EVOLVING GAMEPLAY: INNOVATION DIFFUSION MODEL'S INSIGHTS INTO GAAS AND TRADITIONAL USER DYNAMICS

BUSINESS, ECONOMIC AND FINANCIAL DATA – FINAL PROJECT

GIACOMO VIRGINIO, CRISTIAN GRANCHELLI, LETIZIA CIMBRO

RESEARCH QUESTIONS

How well do Bass Models
fit video games'
active users data?

Are Bass Models
able to classify
different types of video games?

Can an automatic procedure
be developed?

CLASSIFICATION OF VIDEOGAMES

Video games can be classified based on various characteristics, such as Genre, Platform, Perspective, Mode, Distribution.

Characteristic	Traditional	Games as a Service
Release	One-time purchase	Ongoing relationship
Content Updates	Separate expansions or sequels	Continuous updates
Monetization	Initial game purchase, DLC	Microtransactions, Subscription
Community Interaction	Limited after release	Crucial - active engagement
Lifespan	Defined by initial popularity	Designed for a longer lifespan
Example	Football Manager	Dota2

SOURCES OF DATA

STEAM

Digital distribution platform that serves as a marketplace for video games, software, and other digital content

STEAMCHARTS

Website that provides statistics and data related to player counts and gameplay activity on the Steam platform, storing historical data available from Steam's API.

DATA SCRAPING AND CLEANING

Month	Avg. Players
Last 30 Days	*799,362.1*
December 2023	770,094.8
November 2023	714,511.4
October 2023	792,125.0
September 2023	976,138.0
August 2023	922,152.9
July 2023	881,690.9

- Manually collected links for games with most current active users in addition to manually selected ones
- Manually excluded games released before July 2012 (user data not available before such date)
- Performed scraping on the site collecting data from tables on each page
- Excluded games released on the platform less than a year ago
- Removed "Last 30 Days" column

DATA

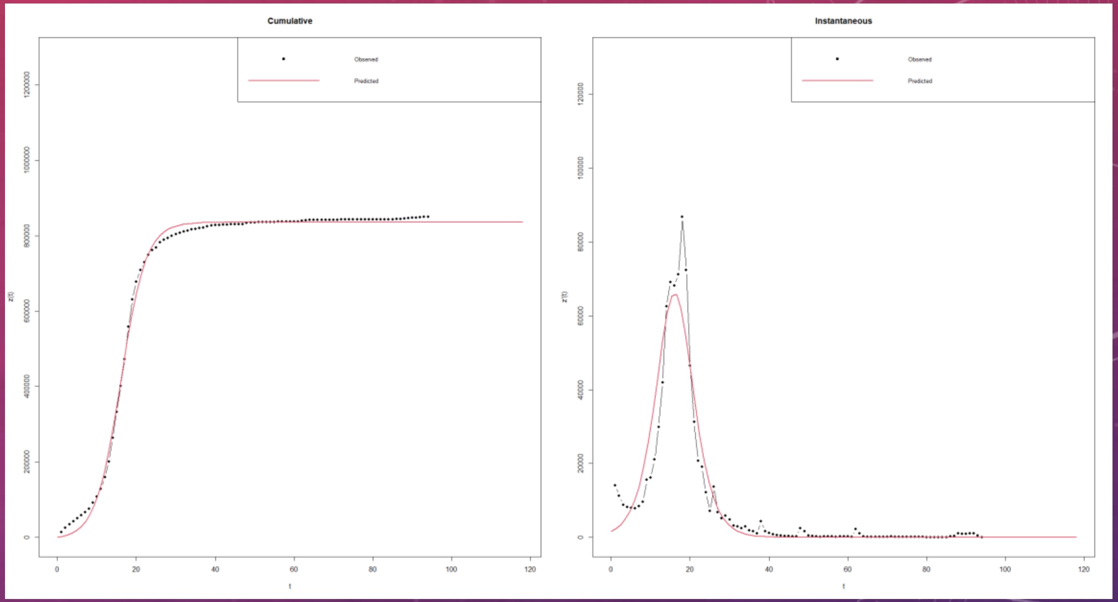
121 videogames

July 2012 → November 2023

Monthly average number of players

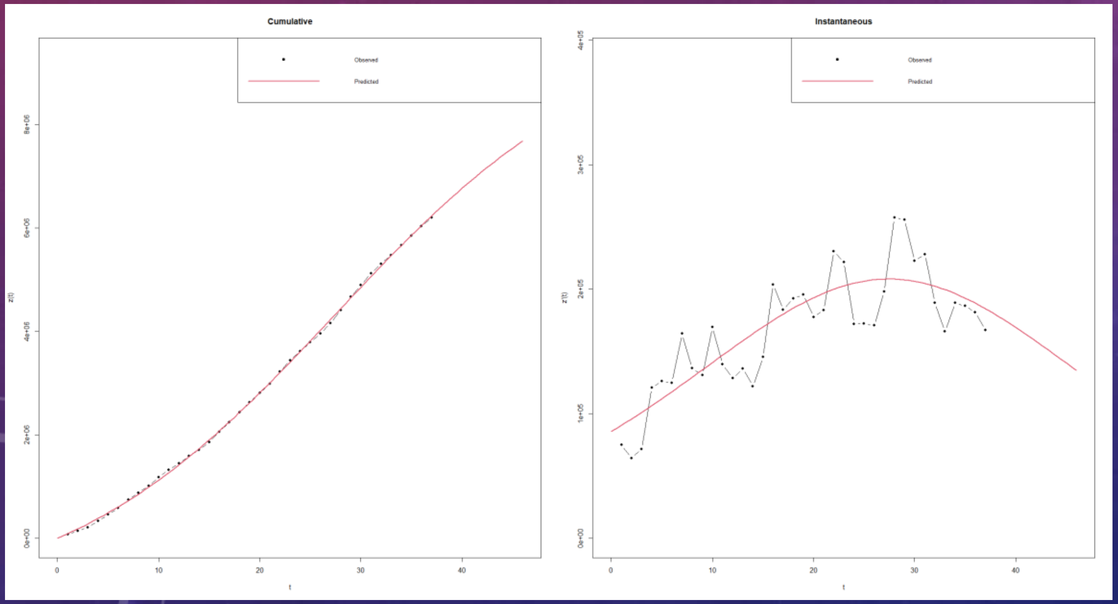
Manual labeling (traditional, GaaS, mixed)

