HOCHSCHULE KAISERSLAUTERN

VISUAL DATA ANALYSIS

Analyse der Disparität der Fertigstellungszeit in Videospielen

Ujwal Subedi
Wissam Alamareen

unter Betreuung von
Prof. Dr. Manfred BRILL

9. Juli 2021

Zusammenfassung

Die Spieler- und Spielstatistiken sind wirklich faszinierend, nicht nur wegen der Art der Spiele, sondern auch wegen der Zeit, die Menschen in sie investieren und versuchen, sie zu meistern. Wie in den Daten zu sehen ist, sind Main Story, Main + Extras, Completionists, Co-Op Multiplayer, Speed Run - Any% , Speed Run - 100% enthalten, wie die Daten genommen werden. In dieser Arbeit werden wir die gescrapten Daten von HowlongtoBeat analysieren und viele versteckte Informationen herausfinden, die zu einigen guten Game-Design-Prinzipien führen könnten.

Inhaltsverzeichnis

1 Einleitung

Die Gegebenheit, die wir hier haben, ist die Spielabschlusszeit, bei der wir versuchen, herauszufinden, welcher bestimmte Parameter die Kunst, ein bestimmtes Spiel abzuschließen, verändern könnte. Wie lange braucht man, um ein bestimmtes Spiel abzuschließen, und was ist die ideale Länge eines Videospiels? Das ist das Hauptproblem, das wir herauszufinden versuchen, damit die Spieleentwickler das Spiel so gestalten können, dass die Spieler gerne Zeit darin verbringen. was dazu führen könnte, dass die Entwickler weniger langweilige Inhalte haben. Die Fertigstellungszeit ist aufgrund vieler Aspekte unterschiedlich, zum Beispiel wegen der Plattform, auf der ein Spieler das Spiel spielt. Bei einigen Spielen können PC-Spieler das Spiel früher beenden, aber bei Spielen für die PlayStation ist das Spiel schneller fertig als für den PC. Darüber hinaus, wie lange hat dieser Spieler schon Videospiele gespielt, welche Art von Skillset hat er, um das Spiel schneller abzuwickeln? Spielen sie das gleiche Genre von Spielen seit langem oder ist es das allererste Mal, dass sie das Spiel spielen? Dieselbe Person braucht auch unterschiedlich lange, um dasselbe Spiel zu beenden, weil sie manchmal einige Taktiken anwenden kann und manchmal nicht.

2 Informationen über Daten und Attribute darin

Aus dem Projekt Howlongtobeat von Kaggle werden uns 4 Dateien zur Verfügung gestellt, die wir analysieren werden. Die Daten, die wir hier verwenden werden, werden über die Website HowLongToBeat gesammelt. HowLongToBeat (HLTB) ist eine großartige Website, um die Zeiten zu ermitteln, die Leute brauchen, um Spiele zu beenden. Wir haben diese Daten auf kaggle gefunden und der Benutzer KasumiL5x hat die API geholt und die Daten gesammelt, die alle bekannten Spiele (zum Zeitpunkt des Schreibens) auf der Website enthalten, wobei er die Spieldaten sowie alle vorhandenen Abschlusseinträge extrahiert hat.

Welche Dateien haben wir und was enthalten sie?

2.0.1 completions.csv

In dieser Datei haben wir die Spiel-ID, die Art der Spielbeendigung, die Plattform, auf der der Spieler das Spiel spielt, sowie den Zeitrahmen, in dem ein Spieler das Spiel beendet hat.

- id -: Spiel-ID, die mit dem obigen Datensatz quer verknüpft werden kann.
- type -: Typ des Abschlusseintrags (Main Story, Main + Extras, Completionists, Co-Op Multiplayer, Speed Run - Any%, Speed Run - 100%).
- platform -: Plattform, auf der der jeweilige Eintrag abgeschlossen wurde.
- time -: Zeit des Eintrags in Stunden und Minuten (z. B. 2h 50m).

