed a hybrid forecasting model for video game sales that combines various techniques, including linear regression, time series analysis, and artificial neural networks. Real-world data from a popular video game was used to test the model, and the results showed that the hybrid model performed better than individual forecasting methods.

In "Forecasting Sales of Video Games Using ARIMA and GARCH Models" (2020), J. F. Rodríguez and E. López-López studied the effectiveness of autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models for forecasting video game sales. The authors collected sales data and used these models to predict sales for the following year. The study revealed that both models were effective, but the GARCH model outperformed the ARIMA model in accuracy.

# DATA SOURCE:

Famitsu,[a] formerly Famicom Tsūshin, is a line of Japanese video game magazines published by Kadokawa Game Linkage (previously known as Gzbrain), a subsidiary of Kadokawa. Famitsu is published weekly and monthly as special topical issues devoted to only one console, video game company, or another theme. Shūkan Famitsū, the original Famitsu publication, is Japan's most widely read and respected video game news magazine. From October 28, 2011, the company began releasing the digital version of the magazine exclusively on BookWalker weekly.

The name Famitsu is a portmanteau abbreviation of Famicom Tsūshin; the word "Famicom" itself comes from a portmanteau abbreviation of "Family Computer" (the Japanese name for the Nintendo Entertainment System)—the dominant video game console in Japan during the 1980s.

1. **VARIABLE CHOICE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VARIABLE** | **DESCRIPTION** | **AFFECT** | **RATIONALE** |
| publisher | The publisher of the game | + | Publisher may influence a game's success based on reputation and marketing strategies. |
| year | The year the data was recorded | + | Time can be a crucial factor in determining a game's success. |
| week\_no | The week number the data was recorded | + | Time can be a crucial factor in determining a game's success. |
| revenue | The revenue generated by the publisher in a particular week | + | Revenue is a key indicator of a game's success. |
| game\_Count | The number of games published by the publisher in a particular week | + | The number of games published may impact a publisher's overall success. |
| genre\_Count | The number of different game genres published by the publisher in a particular week | + | The variety of genres published may impact a publisher's overall success. |
| avg\_price | The average price of the games published by the publisher in a particular week | - | Price can be a factor in a game's success, as consumers may be more likely to purchase games at a lower price point. |
| std\_price | The standard deviation of game prices published by the publisher in a particular week | - | The range of prices published may impact a publisher's overall success. |
| time | The time period (year and week) the data was recorded | + | Time can be a crucial factor in determining a game's success. |
| ranked | Whether or not a game was ranked in the Japanese market in a particular week | + | The presence or absence of a game in the rankings can indicate its success. |
| ranked\_last\_week | Whether or not a game was ranked in the Japanese market in the previous week | + | The previous week's ranking can be an indicator of a game's potential success in the current week. |
| ranked\_last\_month | Whether or not a game was ranked in the Japanese market in the previous month | + | The previous month's ranking can be an indicator of a game's potential success in the current week. |
| ranked\_last\_year | Whether or not a game was ranked in the Japanese market in the previous year | + | The previous year's ranking can be an indicator of a game's potential success in the current week. |
| Variable Created | | | |
| prob\_residual | The difference between a game's actual ranking and its predicted ranking based on the model | - | This variable is used to measure how well the model is predicting game rankings, and may be used as a predictor in other models. |
| DEMOGRAPHIC VARIABLES | | | |
| genre | The genre of the game | +/- | Different game genres may appeal to different audiences, impacting the game's success. For example, action games may be more popular with younger audiences, while simulation games may appeal more to older audiences. |

1. **DESCRIPTIVE ANALYSIS & DATA VISUALIZATIONS:**

Chart, histogram

Description automatically generatedThe graph depicts the number of weeks the publisher made to rank out of 405 weeks.

**8.1CORRELATION POINT:**

****

Referring to the correlation plot, the week\_no and year have a slight relationship, and genre\_count and avg\_price do have a relationship.