BASS MODEL

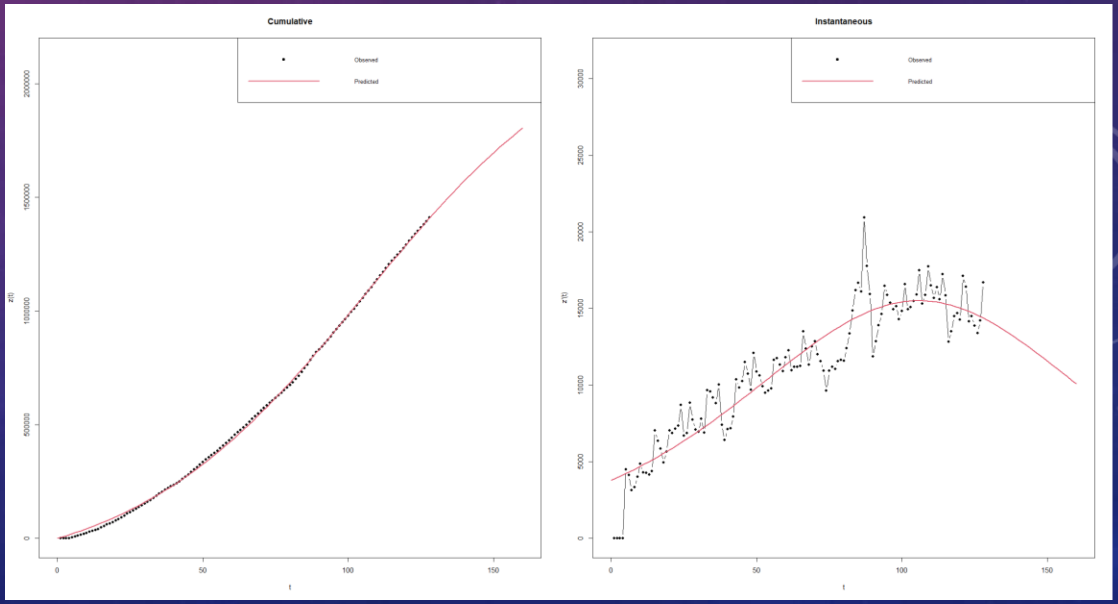


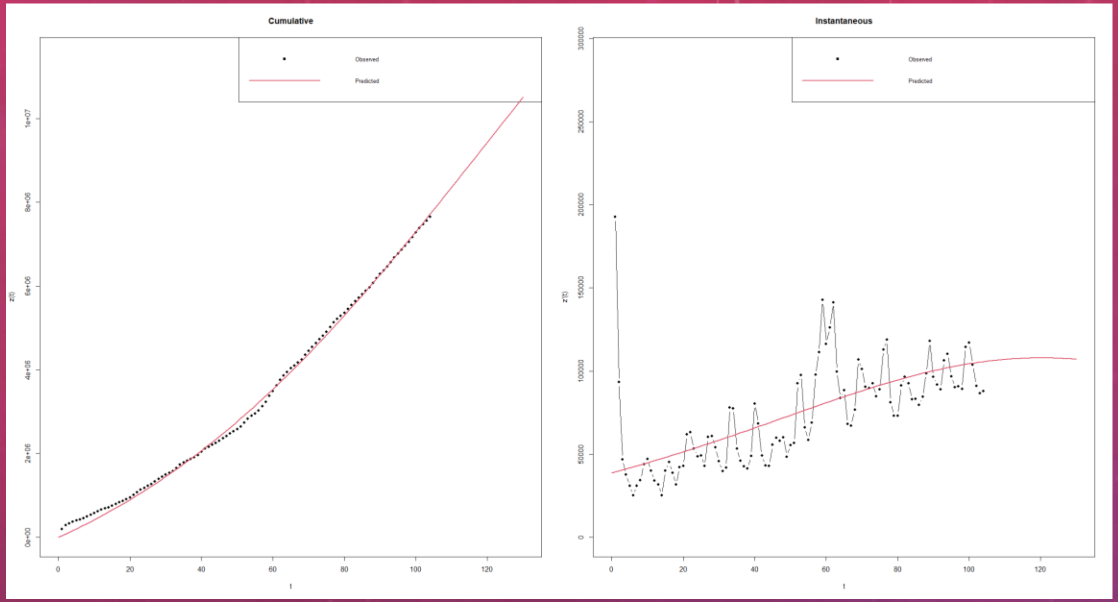
H1Z1, Mixed

Apex Legends, GaaS

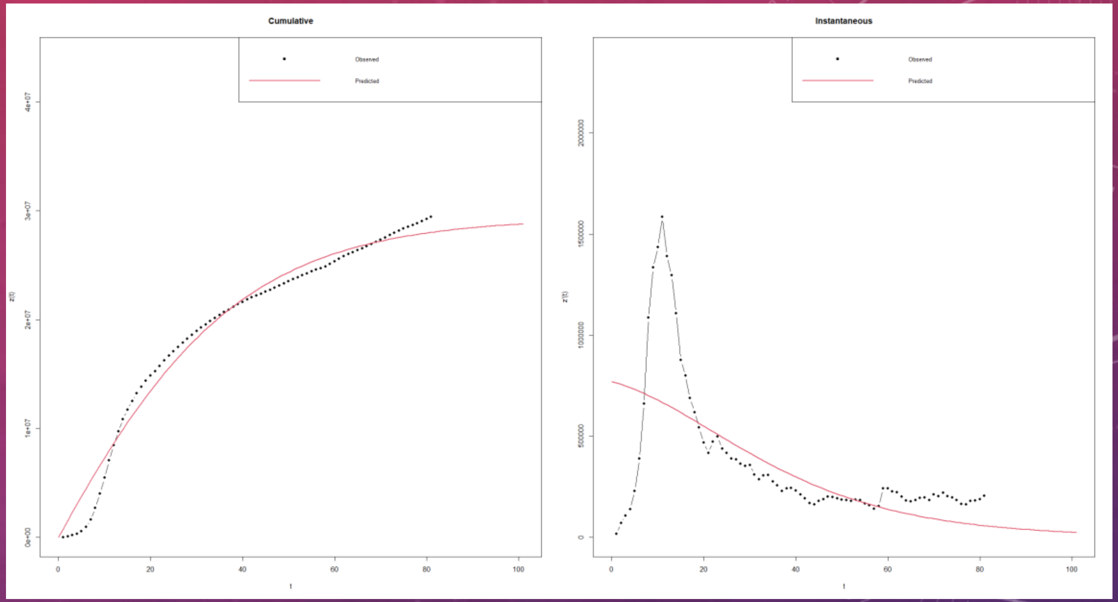


Europa Universalis IV, Traditional



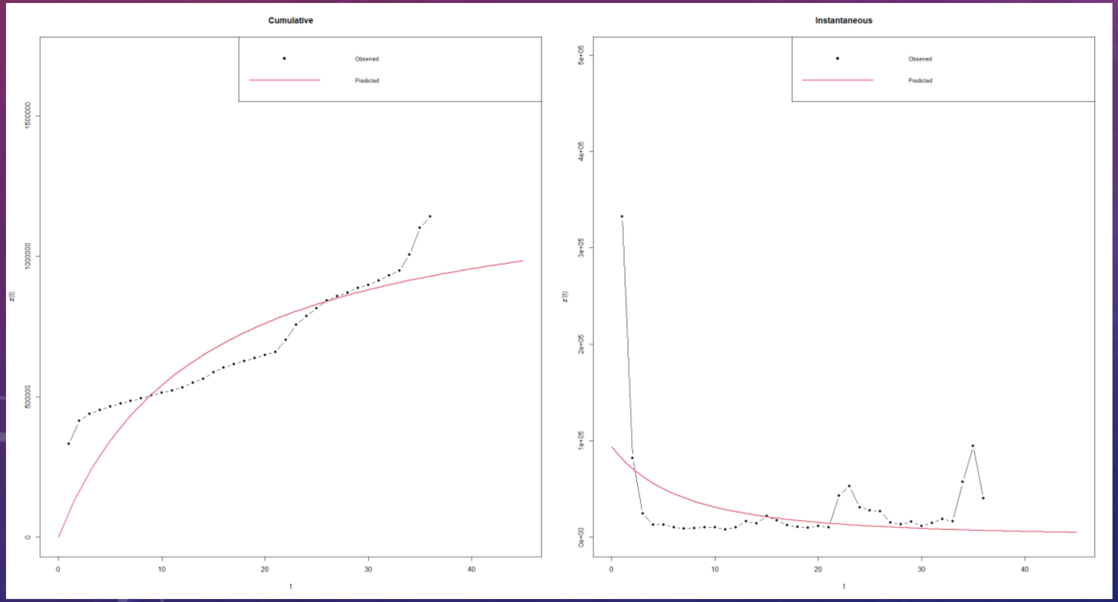


GTA V, Mixed

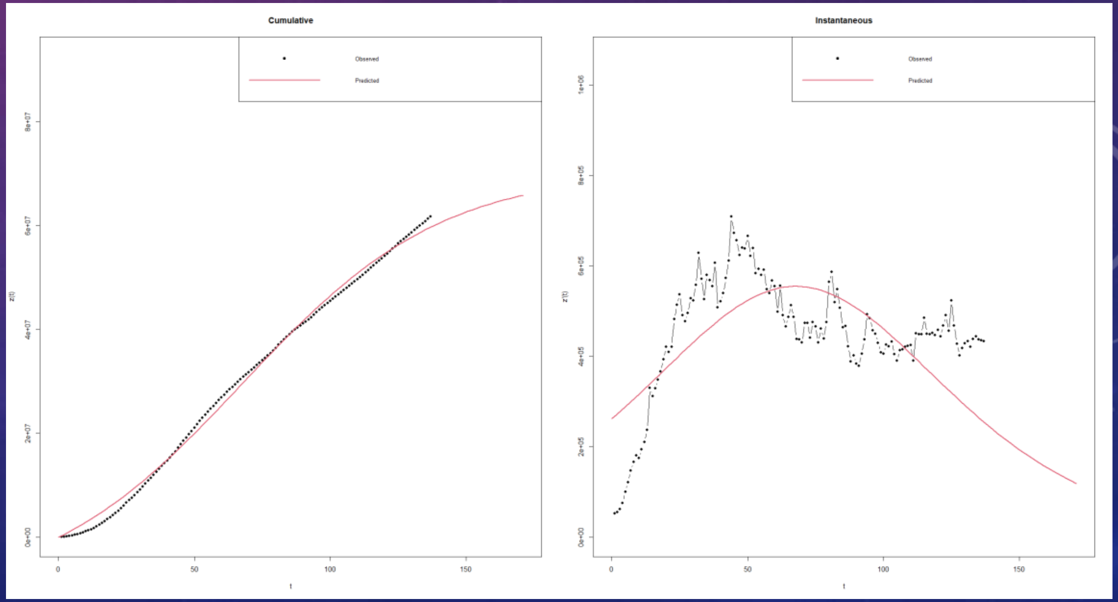


PUBG, Mixed

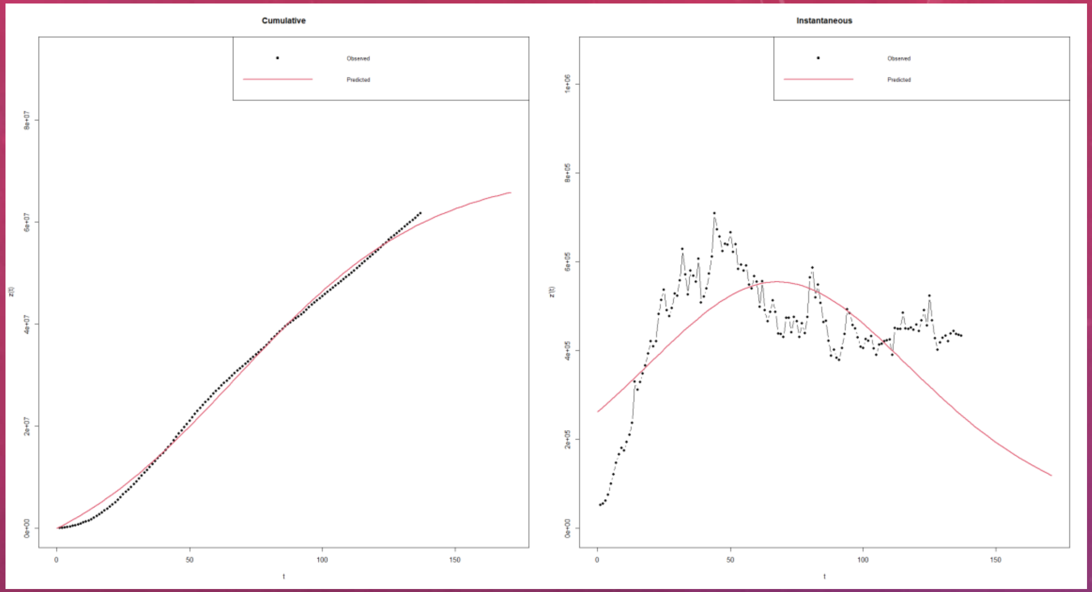
Cyberpunk 2077, Traditional

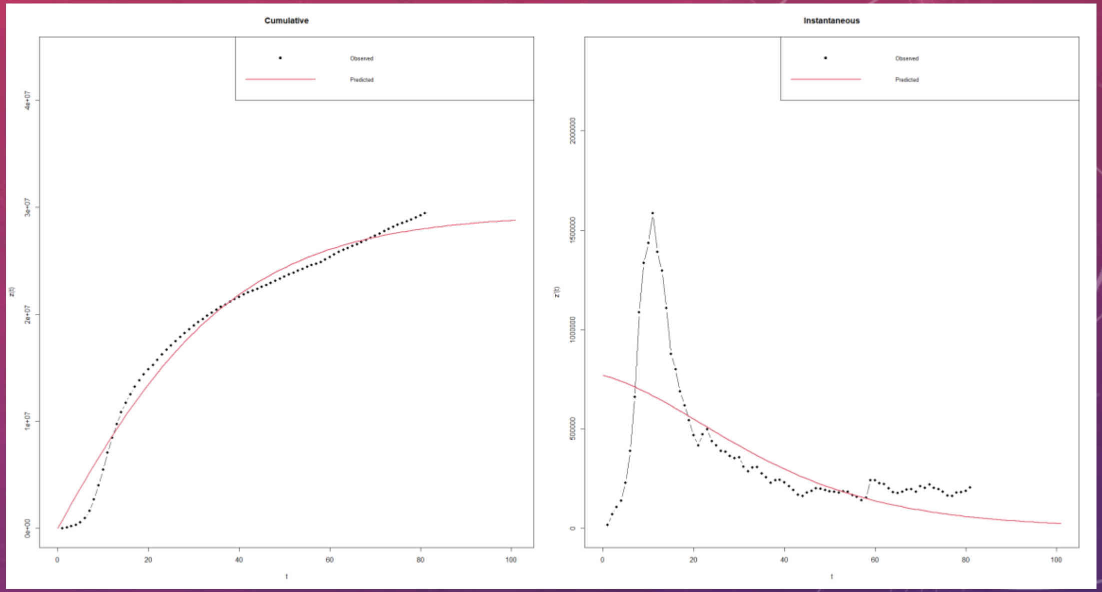


DOTA 2, GaaS



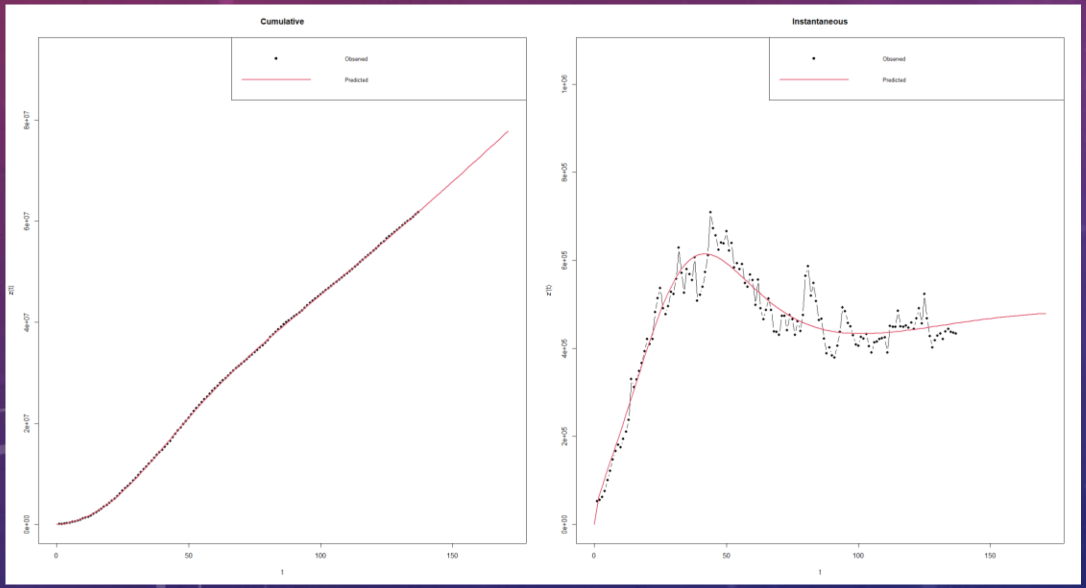
GGM

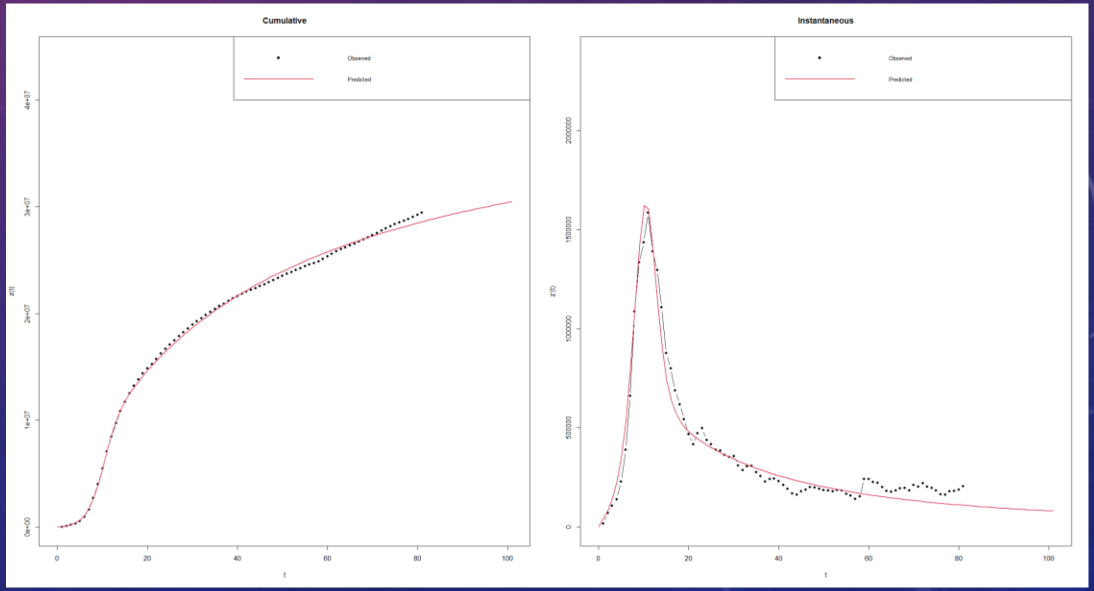




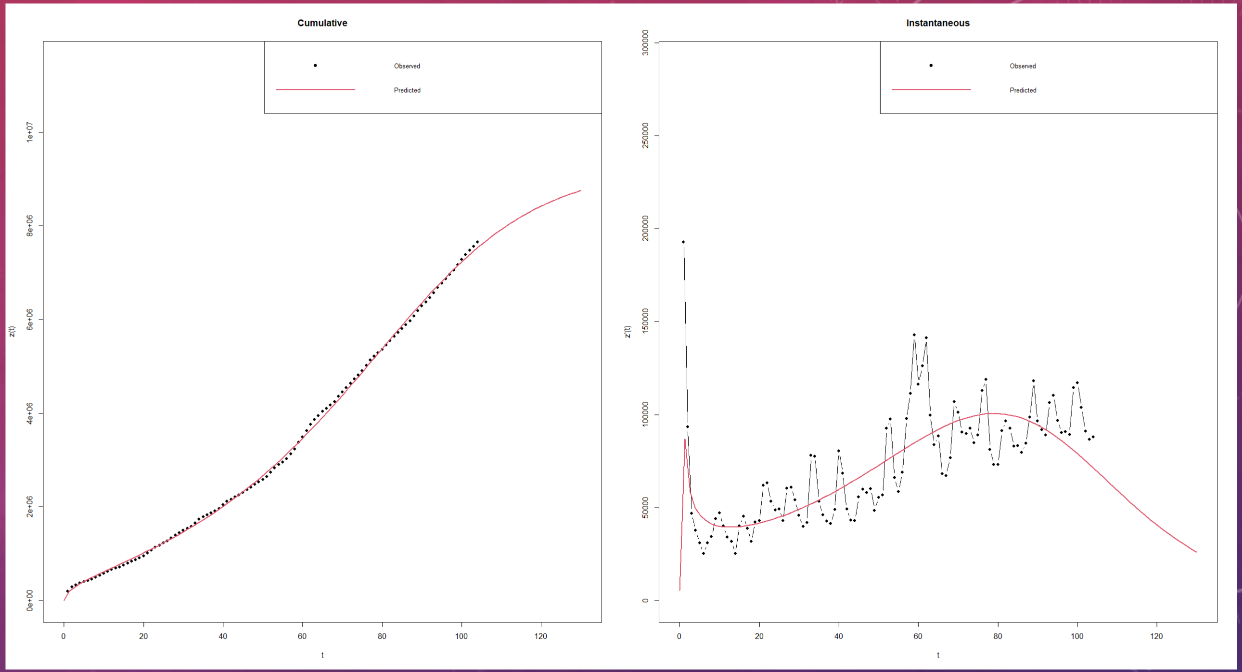
DOTA 2, GaaS