2.0.2 all-completions

Die Datei all-completions.csv enthält genau die gleichen Attribute wie completions.csv und der einzige Unterschied zwischen den beiden Daten ist, dass all-completions.csv nachbereinigt ist.

2.0.3 games.csv

- id -: Spiel-ID von der Website.
- title -: Name des Spiels.
- main_story -: Durchschnittliche Fertigstellungszeit von 'Main Story' in Stunden.
- main_plus_extras -: Durchschnittliche Fertigstellungszeit von 'Main + Extras' in Stunden.
- completionist -: Durchschnittliche Fertigstellungszeit von 'Completionist' in Stunden.
- all_styles -: Durchschnittliche Fertigstellungszeit von 'All Styles' in Stunden.
- coop -: Durchschnittliche Fertigstellungszeit von 'Kooperativer Mehrspielermodus' in Stunden.
- versus -: Durchschnittliche Fertigstellungszeit von 'Vs.' in Stunden.
- type -: Typeneintrag zur Unterscheidung von DLC/Expansion, Mod und ROM Hack von regulären Spieleinträgen.
- developers -: Durch Kommata getrennte Liste aller Entwickler eines Eintrags.
- publishers -: Durch Kommata getrennte Liste aller Publisher eines Eintrags.
- platforms -: Durch Kommata getrennte Liste aller Plattformen, auf denen ein Eintrag verfügbar ist.
- genres -: Kommagetrennte Liste aller Genres eines Eintrags.
- release_na -: Veröffentlichungsdatum in Nordamerika (falls verfügbar).
- release_eu -: Veröffentlichungsdatum in Europa (falls verfügbar).
- release_jp -: Veröffentlichungsdatum in Japan (falls verfügbar).

2.0.4 all-games-processed.csv

Ähnlich wie bei den Completions haben wir hier genau die gleichen Attribute wie bei games.csv, aber die Inhalte darin werden verarbeitet und bereinigt.

3 Vorgang der Datenbereinigung

3.0.1 all-completions

In der Datei all-completions.csv sehen wir die Plattform und die Zeit, wobei die Zeit noch in Stunden und Minuten angegeben ist. Wenn wir die Stunden und Minuten in ihrem eigenen Zustand lassen, können Fehler auftreten und es werden nicht die bestmöglichen Ergebnisse angezeigt (Siehe Abbildung 1).

id	type	platform	time
1000	Main Story	PC	1h 27m
1000	Main Story	PC	1h 30m
1000	Main Story	PC	1h 55m
1000	Main Story	Xbox 360	2h
1000	Main + Extras	PC	2h
1000	Main + Extras	PC	2h
1000	Main + Extras	Xbox 360	4h
1000	Completionists	PC	2h
1000	Completionists	PC	2h 01m
1000	Completionists	PC	3h
1000	Completionists	PlayStation 3	4h

Abbildung 1: Informationen zum Spiel mit der ID 1000 aus den Daten all-completions.csv

Um dieses Problem zu lösen, haben wir versucht, das Zeitformat nur auf Minute zu ändern, sodass es sich um einen einzelnen ganzzahligen Wert handelt (Siehe Abbildung 2).

id	type	platform	time	time_min
1000	Main Story	PC	1h 27m	87
1000	Main Story	PC	1h 30m	90
1000	Main Story	PC	1h 55m	115
1000	Main Story	Xbox 360	2h	120
1000	Main + Extras	PC	2h	120
1000	Main + Extras	PC	2h	120
1000	Main + Extras	Xbox 360	4h	240
1000	Completionists	PC	2h	120
1000	Completionists	PC	2h 01m	121
1000	Completionists	PC	3h	180
1000	Completionists	PlayStation 3	4h	240

Abbildung 2: Neue Spalte time_min, in der die gesamte Zeit nur in Minuten umgerechnet wird

3.0.2 all-games-processed

Wir haben das ähnliche Problem wie zuvor, wo wir wieder in einer anderen Spalte geändert und gespeichert haben (Siehe Abbildung 3 und 4).