**8.2 BOX-AND-WHISKER PLOT:**

**Ranked\_Last\_Week**

** Random effects Vs. Publisher**

****

* 1. **HISTOGRAM OF PROBABILITY RESIDUALS:**

****

# **MODELS:**

**Model 1:**

model <- glmer(ranked ~ ranked\_last\_week + ranked\_last\_month + ranked\_last\_year + (1 | publisher),

+ data = df, family=binomial())

**Model 2:**

game\_count\_model = lmer(game\_count ~ game\_count\_last\_week + game\_count\_last\_month + game\_count\_last\_year +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked, na.action = "na.fail")

**Model 3:**

best\_game\_count\_model = lmer(game\_count ~ game\_count\_last\_week + game\_count\_last\_month + game\_count\_last\_year +

genre\_count\_last\_week +

avg\_price\_last\_week + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked)

**Model 4:**

revenue\_model = lmer(revenue ~ revenue\_last\_week + revenue\_last\_month + revenue\_last\_year +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked, na.action = "na.fail")

**Model 5:**

best\_revenue\_model = lmer(revenue ~ revenue\_last\_week + revenue\_last\_month +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df)

# **STARGAZER OUTPUT:**

# Table Description automatically generated

# **INSIGHTS AND RECOMMENDATIONS:**

* The visualization shows that only a few publishers have consistently made it to the ranking throughout the 405 weeks.
* The correlation plot indicates no strong correlation between revenue and week number or year.
* Adding lagged variables helps us capture trends and patterns over time. We can use these variables to predict the probability of a publisher making it to the ranking in the future.
* Publishers who consistently make it to the ranking could be targeted for partnerships or collaborations. This could increase revenue and visibility for both parties.
* The lack of a strong correlation between revenue and time could indicate that publishers should focus on factors other than time, such as the quality of games, marketing, or pricing strategies, to increase revenue.
* The use of lagged variables can help to predict the probability of a publisher making it to the ranking in the future. This information can inform business decisions such as product development, marketing, and partnerships.

# **LIMITATIONS:**

* Correlation vs. causation: Correlation between the dependent and independent variables does not prove causation. The relationship between the variables may be influenced by other factors not included in the analysis.
* Omitted variable bias: The output does not include all potential variables that could affect the dependent variable. Therefore, the coefficients estimated for the independent variables could be biased.
* Endogeneity: The independent variables are exogenous, meaning that the dependent variable does not affect them. However, this may not be the case, and the estimated coefficients may be biased.
* Model assumptions: The stargazer output assumes that the data meet certain statistical assumptions, such as normality and linearity. Violations of these assumptions could lead to biased estimates.

# APP**ENDIX:**

# **R CODE:**

library(rio)

library(readxl)

library(reshape2)

library(dplyr)

library(tidyr)

library(openxlsx)

library(PanelCount)

library(lme4)

library(zoo)

library(MuMIn)

library(lattice)

############################### Cleaning and Reforming Data###############################

# Read data from Excel into data frames.

# Excel data is already organized as time series(easy to reshape in scraping code - pandas python)

file\_name <- '/Users/sindhu/Desktop/USF/SDM/data\_publisher\_panel.xlsx'

df\_revenue <- read\_excel(file\_name, sheet="Revenue(Mil)");

df\_game\_count <- read\_excel(file\_name, sheet="Count")

df\_genre\_count <- read\_excel(file\_name, sheet="GenreCount")

df\_avg\_price <- read\_excel(file\_name, sheet="AvgPrice")

df\_std\_price <- read\_excel(file\_name, sheet="StdPrice")

# Combining data using merge and reduce

df\_list <- list(df\_revenue, df\_game\_count, df\_genre\_count, df\_avg\_price, df\_std\_price)

df <- Reduce(function(x, y) merge(x, y, by = c("publisher", "year", "week\_no"), all = TRUE), df\_list)

head(df)

publisher year week\_no revenue game\_count genre\_count avg\_price std\_price

1 3goo 2015 21 0 0 0 0 0

2 3goo 2015 23 0 0 0 0 0

3 3goo 2015 24 0 0 0 0 0

4 3goo 2015 25 0 0 0 0 0

5 3goo 2015 26 0 0 0 0 0

6 3goo 2015 27 0 0 0 0 0

# Reshaping to long format with metric as column

df\_reshaped <- melt(df, id.vars = c("publisher", "year", "week\_no"),

variable.name = "metric", value.name = "value")

head(df\_reshaped)

publisher year week\_no metric value

1 3goo 2015 21 revenue 0

2 3goo 2015 23 revenue 0

3 3goo 2015 24 revenue 0

4 3goo 2015 25 revenue 0

5 3goo 2015 26 revenue 0

6 3goo 2015 27 revenue 0

# For analysis, choose df as a data frame with a column in the headers

# Combining year and week in a single column

df$time <- paste(df$year, sprintf("%02d", df$week\_no), sep = "-")

head(df)

publisher year week\_no revenue game\_count genre\_count avg\_price std\_price time