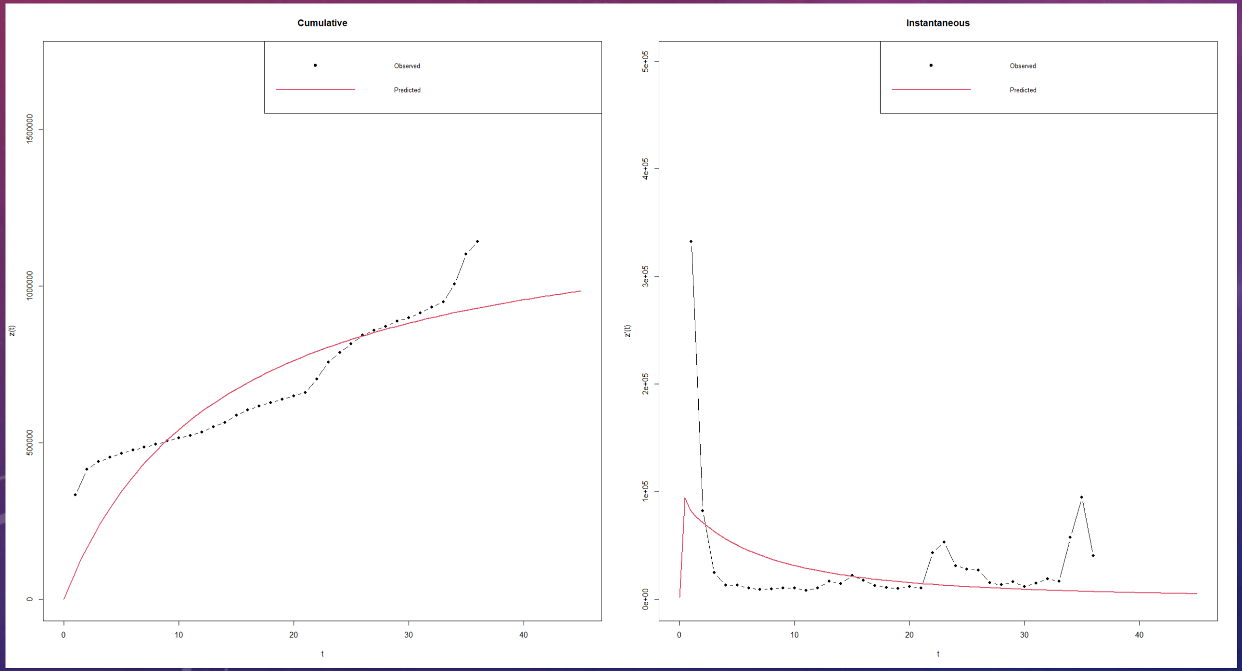
PUBG, Mixed





… but it doesn't always fit well





Sometimes it doesn't fit at all,
getting an error in fitting the hessian

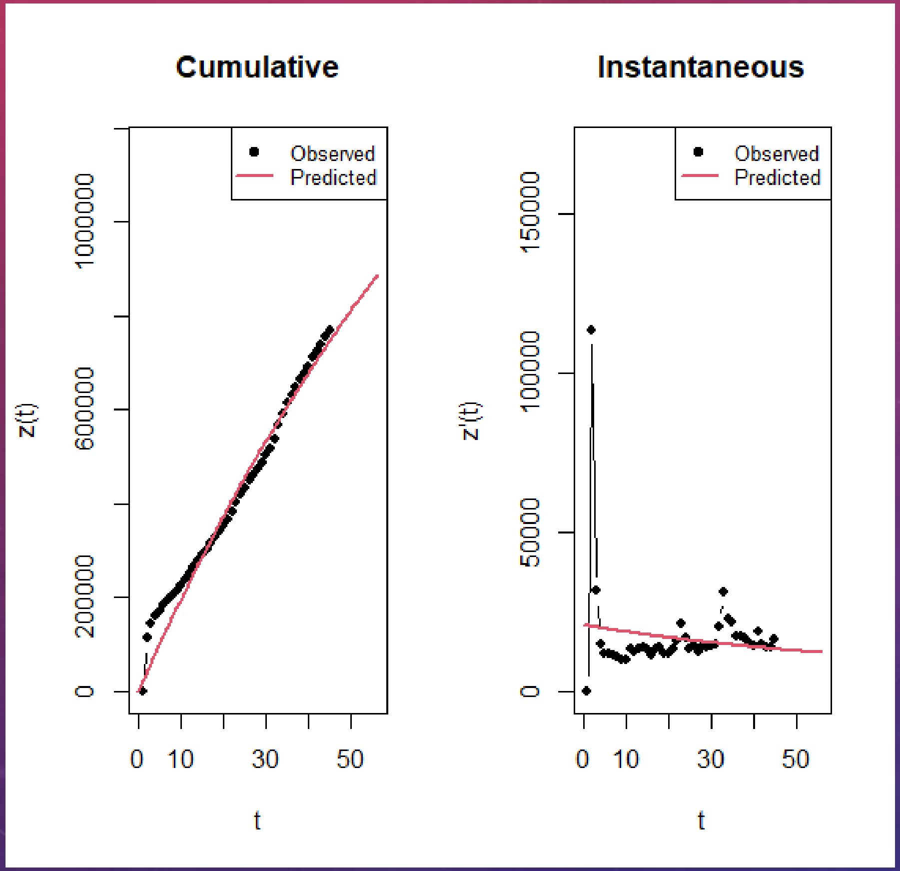
GBM

- We expect traditional videogames to fit well models with shock, in contrast to games-as-a-service (GaaS) models
- We want to use shock related parameters and improved goodness of fit as explanatory variables to classify different types of videogames
- Sources of shocks for traditional videogames:
    - Exponential: game release, update release, DLC.