id	title	main_story	main_plus_extras	completionist	all_styles	coop	versus	type	developers	publishers	platforms	genres	release_na	release_eu	release_jp
1000	Bejeweled 2	1.5	2.5	2.5	2	NA	NA		PopCap Games, Oberon Media, Astraware	PopCap Games, Sony Online Entertainment, Electronic Arts	PC, PlayStation 3, Xbox 360, Xbox One	Puzzle	November 05, 2004		

Abbildung 3: Hier zu sehen sind die Spalten von main_story bis versus, die im Stunden- und Minutenformat als Dezimalwerte vorliegen.

id	title	type	developers	publishers	platforms	genres	release_na	release_eu	release_jp	main_story_min	main_plus_extras_min	completionist_min	all_styles_min	coop_min	versus_min
1000	Bejeweled 2		PopCap Games, Oberon Media, Astraware	PopCap Games, Sony Online Entertainment, Electronic Arts	PC, PlayStation 3, Xbox 360, Xbox One	Puzzle	November 05, 2004			90	150	150	120	NA	NA

Abbildung 4: Die Spalten wurden geändert, wobei die angegebenen Stunden und Minuten nur in Minuten umgewandelt werden.

3.1 All Platforms

In der Abbildung unten sind mehrere Plattformen zu sehen, deren Daten wir erhalten haben, aber wir müssen nicht alle von ihnen in unsere Analyse einbeziehen.

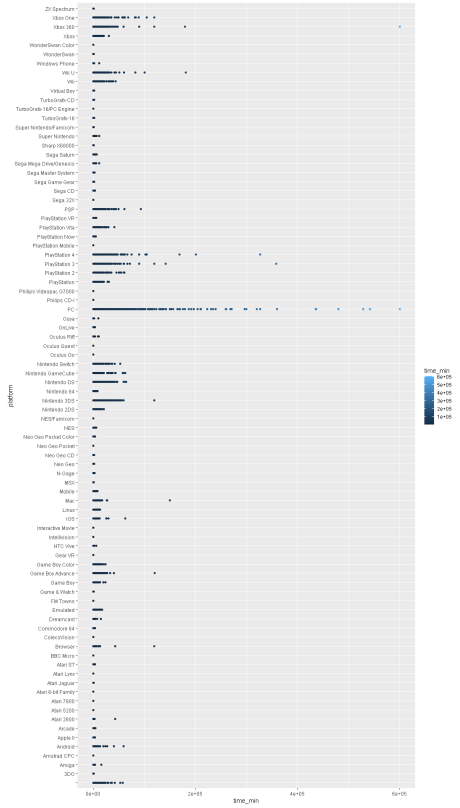

Abbildung 5: alle Plattformen und Spielzeit in jeder

4 Durchschnittliche Fertigstellungszeit von Spielen in Abhängigkeit von ihrem Typ

Von welchen Änderungen hängt die tatsächliche Fertigstellungszeit ab? Was sind die möglichen Gründe, warum die Fertigstellungszeit eines Spiels zwischen den Plattformen variieren kann? Dies sind die Fragen, die wir anhand der uns vorliegenden Daten beantworten werden.

4.1 Werden die Fertigstellungszeiten durch die Plattform beeinflusst?

Die Daten, die wir jetzt sehen, stellen dar, wie lange es dauert, einige Spiele zu beenden, je nachdem, auf welcher Plattform sie gespielt werden. Der Nintendo 2DS scheint im Vergleich zu den anderen Plattformen am längsten zu brauchen, um ein Spiel zu

beenden. Woran kann das liegen? Auf anderen Plattformen könnten ähnliche Spiele einen einzigartigen Controllertyp haben, der deutlich mehr Eingaben hat und somit die Fertigstellungszeit beschleunigt.