1 3goo 2015 21 0 0 0 0 0 2015-21

2 3goo 2015 23 0 0 0 0 0 2015-23

3 3goo 2015 24 0 0 0 0 0 2015-24

4 3goo 2015 25 0 0 0 0 0 2015-25

5 3goo 2015 26 0 0 0 0 0 2015-26

6 3goo 2015 27 0 0 0 0 0 2015-27

write.xlsx(df, file = "data/long\_output.xlsx", rowNames = FALSE)

############################Add lagged variables based on existing columns###############################

# Determine whether the publisher made it to ranking in week

df$ranked\_tf <- ifelse(df$game\_count > 0, 'True', 'False')

df$ranked <- ifelse(df$game\_count > 0, 1, 0)

# Number of times publisher is in ranking past week, month, year

df <- df %>%

group\_by(publisher) %>%

mutate(ranked\_last\_week = lag(ranked, n = 1, default = 0),

ranked\_last\_month = rollapplyr(ranked, width = 5, FUN = sum, fill = 0, align = "right"),

ranked\_last\_year = rollapplyr(ranked, width = 53, FUN = sum, fill = 0, align = "right")) %>%

ungroup()

# Exclude the current row from the rolling sum

df$ranked\_last\_month = df$ranked\_last\_month - df$ranked

df$ranked\_last\_year = df$ranked\_last\_year - df$ranked

df[df$ranked\_last\_month < 0, "ranked\_last\_month"] = 0

df[df$ranked\_last\_year < 0, "ranked\_last\_year"] = 0

# Number of games released by publisher with in the past week, month, year

df <- df %>%

group\_by(publisher) %>%

mutate(game\_count\_last\_week = lag(game\_count, n = 1, default = 0),

game\_count\_last\_month = rollapplyr(game\_count, width = 5, FUN = sum, fill = 0, align = "right"),

game\_count\_last\_year = rollapplyr(game\_count, width = 53, FUN = sum, fill = 0, align = "right")) %>%

ungroup()

# Exclude the current row from the rolling sum

df$game\_count\_last\_month = df$game\_count\_last\_month - df$game\_count

df$game\_count\_last\_year = df$game\_count\_last\_year - df$game\_count

df[df$game\_count\_last\_month < 0, "game\_count\_last\_month"] = 0

df[df$game\_count\_last\_year < 0, "game\_count\_last\_year"] = 0

# Number of distinct genres for a publisher within the past week, month, year

df <- df %>%

group\_by(publisher) %>%

mutate(genre\_count\_last\_week = lag(genre\_count, n = 1, default = 0),

genre\_count\_last\_month = rollapplyr(genre\_count, width = 5, FUN = sum, fill = 0, align = "right"),

genre\_count\_last\_year = rollapplyr(genre\_count, width = 53, FUN = sum, fill = 0, align = "right")) %>%

ungroup()

# Exclude the current row from the rolling sum

df$genre\_count\_last\_month = df$genre\_count\_last\_month - df$genre\_count

df$genre\_count\_last\_year = df$genre\_count\_last\_year - df$genre\_count

df[df$genre\_count\_last\_month < 0, "genre\_count\_last\_month"] = 0

df[df$genre\_count\_last\_year < 0, "genre\_count\_last\_year"] = 0

# Avg revenue for a publisher within the past week, month, year

df <- df %>%

group\_by(publisher) %>%

mutate(revenue\_last\_week = lag(revenue, n = 1, default = 0),

revenue\_last\_month = rollapplyr(revenue, width = 5, FUN = sum, fill = 0, align = "right"),

revenue\_last\_year = rollapplyr(revenue, width = 53, FUN = sum, fill = 0, align = "right")) %>%

ungroup()

# Exclude current row for rolling mean of revenue in last month, year

df$revenue\_last\_month = (df$revenue\_last\_month - df$revenue) / 4

df$revenue\_last\_year = (df$revenue\_last\_year - df$revenue) / 52

df[df$revenue\_last\_month < 0, "revenue\_last\_month"] = 0

df[df$revenue\_last\_year < 0, "revenue\_last\_year"] = 0

# Avg price of all games for a publisher within the past week, month, year

df <- df %>%

group\_by(publisher) %>%

mutate(avg\_price\_last\_week = lag(avg\_price, n = 1, default = 0),

avg\_price\_last\_month = rollapplyr(avg\_price, width = 5, FUN = sum, fill = 0, align = "right"),

avg\_price\_last\_year = rollapplyr(avg\_price, width = 53, FUN = sum, fill = 0, align = "right")) %>%

ungroup()