    - Rectangular: recurrent games (e.g. sports videogames)

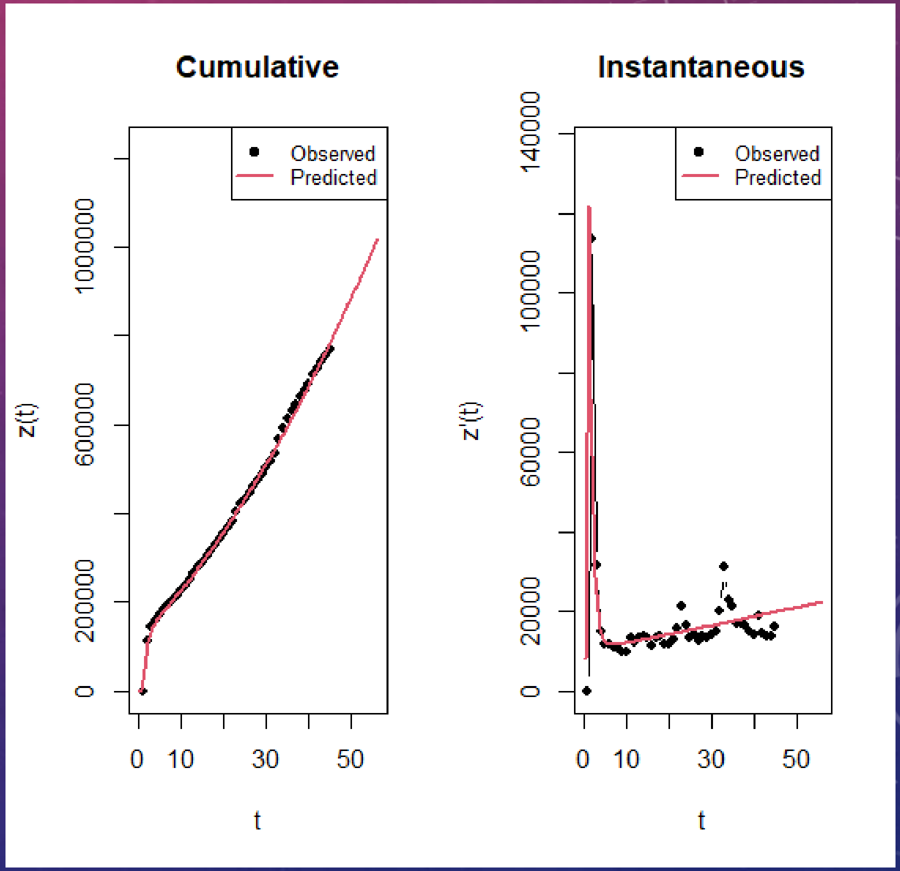
GBM FITTING - FIXED SHOCK ESTIMATION

- Fit a **shock** at the **beginning** of the series:
  - **Exponential**: set as starting point the first data point
  - **Rectangular**: set as period the first year
- 34 models **improved significantly** with an **exponential** shock
- 33 models **improved significantly** with a **rectangular** shock
- **Limitations**:
  - Not possible to fit a shock for some series due to **optimization error**, often caused by **outliers**
  - Procedure doesn't capture **later shocks**