Es ist seltsam, dass die offensichtlichste Plattform, der PC, hier nicht auftaucht. Warum könnte das so sein? Die PC-Spiele sind in der Regel länger und schwieriger als die Spiele, die auf anderen Plattformen erscheinen.

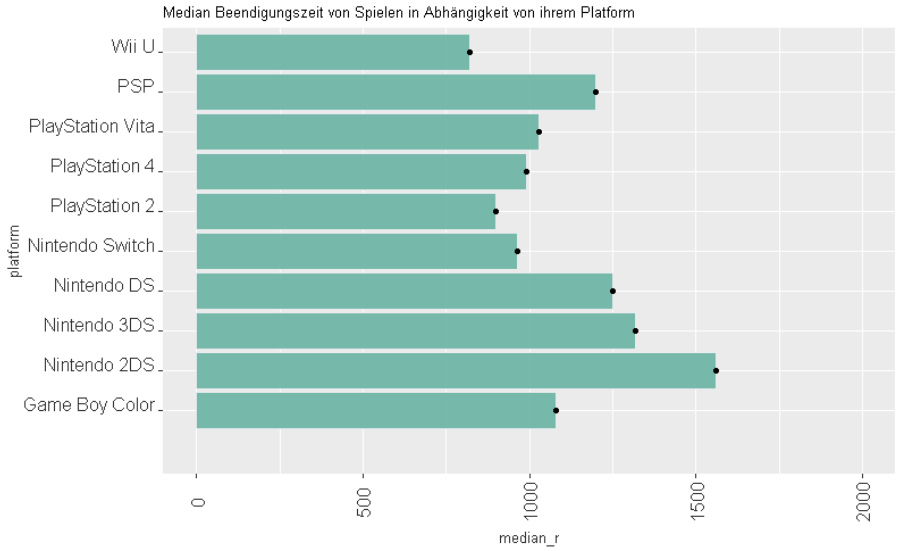

Abbildung 6: Mittlere Fertigstellungszeit je nach Plattform, auf der man spielt

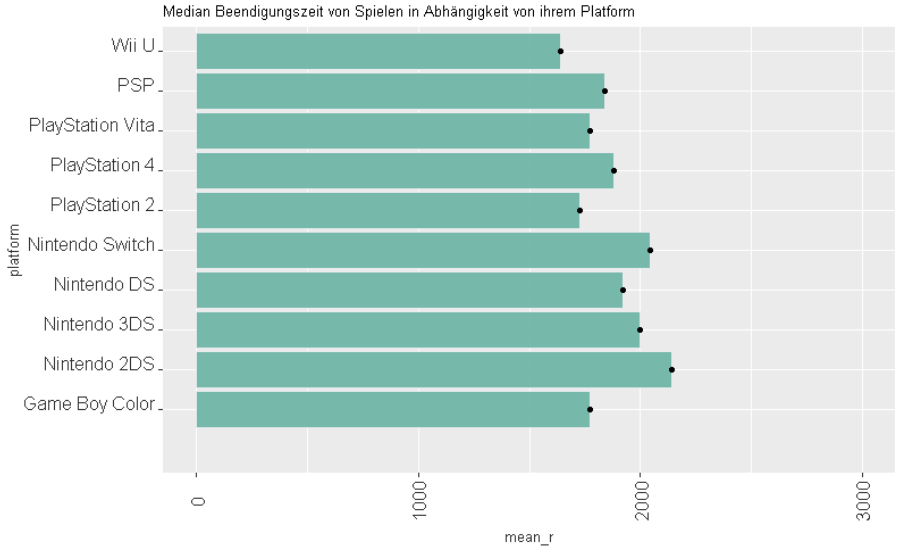

Abbildung 7: Durchschnittliche Fertigstellungszeit je nach Plattform, auf der man spielt

4.2 Art der Spiele

Als Plattform für Spiele hat der PC eindeutig die Nase vorn. PC-Gaming ist im Vergleich zu anderen Plattformen die beliebteste Plattform, und es ist ganz offensichtlich, dass die PlayStation die zweitbeliebteste Plattform ist.