# Exclude current row for rolling mean of avg\_price in last month, year

df$avg\_price\_last\_month = (df$avg\_price\_last\_month - df$avg\_price) / 4

df$avg\_price\_last\_year = (df$avg\_price\_last\_year - df$avg\_price) / 52

df[df$avg\_price\_last\_month < 0, "avg\_price\_last\_month"] = 0

df[df$avg\_price\_last\_year < 0, "avg\_price\_last\_year"] = 0

df = replace(df, is.na(df), 0)

############################### Ranking mixed effects model ###############################

# Predict whether a publisher can make it to ranking using random effects logistic regression

# glmer (logistic regression + random effects (group correlation))

# Plot relation between ranked vs ranked\_last\_week, ranked\_last\_month, ranked\_last\_year

bwplot(ranked\_tf ~ ranked\_last\_week | publisher, data=df)

bwplot(ranked\_tf ~ ranked\_last\_month | publisher, data=df)

bwplot(ranked\_tf ~ ranked\_last\_year | publisher, data=df)

model <- glmer(ranked ~ ranked\_last\_week + ranked\_last\_month + ranked\_last\_year + (1 | publisher),

data = df, family=binomial())

summary(model)

print(fixef(model)) # Fixed effects from the model

(Intercept) ranked\_last\_week ranked\_last\_month ranked\_last\_year

-4.32213764 1.74025824 0.49513036 0.02238743

print(ranef(model)) # Random effects from the model

$publisher

(Intercept)