Standard Bass Model

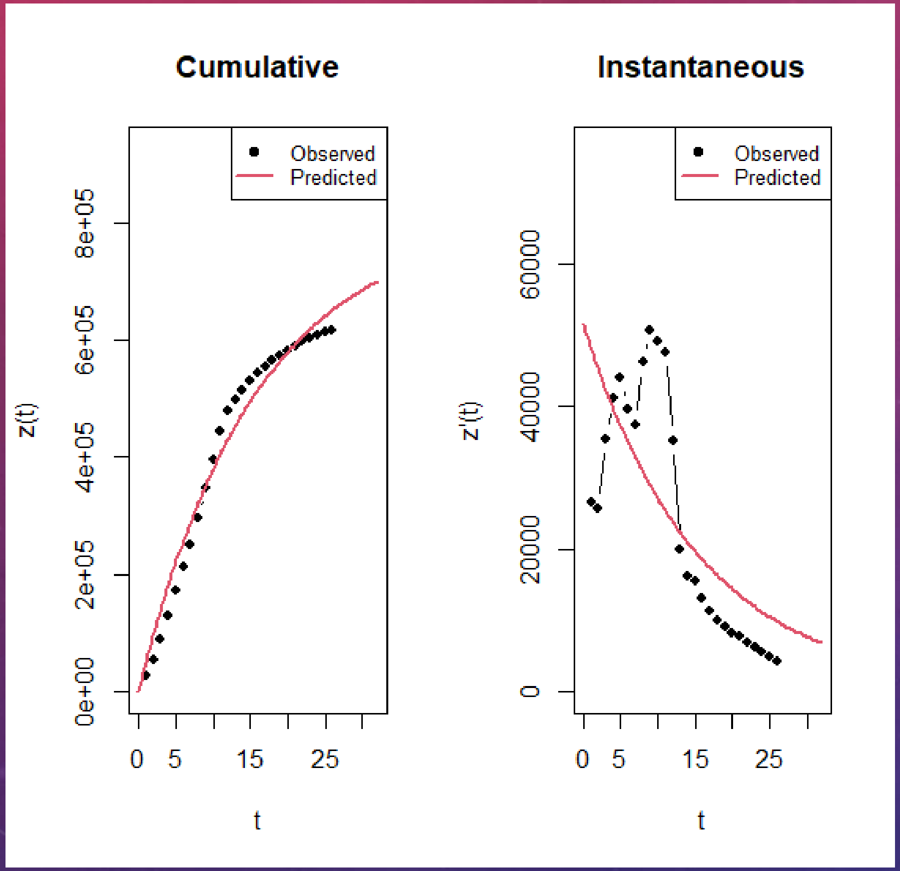


Generalized Bass Model with exponential shock

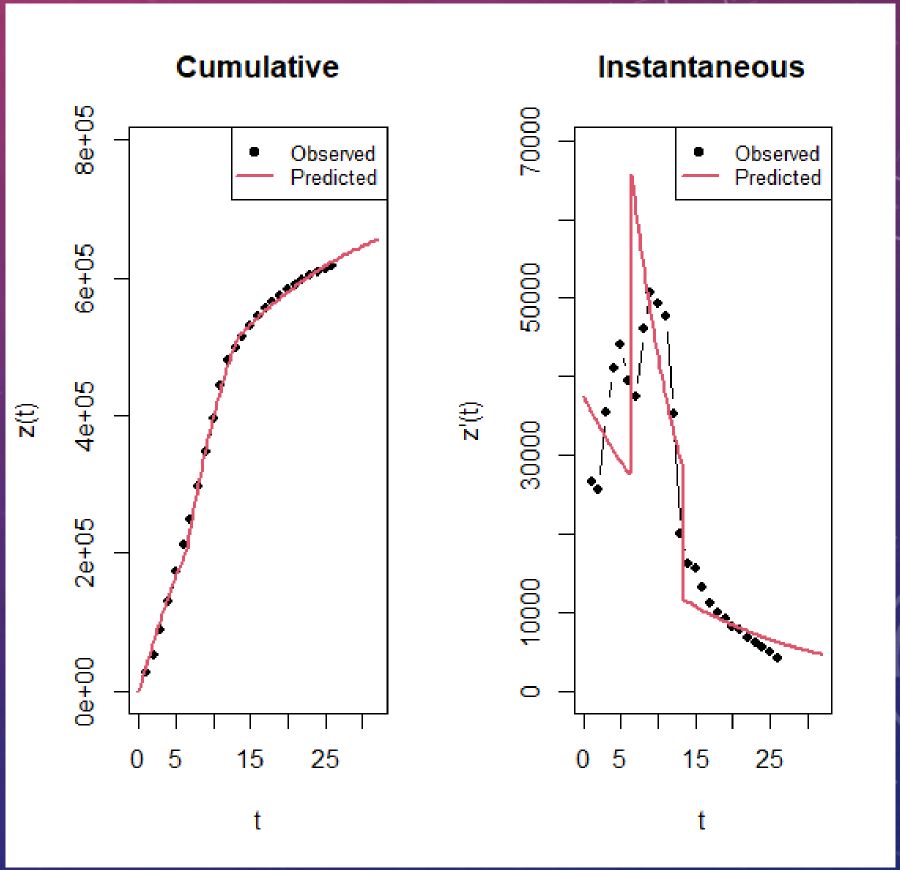


"Mount & Blade II: Bannerlord", Traditional

Standard Bass Model



Generalized Bass Model with rectangular shock



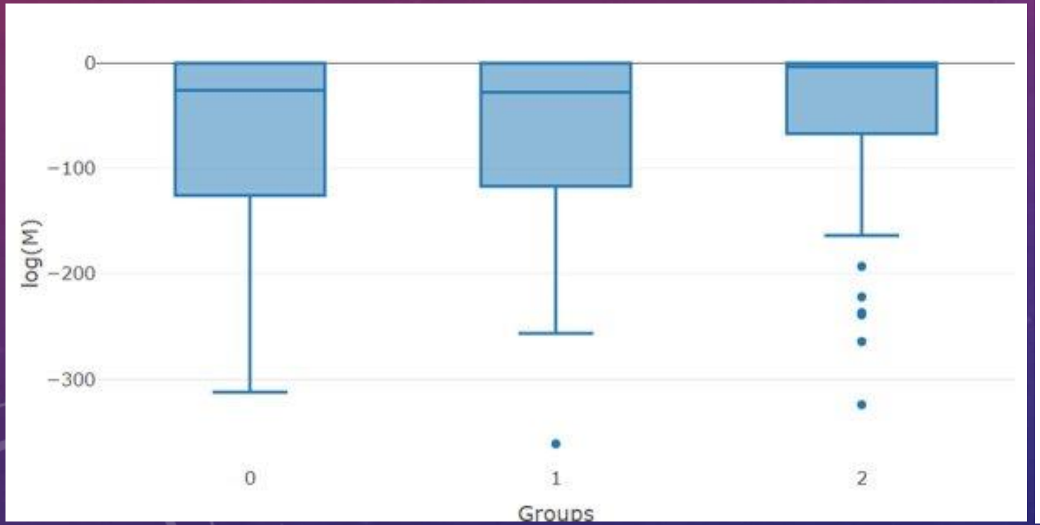
FIFA 22, Mixed

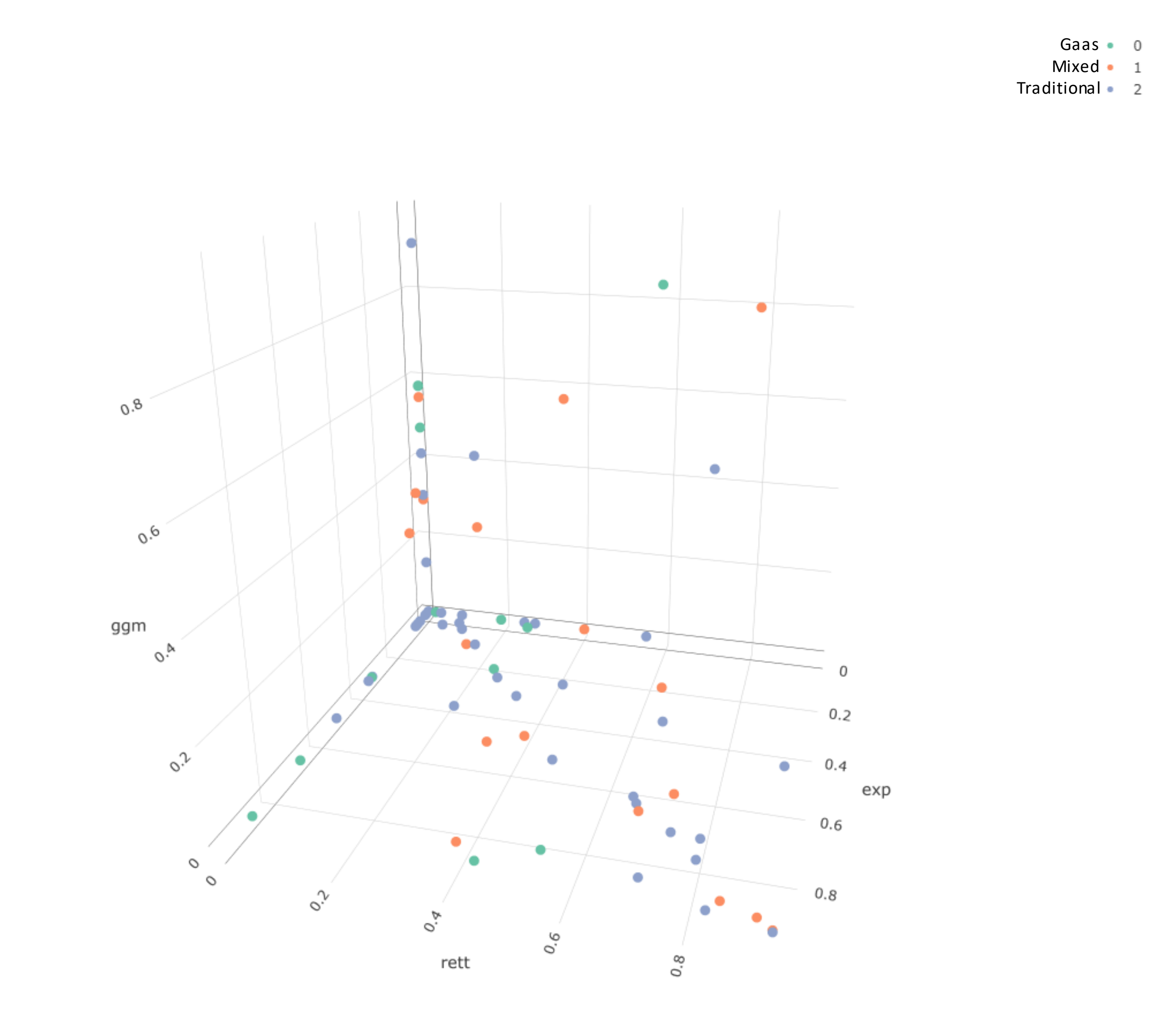
CLASSIFICATION - EXPLANATORY VARIABLES

- **Explanatory variables**:
  - Parameters **m**, **p** and **q,** normalized
  - **Rsq_exp_bm**: variable measuring the (eventual) **improvement** obtained in the **R-squared** with respect to a Bass model, using GBM with **exponential** shock
  - **Rsq_rett_bm**: variable measuring the (eventual) **improvement** obtained in the **R-squared** with respect to a Bass model, using GBM with **rectangular** shock
  - **Rsq_ggm_bm**: variable measuring the (eventual) **improvement** obtained in the **R-squared** with respect to a Bass model, using a **GGM**
  - **naexp**: binary variable equal to 1 if it was possible to fit a GBM with exponential shock, 0 otherwise
  - **narett**: binary variable equal to 1 if it was possible to fit a GBM with rectangular shock, 0 otherwise
  - **naggm**: binary variable equal to 1 if it was possible to fit a GGM, 0 otherwise

VARIABLES' DISTRIBUTION

We used *boxplots* on the chosen explanatory variables and pvalues for the parameters of the models, along with *scatterplots* and *3dscatterplots* of the variables and their logarithm trasformations





CLASSIFICATION - RESULTS

MODEL	Accuracy	Micro F1-score
Ordered Logistic	0.562	0.448
Multinomial Logistic Regression	0.587	0.515
KNN	0.528	0.460

These models tend to be biased towards the majority class (traditional), which contributed to inaccuracies due to the dataset's imbalance.

CLASSIFICATION - REDUCED DATASET

We used the same models considering only the top 60 videogames by average players. This reduced dataset is more balanced.