Wie aus der Grafik ersichtlich ist, würde das Erfüllen von Main + Extra mehr Zeit in Anspruch nehmen als die anderen Arten von Spielabschlüssen. Mit anderen Worten: Spiele mit vielen Nebenquests zusätzlich zur Hauptstory würden mehr Zeit für den Abschluss benötigen als Speed Runs. Aufgrund der Tatsache, dass Menschen ein einzelnes Spiel nicht stundenlang spielen

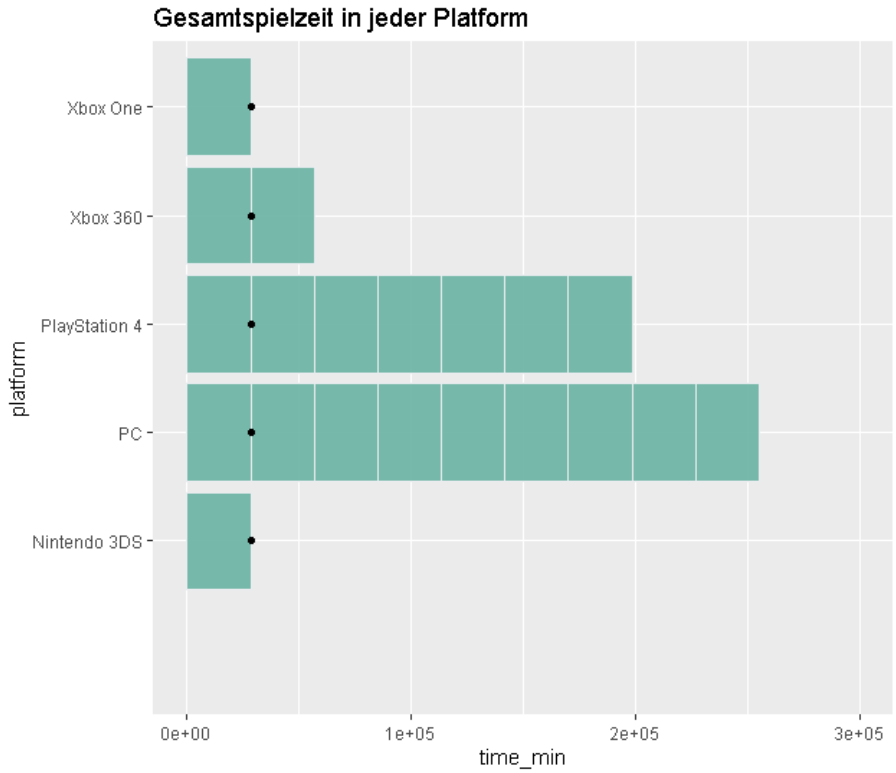

Abbildung 8: Gesamtspiele auf jeder Plattform gespielt

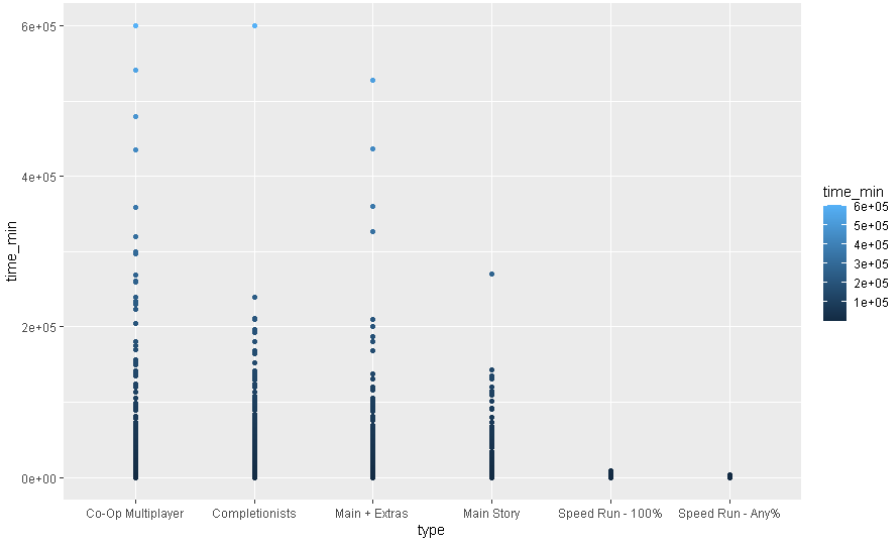

Abbildung 9: Spieltyp und benötigte Zeit

und immer wieder denselben Spielmodus wiederholen möchten, zeigt sich, dass Co-Op-Multiplayer-Spiele weniger Zeit in Anspruch nehmen.

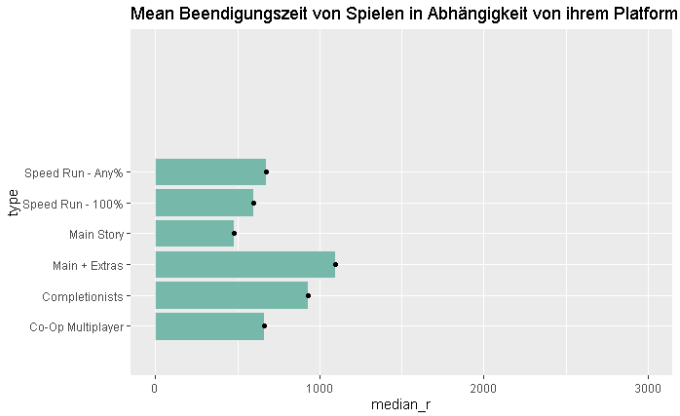

Abbildung 10: Durchschnittliche Fertigstellungszeit

5 Einfache Linear Regression

Die lineare Regression beschreibt die Beziehung zwischen Variablen, indem sie die Linie der besten Anpassung durch die gegebenen Daten liefert, indem sie den Regressionskoeffizienten findet, die den Gesamtfehler des Modells minimieren. [2]
Formel: y = $\alpha + \beta 1x1$.

Nach dem Einlesen der Daten geht es an die Modelldefinition. In unsere Datei ist es nicht klar, welche variablen von den anderen abhängig sind. Demzufolge haben wir angenommen, die abhängige (y-)Variable das all_styles_min in Minuten und die unabhängige (x-)Variable das main_plus_extras_min in Minuten.