3goo 0.28754420

505 Games -1.21169458

5pb. 1.12613445

8-4 -0.83885077

Acquire -0.34139191

Active Gaming Media -0.83884101

Activision 0.11334046

Alchemist -0.84602641

Alfa System -1.21185966

Aniplex 0.80041495

Aqua Plus 0.81389989

Arc System Works 1.19091649

Artdink 0.20140265

Asgard -1.20246433

Atlus 1.64522370

Bandai Namco Entertainment 1.55488282

Bandai Namco Games 2.18485740

Beep Japan -0.55993368

Bergsala-Lightweight -1.21367330

Bethesda Softworks 1.28901556

Blizzard Entertainment -0.83885272

Broccoli 0.70349430

Bushiroad -0.34425087

Capcom 2.19834613

CFK -1.21330266

City Connection -0.02291276

Clouded Leopard Entertainment -0.18194971

Codemasters -1.21367330

Compile Heart 1.35033188

Cygames 0.22400411

D3Publisher 1.46938768

Daewon Media -1.21367330

Deep Silver -0.83839835

Delight Works -1.21367330

DMM Games 0.11614436

Dramatic Create 0.21451597

Edia -1.21297306

El Dia -0.56117670

Electronic Arts 1.88401174

Entergram 0.90971322

Epic Games 0.45762185

EXNOA 0.28617639

Experience 0.38652560

Extreme -1.21150242

Fangamer -1.21256086

Flyhigh Works -1.21367330

Freestyle -1.21367330

From Software 1.33765763

FuRyu 1.49023277

Game Addict -0.20429698

Game Source Entertainment 0.08037107

Gemdrops -1.21247840

Granzella -0.20154666

GungHo Online Entertainment 1.18400111

H2 Interactive 0.35946885

Happinet 0.32340949

Harukaze -1.21367330

Honeybee -1.21179134

Honeybee Black -0.83839835

Hudson -0.07530723

Idea Factory 1.74012425

Imagineer 1.48944474

iMel -1.21367330

Intergrow -0.55897310

Inti Creates 0.47042751

Izanagi Games -0.83873353

Justdan -1.21289063

Kadokawa Games 1.28400905

Kaga Create -1.21170880

Kakehashi Games -1.21367330

Koch Media -0.84862406

Koei Tecmo 1.89444131

Konami 2.20187411

Level 5 1.94121217

Mages. 0.27305641

Marvelous 1.75661956

Mastiff -1.21367330

Matatabi -0.55878050

Mebius -1.21326146

Mediascape -0.83839835

Microsoft Game Studios 2.37618166

Mixi 0.55924953

Moss -0.83839835

NA Publishing -1.21367330

Natsume Atari -0.20334364

Neos 0.13057969

Nihon Falcom 1.40820973

Nintendo 4.96510484

Nippon Columbia 1.58549579

Nippon Cultural Broadcasting Extended -0.83884101

Nippon Ichi Software 1.52054158

Oizumi Amuzio -0.34663954

Onion Games -1.21367330

Piacci -0.83879153

Pikii -1.21367330

Planet G -0.84980159

Playism 0.52363706

Pocket -1.21367330

Pokemon Co. 3.36941180

Ponos 0.87160174

Prototype 0.80964147

PUBG Corporation -0.34724864

QuinRose -1.21170880

Red Flagship -1.21265750

Reef Entertainment -1.21367330

Rejet -0.34366539

Rocket Company -0.23724223

Sega 1.92914673

Shogakukan -1.21367330

Silky`s -1.21170880

SNK 0.27244609

SNK Playmore -0.59856073

Sony Computer Entertainment 1.70380510

Sony Interactive Entertainment 2.02184101

Spike Chunsoft 1.80978375

Sprite -0.83883321

Square Enix 2.41708742

Supergiant Games -1.21367330

SystemSoft Alpha -0.56059860

Tabinomichi -1.21260209

Taito -0.02826598

Takara Tomy 0.83177485

Take-Two Interactive Japan 1.81342209

Takuyo -1.21367330

Technical Group Laboratory -0.33718146

Teyon Japan 1.42428496

THQ Nordic Japan -0.59856073

Tozai Games -1.21367330

Ubisoft 1.67878242

Unties -0.56180351

Warner Entertainment Japan 1.41622289

WB Games 0.19112311

WSS Playground -0.84775040

with conditional variances for “publisher.”

random\_effects = ranef(model)$publisher

random\_effects\_df <- data.frame(Publisher = rownames(random\_effects), Intercept = random\_effects[, 1])

bwplot(Intercept ~ Publisher, data = random\_effects\_df, xlab = "Publisher", ylab = "Random Effects")

# Get predicted probabilities from the model and residuals

df$predicted\_prob <- predict(model, type="response")

df$prob\_residual <- df$ranked - df$predicted\_prob

############################### Game count random effects model ###############################

# Subset of data frames with publishers that are ranked in a given week

df\_ranked <- df[df$ranked == 1, ]

# Random effects model on publisher game count

df\_ranked <- na.omit(df\_ranked)

# Scale price to avoid scaling issues on models

df\_ranked["avg\_price\_last\_week"] <- df\_ranked["avg\_price\_last\_week"] / 1000

df\_ranked["avg\_price\_last\_month"] <- df\_ranked["avg\_price\_last\_month"] / 1000

df\_ranked["avg\_price\_last\_year"] <- df\_ranked["avg\_price\_last\_year"] / 1000

game\_count\_model = lmer(game\_count ~ game\_count\_last\_week + game\_count\_last\_month + game\_count\_last\_year +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked, na.action = "na.fail")

summary(game\_count\_model)

REML criterion at convergence: 13373.7

Scaled residuals:

Min 1Q Median 3Q Max

-6.9510 -0.3626 -0.1119 0.2472 7.8603

Random effects:

Groups Name Variance Std. Dev.

publisher (Intercept) 0.09895 0.3146

Residual 0.65963 0.8122

Number of obs: 5431, groups: publisher, 134

Fixed effects:

Estimate Std. Error t value

(Intercept) 0.4070109 0.1118914 3.638

game\_count\_last\_week 0.3830932 0.0213260 17.964

game\_count\_last\_month 0.0700494 0.0074666 9.382

game\_count\_last\_year 0.0034962 0.0008148 4.291

genre\_count\_last\_week 0.1486979 0.0397909 3.737

genre\_count\_last\_month -0.0210713 0.0137974 -1.527

genre\_count\_last\_year -0.0014819 0.0015388 -0.963

avg\_price\_last\_week -0.0475450 0.0086490 -5.497

avg\_price\_last\_month -0.0024816 0.0118719 -0.209

avg\_price\_last\_year -0.0271507 0.0126701 -2.143

**prob\_residual 0.8401442 0.1093402 7.684**

# t-value of prob\_residual is significant in game\_count\_model, indicating that there is self-selection, i.e., second stage model

# is substantially affected by residuals of the first-stage model

# Use dredge to get the best model based on AIC criteria

game\_count\_model\_comparison <- dredge(game\_count\_model)

print(game\_count\_model\_comparison[which.min(game\_count\_model\_comparison$AICc), ] )