MODEL	Accuracy	Micro F1-score
Ordered Logistic	0.433	0.448
Multinomial Logistic Regression	0.517	0.515
KNN	0.5	0.513

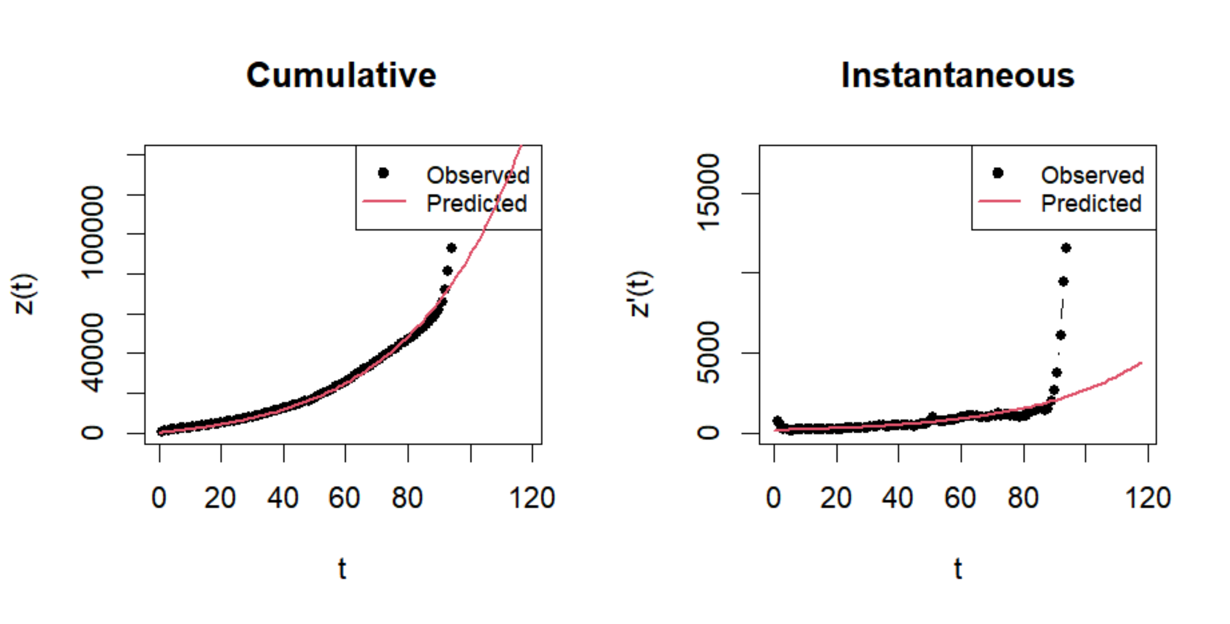
GBM FITTING - AUTOMATIC SHOCK ESTIMATION

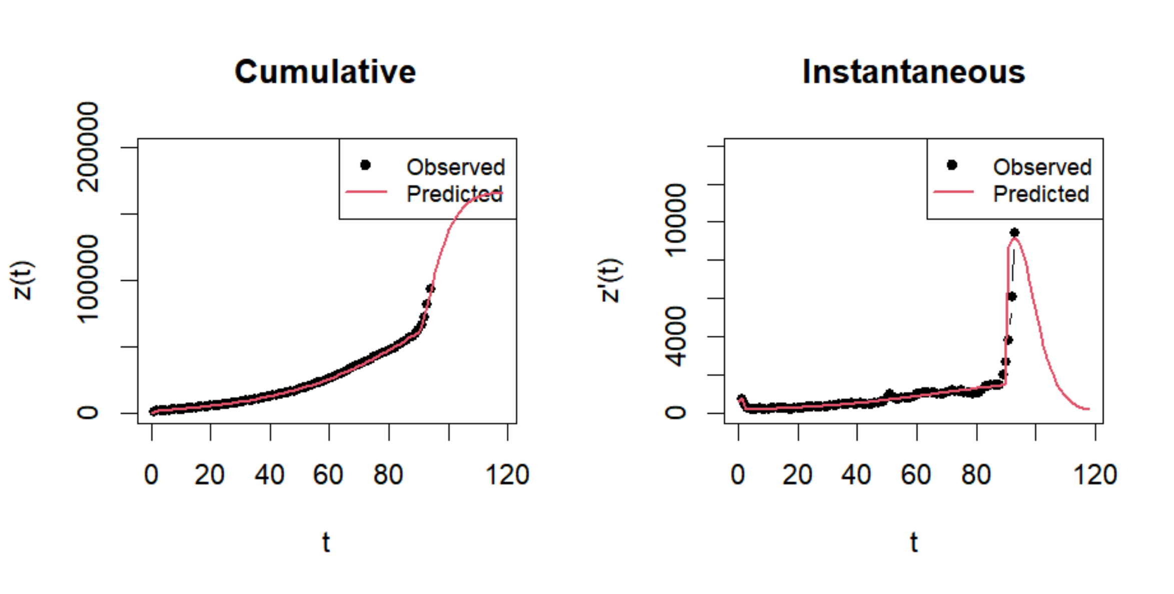
- We developed an **automatic** procedure to estimate **location** and **intensity** of the shocks.
- The procedure works according to the following steps:
  - Break the time series in **k equal parts** (we have chosen k = 10), t_1, t_2, …, t_k
  - Choose a list of values to use as **potential starting point** for the **memory** and **intensity** parameters.
    - Exponential shock: fit a GBM with exponential shock for **every combination of the memory and intensity parameters** using as starting points the last value of t_i for each i.
    - Rectangular shock: fit a GBM with exponential shock for **every intensity parameter** in the list using as starting points the last value of t_i for each i, but the last one, and as ending points the last value of t_i+1 for each i.
  - At each step **keep** the model if the **R-squared is max**.
  - Return the model with the **max R-squared**.

GBM FITTING - AUTOMATIC SHOCK ESTIMATION

- Using the automatic procedure we were able to fit a GBM for each series
- 75 models **improved significantly** with an **exponential** shock
- 69 models **improved significantly** with a **rectangular** shock