5.1 Linear Regression Funktion

Zur einfachen linearen Regression verwendet man die lm()-Funktion. lm steht hierbei für linear Model. Wir definieren uns ein Modell mit dem Namen „Simple_modell“. Hierin soll all_styles_min erklärt werden und wird an den Anfang in der Klammer gestellt, gefolgt von -ünd der erklärenden Variable main_plus_extras_min.
Die Daten kommen aus dem Dataframe „all_games_processed“, weshalb wurde das „data=„-Argument noch angefügt.
Mit der Summary()-Funktion lässt sich die Ergebnisse der Berechnung von Simple_modell ausgeben.

5.2 Interpretation der Ergebnisse der einfachen linearen Regression

So sieht der Output aus. Die Interpretation erfolgt schrittweise unter dem Output.

```
Call:
lm(formula = all_styles_min ~ main_plus_extras_min, data = all_games_processed)

Residuals:
     Min       1Q   Median       3Q      Max
-13890.8    -50.2    -11.1     41.2  19378.6

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         29.623941   5.783915   5.122 3.08e-07 ***
main_plus_extras_min  0.895299   0.003363 266.198  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 515.8 on 11629 degrees of freedom
  (24291 observations deleted due to missingness)
Multiple R-squared:  0.859,     Adjusted R-squared:  0.859
F-statistic: 7.086e+04 on 1 and 11629 DF,  p-value: < 2.2e-16
```

Abbildung 11: Summary(simple_model)