# Best model based on dredge

best\_game\_count\_model = lmer(game\_count ~ game\_count\_last\_week + game\_count\_last\_month + game\_count\_last\_year +

genre\_count\_last\_week +

avg\_price\_last\_week + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked)

summary(best\_game\_count\_model)

REML criterion at convergence: 13353.7

Scaled residuals:

Min 1Q Median 3Q Max

-6.9640 -0.3612 -0.1111 0.2533 7.9305

Random effects:

Groups Name Variance Std. Dev.

publisher (Intercept) 0.0991 0.3148

Residual 0.6599 0.8123

Number of obs: 5431, groups: publisher, 134

Fixed effects:

Estimate Std. Error t value

(Intercept) 0.3430497 0.0989155 3.468

game\_count\_last\_week 0.4047048 0.0183119 22.101

game\_count\_last\_month 0.0604869 0.0047231 12.807

game\_count\_last\_year 0.0027287 0.0002777 9.828

genre\_count\_last\_week 0.1002376 0.0279454 3.587

avg\_price\_last\_week -0.0437126 0.0077422 -5.646

avg\_price\_last\_year -0.0336651 0.0107768 -3.124

**prob\_residual 0.9042545 0.0961949 9.400**

print(AIC(best\_game\_count\_model))

**13373.66**

############################### Sales/Revenue random effects model ###############################

# Subset of data frames with publishers that are ranked in a given week

df\_ranked <- df[df$ranked == 1, ]

# Random effects model on publisher revenue

df\_ranked <- na.omit(df\_ranked)

# Scale price to avoid scaling issues on models

df\_ranked["revenue\_last\_week"] <- df\_ranked["revenue\_last\_week"] / 10

df\_ranked["revenue\_last\_month"] <- df\_ranked["revenue\_last\_month"] / 10

df\_ranked["revenue\_last\_year"] <- df\_ranked["revenue\_last\_year"] / 10

df\_ranked["avg\_price\_last\_week"] <- df\_ranked["avg\_price\_last\_week"] / 1000

df\_ranked["avg\_price\_last\_month"] <- df\_ranked["avg\_price\_last\_month"] / 1000

df\_ranked["avg\_price\_last\_year"] <- df\_ranked["avg\_price\_last\_year"] / 1000

revenue\_model = lmer(revenue ~ revenue\_last\_week + revenue\_last\_month + revenue\_last\_year +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df\_ranked, na.action = "na.fail")

summary(revenue\_model)

Linear mixed model fit by REML ['lmerMod']

Formula: revenue ~ revenue\_last\_week + revenue\_last\_month + revenue\_last\_year +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher)

Data: df\_ranked

REML criterion at convergence: 85187.1

Scaled residuals:

Min 1Q Median 3Q Max

-4.0000 -0.2066 -0.1072 -0.0146 24.5005

Random effects:

Groups Name Variance Std. Dev.

publisher (Intercept) 4036 63.53

Residual 381198 617.41

Number of obs: 5431, groups: publisher, 134

Fixed effects:

Estimate Std. Error t value

(Intercept) 129.0224 64.4430 2.002

revenue\_last\_week 1.4066 0.1672 8.411

revenue\_last\_month 2.1669 0.2806 7.721

revenue\_last\_year 0.8211 0.8754 0.938

genre\_count\_last\_week 24.9654 23.7080 1.053

genre\_count\_last\_month -2.7981 7.3335 -0.382

genre\_count\_last\_year 1.0670 0.4384 2.434

avg\_price\_last\_week -14.0619 6.2310 -2.257

avg\_price\_last\_month -11.8421 8.6121 -1.375

avg\_price\_last\_year -0.2621 8.5030 -0.031

**prob\_residual 14.4319 66.7915 0.216**

# Use dredge to get the best model based on AIC criteria

revenue\_model\_comparison <- dredge(revenue\_model)

print(revenue\_model\_comparison[which.min(revenue\_model\_comparison$AICc), ] )

# Best model based on dredge

best\_revenue\_model = lmer(revenue ~ revenue\_last\_week + revenue\_last\_month +

genre\_count\_last\_week + genre\_count\_last\_month + genre\_count\_last\_year +

avg\_price\_last\_week + avg\_price\_last\_month + avg\_price\_last\_year +

prob\_residual + (1 | publisher), data=df)

print(AIC(best\_revenue\_model))

727014.3