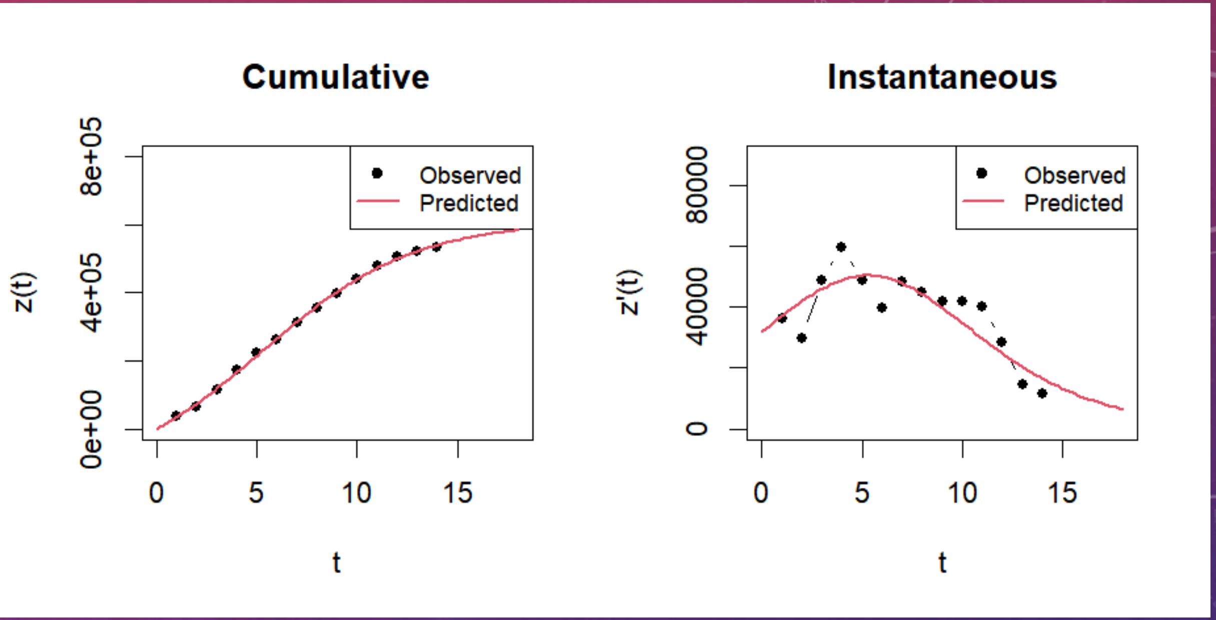
Standard Bass Model

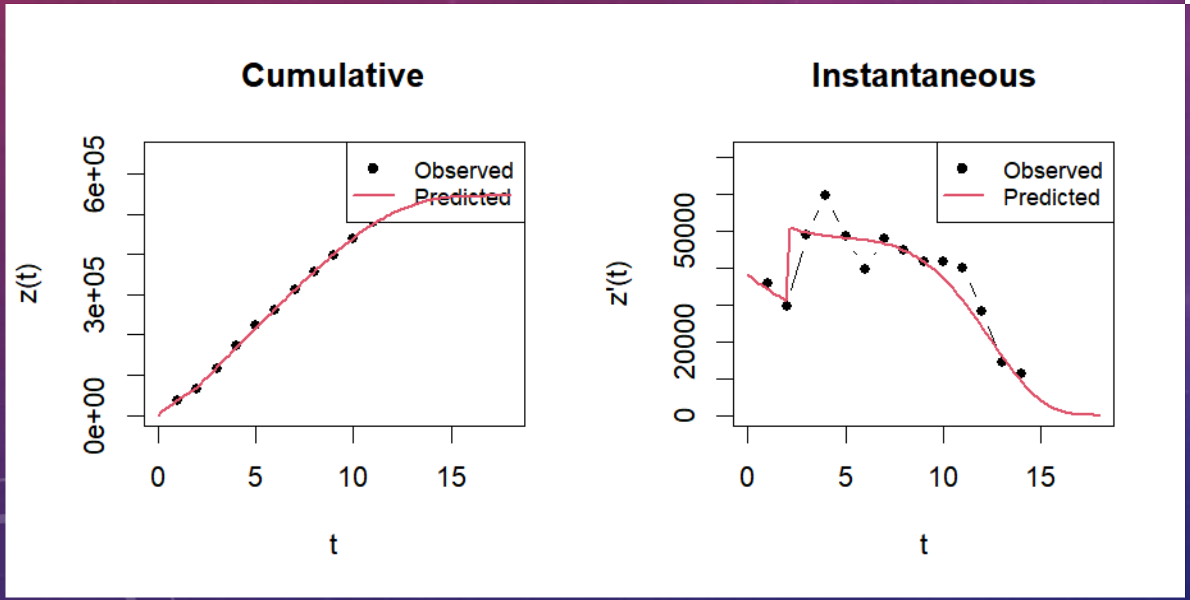




Generalized Bass Model
with rectangular shock

Standard Bass Model

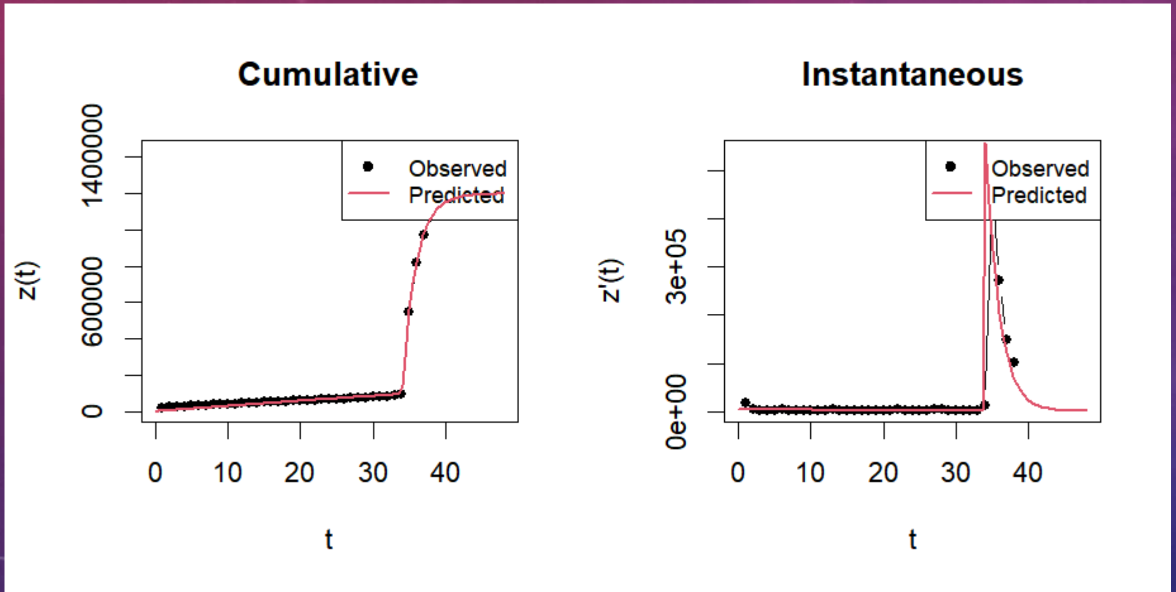


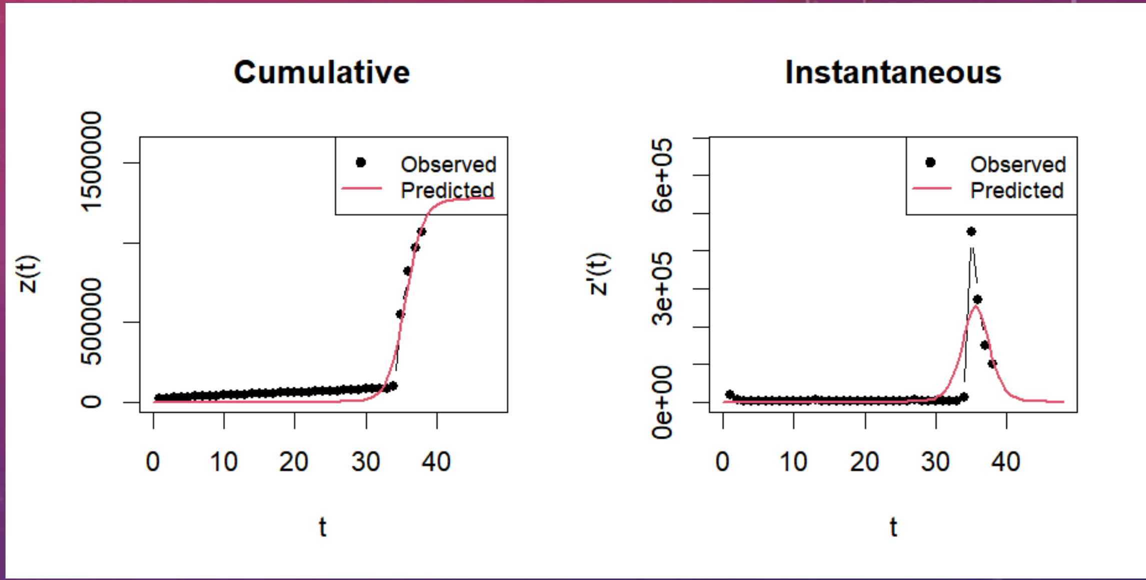


Generalized Bass Model
with exponential shock

A SPECIAL CASE: BETA RELEASE

Standard Bass Model





Generalized Bass Model with exponential shock

Baldur's Gate

CLASSIFICATION - RESULT

MODEL	Accuracy	Micro F1-score
Ordered Logistic	0.669	0.638
Multinomial Logistic Regression	0.719	0.696
Stepwise Multinomial Logistic Regression	0.686	0.640
Stepwise Multinomial Logistic Regression Reduced Dataset	0.75	0.751

CLASSIFICATION - STEPWISE

- The **stepwise** procedure on the reduced dataset selected these variables:
    - m_exp: **market potential** of the GBM with **exponential** shock
    - p_exp: **innovator** parameter of the GBM with **exponential** shock
    - q_exp: **imitator** parameter of the GBM with **exponential** shock
    - b1_exp: **memory** parameter of the GBM with **exponential** shock
    - c1_exp: **intensity** parameter of the GBM with **exponential** shock
    - m_rett: **market potential** of the GBM with **rectangular** shock
    - q_rett: **imitator** parameter of the GBM with **rectangular** shock
    - b1_rett: **intensity** parameter of the GBM with **rectangular** shock

COMPETITION MODEL

- URCDC - Unbalanced competition and regime change diachronic model
- Measures competition (or collaboration) between two videogames
- 2 stages:
  - initial one where only one videogame is in the market
  - second one when the second videogame enters the market allowing competition to rise.