Man beginnt ganz unten bei der F-Statistik. Schreibweise: F(1,11629)=7.086e+04; p <2.2e-16 . Die Signifikanz (p-Wert) sollte einen möglichst kleinen Wert (<0,05) haben. Wenn dem so ist, leistet das Regressionsmodell einen Erklärungsbeitrag. 2.2e-16 ist eine andere Schreibweise für 0,00000000000000022.
Also im Beispiel deutlich unter 0,05. Das Modell leistet in diesem Falle einen signifikanten Erklärungsbeitrag und es kann mit

der Interpretation der weiteren Ergebnisse fortgefahren werden. Ist die Signifikanz über 0,05, leistet das Regressionsmodell keinen signifikanten Erklärungsbeitrag und das Verfahren bzw. die weitere Interpretation ist abzubrechen,

5.3 Güte des Regressionsmodells

Die Güte des Modells der gerechneten Regression wird anhand des Bestimmtheitsmaßes R-Quadrat (R^2) abgelesen. Das R^2 (Multiple R-Squared) ist standardmäßig zwischen 0 und 1 definiert. R^2 gibt an, wie viel Prozent der Varianz der abhängigen Variable (hier: all_styles_min) erklärt werden. Ein niedriger Wert ist hierbei besser.

5.4 Signifikanz und Größe der Koeffizienten

Der Regressionskoeffizient (hier: all_styles_min) sollte signifikant ($p<0,05$) sein. Damit die Nullhypothese nicht fälschlicherweise abgelehnt wird. Die Signifikanz ist mit 0,00000000000000022 deutlich unter 0,05 und somit hat die main_plus_extras_min einen signifikanten Einfluss auf das all_styles_min.

Die interpretierbare Wirkung dieses Koeffizienten sehen Sie unter Ëstimate". Im Regressionsmodell ist die erste Linie der (Intercept). Dies ist die sogenannte Konstante. Ihre Bedeutung ist für den weiteren Verlauf der Untersuchung nicht relevant. Hier ist nur der Estimate interessant. Und der ist eigentlich nur interessant, wenn eine Prognose durchgeführt werden soll. In der

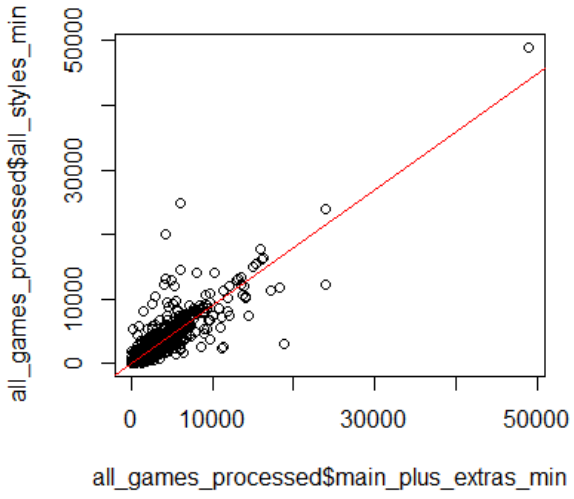

Abbildung 12: Summary(Rplot)

zweiten Zeile steht der Estimate für die main_plus_extras_min . Das ist das all_styles_min, um das sich die abhängige Variable ändert, wenn die unabhängige Variable um 1 steigt – immer! Konkret im Beispiel ist es 0.895299. Das heißt, dass bei einer Steigerung der main_plus_extras_min um eine Einheit das all_styles_min um 0.895299 Minute zunimmt. In der Regel haben positive Koeffizienten einen positiven Einfluss auf die y-Variable und negative Koeffizienten einen negativen Einfluss.