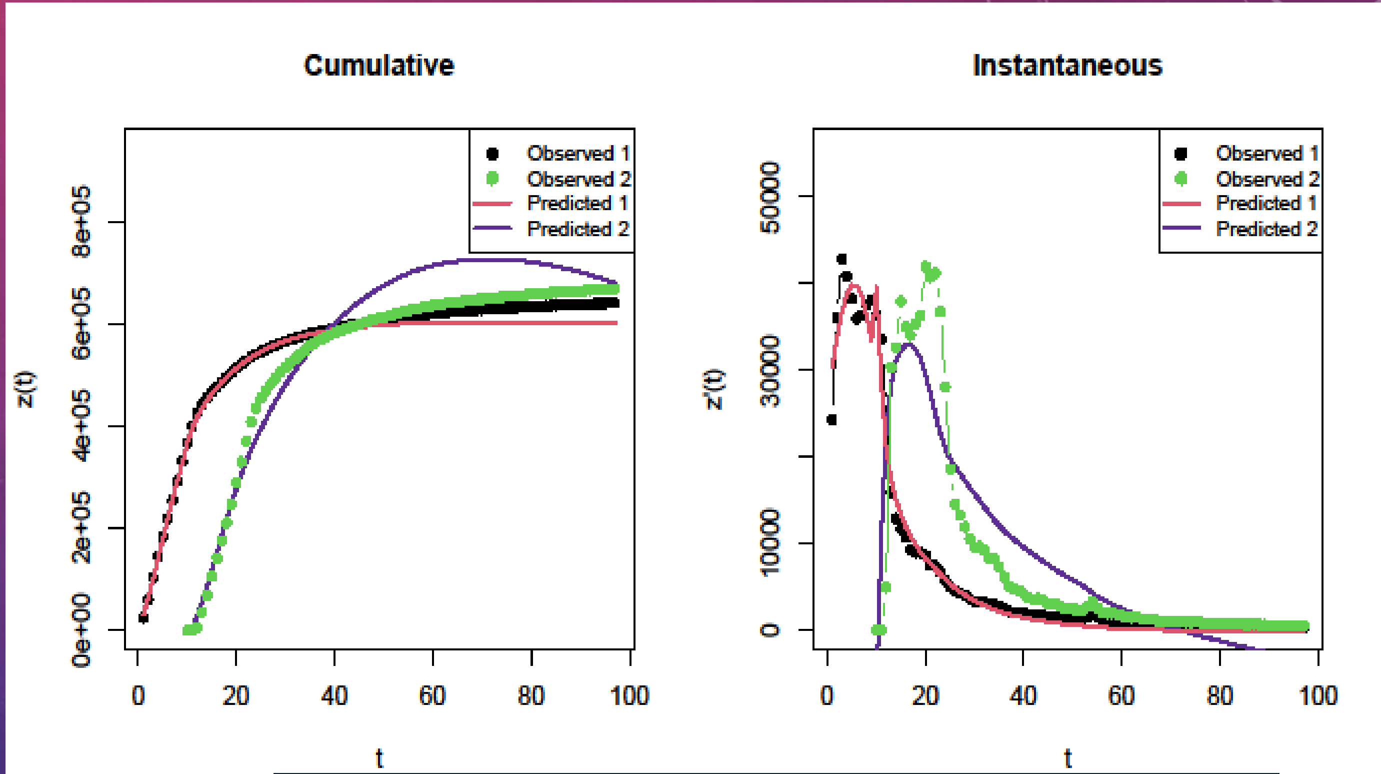
COMPETITION MODEL - EXAMPLE

Football Manager 2016

vs

Football Manager 2017

- Significant p-values
- Full collaboration



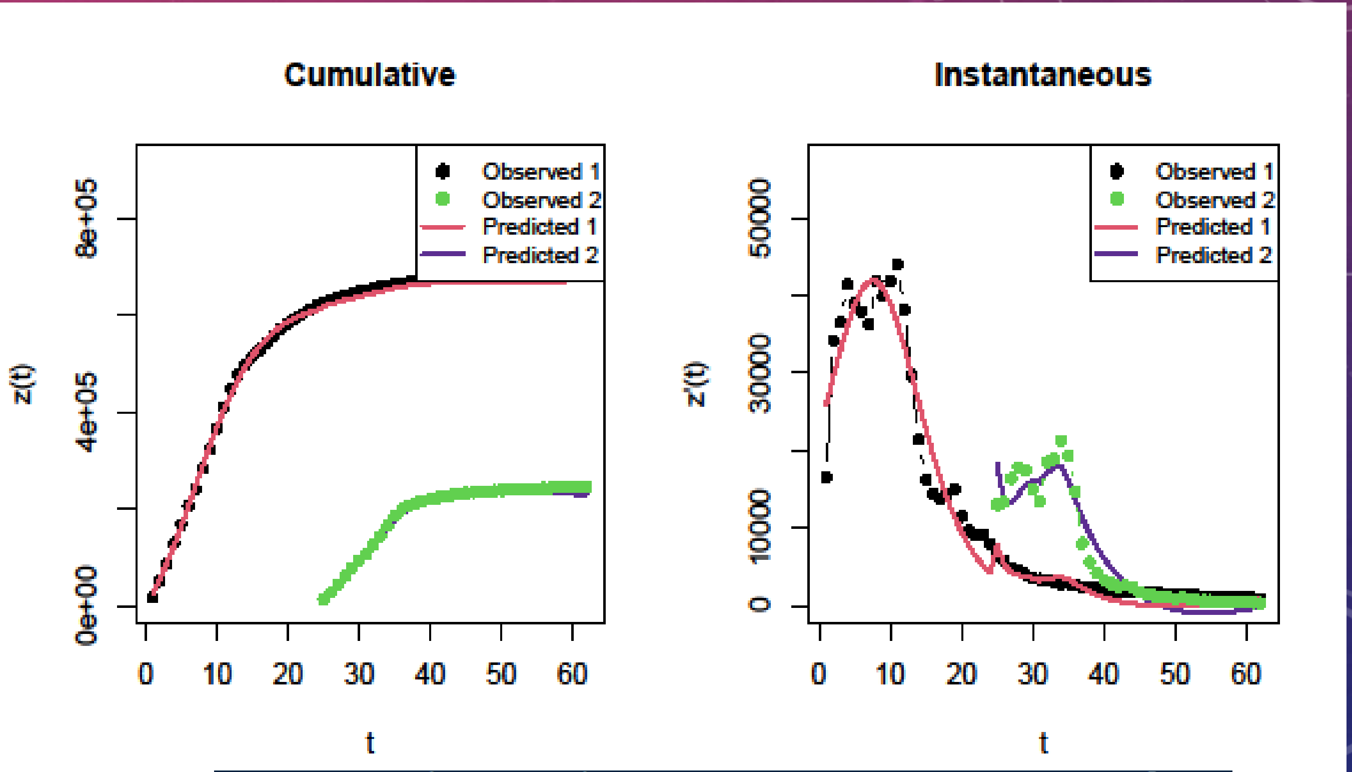
	Estimate	Std.Error	Lower	Upper	p-value	
ma	5.083659e+05	7.124553e+04	3.687273e+05	6.480046e+05	3.82e-04	***
p1a	5.636172e-02	4.929231e-03	4.670060e-02	6.602283e-02	2.68e-05	***
q1a	1.848968e-01	4.619673e-02	9.435293e-02	2.754408e-01	7.10e-03	**
mc	1.279233e+06	4.767830e+03	1.269889e+06	1.288578e+06	3.31e-224	***
p1c	1.622521e-01	2.788848e-02	1.075916e-01	2.169125e-01	2.92e-08	***
p2	-2.306168e-01	2.787716e-02	-2.852550e-01	-1.759785e-01	3.75e-14	***
q1c	9.471677e-02	3.027545e-02	3.537799e-02	1.540556e-01	2.07e-03	**
q2	-8.392113e-02	3.079321e-02	-1.442747e-01	-2.356755e-02	7.10e-03	**
delta	-5.091284e-01	1.148300e-01	-7.341910e-01	-2.840657e-01	1.66e-05	***
gamma	-8.607160e-01	1.153467e-01	-1.086791e+00	-6.346406e-01	4.28e-12	***

COMPETITION MODEL - EXAMPLE

Football Manager 2019

vs

FIFA 21



	Estimate	Std.Error	Lower	Upper	p-value	
ma	6.281640e+05	4.764184e+03	6.188264e+05	6.375016e+05	3.91e-32	***
p1a	3.854593e-02	1.284729e-03	3.602791e-02	4.106395e-02	9.95e-19	***
q1a	1.824127e-01	8.294948e-03	1.661549e-01	1.986705e-01	5.58e-16	***
mc	9.265112e+05	3.184807e+03	9.202691e+05	9.327533e+05	2.57e-108	***
p1c	1.969838e+00	6.441956e-01	7.072374e-01	3.232438e+00	3.17e-03	**
p2	5.593990e+00	7.743604e-01	4.076272e+00	7.111709e+00	5.22e-10	***
q1c	7.322717e-01	2.399588e-01	2.619612e-01	1.202582e+00	3.23e-03	**
q2	2.611297e+00	3.009398e-01	2.021466e+00	3.201128e+00	1.16e-12	***
delta	-3.620548e+00	1.190689e+00	-5.954254e+00	-1.286841e+00	3.33e-03	**
gamma	1.084351e+01	1.429528e+00	8.041685e+00	1.364533e+01	1.14e-10	***

- Significant p-values
- FIFA21 collaborates with FM19
- FM19 competes with FIFA21

COMPETITION MODEL - BATCH

- Fit the UCRCD model between every videogame and save the coefficients and their p-values
- Keep only those that have significant p-values

- Problems:
  - Cannot fit the model in some cases (hessian is negative)
  - The model also captures relationships between video games that are different and aim at different player bases

COMPETITION MODEL - EXAMPLE

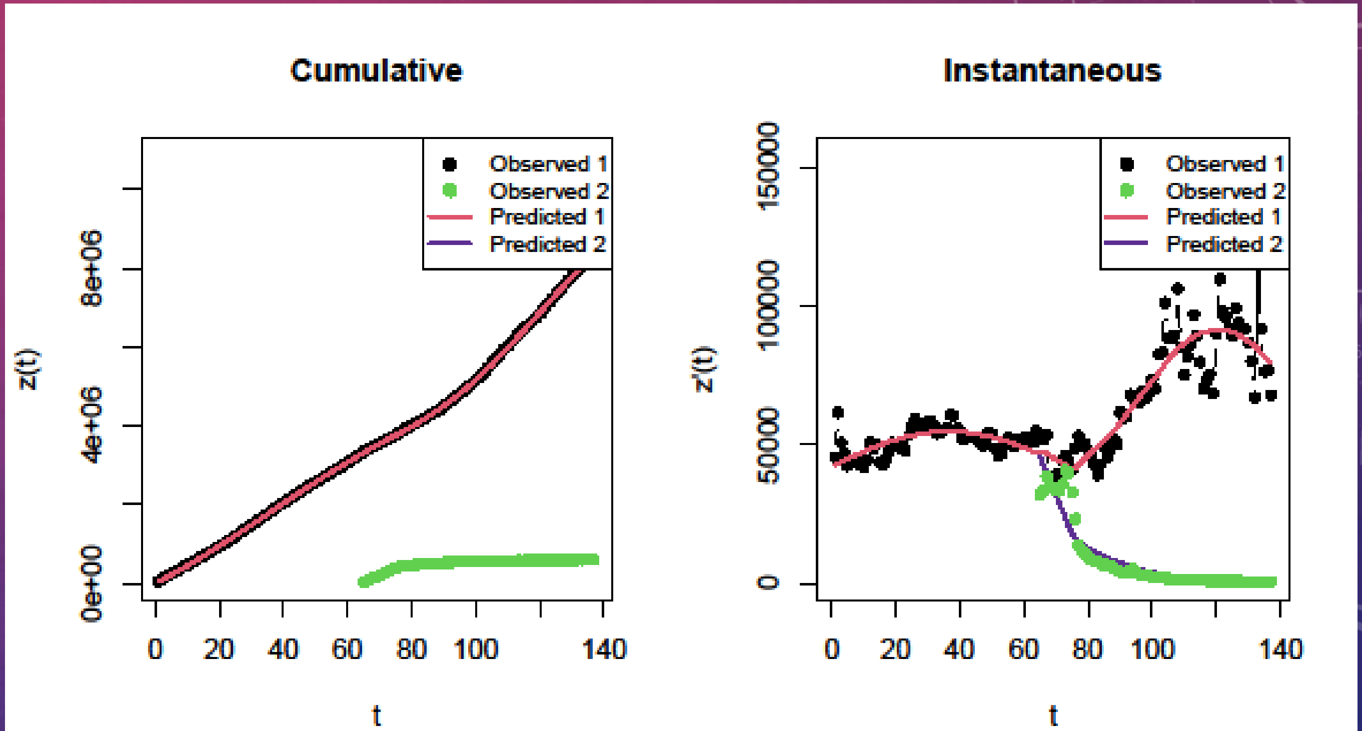
Team Fortress 2

Vs

Football Manager 2018

- Significant p-values
- Full competition

…but they do not aim at the same playerbase…



	Estimate	Std.Error	Lower	Upper	p-value	
ma	5.611264e+06	1.684227e+05	5.281162e+06	5.941367e+06	7.59e-41	***
p1a	7.550802e-03	1.683097e-04	7.220921e-03	7.880683e-03	2.02e-48	***
q1a	2.127694e-02	9.891449e-04	1.933825e-02	2.321563e-02	3.58e-30	***
mc	1.149958e+07	3.061244e+05	1.089959e+07	1.209957e+07	1.14e-74	***
p1c	-1.384241e-02	1.890559e-03	-1.754784e-02	-1.013698e-02	1.80e-11	***
p2	6.594348e-03	6.704309e-04	5.280328e-03	7.908369e-03	1.18e-17	***
q1c	-8.919641e-02	2.067254e-02	-1.297138e-01	-4.867899e-02	3.01e-05	***
q2	-9.836006e-02	1.578148e-02	-1.292912e-01	-6.742893e-02	5.14e-09	***
delta	1.578682e-01	2.573685e-02	1.074249e-01	2.083115e-01	8.40e-09	***
gamma	-9.589431e-02	1.739875e-02	-1.299952e-01	-6.179339e-02	1.67e-07	***

CONCLUSIONS

- How well do Bass Models fit video games' active users data?
  - Bass models don't really fit well with our data, however a good amount of games's data can be fitted well with GBM using premilinary estimated parameters chosen manually or GGM.
- Are Bass Models able to classify different types of video games?
  - Using GBM with good preliminary estimates we are able to get decent results in classification, however results are much worse when limiting ourselves to simpler models.
- Can an automatic procedure be developed?
  - We were able to automatize the process for our limited dataset, but it would be too computationally expensive to apply to a larger dataset.