6 Multiple linearen Regression

In diesem Beispiel versuchen wir die all_styles_min durch die main_plus_extras_min und die main_story_min zu erklären
Formel: $y = \alpha + \beta nxn$.

```
multi_model<-lm(all_styles_min~main_plus_extras_min+main_story_min,data = all_games_processed)
```

demzufolge ist die abhängige (y-)Variable der all_styles_min und die unabhängigen (x-) Variablen der main_plus_extras_min und der main_story_min

6.1 Interpretation der Ergebnisse der multiplen linearen Regression

```
Call:
lm(formula = all_styles_min ~ main_plus_extras_min + main_story_min,
    data = all_games_processed)

Residuals:
    Min      1Q  Median      3Q     Max
-8172.5   -68.6    -7.3    25.6 18624.1

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -25.55059    5.52224  -4.627 3.76e-06 ***
main_plus_extras_min   0.55410    0.00629  88.089  < 2e-16 ***
main_story_min         0.59518    0.01086  54.783  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 434.3 on 10241 degrees of freedom
  (25678 observations deleted due to missingness)
Multiple R-squared:  0.8695,    Adjusted R-squared:  0.8695
F-statistic: 3.411e+04 on 2 and 10241 DF,  p-value: < 2.2e-16
```

Abbildung 13: Summary(multi_model)

F(2,10241)=3.41e+04 p <2,2e-16
Der p-Wert ist im Beispiel deutlich unter 0,05. Das multi_model leistet in diesem Falle einen signifikanten Erklärungsbeitrag und es kann mit der Interpretation der weiteren Ergebnisse fortgefahren werden. wie vorher gesagt: Ist die Signifikanz über 0,05, leistet das Regressionsmodell keinen signifikanten Erklärungsbeitrag und das Verfahren bzw. die weitere Interpretation ist an dieser Stelle abzubrechen.

6.2 Güte des Regressionsmodells

```
Multiple R-squared:  0.8695, Adjusted R-squared:  0.8695
```

Im Beispiel erklärt das Modell 86,95% der Varianz, da das (Multiple R-squared) R²=0,8695 ist. Das korrigierte R² (Adjusted R-squared) adjustiert für eine automatische und ungewollte Zunahme des R². Es ist zusätzlich zum normalen R² zu berichten und ist auch stets kleiner als jenes.

6.3 Signifikanz und Größe der Koeffizienten

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -25.55059    5.52224  -4.627 3.76e-06 ***
main_plus_extras_min   0.55410    0.00629  88.089  < 2e-16 ***
main_story_min         0.59518    0.01086  54.783  < 2e-16 ***
---
```

Abbildung 14: Coefficients

In der zweiten Zeile steht der Estimate für den main_plus_extras_min. Das ist der Teil des all_styles_min, um den sich die abhängige Variable ändert, wenn die unabhängige Variable um 1 steigt - immer! Konkret im Beispiel ist es 0.55410. Das heißt, dass bei einer Steigerung des main_plus_extras_min um eine Einheit der all_styles_min um 0.5541 steigt.

Analog kann man die main_story_min und deren Koeffizient betrachten. Der Koeffizient ist 0.59518 und auch hier ist eine Zunahme der Variable main_story_min um eine Einheit für eine Steigerung um 0.59518 und damit Verbesserung des all_styles_min verantwortlich.

7 Fazit

Basierend auf unserer Analyse lässt sich sagen, dass die Fertigstellungszeit mit verschiedenen Aspekten variiert. Für ein gutes Spieldesign muss es sowohl eine längere Spielzeit als auch eine Mischung aus allen anderen Komponenten geben, die die Benutzerbasis dazu anregen, tatsächlich mehr Zeit mit dem Spiel zu verbringen.

8 Zurkenntnisnahme

Wir möchten uns bei Jonathan Bouchet [1] bedanken, dessen Notizbuch uns geholfen hat, uns auf das angestrebte Ergebnis zu konzentrieren.

Literatur

[1] Jonathan Bouchet. *How long it will take?* URL: https://www.kaggle.com/jonathanbouchet/how-long-it-will-take. (accessed: 20.06.2021).

[2] *Linear Regression*. URL: https://www.scribbr.com/statistics/linear-regression-in-r/. (accessed: 20.06.2021).